Technical Manual and Data Sheet

GoForce 5500

Handheld
Graphics Processing Unit

PROVIDED UNDER NDA

DP-01433-001_v05

HANDHELD

This page left blank intentionally.

Contents

This page left intentionally blank

Revision History

Date	Revision	Description
2/04/05	Advance Release, v01	Creation of first version of this document.
2/11/05	Advance Release, v02	Added mechanical drawing to Specifications Chapter.
3/16/05	Advance Release, v03.000	Updated Ball Maps. Added chapter containing some of the micro class interface information. More to be provided in future revisions. Some of the signals in Chapter 2 had naming conventions that did not match the ball map. These were reconciled. Chapter 3: AC and DC timing characteristics: Where data is not yet available to substantiate expected timing, capacitance, etc. values; these were removed and replaced with TBDs. The tables containing the information have been left visible. The values will be added once data is made available through more modeling and testing.
3/30/05	Advance Release, v03.001	See Change bars throughout document. Details of changes: **Chapter 1**: Overview: Updated block diagram. Corrected signals to Host Controller: Was A[26:2], is A[25:2]; Was MHGP3 (Int#), is MD[2] (MHGP3), added core voltage source connections to embedded SRAM Removed descriptions from 64Bit graphics acceleration that did not belong there, changed VCO frequency range.
		Chapter 2: Functional Descriptions: All edited in general for clarity and accuracy. Overview: added note explaining that the microclass information will be added in a later revision. Edited section on Host Interface to update details. Figure 2.2: updated with DSP block Removed Figure 2.3: 8 MB External, etc. Memory configuration: redundant Section 2.4.1: Memory Controller Introduction: edited introduction for clarity 2.4.2: Added Memory client address map info, Interrupt and bad address info, and reset sequence. Updated information. 2D: added information to functional descriptions list, added section 2.5.3, 2D Engine Interfaces, and section 2.5.4, 2D Engine Clocks and Power Savings. Video Scaler: Added to section 2.6.1 list: source and destination information, performance capability of 3 Blit operations. 2.5.3: Edited for clarity and accuracy Section 2.6.2: Added information about Input Data Formats Video Input: Added to features list: more of the VI capabilities Section 2.7.3: Added to text on functions (Second two paragraphs are new.) Section 2.7.4.4: edited, no change to content ISP: section 2.8.1: Added last two bullets. Section 2.9.3: Interfaces: Added info about Host, VI, and VS inputs to EPP. Section 2.17.2: Clocks overview: Added notes about clock SC15 modules' clock sources, clock dividers, and power management. Added figures on PLL and per-module clock generation, text to go with these. Section 2.20.1: Added to I2S and AC'97 Codec features list.

Date	Revision	Description
		Chapter 3: Signals: Edited signals in this chapter to remove any disagreements between their names and descriptions and those listed on the Ball Maps in Chapter 4. Table 3.11: Discrepancy in signal names. Updated them to match the ball map notation (e.g. AVDDP1, not AVDDP0. Section 3.4: SC15 I/O Power Rails: New section: Contains 3.4, 3.4.1, 3.4.2. Source: B.G. and C.K. Section 3.6: Cleaned up VI Pin descriptions to match ball map. Removed inaccurate descriptions. Table 3.9: changed note on unused IO power to reflect current use and updated information. Table 3.4: Address range changed to [25:2], not [26:2]. Refclk1 definition changed: it's the second reference clock input MD2 definition changed to match MD1 MGHP6 (MD6) changed to MGHP6 (MD3). Same with table 3.15, 3.16, and 3.17 where applicable.
		Chapter 4: Specifications Added note at section 4.1. Explanations that values are based on simulations and will be changed when empirical data is available; some values remain listed as TBD. Table 4.1: added information on core v levels Table 4.2: new table: added IO power information. Table 4.3: Put theoretical values in table (from IO Designer group) Table 4.4: Same thing as table 4.2 Table 4.5: Same thing as table 4.2 Table 4.6 and Table 4.7: Made values TBD where applicable. 4.3.1: Clock: (AC Timings): made TBD 4.3.2: Reset: Now TBD 4.3.3: Power Sequencing: New section, includes 4.3.3.1, 4.3.3.2, 4.3.3.3, and 4.3.3.4. Source: B.G. (from NV Bugs) and C.K.
04/19/05	Advance Release, v03.002	**Chapter 1:** 1.3: Features: Was 32 voice polyphony at 22 kHz, updated to 64 voice polyphony.
		Chapter 2: 2.15.2: Added "SC15 may be put in standby mode so long as the clocks are driven (high or low) and not floated." Corrected typo in 2.16.2.1: design includes, not design include. 2.17.2: corrected typo to read "divide ratios" not "divide rations"
		Chapter 3: Table 3.8 and 3.10, Host Bus Interface Pin Descriptions: Description of REFCLK1 added note that it may be used to input an external clock source. Table 3.13: Display Controller Pins: Notes below table: "other purpose" changed to "other purposes." Table 3.14: Clock Pins: Re-worded descriptions of OSCFI and OSCFO for clarity, and to show that OSCFO may also be used as an input for an external clock source. Added note: Amplitude of input signal on OSCFO must be at least 0.8*HVDD. 3.9: JTAG Interface pins reference voltage was: AOCVDD. Is: HVDD. Added note to tie TRST_ to ground for normal operation. Removed what was figure 3.2 from section 3.4.1.1. Added DVD pin to table 3.8 and 3.10

Date	Revision	Description
		Chapter 4: Table 4.1: Added AVDDOSC, AVDDP1, and AVDDP2 to Voltage Rails table. Figure 4.1: Added AVDDOSC, AVDDP1, and AVDDP2 to Power on/off sequence diagram Section 4.3.3: Added note to initial verbiage that power sequence information presently only for non -use of DPD. Power sequence to include DPD in next revision. All Ball Maps: Section 4.3.4 Was: A2 = TRST_ and G14 = NC Changed to: A2 = NC and G14 = TRST_ Section 4.3.4.1 Changed all Ball to signal mapping information to reflect the A2/G14 change. 4.3.4: Ball Maps: R15 changed to DPD: a new signal. 4.3.4.1: Ball to signal mapping tables: R15 changed to DPD, a new signal.
		Chapter 5: Re-formatted figure titles. No real change to content or chapter.
		Chapter 6: Register Summary Table: New chapter. Lists all registers contained so far in chapter 7, with links to the title and the page number for each.
		Chapter 7: Almost all offset values changed due to the addition of new registers. Host1X_ASYNC_HCONFIG2_0: Auto-increment function for indirect addressing mode removed. HOST1X_ASYNC_ADRINCREG_0 added: auto-increment function. New Registers: HOST1X_ASYNC_RDWAITREG_0 HOST1X_ASYNC_RDWAITREG_0 HOST1X_ASYNC_MODEREG_0 HOST1X_ASYNC_OSCCONFIG_0 HOST1X_ASYNC_G3CCONFIG_0 HOST1X_ASYNC_DLYCTRL_0 HOST1X_ASYNC_LCDPADCAL1_0 HOST1X_ASYNC_VIPADCAL1_0 HOST1X_ASYNC_SDPADCAL1_0 HOST1X_ASYNC_AUDIOPADCAL1_0
6/29/05	Advance Release, v03.003	**Chapter 1:** Separated MPEG and JPEG Encoder Modules in Figure 1.1: they are separate modules. Changed RDY to RDY#/WAIT. Showed VECVDD as source to AVP block. Features: Added under Display Interface: the number of displays increases over time. Added under 32bit Host bus I/F: Synchronous Interface
		Chapter 2: Added Section 2.10.5: Display: Table 2.7: Pin Output Selection Options. Added text below table for some explanation as well. Added in reference to the register set for each module, under each module's section in the chapter. Section 2.5: Added Rotation in 2D Engine: fast rotation information; 2D Engine and VS, Slow Rotation sections VI: Added: 10Bit/clock Bayer Input information (in bullets): take in 12 bit, but only utilize the lower 10 bits. VI GPIO: VHSYNC and VVSYNC: Added information about max period for each. Added Slow rotation information for VI section. Display: Corrections made to Introduction (bulleted features) and Overview sections Table 2.5: Corrections made to table entries to match engineering specs JPEG Encoder: Corrected max frequency value Video Decoder: Overview: Added information in bullets about off loading VLD, and using VLD Clocks: Edited text for clarity, Added Table 2.10 with maximum clock frequencies per module. Added Figure 2.10, Clock Distribution

Date	Revision	Description
		Chapter 3: Table 3.3: Edited entries for clarity Table 3.13: Updated per engineering spec changes. (Change to number of programmable width pulses, etc.) Table 3.16: Changed signals MDQ[31:0], MDQM0, MDQM1, MDQM2, MDQM3 to match signal names on ball map. These became: MD[31:0], MDM0, MDM1, MDM2, MDM3.
		Chapter 4: Section 4.3.3.3: Added notes and power sequence diagrams on use of pin DPD with power sequencing/reset. Figure 4.6: Added SC15-XT Ball Map and ball-to-signal mapping table Section 4.4: Updated mechanical drawing for SC15-NM, -2M, -8M package. Added Figure 4.10 (SC15-8M 10 x 12 mm) package mechanical drawing Added Figure 4.11 (SC15-XT) package mechanical drawing
		Chapter 5: Added Figure 5.4: Channel Map Diagram
		Chapter 6: Updated Register Summary Table with registers in Chapter 7.
		Chapter 7: Added the registers for the following: Display registers for primary display and for secondary a and b displays Added registers to complete the register set documented for this product.
7/6/05	Preliminary Release, v04.000	**Chapter 4:** Added host interface timing diagrams and information.
09/21/05	Preliminary Release, v04.001	**Chapter 1:** Updated H.264 Video Codec information in Functional Descriptions (Overview)
		Chapter 2: Page 1: Updated note about technical manual changes to note that the SC15 is still in preproduction phases. Updated maximum operating frequencies to VI, ISP, JPEG Encoder, MPEG-4 Encoder, Video Decoder, JPEG Decoder, 3D Graphics Engine 2.2 Host Interface Replaced Host Interface Block diagram with an updated version. Added Command Processor, Command Buffer DMA, Read DMA FIFOs, and Module Register Reads sections. Added new diagram: Command DMA Operation Table 2.7: Display Interface: Parallel Host Interfaces Changed notation under 1 clock/pixel, 24bit to refer to R/G/B pixels instead of LD pin names. Tables 2.8 and 2.9: Corrected typos (e.g. if a series of pixels was G2, G2, G0; it was changed to G2, G1, G0 and so on as required.) Added note to MPEG Encoder and Decoder sections about VLD step in Decoding: Done by AVP under certain conditions (depends on bit rate - in which case the host CPU performs the AVP step before sending to SC15. Removed: H.264 VLC occurs in the AVP (DSP) in all cases. 2.6.4.2: Added max frequency information for VCLK 2.16: Power Management: Moved Module Power up/down sequence from Host I/F section to Power Management section. 2.17: Clocks: Added Host I/F and External Memory Controller maximum frequency information to Table 2.11 2.17.4 Frequency calculation: Frequency calculation information updated, Table 2.12 added to define PLL frequency parameters.

Date	Revision	Description
		Chapter 3: Table 3.3: SC15 Power Islands: Added Memory Controller (external) under AOCVDD. Removed note stating the power to SDIO must be on when HVDD is on. Table 3.4: Core Power and Ground Pins: Added DPS under pin description list for VECVDD. Table 3.8 and 3.10: Host Bus Interface Pins MHGP6 (MD3) Changed definition to match that of MD2, for consistency (on both Type A and Type C interfaces.) Table 3.15: Added definition for EMVREF pin - exists on ball map but was undefined elsewhere.
		Chapter 4: Added section 4.2, Temperature Specifications 4.3.4.1: Type A Host Interface: Table 4.8: Signal name in row 3 corrected to read BE_2. Description of 16bit host function for BE_2 and BE_3 had been reversed: corrected this. General: Twras had been drawn in the timing diagrams as being with respect to the rising edge of CS_ instead of falling edge. Corrected this as needed.
		Chapter 6: Simplified Register summary table
		Chapter 7: Register HOST1X_ASYNC_DSPCCONFIG_0: Definition for bits[23:22] and [21:20] changed: updated per engineering spec. Register HOST1X_ASYNC_GPIOODS_0: [15:14] and [12:12]: description changed: updated per engineering spec. Display Registers: Updated with additional descriptions to register fields: no change to functionality.
12/20/05	Preliminary Release, v04.002	**Entire document:** Core voltage range changed from 0.9 V - 1.32 V to 0.95 V - 1.32 V.
		Chapter 1: Number of triangles drawn per second changed from 2.8 million to 2.67 million (in verbiage, last paragraph, first page.) 1.3 Features Under “Audio Engine” and “Decode” added AAC [320 kbps] Under WMV and RealVideo Decoder modified text to read: WMV Decode: QVGA at 15 fps, [SP, Low] Real Video 9 decode: QCIF at 15fps Under “MPEG-4/H.263 Hardware Codec: Modified: MPEG-4 simple profile: was levels “0 to 3”, now states “levels 0 to 5” H.263 Profile 0 now states “Level 50” (this is an increase from 30.) Packaging and Voltage re-worded to show that the 8 MB memory package comes in a different size, and in one size only (10 x 12 mm)

Date	Revision	Description
		Chapter 2: Removed any references to the SC15-NM package option 2.1 Overview: Removed this section/paragraph. It did not contain information yet. 2.6 Video Input (VI) 2.6.1 Introduction Added the following bullet points: Under 10 bit/clock Bayer pattern input "Up to 79 MHz (at 1.0 V) and 105 MHz (at 1.2 V)" Under 8 bit/clock ITU-R BT.656 or TUV422/HS/VS input format: "Up to 79 MHz (50 Mpixels/sec) - at 1.0 V, 105 MHz at 1.2 V." 2.6.2 Overview In Table 2.3: the mapping of pins VD[11:0] to 10Bit Bayer inputs was changed from [VD11:2] mapping to Bayer[9:0] to show that the SC15 VI pins VD[9:0] actually map to the Bayer[9:0]. 2.6.4.2: Added max frequency information for VCLK 2.9.2: MPEG Decoder WMV9 Decode rate was QVGA @ 15 fps, is now QVGA @ 25 fps 2.11: MPEG-4 Encoder Under 2.12.1, Introduction: Changed simple profile L0 - L3 to L0 - L5. 2.12 Video Decoder 2.12.2 MPEG Decode Overview Changed bullet points about H.264 decoding: this should have been CODEC. Was: H.264 Decode (QCIF, QVGA, etc.) Is now: H.264 Codec (simultaneous encode and decode. The performance values for QCIF and QVGA remain the same, as they pertained to the codec function and not just the decode function. 2.16 Clocks Table 2.11: Added Internal Memory Controller parameters. Also added note stating that the maximum EMC frequency depends on the SDRAM. Changed ISP freq to match VI 2.19: I2S Added section 2.19.3 on the I2S timing.
		Chapter 3: 3.9 JTAG Interface Pins: Changed note to state that the JTAG interface pins are referenced to SDVDD. Added: The JTAG reset pin, TRST, is referenced to HVDD.
		Chapter 4: Added new section, 4.4.1, Clock, containing reference clock timing information. Added new section 4.4.2, Reset, with reset timing information. 4.4.3.1 and 4.4.3.2: Power-on Sequence: Changed value of T and changed T1 to be T, since they are now equal. T = 1 ms. Changed power-down time similary. (Both are T, both = 1 ms.) Added new section 4.4.5: Video Input Interface, with Video Parallel input clock timing Figure 4.72: Was SC15 -NM, -2M, and -8M. The drawing now only applies to SC15-NM and SC15-2M. Figure 4.75: 4 balls were missing from diagram - total number appeared to be 284, not 288. diagram corrected to show 288 balls (Diagram with locations 1 through 18 and A through V.) Removed all references, mechanical drawings, signal-to-ball mappings pertaining to the SC15-NM.

Date	Revision	Description
		Chapter 5: Removed Figure 5.3 (Memory map for obsolete -NM configuration)
		Chapter 7: Register DC_CMD_DISPLAY_COMMAND_0 was shown as R/W. It is now RO. Registers DC_COM_PM0_Control_0 and DC_COM_PM1_CONTROL_0 have the following note added to bits [1:0] description: Note: In non-continuous mode, shift clock and pixel clock run continuously, but line clock and frame clock run *only while a frame is being sent*.
1/27/06	Preliminary Release, v04.003	**Entire document: Changed product name from SC15 to GoForce 5500** Wireless Media Processor changed to Handheld Graphics Processor Unit (GPU) Removed references to SC11, it is not a product.
		Chapter 1 Features changed: Was 2.8 Milllion drawn triangles/second Changed to 2.67 million drawn triangles/second Was: 6 simultaneous textures Changed to 5 simultaneous textures Was: 8 surfaces Changed to 7 surfaces (color, Z, texture 1 through 5) Added note: “and lower” to XGA support in 3D mode feature point. H.264 Video Codec: added VLD on Host CPU for bit rates > 1 Mbps Removed references to D3DMobile, and references to 1.0 and 1.1 associated with OpenGL-ES. Changed WMV decode for QVGA to 25 fps (was 15 fps.) Video Input: Changed 3MP at *15 fps* camera preview via ITU-R 656-compliant 8bit interface to be at *10 fps*.
		Chapter 2 2.12.2: MPEG Decode Overview Was “H.264 Simple Profile at Level 3” Is “H.264 Simple Profile Levels 1 through 3” Added, under H.264 Codec: “VLD on Host CPU for bit rates > 1 Mbps” Under WMV9, removed “Simple Profile and Medium Level” It now reads “Decode” Added, after “320 x 240 (QVGA),25 fps,” : 384 kbps, comparable to simple profile medium level QCIF, 15 fps, 96 kbps, full spec for simple profile, low level 2.13 3D Graphics Engine Was: 8 surfaces: color, Z, and texture 1 through 6 Changed to: 7 surfaces: color, Z, and texture 1 through 5 Was: Multi-texture support (up to 6 simultaneous textures) Changed to: Multi-texture support (up to 5 simultaneous textures) Was: 2.8 million drawn triangles/sec Changed to: 2.67 million drawn triangles/sec Was: Standards supported: OpenGL ES 1.0 and 1.1 D3D Mobile Changed to: Supports OpenGL ES with NVIDIA Pixel Shading Extensions Removed phrase: “...future versions of OpenGL ES, and Microsoft’s Mobile D3D APIs”
		Chapter 4 Removed Figure 4.72 - SC11 Ball Map Removed Section 4.4.6.1.2 - SC11 Section

Date	Revision	Description
4/11/06	v05	**Entire Document:** Removed "Preliminary" classification in headers and on cover page. The GoForce 5500 has been released to production, and so the information in this document is no longer classified as preliminary.
		Chapter 2 2.17.3.1 Pull-up and Pull-down Resistors for CMD/DATA Lines: Changed first paragraph too read: The DAT3 line may be used to detect hot card insertion. To use the DAT3 line for detected hot-insertion, a pull-*down* resistor should be used on DAT3. (Normally a pull-*up* resistor is used with DAT3.) Previously implied the write protect could be used for hot card insertion detection, which was erroneous. 2.5.2: Removed ABGR from Overview of Input data formats (RGB 32bpp) - Not supported.
		Chapter 3 3.4 GoForce 5500 I/O Power Rails: Table 3.6: Removed "..and I/O power" from EMVDD description. The memory I/O power rail is not accessible externally. Table 3.8: Added note beneath explaining more about the GPIO pins and what interrupt signal is utilized when these are configured as interrupts. Table 3.13, Clock Pins: Relaxation Oscillator Resistor Pin Description showed the voltage source as AVDD. This was changed to AVDDOSC.
		Chapter 4 4.1: Table 4.2: GoForce 5500 I/O Voltage Rails: removed EMVDD from the voltage rails table. The I/O voltage rails to the EMVDD are not exposed externally. (The core voltage rails are.) Added Note: The core rail-to-rail tolerance is +/- 5% Added Table 4.3: GoForce 5500 Voltage Rails for Additional Memory Table 4.9: Note beneath table stated "...the input voltage should be the same as BVDD..." This should have referenced HVDD instead. BVDD was changed to HVDD. 4.2: Temperature Specifications changed from TBD to specified values 4.3.1 I/O Pin DC Specifications: V_{OL} and V_{OH} values were "TBD." These have been changed to .2*VDD and .8*VDD, respectively. 4.4.1 Clock: Note under Table 4.9: Changed BVDD to HVDD. 4.4.2: Reset: Figure 4.3: Reset Timing: Signal names in figure updated to accurately reflect current GoForce 5500 signal names. 4.4.3 GoForce 5500 Power Sequencing: First paragraph, clarified which sections actually discuss the power on/off sequence. 4.4.3.5 Grounding Considerations: Removed this section, it is obsolete. No replacement information needed. 4.4 AC Characteristics All Type A and Type C Indirect addressing: Auto-increment and Burst read and write timing diagrams: The Wait state had not been tri-stated. This was changed to a tri-state waveform in all such timing diagrams for Type A and Type C throughout the manual. One and Two-channel access for Indirect Addressing: Moved to follow the indirect timing diagrams for both Type A and Type C. Previously they followed the direct addressing timing diagrams. Table 4.49: Updated Min/Max values. 4.4.6: Display Controller Interface Timing: New section added.

Date	Revision	Description
		Chapter 7 HOST1X_ASYNC_ISCCONFIG_0[9]: Description for bit value = 0 stated "...latch data with falling edge" This was changed to "...latch data with rising edge" HOST1X_ASYNC_EMCCONFIG_0 [1] Description erroneously stated bit was used for slow clock enable. This was changed to fast clock enable. HOST1X_ASYNC_INTRMASK_0: Added note to register description that these "..are all level active, positive logic, input interrupts." Bit[6]: added, for clarity: (Enables the SD Module to receive an interrupt from the Host CPU.) HOST1X_SYNC_INTSTATUS_0: Added note for clarification: "Any of the GPIO pins, MHGP[6:4] and MHGP[2:0], when configured as interrupts, utilize the interupt signal INT_ on MHGP3 to output an interrupt from the GoForce 5500 to the Host CPU. MHGP3 contains all the interrupts on the MHGPx GPIOs, the interrupt from the SDIO Module, and the internal module interrupts. The bit in the register below tells which interrupts are pending, or not, on pin MHGP3." HOST1X_SYNC_INTCMASK_0: Added note to end of description above table: "...to interrupt the Host CPU." I2S_FSYNC_ENB_CNTRL_0: changed inactive to falling, and active to rising, in reference to sidnal edge. Former description was unclear. SD_SDGPIN_CONTROL_0 Bit Descriptions were previously names only, no state descriptions. State descriptions added. Removed 9bit SPI interface references from Ch 7. Register VI_PIN_INPUT_ENABLE_0, VI_PIN_OUTPUT_ENABLE_0, VI_PIN_INPUT_DATA_0, and VI_PIN_OUTPUT_DATA_0: Added notes to further explain the registers' functions. VI_PIN_OUTPUT_SELECT_0: Enhanced explanation of register functions: added information about dual-function pins. Added missing definitions to register description fields. Removed HOST1X_OBS and HOST1X_BFM registers - these should not have been documented and are not for use.
		Back Page Updated copyright notice (added two paragraphs) per NVIDIA Corporate Legal advice.

This page left intentionally blank.

Chapter 1 Overview

1.1 Introduction

The NVIDIA® GoForce 5500 Handheld Graphics Processing Unit (GPU) brings new features to your cell phone or handheld device that far exceed expectations. Now, in the device that you carry with you everywhere, you can

- capture pictures you can enlarge to poster-size (20 x 24 in/50 x 60 cm.)
- watch digital (DVB-H) television.
- capture or playback DVD-quality video.
- play awesome 3D games.
- video conference with the same quality as a dedicated, hard-wired system.
- playback WMV or RealVideo formats.
- capture MPEG-4 video at D1 resolution.
- IM or navigate semi-transparent menus while video plays in the background.
- listen to hours of music, regardless of format (e.g. MP3, AAC, WMA, RealAudio, and so on.)
- have support for a tablet-size PC display.

The addition of a dedicated hardware-based H.264/AVC codec makes it possible to watch DVB-H broadcasts any where, any time. By supporting full D1 resolution at an amazing 30 fps during playback, the quality of the videos you watch is comparable to a DVD.

You can also create your own high-quality H.264 movie, or host an H.264 video conference, right on your handheld device, with QVGA resolution at 15 fps.

Incorporating an ISP into GoForce 5500 has increased the supported camera resolution to an unprecedented 10 megapixels. This is higher than many professional digital still cameras, and will enable the production of exceptionally large prints with high levels of detail.

The built-in ISP supports Bayer data and performs operations such as auto-exposure and auto white-balance, as well as edge enhancement, gamma correction, and dead pixel detection. It also collects statistics for auto focus.

A programmable audio engine inside GoForce 5500 supports simultaneous encoding and decoding of AMR or AAC audio in conjunction with video conferencing, camcorder, or video playback -- all with virtually no MIPS requirements on the baseband or CPU.

Additional audio formats (including MP3, WMA, and RealAudio) are supported for listening to music at high-quality bit rates up to 320 kbps.

To support applications in a wide variety of different devices with screen sizes from small to large, the GoForce 5500 can support LCD sizes as large as XGA (1024x768), with over 16 million colors -- even in 3D mode!

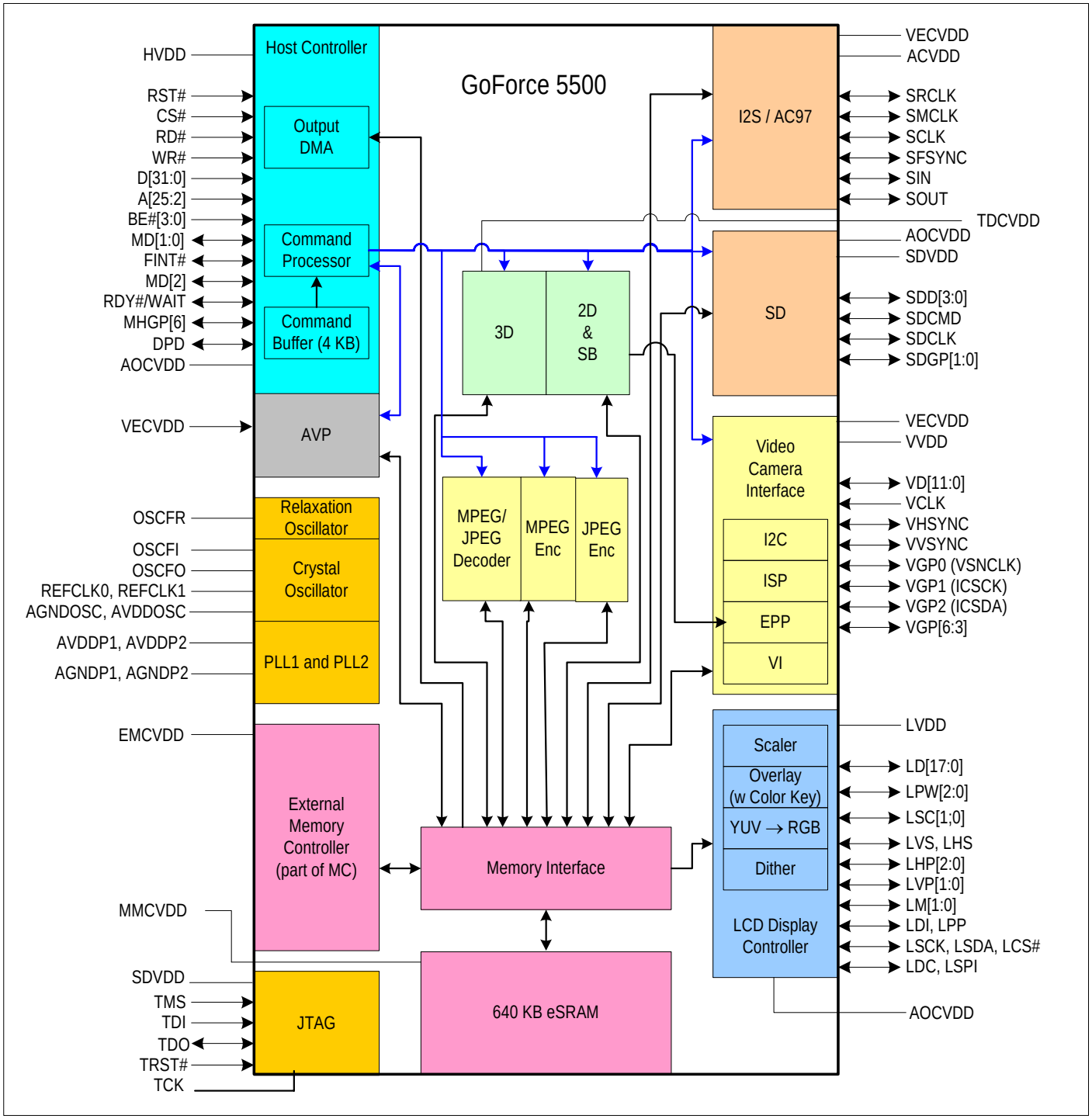
Enhanced alpha-blending modes make it possible to show semi-transparent menus over the top of video.

With a programmable pixel shader, 5 simultaneous textures, 2.67 million triangles drawn per second, and support for Open GL-ES and Java; the GoForce 5500 provides high quality, high performance 3D that exceeds the expectation of anything possible on a low-power handheld device.

1.2 Block Diagram

Figure 1.1 depicts a block diagram of the GoForce 5500. In the signal name descriptions, the symbol "#" denotes a negatively-asserted signal. Throughout this document such signals will be followed by the symbol "_" instead. The "#" used in the block diagram simply makes the signal-assertion level easier to see.

Figure 1.1: GoForce 5500 Block Diagram



1.3 Features

- **3D Graphics Engine V2.0**
  - OpenGL®ES compliant (plus NVIDIA extensions)
  - 200 million pixels/second 3D fill rate
  - 2.67 million drawn triangles/second
  - 128bit interface to internal memory
  - 32bit interface to stacked memory
  - Transform engine
  - 40bit color pipeline
  - 5 simultaneous textures
  - Signed overbright color
  - 7 surfaces (color, Z, texture 1through 5)
  - 16 4bit palettes or one 8bit palette
  - Programmable pixel shader
  - Bilinear/Trilinear texture filtering
  - Fixed and Floating point data
  - XGA [1024x768], and lower, support in 3D mode
  - Setup & pixel processing in hardware

- **Audio Engine**
  - Programmable Core
  - I^2S/AC'97 codec interface

  Decode

  - AMR NB [12.2kbps] and WB [23.5kbps]
  - AAC LC, HE-AAC (AAC+),
  - AAC+ Enhanced [128kbps]
  - MP3 [320 kbps]
  - AAC [320 kbps]
  - WMA, WAV & PCM
  - RealAudio
  - Bluetooth SBC

  Encode

  - AMR NB [12.2kbps] and WB [23.5kbps]
  - AAC LC [128kbps]
  - MP3 [320kbps]
  - Bluetooth SBC

  MIDI

  - Support for SP-MIDI, DLS, XMF
  - 64 voice polyphony at 22 kHz
  - Standard Sound Bank

  Audio Effects

  - Stereo Widening
  - Equalization
  - Noise Cancellation
  - Mixer
  - Acoustic Echo Cancellation
  - Environmental Effects

- **H.264 Video Codec**
  - H.264 Decode at 720x480 at 30fps or 720x576 at 25fps [D1 Resolution]
    VLD on Host CPU for bit rates > 1 Mbps
  - H.264 Encode
    QVGA at15fps [384kbps] (VLC on Host CPU)
    QCIF at 15 fps [128kbps] (VLC on GoForce 5500)
  - H.264 Codec QVGA at15fps [384kbps] (Simultaneous encode and decode)
    QVGA at15fps [384kbps] –
    (VLC on GoForce 5500 and VLD on Host CPU)
    QCIF at 15 fps [128kbps]
    (VLC and VLD both on GoForce 5500)

- **WMV and RealVideo Decoder**
  - WMV Decode: QVGA at 25 fps, [SP, Low]
  - Real Video 9 decode: QCIF at 15fps

- **MPEG-4 / H.263 Hardware Codec**
  - D1 encode or decode at 30fps
  - Full duplex D1 at 30fps
  - MPEG-4 Simple Profile, Levels 0 to 5
  - (ISO/IEC 14496-2)
  - H.263 Profile 0, Level 50
  - Back-end MPEG-4 video processing including hardware color-space conversion and image scaling
  - De-blocking and de-ringing filters to reduce the visibility of compression artifacts during playback
- **JPEG Hardware Codec**
  - 10MP encode or decode using ISO/IEC 10918 Baseline
  - Motion JPEG capture/playback
  - Low shutter lag image capture
  - Composite, framing, and overlay
  - Thumbnail support (store both image and thumbnail in same file)
  - Support Huffman decode for JPEG Programmable quantization table
  - Hardware DCT, RLE, Huffman encode
- **High Resolution Color Display**
  - Support for XGA [1024 x 768] LCD
  - Double-buffering support for VGA and lower resolution display
  - Fast switching between main/sub-LCD
  - Hardware support for sub-LCD display
  - Up to 24bpp panel support
- **SD/SDIO Host Controller**
  - 1-bit and 4-bit SD/SDIO
  - Support for storage or wireless cards
- **Image Signal Processor (ISP)**
  - Optical black calibration
  - “De-knee” compensation
  - Lens-shading (radial) compensation
  - Exposure compensation
  - White balance
  - Defective pixel correction
  - De-mosaicing & de-noising
  - Edge enhancement
  - Color correction to sRGB (or other programmable color spaces)
  - Gamma correction
  - Color conversion (to YUV)
  - Statistics gathering for Auto Exposure, Auto White Balance, and Auto Focus
- **Video Input (Bayer and YUV)**
  - 10MP Bayer camera module support via 10-bit RGGB Bayer Interface
  - 5MP Bayer at 15 fps
3MP at 10fps camera preview via
ITU-R BT656-compliant 8bit interface
  - 96MHz output to camera master clock
  - Horizontal scaling with horizontal averaging and low-pass filtering
  - Vertical averaging
  - I^2C for camera control & programming
  - YUV422 to RGB565 color-space conversion
  - Single- and double-buffering support
  - Double buffering synchronization with graphics controller
  - Image/Video Rotation
- **64Bit 2D Graphics Acceleration**
  - BitBLT with 256 3-operand raster ops
  - Video scaling with range of 8x expansion to 1/64th contraction
  - Mono and solid pattern
  - Mono-to-color expansion
  - Mono source/pattern transparency
  - Destination read/write color transparency
  - All-angle Bresenham line draw
  - Rectangle fill
  - Image/Video Rotation
  - Alpha Blending

- **Display Interface**
  - 16.8 million colors in 24bpp mode
  - 262k colors in 18bpp mode (dithered)
  - Direct interface to Host/CPU-style LCD drivers
  - Built-in timing generator
  - Color TFT at 9, 12, 16, 18, 24-bit/clock
  - Partial pixel-per-clock mode
  - CPU, RGB, Serial, M-CMADS, AMLCD, and LTPS support
  - Support for over 80 popular displays
    This number increases over time, some cases may be limited by software capabilities
- **Graphics Controller**
  - Alpha Blending
  - 16 to 24-bpp color expansion
  - Color Space Conversion (YUV to RGB)
  - Hardware rotation (90 °, 180°, 270°)
  - Flip and mirror
  - Partial display support (any size/position)
  - Triple 6bit look-up-table
  - Overlay support
  - Encode predefined region of display
- **Integrated 640KB 128bit Wide SRAM**
  - 640KB of 128bit wide on-board memory for frame, video, and transactional buffers
- **32Bit Flexible Host Bus Interface**
  - Indirect and direct addressing
  - 16bit or 32bit asynchronous interface for baseband processors (ARM based)
  - 16bit or 32bit synchronous interface
  - Burst mode support
  - Fixed and variable latency host bus
  - Automatic address incrementing for indirect addressing
  - Programmable interrupt
- **Clock Options**
  - On-chip oscillator for 2 to 13MHz crystal
  - Digital bypass mode for external clock sources (e.g. baseband or CPU)
  - Low-power relaxation oscillator
  - Two on-chip PLLs with independent VCOs (range of 50 MHz to 664 MHz)
- **NVIDIA NPower Power Management**
  - Fully-static CMOS technology
  - Low-power 90nm process
  - Individual module enables
  - Automatic shut-off of unused pipeline stages
- **Packaging and Voltage**
  - 284pin BGA, 0.50mm ball spacing, 1.4mm height
  - 10 x 10 mm, (2 MB) or 10 x 12 mm (8 MB)
  - Support for maximum of 32MB external memory (288pin BGA)
  - JTAG boundary scan & BIST
  - 0.95 to 1.32V core, 1.71V to 3.30V I/O

This page left blank intentionally.

Chapter 2 Functional Descriptions

The GoForce 5500 Handheld GPU contains all the modules listed below and this chapter describes each of them. To go to a section to read about a specific module, simply select the module name from this list, and use your mouse to left-click on it.

- Host Interface
- Audio Video Processor (AVP)
- Memory Controller
- 2D Engine
- Video Scaler
- Video Input (VI)
- Image Signal Processor (ISP)
- Encoder Pre-processor (EPP)
- Display Controller
- JPEG Encoder
- MPEG-4 Encoder
- Video Decoder
- 3D Graphics Engine
- Embedded Memory
- Power Management
- Clocks
- SDIO (Secure Digital IO) Interface Host
- Serial Peripheral Bus (SPB)
- I2S and AC'97 Codec Interface

Note: All modules have a module enable/disable function.

Note: Please contact your NVIDIA representative with any questions you have about the technical content of this document, and to be sure that you have the most current GoForce 5500 information available.

2.1 Host Interface

2.1.1 Introduction

The GoForce 5500 Host Interface Module functions as the external host interface for the entire GoForce 5500. The Host Interface is the only asynchronous functional block in the GoForce 5500.

The GoForce 5500 Host Interface Module introduces several new features to deliver increased driver performance through a more efficient host interface:

- A class-based programming API
- Command Buffer DMA
- Multiple channels to the Command FIFO
- Per-module context switching.
- Multiple read DMA ports that can be assigned to any client on the GoForce 5500

These new features The Host Interface registers are listed in *Chapter 7,* "GoForce 5500 Micro-classes" in *Section 7.1,* "Host Registers".

2.1.2 Overview

The GoForce 5500 does not have a chip-specific API. Not having such an API allows hardware flexibility and software driver compatibility. Instead, the GoForce 5500 features a Micro-class Interface with a smart-command processor. The programming interface is based on writing to offsets within a class implementing a function, rather than to specific register offsets and formats.

Features of the GoForce 5500 Host Interface include

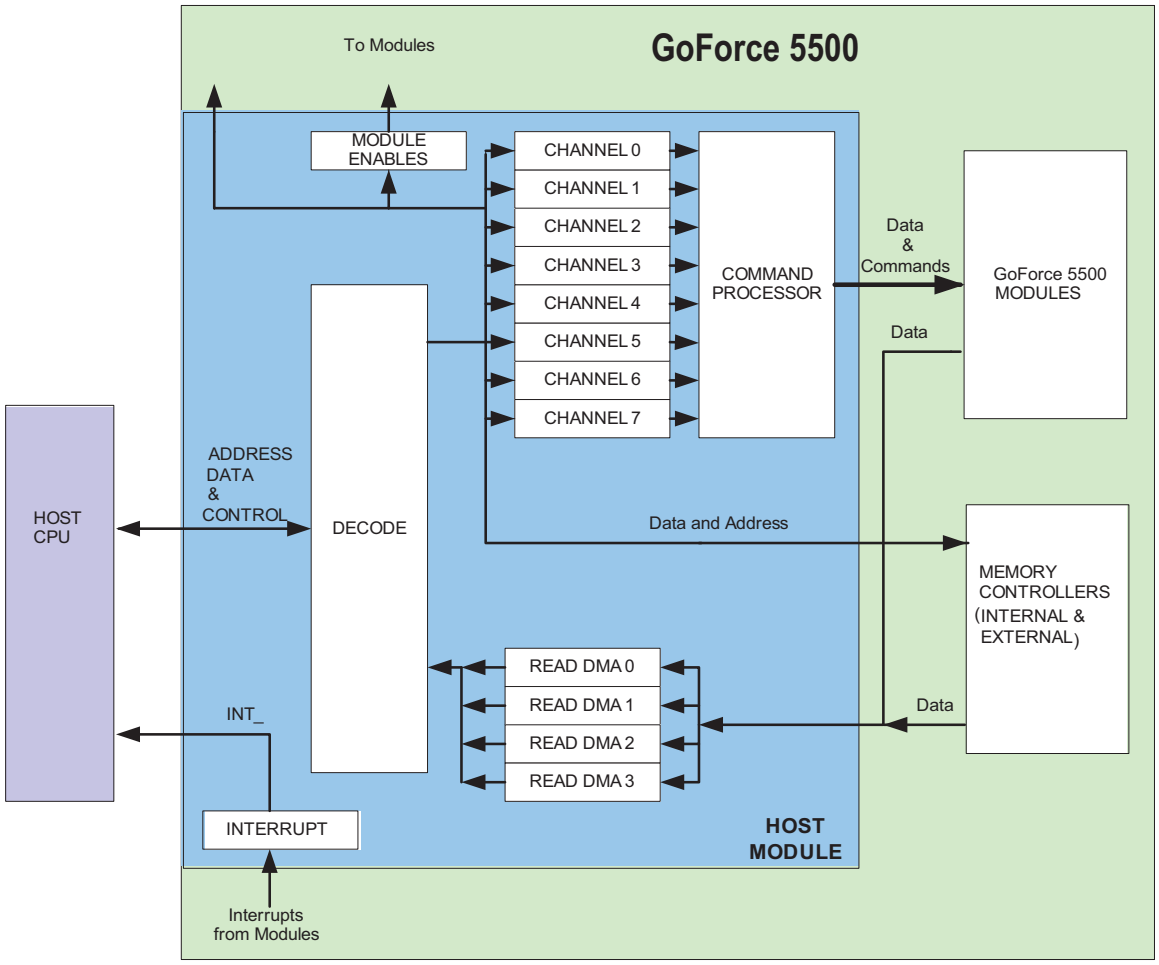
- Asynchronous interfacing to Type A and Type C Host CPUs with
  - 16bit bus width
  - 32bit bus width
- Fixed-cycle interface
  - Does not use the RDY pin (no handshake)
- Handshake interface (uses RDY pin)
- Micro-class interface with smart command processor
  - Eight low-priority channels with deep command buffer (512 x 32bit total) to minimize polling of status (with the no-handshake interface)
    FIFO size per channel is 2048 bits
  - High-priority channel for fast register reads and writes
  - Multiple context channels (up to eight)
  - Flow control to the display double-buffer switch, or to switch between the primary and secondary display
- Indirect addressing
  - Capability for direct linear addressing access to display and external memory, at the same time the host CPU utilizes indirect addressing to the GoForce 5500
- Interrupt controller with active-low interrupt pin
- Write-byte enables
- Read and write byte-swapping option for 16-bit and 32-bit interfaces
  - 4 byte-swap modes
    No swap
    16-bit byte swap
    32-bit byte swap
    32-bit word swap
    Separate Byte Swapping for header and data for the MPEG Encoder
- Separate read and write data swapping for registers/frame buffer addresses
- Increased driver performance (through reduced FIFO polling and contention) using
  - Class-based programming API

    - Command Buffer DMA
    - Multiple channels; each has its own Command FIFO
    - Per-module Context Switching

- Fast output (read) DMA used for
    - Reading MPEG-4 and JPEG encoder outputs from internal memory or from external memory
    - Reading captured video frames from internal or external memory
        - YUV
        - RGB
        - Raw data (Bayer format)
    - Reading a rectangular area from internal or external memory (screen to memory BLT)
    - Utilizing byte swap options
- Optional 2 MB to 8MB external module SDR or DDR memory (either two x16 or one x32.)
- Host Interface Clock
    - 117 MHz maximum operating frequency (at 1.0 V)
    - 175 MHz maximum operating frequency (at 1.2 V)

As was stated before, the Host Interface Module is asynchronous but the rest of the GoForce 5500 is not. This means, for example, the Host CPU reads and writes to internal memory are initiated asynchronously by the Host Interface Module but are carried out synchronously in the Internal Memory Module. This approach is used with other FIFO writes, such as the Video Interface YUV-FIFO writes and the Graphics Engine sequenced frame-buffer writes. (The latter are handled by the front-end Command FIFO.)

Figure 2.1: Host Interface Block Diagram



2.1.3 Host Interface Functional Blocks

2.1.3.1 Interrupt / Status Control

The Host Interface supports one level of interrupt handling, while individual modules may support additional levels. The Host Interface provides an interrupt status, an interrupt mask, and an interrupt clear function. The combined value of the status bits goes to the host CPU via one of the GPIO pins.

Each module can trigger an interrupt to the Host CPU.

2.1.3.2 Module Enables

The Host Interface controls the module enable functions to the on-chip modules. Disabling a module soft-resets that module. The module enable function is combined with chip reset in the Host Interface and sent as a single, asynchronous reset to the module. This reset should not be used to gate anything else, such as a module clock, inside the module. There are separate controls for that in the Host Interface.

Module resets (all are active low)

- Host Interface reset
- 2D reset
- 3D reset
- display reset
- memory controller reset
- video camera interface reset
- video post processor reset
- SD reset
- ISP reset
- I2S reset
- SPB Reset
- MPEG encoder reset
- AVP reset
- Memory Controller (Internal and External) resets
- JPEG encoder reset
- MPEG/JPEG decoder reset

2.1.3.3 Command Processor

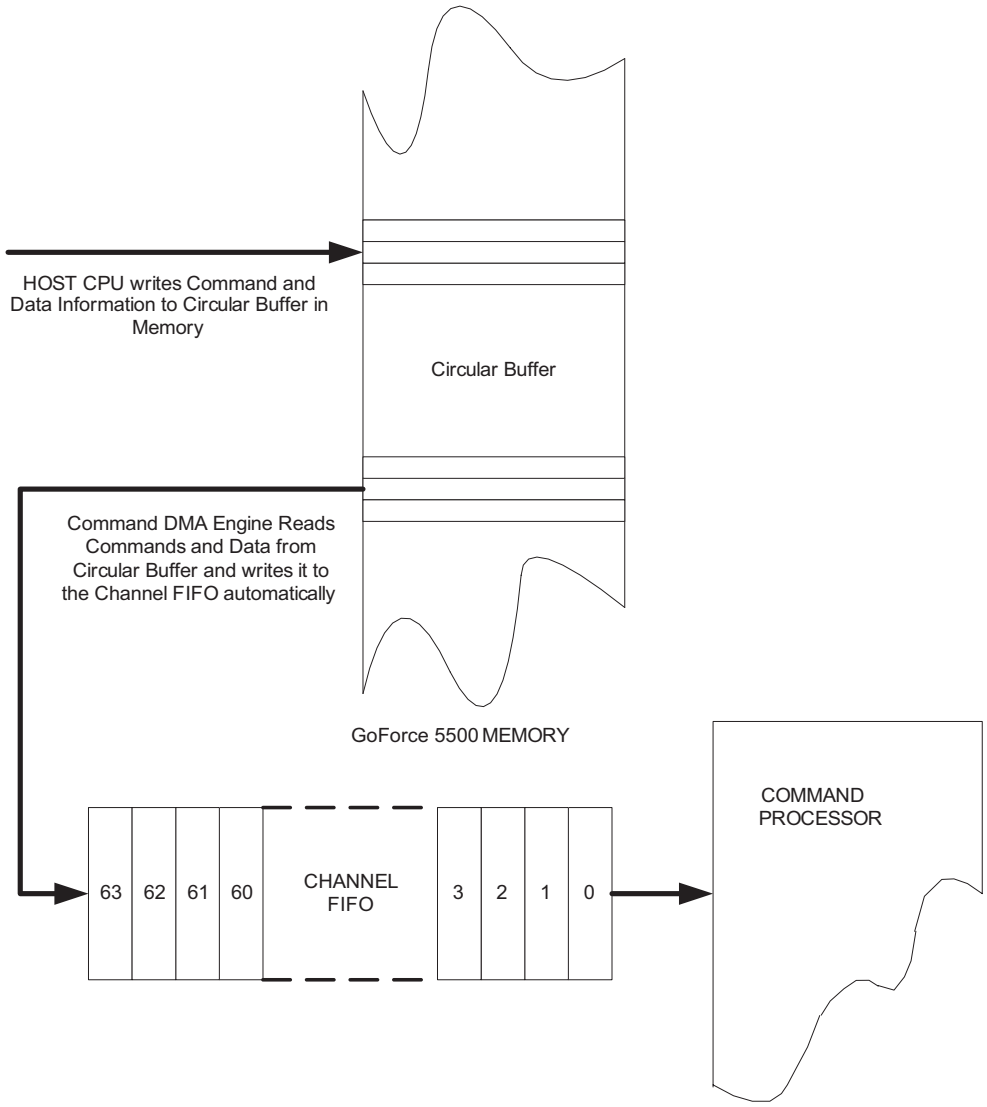
The GoForce 5500 does not have a chip-specific API. Instead, the GoForce 5500 features a Micro-class Interface with a smart-command processor. The programming interface is based on writing to offsets within each class that implements a function, rather than to specific register offsets and formats. Not having a chip-specific API allows hardware flexibility and software driver compatibility with future GoForce products.

The Command Processor is fed with commands and data from up to eight channel FIFOs. The command processor decodes the commands and passes them, along with corresponding data, to the appropriate GoForce 5500 module. Having multiple channels feeding the command processor enables multiple contexts of the GoForce 5500 hardware to be instantiated from a software viewpoint. This architecture allows minimal polling through use of multiple deep command FIFOs. Each command FIFO is 64 x 32bits (64 x 1 dword) deep. In addition to these deep FIFOs, a Command FIFO DMA mechanism can be used to further minimize polling and increase driver performance.

2.1.3.4 Command Buffer DMA

The Command Buffer DMA ensures the channel FIFOs are always full of data and commands. In a typical configuration, the Host CPU must read status registers to verify sufficient free space exists inside a FIFO before writing commands and data to it. When not enough room exists for a complete write, the host CPU must hold off the write until sufficient room becomes available. This process leads to an inefficient driver. The Command Buffer DMA overcomes such a drawback by establishing a circular buffer for each channel within GoForce 5500's internal memory. Commands and Data are then written into this circular buffer, rather than directly into the channel FIFO. There is an internal DMA mechanism that reads commands and data from this circular buffer, and writes them to the channel FIFO. The DMA works to keep the channel FIFO always full, if possible.

Figure 2.2: Command DMA Operation



2.1.3.5 Read DMA FIFOs

The GoForce 5500 Host Interface Module has four Read DMA FIFOs, which can be assigned when the Host CPU is reading large amounts of data from the GoForce 5500. These allow *internal* DMA transfers from the GoForce 5500 memory to the Read DMA FIFO. The Host CPU can then read the FIFO at its own rate. The Read DMA engine keeps the Read FIFO as full as possible. Such uses of the Read FIFO DMA ports include

- Reading MPEG-4, H.264, and JPEG encoder outputs from internal memory or from external memory
- Reading captured video frames from internal or external memory in YUV or RGB
- Reading Camera data (YUV or Bayer format)
- Reading a rectangular area from internal or external memory (screen to memory BLT)

2.1.3.6 Module Register Reads

The GoForce 5500 Module Registers can be read back through a set of read registers associated with each channel. A Module Read command is passed down a channel. When the command is received, the module places the register's data into the READ register associated with that channel. This is not shown on the Host diagram for clarity.

2.1.4 Host Bus Interfaces

The Host Interface has a 32-bit wide data bus that supports 16-bit and 32-bit interfaces in direct and indirect addressing modes. The addressing modes are selected through the mode pins MD[2:0]. (These can be configured as GPIOs after reset.) Refer to *Chapter 3* for the correct mode pin settings.

The Host Interface is optimized for 32-bit and 16-bit bus widths to support 3D applications requirements and video capture application requirements.

To minimize the number of pins, the Host Interface can be operated utilizing the indirect addressing mode. Even in such cases, the option for direct linear addressing reads and writes to the display memory can still be used. In other words, enabling the direct linear addressing access to the display memory means the register access and command writes to various internal engines can still be carried out in indirect addressing mode, if required.

Direct addressing means both the address and data busses are used for host transactions. With indirect addressing the address and data signals for host transactions are multiplexed on the data bus. Indirect addressing uses one address pin to indicate an Address Cycle (A[2] = 1) or a Data Cycle (A[2] = 0), saving twenty-three address pins (when compared to direct addressing) for a 32-bit bus.

Both of the following Host Bus Interface types support direct and indirect addressing modes:

- Type-A Host Bus Interface:
  - 16 bit and 32-bit wide data bus.
  - Separate write enable and read enable signals, both active low.
  - Byte writes controlled through BEn[3:0].

- Type-C Host Bus Interface:
  - 16-bit and 32-bit wide data bus
  - One control signal indicates a write or a read cycle
    - active low for read
    - active high for write
  - Separate write enable signals for each byte.

The Host interface supports three access methods; fixed cycle, ready handshake, and wait handshake accesses. Handshaking is supported through the RDY_ signal. The GoForce 5500 can be configured to support either ready handshake or wait handshake. This method is useful if the host CPU wants to access the GoForce 5500 without polling to avoid overflow. Configure the access methods using mode pins MD[2:0].

The GoForce 5500 has an additional mode pin to indicate synchronous mode, and an Interrupt pin to pass interrupts to the host CPU. It is driven all the time and gets asserted as long as a pre-programmed event happens. The interrupt sources are configurable.

2.1.4.1 Indirect Addressing Mode

Configuring the strap mode option MD[2] puts the GoForce 5500 into indirect addressing mode. See Table 2.1 for a brief description of the address cycles. Indirect addressing requires a single address pin. (Direct addressing requires a 25-bit address bus with a 16-bit host interface.) The A[2] pin is the only address pin required for the indirect addressing mode. It is used as an address/data select pin (i.e., the AD/DSEL pin). It defines whether an on-going CPU Host cycle is an address cycle (where the data bus carries address-related information) or data cycle (where the data bus only carries data).

The GoForce 5500 supports auto-address incrementing. Separate enable controls for write cycles and read cycles allow separate read address and write address controls in indirect addressing mode resulting in increased performance. (The GoForce 5500 holds the write address while the Host CPU changes the read address, or while the Host CPU switches from a write cycle to a read cycle.) There is no specific burst-length limitation for address incrementing as long as the incremented address stays in the GoForce 5500 address range.

Table 2.1 below shows the mapping of Data bits used as address bits and control bits in the 32-bit wide host bus interface.

Table 2.1: 32-Bit Host Indirect Addressing Mode Mapping

Access	A[2] (Address/Data Select)	D[0] (Register Select)	Data/Address Bits
Data Cycle	0	D[0]	Data [31:1]
Address Cycle:			
Register	1	1	D[31:17] = X D[18:1] = Addr[17:2]
Memory	1	0	Address [25:2]

Table 2.2: 16-Bit Host Indirect Addressing Mode Mapping

Access	A[2] (Address/Data Select)	D[1] (Upper-Address)	D[0] (Register Select)	Data/Address Bits
Data Cycle	0	D[1]	D[0]	Data [15:2]
Address Cycle:				
Register 1	1	0	1	Addr[14:1]
Register 2	1	1	1	D[15:5] = 11'bX, D[4:2] = Addr[17:15]
Memory 1	1	0	0	Address [14:1]
Memory 2	1	1	0	D[15:13] = 3'bX, D[12:2] = Addr[25:15]

Note: For non-sequential register accesses two Host CPU clock cycles are required for 16bit register accesses when the GoForce 5500 utilizes indirect addressing. For memory accesses, the GoForce 5500 needs a 25bit address. The memory address for indirect addressing can be transferred to the GoForce 5500 either with one host clock cycle (if there is no change on the upper address bits [25:15]), or it can be transferred with two host clock cycles (if both lower and upper address bits are different from the ones specified previously). Address incrementing removes the need of having address cycles for sequential accesses.

The GoForce 5500 supports burst writing. With indirect addressing mode, as long as the address-incrementing mode is enabled a new address cycle for sequential read or write accesses is not required.

2.1.4.2 Direct Linear Addressing to Display Memory

Note that A[25] = 1 is the address space for direct linear addressing of external memory.

2.1.5 GoForce 5500 Address Map

The address map was designed with the following goals in mind:

- Ability for the memory footprint to scale down when the chip contains less than maximum memory
- Host registers (including the chip ID and memory population information) must be at a fixed place in memory that is accessible the same way regardless of the memory footprint.

2.2 Audio Video Processor (AVP)

2.2.1 Introduction

The GoForce 5500's AVP supports audio and video processing, and has the following features:

- 32-bit processor with 8 KB Instruction (I) cache and 8 KB Data (D) cache
- Audio codecs: AMR, AAC, AC3, MP3, WMA (WMA is decode only)
- MPEG/JPEG bit-stream Variable Length Decode (VLD), and buffer and display management
- Image Signal Processor (ISP) control for auto exposure, auto white-balance, auto focus, and flash control
- Frame-based rate control for MPEG encoder
- Intra prediction and VLC for MPEG encoder
- VLC for H.264 up to 128 kbps

2.2.2 Overview

Using an AVP off loads work from the Host CPU. The GoForce 5500's AVP reduces the Host CPU's work load, and interfaces to the Memory Controller (IMC) of the GoForce 5500.

The Host CPU performs a write operation to all the registers associated with the AVP, and to the registers related to the Clock and Reset Generation module via the Host Interface. The AVP supports audio and video processing, and utilizes the following modules when communicating with the Host Interface:

- Command FIFO
- Clock and Reset Generation Module
- DSP-related Interrupt Controller
- Raise and Wait Vector Generation
- PIF Unit
- MC
- EMC

The AVP uses the Host Interface when communicating with these blocks:

- MPEG-4 and JPEG Decode Modules
- AC'97 I2S Interface Module

2.3 Memory Controller

The registers for the Memory Controller are found in *Chapter 7*, "GoForce 5500 Micro-classes" in Section *7.9*, "EMC Registers" and Section *7.5*, "MC Registers".

2.3.1 Introduction

The GoForce 5500 Memory Controller consists of two parts; the External Memory Controller (EMC) and the Internal Memory Controller (MC). The GoForce 5500 comes with 640KB SRAM and 2MB or 8MB additional memory.

The MC manages the scheduling logic for both the SRAM and DRAM; as well as the SRAM access logic, and features the following:

- 640KB SRAM
- Separate 128bit buses for read and write
- Memory interface source clock driven from
  - relaxation oscillator clock
  - crystal oscillator clock
  - PLL1, PLL2
  - Clock dividing factors for the MC clock
- Assignment of highest arbitration priority to direct Host CPU reads and writes
- Threshold-controlled high and low priority support for all the requesters except
  - Direct Host CPU write and read requests
  - 2D Engine requests

The EMC manages data transfers to and from DRAM and supports the following features:

- Access to 2MB or 8MB DRAM, or up to 32MB External DRAM (SDR or DDR)
- Separate 128bit read and write buses
  - 32Bit bus access to DRAM
- External Memory Controller source clock driven from
  - Relaxation oscillator clock
  - Crystal oscillator clock
  - PLL1, PLL2
  - Clock dividing factors for the EMC clock
- Maximum operating frequencies for both MC and EMC
  - 145 MHz at 1.0 V operation
  - 212 MHz at 1.2 V operation

2.3.2 Overview

The memory client utilizes the following memory client address map:

- SRAM address range: 0 – 0x2000000h (32 MB)
  - Within this range, every 4MB is wrapped
  - 640KB SRAM: 0 – 0x1FFFFFFh
- DRAM address range: 0x2000000h – 0x4000000h (32 MB)
  - 2MB DRAM: 0x2000000h – 0x2200000h
- Host direct memory access address map
  - SRAM address range: 0x0400000h – 0x049FFFFh
  - DRAM address range: (related to the supported external memory size)
    - 2MB: x200000h – 29FFFFh
    - 8MB: x400000h – x49FFFFh
    - 16MB: xC00000h – xC9FFFFh
    - 32MB: x1C00000h – x1C9FFFFh

Interrupts are generated in the MC because of a bad address, or by the MC and EMC in response to a context switch. SRAM and DRAM address detection occur concurrently and result in interrupts as needed. Any captured data is kept until the interrupt is cleared.

- SRAM bad address detection
  - Upper address bits discarded for compare to fit in 4MB
  - Address compared to programmable IMEM_SIZE_KB
  - On bad address, full request info is stored (module ID, client ID, r/w, full address, byte enables, write data)
- DRAM bad address detection
  - Address compared to a programmable size (handled by GFSDK)
  - On bad address occurrence, full request information gets stored: module ID, client ID, read/write, full address, byte enables, write data

The MC is involved in resetting other modules, such as the 2D or 3D graphics engines. While the host registers are mainly involved in module resets, if there are ongoing or outstanding memory transactions involved, the MC plays a role in the reset function.

A typical module reset sequence looks like this, and is accomplished through GFSDK function calls:

- Disable the module to block its memory requests
- Enable module bit in the Host reset register to reset the module
- Enable the module's host reset function to clear the blocked requests seating before arbitration
- Poll the module's out request count till zero
- Disable the module's host reset to release the host reset
- Disable module bit in the Host reset register to release the module reset
- Enable the module to allow new requests to proceed to arbitration

2.4 2D Engine

2.4.1 Introduction

The 2D Engine is a specialized logic processor for graphics operations such as Bit Block Transfers (BitBLTs), Raster Operations (ROPs), area fills, and line drawing. It also provides hardware support for clipping, transparency, and font-color expansion.

Features of the GoForce 5500 2D Engine include

- Input from Host Interface (via Command Buffer) for
  - 2D
  - generic memory to screen BLT for video/audio

- Input from VI and EPP for
  - Stretch
  - YUV-to-RGB color conversion

- Screen to screen XY-swap (for rotation) - destination can overlap with source
- 16bpp and 32bpp
- Rectangle draw and BitBLT with 3-operand raster operation (ROP)
- All-angle Bresenham line drawing with sub-pixel resolution and ROP
- Mono (text) to color expansion
- Mono pattern or mono-source transparency
- Source or destination color transparency
- Alpha blending:
  - fixed alpha
  - source alpha (ARGB1555, ARGB4444, ARGB8888)
  - 1 bit, 2 bit, 4 bit, and 8-bit alpha plane

- Clipping
- Drawing synchronization with Display Module

- Multiple contexts
  - 3 simple-2D classes (host downloading, source copy)
  - 2 full-2D classes
  - 3 Video Scaler (VS) classes

- Multiple Engines
  - BitBlt
  - Line Draw
  - Fast Rotation
  - StretchBlt

- ROP3
  - Pattern path
    - Tile fill (mono only)
    - Mono expansion
    - Fix color fill
    - Rectangle copy (transparency clipping)

  - Source path
    - Mono expansion
    - Fix color fill
    - Rectangle copy (transparency clipping)

  - Destination path
    - Rectangle copy (transparency clipping)

- Alpha Blending and fading
  - 1/2/4/8 bits alpha plane
  - 1/4/8 bits source alpha
    - 1/4 bits alpha requires source 16bits
    - 8 bits alpha requires source 32bits
    - 32bits source blending with 16bits destination
  - Fading only needs source data
- Circular buffer support
  - Two contexts can have circular buffer enables: One triggered by VI, one triggered by the host.
  - Programmable buffer number and size
- Flexible trigger methods
  - Host trigger
    Host writes to a register with offset matching number in trigger registers.
    - 3 trigger registers to hold 6 triggering offset
  - VI trigger
    At the end of one circular buffer or one frame, VI sends trigger to the 2D Graphics engine
  - Link trigger
    At the end of one context command, the linked context is triggered

2.4.2 Overview

Freeing the Host CPU from most of the display rendering functions has three main benefits:

- Accelerated graphics operations produce smooth screen updates without visible start-and-stop or slowdown during heavy Host CPU use by another application.
- Lowered power consumption since the 2D Engine resides on the same chip as the display buffer.
- Increased efficiency: the Host CPU can perform time critical or real-time operations (such as software modem or other I/O functions) while the 2D Engine renders the display images.

Since the Graphics Engine may be shared with other host-controlled applications it can multiplex camera and host-controlled commands.

2.4.3 Rotation in the 2D Engine

- Fast Rotation: 2D Engine only
  - Requires each pixel be read and written
  - Does not require two full buffers
- Slow Rotation: 2D Engine Video Scaler (VS), VI, and EPP modules

2.4.3.1 Fast Rotation

Fast rotation is unique to the GoForce 5500 2D Engine. It is a memory-to-memory command and works in 8bpp, 16bpp, or 32bpp modes. Fast rotation cannot be combined with other 2D functions and is called fast because it works on a tile at a time to make efficient use of memory bandwidth. The tiles must be aligned to the memory boundary, so the following restrictions on the source apply. The source must be aligned to

- 16pixel boundary for 8 bpp
- 8pixel boundary for 16 bpp
- 4pixel boundary for 32 bpp

In-place rotation is possible where the destination rectangle is placed on top of (overlapping with) the source rectangle. However, if the source rectangle is not a square, then the rotated destination rectangle cannot be placed exactly on top of the source, resulting in additional memory consumption.

If the source is X x Y, for in-place rotation

- if X > Y: the memory needed is equivalent to X x X pixels.
- if Y > X: the memory needed is Y x Y pixels.

Fast rotation can also be performed with the destination rectangle stored in a different location which does not overlap the source rectangle. The memory needed would be X x Y pixels (for the source) and X x Y pixels (for the destination) - a total of two frames. Fast rotation consists essentially of an XY swap, horizontal flip, and vertical flip; so there are with possible image transformations possible.

If the source to be rotated is in planar YUV420/422 format, fast rotation must be applied to each of the three planes individually, with three fast rotation commands, in 8bpp mode. However, if the Y, U, and V planes are arranged in memory to form one single 8bpp rectangle, then the resulting YUV rectangle can be rotated with a single fast rotation command. If the fast rotation command in the 2D Engine was triggered by the VI module, then using the single YUV rectangle reduces the number of 2D commands in a single trigger.

The bandwidth required is simply the amount of bandwidth to read source surface, and the amount of bandwidth to write the destination surface.

Bandwidth requirement formula:

X x Y x bpp x 2

Slow rotation occurs in the 2D Engine's VS module and is described in that section.

2.4.4 2D Engine Interfaces

- Program interface with Host Controller
  - 32 bits registers width
  - Data and command using one FIFO
  - Raise/Refcount support

- 128Bit memory interface
- EPP interface
- VI
- Display

2.4.5 2D Engine Clocks and Power Savings

- Free running clock only drives 12 flip flops and Syn-cells in waiting mode
- All second-level gated clocks have register control bits to force enables

2.5 Video Scaler

2.5.1 Introduction

The Video Scaler (VS) is a submodule of the 2D Engine. The VS takes video images stored in the image buffer and makes them smaller or larger in size than the original; then it writes the result back into an area (such as an overlay area) of the image buffer.

The Video Scaler Module supports the following:

- YUV to RGB Color-space conversion (or RGB gain):
  - Input Image data in YCbCr (YUV) 4:2:2 YUV 4:2:0, 16-bit RGB, RGB 32-bit formats. (4:2:0 must be in planar format.)
  - Output data to memory in RGB565, YUV4:2:2, 32-bit RGB formats (YUV output only if YUV input)
- Output data format in YUV444 or RGB888 to EPP
- 2-to-1 interlaced scanning to progressive-scanning conversion
- Image expansion ratio up to 8:1
- Image contraction ratio down to 1/60
- Brightness, contrast, saturation, and hue all adjustable
- Color/chroma key masking
- Source data from host or memory
- Destination data to EPP or memory
- Can perform up to three Blit operations simultaneously:
  e.g. one host-triggered circular buffer, 1 VI-triggered circular buffer, 1 host-triggered full frame.
- Slow rotation

2.5.2 Overview

To start the VS function, system software running on the Host CPU sets up the VS Registers and sends an Execution Start Command to the VS. When the VS is finished it can send an interrupt to the Host CPU.

Input Data Formats

- YUV 422 Non-planar (8 variations)
  - Offset Binary, Two's Complement
  - YUYV, YVYU, UYVY, VYUY
- YUV 420/422 planar (converted to packed by MC)
  - Offset Binary, Two's Complement
- RGB 16 bpp
  - 565 RGB, byte swapped
- RGB 32 bpp
  - ARGB

(Output data formats are same, except for planar modes; and when the input data is YUV, the output is YUV 4:2:2 or YUV 4:4:4.)

2.5.3 2D Engine and the VS

If the Host CPU issues both 2D Engine and VS commands, the 2D Engine serves them on a first come, first served basis. Software basically manages the order of execution. If a VI module trigger issues either 2D Engine or VS commands, those commands receive highest priority and get issued once any current commands in process have finished. So Host-CPU issued and VI-triggered 2D Engine and VS commands run without synchronization.

2.5.3.1 Slow Rotation

The GoForce 5500 VS (as well as the VI and EPP modules) can perform a slow rotation, or XY swap, function. Slow rotation requires each pixel to be read and written, and requires two full buffers. The data can be in either YUV or RGB format. The VS is limited to performing slow XY swap only.

Slow rotation is performed when the source output is written to memory. It works on a raster scan basis instead of a tile basis, which means it is not memory efficient. The pixels in a source scan line (or row) are written vertically into a destination column.

Required Memory Bandwidths:

- RGB surface: to write a column of destination data, the write bandwidth required is 1 memory word (16-byte) per pixel (X x Y x 16), regardless of the number of bits per pixel (bpp).
- YUV planar surface: the write bandwidth required is 1 memory word (16-byte) per color component
  X x Y x 16 x 1.5 for YUV420
  X x Y x 16 x 2 for YUV422 planar

The VS supports slow XY swap (no horizontal or vertical flipping) for 8bpp, 16bpp, and 32bpp pixel depths. To use the VS for a slow XY swap on planar YUV, the three-plane surface must be treated as three surfaces of 8bpp each, or as a single combined YUV surface of 8bpp. Note that the VS cannot write planar formats in memory so the VS slow XY swap on planar YUV works only if the source rectangle is in memory and is declared as an 8bpp color depth.

However, when performed using the VS module, the XY swap can be combined with other VS operations. For example, the VS can read the YUV surface, scale it, convert it to RGB format, and write it to memory with a slow XY swap. If the VS performs a slow XY swap on a source in memory, then an in-place slow XY swap is not possible.

2.6 Video Input (VI)

The registers for configuring the GoForce 5500 VI module can be found in *Chapter 7,* "GoForce 5500 Micro-classes" in Section *2.6,* "Video Input (VI)".

2.6.1 Introduction

This chapter describes the GoForce 5500 Video Input (VI) module. The VI module receives data from video sources such as CMOS sensors, CCD cameras, an MPEG or live video (i.e. TV) decoder, or a Host CPU.

This module features:

- Programmable Y, U, Y, V data format ordering.
- Line averaging (vertical) filter for non-uniform weighted averages.
- Multi-buffering synchronization with the Display Module.
- Programmable horizontal low-pass filtering.
- Programmable horizontal decimation
  - With or without pixel averaging.
  - Maximum decimation ratio of 1/8 with pixel averaging
  - Maximum decimation ratio of 1/15 without pixel averaging
- Programmable vertical decimation
  - With or without line averaging
  - Maximum decimation ratio of 1/8 with pixel averaging
  - Maximum decimation ratio of 1/15 without pixel averaging
- Optional YUV (YCbCr) to RGB888 or RGB565 color-space conversion.
- Status and interrupt generation.
  - VD8, VD9, VD10, VD11, VGP4, VGP5, VGP6-generated interrupts
  - VHsync-generated interrupts
  - VVsync-generated interrupts
- ITU-R BT656 video input port
  - Video Input clock
  - 8-bit multiplexed Y and U/V data
- Support for a digital decoder input port
  - Video Input clock
  - VHSYNC and VSYNC
  - 8-bit multiplexed Y and U/V data
- CPU Host source video with support for 4:2:2 format in planar or non-planar format.
- Clocking by the Video Input clock.
- Camera input reference clock can be driven out on the pin VGP0.
- Programmable bypass function to send JPEG-encoded data stream directly to memory
- 10 bit/clock Bayer pattern input format
  - 12Bit/clock Bayer pattern input, with pins corresponding to bits [1:0] going to ground
  - Upper 10 bits processed by ISP module
  - Up to 79 MHz (at 1.0 V) and 105 MHz (at 1.2 V)
  - >5 MP cameras at 15 fps
- 8-bit/clock ITU-R BT.656 or YUV422/HS/VS input format
  - Up to 79 MHz (50 Mpixels/sec) - at 1.0 V, 105 MHz at 1.2 V.
  - >3 Mpixels camera at 15 fps

- Image Signal Processor for Bayer pattern input format
  - Black level compensation
  - Column and row noise reduction
  - Bad pixel correction
  - Color correction
  - De-mosaic
  - Gamma correction
  - RGB to YUV color space conversion
  - Auto exposure, auto white balancing, and auto focus performed in conjunction with AVP
- Color space conversion from YUV4:2:2 to YUV4:2:0
- Interface to 2D Engine for fast rotation, stretch, and YUV-to-RGB conversion on VI data
- Slow rotation (XY Swap) capability
- Interface to output (read) DMA to send preview capture data to host
- Output to Encoder Pre-Processor for video encoding before or after the decimator, with byte swap options
- Circular Buffer
  Camera-controlled video processing in other modules can occur without software intervention. As video data fills the allocated buffers, the VI notifies the encoders, StretchBLT, or Display modules
- The VI operates off of six possible clock sources
  The camera clock is for data coming from a camera
  PLL for host data only
- Maximum clock frequencies: 79 MHz (1.0 V) or 105 MHz (1.2 V core operation)

2.6.2 Overview

Figure 2.3 shows a simplified block diagram of the GoForce 5500 VI module. Either a camera or the Host Interface sends data into the VI. The VI Module receives the data, which can be YUV 4:2:2 or Raw Bayer data.

Table 2.3, below, shows the mapping of pins VD[11:0] to the different data sources.

Table 2.3: Mapping of Pins VD[11:0] to Bayer and YUV Data Inputs

VI Pin	12Bit Bayer	10Bit Bayer	YUV Data
VD11	Bayer11		
VD10	Bayer10		
VD9	Bayer9	Bayer9	YUV7
VD8	Bayer8	Bayer8	YUV6
VD7	Bayer7	Bayer7	YUV5
VD6	Bayer6	Bayer6	YUV4
VD5	Bayer5	Bayer5	YUV3
VD4	Bayer4	Bayer4	YUV2
VD3	Bayer3	Bayer3	YUV1
VD2	Bayer2	Bayer2	YUV0
VD1	Bayer1: Not used	Bayer1	
VD0	Bayer0: Not used	Bayer0	
Notes	1	2	3

1. Only Bayer[11:2] data is currently processed by the ISP module.

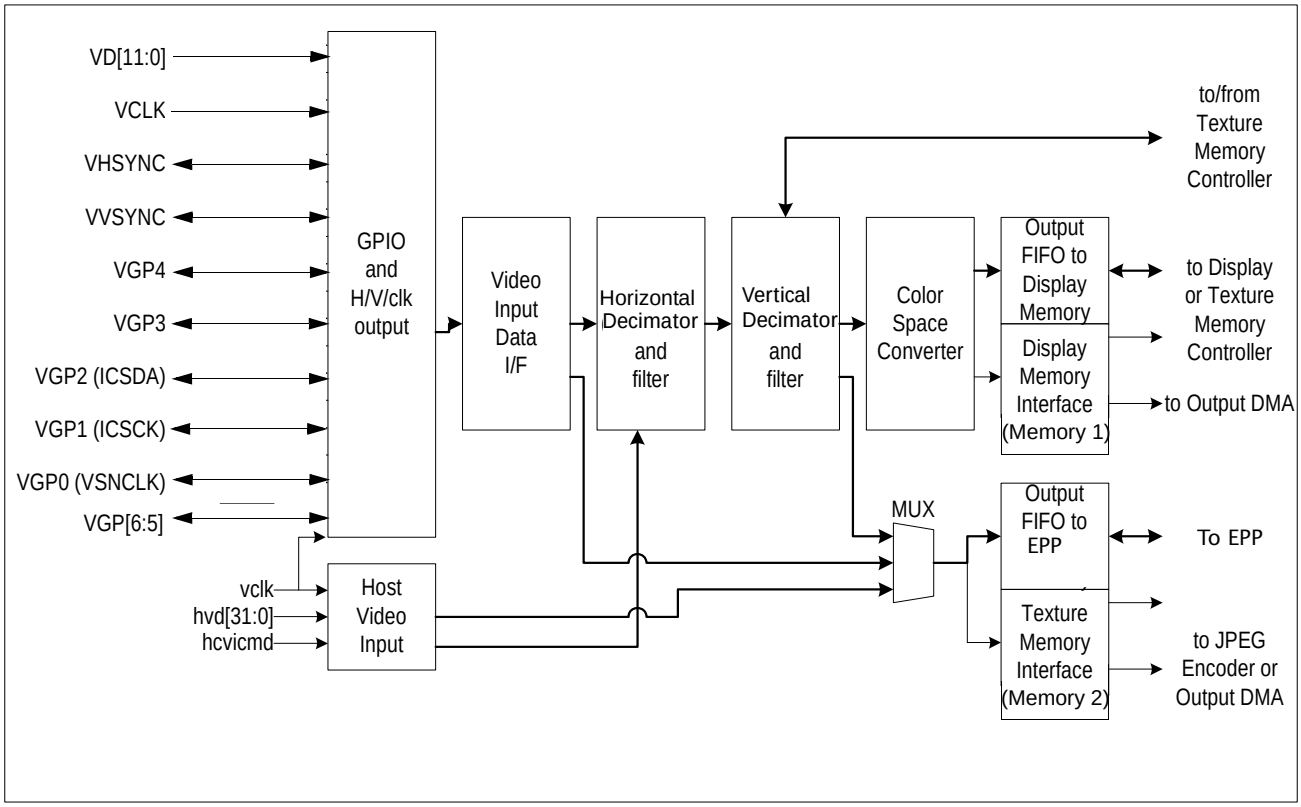
2. Pins VD1 and VD0 should have their inputs disabled. They may be used as general purpose outputs. Bayer[9:0] data is processed by the ISP module.

3. Pins VD[11:10] and VD[1:0] may be used as general purpose IOs.

Once the VI module receives the input data, the data undergoes horizontal low-pass filtering, pixel averaging, horizontal decimation, then line averaging and vertical decimation. The transformed YUV 4:2:2 signal gets color-space converted into RGB565 format and sent to the Video Output FIFO, if previewing is required.

JPEG-encoded stream data input through the VI input port can be sent directly to the JPEG Stream Write Buffers.

Figure 2.3: Simplified Block Diagram of the GoForce 5500 Video Input Module



2.6.3 VI Module Block Functions

Video data can come from the Host Interface, a camera, display memory, or external memory. The blocks shown in Figure 2.6 are described in the following sections.

The VI Module interfaces with the Host, for programming the VI registers and writing the YUV Host Data FIFOs; with the MC for all memory writes and reads; and with the ISP for Bayer conversion to YUV 4:4:4.

Note that the VI module can receive sensor data and host data simultaneously if one of the inputs is Bayer data and goes from the ISP to the EPP module. The other data can go through the VI Core and then to memory. The VI's dual memory output capabilities mean that each memory output can select from pre or post-downscaled video data. Each memory output may select from the output of the color-space converter.

2.6.3.1 Video Signal Processing

Video input downscaler and filters

The horizontal decimator block performs horizontal low-pass filtering (LPF) and filtered sampling-rate conversion on incoming data. The maximum decimation ratio is 1/16; the low-pass filtering is programmable. Decimation and all filtering can be disabled so the input data can pass through without modification.

The vertical decimator block performs line sub-sampling and vertical filtering. It is capable of decimation ratios up to 1/16. The vertical decimator uses part of the texture memory as its line buffer. The decimation and filtering can be disabled to pass the input data without modification. The vertical decimator supports two initial DDA values for field 0 and field 1 to compensate for the vertical position difference between the two fields when the video source is interlaced video. This allows proper conversion from interlaced video source to progressive-scan display. The video input module always converts each odd field or each even field into a single frame per field (Bob mode). The video input module does not support weaving of an odd field and an even field into a single frame (Weave mode).

The YUV4:2:2 signals are selected from either the parallel video input or the host video input and then go into the signal processing datapath. Horizontal signal processing is done first, followed by vertical signal processing followed by optional YUV-to-RGB color space conversion.

Horizontal data may go through a low-pass filter (which can be bypassed.) The data is decimated with or without pixel averaging. Decimation performed without pixel averaging uses the decimation factor defined below.

Decimation Factor (no pixel averaging) = n/m, where:

n is the input horizontal size in pixels
m is the output horizontal size in pixels

Horizontal Decimation performed with pixel averaging uses decimation factors restricted to those shown in Table 2.4 below.

Table 2.4: Horizontal Decimation Factors with Pixel Averaging

Horizontal Decimation Factor	Averaging
1/2	Two-pixel averaging
1/3	Four-pixel averaging
1/4	Four-pixel averaging
1/7	Eight-pixel averaging
1/8	Eight-pixel averaging

Vertical decimation is performed when the vertical data is processed.

Vertical decimation factor = n/m, where:

n is the input vertical size in lines
m is the output vertical size in lines

Vertical decimation may be performed with or without line averaging. The maximum vertical decimation without line averaging is 1/15. Line averaging may be fixed (2 line) or flexible. The vertical decimation factors and corresponding line averaging are shown in Table 2.5 below.

Table 2.5: Vertical Decimation Factors with Line Averaging

Vertical Decimation Factor	Line Averaging
1/2	Two-line averaging
1/3	Four-line averaging
1/4	Four-line averaging
1/7	Eight-line averaging
1/8	Eight-line averaging

The data is sent to the External Memory Controller (sometimes muxed with data directly from the Host Interface), or color-space converter.

2.6.3.2 VI Color-space Converter

The color space converter converts YUV4:2:2 formatted data to RGB565 data. The conversion coefficients are programmable so the color-space converter can also perform brightness, contrast, hue, and saturation adjustments. The converter can be disabled to pass the input data without modification.

2.6.4 VI Module Interfaces

2.6.4.1 Input From the Host Interface

Video input data from the Host Interface Module gets written, typically in YUV4:2:2 (YCbCr) data format to the Video Input Data FIFO. However, the data can come in either of two ways:

- YUV (YCbCr) data is written in the Y-FIFO only
  - 32-bit Y1, V0, Y0, U0 data in little Endian format (bit 31 as MSb and bit 0 as LSb.)
  - Programmable format
  - Recommended method when original video source data is stored in YUV4:2:2 format.
- YUV (YCbCr) data is written in the Y, U, and V FIFOs.
  - Y data written in the Y-FIFO, U data written in the U-FIFO, V data written in the V-FIFO.
  - Recommended method when original video source data is stored in planar YUV4:2:2 format.

The Y-FIFO is 32bits wide by 16 deep, the U and V-FIFOs are both 32bits wide by 8deep.

When the video input data comes from the host interface, the VI module clock can be generated from the relaxation oscillator, PLL1, PLL2, or crystal oscillator. The VI module clock can also come from the VCLK pin, Refclk0, and Refclk1 pins. VCLK should be selected when both the host and camera inputs are used.

2.6.4.2 VI GPIO

The VI Module GPIO block controls signals into and out of the video camera interface. Seven of the signals (VGP[6:0]) may be utilized as GPIOs. (Other VI pins may also be used as GPIOs if they are not utilized in a design. These include unused data pins, or VHSYNC or VVSYNC if embedded syncs are used in place of the latter two signals.) The VI GPIO block generates horizontal and vertical sync outputs (VHSYNC and VVSYNC, respectively) and the Video Clock (VCLK) output signals. These three signals can be output to, for example, the MPEG Decoder for synchronizing video output data. The outputs are generated as follows:

- VCLK
  - VCLK generated from internal video camera interface clock (VCLK) with clock divider from 1 to 8
  - VCLK output polarity is programmable
  - VCLK maximum frequency at 1.0 V is 79 MHz, 105 MHz at 1.2 V

- VHSYNC (Horizontal Sync output)
  - VHSYNC generated at VCLK rising edge
    - Period up to 8192 clock cycles
    - Width of up to 16 clock cycles.
  - VHSYNC can be generated
    - Using internal LCD horizontal sync (LHS) or
    - Using internal LCD horizontal pulse 2 (LHP2)
  - VHSYNC output polarity is programmable.

- VVSYNC (Vertical Sync) output
  - VVSYNC generated at the VHSYNC leading edge (programmable from -2 to 5 VCLK cycles)
    - Period up to 4096 lines
    - Width of up to 16 lines.
  - VVSYNC can be generated
    - Using internal LCD vertical sync (LVS) or
    - Using internal LCD vertical pulse 1 (LVP1)
  - VVSYNC output polarity is programmable.

2.6.4.3 VI Data I/F

The VI Data I/F Block does the following:

- Receives the input data stream from the video input port/pins
- Decodes the EAV and SAV codes (if input stream is compliant to ITU-R BT656)
- Selects the input stream (data, clock, and control signals) from either the video input or from the host video input
- Includes horizontal and vertical counters that generate signals which indicate the capture window area.

The VI Data I/F also does the odd/even field detection in which the odd field is mapped to field 1 and the even field is mapped to field 0.

- For ITU-R BT656 data streams, the odd/even field indicator is embedded in the data stream
- For YUV4:2:2 data streams with H/V sync
  - detects the odd field when the V sync (VVSYNC) active edge occurs and when H sync (VHSYNC) is low
  - detects the even field when the V sync (VVSYNC) active edge occurs and when H sync (VHSYNC) is high.

If video input data comes from the Host Interface and goes directly either to the horizontal decimator and filter or to the External Memory Interface, it bypasses the VI Data I/F.

2.6.4.4 VI Output Memory Interface 1

The Display to Memory Interface consists of three 8-word FIFOs. The width of each FIFO word equals the width of each memory word. Video data may be written to memory in the following formats:

- YUV4:2:0 format (128-bit aligned)
- YUV4:2:2 format (32-bit aligned)
- RGB565 format (16-bit aligned)

If written data is in RGB565 format, data may be written

- with XY coordinates transposed
- with horizontal/vertical flip.

All three types of data format may be written with horizontal and/or vertical flips to achieve 180-degree rotation. RGB data may have 90-degree or 270-degree rotation performed on it with a combination of XY transpose and horizontal/vertical flips.

Data can be written to memory either in two ping-pong frame buffers or in multiple (2 to 8) wrap-around data buffers per frame. If multiple data buffers are used, the beginning of the next frame starts in the next available data buffer. Each data buffer size is defined as a multiple (1, 2, 4, or 8) of 16 YUV-lines or 16 RGB-lines. Therefore the start of each data buffer is always aligned to the 256-bit (32-byte) boundary.

Control signals can be sent to the output DMA module to process (transfer to host) the data buffers written in memory. Frame start and end boundaries are also sent to the output DMA module. Multiple frames can therefore be transferred through the output DMA module. The output DMA module always assumes that lines in the data buffer are all packed and it transfers all data in the data buffer. However, the last data buffer of each frame may not be completely filled if the frame size is not exact multiple of buffer size, in which case it does not get transferred. If transferring YUV or RGB video data through the output DMA module then this data must not be written with XY transpose enabled or with horizontal/vertical flip enabled.

2.6.4.5 Video YUV4:2:0 Write Data Format

When writing YUV4:2:0 data to memory, the VI module always groups the data in multiples of 16 YUV lines, with all Y lines written first; followed by all U lines; then followed by all V lines. The amount of Y data is four times the amount of U (Cb) and V (Cr) data since each U pixel and V pixel are shared between four (2x2) Y pixels. Since the data in the VI Module data path is in YUV4:2:2 format prior to the color space conversion, this YUV4:2:2 data must be converted to YUV4:2:0 data before writing it to memory. Conversion can be done by either throwing away every other U, V line or by averaging pairs of adjacent U lines and V lines.

2.6.5 Slow Rotation

The GoForce 5500 VI module can perform a slow rotation, or XY swap, function. Slow rotation requires each pixel to be read and written, and requires two full buffers. The data can be in either YUV or RGB format. The VI can perform “on the fly” slow rotation, as source output data is written to the destination buffer in memory.

Slow rotation is performed when the output is written to memory. It works on a raster scan basis instead of a tile basis, which means it is not memory efficient. The pixels in a source scan line (or row) are written vertically into a destination column.

Required Memory Bandwidths:

- RGB surface: to write a column of destination data, the write bandwidth required is 1 memory word (16-byte) per pixel (X x Y x 16) regardless of the number of bits per pixel (bpp).
- YUV planar surface: the write bandwidth required is 1 memory word (16-byte) per color component
  X x Y x 16 x 1.5 for YUV420
  X x Y x 16 x 2 for YUV422 planar

Both the VI and EPP modules support slow rotation for

- 16bit RGB
- 32bit RGB
- YUV420 planar
- YUV422 planar: the result becomes YUV422R (rotated) planar.

The advantage of slow rotation (XY swap) is that it can be done by the VI (or the EPP) module without involving the VS module. Also, since it is writing one pixel at a time, the VI Engine does not have the source and destination surfaces’ memory alignment restrictions - it only needs to be pixel-aligned.

2.7 Image Signal Processor (ISP)

The GoForce 5500's Image Signal Processor (ISP) receives raw Bayer data from the VI module, performs functions on the Bayer Data, and converts it to either YUV or RGB data; or bypasses the VI module to send the raw Bayer Data directly to memory.

2.7.1 Introduction

- ISP processes pixel data from CCD or CMOS imaging devices
  - ISP accepts outputs from Bayer Color Filter Array type of imagers
- Input bus consists of
  - 8 to 10 bits of parallel digital data
    - (8 to 10 bits used: MSB[11:10] should be grounded for 12bit sensor buses)
  - H and V Sync pulses
  - Pixel clock
- Black compensation
- Defective pixel concealment
- Lens shading compensation
- Edge enhancement
- De-kneeing
- ISP outputs YUV 4:4:4 image data to the VI module
- Statistics data for auto exposure
- Statistics data for auto focus
- Auto white-balance and statistics gathered for more advanced auto white-balance
- Programmed white selections
- Auto exposure compensation
- Image enhancement
- Noise reduction
- Color processing, gain, contrast, hue, saturation
- Multiple de-mosaicing schemes
- Maximum image size is 4000 pixels per line, 4000 lines
  - 79 MHz maximum pixel clock rate (5 Megapixels at up to 15 fps) - 1.0 V operation
  - 105 MHz maximum pixel clock rate (1.2 V operation)

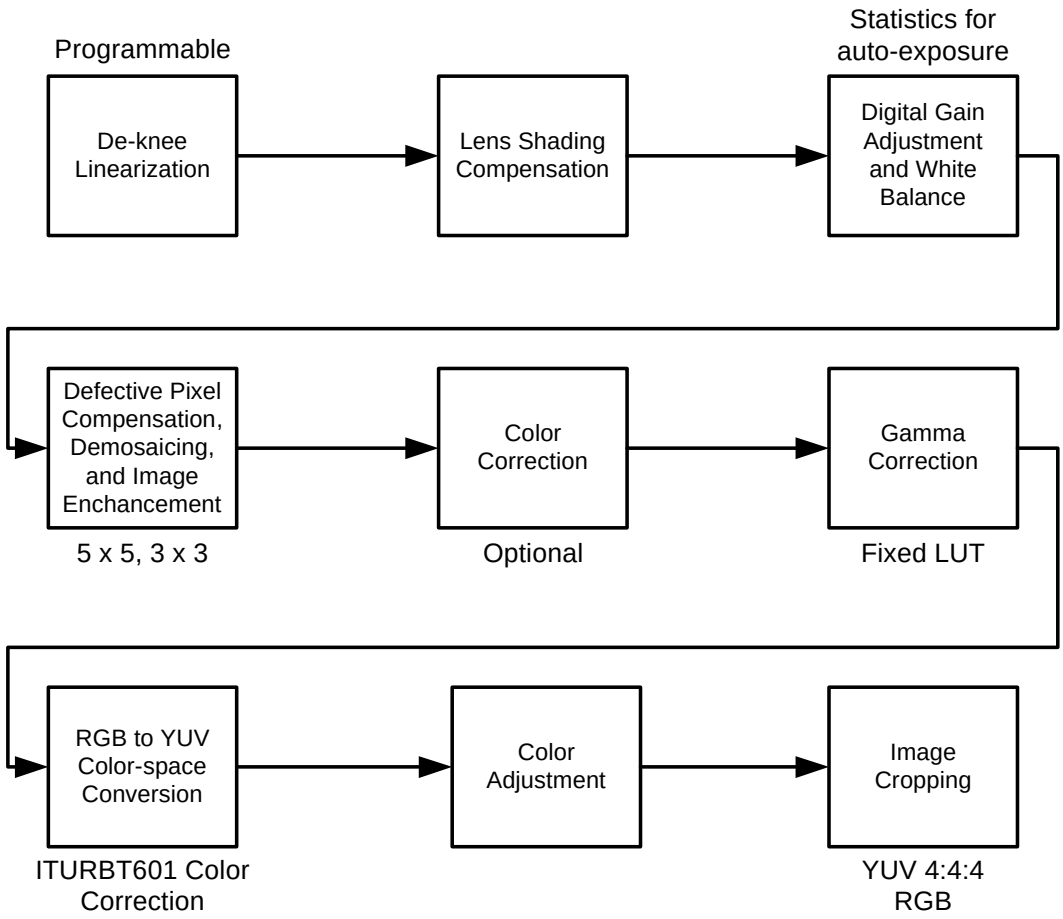
Functional blocks:

- Pixel processing data pipe
- Memory interface (for multi-line buffer memory)
  Multiple-port read/write client to the Buffer memory
- Host interface
- Sync and timing generator

2.7.2 Overview

The ISP module takes Bayer CFA-type images and changes them to RGB-type through such functions as de-mosaicing and color correction. Black-level calibration, de-kneeing, and lens-shading processes are all optional ways of achieving improved image quality. RGB signals may be optionally converted to YUV with a range of color adjustments.

Figure 2.4: Signal Flow in ISP Block: Output to Memory



2.7.3 ISP Functional Blocks

The ISP functional blocks shown in Figure 2.4 are described below.

Optical Black Level Compensation: The optical black level compensation block establishes a reference black level common to all the active pixels. It receives pixels from the portion of an imager, usually masked off so as not to receive any light. It extracts the black level of those pixels, and uses them as the black reference level for all active pixel signals.

Defective Pixel Compensation: Defective Pixel Compensation deals with individual pixel defects. Input levels of individual pixels are compared to those of neighboring pixels. Pixels with significant input level differences can be considered possibly defective, and the ISP can compensate for them. The ISP does not compensate for image sources with vertical or horizontal streaks of defective pixels. (Up to 2 x 2 can be concealed.)

Lens Shading Compensation: An optical lens in front of an imager tends to introduce shading and variations in the strength of incident light. It is strongest at the center and weakest at the farthest point from the center. Actual shading characteristics vary with different optics.

Lens shading compensation applies a gain to the pixels, in proportion to their distances from the center, to boost pixel signal strength. The pixels closest to the center receive little or no signal gain but moving away from the center pixels receive increasing amplification.

De-knee Curve Lookup Table: The De-knee curve lookup table helps to compensate for non-linearity in imaging cell (e.g. CCD) transfer characteristics.

Digital Gain Adjustment: This module applies adjustable gains to pixel signals to maximize full dynamic range. Three or four gain values are selected for RGB (or RG and GB) primaries of the color filter array. The values are determined by referring to the statistics gathered in the modules downstream. So this module must cooperate with the white balance module and its operation modes.

Color Correction: In a strict sense, the color correction module transforms a given color space based on the optical characteristics of the imager to a specific color space, such as that of CCIR 601, handling the color component signals in that standard color space. In practice, the color shade of images is adjustable to suit the viewer's taste by adjusting the gain and offset of R, G, and B pixels independently.

White Balance: In the White Balance block, RGB component signal levels are balanced to render white objects as white. This is a normalization step of the electronic signal of RGB color primaries. (Simple Auto white-balancing, AWB, is available in the ISP. More sophisticated algorithms are implemented in the AVP. The ISP serves to gather useful statistics in the latter case.)

Gamma Correction: Gamma correction is applied to RGB signals, using a gamma value of 2.2, to compensate for the non-linear characteristics of display devices.

RGB-to-YUV Color-space Conversion: RGB signals are converted to YUV signals in this block.

Noise Reduction: Noise reduction minimizeS noise generated by the sensor, or possibly by other ISP operations.

Image Cropping: The input image may be cropped; if so, it will be cropped to a rectangular shape.

2.7.4 Data Input to ISP

The ISP can connect to a 12bit bus, but only supports 8bit and 10bit data inputs. It can accommodate a 12bit interface, but not all 12 bits of incoming data. To connect to the ISP, use the following guide:

- 8Bit YUV422 data: Pins [9:2]
- 10Bit Bayer data: Pins [9:0]
- 12Bit Bayer data: Pins [9:0] (Ground Pins [11:10]: they are not used)

2.8 Encoder Pre-processor (EPP)

2.8.1 Introduction

The Encoder Pre-processor (EPP) receives video frames from the VI module, Display module, or VS module. It supports processing incoming video frames in the following formats (MIPI support):

- RGB888 non-planar
- YUV 4:4:4 non-planar

All of the above data is eventually provided to the EPP in YUV 4:4:4 or RGB 8:8:8 format.

The EPP stores these in memory buffers. the EPP's output to memory is in either planar or non-planar AYUV 4:4:4, ARGB888, YUV 4:2:2, YUV 4:2:0, or Bayer. non-planar format or YUV4:2:0 planar format. The EPP can then send commands to the Host Interface for the output DMA, to the JPEG/MPEG Encoder, or Display module for further image processing.

The EPP module provides the following:

- Video capture from
  - Video (camera) input
  - Video Scaler
  - Display
- Pre-processing filter for MPEG/JPEG encoder
- Circular buffer support
- Slow Rotation (XY Swap)
- Interface to output (read) DMA to send pre-encoded data back to Host CPU

2.8.2 Overview

In supporting the functions listed above, the EPP utilizes the following capabilities:

- Takes input in the following formats:
  - From VI: in YUV4:4:4 non-planar
  - From Display: in RGB888
  - From VS: YUV4:4:4 non-planar or RGB888
- Performs color-space conversion from RGB888 to YUV4:4:4
- Performs conversion from
  - YUV4:4:4 (non-planar) to YUV4:2:2 (non-planar)
  - YUV4:2:2 (non-planar) to YUV4:2:0 (planar) with optional 2-line chroma averaging
- Utilizes a pre-encoding luma filter for removal of high-frequency noise
- Sends output to memory in
  - 32-bit aligned YUV4:2:2 (non-planar)
  - 8-bit aligned YUV4:2:0 planar format
- H (horizontal) and V (vertical) output scanning direction control. Reversing the H output scanning direction utilizes byte swapping within each memory word.
- XY swap for rotation.
- Optional duplications of the first and last pixel of a line for 128-bit boundary alignment.
- Cropping of input frame - creates a subset of the input and makes that an output frame.
- Output multi-buffering (set of 256 buffers) in memory; a frame always starts in a new buffer.

2.8.3 Slow Rotation

The GoForce 5500 EPP module can perform a slow rotation, or XY swap, function. Slow rotation requires each pixel to be read and written, and requires two full buffers. The data can be in either YUV or RGB format. Unlike the VS module, the EPP performs “on the fly” slow rotation, as the source data is written to the destination buffer in memory.

Slow rotation is performed on output written to memory. It works on a raster-scan basis instead of a tile basis, which means it is not memory efficient. The pixels in a source scan line (or row) are written vertically into a destination column.

Required Memory Bandwidths:

- RGB surface: to write a column of destination data, the write bandwidth required is 1 memory word (16-byte) per pixel (X x Y x 16) regardless of the number of bits per pixel (bpp).
- YUV planar surface: the write bandwidth required is 1 memory word (16-byte) per color component
  X x Y x 16 x 1.5 for YUV420
  X x Y x 16 x 2 for YUV422 planar

The EPP module supports slow rotation for

- 16bit RGB
- 32bit RGB
- YUV420 planar
- YUV422 planar: the result becomes YUV422R (rotated) planar.

The advantage of slow rotation (XY swap) done in the EPP module is that it does not need to involve the VS module. Since it writes one pixel at a time, the EPP module does not have the source and destination surfaces’ memory alignment restrictions - it only needs to be pixel aligned.

However, keep in mind that slow rotation performed in the VS module can be combined with other VS operations such as scaling and color-space conversion.

2.8.4 Interfaces

- Interface to JPEG/MPEG Encoder
  This consists of new-buffer and start-of-frame signals. The EPP sends a buffer address to JPEG/MPEG Encoder. The Host CPU must prepare the encoder input buffer ordering to match the EPP output buffer ordering.

- Interface to Host Interface (output) DMA
  The H size (in bytes/line) and line stride should be programmed directly in the output DMA module. The number of lines in the first and last buffer of a frame may be less than the buffer size. For YUV 4:2:0 planar format, three commands per buffer go to the output DMA. The GoForce 5500 sends the Host CPU a buffer index and new buffer signals.

- DIsplay
  The EPP sends the buffer address with frame start and frame end information to the Display.

- Host input to EPP
  Class and register reads and writes are through this interface.

- VI input to EPP:
  VI pass video data and status through stream video data bus, including raise vectors.

- VS input to EPP
  Same as VI to EPP interface

- Display input to EPP
  Same as VI to EPP interface

The EPP Module often receives end-of-buffer or end-of-frame raise information (along with the associated raise vector) when it receives video data from the VI or VS modules. The EPP module returns the raise vector to the host when the specified event occurs and all output data preceding the event has been received by the memory controller.

2.9 Display Controller

The registers for configuring the GoForce 5500 Display are found in *Chapter 7,* "GoForce 5500 Micro-classes" in Section *2.9,* "Display Controller"

2.9.1 Introduction

The Display Module drives the display device connected to the GoForce 5500.

The following is a list of features of the Display Module:

- 1 bpp, 2 bpp, 4 bpp, and 8bpp palettized color depth
  - Converted to 24bpp using a triple palette
  - 1 bpp, 2 bpp, 4 bpp supported on window A only
- 12 bpp (B4G4R4A4), 15bpp (B5G5R5A), and 16bpp (RGB565) color depths
  - Converted to 24 bpp
    Converted using the adaptive color expansion
    Converted using the triple palette
- 32 bpp (R8G8B8A8 or B8G8R8A8)
- YUV packed, YUV 4:2:2 planar, YUV 4:2:0 planar, YUV 4:2:2 rotated planar, and YCbCr (Window B and C)
- Two display modes, Primary and Secondary:
  - Separate set of registers for each display (one for Primary, one for Secondary)
  - Parameters used for dual-resolution display support
- Three windows per display: Window A, Window B, and Window C. Each has
  - Two color key generators
  - Horizontal (H) and Vertical (V) scaling and flip
  - Double buffering
  - Gamma (or palette) Look Up Table (LUT)
- Overlay Blending of the three windows
  - Digital Vibrance
  - The three windows can be supported in both primary and secondary display modes:
    - Display any combination of graphics/video
    - Color key where the two or three windows overlap for graphics/text overlay function
    - Position and size of the three windows are fully programmable within the active display area
- Programmable output window data going to the EPP for encoding
- Fully programmable display resolution and timing limited only by
  - horizontal/vertical counter resolution (12-bit)
  - available display memory size
  - pixel or shift clock frequency
- Double buffering in primary and secondary display modes
- Buffer switching enabled by
  - software
  - 2D engine (at end of a command)
  - video camera interface module (at end of a captured frame)
  - EPP module (at the end of a frame)
  - MPEG Encoder module at the end of either a reconstructed or a reference frame

- Horizontal and vertical image flip (scanning in decrementing x or decrementing y direction)
  - 90-degree and 270-degree image transformation can be achieved using
    - Horizontal or vertical flip in conjunction with XY transpose function in other modules such as the 2D or 3D engine, the video camera interface module, EPP, or the JPEG/MPEG decoder
  - 180-degree image transformation can be achieved by enabling both horizontal and vertical flips

- 64 x 64 or 32 x 32 2bpp hardware cursor with foreground/background color and normal/inverse pixel transparency
- Odd byte and even byte swapping option for all color depths
- Display interface to various displays:
  - Up to 24bpp parallel RGB (1-clock per pixel) direct programmable TFT or PWM-STN interface
  - Parallel host interface (Type A or C) to display (TFT, STN, LTPS, etc)
    - Up to 24bpp 1-clock/pixel (up to 24bits per clock) parallel host interface to the display
    - 16/18/24 bpp 2-clock/pixel (8bit, 9bit, or 12bit per clock) parallel host interface to the display
    - 12bpp 3-clock/2-pixel (8-bit per clock) parallel host interface to the display
    - 18bpp 3-clock/pixel (6-bit per clock) parallel host interface to the display
    - Initialization sequence (IS) supported

  - 1-, 2-, or 3-channel low-voltage differential serial interface
  - Serial Peripheral Interface (SPI)

- 2x2 Ordered dither with matrix rotation (programmable output size from 3 bpp to 18 bpp)
- Ordered dither of 24bpp data down to 18bpp, 16bpp, 15bpp, 12 bpp, to 3bpp

- Error diffusion dithering with programmable output size from 3 bpp to 18 bpp
  - For low-power mode
  - For enhancing the image quality of displays with less than 24 bpp

- Error-diffusion dither of 24bpp data down to 18 bpp, 16 bpp, 15 bpp, 12 bpp, to 3 bpp
  - Max 640 pixels error diffusion line buffer

- Pulse width modulation signals (LPM0, LPM1) for contrast and brightness control
- Three programmable horizontal pulse (LHP0, LHP1, LHP2) signals
- four programmable vertical pulse (LVP0, LVP1, VP2, and VP3) signals (VP2 and VP3 are shared with other pins.)
- Two programmable modulation signals (LM0, LM1) which can toggle every n lines
- Programmable pulse (LPP) with maximum 128 pulses per line and data inversion option
- Three display power sequencing signals: LPW0, LPW1, LPW2
- SPI or serial host interface
  - host SPI, IS SPI, LCD SPI

- Continuous or non-continuous display-refresh operation
  - Frames sent to display either continuously or one frame at a time

2.9.2 Overview

The Display Module drives a display device. Two displays, the primary and secondary, can be connected to the GoForce 5500 at the same time. The Display Module can switch from one to the other quickly and easily due to separate sets of registers for the primary and the secondary displays. The Primary and Secondary displays may have independent timing parameters.

Each display mode can have up to three graphics image windows (*Primary window A*, *Primary window B, Primary window C; and Secondary window A, Secondary window B, Secondary window C*) that can be programmed with independent position, size, and color depths. Color keying and blending is used in the overlap area between the up to three graphics image windows. Each display mode also supports a hardware cursor.

Window A features

- Data Format
  - 1bpp, 2bpp, 4bpp, 8bpp palettized
  - B4G4G4A4, B5G5R5A, and B5G6R5 format
  - B8G8R8A8 and R8G8B8A8 format
- Three 256 x 8bit palette A for palettized data formats and for gamma correction of non-palettized data formats
- Horizontal and Vertical flip
- Horizontal and Vertical scaling
  - Up to 8x1 horizontal downscaling for 16 bpp and less
  - Up to 4x1 horizontal downscaling for 32 bpp
  - From 1 to 15x horizontal and vertical up scaling
  - Up to 16x vertical downscaling
  - No filter (pixel and line replication only)
- Digital Vibrance (8-level)
- Can generate 2 color keys (RGB range, no alpha key)

Window B features

- Data Format
  - 8bpp palettized
  - B4G4G4A4, B5G5R5A and B5G6R5 format
  - B8G8R8A8 and R8G8B8A8 format
  - YUV420, YUV422, YUV422R (planar format)
  - YCbCr420, YCbCr422, YCbCr422R (planar format)
  - YUV422 and YCbCr422 packed
- Three 256x8bit palettes for 8bpp palettized data format and for gamma correction of non-palettized data formats
- Horizontal and Vertical flip
- Horizontal and Vertical scaling
  - Up to 8x1 horizontal downscaling for 16 bpp and less
  - Up to 4x1 horizontal downscaling for 32 bpp
  - From 1 to 15x horizontal and vertical up scaling
  - Up to 16x vertical downscaling
  - 6-tap, 16-phase programmable H filter
  - 2-tap, 16-phase programmable V filter
- YUV to RGB color space converter
- Digital Vibrance (8-level)
- Can generate 2 color keys (RGB or YUV range, no alpha key)

Window C features

Window C is identical to Window B, without the vertical filter.

Window A only supports RGB data formats and should typically be used for the user interface (menu and icons). If it is used for video, color-space conversion from YUV to RGB must be done outside the display module. Windows B and C can be used for either video or graphics overlays and both support YUV-to-RGB conversion.

All three windows support arbitrary scaling functions, with different degrees of filtering. Scaling is typically not performed on the graphics user interface. Therefore, scaling in Window A is supported only with pixel and line replications, and without horizontal and vertical filtering.

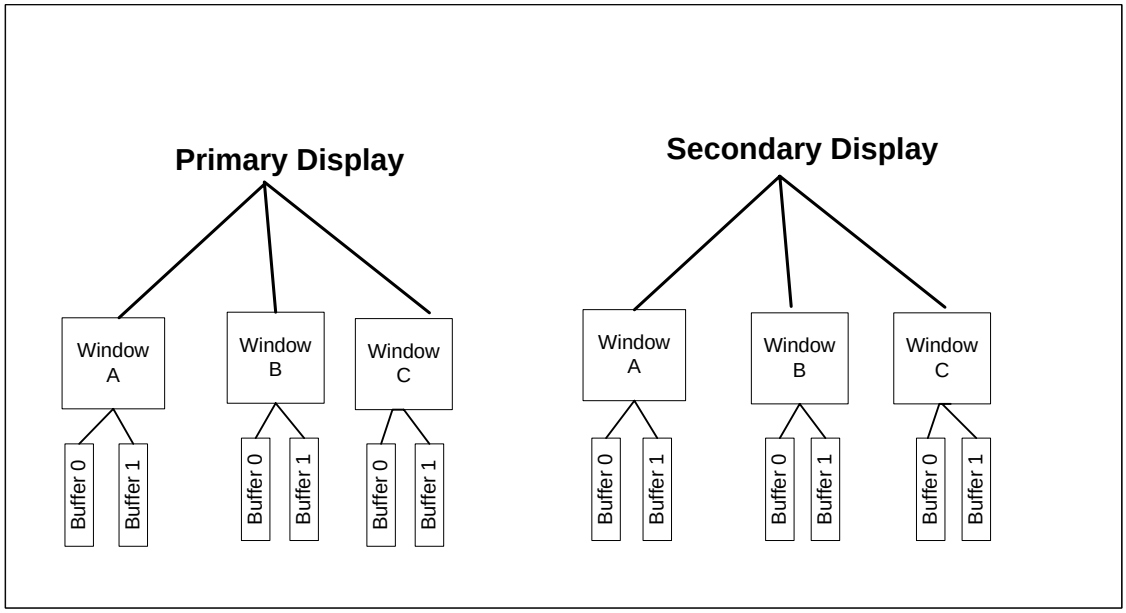
Scaling typically is required for video overlays. So both Window B and Window C support 6-tap, 16-phase horizontal filtering. Window B supports 2-tap 16-phase vertical filtering. Window B can produce better-quality scaling than Window C because of its vertical filter, utilizing twice the memory bandwidth in the process. All horizontal and vertical filter coefficients are independently programmable for both Window B and Window C, However, they are shared between the Primary and Secondary display modes.

A scaling ratio can be programmed independently for each window and between Primary and Secondary modes. Up-scaling does not increase the memory bandwidth requirement. When down-scaling the peak memory bandwidth requirement changes in proportion to the down-scaling ratio. For example, a 2-to-1 down-scaling ratio requires twice the peak memory bandwidth for the down-scaled window. To reduce or minimize the memory bandwidth, perform down-scaling elsewhere in the GoForce 5500 or limit the downscaling to a maximum ratio of 2-to-1.

Higher memory bandwidth usage causes larger power consumption.

Source images for all graphics image windows are stored in image *buffers* in the internal memory. Double buffering (*buffer 0* and *buffer 1*) is supported for each graphics image window. It is possible to switch between primary and secondary display mode and at the same time switch between buffer 0 and buffer 1 of either window.

Figure 2.5: Primary and Secondary Display Block Overview



Supported color depths

- 8Bpp (palettized)
- 1Bpp (palettized)
- 2Bpp (palettized)
- 4Bpp (palettized)
- 12Bpp (B4G4R4A4)
- 15Bpp (B5G5R5A)
- 16Bpp (B5G6R5)
- 32Bpp (B8G8R8A8 and R8G8B8A8)
- YUV/YCbCr 4:2:0 or 4:2:2.

All graphics modes support odd-even byte swapping. Three 256 x 8bit color palettes are used for 1 bpp, 2 bpp, 4 bpp, and 8 bpp to convert them to 24 bpp. The color palette is shared for both graphics image windows in Primary and in Secondary display modes. Also, 4bit, 5bit, and 6bit-to-8bit adaptive color expansion can be configured for 12 bpp, 15 bpp (B5G5R5A) and 16 bpp modes (B5G6R5) to convert them to 24 bpp. Alternatively the three 256 x 8bit color palettes can also be used to convert these modes to 24 bpp or be used to perform gamma correction. YUV/YCbCr data formats are converted to 24 bpp RGB and optionally passed through the three 256 x 8bit color palettes for gamma correction. The YUV-to-RGB color space conversion has programmable coefficients which can be programmed independently for each of window B and window C. However, these coefficients are shared for both Primary and Secondary display modes. Note that data for all three windows are converted to 24 bpp RGB prior to color keying and blending.

The Display module generates either all or most of the necessary functions to refresh the display from the display frame buffer. These include:

- Generating the horizontal and vertical timing signals for the required display device.
- Generating horizontal and vertical timing signals for the graphics image windows.
  - The graphics image resolution might be smaller than the display device resolution.
  - Right and bottom side clipping of the image window is implemented.

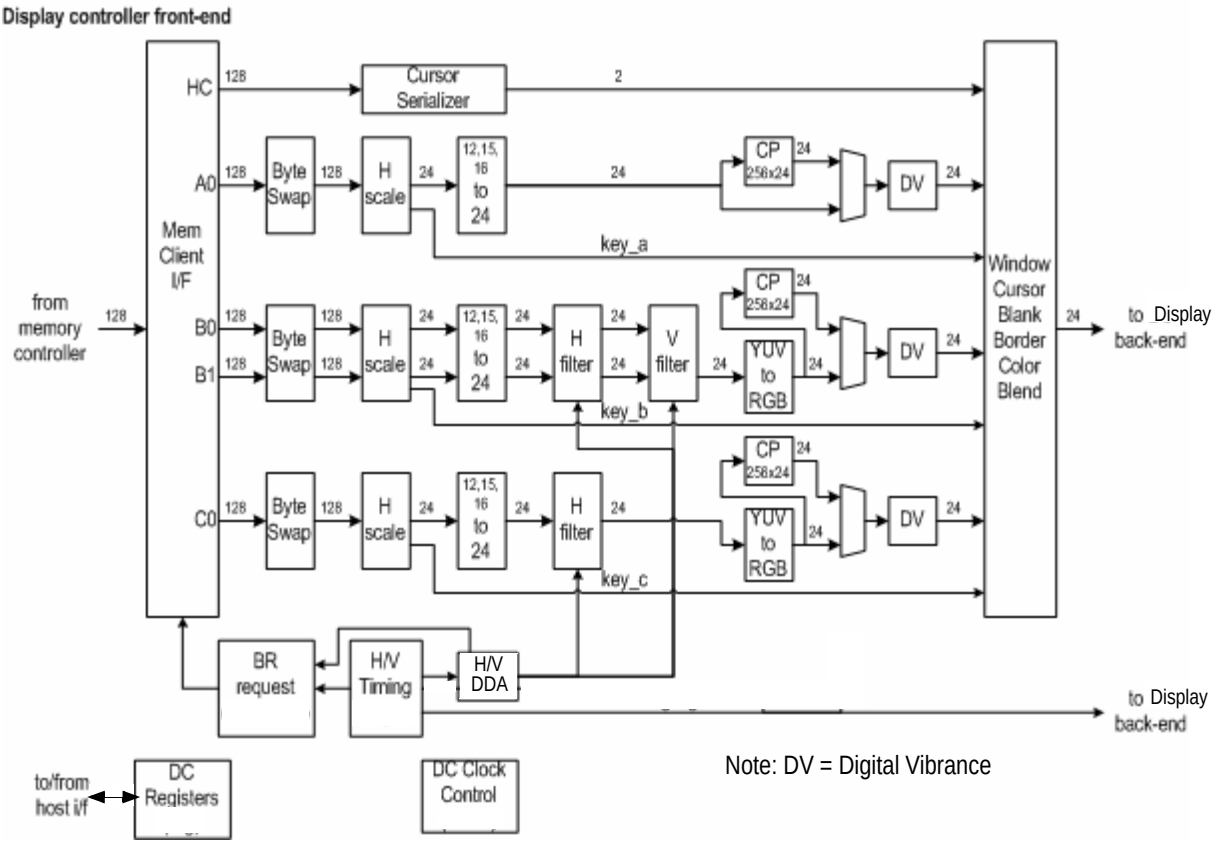
- Generating requests to the memory controller to fetch image lines, and controlling data fetch from the image FIFO.
  - The Display Module performs pixel data serialization.
  - The Display Module supports Odd-even byte and half-word swapping option for all modes; implements color palettes for Red, Green, and Blue color pixels in 8bpp, 4bpp, 2bpp, or 1bpp mode.
  - The Display Module implements 12bit, 15 bit, and 16bit-to-24bit color conversion for 12bpp, 15bpp, and 16bpp modes.

- Generating requests to the memory controller to fetch cursor lines and controlling the cursor position on the display.
  - The Display Module performs cursor data serialization.
  - The Display Module performs insertion of cursor colors in the active display area.

- Performing dynamic power management (software-controlled) to power down the data paths outside the image window and hardware cursor areas. Display Module Block Diagram

Figure 2.6 shows a block diagram of the Display module.

Figure 2.6: Display Module Block Diagram I: Front End



2.9.3 Display Module Functional Blocks

2.9.3.1 Output Window to EPP

The blended window and cursor can be optionally sent to the EPP for MPEG/JPEG encoding or unencoded screen capture. A programmable output window can be defined to crop the display area sent to EPP. Data to EPP is sent in 24-bpp (B8G8R8) format. Any required color space conversion and rotation is done on the EPP. An enable bit in the display module registers can enable/disable this interface. On power on reset, and when Display Module is disabled, this interface is disabled. The enable/disable bit is sampled at every display frame start. EPP should be programmed to receive data from display prior to enabling the display output to EPP.

A DDA-based counter is used to reduce the frame rate for outputting data to the EPP. A 13bit register is programmed with a 12bit fractional value representing the ratio of the frame rate on this interface to the frame rate of display. An internal accumulator is increased by this register value at the beginning of every frame. The frames which cause an overflow in the accumulator are sent out to the EPP on this interface. The other displayed frames are suppressed. The DDA counter allows the frame ratio to be controlled with an accuracy of $1/(2^{12})$. If the value of this register is programmed larger than or equal to 1.0 then every display frame is sent to EPP. This register is not double buffered and should be updated only when the interface is inactive. The DDA accumulator is reset when display is disabled or when this interface is disabled. The first frame encountered after this interface is enabled is always sent to EPP.

The Display output to the EPP interface may be enabled for a single frame (one-shot) or for continuous number of frames. A one-shot burst feature is available when only a single frame needs to be encoded. In continuous mode, frames are sent out to EPP continuously as specified by the frame reduction DDA counter.

2.9.3.2 One shot control

An one-shot burst feature is available on the output to EPP.

2.9.3.3 Color Key and Overlay Blend

When more than one active window overlapping, color key muxes select between blended window data and pure window data. There are 3 windows (A, B, C) and therefore there are potentially 3 regions (A, B, C) where there is no window overlap and 4 regions (AB, AC, BC, ABC) where there is window overlap.

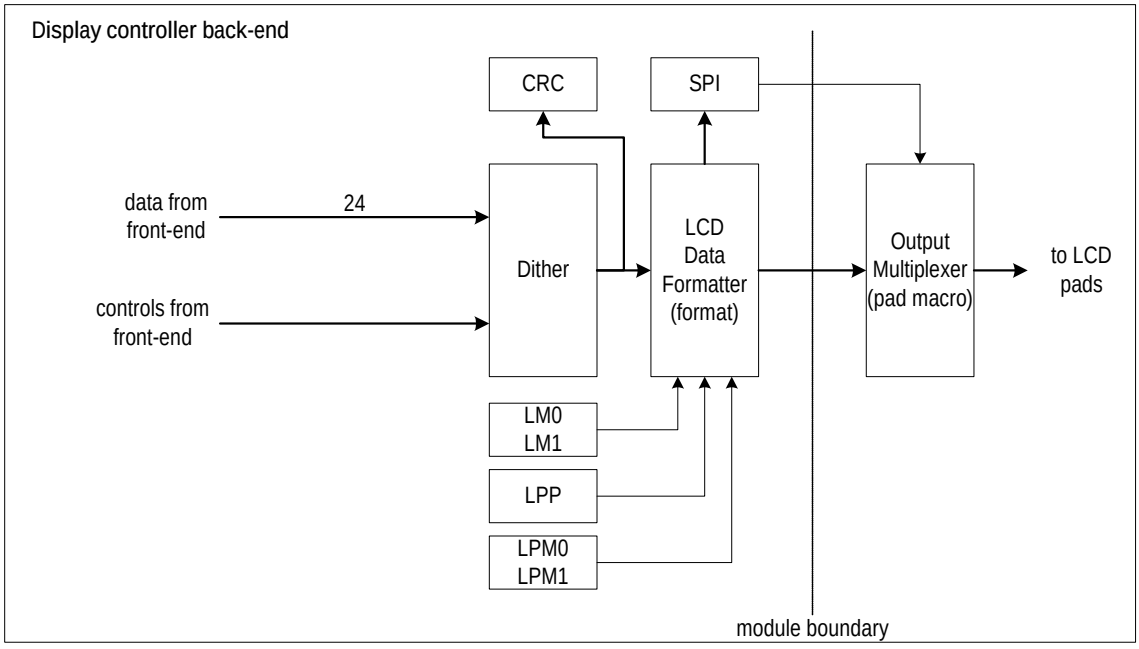
Blended data is the sum of the weighted active window data.

In any of the overlap region, color key can be defined in any of the overlapping window. Typically color key is enabled in one of the overlapping windows only. If color key is enabled in more than one of the overlapping windows then the color key multiplexer uses a priority encoder to select the window to use for color key compare. The order of priority is Window A, then B, then C when multiple windows are enabled for color key.

Data paths are provided for the graphics image windows A, B, and C. The horizontal/vertical timing generator are shared among the data paths. The window datapath and the cursor datapath are merged into a single output stream that goes to the Display module.

The output to the Display Interface is in 8-bit Red, 8-bit Green, and 8-bit Blue (RGB888) format.

Figure 2.7: Display Module Block Diagram II: Back End



2.9.3.4 Display Transformation

The Display is responsible for incrementing (and/or) decrementing x direction scanning, and incrementing (and/or) decrementing y-direction scanning. By itself, the Display is not capable of doing the line scanning in the y direction, which is needed for 90-degree or 270-degree display rotation. Other modules, (the 2D Engine, VS, VI, and EPP) are responsible for writing data in rotated form.

2.9.4 Display Interface to Host

- Synchronous register read/write and class writes
- Primary or Secondary display select and trigger
- Supports raise
- Host write to color palette A and/or B and/or C
  - no host read from color palette A, B, or C
- Interrupts
  - Vertical Active
  - Vertical Sync
  - Display FIFO underflow
  - SPI busy

Table 2.6 contains information about the Display interface for displays with parallel RGB interfaces and displays with low-voltage differential serial interfaces.

Table 2.6: Display Interface: Parallel RGB and Serial Interfaces

Pin Name	RGB 1pixel/clock 18 Bit[1]	Low Voltage Differential Serial I/F	24 Bit Interface (Two Configurations)	
			MSB	LSB
LD17	R5		R7	R1
LD16	R4		R6	R0
LD15	R3		R5	G7
LD14	R2		R4	G6
LD13	R1		R3	G5
LD12	R0		R2	G4
LD11	G5		G7	G3
LD10	G4	SD2 (D2+) [2]	G6	G2
LD9	G3	SD2_ (D2-) [2]	G5	G1
LD8	G2	STP	G4	G0
LD7	G1	SDT	G3	B7
LD6	G0	STH	G2	B6
LD5	B5	SD1 (D1+)	B7	B5
LD4	B4	SD1_ (D1-)	B6	B4
LD3	B3	SD0 (D0+)	B5	B3
LD2	B2	SD0_ (D0-)	B4	B2
LD1	B1	SC(CLK+)	B3	B1
LD0	B0	SC_ (CLK-)	B2	B0
LPW0	PW0		PW0	PW0
LPW1	PW1		PW1	PW1
LPW2	PW2		PW2	PW2
LSC0	SC0		SC0	SC0
LSC1	SC1/DE		SC1/DE	SC1/DE
LVS	V Sync		V Sync	V Sync
LHS	H Sync		H Sync	H Sync
LHP0	H Pulse 0		G1	R5
LHP1	H Pulse 1		B0	R2
LHP2	H Pulse 2		B1	R3
LVP0	V Pulse 0		V Pulse 0	V Pulse 0
LVP1	V Pulse 1		G0	R4
LM0	M0		M0	M0
LM1	M1		M1	M1
LDI	DI	SD2 (D2+)[2]	DI	DI
LPP	PP	SD2_(D2-)[2]	R1	R7
LSCK	SCK		SCK	SCK
LSDA	SDA		SDA	SDA
LCS_	SCS_		SCS_	SCS_
LDC	SDC		R0	R6
LSPI	SPI busy/DE		SPI busy/DE	SPI busy/DE

Notes on the parallel RGB LCD interface and low-voltage differential serial LCD interface:

1. The 18-bit 1-pixel/clock RGB parallel interface can be used when connecting directly to most TFT or PWM STN panels.
2. For a serial LCD interface, LSD2 and LSD2_ can optionally be output either on LD10 and LD9 pins or on LDI and LPP pins correspondingly.

Table 2.7 contains information on the Display Interface pins for displays with parallel host interfaces.

Table 2.7: Display Interface: Parallel Host Interfaces

Pin Name	1 Clock/Pixel, 18 Bit[1]	2 Clocks/Pixel, 18 Bit		2 Clocks/Pixel 16 Bit		3 Clocks/ 2 Pixels, 12 Bit			3 Clocks/Pixel, 18 bit[1]			1 Clock/ Pixel 24 bit
LD17	R5	R5	G2	R5	G2	R5	B5	G15	R5	G5	B5	R1
LD16	R4	R4	G1	R4	G1	R4	B4	G14	R4	G4	B4	R0
LD15	R3	R3	G0	R3	G0	R3	B3	G13	R3	G3	B3	G7
LD14	R2	R2	B5	R2	B5	R2	B2	G12	R2	G2	B2	G6
LD13	R1	R1	B4	R1	B4	G5	R15	B15	R1	G1	B1	G5
LD12	R0	R0	B3	G5	B3	G4	R14	B14	R0	G0	B0	G4
LD11	G5	G5	B2	G4	B2	G3	R13	B13				G3
LD10	G4	G4	B1	G3	B1	G2	R12	B12				G2
LD9	G3	G3	B0									G1
LD8	G2											G0
LD7	G1											B7
LD6	G0											B6
LD5	B5											B5
LD4	B4											B4
LD3	B3											B3
LD2	B2											B2
LD1	B1											B1
LD0	B0											B0
LPW0	PW0 or RST_ (active low reset) [2]											
LPW1	PW1 or RST_ (active low reset) [2]											
LPW2	PW2 or RST_ (active low reset) [2]											
LSC0	Primary display active low write pulse											SC0
LSC1	Secondary-display active low write pulse)											SC1
LVS	Primary/Secondary-display Data/Command											Vsync
LHS												Hsync
LHP0												R5
LHP1												R2
LHP2												R3
LVP0	Primary display active low chip select											V Pulse 0
LVP1	Secondary-display active low chip select											R4
LM0												M0
LM1												M1
LDI												DI
LPP												R7
LSCK												SCK
LSDA												SDA
LCS_												SCS
LDC	LSSF (Secondary-display start frame)											R6
LSPI	LMSF (Primary display start frame)											DE

Notes:

1. For 1-clock/pixel and 3-clock/pixel, data are MSB aligned for panels with less than 18-bits/pixel. For example, R0 and B0 are not output for 16-bit panels and R1-R0, G1-G0, and B1-B0 are not output for 12-bit panels.
2. RST_ is active low reset to the display. RST_ can be assigned to any of the unused signals.

Table 2.8: Parallel Host Interface (I/F) Displays, MSB Aligned, Display Data Pins

Pin Name	1 Clock/Pixel								2 Clocks/Pixel						3 Clocks/ 2 Pixels			3 Clocks/ Pixel		
	18b I/F	16b I/F	15b I/F	12b I/F	9b I/F	8b I/F	6b I/F	3b I/F	24b I/F		18b I/F		16b I/F		12b I/F			18b I/F		
LD17	R5	R4	R4	R3	R2	R2	R1	R0	R7	G3	R5	G2	R4	G2	R03	B03	G13	R5	G5	B5
LD16	R4	R3	R3	R2	R1	R1	R0		R6	G2	R4	G1	R3	G1	R02	B02	G12	R4	G4	B4
LD15	R3	R2	R2	R1	R0	R0			R5	G1	R3	G0	R2	G0	R01	B01	G11	R3	G3	B3
LD14	R2	R1	R1	R0					R4	G0	R2	B5	R1	B4	R00	B00	G10	R2	G2	B2
LD13	R1	R0	R0						R3	B7	R1	B4	R0	B3	G03	R13	B13	R1	G1	B1
LD12	R0								R2	B6	R0	B3	G5	B2	G02	R12	B12	R0	G0	B0
LD11	G5	G5	G4	G3	G2	G2	G1	G0	R1	B5	G5	B2	G4	B1	G01	R11	B11			
LD10	G4	G4	G3	G2	G1	G1	G0		R0	B4	G4	B1	G3	B0	G00	R10	B10			
LD9	G3	G3	G2	G1	G0	G0			G7	B3	G3	B0								
LD8	G2	G2	G1	G0					G6	B2										
LD7	G1	G1	G0						G5	B1										
LD6	G0	G0							G4	B0										
LD5	B5	B4	B4	B3	B2	B2	B1	B0												
LD4	B4	B3	B3	B2	B1	B1	B0													
LD3	B3	B2	B2	B1	B0	B0														
LD2	B2	B1	B1	B0																
LD1	B1	B0	B0																	
LD0	B0																			

Table 2.9: Parallel Host Interface (I/F) Displays, LSB Aligned, Display Data Pins

Pin Name	1 Clock/Pixel								2 Clocks/Pixel						3 Clocks/ 2 Pixels			3 Clocks/ Pixel		
	18b I/F	16b I/F	15b I/F	12b I/F	9b I/F	8b I/F	6b I/F	3b I/F	24b I/F		18b I/F		16b I/F		12b I/F			18b I/F		
LD17	R5																			
LD16	R4																			
LD15	R3	R4																		
LD14	R2	R3	R4																	
LD13	R1	R2	R3																	
LD12	R0	R1	R2																	
LD11	G5	R0	R1	R3					R7	G3										
LD10	G4	G5	R0	R2					R6	G2										
LD9	G3	G4	G4	R1					R5	G1										
LD8	G2	G3	G3	R0	R2				R4	G0	R5	G2								
LD7	G1	G2	G2	G3	R1	R2			R3	B7	R4	G1	R4	G2	R03	B03	G13			
LD6	G0	G1	G1	G2	R0	R1			R2	B6	R3	G0	R3	G1	R02	B02	G12			
LD5	B5	G0	G0	G1	G2	R0	R1		R1	B5	R2	B5	R2	G0	R01	B01	G11	R5	G5	B5
LD4	B4	B4	B4	G0	G1	G2	R0		R0	B4	R1	B4	R1	B4	R00	B00	G10	R4	G4	B4
LD3	B3	B3	B3	B3	G0	G1	G1		G7	B3	R0	B3	R0	B3	G03	R13	B13	R3	G3	B3
LD2	B2	B2	B2	B2	B2	G0	G0	R0	G6	B2	G5	B2	G5	B2	G02	R12	B12	R2	G2	B2
LD1	B1	B1	B1	B1	B1	B1	B1	G0	G5	B1	G4	B1	G4	B1	G01	R11	B11	R1	G1	B1
LD0	B0	B0	B0	B0	B0	B0	B0	B0	G4	B0	G3	B0	G3	B0	G00	R10	B10	R0	G0	B0

Notes on the parallel host LCD interface

1. For the 1 clock/pixel parallel interface, program the output selects for pins LD[17:0] to 0 for pins with active data. Otherwise, program the output selects for pins LD[17:0] to 2 for pins with active data.

2. Dither base color size specifies the number of bits/pixel going to the panel.

3. The 24bit interface can be used with the an external TV encoder or a TMDS transmitter.

4. For the 2 clock/pixel modes, an option is provided to swap data between odd and even clocks.

2.9.5 Pin Output Selection

Table 2.10 lists the GoForce 5500 Display pin output selection choices. The registers used in this selection listed in Chapter 7, *GoForce 5500 Micro-classes*, utilize three bits to select the output for each pin defined below. This table is repeated in Chapter 7 for convenience with Register *DC_COM_PIN_OUTPUT_SELECT0_0*.

Table 2.10: Pin Output Selection Options

Pad Name	0	1	2	3	4	5	6	7
	Output Signal	Output Signal	Output Signal	Output Signal	Output Signal	Output Signal	Output Signal	Output Signal
LD17	LD17	LD17 Out	LPD17	0	0	0	0	0
LD16	LD16	LD16 Out	LPD16	0	0	0	0	0
LD15	LD15	LD15 Out	LPD15	0	0	0	0	0
LD14	LD14	LD14 Out	LPD14	0	0	0	0	0
LD13	LD13	LD13 Out	LPD13	0	0	0	0	0
LD12	LD12	LD12 Out	LPD12	0	0	0	0	0
LD11	LD11	LD11 Out	LPD11	0	0	0	0	0
LD10	LD10	LD10 Out	LPD10	0	SD2	0	0	0
LD9	LD9	LD9 Out	LPD9	0	SD2_	0	0	0
LD8	LD8	LD8 Out	LPD8	0	STP	0	0	0
LD7	LD7	LD7 Out	LPD7	0	SDT	0	0	0
LD6	LD6	LD6Out	LPD6	0	STH	0	0	0
LD5	LD5	LD5 Out	LPD5	0	SD1	0	0	0
LD4	LD4	LD4 Out	LPD4	0	SD1_	0	0	0
LD3	LD3	LD3 Out	LPD3	0	SD0	0	0	0
LD2	LD2	LD2 Out	LPD2	0	SD0_	0	0	0
LD1	LD1	LD1 Out	LPD1	0	SC	0	0	0
LD0	LD0	LD0 Out	LPD0	0	SC_	0	0	0
LPW0	PW0	LPW0 Out	PW1	PM0	PW2	MD0	0	0
LPW1	PW1	LPW1 Out	PW2	PM1	PW3	MD1	0	0
LPW2	PW2	LPW2 Out	PW3	PM0	PW4	MD2	0	0
LSC0	SC0	LSC0 Out	0	0	0	0	0	0
LSC1	SC1	LSC1 Out	DE	0	0	0	0	0
LVS	Vsync	LVS Out	0	PM1	0	MD3	0	0
LHS	Hsync	LHS Out	0	PM0	0	MD2	0	0
LHP0	H Pulse 0	LHP0 Out	LD21	PM0	0	MD0	0	0
LHP1	H Pulse 1	LHP1 Out	LD18	PM1	0	MD1	0	0
LHP2	H Pulse 2	LHP2 Out	LD19	PM0	V Pulse 2	MD2	0	0
LVP0	V Pulse 0	LVP0 Out	0	PM0	0	MD3	0	0
LVP1	V Pulse 1	LVP1 Out	LD20	PM1	PW4	MD3	0	0
LM0	M0	LM0 Out	H Pulse 0	PM0	V Pulse 2	MD0	0	0
LM1	M1	LM1 Out	LD21	PM1	V Pulse 3	MD1	0	0
LDI	D1	LDI Out	LD22	PM0	Sub SCS_	MD2	0	0
LPP	PP	LPP Out	LD23	PM1	V Pulse 3	MD3	0	0
LSCK	SCK	LSCK Out	0	PM0	0	MD0	0	0
LSDA	SDA	LSDA Out	Sub SCS_	PM1	0	MD1	0	0
LCS_	Main SCS_	LCS_ Out	LD22	PM0	0	MD2	0	0
LDC	SDC	LDC Out	LD22	PM1	0	MD3	0	0
LSPI	SPI Busy	LSPI Out	DE	PM0	Pix Clk	MD0	0	0

Notes:
1. LD[23-0] contain pixel data for 1-pixel/1-clock parallel interface
2. LPD[17-0] contain pixel data for non 1-pixel/1-clock parallel interface
3. If output select is set to 1, then corresponding Pin Output Data register value is output (pin is used as general purpose output).

Note that for 24bit, the 6 pins LHP1, LHP2, LVP1, LM1, LDI, LPP are used for pixel data. Two options for aligning 24bit data exist.

P_DISP_DATA_ALIGNMENT:

init=0 Display Data Alignment: enum (MSB, LSB)

This is effective for parallel display data format and the associated Initialization Sequence (IS).

0 = Output data is MSB-aligned

For 1-pixel/1-clock parallel display the output data ordering is the same regardless of display Base Color Size. For 1-pixel/1-clock parallel display data alignment is optimized for 18bpp so the 24-bit data ordering is:

- LD[5:0] is blue data bits 7-2
- LD[11:6] is green data bits 7-2
- LD[17:12] is red data bits 7-2
- LD[19:18] is blue data bits 1-0
- LD[21:20] is green data bits 1-0
- LD[23:22] is red data bits 1-0

Note that LD18 to LD23 signals are multiplexed with control pins.

1 = Output data is LSB-aligned

For 1-pixel/1-clock parallel display the output data ordering is determined by display Base Color Size. For 1-pixel/1-clock parallel display data alignment is optimized for 24-bpp as follows:

- LD[7:0] is blue data bits 7-0
- LD[15:8] is green data bits 7-0
- LD[23:16] is red data bits 7-0

Note that LD18 to LD23 signals are multiplexed with control pins, as shown in Table 2.10.

2.10 JPEG Encoder

2.10.1 Introduction

The GoForce 5500 JPEG Encoder provides real-time baseline JPEG compression of video images produced by camera modules with resolutions up to 10 megapixels, with the following features:

- Input Command received from the VI, Host Interface, or EPP module.
- All compression steps (except some header insertion) performed by JPEG encoder hardware.
- Maximum operating clock frequencies
  - 144 MHz (1.0 V operation)
  - 208 MHz (1.2 V operation)

- Continuous JPEG encoding for images up to 5 MP
  - Provides fluid preview
  - Two or three images may be taken as a series

- Number of frames for continuous JPEG encoding is programmable
- Hardware DCT, RLE, and Huffman encoding
- Software-programmable Q-table values.
- Interrupt generation capability for JPEG Stream Write (non-circular) Buffer overflows.
- Incoming data received in YUV 4:2:0, YUV4:2:2, or YUV 4:2:2R format - no conversion necessary
- Ability to encode any window within a frame and to perform XY swapping
- Direct interface to write buffer through DMA and EPP to send encoded bit stream back to host CPU

The GoForce 5500 JPEG encoder hardware performs all compression steps including the end-of-file marker. Insertion of the interchange header is provided by the software drivers. External software provides the appropriate quantization tables.

The JPEG processor core accepts input data in YUV 4:2:0, YUV 4:2:2, and YUV 4:2:2R formats. The encoding process includes Forward Discrete Cosine Transform (FDCT), Forward Quantization, ZigZag (Run Length) encoding and Huffman encoding algorithms. The resulting encoded data is stored in memory.

2.10.2 Overview

Source data to the GoForce 5500 JPEG Encoder comes from four possible sources; the VI, EPP, or Host CPU Interface. The encoder can process data from cameras with resolutions up to 10 MP in YUV4:2:0, YUV4:2:2, and YUV4:2:2R. The source module sends input buffer index, input buffer ready, and input buffer frame start information to the JPEG Encoder, which triggers it to start the encoding process. Only one module at a time may be active on a given frame boundary. The Output DMA reads the data and sends it to the host CPU.

2.11 MPEG-4 Encoder

2.11.1 Introduction

The MPEG4 Encoder can encode D1 frames up to 30fps and handles up to simple profile level 5.

The MPEG-4 Codec module supports the following features:

- The input image is limited by the image dimensions
  - Maximum width: 720 Pixels
  - Maximum height: 576 Pixels
- Image data formats are 4:2:0.
  - 4:2:0 must be in planar format.
- 8Bit per component signal resolution.
- Maximum operating clock frequencies
  - 113 MHz (1.0 V operation)
  - 163 MHz (1.2 V operation)
- Motion compensation
- Synchronization of audio and video timing
  - Uses an internally generated or an externally supplied clock signal.
- Control Buffer for the following Host CPU-generated information
  - Frame Type (Intra-frame or Inter-frame)
  - AC/DC Prediction Flag
  - Quantization Scale and Style
  - Rate Control Flag and Target
  - Bypass information for Visual Layer Control (VLC) Flag and Reconstruction
  - Short Video Header Encoding
  - Frame addresses
  - Prediction values
- Motion estimation for H.264 Simple Profile
- Accurate time-stamp generation for synchronization with audio stream
- Direct interface to output (read) DMA or via internal AVP to send encoded bit stream back to host
- Byte swap options for output stream
- Simple profile L0-L5.
  - MPEG-4 (ISO/IEC 14496-2) Simple Profile at Level 3 (352x176, 30 fps, 384 Kbps)
  - H.263 Profile 0 (baseline) at Level 30 (352x176, 30fps, 384Kbps) compliant to 3GPP

2.12 Video Decoder

2.12.1 Introduction

The GoForce 5500 Video decoder handles both the JPEG and MPEG-4 decoding requirements.

2.12.2 MPEG Decode Overview

The MPEG-4 decoder handles up to simple profile level 5 (Intra (I) and Predicted (P) frames) and supports the following features:

- Approx 607.5 KB Maximum image size
- Image data format: YUV4:2:0 (planar)
- 8Bit per component signal resolution
- Maximum operating clock frequencies
  - 109 MHz (at 1.0 V operation)
  - 157 MHz (at 1.2 V operation)

- 128Bit memory data bus
- RGB 16bit, 565 output data format
- Bypass mode for direct connection of source image from Host CPU memory or AVP
- 18 image buffers for H.264-decoded data
- H.264 Simple Profile Levels 1 through 3
  - VLD on Host CPU for bit rates > 1 Mbps
  - 720x480, 30 fps
  - 720x576 (D1 Resolution) 25 fps

- H.264 Codec (simultaneous encode and decode)
  - QCIF, 15 fps (128 bps): VLD performed by GoForce 5500
  - QVGA, 15 fps (384bps) VLD performed by Host CPU

- MPEG-4 (ISO/IEC 14496-2) Simple Profile at Level 3
  - 352x176, 30 fps

- H.263 Profile 0 (baseline) at Level 30
  - 352x288 (CIF), 30 fps

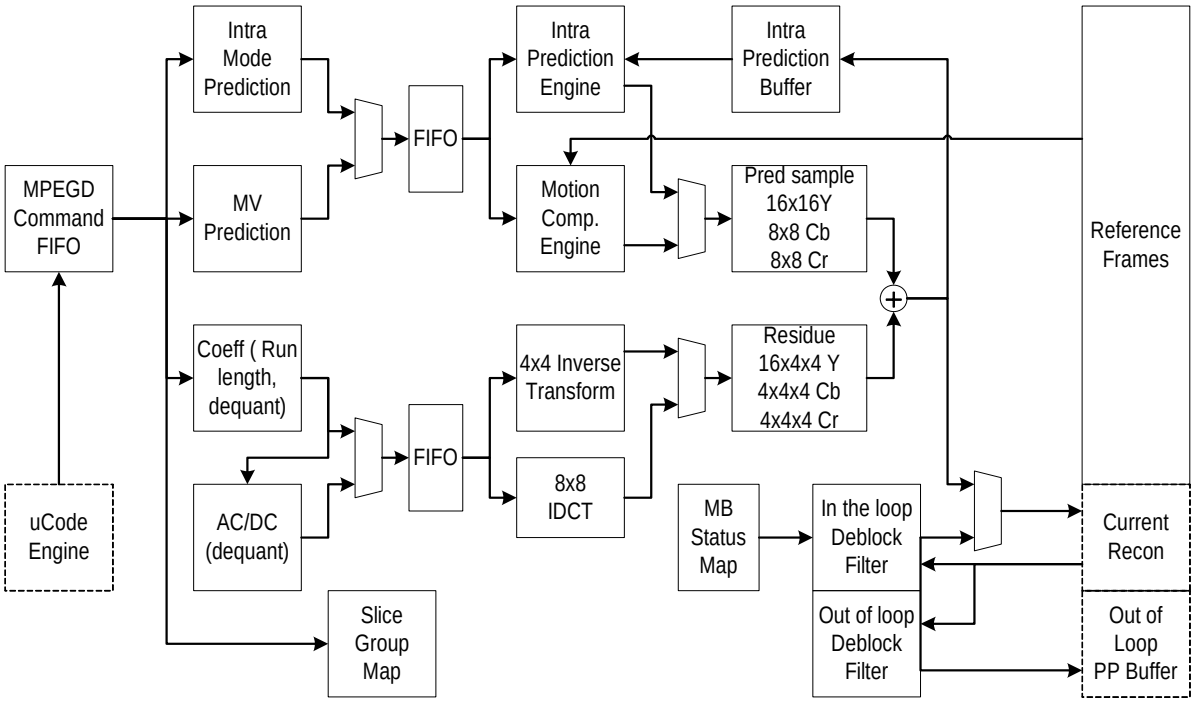
- WMV9 Decode
  - 320 x 240 (QVGA),25 fps, 384 kbps, comparable to simple profile medium level
  - QCIF, 15 fps, 96 kbps, full spec for simple profile, low level

- Bit stream variable length decoding (VLD) supported with internal AVP
  - 4 Mbps when AVP is dedicated 100% for performing VLD (MPEG-4)

- VLD may also be performed by host CPU
- In the loop de-blocking for H.264
- Out of the-loop de-blocking and de-ringing for MPEG-4 and H.263
- YUV4:2:0 output format with swap XY option for rotation
- Image scaling using 2D Engine Video Scaler function
- Color-space conversion using 2D Engine Video Scaler function or Display module
- Intra/Inter VOPs
- AC/DC prediction
- 4 motion vectors per macro block
- Short video header for baseline H.263 baseline
- Error resiliency: video packets, reversible VLC, header extension codes (HEC), data partitioning
- Output: YUV (YCbCr) 4:2:0 prior to video post processing

Note: The MPEG-4 Decoder does not support an X/Y swap function. However, an X/Y swap can be performed by the MPEG-4 Decoder's post-processor.

Figure 2.8: MPEG Decode Path



2.12.3 JPEG Decoder Overview

The JPEG Decoder utilizes the MPEG-4 Decoder hardware.

- Image data in 4:2:0 format (planar), YUV 4:2:2, YUV 4:4:4, YUV 4:2:2R
- 8Bit per component signal resolution.
- Maximum operating clock frequencies
  - 109 MHz (at 1.0 V operation)
  - 157 MHz (at 1.2 V operation)

- 128-Bit memory data bus
- RGB 16bit 565 output data format.
- Bypass mode to input source image directly from Host CPU memory
- YUV 4:2:0 (planar), YUV 4:2:2, YUV 4:4:4, YUV 4:2:2R input data format acceptable.
- Downscaling: 1/4 and 1/16
- Data path to Host Interface (4:2:0, 4:2:2, 4:2:2R formats)
- Bit stream variable length decoding (VLD) supported with internal AVP
- YUV4:2:0, YUV 4:2:2, YUV 4:2:2R output format with swap XY option for rotation
- Image scaling using 2D Engine Video Scaler function
- Color-space conversion using 2D Engine Video Scaler function or display
- Output: YUV (YCbCr) 4:2:0, 4:2:2, or 4:2:2R prior to video post processing

The MPEG and JPEG Decode modules share Input and Output FIFOs, registers, control state machines, and the 8 x 8 Inverse Discrete Cosine Transform (IDCT) function.

The GoForce 5500 always utilizes the software Huffman decode function; the Host CPU processes the frame header, and performs the Huffman decode; it feeds the Huffman-decoded JPEG data to the HW coefficient FIFO.

2.13 3D Graphics Engine

2.13.1 Introduction

The GoForce 5500 has a 3D Graphics Engine, based on OpenGL ES architecture. Features include:

- Transform, clipping and setup engine
- Floating-point and fixed-point input formats
- Support for the following read/write color formats as textures or as frame buffer data with high-quality dithering:
  - A8, I8, L8
  - L8A8
  - B2G3R3
  - B5G6R5, B5G5R5A1, B4G4R4A4
  - A1B5G5R5, A4B4G4R4
  - B8G8R8A8, A8B8G8R8
  - Z16

- Support for the following additional read-only texture formats:

CI4_L8A8,	CI8_L8A8
CI4_B5G6R5,	CI8_B5G6R5
CI4_B5G5R5A1,	CI8_B5G5R5A1
CI4_B4G4R4A4,	CI8_B4G4R4A4
CI4_A1B5G5R5,	CI8_A1B5G5R5
CI4_A4B4G4R4,	CI8_A4B4G4R4
DXT1, DXT1C,	DXT3, DXT5

- OpenGL alpha modes
- Fog
- Anti-aliasing
- Full per-pixel perspective-correct rendering
- 40-bit color pipeline with signed non-integer color (over bright)
- 7 surfaces: color, Z, and texture 1 through 5
- Programmable pixel shader
- Mip-mapping
- Bilinear/trilinear filtered texturing
- 4/8-bit palettized textures
- Multi-texture support (up to 5 simultaneous textures)
- At 200 MHz
  - 200 million pixels/sec
  - 2.67 million drawn triangles/sec

- Maximum operating frequencies
  - 144 MHz (at 1.0 V operation)
  - 208 MHz (at 1.2 V operation)

- Supports OpenGL ES with NVIDIA Pixel Shading Extensions

The 3D Graphics Engine is based on a traditional OpenGL architecture. All of the geometry and pixel processing are executed in hardware. The 3D Graphics Engine is compliant with OpenGL ES.

2.14 Embedded Memory

2.14.1 Introduction

The GoForce 5500 contains 640 KB of embedded memory. Use of the embedded memory depends on the operational mode.

2.14.2 Overview

Embedded memory is shared as follows. (Most of this data can also be stored in the additional 2MB, 8MB, or external memories if those options are chosen.)

- The GoForce 5500 supports three windows (display areas) for each display connected to it: window A, window B, and Window C.
  - The window sizes are programmable; they can be overlapped on the display.
  - Each window has its own buffer in the embedded memory.
  - Each window can be double-buffered.
    As long as there is enough memory, the GoForce 5500 can support two double-buffered windows on both the Primary-LCD and the Secondary-LCD. If there is not enough memory for two LCDs, window buffers can be shared.
- The captured preview video image is stored in memory in RGB format;
  - It is assigned to one of the windows (A or B).
  - If a captured image does not need to be previewed, it can be stored in the memory in YUV format so that it can be read by the host.

- Embedded memory is used for JPEG encoding.
  - JPEG processing buffers (holding 4:2:0 data) are stored in embedded memory.
  - Encoded JPEG stream data is stored in embedded memory; JPEG stream data can be stored in circular buffer fashion.
  - Ping-pong buffering is supported for continuous JPEG encoding.
    Continuous JPEG encoding requires the assigned buffers be non-circular, or continuous. Continuous JPEG encoding requires that the Host CPU be able to read the encoded data out until the encoding for the next (consecutive) frame is completed. The Host CPU must be able to read data out faster than the data comes in, as in burst mode.

2.15 Power Management

2.15.1 Introduction

The GoForce 5500 achieves power management through software module and clock enable controls, and dynamic power-down of modules.

2.15.2 Overview

The GoForce 5500 provides power-enable controls for functional modules and clocks. Software can disable all the unused modules and clocks. After power-up, the GoForce 5500 comes up with all the clocks and all of the disabled modules except for the Host Interface. Address input buffers are enabled only when the host selects the GoForce 5500 through Chip Select. The GoForce 5500 may be put in standby mode so long as the clocks are driven (high or low) and not floated.

The GoForce 5500 also supports dynamic power-down in operational mode. Data and address pipelines are enabled only if there are related activities.

2.15.2.1 Power Islands

The GoForce 5500's design includes power islands, which are power sources going to different groups of modules. By regulating the power to these islands, modules can be turned on and off as needed, enhancing power management of a system containing the GoForce 5500.

The power islands are grouped as follows:

- AOCVDD
  Power for Host Interface
  (Clock Generation, Internal and SDR/DDR Memory Controllers, Display, SD, Test Logic.)
- VECVDD
  Power for Core Logic
  (Powers 2D Engine and I^2S/AC'97)
- MMCVDD
  Power for Core SRAM
- TDCVDD
  Power for 3D Engine

Refer to the signals chapter to see which pins and GPIOs get their power from each power island.

Power Up/Down Sequence

A module's power-up sequence consists of

- turning on module power.
- de-asserting module reset.
- enabling the module clock.

A module's power-down sequence consists of

- disabling the module clock.
- asserting module reset.
- turning off the module power.

2.16 Clocks

2.16.1 Introduction

The GoForce 5500 clocking scheme is very flexible in order to support power management and an assortment of functions concurrently. The GoForce 5500 features independent clock generation for each module.

2.16.2 Overview

The clocks for the GoForce 5500 modules come from six clock sources:

- REFCLK0 (up to 200 MHz)
- REFCLK1 (up to 200 MHz)
- Internal Crystal oscillator (2 to 13 MHz) or external oscillator (100 MHz).
- Relaxation oscillator (ROSC) 10 to 25 MHz
- Two PLLs
  - Maximum VCO: 300 MHz at 1 V $\pm$ 10%
  - Maximum VCO: 664 MHz at 1.2 V $\pm$ 10%

Clock generation:

- External clock input (REFCLK0) or OSCFO
- 2 to 13 MHz built-in low-power crystal oscillator (with external crystal connected)
- Ultra-low power relaxation oscillator
- Programmable low-power PLL with 4-bit divider, 8-bit multiplier, and 50 to 664 MHz VCO
- Optional REFCLK1 clock input for VCO calibration (if needed) or for MPEG encoder time stamp generation

Clock Dividers:

- Even, odd, and half divide ratios (i.e. 1.0, 1.5, 2.0, 2.5, 3.0, and so on.)
- Dynamic divider ratio changes for some modules

Power Management through

- PLL enable
- Gated clocks to dividers
- Second-level clock gating
- Automatic power-down of unused pipelines
- Clock frequency scaling

Table 2.11: Per-module Maximum Clock Frequencies

Module	Maximum Frequency 1.0 V Operation	Maximum Frequency 1.2 V Operation
Host Interface	117 MHz	175 MHz
Audio Video Processor (AVP)	129 MHz	188 MHz
Internal Memory Controller	145 MHz	212 MHz
External Memory Controller	145 MHz	212 MHz*
2D Engine	144 MHz	208 MHz
Video Input (VI)	79 MHz	105 MHz
Image Signal Processor (ISP)	79 MHz	105 MHz
Encoder Pre-processor (EPP)	112 MHz	162 MHz
Display Controller	83 MHz	120 MHz
JPEG Encoder	144 MHz	208 MHz
MPEG-4 Encoder	113 MHz	163 MHz
Video Decoder	109 MHz	157 MHz
3D Graphics Engine	144 MHz	208 MHz
SDIO (Secure Digital IO) Interface Host		
Serial Peripheral Bus (SPB)		
I2S and AC'97 Codec Interface	20 MHz	20 MHz

* The maximum EMC frequency depends on the SDRAM.

Figure 2.9: PLL Clock Generation

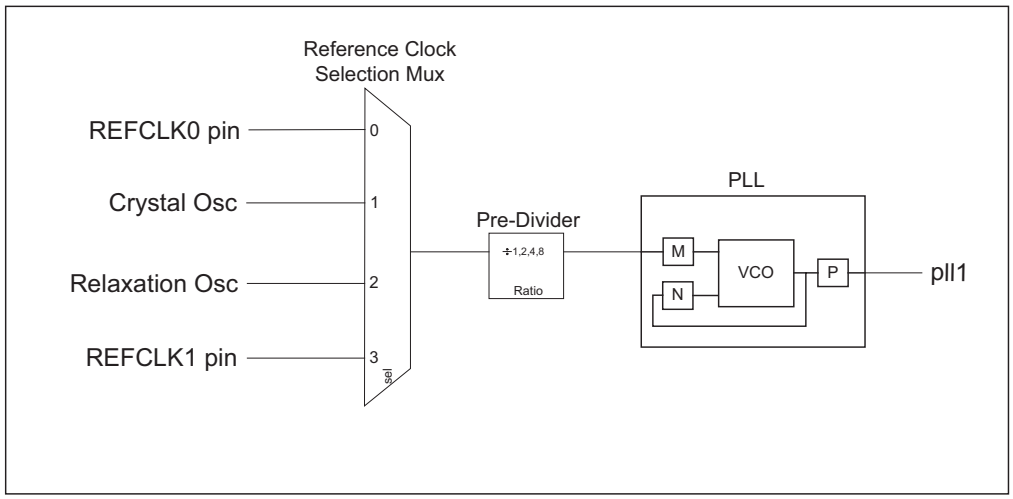
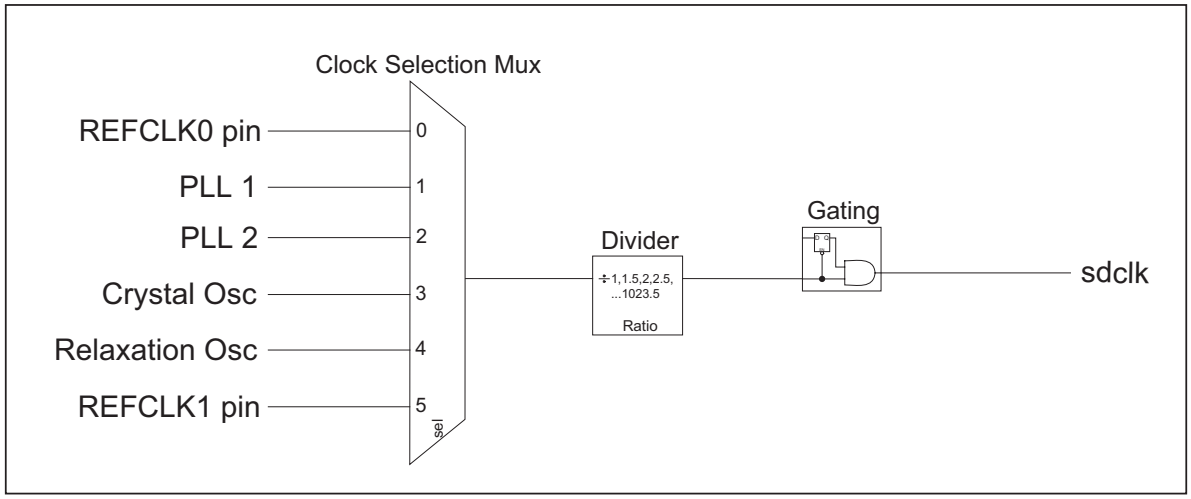


Figure 2.10 depicts a typical module's clock generation; in this case the SD Module. Each module's clock is derived similarly, and separately.

Figure 2.10: Per-module Clock Generation: SD Module Example



Divider ratios may be changed dynamically for some modules; statically for others. For static ratio dividers, GFSDK does the following:

1. Disable the clock
2. Wait for the clock to stop
3. Change the ratio
4. Enable the clock

To dynamically change the clock ratios, GFSDK simply programs the appropriate register to change. The clock control registers are in the host async register set and configure the clocks to be enabled, configure them to be inverted, choose the clock sources, and choose the divider values, all as required.

Modules with dynamic ratio dividers

- Display
- EMC
- Host Interface
- MC
- VI

2.16.3 Relaxation Oscillator

The GoForce 5500 generates a very low power relaxation oscillator, typically used during low-power modes to drive display refresh and other critical functions. The Relaxation Oscillator can be selected as the clock source to many GoForce 5500 functional modules.

The frequency of the relaxation oscillator is selected by software and by choosing the value of an external resistor connected between the OSCR input and AVDD. It is programmable within the range of 10 MHz to 25 MHz. Its exact frequency is approximate and can vary +/- 20% from chip to chip. (The tolerance range does not indicate a frequency tolerance range in a specific circuit; the variation is in the frequency from one chip to another chip, as well as across temperature and voltage ranges.) It requires a 200 kΩ external resistor (1% tolerance), placed between the OSCR input and CVDD.

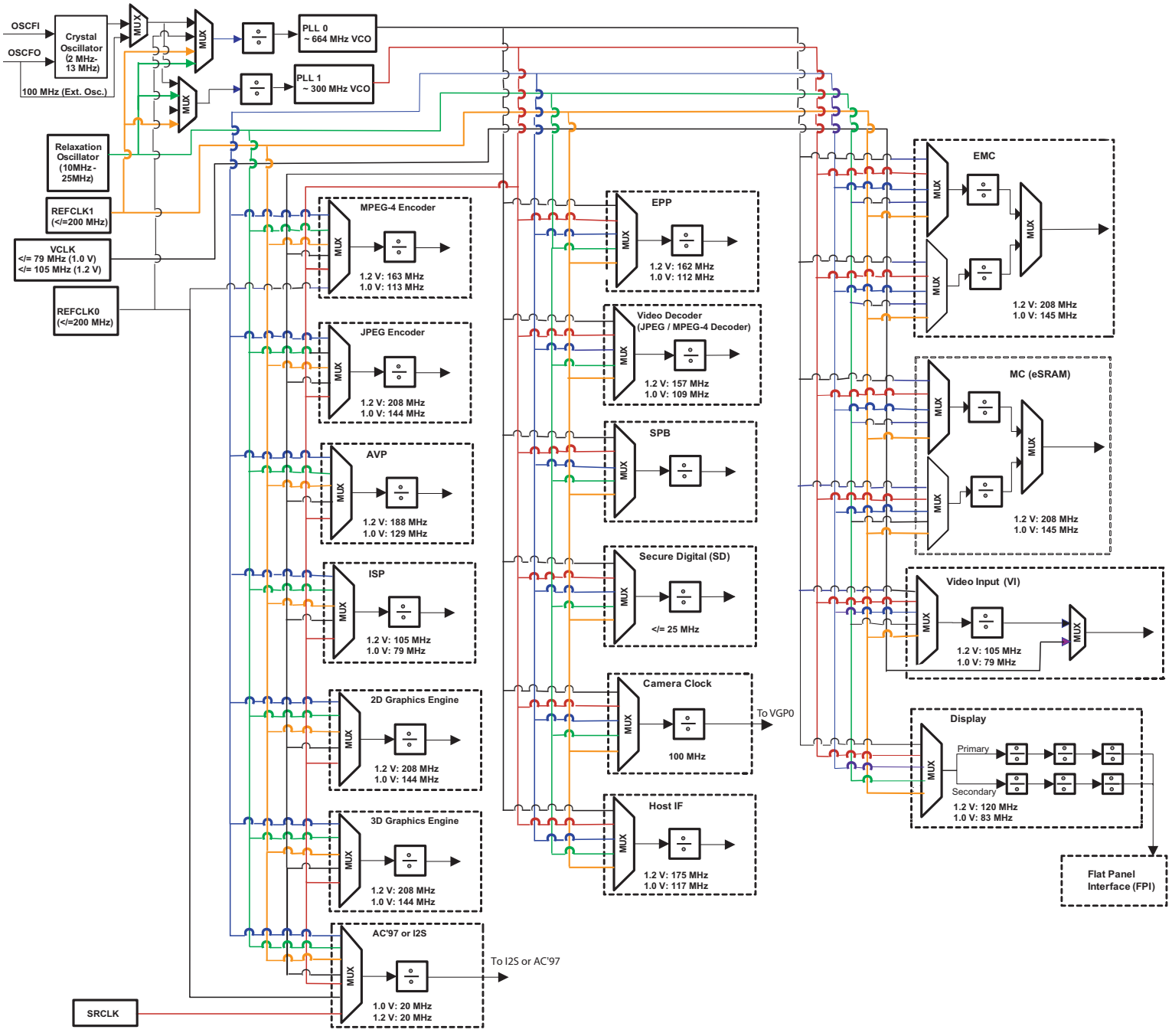
The Relaxation Oscillator can save anywhere from 100 to 400 μA, compared to using the Crystal Oscillator output as the primary clock source. The Relaxation Oscillator should not be used as the source for the camera.

GFSDK selects the frequencies for ROSC by programming the GoForce 5500; the values available are from 10 MHz to 25 MHz in 1 MHz increments (i.e. 10 Mhz, 11 MHz, and so on.)

2.16.3.1 Clock Distribution

The GoForce 5500 internal clock distribution network is very flexible. Many functional blocks can select their clocks from a number of different sources. Management of clocks is an important aspect of GoForce 5500 power management. Designers can trade off between power consumption and required performance on a block-by-block basis. In addition, entire functional blocks can be completely disabled when they are not used for further power savings.

Figure 2.11: GoForce 5500 Clock Distribution



2.16.4 PLL Frequency Calculation

The PLL output frequency, F_o, is determined by the values set in the M, N, and P counters. The PLL output frequency (F_o) is calculated from:

$$F_o = (Fr*N)/(PREDIV*M*2^P)$$

The values for N, M, PREDIV, and P are all contained in the registers HOST1X_ASYNC_PLL1CONFIG2_0 and HOST1X_ASYNC_PLL2CONFIG2_0, in *Chapter 7*.

Note that Fo is the output frequency and Fr is the reference frequency.

Table 2.12: PLL Frequency Calculation Parameter Constraints

Parameter	Definition	Notes
Fr	Base, or reference, frequency	
Fi	PLL Input clock frequency (2 MHz to 6 MHz) Fi = Fr/(PREDIV*M)	
Fo	PLL Output frequency (20 MHz to 500 MHz) $F_o = (Fr*N)/(PREDIV*M*2^P)$	
M[2:0]	000: Not allowed 001: Divide by 1 through 111: Divide by 7 (000 is not a legal value)	Only specific values are legal for M
N[9:0]	10'h000 is not legal Legal values: 10'h001 (divide by 1) through 10'h3ff (divide by 1023)	Only specific values are legal for N
P[2:0]	000 through 111: 000 (Divide by 2**0) = 1 001 (Divide by 2**1) = 2 010 (DIvide by 2**2) = 4 through 111 (Divide by 2**7) = 128	

2.17 SDIO (Secure Digital IO) Interface Host

The registers which configure the SD Module can be found in *7.6*, "SD Registers" in Section *2.17*, "SDIO (Secure Digital IO) Interface Host"

2.17.1 Introduction

The GoForce 5500 interfaces to SD/MMC compliant cards. The Secure Digital Interface (SDIO) Host works in single data pin mode and four-pin mode. Use the four-pin mode to achieve higher data rates (up to 10 Mbytes/sec.) Data can be stored in the secured and the non-secured areas of the Secure Digital (SD) card. The secured area is used primarily for such purposes as saving copyrighted data, like MP3 music. Since an external decoder is necessary to decode such data, applications can use the same decoder to perform encryption and decryption of the secured data.

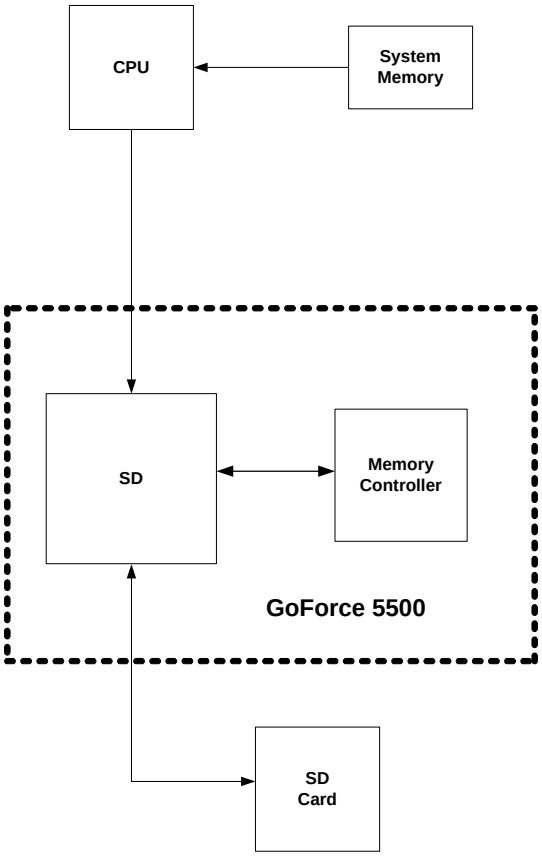
- DMA transfer to/from memory

2.17.2 Overview

Figure 2.12 is a block diagram of the SD Module with a Host CPU write. When the Host CPU Writes to the SD Card the following occurs:

- The Host CPU reads data out of system memory and writes the data into the GoForce 5500.

Figure 2.12: CPU Writes to SD Card



2.17.3 SD Functional Blocks

2.17.3.1 Pull-up and Pull-down Resistors for CMD/DATA Lines

The DAT3 line may be used to detect hot card insertion. To use the DAT3 line for detected hot-insertion, a pull-*down* resistor should be used on DAT3. (Normally a pull-*up* resistor is used with DAT3.)

Pull-up resistors protect the CMD and DAT lines from floating when there is no card connected or when all card drivers are in high-impedance mode. Recommended values for the pull up resistors are 10 kΩ minimum and 100 kΩ maximum. These pull-up resistors can be supplied externally, or can be selected by software.

Note: The pull-up function must be programmed on the SDCLK right after system reset, and disabled right after card detection.

Note: The DAT3 line in the SD card typically has a 50 kΩ pull-up resistor during card insertion. This resistor must be disconnected before starting data transfers to and from the card.

2.17.4 SD Host Transfers

The SD Host Interface enables the transfer of data between the GoForce 5500 and the SD Card; data transfer between the host and SD card is driven by a Command/Response interface. SD card insertion is recognized by the SD module, which then initiates an interrupt to the Host Driver. The Host Driver sends commands to the SD card to read the card's internal registers, which provide all possible operating conditions. The host starts the actual data transfers.

Note: All application-specific commands must be preceded with the Command 55 (ACMD.)

2.17.5 SD Module Interfaces

2.17.5.1 Command Transfers

The command and response interface between the host driver and the SD Host Module occurs through registers and a 4 x 32 Response FIFO. To send a command to the SD card the Host Driver:

- Enables the necessary interrupt masks.
- Programs the Time out function.
- Programs the Command Argument parameters.
- Programs the Command number and all the Command parameters, which triggers the command transfers on the SD CMD pin.)

2.17.5.2 Data Transfers

The Host Driver reads and writes data to and from the SD card through the internal SRAM. The SD Host interfaces to the internal SRAM through read buffer and write buffer clients. The driver negotiates the block length with the SD card. This value will be used in all subsequent transfers. Block length for multi-block transfers, which are preferred for transferring large amounts of data, should always be a multiple of 8 bytes. If the transfer size is not a multiple of 8 bytes, break the transfer into multi-block transfers until the nearest 64-bit boundary. Then reprogram the block length to transfer the last set of data in single-block mode.

2.17.5.3 Transmit (Write) Operation

To start a write operation the Host Driver:

- Sets the Block Length.
- Programs the Start and number of buffers.
- Programs (enables) the Transmit DMA Control and sets the ownership of the buffer to the SD Host Module.
- Programs the command arguments with the start address of the destination memory in the SD Card.
- Programs the transfer command (single-block write or multi-block write) number and the command parameters (write operation).

When a buffer is full the Host driver is interrupted and ownership is returned to it. The process is continued until either the end of the last buffer is reached or the stop transmission command is issued to the SD card.

In case of write errors or CRC errors the data transfer stops and the Status Register gets updated. The driver tracks of the number of correctly-programmed blocks by issuing the command ACMD22 or by counting the number of buffers transmitting.

2.17.5.4 Receive (Read) Operation

To start a read operation the Host Driver:

- Sets the Block length.
- Programs the Start and End addresses of the buffers.
- Programs (enables) the Receive MDA Control function and sets the ownership of the buffer to the SD Host Module.
- Programs the command arguments with the start address of the destination memory in the SD Card.
- Programs the transfer command (single-block read or multi-block read) number and the command parameters (read operation).

The process is continued until either the last buffer is reached or the stop transmission command is issued to the SD card.

In case of read errors or CRC errors the data transfer is stopped and the Status Register is updated.

2.17.6 SD Error Recovery

During multi-block transfers from the SD Host Module to the SD Card, CRC errors or programming in the card cause the SD Host Module to stop further writes to the card. The SD Host Module updates the Error Status in SD07 and resets the transmit FIFO and the Transmit Module in the MIU. The next transfer starts from the last block which had the error.

During multi-block transfers from the SD Card to the SD Host Module, CRC errors in the received block cause the SD Host Module to ignore all future data. It resets the FIFO and the MIU Receive Block, and updates status in SD07. A stop command gets issued to the SD Card and the next transfer starts from the last block with the error.

Note: Do not disable the SD module when stopping the clock between commands to and from the SD card, Disable the SD module only if there are no more commands to or from the SD card. Follow this sequence for enabling or disabling.

Disable

1.Disable the SD Clock.
2.Disable the SD Module.

Enable

1.Enable the SD Module
2.Check for the clock status (running or not).
3.Issue command to start clock if needed.

2.18 Serial Peripheral Bus (SPB)

The registers for configuring the SPB can be found in *Chapter 7*, "GoForce 5500 Microclasses" in Section *7.4*, "SPB Registers".

2.18.1 Introduction

The Serial Peripheral Bus (SPB) provides convenient communication among system components using a simple two-wire interface. The GoForce 5500 SPB implementation relies on hardware for control and data transfer. Standard memory mapped input/output transfers data to and from the SPB master. Interrupts are provided for programmed input and output. The SPB data transfer rate ranges from 100 kbps to 400 kbps, depending on the mode of operation.

The GoForce 5500 SPB is a pure master device; it is not designed to be a slave device.

The SPB supports:

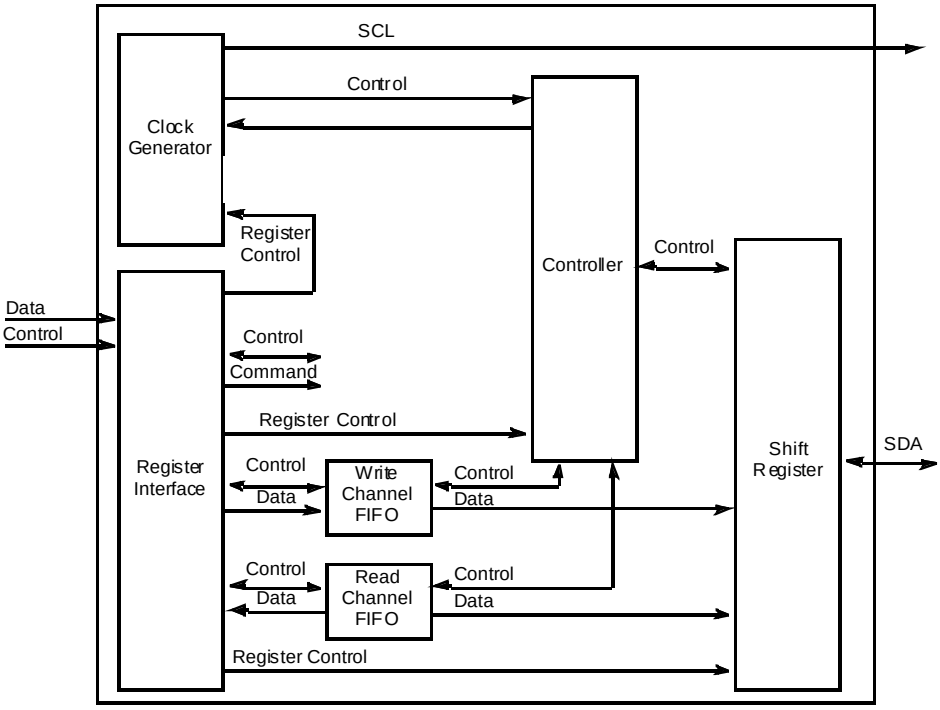
- 7-bit addressing.
- 10-bit addressing.
- Combined 7bit and 10bit addressing.
- Programmable transfer count for transmit and receive.
- Clock synchronization (for multi-master environments).
- Arbitration (for multi-master environments).
- Operation in Standard Mode (100 Kb/second).
- Operation in Fast Mode (400 Kb/second).

The SPB does not support CBUS.

2.18.2 Overview

Figure 2.13 shows the SPB basic architecture. Writes to the register interface initiate transfers. The data FIFOs are accessed from a memory-mapped port. All data is written to the transmit FIFO and read from the receive FIFO.

Figure 2.13: SPB Block Diagram



2.18.3 SPB Functional Blocks

The SPB is a serial, two-wire (plus ground), bi-directional data transfer protocol for communicating between integrated chips within a system. The bus consists of one data line and one clock line. The protocol makes extensive use of the wired-AND function of multiple bus drivers for clock synchronization, arbitration, and acknowledgement.

Between standard and fast mode, the interface is speed-adaptive, and transfers occur based on the speed of the target device.

The SPB supports 7-bit and 10-bit interchangeable addressing. Electrically, the bus uses pull-up resistors to achieve the logical high state and active pull-down circuitry to drive the bus to a logic low. It is friendly to process and logic-family variations because system-level solutions easily implement level shifters to isolate sections.

Figure 2.14: SPB Data Transfer

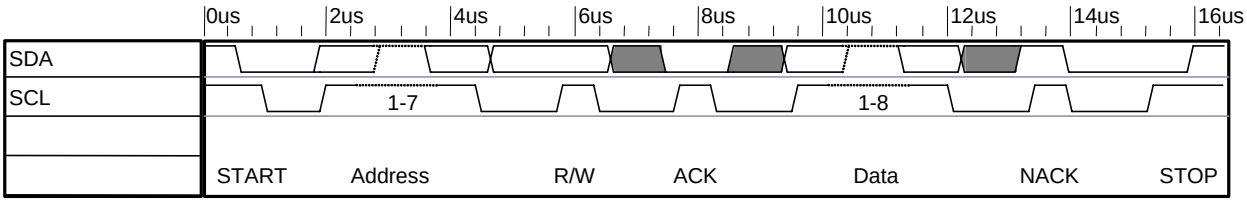


Figure 2.14 shows a typical data transaction. Transfer begins with the transmission of a start condition. Then a seven-bit address is presented, followed by a read/write bit, and the target device acknowledge (ACK). If the master sees a valid acknowledge, data is transferred. A not-acknowledge (NACK) causes a stop condition.

SPB devices fall into one of two categories: master or slave. Master devices initiate and control the data transfer. Slave devices operate on commands received from the master. Master devices sometimes contain logical slave devices. During a write, slave devices are addressed and respond as slave-receivers; during a read they act as slave-transmitters. If a slave does not respond to a master request and does not send and acknowledge, the master discontinues the transaction and takes other actions.

Multiple masters often are on the same physical SPB. Because of this, it is possible for collisions to occur when more than one master initiates a transfer at the same time. The SPB protocol handles this through an arbitration scheme based on the wire AND function. The arbitration results in no loss of data or retry, and the losing master shuts off its output driver when the SPB data does not match its own.

For two masters to carry out arbitration, the SPB protocol provides for clock synchronization. Competing masters drive the clock line; the master with the slow clock determines the low time, and the master with the fast clock determines the high time. Synchronization occurs when each master resets its internal clock generator as the SCL line goes low.

2.18.4 Clocks

All the clock functions are centralized in the Host Interface Module.

2.19 I²S and AC'97 Codec Interface

2.19.1 Introduction

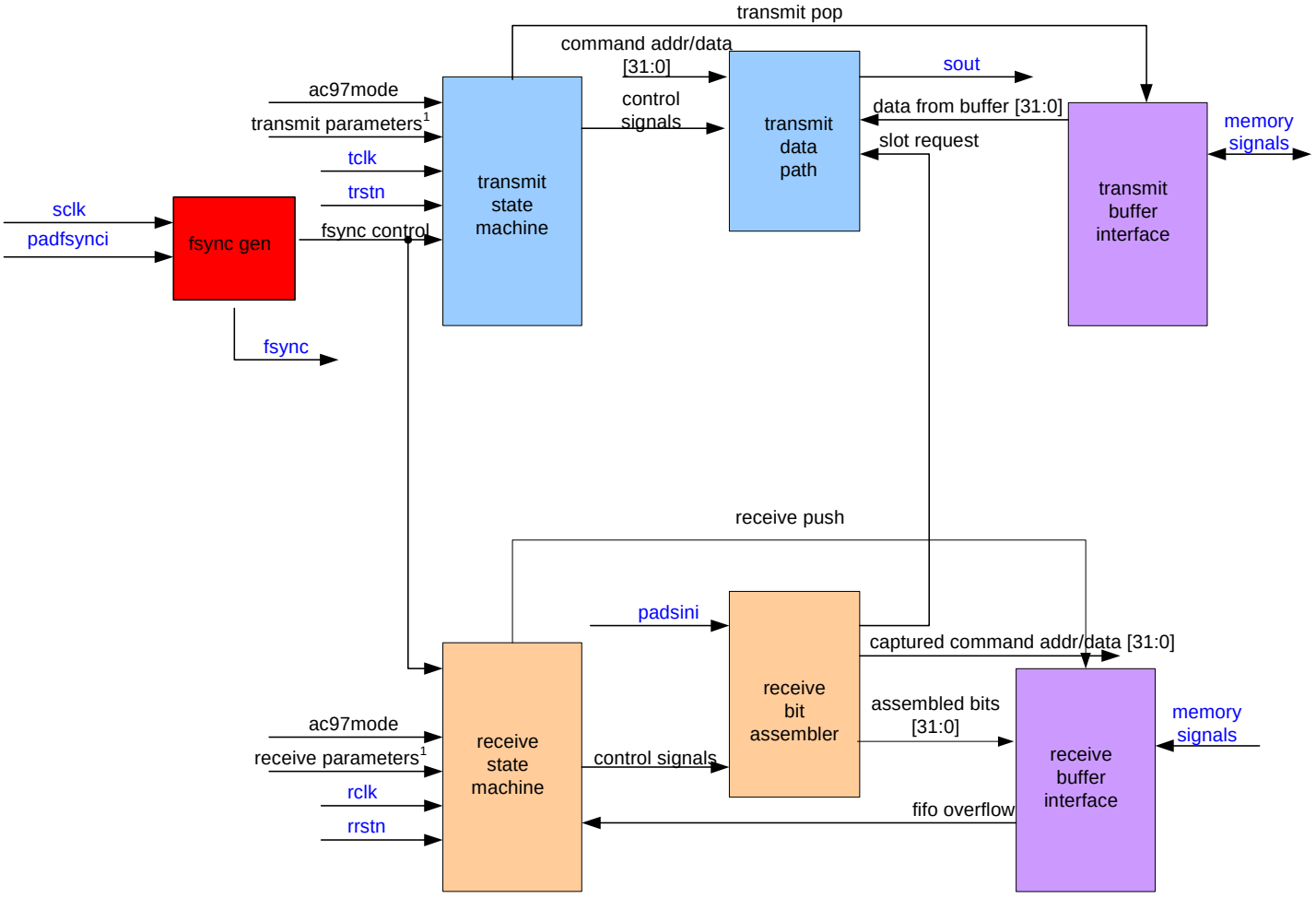
The I²S and AC'97 Codec Interface features the following:

- DMA Transfer to/from memory
- Full-duplex synchronous serial channel interfacing to an external codec.
- Support for AC'97 and non-AC'97 (i.e. I2S and other) data formats.
- Programmable FSYNC and SCLK polarity, stop value, direction
- FSYNC dividers up to 256
- Support for different FSYNC pulse types
- AC97 Features for AC'97 formats
  - Programmable transfer sizes: 8, 16, 18, 20bits
  - Stereo/Mono mode
    - Receive mode allows selection of which data to be kept.
  - Slot select (Left and Right data in slots 3 and 4, 5 and 6, 7 and 8, or 6 and 9.)
  - Variable Sample Rate (for data frame rates less than 48 kHz)
  - Character Time-out Frame Count
  - Command Address/Data write and read
- I²S Features
  - Programmable transfer sizes: 8, 16, 18, 20, 24, 32bits
  - Stereo or Mono mode
  - Master or slave word select (FSYNC)
  - Master or slave serial clock (SCLK)
  - 5 different data formats
  - Transfer rate control: Same audio sample repeated for 1, 2, 4, or 6 frames
  - Transmit Data padding
  - Command Address/Data write and read
- Transmitter (for playback) gets data from memory, can be written by Host Interface or AVP.
- Receiver (for recording) writes data to memory, can be read by Host Interface or AVP.
- Typical frame frequencies are 48 kHz (AC'97 standard) and 44.1 kHz.
- Bit clock rate up to 256 x frame rate (12.288 MHz).
- Non-I²S Features
  - Supports multiple FSYNC types:
    - Even duty cycle
    - Single short pulse per frame
    - Two short pulses per frame
  - Flexible data formats:
    - One sample on FSYNC active edge
    - Two samples on FSYNC active edge
    - One sample on each FSYNC edge
    - Two samples on each FSYNC edge
  - Programmable polarity and stop value for FSYNC and SCLK
  - Positive or negative edge sampling of FSYNC and SIN
  - Can transmit and receive serial data on same clock as FSYNC edge

2.19.2 Overview

The diagram below shows the transmit and receive control signals and data paths. Frame sync control signals are inputs to both the transmitter and receiver. Since the transmitter and receiver can operate independently of each other, each has a clock and reset. The transmitter and receiver interfaces with the memory controller via the buffer interface. When the memory read buffer is ready, the transmitter fetches the data from the memory and serializes it to the output pin, Sout. The receiver gets serial data from the input pin, Sin, and writes the data to memory via the memory write buffer.

Figure 2.15: Top Block Diagram



2.19.3 I2S Timing

The I2S bit clock SCLK, can be configured as master (the GoForce 5500 drives SCLK to the other device) or as slave (other device(2) drive SCLK to the GoForce 5500). The timing requirements for each of these configurations is given in the tables below. Values are in nanoseconds unless otherwise stated.

The setup and hold delays are defined below and are typically met by transmitting on one edge of the clock and receiving on the other edge.

When the GoForce 5500 drives SCLK (master mode) the timings in refer to the rising or falling edge of SCLK. In register HOST1X_ASYNC_ISCCONFIG_0, when SCLK_POL = 0, timing is based on SCLK's rising edge; if the value is 1 then timings are based on SCLK's falling edge.

Table 2.13: SCLK Timings: Master Mode (SCLK Driven by GoForce 5500)

Signal	1.0 V		1.2 V		Conditions
	Setup (ns)	Hold (ns)	Setup (ns)	Hold (ns)	
SIN	3.03	2.09	2.39	1.34	SERIN_SRC_SEL
SIN	-0.06	2.86	0.07	2.16	SERIN_SRC_SEL=SIN_FALL_EDGE
SFSYNC (slave)	6.48	2.76	4.69	1.82	FSYNCIN_SAMPLED=FSYNC_RISING_EDGE
SFSYNC (slave)	-0.15	3.17	-0.01	2.34	FSYNCIN_SAMPLED=FSYNC_FALL_EDGE
	Minimum	**Maximum**	**Minimum**	**Maximum**	
SOUT	-2.46	10.22	-2.65	8.25	
SFSYCNC (master)	-2.70	12.36	-2.82	9.59	

Table 2.14: SCLK Timings: Slave Mode (SCLK Driven by Codec)

Signal	1.0 V		1.2 V		Conditions
	Setup (ns)	Hold (ns)	Setup (ns)	Hold (ns)	
SIN	1.38	6.79	0.94	5.05	SERIN_SRC_SEL=SIN_PAD
SIN	-1.96	6.82	-1.58	5.25	SERIN_SRC_SEL=SIN_FALL_EDGE
SFSYNC (slave)	2.44	7.10	1.63	5.25	FSYNC_RISING_EDGE
SFSYNC (slave)	-2.05	7.13	-1.66	5.44	FSYNCIN_SAMPLED=FSYNC_FALL_EDGE
	Minimum	**Maximum**	**Minimum**	**Maximum**	
SOUT	6.71	21.80	5.18	17.29	
SFSYCNC (master)	6.05	23.98	5.10	18.62	

This page left blank intentionally.

Chapter 3 Signals

3.1 Introduction

This chapter provides a description of the GoForce 5500 signal pins by first summarizing the pin types and conventions and then grouping (and describing) the signals by module or functionality. Table 3.1 lists the package configurations.

As good design practice, ground any unused IO power pads. If this is not possible, then it is acceptable to leave them floating.

Note: The GoForce 5500 is still in stages of design.

Table 3.1: Package Configuration Options

Package Configuration	Host Interface	Internal Memory	Stacked Memory
2	32 bit	640 kB	2 MB
3	32 bit	640 kB	8 MB

3.2 Pin Types and Conventions Used

Table 3.2 provides notations that indicate valid pin types.

Table 3.2: GoForce 5500 Pin Types

Pin Type	Pin Type Description
I	Digital Input Pin.
IS	Schmitt-Trigger CMOS Input Pin.
O	Digital Output Pin.
OD	Open-Drain Output
I / O	Bi-Directional CMOS Input / Output Pin.
IS / O	Bi-Directional CMOS Input / Output Pin with Schmitt-Trigger CMOS Input Pin.
A	Analog Pin.
AI	Analog Input Pin.
AO	Analog Output Pin.
P	Digital Power Pin.
G	Digital Ground Pin.
AP	Analog Power Pin
AG	Analog Ground Pin

The following conventions are used:

- Inputs and outputs are all CMOS buffers.
- The symbol "_" at the end of a pin name indicates an active low signal (e.g. RST_.)
- Digital input buffer is a Schmidt trigger CMOS input buffer with input disable capability.
- Digital output buffer has programmable drive strengths
- Output buffer drive strength is specified in mA for operation at 3.3V.

3.3 Power and Ground Pins

The GoForce 5500 utilizes four power “islands”, or power planes, to supply the core voltages. Since each of these may be shut down to conserve power, you must know which of the islands supplies power to a specific module before shutting that island off, since turning off one unused module’s power might also turn off the power to another module you want to use. The six host bus interfaces are not affected by this.

Table 3.3: GoForce 5500 Power Islands

Name	Modules Powered	Notes: Requirements
AOCVDD	All clock cores	AOCVDD must always be powered on if any IO power is needed. Unused IO power may be floated. Exception: AOCVDD can be off and HVDD can be on if DPD_ is asserted.
	Host Interface	
	Graphics and Display controllers	
	Memory controller (Internal and External)	
	IO Controls	
	SD	
VECVDD	Video Codec (DSP/AVP)	VECVDD must be powered on when any of the modules receiving power from it are in use.
	Camera interface	
	ISP	
	2D Graphics Engine	
	Audio	
MMCVDD	SRAMS	MMCVDD must be powered on if one or more of the following conditions is true: ▪ Internal SRAM is in use ▪ If VECVDD is on ▪ If TDCVDD is on
TDCVDD	3D	TDCVDD must be on when 3D acceleration is needed.

Table 3.4: Core Power and Ground Pins

Name	Type	Pin Description
AOCVDD	P	**Power for core logic** (Always On partition) Power for host I/F including digital portion of clock generation, both memory controllers, display controller, SD, and test logic.
VECVDD	P	**Power for core logic** Power for video (camera) input, EPP, MPEG/JPEG encoder, MPEG/JPEG decoder, 2D engine, I2S/AC'97, and DSP.
TDCVDD	P	**Power for core 3D logic** Power for 3D engine.
MMCVDD	P	**Power for core SRAM internal memory** Power for internal memory.
GND	G	**Ground** Power to Core and I/O ground

Note: The Core Power values and tolerances are listed in *Chapter 4,* "Specifications".

Table 3.5: Clock Power and Ground Pins

Name	Type	Pin Description
AVDD0SC	P	**Analog Power for Crystal Oscillator** This is analog power for crystal oscillator clock pins and internal clock circuitry.
AGND0SC	P	**Analog Ground for Crystal Oscillator** This is analog ground for clock pins and internal clock circuitry.
AVDDP1	P	**Analog Power for PLL1** This is analog power for internal clock PLL1 which must be set to the same voltage as core power supply externally.
AGNDP1	P	**Analog Ground for PLL1** This is analog ground for internal clock PLL1 which must be set to the same voltage as core ground externally.
AVDDP2	P	**Analog Power for PLL2** This is analog power for internal clock PLL2, which must be set to the same voltage as core power supply externally.
AGNDP2	G	**Analog Ground for PLL2** This is analog ground for internal clock PLL2, which must be set to the same voltage as core ground externally.

3.4 GoForce 5500 I/O Power Rails

In addition to the power islands, the GoForce 5500 has six I/O power rails.

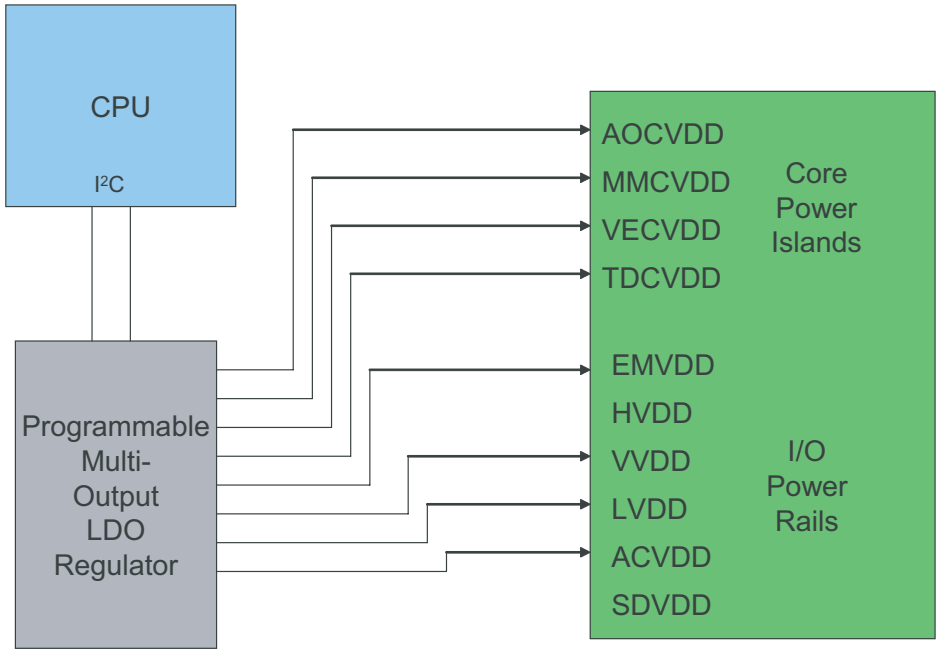
Table 3.6: GoForce 5500 Power Rails

Name	Description	Notes for Power Savings
HVDD	Bus I/O Power	Always keep on when AOCVDD power is on, to minimize leakage current.
VVDD	External Camera I/O power	Turn off when external camera interface is not in use
EMVDD	External memory: External SDRAM	Turn off when not using external memory.
LVDD	Display I/O power	Turn off when display is powered-down
ACVDD	External audio codec I/O power	Turn off when not using external audio Codec support
SDVDD	Secure Digital I/O power	Always keep on when AOCVDD power is on, to minimize leakage current.

3.4.1 Notes on Using the GoForce 5500 I/O Power Rails

- Always power on the core voltage to any given module before powering on the I/O voltage associated with it.
- Power off the I/O voltage before powering off the core voltage.
- Hold all powered-off power rails to a low voltage level. Never float them.
- Do not use HGP[3:0] to enable power supplies since they are used for mode strapping during power-on reset.

Figure 3.1: GoForce 5500 Power Control with Maximum Flexibility



3.4.1.1 Power Savings Tips

- Utilize the Deep power-down function (called DPD_) by connecting the ball on the GoForce 5500 marked DPD_ to a GPIO on the Host CPU. Doing so minimizes the leakage current when AOCVDD goes into sleep mode while HVDD stays powered on, lengthening battery life.
- Limit Dynamic power control to the power rails that consume significant leakage current.
- Use a few GPIOs and more standard Low Dropout (LDO) regulators, rather than multi-output programmable models to control power consumption.
- Example:
  - 3D, Video, and external memory power islands controlled
  - All other power Islands and I/O rails are left powered on
  - Some I/O rails may share an LDO if voltage is consistent. (LVDD and SDVDD share an LDO.)
  - To control power to Video I/O, Display I/O, or embedded memory: use additional LDOs controlled by additional GPIOs.

3.5 Host Bus Interface Pins

The Host Bus Interface Pins are referenced to HVDD. Two types of fixed latency and variable-latency bus interfaces are supported; Type A and Type C. Each interface type may be configured to 16 bit and 32bit widths.

The Type A host interface has separate signals for write and read cycle control (WR_ and RD_, respectively). If the BE_ signals are active, they control enabling bytes for write cycles.

The Type C host interface utilizes a single control signal for write and read (OE_); the low state enables reads, the high state enables writes. Each byte utilizes a separate write enable signal.

The Host Bus Interface Pins are powered by HVDD, the logic is powered by AOCVDD.

Table 3.7: Host Bus Interface Pins Overview

GoForce 5500 Ball Name	Type A	Type C
CS_	CS_	CS_
HD[31:0]	HD[31:0]	HD[31:0]
A[25:2]	A[25:2]	A[25:2]
BE0_	BE0_	WE0_
BE1_	BE1_	WE1_
BE2_	BE2_	WE2_
BE3_ [1]	BE3_	WE3_
RD_	RD_	OE_
WR-	WR_	- [2]
INTR_	INTR_	INTR_
RDY_	RDY_	RDY_

1. BE3_ is used as A[1] with 16bit interfaces.
2. Tie the WR_ pin to low externally with type-C style host interfaces. It is not used for the type-C configuration.

Note: Tie all remaining unused pins high or low as indicated in Table 3.8 or Table 3.10, depending on which Host Interface type you use.

Table 3.8: Type A Style Bus Interface Pins

GoForce 5500 Pin Name	Pin Type	Pin Description
RST_	I	**Reset (active low)** This pin resets the GoForce 5500 when driven low, and puts the GoForce 5500 in sleep mode.
REFCLK0	I	**Clock Input** This pin may be optionally used to input external clock source if needed.
REFCLK1	I/O	**Second Reference Clock Input** This pin may be used to input an external clock source or as a GPIO.
A[25:2]	I	**Host Address** Address for host read/write accesses to the GoForce 5500. This bus should always be driven by the host. A3 and A2 are used in indirect addressing mode. A3: Primary or secondary channel access. 0 = Primary channel access 1 = Secondary channel access A2: Indicates address or data cycle. 0 = Data Cycle 1= Address cycle
BE3_ BE2_ BE1_ BE0_	IS	**Byte Enables (active low)** These pins are driven low by the host for writing to corresponding byte. For a 16-bit interface, use BE0_ and BE1_ as byte enables. Connect BE2_ to ground externally. Use BE3_ as A[1]. For a 32-bit interface, BE0_, BE1_, BE2_, and BE3_ are used as byte enables.
RD_	I	**Read (active low)** This pin is driven low by the host for read cycles and driven high by the host for write cycles.
WR_	I	**Write (active low)** This pin is driven low by the host for write cycles and driven high by the host for read cycles.
CS_	IS	**Chip Select (Active Low)** Asserted by the host to activate the host cycles for the GoForce 5500. The GoForce 5500 decodes the other host inputs only when this signal is asserted.
D[31:16]	IS / O	**Host Data Bus Bits [31:16]** For a 32bit host bus interface: Data bus driven by the host during write accesses and by the GoForce 5500 during read accesses. These pins are tri-stated during reset, and when a 16bit host interface is used.
D[15:0]	IS / O	**Host data (lower 16 bits)** The host drives the write data during write cycles and the GoForce 5500 drives read data during read cycles. These pins are tri-stated with input disabled during reset.
MHGP0 (MD0)	I/O	**Host Controller General Purpose 0 (Mode select 0)** This pin is tri-stated with input enabled during reset and is internally latched when reset is active (low) to decide the host interface type. Following reset, this pin can be configured as general purpose input/output pin.
MHGP1 (MD1)	I/O	**Host Controller General Purpose 1 (Mode select 1)** This pin is tri-stated with input enabled during reset and is internally latched when reset is active (low) to decide the host interface type. Following reset, this pin can be configured as general purpose input/output pin.
MHGP2 (MD2)	I/O	**Host Controller General Purpose 2 (Mode select 2)** This pin is tri-stated with input enabled during reset and is internally latched when reset is active (low) to decide the host interface type. Following reset, this pin can be configured as general purpose input/output pin.
MHGP3 (INT_)	I/O	**Host Controller General Purpose 3 (Interrupt/Status)** This pin is tri-stated with input disabled during reset. This pin can be used to output active low interrupt or internal status. If used as interrupt this pin is configured as active low open-drain output. If not used as interrupt/status pin it can be used as general-purpose input/output.

Table 3.8: Type A Style Bus Interface Pins (Cont.)

GoForce 5500 Pin Name	Pin Type	Pin Description
MHGP4 (RDY)	I/O	**Host controller General Purpose 4 (Ready)** This pin is tri-stated with input disabled during reset. It can be used as general-purpose input/output for no handshake host mode or as ready (RDY) pin for handshake host mode. For active low RDY handshake mode, this pin will be driven inactive (high) by the GoForce 5500 during host read/write access from the beginning of the cycle till the GoForce 5500 is ready. When the GoForce 5500 is ready, it will then drive this pin active (low) until the end of read/write cycle. At the end of the cycle the GoForce 5500 will momentarily drives this pin inactive (high) then tri-state it. For active high RDY handshake mode, this pin will be driven inactive (low) by the GoForce 5500 during host read/write access from the beginning of the cycle till the GoForce 5500 is ready. When the GoForce 5500 is ready, it will then drive this pin active (high) until the end of read/write cycle. At the end of the cycle the GoForce 5500 will momentarily drives this pin inactive (low) then tri-state it.
MHGP5	I/O	**Host controller General Purpose 5** This pin is tri-stated with input disabled during reset. Following reset, it can be used as general purpose input/output pin.
MHGP6 (MD3)	I/O	**Host controller General Purpose 6 (Mode Select 3)** This pin is tri-stated with input enabled during reset and is internally latched when reset is active (low) to decide the host interface type. Following reset, this pin can be configured as general purpose input/output pins.
DPD_	I/O	**Deep Power Down** Connects to a GPIO on the Host CPU and tri-states the Host I/O so AOVDD can be powered down for minimum current leakage. (HVDD stays on, and battery life can be extended. The GoForce 5500 can go into sleep mode.)
HVDD	P	**Power for host interface pins** These are power for host interface pins.

Note: HVDD pins can be set at a voltage level independent from other power supply pins.

Note: Any of the GPIO pins, MHGP[6:4] and MHGP[2:0], when configured as interrupts, utilize the interrupt signal INT_ on MHGP3 to output an interrupt from the GoForce 5500 to the Host CPU. MHGP3 contains all the interrupts on the MHGPx GPIOs, the interrupt from the SDIO Module, and the internal module interrupts. The register HOST1X_SYNC_INTSTATUS_0 contains the interrupt status for HGP3, with a bit corresponding to each possible interrupt source.

Note: Pins MHGP6, MHGP2, MHGP1, and MHGP0 are used for configuring the host interface modes and can be configured for general purpose input or output use following reset.

Table 3.9: Type A and Type C Host Interface Mode Pin Configurations

MHGP6 (MD3)	MHGP2 (MD2)	MHGP1 (MD1)	MHGP0 (MD0)	Definition of Mode
0	0	0	0	Direct addressing fixed cycle mode
0	0	0	1	Reserved
0	0	1	0	Direct addressing active low ready handshake mode
0	0	1	1	Direct addressing active high ready handshake mode
0	1	0	0	Indirect addressing fixed cycle mode
0	1	0	1	Reserved
0	1	1	0	Indirect addressing active low ready handshake mode
0	1	1	1	indirect addressing active high ready handshake mode
1	0	0	0	Synchronous host, direct addressing fixed cycle mode
1	0	0	1	Reserved
1	0	1	0	Synchronous host, direct addressing active low ready handshake mode
1	0	1	1	Synchronous host, direct addressing active high ready handshake mode
1	1	0	0	Synchronous host, indirect addressing fixed cycle mode
1	1	0	1	Reserved
1	1	1	0	Synchronous host, indirect addressing active low ready handshake mode
1	1	1	1	Synchronous host, indirect addressing active high ready handshake mode

Table 3.10: Type C Style Bus Interface Pins

GoForce 5500 Pin Name	Pin Type	Pin Description
RST_	I	**Reset (active low)** This pin resets the GoForce 5500 when driven low and puts the GoForce 5500 in sleep mode.
REFCLK0	I	**Clock Input** This pin may be optionally used to input external clock source if needed.
REFCLK1	I/O	**Second Reference Clock Input** This pin may be used to input an external clock source, or as a GPIO.
CS_	I	**Chip Select (active low)** This pin is active low chip select, which must be asserted for all read/write access to the GoForce 5500.
A[25:2]	I	**Host Address** This is byte address for host read/write accesses to the GoForce 5500. This bus should always be driven by the host. If indirect addressing mode, only A23 is used. A3: Primary or secondary channel access. 0 = Primary channel access 1 = Secondary channel access A2: Indicates address or data cycle. 0: Data Cycle 1: Address cycle
BE3_ BE2_ BE1_ BE0_	IS	**Byte Enables (active low)** These pins are driven low by the host for writing to corresponding byte. For a 16-bit interface, use BE0_ and BE1_ as byte enables. Connect BE2_ to ground externally. Use BE3_ as A[1]. For 32-bit interface, BE0_, BE1_, BE2_, BE3_ are used as byte enables.
OE_	I	**Output Enable (active low)** This pin is used as active low output enable control (OE_). It is driven low by the host for read cycles and driven high by the host for write cycles.
WR_	I	**Always tie this GoForce 5500 pin low when utilizing Type C Host Interfaces**-- Type C Host Interfaces never use the WR_ signal.
CS_	IS	**Chip Select (Active Low)** Asserted by the host to activate the host cycles for the GoForce 5500. The GoForce 5500 decodes the other host inputs only when this signal is asserted.
D[31:16]	IS / O	**Host Data Bus Bits [31:16]** For a 32bit host bus interface: Data bus driven by the host during write accesses and by the GoForce 5500 during read accesses. These pins are tri-stated during reset, and when a 16bit host interface is used.
D[15:0]	IS / O	**Host data (lower 16 bits)** The host drives the write data during write cycles and the GoForce 5500 drives read data during read cycles. These pins are tri-stated with input disabled during reset.
MHGP0 (MD0)	I/O	**Host controller General Purpose 0 (Mode select 0)** This pin is tri-stated with input enabled during reset and is internally latched when reset is active (low) to decide the host interface type. Following reset, this pin can be configured as general purpose input/output pins.
MHGP1 (MD1)	I/O	**Host controller General Purpose 1 (Mode select 1)** This pin is tri-stated with input enabled during reset and is internally latched when reset is active (low) to decide the host interface type. Following reset, this pin can be configured as general purpose input/output pins.
MHGP2 (MD2)	I/O	**Host controller General Purpose 2 (Mode select 2)** This pin is tri-stated with input enabled during reset and is internally latched when reset is active (low) to decide the host interface type. Following reset, this pin can be configured as general purpose input/output pins.
MHGP3 (INT_)		**Host controller General Purpose 3 (Interrupt/Status)** This pin is tri-stated with input disabled during reset. This pin can be used to output active low interrupt or internal status. If used as interrupt this pin is configured as active low open-drain output. If not used as interrupt/status pin it can be used as general-purpose input/output.

Table 3.10: Type C Style Bus Interface Pins (Cont.)

GoForce 5500 Pin Name	Pin Type	Pin Description
MHGP4 (RDY)	I/O	**Host controller General Purpose 4 (Ready)** This pin is tri-stated with input disabled during reset. It can be used as general-purpose input/output for no handshake host mode or as ready (RDY) pin for handshake host mode. For active low RDY handshake mode, this pin will be driven inactive (high) by the GoForce 5500 during host read/write access from the beginning of the cycle till the GoForce 5500 is ready. When the GoForce 5500 is ready, it will then drive this pin active (low) until the end of read/write cycle. At the end of the cycle the GoForce 5500 will momentarily drives this pin inactive (high) then tri-state it. For active high RDY handshake mode, this pin will be driven inactive (low) by the GoForce 5500 during host read/write access from the beginning of the cycle till the GoForce 5500 is ready. When the GoForce 5500 is ready, it will then drive this pin active (high) until the end of read/write cycle. At the end of the cycle the GoForce 5500 will momentarily drives this pin inactive (low) then tri-state it.
MHGP5	I/O	**Host controller General Purpose 5** This pin is tri-stated with input disabled during reset. Following reset, it can be used as general purpose input/output pin.
MHGP6 (MD3)	I/O	**Host controller General Purpose 6 (Mode Select 3)** This pin is tri-stated with input enabled during reset and is internally latched when reset is active (low) to decide the host interface type. Following reset, this pin can be configured as general purpose input/output pins.
DPD_	I/O	**Deep Power Down** Connects to a GPIO on the Host CPU and tri-states the Host I/O so AOVDD can be powered down for minimum current leakage. (HVDD stays on, and battery life can be extended. the GoForce 5500 can go into sleep mode.)
HVDD	P	**Power for host interface pins** These are power for host interface pins.

3.6 Video Input Pins

Table 3.11 lists and provides a brief description of the VI Interface signals.

The VI pins are referenced to VECVDD (which supplies VVDD.)

Table 3.11: Video Input Pins

GoForce 5500 Pin Name	Type	Description
VD[11:0]	IS	**Video Input Data** These pins can be used to input YUV or Bayer data from a video source (camera). If less than 12-bit is transferred, then data should be aligned according to the format in Table 2.3. These pins are tri-stated with input disabled during reset. It can be used as general-purpose input/output if not used to accept video input data.
VCLK VGP13	IS / O	**Video Input Clock** This clock is used to latch video input data. This signal can be programmed as either input or output. On reset, this pin is tri-stated with its input disabled. **Video General Purpose Input/Output 13** This pin can be used as general-purpose input/output.
VHSYNC VGP14	IS / O	**Video Input Horizontal Sync** This signal indicates horizontal sync for incoming video data that can optionally be used as a reference to indicate start of active pixel in video input line. This signal can be programmed as either input or output. On reset, this pin is tri-stated, and its input buffer is disabled. May be used as GPIO if not used to accept video horizontal sync.
VVSYNC VGP15	IS / O	**Video Input Vertical Sync** This signal indicates vertical sync for incoming video data that can optionally be used as a reference to indicate start of active line in video input frame. This signal can be programmed as either input or output. On reset, this pin is tri-stated, and its input buffer is disabled. May be used as a GPIO if not used to accept video vertical sync.
VGP0 VSNCLK	IS / O	**Camera Master Clock** **Video General-Purpose Input/Output 0** This pin is driven low with input disabled during reset and can be used to output clock signal to the video source (camera). Otherwise, it can be used as a general-purpose input/output pin.
VGP1 ICSCK	IS / O	**Video General Purpose Input/Output** These pins are used as general-purpose input/output. Upon reset, these pins are tri-stated and their input buffers are disabled. This pin can be used to output serial clock for programming the video source (camera): When SPB is enabled, VGP1 is the SPB Clock pin.
VGP2 ICSDA	IS/O	**Video (camera) General Purpose Input/Output 2** This pin is tri-stated with input disabled during reset. This pin can be used to output serial data for programming the video source (camera). When SPB is enabled VGP2 is the SPB Data pin.
VGP3	I/O	**Video (camera) General Purpose control 3** This pin is used as general-purpose input/output. Upon reset, these pins are tri-stated and their inputs are disabled.
VGP4	I/O	**Video (camera) General Purpose control 4** This pin is used as general-purpose input/output. Upon reset, these pins are tri-stated and their inputs are disabled.

Table 3.11: Video Input Pins (Cont.)

GoForce 5500 Pin Name	Type	Description
VGP5	I/O	**Video (camera) General Purpose control 5** This pin is tri-stated with input disabled during reset and can be used as a general-purpose input/output pin.
VGP6	I/O	**Video (camera) General Purpose control 6** This pin is tri-stated with input disabled during reset and can be used as a general-purpose input/output pin: VI PWM signal generation: Programmable PWM for driving a flash circuit or possible shutter for a camera.
VVDD	P	**Power for video input interface pins**

3.7 Display Controller Interface Pins

Table 3.12 lists and provides a brief description of the Display Controller Interface signals. The Display Controller pins are powered by LVDD.

Table 3.12: Display Controller Interface Pins

GoForce 5500 Pin Name	Type	Description
LD[17:0]	I/O	**LCD Data** These pins are typically driven with output data for LCD display. These pins are driven low regardless of their polarity during reset and when display controller is disabled. The input buffers are disabled during reset.
LPW0	I/O	**LCD Power control 0** This pin can be used for power sequencing of the display. This pin is driven low regardless of its polarity during reset. The input buffer is disabled during reset.
LPW1	I/O	**LCD Power control 1** This pin can be used for power sequencing of the display. This pin is driven low regardless of its polarity during reset. The input buffer is disabled during reset.
LPW2	I/O	**LCD Power control 2** This pin can be used for power sequencing of the display. This pin is driven low regardless of its polarity during reset. The input buffer is disabled during reset.
LSC0	I/O	**LCD Shift Clock 0** This pin is typically driven with shift clock for LCD display. If programmed active high, falling edge of this signal should be used by the display to latch data. If programmed active low, rising edge of this signal should be used by the display to latch data. This pin is driven low regardless of its polarity during reset and when display controller is disabled. The input buffer is disabled during reset.
LSC1	I/O	**LCD Shift Clock 1** This pin is typically driven with either a second shift clock or driven with display enable signal for LCD display. This pin is driven low regardless of its polarity during reset and when display controller is disabled. The input buffer is disabled during reset.
LVS	I/O	**LCD Vertical Sync** This pin is typically driven with vertical sync signal for LCD display. This pin is driven low regardless of its polarity during reset and when display controller is disabled. The input buffer is disabled during reset.
LHS	I/O	**LCD Horizontal Sync** This pin is typically driven with horizontal sync signal for LCD display. This pin is driven low regardless of its polarity during reset and when display controller is disabled. The input buffer is disabled during reset.
LHP0	I/O	**LCD Horizontal Pulse 0** This pin is typically driven with horizontal pulse 0 signal for LCD display. Up to four programmable width pulse per line can be output on this pin. This pin is driven low regardless of its polarity during reset and when display controller is disabled. The input buffer is disabled during reset.
LHP1	I/O	**LCD Horizontal Pulse 1** This pin is typically driven with horizontal pulse 1 signal for LCD display. Up to four programmable width pulses per line can be output on this pin. This pin is driven low regardless of its polarity during reset and when display controller is disabled. The input buffer is disabled during reset.
LHP2	I/O	**LCD Horizontal Pulse 2** This pin is typically driven with horizontal pulse 2 signal for LCD display. Up to four programmable width pulses per line can be output on this pin. This pin is driven low regardless of its polarity during reset and when display controller is disabled. The input buffer is disabled during reset.

Table 3.12: Display Controller Interface Pins (Cont.)

GoForce 5500 Pin Name	Type	Description
LVP0	I/O	**LCD Vertical Pulse 0** This pin is typically driven with vertical pulse 0 signal for LCD display. Up to three programmable width pulses per frame can be output on this pin. This pin is driven low regardless of its polarity during reset and when display controller is disabled. The input buffer is disabled during reset.
LVP1	I/O	**LCD Vertical Pulse 1** This pin is typically driven with vertical pulse 1 signal for LCD display. Up to three programmable width pulses per frame can be output on this pin. This pin is driven low regardless of its polarity during reset and when display controller is disabled. The input buffer is disabled during reset.
LM0	I/O	**LCD M (modulation) 0** This pin is typically driven with modulation signal 0 for LCD display which can be programmed to toggle either every frame or every 1 to 8 lines. This pin is driven low regardless of its polarity during reset and when display controller is disabled. The input buffer is disabled during reset.
LM1	I/O	**LCD M (modulation) 1** This pin is typically driven with modulation signal 1 for LCD display which can be programmed to toggle either every frame or every 1 to 128 lines. This pin is driven low regardless of its polarity during reset and when display controller is disabled. The input buffer is disabled during reset.
LDI	I/O	**LCD Data Inversion** This pin is typically driven with data inversion signal which is common for PWM STN LCD display. This pin is driven low regardless of its polarity during reset and when display controller is disabled. The input buffer is disabled during reset.
LPP	I/O	**LCD Programmable Pulse** This pin is typically driven with programmable pulse (up to 128 per line) which is commonly used as PWM STN clock but it can also be used for general purpose pulse/signal generator for other type of LCD. This pin is tri-stated with input disabled during reset. The input buffer is disabled during reset.
LSCK	I/O	**LCD Serial Clock** This pin is typically driven with SPI serial clock for register/command programming of some LCD displays. This pin is tri-stated with input disabled during reset. The input buffer is disabled during reset.
LSDA	I/O	**LCD Serial Data** This pin is typically driven with SPI serial data for register/command programming of some LCD displays. This pin is driven low regardless of its polarity during reset and when display controller is disabled. The input buffer is disabled during reset.
LCS_	I/O	**LCD Serial Chip Select** This pin is typically driven with SPI serial chip select (active low) for register/command programming of some LCD displays. This pin is tri-stated with input disabled during reset. The input buffer is disabled during reset.

Table 3.12: Display Controller Interface Pins (Cont.)

GoForce 5500 Pin Name	Type	Description
LDC	I/O	**LCD Serial Data/Command** This pin is typically driven with SPI serial data/command for register/command programming of some LCD displays. This is an optional signal since the serial host interface for some LCD displays does not require an additional pin to indicate data/command. This pin is tri-stated with input disabled during reset. The input buffer is disabled during reset.
LSPI	I/O	**LCD Serial Programming in progress** This pin can be driven with a signal that indicates that serial programming is in progress. Alternatively it can be used to output LDE or to input a tearing-prevention signal for some LCD displays which have a built-in frame buffer and provide this signal. This pin is tri-stated with input disabled during reset. The input buffer is disabled during reset.
LVDD	P	**Power for LCD display controller interface pins** These are power for LCD display controller interface pins.

Note: LVDD pins can be set at a voltage level independent from other power supply pins. All Display Controller pins can be configured as general-purpose input/output pins if necessary.

Note: All control signals except for LSC0 and LSC1 can be configured to output a pulse-width modulation signal (either LPM0 or LPM1) or to output a general-purpose LCD mode (LMD0 or LMD1 or LMD2 or LMD3) signal. The pulse-width modulation signals can typically be used for contrast/brightness control. These options allow unused pins to be used for other purposes. Please refer to the register definitions for these programmability options.

Note: LCD interface pins are sufficient only for a single display or two interleaved displays. More pins need to be added to support true dual-display.

3.8 Clock Pins

The AVDDOSC pins power the clock signals on the GoForce 5500. The clock pin names and functions remain the same as they were in GoForce 4500.

Table 3.13: Clock Pins

GoForce 5500 Pin Name	Pin Type	Pin Description
OSCFI	AI	**Crystal Oscillator Input** Input pin for an external crystal in the range of 2 to 13 MHz.
OSCFO	AI / O	**Crystal Oscillator Output** When an external crystal (2 to 13 MHz) is connected to OSCFI, OSCFO becomes the internal crystal oscillator's output source. When the internal oscillator is bypassed, this input may be driven by an external clock source ranging from 2 MHz to 50 MHz.
OSCFR	A	**Relaxation Oscillator Resistor** OSCFR connects to an external 180 kΩ resistor, which also connects to AVDDOSC. The resistor is needed by all internal clocks: the relaxation oscillator, the internal crystal oscillator, and the clock multipliers.

Note: When using an external crystal (at pin OSCFI), use pin OSCFO as the output for the internal crystal oscillator. When using an external oscillator source, use pin OSCFO as the input pin and leave pin OSCFI floating (not connected)

Note: The voltage level on pin OSCFI must always be between the analog supply (AVDDOSC) and ground (AGNDOSC) voltage levels.

Note: The amplitude of the input signal on pin OSCFO must follow HVDD. That is, Min V_{IH} = 0.8*HVDD.

3.9 JTAG Interface Pins

The JTAG pins are used for interfacing with the GoForce 5500 for diagnostic and testing purposes. Table 3.14 lists the five GoForce 5500 JTAG interface pins. The signal names and functions of the JTAG interface pins remain the same as they were in the GoForce 4500.

Note: The JTAG Interface pins are referenced to SDVDD. The JTAG Reset pin, TRST, is referenced to HVDD.

Table 3.14: Description of the JTAG Interface Pins

Pin Name	Type	Drive (mA)	Description
TCK	I	4	**JTAG Clock** Clock pin for JTAG tap controller
TDI	I	4	**JTAG Data Input** Data from previous JTAG device or controller
TDO	O	4	**JTAG Data Output** Data sent to next JTAG device or controller
TMS	I	4	**JTAG Mode Select** Selects between instruction and data scan
TRST_	I	4	**JTAG Reset** Reset the internal JTAG tap controller

Note: Tie TRST_ to ground for normal operation. TRST_ may also be tied to RST_.

3.10 External Memory Interface

The Memory Interface pins are referenced to EMVDD. EMVDD can be set at a voltage level independent to the other power supply pins. All of these signals are new and completely unique to the GoForce 5500.

Table 3.15: Description of the External Memory Interface Pins

Pin Name	Type	Description
MA[12:0]	O	**Memory Address** These pins provide address for external SDRAM or DDR memory. These pins are driven low regardless of their polarity during reset and when external memory controller is disabled.
MBA[1:0]	O	**Bank Select** These pins provide bank select for external SDRAM or DDR memory. These pins are driven low regardless of their polarity during reset and when external memory controller is disabled.
MD[31:0]	I/O	**Memory Data** These pins are driven by the GoForce 5500 with write data during write cycle and they are driven by the external memory with read data during read cycle. These pins are tri-stated with their input buffer disabled during reset and when external memory controller is disabled.
MDQS0	I/O	**Byte 0 Data Strobe** This pin is driven by the GoForce 5500 and used as byte 0 write data strobe during write cycle and it is driven by the external memory with byte 0 read data strobe during read cycle. This signal is used only with DDR memory. This pin is tri-stated with input buffer disabled during reset and when external memory controller is disabled.
MDQS1	I/O	**Byte 1 Data Strobe** This pin is driven by the GoForce 5500 and used as byte 1 write data strobe during write cycle and it is driven by the external memory with byte 1 read data strobe during read cycle. This signal is used only with DDR memory. This pin is tri-stated with input buffer disabled during reset and when external memory controller is disabled.
MDQS2	I/O	**Byte 2 Data Strobe** This pin is driven by the GoForce 5500 and used as byte 2 write data strobe during write cycle and it is driven by the external memory with byte 2 read data strobe during read cycle. This signal is used only with DDR memory. This pin is tri-stated with input buffer disabled during reset and when external memory controller is disabled.
MDQS3	I/O	**Byte 3 Data Strobe** This pin is driven by the GoForce 5500 and used as byte 3 write data strobe during write cycle and it is driven by the external memory with byte 3 read data strobe during read cycle. This signal is used only with DDR memory. This pin is tri-stated with input buffer disabled during reset and when external memory controller is disabled.
MCS_	O	**Memory Chip Select (active low)** This pin provides chip select for external SDRAM or DDR memory. This pin is driven high during reset and when external memory controller is disabled. The input buffer is disabled during reset.
MRAS_	O	**Row Address Strobe (active low)** This pin provides row address strobe for external SDRAM or DDR memory. This pin is driven high during reset and when external memory controller is disabled. The input buffer is disabled during reset.
MCAS_	O	**Column Address Strobe (active low)** This pin provides column address strobe for external SDRAM or DDR memory. This pin is driven high during reset and when external memory controller is disabled. The input buffer is disabled during reset.
MWE_	O	**Memory Write Enable (active low)** This pin provides write enable for external SDRAM or DDR memory. This pin is driven high during reset and when external memory controller is disabled. The input buffer is disabled during reset.

Table 3.15: Description of the External Memory Interface Pins (Cont.)

Pin Name	Type	Description
MDM0	O	**Byte 0 Data Mask** This pin provides byte 0 write data mask during write cycle. This pin is driven low during reset and when external memory controller is disabled.
MDM1	O	**Byte 1 Data Mask** This pin provides byte 1 write data mask during write cycle. This pin is driven low during reset and when external memory controller is disabled
MDM2	O	**Byte 2 Data Mask** This pin provides byte 2 write data mask during write cycle. This pin is driven low during reset and when external memory controller is disabled
MDM3	O	**Byte 3 Data Mask** This pin provides byte 3 write data mask during write cycle. This pin is driven low during reset and when external memory controller is disabled
MCLK_	O	**Memory Clock (active low)** This pin provides invert of memory clock for the external DDR memory. This pin is driven high during reset and when external memory controller is disabled.
MCKE	O	**Memory Clock Enable** This pin provides memory clock enable for the external memory. If this pin is high the next M0CK edge is valid, else if this pin is low the next M0CK edge is invalid. This pin is driven low during reset and when external memory controller is disabled
EMVREF	I	**External Memory Reference Voltage** For use with GoForce 5500-XT: provides a reference voltage for the External DRAM. Set to EMVDD/2 by using a voltage divider.

3.11 Secure Digital (SD) Interface Pins

Table 3.16 lists and briefly describes the GoForce 5500 SD/GPIO[65:60] pins.

Note: The SD Core is referenced to AOCVDD, the SD Interface pins are referenced to SDVDD.

Table 3.16: Description of the Secure Digital (SD) / GPIO[65:60] Pins

Pin Name	Type	Drive (mA)	Description
SDD3 GPIO[63]	IS/O	4	**Card Detect/Data Line 3 (DAT3/CD)** On power up, the host uses this pin for card detection. Later this pin will be used as DATA line 3 in wide-bus mode. **General Purpose Input/Output**
SDCMD GPIO[65]	IS/O	4	**Secure Digital Command** This pin is used to send commands to the SD card and responses to the SD Host. **General Purpose Input/Output**
SDCLK GPIO[64]	IS/O	4	**Secure Digital Clock** The host provides the clock to the SD card through this pin. The maximum frequency on this pin is 25MHz **General Purpose Input/Output**
SDD0 GPIO[60]	IS/O	4	**Data Line 0 (DAT0)** This pin is used as the data line in both single pin and wide-bus data transfer modes. **General Purpose Input/Output**
SDD1 GPIO[61]	IS/O	4	**Data Line 1(DAT1)** This pin is used as the data line 1 in wide-bus data transfer mode. **General Purpose Input/Output**
SDD2 GPIO[62]	IS/O	4	**Data Line 2 (DAT2)** This pin is used as the data line 2 in wide-bus data transfer mode. **General Purpose Input/Output**
SDGP0	IS/O	4	**SD GPIO0** General Purpose Input/Output.
SDGP1	IS/O	4	**SD GPIO1** General Purpose Input/Output.

3.12 I²S/AC'97 CODEC Interface

The I²S/AC'97 Codec Interface pins are powered by ACVDD, the I2S/AC'97 core is referenced to VECVDD.

Table 3.17: Description of the I²S/AC'97 CODEC Interface Pins

Pin Name	Type	Description
SRCLK	I/O	**Serial Root Clock** This pin can be optionally used to input root clock for the codec interface for all modes of operation. This pin is tri-stated with input buffer enabled after reset. If not used for codec interface, this pin can be used as general-purpose input/output pin.
SMCLK	I/O	**Serial Master Clock** This pin can be used to output master clock to the external codec for all modes of operation. This pin is tri-stated with input buffer enabled after reset. If not used for codec interface, this pin can be used as general-purpose input/output pin.
SCLK	I/O	**Serial Bit Clock** This pin can be used to input serial bit clock in modes 0 and 2 and to output serial bit clock in modes 1 and 3. This pin is tri-stated with input buffer enabled after reset. If not used for codec interface, this pin can be used as general-purpose input/output pin.
SFSYNC	I/O	**Frame Synchronization** This pin can be used to input frame synchronization signal in modes 0, 1 and to output frame synchronization signal in modes 2 and 3. This pin is tri-stated with input buffer enabled after reset. If not used for codec interface, this pin can be used as general-purpose input/output pin.
SIN	I/O	**Serial Data Input** This pin is used to receive serial data input from the codec. This pin is tri-stated with input buffer enabled after reset. If not used for codec interface, this pin can be used as general-purpose input/output pin.
SOUT	I/O	**Serial Data Output** This pin is used to output serial data output to the codec. This pin is tri-stated with input buffer enabled after reset. If not used for codec interface, this pin can be used as general-purpose input/output pin.

Chapter 4 Specifications

4.1 GoForce 5500 Electrical Specifications

In this document you may see "TBD" as the stated value for a given parameter. In such cases "TBD" acts as a place holder and will be replaced with data once it becomes available.

Table 4.1 lists the GoForce 5500 voltage rails; Table 4.2 lists the I/O voltages. Note the two different voltage levels available to the supply voltages. These correspond to operation at different frequencies. The core voltages (Table 4.1) must all work at the same level. (AOCVDD should utilize the same voltage level as MMCVDD, and so on.) The IO voltages do not have to all operate at the same voltage level. For example, HVDD may utilize a different voltage than does LVDD.

Note: The core rail-to-rail tolerance is +/- 10%.

Table 4.1 GoForce 5500 Voltage Rails

Symbol	Parameter	Minimum	Typical	Maximum	Unit
AOCVDD	Supply voltage for the internal core 2 core voltages	0.95	1.0	1.1	V
		1.08	1.2	1.32	V
MMCVDD	Supply voltage for SRAM	0.95	1.0	1.1	V
		1.08	1.2	1.32	V
VECVDD	Supply voltage for Video codec, camera interface, ISP, 2D Graphics Engine, Audio	0.95	1.0	1.1	V
		1.08	1.2	1.32	V
TDCVDD	Supply voltage for 3D	0.95	1.0	1.1	V
		1.08	1.2	1.32	V
AVDDOSC	Supply voltage for crystal oscillator	0.95	1.0	1.1	V
		1.08	1.2	1.32	V
AVDDP1, AVDDP2	Supply voltage for PLL1 and PLL2	0.95	1.0	1.1	V
		1.08	1.2	1.32	V

Table 4.2: GoForce 5500 I/O Voltage Rails

Symbol	Parameter	Minimum	Maximum	Unit
HVDD	Bus I/O Power	1.71	3.3	V
VVDD	External Camera I/O Power	1.71	3.3	V
SDVDD	Secure Digital I/O Power	1.71	3.3	V
LVDD	Display I/O Power	1.71	3.3	V
ACVDD	External audio codec I/O Power	1.71	3.3	V

Table 4.3: GoForce 5500 Voltage Rails for Additional Memory

Symbol	Parameter	Minimum	Maximum	Unit
EMVDD	Supply voltage for External Memory			
	GoForce 5500-XT	1.71	1.89	V
	GoForce 5500-2MI	1.71	1.89	V
	GoForce 5500-8ME	1.71	1.89	V

4.2 Temperature Specifications

Table 4.4: Absolute Maximum Ratings

Symbol	Parameter	Minimum	Maximum	Unit
T_{STG}	Storage temperature: 1 year, 45% to 75% relative humidity (RH)	15	35	°C
T_A	Free-air operating ambient temperature	-35	80	°C

4.3 DC Characteristics

The following subsections provide GoForce 5500 DC characteristics information. Any missing values will be provided in a later revision of this document, once there is sufficient data to provide them.

4.3.1 I/O Pin DC Specifications

Table 4.5 provides the GoForce 5500 I/O pin DC specifications. Currently the values shown are based on simulations and are subject to change based on (future) empirical findings. VOL and VOH are not yet available; they will be described by IBIS modelling and added to this document once available.

Please note in Table 4.5 that the value VDD applies to HVDD, LVDD, and so on.

- Bidirectional pins: Pins such as data bus pins, which both send and receive signal
- All I/O pins: All pins irrespective of whether they are bidirectional, input, or purely output.

Table 4.5: DC Characteristics

Symbol	Parameter	Notes	Minimum	Maximum	Unit
I_{IH}	Input HIGH: Leakage current at typical conditions	Input-only pins	_	7	µA
		Bidirectional pins	_	7	µA
I_{IL}	Input LOW: leakage current at typical conditions	Input-only pins	_	7	µA
		Bidirectional pins	_	7	µA
I_{OZH}	Output HIGH impedance leakage current at typical conditions	All I/O pins	_	13	µA
I_{OZL}	Output LOW impedance leakage current at typical conditions	All I/O pins	_	13	µA
V_{IH}	Input high voltage	All inputs	0.8*VDD	0.7 + VDD	V
V_{IL}	Input low voltage	All inputs	-0.7	0.2*VDD	V
V_{OL}	Output low voltage	All outputs	0.20*VDD		V
V_{OH}	Output high voltage	All outputs	0.80*VDD		V

4.3.2 I/O Pin Load Capacitance

Table 4.6 provides theoretical I/O pin load capacitance (based on simulations) of the GoForce 5500. The actual (empirically measured) values will replace these when they are available.

Table 4.6: Theoretical Pin Load Capacitance

Symbol	Parameter	Maximum	Unit
C_{BID}	Bidirectional buffer capacitance	2.5	pF
C_{IN}	Input capacitance	1.2	pF
C_{OUT}	Output capacitance	2.5	pF

4.4 AC Characteristics

The GoForce 5500 AC Characteristic values will be added in a later revision, once sufficient data is available to publish these parameters.

This section describes the GoForce 5500 AC characteristics, which consist of output delays, input setup and hold times, and signal skew times. All signals are specified relative to an appropriate edge of other signals. Table 4.7 lists the conditions used for testing the GoForce 5500 AC timing.

Table 4.7: AC Test Conditions

Symbol	Parameter	Value	Unit
T_F	Input fall time	2	ns
T_R	Input rise time	2	ns
V_{IH}	Input high voltage	+/- 500 mV from VDD/2	
V_{IL}	Input low voltage	+/- 500 mV from VDD/2	
V_{TEST}	Output trip point	VDD/2	V

4.4.1 Clock

Figure 4.1 shows the reference clock timing.

Figure 4.1 Reference Clock Timing

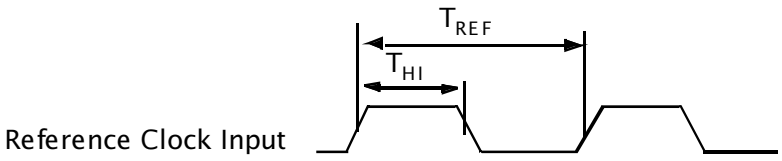


Table 3.6 and Table 3.7 provide the AC timing characteristic values of the input crystal oscillator (OSCFI, used with the internal oscillator clock source) and the external input clock (OSCFO, used when bypassing the internal crystal oscillator.) The values are created by changing the crystal's oscillation signal. The test circuit consists of a 1 MΩ resistor (R1) in parallel with a test crystal, ranging in value from 2 MHz to 13 MHz, and two capacitors (C1 = C2) to ground, which are connected to each side of the crystal/resistor configuration. See Reference Clock Timing Test Circuit: Page 4-5. Refer to the test crystal's specifications to determine the value of C1 and C2, as follows:

$$(C1 = C2 = 2 \times C_{load}\text{Crystal})$$

Figure 4.2: Reference Clock Timing Test Circuit

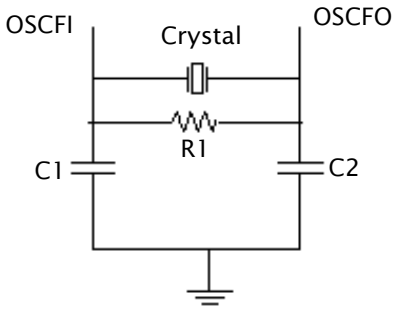


Table 4.8: AC Timing Characteristics - Crystal Input Clock (OSCFI)

Symbol	Parameter	Value		Units
		Min	Max	
T_{REF}	Crystal input frequency	2	13	MHz
T_{HI}/T_{REF}	Clock duty cycle	50%		

Table 4.9: AC Timing Characteristics - External Oscillator Clock Input (OSCFO)

Symbol	Parameter	Value		Units
		Min	Max	
T_{REF}	External oscillator clock frequency	2	50	MHz
T_{HI}/T_{REF}	Clock duty cycle	50%		

Note: When using an external oscillator clock (OSCFO) source, the input voltage should be the same as HVDD (the Host Interface IO power plane.)

4.4.2 Reset

Figure 4.3 Reset Timing

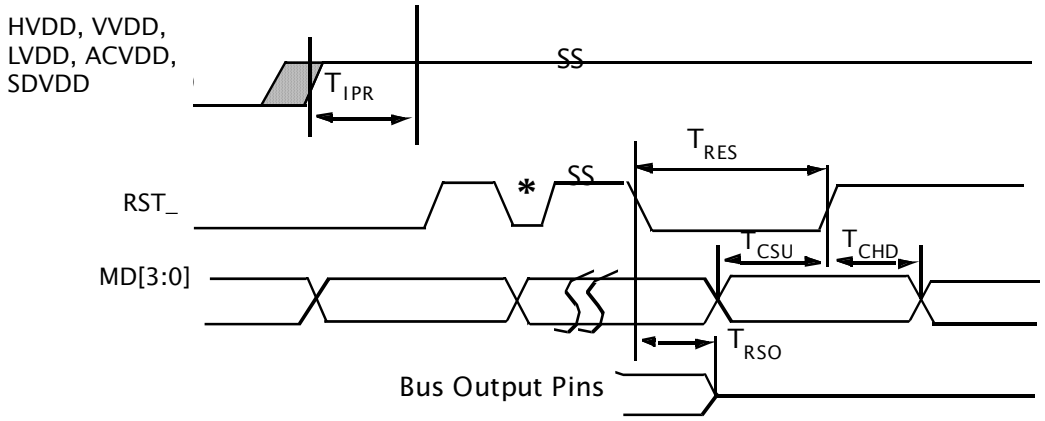


Table 4.10: AC Timing Characteristics - Reset

Symbol	Parameter	Min	Max	Units
T_{IPR}	Reset inactive from power stable[1]	1	–	ms
T_{RES}	Minimum reset pulse width	100	–	µs
T_{RSO}	Reset active to output float delay	–	40	ns
T_{CSU}	Configuration setup time[2]	20	–	ns
T_{CHD}	Configuration hold time	5	–	ns

1. This parameter includes time for internal voltage stabilization of all sections of the chip, start-up and stabilization of the internal clock, and setting of all internal logic to a known state.

2. This parameter specifies the absolute minimum setup time to reliably latch the state of the mode select bits. The recommended configuration bit setup time is T_{RES} to ensure that the chip is in a completely stable state when Reset goes inactive.

4.4.3 GoForce 5500 Power Sequencing

The following two sections, 4.4.3.1 “Power On” and 4.4.3.2 “Power Down”, discuss the GoForce 5500 core power on and power off sequence when the DPD_ function is not utilized. Refer to Figure 4.4 to view the sequencing described.

4.4.3.1 Power On

1. Turn on AOCVDD, then turn on HVDD and SDVDD.
   - Turn AOCVDD on
   - Wait for time T, then turn HVDD and SDVDD on.
   - T = 1 ms, minimum

2. Turn on the rest of the core powers and IO powers.
   - Program the Async Host Registers to enable each core power and IO power.
   - Turn these on in any order with respect to each other.
   - Turn on the first power source a minimum T after HVDD and SDVDD turn on. T = 1 ms, minimum

Note: T is the time between reaching 90% of the AOCVDD AND AVDDOSC power rails, and 10% of the SDVDD, HVDD, AVDDP1, and AVDDP2 power rails. (That is, it is the delay between the first power rails reaching 90%, and the second power rails reaching 10% of their value.)

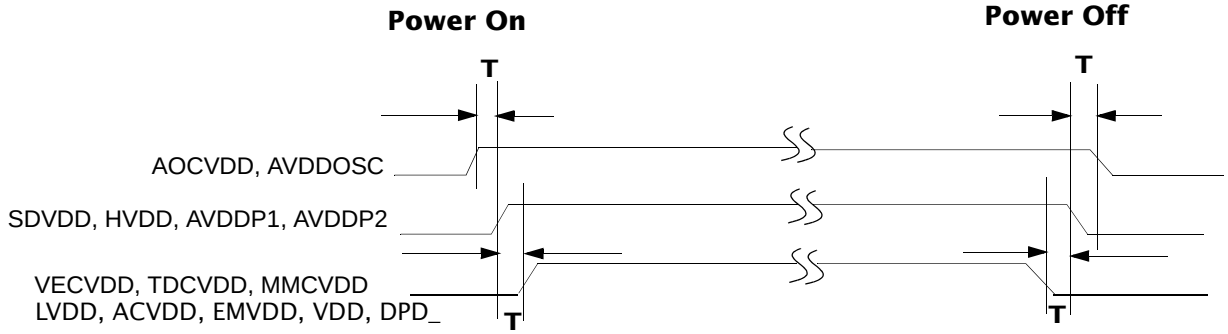
4.4.3.2 Power Down

Power down by reversing the power-on sequence:

1. Program the Async Host Registers (or use the appropriate GFSDK function calls) to disable all Core and IO power sources *except* HVDD, SDVDD, and AOCVDD.
2. Turn the last power source off and allow time T to pass.
   T = 1 ms, maximum
3. Turn HVDD and SDVDD off for time T before turning AOCVDD off:
   T = 1 ms, maximum

Figure 4.4 depicts the GoForce 5500 power on and power off sequencing.

Figure 4.4: GoForce 5500 Power On and Power Off Sequence



4.4.3.3 Sequencing with DPD_ and Reset

Two power sequencing scenarios to consider with DPD_ occur according to whether AOCVDD is powered on first relative to HVDD (Case 1), or HVDD is powered on first (Case 2.) Figure 4.4 (Case 1) and Figure 4.5 (Case 2) illustrate each case.

Figure 4.5: Power Sequence with DPD_ Case 1

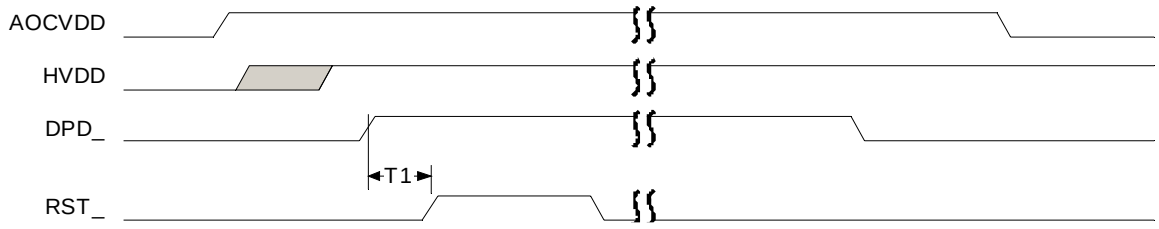
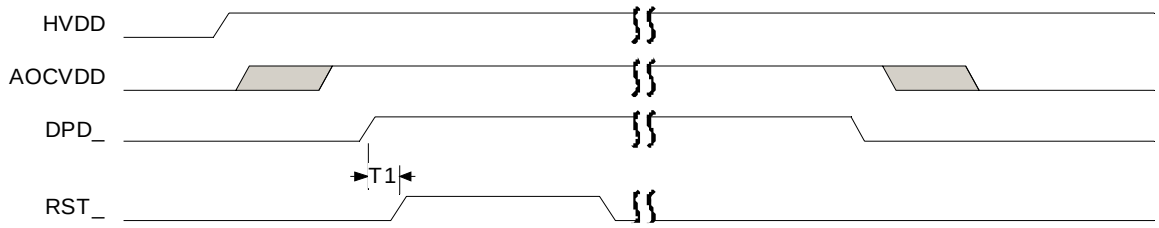


Figure 4.6: Power Sequence with DPD_ Case 2



As long as DPD_ and RST_ are driven as shown in these two figures, there are no requirements between the HVDD and AOCVDD sequencing. Care should be taken to turn off the GoForce 5500 clocks prior to powering-down the cores. T1 should be $\leq 1\ \mu s$.

The AOCVDD power-lost sequence should be as follows:

- GoForce 5500 in the idle state (no host CPU cycles)
- Pull DPD_ low
- Turn off the VDD core power

The AOCVDD power-regain sequence should be as follows:

- DPD_ low
- GoForce 5500 RST_ asserted
- VDD core power turns on
- DPD_ high
- GoForce 5500 RST_ de-asserted

Following the above sequencing avoids corrupting anything already on the Host Interface bus.

4.4.3.4 Registers and GFSDK Function Calls for Core and IO Power Sources

This information will be added in a later revision of this document.

4.4.4 Host Interface

Type A and Type C Host interfaces each have a set of timing parameters and diagrams, detailed in the following three sections. In the tables below all of the timing diagrams are listed and grouped according to Host Interface type. To go to a specific timing diagram, find the title under the appropriate host Interface type, and click it. It should take you directly to the chosen timing diagram.

Table 4.11: Type A Host Interface Timing Diagrams List

Indirect Addressing	
16bit Host Bus Interface	**32bit Host Bus Interface**
Register Write: 16Bit Indirect Type A: Page 4-12	Register Write, 32Bit Indirect Type A: Page 4-20
Register Read: 16Bit Indirect Type A: Page 4-13	Register Read, 32Bit Indirect Type A: Page 4-21
Memory Write: 16Bit Indirect Type A: Page 4-14	Memory Write 32Bit Indirect Type A: Page 4-22
Memory Read: 16Bit Indirect Type A: Page 4-15	Memory Read 32Bit Indirect Type A: Page 4-23
Register Read: Auto-increment 16Bit Indirect Type A: Page 4-16	Register Read: Auto-increment 32Bit Indirect Type A: Page 4-24
Memory Read: Auto-increment 16Bit Indirect Type A: Page 4-17	Memory Read: Auto-increment 32Bit Indirect Type A: Page 4-25
Register Write: Auto-increment 16Bit Indirect Type A: Page 4-18	Register Write: Auto-increment 32Bit Indirect Type A: Page 4-26
Memory Write: Auto-increment 16Bit Indirect Type A: Page 4-19	Memory Write: Auto-increment 32Bit Indirect Type A: Page 4-27
Direct Addressing	
16bit Host Bus Interface	**32bit Host Bus Interface**
WR_-controlled Write: 16Bit Direct Type A: Page 4-29	WR_-controlled Write: 32Bit Direct Type A: Page 4-35
CS_-controlled Write: 16Bit Direct Type A: Page 4-30	CS_-controlled Write: 32Bit Direct Type A: Page 4-36
RD_-controlled Read: 16Bit Direct Type A: Page 4-31	RD_-controlled Read: 32Bit Direct Type A: Page 4-37
CS_-controlled Read: 16Bit Direct Type A: Page 4-32	CS_-controlled Read: 32Bit Direct Type A: Page 4-38
Register or Memory Burst Write: 16Bit Direct Type A: Page 4-33	Register or Memory Burst Write: 32Bit Direct Type A: Page 4-39
Register or Memory Burst Read: 16Bit Direct Type A: Page 4-34	Register or Memory Burst Read: 32Bit Direct Type A: Page 4-40

Table 4.12: Type C Host Interface Timing Diagrams List

Indirect Addressing	
16bit Host Bus Interface	**32bit Host Bus Interface**
Register Write: 16Bit Indirect Type C: Page 4-44	Register Write, 32Bit Indirect Type C: Page 4-52
Register Read: 16Bit Indirect Type C: Page 4-45	Register Read, 32Bit Indirect Type C: Page 4-53
Memory Write: 16Bit Indirect Type C: Page 4-46	Memory Write, 32Bit Indirect Type C: Page 4-54
Memory Read: 16Bit Indirect Type C: Page 4-47	Memory Read: 32Bit Indirect Type C: Page 4-55
Register Auto-increment Read: 16Bit Indirect Type C: Page 4-48	Register Auto-increment Read: 32Bit Indirect Type C: Page 4-56
Memory Auto-increment Read: 16Bit Indirect Type C: Page 4-49	Memory Auto-increment Read: 32Bit Indirect Type C: Page 4-57
Register Auto-increment Write: 16Bit Indirect Type C: Page 4-50	Register Auto-increment Write: 32Bit Indirect Type C: Page 4-58
Memory Auto-increment Write: 16Bit Indirect Type C: Page 4-51	Memory Auto-increment Write: 32Bit Indirect Type C: Page 4-59
Direct Addressing	
16bit Host Bus Interface	**32bit Host Bus Interface**
WE_-controlled Write: 16Bit Direct Type C: Page 4-61	WE_-controlled Write: 32Bit Direct Type C: Page 4-67
CS_-controlled Write: 16Bit Direct Type C: Page 4-62	CS_-controlled Write: 32Bit Direct Type C: Page 4-68
OE_-controlled Read: 16Bit Direct Type C: Page 4-63	OE_-controlled Read: 32Bit Direct Type C: Page 4-69
CS_-controlled Read: 16Bit Direct Type C: Page 4-64	CS_-controlled Read: 32Bit Direct Type C: Page 4-70
Register or Memory Burst Write: 16Bit Direct Type C: Page 4-65	Register or Memory Burst Write: 32Bit Direct Type C: Page 4-71
Register or Memory Burst Read: 16Bit Type C: Page 4-66	Register or Memory Burst Read: 32Bit Direct Type C: Page 4-72

4.4.4.1 Type A Host Interface

This section shows the timing characteristics for Type A Host Bus interface using both direct addressing and indirect addressing. The interface utilizes either a 16 bit or 32-bit bus.

- Indirect Addressing Read/Write Timing Diagrams
  - 16Bit
  - 32Bit
- Direct Addressing Read/Write Timing Diagrams
  - 16Bit
  - 32Bit

All Type A Timing Diagrams refer to timing parameters in Table 4.30.

Note: The WAITn signal used in handshake mode, Burst, and Page writes and reads is tri-stated. Use either a pull-up or a pull-down resistor with this signal, according to the Host CPU specifications.

Table 4.13: Type A Byte-enable Signals for Different Size Host Busses

Type A Host Interface Signal Name	32bit Host Bus	16bit Host Bus
	Function	Function
BE_0	Byte Enable 1 [7:0]	Byte Enable 1 [7:0]
BE_1	Byte Enable 2 [15:8]	Byte Enable 2 [15:8]
BE_2	Byte Enable 3 [23:16]	Not used: Tie low or high externally
BE_3	Byte Enable 4 [31:24]	A1

4.4.4.1.1 Type A Indirect Timing Diagrams: 16Bit Interface

Figure 4.7: Register Write: 16Bit Indirect Type A

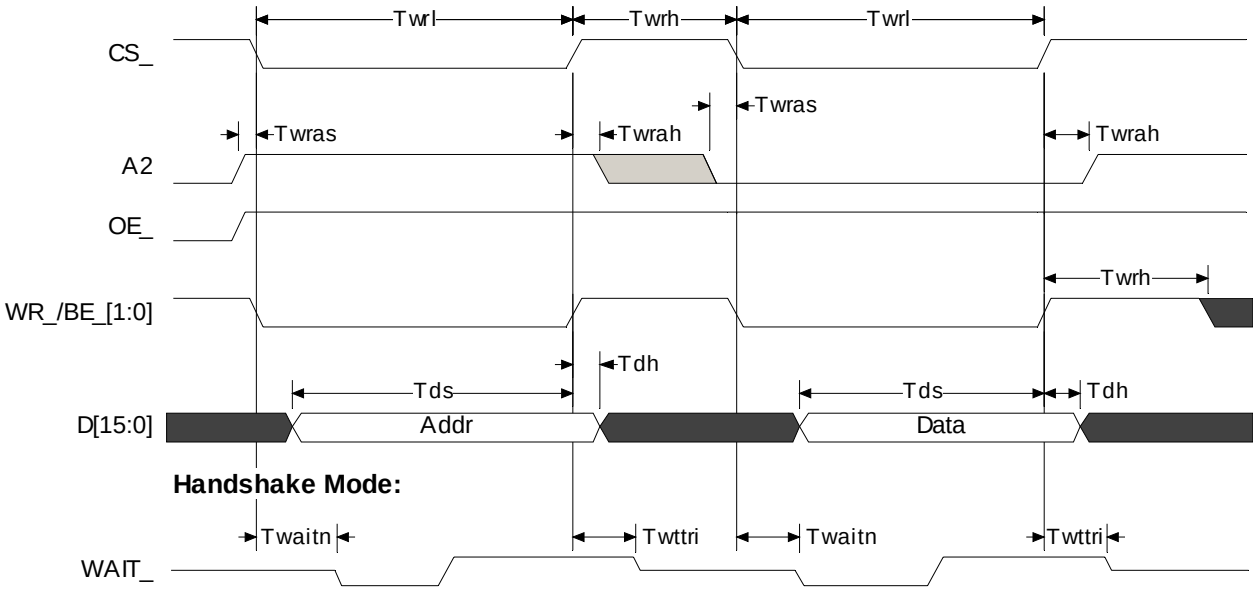


Table 4.14: Register Write D[15:0] Bit Mapping for Addr

Data Bus	Addr (Register 1)	Addr (Register 2)
D[15]	A[14]	X
D[14]	A[13]	X
D[13]	A[12]	X
D[12]	A[11]	X
D[11]	A[10]	X
D[10]	A[9]	X
D[9]	A[8]	X
D[8]	A[7]	X
D[7]	A[6]	X
D[6]	A[5]	X
D[5]	A[4]	X
D[4]	A[3]	A[17]
D[3]	A[2]	A[16]
D[2]	A[1]	A[15]
D[1]	0	1
D[0]	1	1

Figure 4.8: Register Read: 16Bit Indirect Type A

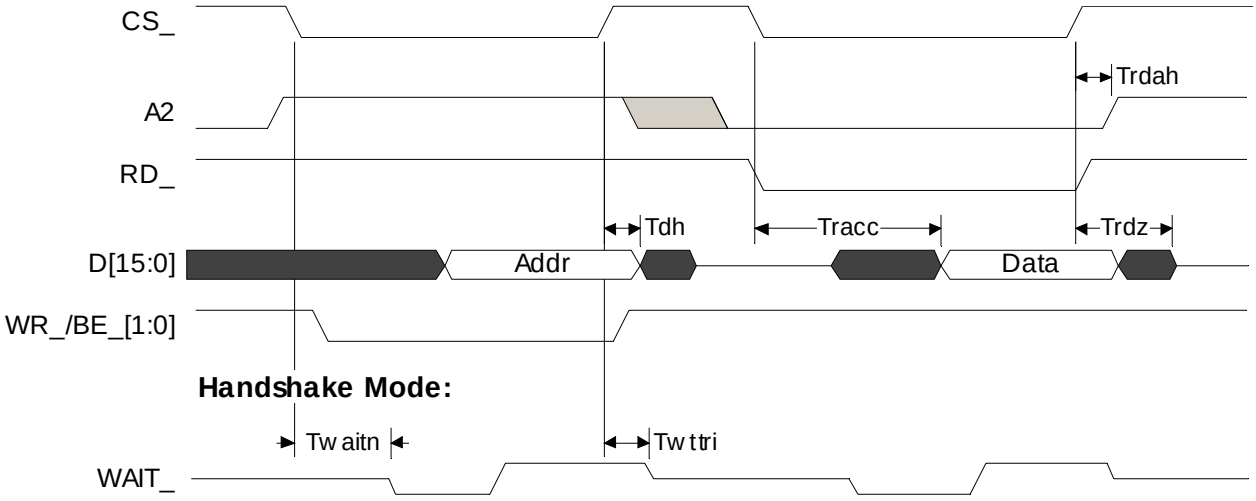


Table 4.15: Register Read D[15:0] Bit Mapping for Addr

Data Bus	Addr (Register 1)	Addr (Register 2)
D[15]	A[14]	X
D[14]	A[13]	X
D[13]	A[12]	X
D[12]	A[11]	X
D[11]	A[10]	X
D[10]	A[9]	X
D[9]	A[8]	X
D[8]	A[7]	X
D[7]	A[6]	X
D[6]	A[5]	X
D[5]	A[4]	X
D[4]	A[3]	A[17]
D[3]	A[2]	A[16]
D[2]	A[1]	A[15]
D[1]	0	1
D[0]	1	1

Figure 4.9: Memory Write: 16Bit Indirect Type A

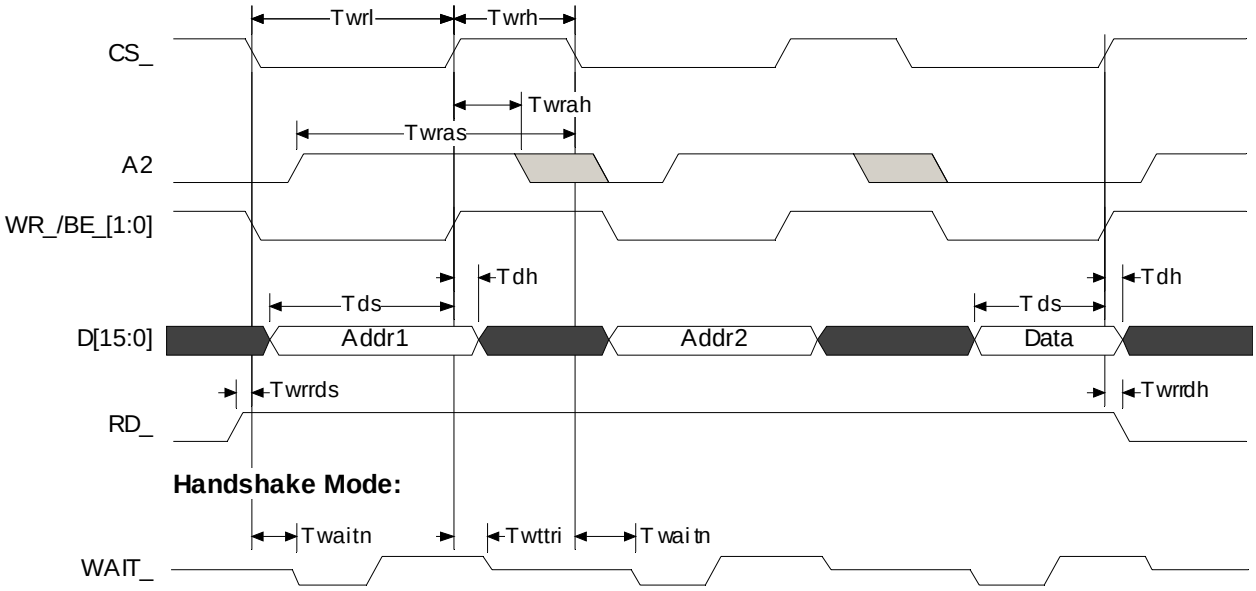


Table 4.16: Memory Write D[15:0] Bit Mapping for Addr1 and Addr2

Data Bus	Address Phase	
	Addr1	Addr2
D[15]	A[14]	X
D[14]	A[13]	X
D[13]	A[12]	X
D[12]	A[11]	A[25]
D[11]	A[10]	A[24]
D[10]	A[9]	A[23]
D[9]	A[8]	A[22]
D[8]	A[7]	A[21]
D[7]	A[6]	A[20]
D[6]	A[5]	A[19]
D[5]	A[4]	A[18]
D[4]	A[3]	A[17]
D[3]	A[2]	A[16]
D[2]	A[1]	A[15]
D[1]	0	1
D[0]	0	0

Figure 4.10: Memory Read: 16Bit Indirect Type A

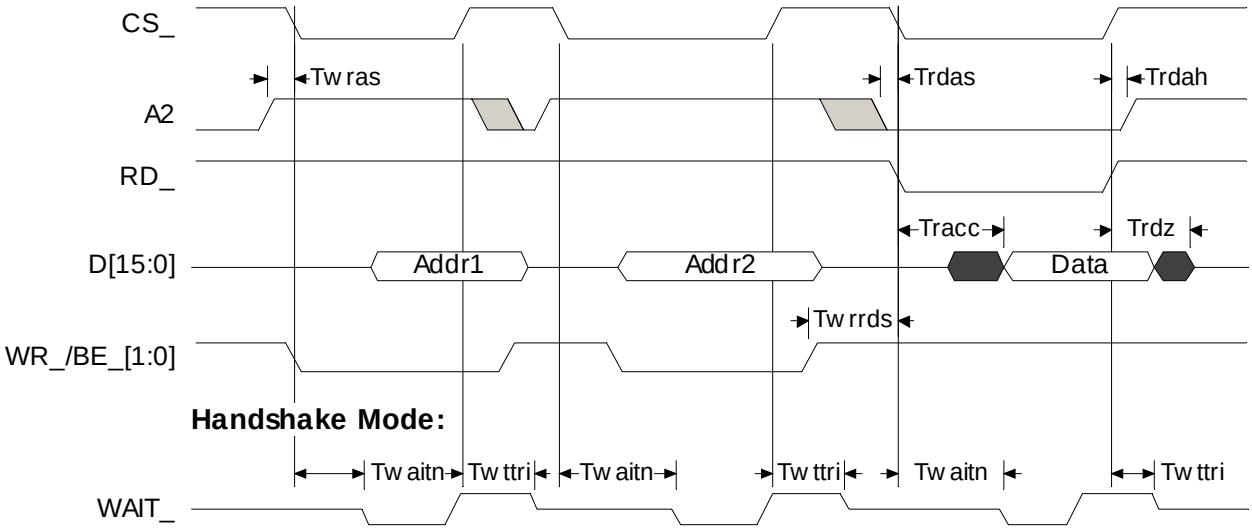


Table 4.17: Memory Read D[15:0] Bit Mapping for Addr1 and Addr2

Data Bus	Address Phase	
	Addr1	Addr2
D[15]	A[14]	X
D[14]	A[13]	X
D[13]	A[12]	X
D[12]	A[11]	A[25]
D[11]	A[10]	A[24]
D[10]	A[9]	A[23]
D[9]	A[8]	A[22]
D[8]	A[7]	A[21]
D[7]	A[6]	A[20]
D[6]	A[5]	A[19]
D[5]	A[4]	A[18]
D[4]	A[3]	A[17]
D[3]	A[2]	A[16]
D[2]	A[1]	A[15]
D[1]	0	1
D[0]	0	0

Figure 4.11: Register Read: Auto-increment 16Bit Indirect Type A

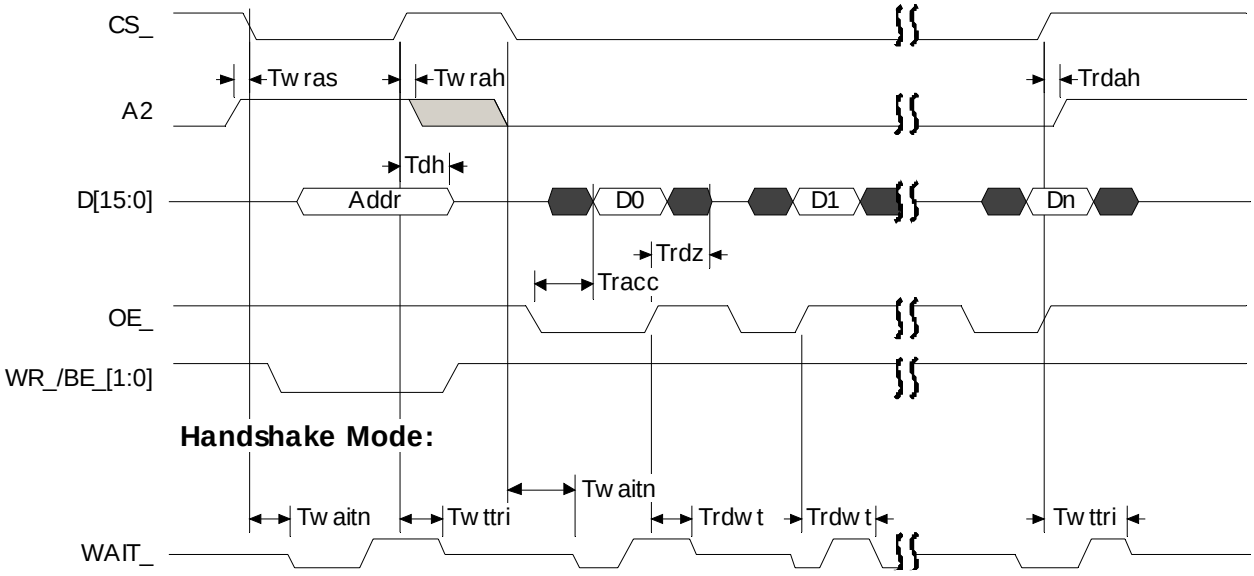


Table 4.18: Register Read D[15:0] Bit Mapping for Addr

Data Bus	Addr (Register 1)	Addr (Register 2)
D[15]	A[14]	X
D[14]	A[13]	X
D[13]	A[12]	X
D[12]	A[11]	X
D[11]	A[10]	X
D[10]	A[9]	X
D[9]	A[8]	X
D[8]	A[7]	X
D[7]	A[6]	X
D[6]	A[5]	X
D[5]	A[4]	X
D[4]	A[3]	A[17]
D[3]	A[2]	A[16]
D[2]	A[1]	A[15]
D[1]	0	1
D[0]	1	1

Figure 4.12: Memory Read: Auto-increment 16Bit Indirect Type A

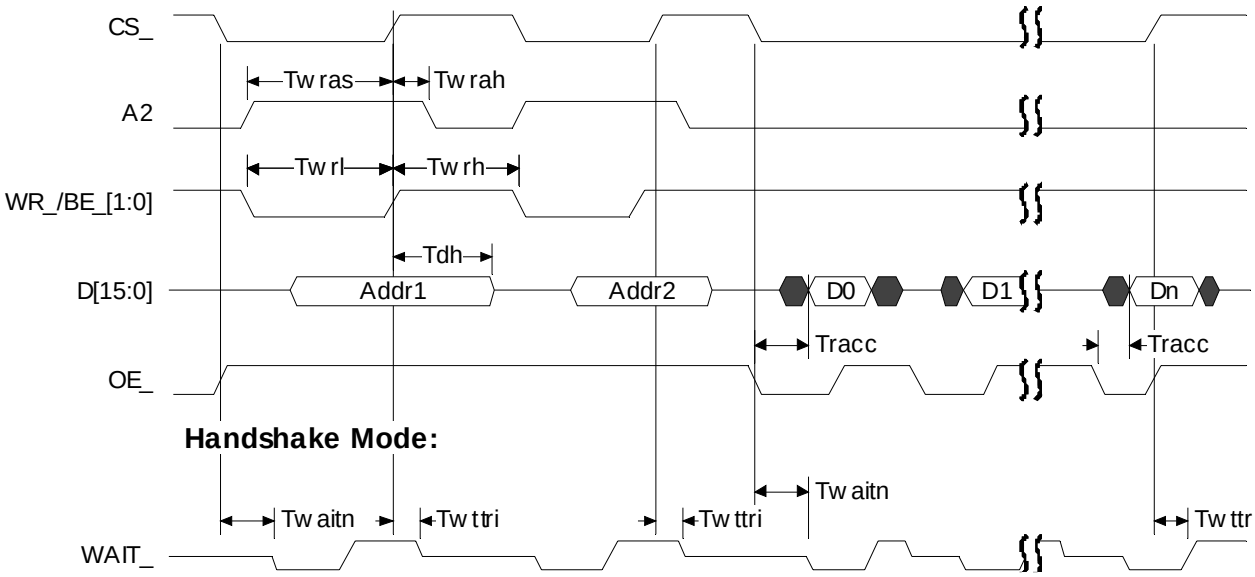


Table 4.19: Memory Read D[15:0] Bit Mapping for Addr1 and Addr2

Data Bus	Address Phase	
	Addr1	Addr2
D[15]	A[14]	X
D[14]	A[13]	X
D[13]	A[12]	X
D[12]	A[11]	A[25]
D[11]	A[10]	A[24]
D[10]	A[9]	A[23]
D[9]	A[8]	A[22]
D[8]	A[7]	A[21]
D[7]	A[6]	A[20]
D[6]	A[5]	A[19]
D[5]	A[4]	A[18]
D[4]	A[3]	A[17]
D[3]	A[2]	A[16]
D[2]	A[1]	A[15]
D[1]	0	1
D[0]	0	0

Figure 4.13: Register Write: Auto-increment 16Bit Indirect Type A

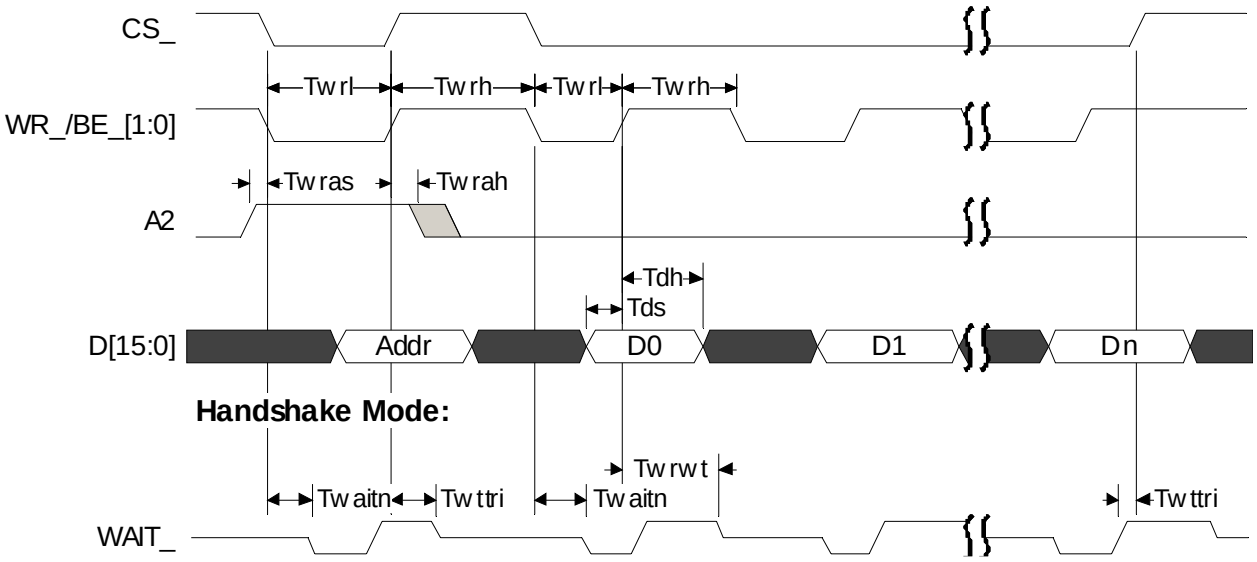


Table 4.20: Register Write D[15:0] Bit Mapping for Addr

Data Bus	Addr (Register 1)	Addr (Register 2)
D[15]	A[14]	X
D[14]	A[13]	X
D[13]	A[12]	X
D[12]	A[11]	X
D[11]	A[10]	X
D[10]	A[9]	X
D[9]	A[8]	X
D[8]	A[7]	X
D[7]	A[6]	X
D[6]	A[5]	X
D[5]	A[4]	X
D[4]	A[3]	A[17]
D[3]	A[2]	A[16]
D[2]	A[1]	A[15]
D[1]	0	1
D[0]	1	1

Figure 4.14: Memory Write: Auto-increment 16Bit Indirect Type A

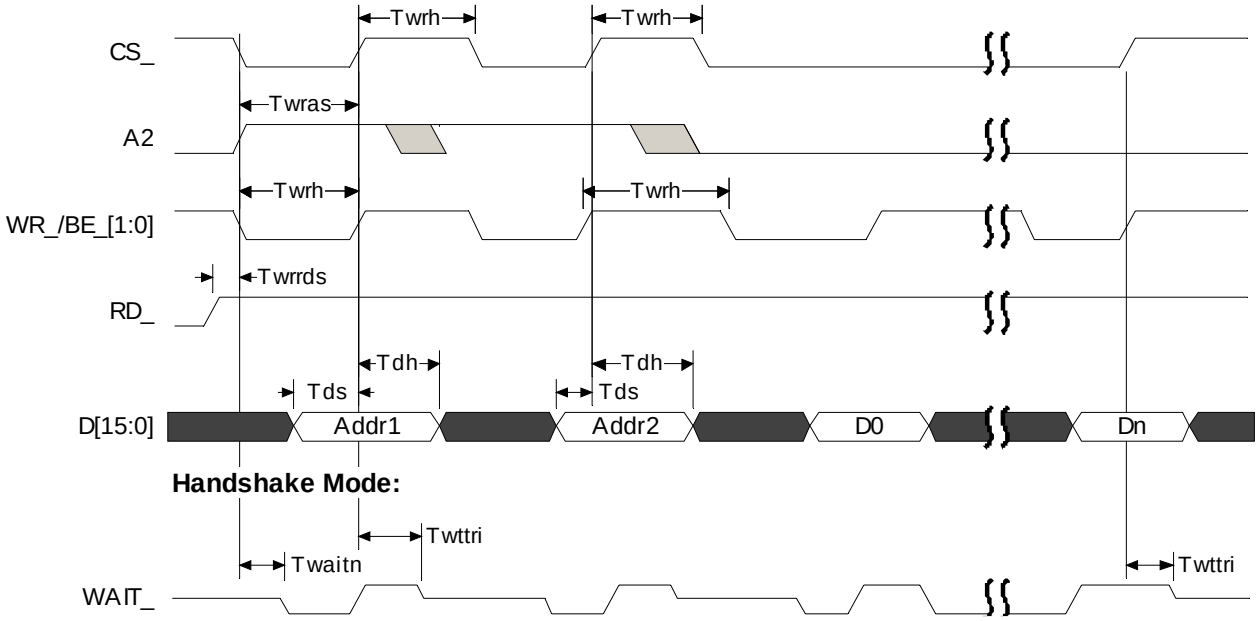


Table 4.21: Memory Write D[15:0] Bit Mapping for Addr1 and Addr2

Data Bus	Address Phase	
	Addr1	Addr2
D[15]	A[14]	X
D[14]	A[13]	X
D[13]	A[12]	X
D[12]	A[11]	A[25]
D[11]	A[10]	A[24]
D[10]	A[9]	A[23]
D[9]	A[8]	A[22]
D[8]	A[7]	A[21]
D[7]	A[6]	A[20]
D[6]	A[5]	A[19]
D[5]	A[4]	A[18]
D[4]	A[3]	A[17]
D[3]	A[2]	A[16]
D[2]	A[1]	A[15]
D[1]	0	1
D[0]	0	0

4.4.4.1.2 Type A Indirect Timing Diagrams: 32Bit Interface

Figure 4.15: Register Write, 32Bit Indirect Type A

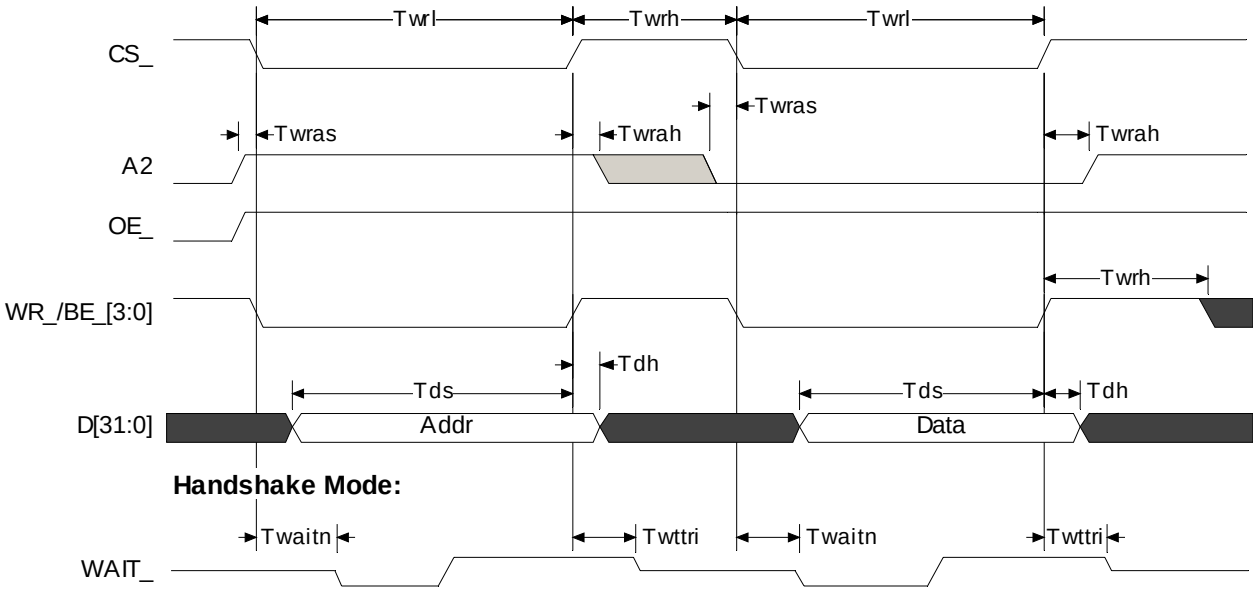


Table 4.22: Register Write D[31:0] Bit Mapping for Addr

Data Bus[31:16]	Addr	Data Bus[15:0]	Addr
D[31]	X	D[15]	A[16]
D[30]	X	D[14]	A[15]
D[29]	X	D[13]	A[14]
D[28]	X	D[12]	A[13]
D[27]	X	D[11]	A[12]
D[26]	X	D[10]	A[11]
D[25]	X	D[9]	A[10]
D[24]	X	D[8]	A[9]
D[23]	X	D[7]	A[8]
D[22]	X	D[6]	A[7]
D[21]	X	D[5]	A[6]
D[20]	X	D[4]	A[5]
D[19]	X	D[3]	A[4]
D[18]	X	D[2]	A[3]
D[17]	X	D[1]	A[2]
D[16]	A[17]	D[0]	1

Figure 4.16: Register Read, 32Bit Indirect Type A

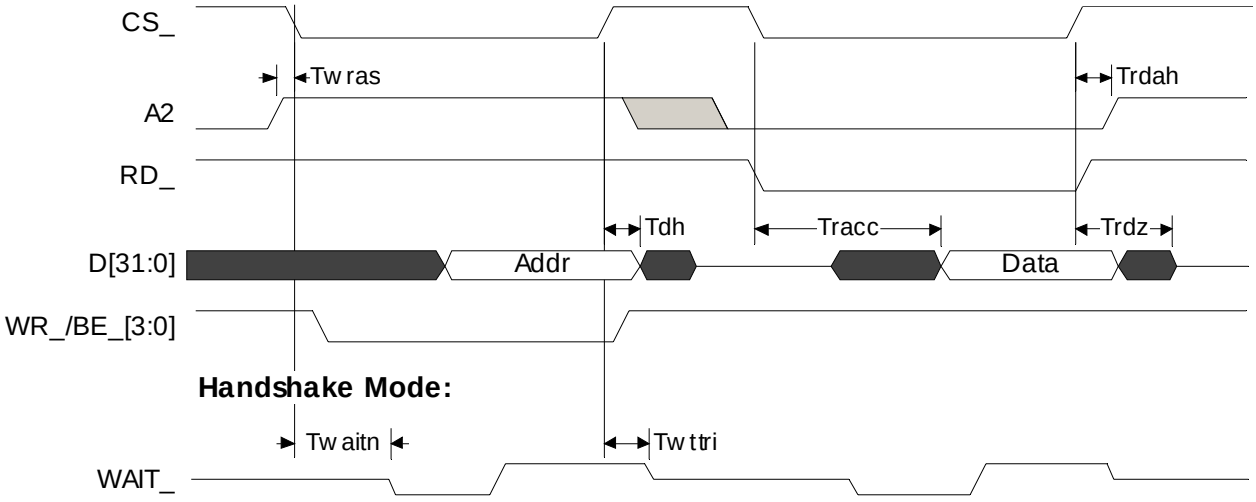


Table 4.23: Register Read D[31:0] Bit Mapping for Addr

Data Bus[31:16]	Addr	Data Bus[15:0]	Addr
D[31]	X	D[15]	A[16]
D[30]	X	D[14]	A[15]
D[29]	X	D[13]	A[14]
D[28]	X	D[12]	A[13]
D[27]	X	D[11]	A[12]
D[26]	X	D[10]	A[11]
D[25]	X	D[9]	A[10]
D[24]	X	D[8]	A[9]
D[23]	X	D[7]	A[8]
D[22]	X	D[6]	A[7]
D[21]	X	D[5]	A[6]
D[20]	X	D[4]	A[5]
D[19]	X	D[3]	A[4]
D[18]	X	D[2]	A[3]
D[17]	X	D[1]	A[2]
D[16]	A[17]	D[0]	1

Figure 4.17: Memory Write 32Bit Indirect Type A

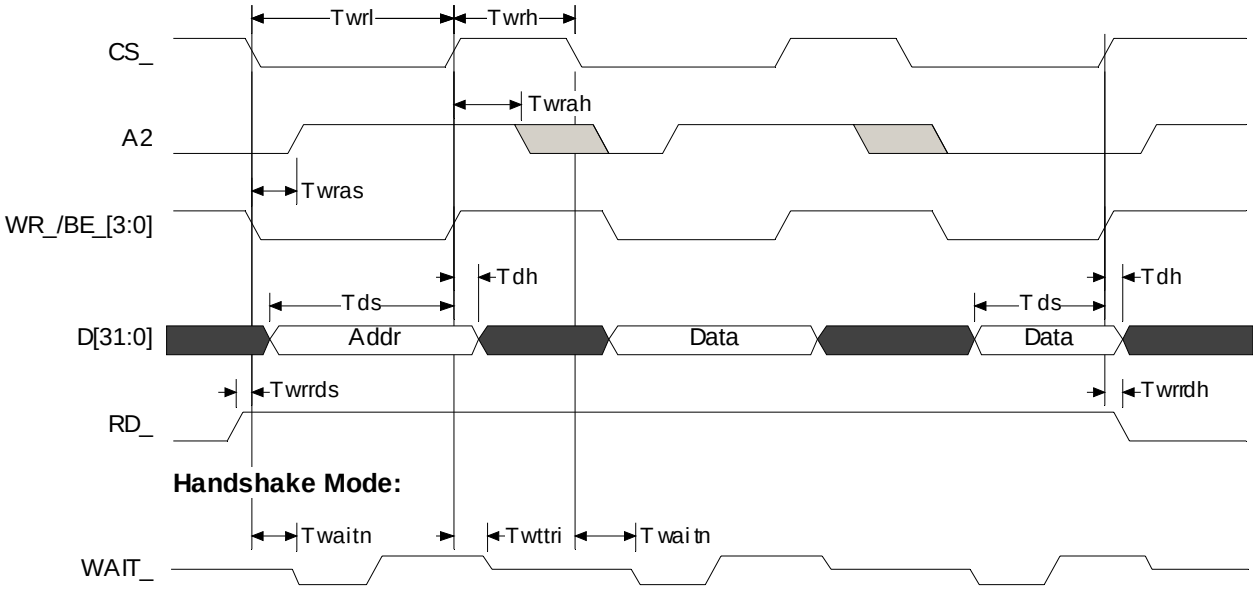


Table 4.24: Memory Write D[31:0] Bit Mapping for Addr

Data Bus[31:16]	Addr	Data Bus[15:0]	Addr
D[31]	X	D[15]	A[16]
D[30]	X	D[14]	A[15]
D[29]	X	D[13]	A[14]
D[28]	X	D[12]	A[13]
D[27]	X	D[11]	A[12]
D[26]	X	D[10]	A[11]
D[25]	X	D[9]	A[10]
D[24]	A[25]	D[8]	A[9]
D[23]	A[24]	D[7]	A[8]
D[22]	A[23]	D[6]	A[7]
D[21]	A[22]	D[5]	A[6]
D[20]	A[21]	D[4]	A[5]
D[19]	A[20]	D[3]	A[4]
D[18]	A[19]	D[2]	A[3]
D[17]	A[18]	D[1]	A[2]
D[16]	A[17]	D[0]	0

Figure 4.18: Memory Read 32Bit Indirect Type A

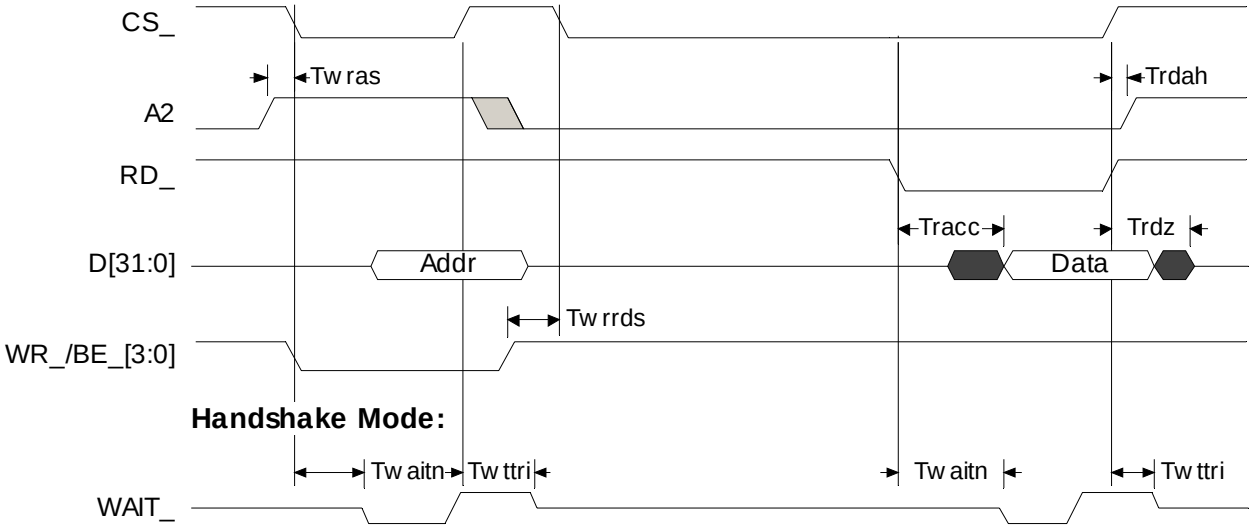


Table 4.25: Memory Read D[31:0] Bit Mapping for Addr

Data Bus[31:16]	Addr	Data Bus[15:0]	Addr
D[31]	X	D[15]	A[16]
D[30]	X	D[14]	A[15]
D[29]	X	D[13]	A[14]
D[28]	X	D[12]	A[13]
D[27]	X	D[11]	A[12]
D[26]	X	D[10]	A[11]
D[25]	X	D[9]	A[10]
D[24]	A[25]	D[8]	A[9]
D[23]	A[24]	D[7]	A[8]
D[22]	A[23]	D[6]	A[7]
D[21]	A[22]	D[5]	A[6]
D[20]	A[21]	D[4]	A[5]
D[19]	A[20]	D[3]	A[4]
D[18]	A[19]	D[2]	A[3]
D[17]	A[18]	D[1]	A[2]
D[16]	A[17]	D[0]	0

Figure 4.19: Register Read: Auto-increment 32Bit Indirect Type A

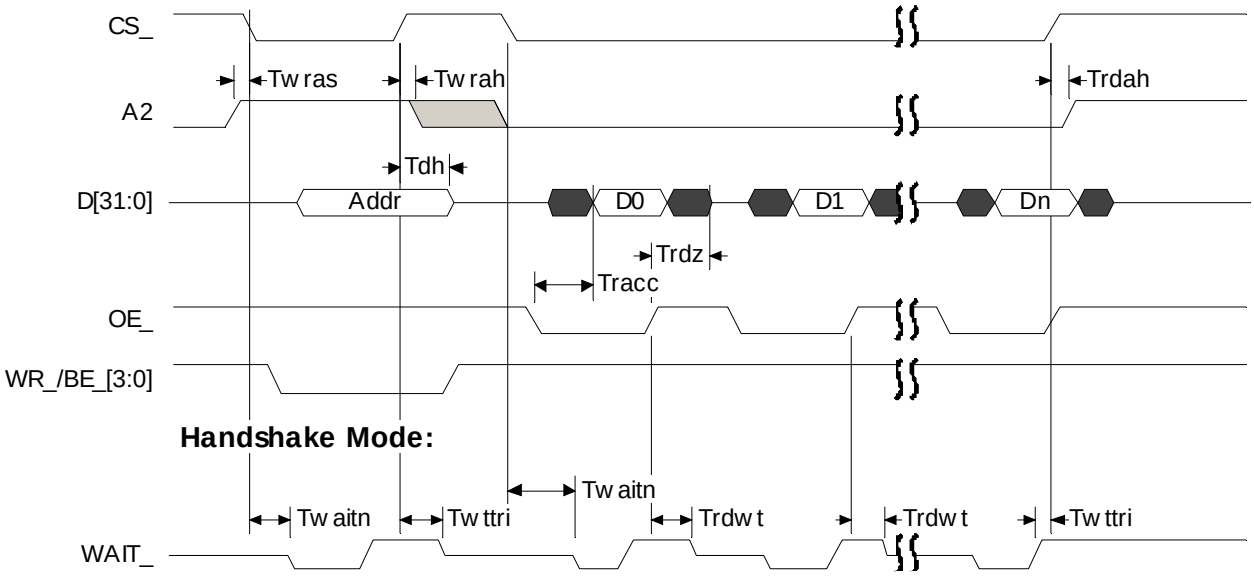


Table 4.26: Register Read D[31:0] Bit Mapping for Addr

Data Bus[31:16]	Addr	Data Bus[15:0]	Addr
D[31]	X	D[15]	A[16]
D[30]	X	D[14]	A[15]
D[29]	X	D[13]	A[14]
D[28]	X	D[12]	A[13]
D[27]	X	D[11]	A[12]
D[26]	X	D[10]	A[11]
D[25]	X	D[9]	A[10]
D[24]	X	D[8]	A[9]
D[23]	X	D[7]	A[8]
D[22]	X	D[6]	A[7]
D[21]	X	D[5]	A[6]
D[20]	X	D[4]	A[5]
D[19]	X	D[3]	A[4]
D[18]	X	D[2]	A[3]
D[17]	X	D[1]	A[2]
D[16]	A[17]	D[0]	1

Figure 4.20: Memory Read: Auto-increment 32Bit Indirect Type A

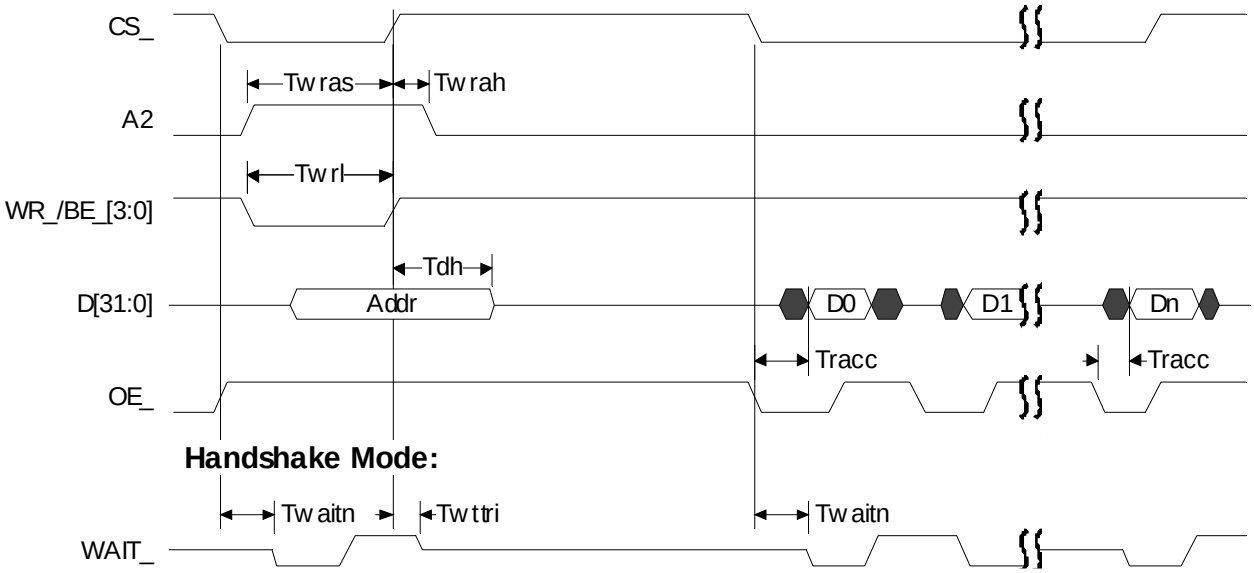


Table 4.27: Memory Read D[31:0] Bit Mapping for Addr

Data Bus[31:16]	Addr	Data Bus[15:0]	Addr
D[31]	X	D[15]	A[16]
D[30]	X	D[14]	A[15]
D[29]	X	D[13]	A[14]
D[28]	X	D[12]	A[13]
D[27]	X	D[11]	A[12]
D[26]	X	D[10]	A[11]
D[25]	X	D[9]	A[10]
D[24]	A[25]	D[8]	A[9]
D[23]	A[24]	D[7]	A[8]
D[22]	A[23]	D[6]	A[7]
D[21]	A[22]	D[5]	A[6]
D[20]	A[21]	D[4]	A[5]
D[19]	A[20]	D[3]	A[4]
D[18]	A[19]	D[2]	A[3]
D[17]	A[18]	D[1]	A[2]
D[16]	A[17]	D[0]	0

Figure 4.21: Register Write: Auto-increment 32Bit Indirect Type A

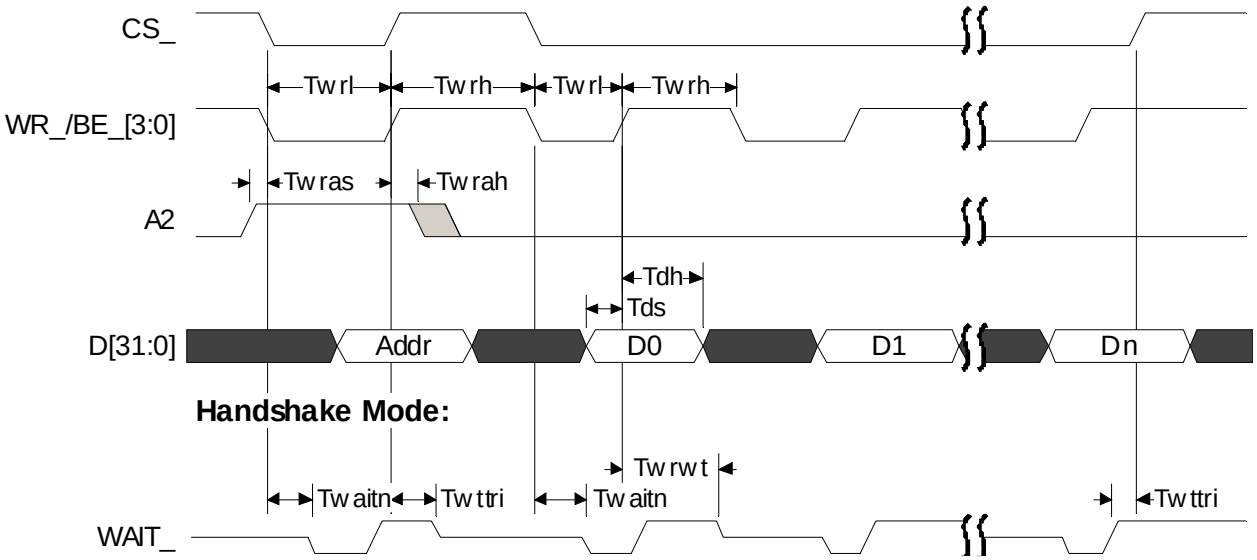


Table 4.28: Register Write D[31:0] Bit Mapping for Addr

Data Bus[31:16]	Addr	Data Bus[15:0]	Addr
D[31]	X	D[15]	A[16]
D[30]	X	D[14]	A[15]
D[29]	X	D[13]	A[14]
D[28]	X	D[12]	A[13]
D[27]	X	D[11]	A[12]
D[26]	X	D[10]	A[11]
D[25]	X	D[9]	A[10]
D[24]	X	D[8]	A[9]
D[23]	X	D[7]	A[8]
D[22]	X	D[6]	A[7]
D[21]	X	D[5]	A[6]
D[20]	X	D[4]	A[5]
D[19]	X	D[3]	A[4]
D[18]	X	D[2]	A[3]
D[17]	X	D[1]	A[2]
D[16]	A[17]	D[0]	1

Figure 4.22: Memory Write: Auto-increment 32Bit Indirect Type A

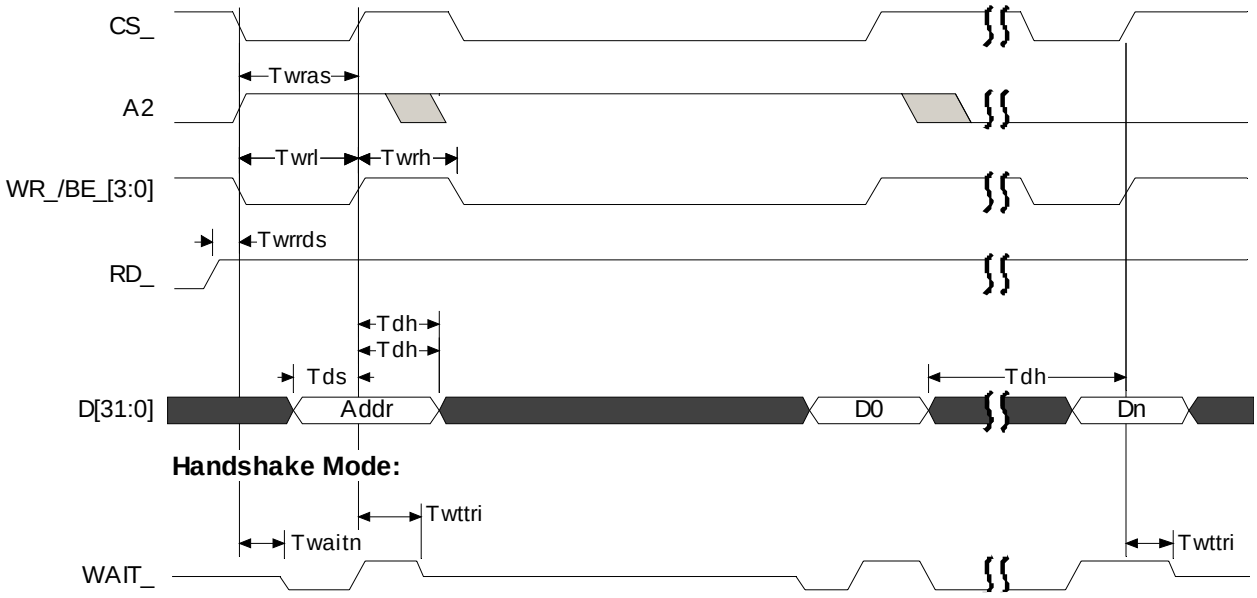
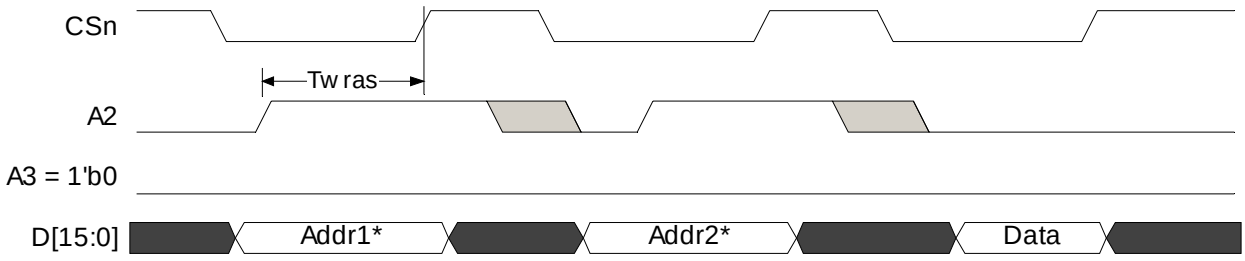


Table 4.29: Memory Write D[31:0] Bit Mapping for Addr

Data Bus[31:16]	Addr	Data Bus[15:0]	Addr
D[31]	X	D[15]	A[16]
D[30]	X	D[14]	A[15]
D[29]	X	D[13]	A[14]
D[28]	X	D[12]	A[13]
D[27]	X	D[11]	A[12]
D[26]	X	D[10]	A[11]
D[25]	X	D[9]	A[10]
D[24]	A[25]	D[8]	A[9]
D[23]	A[24]	D[7]	A[8]
D[22]	A[23]	D[6]	A[7]
D[21]	A[22]	D[5]	A[6]
D[20]	A[21]	D[4]	A[5]
D[19]	A[20]	D[3]	A[4]
D[18]	A[19]	D[2]	A[3]
D[17]	A[18]	D[1]	A[2]
D[16]	A[17]	D[0]	0

4.4.4.1.3 One and Two-channel Access for Indirect Addressing

Figure 4.23: One-channel Access, Indirect Addressing



Note: Recommendation when not using A3 as a secondary latch (i.e. single-channel access): Tie A3 to ground.

Figure 4.24: Two-channel Memory Access, Indirect Addressing

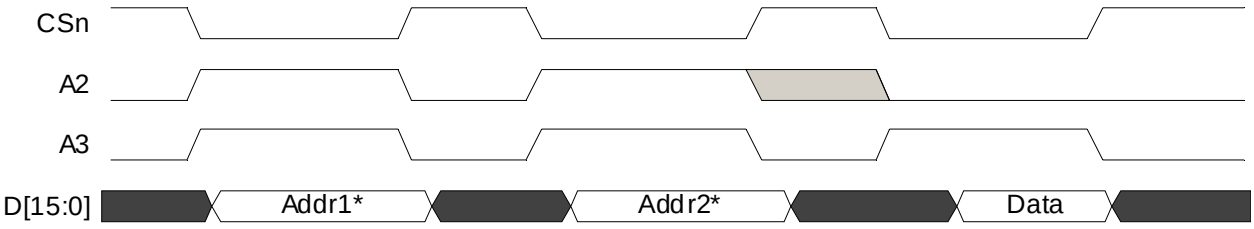
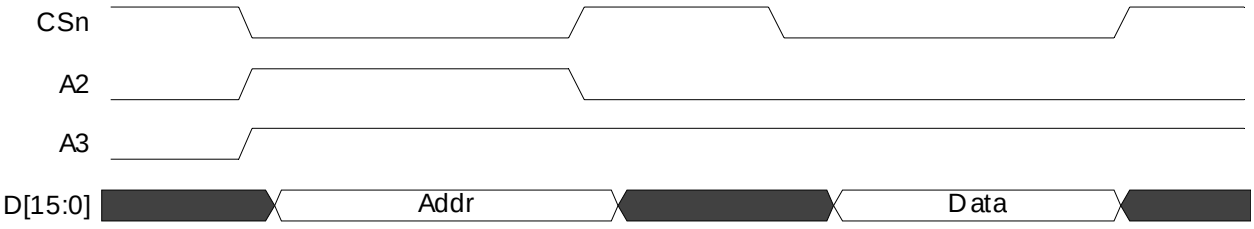


Figure 4.25: Two-channel Register Access, Indirect Addressing



Note in Figure 4.36 and Figure 4.37 that A1 and A2 must toggle with CSn. Also, A2 must be high during the data transmission phase, unlike A1.

4.4.4.1.4 Type A Direct Timing Diagrams: 16Bit Interface

Figure 4.26: WR_-controlled Write: 16Bit Direct Type A

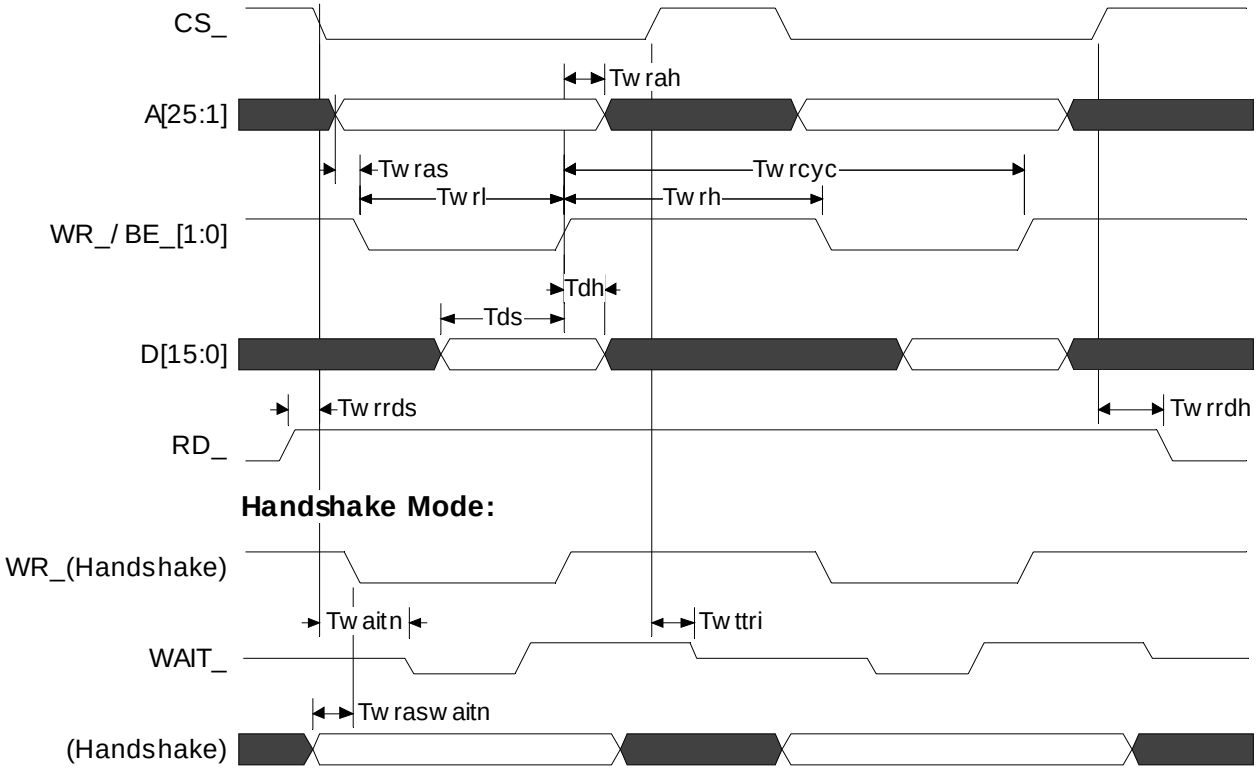


Figure 4.27: CS_-controlled Write: 16Bit Direct Type A

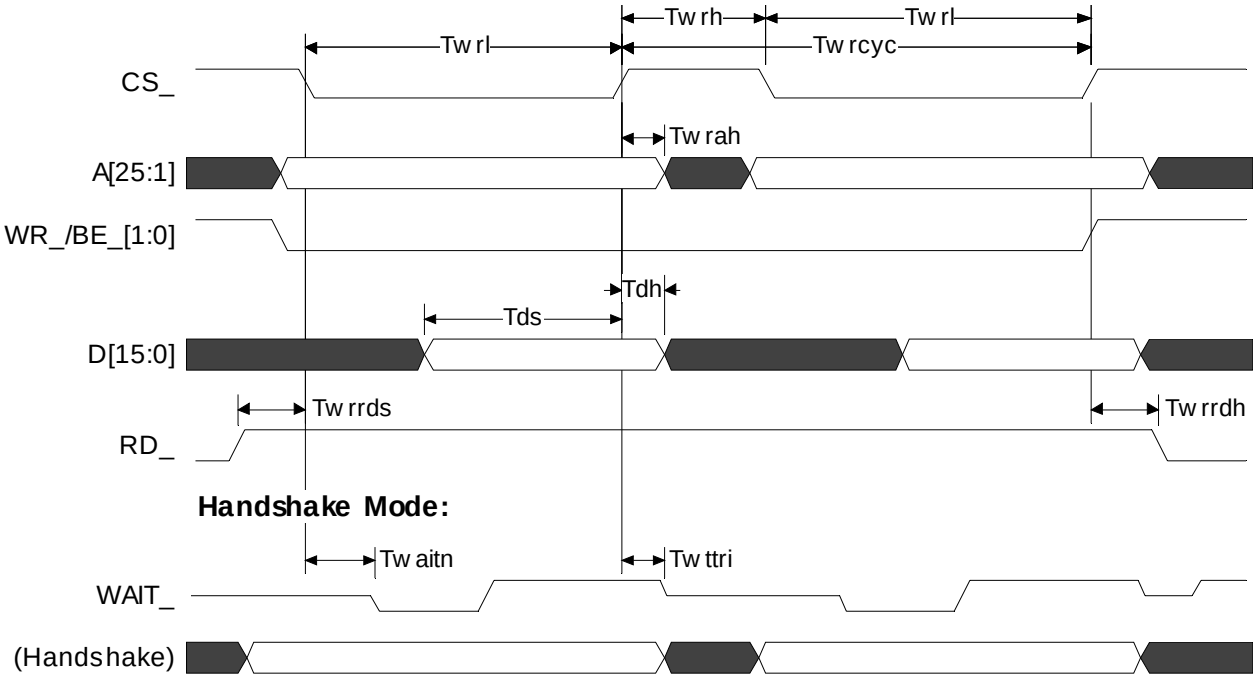


Figure 4.28: RD_-controlled Read: 16Bit Direct Type A

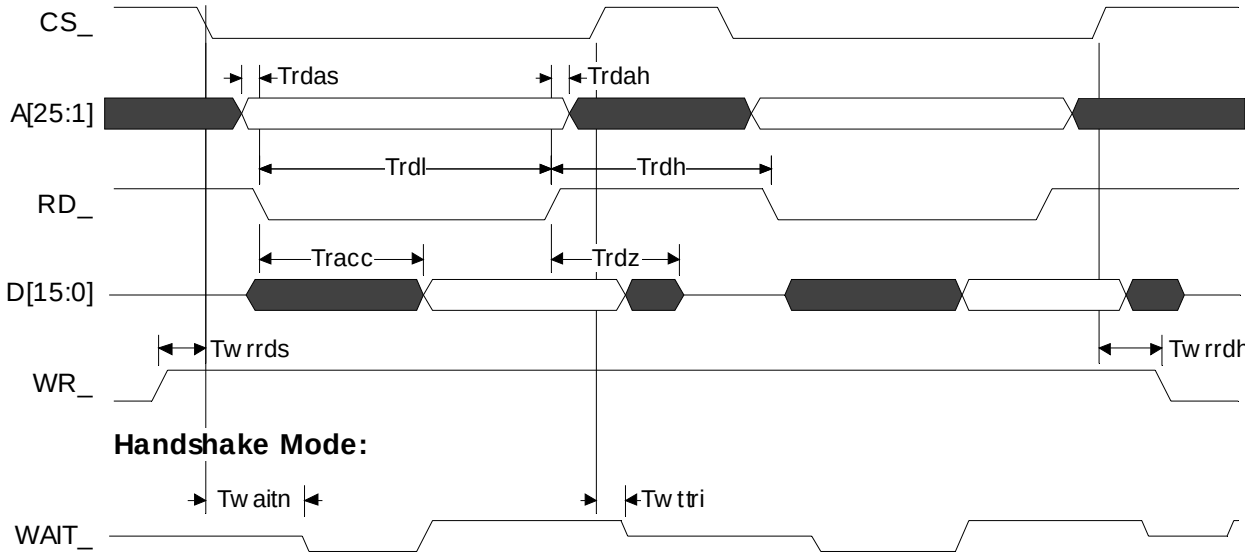


Figure 4.29: CS_-controlled Read: 16Bit Direct Type A

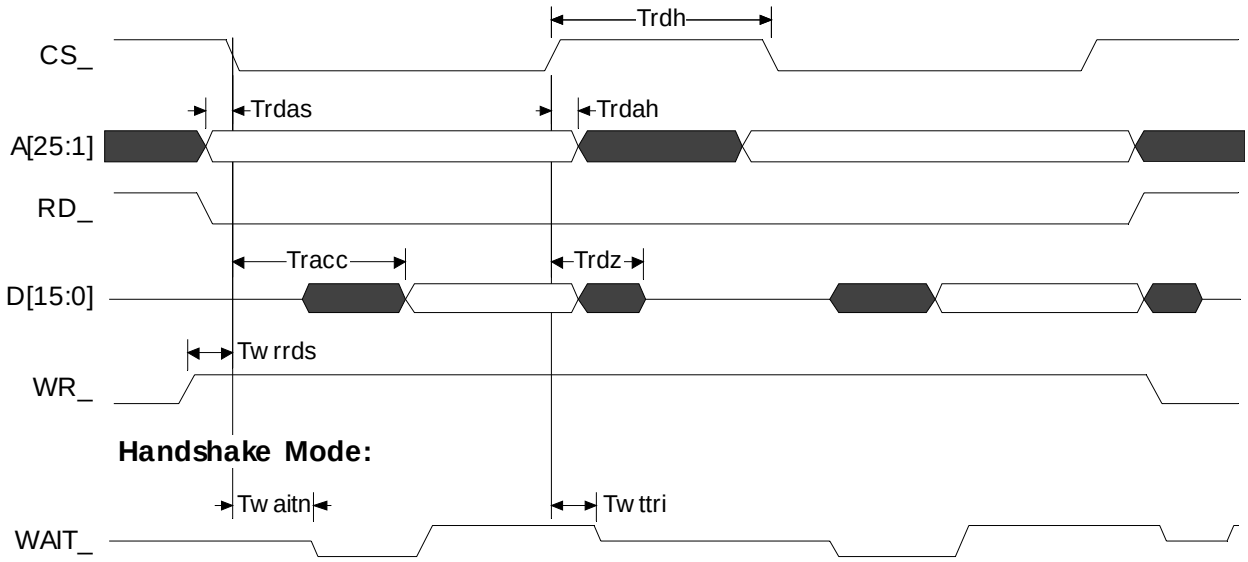


Figure 4.30: Register or Memory Burst Write: 16Bit Direct Type A

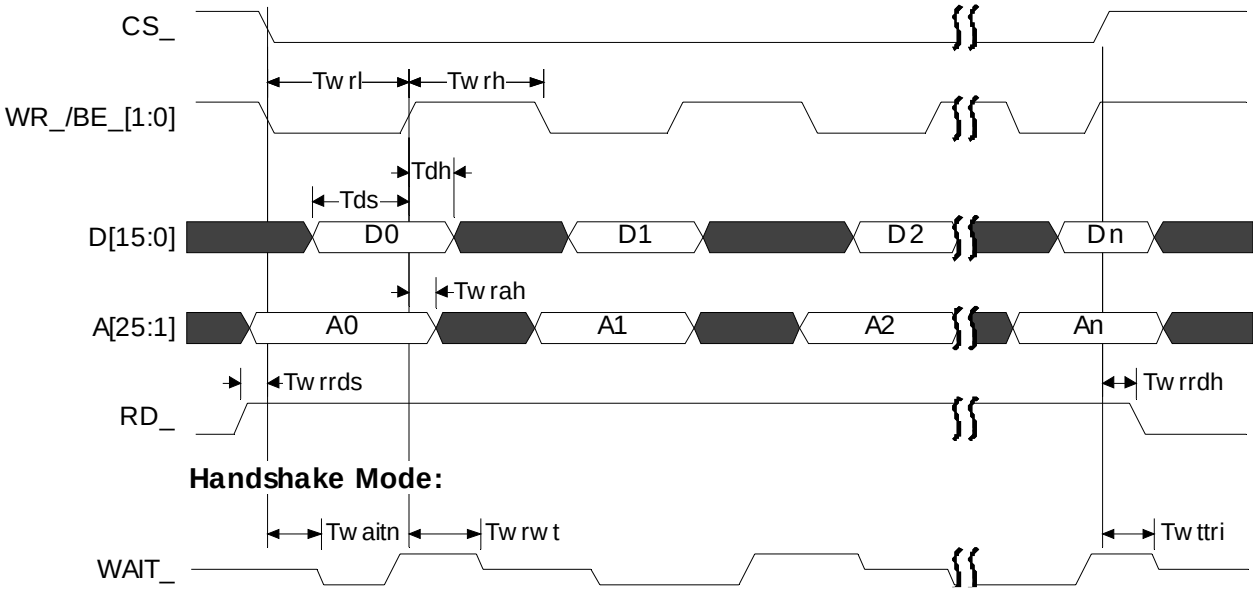
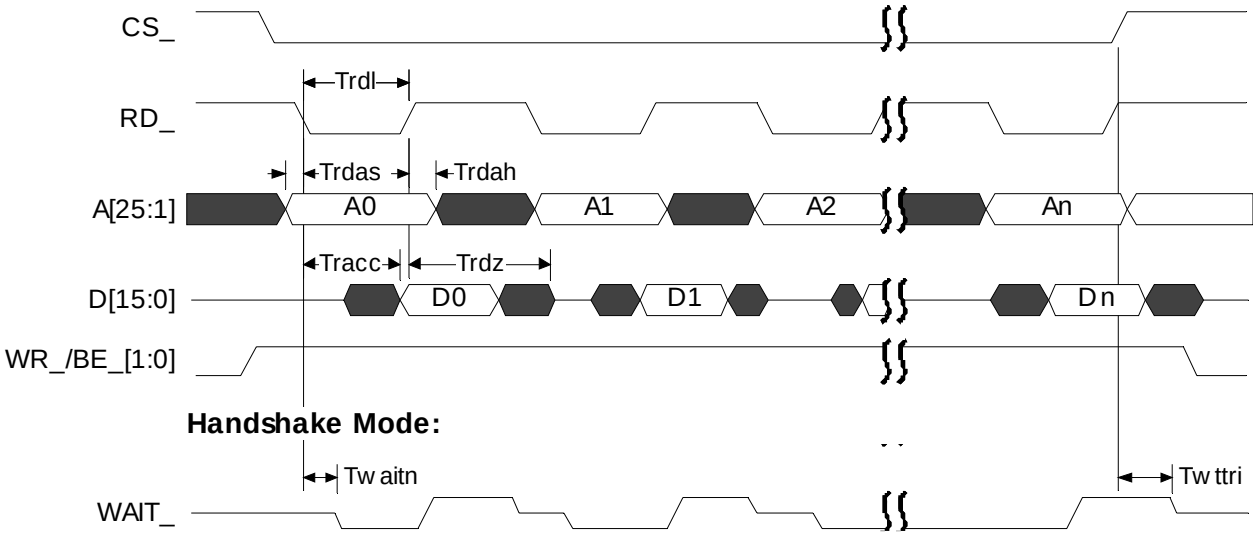


Figure 4.31: Register or Memory Burst Read: 16Bit Direct Type A



4.4.4.1.5 Type A Direct Timing Diagrams: 32Bit Interface

Figure 4.32: WR_-controlled Write: 32Bit Direct Type A

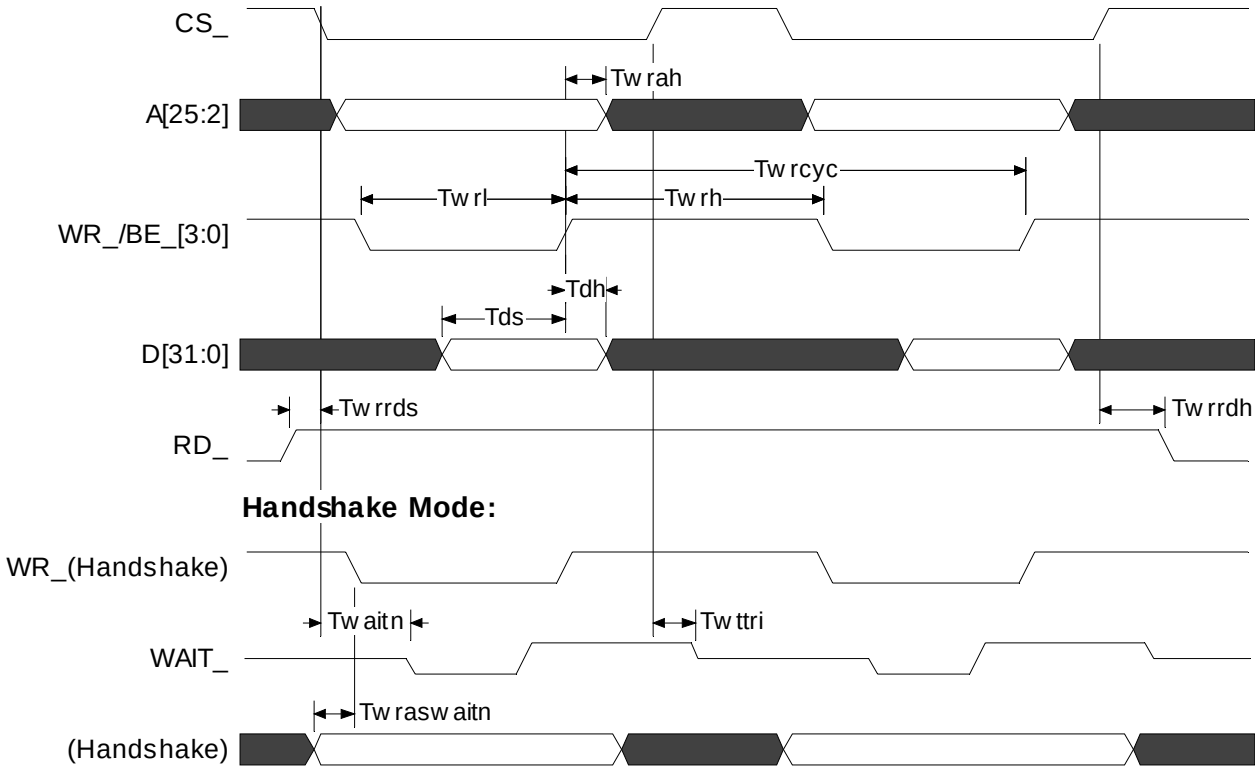


Figure 4.33: CS_-controlled Write: 32Bit Direct Type A

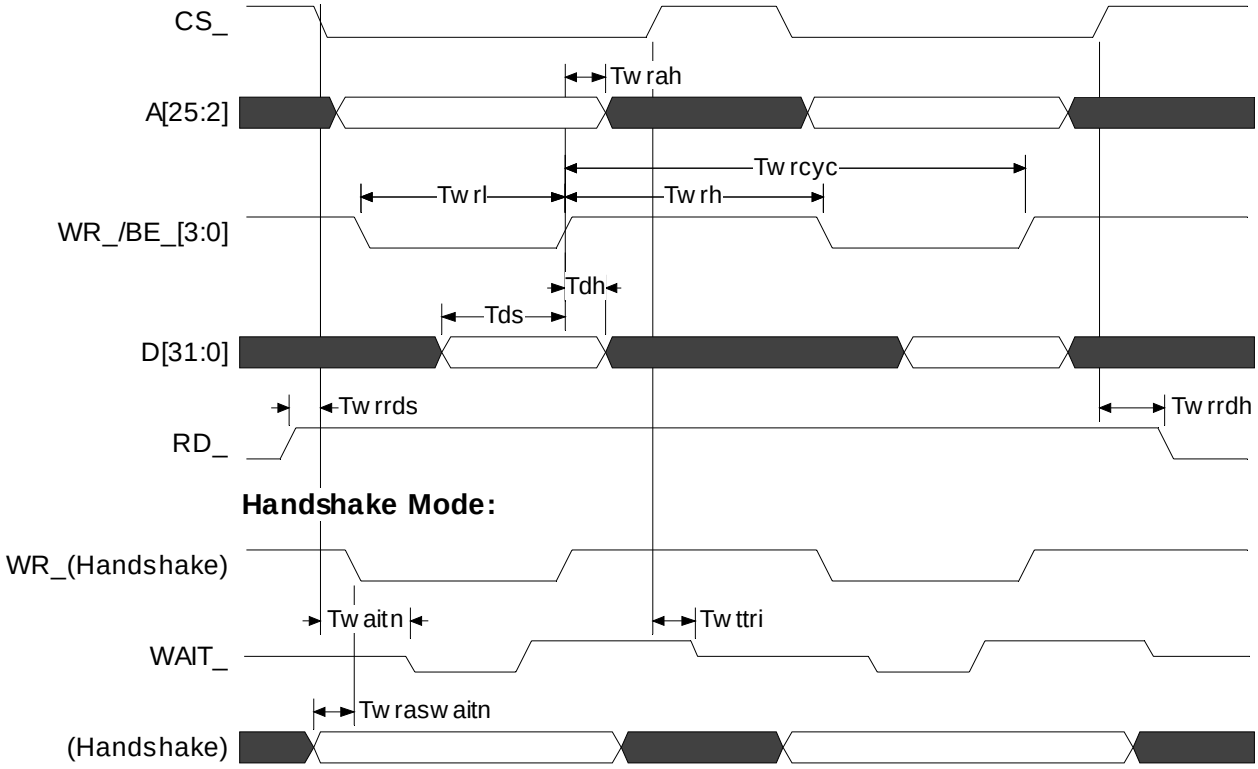


Figure 4.34: RD_-controlled Read: 32Bit Direct Type A

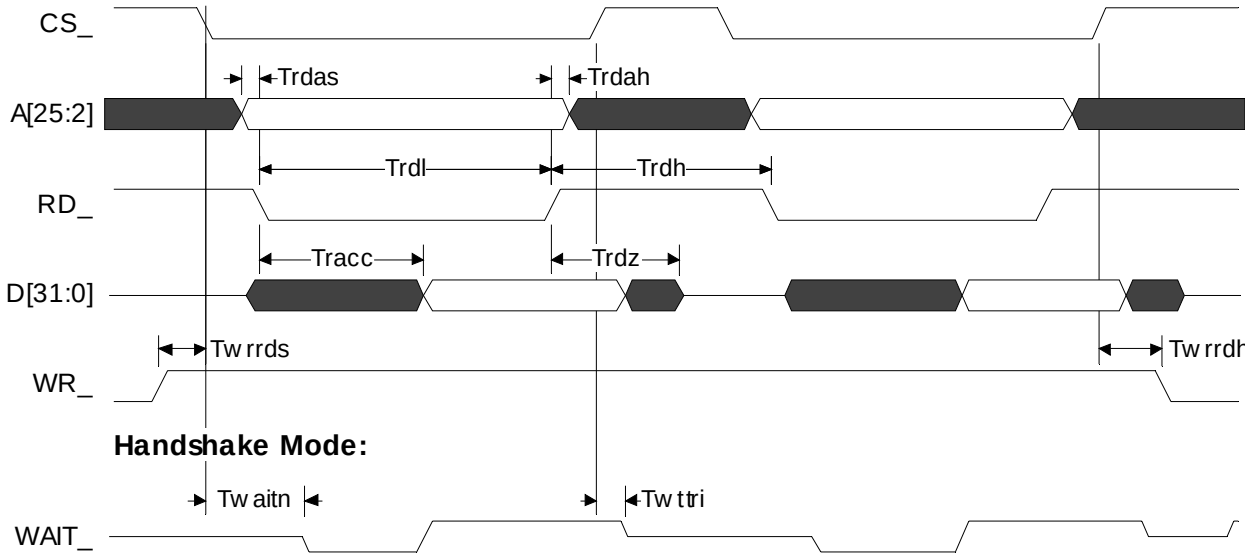


Figure 4.35: CS_-controlled Read: 32Bit Direct Type A

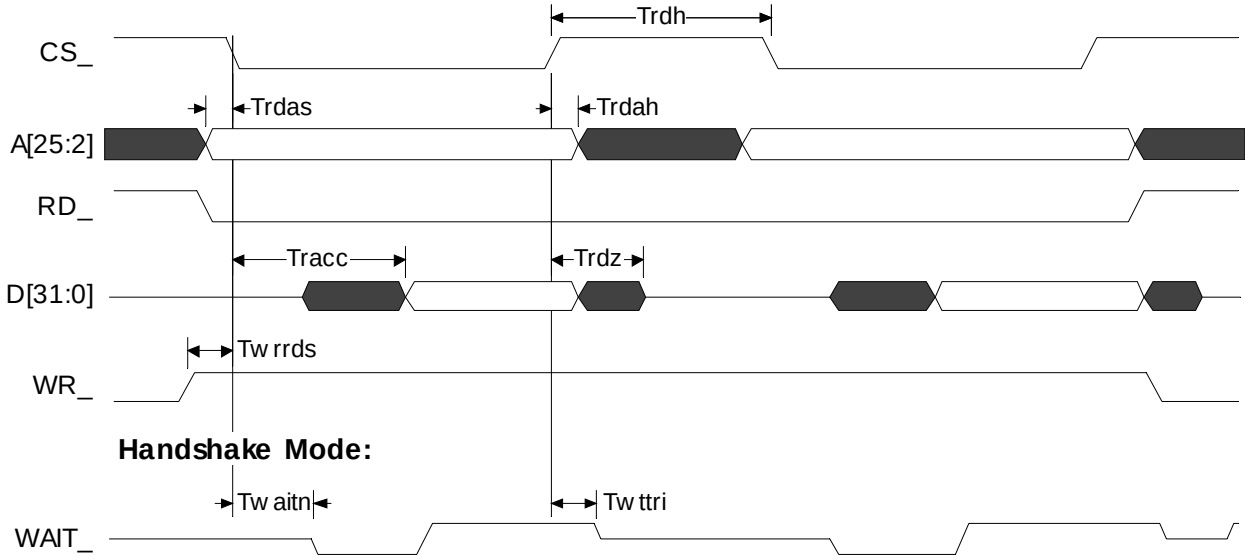


Figure 4.36: Register or Memory Burst Write: 32Bit Direct Type A

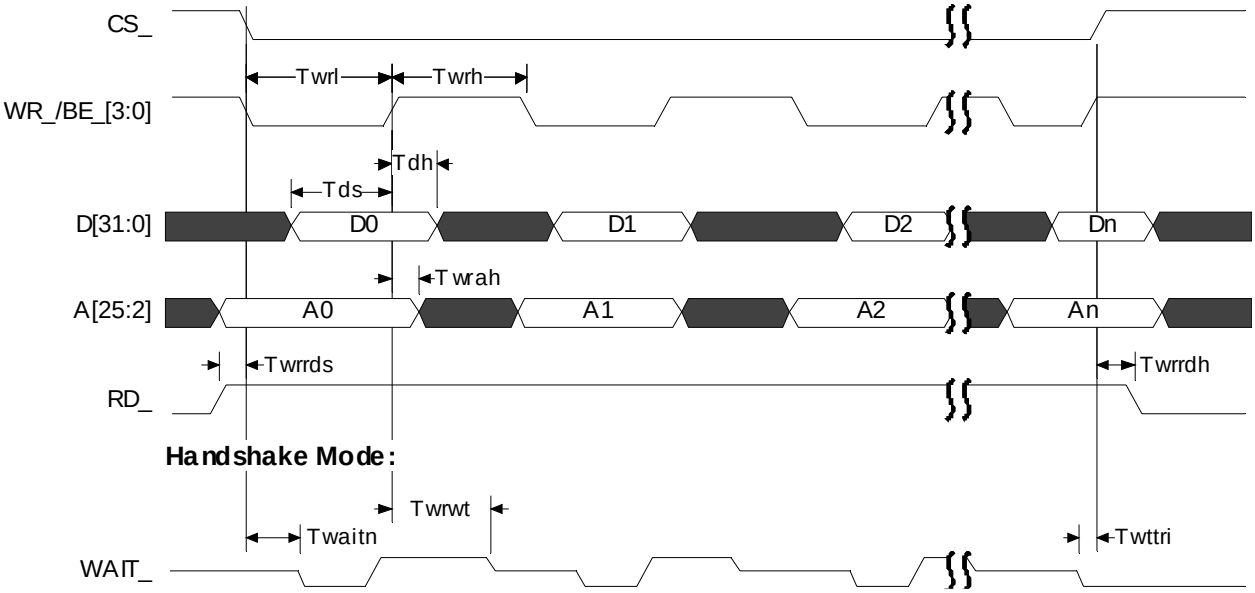
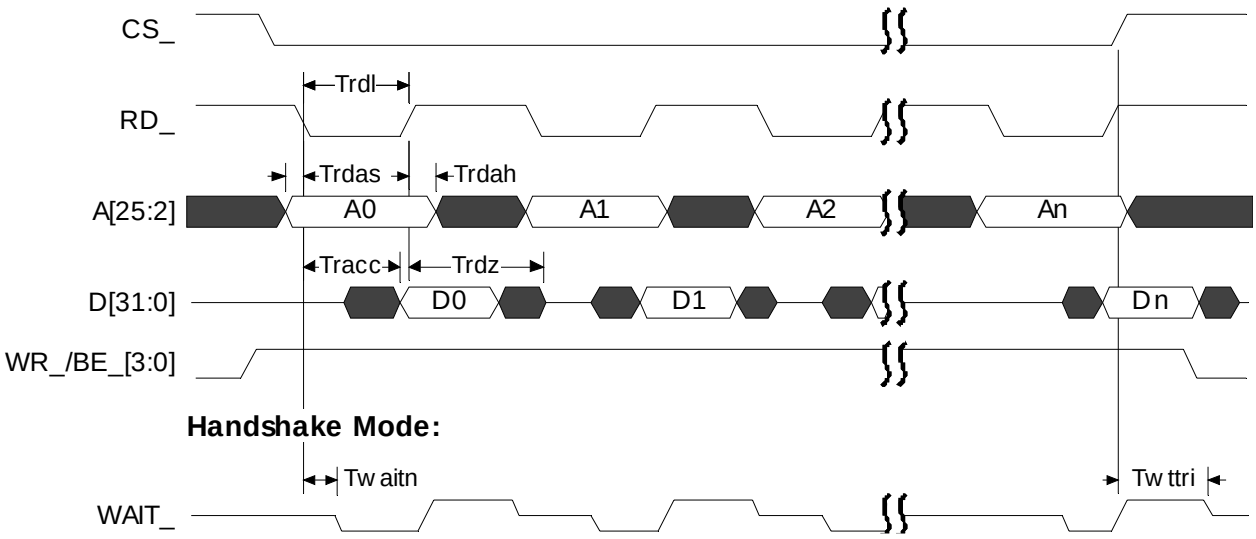


Figure 4.37: Register or Memory Burst Read: 32Bit Direct Type A



4.4.4.2 Type A Host Interface Timing Parameters

Table 4.30 provides the AC timing parameters for the preceding Type A host interface timing diagrams.

Note: **TM** is the memory clock period in ns.
TF is the FIFO clock period in ns, defined according to the following:

- *VI FIFO Status Register:* The frequency of TF is the VI block frequency
- *All other FIFO Status Registers:* The frequency of TF is equal to that of TM.

Table 4.30: Type A Host Interface Timing Parameters

Symbol	Description	Min (ns): Time and Conditions		Max (ns): Time and Conditions
Tdh	**Write cycles:** Data hold time from rising edge of WR_/CS_, whichever comes first	0		N/A
Tds	**Write cycles:** Data setup time to rising edge of WR_/CS_ whichever comes first.	7		N/A
Tracc	**Read cycles:** Maximum read access time from the beginning of the read cycle to the first valid data access.	N/A		Asynchronous Register Access: 26
				Synchronous Register Access: (5*Host Clock) + 21
				SRAM Access: (7*Memory clock) + 21
Trdah	**Read cycles:** Address hold time from rising edge of CS_/RD_, whichever comes first.	0		N/A
Trdas	**Read cycles:** Address setup time to falling edge of CS_/RD_, whichever comes last.	0		N/A
Trdh	**Read cycles:** Read enable Inactive time measured from the end of one read cycle to the beginning of the next read cycle.	5	No handshake	N/A
		26	Handshake	
Trdl	**Read cycles:** Read enable active low time. CS_-controlled read cycles: CS_ low time RD_-controlled read cycles: RD_ low time	Asynchronous Register Access: 26		N/A
		Synchronous Register Access: (5*Host Clock) + 21		
		SRAM Access: (7*Memory clock) + 21		
Trdwt	Time from rising edge of RD_ to falling edge of WAIT_	N/A		20
Trdz	Time from rising edge of CS_ or RD_, whichever comes first, to the data bus floating state	3		15
Twaitn	WAIT_/RDY_ assertion time from the falling edge of CS_.	N/A		20
Twrah	**Write cycles:** Address hold time from the rising edge of CS_/WR_, whichever comes first.	0		N/A
Twras	**Write cycles:** Address valid setup time to the falling edge of CS_/WR_, whichever comes first.	0		N/A

Table 4.30: Type A Host Interface Timing Parameters

Symbol	Description	Min (ns): Time and Conditions		Max (ns): Time and Conditions
Twrcyc	**Write cycle time requirement:** Time from the beginning of one write cycle to the beginning of the next write cycle.	10.5 *OR* Host Clock Period *OR* Memory Clock Period, ***whichever is largest***.		N/A
Twrh	**Write Enable Inactive Time:** Time from the end of one write cycle to the beginning of the next write cycle.	3.5		N/A
Twrl	**Write Enable Active time:** CS_-controlled write cycle: CS_ active time WR_-controlled write cycle: WR_ active time	7		N/A
Twrrdh	**BE_ Assertion Time:** Immediately after a read cycle: Time from the rising edge of CS_ to the falling edge of BE_.	5	No Hand-shake	N/A
		26	Handshake	
	RD_ Assertion Time: immediately following a write cycle: Time from the rising edge of CS_ to the falling edge of RD_.	5	No Hand-shake	N/A
		26	Handshake	
Twrrds	**Time for RD_ to be De-asserted:** Before a write cycle: Time from rising edge of RD_ to falling edge of CS_ for write cycles.	5	No Hand-shake	N/A
		26	Handshake	
	Time for BE_ to be De-asserted: Before a read cycle: Time from the rising edge of BE_ to the falling edge of CS_ for read cycles.	5	No Hand-shake	N/A
		26	Handshake	
Twrwt	Time from WR_ rising edge until WAIT_ falling edge	N/A		20
Twttri	Time from rising edge of CS_ to beginning of tri-state condition of WAIT_	N/A		29

4.4.4.3 Type C Host Interface

This section shows the timing characteristics for a Type C Host Bus interface using both direct addressing and indirect addressing.

- Indirect Addressing Read/Write Timing Diagrams
  - 16Bit
  - 32Bit
- Direct Addressing Read/Write Timing Diagrams
  - 16Bit
  - 32Bit

All Type C Timing Diagrams refer to the timing parameters in Table 4.48.

Note: The WAITn signal in each of the following timing diagrams is a tri-stated signal. Consult the specifications for the Host CPU in your design. Use either a pull-up or a pull-down resistor with this signal, according to the Host CPU specifications.

Note: The GoForce 5500 ball marked WRn does not correspond to a Type C Host Interface signal. The ball ***must*** be tied **low** externally when using the GoForce 5500 in a Type C Host Interface-based design.

Table 4.31: Type C Byte-enable Signals for Different Size Host Busses

Type C Host Interface Signal Name	Function	
	32bit Host Bus	16bit Host Bus
WE_0	Write Enable 1 [7:0]	Write Enable 1 [7:0]
WE_1	Write Enable 2 [15:8]	Write Enable 2 [15:8]
WE_2	Write Enable 3 [23:16]	Not used: Tie low or high externally
WE_3	Write Enable 4 [31:24]	A[1]

4.4.4.3.1 Type C Host Interface Timings: 16Bit Indirect

Figure 4.38: Register Write: 16Bit Indirect Type C

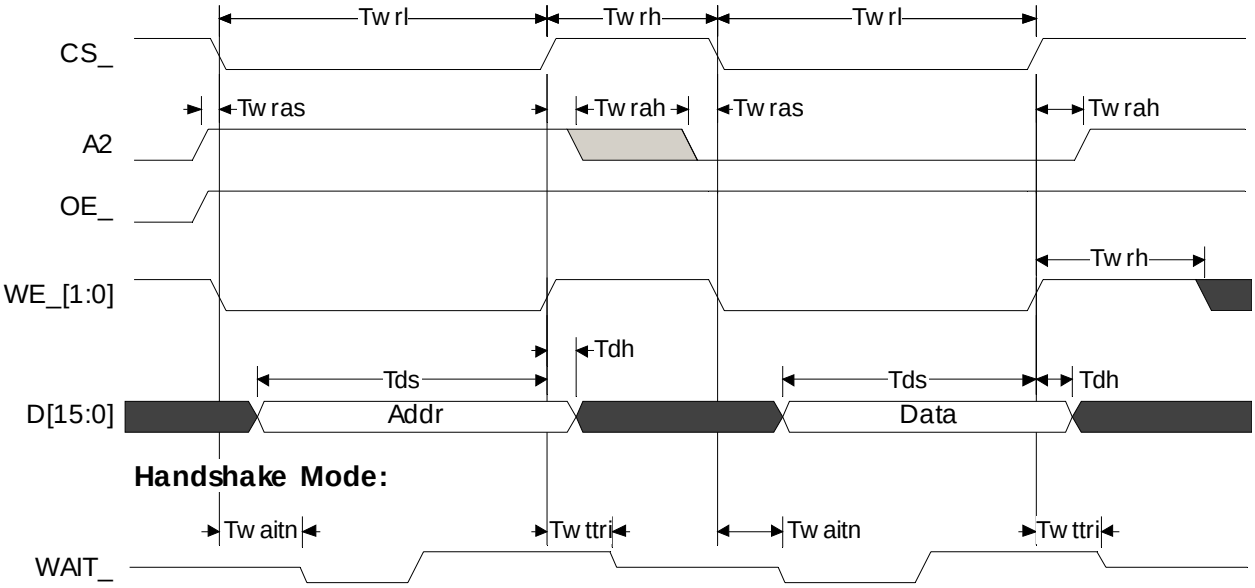


Table 4.32: Register Write D[15:0] Bit Mapping for Addr

Data Bus	Addr (Register 1)	Addr (Register 2)
D[15]	A[14]	X
D[14]	A[13]	X
D[13]	A[12]	X
D[12]	A[11]	X
D[11]	A[10]	X
D[10]	A[9]	X
D[9]	A[8]	X
D[8]	A[7]	X
D[7]	A[6]	X
D[6]	A[5]	X
D[5]	A[4]	X
D[4]	A[3]	A[17]
D[3]	A[2]	A[16]
D[2]	A[1]	A[15]
D[1]	0	1
D[0]	1	1

Figure 4.39: Register Read: 16Bit Indirect Type C

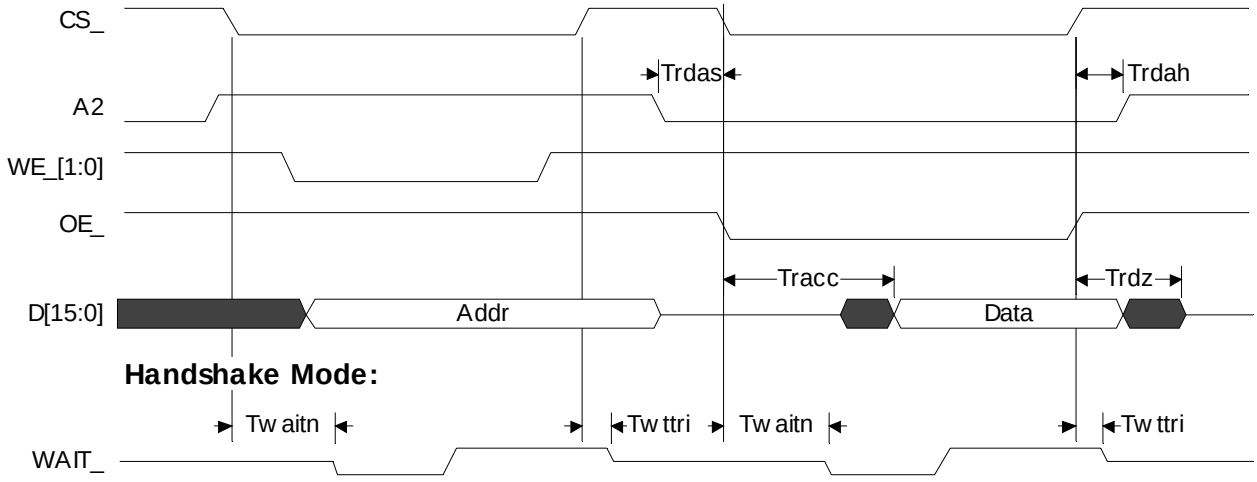


Table 4.33: Register Read D[15:0] Bit Mapping for Addr

Data Bus	Addr (Register 1)	Addr (Register 2)
D[15]	A[14]	X
D[14]	A[13]	X
D[13]	A[12]	X
D[12]	A[11]	X
D[11]	A[10]	X
D[10]	A[9]	X
D[9]	A[8]	X
D[8]	A[7]	X
D[7]	A[6]	X
D[6]	A[5]	X
D[5]	A[4]	X
D[4]	A[3]	A[17]
D[3]	A[2]	A[16]
D[2]	A[1]	A[15]
D[1]	0	1
D[0]	1	1

Figure 4.40: Memory Write: 16Bit Indirect Type C

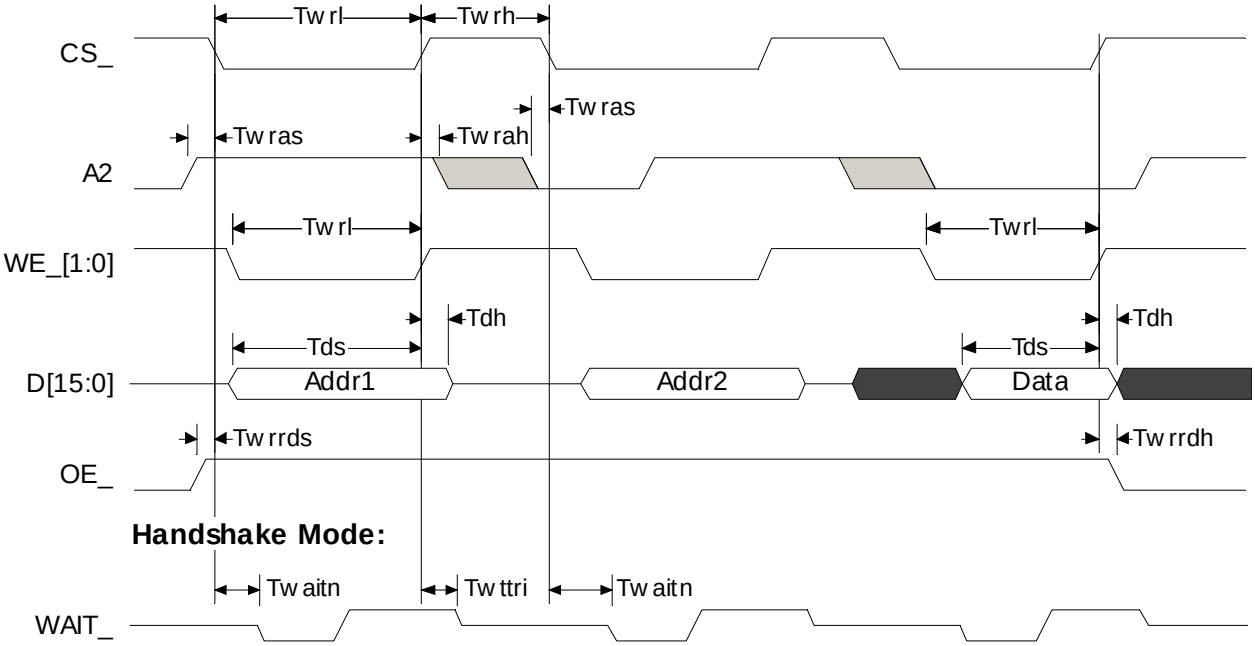


Table 4.34: Memory Write D[15:0] Bit Mapping for Addr1 and Addr2

Data Bus	Address Phase	
	Addr1	Addr2
D[15]	A[14]	X
D[14]	A[13]	X
D[13]	A[12]	X
D[12]	A[11]	A[25]
D[11]	A[10]	A[24]
D[10]	A[9]	A[23]
D[9]	A[8]	A[22]
D[8]	A[7]	A[21]
D[7]	A[6]	A[20]
D[6]	A[5]	A[19]
D[5]	A[4]	A[18]
D[4]	A[3]	A[17]
D[3]	A[2]	A[16]
D[2]	A[1]	A[15]
D[1]	0	1
D[0]	0	0

Figure 4.41: Memory Read: 16Bit Indirect Type C

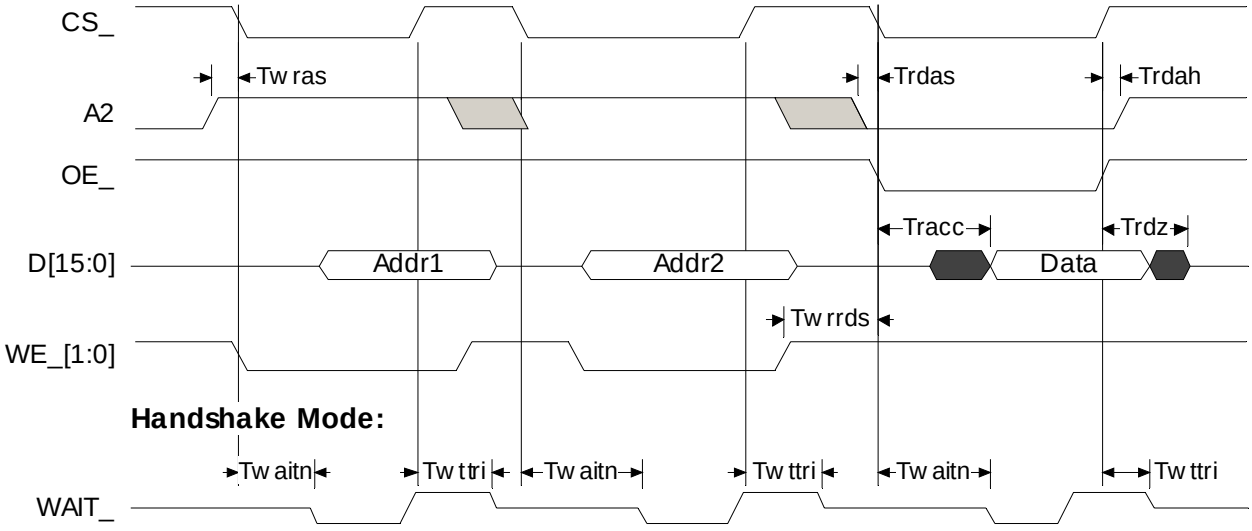


Table 4.35: Memory Read D[15:0] Bit Mapping for Addr1 and Addr2

Data Bus	Address Phase	
	Addr1	Addr2
D[15]	A[14]	X
D[14]	A[13]	X
D[13]	A[12]	X
D[12]	A[11]	A[25]
D[11]	A[10]	A[24]
D[10]	A[9]	A[23]
D[9]	A[8]	A[22]
D[8]	A[7]	A[21]
D[7]	A[6]	A[20]
D[6]	A[5]	A[19]
D[5]	A[4]	A[18]
D[4]	A[3]	A[17]
D[3]	A[2]	A[16]
D[2]	A[1]	A[15]
D[1]	0	1
D[0]	0	0

Figure 4.42: Register Auto-increment Read: 16Bit Indirect Type C

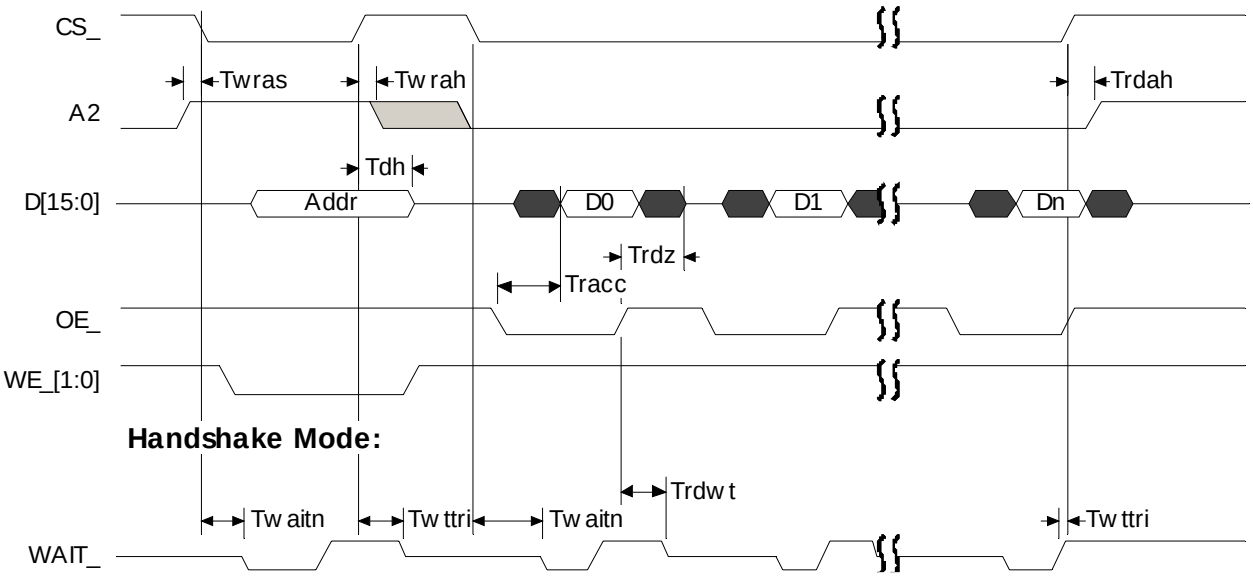


Table 4.36: Register Read D[15:0] Bit Mapping for Addr

Data Bus	Addr (Register 1)	Addr (Register 2)
D[15]	A[14]	X
D[14]	A[13]	X
D[13]	A[12]	X
D[12]	A[11]	X
D[11]	A[10]	X
D[10]	A[9]	X
D[9]	A[8]	X
D[8]	A[7]	X
D[7]	A[6]	X
D[6]	A[5]	X
D[5]	A[4]	X
D[4]	A[3]	A[17]
D[3]	A[2]	A[16]
D[2]	A[1]	A[15]
D[1]	0	1
D[0]	1	1

Figure 4.43: Memory Auto-increment Read: 16Bit Indirect Type C

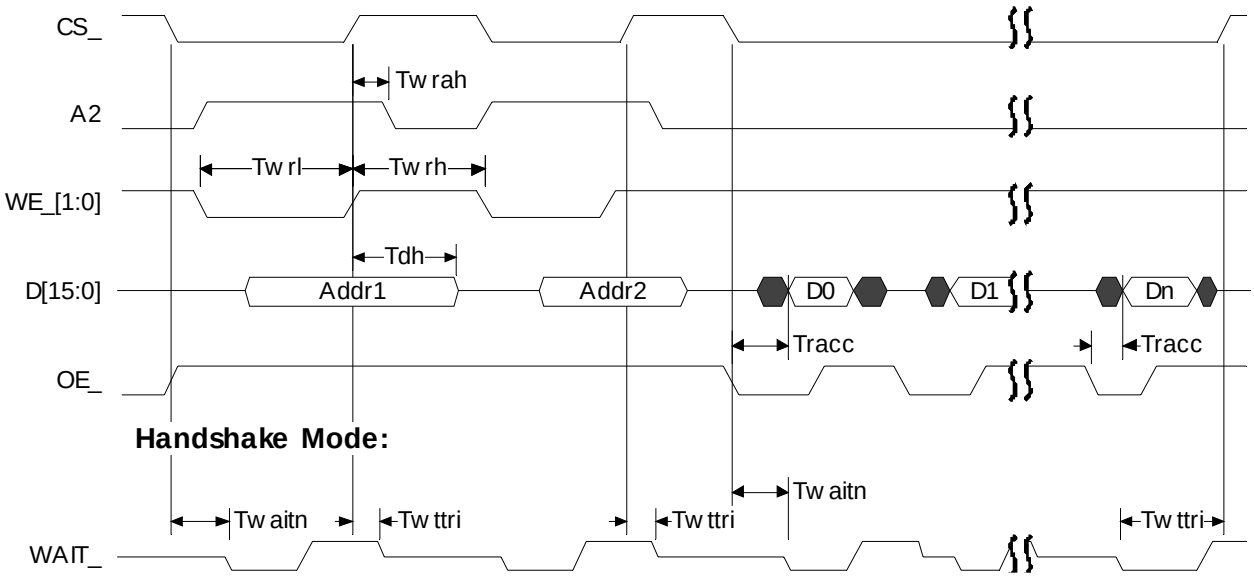


Table 4.37: Memory Read D[15:0] Bit Mapping for Addr1 and Addr2

Data Bus	Address Phase	
	Addr1	Addr2
D[15]	A[14]	X
D[14]	A[13]	X
D[13]	A[12]	X
D[12]	A[11]	A[25]
D[11]	A[10]	A[24]
D[10]	A[9]	A[23]
D[9]	A[8]	A[22]
D[8]	A[7]	A[21]
D[7]	A[6]	A[20]
D[6]	A[5]	A[19]
D[5]	A[4]	A[18]
D[4]	A[3]	A[17]
D[3]	A[2]	A[16]
D[2]	A[1]	A[15]
D[1]	0	1
D[0]	0	0

Figure 4.44: Register Auto-increment Write: 16Bit Indirect Type C

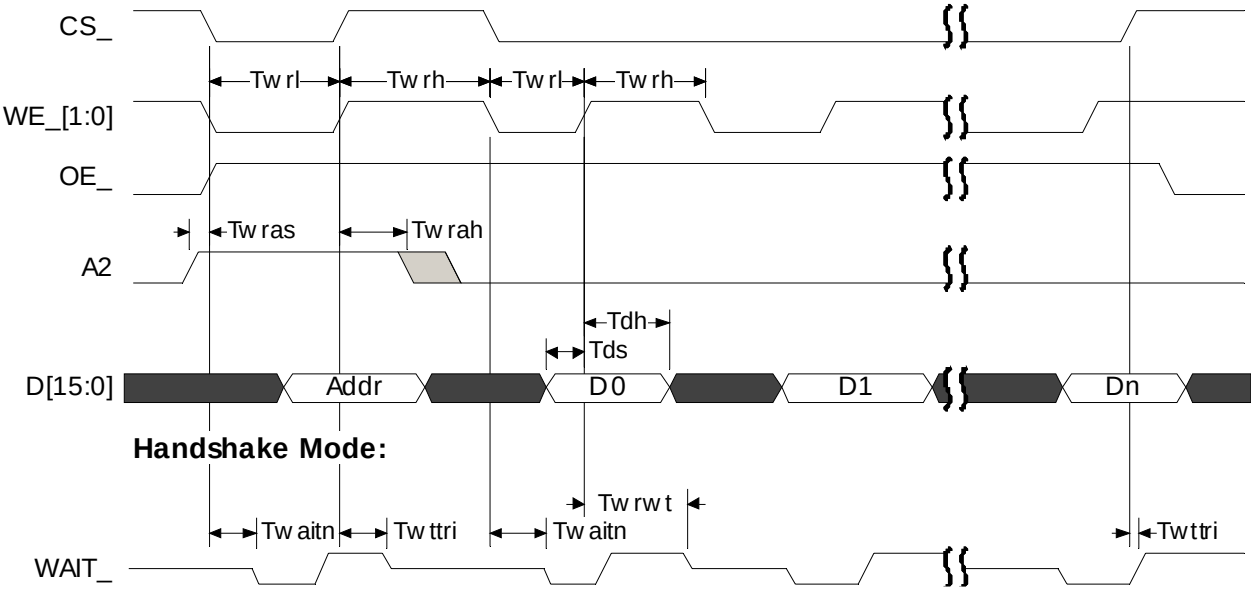


Table 4.38: Register Write D[15:0] Bit Mapping for Addr

Data Bus	Addr (Register 1)	Addr (Register 2)
D[15]	A[14]	X
D[14]	A[13]	X
D[13]	A[12]	X
D[12]	A[11]	X
D[11]	A[10]	X
D[10]	A[9]	X
D[9]	A[8]	X
D[8]	A[7]	X
D[7]	A[6]	X
D[6]	A[5]	X
D[5]	A[4]	X
D[4]	A[3]	A[17]
D[3]	A[2]	A[16]
D[2]	A[1]	A[15]
D[1]	0	1
D[0]	1	1

Figure 4.45: Memory Auto-increment Write: 16Bit Indirect Type C

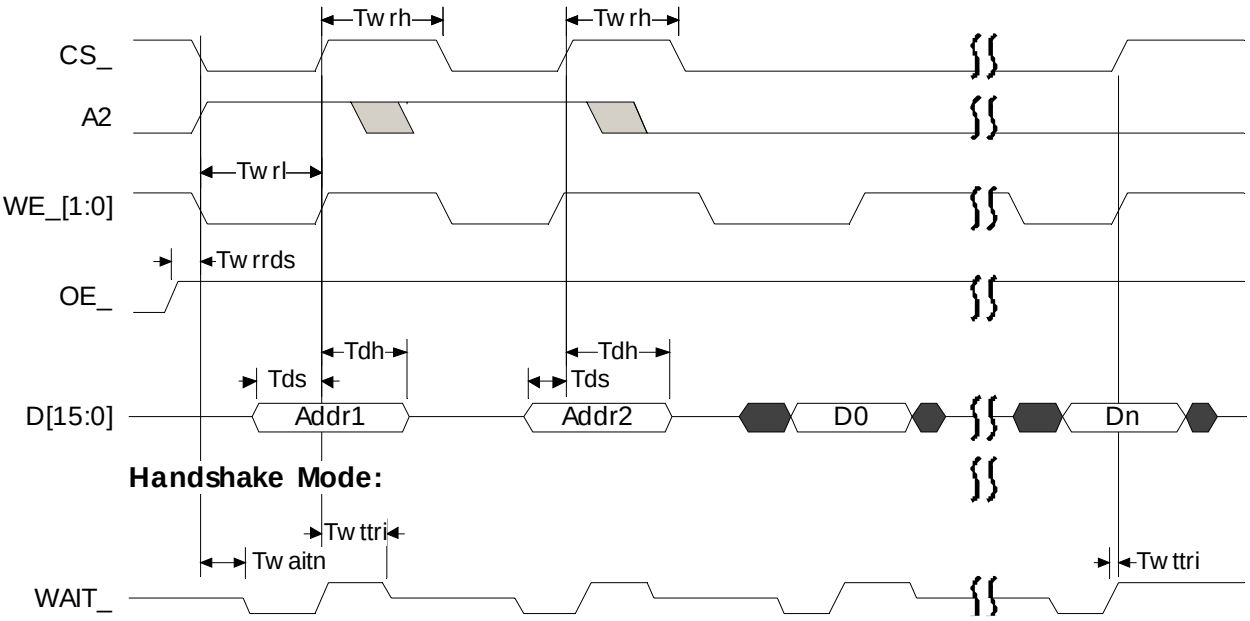


Table 4.39: Memory Write D[15:0] Bit Mapping for Addr1 and Addr2

Data Bus	Address Phase	
	Addr1	Addr2
D[15]	A[14]	X
D[14]	A[13]	X
D[13]	A[12]	X
D[12]	A[11]	A[25]
D[11]	A[10]	A[24]
D[10]	A[9]	A[23]
D[9]	A[8]	A[22]
D[8]	A[7]	A[21]
D[7]	A[6]	A[20]
D[6]	A[5]	A[19]
D[5]	A[4]	A[18]
D[4]	A[3]	A[17]
D[3]	A[2]	A[16]
D[2]	A[1]	A[15]
D[1]	0	1
D[0]	0	0

4.4.4.3.2 Type C Host Interface Timings: 32Bit Indirect

Figure 4.46: Register Write, 32Bit Indirect Type C

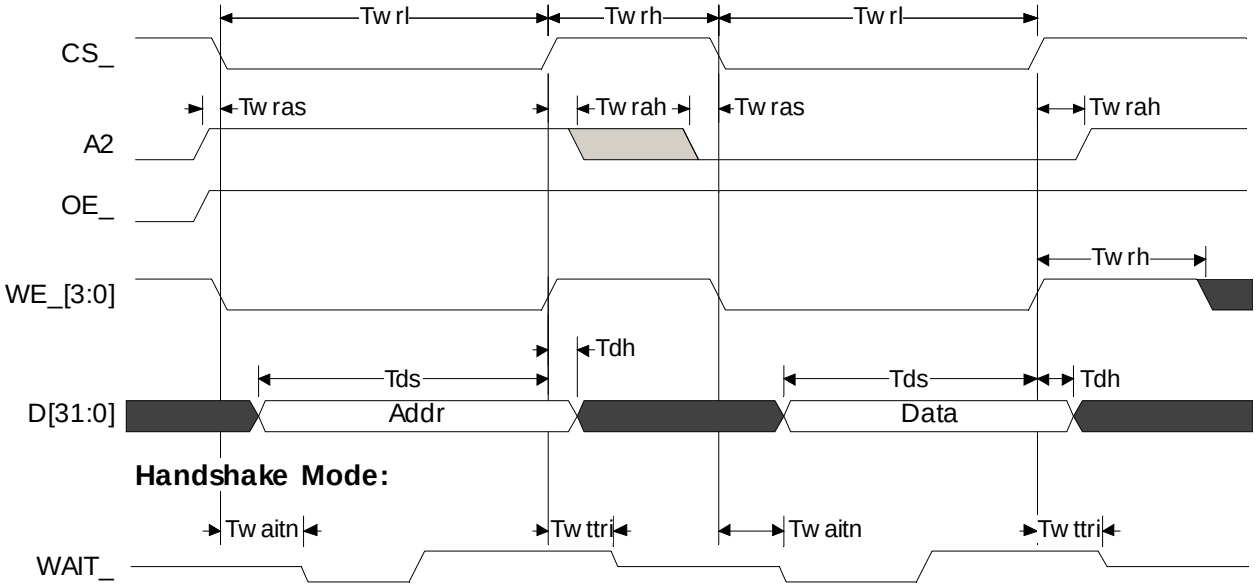


Table 4.40: Register Write D[31:0] Bit Mapping for Addr

Data Bus[31:16]	Addr	Data Bus[15:0]	Addr
D[31]	X	D[15]	A[16]
D[30]	X	D[14]	A[15]
D[29]	X	D[13]	A[14]
D[28]	X	D[12]	A[13]
D[27]	X	D[11]	A[12]
D[26]	X	D[10]	A[11]
D[25]	X	D[9]	A[10]
D[24]	X	D[8]	A[9]
D[23]	X	D[7]	A[8]
D[22]	X	D[6]	A[7]
D[21]	X	D[5]	A[6]
D[20]	X	D[4]	A[5]
D[19]	X	D[3]	A[4]
D[18]	X	D[2]	A[3]
D[17]	X	D[1]	A[2]
D[16]	A[17]	D[0]	1

Figure 4.47: Register Read, 32Bit Indirect Type C

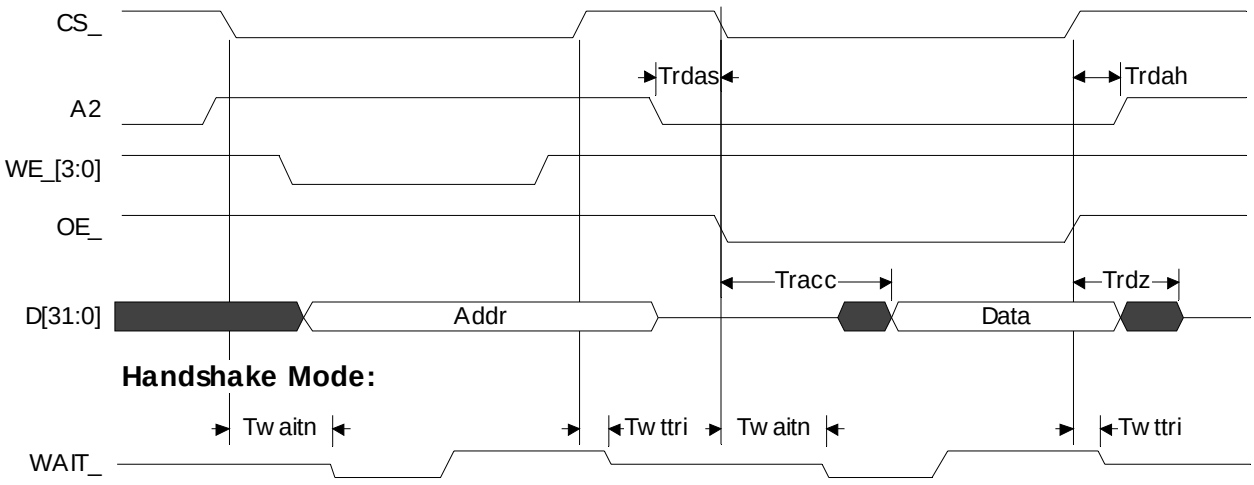


Table 4.41: Register Read D[31:0] Bit Mapping for Addr

Data Bus[31:16]	Addr	Data Bus[15:0]	Addr
D[31]	X	D[15]	A[16]
D[30]	X	D[14]	A[15]
D[29]	X	D[13]	A[14]
D[28]	X	D[12]	A[13]
D[27]	X	D[11]	A[12]
D[26]	X	D[10]	A[11]
D[25]	X	D[9]	A[10]
D[24]	X	D[8]	A[9]
D[23]	X	D[7]	A[8]
D[22]	X	D[6]	A[7]
D[21]	X	D[5]	A[6]
D[20]	X	D[4]	A[5]
D[19]	X	D[3]	A[4]
D[18]	X	D[2]	A[3]
D[17]	X	D[1]	A[2]
D[16]	A[17]	D[0]	1

Figure 4.48: Memory Write, 32Bit Indirect Type C

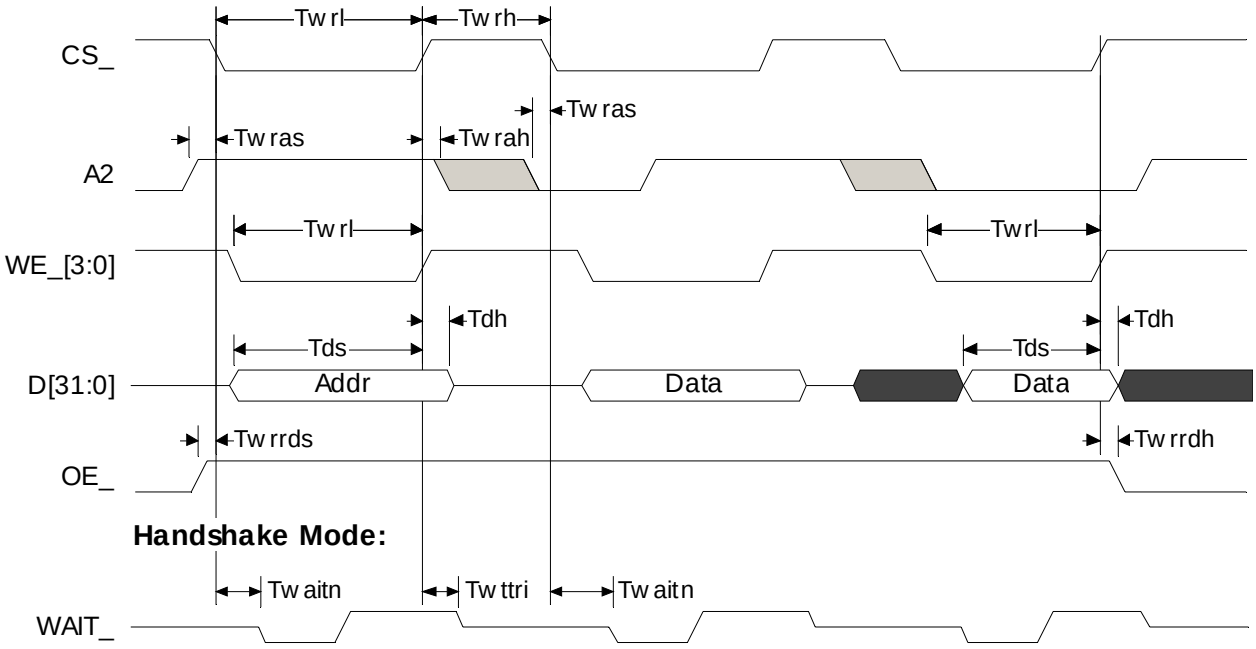


Table 4.42: Memory Write D[31:0] Bit Mapping for Addr

Data Bus[31:16]	Addr	Data Bus[15:0]	Addr
D[31]	X	D[15]	A[16]
D[30]	X	D[14]	A[15]
D[29]	X	D[13]	A[14]
D[28]	X	D[12]	A[13]
D[27]	X	D[11]	A[12]
D[26]	X	D[10]	A[11]
D[25]	X	D[9]	A[10]
D[24]	A[25]	D[8]	A[9]
D[23]	A[24]	D[7]	A[8]
D[22]	A[23]	D[6]	A[7]
D[21]	A[22]	D[5]	A[6]
D[20]	A[21]	D[4]	A[5]
D[19]	A[20]	D[3]	A[4]
D[18]	A[19]	D[2]	A[3]
D[17]	A[18]	D[1]	A[2]
D[16]	A[17]	D[0]	0

Figure 4.49: Memory Read: 32Bit Indirect Type C

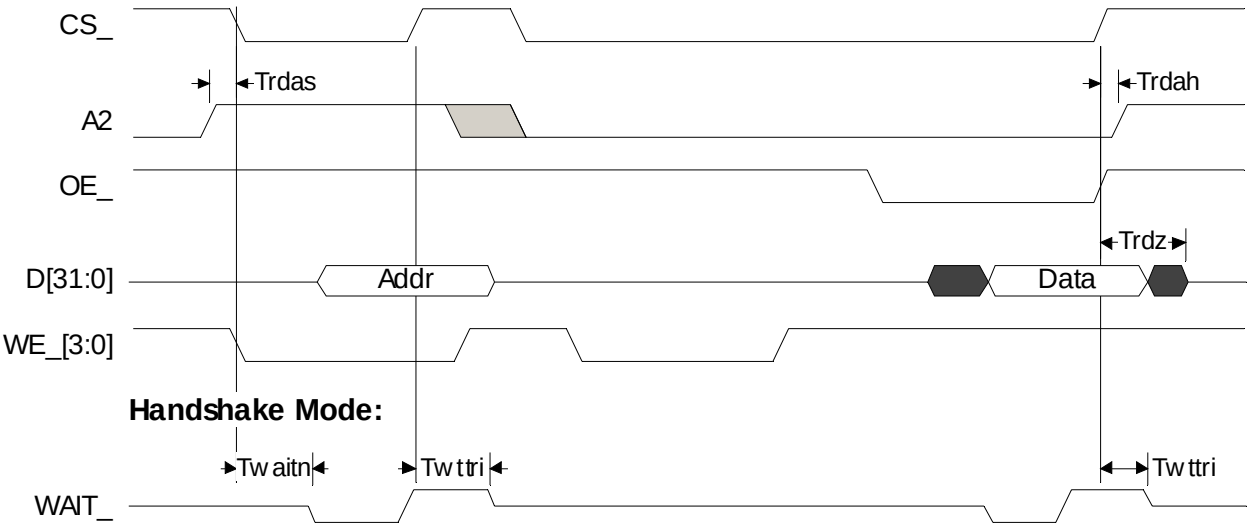


Table 4.43: Memory Read D[31:0] Bit Mapping for Addr and Addr

Data Bus[31:16]	Addr	Data Bus[15:0]	Addr
D[31]	X	D[15]	A[16]
D[30]	X	D[14]	A[15]
D[29]	X	D[13]	A[14]
D[28]	X	D[12]	A[13]
D[27]	X	D[11]	A[12]
D[26]	X	D[10]	A[11]
D[25]	X	D[9]	A[10]
D[24]	A[25]	D[8]	A[9]
D[23]	A[24]	D[7]	A[8]
D[22]	A[23]	D[6]	A[7]
D[21]	A[22]	D[5]	A[6]
D[20]	A[21]	D[4]	A[5]
D[19]	A[20]	D[3]	A[4]
D[18]	A[19]	D[2]	A[3]
D[17]	A[18]	D[1]	A[2]
D[16]	A[17]	D[0]	0

Figure 4.50: Register Auto-increment Read: 32Bit Indirect Type C

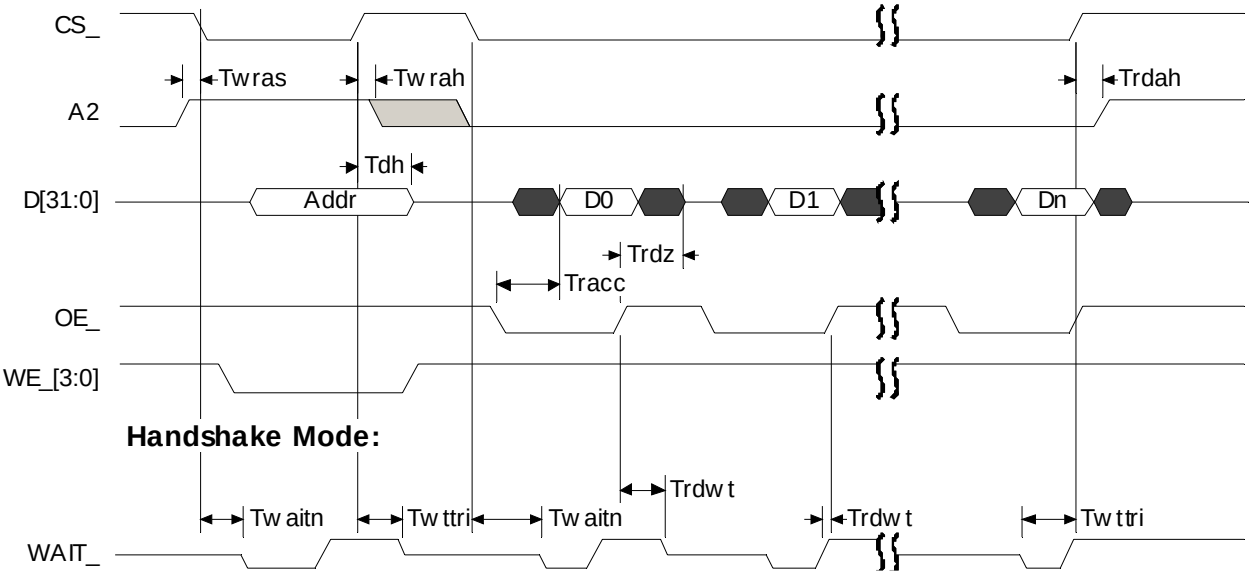


Table 4.44: Register Read D[31:0] Bit Mapping for Addr

Data Bus[31:16]	Addr	Data Bus[15:0]	Addr
D[31]	X	D[15]	A[16]
D[30]	X	D[14]	A[15]
D[29]	X	D[13]	A[14]
D[28]	X	D[12]	A[13]
D[27]	X	D[11]	A[12]
D[26]	X	D[10]	A[11]
D[25]	X	D[9]	A[10]
D[24]	X	D[8]	A[9]
D[23]	X	D[7]	A[8]
D[22]	X	D[6]	A[7]
D[21]	X	D[5]	A[6]
D[20]	X	D[4]	A[5]
D[19]	X	D[3]	A[4]
D[18]	X	D[2]	A[3]
D[17]	X	D[1]	A[2]
D[16]	A[17]	D[0]	1

Figure 4.51: Memory Auto-increment Read: 32Bit Indirect Type C

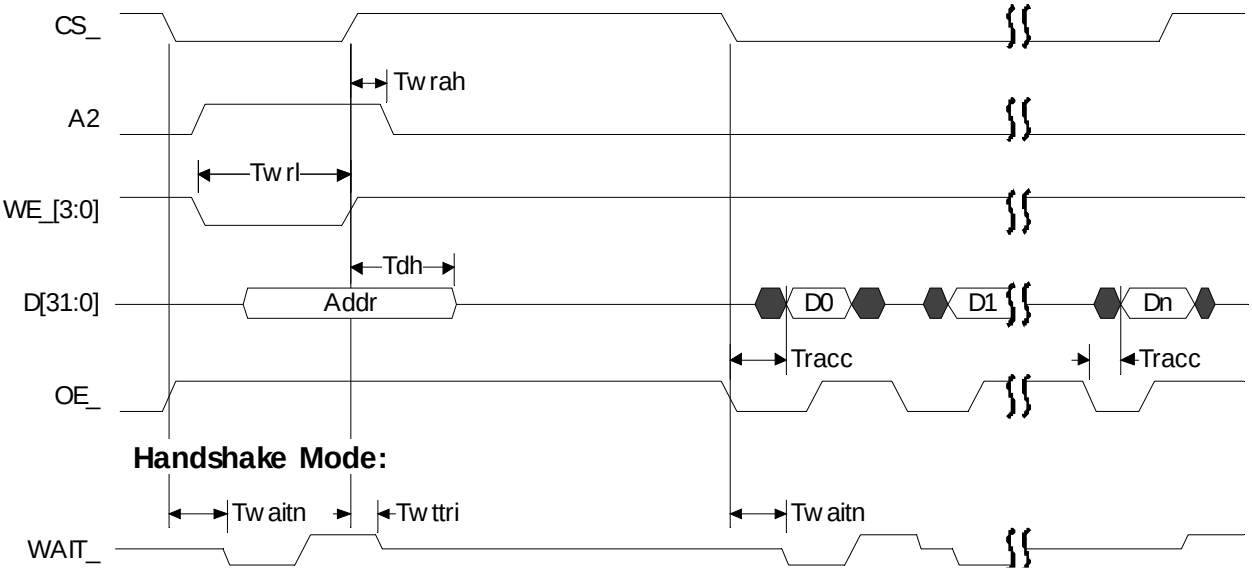


Table 4.45: Memory Read D[31:0] Bit Mapping for Addr

Data Bus[31:16]	Addr	Data Bus[15:0]	Addr
D[31]	X	D[15]	A[16]
D[30]	X	D[14]	A[15]
D[29]	X	D[13]	A[14]
D[28]	X	D[12]	A[13]
D[27]	X	D[11]	A[12]
D[26]	X	D[10]	A[11]
D[25]	X	D[9]	A[10]
D[24]	A[25]	D[8]	A[9]
D[23]	A[24]	D[7]	A[8]
D[22]	A[23]	D[6]	A[7]
D[21]	A[22]	D[5]	A[6]
D[20]	A[21]	D[4]	A[5]
D[19]	A[20]	D[3]	A[4]
D[18]	A[19]	D[2]	A[3]
D[17]	A[18]	D[1]	A[2]
D[16]	A[17]	D[0]	0

Figure 4.52: Register Auto-increment Write: 32Bit Indirect Type C

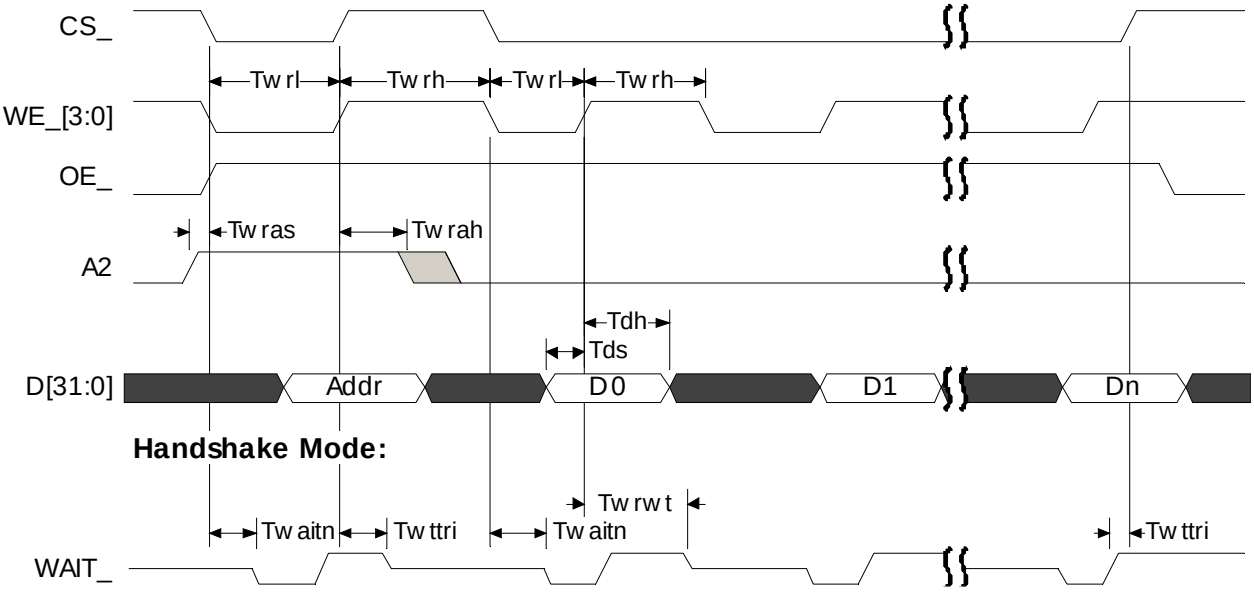


Table 4.46: Register Write D[31:0] Bit Mapping for Addr*

Data Bus[31:16]	Addr	Data Bus[15:0]	Addr
D[31]	X	D[15]	A[16]
D[30]	X	D[14]	A[15]
D[29]	X	D[13]	A[14]
D[28]	X	D[12]	A[13]
D[27]	X	D[11]	A[12]
D[26]	X	D[10]	A[11]
D[25]	X	D[9]	A[10]
D[24]	X	D[8]	A[9]
D[23]	X	D[7]	A[8]
D[22]	X	D[6]	A[7]
D[21]	X	D[5]	A[6]
D[20]	X	D[4]	A[5]
D[19]	X	D[3]	A[4]
D[18]	X	D[2]	A[3]
D[17]	X	D[1]	A[2]
D[16]	A[17]	D[0]	1

Figure 4.53: Memory Auto-increment Write: 32Bit Indirect Type C

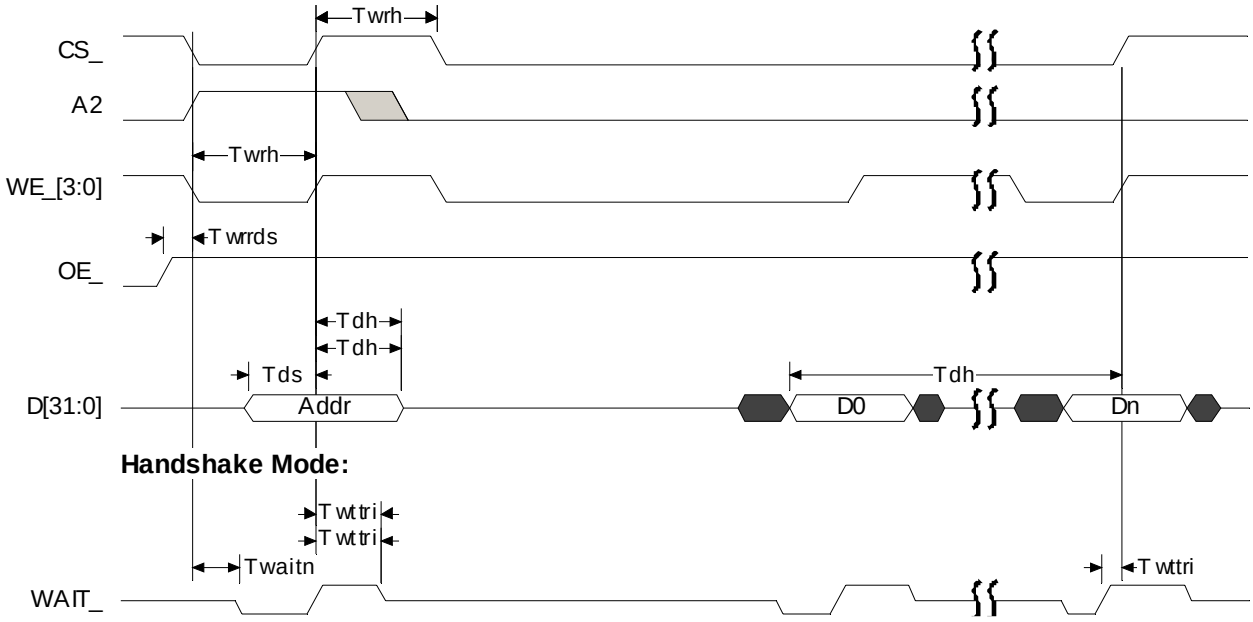
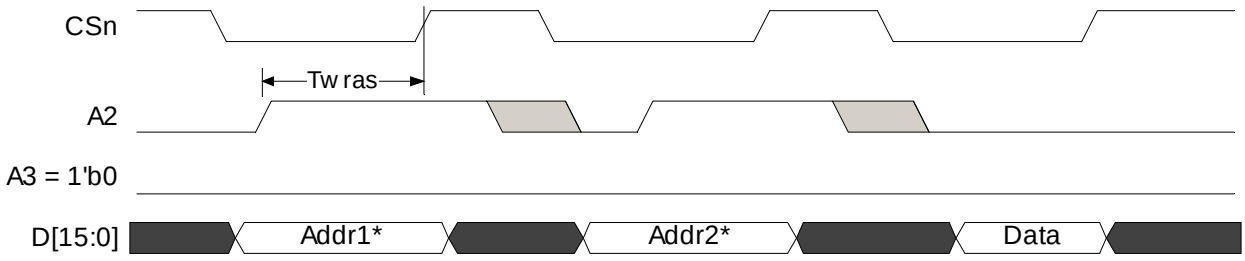


Table 4.47: Memory Write D[31:0] Bit Mapping for Addr1* and Addr2*

Data Bus[31:16]	Addr	Data Bus[15:0]	Addr
D[31]	X	D[15]	A[16]
D[30]	X	D[14]	A[15]
D[29]	X	D[13]	A[14]
D[28]	X	D[12]	A[13]
D[27]	X	D[11]	A[12]
D[26]	X	D[10]	A[11]
D[25]	X	D[9]	A[10]
D[24]	A[25]	D[8]	A[9]
D[23]	A[24]	D[7]	A[8]
D[22]	A[23]	D[6]	A[7]
D[21]	A[22]	D[5]	A[6]
D[20]	A[21]	D[4]	A[5]
D[19]	A[20]	D[3]	A[4]
D[18]	A[19]	D[2]	A[3]
D[17]	A[18]	D[1]	A[2]
D[16]	A[17]	D[0]	0

4.4.4.3.3 One and Two-channel Access for Indirect Addressing

Figure 4.54: One-channel Access, Indirect Addressing



Note: Recommendation when not using A3 as a secondary latch (i.e. single-channel access): Tie A3 to ground.

Figure 4.55: Two-channel Memory Access, Indirect Addressing

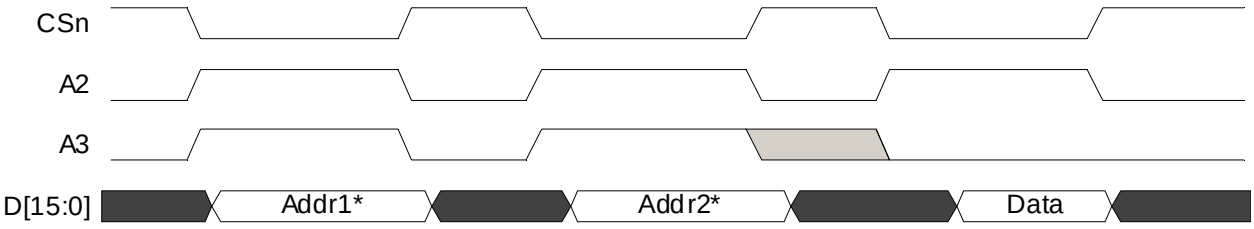
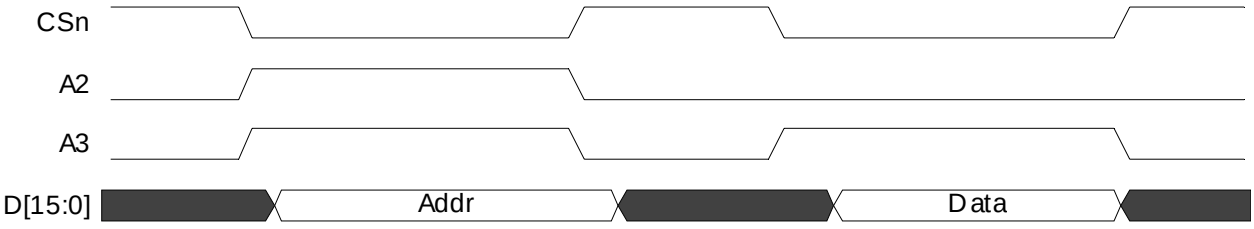


Figure 4.56: Two-channel Register Access, Indirect Addressing



Note in Figure 4.67 and Figure 4.68 that A1 and A2 must toggle with CSn. Also, A2 must be high during the data transmission phase, unlike A1.

4.4.4.3.4 Type C Host Interface Timings: 16Bit Direct

Figure 4.57: WE_-controlled Write: 16Bit Direct Type C

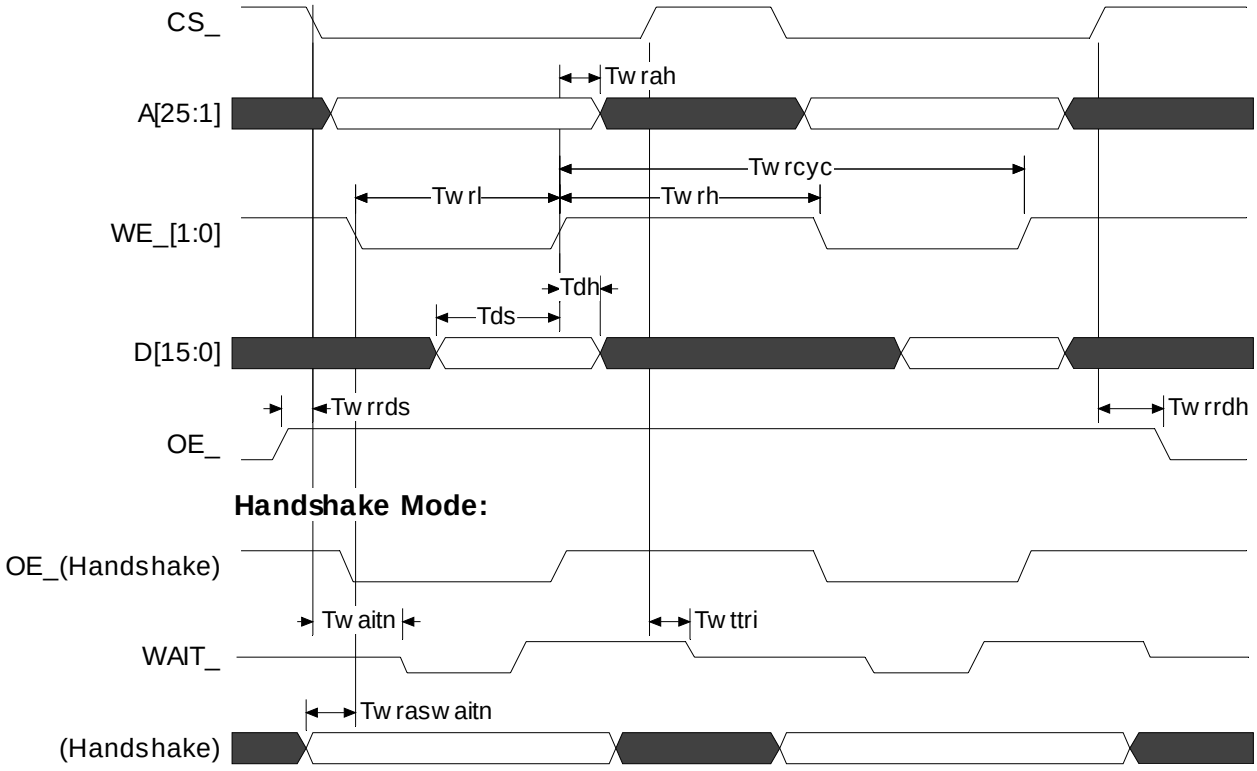


Figure 4.58: CS_-controlled Write: 16Bit Direct Type C

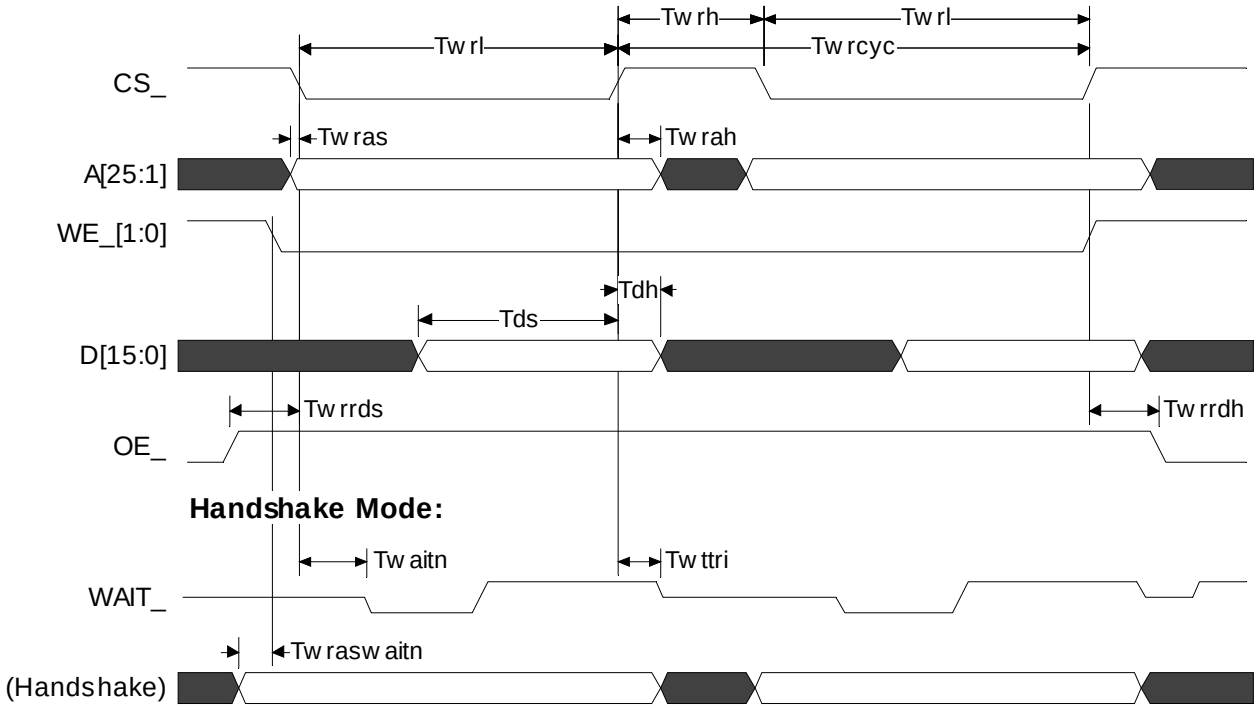


Figure 4.59: OE_-controlled Read: 16Bit Direct Type C

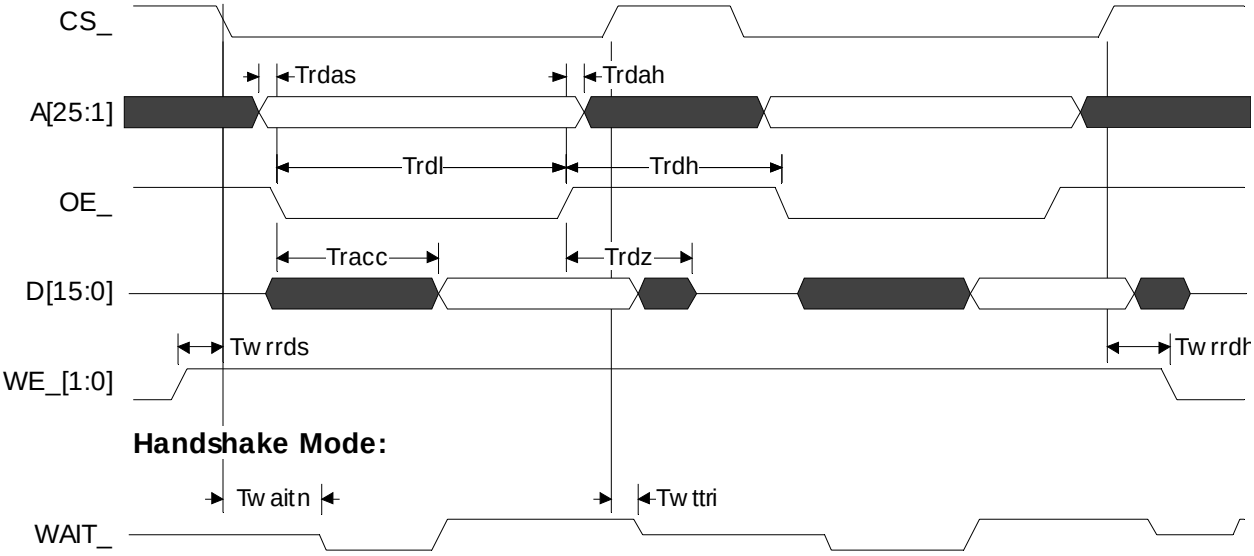


Figure 4.60: CS_-controlled Read: 16Bit Direct Type C

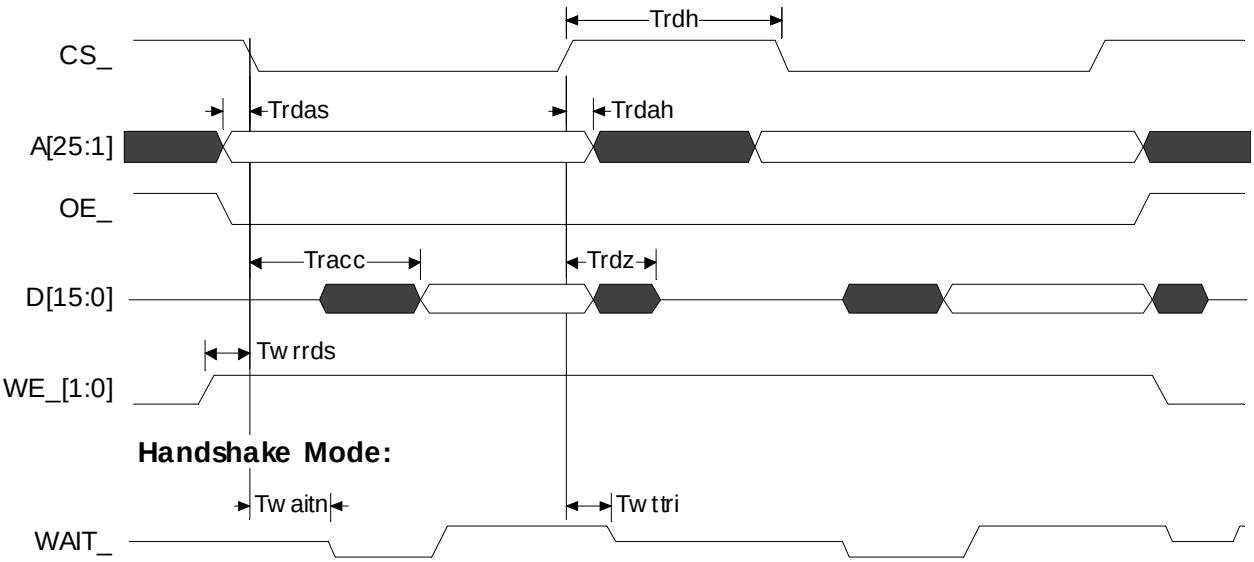


Figure 4.61: Register or Memory Burst Write: 16Bit Direct Type C

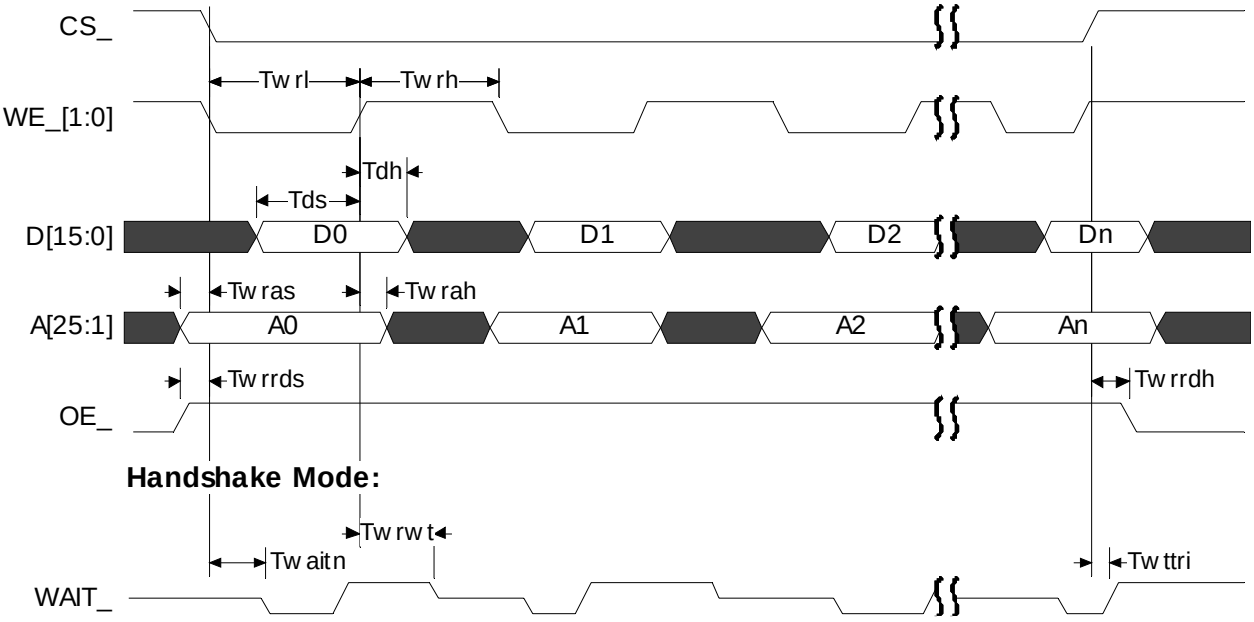
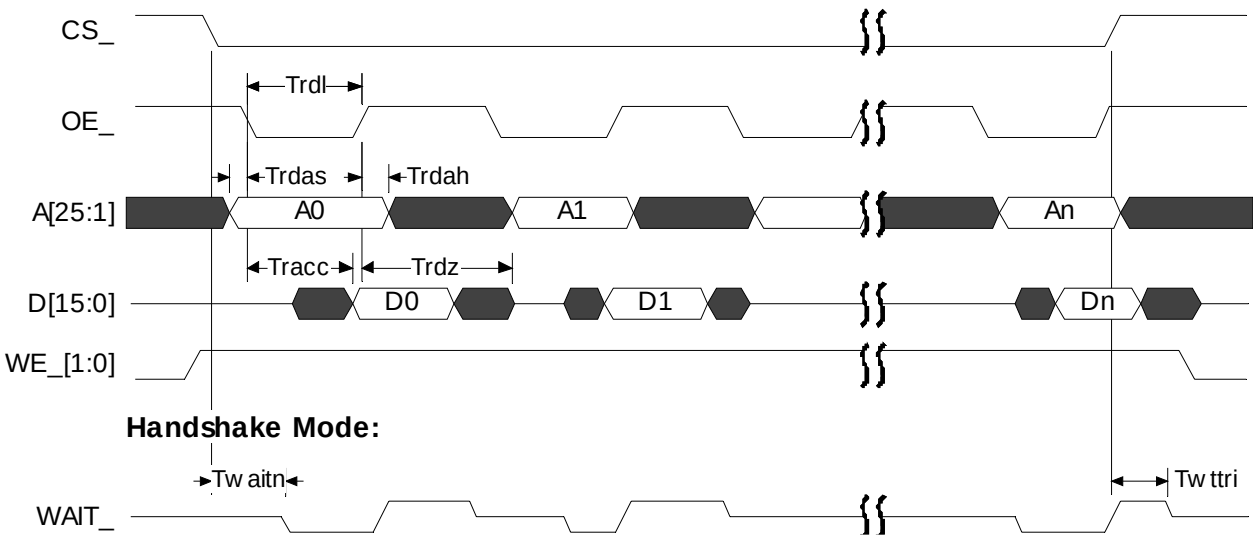


Figure 4.62: Register or Memory Burst Read: 16Bit Type C



4.4.4.3.5 Type C Host Interface Timings: 32Bit Direct

Figure 4.63: WE_-controlled Write: 32Bit Direct Type C

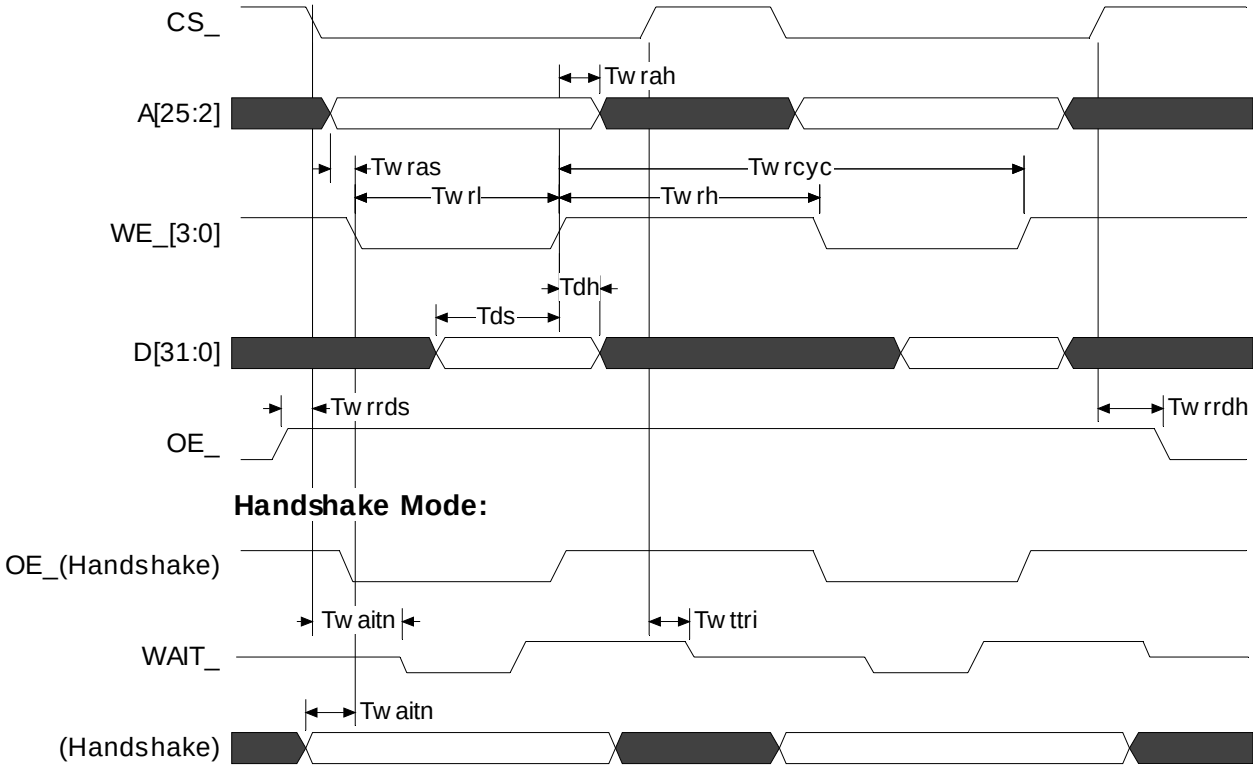


Figure 4.64: CS_-controlled Write: 32Bit Direct Type C

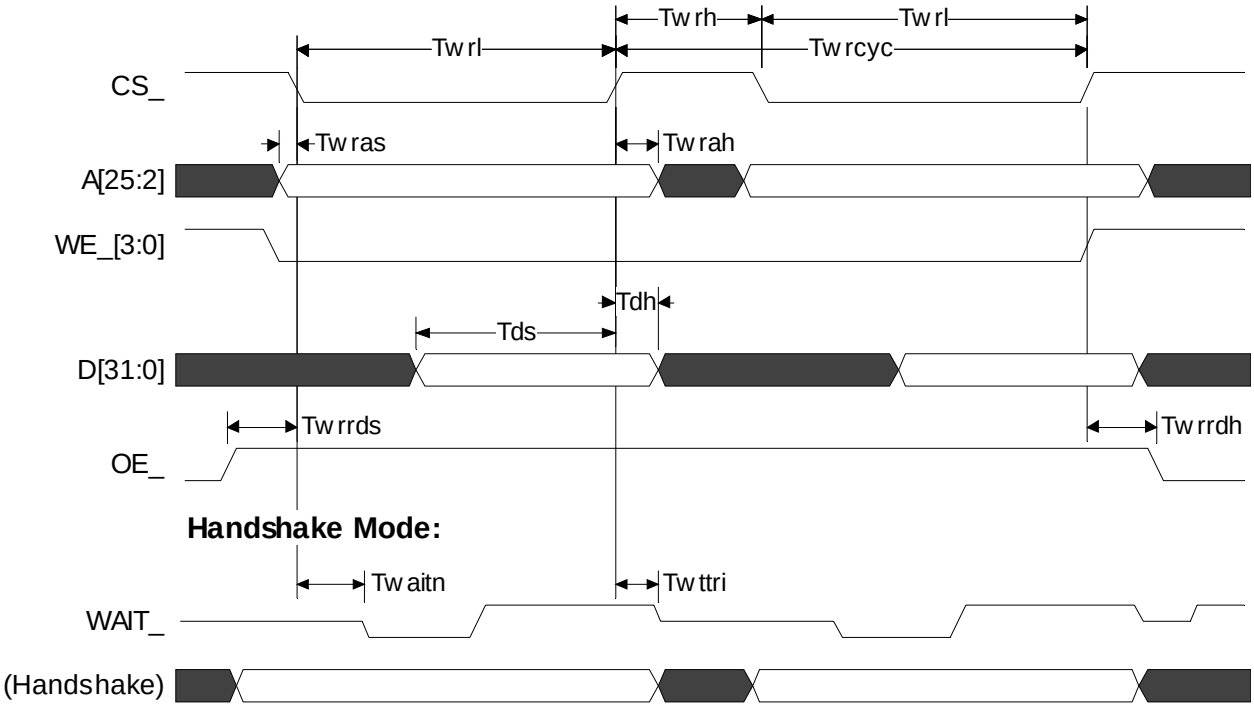


Figure 4.65: OE_-controlled Read: 32Bit Direct Type C

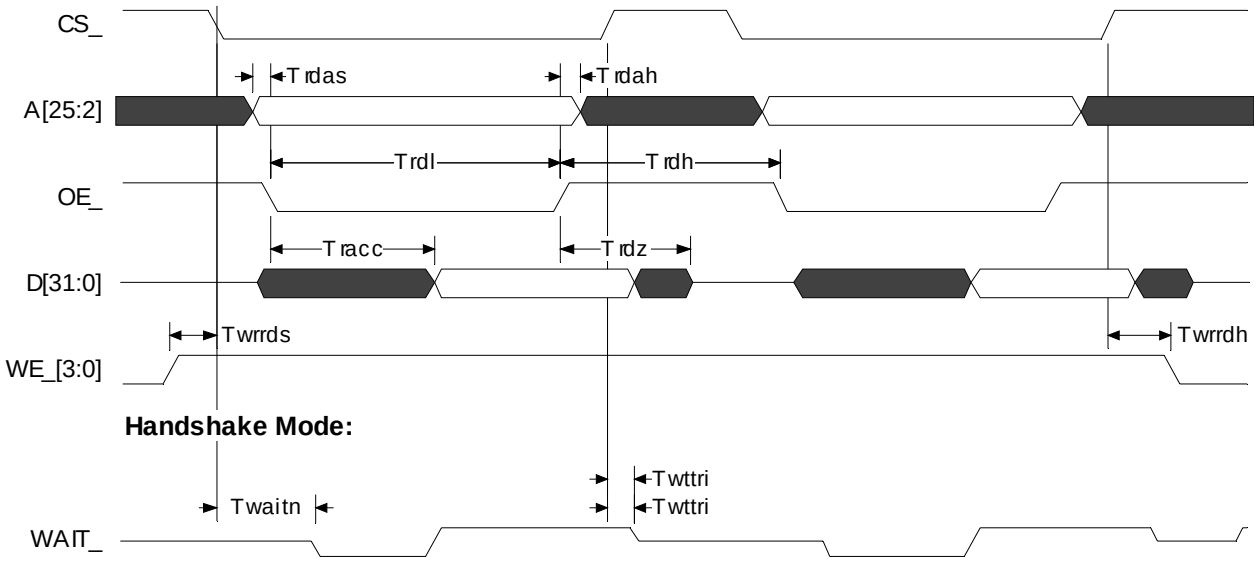


Figure 4.66: CS_-controlled Read: 32Bit Direct Type C

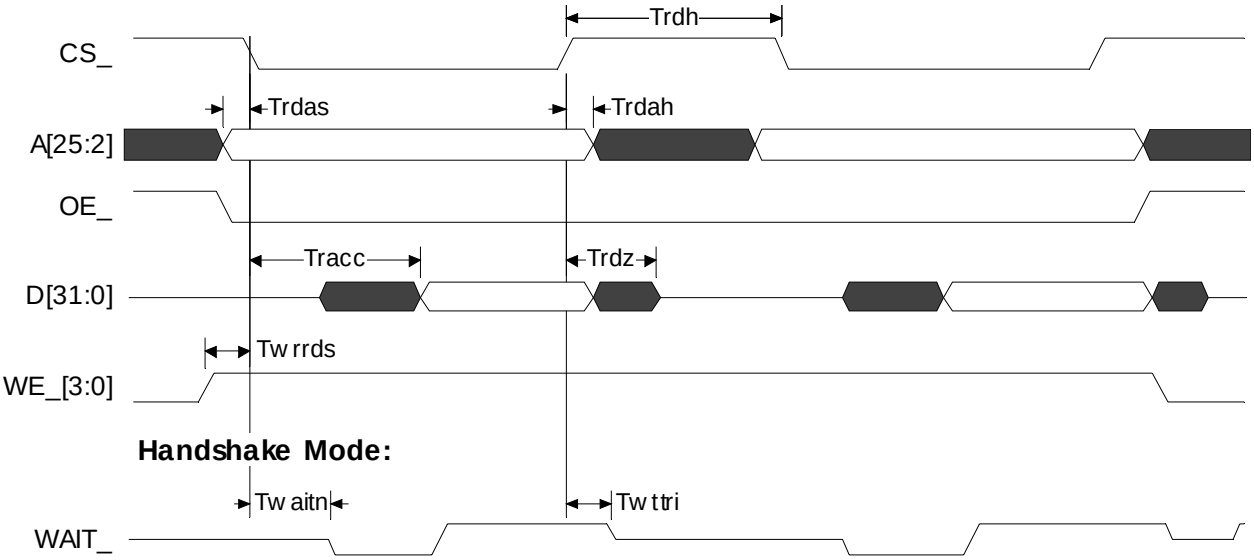


Figure 4.67: Register or Memory Burst Write: 32Bit Direct Type C

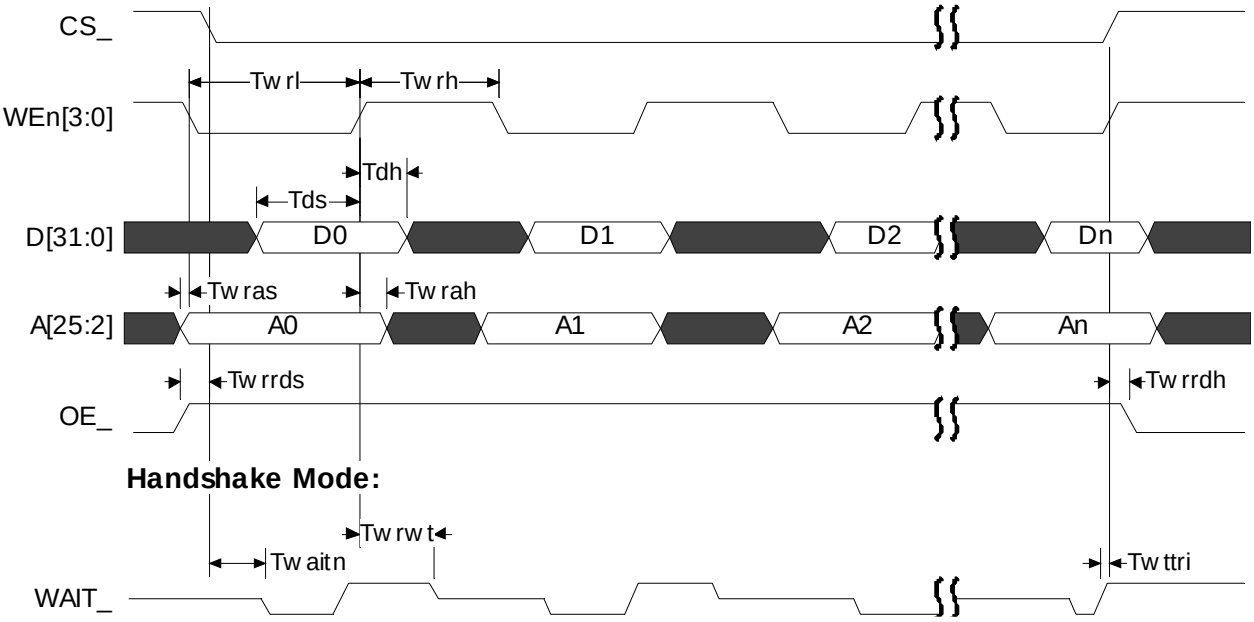
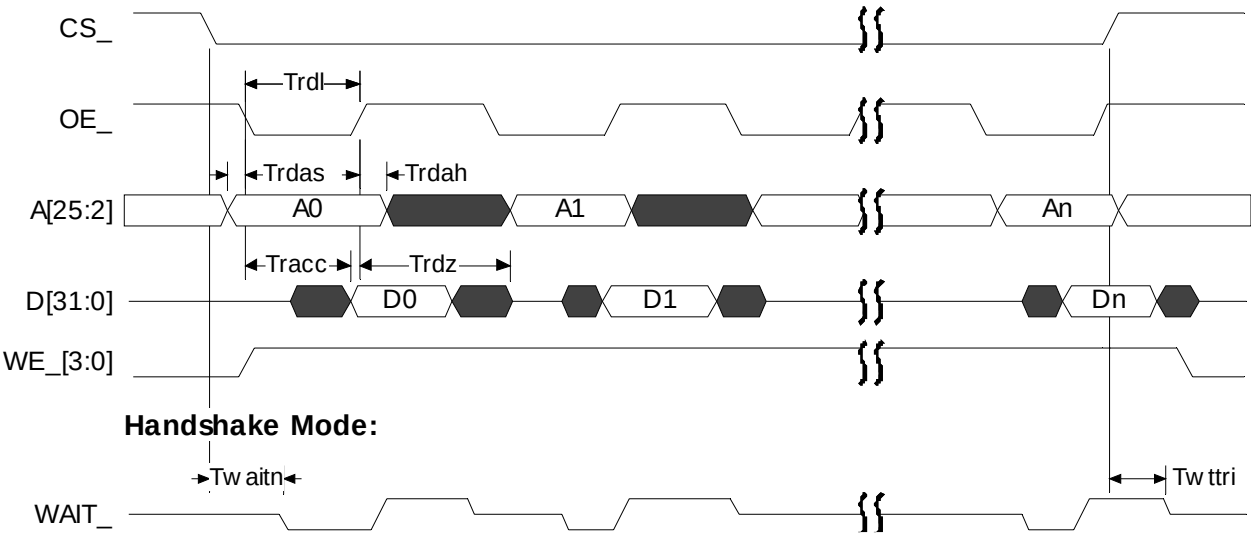


Figure 4.68: Register or Memory Burst Read: 32Bit Direct Type C



4.4.4.4 Type C Host Interface Timing Parameters

Table 4.48 provides the AC timing parameters for the preceding Type C host interface timing diagrams.

Note: **TM** is the memory clock period in ns.
TF is the FIFO clock period in ns, defined according to the following:

- *VI FIFO Status Register:* The frequency of TF is the VI block frequency
- *All other FIFO Status Registers:* The frequency of TF is equal to that of TM.

Table 4.48: Type C Host Interface Timing Parameters

Symbol	Description	Min (ns): Time and Conditions		Max (ns): Time and Conditions
Tdh	**Write cycles:** Data hold time from rising edge of CS_ or WE_, whichever comes first	0		N/A
Tds	**Write cycles:** Data setup time to rising edge of either CS_ or WE_, whichever comes first.	7		N/A
Tracc	**Read cycles:** Maximum read access time from the beginning of the read cycle to the first valid data access.	N/A		Asynchronous Register Access: 26
				Synchronous Register Access: (5*Host Clock) + 21
				SRAM Access: (7*Memory clock) + 21
Trdah	**Read cycles:** Address hold time from rising edge of CS_ or RD_, whichever comes first.	0		N/A
Trdas	**Read cycles:** Address setup time to falling edge of CS_ or OE_, whichever comes last.	0		N/A
Trdh	**Read cycles:** Read enable Inactive time measured from the end of one read cycle to the beginning of the next read cycle.	5	No handshake	N/A
		26	Handshake	
Trdl	**Read cycles:** Read enable active low time. CS_-controlled read cycles: CS_ low time OE_-controlled read cycles: OE_ low time	Asynchronous Register Access: 26		N/A
		Synchronous Register Access: (5*Host Clock) + 21		
		SRAM Access: (7*Memory clock) + 21		
Trdwt	Time from rising edge of OE_ to falling edge of WAIT_	N/A		20
Trdz	Time from rising edge of CS_ or OE_, whichever comes first, to the data bus floating state	3		15
Twaitn	WAIT_/RDY_ assertion time from the falling edge of CS_.	N/A		20
Twrah	**Write cycles:** Address hold time from the rising edge of CS_ or WE_, whichever comes first.	0		N/A
Twras	**Write cycles:** Address valid setup time to the falling edge of CS_ or WE_, whichever comes first.	0		N/A

Table 4.48: Type C Host Interface Timing Parameters

Symbol	Description	Min (ns): Time and Conditions		Max (ns): Time and Conditions
Twrcyc	**Write cycle time requirement:** Time from the beginning of one write cycle to the beginning of the next write cycle.	10.5 *OR* Host Clock Period *OR* Memory Clock Period, ***whichever is largest***.		N/A
Twrh	**Write Enable Inactive Time:** Time from the end of one write cycle to the beginning of the next write cycle.	3.5		N/A
Twrl	**Write Enable Active time:** CS_-controlled write cycle: CS_ active time WR_-controlled write cycle: WR_ active time	7		N/A
Twrrdh	**BE_ Assertion Time:** Immediately after a read cycle: Time from the rising edge of CS_ to the falling edge of WE_.	5	No Hand-shake	N/A
		26	Handshake	
	OE_ Assertion Time: immediately following a write cycle: Time from the rising edge of CS_ to the falling edge of OE_.	5	No Hand-shake	N/A
		26	Handshake	
Twrrds	**Time for OE_ to be De-asserted:** Before a write cycle: Time from rising edge of OE_ to falling edge of CS_ for write cycles.	5	No Hand-shake	N/A
		26	Handshake	
	Time for WE_ to be De-asserted: Before a read cycle: Time from the rising edge of WE_ to the falling edge of CS_ for read cycles.	5	No Hand-shake	N/A
		26	Handshake	
Twrwt	Time from WR_ rising edge until WAIT_ falling edge	N/A		20
Twttri	Time from rising edge of CS_ to beginning of tri-state condition of WAIT_	N/A		29

4.4.5 Video Input Interface

Figure 4.69: Video Parallel Input Clock Timing

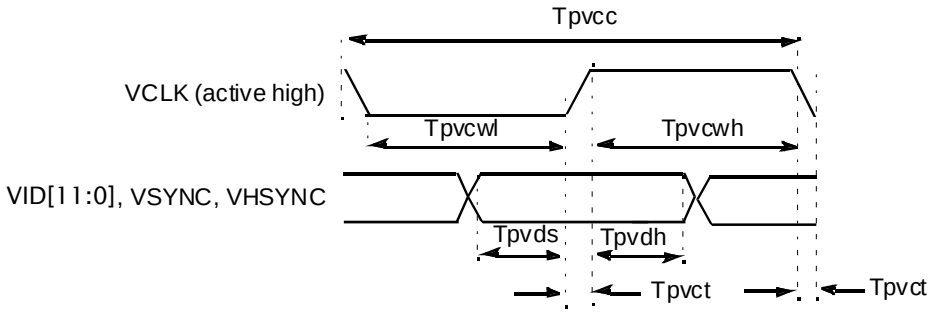


Table 4.49: Video Parallel Input Clock Timing

Symbol	Parameter	Min	Max	Unit
T_{pvcc}	VCLK cycle time	9.35	–	ns
T_{pvcw}	VCLK high/low pulse width	4.66 (low) 4.69 (high)	– –	ns
T_{pvct}	VCLK rise/fall transition time	–	2.08	ns
T_{pvds}	VID[11:0] data setup time	.51	–	ns
T_{pvdh}	VID[11:0] data hold time	.47	–	ns

4.4.6 Display Controller Interface Timing

Figure 4.70: Parallel LCD Data Format (LSC1 and LSC0 Divided by One, 10 pF Load)

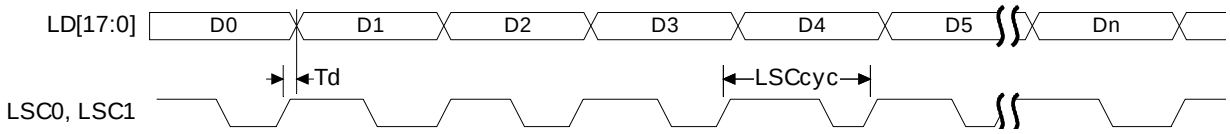


Table 4.50: Parallel Mode Divide-by-one Timing Parameters

Symbol	Parameter	Min	Max	Unit
Td	LD signals output delay from LSC rising edge	0	2	ns
LSC0 and LSC1 clock duty cycle		40	60	%
LSCcyc	LSC0, LSC1 clock cycle time	12.15	N/A	ns

Note: The delay times given in Table 4.51 are with respect to the rising edge of LSC0 and LSC1; 90% of the clock signal's high level. The setup and hold times may also be with respect to the clocks' falling edges. In this case, the times measure when the clock signal is within 10% of its signal's low level.

Figure 4.71: Parallel LCD Data Format (Clock Divided by Two)

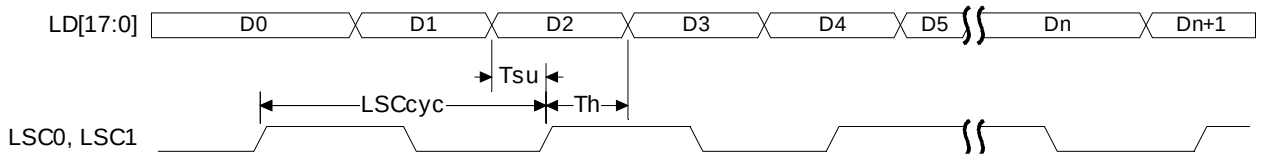


Table 4.51: Parallel Mode Divide-by-two Timing Parameters

Symbol	Parameter	Min	Max	Unit
LSCcyc	LSC0 and LSC1 clock cycle time	24.3	N/A	ns
LSC0 and LSC1 clock duty cycle		40	60	%
Tsu	Signal setup time with respect to LSC0 or LSC1 rising edge	4	N/A	ns
Th	Signal hold time with respect to LSC0 or LSC1 rising edge	4	N/A	ns

Note: The setup and hold times given in Table 4.51 are with respect to the rising edge of LSC0 and LSC1; 90% of the clock signal's high level. The setup and hold times may also be with respect to the clocks' falling edges. In this case, the times measure when the clock signal is within 10% of its signal's low level.

Figure 4.72: Parallel LCD Data Format (LSC[1:0] Divided by Four)

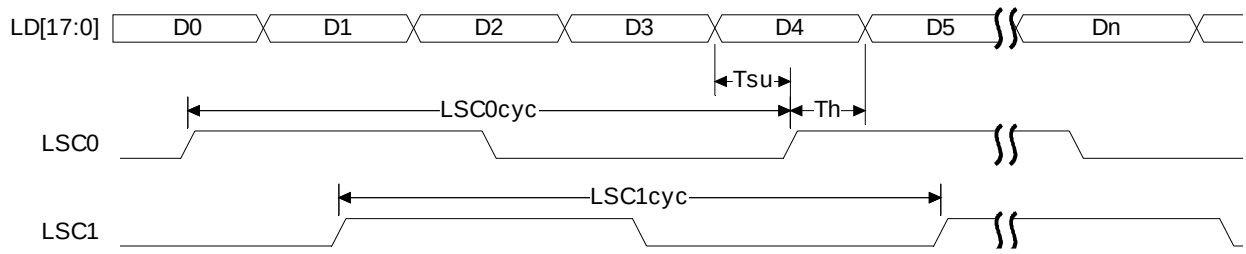


Table 4.52: Parallel Mode Divide-by-four Timing Parameters

Symbol	Parameter	Min	Max	Unit
LSC1cyc LSC0cyc	LSC0 and LSC1 clock cycle time, respectively	24.3	N/A	ns
LSC0 and LSC1 clock duty cycle		40	60	%
Tsu	Signal setup time with respect to LSC0 or LSC1 rising edge	4	N/A	ns
Th	Signal hold time with respect to LSC0 or LSC1 rising edge	4	N/A	ns

Figure 4.73: Serial Mode (10pF Load)

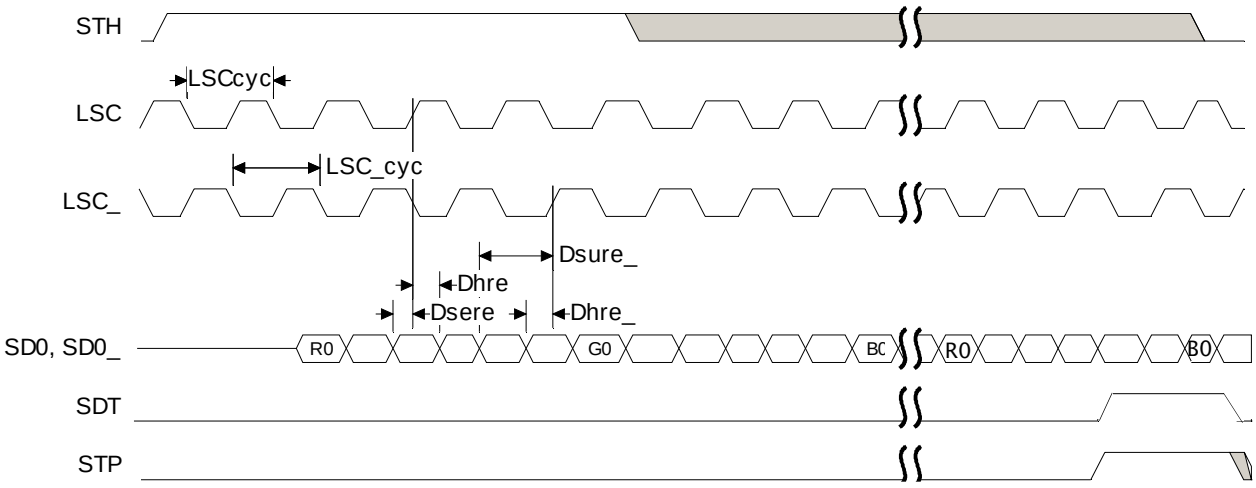


Table 4.53: Serial Mode (10 pF Load) Timing Parameters

Symbol	Parameter	Min	Max	Unit
LSCcyc	LSC clock cycle time	24.3	N/A	ns
Dsure	Data signals setup time from LSC rising edge	5	N/A	ns
Dhre	Data signals hold time from LSC rising edge	4.5	N/A	ns
LSC clock duty cycle		40	60	%
LSCcyc_	LSC_clock cycle time	24.3	N/A	ns
Dsure_	Data signals setup time from LSC_ rising edge	5	N/A	ns
Dhre_	Data signals hold time from LSC_ rising edge	4.5	N/A	ns
LSC_ clock duty cycle		40	60	%

Figure 4.74: SPI (10pF Load) Interface Timing

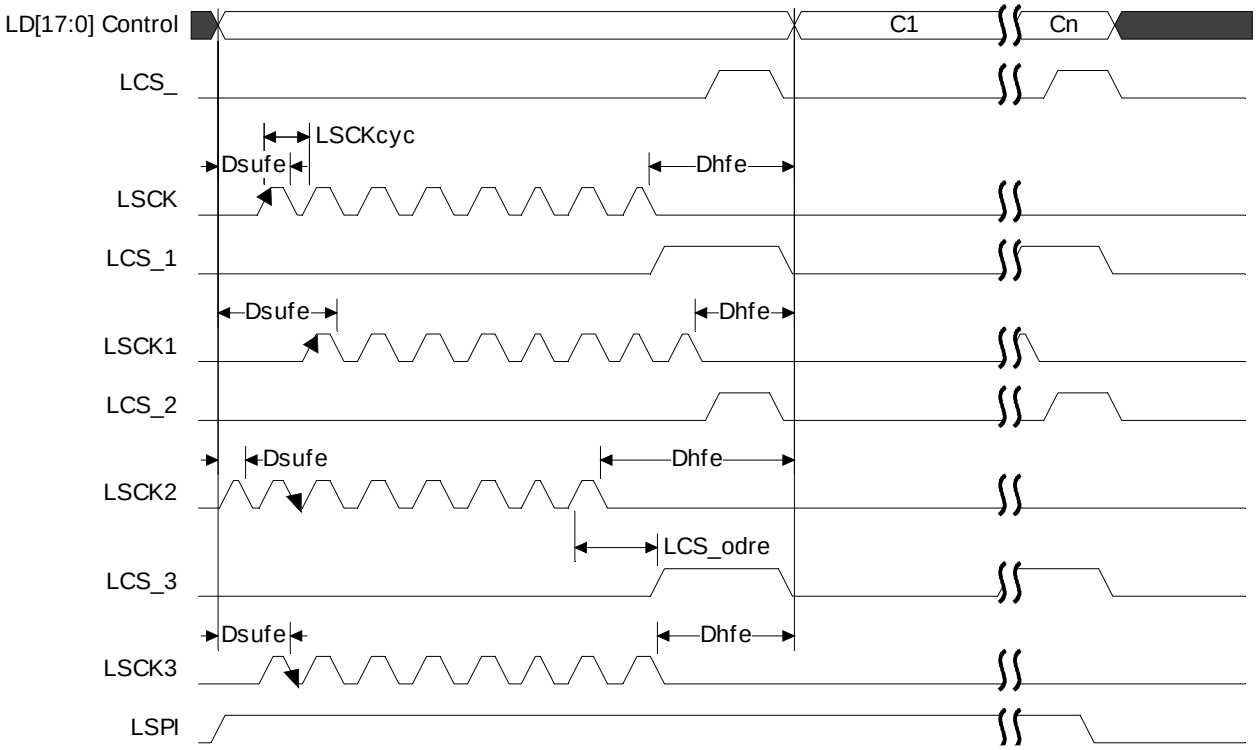


Table 4.54: SPI Timing Parameters

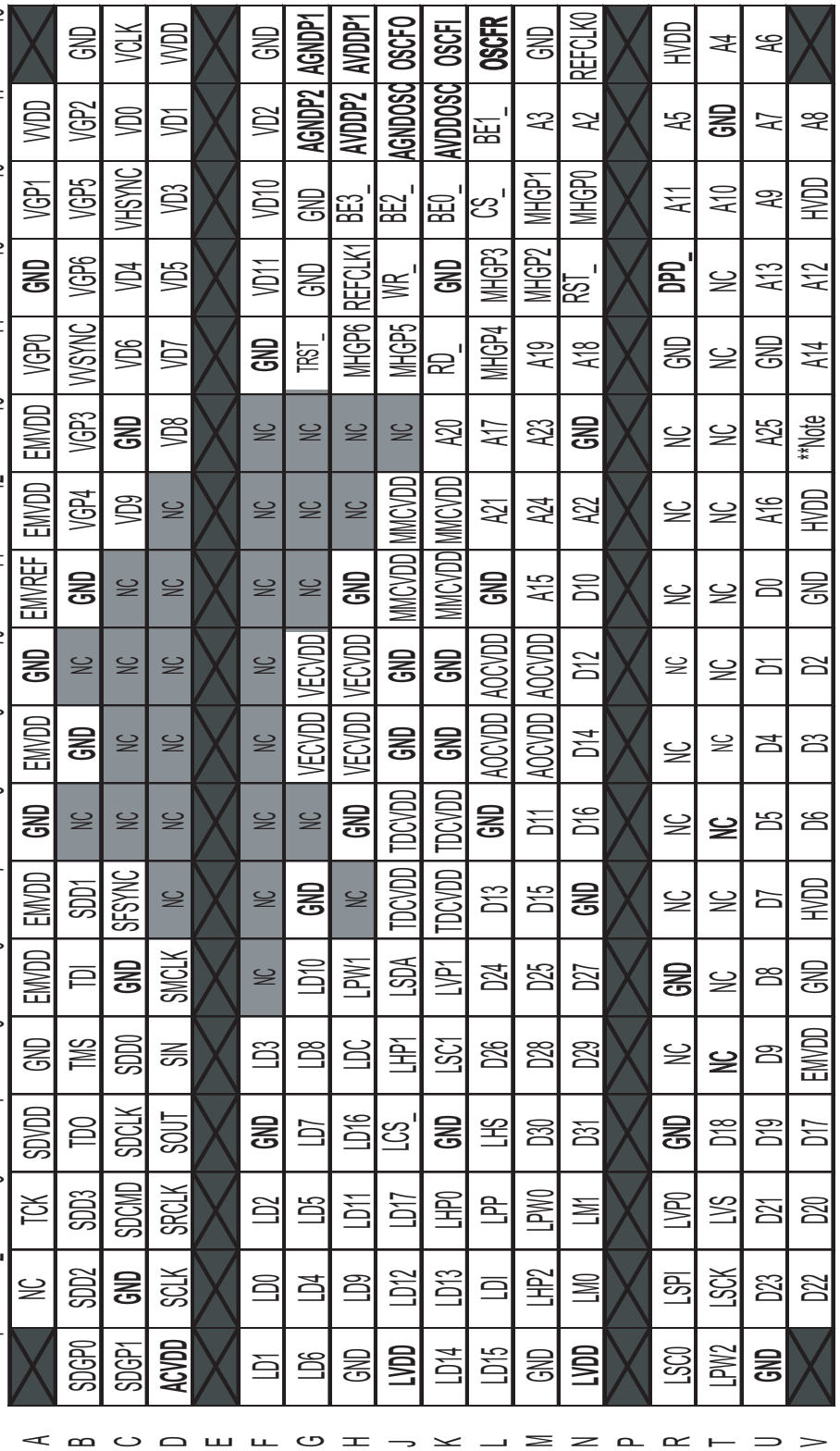
Symbol	Parameter	Min	Max	Unit
LSCKcyc	LSCK clock cycle time	24.3	N/A	ns
Dsufe	Signal setup time from LSCK falling edge	8	N/A	ns
Dhfe	Signal hold time from LSCK falling edge	10	N/A	ns
LSCK clock duty cycle		45	55	%

4.4.7 Ball Map

The following GoForce 5500 Ball Map diagrams illustrate the GoForce 5500-2MI, GoForce 5500-8ME, and GoForce 5500-XT memory configuration options.

- GoForce 5500-2MI: 2 MB additional memory
- GoForce 5500-8ME: 8 MB additional memory

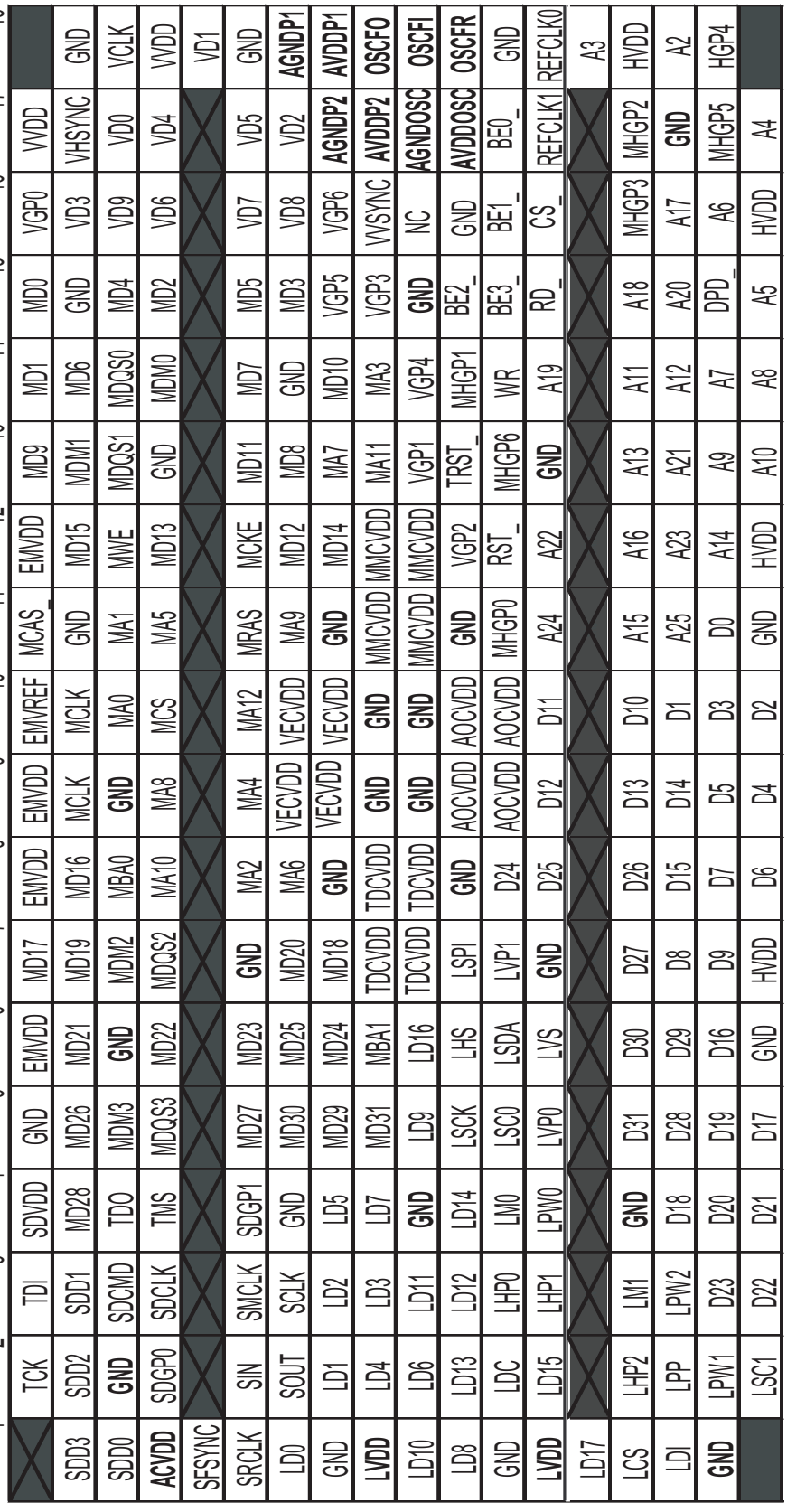
Figure 4.75: GoForce 5500 Ball Map (GoForce 5500-2MI and 8ME)



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
A		NC	TCK	SDVDD	GND	EMVDD	EMVDD	GND	EMVDD	GND	EMVREF	EMVDD	EMVDD	VGP0	GND	VGP1	VVDD	
B	SDGP0	SDD2	SDD3	TDO	TMS	TDI	SDD1	NC	GND	NC	GND	VGP4	VGP3	VVSYNC	VGP6	VGP5	VGP2	GND
C	SDGP1	GND	SDCMD	SDCLK	SDD0	GND	SFSYNC	NC	NC	NC	NC	VD9	GND	VD6	VD4	VHSYNC	VD0	VCLK
D	ACVDD	SCLK	SRCLK	SOUT	SIN	SMCLK	NC	NC	NC	NC	NC	NC	VD8	VD7	VD5	VD3	VD1	VVDD
E																		
F	LD1	LD0	LD2	GND	LD3	NC	NC	NC	NC	NC	NC	NC	NC	GND	VD11	VD10	VD2	GND
G	LD6	LD4	LD5	LD7	LD8	LD10	GND	NC	VECVDD	VECVDD	NC	NC	NC	TRST_	GND	GND	AGNDP2	AGNDP1
H	GND	LD9	LD11	LD16	LDC	LPW1	NC	GND	VECVDD	VECVDD	GND	NC	NC	MHGP6	REFCLK1	BE3_	AVDDP2	AVDDP1
J	LVDD	LD12	LD17	LCS_	LHP1	LSDA	TDCVDD	TDCVDD	GND	GND	MMCVDD	MMCVDD	NC	MHGP5	WR_	BE2_	AGNDOSC	OSCFO
K	LD14	LD13	LHP0	GND	LSC1	LVP1	TDCVDD	TDCVDD	GND	GND	MMCVDD	MMCVDD	A20	RD_	GND	BE0_	AVDDOSC	OSCFI
L	LD15	LDI	LPP	LHS	D26	D24	D13	GND	AOCVDD	AOCVDD	GND	A21	A17	MHGP4	MHGP3	CS_	BE1_	OSCFR
M	GND	LHP2	LPW0	D30	D28	D25	D15	D11	AOCVDD	AOCVDD	A15	A24	A23	A19	MHGP2	MHGP1	A3	GND
N	LVDD	LM0	LM1	D31	D29	D27	GND	D16	D14	D12	D10	A22	GND	A18	RST_	MHGP0	A2	REFCLK0
P																		
R	LSC0	LSPI	LVP0	GND	NC	GND	NC	NC	NC	NC	NC	NC	NC	GND	DPD_	A11	A5	HVDD
T	LPW2	LSCK	LVS	D18	NC	NC	NC	NC	NC	NC	NC	NC	NC	NC	NC	A10	GND	A4
U	GND	D23	D21	D19	D9	D8	D7	D5	D4	D1	D0	A16	A25	GND	A13	A9	A7	A6
V		D22	D20	D17	EMVDD	GND	HVDD	D6	D3	D2	GND	HVDD	**Note	A14	A12	HVDD	A8	

Note: V13 is NC for GoForce 5500 -8ME and EMVDD for GoForce 5500 -2MI

Figure 4.76: GoForce 5500-XT Ball Map



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
A		TCK	TDI	SDVDD	GND	EMVDD	MD17	EMVDD	EMVDD	EMVREF	MCAS_	EMVDD	MD9	MD1	MD0	VGP0	VVDD	
B	SDD3	SDD2	SDD1	MD28	MD26	MD21	MD19	MD16	MCLK	MCLK	GND	MD15	MDM1	MD6	GND	VD3	VHSYNC	GND
C	SDD0	**GND**	SDCMD	TDO	MDM3	**GND**	MDM2	MBA0	**GND**	MA0	MA1	MWE	MDQS1	MDQS0	MD4	VD9	VD0	VCLK
D	**ACVDD**	SDGP0	SDCLK	TMS	MDQS3	MD22	MDQS2	MA10	MA8	MCS	MA5	MD13	GND	MDM0	MD2	VD6	VD4	VVDD
E	SFSYNC																	VD1
F	SRCLK	SIN	SMCLK	SDGP1	MD27	MD23	**GND**	MA2	MA4	MA12	MRAS	MCKE	MD11	MD7	MD5	VD7	VD5	GND
G	LD0	SOUT	SCLK	GND	MD30	MD25	MD20	MA6	VECVDD	VECVDD	MA9	MD12	MD8	GND	MD3	VD8	VD2	**AGNDP1**
H	GND	LD1	LD2	LD5	MD29	MD24	MD18	**GND**	VECVDD	VECVDD	**GND**	MD14	MA7	MD10	VGP5	VGP6	**AGNDP2**	**AVDDP1**
J	**LVDD**	LD4	LD3	LD7	MD31	MBA1	TDCVDD	TDCVDD	**GND**	**GND**	MMCVDD	MMCVDD	MA11	MA3	VGP3	VVSYNC	**AVDDP2**	**OSCF0**
K	LD10	LD6	LD11	**GND**	LD9	LD16	TDCVDD	TDCVDD	**GND**	**GND**	MMCVDD	MMCVDD	VGP1	VGP4	**GND**	NC	**AGNDOSC**	**OSCF1**
L	LD8	LD13	LD12	LD14	LSCK	LHS	LSPI	**GND**	AOCVDD	AOCVDD	**GND**	VGP2	TRST_	MHGP1	BE2_	GND	**AVDDOSC**	**OSCFR**
M	GND	LDC	LHP0	LM0	LSC0	LSDA	LVP1	D24	AOCVDD	AOCVDD	MHGP0	RST_	MHGP6	WR	BE3_	BE1_	BE0_	GND
N	**LVDD**	LD15	LHP1	LPW0	LVP0	LVS	**GND**	D25	D12	D11	A24	A22	**GND**	A19	RD_	CS_	REFCLK1	REFCLK0
P	LD17																	A3
R	LCS	LHP2	LM1	**GND**	D31	D30	D27	D26	D13	D10	A15	A16	A13	A11	A18	MHGP3	MHGP2	HVDD
T	LDI	LPP	LPW2	D18	D28	D29	D8	D15	D14	D1	A25	A23	A21	A12	A20	A17	**GND**	A2
U	**GND**	LPW1	D23	D20	D19	D16	D9	D7	D5	D3	D0	A14	A9	A7	DPD_	A6	MHGP5	HGP4
V		LSC1	D22	D21	D17	GND	HVDD	D6	D4	D2	GND	HVDD	A10	A8	A5	HVDD	A4	

4.4.7.1 Ball to Signal Mapping

4.4.7.1.1 GoForce 5500

In Table 4.55 and Table 4.56 the GoForce 5500 ball-to-signal mappings appear listed by ball symbol in alphabetical order for the GoForce 5500-2MI and GoForce 5500-8ME; and GoForce 5500-XT configurations, respectively.

Greyed-in table cells correspond to unused areas on the GoForce 5500.

Table 4.55: Alphabetically-ordered GoForce 5500 Ball-to-Signal Mapping for GoForce 5500-2MI and GoForce 5500-8ME

Ball	Signal Name	Ball	Signal Name	Ball	Signal Name	Ball	Signal Name	Ball	Signal Name	Ball	Signal Name
A1		C11	NC	G2	LD4	J13	NC	M6	D25	T4	D18
A2	NC	C12	VD9	G3	LD5	J14	MHGP5	M7	D15	T5:T15	NC
A3	TCK	C13	GND	G4	LD7	J15	WR_	M8	D11	T16	A10
A4	SDVDD	C14	VD6	G5	LD8	J16	BE2_	M9	AOCVDD	T17	GND
A5	GND	C15	VD4	G6	LD10	J17	AGNDOSC	M10	AOCVDD	T18	A4
A6	EMVDD	C16	VHSYNC	G7	GND	J18	OSCFO	M11	A15	U1	GND
A7	EMVDD	C17	VD0	G8	NC	K1	LD14	M12	A24	U2	D23
A8	GND	C18	VCLK	G9	VECVDD	K2	LD13	M13	A23	U3	D21
A9	EMVDD	D1	ACVDD	G10	VECVDD	K3	LHP0	M14	A19	U4	D19
A10	GND	D2	SCLK	G11	NC	K4	GND	M15	MHGP2	U5	D9
A11	EMVREF	D3	SRCLK	G12	NC	K5	LSC1	M16	MHGP1	U6	D8
A12	EMVDD	D4	SOUT	G13	NC	K6	LVP1	M17	A3	U7	D7
A13	EMVDD	D5	SIN	G14	TRST_	K7	TDCVDD	M18	GND	U8	D5
A14	VGP0	D6	SMCLK	G15	GND	K8	TDCVDD	N1	LVDD	U9	D4
A15	GND	D7	NC	G16	GND	K9	GND	N2	LM0	U10	D1
A16	VGP1	D8	NC	G17	AGNDP2	K10	GND	N3	LM1	U11	D0
A17	VVDD	D9	NC	G18	AGNDP1	K11	MMCVDD	N4	D31	U12	A16
A18		D10	NC	H1	GND	K12	MMCVDD	N5	D29	U13	A25
B1	SDGP0	D11	NC	H2	LD9	K13	A20	N6	D27	U14	GND
B2	SDD2	D12	NC	H3	LD11	K14	RD_	N7	GND	U15	A13
B3	SDD3	D13	VD8	H4	LD16	K15	GND	N8	D16	U16	A9
B4	TD0	D14	VD7	H5	LDC	K16	BE0_	N9	D14	U17	A7
B5	TMS	D15	VD5	H6	LPW1	K17	AVDDOSC	N10	D12	U18	A6
B6	TDI	D16	VD3	H7	NC	K18	OSCFI	N11	D10	V1	
B7	SDD1	D17	VD1	H8	GND	L1	LD15	N12	A22	V2	D22
B8	NC	D18	VVDD	H9	VECVDD	L2	LDI	N13	GND	V3	D20
B9	GND	E1:E18		H10	VECVDD	L3	LPP	N14	A18	V4	D17
B10	NC	F1	LD1	H11	GND	L4	LHS	N15	RST_	V5	EMVDD
B11	GND	F2	LD0	H12	NC	L5	D26	N16	MHGP0	V6	GND
B12	VGP4	F3	LD2	H13	NC	L6	D24	N17	A2	V7	HVDD
B13	VGP3	F4	GND	H14	MHGP6	L7	D13	N18	REFCLK0	V8	D6
B14	VVSYNC	F5	LD3	H15	REFCLK1	L8	GND	P1:P18		V9	D3
B15	VGP6	F6	NC	H16	BE3_	L9	AOCVDD	R1	LSC0	V10	D2
B16	VGP5	F7	NC	H17	AVDDP2	L10	AOCVDD	R2	LSP1	V11	GND
B17	VGP2	F8	NC	H18	AVDDP1	L11	GND	R3	LVP0	V12	HVDD
B18	GND	F9	NC	J1	LVDD	L12	A21	R4	GND	V13	EMVDD
C1	SDGP1	F10	NC	J2	LD12	L13	A17	R5	NC	V14	A14
C2	GND	F11	NC	J3	LD17	L14	MHGP4	R6	GND	V15	A12
C3	SDCMD	F12	NC	J4	LCS_	L15	MHGP3	R7:R13	NC	V16	HVDD
C4	SDCLK	F13	NC	J5	LHP1	L16	CS_	R14	GND	V17	A8
C5	SDD0	F14	GND	J6	LSDA	L17	BE1_	R15	DPD_	V18	
C6	GND	F15	VD11	J7	TDCVDD	L18	OSCFR	R16	A11		
C7	SFSYNC	F16	VD10	J8	TDCVDD	M1	GND	R17	A5		
C8	NC	F17	VD2	J9	GND	M2	LHP2	R18	HVDD		
C9	NC	F18	GND	J10	GND	M3	LPW0	T1	LPW2		
C10	NC	G1	LD6	J11	MMCVDD	M4	D30	T2	LSCK		
				J12	MMCVDD	M5	D28	T3	LVS		

Table 4.56: Alphabetically Ordered GoForce 5500 Ball-to-signal Mapping for GoForce 5500-XTI

Ball	Signal Name	Ball	Signal Name	Ball	Signal Name	Ball	Signal Name	Ball	Signal Name	Ball	Signal Name
A1		C14	MDQS0	G6	MD25	K1	LD10	M14	WR_	T6	D29
A2	TCK	C15	MD4	G7	MD20	K2	LD6	M15	BE3_	T7	D8
A3	TDI	C16	VD9	G8	MA6	K3	LD11	M16	BE1_	T8	D15
A4	SDVDD	C17	VD0	G9	VECVDD	K4	GND	M17	BE0_	T9	D14
A5	GND	C18	VCLK	G10	VECVDD	K5	LD9	M18	GND	T10	D1
A6	EMVDD	D1	ACVDD	G11	MA9	K6	LD16	N1	LVDD	T11	A25
A7	MD17	D2	SDGP0	G12	MD12	K7	TDCVDD	N2	LD15	T12	A23
A8	EMVDD	D3	SDCLK	G13	MD8	K8	TDCVDD	N3	LHP1	T13	A21
A9	EMVDD	D4	TMS	G14	GND	K9	GND	N4	LPW0	T14	A12
A10	EMVREF	D5	MDQS3	G15	MD3	K10	GND	N5	LVP0	T15	A20
A11	MCAS_	D6	MD22	G16	VD8	K11	MMCVDD	N6	LVS	T16	A17
A12	EMVDD	D7	MDQS2	G17	VD2	K12	MMCVDD	N7	GND	T17	GND
A13	MD9	D8	MA10	G18	AGNDP1	K13	VGP1	N8	D25	T18	A2
A14	MD1	D9	MA8	H1	GND	K14	VGP4	N9	D12	U1	GND
A15	MD0	D10	MCS	H2	LD1	K15	GND	N10	D11	U2	LPW1
A16	VGP0	D11	MA5	H3	LD2	K16	NC	N11	A24	U3	D23
A17	VVDD	D12	MD13	H4	LD5	K17	AGNDOSC	N12	A22	U4	D20
A18		D13	GND	H5	MD29	K18	OSCFI	N13	GND	U5	D19
B1	SDD3	D14	MDM0	H6	MD24	L1	LD8	N14	A19	U6	D16
B2	SDD2	D15	MD2	H7	MD18	L2	LD13	N15	RD_	U7	D9
B3	SDD1	D16	VD6	H8	GND	L3	LD12	N16	CS_	U8	D7
B4	MD28	D17	VD4	H9	VECVDD	L4	LD14	N17	REFCLK1	U9	D5
B5	MD26	D18	VVDD	H10	VECVDD	L5	LSCK	N18	REFCLK0	U10	D3
B6	MD21	E1	SFSYNC	H11	GND	L6	LHS	P1	LD17	U11	D0
B7	MD19	E2:E17		H12	MD14	L7	LSPI	P2:P17		U12	A14
B8	MD16	E18	VD1	H13	MA7	L8	GND	P18	A3	U13	A9
B9	MCLK_	F1	SRCLK	H14	MD10	L9	AOCVDD	R1	LCS	U14	A7
B10	MCLK	F2	SIN	H15	VGP5	L10	AOCVDD	R2	LHP2	U15	DPD_
B11	GND	F3	SMCLK	H16	VGP6	L11	GND	R3	LM1	U16	A6
B12	MD15	F4	SDGP1	H17	AGNDP2	L12	VGP2	R4	GND	U17	MHGP5
B13	MDM1	F5	MD27	H18	AVDDP1	L13	TRST_	R5	D31	U18	MHGP4
B14	MD6	F6	MD23	J1	LVDD	L14	MHGP1	R6	D30	V1	
B15	GND	F7	GND	J2	LD4	L15	BE2_	R7	D27	V2	LSC1
B16	VD3	F8	MA2	J3	LD3	L16	GND	R8	D26	V3	D22
B17	VHSYNC	F9	MA4	J4	LD7	L17	AVDDOSC	R9	D13	V4	D21
B18	GND	F10	MA12	J5	MD31	L18	OSCFR	R10	D10	V5	D17
C1	SDD0	F11	MRAS_	J6	MBA1	M1	GND	R11	A15	V6	GND
C2	GND	F12	MCKE	J7	TDCVDD	M2	LDC	R12	A16	V7	HVDD
C3	SDCMD	F13	MD11	J8	TDCVDD	M3	LHP0	R13	A13	V8	D6
C4	TD0	F14	MD7	J9	GND	M4	LM0	R14	A11	V9	D4
C5	MDM3	F15	MD5	J10	GND	M5	LSC0	R15	A15	V10	D2
C6	GND	F16	VD7	J11	MMCVDD	M6	LSDA	R16	MHGP3	V11	GND
C7	MDM2	F17	VD5	J12	MMCVDD	M7	LVP1	R17	MHGP2	V12	HVDD
C8	MBA0	F18	GND	J13	MA11	M8	D24	R18	HVDD	V13	A10
C9	GND	G1	LD0	J14	MA3	M9	AOCVDD	T1	LD1	V14	A8
C10	MA0	G2	SOUT	J15	VGP3	M10	AOCVDD	T2	LPP	V15	A5
C11	MA1	G3	SCLK	J16	VVSYNC	M11	MHGP0	T3	LPW2	V16	HVDD
C12	MWE	G4	GND	J17	AVDDP2	M12	RST_	T4	D18	V17	A4
C13	MDQS1	G5	MD30	J18	OSCFO	M13	MHGP6	T5	D28	V18	

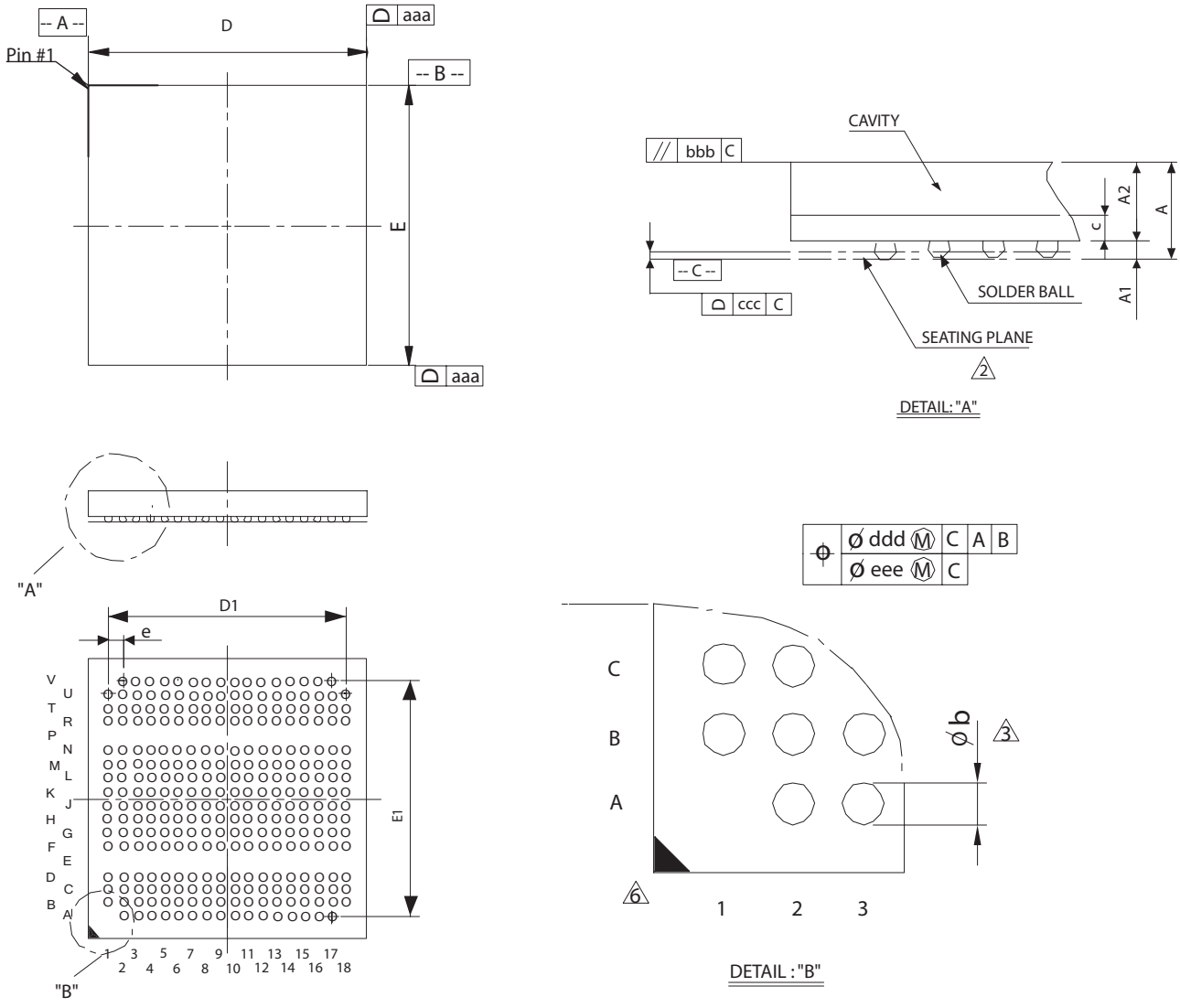
4.5 Mechanical Drawing

The mechanical specifications for the GoForce 5500 follow in this section.

Note: Note there are three different mechanical drawings corresponding to three different types of configuration choice:

- GoForce 5500-2MI 10 x10 mm packages (Figure 4.77)
- GoForce 5500-8ME 10 x 12 mm package (Figure 4.78)
- GoForce 5500-XT package (Figure 4.79)

Figure 4.77: GoForce 5500-2MI 10 X 10 mm Package Mechanical Drawing

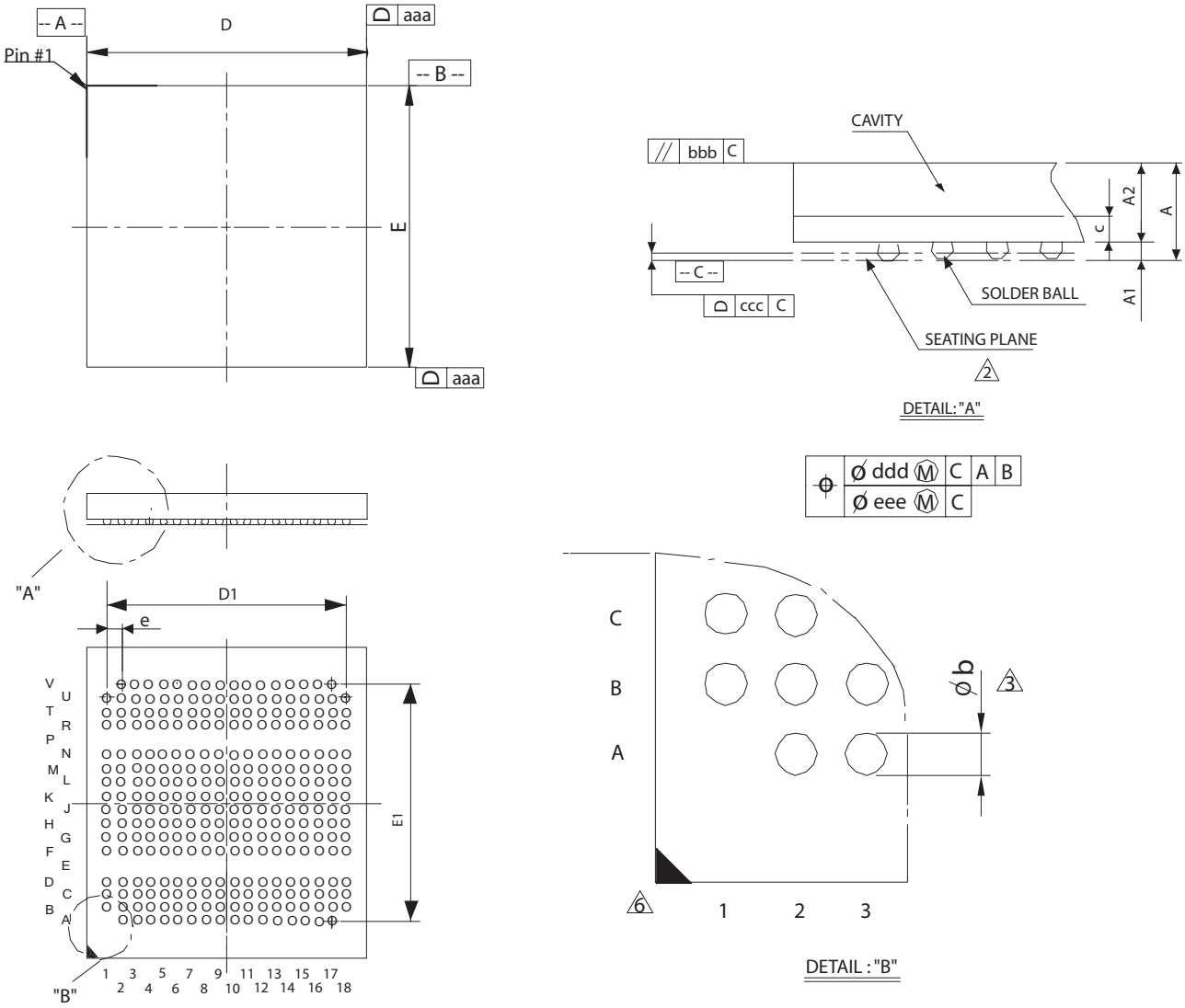


Symbol	Dimension in mm			Dimension in inch		
	MIN	NOM	MAX	MIN	NOM	MAX
A	1.10	1.20	1.40	0.043	0.047	0.055
A1	0.16	0.21	0.26	0.006	0.008	0.010
A2	0.91	0.96	1.01	0.036	0.038	0.040
c	0.22	0.26	0.30	0.009	0.010	0.012
D	9.90	10.00	10.10	0.390	0.394	0.398
E	9.90	10.00	10.10	0.390	0.394	0.398
D1	---	8.50	---	---	0.335	---
E1	---	8.50	---	---	0.335	---
e	---	0.50	---	---	0.020	---
b	0.25	0.30	0.35	0.010	0.012	0.014
aaa	0.15			0.006		
bbb	0.20			0.008		
ccc	0.08			0.003		
ddd	0.15			0.006		
eee	0.08			0.003		
MD/ME	18/18			18/18		
TOTAL No. OF BALLS:	284					

NOTE:

1. CONTROLLING DIMENSION: MILLIMETER
2. PRIMARY DATUM C AND SEATING PLANE ARE DEFINED BY THE SPHERICAL CROWNS OF THE SOLDER BALLS.
3. DIMENSION b IS MEASURED AT THE MAXIMUM SOLDER BALL DIAMETER, PARALLEL TO PRIMARY DATUM C.
4. THERE SHALL BE A MINIMUM CLEARANCE OF 0.25 mm BETWEEN THE EDGE OF THE SOLDER BALL AND THE BODY EDGE.
5. REFERENCE DOCUMENT: JEDEC MO-207.
6. THE PATTERN OF PIN 1 FIDUCIAL IS FOR REFERENCE ONLY.
7. SPECIAL CHARACTERISTICS C CLASS: bbb, ccc

Figure 4.78: GoForce 5500-8ME 10 x 12 mm Package Mechanical Drawing

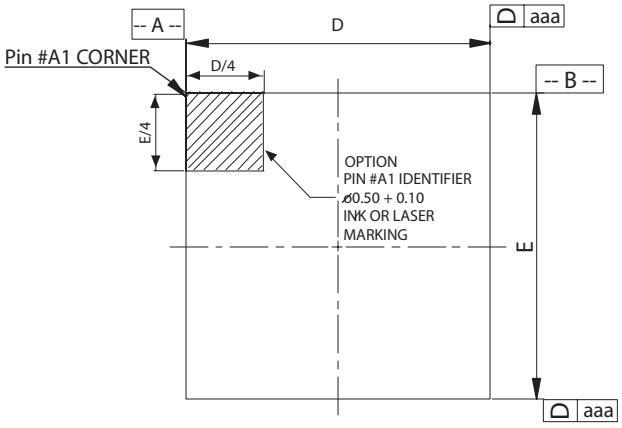


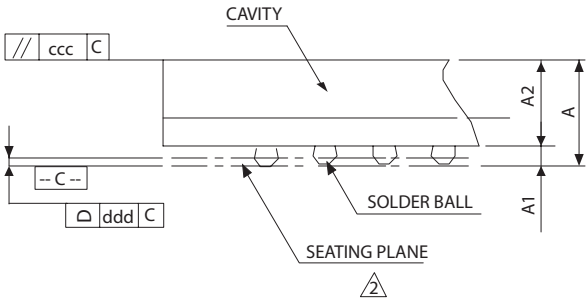
Symbol	Dimension in mm			Dimension in inch		
	MIN	NOM	MAX	MIN	NOM	MAX
A	1.10	1.20	1.40	0.043	0.047	0.055
A1	0.16	0.21	0.26	0.006	0.008	0.010
A2	0.91	0.96	1.01	0.036	0.038	0.040
c	0.22	0.26	0.30	0.009	0.010	0.012
D	9.90	10.00	10.10	0.390	0.394	0.398
E	11.90	12.00	12.10	0.469	0.472	0.476
D1	---	8.50	---	---	0.335	---
E1	---	8.50	---	---	0.335	---
e	---	0.50	---	---	0.020	---
b	0.25	0.30	0.35	0.010	0.012	0.014
aaa	0.15			0.006		
bbb	0.20			0.008		
ccc	0.08			0.003		
ddd	0.15			0.006		
eee	0.08			0.003		
MD/ME	18/18			18/18		
TOTAL No. OF BALLS:	284					

NOTE:

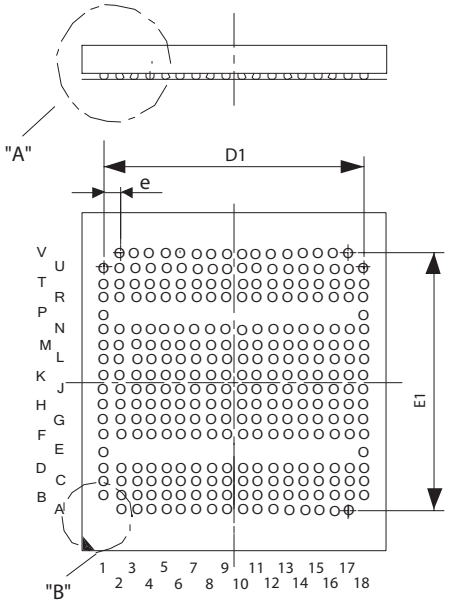
1. CONTROLLING DIMENSION: MILLIMETER
2. PRIMARY DATUM C AND SEATING PLANE ARE DEFINED BY THE SPHERICAL CROWNS OF THE SOLDER BALLS.
3. DIMENSION b IS MEASURED AT THE MAXIMUM SOLDER BALL DIAMETER, PARALLEL TO PRIMARY DATUM C.
4. THERE SHALL BE A MINIMUM CLEARANCE OF 0.25 mm BETWEEN THE EDGE OF THE SOLDER BALL AND THE BODY EDGE.
5. REFERENCE DOCUMENT: JEDEC MO-207.
6. THE PATTERN OF PIN 1 FIDUCIAL IS FOR REFERENCE ONLY.
7. SPECIAL CHARACTERISTICS C CLASS: bbb, ccc

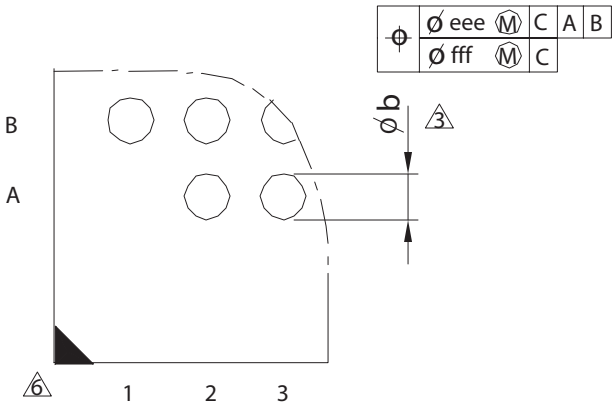
Figure 4.79: GoForce 5500-XT 10 x 12 mm Package Mechanical Drawing





DETAIL: "A"





DETAIL : "B"

Symbol	Dimension in mm			Dimension in inch		
	MIN	NOM	MAX	MIN	NOM	MAX
A	1.10	1.20	1.40	0.043	0.047	0.055
A1	0.16	0.21	0.26	0.006	0.008	0.010
A2	0.65	0.70	0.75	0.026	0.028	0.030
b	0.25	0.30	0.35	0.010	0.012	0.014
D	9.90	10.00	10.10	0.390	0.394	0.398
E	11.90	12.00	12.10	0.469	0.472	0.476
D1	---	8.50	---	---	0.335	---
E1	---	8.50	---	---	0.335	---
e	---	0.50	---	---	0.020	---
aaa	0.15			0.006		
ccc	0.20			0.008		
ddd	0.10			0.004		
eee	0.15			0.006		
fff	0.10			0.004		
MD/ME	18/18			18/18		
TOTAL No. OF BALLS:	288					

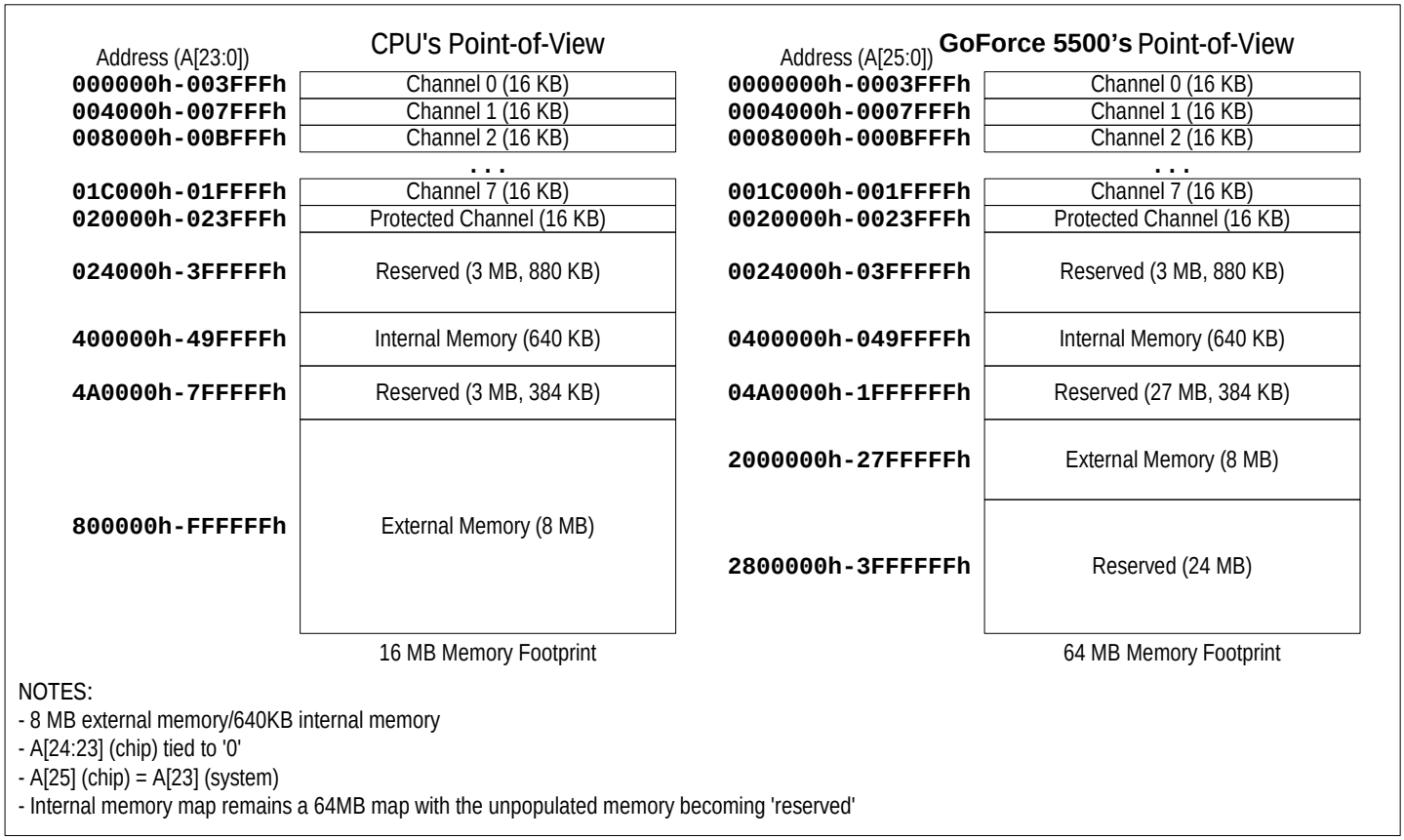
NOTE:

1. CONTROLLING DIMENSION: MILLIMETER
2. PRIMARY DATUM C AND SEATING PLANE ARE DEFINED BY THE SPHERICAL CROWNS OF THE SOLDER BALLS.
3. DIMENSION b IS MEASURED AT THE MAXIMUM SOLDER BALL DIAMETER, PARALLEL TO PRIMARY DATUM C.
4. THERE SHALL BE A MINIMUM CLEARANCE OF 0.25 mm BETWEEN THE EDGE OF THE SOLDER BALL AND THE BODY EDGE.
5. REFERENCE DOCUMENT: JEDEC MO-207.
6. THE PATTERN OF PIN 1 FIDUCIAL IS FOR REFERENCE ONLY.

Chapter 5 Memory Map

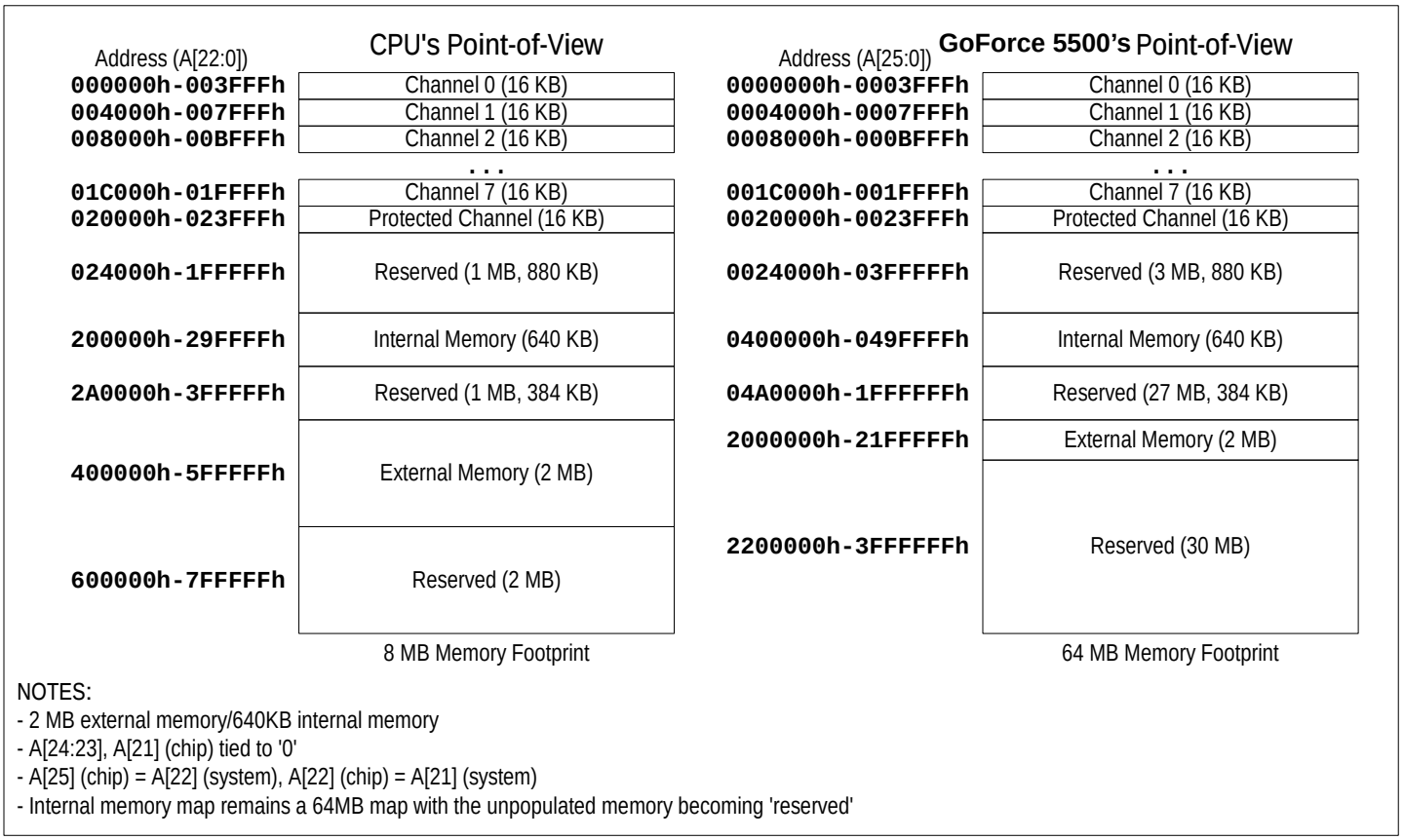
GoForce 5500 Address Map

Figure 5.1: Memory Configuration: 8MB External Memory, 640KB Internal Memory



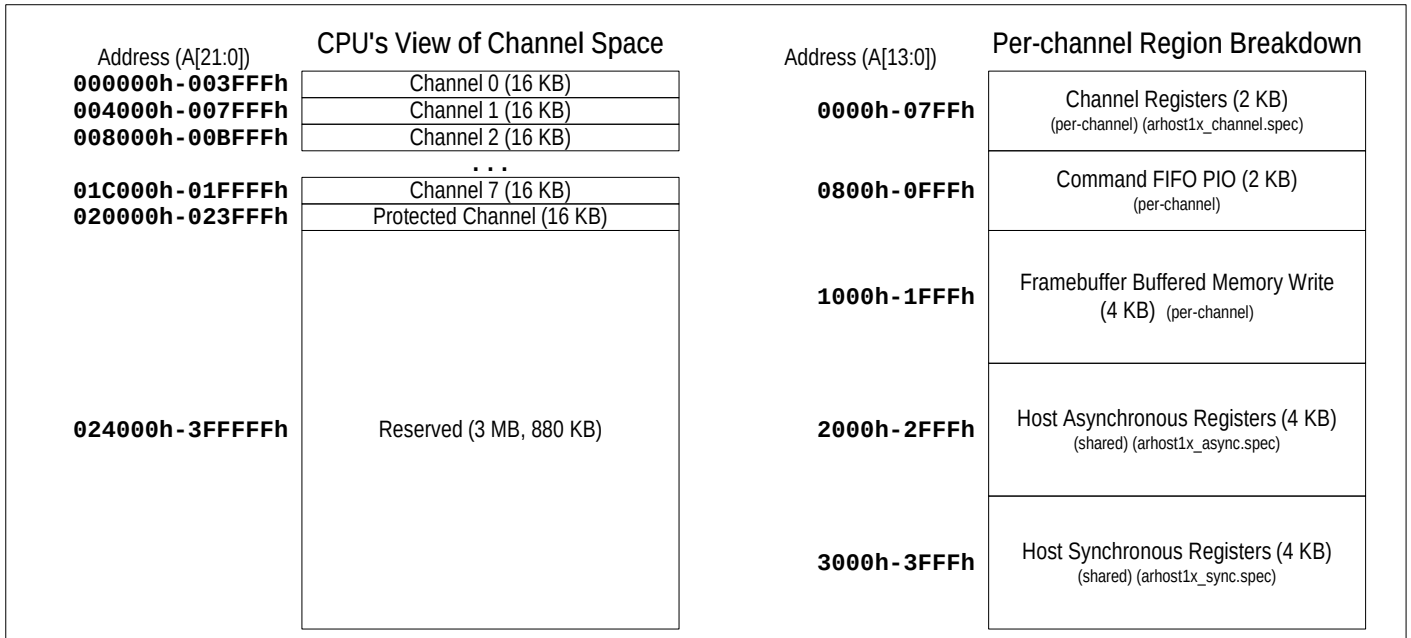
NOTES:
- 8 MB external memory/640KB internal memory
- A[24:23] (chip) tied to '0'
- A[25] (chip) = A[23] (system)
- Internal memory map remains a 64MB map with the unpopulated memory becoming 'reserved'

Figure 5.2: Memory Configuration: 2MB External Memory, 640KB Internal Memory



NOTES:
- 2 MB external memory/640KB internal memory
- A[24:23], A[21] (chip) tied to '0'
- A[25] (chip) = A[22] (system), A[22] (chip) = A[21] (system)
- Internal memory map remains a 64MB map with the unpopulated memory becoming 'reserved'

Figure 5.3: GoForce 5500 Channel Map Diagram



NOTES:
- Some FIFOs, such as the per-channel output FIFO and the read DMA FIFOs, are readable via offsets in the channel registers
- All writes to the Command FIFO PIO region will be pushed into that channel's command FIFO
- Any writes to the Framebuffer Buffered Memory Write region will be written into memory relative to an address programmed in the channel registers. Writes will be accumulated until a 128-bit boundary is crossed, upon which the data will be written to memory. Writing to this region is the preferred way of loading command streams into memory for command DMA.

This page left intentionally blank.

Chapter 6 Register Summary Table

This chapter is a summary of the GoForce 5500 registers, detailed in Chapter 7. Use this table to look up information about the registers or go directly to a specific register by mouse-clicking on the register name in the column marked “Register Name.”

Table 6.1: GoForce 5500 Register Summary

Table 6.1: GoForce 5500 Register Summary

Table 6.1: GoForce 5500 Register Summary

Table 6.1: GoForce 5500 Register Summary

Table 6.1: GoForce 5500 Register Summary

Table 6.1: GoForce 5500 Register Summary

Table 6.1: GoForce 5500 Register Summary

Table 6.1: GoForce 5500 Register Summary

Table 6.1: GoForce 5500 Register Summary

Table 6.1: GoForce 5500 Register Summary

Table 6.1: GoForce 5500 Register Summary

Table 6.1: GoForce 5500 Register Summary

Table 6.1: GoForce 5500 Register Summary

Table 6.1: GoForce 5500 Register Summary

Table 6.1: GoForce 5500 Register Summary

Table 6.1: GoForce 5500 Register Summary

Table 6.1: GoForce 5500 Register Summary

Table 6.1: GoForce 5500 Register Summary

Table 6.1: GoForce 5500 Register Summary

Table 6.1: GoForce 5500 Register Summary

Table 6.1: GoForce 5500 Register Summary

This page left blank intentionally

Chapter 7 GoForce 5500 Micro-classes

This chapter lists the GoForce 5500 microclass architecture components necessary for configuration.

Note: Write all writeable reserved bits with 0 only, unless otherwise specified.

Registers for the following are grouped by module. Mouse-click on a module name below to view information pertaining to it:

- Host Registers
- Host Microclass Registers
- I2S Registers
- SPB Registers
- MC Registers
- SD Registers
- VI Registers
- Display Registers
- EMC Registers

7.1 Host Registers

HOST1X_ASYNC_HCONFIG1_0

Offset: 000h
Read/Write: R/W
Reset: 0000.0000

Host configuration register 1. Includes Host Interface Type and Host Bus Width

Bit	Description
31:2	**Reserved**
3	**HIW – Host interface width.** 0 16Bit Host Interface width 1 32Bit Host Interface width
2:1	**Reserved**
0	**HIT– Host interface type.** 0 Type A Host Interface 1 Type C Host Interface

HOST1X_ASYNC_HCONFIG2_0

Offset: 004h
Read/Write: R/W
Reset: 0000.0000

Host Configuration register 2. Includes input and output byte swapping for both register and memory.

Bit	Description
31:8	**Reserved**
7:6	**MOBS** – Frame Buffer output byte swap control. 00 None 01 Byte16 10 Byte32 11 Word32
5:4	**MIBS** – Frame Buffer input byte swap control. 00 None 01 Byte16 10 Byte32 11 Word32
3:2	**HOBS –** Host output byte swap control**.** 00 None 01 Byte16 10 Byte32 11 Word32
1:0	**HIBS** Host input byte swap control**.** 00 None 01 Byte16 10 Byte32 11 Word32

HOST1X_ASYNC_ADRINCREG_0

Offset: 008h
Read/Write: R/W
Reset: 0000.0000

Address auto increment for indirect addressing mode

Bit	Description
31:2	**Reserved**
1	**IARDINC – Indirect addressing mode, read address auto increment enable.** 0 Disable 1 Enable
0	**IAWRINC – Indirect addressing mode, write address auto increment enable.** 0 Disable 1 Enable

HOST1X_ASYNC_RDWAITREG_0

Offset: 00Ch
Read/Write: R/W
Reset: 0000.0000

Synchronous Host read cycle wait state counter

Bit	Description
31:3	**Reserved**
2:0	**RDWAITST** – synchronous host interface read access wait state. 000 Wait0 001 Wait1 010 Wait2 011 Wait3 100 Wait4 101 Wait5 110 Wait6 111 Wait7

HOST1X_ASYNC_MODEREG_0

Offset: 010h
Read/Write: RO
Reset: 0000.0000

Read only mode pin registers

Bit	Description
31:4	**Reserved**
3:0	**HMD – Host mode (set by MHGP6 and MHGP[2:0] pins at reset)** 0000 Asynchronous No handshake direct host interface 0010 Asynchronous Handshake direct host interface with active-low ready 0011 Asynchronous Handshake direct host interface with active-high ready 0100 Asynchronous No handshake indirect host interface 0110 Asynchronous Handshake indirect host interface with active-low ready 0111 Asynchronous Handshake indirect host interface with active-high ready 1000 Synchronous No handshake direct host interface 1010 Synchronous Handshake direct host interface with active-low ready 1011 Synchronous Handshake direct host interface with active-high ready 1100 Synchronous No handshake indirect host interface 1110 Synchronous Handshake indirect host interface with active-low ready 1111 Synchronous Handshake indirect host interface with active-high ready

HOST1X_ASYNC_RSTREG_0

Offset: 014h
Read/Write: R/W
Reset: 0000.1000

Reset Register. Active low resets, so write a '1' to bring the modules out of reset.

Bit	Description
31:22	**Reserved**
21	**EFUSERSTN: Reset EFUSE.** 0 Enable 1 Disable
20	**CKRSTNI – Reset for clock dividers, need to disable all clocks before reset.** 0 Enable 1 Disable
19	**MCRSTN – Reset MC.** 0 Enable 1 Disable
18	**EMCRSTN – Reset EMC.** 0 Enable 1 Disable
17	**ICRSTN – Reset IC.** 0 Enable 1 Disable
16	**I2SFSRSTN – Reset I2SFS (FIFO).** 0 Enable 1 Disable
15	**I2SSRSTN – Reset I2SS (send).** 0 Enable 1 Disable
14	**I2SRRSTN – Reset I2SR (receive).** 0 Enable 1 Disable
13	**I2STRSTN – Reset I2ST (transmit).** 0 Enable 1 Disable
12	**DSPCORERSTN – Reset DSPCORE.** 0 Disable 1 Enable
11	**DSPRSTN – Reset DSP.** 0 Enable 1 Disable
10	**SDRSTN – Reset SD.** 0 Enable 1 Disable
9	**MERSTN – Reset ME.** 0 Enable 1 Disable
8	**GRMPDRSTN – Reset GRMPD.** 0 Enable 1 Disable

Bit	Description
7	**JERSTN – Reset JE.** 0 Enable 1 Disable
6	**EPPRSTN – Reset EPP.** 0 Enable 1 Disable
5	**ISPRSTN – Reset ISP.** 0 Enable 1 Disable
4	**VIRSTN – Reset VI.** 0 Enable 1 Disable
3	**DCRSTN – Reset DISPLAY.** 0 Enable 1 Disable
2	**NVRSTN – Reset 3D.** 0 Enable 1 Disable
1	**G2RSTN – Reset 2D/SB.** 0 Enable 1 Disable
0	**HCRSTN – Reset synchronous host blocks (command processor, DMA engine).** 0 Enable 1 Disable

HOST1X_ASYNC_PLL1CONFIG1_0

Offset: 018h
Read/Write: R/W
Reset: 0000.0000

CLOCK REGISTERS

To change a clock source or divide ratio, the clock should be turned off first by clearing the clock enable. It is important that no other fields be changed in the register while the clock enable is being cleared. After the new clock source or divide ratio have been programmed, the clock can be turned back on by enabling the clock enable. Enabling the clock and changing the clock source or divide ratio in the same write is not supported.

An exception to the rule above, the Host, Display, MC, EMC, and VI also support changing the clock divide ratio without disabling the clock. The display does this in hardware. Note that the EMC may require setting other registers to maintain memory integrity.

The divide ratio for a 5-bit divider is described in the following table. Other divider sizes are similar.

Value Divide Ratio

------ ------------

00000 1 (Pass through)
00001 1.5
00010 2
00011 2.5
00100 3
00101 3.5
00110 4

.
.
.

11100 15
11101 15.5
11110 16
11111 16.5

Or the general formula is as follows:

ddddh ratio=(dddd+1) + (h*0.5)

The divide ratio for a 11-bit clock divider (audio, IC, I2S, SD) is similar to the above formula with divide ratios of 1, 1.5, 2, 2.5 through 1023, 1023.5, 1024, and 1024.5

All five-bit clock dividers have the same clock ratio: from 1, 1.5, and so through 16.5; just as shown under Value Divide Ratio

PLL1 Config1 Register

Bit	Description
31:10	**Reserved**
9:8	**PLL1TOS – PLL1 Test Output Select.** 0 DC0 1 FO 2 VCO 3 FBDIV

Bit	Description
7:3	**Reserved**
2	**PLL1DCCO – PLL1 Duty cycle control enable.** 0 Disable 1 Enable
1	**PLL1BY – PLL1 bypass enable.** 0 Disable 1 Enable
0	**PLL1E – PLL1 enable. Should be enabled for closed-loop mode.** 0 Disable 1 Enable

HOST1X_ASYNC_PLL1CONFIG2_0

Offset: 01Ch
Read/Write: R/W
Reset: 2069.0010

PLL1 Config2 Register

Bit	Description
31:29	**PLL1P – PLL1 P divider**
28:19	**PLL1N – PLL1 N divider**
18:16	**PLL1M – PLL1 M divider**
15:12	**PLL1VCO – PLL1 VCO Range**
11:8	**PLL1LF – PLL1 Loop Filter**
7:4	**PLL1CP – PLL1 Charge Pump**
3:2	**PLL1RFS – PLL1 Reference clock select.** 0 REFCLK0: External Clock source 1 Crystal Oscillator 2 Relaxation Oscillator, ROSC 3 REFCLK1: External Clock source
1:0	**PLL1_PREDIV – PLL1 source clock pre-divider.** 0 Div1 1 Div2 2 Div4 3 Div8

HOST1X_ASYNC_PLL2CONFIG1_0

Offset: 020h
Read/Write: R/W
Reset: 0000.0000

PLL2 Config1 Register

Bit	Description
31:10	**Reserved**
9:8	**PLL2TOS – PLL2 Test Output Select.** 0 DC0 1 FO 2 VCO 3 FBDIV
3:7	**Reserved**
2	**PLL2DCCON – PLL2 Duty cycle control enable.** 0 Disables PLL2 Duty cycle control 1 Enable PLL2 Duty cycle control
1	**PLL2BYP – PLL2 bypass enable** 0 Disables PLL2 Bypass 1 Enables PLL2 Bypass
0	**PLL2E – PLL2 enable** **Note: Enable for closed-loop mode.** 0 Disables PLL 1 Enables PLL

HOST1X_ASYNC_PLL2CONFIG2_0

Offset: 024h
Read/Write: R/W
Reset: 2069.0010

PLL2 Config2 Register

Bit	Description
31:29	**PLL2P – PLL2 P divider**
28:19	**PLL2N – PLL2 N divider**
18:16	**PLL2M – PLL2 M divider**
15:12	**PLL2VCO – PLL2 VCO Range**
11:8	**PLL2LF – PLL2 Loop Filter**
7:4	**PLL2CP – PLL2 Charge Pump**
3:2	**PLL2RFS – PLL2 Reference clock select.** 0 REFCLK0: External Clock source 1 Crystal Oscillator 2 Relaxation Oscillator, ROSC 3 REFCLK1: External Clock Source
1:0	**PLL2_PREDIV – PLL2 source clock pre-divider.** 0 DIV1 1 DIV2 2 DIV4 3 DIV8

HOST1X_ASYNC_CLKCTRL_0

Offset: 028h
Read/Write: R/W
Reset: 0000.0000

CLK Control Register.
Enable CLK pad to get external clock.

Bit	Description
31:1	**Reserved**
0	**CLKIE – CLK Input Enable** This bit is effective only with asynchronous host interface. 0 Disable 1 Enable

HOST1X_ASYNC_VCLKCTRL_0

Offset: 02Ch
Read/Write: R/W
Reset: 0000.0000

VCLK Control Register. Enable the VCLK pad to get the external clock for VI.

Bit	Description
31:2	**Reserved**
1	**VCLK_PAD_INVERSION: VCLK invert enable.** 0 Disable 1 Enable
0	**VCLK_PAD_IE: VCLK input enable.** 0 Disable 1 Enable

HOST1X_ASYNC_XOCONFIG_0

Offset: 030h
Read/Write: R/W
Reset: 0000.0000

Crystal Oscillator Configuration Register

Bit	Description
31:16	**Reserved**
15:8	**OSCFI_SPARE – Crystal oscillator control**
7:4	**XOFS – Crystal oscillator frequency select**
3	**Reserved**
2	**XOMODE – Crystal oscillator mode (power).** 0 Low power 1 High power
1	**XOBP – Crystal oscillator bypass enable.** 0 Disable Internal Crystal Oscillator Bypass 1 Enable Internal Crystal Oscillator Bypass
0	**XOE – Crystal oscillator enable.** 0 Disable Internal Crystal Oscillator 1 Enable Internal Crystal Oscillator

HOST1X_ASYNC_OSCCONFIG_0

Offset: 034h
Read/Write: R/W
Reset: 0000.0000

Relaxation Oscillator Config Register

Bit	Description
31:16	**Reserved**
15:8	**OSCFR_ESD_SETUP** – Relaxation oscillator miscellaneous setup bits
7:4	**OSCFR_ESD_FSET** – Relaxation oscillator frequency select
3:1	**Reserved**
0	**OSCFR_ESD_EN –** Relaxation oscillator enable 0 Disable 1 Enable

HOST1X_ASYNC_HCCCONFIG_0

Offset: 038h
Read/Write: R/W
Reset: 0000.0000

Host1x Controller Clock Configuration

Bit	Description
31:17	**Reserved**
16	**HC_CLK_OVR_ON – Host1x second level clocks overrides.** 0 Disables Host1x second level clocks override. 1 Enables Host1x second level clocks override.
12:8	**HCCD – Host1x controller clock divider**
6:4	**HCCS – Host1x controller clock select.** 0 REFCLK0 1 PLL1 2 PLL2 3 Crystal Oscillator 4 ROSC 5 REFCLK1
1	**HCCIV: Host1x controller clock invert.** 0 Disables Host1x controller clock invert 1 Enables Host1x controller clock invert
0	**HC_CE: Host1x controller clock enable.** 0 Disables Host1x controller clock 1 Enables Host1x controller clock

HOST1X_ASYNC_DSPCCONFIG_0

Offset: 03Ch
Read/Write: R/W
Reset: 0000.0000

DSP Controller Clock Configuration

Bit	Description
31:24	**Reserved**
23:22	**RCV_TRANS_SEL – clock select for clk2dsp_rcv_inc** 00 Disabled 01 Pulsed once for each cycle of the 32KHz clock pin 10 Pulsed on each 64-bit read from memory. Convert to samples using: TRM_SIZE Samples ---------- --------- TRMSIZE_8 Count * 8 TRMSIZE_16 Count * 4 TRMSIZE_18 Count * 2 TRMSIZE_20 Count * 2 TRMSIZE_24 Count * 2 TRMSIZE_30 Count * 2 Note that the value in the I2S register field TRANSMIT_CTRL.TRM_RATE_CTRL may cause samples to be counted but not be actually transmitted. 11 Pulsed on each received sample written to memory. By this hardware definition of 'samples', a stereo frame of DMODE=ONE_AFTER_FRAME has one left channel and one right channel, and contains TWO samples.
21:20	**TRN_TRANS_SEL – clock select for clk2dsp_trn_inc** 00 Disabled 01 Pulsed once for each cycle of the 32KHz clock pin 10 Pulsed on each 64-bit read from memory. Convert to samples using: TRM_SIZE Samples ---------- --------- TRMSIZE_8 Count * 8 TRMSIZE_16 Count * 4 TRMSIZE_18 Count * 2 TRMSIZE_20 Count * 2 TRMSIZE_24 Count * 2 TRMSIZE_30 Count * 2 Note that the value in the I2S register field TRANSMIT_CTRL.TRM_RATE_CTRL may cause samples to be counted but not be actually transmitted. 11 = Pulsed on each received sample written to memory. By this hardware definition of 'samples', a stereo frame of DMODE=ONE_AFTER_FRAME has one left channel and one right channel, and contains TWO samples.
19:17	**Reserved**
16	**DSP_CLK_OVR_ON – DSP second level clocks overrides.** 0 Disables DSP second level clocks override 1 Enables DSP second level clocks override
15:13	**Reserved**
12:8	**DSPCD – DSP controller clock divider**
7	**Reserved**

Bit	Description
6:4	**DSPCS – DSP controller clock select.** 0 REFCLK0 1 PLL1 2 PLL2 3 Crystal OSC 4 ROSC 5 REFCLK1
3:2	**Reserved**
1	**DSPCIV – DSP controller clock invert.** 0 Disables DSP controller clock invert 1 Enables DSP controller clock invert
0	**DSP_CE – DSP controller clock enable.** 0 Disables DSP Controller clock 1 Enables DSP Controller clock

HOST1X_ASYNC_DCCCONFIG_0

Offset: 040h
Read/Write: R/W
Reset: 0000.0000

Display Controller Clock Configuration

Bit	Description
31:17	**Reserved**
16	**DC_CLK_OVR_ON – Display second level clocks overrides.** 0 Disables Display second level clocks override 1 Enables Display second level clocks override
15:7	**Reserved**
6:4	**DCCS – Display controller clock select.** 0 REFCLK0 1 PLL1 2 PLL2 3 Crystal Oscillator 4 ROSC 5 REFCLK1
3:2	**Reserved**
1	**DCCIV – Display controller clock invert.** 0 Disables Display controller clock invert. 1 Enables Display controller clock invert.
0	**DC_CE – Display controller clock enable.** 0 Disables Display controller clock 1 Enables Display controller clock

HOST1X_ASYNC_VICCONFIG_0

Offset: 044h
Read/Write: R/W
Reset: 0000.0000

Video Input Clock Configuration

Register HOST1X_ASYNC_VICCONFIG_0 selects the clock source for the VI clock divider. Different use cases require different register programming, as follows:

When using a camera as the video input source, the output of the clock divider goes out through VGP0 as the camera reference clock. The camera in turns sends a clock signal back, on pin VCLK, to use as the Video Input source clock. Set VCLK_SEL (bit[3] of this register) to 1 in such cases.

When using the Host CPU as the video input source, the video input source clock is the output of the clock divider selected by VI_PIN_OUTPUT_SELECT_0. Set VCLK_SEL (bit[3] of this register) to 0 for this type of use.

Bit	Description
31:17	**Reserved**
16	**VI_CLK_OVR_ON – Video input second level clocks overrides** 0 Disables Video input second level clocks overrides 1 Enable Video input second level clocks overrides
15:13	**Reserved**
12:8	**VICD – Video input clock divider**
7	**Reserved**
6:4	**VICS – Video input clock select** 0 REFCLK0 1 PLL1 2 PLL2 3 Crystal Oscillator 4 ROSC 5 REFCLK1 6 VCLK
3	**VCLK_SEL** 0 = select Video input clock using VICS, VICD and VICIV enum (VCLK_SEL_VICS, VCLK_SEL_PD2VI_CLK) 1 = Select VCLK as Video input clock. VICS and VICD are used to generate reference clock to vgp0 when VCLK is used as Video input clock
2	**Reserved**
1	**VICIV – Video input clock invert** 0 Disables Video input clock invert 1 Enables Video input clock invert
0	**VI_CE – Video input clock enable** 0 Disables Video input clock 1 Enables Video input clock

HOST1X_ASYNC_ISPCCONFIG_0

Offset: 048h
Read/Write: R/W
Reset: 0000.0000

ISP Clock Configuration

Bit	Description
0	**ISP_CE – ISP clock enable** 0 Disables ISP clock 1 Enables ISP Clock

HOST1X_ASYNC_EPPCCONFIG_0

Offset: 04Ch
Read/Write: R/W
Reset: 0000.0000

Encoder Pre-processor Clock Configuration

Bit	Description
31:17	**Reserved**
16	**EPP_CLK_OVR_ON – Encoder Pre-processor second level clocks overrides.** 0 Disables Encoder Pre-processor second level clocks override 1 Enables Encoder Pre-processor second level clocks override
15:13	**Reserved**
12:8	**EPPCD – Encoder Pre-processor clock divider**
7	**Reserved**
6:4	**EPPCS – Encoder Pre-processor clock select.** 0 REFCLK0 1 PLL1 2 PLL2 3 Crystal OSC 4 ROSC 5 REFCLK1
3:2	**Reserved**
1	**EPPCIV – Encoder Pre-processor clock invert.** 0 Disables Encoder Pre-processor clock invert 1 Enables Encoder Pre-processor clock invert
0	**EPP_CE – Encoder Pre-processor clock enable.** 0 Disable Encoder Pre-processor clock 1 Enable Encoder Pre-processor clock

HOST1X_ASYNC_GRMPDCCONFIG_0

Offset: 050h
Read/Write: R/W
Reset: 0000.0000

JPEG/MPEG Decoder Clock Configuration

Bit	Description
31:17	**Reserved**
16	**GRMPD_CLK_OVR_ON – JPEG/MPEG decoder second level clocks overrides** 0 Disables JPEG/MPEG decoder second level clocks override 1 Enables JPEG/MPEG decoder second level clocks override
15:13	**Reserved**
12:8	**GRMPDCD – JPEG/MPEG decoder clock divider**
7	**Reserved**
6:4	**GRMPDCS JPEG/MPEG decoder clock select** 0 REFCLK0 1 PLL1 2 PLL2 3 Internal Crystal OSC 4 ROSC 5 REFCLK1
3:2	**Reserved**
1	**GRMPDCIV – JPEG/MPEG decoder clock invert** 0 Disables **J**PEG/MPEG decoder clock invert 1 Enables JPEG/MPEG decoder clock invert
0	**GRMPD_CE: JPEG/MPEG decoder clock enable** 0 Disables JPEG/MPEG decoder clock 1 Enables JPEG/MPEG decoder clock

HOST1X_ASYNC_JECCONFIG_0

Offset: 054h
Read/Write: R/W
Reset: 0000.0000

JPEG Encoder Clock Configuration

Bit	Description
16	**JE_CLK_OVR_ON – JPEG encoder second level clocks overrides.** 0 Disables JPEG encoder second level clocks overrides 1 -Enables JPEG encoder second level clocks overrides
12:8	**JECD – JPEG encoder clock divider**
6:4	**JECS – JPEG encoder clock select.** 0 REFCLK0 1 PLL1 2 PLL2 3 Crystal Oscillator 4 ROSC 5 REFCLK1

Bit	Description
1	**JECIV – JPEG encoder clock invert.** 0 Disables JPEG encoder clock invert 1 Enables JPEG encoder clock invert
0	**JE_CE – JPEG encoder clock enable.** 0 Disables JPEG encoder clock 1 Enables JPEG encoder clock

HOST1X_ASYNC_MECCONFIG_0

Offset: 058h
Read/Write: R/W
Reset: 0000.0000

MPEG Encoder Clock Configuration

Bit	Description
31:17	**Reserved**
16	**ME_CLK_OVR_ON – MPEG encoder second level clocks overrides.** 0 = Disables MPEG encoder second level clocks overrides 1 = Enables MPEG encoder second level clocks overrides
15:13	**Reserved**
12:8	**MECD – MPEG encoder clock divider**
7	**Reserved**
6:4	**MECS – MPEG encoder clock select** 0 REFCLK0 1 PLL1 2 PLL2 3 COSC 4 ROSC 5 REFCLK1
3:2	**Reserved**
1	**MECIV – MPEG encoder clock invert** 0 Disables MPEG encoder clock invert 1 Enables MPEG encoder clock invert
0	**ME_CE – MPEG encoder clock enable** 0 Disables MPEG encoder clock 1 Enables MPEG encoder clock

HOST1X_ASYNC_AUDIOCCONFIG_0

Offset: 05Ch
Read/Write: R/W
Reset: 0000.0000

Audio Clock Configuration

Bit	Description
31:27	**Reserved**
26:16	**AUDIOCD – Audio clock divider**
15:7	**Reserved**
6:4	**AUDIOCS – Audio clock select.** 0 REFCLK0 1 PLL1 2 PLL2 3 Crystal OSC 4 ROSC 5 REFCLK1
3:2	**Reserved**
1	**AUDIOCIV – Audio clock invert.** 0 Disables Audio clock invert 1 Enables Audio clock invert
0	**AUDIO_CE – Audio clock enable.** 0 Disables Audio clock 1 Enables Audio clock

HOST1X_ASYNC_ICCCONFIG_0

Offset: 060h
Read/Write: R/W
Reset: 0000.0000

I2C Controller Clock Configuration

Bit	Description
31:27	**Reserved**
26:16	**ICCD – I2C controller clock divider**
15:7	**Reserved**
6:4	**ICCS – I2C controller clock select** 0 REFCLK0 1 PLL1 2 PLL2 3 Crystal OSC 4 ROSC 5 REFCLK1
3:2	**Reserved**
1	**ICCIV – I2C controller clock invert** 0 Disables I2C controller clock invert 1 Enables I2C controller clock invert
0	**IC_CE – I2C controller clock enable** 0 Disables I2C controller clock 1 Enables I2C controller clock

HOST1X_ASYNC_ISCCONFIG_0

Offset: 064h
Read/Write: R/W
Reset: 0000.0000

I2S Controller Clock Configuration Register

Bit	Description
28	**I2S_CLK_OVR_ON – I2S second level clocks overrides** 0 Disables I2S second level clocks overrides 1 Enables I2S second level clocks overrides
10	**SCLK_STOPVAL – SCLK stop value (Effective only when SCLK is an output)** 0 = Low 1 = High
9	**SCLK_POL– SCLK polarity (indicates which edge of SCLK is used to latch data or FSYNC input)** 0 no inversion, latch data with rising edge 1 inverted, latch data with falling edge
8	**SCLK_DIR – set SCLK clock direction to input or output** 0 Output 1 Input
6:4	**I2SCS – I2S controller clock select** 0 REFCLK0 1 PLL1 2 PLL2 3 COSC 4 ROSC 5 REFCLK1
3	**I2SCIV – I2S controller clock invert** 0 Disables I2S controller clock invert 1 Enables I2S controller clock invert
2	**I2ST_CE – I2S transmit FIFO clock enable** 0 Disables I2S transmit FIFO clock 1 Enables I2S transmit FIFO clock
1	**I2SR_CE – I2S receive FIFO clock enable** 0 Disables I2S receive FIFO clock 1 Enables I2S receive FIFO clock
0	**I2S_CE – I2S controller source clock enable** 0 Disable I2S controller source clock 1 Enable I2S controller source clock

HOST1X_ASYNC_ISCCONFIG2_0

Offset: 068h
Read/Write: R/W
Reset: 0000.0000

I2S Controller Clock Config2 Register

Bit	Description
31:22	**Reserved**
21:16	**SCLK_DIV_SEL – SCLK Clock Division Select**
15:11	**Reserved**
10:0	**I2SCD – I2S controller clock divider**

HOST1X_ASYNC_SDCCONFIG_0

Offset: 06Ch
Read/Write: R/W
Reset: 0000.0000

Secure Digital Controller Clock Configuration

Bit	Description
31:29	**Reserved**
28	**SD_CLK_OVR_ON – Secure Digital second level clocks overrides** 0 Disable 1 Enable
27	**Reserved**
26:16	**SDCD – Secure Digital controller clock divider**
15:7	**Reserved**
6:4	**SDCS – Secure Digital controller clock select.** 0 REFCLK0 1 PLL1 2 PLL2 3 COSC 4 ROSC 5 REFCLK1
3:2	**Reserved**
1	**SDCIV – Secure Digital controller clock invert.** 0 Disables Secure digital controller clock invert 1 Enables Secure digital controller clock invert
0	**SD_CE – Secure Digital controller clock enable.** 0 Disables Secure digital controller clock 1 Enables Secure digital controller clock

HOST1X_ASYNC_G2CCONFIG_0

Offset: 070h
Read/Write: R/W
Reset: 0000.0000

2D Graphics Engine Clock Configuration

Bit	Description
31:17	**Reserved**
16	**GR2D_CLK_OVR_ON – 2D graphic engine second level clocks overrides** 0 Disables 2D Graphics Engine Second level clocks override 1 Enables 2D Graphics Engine Second level clocks override
15:13	**Reserved**
12:8	**GR2DCD – 2D graphics engine clock divider**
7	**Reserved**
6:4	**GR2DCS – 2D graphics engine clock select** 0 REFCLK0 1 PLL1 2 PLL2 3 COSC 4 ROSC 5 REFCLK1
3:2	**Reserved**
1	**GR2DCIV – 2D graphics engine clock invert** 0 Disables 2D Graphics engine clock invert 1 Enables 2D Graphics engine clock invert
0	**GR2D_CE – 2D graphics engine (GE) clock enable** 0 Disables 2D GE clock enable 1 Enables 2D GE clock enable

HOST1X_ASYNC_G3CCONFIG_0

Offset: 074h
Read/Write: R/W
Reset: 0000.0000

3D graphics engine clock configuration

Bit	Description
31:24	**Reserved**
23	**GR3D_DWCLK_OVR_OFF –** 3D graphic engine nvdwclk override off. 0 Disable 1 Enable
22	**GR3D_ALUCLK_OVR_OFF –** 3d Graphic Engine Nvaluclk Override Off. 0 Disable 1 Enable
21	**GR3D_DFCLK_OVR_OFF – 3D graphic engine nvdfclk override off.** 0 Disable 1 Enable
20	**GR3D_GKCLK_OVR_OFF – 3D graphic engine nvgkclk override off.** 0 Disable 1 Enable
19	**GR3D_RSCLK_OVR_OFF –** 3d Graphic Engine Nvrsclk Override Off. 0 Disable 1 Enable
18	**GR3D_SCLK_OVR_OFF – 3D graphic engine nvsclk override off.** 0 Disable 1 Enable
17	**GR3D_CLK_OVR_ON – 3D graphic engine clocks override on.** 0 Disable 1 Enable
16	**GR3D_SCLK_OVR_ON – 3D graphic engine nvsclk override on.** 0 Disable 1 Enable
15:13	**Reserved**
12:8	**GR3DCD – 3D graphics engine clock divider**
7	**Reserved**
6:4	**GR3DCS – 3D graphics engine clock select.** 0 REFCLK0 1 PLL1 2 PLL2 3 COSC 4 ROSC 5 REFCLK1
3:2	**Reserved**
1	**GR3DCIV – 3D graphics engine clock invert.** 0 Disable 1 Enable
0	**GR3D_CE – 3D graphics engine clock enable.** 0 Disable 1 Enable

HOST1X_ASYNC_MCCCONFIG_0

Offset: 078h
Read/Write: R/W
Reset: 0000.0000

Memory Controller Clock Configuration

Bit	Description
31:13	**Reserved**
12:8	**MCCD – memory controller clock divider**
7	**Reserved**
6:4	**MCCS – memory controller clock select.** 0 REFCLK0 1 PLL1 2 PLL2 3 COSC 4 ROSC 5 REFCLK1
3:2	**Reserved**
1	**MCCIV – memory controller clock invert.** 0 Disable 1 Enable
0	**MC_CE – memory controller clock enable.** 0 Disable 1 Enable

HOST1X_ASYNC_EMCCONFIG_0

Offset: 07Ch
Read/Write: R/W
Reset: 0000.0000

External Memory Clock Configuration

Bit	Description
31:13	**Reserved**
12:8	**EMCCD – External memory clock divider**
7	**Reserved**
6:4	**EMCCS – External memory clock select.** 0 REFCLK0 1 PLL1 2 PLL2 3 Crystal OSC 4 ROSC 5 REFCLK1
3	**Reserved**
2	**EMCCIV – External memory clock invert.** 0 Disable 1 Enable
1	**EMCFST_CE – External memory fast clock enable.** 0 Disable 1 Enable
0	**EMC_CE – External memory clock enable.** 0 Disable 1 Enable

HOST1X_ASYNC_HIDREV_0

Offset: 084h
Read/Write: RO
Reset: 0000.0000

Chip ID Revision Register

Bit	Description
31:20	**Reserved**
19:16	**MINORREV – Chip ID minor revision**
15:8	**CHIPID – Chip ID**
7:4	**MAJORREV – Chip ID major revision**
3:0	**HIDFAM – Chip ID family register**

HOST1X_ASYNC_COREPWRCONFIG_0

Offset: 088h
Read/Write: R/W
Reset: 0000.0000

Core Power Configuration: Since power is controlled externally, software must set this register to tell the chip which partitions have power. The AO partition must always stay on.

Bit	Description
31:3	**Reserved**
2	**SRAM_COREPWR_EN – Power is on in SRAM partition.** 0 Disables Power in SRAM partition 1 Enables power in SRAM partition
1	**VE_COREPWR_EN: Power is on in VE partition.** 0 Disables power in VE partition 1 Enables power in VE partition
0	**TD_COREPWR_EN: Power is on in TD partition.** 0 Disables power in the TD partition 1 Enables power in the TD partition

HOST1X_ASYNC_IOPWRCONFIG_0

Offset: 08Ch
Read/Write: R/W
Reset: 0000.001f

IO Power Configuration: Since power is controlled externally, software must set this register to tell the chip which IOs have power. HOST stays on always.

Bit	Description
31:5	**Reserved**
4	**LCD_IOPWR_DIS – LCD interface IO power disable.** 0 Disable 1 Enable
3	**VI_IOPWR_DIS – Camera interface IO power disable.** 0 Disable 1 Enable
2	**RESERVE_IOPWR_DIS – Reserved IO power disable.** 0 Disable 1 Enable
1	**AUDIO_IOPWR_DIS – AUDIO IO power disable.** 0 Disable 1 Enable
0	**EM_IOPWR_DIS – External Memory IO power disable.** 0 Disable 1 Enable

HOST1X_ASYNC_GPIOIE_0

Offset: 090h
Read/Write: R/W
Reset: 0000.0083

GPIO input enables. Will enable input data from the pads. Potential power savings if these are disabled.

Bit	Description
31:8	**Reserved**
7	**REFCLK1IE – REFCLK1 pin input enable** 0 Disables REFCLK1 pin input 1 Enables REFCLK1 pin input
6	**MHGP6IE – MHGP6 pin input enable** 0 Disables MHGP6 pin input 1 Enables MHGP6 pin input
5	**MHGP5IE – MHGP5 pin input enable** 0 Disables MHGP5 pin input 1 Enables MHGP5 pin input
4	**MHGP4IE – MHGP4 pin input enable** 0 Disables MHGP4 pin input 1 Enables MHGP4 pin input
3	**MHGP3IE – MHGP3 pin input enable** 0 Disables MHGP3 pin input 1 Enables MHGP3 pin input
2	**MHGP2IE – MHGP2 pin input enable** 0 Disables MHGP2 pin input 1 Enables MHGP2 pin input
	MHGP1IE – MHGP1 pin input enable 0 Disables MHGP1 pin input 1 Enables MHGP1 pin input
0	**MHGP0IE – MHGP0 pin input enable** 0 Disables MHGP0 pin input 1 Enables MHGP0 pin input

HOST1X_ASYNC_GPIOID_0

Offset: 094h
Read/Write: RO
Reset: 0000.0000

GPIO input data. Raw data from pads. Only valid if the corresponding input enable is set.

Bit	Description
7	**REFCLK1ID – REFCLK1I pin input data.** 0 Disables REFCLK1 pin input data 1 Enables REFCLK1 pin input data
6	**MHGP6ID – MHGP6 pin input data.** 0 Disables MHGP6 pin input data 1 Enables MHGP6 pin input data
5	**MHGP5ID: MHGP5 pin input data.** 0 Disables MHGP5 pin input data 1 Enables MHGP5 pin input data
4	**MHGP4ID: MHGP4 pin input data.** 0 Disables MHGP4 pin input data 1 Enables MHGP4 pin input data
3	**MHGP3ID: MHGP3 pin input data.** 0 Disables MHGP3 pin input data 1 Enables MHGP3 pin input data
2	**MHGP2ID: MHGP2 pin input data.** 0 Disables MHGP2 pin input data 1 Enables MHGP2 pin input data
1	**MHGP1ID: MHGP1 pin input data.** 0 Disables MHGP1 pin input data 1 Enables MHGP1 pin input data
0	**MHGP0ID: MHGP0 pin input data.** 0 Disables MHGP0 pin input data 1 Enables MHGP0 pin input data

HOST1X_ASYNC_GPIOOE_0

Offset: 098h
Read/Write: R/W
Reset: 0000.0000

GPIO output enables. Specify whether or not to drive the GPIO pins. Setting MHGP4ODS to 'RDY' overrides MHGP4OE; then MHGP4O gets driven only when necessary according to the requested transaction.

Bit	Description
31:8	**Reserved**
7	**REFCLK1OE – REFCLK1 pin output enable.** 0 Disables REFCLK1 pin output 1 Enables REFCLK1 pin output
6	**MHGP6OE – MHGP6 pin output enable.** 0 Disables MHGP6 pin output 1 Enables MHGP6 pin output
5	**MHGP5OE: MHGP5 pin output enable.** 0 Disables MHGP5 pin output 1 Enables MHGP5 pin output
4	**MHGP4OE: MHGP4 pin output enable. Overridden if MHGP4ODS set to 'RDY'.** 0 Disables MHGP4 pin output 1 Enables MHGP4 pin output
3	**MHGP3OE: MHGP3 pin output enable.** 0 Disables MHGP3 pin output 1 Enables MHGP3 pin output
2	**MHGP2OE: MHGP2 pin output enable.** 0 Disables MHGP2 pin output 1 Enables MHGP2 pin output
1	**MHGP1OE: MHGP1 pin output enable.** 0 Disables MHGP1 pin output 1 Enables MHGP1 pin output
0	**MHGP0OE: MHGP0 pin output enable.** 0 Disables MHGP0 pin output 1 Enables MHGP0 pin output

HOST1X_ASYNC_GPIOOD_0

Offset: 09Ch
Read/Write: R/W
Reset: 0000.0000

GPIO output data. These values will be driven out on the GPIO pins only if the output is enabled and the *OD register is selected by *ODS.

Bit	Description
31:8	**Reserved**
7	**REFCLK1OD – REFCLK1 pin output data** 0 Disables REFCLK1 pin output data 1 Enables REFCLK1 pin output data
6	**MHGP6OD: MHGP6 pin output data** 0 Disables MHGP6 pin output data 1 Enables MHGP6 pin output data
5	**MHGP5OD: MHGP5 pin output data** 0 Disables MHGP5 pin output data 1 Enables MHGP5 pin output data
4	**MHGP4OD: MHGP4 pin output data** 0 Disables MHGP4 pin output data 1 Enables MHGP4 pin output data
3	**MHGP3OD: MHGP3 pin output data** 0 Disables MHGP3 pin output data 1 Enables MHGP3 pin output data
2	**MHGP2OD: MHGP2 pin output data** 0 Disables MHGP2 pin output data 1 Enables MHGP2 pin output data
1	**MHGP1OD: MHGP1 pin output data** 0 Disables MHGP1 pin output data 1 Enables MHGP1 pin output data
0	**MHGP0OD: MHGP0 pin output data** 0 Disables MHGP0 pin output data 1 Enables MHGP0 pin output data

HOST1X_ASYNC_GPIOODS_0

Offset: 0A0h
Read/Write: R/W
Reset: 0000.0100

GPIO output data select

Bit	Description
31:18	**Reserved**
17:16	**REFCLK1ODS – REFCLK1 pin output data select** 0 REFCLK1OD 1 TBD1 2 TBD2 3 TBD3
15:14	**CLK_MONITOR_MHGP6SEL: select monitor clock to MHGP6.** 0 PLL2 1 DCFCLK 2 MCCLK 3 DSPCLK
13:12	**CLK_MONITOR_MHGP5SEL: select monitor clock to MHGP5.** 0 PLL1 1 PLL2 2 COSC 3 ROSC
11	**MHGP56ODS – MHGP5, MHGP6 output data select** 0 MHGP5OD, MHGP6OD 1 clk_monitor_MHGP5, clk_monitor_MHGP6
9:8	**MHGP4ODS – MHGP4 pin output data select. RDY by default** 0 MHGP4OD 1 RDY 2 TBD2 3 TBD3
7:6	**MHGP3ODS – MHGP3 pin output data select** 0 MHGP3OD 1 INT 2 TBD2 3 TBD3
5:4	**MHGP2ODS – MHGP2 pin output data select** 0 MHGP2OD 1 TB1 2 TBD2 3 TBD3
3:2	**MHGP1ODS– MHGP1 pin output data select** 0 MHGP1OD 1 TBD1 2 TBD2 3 TBD3
1:0	**MHGP0ODS – MHGP0 pin output data select** 0 MHGP0OD 1 TB1 2 TBD2 3 TBD3

HOST1X_ASYNC_DLYCTRL_0

Offset: 0A4h
Read/Write: R/W
Reset: 0000.0008

Analog delay control register.

Bit	Description
31:5	**Reserved**
4:0	**RDY_DLY – Delay on RDY output**

HOST1X_ASYNC_CLKMNTREN_0

Offset: 0A8h
Read/Write: R/W
Reset: 0000.0000

clk monitor enable

Bit	Description
31:1	**Reserved**
0	**CLK_MONITOR_EN – clock monitor enable.** 0 Disable 1 Enable

HOST1X_ASYNC_INTRCONFIG_0

Offset: 0ACh
Read/Write: R/W
Reset: 0000.0000

Interrupt pin control registers

Bit	Description
31:2	**Reserved**
1	**MHGP3_INTR_MASK: MHGP3 interrupt pin mask.** 0 Disables MHGP3 interrupt pin mask 1 Enables MHGP3 interrupt pin mask
0	**MHGP3_INTR_POL: MHGP3 interrupt pin polarity.** 0 MHGP3 interrupt pin polarity active low 1 MHGP3 interrupt pin polarity active high

HOST1X_ASYNC_INTRMASK_0

Offset: 0B0h
Read/Write: R/W
Reset: 0000.0000

Asynchronous Interrupt Pin Mask Registers

Note: These are all level-high active input interrupts. These interrupts are not sticky and are cleared when the input signals go low.

Bit	Description
31:7	**Reserved**
6	**SDINTR_MASK – asynchronous input interrupt for SD.** (Enables the SD Module to receive an interrupt from the Host CPU.) 0 Disables asynchronous input interrupt for SD 1 Enables asynchronous input interrupt for SD
5	**MHGP6INTR_MASK – asynchronous input interrupt for MHGP6.** 0 Disables asynchronous input interrupt for MHGP6 1 Enables asynchronous input interrupt for MHGP6
4	**MHGP5INTR_MASK: asynchronous input interrupt for MHGP5.** 0 Disables asynchronous input interrupt for MHGP5 1 Enables asynchronous input interrupt for MHGP5
3	**MHGP4INTR_MASK: asynchronous input interrupt for MHGP4.** 0 Disables asynchronous input interrupt for MHGP4 1 Enables asynchronous input interrupt for MHGP4
2	**MHGP2INTR_MASK: asynchronous input interrupt for MHGP2.** 0 Disables asynchronous input interrupt for MHGP2 1 Enables asynchronous input interrupt for MHGP2
1	**MHGP1INTR_MASK: asynchronous input interrupt for MHGP1.** 0 Disables asynchronous input interrupt for MHGP1 1 Enables asynchronous input interrupt for MHGP1
0	**MHGP0INTR_MASK: asynchronous input interrupt for MHGP0.** 0 Disables asynchronous input interrupt for MHGP0 1 Enables asynchronous input interrupt for MHGP0

HOST1X_ASYNC_EMCPADEN_0

Offset: 0B4h
Read/Write: R/W
Reset: 0000.0003

EMC pad enable registers

Bit	Description
31:3	**Reserved**
2	**EMC_PAD_CKE: CKE output enable.** 0 Enables Normal output 1 Enables Tristate output
1	**EMC_PAD_INPUT_EN: inputs enable for EMC pads.** 0 Disables inputs for EMC pads 1 Enables inputs for EMC pads
0	**EMC_PAD_OUTPUT_EN: outputs enable for EMC pads.** 0 Disables output for EMC pads 1 Enables output for EMC pads

HOST1X_ASYNC_HOSTPADCTRL_0

Offset: 0B8h
Read/Write: R/W
Reset: 0003.3050

Host Pad Control Register

Bit	Description
31:26	**Reserved**
25	**HVDD_E_PWR_DOWN –** Used to latch the hvdd_e_33v_det value and power down the HVDD pad.
24	**HOST_33V_FORCE –** select host 33V control signal from pad or register
23:18	**Reserved**
17:16	**HOST_CTRL_LPMD –** host control pins low power mode select
15:14	**Reserved**
13:12	**HOST_DAT_LPMD –** host data pins low power mode select
11	**Reserved**
10	**HOST_CTRL_SCHMT_EN –** host control pins schmidt enable. 0 Disables Host Control pins schmidt 1 Enables Host Control pins schmidt
9	**Reserved**
8	**HOST_DAT_SCHMT_EN –** host data pins schmidt enable. 0 Disables Host data pins schmidt 1 Enables Host data pins schmidt
7	**Reserved**
6	**HOST_CTRL_HSM_EN –** Host control pins high speed mode enable. 0 Disables Host Control pins high speed mode 1 Enables Host Control pins high speed mode
5	**Reserved**
4	**HOST_DAT_HSM_EN –** host data pins high speed mode enable. 0 Disables Host data pins high speed mode 1 Enables Host data pins high speed mode
3	**Reserved**
2	**HOST_OD_EN –** Host over drain enable **E**nabled only when MHGP3 is used as active low interrupt. 0 Disables Host over drain 1 Enables Host over drain -- MHGP3 must be an active low interrupt
1	**Reserved**
0	**HOST_33V_EN –** Host 3.3V mode enable 0 Disables Host 3.3 V mode 1 Enables Host 3.3 V mode

HOST1X_ASYNC_HOSTPADCAL1_0

Offset: 0BCh
Read/Write: R/W
Reset: 0000.7f7f

Host Pad Calibration Register 1

Bit	Description
31:15	**Reserved**
14:13	**HOST_CAL_DRVUP_SLWF**
12:8	**HOST_CAL_DRVUP**
7	**Reserved**
6:5	**HOST_CAL_DRVDN_SLWR**
4:0	**HOST_CAL_DRVDN**

HOST1X_ASYNC_EMCPADCTRL_0

Offset: 0C0h
Read/Write: R/W
Reset: 000f.f0f1

EMC Pad control register

Bit	Description
31:21	**Reserved**
20	**MEM_PULLD –** EMC pull-down enable 0 Disables EMC pull down 1 Enables EMC pull down
19:18	**MEM1_CTRL_LPMD –** EMC control pins low power mode select
17:16	**MEM0_CTRL_LPMD –** EMC control pins low power mode select
15:14	**MEM1_DAT_LPMD –** EMC data pins low power mode select
13:12	**MEM0_DAT_LPMD –** EMC data pins low power mode select
11:10	**Reserved**
9	**MEM1_DAT_SCHMT_EN –** EMC data pins schmidt enable 0 Disables EMC data pins schmidt 1 Enables EMC data pins schmidt
8	**MEM0_DAT_SCHMT_EN –** EMC data pins schmidt enable 0 Disables EMC data pins schmidt 1 Enables EMC data pins schmidt
7	**MEM1_CTRL_HSM_EN –** EMC control pins high speed mode enable 0 Disables EMC control pins high speed mode 1 Enables EMC control pins high speed mode
6	**MEM0_CTRL_HSM_EN –** EMC control pins high speed mode enable 0 Disables EMC control pins high speed mode 1 Enables EMC control pins high speed mode
5	**MEM1_DAT_HSM_EN –** EMC data pins high speed mode enable 0 Disables EMC data pins high speed mode 1 Enables EMC data pins high speed mode
4	**MEM0_DAT_HSM_EN –** EMC data pins high speed mode enable 0 Disables EMC data pins high speed mode 1 Enables EMC data pins high speed mode
3:2	**Reserved**

Bit	Description
1	**MEM_VREF_EN –** EMC vref enable 0 Disables EMC vref enable 1 Enables EMC vref enable
0	**MEM_33V_EN –** EMC 3.3V mode enable 0 Disables EMC 3.3 V mode 1 Enables EMC 3.3 V mode

HOST1X_ASYNC_MEMPADCAL1_0

Offset: 0C4h
Read/Write: R/W
Reset: 0000.7f7f

EMC pad calibration register 1

Bit	Description
31:15	**Reserved**
14:13	**MEM_CAL_DRVUP_SLWF**
12:8	**MEM_CAL_DRVUP**
7	**Reserved**
6:5	**MEM_CAL_DRVDN_SLWR**
4:0	**MEM_CAL_DRVDN**

HOST1X_ASYNC_LCDPADCTRL_0

Offset: 0C8h
Read/Write: R/W
Reset: 000f.f0f1

LCD Pad Control Register

Bit	Description
31:20	**Reserved**
19:18	**LCD1_CTRL_LPMD –** LCD control pins low power mode select
17:16	**LCD0_CTRL_LPMD –** LCD control pins low power mode select
15:14	**LCD1_DAT_LPMD –** LCD data pins low power mode select
13:12	**LCD0_DAT_LPMD –** LCD data pins low power mode select
11	**Reserved**
10	**LCD_CTRL_SCHMT_EN –** LCD control pins schmidt enable 0 Disables LCD control pins schmidt trigger. 1 Enables LCD control pins schmidt trigger
9	**Reserved**
8	**LCD_DAT_SCHMT_EN –** LCD data pins schmidt enable**.** 0 Disables LCD data pins schmidt trigger. 1 Enables LCD data pins schmidt trigger
7	**LCD1_CTRL_HSM_EN –** LCD control pins high speed mode enable**.** 0 Disables LCD1 control pins schmidt trigger. 1 Enables LCD1 control pins schmidt trigger
6	**LCD0_CTRL_HSM_EN – LCD control pins high speed mode enable.** 0 Disables LCD0 control pins schmidt trigger. 1 Enables LCD0 control pins schmidt trigger

Bit	Description
5	**LCD1_DAT_HSM_EN –** LCD data pins high speed mode enable**.** 0 Disables LCD1 data pins high speed mode. 1 Enables LCD1 data pins high speed mode
4	**LCD0_DAT_HSM_EN –** LCD data pins high speed mode enable. 0 Disables LCD0 data pins high speed mode. 1 Enables LCD0 data pins high speed mode.
3:1	**Reserved**
0	**LCD_33V_EN –** LCD 3.3V mode enable. 0 Disables LCD 3.3 V mode. 1 Enables LCD 3.3 V mode.

HOST1X_ASYNC_LCDPADCAL1_0

Offset: 0CCh
Read/Write: R/W
Reset: 0000.7f7f

LCD pad calibration register 1

Bit	Description
31:15	**Reserved**
14:13	**LCD_CAL_DRVUP_SLWF**
12:8	**LCD_CAL_DRVUP**
7	**Reserved**
6:5	**LCD_CAL_DRVDN_SLWR**
4:0	**LCD_CAL_DRVDN**

HOST1X_ASYNC_VIPADCTRL_0

Offset: 0D0h
Read/Write: R/W
Reset: 0003.3051

VI Pad control register

Bit	Description
31:18	**Reserved**
17:16	**VI_CTRL_LPMD –** VI control pins low power mode select
13:12	**VI_DAT_LPMD –** VI data pins low power mode select
11	**Reserved**
10	**VI_CTRL_SCHMT_EN –** VI control pins schmidt enable**.** 0 Disables VI control pins schmidt trigger 1 Enables VI control pins schmidt trigger
9	**Reserved**
8	**VI_DAT_SCHMT_EN –** VI data pins schmidt enable**.** 0 Disables VI data pins schmidt trigger 1 Enables VI data pins schmidt trigger
7	**Reserved**
6	**VI_CTRL_HSM_EN –** VI control pins high speed mode enable**.** 0 Disables VI control pins high speed mode 1 Enables VI control pins high speed mode
5	**Reserved**

Bit	Description
4	**VI_DAT_HSM_EN –** VI data pins high speed mode enable**.** 0 Disables VI data pins high speed mode 1 Enables VI data pins high speed mode
3:1	**Reserved**
0	**VI_33V_EN –** VI 3.3V mode enable**.** 0 Disables VI 3.3 V mode 1 Enables VI 3.3 V mode

HOST1X_ASYNC_VIPADCAL1_0

Offset: 0d4h
Read/Write: R/W
Reset: 0000.7f7f

VI Pad Calibration Register 1

Bit	Description
31:15	**Reserved**
14:13	**VI_CAL_DRVUP_SLWF**
12:8	**VI_CAL_DRVUP**
7	**Reserved**
6:5	**VI_CAL_DRVDN_SLWR**
4:0	**VI_CAL_DRVDN**

HOST1X_ASYNC_SDPADCTRL_0

Offset: 0D8h
Read/Write: R/W
Reset: 0003.3051

SD Pad control register

Bit	Description
31:24	**Reserved**
23	**SD_CLK_PULLU –** SD clock pin pull-up enable 0 Disables SD clock pin pull-up 1 Enables SD clock pin pull-up
22	**SD_PULLU –** SD data pin (except data pin3) pull-up enable 0 Disables SD data pin pull-up (Does not apply to data pin 3.) 1 Enables SD data pin pull-up (Does not apply to data pin 3.)
21	**SD_SDIO3_PULLU –** SD data pin 3 pull-up enable 0 Disables SD data pin 3 pull-up 1 Enables SD data pin 3 pull-up
20	**SD_SDIO3_PULLD –** SD data pin 3 pull-down enable 0 Disables SD data pin 3 pull-down 1 Enables SD data pin 3 pull-down
19:18	**Reserved**
17:16	**SD_CTRL_LPMD –** SD control pins low power mode select
15:14	**Reserved**
13:12	**SD_DAT_LPMD –** SD data pins low power mode select
11	**Reserved**
10	**SD_CTRL_SCHMT_EN –** SD control pins schmidt enable 0 Disables SD control pin schmidt trigger 1 Enables SD control pin schmidt trigger
9	**Reserved**
8	**SD_DAT_SCHMT_EN –** SD data pins schmidt enable 0 Disables SD data pin schmidt trigger 1 Enables SD data pin schmidt trigger
7	**Reserved**
6	**SD_CTRL_HSM_EN –** SD control pins high speed mode enable 0 Disables SD control pin high speed mode 1 Enables SD control pin high speed mode
5	**Reserved**
4	**SD_DAT_HSM_EN –** SD data pins high speed mode enable 0 Disables SD data pin high speed mode 1 Enables SD data pin high speed mode
3:1	**Reserved**
0	**SD_33V_EN –** SD 3.3V mode enable 0 Disables SD 3.3 V mode 1 Enables SD 3.3 V mode

HOST1X_ASYNC_SDPADCAL1_0

Offset: 0DCh
Read/Write: R/W
Reset: 0000.7f7f

SD Pad Calibration Register 1

Bit	Description
31:15	**Reserved**
14:13	**SD_CAL_DRVUP_SLWF**
12:8	**SD_CAL_DRVUP**
7	**Reserved**
6:5	**SD_CAL_DRVDN_SLWR**
4:0	**SD_CAL_DRVDN**

HOST1X_ASYNC_AUDIOPADCTRL_0

Offset: 0E0h
Read/Write: R/W
Reset: 0003.3051

AUDIO Pad Control Register

Bit	Description
31:18	**Reserved**
17:16	**AUDIO_CTRL_LPMD –** AUDIO control pins low power mode select
15:14	**Reserved**
13:12	**AUDIO_DAT_LPMD –** AUDIO data pins low power mode select
11	**Reserved**
10	**AUDIO_CTRL_SCHMT_EN –** AUDIO control pins schmidt enable 0 Disables Audio control pins schmidt trigger. 1 Enables Audio control pins schmidt trigger
9	**Reserved**
8	**AUDIO_DAT_SCHMT_EN –** AUDIO data pins schmidt enable 0 Disables Audio data pins schmidt trigger. 1 Enables Audio data pins schmidt trigger
7	**Reserved**
6	**AUDIO_CTRL_HSM_EN –** AUDIO control pins high speed mode enable 0 Disables Audio control pins high speed mode. 1 Enables Audio control pins high speed mode
5	**Reserved**
4	**AUDIO_DAT_HSM_EN:** AUDIO data pins high speed mode enable 0 Disables Audio data pins high speed mode. 1 Enables Audio data pins high speed mode
3:1	**Reserved**
0	**AUDIO_33V_EN:** AUDIO 3.3V mode enable 0 Disables Audio 3.3 V mode. 1 Enables Audio 3.3 V mode.

HOST1X_ASYNC_AUDIOPADCAL1_0

Offset: 0e4h
Read/Write: R/W
Reset: 0000.7f7f

AUDIO pad calibration register 1

Bit	Description
31:15	**Reserved**
14:13	**AUDIO_CAL_DRVUP_SLWF**
12:8	**AUDIO_CAL_DRVUP**
7	**Reserved**
6:5	**AUDIO_CAL_DRVDN_SLWR**
4:0	**AUDIO_CAL_DRVDN**

HOST1X_ASYNC_I2CPADCTRL_0

Offset: 0E8h
Read/Write: R/W
Reset: 0000.0000

I2C Pull-up/pull-down Control Register

Bit	Description
3	**I2C_CLK_PULLU –** I2C clock pin pull-up enable 0 Disables I2C Clock pin pull-up 1 Enables I2C Clock pin pull-up
2	**I2C_CLK_PULLD –** I2C clock pin pull-down enable 0 Disables I2C Clock pin pull-down 1 Enables I2C Clock pin pull-down
1	**I2C_D_PULLU –** I2C data pin pull-up enable 0 Disables I2C data pin pull-up 1 Enables I2C data pin pull-up
0	**I2C_D_PULLD –** I2C data pin pull-down enable 0 Disables I2C data pin pull-down 1 Enables I2C data pin pull-down

HOST1X_ASYNC_JTPADCTRL_0

Offset: 0ECh
Read/Write: R/W
Reset: 0000.0000

JTAG Pad control register

Bit	Description
31:1	**Reserved**
0	**JT_33V_EN –** JTAG 3.3V mode enable 0 Disables 3.3 V mode 1 Enables 3.3 V mode

HOST1X_ASYNC_OBSCTRL_0

Offset: 0F8h
Read/Write: R/W
Reset: 0000.0000

Observation Port Control

Bit	Description
31	**OBS_EN –** Observation bus enable 0 Disable Observation bus 1 Enable Observation bus
26:16	**OBS_MOD_SEL –** Module select to determine which module will transmit debug data. Includes partition select.
15:0	**OBS_SIG_SEL –** Module-level mux select for determining which debug signals to send out

HOST1X_ASYNC_OBSDATA_0

Offset: 0FCh
Read/Write: RO
Reset: 0000.0000

Observation port data. This should be the same data that is going out on the observation bus.

Bit	Description
31:0	**OBS_DATA –** Module-level mux select for determining which debug signals to send out

HOST1X_CHANNEL_FIFOSTAT_0

Offset: 000h
Read/Write: RO
Reset: 0000.0000

CFNUMEMPTY is the number of free slots available in the per-channel command FIFO (needed for PIO or polling for completion of a wait).

Bit	Description
31	**INDRDY: Indicates that INDCOUNT==0, so it should be OK to issue another read**
28:24	**OUTFENTRIES: Number of entries available for reading in this channel's output FIFO**
20:16	**REGFNUMEMPTY: Register write/read FIFO free count**
11	**CFGATHER: Indicates whether GATHER is active.** If a GATHER command issued via PIO, software must wait for the GATHER to be IDLE before issuing another command. 0 = Idle 1 = Busy
10	**CFEMPTY: Indicates whether the command FIFO is empty or not.** 0 = Not empty 1 = Empty
9:0	**CFNUMEMPTY: Command FIFO free count**

HOST1X_CHANNEL_INDOFF_0

Offset: 004h
Read/Write: R/W
Reset: 0000.0000

This register (along with INDCNT and INDDATA) is used to indirectly read write modules outside the host. If AUTOINC is set, INDOFFSET increments by 4 on every access of INDDATA. REGFNUMEMPTY is polled to determine when valid data can be read from INDDATA.

Indirect frame buffer write is STRONGLY DISCOURAGED. There are better ways to write to memory (direct and through the channel memory map) and there is limited flow control in the host. It's very easy to get into trouble with indirect frame buffer write.

Bit	Description
31	**AUTOINC: auto increment of read/write address.** 0 = Disable 1 = Enable
30	**ACCTYPE: access type: indirect register or indirect frame buffer.** 0 = Reg 1 = Fb
29	**BUF32B: buffer up 32 bits of register data before sending it.** Otherwise, register writes will be sent as soon as they are received. Does not support byte writes in 16-bit host. Does not affect frame buffer writes. 0 = Nobuf 1 = Buf
28:27	**INDSWAP: Indirect frame buffer access swap control.** 0 = None 1 = Byte16 2 = Byte32 3 = Word32
25:18	**INDMODID: ACCTYPE=REG: register module ID.** 0 = HOST1X 1 = GRMPD 2 = ME 3 = JPEGE 4 = VI 5 = EPP 6 = ISP 7 = DSP 8 = GR2D 9 = GR3D 10 = DISPLAY 11 = I2S 12 = IC 13 = SD 14 = MC 15 = EMC
25:2	**INDOFFSET: ACCTYPE=FB: frame buffer address**
17:2	**INDROFFSET: ACCTYPE=REG: register offset ([15:0])**

HOST1X_CHANNEL_INDCNT_0

Offset: 008h
Read/Write: R/W
Reset: 0000.0000

Indirect register access count

Used to trigger indirect reads. Holds the number of registers/memory locations that will be read out. Channels should not request more than there is space available in their output FIFO. Only the protected channel should make liberal use of this feature for speeding up context switching.

For indirect frame buffer reads, each channel cannot issue more than NV_HOST1X_MAX_IND_FB_READS at once. The read data must return and be written into the per-channel output FIFO before any additional reads can be issued.

Bit	Description
15:0	**INDCOUNT**

HOST1X_CHANNEL_INDDATA_0

Offset: 00ch
Read/Write: R/W
Reset: 0000.0000

This register, when written, writes to the data to the INDOFFSET in INDOFF.

For reads, a REGFNUMEMPTY number of 32-bit values can be read before needing to poll FIFOSTAT again. The per-channel output FIFO (OUTFENTRIES) is readable via this offset. A read of INDDATA will pop an entry off of the per-channel output FIFO.

Bit	Description
31:0	**INDDATA: read or write data**

HOST1X_CHANNEL_RAISE_0

Offset: 010h
Read/Write: RO
Reset: 0000.0000

The general-purpose channels have DMA and RAISE/REFCOUNT functionality

Any raise values returned from a client module are converted to vectors and update the per-channel raise register

Bit	Description
31:0	**RAISE: This channel's RAISE vector**

HOST1X_CHANNEL_REFCNT_0

Offset: 014h
Read/Write: RO
Reset: 0000.0000

Any refcnt value returned from a client module (triggered as part of a method sent to the module) updates this per-channel register.

Bit	Description
15:0	**REFCNT: This channel's REFCNT value**

HOST1X_CHANNEL_DMASTART_0

Offset: 018h
Read/Write: R/W
Reset: 0000.0000

This register triggers a DMA fetch from the FB for this channel, if the Put register doesn't equal the DMA Get register. This value must be 16 byte aligned.

Bit	Description
25:2	**DMASTART: cmdbuf FB offset**

HOST1X_CHANNEL_DMAPUT_0

Offset: 01ch
Read/Write: R/W
Reset: 0000.0000

This register triggers a DMA fetch from the FB for this channel, if the PUT register doesn't equal the GET register. This address is relative to the DMASTART base address. Does not support byte writes. All 4-byte data need to be programmed.

Bit	Description
25:2	**DMAPUT: cmdbuf FB offset**

HOST1X_CHANNEL_DMAGET_0

Offset: 020h
Read/Write: RO
Reset: 0000.0000

This register tracks the FB offset the DMA engine has read up to (incremented as entries are loaded from the channels command buffer into the FIFO). This address is relative to the DMASTART base address.

Bit	Description
25:2	**DMAGET: cmdbuf FB offset**

HOST1X_CHANNEL_DMAEND_0

Offset: 024h
Read/Write: R/W
Reset: 0000.0000

The boundary of illegal addresses (either end of pushbuffer or end of physical memory). This is designed to prevent DMA from prefetching illegal addresses. If DMA reaches this address before seeing a RESTART, it will stop. This would be a software error condition. This value must be 16 byte aligned.

Bit	Description
25:2	**DMAEND: cmdbuf FB offset**

HOST1X_CHANNEL_DMACTRL_0

Offset: 028h
Read/Write: R/W
Reset: 0000.0001

DMA control register

Bit	Description
1	**DMAGETRST: Reset GET pointer to '0'.** Useful for cleaning up crashed channels. 0 = Disable 1 = Enable
0	**DMASTOP: Stop DMA from fetching on this channel.** **NOTE:** A Command DMA channel needs to be enabled for PIO-gather to work. 0 = Run 1 = Stop

HOST1X_CHANNEL_FBBUFBASE_0

Offset: 02ch
Read/Write: R/W
Reset: 0000.0000

Frame buffer- buffered region base Holds the start address of this channel's buffered region. Any writes to HOST1X_CHANNEL_FBBUF_REGION will write into memory relative to this address. For example, if this register is set to 0x1000, a write to (HOST1X_CHANNEL_FBBUF_REGION_BASE + 0x800) will produce a memory write to 0x1800. Writes to this region will be buffered up to 128 bits before being sent to the memory controller.

Bit	Description
25:2	**FBBUFBASE**

HOST1X_CHANNEL_CMDSWAP_0

Offset: 030h
Read/Write: R/W
Reset: 0000.0000

Command swap control. Affects swapping on writes to the PIO region and the frame buffer buffered memory write region.

HOST1X_CHANNEL_FIFOSTAT_0

Offset: 000h
Read/Write: RO
Reset: 0000.0000

CFNUMEMPTY is the number of free slots available in the per-channel command FIFO (needed for PIO or polling for completion of a wait).

Bit	Description
31	**INDRDY: Indicates that INDCOUNT==0, so it should be OK to issue another read**
28:24	**OUTFENTRIES: Number of entries available for reading in this channel's output FIFO**
20:16	**REGFNUMEMPTY: Register write/read FIFO free count**
11	**CFGATHER: Indicates whether GATHER is active.** If a GATHER command issued via PIO, software must wait for the GATHER to be IDLE before issuing another command. 0 = Idle 1 = Busy
10	**CFEMPTY: Indicates whether the command FIFO is empty or not.** 0 = Not empty 1 = Empty
9:0	**CFNUMEMPTY: Command FIFO free count**

HOST1X_CHANNEL_INDOFF_0

Offset: 004h
Read/Write: R/W
Reset: 0000.0000

This register (along with INDCNT and INDDATA) is used to indirectly read write modules outside the host. If AUTOINC is set, INDOFFSET increments by 4 on every access of INDDATA. REGFNUMEMPTY is polled to determine when valid data can be read from INDDATA.

Indirect frame buffer write is STRONGLY DISCOURAGED. There are better ways to write to memory (direct and through the channel memory map) and there is limited flow control in the host. It's very easy to get into trouble with indirect frame buffer write.

Bit	Description
31	**AUTOINC: auto increment of read/write address.** 0 = Disable 1 = Enable
30	**ACCTYPE: access type: indirect register or indirect frame buffer.** 0 = REG 1 = FB
29	**BUF32B: buffer up 32 bits of register data before sending it.** Otherwise, register writes will be sent as soon as they are received. Does not support byte writes in 16-bit host. Does not affect frame buffer writes. 0 = No buf 1 = Buf
28:27	**INDSWAP: Indirect frame buffer access swap control.** 0 = None 1 = Byte16 2 = Byte32 3 = Word32
25:18	**INDMODID: ACCTYPE=REG: register module ID.** 0 = HOST1X 1 = GRMPD 2 = ME 3 = JPEGE 4 = VI 5 = EPP 6 = ISP 7 = DSP 8 = GR2D 9 = GR3D 10 = DISPLAY 11 = I2S 12 = IC 13 = SD 14 = MC 15 = EMC
25:2	**INDOFFSET: ACCTYPE=FB: frame buffer address**
17:2	**INDROFFSET: ACCTYPE=REG: register offset ([15:0])**

HOST1X_CHANNEL_INDCNT_0

Offset: 008h
Read/Write: R/W
Reset: 0000.0000

Indirect register access count

Used to trigger indirect reads. Holds the number of registers/memory locations that will be read out. Channels should not request more than there is space available in their output FIFO. Only the protected channel should make liberal use of this feature for speeding up context switching.

For indirect frame buffer reads, each channel cannot issue more than NV_HOST1X_MAX_IND_FB_READS at once. The read data must return and be written into the per-channel output FIFO before any additional reads can be issued.

Bit	Description
15:0	**INDCOUNT**

HOST1X_CHANNEL_INDDATA_0

Offset: 00ch
Read/Write: R/W
Reset: 0000.0000

This register, when written, writes to the data to the INDOFFSET in INDOFF.

For reads, a REGFNUMEMPTY number of 32-bit values can be read before needing to poll FIFOSTAT again. The per-channel output FIFO (OUTFENTRIES) is readable via this offset. A read of INDDATA will pop an entry off of the per-channel output FIFO.

Bit	Description
31:0	**INDDATA: read or write data**

HOST1X_SYNC_INTSTATUS_0

Offset: 000h
Read/Write: RO
Reset: 0000.0000

HOST CONTROL and STATUS REGISTERS

INTSTATUS - interrupt status

Contains the interrupt status for all of the client modules. These status bits are only status. Writing '1' to them will not clear them. SW must clear the interrupt in the appropriate module, which should be reflected here.

Any of the GPIO pins, MHGP[6:4] and MHGP[2:0], when configured as interrupts, utilize the interrupt signal INT_ on MHGP3 to output an interrupt from the GoForce 5500 to the Host CPU. MHGP3 contains all the interrupts on the MHGPx GPIOs, the interrupt from the SDIO Module, and the internal module interrupts. The bit in the register below tells which interrupts are pending, or not, on pin MHGP3.

Bit	Description
15	**EMC_INT:** 0 = Not_pending 1 = Pending
14	**MC_INT:** 0 = Not_pending 1 = Pending
13	**SD_INT:** 0 = Not_pending 1 = Pending
12	**IC_INT:** 0 = Not_pending 1 = Pending
11	**I2S_INT:** 0 = Not_pending 1 = Pending
10	**DISPLAY_INT:** 0 = Not_pending 1 = Pending
9	**GR3D_INT:** 0 = Not_pending 1 = Pending
8	**GR2D_INT:** 0 = Not_pending 1 = Pending
7	**DSP_INT:** 0 = Not_pending 1 = Pending
6	**ISP_INT:** 0 = Not_pending 1 = Pending
5	**EPP_INT:** 0 = Not_pending 1 = Pending
4	**VI_INT:** 0 = Not_pending 1 = Pending

Bit	Description
3	**JPEGE_INT:** 0 = Not_pending 1 = Pending
2	**ME_INT:** 0 = Not_pending 1 = Pending
1	**GRMPD_INT:** 0 = Not_pending 1 = Pending
0	**HOST_INT:** 0 = Not_pending 1 = Pending

HOST1X_SYNC_INTMASK_0

Offset: 004h
Read/Write: R/W
Reset: 0000.0000

Contains a master interrupt mask for all interrupt signals. If the interface's MASK_ALL bit is disabled, no interrupts will be triggered on that interface.

Bit	Description
1	**DSP_INT_MASK_ALL:** 0 = Disable 1 = Enable
0	**CPU_INT_MASK_ALL:** 0 = Disable 1 = Enable

HOST1X_SYNC_INTCMASK_0

Offset: 008h
Read/Write: R/W
Reset: 0000.0000

INTCMASK - interrupt mask for CPU

Contains the interrupt mask bits for all of the client modules. If the INT_MASK is ENABLED for a particular module, the module's INT bit from INTSTATUS contributes to the global interrupt signal (host2pad_mhgp3) to interrupt the Host CPU.

Bit	Description
15	**EMC_INT_CMASK:** 0 = Disable 1 = Enable
14	**MC_INT_CMASK:** 0 = Disable 1 = Enable
13	**SD_INT_CMASK:** 0 = Disable 1 = Enable

Bit	Description
12	**IC_INT_CMASK:** 0 = Disable 1 = Enable
11	**I2S_INT_CMASK:** 0 = Disable 1 = Enable
10	**DISPLAY_INT_CMASK:** 0 = Disable 1 = Enable
9	**GR3D_INT_CMASK:** 0 = Disable 1 = Enable
8	**GR2D_INT_CMASK:** 0 = Disable 1 = Enable
7	**DSP_INT_CMASK:** 0 = Disable 1 = Enable
6	**ISP_INT_CMASK:** 0 = Disable 1 = Enable
5	**EPP_INT_CMASK:** 0 = Disable 1 = Enable
4	**VI_INT_CMASK:** 0 = Disable 1 = Enable
3	**JPEGE_INT_CMASK:** 0 = Disable 1 = Enable
2	**ME_INT_CMASK:** 0 = DISABLE 1 = ENABLE
1	**GRMPD_INT_CMASK:** 0 = Disable 1 = Enable
0	**HOST_INT_CMASK:** 0 = Disable 1 = Enable

HOST1X_SYNC_INTDMASK_0

Offset: 00ch
Read/Write: R/W
Reset: 0000.0000

INTDMASK - interrupt mask for DSP

Contains the interrupt mask bits for all of the client modules. If the INT_MASK is ENABLED for a particular module, the modules INT bit from INTSTATUS contributes to the global interrupt signal (host2pad_mhgp3)

Bit	Description
15	**EMC_INT_DMASK:** 0 = Disable 1 = Enable
14	**MC_INT_DMASK:** 0 = Disable 1 = Enable
13	**SD_INT_DMASK:** 0 = Disable 1 = Enable
12	**IC_INT_DMASK:** 0 = Disable 1 = Enable
11	**I2S_INT_DMASK:** 0 = Disable 1 = Enable
10	**DISPLAY_INT_DMASK:** 0 = Disable 1 = Enable
9	**GR3D_INT_DMASK:** 0 = Disable 1 = Enable
8	**GR2D_INT_DMASK:** 0 = Disable 1 = Enable
7	**DSP_INT_DMASK:** 0 = Disable 1 = Enable
6	**ISP_INT_DMASK:** 0 = Disable 1 = Enable
5	**EPP_INT_DMASK:** 0 = Disable 1 = Enable
4	**VI_INT_DMASK:** 0 = Disable 1 = Enable
3	**JPEGE_INT_DMASK:** 0 = Disable 1 = Enable
2	**ME_INT_DMASK:** 0 = Disable 1 = Enable

Bit	Description
1	**GRMPD_INT_DMASK:** 0 = Disable 1 = Enable
0	**HOST_INT_DMASK:** 0 = Disable 1 = Enable

HOST1X_SYNC_HINTSTATUS_0

Offset: 010h
Read/Write: RO
Reset: 0000.0000

Host interrupt status

Contains the interrupt status for the various host interrupts. The status is sticky and is PENDING until cleared (write 1's to INTSTATUS to clear).

Bit	Description
29	**CSW_HOST1XW2MC_INT: Host write client FIFO has filled up**
28	**RDMA_DATABUF_THOLD_INT3: Read DMA data fifo in port3 reached high level watermark.** 0 = Not_pending 1 = Pending
27	**RDMA_BUF_THOLD_INT3: Buffer threshold reached in read DMA port3.** 0 = Not_pending 1 = Pending
26	**RDMA_BUF_OFLOW_INT3: Buffer overflow in read DMA port3.** 0 = Not_pending 1 = Pending
25	**RDMA_DATABUF_THOLD_INT2: Read DMA data fifo in port2 reached high level watermark.** 0 = Not_pending 1 = Pending
24	**RDMA_BUF_THOLD_INT2: Buffer threshold reached in read DMA port2.** 0 = Not_pending 1 = Pending
23	**RDMA_BUF_OFLOW_INT2: Buffer overflow in read DMA port2.** 0 = Not_pending 1 = Pending
22	**RDMA_DATABUF_THOLD_INT1: Read DMA data fifo in port1 reached high level watermark.** 0 = Not_pending 1 = Pending
21	**RDMA_BUF_THOLD_INT1: Buffer threshold reached in read DMA port1.** 0 = Not_pending 1 = Pending
20	**RDMA_BUF_OFLOW_INT1: Buffer overflow in read DMA port1.** 0 = Not_pending 1 = Pending
19	**RDMA_DATABUF_THOLD_INT0: Read DMA data fifo in port0 reached high level watermark.** 0 = Not_pending 1 = Pending

Bit	Description
18	**RDMA_BUF_THOLD_INT0: Buffer threshold reached in read DMA port0.** 0 = Not_pending 1 = Pending
17	**RDMA_BUF_OFLOW_INT0: Buffer overflow in read DMA port0.** 0 = Not_pending 1 = Pending
16	**RDMA_INVAL_CLREQ_INT: Invalid client request to read DMA.** 0 = Not_pending 1 = Pending
15	**CTXSW_INT7: Context switch is blocked on channel 7.** 0 = Not_pending 1 = Pending
14	**CTXSW_INT6: Context switch is blocked on channel 6.** 0 = Not_pending 1 = Pending
13	**CTXSW_INT5: Context switch is blocked on channel 5.** 0 = Not_pending 1 = Pending
12	**CTXSW_INT4: Context switch is blocked on channel 4.** 0 = Not_pending 1 = Pending
11	**CTXSW_INT3: Context switch is blocked on channel 3.** 0 = Not_pending 1 = Pending
10	**CTXSW_INT2: Context switch is blocked on channel 2.** 0 = Not_pending 1 = Pending
9	**CTXSW_INT1: Context switch is blocked on channel 1.** 0 = Not_pending 1 = Pending
8	**CTXSW_INT0: Context switch is blocked on channel 0.** 0 = Not_pending 1 = Pending
7	**WAIT_INT7: WAIT has completed on channel 7.** 0 = Not_pending 1 = Pending
6	**WAIT_INT6: WAIT has completed on channel 6.** 0 = Not_pending 1 = Pending
5	**WAIT_INT5: WAIT has completed on channel 5.** 0 = Not_pending 1 = Pending
4	**WAIT_INT4: WAIT has completed on channel 4.** 0 = Not_pending 1 = Pending
3	**WAIT_INT3: WAIT has completed on channel 3.** 0 = Not_pending 1 = Pending
2	**WAIT_INT2: WAIT has completed on channel 2.** 0 = Not_pending 1 = Pending
1	**WAIT_INT1: WAIT has completed on channel 1.** 0 = Not_pending 1 = Pending

Bit	Description
0	**WAIT_INT0: WAIT has completed on channel 0.** 0 = Not_pending 1 = Pending

HOST1X_SYNC_HINTMASK_0

Offset: 014h
Read/Write: R/W
Reset: 0000.0000

Host Interrupt Mask

Contains the interrupt mask bits for all of the host interrupts. If the INT_MASK is ENABLED for a particular module, the modules INT bit from INTSTATUS contributes to the global interrupt signal (host2pad_mhgp3)

Bit	Description
29	**CSW_HOST1XW2MC_INTMASK**
28	**RDMA_DATABUF_THOLD_INTMASK3:** 0 = Disable 1 = Enable
27	**RDMA_BUF_THOLD_INTMASK3:** 0 = Disable 1 = Enable
26	**RDMA_BUF_OFLOW_INTMASK3:** 0 = Disable 1 = Enable
25	**RDMA_DATABUF_THOLD_INTMASK2:** 0 = Disable 1 = Enable
24	**RDMA_BUF_THOLD_INTMASK2:** 0 = Disable 1 = Enable
23	**RDMA_BUF_OFLOW_INTMASK2:** 0 = Disable 1 = Enable
22	**RDMA_DATABUF_THOLD_INTMASK1:** 0 = Disable 1 = Enable
21	**RDMA_BUF_THOLD_INTMASK1:** 0 = Disable 1 = Enable
20	**RDMA_BUF_OFLOW_INTMASK1:** 0 = Disable 1 = Enable
19	**RDMA_DATABUF_THOLD_INTMASK0:** 0 = Disable 1 = Enable
18	**RDMA_BUF_THOLD_INTMASK0:** 0 = Disable 1 = Enable
17	**RDMA_BUF_OFLOW_INTMASK0:** 0 = Disable 1 = Enable

Bit	Description
16	**RDMA_INVAL_CLREQ_INTMASK:** 0 = Disable 1 = Enable
15	**CTXSW_INTMASK7:** 0 = Disable 1 = Enable
14	**CTXSW_INTMASK6:** 0 = Disable 1 = Enable
13	**CTXSW_INTMASK5:** 0 = Disable 1 = Enable
12	**CTXSW_INTMASK4:** 0 = Disable 1 = Enable
11	**CTXSW_INTMASK3:** 0 = Disable 1 = Enable
10	**CTXSW_INTMASK2:** 0 = Disable 1 = Enable
9	**CTXSW_INTMASK1:** 0 = Disable 1 = Enable
8	**CTXSW_INTMASK0:** 0 = Disable 1 = Enable
7	**WAIT_INTMASK7:** 0 = Disable 1 = Enable
6	**WAIT_INTMASK6:** 0 = Disable 1 = Enable
5	**WAIT_INTMASK5:** 0 = Disable 1 = Enable
4	**WAIT_INTMASK4:** 0 = Disable 1 = Enable
3	**WAIT_INTMASK3:** 0 = Disable 1 = Enable
2	**WAIT_INTMASK2:** 0 = Disable 1 = Enable
1	**WAIT_INTMASK1:** 0 = Disable 1 = Enable
0	**WAIT_INTMASK0:** 0 = Disable 1 = Enable

HOST1X_SYNC_CF0_SETUP_0

Offset: 018h
Read/Write: R/W
Reset: 003f.0000

Command FIFO setup registers.

Sets up the various regions of the command FIFO.

Important: This should only be changed when the FIFO is empty.

These are initialized to 8 equal-sized FIFOs.

Bit	Description
24:16	**CF0_LIMIT: Channel 0 FIFO limit (highest address)**
8:0	**CF0_BASE: Channel 0 FIFO base**

HOST1X_SYNC_CF1_SETUP_0

Offset: 01ch
Read/Write: R/W
Reset: 007f.0040

Bit	Description
24:16	**CF1_LIMIT: Channel 1 FIFO limit (highest address)**
8:0	**CF1_BASE: Channel 1 FIFO base**

HOST1X_SYNC_CF2_SETUP_0

Offset: 020h
Read/Write: R/W
Reset: 00bf.0080

Bit	Description
24:16	**CF2_LIMIT: Channel 2 FIFO limit (highest address)**
8:0	**CF2_BASE: Channel 2 FIFO base**

HOST1X_SYNC_CF3_SETUP_0

Offset: 024h
Read/Write: R/W
Reset: 00ff.00c0

Bit	Description
24:16	**CF3_LIMIT: Channel 3 FIFO limit (highest address)**
8:0	**CF3_BASE: Channel 3 FIFO base**

HOST1X_SYNC_CF4_SETUP_0

Offset: 028h
Read/Write: R/W
Reset: 013f.0100

Bit	Description
24:16	**CF4_LIMIT: Channel 4 FIFO limit (highest address)**
8:0	**CF4_BASE: Channel 4 FIFO base**

HOST1X_SYNC_CF5_SETUP_0

Offset: 02ch
Read/Write: R/W
Reset: 017f.0140

Bit	Description
24:16	**CF5_LIMIT: Channel 5 FIFO limit (highest address)**
8:0	**CF5_BASE: Channel 5 FIFO base**

HOST1X_SYNC_CF6_SETUP_0

Offset: 030h
Read/Write: R/W
Reset: 01bf.0180

Bit	Description
24:16	**CF6_LIMIT: Channel 6 FIFO limit (highest address)**
8:0	**CF6_BASE: Channel 6 FIFO base**

HOST1X_SYNC_CF7_SETUP_0

Offset: 034h
Read/Write: R/W
Reset: 01ff.01c0

Bit	Description
24:16	**CF7_LIMIT: Channel 7 FIFO limit (highest address)**
8:0	**CF7_BASE: Channel 7 FIFO base**

HOST1X_SYNC_CF_SETUPDONE_0

Offset: 038h
Read/Write: R/W
Reset: 0000.0000

Write to this register to trigger an update of the FIFO's pointers.

Important: Only do this when the FIFO is empty.

Bit	Description
0	**CF_SETUPDONE: Dummy bit**

HOST1X_SYNC_USEC_CLK_0

Offset: 03ch
Read/Write: R/W
Reset: 0000.007d

Number of host clocks needed to make a microsecond.

Used for the DELAY host method. For example, if the host clock is 250 MHz, this register should be programmed to 250. If the host clock is 150 MHz, this register should be programmed to 150.

Bit	Description
7:0	**USEC_CLKS**

HOST1X_SYNC_HWLOCK0_0

Offset: 040h
Read/Write: RO
Reset: 0000.0001

Hardware lock registers.

These registers are general-purpose and allow software to do an atomic operation to gain ownership of a resource. These registers read back '1' if the reader has obtained ownership and '0' if they have not. This is done by initializing the register to '1'. The act of reading it resets it to '0'. Software must then write '1' back to it when it no longer needs the resource.

Bit	Description
0	**HWLOCK0**

HOST1X_SYNC_HWLOCK1_0

Offset: 044h
Read/Write: RO
Reset: 0000.0001

Bit	Description
0	**HWLOCK1**

HOST1X_SYNC_HWLOCK2_0

Offset: 048h
Read/Write: RO
Reset: 0000.0001

Bit	Description
0	**HWLOCK2**

HOST1X_SYNC_HWLOCK3_0

Offset: 04ch
Read/Write: RO
Reset: 0000.0001

Bit	Description
0	**HWLOCK3**

HOST1X_SYNC_HWLOCK4_0

Offset: 050h
Read/Write: RO
Reset: 0000.0001

Bit	Description
0	**HWLOCK4**

HOST1X_SYNC_HWLOCK5_0

Offset: 054h
Read/Write: RO
Reset: 0000.0001

Bit	Description
0	**HWLOCK5**

HOST1X_SYNC_HWLOCK6_0

Offset: 058h
Read/Write: RO
Reset: 0000.0001

Bit	Description
0	**HWLOCK6**

HOST1X_SYNC_HWLOCK7_0

Offset: 05ch
Read/Write: RO
Reset: 0000.0001

Bit	Description
0	**HWLOCK7**

HOST1X_SYNC_CH_TEARDOWN_0

Offset: 060h
Read/Write: R/W
Reset: 0000.0000

Channel tear down register.

Tells the hardware that a channel has gone away. Will reset that channel's command FIFO and release any locks it has in the arbiter. Will NOT reset that channel's output FIFO, which can be emptied by reading out all remaining entries.

Bit	Description
7	**CH7_TEARDOWN:** 0 = No_action 1 = Teardown
6	**CH6_TEARDOWN:** 0 = No_action 1 = Teardown
5	**CH5_TEARDOWN:** 0 = No_action 1 = Teardown
4	**CH4_TEARDOWN:** 0 = No_action 1 = Teardown
3	**CH3_TEARDOWN:** 0 = No_action 1 = Teardown
2	**CH2_TEARDOWN:** 0 = No_action 1 = Teardown
1	**CH1_TEARDOWN:** 0 = No_action 1 = Teardown
0	**CH0_TEARDOWN:** 0 = No_action 1 = Teardown

HOST1X_SYNC_MOD_TEARDOWN_0

Offset: 064h
Read/Write: R/W
Reset: 0000.0000

Module tear down register.

If a module is reset, the host needs to reset its state with respect to which channels own that module. Whenever a module is reset, the corresponding tear down bit in this register should be written. Module tear down will only work if the command FIFO/command processor are in a good state (the FIFO is empty of traffic for that channel and the command processor is at an opcode boundary). If an entire channel needs to be reset, the CH_TEARDOWN register should be used instead.

Bit	Description
14	**VI_TEARDOWN:** 0 = No_action 1 = Teardown
13	**SD_TEARDOWN:** 0 = No_action 1 = Teardown
12	**ME_TEARDOWN:** 0 = No_action 1 = Teardown
11	**MC_TEARDOWN:** 0 = No_action 1 = Teardown
10	**JPEGE_TEARDOWN:** 0 = No_action 1 = Teardown
9	**ISP_TEARDOWN:** 0 = No_action 1 = Teardown
8	**IC_TEARDOWN:** 0 = No_action 1 = Teardown
7	**I2S_TEARDOWN:** 0 = No_action 1 = Teardown
6	**GRMPD_TEARDOWN:** 0 = No_action 1 = Teardown
5	**GR3D_TEARDOWN:** 0 = No_action 1 = Teardown
4	**GR2D_TEARDOWN:** 0 = No_action 1 = Teardown
3	**EPP_TEARDOWN:** 0 = No_action 1 = Teardown
2	**EMC_TEARDOWN:** 0 = No_action 1 = Teardown

Bit	Description
1	**DSP_TEARDOWN:** 0 = No_action 1 = Teardown
0	**DISPLAY_TEARDOWN:** 0 = No_action 1 = Teardown

HOST1X_SYNC_ARBCONFIG_0

Offset: 068h
Read/Write: R/W
Reset: 0000.0000

Command arbiter config register.

*_CTXSW_MODE selects between automatic and manual context switch mode on a per-client basis. These govern how the arbiter will act when a channel is blocked (another channel owns the client it wants). In AUTO mode, the command arbiter will initiate a context switch whenever the blocking channel reaches a valid context switch point (SETCLASS or CHDONE command). This mode may result in context thrashing if several channels all want to talk to the same client. MANUAL mode gives control to software. If a channel becomes blocked, it triggers an interrupt. When software wants the blocked channel to take over the disputed client, it writes to CH*_CTXSW (see CTXSW register) to enable the channel to continue.

Again, the command arbiter will initiate the context switch whenever the blocking channel reaches a valid context switch point. CH*_CMDPROC_STOP stops issuing commands from the command FIFO. This is useful to stop other channels when a channel tear down is needed to prevent unwanted traffic from happening at the same time.

Bit	Description
22	**VI_CTXSW_MODE:** 0 = Manual 1 = Auto
21	**SD_CTXSW_MODE:** 0 = Manual 1 = Auto
20	**ME_CTXSW_MODE:** 0 = Manual 1 = Auto
19	**MC_CTXSW_MODE:** 0 = Manual 1 = Auto
18	**JPEGE_CTXSW_MODE:** 0 = Manual 1 = Auto
17	**ISP_CTXSW_MODE:** 0 = Manual 1 = Auto
16	**IC_CTXSW_MODE:** 0 = Manual 1 = Auto
15	**I2S_CTXSW_MODE:** 0 = Manual 1 = Auto

Bit	Description
14	**GRMPD_CTXSW_MODE:** 0 = Manual 1 = Auto
13	**GR3D_CTXSW_MODE:** 0 = Manual 1 = Auto
12	**GR2D_CTXSW_MODE:** 0 = Manual 1 = Auto
11	**EPP_CTXSW_MODE:** 0 = Manual 1 = Auto
10	**EMC_CTXSW_MODE:** 0 = Manual 1 = Auto
9	**DSP_CTXSW_MODE:** 0 = Manual 1 = Auto
8	**DISPLAY_CTXSW_MODE:** 0 = Manual 1 = Auto
7	**CH7_CMDPROC_STOP:** 0 = Run 1 = Stop
6	**CH6_CMDPROC_STOP:** 0 = Run 1 = Stop
5	**CH5_CMDPROC_STOP:** 0 = Run 1 = Stop
4	**CH4_CMDPROC_STOP:** 0 = Run 1 = Stop
3	**CH3_CMDPROC_STOP:** 0 = Run 1 = Stop
2	**CH2_CMDPROC_STOP:** 0 = Run 1 = Stop
1	**CH1_CMDPROC_STOP:** 0 = Run 1 = Stop
0	**CH0_CMDPROC_STOP:** 0 = Run 1 = Stop

HOST1X_SYNC_CTXSW_0

Offset: 06ch
Read/Write: R/W
Reset: 0000.0000

Context switch control register. If a channel becomes blocked, it triggers an interrupt. When software wants the blocked channel to take over the disputed client, it writes to CH*_CTXSW to enable the channel to continue. Again, the command arbiter will initiate the context switch whenever the blocking channel reaches a valid context switch point.

Bit	Description
7	**CH7_CTXSW:** 0 = No_action 1 = Allow
6	**CH6_CTXSW:** 0 = No_action 1 = Allow
5	**CH5_CTXSW:** 0 = No_action 1 = Allow
4	**CH4_CTXSW:** 0 = No_action 1 = Allow
3	**CH3_CTXSW:** 0 = No_action 1 = Allow
2	**CH2_CTXSW:** 0 = No_action 1 = Allow
1	**CH1_CTXSW:** 0 = No_action 1 = Allow
0	**CH0_CTXSW:** 0 = No_action 1 = Allow

HOST1X_SYNC_CH0_STATUS_0

Offset: 070h
Read/Write: RO
Reset: 0000.0000

The channel status registers show which clients each channel owns as well as each channel's working class. This is useful for determining which clients need to be cleaned up for a tear down as well as each channel's state with regard to granting context switches. CHOUT_CLASS* holds the requested class in the case of a blocked context switch.

Bit	Description
25:16	**CHOUT_CLASS0: Current or blocked (requested) class for channel 0**
14	**VI_OWNED0:** 0 = Not owned 1 = Owned
13	**SD_OWNED0:** 0 = Not owned 1 = Owned
12	**ME_OWNED0:** 0 = Not owned 1 = Owned
11	**MC_OWNED0:** 0 = Not owned 1 = Owned
10	**JPEGE_OWNED0:** 0 = Not owned 1 = Owned
9	**ISP_OWNED0:** 0 = Not owned 1 = Owned
8	**IC_OWNED0:** 0 = Not owned 1 = Owned
7	**I2S_OWNED0:** 0 = Not owned 1 = Owned
6	**GRMPD_OWNED0:** 0 = Not owned 1 = Owned
5	**GR3D_OWNED0:** 0 = Not owned 1 = Owned
4	**GR2D_OWNED0:** 0 = Not owned 1 = Owned
3	**EPP_OWNED0:** 0 = Not owned 1 = Owned
2	**EMC_OWNED0:** 0 = Not owned 1 = Owned
1	**DSP_OWNED0:** 0 = Not owned 1 = Owned

Bit	Description
0	**DISPLAY_OWNED0:** 0 = Not owned 1 = Owned

HOST1X_SYNC_CH1_STATUS_0

Offset: 074h
Read/Write: RO
Reset: 0000.0000

Bit	Description
25:16	**CHOUT_CLASS1: Current or blocked (requested) class for channel 1**
14	**VI_OWNED1:** 0 = Not owned 1 = Owned
13	**SD_OWNED1:** 0 = Not owned 1 = Owned
12	**ME_OWNED1:** 0 = Not owned 1 = Owned
11	**MC_OWNED1:** 0 = Not owned 1 = Owned
10	**JPEGE_OWNED1:** 0 = Not owned 1 = Owned
9	**ISP_OWNED1:** 0 = Not owned 1 = Owned
8	**IC_OWNED1:** 0 = Not owned 1 = Owned
7	**I2S_OWNED1:** 0 = Not owned 1 = Owned
6	**GRMPD_OWNED1:** 0 = Not owned 1 = Owned
5	**GR3D_OWNED1:** 0 = Not owned 1 = Owned
4	**GR2D_OWNED1:** 0 = Not owned 1 = Owned
3	**EPP_OWNED1:** 0 = Not owned 1 = Owned
2	**EMC_OWNED1:** 0 = Not owned 1 = Owned
1	**DSP_OWNED1:** 0 = Not owned 1 = Owned

Bit	Description
0	**DISPLAY_OWNED1:** 0 = Not owned 1 = Owned

HOST1X_SYNC_CH2_STATUS_0

Offset: 078h
Read/Write: RO
Reset: 0000.0000

Bit	Description
25:16	**CHOUT_CLASS2: Current or blocked (requested) class for channel 2**
14	**VI_OWNED2:** 0 = Not owned 1 = Owned
13	**SD_OWNED2:** 0 = Not owned 1 = Owned
12	**ME_OWNED2:** 0 = Not owned 1 = Owned
11	**MC_OWNED2:** 0 = Not owned 1 = Owned
10	**JPEGE_OWNED2:** 0 = Not owned 1 = Owned
9	**ISP_OWNED2:** 0 = Not owned 1 = Owned
8	**IC_OWNED2:** 0 = Not owned 1 = Owned
7	**I2S_OWNED2:** 0 = Not owned 1 = Owned
6	**GRMPD_OWNED2:** 0 = Not owned 1 = Owned
5	**GR3D_OWNED2:** 0 = Not owned 1 = Owned
4	**GR2D_OWNED2:** 0 = Not owned 1 = Owned
3	**EPP_OWNED2:** 0 = Not owned 1 = Owned
2	**EMC_OWNED2:** 0 = Not owned 1 = Owned
1	**DSP_OWNED2:** 0 = Not owned 1 = Owned

Bit	Description
0	**DISPLAY_OWNED2:** 0 = Not owned 1 = Owned

HOST1X_SYNC_CH3_STATUS_0

Offset: 07ch
Read/Write: RO
Reset: 0000.0000

Bit	Description
25:16	**CHOUT_CLASS3: Current or blocked (requested) class for channel 3**
14	**VI_OWNED3:** 0 = Not owned 1 = Owned
13	**SD_OWNED3:** 0 = Not owned 1 = Owned
12	**ME_OWNED3:** 0 = Not owned 1 = Owned
11	**MC_OWNED3:** 0 = Not owned 1 = Owned
10	**JPEGE_OWNED3:** 0 = Not owned 1 = Owned
9	**ISP_OWNED3:** 0 = Not owned 1 = Owned
8	**IC_OWNED3:** 0 = Not owned 1 = Owned
7	**I2S_OWNED3:** 0 = Not owned 1 = Owned
6	**GRMPD_OWNED3:** 0 = Not owned 1 = Owned
5	**GR3D_OWNED3:** 0 = Not owned 1 = Owned
4	**GR2D_OWNED3:** 0 = Not owned 1 = Owned
3	**EPP_OWNED3:** 0 = Not owned 1 = Owned
2	**EMC_OWNED3:** 0 = Not owned 1 = Owned
1	**DSP_OWNED3:** 0 = Not owned 1 = Owned

Bit	Description
0	**DISPLAY_OWNED3:** 0 = Not owned 1 = Owned

HOST1X_SYNC_CH4_STATUS_0

Offset: 080h
Read/Write: RO
Reset: 0000.0000

Bit	Description
25:16	**CHOUT_CLASS4: Current or blocked (requested) class for channel 4**
14	**VI_OWNED4:** 0 = Not owned 1 = Owned
13	**SD_OWNED4:** 0 = Not owned 1 = Owned
12	**ME_OWNED4:** 0 = Not owned 1 = Owned
11	**MC_OWNED4:** 0 = Not owned 1 = Owned
10	**JPEGE_OWNED4:** 0 = Not owned 1 = Owned
9	**ISP_OWNED4:** 0 = Not owned 1 = Owned
8	**IC_OWNED4:** 0 = Not owned 1 = Owned
7	**I2S_OWNED4:** 0 = Not owned 1 = Owned
6	**GRMPD_OWNED4:** 0 = Not owned 1 = Owned
5	**GR3D_OWNED4:** 0 = Not owned 1 = Owned
4	**GR2D_OWNED4:** 0 = Not owned 1 = Owned
3	**EPP_OWNED4:** 0 = Not owned 1 = Owned
2	**EMC_OWNED4:** 0 = Not owned 1 = Owned
1	**DSP_OWNED4:** 0 = Not owned 1 = Owned

Bit	Description
0	**DISPLAY_OWNED4:** 0 = Not owned 1 = Owned

HOST1X_SYNC_CH5_STATUS_0

Offset: 084h
Read/Write: RO
Reset: 0000.0000

Bit	Description
25:16	**CHOUT_CLASS5: Current or blocked (requested) class for channel 5**
14	**VI_OWNED5:** 0 = Not owned 1 = Owned
13	**SD_OWNED5:** 0 = Not owned 1 = Owned
12	**ME_OWNED5:** 0 = Not owned 1 = Owned
11	**MC_OWNED5:** 0 = Not owned 1 = Owned
10	**JPEGE_OWNED5:** 0 = Not owned 1 = Owned
9	**ISP_OWNED5:** 0 = Not owned 1 = Owned
8	**IC_OWNED5:** 0 = Not owned 1 = Owned
7	**I2S_OWNED5:** 0 = Not owned 1 = Owned
6	**GRMPD_OWNED5:** 0 = Not owned 1 = Owned
5	**GR3D_OWNED5:** 0 = Not owned 1 = Owned
4	**GR2D_OWNED5:** 0 = Not owned 1 = Owned
3	**EPP_OWNED5:** 0 = Not owned 1 = Owned
2	**EMC_OWNED5:** 0 = Not owned 1 = Owned
1	**DSP_OWNED5:** 0 = Not owned 1 = Owned

Bit	Description
0	**DISPLAY_OWNED5:** 0 = Not owned 1 = Owned

HOST1X_SYNC_CH6_STATUS_0

Offset: 088h
Read/Write: RO
Reset: 0000.0000

Bit	Description
25:16	**CHOUT_CLASS6: Current or blocked (requested) class for channel 6**
14	**VI_OWNED6:** 0 = Not owned 1 = Owned
13	**SD_OWNED6:** 0 = Not owned 1 = Owned
12	**ME_OWNED6:** 0 = Not owned 1 = Owned
11	**MC_OWNED6:** 0 = Not owned 1 = Owned
10	**JPEGE_OWNED6:** 0 = Not owned 1 = Owned
9	**ISP_OWNED6:** 0 = Not owned 1 = Owned
8	**IC_OWNED6:** 0 = Not owned 1 = Owned
7	**I2S_OWNED6:** 0 = Not owned 1 = Owned
6	**GRMPD_OWNED6:** 0 = Not owned 1 = Owned
5	**GR3D_OWNED6:** 0 = Not owned 1 = Owned
4	**GR2D_OWNED6:** 0 = Not owned 1 = Owned
3	**EPP_OWNED6:** 0 = Not owned 1 = Owned
2	**EMC_OWNED6:** 0 = Not owned 1 = Owned
1	**DSP_OWNED6:** 0 = Not owned 1 = Owned

Bit	Description
0	**DISPLAY_OWNED6:** 0 = Not owned 1 = Owned

HOST1X_SYNC_CH7_STATUS_0

Offset: 08ch
Read/Write: RO
Reset: 0000.0000

Bit	Description
25:16	**CHOUT_CLASS7: Current or blocked (requested) class for channel 7**
14	**VI_OWNED7:** 0 = Not owned 1 = Owned
13	**SD_OWNED7:** 0 = Not owned 1 = Owned
12	**ME_OWNED7:** 0 = Not owned 1 = Owned
11	**MC_OWNED7:** 0 = Not owned 1 = Owned
10	**JPEGE_OWNED7:** 0 = Not owned 1 = Owned
9	**ISP_OWNED7:** 0 = Not owned 1 = Owned
8	**IC_OWNED7:** 0 = Not owned 1 = Owned
7	**I2S_OWNED7:** 0 = Not owned 1 = Owned
6	**GRMPD_OWNED7:** 0 = Not owned 1 = Owned
5	**GR3D_OWNED7:** 0 = Not owned 1 = Owned
4	**GR2D_OWNED7:** 0 = Not owned 1 = Owned
3	**EPP_OWNED7:** 0 = Not owned 1 = Owned
2	**EMC_OWNED7:** 0 = Not owned 1 = Owned
1	**DSP_OWNED7:** 0 = Not owned 1 = Owned

Bit	Description
0	**DISPLAY_OWNED7:** 0 = Not owned 1 = Owned

HOST1X_SYNC_DISPLAY_STATUS_0

Offset: 090h
Read/Write: RO
Reset: 0000.0000

The per-client status registers indicate which channel owns each client as well as the current working class for each client. This is useful for determining each client's state with regard to granting context switches.

Bit	Description
25:16	**DISPLAY_CURRCL: Current working class**
7	**DISPLAY_OWN7:** 0 = Not owned 1 = Owned
6	**DISPLAY_OWN6:** 0 = Not owned 1 = Owned
5	**DISPLAY_OWN5:** 0 = Not owned 1 = Owned
4	**DISPLAY_OWN4:** 0 = Not owned 1 = Owned
3	**DISPLAY_OWN3:** 0 = Not owned 1 = Owned
2	**DISPLAY_OWN2:** 0 = Not owned 1 = Owned
1	**DISPLAY_OWN1:** 0 = Not owned 1 = Owned
0	**DISPLAY_OWN0:** 0 = Not owned 1 = Owned

HOST1X_SYNC_DSP_STATUS_0

Offset: 094h
Read/Write: RO
Reset: 0000.0000

Bit	Description
25:16	**DSP_CURRCL: Current working class**
7	**DSP_OWN7:** 0 = Not owned 1 = Owned
6	**DSP_OWN6:** 0 = Not owned 1 = Owned
5	**DSP_OWN5:** 0 = Not owned 1 = Owned
4	**DSP_OWN4:** 0 = Not owned 1 = Owned
3	**DSP_OWN3:** 0 = Not owned 1 = Owned
2	**DSP_OWN2:** 0 = Not owned 1 = Owned
1	**DSP_OWN1:** 0 = Not owned 1 = Owned
0	**DSP_OWN0:** 0 = Not owned 1 = Owned

HOST1X_SYNC_EMC_STATUS_0

Offset: 098h
Read/Write: RO
Reset: 0000.0000

Bit	Description
25:16	**EMC_CURRCL: Current working class**
7	**EMC_OWN7:** 0 = Not owned 1 = Owned
6	**EMC_OWN6:** 0 = Not owned 1 = Owned
5	**EMC_OWN5:** 0 = Not owned 1 = Owned
4	**EMC_OWN4:** 0 = Not owned 1 = Owned
3	**EMC_OWN3:** 0 = Not owned 1 = Owned
2	**EMC_OWN2:** 0 = Not owned 1 = Owned
1	**EMC_OWN1:** 0 = Not owned 1 = Owned
0	**EMC_OWN0:** 0 = Not owned 1 = Owned

HOST1X_SYNC_EPP_STATUS_0

Offset: 09ch
Read/Write: RO
Reset: 0000.0000

Bit	Description
25:16	**EPP_CURRCL: Current working class**
7	**EPP_OWN7:** 0 = Not owned 1 = Owned
6	**EPP_OWN6:** 0 = Not owned 1 = Owned
5	**EPP_OWN5:** 0 = Not owned 1 = Owned
4	**EPP_OWN4:** 0 = Not owned 1 = Owned
3	**EPP_OWN3:** 0 = Not owned 1 = Owned
2	**EPP_OWN2:** 0 = Not owned 1 = Owned
1	**EPP_OWN1:** 0 = Not owned 1 = Owned
0	**EPP_OWN0:** 0 = Not owned 1 = Owned

HOST1X_SYNC_GR2D_STATUS_0

Offset: 0a0h
Read/Write: RO
Reset: 0000.0000

Bit	Description
25:16	**GR2D_CURRCL: Current working class**
7	**GR2D_OWN7:** 0 = Not owned 1 = Owned
6	**GR2D_OWN6:** 0 = Not owned 1 = Owned
5	**GR2D_OWN5:** 0 = Not owned 1 = Owned
4	**GR2D_OWN4:** 0 = Not owned 1 = Owned
3	**GR2D_OWN3:** 0 = Not owned 1 = Owned

Bit	Description
2	**GR2D_OWN2:** 0 = Not owned 1 = Owned
1	**GR2D_OWN1:** 0 = Not owned 1 = Owned
0	**GR2D_OWN0:** 0 = Not owned 1 = Owned

HOST1X_SYNC_GR3D_STATUS_0

Offset: 0a4h
Read/Write: RO
Reset: 0000.0000

Bit	Description
25:16	**GR3D_CURRCL: Current working class**
7	**GR3D_OWN7:** 0 = Not owned 1 = Owned
6	**GR3D_OWN6:** 0 = Not owned 1 = Owned
5	**GR3D_OWN5:** 0 = Not owned 1 = Owned
4	**GR3D_OWN4:** 0 = Not owned 1 = Owned
3	**GR3D_OWN3:** 0 = Not owned 1 = Owned
2	**GR3D_OWN2:** 0 = Not owned 1 = Owned
1	**GR3D_OWN1:** 0 = Not owned 1 = Owned
0	**GR3D_OWN0:** 0 = Not owned 1 = Owned

HOST1X_SYNC_GRMPD_STATUS_0

Offset: 0a8h
Read/Write: RO
Reset: 0000.0000

Bit	Description
25:16	**GRMPD_CURRCL: Current working class**
7	**GRMPD_OWN7:** 0 = Not owned 1 = Owned
6	**GRMPD_OWN6:** 0 = Not owned 1 = Owned
5	**GRMPD_OWN5:** 0 = Not owned 1 = Owned
4	**GRMPD_OWN4:** 0 = Not owned 1 = Owned
3	**GRMPD_OWN3:** 0 = Not owned 1 = Owned
2	**GRMPD_OWN2:** 0 = Not owned 1 = Owned
1	**GRMPD_OWN1:** 0 = Not owned 1 = Owned
0	**GRMPD_OWN0:** 0 = Not owned 1 = Owned

HOST1X_SYNC_I2S_STATUS_0

Offset: 0ach
Read/Write: RO
Reset: 0000.0000

Bit	Description
25:16	**I2S_CURRCL: Current working class**
7	**I2S_OWN7:** 0 = Not owned 1 = Owned
6	**I2S_OWN6:** 0 = Not owned 1 = Owned
5	**I2S_OWN5:** 0 = Not owned 1 = Owned
4	**I2S_OWN4:** 0 = Not owned 1 = Owned
3	**I2S_OWN3:** 0 = Not owned 1 = Owned
2	**I2S_OWN2:** 0 = Not owned 1 = Owned
1	**I2S_OWN1:** 0 = Not owned 1 = Owned
0	**I2S_OWN0:** 0 = Not owned 1 = Owned

HOST1X_SYNC_IC_STATUS_0

Offset: 0b0h
Read/Write: RO
Reset: 0000.0000

Bit	Description
25:16	**IC_CURRCL: Current working class**
7	**IC_OWN7:** 0 = Not owned 1 = Owned
6	**IC_OWN6:** 0 = Not owned 1 = Owned
5	**IC_OWN5:** 0 = Not owned 1 = Owned
4	**IC_OWN4:** 0 = Not owned 1 = Owned
3	**IC_OWN3:** 0 = Not owned 1 = Owned
2	**IC_OWN2:** 0 = Not owned 1 = Owned
1	**IC_OWN1:** 0 = Not owned 1 = Owned
0	**IC_OWN0:** 0 = Not owned 1 = Owned

HOST1X_SYNC_ISP_STATUS_0

Offset: 0b4h
Read/Write: RO
Reset: 0000.0000

Bit	Description
25:16	**ISP_CURRCL: Current working class**
7	**ISP_OWN7:** 0 = Not owned 1 = Owned
6	**ISP_OWN6:** 0 = Not owned 1 = Owned
5	**ISP_OWN5:** 0 = Not owned 1 = Owned
4	**ISP_OWN4:** 0 = Not owned 1 = Owned
3	**ISP_OWN3:** 0 = Not owned 1 = Owned

Bit	Description
2	**ISP_OWN2:** 0 = Not owned 1 = Owned
1	**ISP_OWN1:** 0 = Not owned 1 = Owned
0	**ISP_OWN0:** 0 = Not owned 1 = Owned

HOST1X_SYNC_JPEGE_STATUS_0

Offset: 0b8h
Read/Write: RO
Reset: 0000.0000

Bit	Description
25:16	**JPEGE_CURRCL: Current working class**
7	**JPEGE_OWN7:** 0 = Not owned 1 = Owned
6	**JPEGE_OWN6:** 0 = Not owned 1 = Owned
5	**JPEGE_OWN5:** 0 = Not owned 1 = Owned
4	**JPEGE_OWN4:** 0 = Not owned 1 = Owned
3	**JPEGE_OWN3:** 0 = Not owned 1 = Owned
2	**JPEGE_OWN2:** 0 = Not owned 1 = Owned
1	**JPEGE_OWN1:** 0 = Not owned 1 = Owned
0	**JPEGE_OWN0:** 0 = Not owned 1 = Owned

HOST1X_SYNC_MC_STATUS_0

Offset: 0bch
Read/Write: RO
Reset: 0000.0000

Bit	Description
25:16	**MC_CURRCL: Current working class**
7	**MC_OWN7:** 0 = Not owned 1 = Owned
6	**MC_OWN6:** 0 = Not owned 1 = Owned
5	**MC_OWN5:** 0 = Not owned 1 = Owned
4	**MC_OWN4:** 0 = Not owned 1 = Owned
3	**MC_OWN3:** 0 = Not owned 1 = Owned
2	**MC_OWN2:** 0 = Not owned 1 = Owned
1	**MC_OWN1:** 0 = Not owned 1 = Owned
0	**MC_OWN0:** 0 = Not owned 1 = Owned

HOST1X_SYNC_ME_STATUS_0

Offset: 0c0h
Read/Write: RO
Reset: 0000.0000

Bit	Description
25:16	**ME_CURRCL: Current working class**
7	**ME_OWN7:** 0 = Not owned 1 = Owned
6	**ME_OWN6:** 0 = Not owned 1 = Owned
5	**ME_OWN5:** 0 = Not owned 1 = Owned
4	**ME_OWN4:** 0 = Not owned 1 = Owned
3	**ME_OWN3:** 0 = Not owned 1 = Owned
2	**ME_OWN2:** 0 = Not owned 1 = Owned
1	**ME_OWN1:** 0 = Not owned 1 = Owned
0	**ME_OWN0:** 0 = Not owned 1 = Owned

HOST1X_SYNC_SD_STATUS_0

Offset: 0c4h
Read/Write: RO
Reset: 0000.0000

Bit	Description
25:16	**SD_CURRCL: Current working class**
7	**SD_OWN7:** 0 = Not owned 1 = Owned
6	**SD_OWN6:** 0 = Not owned 1 = Owned
5	**SD_OWN5:** 0 = Not owned 1 = Owned
4	**SD_OWN4:** 0 = Not owned 1 = Owned
3	**SD_OWN3:** 0 = Not owned 1 = Owned

Bit	Description
2	**SD_OWN2:** 0 = Not owned 1 = Owned
1	**SD_OWN1:** 0 = Not owned 1 = Owned
0	**SD_OWN0:** 0 = Not owned 1 = Owned

HOST1X_SYNC_VI_STATUS_0

Offset: 0c8h
Read/Write: RO
Reset: 0000.0000

Bit	Description
25:16	**VI_CURRCL: Current working class**
7	**VI_OWN7:** 0 = Not owned 1 = Owned
6	**VI_OWN6:** 0 = Not owned 1 = Owned
5	**VI_OWN5:** 0 = Not owned 1 = Owned
4	**VI_OWN4:** 0 = Not owned 1 = Owned
3	**VI_OWN3:** 0 = Not owned 1 = Owned
2	**VI_OWN2:** 0 = Not owned 1 = Owned
1	**VI_OWN1:** 0 = Not owned 1 = Owned
0	**VI_OWN0:** 0 = Not owned 1 = Owned

HOST1X_SYNC_RDMA_ARB_COUNT_0

Offset: 0cch
Read/Write: R/W
Reset: 0000.0000

Arbitration count:

Any number > 0 allows a count extra acknowledges per ReadDMA port and CDMA channel

Bit	Description
3:0	**RDMA_ARB_COUNT**

HOST1X_SYNC_RDMA_CONFIG_0

Offset: 0d0h
Read/Write: R/W
Reset: 0000.0000

one control bit per Read DMA port, indicating if the port is in rectangular mode configure particular ports to be in rectangular mode

Bit	Description
3:0	**RDMA_RECTANGULAR_BUFFER**

HOST1X_SYNC_RDMA_WRAP_0

Offset: 0d4h
Read/Write: R/W
Reset: 0000.0000

One control bit per read DMA port, indicating it is ok to wrap to base at the end of the buffer chain configure particular ports to be in buffer wrapping mode

Bit	Description
3:0	**RDMA_WRAP**

HOST1X_SYNC_RDMA_STATUS0_0

Offset: 0d8h
Read/Write: RO
Reset: 0000.0000

PORT 0 Registers

Bit	Description
19:16	**RDMA_PEND_MEM_REQ0: pending memory requests on port0**
15:7	**RDMA_PENDBUFRDY0: number of pending buffer_rdy's (pending buffers in memory)**
6:1	**RDMA_OUTFENTRIES0: number of 32-bit words in read DMA fifo**
0	**RDMA_WRAPPED0: read DMA port wrapped to base address**

HOST1X_SYNC_RDMA_BUFFER_THRESHOLD0_0

Offset: 0dch
Read/Write: R/W
Reset: 0000.0000

Bit	Description
12:9	**DATABUF_THRESHOLD0**
8:0	**BUFFER_THRESHOLD0: Threshold of number of buffers pending in memory at which an interrupt is generated**

HOST1X_SYNC_RDMA_CONF0_0

Offset: 0e0h
Read/Write: R/W
Reset: 0000.0000

Bit	Description
21	**RDMA_STORE_HDR0: Store header with bufferdata in DMA fifo (data is sent to DSP/CPU)**
20:9	**RDMA_NUM_LINES0: Rectangular reads: Number of lines per buffer**
8:0	**RDMA_NUM_BUFFERS0: Number of buffers defined for read DMA FIFO0**

HOST1X_SYNC_RDMA_SWAP0_0

Offset: 0e4h
Read/Write: R/W
Reset: 0000.0000

Bit	Description
3:2	**RDMA_HEADER_SWAP0: byte-swap setting for header**
1:0	**RDMA_DATA_SWAP0: byte-swap setting for data**

HOST1X_SYNC_RDMA_LINE0_0

Offset: 0e8h
Read/Write: R/W
Reset: 0000.0000

IMPORTANT:

For rectangular reads (line_stride):
- if line_width is not a multiple of 4 bytes (one word):
if line_stride equals line_width, line_stride should be the exact stride in bytes (eg. line_stride[1:0] not equal to 0)
if line_stride is bigger than line_width, line_stride should be rounded down to the nearest word
(eg. line_stride[1:0] should be zero)

For example: line_width = 42 bytes, line_stride = 82 -> stride should be programmed to 80
line_width = 42 bytes, line_stride = 42 -> stride should be programmed to 42

Bit	Description
23:12	**RDMA_LINE_WIDTH0: rectangular reads: line_width (bytes)**
11:0	**RDMA_LINE_STRIDE0: rectangular reads: line_stride(bytes); JPEGE: buffer_stride (16 byte aligned!!)**

HOST1X_SYNC_RDMA_CLID0_0

Offset: 0ech
Read/Write: R/W
Reset: 0000.0000

Write buffer client attached to port FIFO

Bit	Description
2:0	**RDMA_CLIENTID0: 0 = CPU** 1 = DSP 2 = I2S 3 = SD 4 = MPEGE 5 = JPEGE 6 = EPP 7 = VI

HOST1X_SYNC_RDMA_BADDR0_0

Offset: 0f0h
Read/Write: R/W
Reset: 0000.0000

Buffer base address for port0

Bit	Description
25:2	**RDMA_BASE_ADDR0**

HOST1X_SYNC_RDMA_DMATRIGGER0_0

Offset: 0f4h
Read/Write: R/W
Reset: 0000.0000

NOTE: for these triggers to work, software must write the SRC_DSP field first. (0==CPU,1==DSP)

Bit	Description
4	**RDMA_SRC_DSP0: indicates writes to this register are triggered by DSP(1) or CPU(0)**
2	**RDMA_LAST_BUF0: last_buffer: indicates this is the last buffer in a chain**
1	**RDMA_BUF_RDY0: buffer_rdy : indicates buffer is pending in memory**
0	**RDMA_BUF_INIT0: buffer_init: initializes read DMA port**

HOST1X_SYNC_RDMA_STATUS1_0

Offset: 0f8h
Read/Write: RO
Reset: 0000.0000

PORT 1 Registers

Bit	Description
19:16	**RDMA_PEND_MEM_REQ1: pending memory requests on port1**
15:7	**RDMA_PENDBUFRDY1: number of pending buffer_rdy's (pending buffers in memory)**
6:1	**RDMA_OUTFENTRIES1: number of 32-bit words in read DMA fifo**
0	**RDMA_WRAPPED1: read DMA port wrapped to base address**

HOST1X_SYNC_RDMA_BUFFER_THRESHOLD1_0

Offset: 0fch
Read/Write: R/W
Reset: 0000.0000

Bit	Description
12:9	**DATABUF_THRESHOLD1**
8:0	**BUFFER_THRESHOLD1: Threshold of number of buffers pending in memory at which an interrupt is generated**

HOST1X_SYNC_RDMA_CONF1_0

Offset: 100h
Read/Write: R/W
Reset: 0000.0000

Bit	Description
21	**RDMA_STORE_HDR1: Store header with buffer data in DMA fifo (data is sent to DSP/CPU)**
20:9	**RDMA_NUM_LINES1: Rectangular reads: Number of lines per buffer**
8:0	**RDMA_NUM_BUFFERS1: Number of buffers defined for read DMA FIFO1**

HOST1X_SYNC_RDMA_SWAP1_0

Offset: 104h
Read/Write: R/W
Reset: 0000.0000

Bit	Description
3:2	**RDMA_HEADER_SWAP1: byte-swap setting for header**
1:0	**RDMA_DATA_SWAP1: byte-swap setting for data**

HOST1X_SYNC_RDMA_LINE1_0

Offset: 108h
Read/Write: R/W
Reset: 0000.0000

IMPORTANT:

For rectangular reads (line_stride):
- if line_width is not a multiple of 4 bytes (one word):
if line_stride equals line_width, line_stride should be the exact stride in bytes (eg. line_stride[1:0] not equal to 0)
if line_stride is bigger than line_width, line_stride should be rounded down to the nearest word (eg. line_stride[1:0] should be zero)

For example: line_width = 42 bytes, line_stride = 82 -> stride should be programmed to 80
line_width = 42 bytes, line_stride = 42 -> stride should be programmed to 42

Bit	Description
23:12	**RDMA_LINE_WIDTH1: rectangular reads: line_width (bytes)**
11:0	**RDMA_LINE_STRIDE1: rectangular reads: line_stride(bytes); JPEGE: buffer_stride (16 byte aligned)**

HOST1X_SYNC_RDMA_CLID1_0

Offset: 10ch
Read/Write: R/W
Reset: 0000.0000

Write buffer client attached to port FIFO

Bit	Description
2:0	**RDMA_CLIENTID1: 0 = CPU** 1 = DSP 2 = I2S 3 = SD 4 = MPEGE 5 = JPEGE 6 = EPP 7 = VI

HOST1X_SYNC_RDMA_BADDR1_0

Offset: 110h
Read/Write: R/W
Reset: 0000.0000

buffer base address for port1

Bit	Description
25:2	**RDMA_BASE_ADDR1**

HOST1X_SYNC_RDMA_DMATRIGGER1_0

Offset: 114h
Read/Write: R/W
Reset: 0000.0000

NOTE: for these triggers to work, software needs to write the SRC_DSP field first! (0==CPU,1==DSP)

Bit	Description
4	**RDMA_SRC_DSP1: indicates writes to this register are triggered by DSP(1) or CPU(0)**
2	**RDMA_LAST_BUF1: last_buffer: indicates this is the last buffer in a chain**
1	**RDMA_BUF_RDY1: buffer_rdy : indicates buffer is pending in memory**
0	**RDMA_BUF_INIT1: buffer_init: initializes read DMA port**

HOST1X_SYNC_RDMA_STATUS2_0

Offset: 118h
Read/Write: RO
Reset: 0000.0000

PORT 2 Registers

Bit	Description
19:16	**RDMA_PEND_MEM_REQ2: pending memory requests on port2**
15:7	**RDMA_PENDBUFRDY2: number of pending buffer_rdy's (pending buffers in memory)**
6:1	**RDMA_OUTFENTRIES2: number of 32-bit words in read DMA fifo**
0	**RDMA_WRAPPED2: read DMA port wrapped to base address**

HOST1X_SYNC_RDMA_BUFFER_THRESHOLD2_0

Offset: 11ch
Read/Write: R/W
Reset: 0000.0000

Bit	Description
12:9	**DATABUF_THRESHOLD2**
8:0	**BUFFER_THRESHOLD2: Threshold of number of buffers pending in memory at which an interrupt is generated**

HOST1X_SYNC_RDMA_CONF2_0

Offset: 120h
Read/Write: R/W
Reset: 0000.0000

Bit	Description
21	**RDMA_STORE_HDR2: Store header with buffer data in DMA FIFO (data is sent to DSP/CPU)**
20:9	**RDMA_NUM_LINES2: Rectangular reads: Number of lines per buffer**
8:0	**RDMA_NUM_BUFFERS2: Number of buffers defined for read DMA FIFO1**

HOST1X_SYNC_RDMA_SWAP2_0

Offset: 124h
Read/Write: R/W
Reset: 0000.0000

Bit	Description
3:2	**RDMA_HEADER_SWAP2: byte-swap setting for header**
1:0	**RDMA_DATA_SWAP2: byte-swap setting for data**

HOST1X_SYNC_RDMA_LINE2_0

Offset: 128h
Read/Write: R/W
Reset: 0000.0000

IMPORTANT:

For rectangular reads (line_stride):
- if line_width is not a multiple of 4 bytes (one word):
if line_stride equals line_width, line_stride should be the exact stride in bytes (eg. line_stride[1:0] not equal to 0)
if line_stride is bigger than line_width, line_stride should be rounded down to the nearest word (eg. line_stride[1:0] should be zero)

For example: line_width = 42 bytes, line_stride = 82 -> stride should be programmed to 80
line_width = 42 bytes, line_stride = 42 -> stride should be programmed to 42.

Bit	Description
23:12	**RDMA_LINE_WIDTH2: rectangular reads: line_width (bytes)**
11:0	**RDMA_LINE_STRIDE2: rectangular reads: line_stride(bytes); JPEGE: buffer_stride (16 byte aligned)**

HOST1X_SYNC_RDMA_CLID2_0

Offset: 12ch
Read/Write: R/W
Reset: 0000.0000

Write buffer client attached to port FIFO

Bit	Description
2:0	**RDMA_CLIENTID2: 0 = CPU** 1 = DSP 2 = I2S 3 = SD 4 = MPEGE 5 = JPEGE 6 = EPP 7 = VI

HOST1X_SYNC_RDMA_BADDR2_0

Offset: 130h
Read/Write: R/W
Reset: 0000.0000

Buffer base address for port1

Bit	Description
25:2	**RDMA_BASE_ADDR2**

HOST1X_SYNC_RDMA_DMATRIGGER2_0

Offset: 134h
Read/Write: R/W
Reset: 0000.0000

NOTE: for these triggers to work, software needs to write the SRC_DSP field first! (0==CPU,1==DSP)

Bit	Description
4	**RDMA_SRC_DSP2: indicates writes to this register are triggered by DSP(1) or CPU(0)**
2	**RDMA_LAST_BUF2: last_buffer: indicates this is the last buffer in a chain**
1	**RDMA_BUF_RDY2: buffer_rdy : indicates buffer is pending in memory**
0	**RDMA_BUF_INIT2: buffer_init: initializes read DMA port**

HOST1X_SYNC_RDMA_STATUS3_0

Offset: 138h
Read/Write: RO
Reset: 0000.0000

PORT 3 Registers

Bit	Description
19:16	**RDMA_PEND_MEM_REQ3: pending memory requests on port3**
15:7	**RDMA_PENDBUFRDY3: number of pending buffer_rdy's (pending buffers in memory)**
6:1	**RDMA_OUTFENTRIES3: number of 32-bit words in read DMA fifo**
0	**RDMA_WRAPPED3: read DMA port wrapped to base address**

HOST1X_SYNC_RDMA_BUFFER_THRESHOLD3_0

Offset: 13ch
Read/Write: R/W
Reset: 0000.0000

Bit	Description
12:9	**DATABUF_THRESHOLD3**
8:0	**BUFFER_THRESHOLD3: Threshold of number of buffers pending in memory at which an interrupt is generated**

HOST1X_SYNC_RDMA_CONF3_0

Offset: 140h
Read/Write: R/W
Reset: 0000.0000

Bit	Description
21	**RDMA_STORE_HDR3: Store header with bufferdata in DMA fifo (data is sent to DSP/CPU)**
20:9	**RDMA_NUM_LINES3: Rectangular reads: Number of lines per buffer**
8:0	**RDMA_NUM_BUFFERS3: Number of buffers defined for read DMA FIFO1**

HOST1X_SYNC_RDMA_SWAP3_0

Offset: 144h
Read/Write: R/W
Reset: 0000.0000

Bit	Description
3:2	**RDMA_HEADER_SWAP3: byte-swap setting for header**
1:0	**RDMA_DATA_SWAP3: byte-swap setting for data**

HOST1X_SYNC_RDMA_LINE3_0

Offset: 148h
Read/Write: R/W
Reset: 0000.0000

IMPORTANT:

For rectangular reads (line_stride):
- if line_width is not a multiple of 4 bytes (one word):
if line_stride equals line_width, line_stride should be the exact stride in bytes (eg. line_stride[1:0] not equal to 0)
if line_stride is bigger than line_width, line_stride should be rounded down to the nearest word (eg. line_stride[1:0] should be zero)

For example: line_width = 42 bytes, line_stride = 82 -> stride should be programmed to 80
line_width = 42 bytes, line_stride = 42 -> stride should be programmed to 42.

Bit	Description
23:12	**RDMA_LINE_WIDTH3: rectangular reads: line_width (bytes)**
11:0	**RDMA_LINE_STRIDE3: rectangular reads: line_stride(bytes); JPEGE: buffer_stride (16 byte aligned)**

HOST1X_SYNC_RDMA_CLID3_0

Offset: 14ch
Read/Write: R/W
Reset: 0000.0000

Write buffer client attached to port FIFO

Bit	Description
2:0	**RDMA_CLIENTID3: 0 = CPU** 1 = DSP 2 = I2S 3 = SD 4 = MPEGE 5 = JPEGE 6 = EPP 7 = VI

HOST1X_SYNC_RDMA_BADDR3_0

Offset: 150h
Read/Write: R/W
Reset: 0000.0000

Buffer base address for port1

Bit	Description
25:2	**RDMA_BASE_ADDR3**

HOST1X_SYNC_RDMA_DMATRIGGER3_0

Offset: 154h
Read/Write: R/W
Reset: 0000.0000

NOTE: for these triggers to work, software needs to write the SRC_DSP field first! (0==CPU,1==DSP)

Bit	Description
4	**RDMA_SRC_DSP3: indicates writes to this register are triggered by DSP(1) or CPU(0)**
2	**RDMA_LAST_BUF3: last_buffer: indicates this is the last buffer in a chain**
1	**RDMA_BUF_RDY3: buffer_rdy : indicates buffer is pending in memory**
0	**RDMA_BUF_INIT3: buffer_init: initializes read DMA port**

HOST1X_SYNC_CBREAD0_0

Offset: 158h
Read/Write: RO
Reset: 0000.0000

Command buffer debug read. Will read the bottom of the command FIFO for all channels, including the register FIFO.

Bit	Description
31:0	**CBREAD0: Channel 0 command FIFO read**

HOST1X_SYNC_CBREAD1_0

Offset: 15ch
Read/Write: RO
Reset: 0000.0000

Bit	Description
31:0	**CBREAD1: Channel 1 command FIFO read**

HOST1X_SYNC_CBREAD2_0

Offset: 160h
Read/Write: RO
Reset: 0000.0000

Bit	Description
31:0	**CBREAD2: Channel 2 command FIFO read**

HOST1X_SYNC_CBREAD3_0

Offset: 164h
Read/Write: RO
Reset: 0000.0000

Bit	Description
31:0	**CBREAD3: Channel 3 command FIFO read**

HOST1X_SYNC_CBREAD4_0

Offset: 168h
Read/Write: RO
Reset: 0000.0000

Bit	Description
31:0	**CBREAD4: Channel 4 command FIFO read**

HOST1X_SYNC_CBREAD5_0

Offset: 16ch
Read/Write: RO
Reset: 0000.0000

Bit	Description
31:0	**CBREAD5: Channel 5 command FIFO read**

HOST1X_SYNC_CBREAD6_0

Offset: 170h
Read/Write: RO
Reset: 0000.0000

Bit	Description
31:0	**CBREAD6: Channel 6 command FIFO read**

HOST1X_SYNC_CBREAD7_0

Offset: 174h
Read/Write: RO
Reset: 0000.0000

Bit	Description
31:0	**CBREAD7: Channel 7 command FIFO read**

HOST1X_SYNC_REGF_DATA_0

Offset: 178h
Read/Write: RO
Reset: 0000.0000

Register FIFO debug read. Will read the bottom of the register FIFO, including all offset and control signals.

Bit	Description
31:0	**REGF_DATA: Register FIFO data read**

HOST1X_SYNC_REGF_ADDR_0

Offset: 17ch
Read/Write: RO
Reset: 0000.0000

Bit	Description
28	**REGF_RWN: Register FIFO read/write signal read.** 0 = Write 1 = Read
27:20	**REGF_MODULEID: Register FIFO module ID read.** 0 = HOST1X 1 = GRMPD 2 = ME 3 = JPEGE 4 = VI 5 = EPP 6 = ISP 7 = DSP 8 = GR2D 9 = GR3D 10 = DISPLAY 11 = I2S 12 = IC 13 = SD 14 = MC 15 = EMC
19:16	**REGF_BE: Register FIFO byte enables**
15:0	**REGF_OFFSET: Register FIFO offset read**

HOST1X_SYNC_WAITOVR_0

Offset: 180h
Read/Write: R/W
Reset: 0000.0000

Wait override. When written to '1', steps past a single stuck wait in the command buffer. Needs to be written back to '0' between overrides.

Bit	Description
7	**WAITOVR7: Channel 7 WAIT override.** 0 = Disable 1 = Enable
6	**WAITOVR6: Channel 6 WAIT override.** 0 = Disable 1 = Enable
5	**WAITOVR5: Channel 5 WAIT override.** 0 = Disable 1 = Enable
4	**WAITOVR4: Channel 4 WAIT override.** 0 = Disable 1 = Enable
3	**WAITOVR3: Channel 3 WAIT override.** 0 = Disable 1 = Enable
2	**WAITOVR2: Channel 2 WAIT override.** 0 = Disable 1 = Enable
1	**WAITOVR1: Channel 1 WAIT override.** 0 = Disable 1 = Enable
0	**WAITOVR0: Channel 0 WAIT override.** 0 = Disable 1 = Enable

HOST1X_SYNC_G3D0_STATE_0

Offset: 184h
Read/Write: RO
Reset: 0000.0000

GATHER3D state registers. Readable in case software needs to do context switching within a channel.

Bit	Description
19:18	**G3D0_PRIM: Primitive type**
17:16	**G3D0_VTXSIZE: Vertex size**
15:8	**G3D0_COUNT: Count of all words**
7:0	**G3D0_IDXOFF: Index offset**

HOST1X_SYNC_G3D0_ADDR0_0

Offset: 188h
Read/Write: RO
Reset: 0000.0000

Bit	Description
31:8	**G3D0_ADDRESS0: Base address**
7:4	**G3D0_STRIDE0: Encoded stride**
3:0	**G3D0_WORDS0: Words to gather**

HOST1X_SYNC_G3D0_ADDR1_0

Offset: 18ch
Read/Write: RO
Reset: 0000.0000

Bit	Description
31:8	**G3D0_ADDRESS1: Base address**
7:4	**G3D0_STRIDE1: Encoded stride**
3:0	**G3D0_WORDS1: Words to gather**

HOST1X_SYNC_G3D0_ADDR2_0

Offset: 190h
Read/Write: RO
Reset: 0000.0000

Bit	Description
31:8	**G3D0_ADDRESS2: Base address**
7:4	**G3D0_STRIDE2: Encoded stride**
3:0	**G3D0_WORDS2: Words to gather**

HOST1X_SYNC_G3D0_ADDR3_0

Offset: 194h
Read/Write: RO
Reset: 0000.0000

Bit	Description
31:8	**G3D0_ADDRESS3: Base address**
7:4	**G3D0_STRIDE3: Encoded stride**
3:0	**G3D0_WORDS3: Words to gather**

HOST1X_SYNC_G3D0_ADDR4_0

Offset: 198h
Read/Write: RO
Reset: 0000.0000

Bit	Description
31:8	**G3D0_ADDRESS4: Base address**
7:4	**G3D0_STRIDE4: Encoded stride**
3:0	**G3D0_WORDS4: Words to gather**

HOST1X_SYNC_G3D0_ADDR5_0

Offset: 19ch
Read/Write: RO
Reset: 0000.0000

Bit	Description
31:8	**G3D0_ADDRESS5: Base address**
7:4	**G3D0_STRIDE5: Encoded stride**
3:0	**G3D0_WORDS5: Words to gather**

HOST1X_SYNC_G3D0_ADDR6_0

Offset: 1a0h
Read/Write: RO
Reset: 0000.0000

Bit	Description
31:8	**G3D0_ADDRESS6: Base address**
7:4	**G3D0_STRIDE6: Encoded stride**
3:0	**G3D0_WORDS6: Words to gather**

HOST1X_SYNC_G3D0_ADDR7_0

Offset: 1a4h
Read/Write: RO
Reset: 0000.0000

Bit	Description
31:8	**G3D0_ADDRESS7: Base address**
7:4	**G3D0_STRIDE7: Encoded stride**
3:0	**G3D0_WORDS7: Words to gather**

HOST1X_SYNC_G3D1_STATE_0

Offset: 1a8h
Read/Write: RO
Reset: 0000.0000

Bit	Description
19:18	**G3D1_PRIM: Primitive type**
17:16	**G3D1_VTXSIZE: Vertex size**
15:8	**G3D1_COUNT: Count of all words**
7:0	**G3D1_IDXOFF: Index offset**

HOST1X_SYNC_G3D1_ADDR0_0

Offset: 1ach
Read/Write: RO
Reset: 0000.0000

Bit	Description
31:8	**G3D1_ADDRESS0: Base address**
7:4	**G3D1_STRIDE0: Encoded stride**
3:0	**G3D1_WORDS0: Words to gather**

HOST1X_SYNC_G3D1_ADDR1_0

Offset: 1b0h
Read/Write: RO
Reset: 0000.0000

Bit	Description
31:8	**G3D1_ADDRESS1: Base address**
7:4	**G3D1_STRIDE1: Encoded stride**
3:0	**G3D1_WORDS1: Words to gather**

HOST1X_SYNC_G3D1_ADDR2_0

Offset: 1b4h
Read/Write: RO
Reset: 0000.0000

Bit	Description
31:8	**G3D1_ADDRESS2: Base address**
7:4	**G3D1_STRIDE2: Encoded stride**
3:0	**G3D1_WORDS2: Words to gather**

HOST1X_SYNC_G3D1_ADDR3_0

Offset: 1b8h
Read/Write: RO
Reset: 0000.0000

Bit	Description
31:8	**G3D1_ADDRESS3: Base address**
7:4	**G3D1_STRIDE3: Encoded stride**
3:0	**G3D1_WORDS3: Words to gather**

HOST1X_SYNC_G3D1_ADDR4_0

Offset: 1bch
Read/Write: RO
Reset: 0000.0000

Bit	Description
31:8	**G3D1_ADDRESS4: Base address**
7:4	**G3D1_STRIDE4: Encoded stride**
3:0	**G3D1_WORDS4: Words to gather**

HOST1X_SYNC_G3D1_ADDR5_0

Offset: 1c0h
Read/Write: RO
Reset: 0000.0000

Bit	Description
31:8	**G3D1_ADDRESS5: Base address**
7:4	**G3D1_STRIDE5: Encoded stride**
3:0	**G3D1_WORDS5: Words to gather**

HOST1X_SYNC_G3D1_ADDR6_0

Offset: 1c4h
Read/Write: RO
Reset: 0000.0000

Bit	Description
31:8	**G3D1_ADDRESS6: Base address**
7:4	**G3D1_STRIDE6: Encoded stride**
3:0	**G3D1_WORDS6: Words to gather**

HOST1X_SYNC_G3D1_ADDR7_0

Offset: 1c8h
Read/Write: RO
Reset: 0000.0000

Bit	Description
31:8	**G3D1_ADDRESS7: Base address**
7:4	**G3D1_STRIDE7: Encoded stride**
3:0	**G3D1_WORDS7: Words to gather**

HOST1X_SYNC_G3D2_STATE_0

Offset: 1cch
Read/Write: RO
Reset: 0000.0000

Bit	Description
19:18	**G3D2_PRIM: Primitive type**
17:16	**G3D2_VTXSIZE: Vertex size**
15:8	**G3D2_COUNT: Count of all words**
7:0	**G3D2_IDXOFF: Index offset**

HOST1X_SYNC_G3D2_ADDR0_0

Offset: 1d0h
Read/Write: RO
Reset: 0000.0000

Bit	Description
31:8	**G3D2_ADDRESS0: Base address**
7:4	**G3D2_STRIDE0: Encoded stride**
3:0	**G3D2_WORDS0: Words to gather**

HOST1X_SYNC_G3D2_ADDR1_0

Offset: 1d4h
Read/Write: RO
Reset: 0000.0000

Bit	Description
31:8	**G3D2_ADDRESS1: Base address**
7:4	**G3D2_STRIDE1: Encoded stride**
3:0	**G3D2_WORDS1: Words to gather**

HOST1X_SYNC_G3D2_ADDR2_0

Offset: 1d8h
Read/Write: RO
Reset: 0000.0000

Bit	Description
31:8	**G3D2_ADDRESS2: Base address**
7:4	**G3D2_STRIDE2: Encoded stride**
3:0	**G3D2_WORDS2: Words to gather**

HOST1X_SYNC_G3D2_ADDR3_0

Offset: 1dch
Read/Write: RO
Reset: 0000.0000

Bit	Description
31:8	**G3D2_ADDRESS3: Base address**
7:4	**G3D2_STRIDE3: Encoded stride**
3:0	**G3D2_WORDS3: Words to gather**

HOST1X_SYNC_G3D2_ADDR4_0

Offset: 1e0h
Read/Write: RO
Reset: 0000.0000

Bit	Description
31:8	**G3D2_ADDRESS4: Base address**
7:4	**G3D2_STRIDE4: Encoded stride**
3:0	**G3D2_WORDS4: Words to gather**

HOST1X_SYNC_G3D2_ADDR5_0

Offset: 1e4h
Read/Write: RO
Reset: 0000.0000

Bit	Description
31:8	**G3D2_ADDRESS5: Base address**
7:4	**G3D2_STRIDE5: Encoded stride**
3:0	**G3D2_WORDS5: Words to gather**

HOST1X_SYNC_G3D2_ADDR6_0

Offset: 1e8h
Read/Write: RO
Reset: 0000.0000

Bit	Description
31:8	**G3D2_ADDRESS6: Base address**
7:4	**G3D2_STRIDE6: Encoded stride**
3:0	**G3D2_WORDS6: Words to gather**

HOST1X_SYNC_G3D2_ADDR7_0

Offset: 1ech
Read/Write: RO
Reset: 0000.0000

Bit	Description
31:8	**G3D2_ADDRESS7: Base address**
7:4	**G3D2_STRIDE7: Encoded stride**
3:0	**G3D2_WORDS7: Words to gather**

HOST1X_SYNC_G3D3_STATE_0

Offset: 1f0h
Read/Write: RO
Reset: 0000.0000

Bit	Description
19:18	**G3D3_PRIM: Primitive type**
17:16	**G3D3_VTXSIZE: Vertex size**
15:8	**G3D3_COUNT: Count of all words**
7:0	**G3D3_IDXOFF: Index offset**

HOST1X_SYNC_G3D3_ADDR0_0

Offset: 1f4h
Read/Write: RO
Reset: 0000.0000

Bit	Description
31:8	**G3D3_ADDRESS0: Base address**
7:4	**G3D3_STRIDE0: Encoded stride**
3:0	**G3D3_WORDS0: Words to gather**

HOST1X_SYNC_G3D3_ADDR1_0

Offset: 1f8h
Read/Write: RO
Reset: 0000.0000

Bit	Description
31:8	**G3D3_ADDRESS1: Base address**
7:4	**G3D3_STRIDE1: Encoded stride**
3:0	**G3D3_WORDS1: Words to gather**

HOST1X_SYNC_G3D3_ADDR2_0

Offset: 1fch
Read/Write: RO
Reset: 0000.0000

Bit	Description
31:8	**G3D3_ADDRESS2: Base address**
7:4	**G3D3_STRIDE2: Encoded stride**
3:0	**G3D3_WORDS2: Words to gather**

HOST1X_SYNC_G3D3_ADDR3_0

Offset: 200h
Read/Write: RO
Reset: 0000.0000

Bit	Description
31:8	**G3D3_ADDRESS3: Base address**
7:4	**G3D3_STRIDE3: Encoded stride**
3:0	**G3D3_WORDS3: Words to gather**

HOST1X_SYNC_G3D3_ADDR4_0

Offset: 204h
Read/Write: RO
Reset: 0000.0000

Bit	Description
31:8	**G3D3_ADDRESS4: Base address**
7:4	**G3D3_STRIDE4: Encoded stride**
3:0	**G3D3_WORDS4: Words to gather**

HOST1X_SYNC_G3D3_ADDR5_0

Offset: 208h
Read/Write: RO
Reset: 0000.0000

Bit	Description
31:8	**G3D3_ADDRESS5: Base address**
7:4	**G3D3_STRIDE5: Encoded stride**
3:0	**G3D3_WORDS5: Words to gather**

HOST1X_SYNC_G3D3_ADDR6_0

Offset: 20ch
Read/Write: RO
Reset: 0000.0000

Bit	Description
31:8	**G3D3_ADDRESS6: Base address**
7:4	**G3D3_STRIDE6: Encoded stride**
3:0	**G3D3_WORDS6: Words to gather**

HOST1X_SYNC_G3D3_ADDR7_0

Offset: 210h
Read/Write: RO
Reset: 0000.0000

Bit	Description
31:8	**G3D3_ADDRESS7: Base address**
7:4	**G3D3_STRIDE7: Encoded stride**
3:0	**G3D3_WORDS7: Words to gather**

HOST1X_SYNC_G3D4_STATE_0

Offset: 214h
Read/Write: RO
Reset: 0000.0000

Bit	Description
19:18	**G3D4_PRIM: Primitive type**
17:16	**G3D4_VTXSIZE: Vertex size**
15:8	**G3D4_COUNT: Count of all words**
7:0	**G3D4_IDXOFF: Index offset**

HOST1X_SYNC_G3D4_ADDR0_0

Offset: 218h
Read/Write: RO
Reset: 0000.0000

Bit	Description
31:8	**G3D4_ADDRESS0: Base address**
7:4	**G3D4_STRIDE0: Encoded stride**
3:0	**G3D4_WORDS0: Words to gather**

HOST1X_SYNC_G3D4_ADDR1_0

Offset: 21ch
Read/Write: RO
Reset: 0000.0000

Bit	Description
31:8	**G3D4_ADDRESS1: Base address**
7:4	**G3D4_STRIDE1: Encoded stride**
3:0	**G3D4_WORDS1: Words to gather**

HOST1X_SYNC_G3D4_ADDR2_0

Offset: 220h
Read/Write: RO
Reset: 0000.0000

Bit	Description
31:8	**G3D4_ADDRESS2: Base address**
7:4	**G3D4_STRIDE2: Encoded stride**
3:0	**G3D4_WORDS2: Words to gather**

HOST1X_SYNC_G3D4_ADDR3_0

Offset: 224h
Read/Write: RO
Reset: 0000.0000

Bit	Description
31:8	**G3D4_ADDRESS3: Base address**
7:4	**G3D4_STRIDE3: Encoded stride**
3:0	**G3D4_WORDS3: Words to gather**

HOST1X_SYNC_G3D4_ADDR4_0

Offset: 228h
Read/Write: RO
Reset: 0000.0000

Bit	Description
31:8	**G3D4_ADDRESS4: Base address**
7:4	**G3D4_STRIDE4: Encoded stride**
3:0	**G3D4_WORDS4: Words to gather**

HOST1X_SYNC_G3D4_ADDR5_0

Offset: 22ch
Read/Write: RO
Reset: 0000.0000

Bit	Description
31:8	**G3D4_ADDRESS5: Base address**
7:4	**G3D4_STRIDE5: Encoded stride**
3:0	**G3D4_WORDS5: Words to gather**

HOST1X_SYNC_G3D4_ADDR6_0

Offset: 230h
Read/Write: RO
Reset: 0000.0000

Bit	Description
31:8	**G3D4_ADDRESS6: Base address**
7:4	**G3D4_STRIDE6: Encoded stride**
3:0	**G3D4_WORDS6: Words to gather**

HOST1X_SYNC_G3D4_ADDR7_0

Offset: 234h
Read/Write: RO
Reset: 0000.0000

Bit	Description
31:8	**G3D4_ADDRESS7: Base address**
7:4	**G3D4_STRIDE7: Encoded stride**
3:0	**G3D4_WORDS7: Words to gather**

HOST1X_SYNC_G3D5_STATE_0

Offset: 238h
Read/Write: RO
Reset: 0000.0000

Bit	Description
19:18	**G3D5_PRIM: Primitive type**
17:16	**G3D5_VTXSIZE: Vertex size**
15:8	**G3D5_COUNT: Count of all words**
7:0	**G3D5_IDXOFF: Index offset**

HOST1X_SYNC_G3D5_ADDR0_0

Offset: 23ch
Read/Write: RO
Reset: 0000.0000

Bit	Description
31:8	**G3D5_ADDRESS0: Base address**
7:4	**G3D5_STRIDE0: Encoded stride**
3:0	**G3D5_WORDS0: Words to gather**

HOST1X_SYNC_G3D5_ADDR1_0

Offset: 240h
Read/Write: RO
Reset: 0000.0000

Bit	Description
31:8	**G3D5_ADDRESS1: Base address**
7:4	**G3D5_STRIDE1: Encoded stride**
3:0	**G3D5_WORDS1: Words to gather**

HOST1X_SYNC_G3D5_ADDR2_0

Offset: 244h
Read/Write: RO
Reset: 0000.0000

Bit	Description
31:8	**G3D5_ADDRESS2: Base address**
7:4	**G3D5_STRIDE2: Encoded stride**
3:0	**G3D5_WORDS2: Words to gather**

HOST1X_SYNC_G3D5_ADDR3_0

Offset: 248h
Read/Write: RO
Reset: 0000.0000

Bit	Description
31:8	**G3D5_ADDRESS3: Base address**
7:4	**G3D5_STRIDE3: Encoded stride**
3:0	**G3D5_WORDS3: Words to gather**

HOST1X_SYNC_G3D5_ADDR4_0

Offset: 24ch
Read/Write: RO
Reset: 0000.0000

Bit	Description
31:8	**G3D5_ADDRESS4: Base address**
7:4	**G3D5_STRIDE4: Encoded stride**
3:0	**G3D5_WORDS4: Words to gather**

HOST1X_SYNC_G3D5_ADDR5_0

Offset: 250h
Read/Write: RO
Reset: 0000.0000

Bit	Description
31:8	**G3D5_ADDRESS5: Base address**
7:4	**G3D5_STRIDE5: Encoded stride**
3:0	**G3D5_WORDS5: Words to gather**

HOST1X_SYNC_G3D5_ADDR6_0

Offset: 254h
Read/Write: RO
Reset: 0000.0000

Bit	Description
31:8	**G3D5_ADDRESS6: Base address**
7:4	**G3D5_STRIDE6: Encoded stride**
3:0	**G3D5_WORDS6: Words to gather**

HOST1X_SYNC_G3D5_ADDR7_0

Offset: 258h
Read/Write: RO
Reset: 0000.0000

Bit	Description
31:8	**G3D5_ADDRESS7: Base address**
7:4	**G3D5_STRIDE7: Encoded stride**
3:0	**G3D5_WORDS7: Words to gather**

HOST1X_SYNC_G3D6_STATE_0

Offset: 25ch
Read/Write: RO
Reset: 0000.0000

Bit	Description
19:18	**G3D6_PRIM: Primitive type**
17:16	**G3D6_VTXSIZE: Vertex size**
15:8	**G3D6_COUNT: Count of all words**
7:0	**G3D6_IDXOFF: Index offset**

HOST1X_SYNC_G3D6_ADDR0_0

Offset: 260h
Read/Write: RO
Reset: 0000.0000

Bit	Description
31:8	**G3D6_ADDRESS0: Base address**
7:4	**G3D6_STRIDE0: Encoded stride**
3:0	**G3D6_WORDS0: Words to gather**

HOST1X_SYNC_G3D6_ADDR1_0

Offset: 264h
Read/Write: RO
Reset: 0000.0000

Bit	Description
31:8	**G3D6_ADDRESS1: Base address**
7:4	**G3D6_STRIDE1: Encoded stride**
3:0	**G3D6_WORDS1: Words to gather**

HOST1X_SYNC_G3D6_ADDR2_0

Offset: 268h
Read/Write: RO
Reset: 0000.0000

Bit	Description
31:8	**G3D6_ADDRESS2: Base address**
7:4	**G3D6_STRIDE2: Encoded stride**
3:0	**G3D6_WORDS2: Words to gather**

HOST1X_SYNC_G3D6_ADDR3_0

Offset: 26ch
Read/Write: RO
Reset: 0000.0000

Bit	Description
31:8	**G3D6_ADDRESS3: Base address**
7:4	**G3D6_STRIDE3: Encoded stride**
3:0	**G3D6_WORDS3: Words to gather**

HOST1X_SYNC_G3D6_ADDR4_0

Offset: 270h
Read/Write: RO
Reset: 0000.0000

Bit	Description
31:8	**G3D6_ADDRESS4: Base address**
7:4	**G3D6_STRIDE4: Encoded stride**
3:0	**G3D6_WORDS4: Words to gather**

HOST1X_SYNC_G3D6_ADDR5_0

Offset: 274h
Read/Write: RO
Reset: 0000.0000

Bit	Description
31:8	**G3D6_ADDRESS5: Base address**
7:4	**G3D6_STRIDE5: Encoded stride**
3:0	**G3D6_WORDS5: Words to gather**

HOST1X_SYNC_G3D6_ADDR6_0

Offset: 278h
Read/Write: RO
Reset: 0000.0000

Bit	Description
31:8	**G3D6_ADDRESS6: Base address**
7:4	**G3D6_STRIDE6: Encoded stride**
3:0	**G3D6_WORDS6: Words to gather**

HOST1X_SYNC_G3D6_ADDR7_0

Offset: 27ch
Read/Write: RO
Reset: 0000.0000

Bit	Description
31:8	**G3D6_ADDRESS7: Base address**
7:4	**G3D6_STRIDE7: Encoded stride**
3:0	**G3D6_WORDS7: Words to gather**

HOST1X_SYNC_G3D7_STATE_0

Offset: 280h
Read/Write: RO
Reset: 0000.0000

Bit	Description
19:18	**G3D7_PRIM: Primitive type**
17:16	**G3D7_VTXSIZE: Vertex size**
15:8	**G3D7_COUNT: Count of all words**
7:0	**G3D7_IDXOFF: Index offset**

HOST1X_SYNC_G3D7_ADDR0_0

Offset: 284h
Read/Write: RO
Reset: 0000.0000

Bit	Description
31:8	**G3D7_ADDRESS0: Base address**
7:4	**G3D7_STRIDE0: Encoded stride**
3:0	**G3D7_WORDS0: Words to gather**

HOST1X_SYNC_G3D7_ADDR1_0

Offset: 288h
Read/Write: RO
Reset: 0000.0000

Bit	Description
31:8	**G3D7_ADDRESS1: Base address**
7:4	**G3D7_STRIDE1: Encoded stride**
3:0	**G3D7_WORDS1: Words to gather**

HOST1X_SYNC_G3D7_ADDR2_0

Offset: 28ch
Read/Write: RO
Reset: 0000.0000

Bit	Description
31:8	**G3D7_ADDRESS2: Base address**
7:4	**G3D7_STRIDE2: Encoded stride**
3:0	**G3D7_WORDS2: Words to gather**

HOST1X_SYNC_G3D7_ADDR3_0

Offset: 290h
Read/Write: RO
Reset: 0000.0000

Bit	Description
31:8	**G3D7_ADDRESS3: Base address**
7:4	**G3D7_STRIDE3: Encoded stride**
3:0	**G3D7_WORDS3: Words to gather**

HOST1X_SYNC_G3D7_ADDR4_0

Offset: 294h
Read/Write: RO
Reset: 0000.0000

Bit	Description
31:8	**G3D7_ADDRESS4: Base address**
7:4	**G3D7_STRIDE4: Encoded stride**
3:0	**G3D7_WORDS4: Words to gather**

HOST1X_SYNC_G3D7_ADDR5_0

Offset: 298h
Read/Write: RO
Reset: 0000.0000

Bit	Description
31:8	**G3D7_ADDRESS5: Base address**
7:4	**G3D7_STRIDE5: Encoded stride**
3:0	**G3D7_WORDS5: Words to gather**

HOST1X_SYNC_G3D7_ADDR6_0

Offset: 29ch
Read/Write: RO
Reset: 0000.0000

Bit	Description
31:8	**G3D7_ADDRESS6: Base address**
7:4	**G3D7_STRIDE6: Encoded stride**
3:0	**G3D7_WORDS6: Words to gather**

HOST1X_SYNC_G3D7_ADDR7_0

Offset: 2a0h
Read/Write: RO
Reset: 0000.0000

Bit	Description
31:8	**G3D7_ADDRESS7: Base address**
7:4	**G3D7_STRIDE7: Encoded stride**
3:0	**G3D7_WORDS7: Words to gather**

HOST1X_SYNC_MCCIF_THCTRL_0

Offset: 2a4h
Read/Write: R/W
Reset: 0000.0000

Memory write client FIFO status.

Reads out the available 128-bit entries. If writing through the buffered frame buffer write region, writes are accumulated up to 128 bits before being flushed, so 10 entries can mean up to 40 writes.

Memory client high-priority threshold control register. Sets the threshold of when the client becomes high-priority.

Bit	Description
13:8	**CSW_HOST1XW2MC_HPTH**
5:0	**CSW_HOST1XW_FIFOSTAT**

HOST1X_SYNC_HC_MCCIF_FIFOCTRL_0

Offset: 2a8h
Read/Write: R/W
Reset: 0000.0000

Memory Client Interface Async Fifo Optimization Register

Memory Client Interface FIFO Control Register.

The registers below allow to optimize the synchronization timing in the memory client asynchronous fifos. When they can be used depend on the client and memory controller clock ratio.

Additionally, the RDMC_RDFAST/RDCL_RDFAST fields can increase power consumption if the asynchronous fifo is implemented as a real RAM. There is no power impact on latch-based fifos. Flipflop-based fifos do not use these fields.

Important: See recommended settings below.

The register fields can only be changed when the memory client async FIFOs are empty.

The register field ending with WRCL_MCLE2X (if any) can be set to improve async fifo synchronization on the write side by one client clock cycle if the memory controller clock frequency is less or equal to twice the client clock frequency:

mcclk_freq <= 2 * clientclk_freq

The register field ending with WRMC_CLLE2X (if any) can be set to improve async fifo synchronization on the write side by one memory controller clock cycle if the client clock frequency is less or equal to twice the memory controller clock frequency:

clientclk_freq <= 2 * mcclk_freq

The register field ending with RDMC_RDFAST (if any) can be set to improve async fifo synchronization on the read side by one memory controller clock cycle.

Warning: RDMC_RDFAST can be used along with WRCL_MCLE2X only when

mcclk_freq <= clientclk_freq

The register field ending with RDCL_RDFAST (if any) can be set to improve async fifo synchronization on the read side by one client clock cycle.

Warning: RDCL_RDFAST can be used along with WRMC_CLLE2X only when:

clientclk_freq <= mcclk_freq

RECOMMENDED SETTINGS

Client writing to fifo, memory controller reading from fifo

- mcclk_freq <= clientclk_freq

You can enable both RDMC_RDFAST and WRCL_CLLE2X. If one of the fifos is a real RAM and power is a concern, you should avoid enabling RDMC_RDFAST.

- clientclk_freq < mcclk_freq <= 2 * clientclk_freq

You can enable RDMC_RDFAST or WRCL_MCLE2X, but because the client clock is slower, you should enable only WRCL_MCLE2X.

- 2 * clientclk_freq < mcclk_freq

You can only enable RDMC_RDFAST. If one of the fifos is a real RAM and power is a concern, you should avoid enabling RDMC_RDFAST.

Memory controller writing to fifo, client reading from fifo

- clientclk_freq <= mcclk_freq

You can enable both RDCL_RDFAST and WRMC_CLLE2X. If one of the fifos is a real RAM and power is a concern, you should avoid enabling RDCL_RDFAST.

- mcclk_freq < clientclk_freq <= 2 * mcclk_freq

You can enable RDCL_RDFAST or WRMC_CLLE2X, but because the memory controller clock is slower, you should enable only WRMC_CLLE2X.

- 2 * mcclk_freq < clientclk_freq

You can only enable RDCL_RDFAST. If one of the fifos is a real RAM and power is a concern, you should avoid enabling RDCL_RDFAST.

Bit	Description
3	**HC_MCCIF_RDCL_RDFAST:** 0 = Disable 1 = Enable
2	**HC_MCCIF_WRMC_CLLE2X:** 0 = Disable 1 = Enable
1	**HC_MCCIF_RDMC_RDFAST:** 0 = Disable 1 = Enable
0	**HC_MCCIF_WRCL_MCLE2X:** 0 = Disable 1 = Enable

7.2 Host Microclass Registers

NV_CLASS_HOST_CLEAR_0

Offset: 000h
Read/Write: R/W
Reset: 0000.0000

Clear method. Any bits set in VECTOR will be cleared in the channel's RAISE vector.

Bit	Description
31:0	**VECTOR**

NV_CLASS_HOST_WAIT_0

Offset: 001h
Read/Write: R/W
Reset: 0000.0000

Wait method. Command dispatch will stall until any of the bits set in VECTOR become set in the channel's RAISE vector.

Bit	Description
31:0	**VECTOR**

NV_CLASS_HOST_WAIT_WITH_INTR_0

Offset: 002h
Read/Write: R/W
Reset: 0000.0000

Wait w/ interrupt method. Identical to the WAIT method except an interrupt will be triggered when the WAIT requirement is satisfied.

Bit	Description
31:0	**VECTOR**

NV_CLASS_HOST_DELAY_USEC_0

Offset: 003h
Read/Write: R/W
Reset: 0000.0000

Delay number of microseconds. Command dispatch will stall until the number of microseconds indicated in NUSEC has passed. The timing of microseconds is controlled by the USEC_CLK register.

Bit	Description
19:0	**NUSEC: Enough for 1.05 seconds**

NV_CLASS_HOST_INDOFF_0

Offset: 020h
Read/Write: R/W
Reset: 0000.0000

Indirect address

This register (along with INDDATA) is used to indirectly read/write either register or memory. Host registers are not accessible using this interface. If AUTOINC is set, INDOFFSET increments by 4 on every access of INDDATA.

Bit	Description
31:28	**INDBE: Byte enables.** Will apply to all subsequent data transactions. Not applicable for reads.
27	**AUTOINC: Auto increment of read/write address.** 0 = Disable 1 = Enable
25:18	**INDMODID: ACCTYPE=REG: register module ID.** 0 = HOST1X 1 = GRMPD 2 = ME 3 = JPEGE 4 = VI 5 = EPP 6 = ISP 7 = DSP 8 = GR2D 9 = GR3D 10 = DISPLAY 11 = I2S 12 = IC 13 = SD 14 = MC 15 = EMC
25:2	**INDOFFSET: ACCTYPE=FB: frame buffer address**
17:2	**INDROFFSET: ACCTYPE=REG: register offset ([15:0])**
1	**ACCTYPE: Access type: indirect register or indirect frame buffer.** 0 = REG 1 = FB
0	**RWN: Read/write.** 0 = WRITE 1 = READ

NV_CLASS_HOST_INDDATA_0

Offset: 021h..03fh
Read/Write: R/W
Reset: 0000.0000

This is an array of 31 identical register entries; the register fields below apply to each entry. These registers, when written, either writes to the data to the INDOFFSET in INDOFF or triggers a read of the offset at INDOFFSET.

Bit	Description
31:0	**INDDATA: read or write data**

7.3 I2S Registers

I2S_CTXSW_0

Offset: 000h
Read/Write: R/W
Reset: fffff000.f000

Contains temporary buffer related parameters

Context switch register. Should be common to all modules. Includes the current channel/class (which is writable by SW) and the next channel/class (which the hardware sets when it receives a context switch).

Context switch works like this:

Any context switch request triggers an interrupt to the host and causes the new channel/class to be stored in NEXT_CHANNEL/NEXT_CLASS (see vmod/chexample). SW sees that there is a context switch interrupt and does the necessary operations to make the module ready to receive traffic from the new context. It clears the context switch interrupt and writes CURR_CHANNEL/CLASS to the same value as NEXT_CHANNEL/CLASS, which causes a context switch acknowledge packet to be sent to the host. This completes the context switch and allows the host to continue sending data to the module.

Bit	Description
31:28	**NEXT_CHANNEL: Next requested channel**
25:16	**NEXT_CLASS: Next requested class**
15:12	**CURR_CHANNEL: Current working channel, reset to 'invalid'**
9:0	**CURR_CLASS: Current working class**

I2S_CLK_FSYNC_CNTRL_0

Offset: 001h
Read/Write: R/W
Reset: 0000.0002

FSYNC and Sclk control
FSYNC Control

Bit	Description
14	**FSYNCIN_SAMPLED: FSYNC Input Source select.** 0 = FSYNC_FALL_EDGE 1 = FSYNC_RISING_EDGE
13	**FSYNC_STOPVAL: FSYNC stop value.** 0 = Low 1 = Hi
11:10	**FSYNC_PLS: FSYNC pulse type.** 0 = Even_duty 1 = Once 2 = Twice 3 = Low

Bit	Description
9:7	**FSYNC_DIV: FSYNC divider control.** 0 = FP_4 1 = FP_8 2 = FP_16 3 = FP_32 4 = FP_64 5 = FP_128 6 = DIV_265 7 = FSYNC_6
6:5	**FSYNC_FP: FSYNC predivider. Parameter is used to generate FSYNC.** 0 = FP_DIV1 1 = FP_DIV2 2 = FP_DIV3 3 = FP_DIV5
4	**FSYNC_POL: FSYNC polarity (determines the active edge of FSYNC)..** 0 = Falling_edge 1 = Rising_edge
2	**SOUT_DLY: Serial data out delay (1 SCLK cycle) .** 0 = No_delay 1 = Delay
1	**FSYNC_DIR: Sets FSYNC clock direction to input or output** 0 = Input 1 = Output

I2S_FSYNC_ENB_CNTRL_0

Offset: 002h
Read/Write: R/W
Reset: 0000.0000

FSYNC enable and edge select

Bit	Description
2	**FSYNC_STS_EDGE: FSYNC status edge select** Select which edge of FSYNC is used to set FSYNC_STS. 0 = set when falling edge is detected. In AC97 mode, falling edge signals start of tag phase. (Use this for AC97) 1 = set when rising edge is detected
0	**FSYNC_EN: Enables FSYNC pulse** 0 = FSYNC stopped, and will be held at value of FSYNC_STOPVAL 1 = FSYNC enabled

I2S_RCV_SRC_SEL_0

Offset: 003h
Read/Write: R/W
Reset: 0000.0000

Receive Source Select
SIN Control

Bit	Description
1:0	**SERIN_SRC_SEL: Serial data source select.** 0 = SIN_FALL_EDGE 1 = SIN_PAD 2 = SOUT 3 = SOUT_PAD

I2S_AC97_CMDSTS_CNTRL_0

Offset: 004h
Read/Write: R/W
Reset: 0000.0000

AC97 Command/Status Control
AC97 Control

Bit	Description
5	**STS_RCVED: Status Received** Set every time status is received from Codec. It is cleared every time AC97_STS register is read. 0 = Status has not been received since last read of AC97_STS. 1 = New Status has not been received since last read of AC97_STS.
4	**CMD_BUSY: Command Busy** 0 = no pending command transmission. Next command can be sent. 1 = pending command/status transmission. No command/status should be sent until current transmission is completed.
0	**CMDSTS_EN: AC97 Command/Status Enable** Also enable clock to transmit/receive block. 0 = AC97 Command/Status logic disabled. Clear command/status transmit/receive bits (CMDSTS_BUSY and CMDSTS_RCVED) 1 = AC97 Command/Status logic enabled.

I2S_AC97_CMD_0

Offset: 005h
Read/Write: R/W
Reset: 0000.0000

AC97 Command Data/Address Write

Bit	Description
31:16	**CMD_WDATA: AC97 Command Data** Data written to this field gets transmitted during slot 2 in bit positions 19-4.
15:8	**CMD_WADDR: AC97 Command Address** Data written to this field gets transmitted during slot 1 in bit positions 19-12. CMD_WADDR[23] is the Command Read/Write command, and the remaining bits is the control register index.
7:0	**CMD_WADDR_PAD: AC97 Command Address Pad** These bit should be programmed to 0 for normal operation.

I2S_AC97_STS_0

Offset: 006h
Read/Write: RO
Reset: 0000.0000

AC97 Status Data/Address Read

Bit	Description
31:16	**STS_RDATA: AC97 Status Data** Data read from this field comes from slot2 bit positions 19-4.
15:8	**STS_RADDR: AC97 Status Address** Data read from this field comes from slot1 bit positions 19-12.

I2S_TRANSMIT_CNTRL_0

Offset: 007h
Read/Write: R/W
Reset: 0000.0000

Transmit Control

Bit	Description
22	**SLOT_REQ_LORR: Slot Request Send Data Mode (AC97 mode only) .** 0 = SEND_BOTH 1 = SEND_EITHER
21	**VAR_SMPLE_RATE: Transmit Variable Sample Rate Enable (AC97 mode only)** Transmit data only when previous frame had either, or both, left/right SLOTREQ bits active. SLOTREQ bits are sampled during Slot1, and are valid regardless of the valid bits in tag slot. Below shows the bit position in Slot1 corresponding to active slots. Slot1 Bit Position Slot 11 3 10 4 9 5 8 6 7 7 6 8 5 9 0 = disabled 1 = enabled
20:19	**TRM_SLOT_SEL: Transmit Slot Select (AC97 mode only).** 0 = USE_SLOT_3_4 1 = USE_SLOT_5_6 2 = USE_SLOT_7_8 3 = USE_SLOT_6_9
17:16	**TRM_RATE_CNTRL: Transmit Sample Rate Control (non-Ac97 mode only).** 0 = Normal 1 = Repeat_2 2 = Repeat_4 3 = Repeat_6
14:12	**TRM_DMODE: Transmit Data Mode (non-AC97 mode only).** 0 = ONE_AFTER_FSYNC 3 = TWO_AFTER_FSYNC 4 = ONE_AFTER_FRAME 5 = FOUT_AFTER_FSYNC 7 = CONTINUOUS_AFTER_FRAME
10	**TRM_MONO: Transmit Mono Audio.** 0 = STEREO 1 = MONO
8:6	**TRM_SIZE: Transmit width.** 0 = TRMSIZE_8 1 = TRMSIZE_16 2 = TRMSIZE_18 3 = TRMSIZE_20 4 = TRMSIZE_24 5 = TRMSIZE_32

I2S_TRM_DATA_PAD_0

Offset: 008h
Read/Write: R/W
Reset: 0000.0000

Transmit Data Pad

Bit	Description
31:0	**TDATA_PAD: Used to output additional data, and should only be changed when transmitter is disabled.**

I2S_TRM_BUF_START_ADDR_0

Offset: 009h
Read/Write: R/W
Reset: 0000.0000

Transmit (Read) Buffer Start Address (bits [1:0] are not used)

Bit	Description
25:0	**TRMBUF_ADDR**

I2S_TRM_BUF_NUM_0

Offset: 00ah
Read/Write: R/W
Reset: 0000.0000

Number of transmit (read) buffers

Bit	Description
8:0	**TRMBUF_NUM**

I2S_TRM_BUF_CONFIG_0

Offset: 00bh
Read/Write: R/W
Reset: 0000.0000

RDMA buffer configuration

Bit	Description
2	**TRMBUF_WRAP**
1	**TRMBUF_STOREHDR**
0	**TRMBUF_EN: Enables and initializes transmit.**

I2S_TRM_BUF_CNTRL_0

Offset: 00ch
Read/Write: RO
Reset: 0000.0000

DMA enable, buffer related signals

Bit	Description
1	**TRMBUF_LASTBUF: Marks last buffer. Use with TRMBUF_RDY.**
0	**TRMBUF_RDY: Write-one to mark buffer in memory ready to transmit.**

I2S_TRM_BUFS_INT_LIMITS_0

Offset: 00dh
Read/Write: R/W
Reset: 0000.0000

Buffer low and high limit values for TRM_BUFS_LOW and TRM_BUFS_HIGH interrupts.

Bit	Description
24:16	**TRM_BUFS_HIGH: The TRM_BUFS_HIGH_STS status bit is set when the TRM buffers available climbs above this value.**
8:0	**TRM_BUFS_LOW: The TRM_BUFS_LOW_STS status bit is set when the TRM buffers available count falls below this value.**

I2S_TRM_DMA_STATUS_0

Offset: 00eh
Read/Write: RO
Reset: 0000.0000

The status vector for the read buffer.

Bit	Description
31	**TRMBUF_WRAPPED: Wrapped to base.**
28:25	**TRMBUF_NUMENTRIES: Pending entries in data fifo.**
24:16	**TRMBUF_NUMBUFFERS: Pending buffers in memory.**
15:0	**TRMBUF_NUMBYTES: Pending bytes in current buffer.**

I2S_RECEIVE_CNTRL_0

Offset: 00fh
Read/Write: R/W
Reset: 0000.0000

Receive Control

Bit	Description
27:24	**CHAR_TIMEOUT_FCNT: Character Timeout Frame Count (AC97 mode only)** Number of consecutive frames when no left/right data is received for timeout to occur. Generates status interrupt CHAR_TIMEOUT_STS. Valid values range from 0 (no character timeout) to 15.
20:19	**RCV_SLOT_SEL: Receive Slot Select (AC97 mode only).** 0 = USE_SLOT_3_4 1 = USE_SLOT_5_6 2 = USE_SLOT_7_8 3 = USE_SLOT_6_9
17:16	**RCV_RATE_CNTRL: Receive Rate Control.** 0 = ALL_SAMPLE 1 = HALF_SAMPLE 2 = QUARTER_SAMPLE 3 = SIXTH_SAMPLE
15	**EARLY_PUSH: Early Push Enable (non-AC97 mode only)** Set if number of clocks in frame period is expected to be less than what is needed to transfer receive data. Receive data in bit positions that could not be received because of short frame period will be left with zeroes. Effective in all receive data modes.
14:12	**RCV_DMODE: Receive Data Mode (non-AC97 mode only).** 0 = ONE_AFTER_FSYNC 3 = TWO_AFTER_FSYNC 4 = ONE_AFTER_FRAME 5 = FOUT_AFTER_FSYNC 7 = CONTINUOUS_AFTER_FRAME
11	**RCV_MONO_LR: Receive Mono Left/Right Select.** 0 = Left 1 = Right
10	**RCV_MONO: Receive Mono mode.** 0 = Stereo 1 = Mono
8:6	**RCV_SIZE: Receive width.** 0 = RCVSIZE_8 1 = RCVSIZE_16 2 = RCVSIZE_18 3 = RCVSIZE_20 4 = RCVSIZE_24 5 = RCVSIZE_32
3	**CHAR_TIMEOUT_FLUSH: Character Timeout Flush enable** Generate a fifo flush to get remaining fifo data written to memory. 0 = character timeout will not generate a flush 1 = character timeout will generate a flush
2	**STOP_RCV: Stop Receive** If receive fifo is not empty, flush is generated such that all data in fifo is written to memory. 0 = normal operation 1 = receive operation stopped after current frame

I2S_RCV_BUF_START_ADDR_0

Offset: 010h
Read/Write: R/W
Reset: 0000.0000

Receive (Write) Buffer Start Address (bits [1:0] are not used)

Bit	Description
25:0	**RCVBUF_ADDR**

I2S_RCV_BUF_SIZE_0

Offset: 011h
Read/Write: R/W
Reset: 0000.0000

Receive (Write) Buffer Size (bits [1:0] are not used)

Bit	Description
15:0	**RCVBUF_SIZE**

I2S_RCV_BUF_NUM_0

Offset: 012h
Read/Write: R/W
Reset: 0000.0000

Number of receive (write) buffers

Bit	Description
8:0	**RCVBUF_NUM**

I2S_RCV_BUF_CONFIG_0

Offset: 013h
Read/Write: R/W
Reset: 0000.0000

Bit	Description
0	**RCVBUF_EN: DMA enable.**

I2S_RCV_BUF_CNTRL_0

Offset: 014h
Read/Write: RO
Reset: 0000.0000

DSP direct receive buffer access controls (when not using host RDMA).

Bit	Description
0	**RCV_BUFFER_READ: Write 1 to let receive know that buffer is available to be written again.**

I2S_RCV_BUFS_INT_LIMITS_0

Offset: 015h
Read/Write: R/W
Reset: 0000.0000

Buffer low and high limit values for RCV_BUFS_LOW and RCV_BUFS_HIGH interrupts.

Bit	Description
24:16	**RCV_BUFS_HIGH: The RCV_BUFS_HIGH_STS status bit is set when the RCV buffers filled climbs above this value.**
8:0	**RCV_BUFS_LOW: The RCV_BUFS_LOW_STS status bit is set when the RCV buffers filled count falls below this value.**

I2S_RCV_DMA_STATUS0_0

Offset: 016h
Read/Write: RO
Reset: 0000.0000

Receive buffer status

Bit	Description
31	**RCVBUF_WRAPPED: Wrapped to base.**
30	**RCVBUF_FULL: Buffer chain is full.**
24:16	**RCVBUF_BUFINDEX: Current buffer index.**
15:0	**RCVBUF_NUMBYTES: Number of bytes in current buffer.**

I2S_RCV_DMA_STATUS1_0

Offset: 017h
Read/Write: RO
Reset: 0000.0000

Receive buffer status

Bit	Description
8:0	**RCVBUF_PENDING: Pending buffers in memory.**

I2S_TRM_HEADER_RAISE_CNTRL_0

Offset: 018h
Read/Write: R/W
Reset: 0000.0000

Raise field in transmit buffer header to be reported as raise events.

Bit	Description
31	**TRM_HDR_RAISE_EN**

I2S_RCV_RAISE_CNTRL_0

Offset: 019h
Read/Write: R/W
Reset: 0000.0000

When written by CPU, raise event will be sent on next complete receive buffer.

Bit	Description
19:16	**RCV_BUF_CHANNEL**
4:0	**RCV_BUF_RAISEVAL**

I2S_RCV_HEADER_CNTRL_0

Offset: 01ah
Read/Write: R/W
Reset: 0000.0000

Header field in receive buffer headers will be set to these values.

Bit	Description
31	**RCV_HDR_RAISE_EN**
19:16	**RCV_HDR_CHANNEL**
4:0	**RCV_HDR_RAISEVAL**

I2S_CMD_TRM_RAISE_CNTRL_0

Offset: 01bh
Read/Write: R/W
Reset: 0000.0000

When written by CPU, raise event will be sent when the next AC'97 command completes.

Bit	Description
19:16	**CMD_TRM_CHANNEL**
4:0	**CMD_TRM_RAISEVAL**

I2S_INTERRUPT_MASK_0

Offset: 01ch
Read/Write: R/W
Reset: 0000.0000

Interrupt Source Enabled

Bit	Description
16	**RFIFO_OVERFLOW_INTE: Receive Fifo overflow interrupt enable** 0 = no interrupt 1 = generate interrupt
15	**RCOMPLETE_INTE: Receive Complete interrupt enable** Receive complete is signaled when the last receive buffer is written to memory. 0 = no interrupt 1 = generate interrupt
14	**RCV_BUF_COMPLETE_INTE: Receive Buffer Complete interrupt enable** Effective when receiver is enabled. Occurs when each buffer ready for host RDMA. 0 = no interrupt 1 = generate interrupt
13	**RCV_BUFS_HIGH_INTE: Receive buffers high water mark interrupt enable** 0 = no interrupt 1 = generate interrupt
12	**RCV_BUFS_LOW_INTE: Receive buffers low water mark interrupt enable** 0 = no interrupt 1 = generate interrupt
11	**TFIFO_EMPTY_INTE: Transmit Fifo Out of Data interrupt enable** Generates an interrupt when transmit fifo is empty but transfer is not complete. 0 = no interrupt 1 = generate interrupt
10	**TCOMPLETE_INTE: Transmit Complete interrupt enable** Transmit complete is signaled when the DMA transmit is completed and transmit fifo runs out of data. 0 = no interrupt 1 = generate interrupt
9	**TRM_BUF_COMPLETE_INTE: Transmite Buffer Complete interrupt enable** Effective when transmitter is enabled. Occurs when raise field in buffer header is set and complete buffer has been transmitted. 0 = no interrupt 1 = generate interrupt
8	**TRM_BUFS_HIGH_INTE: Transmit buffers high water mark interrupt enable** Effective when transmitter is enabled. 0 = no interrupt 1 = generate interrupt
7	**TRM_BUFS_LOW_INTE: Transmit buffers low water mark interrupt enable** Effective when transmitter is enabled. 0 = no interrupt 1 = generate interrupt
6	**CHAR_TIMEOUT_INTE: AC97 Character Time out interrupt enable** Effective when receiver is enabled. Time out generated when no PCM left/right data is received within number of time out frames programmed in CHAR_TIMEOUT_FCNT 0 = no interrupt 1 = generate interrupt
5	**FSYNC_EDGE_INTE: FSYNC Edge interrupt enable** Generate an interrupt when an FSYNC edge is detected. FSYNC_STS_EDGE determines whether active edge or inactive edge is used to generate the interrupt. 0 = no interrupt 1 = generate interrupt

Bit	Description
4	**FSYNC_STOP_INTE: FSYNC Stopped interrupt enable** Effective only if FSYNC is generated internally, when FSYNC is stopped by clearing FSYNC_EN. 0 = no interrupt 1 = generate interrupt
3	**CMD_TRM_INTE: AC97 Command transmitted interrupt enable** 0 = no interrupt 1 = generate interrupt
2	**STS_RCVED_INTE: AC97 Status Received interrupt enable** 0 = no interrupt 1 = generate interrupt
1	**CRDY_INTE: AC97 Primary Codec Ready interrupt enable** 0 = no interrupt 1 = generate interrupt
0	**CTXSW_INTE: Context Switch interrupt Status**

I2S_INTSTATUS_0

Offset: 01dh
Read/Write: RO
Reset: 0000.0000

Interrupt Status

Bit	Description
16	**RFIFO_OVERFLOW_INT: Receive Fifo overflow Status** 0 = no status 1 = status detected
15	**RCOMPLETE_INT: Receive Complete interrupt enable** Receive complete is signaled when the last receive buffer is written to memory. 0 = no status 1 = status detected
14	**RCV_BUF_COMPLETE_INT: Receive Buffer Complete Status** Effective when receiver is enabled. Occurs when each buffer ready for host RDMA. 0 = no status 1 = status detected
13	**RCV_BUFS_HIGH_INT: Receive buffers high water mark Status** 0 = no status 1 = status detected
12	**RCV_BUFS_LOW_INT: Receive buffers low water mark Status** 0 = no status 1 = status detected
11	**TFIFO_EMPTY_INT: Transmit Fifo Out of Data Status** Set when transmit fifo is empty but transfer is not complete. 0 = no status 1 = status detected
10	**TCOMPLETE_INT: Transmit Complete Status** Transmit complete is signaled when the DMA transmit is completed and transmit fifo runs out of data. 0 = no status 1 = status detected
9	**TRM_BUF_COMPLETE_INT: Transmit Buffer Complete Status** Effective when transmitter is enabled. Occurs when raise field in buffer header is set and complete buffer has been transmitted. 0 = no status 1 = status detected

Bit	Description
8	**TRM_BUFS_HIGH_INT: Transmit buffers high water mark Status** Effective when transmitter is enabled. 0 = no status 1 = status detected
7	**TRM_BUFS_LOW_INT: Transmit buffers low water mark Status** Effective when transmitter is enabled. 0 = no status 1 = status detected
6	**CHAR_TIMEOUT_INT: AC97 Character Time out Status** Effective when receiver is enabled. Time out generated when no PCM left/right data is received within number of time out frames programmed in CHAR_TIMEOUT_FCNT 0 = no status 1 = status detected
5	**FSYNC_EDGE_INT: FSYNC Edge Status** Set when an FSYNC edge is detected. FSYNC_STS_EDGE determines whether active edge or inactive edge is used to generate the status. 0 = no status 1 = status detected
4	**FSYNC_STOP_INT: FSYNC Stopped Status** Effective only if FSYNC is generated internally, when FSYNC is stopped by clearing FSYNC_EN. 0 = no status 1 = status detected
3	**CMD_TRM_INT: AC97 Command transmitted Status** 0 = no status 1 = status detected
2	**STS_RCVED_INT: AC97 Status Received Status** 0 = no status 1 = status detected
1	**CRDY_INT: AC97 Primary Codec Ready Status** 0 = no status 1 = status detected
0	**CTXSW_INT: Context Switch interrupt Status** To allow for software polling, interrupt status bits will be set when their interrupting conditions occur, even if their corresponding interrupt enable bits are cleared. The status bits can be cleared by writing 1 to bit positions.

I2S_I2S_CLOCK_EN_0

Offset: 01eh
Read/Write: R/W
Reset: 0000.0000

I2S Clock Enable
GPIO Pins

Bit	Description
0	**SMCLK_EN: SMCLK Clock Enable** When clock is stopped, logic will wait till root clock goes low before holding it at low value (avoid generating short pulses). 0 = SMCLK stopped (held low) 1 = SMCLK enabled

I2S_GPIO_PIN_CNTRL_0

Offset: 01fh
Read/Write: R/W
Reset: 0000.0000

GPIO Pin Control

Bit	Description
21	**GPIO5_OUT_SEL: GPIO5 Output Select.** 0 = SOUT 1 = GPIO_5
20	**GPIO4_OUT_SEL: GPIO4 Output Select.** 0 = SIN 1 = GPIO_4
19	**GPIO3_OUT_SEL: GPIO3 Output Select.** 0 = FSYNC 1 = GPIO_3
18	**GPIO2_OUT_SEL: GPIO2 Output Select.** 0 = SCLK 1 = GPIO_2
17	**GPIO1_OUT_SEL: GPIO1 Output Select.** 0 = SMCLK 1 = GPIO_1
16	**GPIO0_OUT_SEL: GPIO0 Output Select.** 0 = SRCLK 1 = GPIO_0
11:10	**GPIO5_PIN_CNTRL: Controls input/output of GPIO5 (SOUT).** 0 = Input disabled, output disabled 1 = Input enabled, output disabled 2 = Input disabled, output enabled 3 = Input enabled, output enabled
9:8	**GPIO4_PIN_CNTRL: Controls input/output of GPIO4 (SIN).** 0 = Input disabled, output disabled 1 = Input enabled, output disabled 2 = Input disabled, output enabled 3 = Input enabled, output enabled
7:6	**GPIO3_PIN_CNTRL: Controls input/output of GPIO3 (FSYNC).** 0 = Input disabled, output disabled 1 = Input enabled, output disabled 2 = Input disabled, output enabled 3 = Input enabled, output enabled
5:4	**GPIO2_PIN_CNTRL: Controls input/output of GPIO2 (SCLK).** 0 = Input disabled, output disabled 1 = Input enabled, output disabled 2 = Input disabled, output enabled 3 = Input enabled, output enabled
3:2	**GPIO1_PIN_CNTRL: Controls input/output of GPIO1 (SMCLK).** 0 = Input disabled, output disabled 1 = Input enabled, output disabled 2 = Input disabled, output enabled 3 = Input enabled, output enabled
1:0	**GPIO0_PIN_CNTRL: Controls input/output of GPIO0 (SRCLK).** 0 = Input disabled, output disabled 1 = Input enabled, output disabled 2 = Input disabled, output enabled 3 = Input enabled, output enabled

I2S_GPIO_IN_OUT_DATA_0

Offset: 020h
Read/Write: R/W
Reset: 0000.0000

GPIO Input/Output Data

Bit	Description
21	**GPIO5_OUT: GPIO5 Output Data**
20	**GPIO4_OUT: GPIO4 Output Data**
19	**GPIO3_OUT: GPIO3 Output Data**
18	**GPIO2_OUT: GPIO2 Output Data**
17	**GPIO1_OUT: GPIO1 Output Data**
16	**GPIO0_OUT: GPIO0 Output Data**
5	**GPIO5_IN: GPIO5 Input Data**
4	**GPIO4_IN: GPIO4 Input Data**
3	**GPIO3_IN: GPIO3 Input Data**
2	**GPIO2_IN: GPIO2 Input Data**
1	**GPIO1_IN: GPIO1 Input Data**
0	**GPIO0_IN: GPIO0 Input Data**

I2S_I2SR_MCCIF_FIFOCTRL_0

Offset: 023h
Read/Write: R/W
Reset: 0000.0000

Memory Client Interface Async Fifo Optimization Register Memory Client Interface Fifo Control Register. The registers below allow to optimize the synchronization timing in the memory client asynchronous fifos. When they can be used depend on the client and memory controller clock ratio.

Additionally, the RDMC_RDFAST/RDCL_RDFAST fields can increase power consumption if the asynchronous fifo is implemented as a real RAM. There is no power impact on latch-based fifos. Flipflop-based fifos do not use these fields. See recommended settings below.

Important: The register fields can only be changed when the memory client async fifos are empty.

The register field ending with WRCL_MCLE2X (if any) can be set to improve async fifo synchronization on the write side by one client clock cycle if the memory controller clock frequency is less or equal to twice the client clock frequency:

mcclk_freq <= 2 * clientclk_freq

The register field ending with WRMC_CLLE2X (if any) can be set to improve async fifo synchronization on the write side by one memory controller clock cycle if the client clock frequency is less or equal to twice the memory controller clock frequency:

clientclk_freq <= 2 * mcclk_freq

The register field ending with RDMC_RDFAST (if any) can be set to improve async fifo synchronization on the read side by one memory controller clock cycle.

Warning: RDMC_RDFAST can be used along with WRCL_MCLE2X only when:

mcclk_freq <= clientclk_freq

The register field ending with RDCL_RDFAST (if any) can be set to improve async fifo synchronization on the read side by one client clock cycle.

Warning: RDCL_RDFAST can be used along with WRMC_CLLE2X only when:

clientclk_freq <= mcclk_freq

RECOMMENDED SETTINGS

Client writing to fifo, memory controller reading from fifo

- mcclk_freq <= clientclk_freq

You can enable both RDMC_RDFAST and WRCL_CLLE2X. If one of the fifos is a real RAM and power is a concern, you should avoid enabling RDMC_RDFAST.

- clientclk_freq < mcclk_freq <= 2 * clientclk_freq

You can enable RDMC_RDFAST or WRCL_MCLE2X, but because the client clock is slower, you should enable only WRCL_MCLE2X.

- 2 * clientclk_freq < mcclk_freq

You can only enable RDMC_RDFAST. If one of the fifos is a real RAM and power is a concern, you should avoid enabling RDMC_RDFAST.

Memory controller writing to fifo, client reading from fifo

- clientclk_freq <= mcclk_freq

You can enable both RDCL_RDFAST and WRMC_CLLE2X. If one of the fifos is a real RAM and power is a concern, you should avoid enabling RDCL_RDFAST.

- mcclk_freq < clientclk_freq <= 2 * mcclk_freq

You can enable RDCL_RDFAST or WRMC_CLLE2X, but because the memory controller clock is slower, you should enable only WRMC_CLLE2X.

- 2 * mcclk_freq < clientclk_freq

You can only enable RDCL_RDFAST. If one of the fifos is a real RAM and power is a concern, you should avoid enabling RDCL_RDFAST.

Bit	Description
1	**I2SR_MCCIF_RDMC_RDFAST:** 0 = Disable 1 = Enable
0	**I2SR_MCCIF_WRCL_MCLE2X:** 0 = Disable 1 = Enable

I2S_I2ST_MCCIF_FIFOCTRL_0

Offset: 024h
Read/Write: R/W
Reset: 0000.0000

Bit	Description
3	**I2ST_MCCIF_RDCL_RDFAST:** 0 = Disable 1 = Enable
2	**I2ST_MCCIF_WRMC_CLLE2X:** 0 = Disable 1 = Enable
1	**I2ST_MCCIF_RDMC_RDFAST:** 0 = Disable 1 = Enable
0	**I2ST_MCCIF_WRCL_MCLE2X:** 0 = Disable 1 = Enable

7.4 SPB Registers

IC_CTXSW_0

Offset: 000h
Read/Write: R/W
Reset: fffff000.f000

Context switch register. Should be common to all modules.

Includes the current channel/class (which is writable by SW) and the next channel/class (which the hardware sets when it receives a context switch).

Context switch works like this:

Any context switch request triggers an interrupt to the host and causes the new channel/class to be stored in NEXT_CHANNEL/NEXT_CLASS (see vmod/chexample). SW sees that there is a context switch interrupt and does the necessary operations to make the module ready to receive traffic from the new context. It clears the context switch interrupt and writes CURR_CHANNEL/CLASS to the same value as NEXT_CHANNEL/CLASS, which causes a context switch acknowledge packet to be sent to the host. This completes the context switch and allows the host to continue sending data to the module.

Bit	Description
31:28	**NEXT_CHANNEL: Next requested channel**
25:16	**NEXT_CLASS: Next requested class**
15:12	**CURR_CHANNEL: Current working channel, reset to 'invalid'**
9:0	**CURR_CLASS: Current working class**

IC_STOPSTART_WAIT_0

Offset: 001h
Read/Write: R/W
Reset: 0000.0000

Wait count between consecutive STOP/START

Bit	Description
7:1	**SPSTWAITTIME: the Philips I2C bus spec defines the bus free time (tBUG) between STOP and START conditions to be 4.7us in STANDARD-MODE and 1.3us is FAST-MODE**
0	**SPSTWAITEN:** 0 = DISABLE_WAIT_STATE 1 = EN_WAIT_STATE

IC_IC_CONFIG_0

Offset: 002h
Read/Write: R/W
Reset: 0000.0000

Bit	Description
30	**LACK:** 0 = NO_ACK 1 = ACK
29:28	**THDCTL: enum (STD_TIMING=0x0, ADD_1_CLK=0x1, ADD_1_CLK=02)** This bit controls the behavior of i2c in Fast Mode. If the operating mode is set to F/S and the device is on a mixed speed bus, this bit will control the hold time of the device to conform to Standard Mode timing. 00 No Change 01 Add one clock to hold time 10 Add two clocks to hold time 11 Reserved
8	**IC_RESP_TO_NEXT:** 0 = CONTINUE_SENDING 1 = NEXT_CMD
7	**CKSYEN:** 0 = CLK_SYNC_EN 1 = CLK_SYNC_DISABLED
6	**CMD_CMB:** 0 = NO_COMBO 1 = COMBO
2	**ICCFSMRST:** 0 = XFER_FIFO_ACTIVE 1 = XFER_FIFO_RST
1	**ICFRST:** 0 = RD_FIFO_ACTIVE 1 = RD_FIFO_RST
0	**ICEN:** 0 = RST 1 = ACTIVE

IC_RESP_TIMEOUT_0

Offset: 003h
Read/Write: R/W
Reset: 0fff.ffff

Response time out Counter Control

If Ack is not received, start the time out counter and enter error state if time out occurs with no ack after retrying if i2c times out and enters error state, Software will reset everything and restart from the fail point

Bit	Description
31:0	**RESP_TO**

IC_TCOMMAND_0

Offset: 004h
Read/Write: R/W
Reset: 0000.0000

31:0 rw CMD i=0x0

This is the address register used in I2C master transfers. These parameters pass through the command manager/FIFO circuitry.

Bit	Description
30:28	**GEN_CALL_CTL:** 0 = USE_ADDR_REG 1 = RESERVED1 2 = RESERVED2 3 = GENERAL_CALL
25:16	**XFER_CNT: 25:16 transfer count** This field holds and initiates the number of bytes for a transfer. The transfer is complete when the "transfer done" bit is set. This field differs from the Transfer Count Status in that it holds the expected byte count. Use the Transfer Count Status IC04R[21:12] to get the current transfer status. The Transfer Count should be programmed to be the number of bytes to be transferred minus one. This means the smallest possible transfer is one byte or a register setting of zero.
15	**XFER_SLAVE_MODE:** 0 = STD_MODE 1 = ADDR_ONLY
13:12	**SPEED_CTL:** 0 = STD_MODE 1 = FAST_MODE 2 = RSVD1 3 = RSVD3
11	**ADDR_MOD:** 0 = SEVEN_BIT 1 = TEN_BIT
10:1	**ADDR: 10:1 address for r/w access** this field holds the address for the i2c access. if the 7/10 bit addressing mode is set to 7-bit access, only bits 7:1 are valid. The master will always send data to the address specified in this register.
0	**RW_CTL:** 0 = WRITE 1 = READ

IC_TWDATA_0

Offset: 005h
Read/Write: R/W
Reset: 0000.0000

Transfer write data

Bit	Description
31:0	**tdata: This register is used to write the data for the I2C master write access.** The master will always transfer data to the I2C address specified in the address register.

IC_TRDATA_0

Offset: 006h
Read/Write: RO
Reset: 0000.0000

Transfer read data

Bit	Description
31:0	**ICRFFDATA: This register is used to read the data for the I2C master read access.** The master will always transfer data from the I2C address specified in the address register.

IC_TRDATA_POP_0

Offset: 007h
Read/Write: WO
Reset: 0000.0000

transfer read data through class path

Bit	Description
0	**IC_R_DATA_POP: When setting this bit to 1'b1, the receive data will be popped off of the fifo and pushed into the host read fifo.** This enables synchronization of reads through the host command fifo.

IC_TFSTATUS_0

Offset: 008h
Read/Write: RO
Reset: 0000.0000

Transfer fifo status

This register contains control bits that provide I2C master transfer status. This is a read only register.

Bit	Description
28:25	**ICCFFWSTS: command fifo status** These bits contain the write status of the command FIFO. During programmed I/O, the status may be used to tell the host when to write to the FIFO.
23:14	**CUR_CNTSTATUS: This field holds the number of bytes remaining to transfer.** When transfer has completed, the Transfer Done bit is set. This field differs from the Transfer Count in that it reflects the current byte count.
9:7	**ICRFFSTS: read fifo status** This bit contains the status of the I2C read FIFO in master-receive mode. This status refers to the read side of the FIFO used during transfer of data from an I2C peripheral to the host. During programmed I/O, the status may be used to tell the host when to read from the FIFO.

Bit	Description
3:0	**ICTFFSTS: write fifo status** This bit contains the status of the master write FIFO in master-transmit mode. This status refers to the write side of the FIFO used during transfer of data from the host to an I2C peripheral. During programmed I/O, the status may be used to tell the host when to write to the FIFO. 0000 FIFO is empty. 0001 1 FIFO slot is filled. 0010 2 FIFO slots are filled. 1000 8 FIFO slots are filled.

IC_CSTATUS_0

Offset: 009h
Read/Write: RO
Reset: 0000.0000

Command status

This register contains control bits that provide I2C master transfer status. This is a read only register.

Bit	Description
24	**CMD_DONE: transfer done.** 0 = XFER_IN_PROGRESS 1 = XFER_DONE
11	**IC_CONT_RD_SHIFTSTART: receive complete.** 0 = XFER_IN_PROGRESS 1 = XFER_DONE
5	**IC_CONT_WR_SHIFTSTART: transmit complete.** 0 = XFER_IN_PROGRESS 1 = XFER_DONE
1	**IC_CONT_TDONE_R: transmit transfer done** 0 master transmit transfer is incomplete or not initiated 1 master transmit transfer is complete
0	**IC_CONT_RDONE_R: receive transfer done** 0 master receive transfer is incomplete or not initiated 1 master receive transfer ic complete

IC_INTMASK_0

Offset: 00ah
Read/Write: R/W
Reset: 0000.0000

Interrupt mask for I2C.

Bit	Description
8	**TO_INT_EN**
7	**CTXSW_INT_EN**
6	**CHF_INT_EN**
5	**RHF_INT_EN**

Bit	Description
4	**WHF_INT_EN**
3	**CF_INT_EN**
2	**TC_INT_EN**
1	**RF_INT_EN**
0	**WF_INT_EN**

IC_INTSTATUS_0

Offset: 00bh
Read/Write: RO
Reset: 0000.0000

This register contains control bits to provide interrupt status.

An interrupt may be cleared by writing a "1" to the corresponding status bit in IC_INTERRUPT_CLEAR followed immediately with a "0" to the same bit.

Bit	Description
8	**RESP_TO_INT: time out interrupt** as requested by SW, I2C will always flag an interrupt if the response time out is reached. There is no interrupt mask/enable for this interrupt
7	**CTXSW_INT: context switch**
6	**CHMPTINT_R: command fifo is half full**
5	**RHMPTINT_R: master read fifo is half full**
4	**THMPTINT_R: master write fifo is half empty**
3	**CMPTINT_R: command fifo is empty**
2	**T_COMPLETE_R: enables generation of interrupts when the transfer has completed**
1	**RMPTINT_R: master read fifo is full**
0	**TMPTINT_R: master write fifo is empty**

IC_RAISE_CFIFO_EMPTY_0

Offset: 00ch
Read/Write: R/W
Reset: 0000.0000

Bit	Description
19:16	**RAISE_CFE_CHANNEL**
4:0	**RAISE_CFE_VALUE**

IC_RAISE_TFIFO_EMPTY_0

Offset: 00dh
Read/Write: R/W
Reset: 0000.0000

Bit	Description
19:16	**RAISE_TFE_CHANNEL**
4:0	**RAISE_TFE_VALUE**

IC_RAISE_RFIFO_EMPTY_0

Offset: 00eh
Read/Write: R/W
Reset: 0000.0000

Bit	Description
19:16	**RAISE_RFE_CHANNEL**
4:0	**RAISE_RFE_VALUE**

IC_RAISE_CFIFO_HALFEMPTY_0

Offset: 00fh
Read/Write: R/W
Reset: 0000.0000

Bit	Description
19:16	**RAISE_CFHE_CHANNEL**
4:0	**RAISE_CFHE_VALUE**

IC_RAISE_TFIFO_HALFEMPTY_0

Offset: 010h
Read/Write: R/W
Reset: 0000.0000

Bit	Description
19:16	**RAISE_TFHE_CHANNEL**
4:0	**RAISE_TFHE_VALUE**

IC_RAISE_RFIFO_HALFEMPTY_0

Offset: 011h
Read/Write: R/W
Reset: 0000.0000

Bit	Description
19:16	**RAISE_RFHE_CHANNEL**
4:0	**RAISE_RFHE_VALUE**

IC_RAISE_CMD_DONE_0

Offset: 012h
Read/Write: R/W
Reset: 0000.0000

Bit	Description
19:16	**RAISE_CMD_DONE_CHANNEL**
4:0	**RAISE_CMD_DONE_VALUE**

IC_RAISE_RDONE_0

Offset: 013h
Read/Write: R/W
Reset: 0000.0000

Bit	Description
19:16	**RAISE_RDONE_CHANNEL**
4:0	**RAISE_RDONE_VALUE**

IC_RAISE_TDONE_0

Offset: 014h
Read/Write: R/W
Reset: 0000.0000

Bit	Description
19:16	**RAISE_TDONE_CHANNEL**
4:0	**RAISE_TDONE_VALUE**

IC_REFCOUNT_0

Offset: 015h
Read/Write: R/W
Reset: 0000.0000

Bit	Description
15:0	**REF_COUNT_VAL**

ARIC_BFM_DEVICE_0

Offset: 1000h
Read/Write: R/W
Reset: 0000.0000

Bit	Description
22	**SUBADDRESS_TEST_MODE:** 0 = OFF 1 = ON
21	**BUS_TIMING_CHECK_SPEED:** 0 = STD 1 = FAST
20	**BUS_TIMING_CHECKS:** 0 = OFF 1 = ON
19	**GENERAL_CALL_RESPONSE:** 0 = OFF 1 = ON
18	**TIMEOUT_ENABLE:** 0 = OFF 1 = ON
17	**ADDRESS_MODE:** 0 = _7_BIT 1 = _10_BIT
16	**SUBADDRESS_MODE:** 0 = OFF 1 = ON
9:0	**DEVICE_ADDRESS**

ARIC_BFM_TIMEOUT_0

Offset: 1001h
Read/Write: R/W
Reset: 0000.0010

Bit	Description
15:0	**TIMEOUT_DELAY**

ARIC_BFM_STATUS_0

Offset: 1002h
Read/Write: RO
Reset: 0000.0000

Bit	Description
1	**TIMING_ERROR**
0	**STATUS_IDLE**

ARIC_BFM_MEMORY_INDIRECT_ADDRESS_0

Offset: 1003h
Read/Write: R/W
Reset: 0000.0000

Bit	Description
7:2	**INDIRECT_ADDRESS**

ARIC_BFM_MEMORY_INDIRECT_DATA_WRITE_0

Offset: 1004h
Read/Write: WO
Reset: 0000.0000

Bit	Description
31:0	**INDIRECT_DATA_WRITE**

ARIC_BFM_MEMORY_INDIRECT_DATA_READ_0

Offset: 1005h
Read/Write: RO
Reset: 0000.0000

Bit	Description
31:0	**INDIRECT_DATA_READ**

ARIC_BFM_MEMORY_INDIRECT_CTL_0

Offset: 1006h
Read/Write: R/W
Reset: 0000.0000

Bit	Description
3	**INDIRECT_READY**
2	**INDIRECT_READ**
1	**INDIRECT_WRITE**
0	**INDIRECT_ENABLE**

7.5 MC Registers

MC_CTXSW_0

Offset: 000h
Read/Write: R/W
Reset: fffff000.f000

Module IDs for use in enums

Channel IDs

Context switch register. Should be common to all modules. Includes the current channel/class (which is writable by SW) and the next channel/class (which the hardware sets when it receives a context switch).

Context switch works like this:

Any context switch request triggers an interrupt to the host and causes the new channel/class to be stored in NEXT_CHANNEL/NEXT_CLASS (see vmod/chexample). SW sees that there is a context switch interrupt and does the necessary operations to make the module ready to receive traffic from the new context. It clears the context switch interrupt and writes CURR_CHANNEL/CLASS to the same value as NEXT_CHANNEL/CLASS, which causes a context switch acknowledge packet to be sent to the host. This completes the context switch and allows the host to continue sending data to the module.

Bit	Description
31:28	**NEXT_CHANNEL: Next requested channel**
25:16	**NEXT_CLASS: Next requested class**
15:12	**CURR_CHANNEL: Current working channel, reset to 'invalid'**
9:0	**CURR_CLASS: Current working class**

MC_INTSTATUS_0

Offset: 001h
Read/Write: RO
Reset: 0000.0000

Interrupt Status Register. Clear on 1-write. Init value is clear.

Bit	Description
2	**EMEM_BADADR_INT: External Memory Bad address.** 0 = CLEAR 1 = SET
1	**IMEM_BADADR_INT: Internal Memory Bad address.** 0 = CLEAR 1 = SET
0	**CTXSW_INT: Context Switch.** 0 = CLEAR 1 = SET

MC_INTMASK_0

Offset: 002h
Read/Write: R/W
Reset: 0000.0000

Interrupt Mask Register. Init value is masked.

Bit	Description
2	**EMEM_BADADR_INTMASK: External Memory Bad address.** 0 = MASKED 1 = UNMASKED
1	**IMEM_BADADR_INTMASK: Internal Memory Bad address.** 0 = MASKED 1 = UNMASKED
0	**CTXSW_INTMASK: Context Switch.** 0 = MASKED 1 = UNMASKED

MC_IMEM_CFG_0

Offset: 003h
Read/Write: R/W
Reset: 0002.0280

Internal memory config Register

Internal memory size (in KBytes): This is used to check for out-of-bound accesses.
Internal memory latency. Init value is 2. This register defines the number of memory cycles from when the memory controller sends a read request to the internal memory, till the data is available for capture.

With the default SRAM, the control signals are registered inside the SRAM block before driving the SRAM array. The read data is registered inside the SRAM wrapper because the read latency is close to 3/4th of the memory clock period. Hence the default init value of 2 (cycles). The maximum value is defined by NV_MC_SRAM_MAXLATENCY.

Bit	Description
18:16	**IMEM_LATENCY**
15:0	**IMEM_SIZE_KB**

MC_EMEM_CFG_0

Offset: 004h
Read/Write: R/W
Reset: 0000.2000

External memory config Register.

External memory size (in KBytes): This is used to check for out-of-bound accesses.

External memory address translation config: The config value indicates how the translation from linear address to bank/row/column is performed.

The ri1/ri2/ri3/ri4 bits are defined in project_ar_mc.spec.

For GoForce 5500:

Number of bank/row/column bits and dram page size for each config:
- CFG0 -> 2 bank bits, 8 column bits, up to 13 row bits, dram page size is 1 KByte.
- CFG1 -> 2 bank bits, 9 column bits, up to 12 row bits, dram page size is 2 KBytes.
- CFG2 -> 1 bank bit, 7 column bits, up to 13 row bits, dram page size is 0.5 KByte.
- CFG3 -> 1 bank bit, 7 column bits, up to 13 row bits, dram page size is 0.5 KByte.
- CFG6 -> 1 bank bit, 8 column bits, up to 12 row bits, dram page size is 1 KByte.
- CFG7 -> 1 bank bit, 8 column bits, up to 12 row bits, dram page size is 1 KByte.

Definitions of ri1/ri2/ri3/ri4 bits:
- ri1 = (adr[..:12] == prev_adr[..:12])
- ri2 = adr[6]
- ri3 = adr[7]
- ri4 = (adr[11:10] == prev_adr[11:10])

```
ri1 ri4 ri3 ri2
|---------------|-----| | |
adr ..... 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
A3 A2 A1 A0 - -
L1 ..... r2 r1 r0 A7 A6 A5 A4 B1 B0 SDR Mobile Elpida 8 MBytes
L2 ..... r2 r1 r0 A7 A6 A5 A4 B1 B0 SDR Mobile Micron 16 MBytes
L3 ..... r1 r0 A8 A7 A6 A5 A4 B1 B0 SDR Mobile Micron 32 MBytes
L4 ..... r4 r3 r2 r1 r0 A6 A5 A4 B0 SDR Inapac 2/4 MBytes
L5 ..... r4 r3 r2 r1 r0 A6 A5 B0 A4 SDR Inapac 2/4 MBytes
L6 ..... r2 r1 r0 A7 A6 A5 A4 B1 B0 SDR Mobile Samsung 8/16 MBytes
L7 ..... r1 r0 A8 A7 A6 A5 A4 B1 B0 SDR Mobile Samsung 32 MBytes
L8 ..... r4 r3 r2 r1 r0 A6 A5 A4 B0 SDR Etron 2 MBytes
L9 ..... r4 r3 r2 r1 r0 A6 A5 B0 A4 SDR Etron 2 MBytes

L10 ..... r2 r1 r0 A7 A6 A5 A4 B1 B0 DDR Mobile Elpida 32 MBytes
L11 ..... r2 r1 r0 A7 A6 A5 A4 B1 B0 DDR Inapac 8 MBytes
L12 ..... r2 r1 r0 A7 A6 A5 A4 B1 B0 DDR Mobile Samsung 32 MBytes

L13 ..... r3 r2 r1 r0 A7 A6 A5 A4 B0 SDR Etron 8 MBytes
L14 ..... r3 r2 r1 r0 A7 A6 A5 B0 A4 SDR Etron 8 MBytes
```

SDRAM address configuration:
- CFG0 -> L1/L2/L6/L10/L11/L12 B0=ri2, B1=ri3, SamePage= ri1 & (B0==prev_B0) & (B1==prev_B1)
- CFG1 -> L3/L7 B0=ri2, B1=ri3, SamePage= ri1 & (B0==prev_B0) & (B1==prev_B1)
- CFG2 -> L4/L8 B0=ri2, B1=0, SamePage= ri1 & (B0==prev_B0) & ri4
- CFG3 -> L5/L9 B0=ri3, B1=0, SamePage= ri1 & (B0==prev_B0) & ri4
- CFG6 -> L13 B0=ri2, B1=0, SamePage= ri1 & (B0==prev_B0) & ri4
- CFG7 -> L14 B0=ri3, B1=0, SamePage= ri1 & (B0==prev_B0) & ri4

With CFG6/CFG7, we use ri4 in the same page calculation. Therefore the calculation is as if the dram page size was twice smaller (0.5 KByte instead of 1 KByte). That is the same page calculation is pessimistic.

With CFG0, we change bank every 2^6 = 64 Bytes, we change page every 2^12= 4 KBytes.
With CFG1, we change bank every 2^6 = 64 Bytes, we change page every 2^13= 8 KBytes.
With CFG2, we change bank every 2^6 = 64 Bytes, we change page every 2^10= 1 KByte.
With CFG3, we change bank every 2^7 = 128 Bytes, we change page every 2^10= 1 KByte.
With CFG6, we change bank every 2^6 = 64 Bytes, we change page every 2^10= 1 KByte.
With CFG7, we change bank every 2^7 = 128 Bytes, we change page every 2^10= 1 KByte.

Again here with CFG6/CFG7, we should change page every 2^11= 2 KBytes, but because the same page calculation is pessimistic, we consider that we change page every 1 KByte. That is the same page calculation is pessimistic.

CFG4/CFG5 are not defined yet.

Reserved fields.
A few reserved fields in case we need a SW accessible register for an ECO.

Bit	Description
23	**EMEM_CFG_RESERVED4:** 0 = DISABLED 1 = ENABLED
22	**EMEM_CFG_RESERVED3:** 0 = DISABLED 1 = ENABLED
21	**EMEM_CFG_RESERVED2:** 0 = DISABLED 1 = ENABLED
20	**EMEM_CFG_RESERVED1:** 0 = DISABLED 1 = ENABLED
19	**EMEM_CFG_RESERVED0:** 0 = DISABLED 1 = ENABLED
18:16	**EMEM_ADR_CFG:** 0 = CFG0 1 = CFG1 2 = CFG2 3 = CFG3 4 = CFG4 5 = CFG5 6 = CFG6 7 = CFG7
15:0	**EMEM_SIZE_KB**

MC_EMEM_ARB_CFG_0

Offset: 005h
Read/Write: R/W
Reset: 0404.8a10

External memory arbiter config

The read/write and write/read counter thresholds are used to generate the read/write switch for the dram arbiter when compared to the read/write counter. The counter is incremented every cycle and cleared when the arbitration winner changes from read to write or write to read.

The bank counter threshold is used to block non same-page requests until the number of cycles since the last arbitration bank winner was the requested bank has reached the threshold. The non same-page bank counter threshold is used to block non same-page requests until the number of cycles since the last arbitration bank winner was the requested bank and it changed the same page, has reached the threshold. The bank counter thresholds are incremented only when the rest of the memory controller pipeline after the arbiter is ready, so that they are kept in sync with the requests sent to the external memory controller. There is a one-cycle imprecision on number of cycles related to the bank counter thresholds.

The emem_lowestfp_norw field allows to disable the read/write switch in the client priority vector if the client fixed priority is 0 (lowest). The emem_swap_fp_rw field allows to swap the read/write switch and the client fixed priority in the client priority vector.

Bit	Description
31	**EMEM_SWAP_FP_RW:** 0 = DISABLE 1 = ENABLE
30	**EMEM_LOWESTFP_NORW:** 0 = DISABLE 1 = ENABLE
29:22	**EMEM_WRCNT_TH**
21	**EMEM_CLEAR_AP_PREV_SPREQ:** 0 = DISABLE 1 = ENABLE
20	**EMEM_CLEAR_SP_ON_AUTOPC:** 0 = DISABLE 1 = ENABLE
19:14	**EMEM_BANKCNT_NSP_TH**
13:8	**EMEM_BANKCNT_TH**
7:0	**EMEM_RWCNT_TH**

MC_PARTITION_CONFLICT_CFG_0

Offset: 006h
Read/Write: R/W
Reset: 0000.0003

Partition client conflict configuration.

This register controls the behavior of the client data paths and the memory return data paths. When 'SINGLE', at most one (return) data path is active at any time. When 'MULTI', two or more (return) data paths can be active at the same time.

Bit	Description
1	**MEM_RDPATHS:** 0 = SINGLE 1 = MULTI
0	**PC_DATAPATHS:** 0 = SINGLE 1 = MULTI

MC_TIMEOUT_CTRL_0

Offset: 007h
Read/Write: R/W
Reset: 0000.0000

Time-out control Register.

Time-out clock counter scale factors. Reset value is 0. A value of zero means that the time-out clock counter is disabled. A value of N (non-zero) means that the time-out clock counter is decremented every 2^(N+1) memory cycles. This is used in conjunction with the _TMVAL registers defined in mc_regs.spec.

Bit	Description
6	**TMCREDITS:** 0 = FROM_CIF_FIFO 1 = ONE
5:3	**EMEM_TM_SFACTOR**
2:0	**IMEM_TM_SFACTOR**

MC_IBA_STATUS_0

Offset: 008h
Read/Write: RO
Reset: 0000.0000

Internal Memory Bad Address Status Register. This register stores information about the first detected out-of-bound access to internal memory.

Bit	Description
12:6	**IMEM_BADADR_GCID: Global client identifier per MC name (more accurate)**

Bit	Description
5:1	**IMEM_BADADR_CID: Client identifier.** 0 = DC 1 = DSP 2 = EPP 3 = G2 4 = HC 5 = I2SR 6 = I2ST 7 = ISP 8 = JE 9 = ME 10 = MPD 11 = NV 12 = SD 13 = VI
0	**IMEM_BADADR_RW: Type of request.** 0 = READ 1 = WRITE

MC_IBA_ADR_0

Offset: 009h
Read/Write: RO
Reset: 0000.0000

Internal Memory Bad Address Value Register.

This register stores the address value of the first detected out-of-bound access to internal memory.

Bit	Description
31:0	**IMEM_BADADR_ADR: Bad address - Value**

MC_IBA_BE_0

Offset: 00ah
Read/Write: RO
Reset: 0000.0000

Internal Memory Bad Address Byte Enables Value Register.

This register stores the byte enables value (if any) of the first detected out-of-bound access to internal memory.

Bit	Description
15:0	**IMEM_BADADR_BE: Bad address - Byte Enables**

MC_IBA_WRDATA0_0

Offset: 00bh
Read/Write: RO
Reset: 0000.0000

Internal Memory Bad Address Write Data Value Registers. This register stores the write data value (if any) of the first detected out-of-bound access to internal memory.

Bit	Description
31:0	**IMEM_BADADR_WRDATA0: Bad address - Write data 31:0**

MC_IBA_WRDATA1_0

Offset: 00ch
Read/Write: RO
Reset: 0000.0000

Bit	Description
31:0	**IMEM_BADADR_WRDATA1: Bad address - Write data 63:32**

MC_IBA_WRDATA2_0

Offset: 00dh
Read/Write: RO
Reset: 0000.0000

Bit	Description
31:0	**IMEM_BADADR_WRDATA2: Bad address - Write data 95:64**

MC_IBA_WRDATA3_0

Offset: 00eh
Read/Write: RO
Reset: 0000.0000

Bit	Description
31:0	**IMEM_BADADR_WRDATA3: Bad address - Write data 127:96**

MC_EBA_STATUS_0

Offset: 00fh
Read/Write: RO
Reset: 0000.0000

External Memory Bad Address Status Register.

This register stores information about the first detected out-of-bound access to external memory.

Bit	Description
12:6	**EMEM_BADADR_GCID: Global client identifier per MC name (more accurate)**
5:1	**EMEM_BADADR_CID: Client identifier per SW name.** 0 = DC 1 = DSP 2 = EPP 3 = G2 4 = HC 5 = I2SR 6 = I2ST 7 = ISP 8 = JE 9 = ME 10 = MPD 11 = NV 12 = SD 13 = VI
0	**EMEM_BADADR_RW: Type of request (read/write).** 0 = READ 1 = WRITE

MC_EBA_ADR_0

Offset: 010h
Read/Write: RO
Reset: 0000.0000

External Memory Bad Address Value Register.

This register stores the address value of the first detected out-of-bound access to external memory.

Bit	Description
31:0	**EMEM_BADADR_ADR: Bad address - Value**

MC_EBA_BE_0

Offset: 011h
Read/Write: RO
Reset: 0000.0000

External Memory Bad Address Byte Enables Value Register.

This register stores the byte enables value (if any) of the first detected out-of-bound access to external memory.

Bit	Description
15:0	**EMEM_BADADR_BE: Bad address - Byte Enables**

MC_EBA_WRDATA0_0

Offset: 012h
Read/Write: RO
Reset: 0000.0000

External Memory Bad Address Write Data Value (lower 32 bits) Register.

This register stores the write data value (if any) of the first detected out-of-bound access to external memory.

Bit	Description
31:0	**EMEM_BADADR_WRDATA0: Bad address - Write data 31:0**

MC_EBA_WRDATA1_0

Offset: 013h
Read/Write: RO
Reset: 0000.0000

Bit	Description
31:0	**EMEM_BADADR_WRDATA1: Bad address - Write data 63:32**

MC_EBA_WRDATA2_0

Offset: 014h
Read/Write: RO
Reset: 0000.0000

Bit	Description
31:0	**EMEM_BADADR_WRDATA2: Bad address - Write data 95:64**

MC_EBA_WRDATA3_0

Offset: 015h
Read/Write: RO
Reset: 0000.0000

Bit	Description
31:0	**EMEM_BADADR_WRDATA3: Bad address - Write data 127:96**

MC_CLIENT_CTRL_0

Offset: 016h
Read/Write: R/W
Reset: 0000.3fff

Include additional registers.

This file is generated from the project_ar_*.spec files and should be located in the include/project directory. Memory Client Control Register. Active high. Reset value is enabled. When enabled, the client requests are free to go to arbitration. When disabled, the client requests are blocked before arbitration.

Bit	Description
13	**VI_ENABLE** 0 = Disable 1 = Enable
12	**SD_ENABLE** 0 = Disable 1 = Enable
11	**NV_ENABLE** 0 = Disable 1 = Enable
10	**MPD_ENABLE** 0 = Disable 1 = Enable
9	**ME_ENABLE** 0 = Disable 1 = Enable
8	**JE_ENABLE** 0 = Disable 1 = Enable
7	**ISP_ENABLE** 0 = Disable 1 = Enable
6	**I2ST_ENABLE** 0 = Disable 1 = Enable
5	**I2SR_ENABLE** 0 = Disable 1 = Enable
4	**HC_ENABLE** 0 = Disable 1 = Enable
3	**G2_ENABLE** 0 = Disable 1 = Enable

Bit	Description
2	**EPP_ENABLE** 0 = Disable 1 = Enable
1	**DSP_ENABLE** 0 = Disable 1 = Enable
0	**DC_ENABLE** 0 = Disable 1 = Enable

MC_CLIENT_HOTRESETN_0

Offset: 017h
Read/Write: R/W
Reset: 0000.3fff

Memory Client Hot Reset Register. Active low. Reset value is no hot reset. When enabled, the client requests before arbitration are cleared.

A proper client reset sequence is as followed:

- Clear client bit in the MC_ENABLEREG register to block its requests.
- Clear module bit in the HOST_RSTREG register to reset the module, and clear client bit in the MC_HOTRSTREG register to clear the client requests seating before arbitration.

- Poll the _OUTREQCNT register till zero.
- Set client bit in the MC_HOTRSTREG register to release the hot reset.
- Set module bit in the HOST_RSTREG register to release the module reset.
- Set client bit in the MC_ENABLEREG register to allow new requests to proceed to arbitration.

Bit	Description
13	**VI_HOTRESETN** 0 = ENABLE 1 = DISABLE
12	**SD_HOTRESETN** 0 = ENABLE 1 = DISABLE
11	**NV_HOTRESETN** 0 = Enable 1 = Disable
10	**MPD_HOTRESETN** 0 = Enable 1 = Disable
9	**ME_HOTRESETN** 0 = Enable 1 = Disable
8	**JE_HOTRESETN** 0 = Enable 1 = Disable
7	**ISP_HOTRESETN** 0 = Enable 1 = Disable
6	**I2ST_HOTRESETN** 0 = Enable 1 = Disable

Bit	Description
5	**I2SR_HOTRESETN** 0 = Enable 1 = Disable
4	**HC_HOTRESETN** 0 = Enable 1 = Disable
3	**G2_HOTRESETN** 0 = Enable 1 = Disable
2	**EPP_HOTRESETN** 0 = Enable 1 = Disable
1	**DSP_HOTRESETN** 0 = Enable 1 = Disable
0	**DC_HOTRESETN** 0 = Enable 1 = Disable

MC_DC_ORRC_0

Offset: 018h
Read/Write: RO
Reset: 0000.0000

DC Outstanding Request Register. Read-only. This reports the number of outstanding client requests after arbitration. It is incremented by HW when a client request is granted by the arbiter. For read clients, it is decremented by HW when a return data is sent back to the client.

Bit	Description
7:0	**DC_OUTREQCNT**

MC_DSP_ORRC_0

Offset: 019h
Read/Write: RO
Reset: 0000.0000

DSP Outstanding Request Register. Read-only. This reports the number of outstanding client requests after arbitration. It is incremented by HW when a client request is granted by the arbiter. For read clients, it is decremented by HW when a return data is sent back to the client.

Bit	Description
7:0	**DSP_OUTREQCNT**

MC_EPP_ORRC_0

Offset: 01ah
Read/Write: RO
Reset: 0000.0000

EPP Outstanding Request Register. Read-only.

This reports the number of outstanding client requests after arbitration. It is incremented by HW when a client request is granted by the arbiter. For read clients, it is decremented by HW when a return data is sent back to the client.

Bit	Description
7:0	**EPP_OUTREQCNT**

MC_G2_ORRC_0

Offset: 01bh
Read/Write: RO
Reset: 0000.0000

G2 Outstanding Request Register. Read-only.

This reports the number of outstanding client requests after arbitration. It is incremented by HW when a client request is granted by the arbiter. For read clients, it is decremented by HW when a return data is sent back to the client.

Bit	Description
7:0	**G2_OUTREQCNT**

MC_HC_ORRC_0

Offset: 01ch
Read/Write: RO
Reset: 0000.0000

HC Outstanding Request Register. Read-only.

This reports the number of outstanding client requests after arbitration. It is incremented by HW when a client request is granted by the arbiter. For read clients, it is decremented by HW when a return data is sent back to the client.

Bit	Description
7:0	**HC_OUTREQCNT**

MC_I2SR_ORRC_0

Offset: 01dh
Read/Write: RO
Reset: 0000.0000

I2SR Outstanding Request Register. Read-only. This reports the number of outstanding client requests after arbitration. It is incremented by HW when a client request is granted by the arbiter. For read clients, it is decremented by HW when a return data is sent back to the client.

Bit	Description
7:0	**I2SR_OUTREQCNT**

MC_I2ST_ORRC_0

Offset: 01eh
Read/Write: RO
Reset: 0000.0000

I2ST Outstanding Request Register. Read-only.

This reports the number of outstanding client requests after arbitration. It is incremented by HW when a client request is granted by the arbiter. For read clients, it is decremented by HW when a return data is sent back to the client.

Bit	Description
7:0	**I2ST_OUTREQCNT**

MC_ISP_ORRC_0

Offset: 01fh
Read/Write: RO
Reset: 0000.0000

ISP Outstanding Request Register. Read-only.

This reports the number of outstanding client requests after arbitration. It is incremented by HW when a client request is granted by the arbiter. For read clients, it is decremented by HW when a return data is sent back to the client.

Bit	Description
7:0	**ISP_OUTREQCNT**

MC_JE_ORRC_0

Offset: 020h
Read/Write: RO
Reset: 0000.0000

JE Outstanding Request Register. Read-only.

This reports the number of outstanding client requests after arbitration. It is incremented by HW when a client request is granted by the arbiter. For read clients, it is decremented by HW when a return data is sent back to the client.

Bit	Description
7:0	**JE_OUTREQCNT**

MC_ME_ORRC_0

Offset: 021h
Read/Write: RO
Reset: 0000.0000

ME Outstanding Request Register. Read-only.

This reports the number of outstanding client requests after arbitration. It is incremented by HW when a client request is granted by the arbiter. For read clients, it is decremented by HW when a return data is sent back to the client.

Bit	Description
7:0	**ME_OUTREQCNT**

MC_MPD_ORRC_0

Offset: 022h
Read/Write: RO
Reset: 0000.0000

MPD Outstanding Request Register. Read-only.

This reports the number of outstanding client requests after arbitration. It is incremented by HW when a client request is granted by the arbiter. For read clients, it is decremented by HW when a return data is sent back to the client.

Bit	Description
7:0	**MPD_OUTREQCNT**

MC_NV_ORRC_0

Offset: 023h
Read/Write: RO
Reset: 0000.0000

NV Outstanding Request Register. Read-only.

This reports the number of outstanding client requests after arbitration. It is incremented by HW when a client request is granted by the arbiter. For read clients, it is decremented by HW when a return data is sent back to the client.

Bit	Description
7:0	**NV_OUTREQCNT**

MC_SD_ORRC_0

Offset: 024h
Read/Write: RO
Reset: 0000.0000

SD Outstanding Request Register. Read-only.

This reports the number of outstanding client requests after arbitration. It is incremented by HW when a client request is granted by the arbiter. For read clients, it is decremented by HW when a return data is sent back to the client.

Bit	Description
7:0	**SD_OUTREQCNT**

MC_VI_ORRC_0

Offset: 025h
Read/Write: RO
Reset: 0000.0000

VI Outstanding Request Register. Read-only.

This reports the number of outstanding client requests after arbitration. It is incremented by HW when a client request is granted by the arbiter. For read clients, it is decremented by HW when a return data is sent back to the client.

Bit	Description
7:0	**VI_OUTREQCNT**

MC_AP_CTRL_0_0

Offset: 026h
Read/Write: R/W
Reset: 0000.0000

Memory Client Auto-Precharge Register 0. Reset value is disabled.

Control whether an auto-precharge request is sent when the client sends a request to access the external memory.

Bit	Description
31	**RECON_APVAL:** 0 = Disable 1 = Enable
30	**POSTPROC_APVAL:** 0 = Disable 1 = Enable
29	**EPPY_APVAL:** 0 = Disable 1 = Enable
28	**EPPV_APVAL:** 0 = Disable 1 = Enable
27	**EPPU_APVAL:** 0 = Disable 1 = Enable
26	**VLCDMAR_APVAL:** 0 = Disable 1 = Enable
25	**SDTR_APVAL:** 0 = Disable 1 = Enable
24	**ISPR_APVAL:** 0 = Disable 1 = Enable
23	**I2SR_APVAL** 0 = Disable 1 = Enable
22	**HOST1XR_APVAL** 0 = Disable 1 = Enable
21	**HOST1XDMAR_APVAL** 0 = Disable 1 = Enable
20	**GRDFZ_APVAL** 0 = DISABLE 1 = ENABLE
19	**GRDFT_APVAL** 0 = DISABLE 1 = ENABLE
18	**GRDFC_APVAL** 0 = DISABLE 1 = ENABLE
17	**G2DR_APVAL** 0 = DISABLE 1 = ENABLE

Bit	Description
16	**DSPR_APVAL** 0 = DISABLE 1 = ENABLE
15	**DISPLAYHC_APVAL** 0 = DISABLE 1 = ENABLE
14	**ACDCR_APVAL** 0 = DISABLE 1 = ENABLE
13	**YUVREF_APVAL** 0 = DISABLE 1 = ENABLE
12	**YUVCUR_APVAL** 0 = Disable 1 = Enable
11	**VIRUV_APVAL** 0 = Disable 1 = Enable
10	**MOTCOMP_APVAL** 0 = Disable 1 = Enable
9	**JPEGER_APVAL** 0 = Disable 1 = Enable
8	**G2SR_APVAL** 0 = Disable 1 = Enable
7	**G2PR_APVAL** 0 = Disable 1 = Enable
6	**EPPVP_APVAL** 0 = Disable 1 = Enable
5	**EPPUP_APVAL** 0 = Disable 1 = Enable
4	**DISPLAY1B_APVAL** 0 = Disable 1 = Enable
3	**DISPLAY0C_APVAL** 0 = Disable 1 = Enable
2	**DISPLAY0B_APVAL** 0 = Disable 1 = Enable
1	**DISPLAY0A_APVAL** 0 = Disable 1 = Enable
0	**DBR_APVAL** 0 = Disable 1 = Enable

MC_AP_CTRL_1_0

Offset: 027h
Read/Write: R/W
Reset: 0000.0000

Memory Client Auto-Precharge Register 1. Reset value is disabled.

Control whether an auto-precharge request is sent when the client sends a request to access the external memory.

Bit	Description
15	**VLCDMAW_APVAL** 0 = Disable 1 = Enable
14	**SDTW_APVAL** 0 = Disable 1 = Enable
13	**JPEGEW_APVAL** 0 = Disable 1 = Enable
12	**ISPW_APVAL** 0 = Disable 1 = Enable
11	**I2SW_APVAL** 0 = Disable 1 = Enable
10	**HOST1XW_APVAL** 0 = Disable 1 = Enable
9	**GRDWZ_APVAL** 0 = Disable 1 = Enable
8	**GRDWC_APVAL** 0 = DISABLE 1 = ENABLE
7	**DSPW_APVAL** 0 = DISABLE 1 = ENABLE
6	**ACDCW_APVAL** 0 = DISABLE 1 = ENABLE
5	**G2DW_APVAL** 0 = Disable 1 = Enable
4	**YUVWR_APVAL** 0 = Disable 1 = Enable
3	**VIWY_APVAL** 0 = Disable 1 = Enable
2	**VIWV_APVAL** 0 = Disable 1 = Enable
1	**VIWU_APVAL** 0 = Disable 1 = Enable

Bit	Description
0	**VIWSB_APVAL** 0 = Disable 1 = Enable

MC_FPRI_CTRL_DC_0

Offset: 028h
Read/Write: R/W
Reset: 0000.0000

Fixed-priority Register for dc clients. Reset value is lowest.

Bit	Description
9:8	**DISPLAYHC_PRIVAL** 0 = LOWEST 1 = LOW 2 = MED 3 = HIGH
7:6	**DISPLAY1B_PRIVAL** 0 = LOWEST 1 = LOW 2 = MED 3 = HIGH
5:4	**DISPLAY0C_PRIVAL** 0 = LOWEST 1 = LOW 2 = MED 3 = HIGH
3:2	**DISPLAY0B_PRIVAL** 0 = LOWEST 1 = LOW 2 = MED 3 = HIGH
1:0	**DISPLAY0A_PRIVAL** 0 = LOWEST 1 = LOW 2 = MED 3 = HIGH

MC_FPRI_CTRL_DSP_0

Offset: 029h
Read/Write: R/W
Reset: 0000.0000

Fixed-priority Register for DSP clients. Reset value is lowest.

Bit	Description
3:2	**DSPW_PRIVAL** 0 = LOWEST 1 = LOW 2 = MED 3 = HIGH
1:0	**DSPR_PRIVAL** 0 = LOWEST 1 = LOW 2 = MED 3 = HIGH

MC_FPRI_CTRL_EPP_0

Offset: 02ah
Read/Write: R/W
Reset: 0000.0000

Fixed-priority Register for EPP clients. Reset value is lowest.

Bit	Description
9:8	**EPPY_PRIVAL** 0 = LOWEST 1 = LOW 2 = MED 3 = HIGH
7:6	**EPPV_PRIVAL** 0 = LOWEST 1 = LOW 2 = MED 3 = HIGH
5:4	**EPPU_PRIVAL** 0 = LOWEST 1 = LOW 2 = MED 3 = HIGH
3:2	**EPPVP_PRIVAL** 0 = LOWEST 1 = LOW 2 = MED 3 = HIGH
1:0	**EPPUP_PRIVAL** 0 = LOWEST 1 = LOW 2 = MED 3 = HIGH

MC_FPRI_CTRL_G2_0

Offset: 02bh
Read/Write: R/W
Reset: 0000.0000

Fixed-priority Register for g2 clients. Reset value is lowest.

Bit	Description
7:6	**G2DW_PRIVAL** 0 = LOWEST 1 = LOW 2 = MED 3 = HIGH
5:4	**G2DR_PRIVAL** 0 = LOWEST 1 = LOW 2 = MED 3 = HIGH
3:2	**G2SR_PRIVAL** 0 = LOWEST 1 = LOW 2 = MED 3 = HIGH
1:0	**G2PR_PRIVAL** 0 = LOWEST 1 = LOW 2 = MED 3 = HIGH

MC_FPRI_CTRL_HC_0

Offset: 02ch
Read/Write: R/W
Reset: 0000.0000

Fixed-priority Register for HC clients. Reset value is lowest.

Bit	Description
5:4	**HOST1XW_PRIVAL** 0 = LOWEST 1 = LOW 2 = MED 3 = HIGH
3:2	**HOST1XR_PRIVAL** 0 = LOWEST 1 = LOW 2 = MED 3 = HIGH
1:0	**HOST1XDMAR_PRIVAL** 0 = LOWEST 1 = LOW 2 = MED 3 = HIGH

MC_FPRI_CTRL_I2SR_0

Offset: 02dh
Read/Write: R/W
Reset: 0000.0000

Fixed-priority Register for i2sr clients. Reset value is lowest.

Bit	Description
1:0	**I2SW_PRIVAL** 0 = LOWEST 1 = LOW 2 = MED 3 = HIGH

MC_FPRI_CTRL_I2ST_0

Offset: 02eh
Read/Write: R/W
Reset: 0000.0000

Fixed-priority Register for i2st clients. Reset value is lowest.

Bit	Description
1:0	**I2SR_PRIVAL** 0 = LOWEST 1 = LOW 2 = MED 3 = HIGH

MC_FPRI_CTRL_ISP_0

Offset: 02fh
Read/Write: R/W
Reset: 0000.0000

Fixed-priority Register for ISP clients. Reset value is lowest.

Bit	Description
3:2	**ISPW_PRIVAL** 0 = LOWEST 1 = LOW 2 = MED 3 = HIGH
1:0	**ISPR_PRIVAL** 0 = LOWEST 1 = LOW 2 = MED 3 = HIGH

MC_FPRI_CTRL_JE_0

Offset: 030h
Read/Write: R/W
Reset: 0000.0000

Fixed-priority Register for JE clients. Reset value is lowest.

Bit	Description
3:2	**JPEGEW_PRIVAL** 0 = LOWEST 1 = LOW 2 = MED 3 = HIGH
1:0	**JPEGER_PRIVAL** 0 = LOWEST 1 = LOW 2 = MED 3 = HIGH

MC_FPRI_CTRL_ME_0

Offset: 031h
Read/Write: R/W
Reset: 0000.0000

Fixed-priority Register for me clients. Reset value is lowest.

Bit	Description
13:12	**VLCDMAW_PRIVAL** 0 = LOWEST 1 = LOW 2 = MED 3 = HIGH
11:10	**ACDCW_PRIVAL** 0 = LOWEST 1 = LOW 2 = MED 3 = HIGH
9:8	**YUVWR_PRIVAL** 0 = LOWEST 1 = LOW 2 = MED 3 = HIGH
7:6	**VLCDMAR_PRIVAL** 0 = LOWEST 1 = LOW 2 = MED 3 = HIGH
5:4	**ACDCR_PRIVAL** 0 = LOWEST 1 = LOW 2 = MED 3 = HIGH

Bit	Description
3:2	**YUVREF_PRIVAL** 0 = LOWEST 1 = LOW 2 = MED 3 = HIGH
1:0	**YUVCUR_PRIVAL** 0 = LOWEST 1 = LOW 2 = MED 3 = HIGH

MC_FPRI_CTRL_MPD_0

Offset: 032h
Read/Write: R/W
Reset: 0000.0000

Fixed-priority Register for MPD clients. Reset value is lowest.

Bit	Description
7:6	**RECON_PRIVAL** 0 = LOWEST 1 = LOW 2 = MED 3 = HIGH
5:4	**POSTPROC_PRIVAL** 0 = LOWEST 1 = LOW 2 = MED 3 = HIGH
3:2	**MOTCOMP_PRIVAL** 0 = LOWEST 1 = LOW 2 = MED 3 = HIGH
1:0	**DBR_PRIVAL** 0 = LOWEST 1 = LOW 2 = MED 3 = HIGH

MC_FPRI_CTRL_NV_0

Offset: 033h
Read/Write: R/W
Reset: 0000.0000

Fixed-priority Register for nv clients. Reset value is lowest.

Bit	Description
9:8	**GRDWZ_PRIVAL** 0 = LOWEST 1 = LOW 2 = MED 3 = HIGH
7:6	**GRDWC_PRIVAL** 0 = LOWEST 1 = LOW 2 = MED 3 = HIGH
5:4	**GRDFZ_PRIVAL** 0 = LOWEST 1 = LOW 2 = MED 3 = HIGH
3:2	**GRDFT_PRIVAL** 0 = LOWEST 1 = LOW 2 = MED 3 = HIGH
1:0	**GRDFC_PRIVAL** 0 = LOWEST 1 = LOW 2 = MED 3 = HIGH

MC_FPRI_CTRL_SD_0

Offset: 034h
Read/Write: R/W
Reset: 0000.0000

Fixed-priority Register for sd clients. Reset value is lowest.

Bit	Description
3:2	**SDTW_PRIVAL** 0 = LOWEST 1 = LOW 2 = MED 3 = HIGH
1:0	**SDTR_PRIVAL** 0 = LOWEST 1 = LOW 2 = MED 3 = HIGH

MC_FPRI_CTRL_VI_0

Offset: 035h
Read/Write: R/W
Reset: 0000.0000

Fixed-priority Register for vi clients. Reset value is lowest.

Bit	Description
9:8	**VIWY_PRIVAL** 0 = LOWEST 1 = LOW 2 = MED 3 = HIGH
7:6	**VIWV_PRIVAL** 0 = LOWEST 1 = LOW 2 = MED 3 = HIGH
5:4	**VIWU_PRIVAL** 0 = LOWEST 1 = LOW 2 = MED 3 = HIGH
3:2	**VIWSB_PRIVAL** 0 = LOWEST 1 = LOW 2 = MED 3 = HIGH
1:0	**VIRUV_PRIVAL** 0 = LOWEST 1 = LOW 2 = MED 3 = HIGH

MC_TIMEOUT_DC_0

Offset: 036h
Read/Write: R/W
Reset: 0000.0000

Request time-out Register for dc clients. Reset value is 0.

If not zero, the client request time-out counter is reloaded with this value when it reaches zero. It is decremented when there is a pending request and the time-out clock counter for the request destination (internal/external memory) reaches zero.

When the client request time-out counter reaches zero, a number of time-out credits equal to the number of its requests pending inside the memory controller client fifo are allocated to this client. The client fifo depth is NV_2MC_RFIFO_MAXCREDITS if defined, otherwise NV_MC_RFIFO_MAXCREDITS_DEFAULT.

The number of time-out credits is decremented each time a request is granted. As long as there are credits, the client time-out priority bit is set.

Bit	Description
14:12	**DISPLAYHC_TMVAL**
11:9	**DISPLAY1B_TMVAL**
8:6	**DISPLAY0C_TMVAL**
5:3	**DISPLAY0B_TMVAL**
2:0	**DISPLAY0A_TMVAL**

MC_TIMEOUT_DSP_0

Offset: 037h
Read/Write: R/W
Reset: 0000.0000

Request time-out Register for DSP clients. Reset value is 0.

If not zero, the client request time-out counter is reloaded with this value when it reaches zero. It is decremented when there is a pending request and the time-out clock counter for the request destination (internal/external memory) reaches zero.

When the client request time-out counter reaches zero, a number of time-out credits equal to the number of its requests pending inside the memory controller client fifo are allocated to this client. The client fifo depth is NV_2MC_RFIFO_MAXCREDITS if defined, otherwise NV_MC_RFIFO_MAXCREDITS_DEFAULT.

The number of time-out credits is decremented each time a request is granted. As long as there are credits, the client time-out priority bit is set.

Bit	Description
5:3	**DSPW_TMVAL**
2:0	**DSPR_TMVAL**

MC_TIMEOUT_EPP_0

Offset: 038h
Read/Write: R/W
Reset: 0000.0000

Request time-out Register for EPP clients. Reset value is 0.

If not zero, the client request time-out counter is reloaded with this value when it reaches zero. It is decremented when there is a pending request and the time-out clock counter for the request destination (internal/external memory) reaches zero.

When the client request time-out counter reaches zero, a number of time-out credits equal to the number of its requests pending inside the memory controller client fifo are allocated to this client. The client fifo depth is NV_2MC_RFIFO_MAXCREDITS if defined, otherwise NV_MC_RFIFO_MAXCREDITS_DEFAULT.

The number of time-out credits is decremented each time a request is granted. As long as there are credits, the client time-out priority bit is set.

Bit	Description
14:12	**EPPY_TMVAL**
11:9	**EPPV_TMVAL**
8:6	**EPPU_TMVAL**
5:3	**EPPVP_TMVAL**
2:0	**EPPUP_TMVAL**

MC_TIMEOUT_G2_0

Offset: 039h
Read/Write: R/W
Reset: 0000.0000

Request time-out Register for g2 clients. Reset value is 0.

If not zero, the client request time-out counter is reloaded with this value when it reaches zero. It is decremented when there is a pending request and the time-out clock counter for the request destination (internal/external memory) reaches zero.

When the client request time-out counter reaches zero, a number of time-out credits equal to the number of its requests pending inside the memory controller client fifo are allocated to this client. The client fifo depth is NV_2MC_RFIFO_MAXCREDITS if defined, otherwise NV_MC_RFIFO_MAXCREDITS_DEFAULT.

The number of time-out credits is decremented each time a request is granted. As long as there are credits, the client time-out priority bit is set.

Bit	Description
11:9	**G2DW_TMVAL**
8:6	**G2DR_TMVAL**
5:3	**G2SR_TMVAL**
2:0	**G2PR_TMVAL**

MC_TIMEOUT_HC_0

Offset: 03ah
Read/Write: R/W
Reset: 0000.0000

Request time-out Register for HC clients. Reset value is 0.

If not zero, the client request time-out counter is reloaded with this value when it reaches zero. It is decremented when there is a pending request and the time-out clock counter for the request destination (internal/external memory) reaches zero.

When the client request time-out counter reaches zero, a number of time-out credits equal to the number of its requests pending inside the memory controller client fifo are allocated to this client. The client fifo depth is NV_2MC_RFIFO_MAXCREDITS if defined, otherwise NV_MC_RFIFO_MAXCREDITS_DEFAULT.

The number of time-out credits is decremented each time a request is granted. As long as there are credits, the client time-out priority bit is set.

Bit	Description
8:6	**HOST1XW_TMVAL**
5:3	**HOST1XR_TMVAL**
2:0	**HOST1XDMAR_TMVAL**

MC_TIMEOUT_I2SR_0

Offset: 03bh
Read/Write: R/W
Reset: 0000.0000

Request time-out Register for i2sr clients. Reset value is 0.

If not zero, the client request time-out counter is reloaded with this value when it reaches zero. It is decremented when there is a pending request and the time-out clock counter for the request destination (internal/external memory) reaches zero.

When the client request time-out counter reaches zero, a number of time-out credits equal to the number of its requests pending inside the memory controller client fifo are allocated to this client. The client fifo depth is NV_2MC_RFIFO_MAXCREDITS if defined, otherwise NV_MC_RFIFO_MAXCREDITS_DEFAULT.

The number of time-out credits is decremented each time a request is granted. As long as there are credits, the client time-out priority bit is set.

Bit	Description
2:0	**I2SW_TMVAL**

MC_TIMEOUT_I2ST_0

Offset: 03ch
Read/Write: R/W
Reset: 0000.0000

Request time-out Register for i2st clients. Reset value is 0.

If not zero, the client request time-out counter is reloaded with this value when it reaches zero. It is decremented when there is a pending request and the time-out clock counter for the request destination (internal/external memory) reaches zero.

When the client request time-out counter reaches zero, a number of time-out credits equal to the number of its requests pending inside the memory controller client fifo are allocated to this client. The client fifo depth is NV_2MC_RFIFO_MAXCREDITS if defined, otherwise NV_MC_RFIFO_MAXCREDITS_DEFAULT.

The number of time-out credits is decremented each time a request is granted. As long as there are credits, the client time-out priority bit is set.

Bit	Description
2:0	**I2SR_TMVAL**

MC_TIMEOUT_ISP_0

Offset: 03dh
Read/Write: R/W
Reset: 0000.0000

Request time-out Register for ISP clients. Reset value is 0.

If not zero, the client request time-out counter is reloaded with this value when it reaches zero. It is decremented when there is a pending request and the time-out clock counter for the request destination (internal/external memory) reaches zero.

When the client request time-out counter reaches zero, a number of time-out credits equal to the number of its requests pending inside the memory controller client fifo are allocated to this client. The client fifo depth is NV_2MC_RFIFO_MAXCREDITS if defined, otherwise NV_MC_RFIFO_MAXCREDITS_DEFAULT.

The number of time-out credits is decremented each time a request is granted. As long as there are credits, the client time-out priority bit is set.

Bit	Description
5:3	**ISPW_TMVAL**
2:0	**ISPR_TMVAL**

MC_TIMEOUT_JE_0

Offset: 03eh
Read/Write: R/W
Reset: 0000.0000

Request time-out Register for JE clients. Reset value is 0.

If not zero, the client request time-out counter is reloaded with this value when it reaches zero. It is decremented when there is a pending request and the time-out clock counter for the request destination (internal/external memory) reaches zero.

When the client request time-out counter reaches zero, a number of time-out credits equal to the number of its requests pending inside the memory controller client fifo are allocated to this client. The client fifo depth is NV_2MC_RFIFO_MAXCREDITS if defined, otherwise NV_MC_RFIFO_MAXCREDITS_DEFAULT.

The number of time-out credits is decremented each time a request is granted. As long as there are credits, the client time-out priority bit is set.

Bit	Description
5:3	**JPEGEW_TMVAL**
2:0	**JPEGER_TMVAL**

MC_TIMEOUT_ME_0

Offset: 03fh
Read/Write: R/W
Reset: 0000.0000

Request time-out Register for me clients. Reset value is 0.

If not zero, the client request time-out counter is reloaded with this value when it reaches zero. It is decremented when there is a pending request and the time-out clock counter for the request destination (internal/external memory) reaches zero.

When the client request time-out counter reaches zero, a number of time-out credits equal to the number of its requests pending inside the memory controller client fifo are allocated to this client. The client fifo depth is NV_2MC_RFIFO_MAXCREDITS if defined, otherwise NV_MC_RFIFO_MAXCREDITS_DEFAULT.

The number of time-out credits is decremented each time a request is granted. As long as there are credits, the client time-out priority bit is set.

Bit	Description
20:18	**VLCDMAW_TMVAL**
17:15	**ACDCW_TMVAL**
14:12	**YUVWR_TMVAL**
11:9	**VLCDMAR_TMVAL**
8:6	**ACDCR_TMVAL**
5:3	**YUVREF_TMVAL**
2:0	**YUVCUR_TMVAL**

MC_TIMEOUT_MPD_0

Offset: 040h
Read/Write: R/W
Reset: 0000.0000

Request time-out Register for MPD clients. Reset value is 0. If not zero, the client request time-out counter is reloaded with this value when it reaches zero. It is decremented when there is a pending request and the time-out clock counter for the request destination (internal/external memory) reaches zero.

When the client request time-out counter reaches zero, a number of time-out credits equal to the number of its requests pending inside the memory controller client fifo are allocated to this client. The client fifo depth is NV_2MC_RFIFO_MAXCREDITS if defined, otherwise NV_MC_RFIFO_MAXCREDITS_DEFAULT.

The number of time-out credits is decremented each time a request is granted. As long as there are credits, the client time-out priority bit is set.

Bit	Description
11:9	**RECON_TMVAL**
8:6	**POSTPROC_TMVAL**
5:3	**MOTCOMP_TMVAL**
2:0	**DBR_TMVAL**

MC_TIMEOUT_NV_0

Offset: 041h
Read/Write: R/W
Reset: 0000.0000

Request time-out Register for nv clients. Reset value is 0. If not zero, the client request time-out counter is reloaded with this value when it reaches zero. It is decremented when there is a pending request and the time-out clock counter for the request destination (internal/external memory) reaches zero.

When the client request time-out counter reaches zero, a number of time-out credits equal to the number of its requests pending inside the memory controller client fifo are allocated to this client. The client fifo depth is NV_2MC_RFIFO_MAXCREDITS if defined, otherwise NV_MC_RFIFO_MAXCREDITS_DEFAULT.

The number of time-out credits is decremented each time a request is granted. As long as there are credits, the client time-out priority bit is set.

Bit	Description
14:12	**GRDWZ_TMVAL**
11:9	**GRDWC_TMVAL**
8:6	**GRDFZ_TMVAL**
5:3	**GRDFT_TMVAL**
2:0	**GRDFC_TMVAL**

MC_TIMEOUT_SD_0

Offset: 042h
Read/Write: R/W
Reset: 0000.0000

Request time-out Register for sd clients. Reset value is 0. If not zero, the client request time-out counter is reloaded with this value when it reaches zero. It is decremented when there is a pending request and the time-out clock counter for the request destination (internal/external memory) reaches zero.

When the client request time-out counter reaches zero, a number of time-out credits equal to the number of its requests pending inside the memory controller client fifo are allocated to this client. The client fifo depth is NV_2MC_RFIFO_MAXCREDITS if defined, otherwise NV_MC_RFIFO_MAXCREDITS_DEFAULT.

The number of time-out credits is decremented each time a request is granted. As long as there are credits, the client time-out priority bit is set.

Bit	Description
5:3	**SDTW_TMVAL**
2:0	**SDTR_TMVAL**

MC_TIMEOUT_VI_0

Offset: 043h
Read/Write: R/W
Reset: 0000.0000

Request time-out Register for vi clients. Reset value is 0. If not zero, the client request time-out counter is reloaded with this value when it reaches zero. It is decremented when there is a pending request and the time-out clock counter for the request destination (internal/external memory) reaches zero.

When the client request time-out counter reaches zero, a number of time-out credits equal to the number of its requests pending inside the memory controller client fifo are allocated to this client. The client fifo depth is NV_2MC_RFIFO_MAXCREDITS if defined, otherwise NV_MC_RFIFO_MAXCREDITS_DEFAULT.

The number of time-out credits is decremented each time a request is granted. As long as there are credits, the client time-out priority bit is set.

Bit	Description
14:12	**VIWY_TMVAL**
11:9	**VIWV_TMVAL**
8:6	**VIWU_TMVAL**
5:3	**VIWSB_TMVAL**
2:0	**VIRUV_TMVAL**

MC_OBS_HOSTIF_MAIN_HWR_0

Offset: 000h
Read/Write: RO
Reset: 0000.0000

HOST interface

Bit	Description
24:9	**HOSTIF_HWR_HOST1X2MC_OFFSET**
8:5	**HOSTIF_HWR_HOST1X2MC_BE**
4	**HOSTIF_HWR_HOST1X2MC_CTXSW**
3	**HOSTIF_HWR_HOST1X2MC_COR**
2	**HOSTIF_HWR_HOST1X2MC_RWN**
1	**HOSTIF_HWR_HOST1X2MC_STALL**
0	**HOSTIF_HWR_HOST1X2MC_VALID**

MC_OBS_HOSTIF_DATA0_HWR_0

Offset: 001h
Read/Write: RO
Reset: 0000.0000

Bit	Description
24:9	**HOSTIF_DATA0_HWR_HOST1X2MC_DATA**
8:5	**HOSTIF_DATA0_HWR_HOST1X2MC_BE**
4	**HOSTIF_DATA0_HWR_HOST1X2MC_CTXSW**
3	**HOSTIF_DATA0_HWR_HOST1X2MC_COR**
2	**HOSTIF_DATA0_HWR_HOST1X2MC_RWN**
1	**HOSTIF_DATA0_HWR_HOST1X2MC_STALL**
0	**HOSTIF_DATA0_HWR_HOST1X2MC_VALID**

MC_OBS_HOSTIF_DATA1_HWR_0

Offset: 002h
Read/Write: RO
Reset: 0000.0000

Bit	Description
24:9	**HOSTIF_DATA1_HWR_HOST1X2MC_DATA**
8:5	**HOSTIF_DATA1_HWR_HOST1X2MC_BE**
4	**HOSTIF_DATA1_HWR_HOST1X2MC_CTXSW**
3	**HOSTIF_DATA1_HWR_HOST1X2MC_COR**
2	**HOSTIF_DATA1_HWR_HOST1X2MC_RWN**
1	**HOSTIF_DATA1_HWR_HOST1X2MC_STALL**
0	**HOSTIF_DATA1_HWR_HOST1X2MC_VALID**

MC_OBS_HOSTIF_DATA0_HRD_0

Offset: 003h
Read/Write: RO
Reset: 0000.0000

Bit	Description
23:8	**HOSTIF_DATA0_HRD_MC2HOST1X_DATA**
7:6	**HOSTIF_DATA0_HRD_MC2HOST1X_TYPE**
5:2	**HOSTIF_DATA0_HRD_MC2HOST1X_CHANNEL**
1	**HOSTIF_DATA0_HRD_MC2HOST1X_STALL**
0	**HOSTIF_DATA0_HRD_MC2HOST1X_VALID**

MC_OBS_HOSTIF_DATA1_HRD_0

Offset: 004h
Read/Write: RO
Reset: 0000.0000

Bit	Description
23:8	**HOSTIF_DATA1_HRD_MC2HOST1X_DATA**
7:6	**HOSTIF_DATA1_HRD_MC2HOST1X_TYPE**
5:2	**HOSTIF_DATA1_HRD_MC2HOST1X_CHANNEL**
1	**HOSTIF_DATA1_HRD_MC2HOST1X_STALL**
0	**HOSTIF_DATA1_HRD_MC2HOST1X_VALID**

MC_OBS_HOSTPROC_REG_0

Offset: 005h
Read/Write: RO
Reset: 0000.0000

Bit	Description
27	**hostproc_int_clear**
26	**hostproc_hrd_mc2host1x_intr**
25:22	**hostproc_hrd_channel**
21:20	**hostproc_hrd_type**
19	**hostproc_hrd_write**
18	**hostproc_hwr_stall**
17:2	**hostproc_reg_offset**
1	**hostproc_reg_rd_en**
0	**hostproc_reg_wr_en**

MC_OBS_HOSTPROC_OTHER_0

Offset: 006h
Read/Write: RO
Reset: 0000.0000

Bit	Description
20	**hostproc_ctxsw_ack_pending**
19	**hostproc_ctxsw_ack_write**
18	**hostproc_ctxsw_match**
17	**hostproc_ctxsw_req**
16	**hostproc_refcount_wr**
15	**hostproc_refcount_req**
14	**hostproc_raise_wr**
13	**hostproc_raise_req**
12:1	**hostproc_class_offset**
0	**hostproc_class_write_req**

MC_OBS_HOSTREQ_ADR_0

Offset: 007h
Read/Write: RO
Reset: 0000.0000

Host request

Bit	Description
26:3	**HOSTREQ_ADR_HOST1X2MC_ADR**
2	**HOSTREQ_ADR_HOST1X2MC_WE**
1	**HOSTREQ_ADR_HOST1X2MC_RDY**
0	**HOSTREQ_ADR_HOST1X2MC_REQ**

MC_OBS_HOSTREQ_BE_WDO0_0

Offset: 008h
Read/Write: RO
Reset: 0000.0000

Bit	Description
22:7	**HOSTREQ_BE_WDO0_HOST1X2MC_WDO**
6:3	**HOSTREQ_BE_WDO0_HOST1X2MC_BE**
2	**HOSTREQ_BE_WDO0_HOST1X2MC_WE**
1	**HOSTREQ_BE_WDO0_HOST1X2MC_RDY**
0	**HOSTREQ_BE_WDO0_HOST1X2MC_REQ**

MC_OBS_HOSTREQ_BE_WDO1_0

Offset: 009h
Read/Write: RO
Reset: 0000.0000

Bit	Description
22:7	**HOSTREQ_BE_WDO1_HOST1X2MC_WDO**
6:3	**HOSTREQ_BE_WDO1_HOST1X2MC_BE**
2	**HOSTREQ_BE_WDO1_HOST1X2MC_WE**
1	**HOSTREQ_BE_WDO1_HOST1X2MC_RDY**
0	**HOSTREQ_BE_WDO1_HOST1X2MC_REQ**

MC_OBS_ARB_0

Offset: 00ah
Read/Write: RO
Reset: 0000.0000

Arbiter

Bit	Description
27	**ARB_EMEM_SAMEWIN**
26	**ARB_EMEM_SPCWIN**
25	**ARB_EMEM_SPWIN**
24:23	**ARB_EMEM_BANKWIN**
22	**ARB_EMEM_RWWIN**
21:16	**ARB_IMEM_ALL_IRDY**
15:10	**ARB_IMEM_ALL_IANYREQ**
9:4	**ARB_IMEM_IANYREQ**
3	**ARB_ARB2SEQ_ERDY**
2	**ARB_ARB2SEQ_EANYREQ**
1	**ARB_ARB2SEQ_IRDY**
0	**ARB_ARB2SEQ_IANYREQ**

MC_OBS_SEQ_MAIN_0

Offset: 00bh
Read/Write: RO
Reset: 0000.0000

Sequencer - Request side

Bit	Description
21:16	**SEQ_EREQ_PC**
15:10	**SEQ_IREQ_PC**
9	**SEQ_EDORDY_GO**
8	**SEQ_IDORDY_GO**
7	**SEQ_EDORDY_STALL**
6	**SEQ_IDORDY_STALL**

Bit	Description
5	**SEQ_PC_DPATHS_SINGLE**
4	**SEQ_PC_ACCESS**
3	**ARB2SEQ_ERDY**
2	**ARB2SEQ_EANYREQ**
1	**ARB2SEQ_IRDY**
0	**ARB2SEQ_IANYREQ**

MC_OBS_SEQ_IMEM_0

Offset: 00ch
Read/Write: RO
Reset: 0000.0000

Bit	Description
20	**SEQ_IMEM_DFIFO_RG**
19	**SEQ_IMEM_PMEM_RG**
18:13	**SEQ_IMEM_DFIFO_REQ_ID**
12:7	**SEQ_IMEM_PMEM_REQ_ID**
6	**SEQ_IMEM_DFIFO_WE**
5	**SEQ_IMEM_DFIFO_RD_RDY**
4	**SEQ_IMEM_DFIFO_RD_REQ**
3	**SEQ_IMEM_PMEM_WE**
2	**SEQ_IMEM_PMEM_REQ**
1	**SEQ_IMEM_DFIFO_WR_RDY**
0	**SEQ_IMEM_DFIFO_WR_REQ**

MC_OBS_SEQ_EMEM_0

Offset: 00dh
Read/Write: RO
Reset: 0000.0000

Bit	Description
18:13	**SEQ_EMEM_DFIFO_REQ_ID**
12:7	**SEQ_EMEM_PMEM_REQ_ID**
6	**SEQ_EMEM_DFIFO_WE**
5	**SEQ_EMEM_DFIFO_RD_RDY**
4	**SEQ_EMEM_DFIFO_RD_REQ**
3	**SEQ_EMEM_PMEM_WE**
2	**SEQ_EMEM_PMEM_REQ**
1	**SEQ_EMEM_DFIFO_WR_RDY**
0	**SEQ_EMEM_DFIFO_WR_REQ**

MC_OBS_SEQ_RDI_0

Offset: 00eh
Read/Write: RO
Reset: 0000.0000

Sequencer - Return data side

Bit	Description
13:9	**SEQ_RDI_MC2EMC_RDI_ID**
8	**SEQ_RDI_MC2EMC_DIVLD**
7	**SEQ_RDI_RG_PICK**
6	**SEQ_RDI_RG_REQ**
5:1	**SEQ_RDI_MC2SRAM_RDI_ID**
0	**SEQ_RDI_MC2SRAM_DIVLD**

MC_OBS_SRAMIF_MAIN_0

Offset: 00fh
Read/Write: RO
Reset: 0000.0000

SRAM interface (including clock enables)

Bit	Description
18:3	**SRAMIF_MC2SRAM_ADR**
2	**SRAMIF_MC2SRAM_WE**
1	**SRAMIF_MC2SRAM_RDY**
0	**SRAMIF_MC2SRAM_REQ**

MC_OBS_SRAMIF_BE_0

Offset: 010h
Read/Write: RO
Reset: 0000.0000

Bit	Description
17:2	**SRAMIF_BE_MC2SRAM_BE**
1	**SRAMIF_BE_MC2SRAM_WE**
0	**SRAMIF_BE_MC2SRAM_REQ**

MC_OBS_SRAMIF_WDO0_0

Offset: 011h
Read/Write: RO
Reset: 0000.0000

Bit	Description
29:2	**SRAMIF_WDO0_MC2SRAM_WDO**
1	**SRAMIF_WDO0_MC2SRAM_WE**
0	**SRAMIF_WDO0_MC2SRAM_REQ**

MC_OBS_SRAMIF_WDO1_0

Offset: 012h
Read/Write: RO
Reset: 0000.0000

Bit	Description
29:2	**SRAMIF_WDO1_MC2SRAM_WDO**
1	**SRAMIF_WDO1_MC2SRAM_WE**
0	**SRAMIF_WDO1_MC2SRAM_REQ**

MC_OBS_SRAMIF_WDO2_0

Offset: 013h
Read/Write: RO
Reset: 0000.0000

Bit	Description
29:2	**SRAMIF_WDO2_MC2SRAM_WDO**
1	**SRAMIF_WDO2_MC2SRAM_WE**
0	**SRAMIF_WDO2_MC2SRAM_REQ**

MC_OBS_SRAMIF_WDO3_0

Offset: 014h
Read/Write: RO
Reset: 0000.0000

Bit	Description
29:2	**SRAMIF_WDO3_MC2SRAM_WDO**
1	**SRAMIF_WDO3_MC2SRAM_WE**
0	**SRAMIF_WDO3_MC2SRAM_REQ**

MC_OBS_SRAMIF_WDO4_0

Offset: 015h
Read/Write: RO
Reset: 0000.0000

Bit	Description
17:2	**SRAMIF_WDO4_MC2SRAM_WDO**
1	**SRAMIF_WDO4_MC2SRAM_WE**
0	**SRAMIF_WDO4_MC2SRAM_REQ**

MC_OBS_SRAMIF_DIVLD_0

Offset: 016h
Read/Write: RO
Reset: 0000.0000

Bit	Description
6	**SRAMIF_MC2SRAM_RG**
5:1	**SRAMIF_MC2SRAM_RDI_ID**
0	**SRAMIF_MC2SRAM_DIVLD**

MC_OBS_SRAMIF_RDI0_0

Offset: 017h
Read/Write: RO
Reset: 0000.0000

Bit	Description
28:1	**SRAMIF_RDI0_MC2SRAM_RDI**
0	**SRAMIF_RDI0_MC2SRAM_DIVLD**

MC_OBS_SRAMIF_RDI1_0

Offset: 018h
Read/Write: RO
Reset: 0000.0000

Bit	Description
28:1	**SRAMIF_RDI1_MC2SRAM_RDI**
0	**SRAMIF_RDI1_MC2SRAM_DIVLD**

MC_OBS_SRAMIF_RDI2_0

Offset: 019h
Read/Write: RO
Reset: 0000.0000

Bit	Description
28:1	**SRAMIF_RDI2_MC2SRAM_RDI**
0	**SRAMIF_RDI2_MC2SRAM_DIVLD**

MC_OBS_SRAMIF_RDI3_0

Offset: 01ah
Read/Write: RO
Reset: 0000.0000

Bit	Description
28:1	**SRAMIF_RDI3_MC2SRAM_RDI**
0	**SRAMIF_RDI3_MC2SRAM_DIVLD**

MC_OBS_SRAMIF_RDI4_0

Offset: 01bh
Read/Write: RO
Reset: 0000.0000

Bit	Description
16:1	**SRAMIF_RDI4_MC2SRAM_RDI**
0	**SRAMIF_RDI4_MC2SRAM_DIVLD**

MC_OBS_SRAMIF_CLKEN_0

Offset: 01ch
Read/Write: RO
Reset: 0000.0000

Bit	Description
24:9	**SRAMIF_CLKEN_MC2SRAM_BE**
8	**SRAMIF_CLKEN_SRAMCLKEN**
7	**SRAMIF_CLKEN_SRAM3_CLKEN**
6	**SRAMIF_CLKEN_SRAM2_CLKEN**
5	**SRAMIF_CLKEN_SRAM1_CLKEN**
4	**SRAMIF_CLKEN_SRAM0_CLKEN**
3	**SRAMIF_CLKEN_MC2SRAM_DIVLD**
1	**SRAMIF_CLKEN_MC2SRAM_WE**
0	**SRAMIF_CLKEN_MC2SRAM_REQ**

MC_OBS_EMCIF_MAIN_0

Offset: 01dh
Read/Write: RO
Reset: 0000.0000

EMC interface

Bit	Description
23:11	**EMCIF_MC2EMC_ROW**
10:9	**EMCIF_MC2EMC_BANK**
8:4	**EMCIF_MC2EMC_REQ_ID**
3	**EMCIF_MC2EMC_AP**
2	**EMCIF_MC2EMC_WE**
1	**EMCIF_MC2EMC_RDY**
0	**EMCIF_MC2EMC_REQ**

MC_OBS_EMCIF_ADR_0

Offset: 01eh
Read/Write: RO
Reset: 0000.0000

Bit	Description
24:18	**EMCIF_ADR_MC2EMC_COL**
17:5	**EMCIF_ADR_MC2EMC_ROW**
4:3	**EMCIF_ADR_MC2EMC_BANK**
2	**EMCIF_ADR_MC2EMC_WE**
1	**EMCIF_ADR_MC2EMC_RDY**
0	**EMCIF_ADR_MC2EMC_REQ**

MC_OBS_EMCIF_BE_0

Offset: 01fh
Read/Write: RO
Reset: 0000.0000

Bit	Description
18:3	**EMCIF_BE_MC2EMC_BE**
2	**EMCIF_BE_MC2EMC_WE**
1	**EMCIF_BE_MC2EMC_RDY**
0	**EMCIF_BE_MC2EMC_REQ**

MC_OBS_MCCIF_DC_0

Offset: 020h
Read/Write: RO
Reset: 0000.0000

Client interface

Bit	Description
19	**MCCIF_DIVLD_DISPLAYHC**
18	**MCCIF_DORDY_DISPLAYHC**
17	**MCCIF_CREDIT_DISPLAYHC**
16	**MCCIF_REQ_DISPLAYHC**
15	**MCCIF_DIVLD_DISPLAY1B**
14	**MCCIF_DORDY_DISPLAY1B**
13	**MCCIF_CREDIT_DISPLAY1B**
12	**MCCIF_REQ_DISPLAY1B**
11	**MCCIF_DIVLD_DISPLAY0C**
10	**MCCIF_DORDY_DISPLAY0C**
9	**MCCIF_CREDIT_DISPLAY0C**
8	**MCCIF_REQ_DISPLAY0C**
7	**MCCIF_DIVLD_DISPLAY0B**
6	**MCCIF_DORDY_DISPLAY0B**
5	**MCCIF_CREDIT_DISPLAY0B**
4	**MCCIF_REQ_DISPLAY0B**
3	**MCCIF_DIVLD_DISPLAY0A**
2	**MCCIF_DORDY_DISPLAY0A**
1	**MCCIF_CREDIT_DISPLAY0A**
0	**MCCIF_REQ_DISPLAY0A**

MC_OBS_MCCIF_DSP_0

Offset: 021h
Read/Write: RO
Reset: 0000.0000

Bit	Description
6	**MCCIF_DORDY_DSPW**
5	**MCCIF_CREDIT_DSPW**
4	**MCCIF_REQ_DSPW**
3	**MCCIF_DIVLD_DSPR**
2	**MCCIF_DORDY_DSPR**
1	**MCCIF_CREDIT_DSPR**
0	**MCCIF_REQ_DSPR**

MC_OBS_MCCIF_EPP_0

Offset: 022h
Read/Write: RO
Reset: 0000.0000

Bit	Description
16	**MCCIF_DORDY_EPPY**
15	**MCCIF_CREDIT_EPPY**
14	**MCCIF_REQ_EPPY**
13	**MCCIF_DORDY_EPPV**
12	**MCCIF_CREDIT_EPPV**
11	**MCCIF_REQ_EPPV**
10	**MCCIF_DORDY_EPPU**
9	**MCCIF_CREDIT_EPPU**
8	**MCCIF_REQ_EPPU**
7	**MCCIF_DIVLD_EPPVP**
6	**MCCIF_DORDY_EPPVP**
5	**MCCIF_CREDIT_EPPVP**
4	**MCCIF_REQ_EPPVP**
3	**MCCIF_DIVLD_EPPUP**
2	**MCCIF_DORDY_EPPUP**
1	**MCCIF_CREDIT_EPPUP**
0	**MCCIF_REQ_EPPUP**

MC_OBS_MCCIF_G2_0

Offset: 023h
Read/Write: RO
Reset: 0000.0000

Bit	Description
14	**MCCIF_DORDY_G2DW**
13	**MCCIF_CREDIT_G2DW**
12	**MCCIF_REQ_G2DW**
11	**MCCIF_DIVLD_G2DR**
10	**MCCIF_DORDY_G2DR**
9	**MCCIF_CREDIT_G2DR**
8	**MCCIF_REQ_G2DR**
7	**MCCIF_DIVLD_G2SR**
6	**MCCIF_DORDY_G2SR**
5	**MCCIF_CREDIT_G2SR**
4	**MCCIF_REQ_G2SR**
3	**MCCIF_DIVLD_G2PR**
2	**MCCIF_DORDY_G2PR**
1	**MCCIF_CREDIT_G2PR**
0	**MCCIF_REQ_G2PR**

MC_OBS_MCCIF_HC_0

Offset: 024h
Read/Write: RO
Reset: 0000.0000

Bit	Description
10	**MCCIF_DORDY_HOST1XW**
9	**MCCIF_CREDIT_HOST1XW**
8	**MCCIF_REQ_HOST1XW**
7	**MCCIF_DIVLD_HOST1XR**
6	**MCCIF_DORDY_HOST1XR**
5	**MCCIF_CREDIT_HOST1XR**
4	**MCCIF_REQ_HOST1XR**
3	**MCCIF_DIVLD_HOST1XDMAR**
2	**MCCIF_DORDY_HOST1XDMAR**
1	**MCCIF_CREDIT_HOST1XDMAR**
0	**MCCIF_REQ_HOST1XDMAR**

MC_OBS_MCCIF_I2SR_0

Offset: 025h
Read/Write: RO
Reset: 0000.0000

Bit	Description
2	**MCCIF_DORDY_I2SW**
1	**MCCIF_CREDIT_I2SW**
0	**MCCIF_REQ_I2SW**

MC_OBS_MCCIF_I2ST_0

Offset: 026h
Read/Write: RO
Reset: 0000.0000

Bit	Description
3	**MCCIF_DIVLD_I2SR**
2	**MCCIF_DORDY_I2SR**
1	**MCCIF_CREDIT_I2SR**
0	**MCCIF_REQ_I2SR**

MC_OBS_MCCIF_ISP_0

Offset: 027h
Read/Write: RO
Reset: 0000.0000

Bit	Description
6	**MCCIF_DORDY_ISPW**
5	**MCCIF_CREDIT_ISPW**
4	**MCCIF_REQ_ISPW**
3	**MCCIF_DIVLD_ISPR**
2	**MCCIF_DORDY_ISPR**
1	**MCCIF_CREDIT_ISPR**
0	**MCCIF_REQ_ISPR**

MC_OBS_MCCIF_JE_0

Offset: 028h
Read/Write: RO
Reset: 0000.0000

Bit	Description
6	**MCCIF_DORDY_JPEGEW**
5	**MCCIF_CREDIT_JPEGEW**
4	**MCCIF_REQ_JPEGEW**
3	**MCCIF_DIVLD_JPEGER**
2	**MCCIF_DORDY_JPEGER**
1	**MCCIF_CREDIT_JPEGER**
0	**MCCIF_REQ_JPEGER**

MC_OBS_MCCIF_ME_0

Offset: 029h
Read/Write: RO
Reset: 0000.0000

Bit	Description
24	**MCCIF_DORDY_VLCDMAW**
23	**MCCIF_CREDIT_VLCDMAW**
22	**MCCIF_REQ_VLCDMAW**
21	**MCCIF_DORDY_ACDCW**
20	**MCCIF_CREDIT_ACDCW**

Bit	Description
19	**MCCIF_REQ_ACDCW**
18	**MCCIF_DORDY_YUVWR**
17	**MCCIF_CREDIT_YUVWR**
16	**MCCIF_REQ_YUVWR**
15	**MCCIF_DIVLD_VLCDMAR**
14	**MCCIF_DORDY_VLCDMAR**
13	**MCCIF_CREDIT_VLCDMAR**
12	**MCCIF_REQ_VLCDMAR**
11	**MCCIF_DIVLD_ACDCR**
10	**MCCIF_DORDY_ACDCR**
9	**MCCIF_CREDIT_ACDCR**
8	**MCCIF_REQ_ACDCR**
7	**MCCIF_DIVLD_YUVREF**
6	**MCCIF_DORDY_YUVREF**
5	**MCCIF_CREDIT_YUVREF**
4	**MCCIF_REQ_YUVREF**
3	**MCCIF_DIVLD_YUVCUR**
2	**MCCIF_DORDY_YUVCUR**
1	**MCCIF_CREDIT_YUVCUR**
0	**MCCIF_REQ_YUVCUR**

MC_OBS_MCCIF_MPD_0

Offset: 02ah
Read/Write: RO
Reset: 0000.0000

Bit	Description
13	**MCCIF_DORDY_RECON**
12	**MCCIF_CREDIT_RECON**
11	**MCCIF_REQ_RECON**
10	**MCCIF_DORDY_POSTPROC**
9	**MCCIF_CREDIT_POSTPROC**
8	**MCCIF_REQ_POSTPROC**
7	**MCCIF_DIVLD_MOTCOMP**
6	**MCCIF_DORDY_MOTCOMP**
5	**MCCIF_CREDIT_MOTCOMP**
4	**MCCIF_REQ_MOTCOMP**
3	**MCCIF_DIVLD_DBR**
2	**MCCIF_DORDY_DBR**
1	**MCCIF_CREDIT_DBR**
0	**MCCIF_REQ_DBR**

MC_OBS_MCCIF_NV_0

Offset: 02bh
Read/Write: RO
Reset: 0000.0000

Bit	Description
17	**MCCIF_DORDY_GRDWZ**
16	**MCCIF_CREDIT_GRDWZ**
15	**MCCIF_REQ_GRDWZ**
14	**MCCIF_DORDY_GRDWC**
13	**MCCIF_CREDIT_GRDWC**
12	**MCCIF_REQ_GRDWC**
11	**MCCIF_DIVLD_GRDFZ**
10	**MCCIF_DORDY_GRDFZ**
9	**MCCIF_CREDIT_GRDFZ**
8	**MCCIF_REQ_GRDFZ**
7	**MCCIF_DIVLD_GRDFT**
6	**MCCIF_DORDY_GRDFT**
5	**MCCIF_CREDIT_GRDFT**
4	**MCCIF_REQ_GRDFT**
3	**MCCIF_DIVLD_GRDFC**
2	**MCCIF_DORDY_GRDFC**
1	**MCCIF_CREDIT_GRDFC**
0	**MCCIF_REQ_GRDFC**

MC_OBS_MCCIF_SD_0

Offset: 02ch
Read/Write: RO
Reset: 0000.0000

Bit	Description
6	**MCCIF_DORDY_SDTW**
5	**MCCIF_CREDIT_SDTW**
4	**MCCIF_REQ_SDTW**
3	**MCCIF_DIVLD_SDTR**
2	**MCCIF_DORDY_SDTR**
1	**MCCIF_CREDIT_SDTR**
0	**MCCIF_REQ_SDTR**

MC_OBS_MCCIF_VI_0

Offset: 02dh
Read/Write: RO
Reset: 0000.0000

Bit	Description
15	**MCCIF_DORDY_VIWY**
14	**MCCIF_CREDIT_VIWY**
13	**MCCIF_REQ_VIWY**
12	**MCCIF_DORDY_VIWV**
11	**MCCIF_CREDIT_VIWV**
10	**MCCIF_REQ_VIWV**
9	**MCCIF_DORDY_VIWU**
8	**MCCIF_CREDIT_VIWU**
7	**MCCIF_REQ_VIWU**
6	**MCCIF_DORDY_VIWSB**
5	**MCCIF_CREDIT_VIWSB**
4	**MCCIF_REQ_VIWSB**
3	**MCCIF_DIVLD_VIRUV**
2	**MCCIF_DORDY_VIRUV**
1	**MCCIF_CREDIT_VIRUV**
0	**MCCIF_REQ_VIRUV**

MC_OBS_CIF_DC_0

Offset: 02eh
Read/Write: RO
Reset: 0000.0000

Bit	Description
24	**CIF_HP_DISPLAYHC**
23	**CIF_TM_DISPLAYHC**
22	**CIF_RDY_DISPLAYHC**
21	**CIF_EREQ_DISPLAYHC**
20	**CIF_IREQ_DISPLAYHC**
19	**CIF_HP_DISPLAY1B**
18	**CIF_TM_DISPLAY1B**
17	**CIF_RDY_DISPLAY1B**
16	**CIF_EREQ_DISPLAY1B**
15	**CIF_IREQ_DISPLAY1B**
14	**CIF_HP_DISPLAY0C**
13	**CIF_TM_DISPLAY0C**
12	**CIF_RDY_DISPLAY0C**
11	**CIF_EREQ_DISPLAY0C**
10	**CIF_IREQ_DISPLAY0C**
9	**CIF_HP_DISPLAY0B**
8	**CIF_TM_DISPLAY0B**
7	**CIF_RDY_DISPLAY0B**

Bit	Description
6	**CIF_EREQ_DISPLAY0B**
5	**CIF_IREQ_DISPLAY0B**
4	**CIF_HP_DISPLAY0A**
3	**CIF_TM_DISPLAY0A**
2	**CIF_RDY_DISPLAY0A**
1	**CIF_EREQ_DISPLAY0A**
0	**CIF_IREQ_DISPLAY0A**

MC_OBS_CIF_DSP_0

Offset: 02fh
Read/Write: RO
Reset: 0000.0000

Bit	Description
9	**CIF_HP_DSPW**
8	**CIF_TM_DSPW**
7	**CIF_RDY_DSPW**
6	**CIF_EREQ_DSPW**
5	**CIF_IREQ_DSPW**
4	**CIF_HP_DSPR**
3	**CIF_TM_DSPR**
2	**CIF_RDY_DSPR**
1	**CIF_EREQ_DSPR**
0	**CIF_IREQ_DSPR**

MC_OBS_CIF_EPP_0

Offset: 030h
Read/Write: RO
Reset: 0000.0000

Bit	Description
24	**CIF_HP_EPPY**
23	**CIF_TM_EPPY**
22	**CIF_RDY_EPPY**
21	**CIF_EREQ_EPPY**
20	**CIF_IREQ_EPPY**
19	**CIF_HP_EPPV**
18	**CIF_TM_EPPV**
17	**CIF_RDY_EPPV**
16	**CIF_EREQ_EPPV**
15	**CIF_IREQ_EPPV**
14	**CIF_HP_EPPU**
13	**CIF_TM_EPPU**
12	**CIF_RDY_EPPU**
11	**CIF_EREQ_EPPU**

Bit	Description
10	**CIF_IREQ_EPPU**
9	**CIF_HP_EPPVP**
8	**CIF_TM_EPPVP**
7	**CIF_RDY_EPPVP**
6	**CIF_EREQ_EPPVP**
5	**CIF_IREQ_EPPVP**
4	**CIF_HP_EPPUP**
3	**CIF_TM_EPPUP**
2	**CIF_RDY_EPPUP**
1	**CIF_EREQ_EPPUP**
0	**CIF_IREQ_EPPUP**

MC_OBS_CIF_G2_0

Offset: 031h
Read/Write: RO
Reset: 0000.0000

Bit	Description
15	**CIF_TM_G2DW**
14	**CIF_RDY_G2DW**
13	**CIF_EREQ_G2DW**
12	**CIF_IREQ_G2DW**
11	**CIF_TM_G2DR**
10	**CIF_RDY_G2DR**
9	**CIF_EREQ_G2DR**
8	**CIF_IREQ_G2DR**
7	**CIF_TM_G2SR**
6	**CIF_RDY_G2SR**
5	**CIF_EREQ_G2SR**
4	**CIF_IREQ_G2SR**
3	**CIF_TM_G2PR**
2	**CIF_RDY_G2PR**
1	**CIF_EREQ_G2PR**
0	**CIF_IREQ_G2PR**

MC_OBS_CIF_HC_0

Offset: 032h
Read/Write: RO
Reset: 0000.0000

Bit	Description
12	**CIF_HP_HOST1XW**
11	**CIF_TM_HOST1XW**
10	**CIF_RDY_HOST1XW**
9	**CIF_EREQ_HOST1XW**
8	**CIF_IREQ_HOST1XW**
7	**CIF_TM_HOST1XR**
6	**CIF_RDY_HOST1XR**
5	**CIF_EREQ_HOST1XR**
4	**CIF_IREQ_HOST1XR**
3	**CIF_TM_HOST1XDMAR**
2	**CIF_RDY_HOST1XDMAR**
1	**CIF_EREQ_HOST1XDMAR**
0	**CIF_IREQ_HOST1XDMAR**

MC_OBS_CIF_I2SR_0

Offset: 033h
Read/Write: RO
Reset: 0000.0000

Bit	Description
3	**CIF_TM_I2SW**
2	**CIF_RDY_I2SW**
1	**CIF_EREQ_I2SW**
0	**CIF_IREQ_I2SW**

MC_OBS_CIF_I2ST_0

Offset: 034h
Read/Write: RO
Reset: 0000.0000

Bit	Description
3	**CIF_TM_I2SR**
2	**CIF_RDY_I2SR**
1	**CIF_EREQ_I2SR**
0	**CIF_IREQ_I2SR**

MC_OBS_CIF_ISP_0

Offset: 035h
Read/Write: RO
Reset: 0000.0000

Bit	Description
9	**CIF_HP_ISPW**
8	**CIF_TM_ISPW**
7	**CIF_RDY_ISPW**
6	**CIF_EREQ_ISPW**
5	**CIF_IREQ_ISPW**
4	**CIF_HP_ISPR**
3	**CIF_TM_ISPR**
2	**CIF_RDY_ISPR**
1	**CIF_EREQ_ISPR**
0	**CIF_IREQ_ISPR**

MC_OBS_CIF_JE_0

Offset: 036h
Read/Write: RO
Reset: 0000.0000

Bit	Description
7	**CIF_TM_JPEGEW**
6	**CIF_RDY_JPEGEW**
5	**CIF_EREQ_JPEGEW**
4	**CIF_IREQ_JPEGEW**
3	**CIF_TM_JPEGER**
2	**CIF_RDY_JPEGER**
1	**CIF_EREQ_JPEGER**
0	**CIF_IREQ_JPEGER**

MC_OBS_CIF_ME_0

Offset: 037h
Read/Write: RO
Reset: 0000.0000

Bit	Description
27	**CIF_TM_VLCDMAW**
26	**CIF_RDY_VLCDMAW**
25	**CIF_EREQ_VLCDMAW**
24	**CIF_IREQ_VLCDMAW**
23	**CIF_TM_ACDCW**
22	**CIF_RDY_ACDCW**
21	**CIF_EREQ_ACDCW**
20	**CIF_IREQ_ACDCW**

Bit	Description
19	**CIF_TM_YUVWR**
18	**CIF_RDY_YUVWR**
17	**CIF_EREQ_YUVWR**
16	**CIF_IREQ_YUVWR**
15	**CIF_TM_VLCDMAR**
14	**CIF_RDY_VLCDMAR**
13	**CIF_EREQ_VLCDMAR**
12	**CIF_IREQ_VLCDMAR**
11	**CIF_TM_ACDCR**
10	**CIF_RDY_ACDCR**
9	**CIF_EREQ_ACDCR**
8	**CIF_IREQ_ACDCR**
7	**CIF_TM_YUVREF**
6	**CIF_RDY_YUVREF**
5	**CIF_EREQ_YUVREF**
4	**CIF_IREQ_YUVREF**
3	**CIF_TM_YUVCUR**
2	**CIF_RDY_YUVCUR**
1	**CIF_EREQ_YUVCUR**
0	**CIF_IREQ_YUVCUR**

MC_OBS_CIF_MPD_0

Offset: 038h
Read/Write: RO
Reset: 0000.0000

Bit	Description
16	**CIF_TM_RECON**
15	**CIF_RDY_RECON**
14	**CIF_EREQ_RECON**
13	**CIF_IREQ_RECON**
12	**CIF_TM_POSTPROC**
11	**CIF_RDY_POSTPROC**
10	**CIF_EREQ_POSTPROC**
9	**CIF_IREQ_POSTPROC**
8	**CIF_HP_MOTCOMP**
7	**CIF_TM_MOTCOMP**
6	**CIF_RDY_MOTCOMP**
5	**CIF_EREQ_MOTCOMP**
4	**CIF_IREQ_MOTCOMP**
3	**CIF_TM_DBR**
2	**CIF_RDY_DBR**
1	**CIF_EREQ_DBR**
0	**CIF_IREQ_DBR**

MC_OBS_CIF_NV_0

Offset: 039h
Read/Write: RO
Reset: 0000.0000

Bit	Description
19	**CIF_TM_GRDWZ**
18	**CIF_RDY_GRDWZ**
17	**CIF_EREQ_GRDWZ**
16	**CIF_IREQ_GRDWZ**
15	**CIF_TM_GRDWC**
14	**CIF_RDY_GRDWC**
13	**CIF_EREQ_GRDWC**
12	**CIF_IREQ_GRDWC**
11	**CIF_TM_GRDFZ**
10	**CIF_RDY_GRDFZ**
9	**CIF_EREQ_GRDFZ**
8	**CIF_IREQ_GRDFZ**
7	**CIF_TM_GRDFT**
6	**CIF_RDY_GRDFT**
5	**CIF_EREQ_GRDFT**
4	**CIF_IREQ_GRDFT**
3	**CIF_TM_GRDFC**
2	**CIF_RDY_GRDFC**
1	**CIF_EREQ_GRDFC**
0	**CIF_IREQ_GRDFC**

MC_OBS_CIF_SD_0

Offset: 03ah
Read/Write: RO
Reset: 0000.0000

Bit	Description
7	**CIF_TM_SDTW**
6	**CIF_RDY_SDTW**
5	**CIF_EREQ_SDTW**
4	**CIF_IREQ_SDTW**
3	**CIF_TM_SDTR**
2	**CIF_RDY_SDTR**
1	**CIF_EREQ_SDTR**
0	**CIF_IREQ_SDTR**

MC_OBS_CIF_VI_0

Offset: 03bh
Read/Write: RO
Reset: 0000.0000

Bit	Description
24	**CIF_HP_VIWY**
23	**CIF_TM_VIWY**
22	**CIF_RDY_VIWY**
21	**CIF_EREQ_VIWY**
20	**CIF_IREQ_VIWY**
19	**CIF_HP_VIWV**
18	**CIF_TM_VIWV**
17	**CIF_RDY_VIWV**
16	**CIF_EREQ_VIWV**
15	**CIF_IREQ_VIWV**
14	**CIF_HP_VIWU**
13	**CIF_TM_VIWU**
12	**CIF_RDY_VIWU**
11	**CIF_EREQ_VIWU**
10	**CIF_IREQ_VIWU**
9	**CIF_HP_VIWSB**
8	**CIF_TM_VIWSB**
7	**CIF_RDY_VIWSB**
6	**CIF_EREQ_VIWSB**
5	**CIF_IREQ_VIWSB**
4	**CIF_HP_VIRUV**
3	**CIF_TM_VIRUV**
2	**CIF_RDY_VIRUV**
1	**CIF_EREQ_VIRUV**
0	**CIF_IREQ_VIRUV**

7.6 SD Registers

SD_CTXSW_0

Offset: 000h
Read/Write: R/W
Reset: f000.f000

Context switch register. Should be common to all modules. Includes the current channel/class (which is writable by SW) and the next channel/class (which the hardware sets when it receives a context switch).

Context switch works like this:

Any context switch request triggers an interrupt to the host and causes the new channel/class to be stored in NEXT_CHANNEL/NEXT_CLASS (see vmod/chexample). SW sees that there is a context switch interrupt and does the necessary operations to make the module ready to receive traffic from the new context. It clears the context switch interrupt and writes CURR_CHANNEL/CLASS to the same value as NEXT_CHANNEL/CLASS, which causes a context switch acknowledge packet to be sent to the host. This completes the context switch and allows the host to continue sending data to the module.

Bit	Description
31:28	**NEXT_CHANNEL: Next requested channel**
25:16	**NEXT_CLASS: Next requested class**
15:12	**CURR_CHANNEL: Current working channel, reset to 'invalid'**
9:0	**CURR_CLASS: Current working class**

SD_ENABLE_REG_0

Offset: 001h
Read/Write: R/W
Reset: 0000.0000

Bit	Description
15	**RDBITSWAP: Bit Swap on Rx data.** This bit can be used by the software to swap the bits in a byte before writing the data to the internal memory. 0 Bits within a byte are not swapped, 1 Bits within a byte are swapped. 0 = Non Swapped 1 = Swapped
14	**RDBYTESWAP: Byte Swap on Rx data.** This bit can be used by the software to swap the bytes before writing the data to the internal memory. 0 Consecutive bytes are not swapped, 1 Consecutive bytes are swapped. 0 = Non Swapped 1 = Swapped
13	**WRBITSWAP: Bit Swap on Tx data.** This bit can be used by the software to swap the bits in a byte before writing the data to the card. 0 Bits within a byte are not swapped, 1 Bits within a byte are swapped. 0 = Non Swapped 1 = Swapped

Bit	Description
12	**WRBYTESWAP: Byte Swap on Tx data.** This bit can be used by the software to swap the bytes before writing the data to the card. 0 Consecutive bytes are not swapped, 1 Consecutive bytes are swapped. 0 = GPIO Mode 1 = SD MODE
2	**GPIO63MOD: GPIO Mode Select for the GPIO63 pin.** If the SD module is being used in the single-bit mode pin gpio63 (Data3) can be used as GPIO. If this bit is "0" the gpio63 pin will be used as GPIO. 0 = GPIO Mode 1 = SD Mode
1	**GPIO6162MOD: GPIO Mode Select for pins gpio61 and gpio62.** If the SD module is being used in the single-bit mode pins gpio61 and gpio62 (Data1 and Data2) can be used as GPIOs. If this bit is "0" the gpio61 and gpio62 pins will be used as GPIOs. 0 = GPIO Mode 1 = SD Mode
0	**SDEN: SD Module Enable/GPIO Mode Select for pins gpio60, gpio64, and gpio65.** This bit is used to Enable/Disable the SD module. If the block is disabled the SD module logic will be in reset state and the GPIO60, 64 and 65 will be in the GPIO mode. The following is the pin map between the SD pins and the GPIOs. GPIO60 = Data 0 GPIO61 = Data 1 GPIO62 = Data 2 GPIO63 = Data 3 GPIO64 = Sdclk GPIO65 = Sdcmd. 0 = Disable 1 = Enable

SD_CONTROL_0

Offset: 002h
Read/Write: R/W
Reset: 0000.0000

Bit	Description
30	**SDRXDMAEN: SD Receive DMA Enable** This bit enables/disables the DMA transfer from SD Receive buffer to internal memory transfers. 0 = Disable_DMA 1 = Enable_DMA
28	**SDTXDMAEN: SD Transmit DMA Enable** This bit enables/disables the DMA transfer from internal memory to SD transmit buffers. 0 = Disable_DMA 1 = Enable_DMA
18	**SDIOEN: SDIO enable.** This bit should be set if the SDIO functionality is needed. 0 = Disabled 1 = Enabled
17	**RDWAITEN: Read Wait enable.** This bit can be used by the software to insert the Read wait control using DAT2 if the SD card supports the read wait control. The software should set this bit if it needs to stall the data between a read multiple block. The software can set this bit asynchronously. The hardware will force the DAT2 line to low only at the end of a block. It will de assert it as soon as the software resets this bit 0 = Disables Read Wait control 1 = Enables Read Wait control.

Bit	Description
16	**SDIOSUSINTEN: Interrupt Enable during Suspend mode (for SDIO only)** The software can use this bit to enable the interrupt sampling on the DAT[1] line when the host puts one of the SDIO functions into suspend mode. 0 = NO_SAMPLE_IRQ (HW does not sample the DAT1/IRQ line) 1 = SAMPLE_IRQ (HW samples the DAT1/IRQ line)
11	**RESP_SRST: Response FIFO Software reset.** This bit is set by the software to reset the Response FIFO. Software should write a "1" to reset the FIFO and clear it later. 1 = RX_FIFO_RST
10	**SDRXFFSWRST: Receive FIFO Software reset.** This bit is set by the software to reset the Receive FIFO. Software should write a "1" to reset the FIFO and clear it later. 1 = RX_FIFO_RST
9	**SDTXFFSWRST: Transmit FIFO Software reset.** This bit is set by the software to reset the Transmit FIFO. Software should write a "1" to reset the FIFO and clear it later. 1 = TX_FIFO_RST
1	**STCLKCMD: This bit is used to start the SD clock to the SD card.** Writing a 1 to this bit will override all other power management logic and will force the clock to run. This bit is cleared by the hardware and will always return a "0". 1 = START_CLOCK
0	**ENDCLKCMD: this bit is used to stop the sd clock to the sd card.** Writing a 1 to this bit will override all other clock control logic and will stop the clock to the sd card immediately. the clock will be forced high. This bit has a higher priority over the bit "1" of this register. This bit is cleared by the hardware and will always return a "0".

SD_BLOCK_CONTROL_0

Offset: 003h
Read/Write: R/W
Reset: 0000.0000

Bit	Description
31:16	**NUMOFBLOCKS: Number of Blocks in multiple block transfer mode.** The Software will program the number of blocks that will be transferred for each multiple block transfer command. This value will be used by the SD DMA engine only if bit 11 of SD Command Start Register is set.
11:0	**BLKLEN: Block Length** The Software will program the Block length that will be used for the next data transfer. The block length is in number of bytes.

SD_READTIMEOUT_0

Offset: 004h
Read/Write: R/W
Reset: 0000.0000

SDMTR

Bit	Description
15:0	**RCVDTO: Read Time Out Value** The Software will program the number of SD clocks the host needs to wait for the next read block before it signals a read time out.

SD_RESPTIMEOUT_0

Offset: 005h
Read/Write: R/W
Reset: 0000.0000

SD15R

Bit	Description
24:0	**RESTO: Response Time Out Value** The Software will program the number of SD clocks the host needs to wait for the response before it signals a response time out.

SD_INTMASK_0

Offset: 006h
Read/Write: R/W
Reset: 0000.0000

SDMIR
Interrupt Mask for SD

Bit	Description
19	**WRITE_BUFFER_RDY_INT_EN: write buffer ready interrupt enable**
18	**READ_BUFFER_LW_INT_EN: read buffer low watermark interrupt enable interrupts for write buffer**
17	**READ_BUFFER_HW_INT_EN: read buffer high watermark interrupt enable**
16	**CTXSW_INT_EN: new for GoForce 5500: interrupts for read buffer**
11	**SDIO_INT_EN**
10	**SD_CARD_DETECT_INT_EN**
7	**RESP_RECEIVE_INT_EN**
6	**PROG_DONE_INT_EN**
5	**DATA_TRANS_DONE_INT_EN**
4	**RESP_CRC_ERR_INT_EN**
3	**READ_CRC_ERR_INT_EN**
2	**WRITE_CRC_ERR_INT_EN**
1	**RESP_TO_INT_EN**
0	**READ_TO_INT_EN**

SD_INTSTATUS_0

Offset: 007h
Read/Write: RO
Reset: 0000.0000

Interrupt Status Register

Bit	Description
19	**WRITE_BUFFER_RDY_INT: Write Buffer's Buffer-ready Interrupt** 0 Write buffer has not been filled 1 Write buffer has been filled and waiting in memory
18	**READ_BUFFER_LW_INT: Read Buffer Low Watermark Interrupt** 0 SD read buffer low watermark has not been reached 1 SD read buffer low watermark has been reached
17	**READ_BUFFER_HW_INT: High Watermark Interrupt for the Read Buffer** 0 SD read buffer high watermark has not been reached 1 SD read buffer high watermark has been reached
16	**CTXSW_INT – Context Switch Interrupt** 0 SD has not detected a context switch 1 SD detected a context switch
15	**RESP_FIFO_STATUS_INT – Response FIFO status Interrupt** This bit specify that the Response FIFO IS FULL. The status bit is sticky and only gets cleared if written. 0 Response FIFO is not full 1 Response FIFO is full
14:13	**ERROR_CODE_STATUS: these bits specify whether any error occured during writes to SD card and if so they specify the type of error.** SD host sets these bits 00 no error 01 CRC error 10 write error 11 illegal
12	**SD_CLOCK_STATUS –SD Clock Status** **Status of sd clock to sd device. When this bit is enabled, sd host does not send an interrupt.** 0 SD clock has stopped 1 SD clock is running
11	**SDIO_INT – SDIO Interrupt** 0 The sdio interrupt is not active 1 The sdio interrupt is pending
10	**SD_CARD_DETECT_INT – SD Card Detect Interrupt** 0 The sd card is not inserted 1 The sd card is inserted
7	**RESP_RECEIVE_INT – Response Receive Interrupt** 0 Command response has yet to be recieved 1 Command response has been received.
6	**PROG_DONE_INT – Programming Done Interrupt** This bit gets set after the data0 is pulled high (card indicating the previous write block has been programmed) at the end of completion of the last write block. The SD Module sets this bit. This bit will be cleared by the software by writing a "1" to this bit. 0 Card programming is not yet done 1 Card programming is finished
5	**DATA_TRANS_DONE_INT – Data Transfer Done Interrupt** 0 Write data transfer is not yet done 1 Write data transfer is finished

Bit	Description
4	**RESP_CRC_ERR_INT – Response Error Interrupt** 0 The response received is correct 1 The received response has a crc error
3	**READ_CRC_ERR_INT – Read CRC Error Interrupt** 0 The received data block is correct 1 The received data block has a crc error
2	**WRITE_CRC_ERR_INT – Write CRC Error Interrupt** 0 Write transaction went through 1 Write data block as a crc error
1	**RESP_TO_INT – Response Time Out Interrupt** 0 No timeout on the command/response transation 1 Timeout occurred while waiting for the response from the card
0	**READ_TO_INT – Read Time Out Interrupt** 0 No time out on the read transaction 1 Timout occurred while waiting for the read data from the card

SD_CMD_ARG_0

Offset: 008h
Read/Write: R/W
Reset: 0000.0000

Command Argument

The Software will program the command argument in this register. These 32 bits along with the command number, Start bit, Transmitter bit, CRC and End bit constitute the 48-bit length of the command.

Bit	Description
31:24	**CMDreg4**
23:16	**CMDreg3**
15:8	**CMDreg2**
7:0	**CMDreg1**

SD_CMDSTART_0

Offset: 009h
Read/Write: R/W
Reset: 0000.0000

Bit	Description
15	**WIDEBUS: Wide Bus.** This bit specifies whether the data transfer of the current command is in wide bus mode (D3 to D0) or not. The wide bus can be used for higher throughputs. Single pin (D0) is used for the data transfer. Four pin means D3-D0 are used for the data transfers. 0 = SINGLE_PIN 1 = FOUR_PIN
14	**NOCMD: No command.** This bit if set will stop the SD module from triggering the command. This will enable programming this register without triggering a command. Use_Reg means aew Command is sent to the SD card based on the control parameters set in this register. N0_CMD means a new command is not sent to the SD card. 0 = USE_REG 1 = NO_CMD
13	**INIT: Send Initialization sequence.** This bit if set will trigger an initialization sequence of 80 clocks on the SD clock pin. 0 = UNCHANGED: The state of the SD clock pin is not changed 1 = DWN_STRM_INT: The Initialization sequence will be triggered downstream
12	**BUSYBIT: Busy Bit.** This bit specifies whether to expect a busy signal for the current command. 0 The current command will not be returned with a busy signal from the SD card. 1 The current command may be returned with a busy signal from the SD card.. 0 = BUSY 1 = IDLE
11	**NOBON: Number of Blocks Enable.** This bit enables the use of the Number of Blocks, specified in the SD_BLOCK_CONTROL_0[NUMOFBLOCKS] register, in the multiple block mode. 0 = NO_USE_NUMOFBLOCKS: The number of blocks can vary from the programmed value in the SD_BLOCK_CONTROL_0[NUMOFBLOCKS] 1 = USE_NUMOFBLOCKS: The number of blocks that will be transferred in a multiple block mode will equal the programmed value in the SD_BLOCK_CONTROL_0[NUMOFBLOCKS].

Bit	Description
10	**MULBLK: Multiple Block Mode.** This bit specifies whether the current data transfer (if any) is a multiple block or a single block operation. The size of the block and the number of blocks is programmed in the SDBLK register. 0 = SINGLE_BLOCK: The current operation is a single block operation 1 = MULTI_BLOCK: The current operation is a multiple block operation.
9	**WRRDB: Write/Read.** This bit specifies whether the current data transfer (if any) is a write or read operation. 0 = READ (Current operation is a read operation) 1 = WRITE (Current operation is a write operation)
8	**DATEN: Data Enable.** The software will set this bit if the current command involves a data transfer. 0 = CMD_NO_DATA (Current command does not involve any date transfer) 1 = CMD_DATA (Current command involves a data transfer)
7:6	**RESNUM: Form of the Expected Response.** The software will program the expected response number for the current command programmed in the bits 5-0 of this register. 00 No Response. 01 Format of the response is R1/R1b/R6/R5 10 Format of the response is R2 11 Format of the response is R3/R4.. 00 = NO_RSP (No Response) 01 = RSP_R1(Response in format R1/R1b/R6/R5) 10 = RSP_R2 (Response in format R2) 11 = RSP_R3 (Response in format R3/R4)
5:0	**CMDreg0: Command Number.** These bits specify the command number. All Application specific commands should be preceded by the APP_CMD (Command Number 55).

SD_RESPONSEFIFO_0

Offset: 00ah
Read/Write: RO
Reset: 0000.0000

Bit	Description
31:0	**RESP_DATAOUT: Response FIFO** Reads to the Response FIFO are done at this register address.

SD_TRANSMIT_DMA_STATUS_0

Offset: 00bh
Read/Write: RO
Reset: 0000.0000

Bit	Description
31	**TBUF_EMPTY: the read buffer is empty**
30:22	**TBUF_NUMBUFFERS: pending buffers in memory**
21:6	**TBUF_NUMBYTES: pending byes in current buffer**
5:2	**TBUF_NUMENTRIES: pending entries in data fifo**
1	**TBUF_WRAPPED: wrapped to base**
0	**SD_TDMA_EN: DMA Enable** This bit enables/disables the DMA transfer from internal memory to SD transmit buffers. 0 = DISABLE_DMA 1 = ENABLE_DMA

SD_TRANSMIT_ADDR_0

Offset: 00ch
Read/Write: R/W
Reset: 0000.0000

Bit	Description
25:0	**RB_BASE_ADDR: read buffer base address**

SD_TRANSMIT_BUF_CONTROL_0

Offset: 00dh
Read/Write: R/W
Reset: 0000.0000

Bit	Description
8:0	**RB_NUMBER_BUF: number of read buffers in transmit**

SD_TRANSMIT_BUF_READY_0

Offset: 00eh
Read/Write: R/W
Reset: 0000.0000

Bit	Description
1	**TXDMA_RB_LAST_BUFFER**
0	**TXDMA_RB_BUFFER_RDY**

SD_TRANSMIT_BUF_CONFIG_0

Offset: 00fh
Read/Write: R/W
Reset: 0000.0000

Bit	Description
1	**RB_WRAP**
0	**RB_STORE_HDR**

SD_TRANSMIT_READ_BUFFER_WATERMARK_0

Offset: 010h
Read/Write: R/W
Reset: 0000.01ff

Bit	Description
24:16	**LOW_WATERMARK**
8:0	**HIGH_WATERMARK**

SD_RECEIVE_BUF_CONFIG1_0 Offset: 011h
Read/Write: R/W
Reset: 0000.0000

Write buffer configuration

These go directly to the receive transaction DMA block which houses the write buffer

Bit	Description
29:25	**CL_WB_CHANNEL**
24:16	**CL_WB_NUMBER_BUF**
15:0	**CL_WB_BUFFER_SIZE**

SD_RECEIVE_BUF_CONFIG2_0

Offset: 012h
Read/Write: R/W
Reset: 0000.0000

Write buffer configuation

These go directly to the receive transaction DMA block which houses the write buffer

Bit	Description
6:2	**CL_WB_RAISE_VEC**
1	**CL_WB_RAISE_EN**
0	**CL_WB_DMA_EN**

SD_RECEIVE_BUF_CONFIG_BASE_ADDR_0

Offset: 013h
Read/Write: R/W
Reset: 0000.0000

Write buffer configuration base address of the first write buffer

Bit	Description
25:0	**CL_WB_BASE_ADDR: the base address for the data must be 32 bit byte aligned**

SD_RECEIVE_BUF_STATUS_0

Offset: 014h
Read/Write: RO
Reset: 0000.0000

Bit	Description
28	**RBUF_LAST_BUF: last buffer**
27:19	**RBUF_NUMBUFFERS: number of buffers pending in memory**
18:3	**RBUF_NUMBYTES: number of bytes in current buffer**
2	**RBUF_WRAPPED: wrapped to base**
1	**RBUF_BUF_FULL: Part of the Write Buffer Client Buffer Status.** The Write Buffer Client is waiting for an ACK from the DMA The buffer chain is full
0	**SD_RDMA_EN: DMA Enable** This bit enables/disables the DMA transfer from the SD Receive buffer to internal memory transfers. This bit reflects SDRXDMAEN. It is reset if there is a crc receive error 0 Disable DMA 1 Enable DMA

SD_RECEIVE_BUF_BSTATUS_0

Offset: 015h
Read/Write: RO
Reset: 0000.0000

Bit	Description
8:0	**RBUF_PEND_BUFCOUNT: number of buffers pending in memory**

SD_DATA_COUNT_0

Offset: 016h
Read/Write: RO
Reset: 0000.0000

Bit	Description
11:0	**BYTE_COUNT: current byte count transmitted or received**

SD_DATA_TRANSMIT_BCOUNT_0

Offset: 017h
Read/Write: RO
Reset: 0000.0000

Bit	Description
31:16	**RBLOCK_COUNT: current block count received**
15:0	**WBLOCK_COUNT: current block count transmitted**

SD_GPIO_DCONTROL_0

Offset: 018h
Read/Write: R/W
Reset: 0000.0fff

Bit	Description
11:0	**GPIO_IEN_OEN: 1:0 sd_data_0 input enable output enable Control.** These bits control sd_data_0 pin when it is used as general purpose input/output 00 Input disabled, Output enabled 01 Input disabled, Output disabled 10 Input enabled, Output enabled 11 Input enabled, Output disabled enum (ID_OD =0x0, ID_OE =0x1, IE_OD =0x2, IE_OE=0x3) 3:2 sd_data_1 input enable output enable Control. 5:4 sd_data_2 input enable output enable Control. 7:6 sd_data_3 input enable output enable Control. 9:8 sd_clk input enable output enable Control. 11:10 sd_cmd input enable output enable Control.

SD_SDGPIN_CONTROL_0

Offset: 019h
Read/Write: R/W
Reset: 0000.0000

Bit	Description
10	**PAD2SD_SDGP1** GPIO data from pad; read only
9	**PAD2SD_SDGP0** GPIO data from pad; read only
5	**SD2PAD_SDGP1** GPIO data driven to pin
4	**SD2PAD_SDGP1_OEN: GPIO Output Enable** 0 Output enabled 1 Output not enabled
3	**SD2PAD_SDGP1_IE: GPIO Input Enable** 0 Input enabled 1 Input not enabled
2	**SD2PAD_SDGP0: GPIO data driven to pin**
1	**SD2PAD_SDGP0_OEN: GPIO Output Enable** 0 Output enabled 1 Output not enabled
0	**SD2PAD_SDGP0_IE: GPIO Input Enable** 0 Input enabled 1 Input not enabled

SD_GPIO_IODATA_0

Offset: 01ah
Read/Write: R/W
Reset: 0000.0000

Bit	Description
21:16	**SD12RDAT: GPIO[65:60] pins output data.** These data bits can be output on the corresponding GPIO[65:60] pins if the corresponding output buffer is enabled (the corresponding SD11R[11:0] areprogrammed to be in output mode). 0 The corresponding GPIO[65:60] pin output data is low. 1 The corresponding GPIO[65:60] pin output data is high.
5:0	**GPIOIN: R. GPIO[65:60] pins input data (Read only)** These bits allow the host to read the state of the corresponding GPIO[65:60] pins input buffer when they are enabled (corresponding SDGPIO_DCONTROL[11:0] bits are programmed to be in input mode). If the corresponding GPIO[65:60] input buffers are disabled then the corresponding bits will return 1.

SD_TEST_CONTROL_0

Offset: 01bh
Read/Write: R/W
Reset: 0000.0000

Bit	Description
31:16	**CRCTESTDATA: Test Data Output** These bits will output the CRC selected by the bits 2-0 of this register.

Bit	Description
2:0	**DATACRCEN: Test Output Select** These bits will select the required CRC to be output in bits 31-16 of this register 000 CRC_22_16: Response CRC will be output in bits 22-16 of this register 001 D0_CRC: Data Line 0 CRC will be output in bits 31-16 of this register 010 D1_CRC: Data Line 1 CRC will be output in bits 31-16 of this register 011 D2_CRC: Data Line 2 CRC will be output in bits 31-16 of this register 100 D3_CRC: Data Line 3 CRC will be output in bits 31-16 of this register 101 Reserved 110 Reserved 111 Reserved

SD_RAISE_RESP_RECEIVED_0

Offset: 01ch
Read/Write: R/W
Reset: 0000.0000

Bit	Description
19:16	**RAISE_RESP_RECEIVED_CHANNEL**
4:0	**RAISE_RESP_RECEIVED_VECTOR**

SD_RAISE_COMMAND_DONE_0

Offset: 01dh
Read/Write: R/W
Reset: 0000.0000

Bit	Description
19:16	**RAISE_COMMAND_DONE_CHANNEL**
4:0	**RAISE_COMMAND_DONE_VECTOR**

SD_RAISE_READ_DONE_0

Offset: 01eh
Read/Write: R/W
Reset: 0000.0000

Bit	Description
19:16	**RAISE_READ_DONE_CHANNEL**
4:0	**RAISE_READ_DONE_VECTOR**

SD_RAISE_WRITE_DONE_0

Offset: 01fh
Read/Write: R/W
Reset: 0000.0000

Bit	Description
19:16	**RAISE_WRITE_DONE_CHANNEL**

Bit	Description
4:0	**RAISE_WRITE_DONE_VECTOR**

SD_REFCOUNT_0

Offset: 020h
Read/Write: R/W
Reset: 0000.0000

Bit	Description
15:0	**REF_COUNT_VAL**

SD_SD_MCCIF_FIFOCTRL_0

Offset: 021h
Read/Write: R/W
Reset: 0000.0000

Memory Client Interface Async Fifo Optimization Register

Memory Client Interface Fifo Control Register: The registers below allow to optimize the synchronization timing in the memory client asynchronous fifos. When they can be used depend onthe client and memory controller clock ratio. Additionally, the RDMC_RDFAST/RDCL_RDFAST fields can increase power consumption if the asynchronous fifo is implemented as a real RAM. There is no power impact on latch-based fifos. Flipflop-based fifos do not use these fields. See recommended settings below.

Important: The register fields can only be changed when the memory client async fifos are empty.

The register field ending with WRCL_MCLE2X (if any) can be set to improve async fifo synchronization on the write side by one client clock cycle if the memory controller clock frequency is less or equal to twice the client clock frequency:

mcclk_freq <= 2 * clientclk_freq

The register field ending with WRMC_CLLE2X (if any) can be set to improve async fifo synchronization on the write side by one memory controller clock cycle if the client clock frequency is less or equal to twice the memory controller clock frequency:

clientclk_freq <= 2 * mcclk_freq

The register field ending with RDMC_RDFAST (if any) can be set to improve async fifo synchronization on the read side by one memory controller clock cycle.

Warning: RDMC_RDFAST can be used along with WRCL_MCLE2X only when:

mcclk_freq <= clientclk_freq

The register field ending with RDCL_RDFAST (if any) can be set to improve async fifo synchronization on the read side by one client clock cycle.

Warning: RDCL_RDFAST can be used along with WRMC_CLLE2X only when:

clientclk_freq <= mcclk_freq

RECOMMENDED SETTINGS

Client writing to fifo, memory controller reading from fifo
- mcclk_freq <= clientclk_freq

You can enable both RDMC_RDFAST and WRCL_CLLE2X. If one of the fifos is a real RAM and power is a concern, you should avoid enabling RDMC_RDFAST.
- clientclk_freq < mcclk_freq <= 2 * clientclk_freq

You can enable RDMC_RDFAST or WRCL_MCLE2X, but because the client clock is slower, you should enable only WRCL_MCLE2X.
- 2 * clientclk_freq < mcclk_freq

You can only enable RDMC_RDFAST. If one of the fifos is a real RAM and power is a concern, you should avoid enabling RDMC_RDFAST.

Memory controller writing to fifo, client reading from fifo

- clientclk_freq <= mcclk_freq

You can enable both RDCL_RDFAST and WRMC_CLLE2X. If one of the fifos is a real RAM and power is a concern, you should avoid enabling RDCL_RDFAST.

- mcclk_freq < clientclk_freq <= 2 * mcclk_freq

You can enable RDCL_RDFAST or WRMC_CLLE2X, but because the memory controller clock is slower, you should enable only WRMC_CLLE2X.

- 2 * mcclk_freq < clientclk_freq

You can only enable RDCL_RDFAST. If one of the fifos is a real RAM and power is a concern, you should avoid enabling RDCL_RDFAST.

Bit	Description
3	**SD_MCCIF_RDCL_RDFAST** 0 = Disable 1 = Enable
2	**SD_MCCIF_WRMC_CLLE2X** 0 = Disable 1 = Enable
1	**SD_MCCIF_RDMC_RDFAST** 0 = Disable 1 = Enable
0	**SD_MCCIF_WRCL_MCLE2X** 0 = Disable 1 = Enable

7.7 VI Registers

VI_CTXSW_0

Offset: 000h
Read/Write: R/W
Reset: fffff000.f000

Context switch register.

Should be common to all modules. Includes the current channel/class (which is writable by SW) and the next channel/class (which the hardware sets when it receives a context switch).

Context switch works like this:

Any context switch request triggers an interrupt to the host and causes the new channel/class to be stored in NEXT_CHANNEL/NEXT_CLASS (see vmod/chexample). SW sees that there is a context switch interrupt and does the necessary operations to make the module ready to receive traffic from the new context. It clears the context switch interrupt and writes CURR_CHANNEL/CLASS to the same value as NEXT_CHANNEL/CLASS, which causes a context switch acknowledge packet to be sent to the host. This completes the context switch and allows the host to continue sending data to the module.

Bit	Description
31:28	**NEXT_CHANNEL: Next requested channel**
25:16	**NEXT_CLASS: Next requested class**
15:12	**CURR_CHANNEL: Current working channel, reset to 'invalid'**
9:0	**CURR_CLASS: Current working class**

VI_INTSTATUS_0

Offset: 001h
Read/Write: RO
Reset: 0000.0000

Bit	Description
0	**CTXSW_INT: Context switch interrupt status (clear on write)**

VI_VI_INPUT_CONTROL_0

Offset: 002h
Read/Write: R/W
Reset: 0000.0000

VI Input Control

Bit	Description
30	**V_COUNTER: Vertical Counter.** 0 = Enabled 1 = Disabled
29	**H_COUNTER: Horizontal Counter.** 0 = Enabled 1 = Disabled
28	**FIELD_TYPE: Odd/Even Field type.** 0 = Topodd 1 = Topeven
27	**FIELD_DETECT: Interlaced video Field Detection.** 0 = Disabled 1 = Enabled
26:25	**SYNC_FORMAT: Horizontal and Vertical Sync Format.** 0 = YUV422 1 = ITU656 2 = INTHVS
24	**VVS_IN_EDGE: VVS input signal active edge which is used.** 0 = Rising 1 = Falling
23	**VHS_IN_EDGE: VHS input signal active edge which is used.** 0 = Rising 1 = Falling
9:8	**YUV_INPUT_FORMAT: YUV Input Format.** 0 = UYVY 1 = VYUY 2 = YUYV 3 = YVYU
7:6	**HOST_FORMAT: Host data Format.** 0 = NONPLANAR 1 = PLANAR 2 = BAYER8 3 = BAYER12
5:2	**INPUT_PORT_FORMAT: Input port data Format.** 0 = YUV422 1 = Reserved_1 2 = Bayer 3 = Reserved_2 4 = Pattern_a 5 = Pattern_b 6 = Pattern_c 7 = Pattern_c_raw 8 = Reserved_3 9 = Reserved 10 = RGB565 11 = RGB888 12 = RGB444
1	**VIP_INPUT_ENABLE: VIP Input Enable.** 0 = Disabled 1 = Enabled

Bit	Description
0	**HOST_INPUT_ENABLE: Host Input Enable.** 0 = Disabled 1 = Enabled

VI_VI_CORE_CONTROL_0

Offset: 003h
Read/Write: R/W
Reset: 0000.0000

VI Core Control and Output to EPP/ISP

Bit	Description
19	**V_DOWNSCALING: Vertical Down-scaling.** 0 = Disabled 1 = Enabled
18	**V_AVERAGING: Vertical Averaging.** 0 = Disabled 1 = Enabled
17	**H_DOWNSCALING: Horizontal Down-scaling.** 0 = Disabled 1 = Enabled
16	**H_AVERAGING: Horizontal Averaging.** 0 = Disabled 1 = Enabled
11	**CSC_INPUT_SEL: Color Space Conversion Input select.** 0 = Yuv422post 1 = Yuv422pre
10	**PLANAR_CONV_INPUT_SEL: Planar Conversion Module Input select.** 0 = Yuv422post 1 = Yuv422pre
9:8	**INPUT_TO_CORE: Input to VI Core.** 0 = VIP 1 = HOST 2 = ISP
6:5	**ISP_DOWNSAMPLE: Downsample from YUV444 to YUV422.** 0 = COSITED_EVEN 1 = COSITED_ODD 2 = NONCOSITED 3 = AVERAGED
4:2	**OUTPUT_TO_EPP: Output to EPP enable.** 0 = DISABLED 1 = YUV444POST 2 = YUV444PRE 3 = YUV444ISP 4 = RGB
1:0	**OUTPUT_TO_ISP: Output to ISP.** 0 = Disabled 1 = VIP 2 = HOST

VI_VI_FIRST_OUTPUT_CONTROL_0

Offset: 004h
Read/Write: R/W
Reset: 0000.0000

VI Output Control of YUV/RGB and YUV420P

Bit	Description
21	**XY_SWAP: XY-Swap in internal memory**
20	**V_DIRECTION: V-direction in internal memory**
19	**H_DIRECTION: H-direction in internal memory**
18:17	**YUV_OUTPUT_FORMAT: YUV Output Format.** 0 = UYVY 1 = VYUY 2 = YUYV 3 = YVYU
16	**OUTPUT_BYTE_SWAP: Output Byte Swap.** 0 = Disabled 1 = Enabled
8	**LAST_PIXEL_DUPLICATION: For Planar Output Only, enabling this register.** 0 = Disabled 1 = Enabled
2:0	**OUTPUT_FORMAT: Output data Format.** 0 = RGB16 1 = RGB16D 2 = RGB24 3 = YUV422POST 4 = YUV422PRE 5 = YUV422P 6 = YUV420P 7 = YUV420PA

VI_VI_SECOND_OUTPUT_CONTROL_0

Offset: 005h
Read/Write: R/W
Reset: 0000.0000

VI Second Output Control of YUV422NP and RGB

Bit	Description
21	**SECOND_XY_SWAP: Second output's XY-Swap in internal memory**
20	**SECOND_V_DIRECTION: Second output's V-direction in internal memory**
19	**SECOND_H_DIRECTION: Second output's H-direction in internal memory**
18:17	**YUV_SECOND_OUTPUT_FORMAT: YUV Second Output Format.** 0 = UYVY 1 = VYUY 2 = YUYV 3 = YVYU

Bit	Description
16	**SECOND_OUTPUT_BYTE_SWAP: Output Byte Swap.** 0 = Disabled 1 = Enabled
2:0	**SECOND_OUTPUT_FORMAT: Secondary Output to MC.** 0 = RGB16 1 = RGB16D 2 = RGB24 3 = YUV422POST 4 = YUV422PRE 5 = JPEG_STREAM

VI_HOST_INPUT_FRAME_SIZE_0

Offset: 006h
Read/Write: R/W
Reset: 0000.0000

Host Input Frame Width

Input Frame Width and Height give the total input data dimensions. The VI input stage will cull/clip pixels outside the Active Region (see register VI_HOST_H_ACTIVE & VI_HOST_V_ACTIVE). The amount of data per frame is expected to be INPUT_WIDTH * INPUT_HEIGHT * the bytes per pixel (determined from the INPUT_HOST_FORMAT). For Planar, the BPP is 1 for the Y fifo, 1/2 for U and V. For non planar it is 2.

The Bayer data is treated as 1 byte per pixel, so if it is more, then the input width and the H_ACTIVE should be scaled accordingly, so that internally generated hsync and vsyncs for ISP are correct. For Bayer input, it is important to insert blanking data for horizontal and vertical, allowing ISP to do side band calculations.

Bit	Description
27:16	**INPUT_FRAME_HEIGHT: Host Input Frame Height** Specifies in terms of lines the height of the input data coming from host.
11:0	**INPUT_FRAME_WIDTH: Specifies in terms of pixels the width of the input data coming from host.**

VI_HOST_H_ACTIVE_0

Offset: 007h
Read/Write: R/W
Reset: 0000.0000

VI Horizontal Active

This register defines the horizontal active area of the input video source with respect to the internally generated horizontal sync. (This is for data coming in from host.)

Bit	Description
27:16	**HOST_H_ACTIVE_PERIOD: Horizontal Active Period** This parameter specifies the number of pixels in the horizontal active area. H_ACTIVE_START + H_ACTIVE_PERIOD should be less than 4096. This parameter should be be programmed with even number (bit 16 is ignored internally).
10:0	**HOST_H_ACTIVE_START: Horizontal Active Start (offset to active)** This parameter specifies the number of pixels to be discarded until the first active pixel. If programmed to 0, the first active pixel is the first pixel popped from the Host YUV FIFO.

VI_HOST_V_ACTIVE_0

Offset: 008h
Read/Write: R/W
Reset: 0000.0000

Vertical Active

This register defines the vertical active area of the input video source with respect to the internally generated vertical sync. (This is for data coming in from host.)

Bit	Description
27:16	**HOST_V_ACTIVE_PERIOD: Vertical Active Period** This parameter specifies the number of lines in the vertical active area. V_ACTIVE_START + V_ACTIVE_PERIOD should be less than 4096.
10:0	**HOST_V_ACTIVE_START: Vertical Active Start (offset to active)** This parameter specifies the number of horizontal sync active edges from vertical sync active edge to the first vertical active line. If programmed to 0, the first active line starts after the first horizontal sync active edge following the vertical sync active edge.

VI_VIP_H_ACTIVE_0

Offset: 009h
Read/Write: R/W
Reset: 0000.0000

VI Horizontal Active

This register defines the horizontal active area of the input video source with respect to horizontal sync. (This is for VIP data.)

Bit	Description
27:16	**VIP_H_ACTIVE_PERIOD: Horizontal Active Period** This parameter specifies the number of pixels in the horizontal active area. H_ACTIVE_START + H_ACTIVE_PERIOD should be less than 4096. This parameter should be be programmed with even number (bit 16 is ignored internally).
10:0	**VIP_H_ACTIVE_START: Horizontal Active Start (offset to active)** This parameter specifies the number of clock active edges from horizontal sync active edge to the first horizontal active pixel. If programmed to 0, thefirst active line starts after the first active clock edge following the horizontal sync active edge.

VI_VIP_V_ACTIVE_0

Offset: 00ah
Read/Write: R/W
Reset: 0000.0000

Vertical Active
This register defines the vertical active area of the input video source with respect to vertical sync. (This is for VIP data.)

Bit	Description
27:16	**VIP_V_ACTIVE_PERIOD: Vertical Active Period** This parameter specifies the number of lines in the vertical active area. V_ACTIVE_START + V_ACTIVE_PERIOD should be less than 4096.
10:0	**VIP_V_ACTIVE_START: Vertical Active Start (offset to active)** This parameter specifies the number of horizontal sync active edges from vertical sync active edge to the first vertical active line. If programmed to 0, the first active line starts after the first horizontal sync active edge following the vertical sync active edge.

VI_VI_PEER_CONTROL_0

Offset: 00bh
Read/Write: R/W
Reset: 0000.0000

VI Peer to Peer Control

For all fields except DMA_REQUEST:

00 = Disabled
01 = First memory
10 = Second memory

Bit	Description
7:6	**SB_CONTROL: VI to StretchBLT Control Bus enable.** 0 = Disabled 1 = First 2 = Second
5:4	**ENCODER_CONTROL: VI to JPEGE & MPEGE Control Bus enable.** 0 = Disabled 1 = First 2 = Second
3:2	**DISPLAY_CONTROL: VI to Display Control Bus enable.** 0 = Disabled 1 = First 2 = Second
1:0	**DMA_REQUEST: Host DMA Request enable at end of block.** 0 = Disabled 1 = Stream 2 = First 3 = Second

VI_HOST_DMA_WRITE_BUFFER_0

Offset: 00ch
Read/Write: R/W
Reset: 0000.0000

Host DMA Write Buffer Configuration Registers

Bit	Description
25	**DMA_ENABLE: DMA Enable.** 0 = Disabled 1 = Enabled
24:16	**BUFFER_NUMBER: Buffer Number**
15:0	**BUFFER_SIZE: Buffer Size**

VI_HOST_DMA_BASE_ADDRESS_0

Offset: 00dh
Read/Write: R/W
Reset: 0000.0000

Host DMA Write Buffer Configuration Registers

Bit	Description
25:0	**DMA_BASE_ADDR: Base Address**

VI_HOST_DMA_WRITE_BUFFER_STATUS_0

Offset: 00eh
Read/Write: RO
Reset: 0000.0000

Host DMA Write Buffer Status Register

Bit	Description
26:0	**WB_STATUS: Read Only**

VI_HOST_DMA_WRITE_PEND_BUFCOUNT_0

Offset: 00fh
Read/Write: RO
Reset: 0000.0000

Host DMA Write Buffer Pending Buffer Count

Bit	Description
8:0	**PEND_BUFCOUNT: Read Only**

VI_VB0_START_ADDRESS_FIRST_0

Offset: 010h
Read/Write: R/W
Reset: 0000.0000

Video Buffer O Start Address for First Output

FIRST OUTPUT Registers

These registers are used to setup the first of two memory outputs for VI Address Y, U, V; Frame size; Count; Size (line stride and block height); and Buffer Stride

Bit	Description
25:0	**VB0_START_ADDRESS_1: This is byte address of video buffer 0 if output data format is RGB or YUV non-planar.** This is byte address of video buffer 0 Y-plane if output data format is YUV planar.

VI_VB0_START_ADDRESS_U_0

Offset: 011h
Read/Write: R/W
Reset: 0000.0000

Video Buffer O Start Address U (linked to First Output)

Bit	Description
25:0	**VB0_START_ADDRESS_U: This is byte address of video buffer 0 u-plane if output data format is YUV planar.** Output data format is YUV planar.

VI_VB0_START_ADDRESS_V_0

Offset: 012h
Read/Write: R/W
Reset: 0000.0000

Video Buffer O Start Address V (linked to First Output)

Bit	Description
25:0	**VB0_START_ADDRESS_V: This is byte address of video buffer 0 V-plane if output data format is YUV planar.** Output data format is YUV planar.

VI_VB0_SCRATCH_ADDRESS_UV_0

Offset: 013h
Read/Write: R/W
Reset: 0000.0000

Video Buffer O Scratch Address UV (linked to First Output)

Bit	Description
25:0	**VB0_SCRATCH_ADDRESS_UV: If OUTPUT_FORMAT is YUV420PA, this is used.** This is byte address of video buffer 0 UV intermediate data is saved here during the YUV422 to YUV420PA conversion. The size allocated needs to match the FIRST_FRAME_WIDTH register setting

VI_FIRST_OUTPUT_FRAME_SIZE_0

Offset: 014h
Read/Write: R/W
Reset: 0000.0000

Width and height of first output frame

This is the size of the frame being written to memory. Apply decimation or averaging to calculate the output frame size. Whether or not downscaling is used, specify the size of the frame being written to memory.

Bit	Description
27:16	**FIRST_FRAME_HEIGHT: frame height in lines which VI needs to process**
11:0	**FIRST_FRAME_WIDTH: frame width in pixel which VI needs to process**

VI_VB0_COUNT_FIRST_0

Offset: 015h
Read/Write: R/W
Reset: 0000.0000

Video Buffer Set 0 Count for First Output

Bit	Description
7:0	**VB0_COUNT_1: Video Buffer Set 0 Count** This specifies the number of buffers in video buffer set 0.

VI_VB0_SIZE_FIRST_0

Offset: 016h
Read/Write: R/W
Reset: 0000.0000

Video Buffer Set 0 Size for First Output

Bit	Description
27:16	**VB0_V_SIZE_1: Video Buffer Set 0 Vertical Size** This specifies the number of lines in each buffer in video buffer set 0. This value will be divided by 2 for chroma buffers for YUV420 planar formats
11:0	**VB0_H_SIZE_1: Video Buffer Set 0 Horizontal Size** This parameter specifies the line stride (in pixels) for lines in the video buffer set 0. For YUV non-planar format, this parameter must be programmed as multiple of 2 pixels (bit 0 is ignored). For YUV planar format, this parameter must be programmed as multiple of 8 pixels (bits 2-0 are ignored) and it specifies the luma line stride or twice the chroma line stride. This value will be divided by 2 for chroma buffers for YUV422 and YUV420 planar formats

VI_VB0_BUFFER_STRIDE_FIRST_0

Offset: 017h
Read/Write: R/W
Reset: 0000.0000

Video Buffer Set 0 Buffer Stride

Bit	Description
31:30	**VB0_BUFFER_STRIDE_C: Video Buffer Set 0 Chroma Buffer Stride.** 0 = CBS1X 1 = CBS2X 2 = CBS4X
25:0	**VB0_BUFFER_STRIDE_L: Video Buffer Set 0 Luma Buffer Stride** This is luma buffer stride (in bytes)

VI_VB0_START_ADDRESS_SECOND_0

Offset: 018h
Read/Write: R/W
Reset: 0000.0000

Video Buffer O Start Address for Second Output

SECOND OUTPUT Registers

These registers are used to setup the second of two memory outputs for VI Address; Frame size; Count; Size (line stride and block height); and Buffer Stride.

Bit	Description
25:0	**VB0_START_ADDRESS_2: This is byte address of video buffer 0 if output data format is RGB or YUV non-planar.** This is byte address of video buffer 0. This output data is read by the SB

VI_SECOND_OUTPUT_FRAME_SIZE_0

Offset: 019h
Read/Write: R/W
Reset: 0000.0000

Width and height of second output frame

Bit	Description
27:16	**SECOND_FRAME_HEIGHT: frame height in lines which VI needs to process**
11:0	**SECOND_FRAME_WIDTH: frame width in pixel which VI needs to process**

VI_VB0_COUNT_SECOND_0

Offset: 01ah
Read/Write: R/W
Reset: 0000.0000

Video Buffer Set 0 Count for Second Output

Bit	Description
7:0	**VB0_COUNT_2:** This specifies the number of buffers in video buffer set 0.

VI_VB0_SIZE_SECOND_0

Offset: 01bh
Read/Write: R/W
Reset: 0000.0000

Video Buffer Set 0 Size for Second Output

Bit	Description
27:16	**VB0_V_SIZE_2: Video Buffer Set 0 Vertical Size** This specifies the number of lines in each buffer in video buffer set 0.
11:0	**VB0_H_SIZE_2: Video Buffer Set 0 Horizontal Size** This parameter specifies the line stride (in pixels) for lines in the video buffer set 0. For YUV non-planar format, this parameter must be programmed as multiple of 2 pixels (bit 0 is ignored).

VI_VB0_BUFFER_STRIDE_SECOND_0

Offset: 01ch
Read/Write: R/W
Reset: 0000.0000

Video Buffer Set 0 Buffer Stride for Second Output

Bit	Description
25:0	**VB0_BUFFER_STRIDE_2: Video Buffer Set 0 Luma Buffer Stride** This is luma buffer stride (in bytes)

VI_MC_HP_THRESHOLD_0

Offset: 01dh
Read/Write: R/W
Reset: 0000.0000

Memory Client High Priority Threshold

Bit	Description
30:24	**CBW_VIWSB2MC_HPTH: Write High Priority Threshold for Second Output** If the fifo threshold value is 0, the high priority signal is disabled
22:16	**CBW_VIWY2MC_HPTH: Write High Priority Threshold for First Output** If the fifo threshold value is 0, the high priority signal is disabled
13:8	**CBR_VIRUV2MC_HPTM: Read High Priority Timer** If the timer value is 0, the high priority signal is disabled
4:0	**CBR_VIRUV2MC_HPTH: Read High Priority Threshold** If the fifo threshold value is 0, the high priority signal is disabled

VI_H_LPF_CONTROL_0

Offset: 01eh
Read/Write: R/W
Reset: 0240.0240

VI Horizontal Low-Pass Filter (LPF) Control

This register controls horizontal low-pass filtering which can be enabled to improve quality of the decimated image. The only valid programming values for this register are

0x02400240 No filtering
0x0DBE092E 1-HPF^3
0x01B60126 1-HPF^2
0x05B70127 (1-HPF^2+LPF)/2
0x06480248 LPF
0x04910001 (LPF+LPF^2)/2
0x00900000 LPF^2
0x04980008 LPF^3
0x07980308 LPF^2 * (0.5,0,0.5)
0x07f80368 LPF * (0.5,0,0.5) * (2,-3,2)

The above list is ordered from the widest band-pass filter to the narrowest band-pass filter.

Bit	Description
28:16	**H_LPF_C: Horizontal LPF Chrominance filter** This controls low pass filter for U V data.
12:0	**H_LPF_L: Horizontal LPF Luminance filter** This controls low pass filter for Y data.

VI_H_DOWNSCALE_CONTROL_0

Offset: 01fh
Read/Write: R/W
Reset: 0000.0000

VI Horizontal Down-scaling Control

Horizontal pixel processing starts with horizontal low-pass filtering.

Following horizontal low-pass filtering, horizontal down-scaling (decimation) can then be performed with or without horizontal averaging. If horizontal down-scaling (decimation) is performed without horizontal averaging, the down-scaling factor is specified by input active period and output frame size. Because the VI has two input methods (VIP and HOST) and two memory outputs, there are mux selects to indicate which registers to use in calculating the input and output frame sizes. If horizontal down-scaling is performed with horizontal averaging, the down-scaling factors are limited to few factors determined by H_AVG_CONTROL. When enabling averaging PLEASE be careful that the input and output ratios match the formula for the averaging decimation ratio exactly to the pixel/line. The formula for each of the Averaging Decimation Ratio is as follows:

Averaging Decimation Formalae

x = input size
y(x) = output size
2-pixel averaging and 1/2 downscaling: y(x) = Floor(x/2)
4-pixel averaging and 1/3 downscaling: y(x) = Floor((x-1)/3)
4-pixel averaging and 1/4 downscaling: y(x) = Floor(x/4)
8-pixel averaging and 1/7 downscaling: y(x) = Floor((x-1)/7)
8-pixel averaging and 1/8 downscaling: y(x) = Floor(x/8)

Horizontal Decimation Algorithm:

The Horizontal Decimator decides which pixels to drop by using a simple DDA algorithm. The accumulator will continue to add the value of the output width (numerator) for each pixel until the sum is equal or greater than the input width (denominator). When the sum is greater or equal to the input width (denominator), the hardware will flag that pixel as a pixel to be written out to memory. At the same time, the input width (denominator) will be subtracted from the sum and the difference will be loaded back into the accumulator for the next line. By default the accumulator is initialized with 0's upon reset. However the user can set the H_DEC_INIT_VAL to initialize the accumulator with a certain value from 0 to the input width (denominator). Any H_DEC_INIT_VAL that is greater or equal to the difference of the input width (denominator) and the output width (numerator) will cause the first pixel to be written out to memory. This register shifts the phase of the decimation pattern.

Bit	Description
27:16	**H_DEC_INIT_VAL: Horizontal Decimation Accumulator Initial Value** The user may initialized the H-Dec accumulator with a value between 0-(H_ACTIVE_PERIOD) to change the phase of the decimation pattern. This will allow the user to decide which is the first pixel to keep.
10:8	**H_AVG_CONTROL: Horizontal Averaging Control.** 0 = A2D2 1 = A4D3 2 = A4D4 3 = A8D7 4 = A8D8

Bit	Description
1	**OUTPUT_H_SIZE_SEL: Output Horizontal Size Select.** 0 = FIRST 1 = SECOND
0	**INPUT_H_SIZE_SEL: Input Horizontal Size Select.** 0 = VIP 1 = HOST

VI_V_DOWNSCALE_CONTROL_0

Offset: 020h
Read/Write: R/W
Reset: 0000.0000

VI Vertical Down-scaling Control

Vertical processing consists of optional vertical down-scaling (decimation) which can be performed with or without vertical averaging. If vertical down-scaling (decimation) is performed without vertical averaging, the down-scaling factor is specified by input active period and output frame size. Because the VI has two input methods (VIP and HOST) and two memory outputs, there are mux selects to indicate which registers to use in calculating the input and output frame sizes. If horizontal down-scaling is performed with vertical averaging, the down-scaling factors are limited to few factors determined by V_AVG_CONTROL. When enabling averaging PLEASE be careful that the input and output ratios match the formula for the averaging decimation ratio exactly to the pixel/line. The formula for each of the Averaging Decimation Ratio is as follows:

Averaging Decimation Formalae

x = input size
y(x) = output size
2-pixel averaging and 1/2 downscaling: y(x) = Floor(x/2)
4-pixel averaging and 1/3 downscaling: y(x) = Floor((x-1)/3)
4-pixel averaging and 1/4 downscaling: y(x) = Floor(x/4)
8-pixel averaging and 1/7 downscaling: y(x) = Floor((x-1)/7)
8-pixel averaging and 1/8 downscaling: y(x) = Floor(x/8)

Vertical Decimation Algorithm: (same as the Horizontal Decimation Algorithm)

The Vertical Decimator decides which pixels to drop by using a simple DDA algorithm. The accumulator will continue to add the value of the output height (numerator) for each line until the sum is equal or greater than the input height (denominator). When the sum is greater or equal to the input height (denominator), the hardware will flag that line as a line to be written out to memory. At the same time, the input height (denominator) will be subtracted from the sum and the difference will be loaded back into the accumulator for the next line. By default the accumulator is initialized with 0's upon reset. However the user can set the V_DEC_INIT_VAL to initialize the accumulator with a certain value from 0 to the input height (denominator). Any V_DEC_INIT_VAL that is greater or equal to the difference of the input height (denominator) and the output height (numerator) will cause the first line to be written out to memory. This register shifts the phase of the decimation pattern.

Bit	Description
28	**IGNORE_FIELD: Specifies whether odd/even field affects vertical.** 0 = Disabled 1 = Enabled
27:16	**V_DEC_INIT_VAL: Vertical Decimation Accumulator Initial Value** The user may initialized the V-Dec accumulator with a value between 0-(V_ACTIVE_PERIOD) to change the phase of the decimation pattern. This will allow the user to decide which is the first line to keep.
13	**MULTI_TAP_V_AVG_FILTER: Multi-Tap Vertical Averaging Filter.** 0 = Disabled 1 = Enabled
12	**FLEXIBLE_VSCALE: Flexible Vertical Scaling.** 0 = Disabled 1 = Enabled
10:8	**V_AVG_CONTROL: Vertical Averaging Control.** 0 = A2D2 1 = A4D3 2 = A4D4 3 = A8D7 4 = A8D8
1	**OUTPUT_V_SIZE_SEL: Output Vertical Size Select.** 0 = First 1 = Second
0	**INPUT_V_SIZE_SEL: Input Vertical Size Select.** 0 = VIP 1 = HOST

VI_CSC_Y_0

Offset: 021h
Read/Write: R/W
Reset: 0000.0000

CSC Y Offset and Gain
Color Space Conversion coefficients.
The CSC can be used for YUV to RGB conversion with brightness and hue/saturation control.

For Y color, the Y offset is applied first and saturation (clipping) is performed immediately after the Y offset is applied.

R = sat(KYRGB * sat(Y + YOF) + KUR * U + KVR * V)
G = sat(KYRGB * sat(Y + YOF) + KUG * U + KVG * V)
B = sat(KYRGB * sat(Y + YOF) + KUB * U + KVB * V)
Saturation and rounding is performed in the range of 0 to 255 for the above equations.

Typical values are:

YOF = -16.000, KYRGB = 1.1644
KUR = 0.0000, KVR = -1.5960
KUG = -0.3918, KVG = -0.8130
KUB = 2.0172, KVB = 0.0000
KUR and KVB are typically 0.0000 but they may be programmed non-zero for hue rotation.

The CSC can also take RGB input, in which case YOF, KVB, KUG, KUR should be programmed to 0 and KYRGB will be forced to 0 by the hardware for generating R and B. KYRGB will not be forced to 0 for generating G. KVR, KYRGB, and KUB can be programmed to 1.0 or used as gain control for R, G, B correspondingly. Note that color value ranges from 0 to 255 for Y, R, G, B and -128 to 127 for U and V.

Bit	Description
25:16	**KYRGB: Y Gain for R, G, B colors in 2.8 format**
7:0	**YOF: Y Offset in s.7.0 format**

VI_CSC_UV_R_0

Offset: 022h
Read/Write: R/W
Reset: 0000.0000

CSC U & V coefficent for R

Bit	Description
26:16	**KVR: V coefficients for R in s.2.8 format**
10:0	**KUR: U coefficients for R in s.2.8 format**

VI_CSC_UV_G_0

Offset: 023h
Read/Write: R/W
Reset: 0000.0000

CSC U & V coefficent for G

Bit	Description
25:16	**KVG: V coefficients for G in s.1.8 format**
9:0	**KUG: U coefficients for G in s.1.8 format**

VI_CSC_UV_B_0

Offset: 024h
Read/Write: R/W
Reset: 0000.0000

CSC U & V coefficent for B

Bit	Description
26:16	**KVB: V coefficients for B in s.2.8 format**
10:0	**KUB: U coefficients for B in s.2.8 format**

VI_CSC_ALPHA_0

Offset: 025h
Read/Write: R/W
Reset: 0000.0000

RGB Color Space Converter Alpha value

Bit	Description
7:0	**RGB888_ALPHA:** When output format to memory is selected for RGB888, the pixel data is 32-bit aligned. The value programmed here will be appended to the RGB888 data as the 8 MSBs and can be used as an alpha value.

VI_HOST_VSYNC_0

Offset: 026h
Read/Write: RO
Reset: 0000.0000

Valid when INPUT_SOURCE is HOST

Bit	Description
0	**HOST_VSYNC_TRIGGER: This triggers VI's internal VSYNC generation** Always write once to this register with any data before writing the Frame's data to Y_FIFO_DATA

VI_COMMAND_0

Offset: 027h
Read/Write: R/W
Reset: 0000.0000

VI Command

This register is used initialize VI module when INPUT_SOURCE is HOST.

This register has a dual use purpose. Host input VSYNC is created by writing to this register.

Bit	Description
27:16	**V_COUNTER_THRESHOLD: Vertical Counter Threshold** This specifies a threshold which, when exceeded, would generate the vertical counter interrupt if the interrupt is enabled. This is used to detect the case when the host is sending too many input data than expected by VI module.
11:8	**Y_FIFO_THRESHOLD: Y-FIFO Threshold** This specifies maximum number of filled locations in Y-FIFO for the Y-FIFO Threshold Status bit.
0	**PROCESS_FIELD: Process Odd/Even field.** 0 = ODD 1 = EVEN

VI_HOST_FIFO_STATUS_0

Offset: 028h
Read/Write: RO
Reset: 0000.0000

Host FIFO status

Note: This is not needed if host input video goes through command buffer interface.

Bit	Description
14:12	**V_FIFO_STATUS: This indicates the number of filled locations in V-FIFO.** If the returned value is 3'h0, the fifo is empty and if the returned value is 3'h7 then the fifo is full.
10:8	**U_FIFO_STATUS: This indicates the number of filled locations in U-FIFO.** If the returned value is 3'h0, the fifo is empty and if the returned value is 3'h7 then the fifo is full.
3:0	**Y_FIFO_STATUS: This indicates the number of filled locations in Y-FIFO.** If the returned value is 4'h0, the fifo is empty and if the returned value is 4'hF then the fifo is full.

VI_INTERRUPT_MASK_0

Offset: 029h
Read/Write: R/W
Reset: 0000.0000

Interrupt Mask

Bit	Description
19	**DMA_STALL_INT_MASK: Write Buffer DMA to VI Stalls VI and causes an error.** 0 = Disabled 1 = Enabled
18	**SECOND_OUTPUT_PEER_STALL_INT_MASK: VI to Peer stall - Second Memory Output.** 0 = Disabled 1 = Enabled
17	**FIRST_OUTPUT_PEER_STALL_INT_MASK: VI to Peer stall - First Memory Output.** 0 = Disabled 1 = Enabled
16	**YUV420PA_ERROR_INT_MASK: YUV420PA Error Interrupt Mask.** 0 = Disabled 1 = Enabled
15	**EPP_ERROR_INT_MASK: VI to EPP Error Interrupt Mask.** 0 = Disabled 1 = Enabled
14	**FRAME_SECOND_OUTPUT_INT_MASK: Buffer Done Second Output Interrupt Mask.** 0 = Disabled 1 = Enabled
13	**BUFFER_SECOND_OUTPUT_INT_MASK: Buffer Done Second Output Interrupt Mask.** 0 = Disabled 1 = Enabled

Bit	Description
12	**FRAME_FIRST_OUTPUT_INT_MASK: Frame Done First Output Interrupt Mask.** 0 = Disabled 1 = Enabled
11	**BUFFER_FIRST_OUTPUT_INT_MASK: Buffer Done First Output Interrupt Mask.** 0 = Disabled 1 = Enabled
10	**Y_THRESHOLD_INT_MASK: Y-FIFO Threshold Interrupt Mask.** 0 = Disabled 1 = Enabled
9	**V_COUNTER_INT_MASK: Vertical Counter Interrupt Mask.** 0 = Disabled 1 = Enabled
8	**VVS_INT_MASK: VVS pin Interrupt Mask.** 0 = Disabled 1 = Enabled
7	**VHS_INT_MASK: VHS pin Interrupt Mask.** 0 = Disabled 1 = Enabled
6	**VGP6_INT_MASK: VGP6 pin Interrupt Mask.** 0 = Disabled 1 = Enabled
5	**VGP5_INT_MASK: VGP5 pin Interrupt Mask.** 0 = Disabled 1 = Enabled
4	**VGP4_INT_MASK: VGP4 pin Interrupt Mask.** 0 = Disabled 1 = Enabled
3	**VD11_INT_MASK: VD11 pin Interrupt Mask.** 0 = Disabled 1 = Enabled
2	**VD10_INT_MASK: VD10 pin Interrupt Mask.** 0 = Disabled 1 = Enabled
1	**VD9_INT_MASK: VD9 pin Interrupt Mask.** 0 = Disabled 1 = Enabled
0	**VD8_INT_MASK: VD8 pin Interrupt Mask.** 0 = Disabled 1 = Enabled

VI_INTERRUPT_TYPE_SELECT_0

Offset: 02ah
Read/Write: R/W
Reset: 0000.0000

Interrupt Type Select

Bit	Description
8	**VVS_INT_TYPE: VVS pin Interrupt Type.** 0 = Edge 1 = Level
7	**VHS_INT_TYPE: VHS pin Interrupt Type.** 0 = Edge 1 = Level
6	**VGP6_INT_TYPE: VGP6 pin Interrupt Type.** 0 = Edge 1 = Level
5	**VGP5_INT_TYPE: VGP5 pin Interrupt Type.** 0 = Edge 1 = Level
4	**VGP4_INT_TYPE: VGP4 pin Interrupt Type.** 0 = Edge 1 = Level
3	**VD11_INT_TYPE: VD11 pin Interrupt Type.** 0 = Edge 1 = Level
2	**VD10_INT_TYPE: VD10 pin Interrupt Type.** 0 = Edge 1 = Level
1	**VD9_INT_TYPE: VD9 pin Interrupt Type.** 0 = Edge 1 = Level
0	**VD8_INT_TYPE: VD8 pin Interrupt Type.** 0 = Edge 1 = Level

VI_INTERRUPT_POLARITY_SELECT_0

Offset: 02bh
Read/Write: R/W
Reset: 0000.0000

Interrupt Polarity Select

Bit	Description
8	**VVS_INT_POLARITY: VVS pin Interrupt Type.** 0 = Low 1 = High
7	**VHS_INT_POLARITY: VHS pin Interrupt Type.** 0 = Low 1 = High
6	**VGP6_INT_POLARITY: VGP6 pin Interrupt Type.** 0 = Low 1 = High

Bit	Description
5	**VGP5_INT_POLARITY: VGP5 pin Interrupt Type.** 0 = Low 1 = High
4	**VGP4_INT_POLARITY: VGP4 pin Interrupt Type.** 0 = Low 1 = High
3	**VD11_INT_POLARITY: VD11 pin Interrupt Type.** 0 = Low 1 = High
2	**VD10_INT_POLARITY: VD10 pin Interrupt Type.** 0 = Low 1 = High
1	**VD9_INT_POLARITY: VD9 pin Interrupt Type.** 0 = Low 1 = High
0	**VD8_INT_POLARITY: VD8 pin Interrupt Type.** 0 = Low 1 = High

VI_INTERRUPT_STATUS_0

Offset: 02ch
Read/Write: RO
Reset: 0000.0000

Interrupt Enable

This register returns interrupt status when read. Except for bits 15-14, when this register is written, the interrupt status corresponding to the bits written with 1 will be reset. Interrupt status corresponding to the bits written with 0 will be left unchanged.

Note that interrupt status bits can be set even when their corresponding interrupt enable bits, in VI10R, are cleared. When these bits are set and their corresponding interrupt enable bits are set, an interrupt is generated. The interrupt can be cleared, or left unchanged, by writing 1, or 0, respectively to the corresponding bits in this register. Clearing the interrupt status bits does not affect the interrupt enable bits.

Bit	Description
20	**FIELD_STATUS: Top or Bottom Field Status.** 0 = Bottom 1 = Top
19	**DMA_STALL_INT_STATUS** 0 = NOINTR 1 = INTR
18	**SECOND_OUTPUT_PEER_STALL_INT_STATUS** 0 = NOINTR 1 = INTR
17	**FIRST_OUTPUT_PEER_STALL_INT_STATUS** 0 = NOINTR 1 = INTR
16	**YUV420PA_ERROR_INT_STATUS: YUV420PA Error Interrupt Enable.** 0 = NOINTR 1 = INTR

Bit	Description
15	**EPP_ERROR_INT_STATUS: VI to EPP Error Interrupt Enable.** 0 = NOINTR 1 = INTR
14	**FRAME_SECOND_OUTPUT_INT_STATUS: Frame Done Second Output Interrupt Status.** 0 = NOINTR 1 = INTR
13	**BUFFER_SECOND_OUTPUT_INT_STATUS: Buffer Done Second Output Interrupt Status.** 0 = NOINTR 1 = INTR
12	**FRAME_FIRST_OUTPUT_INT_STATUS: Frame Done First Output Interrupt Status.** 0 = NOINTR 1 = INTR
11	**BUFFER_FIRST_OUTPUT_INT_STATUS: Buffer Done First Output Interrupt Status.** 0 = NOINTR 1 = INTR
10	**Y_THRESHOLD_INT_STATUS: Y-FIFO Threshold Interrupt Enable.** 0 = NOINTR 1 = INTR
9	**V_COUNTER_INT_STATUS: Vertical Counter Interrupt Status.** 0 = NOINTR 1 = INTR
8	**VVS_INT_STATUS: VVS pin Interrupt Status.** 0 = NOINTR 1 = INTR
7	**VHS_INT_STATUS: VHS pin Interrupt Status.** 0 = NOINTR 1 = INTR
6	**VGP6_INT_STATUS: VGP6 pin Interrupt Status.** 0 = NOINTR 1 = INTR
5	**VGP5_INT_STATUS: VGP5 pin Interrupt Status.** 0 = NOINTR 1 = INTR
4	**VGP4_INT_STATUS: VGP4 pin Interrupt Status.** 0 = NOINTR 1 = INTR
3	**VD11_INT_STATUS: VD11 pin Interrupt Status.** 0 = NOINTR 1 = INTR
2	**VD10_INT_STATUS: VD10 pin Interrupt Status.** 0 = NOINTR 1 = INTR
1	**VD9_INT_STATUS: VD9 pin Interrupt Status.** 0 = NOINTR 1 = INTR
0	**VD8_INT_STATUS: VD8 pin Interrupt Status.** 0 = NOINTR 1 = INTR

VI_VIP_INPUT_STATUS_0

Offset: 02dh
Read/Write: RO
Reset: 0000.0000

Video Input Port status

Bit	Description
31:16	**FRAME_COUNT: The number of frames received (vsyncs)** Any write to this register, clears.
15:0	**LINE_COUNT: The number of lines received (hsyncs)**

VI_VIDEO_BUFFER_STATUS_0

Offset: 02eh
Read/Write: RO
Reset: 0000.0000

Interrupt Enable

Bit	Description
19:16	**RAW_STREAM_WRITE_COUNT: Write count of the Raw Stream Write FIFO** This is the fifo used to synchronize the data coming from pads into the vi clock domain.
15:8	**SECOND_VIDEO_BUFFER_STATUS: Buffer status** This specifies the buffer number of the the last video data field written to memory
7:0	**FIRST_VIDEO_BUFFER_STATUS: Buffer status** This specifies the buffer number of the the last video data field written to memory

VI_SYNC_OUTPUT_0

Offset: 02fh
Read/Write: R/W
Reset: 0000.0000

VI H and V sync Output control

This register controls VHS and VVS output when H/V syncs are generated internally in the VI module (VIDEO_SOURCE is VIP and SYNC_FORMAT is INTHVS). The generated VHS and VVS signal can be sent to external video source device and used to synchronize the video data transfer from the video source to the VI module. VHS and VVS pin should be configured in output mode to output the internally generated H/V syncs. Also in this case, the internally generate H/V syncs can be used by the VI module as horizontal and vertical reference signals for the incoming video data.

Bit	Description
30:19	**VVS_OUTPUT_PERIOD: This specifies VVS output pulse period in term of number of VHS cycles.** Programmed value is actual value - 1: The valid value ranges from 2 to 4096.
18:16	**VVS_OUTPUT_WIDTH: This specifies VVS output pulse width in term of number of VHS cycles.** Programmed value is actual value - 1: The valid value ranges from 1 to 8.
15:3	**VHS_OUTPUT_PERIOD: This specifies VHS output pulse period in term of number of VI clock cycles.** Programmed value is actual value - 1: The valid value ranges from 32 to 8192.
2:0	**VHS_OUTPUT_WIDTH: This specifies VHS output pulse width in term of number of VI clock cycles.** Programmed value is actual value - 1: The valid value ranges from 1 to 8.

VI_VVS_OUTPUT_DELAY_0

Offset: 030h
Read/Write: R/W
Reset: 0000.0000

VI V sync Output Delay

Bit	Description
3:0	**VVS_OUTPUT_DELAY: This specifies the number of VI clock cycles from leading edge of VHS to leading edge of VVS.** Programmed value is actual value + 2: The valid value ranges from -2 to 13.

VI_PWM_CONTROL_0

Offset: 031h
Read/Write: R/W
Reset: 0000.0000

VI Pulse Width Modulation Control

VI Pulse Width Modulation signal generation PWM signal generation logic can generate up to 128 pulses per line internally and the PWM pulse select registers determines which of the 128 pulses will be output. Any of the 128 internally generated pulse can be independently selected as output if they occur within one line time.

PWM signal can be output on the VGP6 pin if VGP6 output is enabled and the output select is set to PWM. The PWM will be triggered by the first vsync after the PWM_ENABLE bit has been set.

Bit	Description
31:24	**PWM_COUNTER: PWM Counter** 8-bit value used when PWM_MODE is set to COUNTER to determine how many times the PWM will cycle through the 128 cycles before stopping.
21:20	**PWM_MODE: PWM Mode.** 0 = Continuous 1 = Single 2 = Counter
15:12	**PWM_LOW_PULSE: PWM Low Pulse (1 to 16)** 19:16 reserved
11:8	**PWM_HIGH_PULSE: PWM High Pulse (1 to 16)**
4	**PWM_DIRECTION: PWM Direction.** 0 = Incr 1 = Decr
0	**PWM_ENABLE: PWM Enable.** 0 = Disabled 1 = Enabled

VI_PWM_SELECT_PULSE_A_0

Offset: 032h
Read/Write: R/W
Reset: 0000.0000

PWM Pulse Select A

The next 4 registers select which of the internal 128 pulses to be output. Each bit in the four registers correspond to one internal pulse.

Bit	Description
31:0	**PWM_SELECT_A: PWM Select bits 31 to 0**

VI_PWM_SELECT_PULSE_B_0

Offset: 033h
Read/Write: R/W
Reset: 0000.0000

PWM Pulse Select B

Bit	Description
31:0	**PWM_SELECT_B: PWM Select bits 63 to 32**

VI_PWM_SELECT_PULSE_C_0

Offset: 034h
Read/Write: R/W
Reset: 0000.0000

PWM Pulse Select C

Bit	Description
31:0	**PWM_SELECT_C: PWM Select bits 95 to 64**

VI_PWM_SELECT_PULSE_D_0

Offset: 035h
Read/Write: R/W
Reset: 0000.0000

PWM Pulse Select D

Bit	Description
31:0	**PWM_SELECT_D: PWM Select bits 127 to 96**

VI_VI_DATA_INPUT_CONTROL_0

Offset: 036h
Read/Write: R/W
Reset: 0000.0fff

VI Input Mask

Bit	Description
11:0	**VI_DATA_INPUT_MASK: Mask the VD[11:0] pin inputs to the VI core and ISP** The mask is not applied to the Host GPIO read value

VI_PIN_INPUT_ENABLE_0

The VI pins may be used as either regular input (i.e. VSYNC and HSYNC) or GPIOs.

Writing bits in Register VI_PIN_INPUT_ENABLE_0 to 1 allows the VI to sample whatever is on the pin. When the input to a pin is enabled in this register, the value on that pin can be read from Input Data Read Register VI_PIN_INPUT_DATA_0.

Offset: 037h
Read/Write: R/W
Reset: 0000.0000

VI pins Input Enable

Bit	Description
21	**VGP6_INPUT_ENABLE: VGP6 pin Input Enable.** 0 = Disabled 1 = Enabled
20	**VGP5_INPUT_ENABLE: VGP5 pin Input Enable.** 0 = Disabled 1 = Enabled
19	**VGP4_INPUT_ENABLE: VGP4 pin Input Enable.** 0 = Disabled 1 = Enabled
18	**VGP3_INPUT_ENABLE: VGP3 pin Input Enable.** 0 = Disabled 1 = Enabled
17	**VGP2_INPUT_ENABLE: VGP2 pin Input Enable.** 0 = Disabled 1 = Enabled
16	**VGP1_INPUT_ENABLE: VGP1 pin Input Enable.** 0 = Disabled 1 = Enabled
15	**VGP0_INPUT_ENABLE: VGP0 pin Input Enable.** 0 = Disabled 1 = Enabled
14	**VVS_INPUT_ENABLE: VVS pin Input Enable.** 0 = Disabled 1 = Enabled

Bit	Description
13	**VHS_INPUT_ENABLE: VHS pin Input Enable.** 0 = Disabled 1 = Enabled
11	**VD11_INPUT_ENABLE: VD11 pin Input Enable.** 0 = Disabled 1 = Enabled
10	**VD10_INPUT_ENABLE: VD10 pin Input Enable.** 0 = Disabled 1 = Enabled
9	**VD9_INPUT_ENABLE: VD9 pin Input Enable.** 0 = Disabled 1 = Enabled
8	**VD8_INPUT_ENABLE: VD8 pin Input Enable.** 0 = Disabled 1 = Enabled
7	**VD7_INPUT_ENABLE: VD7 pin Input Enable.** 0 = Disabled 1 = Enabled
6	**VD6_INPUT_ENABLE: VD6 pin Input Enable.** 0 = Disabled 1 = Enabled
5	**VD5_INPUT_ENABLE: VD5 pin Input Enable.** 0 = Disabled 1 = Enabled
4	**VD4_INPUT_ENABLE: VD4 pin Input Enable.** 0 = Disabled 1 = Enabled
3	**VD3_INPUT_ENABLE: VD3 pin Input Enable.** 0 = Disabled 1 = Enabled
2	**VD2_INPUT_ENABLE: VD2 pin Input Enable.** 0 = Disabled 1 = Enabled
1	**VD1_INPUT_ENABLE: VD1 pin Input Enable.** 0 = Disabled 1 = Enabled
0	**VD0_INPUT_ENABLE: VD0 pin Input Enable.** 0 = Disabled 1 = Enabled

VI_PIN_OUTPUT_ENABLE_0

Offset: 038h
Read/Write: R/W
Reset: 0000.0000

VI pins Output Enable

Setting one or more bits in this register (to 1) causes the VI to drive a value onto the corresponding pin or pins.

Bit	Description
21	**VGP6_OUTPUT_ENABLE: VGP6 pin Output Enable.** 0 = Disabled 1 = Enabled
20	**VGP5_OUTPUT_ENABLE: VGP5 pin Output Enable.** 0 = Disabled 1 = Enabled
19	**VGP4_OUTPUT_ENABLE: VGP4 pin Output Enable.** 0 = Disabled 1 = Enabled
18	**VGP3_OUTPUT_ENABLE: VGP3 pin Output Enable.** 0 = Disabled 1 = Enabled
17	**VGP2_OUTPUT_ENABLE: VGP2 pin Output Enable.** 0 = Disabled 1 = Enabled
16	**VGP1_OUTPUT_ENABLE: VGP1 pin Output Enable.** 0 = Disabled 1 = Enabled
15	**VGP0_OUTPUT_ENABLE: VGP0 pin Output Enable.** 0 = Disabled 1 = Enabled
14	**VVS_OUTPUT_ENABLE: VVS pin Output Enable.** 0 = Disabled 1 = Enabled
13	**VHS_OUTPUT_ENABLE: VHS pin Output Enable.** 0 = Disabled 1 = Enabled
12	**VSCK_OUTPUT_ENABLE: VSCK pin Output Enable.** 0 = Disabled 1 = Enabled
11	**VD11_OUTPUT_ENABLE: VD11 pin Output Enable.** 0 = Disabled 1 = Enabled
10	**VD10_OUTPUT_ENABLE: VD10 pin Output Enable.** 0 = Disabled 1 = Enabled
9	**VD9_OUTPUT_ENABLE: VD9 pin Output Enable.** 0 = Disabled 1 = Enabled
8	**VD8_OUTPUT_ENABLE: VD8 pin Output Enable.** 0 = Disabled 1 = Enabled
7	**VD7_OUTPUT_ENABLE: VD7 pin Output Enable.** 0 = Disabled 1 = Enabled

Bit	Description
6	**VD6_OUTPUT_ENABLE: VD6 pin Output Enable.** 0 = Disabled 1 = Enabled
5	**VD5_OUTPUT_ENABLE: VD5 pin Output Enable.** 0 = Disabled 1 = Enabled
4	**VD4_OUTPUT_ENABLE: VD4 pin Output Enable.** 0 = Disabled 1 = Enabled
3	**VD3_OUTPUT_ENABLE: VD3 pin Output Enable.** 0 = Disabled 1 = Enabled
2	**VD2_OUTPUT_ENABLE: VD2 pin Output Enable.** 0 = Disabled 1 = Enabled
1	**VD1_OUTPUT_ENABLE: VD1 pin Output Enable.** 0 = Disabled 1 = Enabled
0	**VD0_OUTPUT_ENABLE: VD0 pin Output Enable.** 0 = Disabled 1 = Enabled

VI_PIN_INVERSION_0

Offset: 039h
Read/Write: R/W
Reset: 0000.0000

VI pins input/output Inversion: 0 reserved

Bit	Description
18	**VVS_OUT_INVERSION: VVS pin Output Inversion.** 0 = Disabled 1 = Enabled
17	**VHS_OUT_INVERSION: VHS pin Output Inversion.** 0 = Disabled 1 = Enabled
16	**VSCK_OUT_INVERSION: VSCK pin Output Inversion.** 0 = Disabled 1 = Enabled
2	**VVS_IN_INVERSION: VVS pin Input Inversion.** 0 = Disabled 1 = Enabled
1	**VHS_IN_INVERSION: VHS pin Input Inversion.** 0 = Disabled 1 = Enabled

VI_PIN_INPUT_DATA_0

Offset: 03ah
Read/Write: RO
Reset: 0000.0000

VI pins Input Data

This register contains input data when the video camera interface pins are used as general-purpose input pins. The pin data read from this register is not affected by the pin input inversion bits.

When the corresponding register bits in VI_PIN_INPUT_ENABLE_0 are "1", the value of the pin can be read from this register. (E.g. If VI_PIN_INPUT_ENABLE_0[11] = 1, the value on pin VD11 is located in bit 11 of this register.)

Bit	Description
21	**VGP6_INPUT_DATA: VGP6 pin Input Data** (effective if VGP6_INPUT_ENABLE is ENABLED) 0= VGP6 input low 1= VGP6 input high
20	**VGP5_INPUT_DATA: VGP5 pin Input Data** (effective if VGP5_INPUT_ENABLE is ENABLED) 0= VGP5 input low 1= VGP5 input high
19	**VGP4_INPUT_DATA: VGP4 pin Input Data** (effective if VGP4_INPUT_ENABLE is ENABLED) 0= VGP4 input low 1= VGP4 input high
18	**VGP3_INPUT_DATA: VGP3 pin Input Data** (effective if VGP3_INPUT_ENABLE is ENABLED) 0= VGP3 input low 1= VGP3 input high
17	**VGP2_INPUT_DATA: VGP2 pin Input Data** (effective if VGP2_INPUT_ENABLE is ENABLED) 0= VGP2 input low 1= VGP2 input high
16	**VGP1_INPUT_DATA: VGP1 pin Input Data** (effective if VGP1_INPUT_ENABLE is ENABLED) 0= VGP1 input low 1= VGP1 input high
15	**VGP0_INPUT_DATA: VGP0 pin Input Data** (effective if VGP0_INPUT_ENABLE is ENABLED) 0= VGP0 input low 1= VGP0 input high
14	**VVS_INPUT_DATA: VVS pin Input Data** (effective if VVS_INPUT_ENABLE is ENABLED) 0= VVS input low 1= VVS input high
13	**VHS_INPUT_DATA: VHS pin Input Data** (effective if VHS_INPUT_ENABLE is ENABLED) 0= VHS input low 1= VHS input high
12	**VSCK_INPUT_DATA: VSCK pin Input Data** (effective if VSCK_INPUT_ENABLE is ENABLED) 0= VSCK input low 1= VSCK input high

Bit	Description
11	**VD11_INPUT_DATA: VD11 pin Input Data** (effective if VD11_INPUT_ENABLE is ENABLED) 0= VD11 input low 1= VD11 input high
10	**VD10_INPUT_DATA: VD10 pin Input Data** (effective if VD10_INPUT_ENABLE is ENABLED) 0= VD10 input low 1= VD10 input high
9	**VD9_INPUT_DATA: VD9 pin Input Data** (effective if VD9_INPUT_ENABLE is ENABLED) 0= VD9 input low 1= VD9 input high
8	**VD8_INPUT_DATA: VD8 pin Input Data** (effective if VD8_INPUT_ENABLE is ENABLED) 0= VD8 input low 1= VD8 input high
7	**VD7_INPUT_DATA: VD7 pin Input Data** (effective if VD7_INPUT_ENABLE is ENABLED) 0= VD7 input low 1= VD7 input high
6	**VD6_INPUT_DATA: VD6 pin Input Data** (effective if VD6_INPUT_ENABLE is ENABLED) 0= VD6 input low 1= VD6 input high
5	**VD5_INPUT_DATA: VD5 pin Input Data** (effective if VD5_INPUT_ENABLE is ENABLED) 0= VD5 input low 1= VD5 input high
4	**VD4_INPUT_DATA: VD4 pin Input Data** (effective if VD4_INPUT_ENABLE is ENABLED) 0= VD4 input low 1= VD4 input high
3	**VD3_INPUT_DATA: VD3 pin Input Data** (effective if VD3_INPUT_ENABLE is ENABLED) 0= VD3 input low 1= VD3 input high
2	**VD2_INPUT_DATA: VD2 pin Input Data** (effective if VD2_INPUT_ENABLE is ENABLED) 0= VD2 input low 1= VD2 input high
1	**VD1_INPUT_DATA: VD1 pin Input Data** (effective if VD1_INPUT_ENABLE is ENABLED) 0= VD1 input low 1= VD1 input high
0	**VD0_INPUT_DATA: VD0 pin Input Data**

VI_PIN_OUTPUT_DATA_0

Offset: 03bh
Read/Write: R/W
Reset: 0000.0000

VI pins Output Data

This register contains output data when the video camera interface pins are used as general-purpose output pins.

This register can be used to write the output driven on each pin. The pin must be output-enabled, and the output control select bits must be appropriately enabled. (See Register VI_PIN_OUTPUT_SELECT_0.)

Note: The output signal at the pin IS affected by the corresponding pin output inversion bit.

Bit	Description
21	**VGP6_OUTPUT_DATA: VGP6 pin Output Data** (effective if VGP6_OUTPUT_ENABLE is ENABLED and VGP6_OUTPUT_SELECT is DATA)
20	**VGP5_OUTPUT_DATA: VGP5 pin Output Data** (effective if VGP5_OUTPUT_ENABLE is ENABLED and VGP5_OUTPUT_SELECT is DATA)
19	**VGP4_OUTPUT_DATA: VGP4 pin Output Data** (effective if VGP4_OUTPUT_ENABLE is ENABLED and VGP4_OUTPUT_SELECT is DATA)
18	**VGP3_OUTPUT_DATA: VGP3 pin Output Data** (effective if VGP3_OUTPUT_ENABLE is ENABLED and VGP3_OUTPUT_SELECT is DATA)
17	**VGP2_OUTPUT_DATA: VGP2 pin Output Data** (effective if VGP2_OUTPUT_ENABLE is ENABLED and VGP2_OUTPUT_SELECT is DATA)
16	**VGP1_OUTPUT_DATA: VGP1 pin Output Data** (effective if VGP1_OUTPUT_ENABLE is ENABLED and VGP1_OUTPUT_SELECT is DATA)
15	**VGP0_OUTPUT_DATA: VGP0 pin Output Data** (effective if VGP0_OUTPUT_ENABLE is ENABLED and VGP0_OUTPUT_SELECT is DATA)
14	**VVS_OUTPUT_DATA: VVS pin Output Data** (effective if VVS_OUTPUT_ENABLE is ENABLED and VVS_OUTPUT_SELECT is DATA)
13	**VHS_OUTPUT_DATA: VHS pin Output Data** (effective if VHS_OUTPUT_ENABLE is ENABLED and VHS_OUTPUT_SELECT is DATA)
12	**VSCK_OUTPUT_DATA: VSCK pin Output Data** (effective if VSCK_OUTPUT_ENABLE is ENABLED and VSCK_OUTPUT_SELECT is DATA)
11	**VD11_OUTPUT_DATA: VD11 pin Output Data** (effective if VD11_OUTPUT_ENABLE is ENABLED and VD11_OUTPUT_SELECT is DATA)
10	**VD10_OUTPUT_DATA: VD10 pin Output Data** (effective if VD10_OUTPUT_ENABLE is ENABLED and VD10_OUTPUT_SELECT is DATA)
9	**VD9_OUTPUT_DATA: VD9 pin Output Data** (effective if VD9_OUTPUT_ENABLE is ENABLED and VD9_OUTPUT_SELECT is DATA)
8	**VD8_OUTPUT_DATA: VD8 pin Output Data** (effective if VD8_OUTPUT_ENABLE is ENABLED and VD8_OUTPUT_SELECT is DATA)
7	**VD7_OUTPUT_DATA: VD7 pin Output Data** (effective if VD7_OUTPUT_ENABLE is ENABLED and VD7_OUTPUT_SELECT is DATA)
6	**VD6_OUTPUT_DATA: VD6 pin Output Data** (effective if VD6_OUTPUT_ENABLE is ENABLED and VD6_OUTPUT_SELECT is DATA)
5	**VD5_OUTPUT_DATA: VD5 pin Output Data** (effective if VD5_OUTPUT_ENABLE is ENABLED and VD5_OUTPUT_SELECT is DATA)

Bit	Description
4	**VD4_OUTPUT_DATA: VD4 pin Output Data** (effective if VD4_OUTPUT_ENABLE is ENABLED and VD4_OUTPUT_SELECT is DATA)
3	**VD3_OUTPUT_DATA: VD3 pin Output Data** (effective if VD3_OUTPUT_ENABLE is ENABLED and VD3_OUTPUT_SELECT is DATA)
2	**VD2_OUTPUT_DATA: VD2 pin Output Data** (effective if VD2_OUTPUT_ENABLE is ENABLED and VD2_OUTPUT_SELECT is DATA)
1	**VD1_OUTPUT_DATA: VD1 pin Output Data** (effective if VD1_OUTPUT_ENABLE is ENABLED and VD1_OUTPUT_SELECT is DATA)
0	**VD0_OUTPUT_DATA: VD0 pin Output Data** (effective if VD0_OUTPUT_ENABLE is ENABLED and VD0_OUTPUT_SELECT is DATA)

VI_PIN_OUTPUT_SELECT_0

Offset: 03ch
Read/Write: R/W
Reset: 0000.0000

VI pins Output Select

Some GoForce 5500 pins have dual functions, so Register VI_PIN_OUTPUT_SELECT_0 acts as a mux to ensure the selected one of two possible signal choice gets driven on such pins. In this register, if a pin has one purpuse only, it is shown in the bit description.

For example, VGP6 has two functions. So setting bit[21] to 1 in this register drives the PWM on VGP6 instead of driving data on that pin as specified in VI_PIN_OUPUT_DATA_0[21].

VCLK, VHSYNC, VVSYNC
This register selects between the register programmed GPIO outputs (when set to 0) and the internally generated VI signals (when set to 1) for these pins.

VGP1through VGP2
This register selects between the I2C outputs (when set to 0) and the VI register-programmed GPIO outputs (when set to 1) for these pins

For VD0 through VD11, and VGP3-VGP5
These pins do not have a secondary function. So the “1” state is a reserved state for these bits.

Bit	Description
21	**PIN_OUTPUT_SELECT_vgp6: Pin Output Select VGP6.** 0 = DATA 1 = PWM
20	**PIN_OUTPUT_SELECT_vgp5: Pin Output Select VGP5** 0 = VGP5 GPIO output data 1 = Reserved
19	**PIN_OUTPUT_SELECT_vgp4: Pin Output Select VGP4** 0 = VGP4 GPIO output data 1 = Reserved
18	**PIN_OUTPUT_SELECT_vgp3: Pin Output Select VGP3** 0 = VGP3 GPIO output data 1 = Reserved
17	**PIN_OUTPUT_SELECT_vgp2: Pin Output Select VGP2** 0 = I2C SDA pin 1 = VGP2 output register

Bit	Description
16	**PIN_OUTPUT_SELECT_vgp1: Pin Output Select VGP1** 0 = I2C SCK pin 1 = VGP1 output register
15	**PIN_OUTPUT_SELECT_vgp0: Pin Output Select VGP0** 0 = VGP0 GPIO output data 1 = REFCLK
14	**PIN_OUTPUT_SELECT_vvs: Pin Output Select VVSYNC** 0 = VSYNC GPIO output data 1 = Internally generated VVSYNC
13	**PIN_OUTPUT_SELECT_vhs: Pin Output Select VHSYNC** 0 = HSYNC GPIO output data 1 = Internally generated VHSYNC
9	**PIN_OUTPUT_SELECT_vd9: Pin Output Select VD9** 0 = VD9 GPIO output data 1 = Reserved
8	**PIN_OUTPUT_SELECT_vd8: Pin Output Select VD8** 0 = VD8 GPIO output data 1 = Reserved
7	**PIN_OUTPUT_SELECT_vd7: Pin Output Select VD7** 0 = VD7 GPIO output data 1 = Reserved
6	**PIN_OUTPUT_SELECT_vd6: Pin Output Select VD6** 0 = VD6 GPIO output data 1 = Reserved
5	**PIN_OUTPUT_SELECT_vd5: Pin Output Select VD5** 0 = VD5 GPIO output data register 1 = Reserved
4	**PIN_OUTPUT_SELECT_vd4: Pin Output Select VD4** 0 = VD4 GPIO output data register 1 = Reserved
3	**PIN_OUTPUT_SELECT_vd3: Pin Output Select VD3** 0 = VD3 GPIO output data register 1 = Reserved
2	**PIN_OUTPUT_SELECT_vd2: Pin Output Select VD2** 0 = VD2 GPIO output data 1 = Reserved
1	**PIN_OUTPUT_SELECT_vd1: Pin Output Select VD1** 0 = VD1 GPIO output data 1 = Reserved
0	**PIN_OUTPUT_SELECT_vd0: Pin Output Select VD0** 0 = VD0 GPIO output data 1 = Reserved

VI_RAISE_VIP_BUFFER_FIRST_OUTPUT_0

Offset: 03dh
Read/Write: R/W
Reset: 0000.0000

Raise vector at buffer end

Raise vectors are received from host. If host is the input source, host will send a raise vector at the end of a line, and VI return it when that has been written to memory. A raise

written when decimation or averaging is selected in vi, is not supported. If Video Input Port is the input source, host should program raise vectors to either raise at buffer end or at frame end.

Since there are 2 memory outputs for vi, there are two separate raise vectors for buffer/frame.

Bit	Description
4:0	**RAISE_BUFFER_1_VECTOR**

VI_RAISE_VIP_FRAME_FIRST_OUTPUT_0

Offset: 03eh
Read/Write: R/W
Reset: 0000.0000

Raise vector at frame end

Bit	Description
4:0	**RAISE_FRAME_1_VECTOR**

VI_RAISE_VIP_BUFFER_SECOND_OUTPUT_0

Offset: 03fh
Read/Write: R/W
Reset: 0000.0000

Raise vector at buffer end

Bit	Description
4:0	**RAISE_BUFFER_2_VECTOR**

VI_RAISE_VIP_FRAME_SECOND_OUTPUT_0

Offset: 040h
Read/Write: R/W
Reset: 0000.0000

Raise vector at frame end

Bit	Description
4:0	**RAISE_FRAME_2_VECTOR**

VI_RAISE_HOST_FIRST_OUTPUT_0

Offset: 041h
Read/Write: R/W
Reset: 0000.0000

Raise vector when from host

Bit	Description
4:0	**RAISE_HOST_1_VECTOR**

VI_RAISE_HOST_SECOND_OUTPUT_0

Offset: 042h
Read/Write: R/W
Reset: 0000.0000

Raise vector when from host

Bit	Description
4:0	**RAISE_HOST_2_VECTOR**

VI_RAISE_EPP_0

Offset: 043h
Read/Write: R/W
Reset: 0000.0000

Raise vector at line end
EPP receives the raise request via the Simple Stream Video Data bus see arepp.spec for details This raise needs to be written during the horizontal blanking period. (After end of line.)

This is only valid if the input source is host.

Bit	Description
4:0	**RAISE_EPP_VECTOR**

VI_CLASS_REFCOUNT_0

Offset: 044h
Read/Write: RO
Reset: 0000.0000

Null refcount

Bit	Description
15:0	**REFCOUNT**

VI_VI_ENABLE_0

Offset: 045h
Read/Write: R/W
Reset: 0000.0000

VI Enables

Bit	Description
4	**STOP_CAPTURE: Disables camera capturing VI_ENABLE after the .** 0 = Disabled 1 = Enabled
3	**SECOND_OUTPUT_TO_MEMORY: Second Output to Memory.** 0 = Enabled 1 = Disabled
2	**FIRST_OUTPUT_TO_MEMORY: First Output to Memory.** 0 = Enabled 1 = Disabled
1	**TEST_MODE_ENABLE: Test Mode Enable.** 0 = Disabled 1 = Enabled
0	**VI_ENABLE: VI camera input module Enable.** 0 = NOEDIT 1 = ENABLED

VI_Y_FIFO_WRITE_0

Offset: 046h
Read/Write: RO
Reset: 0000.0000

YUV 4:2:0 Planar Y-FIFO, YUV 4:2:2 non-Planar YUV FIFO

**** Host YUV FIFO offsets. This register space is used for Host Video Data writes.
**** YUV 4:2:0 planar for re-encoding as well as YUV 4:2:2 data.

Bit	Description
31:0	**Y_FIFO_DATA**

VI_U_FIFO_WRITE_0

Offset: 047h
Read/Write: RO
Reset: 0000.0000

YUV 4:2:0 Planar U-FIFO

Bit	Description
31:0	**U_FIFO_DATA**

VI_V_FIFO_WRITE_0

Offset: 048h
Read/Write: RO
Reset: 0000.0000

YUV 4:2:0 Planar V-FIFO

Bit	Description
31:0	**V_FIFO_DATA**

VI_VI_MCCIF_FIFOCTRL_0

Memory Client Interface Async Fifo Optimization Register

Memory Client Interface Fifo Control Register: The registers below allow to optimize the synchronization timing in the memory client asynchronous fifos. When they can be used depend on the client and memory controller clock ratio. Additionally, the RDMC_RDFAST/RDCL_RDFAST fields can increase power consumption if the asynchronous fifo is implemented as a real RAM. There is no power impact on latch-based fifos. Flipflop-based fifos do not use these fields. See recommended settings below.

Important: The register fields can only be changed when the memory client async fifos are empty.

The register field ending with WRCL_MCLE2X (if any) can be set to improve async fifo synchronization on the write side by one client clock cycle if the memory controller clock frequency is less or equal to twice the client clock frequency:

mcclk_freq <= 2 * clientclk_freq

The register field ending with WRMC_CLLE2X (if any) can be set to improve async fifo synchronization on the write side by one memory controller clock cycle if the client clock frequency is less or equal to twice the memory controller clock frequency:

clientclk_freq <= 2 * mcclk_freq

The register field ending with RDMC_RDFAST (if any) can be set to improve async fifo synchronization on the read side by one memory controller clock cycle.

Warning: RDMC_RDFAST can be used along with WRCL_MCLE2X only when:

mcclk_freq <= clientclk_freq

The register field ending with RDCL_RDFAST (if any) can be set to improve async fifo synchronization on the read side by one client clock cycle.

Warning: RDCL_RDFAST can be used along with WRMC_CLLE2X only when:

clientclk_freq <= mcclk_freq

RECOMMENDED SETTINGS

Client writing to fifo, memory controller reading from fifo

- mcclk_freq <= clientclk_freq

You can enable both RDMC_RDFAST and WRCL_CLLE2X. If one of the fifos is a real RAM and power is a concern, you should avoid enabling RDMC_RDFAST.

- clientclk_freq < mcclk_freq <= 2 * clientclk_freq

You can enable RDMC_RDFAST or WRCL_MCLE2X, but because the client clock is slower, you should enable only WRCL_MCLE2X.

- 2 * clientclk_freq < mcclk_freq

You can only enable RDMC_RDFAST. If one of the fifos is a real RAM and power is a concern, you should avoid enabling RDMC_RDFAST.

Memory controller writing to fifo, client reading from fifo

- clientclk_freq <= mcclk_freq

You can enable both RDCL_RDFAST and WRMC_CLLE2X. If one of the fifos is a real RAM and power is a concern, you should avoid enabling RDCL_RDFAST.

- mcclk_freq < clientclk_freq <= 2 * mcclk_freq

You can enable RDCL_RDFAST or WRMC_CLLE2X, but because the memory controller clock is slower, you should enable only WRMC_CLLE2X.

- 2 * mcclk_freq < clientclk_freq

You can only enable RDCL_RDFAST. If one of the fifos is a real RAM and power is a concern, avoid enabling RDCL_RDFAST.

Bit	Description
3	**VI_MCCIF_RDCL_RDFAST** 0 = Disable 1 = Enable
2	**VI_MCCIF_WRMC_CLLE2X** 0 = Disable 1 = Enable
1	**VI_MCCIF_RDMC_RDFAST** 0 = Disable 1 = Enable
0	**VI_MCCIF_WRCL_MCLE2X** 0 = Disable 1 = Enable

7.8 Display Registers

DC_CMD_CTXSW_0

Offset: 000h
Read/Write: R/W
Reset: F000.F000

Context switch registers for class and channel Context switch register. Should be common to all modules. Includes the current channel/class (which is writable by SW) and the next channel/class (which the hardware sets when it receives a context switch).

Context switch works like this:
Any context switch request triggers an interrupt to the host and causes the new channel/class to be stored in NEXT_CHANNEL/NEXT_CLASS (see vmod/chexample). SW sees that there is a context switch interrupt and does the necessary operations to make the module ready to receive traffic from the new context. It clears the context switch interrupt and writes CURR_CHANNEL/CLASS to the same value as NEXT_CHANNEL/CLASS, which causes a context switch acknowledge packet to be sent to the host. This completes the context switch and allows the host to continue sending data to the module.

Bit	Description
31:28	**NEXT_CHANNEL: Next requested channel**
25:16	**NEXT_CLASS: Next requested class**
15:12	**CURR_CHANNEL: Current working channel, reset to 'invalid'**
9:0	**CURR_CLASS: Current working class**

DC_CMD_DISPLAY_COMMAND_OPTION0_0

Offset: 001h
Read/Write: R/W
Reset: 0000.0000

Display Controller Option 0

This register is not effective until DISPLAY_COMMAND is written.

Class: Display Command

This command allow switching of active display set (primary, secondary, etc), continuous and non-continuous sending of display frame, and also switching of window buffer assuming that the corresponding window buffer switching is controlled by host.

Note that if the corresponding window buffer switching is not controlled by host then the window buffer switching command will not be effective. Active display set can be changed only on frame boundary. Buffer switching, however, can be changed either at the next line boundary or at the next frame boundary (effective after some delay).

A raise can be enabled so that a raise vector can be generated at the end of the command. In case there is a change in active display, the raise vector will be returned at the next frame boundary. In case there is no change in active display, the raise vector will be returned at buffer switching time (either at the next line boundary or at the next frame boundary).

Bit	Description
18	**WINDOW_C_NC_DISPLAY: Window C Non-Continuous Display.** 0 Disable 1 Enable This is effective only in Non-Continuous Display mode when window C buffer switching is not controlled by host. If this bit is enabled, a frame from each active display is sent whenever Window C buffer is switched.
17	**WINDOW_B_NC_DISPLAY: Window B Non-Continuos Display.** 0 Disable 1 Enable This is effective only in Non-Continuous Display mode when window C buffer switching is not controlled by host. If this bit is enabled, a frame from each active display is sent whenever Window B buffer is switched.
16	**WINDOW_A_NC_DISPLAY: Window A Non-Continuos Display.** 0 Disable 1 Enable This is effective only in Non-Continuous Display mode when window A buffer switching is not controlled by host. If this bit is enabled, a frame from each active display is sent whenever Window A buffer is switched.
6	**SECONDARY_SSF: Secondary Sub-Display Stop Frame (SSF) input.** 0 Disable 1 Enable This is effective only in Non-Continuous Display mode This enable is checked if the next frame to be sent is using Primary register set. When enabled, SSF signal can be input through LDC pin. When SSF is enabled a trigger to send a frame in Non-Continuous Display mode will be delayed until SSF is active.

Bit	Description
5	**PRIMARY_SSF: Primary Sub-Display Stop Frame (SSF) input.** 0 Disable 1 Enable This is effective only in Non-Continuous Display mode This enable is checked if the next frame to be sent is using Primary register set. When enabled, SSF signal can be input through LDC pin. When SSF is enabled a trigger to send a frame in Non-Continuous Display mode will be delayed until SSF is active.
4	**SSF_POLARITY: Sub-Display Stop Frame (SSF) Polarity** 0 Active High 1 Active Low
2	**SECONDARY_MSF: Secondary Main-Display Stop Frame (MSF) input** 0 Disable 1 Enable This is effective only in Non-Continuous Display mode This enable is checked if the next frame to be sent is using Secondary register set. When enabled, MSF signal can be input through LSPI pin. When MSF is enabled a trigger to send a frame in Non-Continuous Display mode will be delayed until MSF is active.
1	**PRIMARY_MSF: Primary Main-Display Stop Frame (MSF) input.** 0 Disable 1 Enable This is effective only in non-continuous display mode. This enable is checked if the next frame to be sent is using Primary register set. When enabled, MSF signal can be input through LSPI pin. When MSF is enabled a // trigger to send a frame in Non-Continuous Display mode will be delayed until MSF is active.
0	**MSF_POLARITY: Main-Display Stop Frame (MSF) Polarity** 0 Active High 1 Active Low

DC_CMD_DISPLAY_COMMAND_OPTION1_0

Offset: 002h
Read/Write: R/W
Reset: 0000.0000

Display Command Option 1

This register is not effective until DISPLAY_COMMAND is written. Buffer Field changes are effective at the same time with buffer switching specified in DISPLAY_COMMAND. So the buffer must be "switched" even if only the buffer field needs to be changed. It is possible to "switched" to the same buffer if only the buffer field needs to be changed.

Bit	Description
10	**S_C_BUFFER_FIELD: Secondary Window C Buffer Field.** 0 Top 1 Bottom
9	**S_B_BUFFER_FIELD: Secondary Window B Buffer Field.** 0 Top 1 Bottom
8	**S_A_BUFFER_FIELD: Secondary Window A Buffer Field.** 0 Top 1 Bottom
2	**P_C_BUFFER_FIELD: Primary Window C Buffer Field.** 0 Top 1 Bottom

Bit	Description
1	**P_B_BUFFER_FIELD: Primary Window B Buffer Field.** 0 Top 1 Bottom
0	**P_A_BUFFER_FIELD: Primary Window A Buffer Field.** 0 Top 1 Bottom

DC_CMD_DISPLAY_COMMAND_0

Offset: 003h
Read/Write: RO
Reset: 0000.0004

Display Command

Bit	Description
30:27	**DISP_COMMAND_RAISE_CHANNEL_ID: Display Command Channel ID**
26:22	**DISP_COMMAND_RAISE_VECTOR: Display Command Raise Vector** This raise vector is returned at the next frame boundary if active display is changed or either at the next line/frame boundary if active display is not changed depending on buffer switching time.
21:20	**S_C_BUFFER_SELECT: Secondary Window C Buffer Select.** 0 = no buffer switching 1 = switch to buffer 0 2 = switch to buffer 1
19:18	**S_B_BUFFER_SELECT: Secondary Window B Buffer Select.** 0 = no buffer switching 1 = switch to buffer 0 2 = switch to buffer 1
17:16	**S_A_BUFFER_SELECT: Secondary Window A Buffer Select.** 0 = no buffer switching 1 = switch to buffer 0 2 = switch to buffer 1
13:12	**P_C_BUFFER_SELECT: Primary Window C Buffer Select.** 0 = no buffer switching 1 = switch to buffer 0 2 = switch to buffer 1
11:10	**P_B_BUFFER_SELECT: Primary Window B Buffer Select.** 0 = no buffer switching 1 = switch to buffer 0 2 = switch to buffer 1
9:8	**P_A_BUFFER_SELECT: Primary Window A Buffer Select.** 0 = no buffer switching 1 = switch to buffer 0 2 = switch to buffer 1

Bit	Description
6:5	**DISPLAY_CTRL_MODE: Display Controller Mode.** 0= Stop Display, this can be used to stop sending frame at the next frame boundary. This is automatically generated in Non-Continous Display after sending one frame of each active display. If this is issued when display controller is already stopped then there is no frame is sent, buffer changes if there is any is executed immediately and raise vector (if raise is enabled) is also returned immediately. This command can also be used in non-continuous display mode to stop accepting non-host trigger conditions from other clients. 1= Continuous Display, the display controller will continuously send frame. Continuous display mode can be stopped by switching to Non-Continuous Display. or by issuing Stop Display. 2= Non-Continuous Display, the display controller is forced to send one frame of each active display and then wait for the next time this command is issued or for other (non-host) trigger conditions to send frame. The sending of frames may be delayed by MSF or SSF input signals from the display device. If a Stop Display is issued while in non-continous display mode then non-host trigger conditions will no longer be accepted until the next time Non-Continuous Display is issued 3= no change to display controller mode
4	**WIN_BUFFER_SWITCH_TIME: Window Buffer Switch Time.** 0 = switch window buffer at frame start (delayed buffer switch) 1 = switch window buffer at line start (immediate buffer switch)
3:2	**ACTIVE_DISPLAY_SELECT: Active Display Select.** 0 = no change to active display settings 1 = Primary Display active 2 = Secondary Display active 3 = both Primary and Secondary displays active. In this case, Primary display frame will be sent first followed by Secondary display frame. This takes effect on frame boundary.
0	**DISP_COMMAND_RAISE: Display Command Raise.** 0 Disable 1 Enable

DC_CMD_DISP_STATUS_0

Offset: 004h
Read/Write: RO
Reset: 0000.0000

Display Status

Read-only register to read the status of active display, buffer selection and field selection. These reflects values are set through the Display Command class.

Bit	Description
14	**WIN_C_FIELD_SELECT_STATUS: Window C Field Select Status**
13	**WIN_B_FIELD_SELECT_STATUS: Window B Field Select Status**
12	**WIN_A_FIELD_SELECT_STATUS: Window A Field Select Status**
6	**WIN_C_BUFFER_SELECT_STATUS: Window C Buffer Select Status**
5	**WIN_B_BUFFER_SELECT_STATUS: Window B Buffer Select Status**
4	**WIN_A_BUFFER_SELECT_STATUS: Window A Buffer Select Status**
1	**SECONDARY_DISPLAY_STATUS: Secondary Display Status**
0	**PRIMARY_DISPLAY_STATUS: Primary Display Status**

DC_CMD_SIGNAL_RAISE_0

Offset: 005h
Read/Write: R/W
Reset: 0000.0000

When written, next occurrence of the selected SIGNAL will cause a RAISE to be sent to the host. Software must not write this register if a previous request (from previous write) is still outstanding.

Bit	Description
19:16	**SIGNAL_RAISE_CHANNEL_ID: Signal Raise Channel ID**
9:8	**SIGNAL_RAISE_SELECT: which signal to raise on.** 0 = None, no raise sent back 1 = Frame End Signal 2 = V Blank Signal 3 = V pulse 3 signal
4:0	**SIGNAL_RAISE_VECTOR: bit number to raise**

DC_CMD_SIGNAL_REFCOUNT_0

Offset: 006h
Read/Write: R/W
Reset: 0000.0000

When written, next occurrence of the selected SIGNAL will cause a REFCOUNT to be sent to the host. Software must not write this register if a previous request (from previous write) is still outstanding.

Bit	Description
16	**SIGNAL_REFCOUNT_SELECT: which signal to return refcount on.** 0 = none, return refcount immediately 1 = Frame End signal
15:0	**SIGNAL_REFCOUNT_VECTOR: bit number to refcount**

DC_CMD_WIN_G_DDA_INCREMENT_0

Offset: 007h
Read/Write: R/W
Reset: 0000.0000

Window G DDA Increment
Window G DDA Increment

Bit	Description
16:4	**WIN_G_DDA_INCREMENT: Window G DDA Increment (1.12)** This consists of 1 bit integer and 12-bit fraction and it must not be programmed larger than 1.0. This DDA controls how often a window G frame is output. Initial value of the DDA is reset to 1.0 whenever window G is disabled or whenever display module is in reset state.

DC_CMD_WIN_G_TRIGGER_0

Offset: 008h
Read/Write: R/W
Reset: 0000.0000

Window G Trigger
This can be used to trigger the outputting of Window G. Window G defines part of display active area which can be sent to encoder pre-processor (EPP) module. Pixel data sent to the EPP is in RGB888 format and this data can be converted by the EPP to planar YUV format for encoding.
The rate of window G output frames when sending continuously are determined by the Window G DDA Increment parameter. The initial value of the DDA is reset to 1.0 whenever Window G frames are not output or whenever display module is in reset/disabled state. The value of the DDA is immediately reset after the last output frame is sent so even if a back-to-back triggers are executed, the DDA will always be reset in between.

Bit	Description
28:24	**WIN_G_RAISE_VECTOR: Win G Raise Vector** This raise vector is returned as described by WIN_G_COMMAND.
19:16	**WIN_G_RAISE_CHANNEL_ID: Win G Channel ID**
3:2	**WIN_G_COMMAND: Window G Command.** 0= Disable continuous output frames at the end of current frame. If raise is enabled, the raise vector is returned after the current output frame is completely sent. 1= Enable continuous output frames starting with the next frame boundary. If raise is enabled, the raise vector is returned as soon as the first output frame starts. 2= Output only 1 frame. If raise is enabled, the raise vector is returned after the output frame is completely sent.
1	**WIN_G_RAISE_OUTPUT: Window G Raise Output.** 0= Disable - Window G raise vector is returned by display module at the end of command completion. 1= Enable - Window G raise vector is not returned by display module but instead it is sent to EPP together with channel ID at the end of command completion. If raise is enabled for continuous output frame then the raise vector is sent to EPP just prior to the first frame pixel data.
0	**WIN_G_RAISE: Window G Raise.** 0 = Disable 1 = Enable - raise vector will be returned at the end of command completion

DC_CMD_DISPLAY_POWER_CONTROL_0

Offset: 009h
Read/Write: R/W
Reset: 0000.0000

Display Power Control
Display Power Control

Bit	Description
25	**HSPI_ENABLE: Host SPI write cycle Enable.** SPI_ENABLE must be enabled for this bit to be effective. 0 = Disable 1 = Enable
24	**SPI_ENABLE: SPI interface Enable.** 0 = Disable 1 = Enable - this enables clock to SPI interface logic for Host SPI, IS SPI, and LCD SPI.

Bit	Description
18	**PM1_ENABLE: PM1 signal Enable.** 0 = Disable 1 = Enable
16	**PM0_ENABLE: PM0 signal Enable.** 0 = Disable 1 = Enable This signal can be output at the pad for display power sequencing.
8	**PW4_ENABLE: PW4 signal Enable.** 0 = Disable 1 = Enable This signal can be output at the pad for display power sequencing.
6	**PW3_ENABLE: PW3 signal Enable.** 0 = Disable 1 = Enable This signal can be output at the pad for display power sequencing.
4	**PW2_ENABLE: PW2 signal Enable.** This signal controls pixel data processing. It should be enabled during V blank time. This signal also controls the time when pin polarity takes effect at the pad. This signal can be output at the pad for display power sequencing. 0 = Disable 1 = Enable
2	**PW1_ENABLE: PW1 signal Enable.** 0 = Disable 1 = Enable This signal can be output at the pad for display power sequencing.
0	**PW0_ENABLE: PW0 signal Enable.** This signal controls the display H and V counters. It must be enabled first and disabled last during display power sequencing. This signal can be output at the pad for display power sequencing. 0 = Disable 1 = Enable

DC_CMD_INT_STATUS_0

Offset: 00ah
Read/Write: RO
Reset: 0000.0000

Interrupt Status

This reflects status of all pending interrupts which is valid as long as the interrupt is not cleared even if the interrupt is masked. A pending interrupt can be cleared by writing a '1' to this the corresponding interrupt status bit in this register.

Display Interrupt and Status

Bit	Description
13	**SSF_INT: Sub-Display Stop Frame Interrupt Status** 0= interrupt not pending 1= interrupt pending
12	**MSF_INT: Main-Display Stop Frame Interrupt Status** 0= interrupt not pending 1= interrupt pending
11	**EPP_OF_INT: Display2epp Overflow Interrupt Status** 0= interrupt not pending 1= interrupt pending

Bit	Description
10	**WIN_C_UF_INT: Window C Underflow Interrupt Status** 0= interrupt not pending 1= interrupt pending
9	**WIN_B_UF_INT: Window B Underflow Interrupt Status** 0= interrupt not pending 1= interrupt pending
8	**WIN_A_UF_INT: Window A Underflow Interrupt Status** 0= interrupt not pending 1= interrupt pending
7	**SPI_BUSY_INT: SPI Busy Interrupt Status** 0= interrupt not pending 1= interrupt pending
4	**V_PULSE3_INT: Vertical Pulse 3 Interrupt** 0= interrupt not pending 1= interrupt pending
3	**H_BLANK_INT: Horizontal Blank Interrupt** 0= interrupt not pending 1= interrupt pending
2	**V_BLANK_INT: Vertical Blank Interrupt** 0= interrupt not pending 1= interrupt pending
1	**FRAME_END_INT: Frame End Interrupt** 0= interrupt not pending 1= interrupt pending
0	**CTXSW_INT: Context Switch Interrupt Status** (This is cleared on write) 0= interrupt not pending 1= interrupt pending

DC_CMD_INT_MASK_0

Offset: 00bh
Read/Write: R/W
Reset: 0000.0000

Interrupt Mask

Setting bits in this register masked the corresponding interrupt but does not clear a pending interrupt and does not prevent a pending interrupt to be generated. Masking an interrupt also does not clear a pending interrupt status and does not a pending interrupt status to be generated.

Bit	Description
13	**SSF_INT_MASK: Sub-Display Stop Frame Interrupt Mask.** 0 = Interrupt masked 1 = Interrrupt not masked
12	**MSF_INT_MASK: Main-Display Stop Frame Interrupt Mask.** 0 = Interrupt masked 1 = Interrrupt not masked
11	**EPP_OF_INT_MASK: Display2epp Overflow Interrupt Mask.** 0 = Interrupt masked 1 = Interrrupt not masked
10	**WIN_C_UF_INT_MASK: Window C Underflow Interrupt Mask.** 0 = Interrupt masked 1 = Interrrupt not masked

Bit	Description
9	**WIN_B_UF_INT_MASK: Window B Underflow Interrupt Mask.** 0 = Interrupt masked 1 = Interrrupt not masked
8	**WIN_A_UF_INT_MASK: Window A Underflow Interrupt Mask.** 0 = Interrupt masked 1 = Interrrupt not masked
7	**SPI_BUSY_INT_MASK: SPI Busy Interrupt Mask.** 0 = Interrupt masked 1 = Interrrupt not masked
6:5	**Reserved**
4	**V_PULSE3_INT_MASK: Vertical Pulse 3 Interrupt Mask.** 0 = Interrupt masked 1 = Interrrupt not masked
3	**H_BLANK_INT_MASK: Horizontal Blank Interrupt Mask.** 0 = Interrupt masked 1 = Interrrupt not masked
2	**V_BLANK_INT_MASK: Vertical Blank Interrupt Mask.** 0 = Interrupt masked 1 = Interrrupt not masked
1	**FRAME_END_INT_MASK: Frame End Interrupt Mask.** 0 = Interrupt masked 1 = Interrrupt not masked
0	**CTXSW_INT_MASK: Context Switch Interrupt Mask.** 0 = Interrupt masked 1 = Interrrupt not masked

DC_CMD_INT_ENABLE_0

Offset: 00ch
Read/Write: R/W
Reset: 0000.0001

Interrupt Enable

Setting bits in this register enable the corresponding interrupt event to generate a pending interrupt. Interrupt output signal will be activated only if the corresponding interrupt is not masked. Disabling an interrupt will not clear a corresponding pending interrupt - it only prevent a new interrupt event to generate a pending interrupt.

Bit	Description
13	**SSF_INT_ENABLE: Sub-Display Stop Frame Interrupt Enable.** 0 = Disable interrupt 1 = Enable interrupt
12	**MSF_INT_ENABLE: Main-Display Stop Frame Interrupt Enable.** 0 = Disableinterrupt 1 = Enable interrupt
11	**EPP_OF_INT_ENABLE: Display2epp Overflow Interrupt Enable.** 0 = Disable interrupt 1 = Enable interrupt
10	**WIN_C_UF_INT_ENABLE: Window C Underflow Interrupt Enable.** 0 = Disable interrupt 1 = Enable interrupt
9	**WIN_B_UF_INT_ENABLE: Window B Underflow Interrupt Enable.** 0 = Disable interrupt 1 = Enable interrupt

Bit	Description
8	**WIN_A_UF_INT_ENABLE: Window A Underflow Interrupt Enable.** 0 = Disable interrupt 1 = Enable interrupt
7	**SPI_BUSY_INT_ENABLE: SPI Busy Interrupt Enable.** 0 = Disable interrupt 1 = Enable interrupt
4	**V_PULSE3_INT_ENABLE: Vertical Pulse 3 Interrupt Enable.** 0 = Interrupt masked 1 = Interrupt not masked
3	**H_BLANK_INT_ENABLE: Horizontal Blank Interrupt Enable.** 0 = Interrupt masked 1 = Interrupt not masked
2	**V_BLANK_INT_ENABLE: Vertical Blank Interrupt Enable.** 0 = Interrupt masked 1 = Interrupt not masked
1	**FRAME_END_INT_ENABLE: Frame End Interrupt Enable.** 0 = Interrupt masked 1 = Interrupt not masked
0	**CTXSW_INT_ENABLE: Context Switch Interrupt Enable.** 0 = Disable interrupt 1 = Enable interrupt

DC_CMD_INT_TYPE_0

Offset: 00dh
Read/Write: R/W
Reset: 0000.0000

Interrupt Type

Two interrupt types are available:

a. Edge interrupt - transition on input signal/event generates pending interrupt

b. Level interrupt - active level on input signal/event generates pending interrupt

Bit	Description
13	**SSF_INT_TYPE: Sub-Display Stop Frame Interrupt Type.** 0 = Edge interrupt 1 = Level interrupt
12	**MSF_INT_TYPE: Main-Display Stop Frame Interrupt Type.** 0 = Edge interrupt 1 = Level interrupt
11	**EPP_OF_INT_TYPE: Display2epp Overflow Interrupt Type.** 0 = Edge interrupt 1 = Level interrupt
10	**WIN_C_UF_INT_TYPE: Window C Underflow Interrupt Type.** 0 = Edge interrupt 1 = Level interrupt
9	**WIN_B_UF_INT_TYPE: Window B Underflow Interrupt Type.** 0 = Edge interrupt 1 = Level interrupt
8	**WIN_A_UF_INT_TYPE: Window A Underflow Interrupt Type.** 0 = Edge interrupt 1 = Level interrupt

Bit	Description
7	**SPI_BUSY_INT_TYPE: SPI Busy Interrupt Type.** 0 = Edge interrupt 1 = Level interrupt
4	**V_PULSE3_INT_TYPE: Vertical Pulse 3 Interrupt Type.** 0 = Edge interrupt 1 = Level interrupt
3	**H_BLANK_INT_TYPE: Horizontal Blank Interrupt Type.** 0 = Edge interrupt 1 = Level interrupt
2	**V_BLANK_INT_TYPE: Vertical Blank Interrupt Type.** 0 = Edge interrupt 1 = Level interrupt
1	**FRAME_END_INT_TYPE: Frame End Interrupt Type.** 0 = Edge interrupt 1 = Level interrupt

DC_CMD_INT_POLARITY_0

Offset: 00eh
Read/Write: R/W
Reset: 0000.0000

Interrupt Polarity

For edge interrupt, these bits specify whether a pending interrupt is generated on falling edge or on rising edge of the corresponding input signal/event. For level interrupt, these bits specify whether a pending interrupt is generated on low level or on high level of the corresponding input signal/event.

0 rw CTXSW_INT_POLARITY init=0 Context Switch Interrupt Polarity enum (LOW, HIGH)

0 = falling edge or low level interrupt

1 = rising edge or high level interrupt

Bit	Description
13	**SSF_INT_POLARITY: Sub-Display Stop Frame Interrupt Polarity.** 0 = Falling edge or low-level interrupt 1 = Rising edge or high-level interrupt
12	**MSF_INT_POLARITY: Main-Display Stop Frame Interrupt Polarity.** 0 = Falling edge or low-level interrupt 1 = Rising edge or high-level interrupt
11	**EPP_OF_INT_POLARITY: Display2epp Overflow Interrupt Polarity.** 0 = Falling edge or low-level interrupt 1 = Rising edge or high-level interrupt
10	**WIN_C_UF_INT_POLARITY: Window A Underflow Interrupt Polarity.** 0 = Falling edge or low-level interrupt 1 = Rising edge or high-level interrupt
9	**WIN_B_UF_INT_POLARITY: Window A Underflow Interrupt Polarity.** 0 = Falling edge or low-level interrupt 1 = Rising edge or high-level interrupt
8	**WIN_A_UF_INT_POLARITY: Window A Underflow Interrupt Polarity.** 0 = Falling edge or low-level interrupt 1 = Rising edge or high-level interrupt

Bit	Description
7	**SPI_BUSY_INT_POLARITY: SPI Busy Interrupt Polarity.** 0 = Falling edge or low-level interrupt 1 = Rising edge or high-level interrupt
4	**V_PULSE3_INT_POLARITY: V Pulse 3 Interrupt Polarity.** 0 = Falling edge or low-level interrupt 1 = Rising edge or high-level interrupt
3	**H_BLANK_INT_POLARITY: H Blank Interrupt Polarity.** 0 = Falling edge or low-level interrupt 1 = Rising edge or high-level interrupt
2	**V_BLANK_INT_POLARITY: V Blank Interrupt Polarity.** 0 = Falling edge or low-level interrupt 1 = Rising edge or high-level interrupt
1	**FRAME_END_INT_POLARITY: Frame End Interrupt Polarity.** 0 = Falling edge or low-level interrupt 1 = Rising edge or high-level interrupt

DC_H_DISPLAY_HEADER_0

Offset: 100h
Read/Write: R/W
Reset: 0000.0000

Display Header for programming primary and/or secondary display classes except for primary/secondary display windows. For primary/secondary display windows, use the DISPLAY_WINDOW_HEADER.

Class: Display Programming Header

Bit	Description
1	**S_DISPLAY_SELECT: Secondary Display Select.** 0 = Disable 1 = Enable
0	**P_DISPLAY_SELECT: Primary Display Select.** 0 = Disable 1 = Enable

DC_WH_DISPLAY_WINDOW_HEADER_0

Offset: 200h
Read/Write: R/W
Reset: 0000.0000

Display Window Header for programming display windows and their corresponding buffer start addresses.

Class: Display Window Programming Header

Bit	Description
13	**BUFFER_1_SELECT: Buffer 1 Select.** 0 = Disable 1 = Enable

Bit	Description
12	**BUFFER_0_SELECT: Buffer 0 Select.** 0 = Disable 1 = Enable
6	**WINDOW_C_SELECT: Window C Select.** 0 = Disable 1 = Enable
5	**WINDOW_B_SELECT: Window B Select.** 0 = Disable 1 = Enable
4	**WINDOW_A_SELECT: Window A Select.** 0 = Disable 1 = Enable
1	**S_W_DISPLAY_SELECT: Secondary Display Select.** 0 = Disable 1 = Enable This bit is ignored when writing to color palettes since color palettes are shared for both primary and secondary displays.
0	**P_W_DISPLAY_SELECT: Primary Display Select.** 0 = Disable 1 = Enable This bit is ignored when writing to color palettes since color palettes are shared for both primary and secondary displays.

DC_COM_CRC_CONTROL_0

Offset: 800h
Read/Write: R/W
Reset: 0000.0000

CRC Control

CRC is provided for at speed testing and diagnostic. When CRC is enabled, the CRC logic waits for the next VSync pulse or the one after that (depending on CRC_WAIT) and then it captures one frame of data at the end of display pipeline and computes the CRC value. After, one frame of data is captured, the CRC logic will stop capturing data.

DISPLAY_COMMAND.DISPLAY_CTRL_MODE should be programmed to C_DISPLAY so that CRC works properly. CRC logic takes 8-bit of control signals and 24-bit RGB pixel after dither and after display color (R and B) swap option.

Input [31:0] is {LVP1, LPV0, LHP2, LHP1, LHP0, VSYNC, HSYNC, ACTIVE, R[7:0], G[7:0], B[7:0]

Bit	Description
1	**CRC_WAIT: CRC Wait** 0= 1 Vsync 1= 2 Vsync
0	**CRC_ENABLE: CRC Enable.** 0 = Disable 1 = Enable

DC_COM_CRC_CHECKSUM_0

Offset: 801h
Read/Write: RO
Reset: 0000.0000

CRC Checksum

This register can be read by host after CRC logic stops capturing data.

Bit	Description
31:0	**CRC_CHECKSUM: CRC Checksum**

DC_COM_PIN_OUTPUT_ENABLE0_0

Offset: 802h
Read/Write: R/W
Reset: 0000.0000

Pin Output Enable 0
Pin Output Enable registers

Bit	Description
30	**LD15_OUTPUT_ENABLE: LD15 pin output enable.** 0 = Enable 1 = Disable
28	**LD14_OUTPUT_ENABLE: LD14 pin output enable.** 0 = Enable 1 = Disable
26	**LD13_OUTPUT_ENABLE: LD13 pin output enable.** 0 = Enable 1 = Disable
24	**LD12_OUTPUT_ENABLE: LD12 pin output enable.** 0 = Enable 1 = Disable
22	**LD11_OUTPUT_ENABLE: LD11 pin output enable.** 0 = Enable 1 = Disable
20	**LD10_OUTPUT_ENABLE: LD10 pin output enable.** 0 = Enable 1 = Disable
18	**LD9_OUTPUT_ENABLE: LD9 pin output enable.** 0 = Enable 1 = Disable
16	**LD8_OUTPUT_ENABLE: LD8 pin output enable.** 0 = Enable 1 = Disable
14	**LD7_OUTPUT_ENABLE: LD7 pin output enable.** 0 = Enable 1 = Disable
12	**LD6_OUTPUT_ENABLE: LD6 pin output enable.** 0 = Enable 1 = Disable
10	**LD5_OUTPUT_ENABLE: LD5 pin output enable.** 0 = Enable 1 = Disable

Bit	Description
8	**LD4_OUTPUT_ENABLE: LD4 pin output enable.** 0 = Enable 1 = Disable
6	**LD3_OUTPUT_ENABLE: LD3 pin output enable.** 0 = Enable 1 = Disable
4	**LD2_OUTPUT_ENABLE: LD2 pin output enable.** 0 = Enable 1 = Disable
2	**LD1_OUTPUT_ENABLE: LD1 pin output enable.** 0 = Enable 1 = Disable
0	**LD0_OUTPUT_ENABLE: LD0 pin output enable.** 0 = Enable 1 = Disable

DC_COM_PIN_OUTPUT_ENABLE1_0

Offset: 803h
Read/Write: R/W
Reset: 0000.0000

Pin Output Enable 1

Bit	Description
30	**LHS_OUTPUT_ENABLE: LHS pin output enable.** 0 = Enable 1 = Disable
28	**LVS_OUTPUT_ENABLE: LVS pin output enable.** 0 = Enable 1 = Disable
26	**LSC1_OUTPUT_ENABLE: LSC1 pin output enable.** 0 = Enable 1 = Disable
24	**LSC0_OUTPUT_ENABLE: LSC0 pin output enable.** 0 = Enable 1 = Disable
20	**LPW2_OUTPUT_ENABLE: LPW2 pin output enable.** 0 = Enable 1 = Disable
18	**LPW1_OUTPUT_ENABLE: LPW1 pin output enable.** 0 = Enable 1 = Disable
16	**LPW0_OUTPUT_ENABLE: LPW0 pin output enable.** 0 = Enable 1 = Disable
2	**LD17_OUTPUT_ENABLE: LD17 pin output enable.** 0 = Enable 1 = Disable
0	**LD16_OUTPUT_ENABLE: LD16 pin output enable.** 0 = Enable 1 = Disable

DC_COM_PIN_OUTPUT_ENABLE2_0

Offset: 804h
Read/Write: R/W
Reset: 0050.0000

Pin Output Enable 2

Bit	Description
22	**LPP_OUTPUT_ENABLE: LPP pin output enable.** 0 = Enable 1 = Disable (default after reset)
20	**LDI_OUTPUT_ENABLE: LDI pin output enable.** 0 = Enable 1 = Disable (default after reset)
18	**LM1_OUTPUT_ENABLE: LM1 pin output enable.** 0 = Enable 1 = Disable
16	**LM0_OUTPUT_ENABLE: LM0 pin output enable.** 0 = Enable 1 = Disable
10	**LVP1_OUTPUT_ENABLE: LVP1 pin output enable.** 0 = Enable 1 = Disable
8	**LVP0_OUTPUT_ENABLE: LVP0 pin output enable.** 0 = Enable 1 = Disable
4	**LHP2_OUTPUT_ENABLE: LHP2 pin output enable.** 0 = Enable 1 = Disable
2	**LHP1_OUTPUT_ENABLE: LHP1 pin output enable.** 0 = Enable 1 = Disable
0	**LHP0_OUTPUT_ENABLE: LHP0 pin output enable.** 0 = Enable 1 = Disable

DC_COM_PIN_OUTPUT_ENABLE3_0

Offset: 805h
Read/Write: R/W
Reset: 0000.0155

Pin Output Enable 3

Bit	Description
8	**LSPI_OUTPUT_ENABLE: LSPI pin output enable.** 0 = Enable 1 = Disable
6	**LDC_OUTPUT_ENABLE: LDC pin output enable.** 0 = Enable 1 = Disable

Bit	Description
4	**LCSN_OUTPUT_ENABLE: LCSN pin output enable.** 0 = Enable 1 = Disable
2	**LSDA_OUTPUT_ENABLE: LSDA pin output enable.** 0 = Enable 1 = Disable
0	**LSCK_OUTPUT_ENABLE: LSCK pin output enable.** 0 = Enable 1 = Disable

DC_COM_PIN_OUTPUT_POLARITY0_0

Offset: 806h
Read/Write: R/W
Reset: 0000.0000

Pin Output Polarity 0

Pin Output Polarity Registers

Bit	Description
30	**LD15_OUTPUT_POLARITY: LD15 pin output polarity.** 0 = Active High 1 = Active Low
28	**LD14_OUTPUT_POLARITY: LD14 pin output polarity.** 0 = Active High 1 = Active Low
26	**LD13_OUTPUT_POLARITY: LD13 pin output polarity.** 0 = Active High 1 = Active Low
24	**LD12_OUTPUT_POLARITY: LD12 pin output polarity.** 0 = Active High 1 = Active Low
22	**LD11_OUTPUT_POLARITY: LD11 pin output polarity.** 0 = Active High 1 = Active Low
20	**LD10_OUTPUT_POLARITY: LD10 pin output polarity.** 0 = Active High 1 = Active Low
18	**LD9_OUTPUT_POLARITY: LD9 pin output polarity.** 0 = Active High 1 = Active Low
16	**LD8_OUTPUT_POLARITY: LD8 pin output polarity.** 0 = Active High 1 = Active Low
14	**LD7_OUTPUT_POLARITY: LD7 pin output polarity.** 0 = Active High 1 = Active Low
12	**LD6_OUTPUT_POLARITY: LD6 pin output polarity.** 0 = Active High 1 = Active Low
10	**LD5_OUTPUT_POLARITY: LD5 pin output polarity.** 0 = Active High 1 = Active Low

Bit	Description
8	**LD4_OUTPUT_POLARITY: LD4 pin output polarity.** 0 = Active High 1 = Active Low
6	**LD3_OUTPUT_POLARITY: LD3 pin output polarity.** 0 = Active High 1 = Active Low
4	**LD2_OUTPUT_POLARITY: LD2 pin output polarity.** 0 = Active High 1 = Active Low
2	**LD1_OUTPUT_POLARITY: LD1 pin output polarity.** 0 = Active High 1 = Active Low
0	**LD0_OUTPUT_POLARITY: LD0 pin output polarity.** 0 = Active High 1 = Active Low

DC_COM_PIN_OUTPUT_POLARITY1_0

Offset: 807h
Read/Write: R/W
Reset: 0000.0000

Pin Output Polarity 1

Bit	Description
30	**LHS_OUTPUT_POLARITY: LHS pin output polarity.** 0 = Active High 1 = Active Low
28	**LVS_OUTPUT_POLARITY: LVS pin output polarity.** 0 = Active High 1 = Active Low
26	**LSC1_OUTPUT_POLARITY: LSC1 pin output polarity.** 0 = Active High 1 = Active Low
24	**LSC0_OUTPUT_POLARITY: LSC0 pin output polarity.** 0 = Active High 1 = Active Low
20	**LPW2_OUTPUT_POLARITY: LPW2 pin output polarity.** 0 = Active High 1 = Active Low
18	**LPW1_OUTPUT_POLARITY: LPW1 pin output polarity.** 0 = Active High 1 = Active Low
16	**LPW0_OUTPUT_POLARITY: LPW0 pin output polarity.** 0 = Active High 1 = Active Low
2	**LD17_OUTPUT_POLARITY: LD17 pin output polarity.** 0 = Active High 1 = Active Low
0	**LD16_OUTPUT_POLARITY: LD16 pin output polarity.** 0 = Active High 1 = Active Low

DC_COM_PIN_OUTPUT_POLARITY2_0

Offset: 808h
Read/Write: R/W
Reset: 0000.0000

Pin Output Polarity 2: High means active high, low means active low.

Bit	Description
22	**LPP_OUTPUT_POLARITY: LPP pin output polarity.** 0 = High 1 = Low
20	**LDI_OUTPUT_POLARITY: LDI pin output polarity.** 0 = High 1 = Low
18	**LM1_OUTPUT_POLARITY: LM1 pin output polarity.** 0 = High 1 = Low
16	**LM0_OUTPUT_POLARITY: LM0 pin output polarity.** **0 = High** **1 = Low**
10	**LVP1_OUTPUT_POLARITY: LVP1 pin output polarity.** 0 = HIGH 1 = LOW
8	**LVP0_OUTPUT_POLARITY: LVP0 pin output polarity.** 0 = HIGH 1 = LOW
4	**LHP2_OUTPUT_POLARITY: LHP2 pin output polarity.** 0 = High 1 = Low
2	**LHP1_OUTPUT_POLARITY: LHP1 pin output polarity.** 0 = High 1 = Low
0	**LHP0_OUTPUT_POLARITY: LHP0 pin output polarity.** 0 = High 1 = Low

DC_COM_PIN_OUTPUT_POLARITY3_0

Offset: 809h
Read/Write: R/W
Reset: 0000.0000

Pin Output Polarity 3: High means active high, Low means active low.

Bit	Description
8	**LSPI_OUTPUT_POLARITY: LSPI pin output polarity.** 0 = High 1 = Low
6	**LDC_OUTPUT_POLARITY: LDC pin output polarity.** 0 = High 1 = Low
4	**LCSN_OUTPUT_POLARITY: LCSN pin output polarity.** 0 = High 1 = Low

Bit	Description
2	**LSDA_OUTPUT_POLARITY: LSDA pin output polarity.** 0 = High 1 = Low
0	**LSCK_OUTPUT_POLARITY: LSCK pin output polarity.** 0 = High 1 = Low

DC_COM_PIN_OUTPUT_DATA0_0

Offset: 80ah
Read/Write: R/W
Reset: 0000.0000

Pin Output Data 0
Pin Output data registers
To change output data, the corresponding mask should be disabled (not masked).

Bit	Description
31	**LD15_OUTPUT_DATA_MASK: LD15 pin output data mask.** 0 = Masked 1 = Notmasked
30	**LD15_OUTPUT_DATA: LD15 pin output data.** 0 = Low 1 = High
29	**LD14_OUTPUT_DATA_MASK: LD14 pin output data mask.** 0 = Masked 1 = Not masked
28	**LD14_OUTPUT_DATA: LD14 pin output data.** 0 = Low 1 = High
27	**LD13_OUTPUT_DATA_MASK: LD13 pin output data mask.** 0 = Masked 1 = Not masked
26	**LD13_OUTPUT_DATA: LD13 pin output data.** 0 = Low 1 = High
25	**LD12_OUTPUT_DATA_MASK: LD12 pin output data mask.** 0 = Masked 1 = Not masked
24	**LD12_OUTPUT_DATA: LD12 pin output data.** 0 = Low 1 = High
23	**LD11_OUTPUT_DATA_MASK: LD11 pin output data mask.** 0 = Masked 1 = Not masked
22	**LD11_OUTPUT_DATA: LD11 pin output data.** 0 = Low 1 = High
21	**LD10_OUTPUT_DATA_MASK: LD10 pin output data mask.** 0 = Masked 1 = Not masked
20	**LD10_OUTPUT_DATA: LD10 pin output data.** 0 = Low 1 = High

Bit	Description
19	**LD9_OUTPUT_DATA_MASK: LD9 pin output data mask.** 0 = Masked 1 = Not masked
18	**LD9_OUTPUT_DATA: LD9 pin output data.** 0 = Low 1 = High
17	**LD8_OUTPUT_DATA_MASK: LD8 pin output data mask.** 0 = Masked 1 = Not masked
16	**LD8_OUTPUT_DATA: LD8 pin output data.** 0 = Low 1 = High
15	**LD7_OUTPUT_DATA_MASK: LD7 pin output data mask.** 0 = Masked 1 = Not masked
14	**LD7_OUTPUT_DATA: LD7 pin output data.** 0 = Low 1 = High
13	**LD6_OUTPUT_DATA_MASK: LD6 pin output data mask.** 0 = Masked 1 = Not masked
12	**LD6_OUTPUT_DATA: LD6 pin output data.** 0 = Low 1 = High
11	**LD5_OUTPUT_DATA_MASK: LD5 pin output data mask.** 0 = Masked 1 = Not masked
10	**LD5_OUTPUT_DATA: LD5 pin output data.** 0 = Low 1 = High
9	**LD4_OUTPUT_DATA_MASK: LD4 pin output data mask.** 0 = Masked 1 = Not masked
8	**LD4_OUTPUT_DATA: LD4 pin output data.** 0 = Low 1 = High
7	**LD3_OUTPUT_DATA_MASK: LD3 pin output data mask.** 0 = Masked 1 = Not masked
6	**LD3_OUTPUT_DATA: LD3 pin output data.** 0 = Low 1 = High
5	**LD2_OUTPUT_DATA_MASK: LD2 pin output data mask.** 0 = Masked 1 = Not masked
4	**LD2_OUTPUT_DATA: LD2 pin output data.** 0 = Low 1 = High
3	**LD1_OUTPUT_DATA_MASK: LD1 pin output data mask.** 0 = Masked 1 = Not masked
2	**LD1_OUTPUT_DATA: LD1 pin output data.** 0 = Low 1 = High

Bit	Description
1	**LD0_OUTPUT_DATA_MASK: LD0 pin output data mask.** 0 = Masked 1 = Not masked
0	**LD0_OUTPUT_DATA: LD0 pin output data.** 0 = Low 1 = High

DC_COM_PIN_OUTPUT_DATA1_0

Offset: 80bh
Read/Write: R/W
Reset: 0000.0000

Pin Output Data 1

Bit	Description
31	**LHS_OUTPUT_DATA_MASK: LHS pin output data mask.** 0 = Masked 1 = Not masked
30	**LHS_OUTPUT_DATA: LHS pin output data.** 0 = Low 1 = High
29	**LVS_OUTPUT_DATA_MASK: LVS pin output data mask.** 0 = Masked 1 = Not masked
28	**LVS_OUTPUT_DATA: LVS pin output data.** 0 = Low 1 = High
27	**LSC1_OUTPUT_DATA_MASK: LSC1 pin output data mask.** 0 = Masked 1 = Not masked
26	**LSC1_OUTPUT_DATA: LSC1 pin output data.** 0 = Low 1 = High
25	**LSC0_OUTPUT_DATA_MASK: LSC0 pin output data mask.** 0 = Masked 1 = Not masked
24	**LSC0_OUTPUT_DATA: LSC0 pin output data.** 0 = Low 1 = High
21	**LPW2_OUTPUT_DATA_MASK: LPW2 pin output data mask.** 0 = Masked 1 = Not masked
20	**LPW2_OUTPUT_DATA: LPW2 pin output data.** 0 = Low 1 = High
19	**LPW1_OUTPUT_DATA_MASK: LPW1 pin output data mask.** 0 = Masked 1 = Not masked
18	**LPW1_OUTPUT_DATA: LPW1 pin output data.** 0 = Low 1 = High
17	**LPW0_OUTPUT_DATA_MASK: LPW0 pin output data mask.** 0 = Masked 1 = Not masked

Bit	Description
16	**LPW0_OUTPUT_DATA: LPW0 pin output data.** 0 = Low 1 = High
3	**LD17_OUTPUT_DATA_MASK: LD17 pin output data mask.** 0 = Masked 1 = Not masked
2	**LD17_OUTPUT_DATA: LD17 pin output data.** 0 = Low 1 = High
1	**LD16_OUTPUT_DATA_MASK: LD16 pin output data mask.** 0 = Masked 1 = Not masked
0	**LD16_OUTPUT_DATA: LD16 pin output data.** 0 = Low 1 = High

DC_COM_PIN_OUTPUT_DATA2_0

Offset: 80ch
Read/Write: R/W
Reset: 0000.0000

Pin Output Data 2

Bit	Description
23	**LPP_OUTPUT_DATA_MASK: LPP pin output data mask.** 0 = Masked 1 = Not masked
22	**LPP_OUTPUT_DATA: LPP pin output data.** 0 = Low 1 = High
21	**LDI_OUTPUT_DATA_MASK: LDI pin output data mask.** 0 = Masked 1 = Not masked
20	**LDI_OUTPUT_DATA: LDI pin output data.** 0 = Low 1 = High
19	**LM1_OUTPUT_DATA_MASK: LM1 pin output data mask.** 0 = Masked 1 = Not masked
18	**LM1_OUTPUT_DATA: LM1 pin output data.** 0 = Low 1 = High
17	**LM0_OUTPUT_DATA_MASK: LM0 pin output data mask.** 0 = Masked 1 = Not masked
16	**LM0_OUTPUT_DATA: LM0 pin output data.** 0 = Low 1 = High
11	**LVP1_OUTPUT_DATA_MASK: LVP1 pin output data mask.** 0 = Masked 1 = Not masked
10	**LVP1_OUTPUT_DATA: LVP1 pin output data.** 0 = Low 1 = High

Bit	Description
9	**LVP0_OUTPUT_DATA_MASK: LVP0 pin output data mask.** 0 = Masked 1 = Not masked
8	**LVP0_OUTPUT_DATA: LVP0 pin output data.** 0 = Low 1 = High
5	**LHP2_OUTPUT_DATA_MASK: LHP2 pin output data mask.** 0 = Masked 1 = Not masked
4	**LHP2_OUTPUT_DATA: LHP2 pin output data.** 0 = Low 1 = High
3	**LHP1_OUTPUT_DATA_MASK: LHP1 pin output data mask.** 0 = Masked 1 = Not masked
2	**LHP1_OUTPUT_DATA: LHP1 pin output data.** 0 = Low 1 = High
1	**LHP0_OUTPUT_DATA_MASK: LHP0 pin output data mask.** 0 = Masked 1 = Not masked
0	**LHP0_OUTPUT_DATA: LHP0 pin output data.** 0 = Low 1 = High

DC_COM_PIN_OUTPUT_DATA3_0

Offset: 80dh
Read/Write: R/W
Reset: 0000.0000

Pin Output Data 3
Pin Output data registers

Bit	Description
9	**LSPI_OUTPUT_DATA_MASK: LSPI pin output data mask.** 0 = Masked 1 = Not masked
8	**LSPI_OUTPUT_DATA: LSPI pin output data.** 0 = Low 1 = High
7	**LDC_OUTPUT_DATA_MASK: LDC pin output data mask.** 0 = Masked 1 = Not masked
6	**LDC_OUTPUT_DATA: LDC pin output data.** 0 = Low 1 = High
5	**LCSN_OUTPUT_DATA_MASK: LCSN pin output data mask.** 0 = Masked 1 = Not masked
4	**LCSN_OUTPUT_DATA: LCSN pin output data.** 0 = Low 1 = High

Bit	Description
3	**LSDA_OUTPUT_DATA_MASK: LSDA pin output data mask.** 0 = Masked 1 = Not masked
2	**LSDA_OUTPUT_DATA: LSDA pin output data.** 0 = Low 1 = High
1	**LSCK_OUTPUT_DATA_MASK: LSCK pin output data mask.** 0 = Masked 1 = Not masked
0	**LSCK_OUTPUT_DATA: LSCK pin output data.** 0 = Low 1 = High

DC_COM_PIN_INPUT_ENABLE0_0

Offset: 80eh
Read/Write: R/W
Reset: 0000.0000

Pin Input Enable 0
Pin Input Enable registers

Bit	Description
30	**LD15_INPUT_ENABLE: LD15 pin input enable.** 0 = Disable 1 = Enable
28	**LD14_INPUT_ENABLE: LD14 pin input enable.** 0 = Disable 1 = Enable
26	**LD13_INPUT_ENABLE: LD13 pin input enable.** 0 = Disable 1 = Enable
24	**LD12_INPUT_ENABLE: LD12 pin input enable.** 0 = Disable 1 = Enable
22	**LD11_INPUT_ENABLE: LD11 pin input enable.** 0 = Disable 1 = Enable
20	**LD10_INPUT_ENABLE: LD10 pin input enable.** 0 = Disable 1 = Enable
18	**LD9_INPUT_ENABLE: LD9 pin input enable.** 0 = Disable 1 = Enable
16	**LD8_INPUT_ENABLE: LD8 pin input enable.** 0 = Disable 1 = Enable
14	**LD7_INPUT_ENABLE: LD7 pin input enable.** 0 = Disable 1 = Enable
12	**LD6_INPUT_ENABLE: LD6 pin input enable.** 0 = Disable 1 = Enable

Bit	Description
10	**LD5_INPUT_ENABLE: LD5 pin input enable.** 0 = Disable 1 = Enable
8	**LD4_INPUT_ENABLE: LD4 pin input enable.** 0 = Disable 1 = Enable
6	**LD3_INPUT_ENABLE: LD3 pin input enable.** 0 = Disable 1 = Enable
4	**LD2_INPUT_ENABLE: LD2 pin input enable.** 0 = Disable 1 = Enable
2	**LD1_INPUT_ENABLE: LD1 pin input enable.** 0 = Disable 1 = Enable
0	**LD0_INPUT_ENABLE: LD0 pin input enable.** 0 = Disable 1 = Enable

DC_COM_PIN_INPUT_ENABLE1_0

Offset: 80fh
Read/Write: R/W
Reset: 0000.0000

Pin Input Enable 1

Bit	Description
30	**LHS_INPUT_ENABLE: LHS pin input enable.** 0 = Disable 1 = Enable
28	**LVS_INPUT_ENABLE: LVS pin input enable.** 0 = Disable 1 = Enable
26	**LSC1_INPUT_ENABLE: LSC1 pin input enable.** 0 = Disable 1 = Enable
24	**LSC0_INPUT_ENABLE: LSC0 pin input enable.** 0 = Disable 1 = Enable
20	**LPW2_INPUT_ENABLE: LPW2 pin input enable.** 0 = Disable 1 = Enable
18	**LPW1_INPUT_ENABLE: LPW1 pin input enable.** 0 = Disable 1 = Enable
16	**LPW0_INPUT_ENABLE: LPW0 pin input enable.** 0 = Disable 1 = Enable
2	**LD17_INPUT_ENABLE: LD17 pin input enable.** 0 = Disable 1 = Enable
0	**LD16_INPUT_ENABLE: LD16 pin input enable.** 0 = Disable 1 = Enable

DC_COM_PIN_INPUT_ENABLE2_0

Offset: 810h
Read/Write: R/W
Reset: 0000.0000

Pin Input Enable 2

Bit	Description
22	**LPP_INPUT_ENABLE: LPP pin input enable.** 0 = Disable 1 = Enable
20	**LDI_INPUT_ENABLE: LDI pin input enable.** 0 = Disable 1 = Enable
18	**LM1_INPUT_ENABLE: LM1 pin input enable.** 0 = Disable 1 = Enable
16	**LM0_INPUT_ENABLE: LM0 pin input enable.** 0 = Disable 1 = Enable
10	**LVP1_INPUT_ENABLE: LVP1 pin input enable.** 0 = Disable 1 = Enable
8	**LVP0_INPUT_ENABLE: LVP0 pin input enable.** 0 = Disable 1 = Enable
4	**LHP2_INPUT_ENABLE: LHP2 pin input enable.** 0 = Disable 1 = Enable
2	**LHP1_INPUT_ENABLE: LHP1 pin input enable.** 0 = Disable 1 = Enable
0	**LHP0_INPUT_ENABLE: LHP0 pin input enable.** 0 = Disable 1 = Enable

DC_COM_PIN_INPUT_ENABLE3_0

Offset: 811h
Read/Write: R/W
Reset: 0000.0000

Pin Input Enable 3

Bit	Description
8	**LSPI_INPUT_ENABLE: LSPI pin input enable.** 0 = Disable 1 = Enable
6	**LDC_INPUT_ENABLE: LDC pin input enable.** 0 = Disable 1 = Enable
4	**LCSN_INPUT_ENABLE: LCSN pin input enable.** 0 = Disable 1 = Enable

Bit	Description
2	**LSDA_INPUT_ENABLE: LSDA pin input enable.** 0 = Disable 1 = Enable
0	**LSCK_INPUT_ENABLE: LSCK pin input enable.** 0 = Disable 1 = Enable

DC_COM_PIN_INPUT_DATA0_0

Offset: 812h
Read/Write: RO
Reset: 0000.0000

Pin Input Data 0
Pin Input Data registers (read-only)

Bit	Description
17	**LD17_INPUT_DATA: LD17 pin input data**
16	**LD16_INPUT_DATA: LD16 pin input data**
15	**LD15_INPUT_DATA: LD15 pin input data**
14	**LD14_INPUT_DATA: LD14 pin input data**
13	**LD13_INPUT_DATA: LD13 pin input data**
12	**LD12_INPUT_DATA: LD12 pin input data**
11	**LD11_INPUT_DATA: LD11 pin input data**
10	**LD10_INPUT_DATA: LD10 pin input data**
9	**LD9_INPUT_DATA: LD9 pin input data**
8	**LD8_INPUT_DATA: LD8 pin input data**
7	**LD7_INPUT_DATA: LD7 pin input data**
6	**LD6_INPUT_DATA: LD6 pin input data**
5	**LD5_INPUT_DATA: LD5 pin input data**
4	**LD4_INPUT_DATA: LD4 pin input data**
3	**LD3_INPUT_DATA: LD3 pin input data**
2	**LD2_INPUT_DATA: LD2 pin input data**
1	**LD1_INPUT_DATA: LD1 pin input data**
0	**LD0_INPUT_DATA: LD0 pin input data**

DC_COM_PIN_INPUT_DATA1_0

Offset: 813h
Read/Write: RO
Reset: 0000.0000

Pin Input Data 1

Bit	Description
24	**LSPI_INPUT_DATA: LSPI pin input data**
23	**LDC_INPUT_DATA: LDC pin input data**
22	**LCSN_INPUT_DATA: LCSN pin input data**
21	**LSDA_INPUT_DATA: LSDA pin input data**
20	**LSCK_INPUT_DATA: LSCK pin input data**
19	**LPP_INPUT_DATA: LPP pin input data**
18	**LDI_INPUT_DATA: LDI pin input data**
17	**LM1_INPUT_DATA: LM1 pin input data**
16	**LM0_INPUT_DATA: LM0 pin input data**
13	**LVP1_INPUT_DATA: LVP1 pin input data**
12	**LVP0_INPUT_DATA: LVP0 pin input data**
10	**LHP2_INPUT_DATA: LHP2 pin input data**
9	**LHP1_INPUT_DATA: LHP1 pin input data**
8	**LHP0_INPUT_DATA: LHP0 pin input data**
7	**LHS_INPUT_DATA: LHS pin input data**
6	**LVS_INPUT_DATA: LVS pin input data**
5	**LSC1_INPUT_DATA: LSC1 pin input data**
4	**LSC0_INPUT_DATA: LSC0 pin input data**
2	**LPW2_INPUT_DATA: LPW2 pin input data**
1	**LPW1_INPUT_DATA: LPW1 pin input data**
0	**LPW0_INPUT_DATA: LPW0 pin input data**

DC_COM_PIN_OUTPUT_SELECT0_0

Offset: 814h
Read/Write: R/W
Reset: 0000.0000

Pin Output Select 0
Pin Output Select
3 bits are used to select output on each pins and they are defined as follows:

Pin Output Select

Pad Name	0	1	2	3	4	5	6	7
	Output Signal	Output Signal	Output Signal	Output Signal	Output Signal	Output Signal	Output Signal	Output Signal
LD17	LD17	LD17 Out	LPD17	0	0	0	0	0
LD16	LD16	LD16 Out	LPD16	0	0	0	0	0
LD15	LD15	LD15 Out	LPD15	0	0	0	0	0
LD14	LD14	LD14 Out	LPD14	0	0	0	0	0
LD13	LD13	LD13 Out	LPD13	0	0	0	0	0
LD12	LD12	LD12 Out	LPD12	0	0	0	0	0
LD11	LD11	LD11 Out	LPD11	0	0	0	0	0
LD10	LD10	LD10 Out	LPD10	0	SD2	0	0	0
LD9	LD9	LD9 Out	LPD9	0	SD2_	0	0	0
LD8	LD8	LD8 Out	LPD8	0	STP	0	0	0
LD7	LD7	LD7 Out	LPD7	0	SDT	0	0	0
LD6	LD6	LD6Out	LPD6	0	STH	0	0	0
LD5	LD5	LD5 Out	LPD5	0	SD1	0	0	0
LD4	LD4	LD4 Out	LPD4	0	SD1_	0	0	0
LD3	LD3	LD3 Out	LPD3	0	SD0	0	0	0
LD2	LD2	LD2 Out	LPD2	0	SD0_	0	0	0
LD1	LD1	LD1 Out	LPD1	0	SC	0	0	0
LD0	LD0	LD0 Out	LPD0	0	SC_	0	0	0
LPW0	PW0	LPW0 Out	PW1	PM0	PW2	MD0	0	0
LPW1	PW1	LPW1 Out	PW2	PM1	PW3	MD1	0	0
LPW2	PW2	LPW2 Out	PW3	PM0	PW4	MD2	0	0
LSC0	SC0	LSC0 Out	P23	0	0	0	0	0
LSC1	SC1	LSC1 Out	DE	0	0	0	0	0
LVS	Vsync	LVS Out	0	PM1	0	MD3	0	0
LHS	Hsync	LHS Out	0	PM0	0	MD2	0	0
LHP0	H Pulse 0	LHP0 Out	LD21	PM0	0	MD0	0	0
LHP1	H Pulse 1	LHP1 Out	LD18	PM1	0	MD1	0	0
LHP2	H Pulse 2	LHP2 Out	LD19	PM0	V Pulse 2	MD2	0	0
LVP0	V Pulse 0	LVP0 Out	0	PM0	0	MD3	0	0
LVP1	V Pulse 1	LVP1 Out	LD20	PM1	PW4	MD3	0	0
LM0	M0	LM0 Out	H Pulse 0	PM0	V Pulse 2	MD0	0	0
LM1	M1	LM1 Out	LD21	PM1	V Pulse 3	MD1	0	0
LD1	D1	LDI Out	LD22	PM0	Sub SCS_	MD2	0	0
LPP	PP	LPP Out	LD23	PM1	V Pulse 3	MD3	0	0
LSCK	SCK	LSCK Out	0	PM0	0	MD0	0	0
LSDA	SDA	LSDA Out	Sub SCS_	PM1	0	MD1	0	0
LCS_	Main SCS_	LCS_ Out	LD22	PM0	0	MD2	0	0
LDC	SDC	LDC Out	LD22	PM1	0	MD3	0	0
LSPI	SPI Busy	LSPI Out	DE	PM0	Pix Clk	MD0	0	0

Notes:
1. LD[17-0] contain pixel data for 1-pixel/1-clock parallel interface
2. LPD[17-0] contain pixel data for non 1-pixel/1-clock parallel interface
3. If output select is set to 1, then corresponding Pin Output Data register value is output (pin is used as general purpose output).

Bit	Description
30:28	**LD7_OUTPUT_SELECT: LD7 pin output select**
26:24	**LD6_OUTPUT_SELECT: LD6 pin output select**
22:20	**LD5_OUTPUT_SELECT: LD5 pin output select**
18:16	**LD4_OUTPUT_SELECT: LD4 pin output select**
14:12	**LD3_OUTPUT_SELECT: LD3 pin output select**
10:8	**LD2_OUTPUT_SELECT: LD2 pin output select**
6:4	**LD1_OUTPUT_SELECT: LD1 pin output select**
2:0	**LD0_OUTPUT_SELECT: LD0 pin output select**

DC_COM_PIN_OUTPUT_SELECT1_0

Offset: 815h
Read/Write: R/W
Reset: 0000.0000

Pin Output Select 1: See 'Pin Output Select" Table on page 7-304 for more details.

Bit	Description
30:28	**LD15_OUTPUT_SELECT: LD15 pin output select**
26:24	**LD14_OUTPUT_SELECT: LD14 pin output select**
22:20	**LD13_OUTPUT_SELECT: LD13 pin output select**
18:16	**LD12_OUTPUT_SELECT: LD12 pin output select**
14:12	**LD11_OUTPUT_SELECT: LD11 pin output select**
10:8	**LD10_OUTPUT_SELECT: LD10 pin output select**
6:4	**LD9_OUTPUT_SELECT: LD9 pin output select**
2:0	**LD8_OUTPUT_SELECT: LD8 pin output select**

DC_COM_PIN_OUTPUT_SELECT2_0

Offset: 816h
Read/Write: R/W
Reset: 0000.0000

Pin Output Select 2: See 'Pin Output Select" Table on page 7-304 for more details.

Bit	Description
6:4	**LD17_OUTPUT_SELECT: LD17 pin output select**
2:0	**LD16_OUTPUT_SELECT: LD16 pin output select**

DC_COM_PIN_OUTPUT_SELECT3_0

Offset: 817h
Read/Write: R/W
Reset: 0000.0000

Pin Output Select 3: See 'Pin Output Select" Table on page 7-304 for more details.

Bit	Description
30:28	**LHS_OUTPUT_SELECT: LHS pin output select**
26:24	**LVS_OUTPUT_SELECT: LVS pin output select**
22:20	**LSC1_OUTPUT_SELECT: LSC1 pin output select**
18:16	**LSC0_OUTPUT_SELECT: LSC0 pin output select**
10:8	**LPW2_OUTPUT_SELECT: LPW2 pin output select**
6:4	**LPW1_OUTPUT_SELECT: LPW1 pin output select**
2:0	**LPW0_OUTPUT_SELECT: LPW0 pin output select**

DC_COM_PIN_OUTPUT_SELECT4_0

Offset: 818h
Read/Write: R/W
Reset: 0000.0000

Pin Output Select 4: See 'Pin Output Select" Table on page 7-304 for more details.

Bit	Description
22:20	**LVP1_OUTPUT_SELECT: LVP1 pin output select**
18:16	**LVP0_OUTPUT_SELECT: LVP0 pin output select**
10:8	**LHP2_OUTPUT_SELECT: LHP2 pin output select**
6:4	**LHP1_OUTPUT_SELECT: LHP1 pin output select**
2:0	**LHP0_OUTPUT_SELECT: LHP0 pin output select**

DC_COM_PIN_OUTPUT_SELECT5_0

Offset: 819h
Read/Write: R/W
Reset: 0000.0000

Pin Output Select 5: See 'Pin Output Select" Table on page 7-304 for more details.

Bit	Description
14:12	**LPP_OUTPUT_SELECT: LPP pin output select**
10:8	**LDI_OUTPUT_SELECT: LDI pin output select**
6:4	**LM1_OUTPUT_SELECT: LM1 pin output select**
2:0	**LM0_OUTPUT_SELECT: LM0 pin output select**

DC_COM_PIN_OUTPUT_SELECT6_0

Offset: 81ah
Read/ Write: R/W
Reset: 0000.0000

Pin Output Select 6: See 'Pin Output Select" Table on page 7-304 for more details.

Bit	Description
18:16	**LSPI_OUTPUT_SELECT: LSPI pin output select**
14:12	**LDC_OUTPUT_SELECT: LDC pin output select**
10:8	**LCSN_OUTPUT_SELECT: LCSN pin output select**
6:4	**LSDA_OUTPUT_SELECT: LSDA pin output select**
2:0	**LSCK_OUTPUT_SELECT: LSCK pin output select**

DC_COM_PIN_MISC_CONTROL_0

Offset: 81bh
Read/Write: R/W
Reset: 0000.0000

Pin Miscellaneous Control

Bit	Description
2	**DISP_CLOCK_OUTPUT: Display Clock (DCLK) Enable.** 0 = Disable 1 = Enable display clock to be output on LSPI pin (LSPI output select must be appropriately programmed for this to be effective)

DC_COM_PM0_CONTROL_0

Offset: 81ch
Read/Write: R/W
Reset: 0000.0000

PM0 signal Control

Class: Pulse Width Modulation

PM0 signal is programmable pulse width modulation signal that can be output on several pins. The control register should be initialized once before PM0 is enabled.

Bit	Description
23:18	**PM0_PERIOD: PM0 Period (4, 8, ... , 256 clock cycles)**
9:4	**PM0_CLOCK_DIVIDER: PM0 Clock Divider (1 to 64)**
1:0	**PM0_CLOCK_SELECT: PM0 Clock Select** 0= Output Of Shift Clock Divider 1= Pixel Clock 2= Line Clock 3= Frame Clock Note that pixel clock, line clock, and frame clock run only when PW0 signal is enabled. Note: In non-continuous mode, shift clock and pixel clock run contiuously, but line clock and frame clock run *only while a frame is being sent*.

DC_COM_PM0_DUTY_CYCLE_0

Offset: 81dh
Read/Write: R/W
Reset: 0000.0000

PM0 Duty Cycle

Bit	Description
7:0	**PM0_DUTY_CYCLE: PM0 Duty Cycle** From 1/P to P/P pulse high time where P is the period. This must not be larger than the period.

DC_COM_PM1_CONTROL_0

Offset: 81eh
Read/Write: R/W
Reset: 0000.0000

PM1 signal Control
PM1 signal is programmable pulse width modulation signal that can be output on several pins. The control register should be initialized once before PM1 is enabled.

Bit	Description
23:18	**PM1_PERIOD: PM1 Period (4, 8, ... , 256 clock cycles)**
9:4	**PM1_CLOCK_DIVIDER: PM1 Clock Divider (1 to 64)**
1:0	**PM1_CLOCK_SELECT: PM1 Clock Select** 0= output of shift clock divider 1= pixel clock 2= line clock 3= frame clock Note that pixel clock, line clock, and frame clock is running only when PW0 signal is enabled. Note: In non-continuous mode, shift clock and pixel clock run contiuously, but line clock and frame clock run *only while a frame is being sent*.

DC_COM_PM1_DUTY_CYCLE_0

Offset: 81fh
Read/Write: R/W
Reset: 0000.0000

PM1 Duty Cycle

Bit	Description
7:0	**PM1_DUTY_CYCLE: PM1 Duty Cycle** from 1/P to P/P pulse high time where P is the period. This must not be larger than the period.

DC_COM_SPI_CONTROL_0

Offset: 820h
Read/Write: R/W
Reset: 0000.0000

SPI Control

Class: Serial Peripheral Interface (SPI)

SPI interface is a 3-pin or 4-pin serial interface which is typically used to program registers in display device. However, for display with built-in frame buffer, it can also be used to write pixel data to the built-in frame buffer.

Currently only write cycles are supported. LCD SPI interface signals consists of

1. SPI Clock (SCK) which can be output on LSCK pin.
2. SPI Data (SDA) which can be output on LSDA pin.
3. Optional SPI Data/Command (SDC) which can be output on LDC pin.
4. Main-Display SPI Chip Select (Main SCS_) signal which can be output on LCS_ pin
5. Sub-Display SPI Chip Select (Sub SCS_) signal which can be output on LM0 or LDI or LSDA pins - this is only used only if there is a sub display.

Theoretically if two LCD panels (main and sub displays) are connected, then SPI interface can be connected to both LCD panels and shared between the two panels. In this case two Chip Select pins are required and LCS_ should be connected to the main display and other pin has to be selected to output the sub display SCS_. Internally two SPI chip select signals (lmscs_ and lsscs_ for main and sub display SPI chip select) are generated.

When SPI is enabled, there are three possible SPI transactions:

a. SPI write can be triggered by host by writing to HSPI_CS_DC register
b. SPI write can be triggered to send pixel data from frame buffer
c. SPI write can be triggered to send Initialization Sequence (IS). Frame initialization sequence is sent one line prior to active display line.

For LCD SPI, pixel data can only be sent to either Main-Display or Sub-Display but not to both but host SPI and IS SPI can be sent to both Main and Sub displays simultaneously. In case a. and c. the SPI root clock is derived from output of shift clock divider which is then further divided by SPI_CLK_DIVIDER. For both cases, the SPI number of bits/cycle is determined by SPI_BITS_PER_CYCLE and the SPI data direction is determined by SPI_DATA_DIRECTION.

In case b. the SPI root clock is derived from output of shift clock divider with no further division. For this case, the SPI number of bits/cycle is determined by correct programming of pixel clock divider.

If case a. is enabled at the same time with case b. and c. then host SPI triggers is delayed to the beginning of horizontal display active time of a line that does not have IS SPI or LCD SPI cycles. This means that if all cases are enabled, the vertical blank time must be at least 2 lines otherwise host SPI cycles cannot be executed.

Bit	Description
25:24	**SPI_STATUS_ENABLE: SPI Status Enable** 0 = SPI status disable 1 = SPI status enabled for host SPI only 2 = SPI status enabled for IS and LCD SPI only 3 = SPI status enabled for all SPI cycles SPI status is reflected in SPI_BUSY bit and can generate interrupt. SPI status indicates when SPI module is busy (SPI write cycles are in progress) so its falling edge should be used to generate interrupt. SPI status can also be output on LSPI pin. When Host SPI is triggered, the SPI busy is asserted within three display clock cycles after the end of the host write cycle. SPI status is disabled when SPI_ENABLE bit is disabled.
20:16	**SPI_CLK_DIVIDER: SPI Clock Divider (1 to 32)** This clock divider is used only if SPI is enabled for host writes (Host SPI) or for sending initialization sequence (IS SPI). Programmed value is 1 less than the desired (actual) clock divider value. This parameter is forced to 0 (clock divide by 1) for LCD SPI.
7:4	**SPI_BITS_PER_CYCLE: SPI Bits per Cycle.** 0 = SPI8 1 = SPI8DC 2 = SPI12 3 = SPI16 4 = SPI16DC 5 = SPI16SB 6 = SPI18 7 = SPI24
3	**SPI_DATA_DIRECTION: SPI Data Direction.** 0 = Most Significan Bit to Least Significant Bit 1 **=** Least Significant Bit to Most Significant Bit
1:0	**SPI_SERIAL_CLK_CONTROL: SPI Serial Clock Control** 0= SCK rising edge is active edge: 1-clock chip select and no SCK clock edge to latch chip select 1= SCK rising edge is active edge 2-clock chip select with SCK rising clock edge to latch it 2= SCK falling edge is active edge 1-clock chip select and no SCK clock edge to latch chip select 3= SCK falling edge is active edge 2-clock chip select with SCK falling clock edge to latch it This is valid for Host, IS, and LCD SPI

DC_COM_SPI_START_BYTE_0

Offset: 821h
Read/Write: R/W
Reset: 0000.0000

SPI Start Byte

SPI Start Bytes are used only for SPI16SB mode (start byte plus 16-bit data.)
Note that data direction does not affect the start byte direction (always MSb to LSb) and position (always first 8-bit of serial data).

Bit	Description
15:8	**SPI_COMMAND_START_BYTE: SPI Command Start Byte** This is valid for Host, IS, and LCD SPI.
7:0	**SPI_DATA_START_BYTE: SPI Data Start Byte** This is valid for Host, IS, and LCD SPI.

DC_COM_HSPI_WRITE_DATA_AB_0

Offset: 822h
Read/Write: R/W
Reset: 0000.0000

Host SPI Write Data A & B
These registers are used to send write data for Host SPI write cycles (HSPI_ENABLE = 1).

HSPIWDA = HSPI_WRITE_DATA_A
HSPIWDB = HSPI_WRITE_DATA_B
HSPIWDC = HSPI_WRITE_DATA_C
HSPIWDD = HSPI_WRITE_DATA_D
HSPIMCS = HSPI_MAIN_CS
HSPISCS = HSPI_SUB_CS
HSPIDC = HSPI_DC

For 8-bit SPI, up to 4 write cycles can be performed. (SPI_BITS_PER_CYCLE equals SPI8 or SPI8DC) Note that for SPI8DC mode, nine cycles of SPI are generated where the first data bit sent comes from HSPI_DC register and the other eight bits come as shown in the table below (e.g. data equals HSPIDC[0],HSPIWDA[7-0] for first cycle):

Write cycle	0	1	2	3
data	HSPIWDA[7-0]	HSPIWDA[15-8]	HSPIWDB[7-0]	HSPIWDB[15-8]
Main SCS_	HSPIMCS[0]	HSPIMCS[1]	HSPIMCS[2]	HSPIMCS[3]
Sub SCS_	HSPISCS[0]	HSPISCS[1]	HSPISCS[2]	HSPICSC[3]
SDC	HSPIDC[0]	HSPIDC[1]	HSPIDC[2]	HSPIDC[3]

For 12-bit SPI, up to 4 write cycles can be performed (SPI12):

Write cycle	0	1	2	3
data	HSPIWDA[11-0]	HSPIWDB[11-0]	HSPIWDC[11-0]	HSPIWDD[11-0]
Main SCS_	HSPIMCS[0]	HSPIMCS[1]	HSPIMCS[2]	HSPIMCS[3]
Sub SCS_	HSPISCS[0]	HSPISCS[1]	HSPISCS[2]	HSPISCS[3]
SDC	HSPIDC[0]	HSPIDC[1]	HSPIDC[2]	HSPIDC[3]

For 16-bit SPI, up to 4 write cycles can be performed (SPI16, SPI16DC, SPI16SB):

Write cycle	0	1	2	3
data	HSPIWDA[15-0]	HSPIWDB[15-0]	HSPIWDC[15-0]	HSPIWDD[15-0]
Main SCS_	HSPIMCS[0]	HSPIMCS[1]	HSPIMCS[2]	HSPIMCS[3]
Sub SCS_	HSPISCS[0]	HSPISCS[1]	HSPISCS[2]	HSPISCS[3]
SDC	HSPIDC[0]	HSPIDC[1]	HSPIDC[2]	HSPIDC[3]

For 18-bit SPI, up to 2 write cycles can be performed (SPI18):

Write cycle	0	1
data	HSPIWDB[1-0], HSPIWDA[15-0]	HSPIWDC[1-0], HSPIWDD[15-0]
Main SCS_	HSPIMCS[0]	HSPIMCS[1]
Sub SCS_	HSPISCS[0]	HSPISCS[1]
SDC	HSPIDC[0]	HSPIDC[1]

For 24-bit SPI, up to 2 write cycles can be performed (SPI24):

Write cycle	0	1
data	HSPIWDB[7-0], HSPIWDA[15-0]	HSPIWDC[7-0], HSPIWDD[15-0]
Main SCS_	HSPIMCS[0]	HSPIMCS[1]

Sub SCS_	HSPISCS[0]	HSPISCS[1]
SDC	HSPIDC[0]	HSPIDC[1]

Bit	Description
31:16	**HSPI_WRITE_DATA_B: Host SPI Write Data B bits 15-0**
15:0	**HSPI_WRITE_DATA_A: Host SPI Write Data A bits 15-0**

DC_COM_HSPI_WRITE_DATA_CD_0

Offset: 823h
Read/Write: R/W
Reset: 0000.0000

Host SPI Write Data C & D

Bit	Description
31:16	**HSPI_WRITE_DATA_D: Host SPI Write Data D bits 15-0**
15:0	**HSPI_WRITE_DATA_C: Host SPI Write Data C bits 15-0**

DC_COM_HSPI_CS_DC_0

Offset: 824h
Read/Write: R/W
Reset: 0000.0000

Host SPI Chip Select and Data/Command_
A write to this register will trigger the Host SPI write cycle if SPI_ENABLE and HSPI_ENABLE are both enabled. Writing to this register with HSPI_RAISE enabled will cause the raise vector to be returned after all the host SPI cycles are completed. Up to four host SPI cycles can be executed with a single trigger.

This register should not be written if previous host SPI write cycle is in progress. HSPI_MAIN_CS and HSPI_SUB_CS controls the main and sub display chip selects and therefore also determine how many SPI write cycles to main and sub displays and the write data position.

Bit	Description
28:24	**HSPI_RAISE_VECTOR: Host SPI Raise Vector** This raise vector is returned after all the triggered host SPI cycles are executed.
19:16	**HSPI_RAISE_CHANNEL_ID: Win G Channel ID**
15:12	**HSPI_SUB_CS: Host SPI Sub display Chip Select (Sub SCS_)** 0= Sub display not selected (Sub SCS_=1) 1= Sub display selected (Sub SCS_=0) This is valid for Host SPI only. Each bit of this parameter corresponds to the four possible host SPI cycles.
11:8	**HSPI_MAIN_CS: Host SPI Main display Chip Select (Main SCS_)** 0= Main display not selected (Main SCS_=1) 1= Main display selected (Main SCS_=0) This is valid for Host SPI only. Each bit of this parameter corresponds to the four possible host SPI cycles.

Bit	Description
7:4	**HSPI_DC: Host SPI Data/Command_ (SDC)** 0= Command cycle (SDC=0) 1= Data cycle (SDC=1) This is valid for Host SPI only. Each bit of this parameter corresponds to the four possible host SPI cycles.
0	**HSPI_RAISE: Host SPI Raise.** 0 = Disable 1 = Enable raise vector will be returned at the end of the host SPI write cycles

DC_COM_SCRATCH_REGISTER_A_0

Offset: 825h
Read/Write: R/W
Reset: 0000.0000

Scratch Register A

Class: Software Scratch Registers

Bit	Description
31:0	**SCRATCH_REGISTER_A: Scratch Register A**

DC_COM_SCRATCH_REGISTER_B_0

Offset: 826h
Read/Write: R/W
Reset: 0000.0000

Scratch Register B

Bit	Description
31:0	**SCRATCH_REGISTER_B: Scratch Register B**

DC_WINC_A_COLOR_PALETTE_0

Offset: 400h..4ffh
Read/Write: WO
Reset: 0000.0000

Window B Color Palette

This is an array of 256 identical register entries; the register fields below apply to each entry.

Color palette

This is used for palettized data format (color depth of 8-bpp or less) or for gamma correction for non-palettized data formats (color depth of more than 8-bpp). Each window has its own color palette which consists of three 256x8 register file which can be written by host and indexed (read) by the window.

For palettized data format less than 8-bpp the pixel data is aligned to least significant bits of the palette index (address) and the remaining upper bits are filled with the corresponding bits of the Palette Color Extension. For example, for 4-bpp mode, the pixel data occupies bits 3-0 of the palette index and bits 7-4 of the palette index are set to bits 7-4 of the Palette Color Extension.

Note that host read is assumed to be not needed - software can cache the color palette in system memory. The same color palettes are used for both primary and secondary displays and they can be written as either primary or secondary display color palette.

Bit	Description
23:16	**A_COLOR_PALETTE_B: Blue Color Palette**
15:8	**A_COLOR_PALETTE_G: Green Color Palette**
7:0	**A_COLOR_PALETTE_R: Red Color Palette**

DC_WINC_A_DV_CONTROL_0

Offset: 500h
Read/Write: R/W
Reset: 0000.0000

Window B Digital Vibrance Control

Digital Vibrance (DV) control
If enabled, Digital Vibrance is applied after H and V scaling and after color palette or color space conversion logic but before color keying multiplexer and before cursor multiplexer.

After DV, new R = R + (2R - G - B) * FR, where FR is fraction from 0 to 7/8
After DV, new G = G + (2G - R - B) * FG, where FG is fraction from 0 to 7/8
After DV, new B = B + (2B - R - G) * FB, where FB is fraction from 0 to 7/8

Bit	Description
18:16	**A_DV_CONTROL_B: Digital Vibrance control for B**
10:8	**A_DV_CONTROL_G: Digital Vibrance control for G**
2:0	**A_DV_CONTROL_R: Digital Vibrance control for R**

DC_WINC_B_COLOR_PALETTE_0

Offset: 400h..4ffh
Read/Write: WO
Reset: 0000.0000

Window B Color Palette
This is an array of 256 identical register entries; the register fields below apply to each entry.

Color palette
This is used for palettized data format (color depth of 8-bpp or less) or for gamma correction for non-palettized data formats (color depth of more than 8-bpp). Each window has its own color palette which consists of three 256x8 register file which can be written by host and indexed (read) by the window.

For palettized data format less than 8-bpp the pixel data is aligned to least significant bits of the palette index (address) and the remaining upper bits are filled with the corresponding bits of the Palette Color Extension. For example, for 4-bpp mode, the pixel data occupies bits 3-0 of the palette index and bits 7-4 of the palette index are set to bits 7-4 of the Palette Color Extension.

Note that host read is assumed to be not needed - software can cache the color palette in system memory. The same color palettes are used for both primary and secondary displays and they can be written as either primary or secondary display color palette.

Bit	Description
23:16	**B_COLOR_PALETTE_B: Blue Color Palette**
15:8	**B_COLOR_PALETTE_G: Green Color Palette**
7:0	**B_COLOR_PALETTE_R: Red Color Palette**

DC_WINC_B_DV_CONTROL_0

Offset: 500h
Read/Write: R/W
Reset: 0000.0000

Window B Digital Vibrance Control

Digital Vibrance (DV) control
If enabled, Digital Vibrance is applied after H and V scaling and after color palette or color space conversion logic but before color keying multiplexer and before cursor multiplexer.

After DV, new R = R + (2R - G - B) * FR, where FR is fraction from 0 to 7/8
After DV, new G = G + (2G - R - B) * FG, where FG is fraction from 0 to 7/8
After DV, new B = B + (2B - R - G) * FB, where FB is fraction from 0 to 7/8

Bit	Description
18:16	**B_DV_CONTROL_B: Digital Vibrance control for B**
10:8	**B_DV_CONTROL_G: Digital Vibrance control for G**
2:0	**B_DV_CONTROL_R: Digital Vibrance control for R**

DC_WINC_B_H_FILTER_P00_0

Offset: 501h
Read/Write: R/W
Reset: 0000.0000

Window B Horizontal Filter phase 00

Horizontal scaling filter coefficients.
Horizontal scaling filter is a 6-tap filter with 4-bit positional phase:
Coefficients 0 and 5 are 3-bit signed value ranging from -4 to 3.
Coefficients 1 and 4 are 5-bit signed value ranging from -16 to 15.
Coefficients 2 and 3 are 8-bit unsigned value ranging from 0 to 128.
Coefficient 0 is the multiplier for the earliest pixel (P0) in the group of 6-pixel and coefficient 5 is the multiplier for the latest pixel (P5) in the group. The output pixel positional phase is defined as centered in P2 if the positional phase is 0 or proportionally in between P2 and P3 if the positional phase is larger than 0.

Sum of all coefficients for each phase should be 128 typically and software should never program the sum of all coefficients for a phase to be more than 128. For each horizontal positional phase, the 6 filter coefficients requires 32 reg bits. Note that color value ranges from 0 to 255 for Y, R, G, B and -128 to 127 for U and V. The same H filter coefficients are used for both primary and secondary displays and they can be written as either primary or secondary display H filter coefficients.

Bit	Description
31:29	**B_H_FILTER_P00C5: Phase 00 coefficient 5 (typically 0)**
28:24	**B_H_FILTER_P00C4: Phase 00 coefficient 4 (typically 0)**
23:16	**B_H_FILTER_P00C3: Phase 00 coefficient 3 (typically 0)**
15:8	**B_H_FILTER_P00C2: Phase 00 coefficient 2 (typically 128)**
7:3	**B_H_FILTER_P00C1: Phase 00 coefficient 1 (typically 0)**
2:0	**B_H_FILTER_P00C0: Phase 00 coefficient 0 (typically 0)**

DC_WINC_B_H_FILTER_P01_0

Offset: 502h
Read/Write: R/W
Reset: 0000.0000

Window B Horizontal Filter phase 01

Bit	Description
31:29	**B_H_FILTER_P01C5: Phase 01 coefficient 5 (typically 1)**
28:24	**B_H_FILTER_P01C4: Phase 01 coefficient 4 (typically -2)**
23:16	**B_H_FILTER_P01C3: Phase 01 coefficient 3 (typically 8)**
15:8	**B_H_FILTER_P01C2: Phase 01 coefficient 2 (typically 124)**
7:3	**B_H_FILTER_P01C1: Phase 01 coefficient 1 (typically -4)**
2:0	**B_H_FILTER_P01C0: Phase 01 coefficient 0 (typically 1)**

DC_WINC_B_H_FILTER_P02_0

Offset: 503h
Read/Write: R/W
Reset: 0000.0000

Window B Horizontal Filter phase 02

Bit	Description
31:29	**B_H_FILTER_P02C5: Phase 02 coefficient 5 (typically 1)**
28:24	**B_H_FILTER_P02C4: Phase 02 coefficient 4 (typically -5)**
23:16	**B_H_FILTER_P02C3: Phase 02 coefficient 3 (typically 17)**
15:8	**B_H_FILTER_P02C2: Phase 02 coefficient 2 (typically 122)**
7:3	**B_H_FILTER_P02C1: Phase 02 coefficient 1 (typically -8)**
2:0	**B_H_FILTER_P02C0: Phase 02 coefficient 0 (typically 1)**

DC_WINC_B_H_FILTER_P03_0

Offset: 504h
Read/Write: R/W
Reset: 0000.0000

Window B Horizontal Filter phase 03

Bit	Description
31:29	**B_H_FILTER_P03C5: Phase 03 coefficient 5 (typically 2)**
28:24	**B_H_FILTER_P03C4: Phase 03 coefficient 4 (typically -7)**
23:16	**B_H_FILTER_P03C3: Phase 03 coefficient 3 (typically 27)**
15:8	**B_H_FILTER_P03C2: Phase 03 coefficient 2 (typically 115)**
7:3	**B_H_FILTER_P03C1: Phase 03 coefficient 1 (typically -11)**
2:0	**B_H_FILTER_P03C0: Phase 03 coefficient 0 (typically 2)**

DC_WINC_B_H_FILTER_P04_0

Offset: 505h
Read/Write: R/W
Reset: 0000.0000

Window B Horizontal Filter phase 04

Bit	Description
31:29	**B_H_FILTER_P04C5: Phase 04 coefficient 5 (typically 2)**
28:24	**B_H_FILTER_P04C4: Phase 04 coefficient 4 (typically -9)**
23:16	**B_H_FILTER_P04C3: Phase 04 coefficient 3 (typically 37)**
15:8	**B_H_FILTER_P04C2: Phase 04 coefficient 2 (typically 109)**
7:3	**B_H_FILTER_P04C1: Phase 04 coefficient 1 (typically -13)**
2:0	**B_H_FILTER_P04C0: Phase 04 coefficient 0 (typically 2)**

DC_WINC_B_H_FILTER_P05_0

Offset: 506h
Read/Write: R/W
Reset: 0000.0000

Window B Horizontal Filter phase 05

Bit	Description
31:29	**B_H_FILTER_P05C5: Phase 05 coefficient 5 (typically 2)**
28:24	**B_H_FILTER_P05C4: Phase 05 coefficient 4 (typically -11)**
23:16	**B_H_FILTER_P05C3: Phase 05 coefficient 3 (typically 47)**
15:8	**B_H_FILTER_P05C2: Phase 05 coefficient 2 (typically 102)**
7:3	**B_H_FILTER_P05C1: Phase 05 coefficient 1 (typically -15)**
2:0	**B_H_FILTER_P05C0: Phase 05 coefficient 0 (typically 3)**

DC_WINC_B_H_FILTER_P06_0

Offset: 507h
Read/Write: R/W
Reset: 0000.0000

Window B Horizontal Filter phase 06

Bit	Description
31:29	**B_H_FILTER_P06C5: Phase 06 coefficient 5 (typically 3)**
28:24	**B_H_FILTER_P06C4: Phase 06 coefficient 4 (typically -13)**
23:16	**B_H_FILTER_P06C3: Phase 06 coefficient 3 (typically 56)**
15:8	**B_H_FILTER_P06C2: Phase 06 coefficient 2 (typically 94)**
7:3	**B_H_FILTER_P06C1: Phase 06 coefficient 1 (typically -15)**
2:0	**B_H_FILTER_P06C0: Phase 06 coefficient 0 (typically 3)**

DC_WINC_B_H_FILTER_P07_0

Offset: 508h
Read/Write: R/W
Reset: 0000.0000

Window B Horizontal Filter phase 07

Bit	Description
31:29	**B_H_FILTER_P07C5: Phase 07 coefficient 5 (typically 3)**
28:24	**B_H_FILTER_P07C4: Phase 07 coefficient 4 (typically -14)**
23:16	**B_H_FILTER_P07C3: Phase 07 coefficient 3 (typically 67)**
15:8	**B_H_FILTER_P07C2: Phase 07 coefficient 2 (typically 85)**
7:3	**B_H_FILTER_P07C1: Phase 07 coefficient 1 (typically -16)**
2:0	**B_H_FILTER_P07C0: Phase 07 coefficient 0 (typically 3)**

DC_WINC_B_H_FILTER_P08_0

Offset: 509h
Read/Write: R/W
Reset: 0000.0000

Window B Horizontal Filter phase 08

Bit	Description
31:29	**B_H_FILTER_P08C5: Phase 08 coefficient 5 (typically 3)**
28:24	**B_H_FILTER_P08C4: Phase 08 coefficient 4 (typically -15)**
23:16	**B_H_FILTER_P08C3: Phase 08 coefficient 3 (typically 76)**
15:8	**B_H_FILTER_P08C2: Phase 08 coefficient 2 (typically 76)**
7:3	**B_H_FILTER_P08C1: Phase 08 coefficient 1 (typically -15)**
2:0	**B_H_FILTER_P08C0: Phase 08 coefficient 0 (typically 3)**

DC_WINC_B_H_FILTER_P09_0

Offset: 50ah
Read/Write: R/W
Reset: 0000.0000

Window B Horizontal Filter phase 09

Bit	Description
31:29	**B_H_FILTER_P09C5: Phase 09 coefficient 5 (typically 3)**
28:24	**B_H_FILTER_P09C4: Phase 09 coefficient 4 (typically -16)**
23:16	**B_H_FILTER_P09C3: Phase 09 coefficient 3 (typically 85)**
15:8	**B_H_FILTER_P09C2: Phase 09 coefficient 2 (typically 67)**
7:3	**B_H_FILTER_P09C1: Phase 09 coefficient 1 (typically -14)**
2:0	**B_H_FILTER_P09C0: Phase 09 coefficient 0 (typically 3)**

DC_WINC_B_H_FILTER_P0A_0

Offset: 50bh
Read/Write: R/W
Reset: 0000.0000

Window B Horizontal Filter phase 0A

Bit	Description
31:29	**B_H_FILTER_P0AC5: Phase 0A coefficient 5 (typically 3)**
28:24	**B_H_FILTER_P0AC4: Phase 0A coefficient 4 (typically -15)**
23:16	**B_H_FILTER_P0AC3: Phase 0A coefficient 3 (typically 94)**
15:8	**B_H_FILTER_P0AC2: Phase 0A coefficient 2 (typically 56)**
7:3	**B_H_FILTER_P0AC1: Phase 0A coefficient 1 (typically -13)**
2:0	**B_H_FILTER_P0AC0: Phase 0A coefficient 0 (typically 3)**

DC_WINC_B_H_FILTER_P0B_0

Offset: 50ch
Read/Write: R/W
Reset: 0000.0000

Window B Horizontal Filter phase 0B

Bit	Description
31:29	**B_H_FILTER_P0BC5: Phase 0B coefficient 5 (typically 3)**
28:24	**B_H_FILTER_P0BC4: Phase 0B coefficient 4 (typically -15)**
23:16	**B_H_FILTER_P0BC3: Phase 0B coefficient 3 (typically 102)**
15:8	**B_H_FILTER_P0BC2: Phase 0B coefficient 2 (typically 47)**
7:3	**B_H_FILTER_P0BC1: Phase 0B coefficient 1 (typically -11)**
2:0	**B_H_FILTER_P0BC0: Phase 0B coefficient 0 (typically 2)**

DC_WINC_B_H_FILTER_P0C_0

Offset: 50dh
Read/Write: R/W
Reset: 0000.0000

Window B Horizontal Filter phase 0C

Bit	Description
31:29	**B_H_FILTER_P0CC5: Phase 0C coefficient 5 (typically 2)**
28:24	**B_H_FILTER_P0CC4: Phase 0C coefficient 4 (typically -13)**
23:16	**B_H_FILTER_P0CC3: Phase 0C coefficient 3 (typically 109)**
15:8	**B_H_FILTER_P0CC2: Phase 0C coefficient 2 (typically 37)**
7:3	**B_H_FILTER_P0CC1: Phase 0C coefficient 1 (typically -9)**
2:0	**B_H_FILTER_P0CC0: Phase 0C coefficient 0 (typically 2)**

DC_WINC_B_H_FILTER_P0D_0

Offset: 50eh
Read/Write: R/W
Reset: 0000.0000

Window B Horizontal Filter phase 0D

Bit	Description
31:29	**B_H_FILTER_P0DC5: Phase 0D coefficient 5 (typically 2)**
28:24	**B_H_FILTER_P0DC4: Phase 0D coefficient 4 (typically -11)**
23:16	**B_H_FILTER_P0DC3: Phase 0D coefficient 3 (typically 115)**
15:8	**B_H_FILTER_P0DC2: Phase 0D coefficient 2 (typically 27)**
7:3	**B_H_FILTER_P0DC1: Phase 0D coefficient 1 (typically -7)**
2:0	**B_H_FILTER_P0DC0: Phase 0D coefficient 0 (typically 2)**

DC_WINC_B_H_FILTER_P0E_0

Offset: 50fh
Read/Write: R/W
Reset: 0000.0000

Window B Horizontal Filter phase 0E

Bit	Description
31:29	**B_H_FILTER_P0EC5: Phase 0E coefficient 5 (typically 1)**
28:24	**B_H_FILTER_P0EC4: Phase 0E coefficient 4 (typically -8)**
23:16	**B_H_FILTER_P0EC3: Phase 0E coefficient 3 (typically 122)**
15:8	**B_H_FILTER_P0EC2: Phase 0E coefficient 2 (typically 17)**
7:3	**B_H_FILTER_P0EC1: Phase 0E coefficient 1 (typically -5)**
2:0	**B_H_FILTER_P0EC0: Phase 0E coefficient 0 (typically 1)**

DC_WINC_B_H_FILTER_P0F_0

Offset: 510h
Read/Write: R/W
Reset: 0000.0000

Window B Horizontal Filter phase 0F

Bit	Description
31:29	**B_H_FILTER_P0FC5: Phase 0F coefficient 5 (typically 1)**
28:24	**B_H_FILTER_P0FC4: Phase 0F coefficient 4 (typically -4)**
23:16	**B_H_FILTER_P0FC3: Phase 0F coefficient 3 (typically 124)**
15:8	**B_H_FILTER_P0FC2: Phase 0F coefficient 2 (typically 8)**
7:3	**B_H_FILTER_P0FC1: Phase 0F coefficient 1 (typically -2)**
2:0	**B_H_FILTER_P0FC0: Phase 0F coefficient 0 (typically 1)**

DC_WINC_B_CSC_YOF_0

Offset: 511h
Read/Write: R/W
Reset: 0000.0000

Window B CSC Y Offset
Color Space Conversion coefficients.
The CSC can be used for YUV to RGB conversion with brightness and hue/saturation control. The CSC can only be enabled for Window B controlled by (P/S)_B_CSC_ENABLE register bits. For Y color, the Y offset is applied first and saturation (clipping) is performed immediately after the Y offset is applied.

R = sat(KYRGB * sat(Y + YOF) + KUR * U + KVR * V)
G = sat(KYRGB * sat(Y + YOF) + KUG * U + KVG * V)
B = sat(KYRGB * sat(Y + YOF) + KUB * U + KVB * V)
Saturation and rounding is performed in the range of 0 to 255 for the above equations.

Typical values are
YOF = -16.000, KYRGB = 1.1644
KUR = 0.0000, KVR = -1.5960
KUG = -0.3918, KVG = -0.8130
KUB = 2.0172, KVB = 0.0000
KUR and KVB are typically 0.0000 but they may be programmed non-zero for hue rotation.

The CSC can also take RGB input, in which case YOF, KVB, KUG, KUR should be programmed to 0 and KYRGB will be forced to 0 by the hardware for generating R and B. KYRGB will not be forced to 0 for generating G. KVR, KYRGB, and KUB can be programmed to 1.0 or used as gain control for R, G, B correspondingly. Note that color value ranges from 0 to 255 for Y, R, G, B and -128 to 127 for U and V.

Bit	Description
7:0	**B_CSC_YOF: Y Offset in s.7.0 format**

DC_WINC_B_CSC_KYRGB_0

Offset: 512h
Read/Write: R/W
Reset: 0000.0000

Window B CSC Y Coefficient (gain) for RGB

Bit	Description
9:0	**B_CSC_KYRGB: Y Gain for R, G, B colors in 2.8 format**

DC_WINC_B_CSC_KUR_0

Offset: 513h
Read/Write: R/W
Reset: 0000.0000

Window B CSC U coefficient for R

Bit	Description
10:0	**B_CSC_KUR: U coefficients for R in s.2.8 format**

DC_WINC_B_CSC_KVR_0

Offset: 514h
Read/Write: R/W
Reset: 0000.0000

Window B CSC V coefficient for R.

Bit	Description
10:0	**B_CSC_KVR: V coefficients for R in s.2.8 format**

DC_WINC_B_CSC_KUG_0

Offset: 515h
Read/Write: R/W
Reset: 0000.0000

Window B CSC U coefficient for G

Bit	Description
9:0	**B_CSC_KUG: U coefficients for G in s.1.8 format**

DC_WINC_B_CSC_KVG_0

Offset: 516h
Read/Write: R/W
Reset: 0000.0000

Window B CSC V coefficient for G

Bit	Description
9:0	**B_CSC_KVG: V coefficients for G in s.1.8 format**

DC_WINC_B_CSC_KUB_0

Offset: 517h
Read/Write: R/W
Reset: 0000.0000

Window B CSC U coefficient for B

Bit	Description
10:0	**B_CSC_KUB: U coefficients for B in s.2.8 format**

DC_WINC_B_CSC_KVB_0

Offset: 518h
Read/Write: R/W
Reset: 0000.0000

Window B CSC V coefficient for B

Bit	Description
10:0	**B_CSC_KVB: V coefficients for B in s.2.8 format**

DC_WINC_B_V_FILTER_P00_0

Offset: 519h
Read/Write: R/W
Reset: 0000.0000

Window B Vertical Filter phase 00

Vertical scaling filter coefficients

Vertical scaling filter is a 2-tap filter with 4-bit positional phase. Coefficients 0 and 1 are 8-bit unsigned value ranging from 0 to 128. Coefficient 0 is the multiplier for the earlier pixel (P0) in the group of 2-pixel and coefficient 1 is the multiplier for the later pixel (P1) in the group. The output pixel positional phase is defined as centered in P0 if the positional phase is 0 or proportionally in between P0 and P1 if the positional phase is larger than 0. Sum of all coefficients for each phase should be 128 typically therefore coefficient 1 can be calculated from (1 - coefficient 0) and therefore only coefficient 0 is programmed. For each vertical positional phase, the filter coefficient requires 8 reg bits. Note that color value ranges from 0 to 255 for Y, R, G, B and -128 to 127 for U and V. The same V filter coefficients are used for both primary and secondary displays and they can be written as either primary or secondary display V filter coefficients.

Bit	Description
7:0	**B_V_FILTER_P00C0: Phase 00 coefficient 0 (typically 128)**

DC_WINC_B_V_FILTER_P01_0

Offset: 51ah
Read/Write: R/W
Reset: 0000.0000

Window B Vertical Filter phase 01

Bit	Description
7:0	**B_V_FILTER_P01C0: Phase 01 coefficient 0 (typically 120)**

DC_WINC_B_V_FILTER_P02_0

Offset: 51bh
Read/Write: R/W
Reset: 0000.0000

Window B Vertical Filter phase 02

Bit	Description
7:0	**B_V_FILTER_P02C0: Phase 02 coefficient 0 (typically 112)**

DC_WINC_B_V_FILTER_P03_0

Offset: 51ch
Read/Write: R/W
Reset: 0000.0000

Window B Vertical Filter phase 03

Bit	Description
7:0	**B_V_FILTER_P03C0: Phase 03 coefficient 0 (typically 104)**

DC_WINC_B_V_FILTER_P04_0

Offset: 51dh
Read/Write: R/W
Reset: 0000.0000

Window B Vertical Filter phase 04

Bit	Description
7:0	**B_V_FILTER_P04C0: Phase 04 coefficient 0 (typically 96)**

DC_WINC_B_V_FILTER_P05_0

Offset: 51eh
Read/Write: R/W
Reset: 0000.0000

Window B Vertical Filter phase 05

Bit	Description
7:0	**B_V_FILTER_P05C0: Phase 05 coefficient 0 (typically 88)**

DC_WINC_B_V_FILTER_P06_0

Offset: 51fh
Read/Write: R/W
Reset: 0000.0000

Window B Vertical Filter phase 06

Bit	Description
7:0	**B_V_FILTER_P06C0: Phase 06 coefficient 0 (typically 80)**

DC_WINC_B_V_FILTER_P07_0

Offset: 520h
Read/Write: R/W
Reset: 0000.0000

Window B Vertical Filter phase 07

Bit	Description
7:0	**B_V_FILTER_P07C0: Phase 07 coefficient 0 (typically 72)**

DC_WINC_B_V_FILTER_P08_0

Offset: 521h
Read/Write: R/W
Reset: 0000.0000

Window B Vertical Filter phase 08

Bit	Description
7:0	**B_V_FILTER_P08C0: Phase 08 coefficient 0 (typically 64)**

DC_WINC_B_V_FILTER_P09_0

Offset: 522h
Read/Write: R/W
Reset: 0000.0000

Window B Vertical Filter phase 09

Bit	Description
7:0	**B_V_FILTER_P09C0: Phase 09 coefficient 0 (typically 56)**

DC_WINC_B_V_FILTER_P0A_0

Offset: 523h
Read/Write: R/W
Reset: 0000.0000

Window B Vertical Filter phase 0A

Bit	Description
7:0	**B_V_FILTER_P0AC0: Phase 0A coefficient 0 (typically 48)**

DC_WINC_B_V_FILTER_P0B_0

Offset: 524h
Read/Write: R/W
Reset: 0000.0000

Window B Vertical Filter phase 0B

Bit	Description
7:0	**B_V_FILTER_P0BC0: Phase 0B coefficient 0 (typically 40)**

DC_WINC_B_V_FILTER_P0C_0

Offset: 525h
Read/Write: R/W
Reset: 0000.0000

Window B Vertical Filter phase 0C

Bit	Description
7:0	**B_V_FILTER_P0CC0: Phase 0C coefficient 0 (typically 32)**

DC_WINC_B_V_FILTER_P0D_0

Offset: 526h
Read/Write: R/W
Reset: 0000.0000

Window B Vertical Filter phase 0D

Bit	Description
7:0	**B_V_FILTER_P0DC0: Phase 0D coefficient 0 (typically 24)**

DC_WINC_B_V_FILTER_P0E_0

Offset: 527h
Read/Write: R/W
Reset: 0000.0000

Window B Vertical Filter phase 0E

Bit	Description
7:0	**B_V_FILTER_P0EC0: Phase 0E coefficient 0 (typically 16)**

DC_WINC_B_V_FILTER_P0F_0

Offset: 528h
Read/Write: R/W
Reset: 0000.0000

Window B Vertical Filter phase 0F

Bit	Description
7:0	**B_V_FILTER_P0FC0: Phase 0F coefficient 0 (typically 8)**

DC_WINC_B_COLOR_PALETTE_0

Offset: 400h..4ffh
Read/Write: WO
Reset: 0000.0000

Window B Color Palette

This is an array of 256 identical register entries; the register fields below apply to each entry.

Color palette

This is used for palettized data format (color depth of 8-bpp or less) or for gamma correction for non-palettized data formats (color depth of more than 8-bpp). Each window has its own color palette which consists of three 256x8 register file which can be written by host and indexed (read) by the window. For palettized data format less than 8-bpp the pixel data is aligned to least significant bits of the palette index (address) and the remaining upper bits are filled with the corresponding bits of the Palette Color Extension. For example, for 4-bpp mode, the pixel data occupies bits 3-0 of the palette index and bits 7-4 of the palette index are set to bits 7-4 of the Palette Color Extension.

Note that host read is assumed to be not needed - software can cache the color palette in system memory. The same color palettes are used for both primary and secondary displays and they can be written as either primary or secondary display color palette.

Bit	Description
23:16	**B_COLOR_PALETTE_B: Blue Color Palette**
15:8	**B_COLOR_PALETTE_G: Green Color Palette**
7:0	**B_COLOR_PALETTE_R: Red Color Palette**

DC_WINC_B_DV_CONTROL_0

Offset: 500h
Read/Write: R/W
Reset: 0000.0000

Window B Digital Vibrance Control

Digital Vibrance (DV) control

If enabled, Digital Vibrance is applied after H and V scaling and after color palette or color space conversion logic but before color keying multiplexer and before cursor multiplexer.

After DV, new R = R + (2R - G - B) * FR, where FR is fraction from 0 to 7/8
After DV, new G = G + (2G - R - B) * FG, where FG is fraction from 0 to 7/8
After DV, new B = B + (2B - R - G) * FB, where FB is fraction from 0 to 7/8

Bit	Description
18:16	**B_DV_CONTROL_B: Digital Vibrance control for B**
10:8	**B_DV_CONTROL_G: Digital Vibrance control for G**
2:0	**B_DV_CONTROL_R: Digital Vibrance control for R**

DC_WINC_B_H_FILTER_P00_0

Offset: 501h
Read/Write: R/W
Reset: 0000.0000

Window B Horizontal Filter phase 00

Horizontal scaling filter coefficients.

Horizontal scaling filter is a 6-tap filter with 4-bit positional phase.
Coefficients 0 and 5 are 3-bit signed value ranging from -4 to 3.
Coefficients 1 and 4 are 5-bit signed value ranging from -16 to 15.
Coefficients 2 and 3 are 8-bit unsigned value ranging from 0 to 128.
Coefficient 0 is the multiplier for the earliest pixel (P0) in the group of 6-pixel and coefficient 5 is the multiplier for the latest pixel (P5) in the group.

The output pixel positional phase is defined as centered in P2 if the positional phase is 0 or proportionally in between P2 and P3 if the positional phase is larger than 0. Sum of all coefficients for each phase should be 128 typically and software should never program the the sum of all coefficients for a phase to be more than 128.

For each horizontal positional phase, the 6 filter coefficients requires 32 reg bits. Note that color value ranges from 0 to 255 for Y, R, G, B and -128 to 127 for U and V. The same H filter coefficients are used for both primary and secondary displays and they can be written as either primary or secondary display H filter coefficients.

Bit	Description
31:29	**B_H_FILTER_P00C5: Phase 00 coefficient 5 (typically 0)**
28:24	**B_H_FILTER_P00C4: Phase 00 coefficient 4 (typically 0)**
23:16	**B_H_FILTER_P00C3: Phase 00 coefficient 3 (typically 0)**
15:8	**B_H_FILTER_P00C2: Phase 00 coefficient 2 (typically 128)**
7:3	**B_H_FILTER_P00C1: Phase 00 coefficient 1 (typically 0)**
2:0	**B_H_FILTER_P00C0: Phase 00 coefficient 0 (typically 0)**

DC_WINC_B_H_FILTER_P01_0

Offset: 502h
Read/Write: R/W
Reset: 0000.0000

Window B Horizontal Filter phase 01

Bit	Description
31:29	**B_H_FILTER_P01C5: Phase 01 coefficient 5 (typically 1)**
28:24	**B_H_FILTER_P01C4: Phase 01 coefficient 4 (typically -2)**
23:16	**B_H_FILTER_P01C3: Phase 01 coefficient 3 (typically 8)**
15:8	**B_H_FILTER_P01C2: Phase 01 coefficient 2 (typically 124)**
7:3	**B_H_FILTER_P01C1: Phase 01 coefficient 1 (typically -4)**
2:0	**B_H_FILTER_P01C0: Phase 01 coefficient 0 (typically 1)**

DC_WINC_B_H_FILTER_P02_0

Offset: 503h
Read/Write: R/W
Reset: 0000.0000

Window B Horizontal Filter phase 02

Bit	Description
31:29	**B_H_FILTER_P02C5: Phase 02 coefficient 5 (typically 1)**
28:24	**B_H_FILTER_P02C4: Phase 02 coefficient 4 (typically -5)**
23:16	**B_H_FILTER_P02C3: Phase 02 coefficient 3 (typically 17)**
15:8	**B_H_FILTER_P02C2: Phase 02 coefficient 2 (typically 122)**
7:3	**B_H_FILTER_P02C1: Phase 02 coefficient 1 (typically -8)**
2:0	**B_H_FILTER_P02C0: Phase 02 coefficient 0 (typically 1)**

DC_WINC_B_H_FILTER_P03_0

Offset: 504h
Read/Write: R/W
Reset: 0000.0000

Window B Horizontal Filter phase 03

Bit	Description
31:29	**B_H_FILTER_P03C5: Phase 03 coefficient 5 (typically 2)**
28:24	**B_H_FILTER_P03C4: Phase 03 coefficient 4 (typically -7)**
23:16	**B_H_FILTER_P03C3: Phase 03 coefficient 3 (typically 27)**
15:8	**B_H_FILTER_P03C2: Phase 03 coefficient 2 (typically 115)**
7:3	**B_H_FILTER_P03C1: Phase 03 coefficient 1 (typically -11)**
2:0	**B_H_FILTER_P03C0: Phase 03 coefficient 0 (typically 2)**

DC_WINC_B_H_FILTER_P04_0

Offset: 505h
Read/Write: R/W
Reset: 0000.0000

Window B Horizontal Filter phase 04

Bit	Description
31:29	**B_H_FILTER_P04C5: Phase 04 coefficient 5 (typically 2)**
28:24	**B_H_FILTER_P04C4: Phase 04 coefficient 4 (typically -9)**
23:16	**B_H_FILTER_P04C3: Phase 04 coefficient 3 (typically 37)**
15:8	**B_H_FILTER_P04C2: Phase 04 coefficient 2 (typically 109)**
7:3	**B_H_FILTER_P04C1: Phase 04 coefficient 1 (typically -13)**
2:0	**B_H_FILTER_P04C0: Phase 04 coefficient 0 (typically 2)**

DC_WINC_B_H_FILTER_P05_0

Offset: 506h
Read/Write: R/W
Reset: 0000.0000

Window B Horizontal Filter phase 05

Bit	Description
31:29	**B_H_FILTER_P05C5: Phase 05 coefficient 5 (typically 2)**
28:24	**B_H_FILTER_P05C4: Phase 05 coefficient 4 (typically -11)**
23:16	**B_H_FILTER_P05C3: Phase 05 coefficient 3 (typically 47)**
15:8	**B_H_FILTER_P05C2: Phase 05 coefficient 2 (typically 102)**
7:3	**B_H_FILTER_P05C1: Phase 05 coefficient 1 (typically -15)**
2:0	**B_H_FILTER_P05C0: Phase 05 coefficient 0 (typically 3)**

DC_WINC_B_H_FILTER_P06_0

Offset: 507h
Read/Write: R/W
Reset: 0000.0000

Window B Horizontal Filter phase 06

Bit	Description
31:29	**B_H_FILTER_P06C5: Phase 06 coefficient 5 (typically 3)**
28:24	**B_H_FILTER_P06C4: Phase 06 coefficient 4 (typically -13)**
23:16	**B_H_FILTER_P06C3: Phase 06 coefficient 3 (typically 56)**
15:8	**B_H_FILTER_P06C2: Phase 06 coefficient 2 (typically 94)**
7:3	**B_H_FILTER_P06C1: Phase 06 coefficient 1 (typically -15)**
2:0	**B_H_FILTER_P06C0: Phase 06 coefficient 0 (typically 3)**

DC_WINC_B_H_FILTER_P07_0

Offset: 508h
Read/Write: R/W
Reset: 0000.0000

Window B Horizontal Filter phase 07

Bit	Description
31:29	**B_H_FILTER_P07C5: Phase 07 coefficient 5 (typically 3)**
28:24	**B_H_FILTER_P07C4: Phase 07 coefficient 4 (typically -14)**
23:16	**B_H_FILTER_P07C3: Phase 07 coefficient 3 (typically 67)**
15:8	**B_H_FILTER_P07C2: Phase 07 coefficient 2 (typically 85)**
7:3	**B_H_FILTER_P07C1: Phase 07 coefficient 1 (typically -16)**
2:0	**B_H_FILTER_P07C0: Phase 07 coefficient 0 (typically 3)**

DC_WINC_B_H_FILTER_P08_0

Offset: 509h
Read/Write: R/W
Reset: 0000.0000

Window B Horizontal Filter phase 08

Bit	Description
31:29	**B_H_FILTER_P08C5: Phase 08 coefficient 5 (typically 3)**
28:24	**B_H_FILTER_P08C4: Phase 08 coefficient 4 (typically -15)**
23:16	**B_H_FILTER_P08C3: Phase 08 coefficient 3 (typically 76)**
15:8	**B_H_FILTER_P08C2: Phase 08 coefficient 2 (typically 76)**
7:3	**B_H_FILTER_P08C1: Phase 08 coefficient 1 (typically -15)**
2:0	**B_H_FILTER_P08C0: Phase 08 coefficient 0 (typically 3)**

DC_WINC_B_H_FILTER_P09_0

Offset: 50ah
Read/Write: R/W
Reset: 0000.0000

Window B Horizontal Filter phase 09

Bit	Description
31:29	**B_H_FILTER_P09C5: Phase 09 coefficient 5 (typically 3)**
28:24	**B_H_FILTER_P09C4: Phase 09 coefficient 4 (typically -16)**
23:16	**B_H_FILTER_P09C3: Phase 09 coefficient 3 (typically 85)**
15:8	**B_H_FILTER_P09C2: Phase 09 coefficient 2 (typically 67)**
7:3	**B_H_FILTER_P09C1: Phase 09 coefficient 1 (typically -14)**
2:0	**B_H_FILTER_P09C0: Phase 09 coefficient 0 (typically 3)**

DC_WINC_B_H_FILTER_P0A_0

Offset: 50bh
Read/Write: R/W
Reset: 0000.0000

Window B Horizontal Filter phase 0A

Bit	Description
31:29	**B_H_FILTER_P0AC5: Phase 0A coefficient 5 (typically 3)**
28:24	**B_H_FILTER_P0AC4: Phase 0A coefficient 4 (typically -15)**
23:16	**B_H_FILTER_P0AC3: Phase 0A coefficient 3 (typically 94)**
15:8	**B_H_FILTER_P0AC2: Phase 0A coefficient 2 (typically 56)**
7:3	**B_H_FILTER_P0AC1: Phase 0A coefficient 1 (typically -13)**
2:0	**B_H_FILTER_P0AC0: Phase 0A coefficient 0 (typically 3)**

DC_WINC_B_H_FILTER_P0B_0

Offset: 50ch
Read/Write: R/W
Reset: 0000.0000

Window B Horizontal Filter phase 0B

Bit	Description
31:29	**B_H_FILTER_P0BC5: Phase 0B coefficient 5 (typically 3)**
28:24	**B_H_FILTER_P0BC4: Phase 0B coefficient 4 (typically -15)**
23:16	**B_H_FILTER_P0BC3: Phase 0B coefficient 3 (typically 102)**
15:8	**B_H_FILTER_P0BC2: Phase 0B coefficient 2 (typically 47)**
7:3	**B_H_FILTER_P0BC1: Phase 0B coefficient 1 (typically -11)**
2:0	**B_H_FILTER_P0BC0: Phase 0B coefficient 0 (typically 2)**

DC_WINC_B_H_FILTER_P0C_0

Offset: 50dh
Read/Write: R/W
Reset: 0000.0000

Window B Horizontal Filter phase 0C

Bit	Description
31:29	**B_H_FILTER_P0CC5: Phase 0C coefficient 5 (typically 2)**
28:24	**B_H_FILTER_P0CC4: Phase 0C coefficient 4 (typically -13)**
23:16	**B_H_FILTER_P0CC3: Phase 0C coefficient 3 (typically 109)**
15:8	**B_H_FILTER_P0CC2: Phase 0C coefficient 2 (typically 37)**
7:3	**B_H_FILTER_P0CC1: Phase 0C coefficient 1 (typically -9)**
2:0	**B_H_FILTER_P0CC0: Phase 0C coefficient 0 (typically 2)**

DC_WINC_B_H_FILTER_P0D_0

Offset: 50eh
Read/Write: R/W
Reset: 0000.0000

Window B Horizontal Filter phase 0D

Bit	Description
31:29	**B_H_FILTER_P0DC5: Phase 0D coefficient 5 (typically 2)**
28:24	**B_H_FILTER_P0DC4: Phase 0D coefficient 4 (typically -11)**
23:16	**B_H_FILTER_P0DC3: Phase 0D coefficient 3 (typically 115)**
15:8	**B_H_FILTER_P0DC2: Phase 0D coefficient 2 (typically 27)**
7:3	**B_H_FILTER_P0DC1: Phase 0D coefficient 1 (typically -7)**
2:0	**B_H_FILTER_P0DC0: Phase 0D coefficient 0 (typically 2)**

DC_WINC_B_H_FILTER_P0E_0

Offset: 50fh
Read/Write: R/W
Reset: 0000.0000

Window B Horizontal Filter phase 0E

Bit	Description
31:29	**B_H_FILTER_P0EC5: Phase 0E coefficient 5 (typically 1)**
28:24	**B_H_FILTER_P0EC4: Phase 0E coefficient 4 (typically -8)**
23:16	**B_H_FILTER_P0EC3: Phase 0E coefficient 3 (typically 122)**
15:8	**B_H_FILTER_P0EC2: Phase 0E coefficient 2 (typically 17)**
7:3	**B_H_FILTER_P0EC1: Phase 0E coefficient 1 (typically -5)**
2:0	**B_H_FILTER_P0EC0: Phase 0E coefficient 0 (typically 1)**

DC_WINC_B_H_FILTER_P0F_0

Offset: 510h
Read/Write: R/W
Reset: 0000.0000

Window B Horizontal Filter phase 0F

Bit	Description
31:29	**B_H_FILTER_P0FC5: Phase 0F coefficient 5 (typically 1)**
28:24	**B_H_FILTER_P0FC4: Phase 0F coefficient 4 (typically -4)**
23:16	**B_H_FILTER_P0FC3: Phase 0F coefficient 3 (typically 124)**
15:8	**B_H_FILTER_P0FC2: Phase 0F coefficient 2 (typically 8)**
7:3	**B_H_FILTER_P0FC1: Phase 0F coefficient 1 (typically -2)**
2:0	**B_H_FILTER_P0FC0: Phase 0F coefficient 0 (typically 1)**

DC_WINC_B_CSC_YOF_0

Offset: 511h
Read/Write: R/W
Reset: 0000.0000

Window B CSC Y Offset

Color Space Conversion coefficients.
The CSC can be used for YUV to RGB conversion with brightness and hue/saturation control. The CSC can only be enabled for Window B controlled by (P/S)_B_CSC_ENABLE register bits. For Y color, the Y offset is applied first and saturation (clipping) is performed immediately after the Y offset is applied.

R = sat(KYRGB * sat(Y + YOF) + KUR * U + KVR * V)
G = sat(KYRGB * sat(Y + YOF) + KUG * U + KVG * V)
B = sat(KYRGB * sat(Y + YOF) + KUB * U + KVB * V)
Saturation and rounding is performed in the range of 0 to 255 for the above equations.

Typical values are:
YOF = -16.000, KYRGB = 1.1644
KUR = 0.0000, KVR = -1.5960
KUG = -0.3918, KVG = -0.8130
KUB = 2.0172, KVB = 0.0000
KUR and KVB are typically 0.0000 but they may be programmed non-zero for hue rotation.

The CSC can also take RGB input, in which case YOF, KVB, KUG, KUR should be programmed to 0 and KYRGB will be forced to 0 by the hardware for generating R and B. KYRGB will not be forced to 0 for generating G. KVR, KYRGB, and KUB can be programmed to 1.0 or used as gain control for R, G, B, respectively.

Note that color value ranges from 0 to 255 for Y, R, G, B and -128 to 127 for U and V.

Bit	Description
7:0	**B_CSC_YOF: Y Offset in s.7.0 format**

DC_WINC_B_CSC_KYRGB_0

Offset: 512h
Read/Write: R/W
Reset: 0000.0000

Window B CSC Y Coefficient (gain) for RGB

Bit	Description
9:0	**B_CSC_KYRGB: Y Gain for R, G, B colors in 2.8 format**

DC_WINC_B_CSC_KUR_0

Offset: 513h
Read/Write: R/W
Reset: 0000.0000

Window B CSC U coefficient for R

Bit	Description
10:0	**B_CSC_KUR: U coefficients for R in s.2.8 format**

DC_WINC_B_CSC_KVR_

Offset: 514h
Read/Write: R/W
Reset: 0000.0000

Window B CSC V coefficient for R

Bit	Description
10:0	**B_CSC_KVR: V coefficients for R in s.2.8 format**

DC_WINC_B_CSC_KUG_

Offset: 515h
Read/Write: R/W
Reset: 0000.0000

Window B CSC U coefficient for G

Bit	Description
9:0	**B_CSC_KUG: U coefficients for G in s.1.8 format**

DC_WINC_B_CSC_KVG_0

Offset: 516h
Read/Write: R/W
Reset: 0000.0000

Window B CSC V coefficient for G

Bit	Description
9:0	**B_CSC_KVG: V coefficients for G in s.1.8 format**

DC_WINC_B_CSC_KUB_0

Offset: 517h
Read/Write: R/W
Reset: 0000.0000

Window B CSC U coefficient for B

Bit	Description
10:0	**B_CSC_KUB: U coefficients for B in s.2.8 format**

DC_WINC_B_CSC_KVB_0

Offset: 518h
Read/Write: R/W
Reset: 0000.0000

Window B CSC V coefficient for B

Bit	Description
10:0	**B_CSC_KVB: V coefficients for B in s.2.8 format**

DC_WINC_B_V_FILTER_P00_0

Offset: 519h
Read/Write: R/W
Reset: 0000.0000

Window B Vertical Filter phase 00

Vertical scaling filter coefficients

Vertical scaling filter is a 2-tap filter with 4-bit positional phase. Coefficients 0 and 1 are 8-bit unsigned value ranging from 0 to 128. Coefficient 0 is the multiplier for the earlier pixel (P0) in the group of 2-pixel and coefficient 1 is the multiplier for the later pixel (P1) in the group. The output pixel positional phase is defined as centered in P0 if the positional phase is 0 or proportionally in between P0 and P1 if the positional phase is larger than 0.

Sum of all coefficients for each phase should be 128 typically; therefore, coefficient 1 can be calculated from (1 - coefficient 0) and therefore only coefficient 0 is programmed. For each vertical positional phase, the filter coefficient requires 8 reg bits. Note that color value ranges from 0 to 255 for Y, R, G, B and -128 to 127 for U and V. The same V filter coefficients are used for both primary and secondary displays and they can be written as either primary or secondary display V filter coefficients.

Bit	Description
7:0	**B_V_FILTER_P00C0: Phase 00 coefficient 0 (typically 128)**

DC_WINC_B_V_FILTER_P01_0

Offset: 51ah
Read/Write: R/W
Reset: 0000.0000

Window B Vertical Filter phase 01

Bit	Description
7:0	**B_V_FILTER_P01C0: Phase 01 coefficient 0 (typically 120)**

DC_WINC_B_V_FILTER_P02_0

Offset: 51bh
Read/Write: R/W
Reset: 0000.0000

Window B Vertical Filter phase 02

Bit	Description
7:0	**B_V_FILTER_P02C0: Phase 02 coefficient 0 (typically 112)**

DC_WINC_B_V_FILTER_P03_0

Offset: 51ch
Read/Write: R/W
Reset: 0000.0000

Window B Vertical Filter phase 03

Bit	Description
7:0	**B_V_FILTER_P03C0: Phase 03 coefficient 0 (typically 104)**

DC_WINC_B_V_FILTER_P04_0

Offset: 51dh
Read/Write: R/W
Reset: 0000.0000

Window B Vertical Filter phase 04

Bit	Description
7:0	**B_V_FILTER_P04C0: Phase 04 coefficient 0 (typically 96)**

DC_WINC_B_V_FILTER_P05_0

Offset: 51eh
Read/Write: R/W
Reset: 0000.0000

Window B Vertical Filter phase 05

Bit	Description
7:0	**B_V_FILTER_P05C0: Phase 05 coefficient 0 (typically 88)**

DC_WINC_B_V_FILTER_P06_0

Offset: 51fh
Read/Write: R/W
Reset: 0000.0000

Window B Vertical Filter phase 06

Bit	Description
7:0	**B_V_FILTER_P06C0: Phase 06 coefficient 0 (typically 80)**

DC_WINC_B_V_FILTER_P07_0

Offset: 520h
Read/Write: R/W
Reset: 0000.0000

Window B Vertical Filter phase 07

Bit	Description
7:0	**B_V_FILTER_P07C0: Phase 07 coefficient 0 (typically 72)**

DC_WINC_B_V_FILTER_P08_0

Offset: 521h
Read/Write: R/W
Reset: 0000.0000

Window B Vertical Filter phase 08

Bit	Description
7:0	**B_V_FILTER_P08C0: Phase 08 coefficient 0 (typically 64)**

DC_WINC_B_V_FILTER_P09_0

Offset: 522h
Read/Write: R/W
Reset: 0000.0000

Window B Vertical Filter phase 09

Bit	Description
7:0	**B_V_FILTER_P09C0: Phase 09 coefficient 0 (typically 56)**

DC_WINC_B_V_FILTER_P0A_0

Offset: 523h
Read/Write: R/W
Reset: 0000.0000

Window B Vertical Filter phase 0A

Bit	Description
7:0	**B_V_FILTER_P0AC0: Phase 0A coefficient 0 (typically 48)**

DC_WINC_B_V_FILTER_P0B_0

Offset: 524h
Read/Write: R/W
Reset: 0000.0000

Window B Vertical Filter phase 0B

Bit	Description
7:0	**B_V_FILTER_P0BC0: Phase 0B coefficient 0 (typically 40)**

DC_WINC_B_V_FILTER_P0C_0

Offset: 525h
Read/Write: R/W
Reset: 0000.0000

Window B Vertical Filter phase 0C

Bit	Description
7:0	**B_V_FILTER_P0CC0: Phase 0C coefficient 0 (typically 32)**

DC_WINC_B_V_FILTER_P0D_0

Offset: 526h
Read/Write: R/W
Reset: 0000.0000

Window B Vertical Filter phase 0D

Bit	Description
7:0	**B_V_FILTER_P0DC0: Phase 0D coefficient 0 (typically 24)**

DC_WINC_B_V_FILTER_P0E_0

Offset: 527h
Read/Write: R/W
Reset: 0000.0000

Window B Vertical Filter phase 0E

Bit	Description
7:0	**B_V_FILTER_P0EC0: Phase 0E coefficient 0 (typically 16)**

DC_WINC_B_V_FILTER_P0F_0

Offset: 528h
Read/Write: R/W
Reset: 0000.0000

Window B Vertical Filter phase 0F

Bit	Description
7:0	**B_V_FILTER_P0FC0: Phase 0F coefficient 0 (typically 8)**

DC_BUF_START_ADDR_

Offset: 300h
Read/Write: R/W
Reset: 0000.0000

Window B Buffer 0 Start Address

In the case where host (software) controls the window buffer, the buffer start addresses are used as byte address to indicate the start address of the buffer. The least significant bit of the start address is not used for 16-bpp non-planar pixel format and the least two significant bits of the start address are not used for 32-bpp non-planar pixel format.

In the case where window buffer is not controlled by host (software), the buffer 0 start addresses are used as buffer 0 start address sent by the controlling module and buffer 1 start addresses are used as buffer strides. For planar YUV, there are only two buffer strides, one for luma and one for chroma, so Buffer 1 Start Address is used for luma buffer stride and Buffer 1 Start Address for U is used for chroma buffer stride and Buffer 1 Start Address for V is not used. Buffer 0 start addresses must include necessary buffer offset that is due to H/V flip or cropping.

Note that in this case, the controlling module will send buffer index and display module will calculate the actual start addresses of the buffer based on buffer 0 start address, the buffer strides and the buffer index. In the case where window buffer is not controlled by host (software) then a frame may be stored in multiple buffers. In this case, the buffers must be contiguous in the memory because display will use the same luma or chroma line strides for all lines in the frame. Also buffer wraparound must not occur in the middle of the displayed part of the frame. The controlling module will send frame start and frame end indicators (flags) to display module to indicate the beginning and end of frame. Buffer start address is latched when frame start flag is active but the actual buffer start address is not switched until frame end flag is active. In the case where one buffer correspond to one frame then frame start and frame end flag are active every time a buffer index is sent.

Bit	Description
25:0	**START_ADDR: Window B Buffer 0 Start Address** This is a byte address. The LSB is not used for 16-bpp non-planar pixel format and the last 2 LSB are not used for 32-bpp non-planar pixel format. For YUV planar pixel format, this specifies start address for the Y plane.

DC_BUF_START_ADDR_U_0

Offset: 301h
Read/Write: R/W
Reset: 0000.0000

Window B Buffer 0 Start Address for U plane

Bit	Description
25:0	**START_ADDR_U: Window B Buffer 0 Start Address for U plane** This is a byte address.

DC_BUF_START_ADDR_V_0

Offset: 302h
Read/Write: R/W
Reset: 0000.0000

Window B Buffer 0 Start Address for V plane

Bit	Description
25:0	**START_ADDR_V: Window B Buffer 0 Start Address for V plane** This is a byte address.

DC_WINC_C_COLOR_PALETTE_0

Offset: 400h..4ffh
Read/Write: WO
Reset: 0000.0000

Window B Color Palette

This is an array of 256 identical register entries; the register fields below apply to each entry.

Color palette

This is used for palettized data format (color depth of 8-bpp or less) or for gamma correction for non-palettized data formats (color depth of more than 8-bpp). Each window has its own color palette which consists of three 256x8 register file which can be written by host and indexed (read) by the window. For palettized data format less than 8-bpp the pixel data is aligned to least significant bits of the palette index (address) and the remaining upper bits are filled with the corresponding bits of the Palette Color Extension. For example, for 4-bpp mode, the pixel data occupies bits 3-0 of the palette index and bits 7-4 of the palette index are set to bits 7-4 of the Palette Color Extension.

Note that host read is assumed to be not needed - software can cache the color palette in system memory. The same color palettes are used for both primary and secondary displays and they can be written as either primary or secondary display color palette.

Bit	Description
23:16	**C_COLOR_PALETTE_B: Blue Color Palette**
15:8	**C_COLOR_PALETTE_G: Green Color Palette**
7:0	**C_COLOR_PALETTE_R: Red Color Palette**

DC_WINC_C_DV_CONTROL_0

Offset: 500h
Read/Write: R/W
Reset: 0000.0000

Window B Digital Vibrance (DV) Control

If enabled, Digital Vibrance is applied after H and V scaling and after color palette or color space conversion logic but before color keying multiplexer and before cursor multiplexer.

After DV, new R = R + (2R - G - B) * FR, where FR is fraction from 0 to 7/8
After DV, new G = G + (2G - R - B) * FG, where FG is fraction from 0 to 7/8
After DV, new B = B + (2B - R - G) * FB, where FB is fraction from 0 to 7/8

Bit	Description
18:16	**C_DV_CONTROL_B: Digital Vibrance control for B**
10:8	**C_DV_CONTROL_G: Digital Vibrance control for G**
2:0	**C_DV_CONTROL_R: Digital Vibrance control for R**

DC_WINC_C_H_FILTER_P00_0

Offset: 501h
Read/Write: R/W
Reset: 0000.0000

Window B Horizontal Filter phase 00

Horizontal scaling filter coefficients.

Horizontal scaling filter is a 6-tap filter with 4-bit positional phase.
Coefficients 0 and 5 are 3-bit signed value ranging from -4 to 3.
Coefficients 1 and 4 are 5-bit signed value ranging from -16 to 15.
Coefficients 2 and 3 are 8-bit unsigned value ranging from 0 to 128.
Coefficient 0 is the multiplier for the earliest pixel (P0) in the group of 6-pixel and coefficient 5 is the multiplier for the latest pixel (P5) in the group.

The output pixel positional phase is defined as centered in P2 if the positional phase is 0 or proportionally in between P2 and P3 if the positional phase is larger than 0. Sum of all coefficients for each phase should be 128 typically and software should never program the the sum of all coefficients for a phase to be more than 128. For each horizontal positional phase, the 6 filter coefficients requires 32 reg bits. Note that color value ranges from 0 to 255 for Y, R, G, B and -128 to 127 for U and V. The same H filter coefficients are used for both primary and secondary displays and they can be written as either primary or secondary display H filter coefficients.

Bit	Description
31:29	**C_H_FILTER_P00C5: Phase 00 coefficient 5 (typically 0)**
28:24	**C_H_FILTER_P00C4: Phase 00 coefficient 4 (typically 0)**
23:16	**C_H_FILTER_P00C3: Phase 00 coefficient 3 (typically 0)**
15:8	**C_H_FILTER_P00C2: Phase 00 coefficient 2 (typically 128)**
7:3	**C_H_FILTER_P00C1: Phase 00 coefficient 1 (typically 0)**
2:0	**C_H_FILTER_P00C0: Phase 00 coefficient 0 (typically 0)**

DC_WINC_C_H_FILTER_P01_0

Offset: 502h
Read/Write: R/W
Reset: 0000.0000

Window B Horizontal Filter phase 01

Bit	Description
31:29	**C_H_FILTER_P01C5: Phase 01 coefficient 5 (typically 1)**
28:24	**C_H_FILTER_P01C4: Phase 01 coefficient 4 (typically -2)**
23:16	**C_H_FILTER_P01C3: Phase 01 coefficient 3 (typically 8)**
15:8	**C_H_FILTER_P01C2: Phase 01 coefficient 2 (typically 124)**
7:3	**C_H_FILTER_P01C1: Phase 01 coefficient 1 (typically -4)**
2:0	**C_H_FILTER_P01C0: Phase 01 coefficient 0 (typically 1)**

DC_WINC_C_H_FILTER_P02_0

Offset: 503h
Read/Write: R/W
Reset: 0000.0000

Window B Horizontal Filter phase 02

Bit	Description
31:29	**C_H_FILTER_P02C5: Phase 02 coefficient 5 (typically 1)**
28:24	**C_H_FILTER_P02C4: Phase 02 coefficient 4 (typically -5)**
23:16	**C_H_FILTER_P02C3: Phase 02 coefficient 3 (typically 17)**
15:8	**C_H_FILTER_P02C2: Phase 02 coefficient 2 (typically 122)**
7:3	**C_H_FILTER_P02C1: Phase 02 coefficient 1 (typically -8)**
2:0	**C_H_FILTER_P02C0: Phase 02 coefficient 0 (typically 1)**

DC_WINC_C_H_FILTER_P03_0

Offset: 504h
Read/Write: R/W
Reset: 0000.0000

Window B Horizontal Filter phase 03

Bit	Description
31:29	**C_H_FILTER_P03C5: Phase 03 coefficient 5 (typically 2)**
28:24	**C_H_FILTER_P03C4: Phase 03 coefficient 4 (typically -7)**
23:16	**C_H_FILTER_P03C3: Phase 03 coefficient 3 (typically 27)**
15:8	**C_H_FILTER_P03C2: Phase 03 coefficient 2 (typically 115)**
7:3	**C_H_FILTER_P03C1: Phase 03 coefficient 1 (typically -11)**
2:0	**C_H_FILTER_P03C0: Phase 03 coefficient 0 (typically 2)**

DC_WINC_C_H_FILTER_P04_0

Offset: 505h
Read/Write: R/W
Reset: 0000.0000

Window B Horizontal Filter phase 04

Bit	Description
31:29	**C_H_FILTER_P04C5: Phase 04 coefficient 5 (typically 2)**
28:24	**C_H_FILTER_P04C4: Phase 04 coefficient 4 (typically -9)**
23:16	**C_H_FILTER_P04C3: Phase 04 coefficient 3 (typically 37)**
15:8	**C_H_FILTER_P04C2: Phase 04 coefficient 2 (typically 109)**
7:3	**C_H_FILTER_P04C1: Phase 04 coefficient 1 (typically -13)**
2:0	**C_H_FILTER_P04C0: Phase 04 coefficient 0 (typically 2)**

DC_WINC_C_H_FILTER_P05_0

Offset: 506h
Read/Write: R/W
Reset: 0000.0000

Window B Horizontal Filter phase 05

Bit	Description
31:29	**C_H_FILTER_P05C5: Phase 05 coefficient 5 (typically 2)**
28:24	**C_H_FILTER_P05C4: Phase 05 coefficient 4 (typically -11)**
23:16	**C_H_FILTER_P05C3: Phase 05 coefficient 3 (typically 47)**
15:8	**C_H_FILTER_P05C2: Phase 05 coefficient 2 (typically 102)**
7:3	**C_H_FILTER_P05C1: Phase 05 coefficient 1 (typically -15)**
2:0	**C_H_FILTER_P05C0: Phase 05 coefficient 0 (typically 3)**

DC_WINC_C_H_FILTER_P06_0

Offset: 507h
Read/Write: R/W
Reset: 0000.0000

Window B Horizontal Filter phase 06

Bit	Description
31:29	**C_H_FILTER_P06C5: Phase 06 coefficient 5 (typically 3)**
28:24	**C_H_FILTER_P06C4: Phase 06 coefficient 4 (typically -13)**
23:16	**C_H_FILTER_P06C3: Phase 06 coefficient 3 (typically 56)**
15:8	**C_H_FILTER_P06C2: Phase 06 coefficient 2 (typically 94)**
7:3	**C_H_FILTER_P06C1: Phase 06 coefficient 1 (typically -15)**
2:0	**C_H_FILTER_P06C0: Phase 06 coefficient 0 (typically 3)**

DC_WINC_C_H_FILTER_P07_0

Offset: 508h
Read/Write: R/W
Reset: 0000.0000

Window B Horizontal Filter phase 07

Bit	Description
31:29	**C_H_FILTER_P07C5: Phase 07 coefficient 5 (typically 3)**
28:24	**C_H_FILTER_P07C4: Phase 07 coefficient 4 (typically -14)**
23:16	**C_H_FILTER_P07C3: Phase 07 coefficient 3 (typically 67)**
15:8	**C_H_FILTER_P07C2: Phase 07 coefficient 2 (typically 85)**
7:3	**C_H_FILTER_P07C1: Phase 07 coefficient 1 (typically -16)**
2:0	**C_H_FILTER_P07C0: Phase 07 coefficient 0 (typically 3)**

DC_WINC_C_H_FILTER_P08_0

Offset: 509h
Read/Write: R/W
Reset: 0000.0000

Window B Horizontal Filter phase 08

Bit	Description
31:29	**C_H_FILTER_P08C5: Phase 08 coefficient 5 (typically 3)**
28:24	**C_H_FILTER_P08C4: Phase 08 coefficient 4 (typically -15)**
23:16	**C_H_FILTER_P08C3: Phase 08 coefficient 3 (typically 76)**
15:8	**C_H_FILTER_P08C2: Phase 08 coefficient 2 (typically 76)**
7:3	**C_H_FILTER_P08C1: Phase 08 coefficient 1 (typically -15)**
2:0	**C_H_FILTER_P08C0: Phase 08 coefficient 0 (typically 3)**

DC_WINC_C_H_FILTER_P09_0

Offset: 50ah
Read/Write: R/W
Reset: 0000.0000

Window B Horizontal Filter phase 09

Bit	Description
31:29	**C_H_FILTER_P09C5: Phase 09 coefficient 5 (typically 3)**
28:24	**C_H_FILTER_P09C4: Phase 09 coefficient 4 (typically -16)**
23:16	**C_H_FILTER_P09C3: Phase 09 coefficient 3 (typically 85)**
15:8	**C_H_FILTER_P09C2: Phase 09 coefficient 2 (typically 67)**
7:3	**C_H_FILTER_P09C1: Phase 09 coefficient 1 (typically -14)**
2:0	**C_H_FILTER_P09C0: Phase 09 coefficient 0 (typically 3)**

DC_WINC_C_H_FILTER_P0A_0

Offset: 50bh
Read/Write: R/W
Reset: 0000.0000

Window B Horizontal Filter phase 0A

Bit	Description
31:29	**C_H_FILTER_P0AC5: Phase 0A coefficient 5 (typically 3)**
28:24	**C_H_FILTER_P0AC4: Phase 0A coefficient 4 (typically -15)**
23:16	**C_H_FILTER_P0AC3: Phase 0A coefficient 3 (typically 94)**
15:8	**C_H_FILTER_P0AC2: Phase 0A coefficient 2 (typically 56)**
7:3	**C_H_FILTER_P0AC1: Phase 0A coefficient 1 (typically -13)**
2:0	**C_H_FILTER_P0AC0: Phase 0A coefficient 0 (typically 3)**

DC_WINC_C_H_FILTER_P0B_0

Offset: 50ch
Read/Write: R/W
Reset: 0000.0000

Window B Horizontal Filter phase 0B

Bit	Description
31:29	**C_H_FILTER_P0BC5: Phase 0B coefficient 5 (typically 3)**
28:24	**C_H_FILTER_P0BC4: Phase 0B coefficient 4 (typically -15)**
23:16	**C_H_FILTER_P0BC3: Phase 0B coefficient 3 (typically 102)**
15:8	**C_H_FILTER_P0BC2: Phase 0B coefficient 2 (typically 47)**
7:3	**C_H_FILTER_P0BC1: Phase 0B coefficient 1 (typically -11)**
2:0	**C_H_FILTER_P0BC0: Phase 0B coefficient 0 (typically 2)**

DC_WINC_C_H_FILTER_P0C_0

Offset: 50dh
Read/Write: R/W
Reset: 0000.0000

Window B Horizontal Filter phase 0C

Bit	Description
31:29	**C_H_FILTER_P0CC5: Phase 0C coefficient 5 (typically 2)**
28:24	**C_H_FILTER_P0CC4: Phase 0C coefficient 4 (typically -13)**
23:16	**C_H_FILTER_P0CC3: Phase 0C coefficient 3 (typically 109)**
15:8	**C_H_FILTER_P0CC2: Phase 0C coefficient 2 (typically 37)**
7:3	**C_H_FILTER_P0CC1: Phase 0C coefficient 1 (typically -9)**
2:0	**C_H_FILTER_P0CC0: Phase 0C coefficient 0 (typically 2)**

DC_WINC_C_H_FILTER_P0D_0

Offset: 50eh
Read/Write: R/W
Reset: 0000.0000

Window B Horizontal Filter phase 0D

Bit	Description
31:29	**C_H_FILTER_P0DC5: Phase 0D coefficient 5 (typically 2)**
28:24	**C_H_FILTER_P0DC4: Phase 0D coefficient 4 (typically -11)**
23:16	**C_H_FILTER_P0DC3: Phase 0D coefficient 3 (typically 115)**
15:8	**C_H_FILTER_P0DC2: Phase 0D coefficient 2 (typically 27)**
7:3	**C_H_FILTER_P0DC1: Phase 0D coefficient 1 (typically -7)**
2:0	**C_H_FILTER_P0DC0: Phase 0D coefficient 0 (typically 2)**

DC_WINC_C_H_FILTER_P0E_0

Offset: 50fh
Read/Write: R/W
Reset: 0000.0000

Window B Horizontal Filter phase 0E

Bit	Description
31:29	**C_H_FILTER_P0EC5: Phase 0E coefficient 5 (typically 1)**
28:24	**C_H_FILTER_P0EC4: Phase 0E coefficient 4 (typically -8)**
23:16	**C_H_FILTER_P0EC3: Phase 0E coefficient 3 (typically 122)**
15:8	**C_H_FILTER_P0EC2: Phase 0E coefficient 2 (typically 17)**
7:3	**C_H_FILTER_P0EC1: Phase 0E coefficient 1 (typically -5)**
2:0	**C_H_FILTER_P0EC0: Phase 0E coefficient 0 (typically 1)**

DC_WINC_C_H_FILTER_P0F_0

Offset: 510h
Read/Write: R/W
Reset: 0000.0000

Window B Horizontal Filter phase 0F

Bit	Description
31:29	**C_H_FILTER_P0FC5: Phase 0F coefficient 5 (typically 1)**
28:24	**C_H_FILTER_P0FC4: Phase 0F coefficient 4 (typically -4)**
23:16	**C_H_FILTER_P0FC3: Phase 0F coefficient 3 (typically 124)**
15:8	**C_H_FILTER_P0FC2: Phase 0F coefficient 2 (typically 8)**
7:3	**C_H_FILTER_P0FC1: Phase 0F coefficient 1 (typically -2)**
2:0	**C_H_FILTER_P0FC0: Phase 0F coefficient 0 (typically 1)**

DC_WINC_C_CSC_YOF_0

Offset: 511h
Read/Write: R/W
Reset: 0000.0000

Window B CSC Y Offset

Color Space Conversion coefficients.

The CSC can be used for YUV to RGB conversion with brightness and hue/saturation control. The CSC can only be enabled for Window B controlled by (P/S)_B_CSC_ENABLE register bits. For Y color, the Y offset is applied first and saturation (clipping) is performed immediately after the Y offset is applied.
R = sat(KYRGB * sat(Y + YOF) + KUR * U + KVR * V)
G = sat(KYRGB * sat(Y + YOF) + KUG * U + KVG * V)
B = sat(KYRGB * sat(Y + YOF) + KUB * U + KVB * V)
Saturation and rounding is performed in the range of 0 to 255 for the above equations.

Typical values are:
YOF = -16.000, KYRGB = 1.1644
KUR = 0.0000, KVR = -1.5960
KUG = -0.3918, KVG = -0.8130
KUB = 2.0172, KVB = 0.0000
KUR and KVB are typically 0.0000 but they may be programmed non-zero for hue rotation.

The CSC can also take RGB input, in which case YOF, KVB, KUG, KUR should be programmed to 0 and KYRGB will be forced to 0 by the hardware for generating R and B. KYRGB will not be forced to 0 for generating G. KVR, KYRGB, and KUB can be programmed to 1.0 or used as gain control for R, G, B, respectively.

Note that color value ranges from 0 to 255 for Y, R, G, B and -128 to 127 for U and V.

Bit	Description
7:0	**C_CSC_YOF: Y Offset in s.7.0 format**

DC_WINC_C_CSC_KYRGB_0

Offset: 512h
Read/Write: R/W
Reset: 0000.0000

Window B CSC Y Coefficient (gain) for RGB

Bit	Description
9:0	**C_CSC_KYRGB: Y Gain for R, G, B colors in 2.8 format**

DC_WINC_C_CSC_KUR_0

Offset: 513h
Read/Write: R/W
Reset: 0000.0000

Window B CSC U coefficient for R

Bit	Description
10:0	**C_CSC_KUR: U coefficients for R in s.2.8 format**

DC_WINC_C_CSC_KVR_0

Offset: 514h
Read/Write: R/W
Reset: 0000.0000

Window B CSC V coefficient for R

Bit	Description
10:0	**C_CSC_KVR: V coefficients for R in s.2.8 format**

DC_WINC_C_CSC_KUG_0

Offset: 515h
Read/Write: R/W
Reset: 0000.0000

Window B CSC U coefficient for G

Bit	Description
9:0	**C_CSC_KUG: U coefficients for G in s.1.8 format**

DC_WINC_C_CSC_KVG_0

Offset: 516h
Read/Write: R/W
Reset: 0000.0000

Window B CSC V coefficient for G

Bit	Description
9:0	**C_CSC_KVG: V coefficients for G in s.1.8 format**

DC_WINC_C_CSC_KUB_0

Offset: 517h
Read/Write: R/W
Reset: 0000.0000

Window B CSC U coefficient for B

Bit	Description
10:0	**C_CSC_KUB: U coefficients for B in s.2.8 format**

DC_WINC_C_CSC_KVB_0

Offset: 518h
Read/Write: R/W
Reset: 0000.0000

Window B CSC V coefficient for B

Bit	Description
10:0	**C_CSC_KVB: V coefficients for B in s.2.8 format**

DC_DISP_DISP_SIGNAL_OPTIONS0_0

Offset: 101h
Read/Write: R/W
Reset: 0000.0000

Display Signal Options 0

Bit	Description
26	**LM1_ENABLE: LCD Modulation 1 Enable.** 0 = Disable 1 = Enable
24	**LM0_ENABLE: LCD Modulation 0 Enable.** 0 = Disable 1 = Enable
20	**V_PULSE3_ENABLE: V Pulse 3 Enable.** 0 = Disable 1 = Enable
19	**V_PULSE2_ENABLE: V Pulse 2 Enable.** 0 = Disable 1 = Enable
18	**V_PULSE1_ENABLE: Vertical Pulse 1 Enable.** 0 = Disable 1 = Enable
16	**V_PULSE0_ENABLE: Vertical Pulse 0 Enable.** 0 = Disable 1 = Enable
12	**H_PULSE2_ENABLE: Horizontal Pulse 2 Enable.** 0 = Disable 1 = Enable

Bit	Description
10	**H_PULSE1_ENABLE: Horizontal Pulse 1 Enable.** 0 = Disable 1 = Enable
8	**H_PULSE0_ENABLE: Horizontal Pulse 0 Enable.** 0 = Disable 1 = Enable

DC_DISP_DISP_SIGNAL_OPTIONS1_0

Offset: 102h
Read/Write: R/W
Reset: 0000.0000

Display Signal Options 1

Bit	Description
18	**PP_ENABLE: LCD Programmable pulse Enable.** 0 = Disable 1 = Enable
16	**DI_ENABLE: LCD Data Inversion Enable.** 0 = Disable 1 = Enable

DC_DISP_DISP_WIN_OPTIONS_0

Offset: 103h
Read/Write: R/W
Reset: 0000.0000

Display Window Options

Bit	Description
24	**WIN_G_ENABLE: Window G Enable.** 0 = Disable 1 = Enable
16	**CURSOR_ENABLE: Cursor Enable.** 0 = Disable 1 = Enable
4	**WIN_C_ENABLE: Window C Enable.** 0 = Disable 1 = Enable
2	**WIN_B_ENABLE: Window B Enable.** 0 = Disable 1 = Enable
0	**WIN_A_ENABLE: Window A Enable.** 0 = Disable 1 = Enable

DC_DISP_MEM_HIGH_PRIORITY_0

Offset: 104h
Read/Write: R/W
Reset: 0000.0000

Memory High Priority request control

Display Memory High Priority Threshold
This controls high priority request for memory read access for each display window and for cursor. High priority request threshold should be increased in scenarios where memory access latency is high.

Bit	Description
26:24	**CSR_DISPLAYHC2MC_HPTH: Cursor Memory High Priority enable** 0= memory access for cursor is normal priority 1= memory access for cursor is high priority
22:16	**CBR_DISPLAY0C2MC_HPTH: Window C Memory High Priority threshold** Memory access for this window is high priority if the number of empty location in the read data FIFO is less than or equal to this value. Setting this parameter to 0 disables high priority memory request.
14:8	**CBR_DISPLAYB2MC_HPTH: Window B Memory High Priority threshold** Memory access for this window is high priority if the number of empty location in the read data FIFO is less than or equal to this value. Setting this parameter to 0 disables high priority memory request. This register is used for both window B) and B1
5:0	**CBR_DISPLAY0A2MC_HPTH: Window A Memory High Priority threshold** Memory access for this window is high priority if the number of empty location in the read data FIFO is less than or equal to this value. Setting this parameter to 0 disables high priority memory request.

DC_DISP_MEM_HIGH_PRIORITY_TIMER_0

Offset: 105h
Read/Write: R/W
Reset: 0000.0000

Memory High Priority request control

Bit	Description
29:24	**CSR_DISPLAYHC2MC_HPTM: Cursor Memory High Priority timer**
21:16	**CBR_DISPLAY0C2MC_HPTM: Window C Memory High Priority timer**
13:8	**CBR_DISPLAYB2MC_HPTM: Window B Memory High Priority timer** This register is used for both window B) and B1
5:0	**CBR_DISPLAY0A2MC_HPTM: Window A Memory High Priority timer**

DC_DISP_DISP_TIMING_OPTIONS_0

Offset: 106h
Read/Write: R/W
Reset: 0000.0000

Display Timing_Options

Class: Display Standard Timings

This register specifies display timing options for HSYNC and VSYNC

Bit	Description
23:16	**DISP_VSYNC_OPTIONS: VSYNC Options** This is reserved for future use.
7:0	**DISP_HSYNC_OPTIONS: HSYNC Options** This is reserved for future use.

DC_DISP_REF_TO_SYNC_0

Offset: 107h
Read/Write: R/W
Reset: 0000.0000

H/V Reference to Sync

This register specifies the start position of HSYNC and VSYNC with respect to H and V reference point (line and frame start) correspondingly. The H and V reference points correspond to the time when H and V display timing counter is re-initialized to zero correspondingly.

The H reference point also determines the point where V display timing counter is incremented so this point the horizontal relationship between HSYNC and VSYNC.

Bit	Description
27:16	**V_REF_TO_SYNC: V reference to VSYNC** (minimum 1 line clock)
11:0	**H_REF_TO_SYNC: H reference to HSYNC** (minimum 0 pixel clock)

DC_DISP_SYNC_WIDTH_0

Offset: 108h
Read/Write: R/W
Reset: 0000.0000

H/V SYNC Pulse Width
This register specifies the width of HSYNC and VSYNC pulses.

Bit	Description
27:16	**V_SYNC_WIDTH: VSYNC pulse width** (minimum 1 line clock)
11:0	**H_SYNC_WIDTH: HSYNC pulse width** (minimum 1 pixel clock)

DC_DISP_BACK_PORCH_0

Offset: 109h
Read/Write: R/W
Reset: 0000.0000

H/V Back Porch
This register specifies the distance between H/V SYNC trailing edge to beginning of display active area. This is 2's complement value and negative value indicates that H/V SYNC overlaps with the corresponding display active area.

Constraint: P_H_REF + P_H_SYNC_WIDTH + P_H_BACK_PORCH must be larger than 11.
Constraint: P_V_REF + P_V_SYNC_WIDTH + P_V_BACK_PORCH must be larger than 1.

Bit	Description
27:16	**V_BACK_PORCH: V back porch**
11:0	**H_BACK_PORCH: H back porch**

DC_DISP_DISP_ACTIVE_0

Offset: 10ah
Read/Write: R/W
Reset: 0000.0000

H/V Display Active width
This register specifies the width of H/V display active area.

Bit	Description
27:16	**V_DISP_ACTIVE: V display active height** (minimum 16 lines)
11:0	**H_DISP_ACTIVE: H display active width** (minimum 16 pixels)

DC_DISP_FRONT_PORCH_0

Offset: 10bh
Read/Write: R/W
Reset: 0000.0000

H/V Front Porch
This register specifies the distance between end of H/V display active area to the leading edge of the corresponding H/V SYNC.

Design Note: H/V active end plus the H/V front porch value minus the H/V reference to H/VSYNC determines the H/V total (final H/V count value for the H/V display counter).

Bit	Description
27:16	**V_FRONT_PORCH: VSYNC front porch** (minimum P_V_REF_TO_SYNC + 0)
11:0	**H_FRONT_PORCH: HSYNC front porch** (minimum P_H_REF_TO_SYNC + 1)

DC_DISP_H_PULSE0_CONTROL_0

Offset: 10ch
Read/Write: R/W
Reset: 0000.0000

H Pulse 0 Control

Class: Display Extended Timings

Horizontal pulse 0 is programmable pulse that repeats every line: In the NORMAL mode, this signal can have several pulses (A to D) per line with programmable width as defined by the pairs of start and end positions. The pulses must not overlap and must occur in sequence: pulse A, then pulse B, etc.

In this case, the Enable field must be set to one of the End position. If the Enable field is set to one of the Start position then the pulse generator will stop as if the Enable field is set to the previous End position. If Enable field is set to Start A position then no pulse is generated.

In the ONE_CLOCK mode this signal can have up to twice the number of pulses per line with each pulse having a width of 1 pixel clock. In this mode, the position of the one-clock pulses correspond to the enabled Start and End positions. Regardless of the mode, the pulse generator processes the pairs of start and end position sequentially in the order of: Start A, End A, Start B, End B, etc. So these start/end positions should be programmed in increasing order. If any of the positions are programmed in non-increasing order (has invalid value) then the pulse generator will stop at the last valid position.

Polarity adjustment is made prior to V display qualification. This register specifies options for Horizontal pulse 0.

Bit	Description
11:8	**H_PULSE0_LAST: H Pulse 0 Last point.** 0 = end on Start A position 1 = end on End A position 2 = end on Start B position 3 = end on End B position 4 = end on Start C position 5 = end on End C position 6 = end on Start D position 7 = end on End D position All other values = reserved
7:6	**H_PULSE0_V_QUAL: H Pulse 0 Vertical Qualifier.** 0 = always running 2 = run during vertical active area 3 = run during vertical active plus 1 line
4	**H_PULSE0_POLARITY: H Pulse 0 Polarity.** 0 = High 1 = Low Polarity adjustment is done before the vertical qualifier is applied.
3	**H_PULSE0_MODE: H Pulse 0 Mode.** 0 = Normal 1 = Single-clock mode

DC_DISP_H_PULSE0_POSITION_A_0

Offset: 10dh
Read/Write: R/W
Reset: 0000.0000

H Pulse 0 Position A

Bit	Description
27:16	**H_PULSE0_END_A: H Pulse 0 End A** (minimum P_H_PULSE0_START_A+1)
11:0	**H_PULSE0_START_A: H Pulse 0 Start A** (minimum 0)

DC_DISP_H_PULSE0_POSITION_B_0

Offset: 10eh
Read/Write: R/W
Reset: 0000.0000

H Pulse 0 Position B

Bit	Description
27:16	**H_PULSE0_END_B: H Pulse 0 End B** (minimum P_H_PULSE0_START_B+1)
11:0	**H_PULSE0_START_B: H Pulse 0 Start B** (minimum P_H_PULSE0_END_A+1)

DC_DISP_H_PULSE0_POSITION_C_0

Offset: 10fh
Read/Write: R/W
Reset: 0000.0000

H Pulse 0 Position C

Bit	Description
27:16	**H_PULSE0_END_C: H Pulse 0 End C** (minimum P_H_PULSE0_START_C+1)
11:0	**H_PULSE0_START_C: H Pulse 0 Start C** (minimum P_H_PULSE0_END_B+1)

DC_DISP_H_PULSE0_POSITION_D_0

Offset: 110h
Read/Write: R/W
Reset: 0000.0000

H Pulse 0 Position D

Bit	Description
27:16	**H_PULSE0_END_D: H Pulse 0 End D** (minimum P_H_PULSE0_START_D+1)
11:0	**H_PULSE0_START_D: H Pulse 0 Start D** (minimum P_H_PULSE0_END_C+1)

DC_DISP_H_PULSE1_CONTROL_0

Offset: 111h
Read/Write: R/W
Reset: 0000.0000

H Pulse 1 Control
Horizontal pulse 1 is programmable pulse that repeats every line. In the NORMAL mode, this signal can have several pulses (A to D) per line with programmable width as defined by the pairs of start and end positions. The pulses must not overlap and must occur in sequence: pulse A, then pulse B, etc.

In this case, the Enable field must be set to one of the End position. If the Enable field is set to one of the Start position then the pulse generator will stop as if the Enable field is set to the previous End position. If Enable field is set to Start A position then no pulse is generated.

In the ONE_CLOCK mode this signal can have up to twice the number of pulses per line with each pulse having a width of 1 pixel clock. In this mode, the position of the one-clock pulses correspond to the enabled Start and End positions. Regardless of the mode, the pulse generator processes the pairs of start and end position sequentially in the order of: Start A, End A, Start B, End B, etc. So these start/end positions should be programmed in increasing order. If any of the positions are programmed in non-increasing order (has invalid value) then the pulse generator will stop at the last valid position.

Polarity adjustment is made prior to V display qualification. This register specifies options for Horizontal pulse 1.

Bit	Description
11:8	**H_PULSE1_LAST: H Pulse 1 Last point.** 0 = end on Start A position 1 = end on End A position 2 = end on Start B position 3 = end on End B position 4 = end on Start C position 5 = end on End C position 6 = end on Start D position 7 = end on End D position All others = reserved

Bit	Description
7:6	**H_PULSE1_V_QUAL: H Pulse 1 Vertical Qualifier.** 0 = always running 2 = run during vertical active area 3 = run during vertical active plus 1 line
4	**H_PULSE1_POLARITY: H Pulse 1 Polarity.** Polarity adjustment is performed before the vertical qualifier is applied 0 = High 1 = Low
3	**H_PULSE1_MODE: H Pulse 1 Mode.** 0 = Normal 1 = Single-clock mode

DC_DISP_H_PULSE1_POSITION_A_0

Offset: 112h
Read/Write: R/W
Reset: 0000.0000

H Pulse 1 Position A

Bit	Description
27:16	**H_PULSE1_END_A: H Pulse 1 End A** (minimum P_H_PULSE1_START_A+1)
11:0	**H_PULSE1_START_A: H Pulse 1 Start A** (minimum 0)

DC_DISP_H_PULSE1_POSITION_B_0

Offset: 113h
Read/Write: R/W
Reset: 0000.0000

H Pulse 1 Position B

Bit	Description
27:16	**H_PULSE1_END_B: H Pulse 1 End B** (Minimum P_h_pulse1_start_b+1)
11:0	**H_PULSE1_START_B: H Pulse 1 Start B** (Minimum P_H_PULSE1_END_A+1)

DC_DISP_H_PULSE1_POSITION_C_0

Offset: 114h
Read/Write: R/W
Reset: 0000.0000

H Pulse 1 Position C

Bit	Description
27:16	**H_PULSE1_END_C: H Pulse 1 End C** (minimum P_H_PULSE1_START_C+1)
11:0	**H_PULSE1_START_C: H Pulse 1 Start C** (minimum P_H_PULSE1_END_B+1)

DC_DISP_H_PULSE1_POSITION_D_0

Offset: 115h
Read/Write: R/W
Reset: 0000.0000

H Pulse 1 Position D

Bit	Description
27:16	**H_PULSE1_END_D: H Pulse 1 End D** (minimum P_H_PULSE1_START_D+1)
11:0	**H_PULSE1_START_D: H Pulse 1 Start D** (minimum P_H_PULSE1_END_C+1)

DC_DISP_H_PULSE2_CONTROL_0

Offset: 116h
Read/Write: R/W
Reset: 0000.0000

H Pulse 2 Control

Horizontal pulse 2 is programmable pulse that repeats every line. In the NORMAL mode, this signal can have several pulses (A to D) per line with programmable width as defined by the pairs of start and end positions. The pulses must not overlap and must occur in sequence: pulse A, then pulse B, etc.

In this case, the Enable field must be set to one of the End position. If the Enable field is set to one of the Start position then the pulse generator will stop as if the Enable field is set to the previous End position. If Enable field is set to Start A position then no pulse is generated.

In the ONE_CLOCK mode this signal can have up to twice the number of pulses per line with each pulse having a width of 1 pixel clock. In this mode, the position of the one-clock pulses correspond to the enabled Start and End positions. Regardless of the mode, the pulse generator processes the pairs of start and end position sequentially in the order of: Start A, End A, Start B, End B, etc.

So these start/end positions should be programmed in increasing order. If any of the positions are programmed in non-increasing order (has invalid value) then the pulse generator will stop at the last valid position. Polarity adjustment is made prior to V display qualification. This register specifies options for Horizontal pulse 2.

Bit	Description
11:8	**H_PULSE2_LAST: H Pulse 2 Last point.** 0 = end on Start A position 1 = end on End A position 2 = end on Start B position 3 = end on End B position 4 = end on Start C position 5 = end on End C position 6 = end on Start D position 7 = end on End D position All others = reserved

Bit	Description
7:6	**H_PULSE2_V_QUAL: H Pulse 2 Vertical Qualifier.** 0 = always running 2 = run during vertical active area 3 = run during vertical active plus 1 line
4	**H_PULSE2_POLARITY: H Pulse 2 Polarity.** Polarity adjustment is performed before the vertical qualifier is applied 0 = High 1 = Low
3	**H_PULSE2_MODE: H Pulse 2 Mode.** 0 = Normal 1 = Single-clock mode

DC_DISP_H_PULSE2_POSITION_A_0

Offset: 117h
Read/Write: R/W
Reset: 0000.0000

H Pulse 2 Position A

Bit	Description
27:16	**H_PULSE2_END_A: H Pulse 2 End A** (minimum P_H_PULSE2_START_A+1)
11:0	**H_PULSE2_START_A: H Pulse 2 Start A** (minimum 0)

DC_DISP_H_PULSE2_POSITION_B_0

Offset: 118h
Read/Write: R/W
Reset: 0000.0000

H Pulse 2 position B

Bit	Description
27:16	**H_PULSE2_END_B: H Pulse 2 End B** (minimum P_H_PULSE2_START_B+1)
11:0	**H_PULSE2_START_B: H Pulse 2 Start B** (minimum P_H_PULSE2_END_A+1)

DC_DISP_H_PULSE2_POSITION_C_0

Offset: 119h
Read/Write: R/W
Reset: 0000.0000

H Pulse 2 Position C

Bit	Description
27:16	**H_PULSE2_END_C: H Pulse 2 End C** (minimum P_H_PULSE2_START_C+1)

Bit	Description
11:0	**H_PULSE2_START_C: H Pulse 2 Start C** (minimum P_H_PULSE2_END_B+1)

DC_DISP_H_PULSE2_POSITION_D_0

Offset: 11ah
Read/Write: R/W
Reset: 0000.0000

H Pulse 2 Position D

Bit	Description
27:16	**H_PULSE2_END_D: H Pulse 2 End D** (minimum P_H_PULSE2_START_D+1)
11:0	**H_PULSE2_START_D: H Pulse 2 Start D** (minimum P_H_PULSE2_END_C+1)

DC_DISP_V_PULSE0_CONTROL_0

Offset: 11bh
Read/Write: R/W
Reset: 0000.0000

V Pulse 0 Control

Vertical pulse 0 is programmable pulse that repeats every frame. This signal can have several pulses (A to C) per frame with programmable width as defined by the pairs of start and end positions. The pulses must not overlap and must occur in sequence: pulse A, then pulse B, etc.

In this case, the Enable field must be set to one of the End position. If the Enable field is set to one of the Start position then the pulse generator will stop as if the Enable field is set to the previous End position. If Enable field is set to Start A position then no pulse is generated. The pulse generator processes the pairs of start and end position sequentially in the order of: Start A, End A, Start B, End B, etc.

So these start/end positions should be programmed in increasing order. If any of the positions are programmed in non-increasing order (has invalid value) then the pulse generator will stop at the last valid position. This register specifies options for Vertical pulse 0.

Bit	Description
11:8	**V_PULSE0_LAST: V Pulse 0 Last point.** 0 = end on Start A position 1 = end on End A position 2 = end on Start B position 3 = end on End B position 4 = end on Start C position 5 = end on End C position All others = reserved
7:6	**V_PULSE0_DELAY: V Pulse 0 Delay.** 0 = no delay 1 = 1-line delay 2 = 2-line delay 3 = reserved

Bit	Description
4	**V_PULSE0_POLARITY: V Pulse 0 Polarity.** 0 = High 1 = Low

DC_DISP_V_PULSE0_POSITION_A_0

Offset: 11ch
Read/Write: R/W
Reset: 0000.0000

V Pulse 0 Position A

Bit	Description
27:16	**V_PULSE0_END_A: V Pulse 0 End A** (minimum P_V_PULSE0_START_A+1)
11:0	**V_PULSE0_START_A: V Pulse 0 Start A** (minimum 0)

DC_DISP_V_PULSE0_POSITION_B_0

Offset: 11dh
Read/Write: R/W
Reset: 0000.0000

V Pulse 0 Position B

Bit	Description
27:16	**V_PULSE0_END_B: V Pulse 0 End B** (minimum P_V_PULSE0_START_B+1)
11:0	**V_PULSE0_START_B: V Pulse 0 Start B** (minimum P_V_PULSE0_END_A+1)

DC_DISP_V_PULSE0_POSITION_C_0

Offset: 11eh
Read/Write: R/W
Reset: 0000.0000

V Pulse 0 Position C

Bit	Description
27:16	**V_PULSE0_END_C: V Pulse 0 End C** (minimum P_V_PULSE0_START_C+1)
11:0	**V_PULSE0_START_C: V Pulse 0 Start C** (minimum P_V_PULSE0_END_B+1)

DC_DISP_V_PULSE1_CONTROL_0

Offset: 11fh
Read/Write: R/W
Reset: 0000.0000

V pulse 1 Control

Vertical pulse 1 is programmable pulse that repeats every frame. This signal can have several pulses (A to C) per frame with programmable width as defined by the pairs of start and end positions. The pulses must not overlap and must occur in sequence: pulse A, then pulse B, etc.

In this case, the Enable field must be set to one of the End position. If the Enable field is set to one of the Start position then the pulse generator will stop as if the Enable field is set to the previous End position. If Enable field is set to Start A position then no pulse is generated. The pulse generator processes the pairs of start and end position sequentially in the order of: Start A, End A, Start B, End B, etc.

So these start/end positions should be programmed in increasing order. If any of the positions are programmed in non-increasing order (has invalid value) then the pulse generator will stop at the last valid position. This register specifies options for Vertical pulse 1.

Bit	Description
11:8	**V_PULSE1_LAST: V pulse 1 Last point.** 0 = end on Start A position 1 = end on End A position 2 = end on Start B position 3 = end on End B position 4 = end on Start C position 5 = end on End C position All others = reserved
7:6	**V_PULSE1_DELAY: V pulse 1 Delay.** 0 = no delay 1 = 1-line delay 2 = 2-line delay 3 = reserved
4	**V_PULSE1_POLARITY: V pulse 1 Polarity.** 0 = High 1 = Low

DC_DISP_V_PULSE1_POSITION_A_0

Offset: 120h
Read/Write: R/W
Reset: 0000.0000

V Pulse 1 Position A

Bit	Description
27:16	**V_PULSE1_END_A: V Pulse 1 End A** (minimum P_V_PULSE1_START_A+1)

Bit	Description
11:0	**V_PULSE1_START_A: V Pulse 1 Start A** (minimum 0)

DC_DISP_V_PULSE1_POSITION_B_0

Offset: 121h
Read/Write: R/W
Reset: 0000.0000

V Pulse 1 Position B

Bit	Description
27:16	**V_PULSE1_END_B: V Pulse 1 End B** (minimum P_V_PULSE1_START_B+1)
11:0	**V_PULSE1_START_B: V Pulse 1 Start B** (minimum P_V_PULSE1_END_A+1)

DC_DISP_V_PULSE1_POSITION_C_0

Offset: 122h
Read/Write: R/W
Reset: 0000.0000

V Pulse 1 Position C

Bit	Description
27:16	**V_PULSE1_END_C: V Pulse 1 End C** (minimum P_V_PULSE1_START_C+1)
11:0	**V_PULSE1_START_C: V Pulse 1 Start C** (minimum P_V_PULSE1_END_B+1)

DC_DISP_V_PULSE2_CONTROL_0

Offset: 123h
Read/Write: R/W
Reset: 0000.0000

V pulse 2 Control

Vertical pulse 2 is programmable pulse that repeats every frame. This signal can have one pulse (A) per frame with programmable width as defined by the pair of start and end positions.

In this case, the Enable field must be set to one of the End position. If the Enable field is set to one of the Start position then the pulse generator will stop as if the Enable field is set to the previous End position. If Enable field is set to Start A position then no pulse is generated.

So these start/end positions should be programmed in increasing order. If any of the positions are programmed in non-increasing order (has invalid value) then the pulse generator will stop at the last valid position.

This register specifies options for Vertical pulse 2.

Bit	Description
8	**V_PULSE2_LAST: V pulse 2 Last point.** 0 = Start_a 1 = End_a
4	**V_PULSE2_POLARITY: V pulse 2 Polarity.** 0 = High 1 = Low

DC_DISP_V_PULSE2_POSITION_A_0

Offset: 124h
Read/Write: R/W
Reset: 0000.0000

V Pulse 2 Position A

Bit	Description
27:16	**V_PULSE2_END_A: V Pulse 2 End A** (minimum P_V_PULSE2_START_A+1)
11:0	**V_PULSE2_START_A: V Pulse 2 Start A** (minimum 0)

DC_DISP_V_PULSE3_CONTROL_0

Offset: 125h
Read/Write: R/W
Reset: 0000.0000

V pulse 3 Control

Vertical pulse 3 is programmable pulse that repeats every frame. This signal can have one pulse (A) per frame with programmable width as defined by the pair of start and end positions. In this case, the Enable field must be set to one of the End position. If the Enable field is set to one of the Start position then the pulse generator will stop as if the Enable field is set to the previous End position. If Enable field is set to Start A position then no pulse is generated.

So these start/end positions should be programmed in increasing order. If any of the positions are programmed in non-increasing order (has invalid value) then the pulse generator will stop at the last valid position.

This register specifies options for Vertical pulse 2.

Bit	Description
8	**V_PULSE3_LAST: V pulse 3 Last point.** 0 = End on Start A position 1 = Start on End A position
4	**V_PULSE3_POLARITY: V pulse 3 Polarity.** 0 = HIGH 1 = LOW

DC_DISP_V_PULSE3_POSITION_A_0

Offset: 126h
Read/Write: R/W
Reset: 0000.0000

V Pulse 3 Position A

Bit	Description
27:16	**V_PULSE3_END_A: V Pulse 3 End A** (minimum P_V_PULSE3_START_A+1)
11:0	**V_PULSE3_START_A: V Pulse 3 Start A** (minimum 0)

DC_DISP_M0_CONTROL_0

Offset: 127h
Read/Write: R/W
Reset: 0000.0000

M0 Control

Display M0 signal

M0 signal can be generated either using a line (horizontal) or a frame (vertical) clock and it can be horizontally positioned with respect to H reference point. This signal is typically output on LM0 pin.

This register specifies options for M0 signal.

Bit	Description
27:16	**M0_H_POSITION: M0 Horizontal Position** This parameter specifies the position where M0 can toggle with respect to H reference point.
12:8	**M0_PERIOD: M0 Period** This should be program to the half of the desired M0 period (in lines) minus 1.
7	**M0_POLARITY: M0 Polarity.** 0 = High 1 = Low Polarity adjustment is applied last after phase control is applied.
6	**M0_PHASE_RESET: M0 Phase Reset.** This bit is effective only when M0 is not free running. 0 = frequency (phase) counter is not reset 1 = frequency (phase) counter is reset at beginning of vertical active display if phase control is set to VACTIVE_RESTART or at beginning of frame if phase control is set to FRAME_INVERT
5:4	**M0_PHASE_CONTROL: M0 Phase Control.** 0 = free-running 1 = reserved 2 = reset at beginning of vertical active display 3 = invert at beginning of frame. This should be set to free-running if frame clock is used.
1:0	**M0_CLOCK_SELECT: M0 Clock Select.** 0 = pixel clock (for diagnostic) 1 = reserved 2 = line clock 3 = frame clock

DC_DISP_M1_CONTROL_0

Offset: 128h
Read/Write: R/W
Reset: 0000.0000

M1 Control

Display M1 signal: The M1 signal can be generated either using a line (horizontal) or a frame (vertical) clock and it can be horizontally positioned with respect to H reference point. This signal is typically output on LM1 pin.

This register specifies options for M1 signal.

Bit	Description
27:16	**M1_H_POSITION: M1 Horizontal Position** This parameter specifies the position where M0 can toggle with respect to H reference point.
12:8	**M1_PERIOD: M1 Period** This should be program to the half of the desired M1 period (in lines) minus 1.
7	**M1_POLARITY: M1 Polarity.** 0 = High 1 = Low Polarity adjustment is applied last after phase control is applied.
6	**M1_PHASE_RESET: M1 Phase Reset.** This bit is effective only when M1 is not free running. 0 = frequency (phase) counter is not reset 1 = frequency (phase) counter is reset at beginning of vertical active display if phase control is set to VACTIVE_RESTART or at beginning of frame if phase control is set to FRAME_INVERT
5:4	**M1_PHASE_CONTROL: M1 Phase Control.** 0 = free-running 1 = reserved 2 = reset at beginning of vertical active display 3 = invert at beginning of frame This should be set to free-running if frame clock is used.
1:0	**M1_CLOCK_SELECT: M1 Clock Select.** 0 = pixel clock (for diagnostic) 1 = synchronous to M0 provided that M0 is generated using line clock. This will not work if M0 is not generated using line clock. In this case, M1 is controlled by P_M0_PHASE_RESET and P_M0_PERIOD, P_M1_PHASE_CONTROL and P_M1_POLARITY. 2 = line clock 3 = frame clock

DC_DISP_DI_CONTROL_0

Offset: 129h
Read/Write: R/W
Reset: 0000.0000

DI Control

Display Data Inversion (DI) signal generation: This signal is typically needed to control data inversion for PWM panels and is typically output on LDI pin. Horizontal position of this signal with respect to horizontal reference point can be programmed.

DI signal together with M0 may also be used to control the actual pixel data inversion. Pixel data may be controlled by either DI only or by (DI ^ M0) as specified by P_PIXDATA_INV_SELECT. The inversion control signal is then used to control pixel data inversion as specified by P_PIXDATA_INV_CONTROL. Note that even if the DI signal is disabled, pixel data inversion could still occur depending on the setting of P_PIXDATA_INV_CONTROL. Data inversion is limited to only active area. For the purpose of pixel data inversion, DI and M0 signals are used before the corresponding horizontal positioning so that these signals are always stable during active area.

In case M0 signal is used to control data inversion then it should be generated using line clock. M0 polarity control is not accounted when M0 is used to generate DI signal or to control pixel data inversion.

This register specifies options for DI signal as well as pixel data inversion.

Bit	Description
27:16	**DI_H_POSITION: DI signal Horizontal Position** This parameter specifies the position where di signal can toggle with respect to h reference point. it should not be programmed larger than p_pp_h_position if di is used to control pp **signal generation.**
7:6	**PIXDATA_INV_CONTROL: Pixel Data Inversion Control.** The control signal for pixel data inversion is defined by P_PIXDATA_INV_SELECT 0 = no pixel data inversion regardless of control signal state. 1 = Pixels 0, 2, 4 ... are inverted if control signal is high. Pixels 1, 3, 5 ... are inverted if control signal is low. 2 = Pixels 1, 3, 5 ... are inverted if control signal is high. Pixels 0, 2, 4 ... are inverted if control signal is low. 3 = all pixel data is inverted if control signal is high. NOTE: Pixel data inversion is NOT supported for 2-pixel/3-clock, 12-bit parallel display data format !
4	**PIXDATA_INV_SELECT: Pixel Data Inversion Select.** 0 = DI signal controls pixel data inversion 1 = DI or M0, (exclusive or condition) controls pixel data inversion.
1:0	**DI_MODE: DI signal Mode** 0 = DI is always low 1= DI is always high 2= DI is forced high every time M0 (before polarity adjustment) toggles from low to high; otherwise then DI toggles every line 3= DI has same frequency (phase) as M0 (before M0 polarity adjustment)

DC_DISP_PP_CONTROL_0

Offset: 12ah
Read/Write: R/W
Reset: 0000.0000

PP Control

Display Programmable Pulse (PP) signal generation: PP signal generation logic can generate up to 128 pulses per line internally and the PP pulse select registers determines which of the 128 pulses will be output. Any of the 128 internally generated pulse can be independently selected as output if they occur within one line time. PP signal is typically output on LPP pin.

Note that DI signal may impact PP generation as controlled by P_PP_REVERSAL_CONTROL. PP signal generation may be delayed (positioned) from H reference point (line start) controlled by P_PP_H_DELAY. Delaying PP may cause the last few internal PP pulses to overflow to the next line. PP is always generated using the display clock after the shift clock divider. This register specifies options for PP signal.

Bit	Description
15:12	**PP_LOW_PULSE: PP Low Pulse width (1 to 16)**
11:8	**PP_HIGH_PULSE: PP High Pulse width (1 to 16)**
7:4	**PP_H_DELAY: PP signal Horizontal Delay (0 to 15)** This parameter specifies the position where PP signal generation starts with respect to H reference point. If DI is used to generate PP signal then this parameter should not be smaller than P_DI_H_POSITION.
3:2	**PP_V_QUALIFIER: PP Vertical Qualifier.** 0 = free running (not qualified) 1 = V Pulse 1 qualified 2 = V Pulse 2 qualified 3 = V Pulse 3 qualified
1:0	**PP_DIRECTION: PP Direction (incrementing or decrementing).** 0 = always from pulse 0 to 127 (regardless of DI signal) 1 = 0 to 127 if DI=0 and 127 to 0 if DI =1 2 = 127 to 0 if DI=0 and 0 to 127 if DI =1 3 = always 127 to 0 regardless of DI

DC_DISP_PP_SELECT_A_0

Offset: 12bh
Read/Write: R/W
Reset: 0000.0000

PP Select A

The next 4 registers and P_PP_DIRECTION which of the internal 128 pulses to be output. Each bit in the four registers correspond to one internal pulse.

Bit	Description
31:0	**PP_SELECT_A: PP Select bits 31 to 0**

DC_DISP_PP_SELECT_B_0

Offset: 12ch
Read/Write: R/W
Reset: 0000.0000

PP Select B

Bit	Description
31:0	**PP_SELECT_B: PP Select bits 63 to 32**

DC_DISP_PP_SELECT_C_0

Offset: 12dh
Read/Write: R/W
Reset: 0000.0000

PP Select C

Bit	Description
31:0	**PP_SELECT_C: PP Select bits 95 to 64**

DC_DISP_PP_SELECT_D_0

Offset: 12eh
Read/Write: R/W
Reset: 0000.0000

PP Select D

Bit	Description
31:0	**PP_SELECT_D: PP Select bits 127 to 96**

DC_DISP_DISP_CLOCK_CONTROL_0

Offset: 12fh
Read/Write: R/W
Reset: 0000.0006

Display Clock Control

The shift clock divider is used to divide root clock for display controller module to generate internal shift clock for shifting data to the display. Output of this divider is typically used to generate the external shift clock which is sent to the display (SC0 and/or SC1) except for 1-pixel/1-clock parallel display. The output of this divider is also used to generate Programmable Pulse (PP) signal. For 1-pixel/1-clock parallel display, SC0 and SC1 are generated using the output of pixel clock divider which can be set to 1, 2, or 4 for 1-pixel/1-clock parallel display. The reason pixel clock divider 2 and 4 are allowed for 1-pixel/1-clock parallel display interface is so that the clock that generates PP can be generated with 2x or 4x higher frequency than pixel clock and therefore can produce higher resolution PP pulse positions. For all cases of parallel display, SC0 and SC1 can be further divided by 1, 2 or 4.

Class: Display Interface Settings

This register controls generation of shift clock to the display and internal pixel clock. Internal display pipeline runs with pixel clock and processes 1 pixel per clock.

Bit	Description
11:8	**PIXEL_CLK_DIVIDER: Pixel Clock Divider.** 0 = Divide by 1 1 = Divide by 1.5 2 = Divide by 2 3 = Divide by 3 4 = Divide by 4 5 = Divide by 6 6 = Divide by 8 7 = Divide by 9 8 = Divide by 12 9 = Divide by 16 10 = Divide by 18 11 = Divide by 24

Bit	Description
7:0	**SHIFT_CLK_DIVIDER: Shift Clock Divider** 0 = divide by 1 1 = divide by 1.5 2 = divide by 2 3 = divide by 2.5 4 = divide by 3 : : : : 254 = divide by 128 255 = divide by 128.5 Pixel clock divider is used to divide output of internal shift clock divider to generate internal pixel clock which is used to clock the internal horizontal and vertical counters. This divider also determine the output format for parallel interface, serial interface, and LCD SPI interface in conjunction with Display Data Format parameter. For 1-pixel/1-clock parallel display interface, valid settings are PCD1, PCD2, and PCD4. Note that the main reason to use PCD2 and PCD4 is to get higher frequency PP clock because the PP clock is always generated from the output of shift clock divider. For non 1-pixel/1-clock parallel display interface, valid settings are, PCD1H (2-pixel/3-clock), PCD2 (1-pixel/2-clock), and PCD3 (1-pixel/3-clock). For 1-channel serial display interface, valid settings are PCD3 (3-bpp 1-ch), PCD4 (3-bpp 1-ch), PCD6 (6-bpp 1-ch), PCD9 (9-bpp 1-ch), PCD12 (12-bpp 1-ch), PCD16 (16-bpp 1-ch), PCD18 (18-bpp 1-ch). For 2-channel serial display interface, valid settings are PCD2 (3-bpp 2-ch), PCD3 (6-bpp 2-ch), PCD6 (12-bpp 2-ch), PCD8 (16-bpp 2-ch), PCD9 (18-bpp 2-ch). For 3-channel serial display interface, valid settings are PCD1 (3-bpp 3-ch), PCD2 (6-bpp 3-ch), PCD3 (9-bpp 3-ch), PCD4 (12-bpp 3-ch), PCD6 (18-bpp 3-ch). For LCD SPI interface, valid settings are PCD12 (B4G4R4), PCD16 (B5G6R5), PCD18 (B6G6R6), PCD24 (B8G8R8), PCD8 (B5G6R5 with data/command bit), and PCD6 (B5G6R5 with data/command start byte - depending on data/command bit).

DC_DISP_DISP_INTERFACE_CONTROL_0

Offset: 130h
Read/Write: R/W
Reset: 0000.0000

Display Interface Control

This register specifies display interface options.

Bit	Description
9	**DISP_DATA_ORDER: Display Data Order.** This is effective only for 1-pixel/2-clock, 16-/18-/24- bit parallel interfaces 0 = Red pixel is output in the first clock and blue pixel is output in the second cycle 1 = Blue pixel is output in the first clock cycle and red pixel is output in the second clock cycle

Bit	Description
8	**DISP_DATA_ALIGNMENT: Display Data Alignment.** This is effective for parallel display data format and the associated Initialization Sequence (IS). 0 = Output data is MSB-aligned For 1-pixel/1-clock parallel display the output data ordering is the same regardless of display Base Color Size. For 1-pixel/1-clock parallel display data alignment is optimized for 18-bpp so the 24-bit data ordering is: LD[5:0] is blue data bits 7-2 LD[11:6] is green data bits 7-2 LD[17:12] is red data bits 7-2 LD[19:18] is blue data bits 1-0 LD[21:20] is green data bits 1-0 LD[23:22] is red data bits 1-0 Note that LD18 to LD23 signals are multiplexed with control pins (see Pin Output Select definition) 1 = Output data is LSB-aligned For 1-pixel/1-clock parallel display the output data ordering is determined by display Base Color Size. For 1-pixel/1-clock parallel display data alignment is optimized for 24-bpp as follows: LD[7:0] is blue data bits 7-0 LD[15:8] is green data bits 7-0 LD[23:16] is red data bits 7-0 Note that LD18 to LD23 signals are multiplexed with control pins (see Pin Output Select definition)
3:0	**DISP_DATA_FORMAT: Display Data Format.** Pixel Clock Divider is used together with this parameter to determine the exact display data format. 0 = 1-pixel/1-clock up to 24-bit parallel 1 = 1-pixel/2-clock 24-bit parallel 2 = 1-pixel/2-clock 18-bit parallel or 2-pixel/3-clock 12-bit parallel or 1-pixel/3-clock 18-bit parallel 3 = 1-pixel/2-clock 16-bit parallel 4 = 1-channel serial 5 = 2-channel serial 6 = 3-channel serial 7 = SPI serial NOTE: for 2-pixel/3-clock 12-bit parallel, the horizontal display active time must be even number of pixels. NOTE: 1-/2-/3-channel serial display interface supported is a low-voltage differential serial interface.

DC_DISP_DISP_COLOR_CONTROL_0

Offset: 131h
Read/Write: R/W
Reset: 0000.0000

Display Color Control

Bit	Description
27	**LCD_MD3: LCD Mode 3 signal.** 0 = Low 1 = High
26	**LCD_MD2: LCD Mode 2 signal.** 0 = Low 1 = High

Bit	Description
25	**LCD_MD1: LCD Mode 1 signal.** 0 = Low 1 = High
24	**LCD_MD0: LCD Mode 0 signal.** 0 = Low 1 = High
18	**NON_BASE_COLOR: Non Base Color** 0= zeros 1= ones MD0-3 signals are general purpose mode signals that can be output in various pins (see Pin Output Select) to configure the display device. These bits are effective at start of frame. Typically these can be changed also when changing between display register set (Primary, Secondary, etc).
17	**BLANK_COLOR: Blank Color** 0= zeros 1= ones Non Base Color applies to least significant color bits which are not part of base color and it has higher priority over Border Color but lower priority over Blank color.
16	**DISP_COLOR_SWAP: Display Color Swap.** 0 = RGB (normal) 1 = BGR (red-blue reverse)
13:12	**ORD_DITHER_ROTATION: Ordered Dither Frame Rotation** This parameter specifies the rotation frequency of the dither matrix in terms of number of frames. If programmed to 0, there is no dither matrix rotation. If programmed to N where N is larger than 0, the dither matrix is rotated clockwise every N frame.
9:8	**DITHER_CONTROL: Dither Control.** 0 = Disable 2 = Ordered dither 3 = Error diffusion dither
3:0	**BASE_COLOR_SIZE: Display Base Color Size.** This parameter determines the number of bits per color after dither. 0 = 6 bits per color (e.g. RGB 666) 1 = 1 bit per color 2 = 2 bits per color 3 = 3 bits per color 4 = 4 bits per color 5 = 5 bits per color 6 = 5 bits for R,B and 6 bits for G 7 = 3 bits for R,G and 2 bits for B 8 = 8 bits, this also forces dither to be disabled. This setting can be used to output 24-bit data in 1-pixel/clock parallel display data format.

DC_DISP_SHIFT_CLOCK_OPTIONS_0

Offset: 132h
Read/Write: R/W
Reset: 0000.0000

Shift Clock options

This register specifies options for both display shift clock 0 (SC0) and display shift clock 1 (SC1). SC0 signal is typically output on LSC0 pin and SC1 signal is typically output on LSC1 pin.

Bit	Description
23:22	**SC1_CLK_DIVIDER: SC1 Clock Divider.** 0 = divide by 1 - this is valid for all display interfaces 1 = divide by 2 - this is valid only for 1-pixel/1-clock parallel display and 2-pixel/1-clock parallel display 2 = divide by 4 - this is valid only for 1-pixel/1-clock parallel display 3 = reserved If SC1 is divided by 2 then it is synchronously reset at the beginning of the horizontal qualifier such that rising edge of SC1 is generated for the first horizontally qualified 'pixel'. If SC1 is divided by 4 then it is synchronously reset at the beginning of the horizontal qualifier such that rising edge of LSC1 is generated for the second horizontally qualified 'pixel'. In the case where there is no horizontal qualifier start of horizontal display active will be used to generate the synchronous reset. If Initialization Sequence (IS) is enabled on parallel interface then only divide by 1 is allowed for SC1 Clock Divider and SC1 must have vertical and horizontal qualifiers enabled.
21:19	**SC1_V_QUALIFIER: SC1 Vertical Qualifier.** 0 = no vertical qualifer 2 = vertical display active 3 = 1-line extended vertical display active 4 = V Pulse 1 (VP1) 5 = 1-line extended V Pulse 1 All other values = reserved
18:16	**SC1_H_QUALIFIER: SC1 Horizontal Qualifier.** 0 = disable (regardless of vertical qualifier) 1 = no horizontal qualifier (V qualifer only) 2 = horizontal display active 3 = 1-clock early & extended H display active 4 = H Pulse 1 (HP1) 5 = 1-clock early & extended H Pulse 1 All other values = reserved
7:6	**SC0_CLK_DIVIDER: SC0 Clock Divider.** 0 = divide by 1 - this is valid for all display interface 1 = divide by 2 - this is valid only for 1-pixel/1-clock parallel display and 2-pixel/1-clock parallel display 2 = divide by 4 - this is valid only for 1-pixel/1-clock parallel display 3 = reserved If SC0 is divided by 2 or 4 then it is synchronously reset at the beginning of the horizontal qualifier such that rising edge of SC0 is generated for the first horizontally qualified 'pixel'. In the case where there is no horizontal qualifier start of horizontal display active will be used to generate the synchronous reset. If Initialization Sequence (IS) is enabled on parallel interface then only divide by 1 is allowed for SC0 Clock Divider and SC0 must have vertical and horizontal qualifiers enabled.
5:3	**SC0_V_QUALIFIER: SC0 Vertical Qualifier.** 0 = no vertical qualifer 2 = vertical display active 3 = 1-line extended vertical display active 4 = V Pulse 0 (VP0) 5 = 1-line extended V Pulse 0 All other values= reserved
2:0	**SC0_H_QUALIFIER: SC0 Horizontal Qualifier.** 0 = disable (regardless of vertical qualifier) 1 = no horizontal qualifier (V qualifer only) 2 = horizontal display active 3 = 1-clock early & extended H display active 4 = H Pulse 0 (HP0) 5 = 1-clock early & extended H Pulse 0 All other values = reserved

DC_DISP_DATA_ENABLE_OPTIONS_0

Offset: 133h
Read/Write: R/W
Reset: 0000.0000

Data Enable options

DE signal is display Data Enable signal which can be used to indicate valid data area and it can be output on LSC1 pin if needed.

Bit	Description
3:2	**DE_CONTROL: DE (Data Enable) Control** 0 = 1-pixel clock pulse preceding active line (1-clock DE) 1 = LDE active for horizontal display active time (normal DE) 2 = LDE starts 1-pixel clock preceding active line but stays high on horizontal display active (early and extended DE) 3 = 1-pixel clock early horizontal display active (early DE)
0	**DE_SELECT: DE (Data Enable) Select.** 0 = DE is generated on every lines (active & blank) 1 = DE is generated only for active lines This bit also controls STH for serial display interface in the same manner.

DC_DISP_SERIAL_INTERFACE_OPTIONS_0

Offset: 134h
Read/Write: R/W
Reset: 0000.0000

Serial display interface options: Serial display interface control signals

Controls signals for the low-voltage differential serial display interface consists of the following:

SDT, STP and STH signals.
SDT and STP are asserted high if current pixel is same as previous pixel; in this case,
SDT is toggled low sometime later but STP is either toggled low at same time as SDT (if next pixel is different than current pixel) or remains high if next pixel is same as current pixel.

When doing pixel comparison, output of dither is used, so pixel comparison depends on the base color (which maybe different than the number of output data bits). Both SDT and STP are always low (disabled) if the pixel clock divider is 4 or less. STH is used to indicate the beginning of line and it is asserted high once at the beginning of each line. The STH pulse exact timing width is dependent on the exact mode.

STH is generated from Data Enable therefore Data Enable Select bit also controls STH generation and can be used to generate STH either only for active lines or both for active and blank lines. If STH is sent during blank lines then the blank lines are also transmitted.

Bit	Description
7	**STP_CONTROL: STP signal control.** 0= STP is not OR-ed with H Pulse 2 and vertical blank 1= STP is OR-ed with H Pulse 2 and vertical blank. This may be set to 1 when STP needs to be forced high during blank time, in which case H Pulse 2 should be programmed when STP needs to be forced high. Vertical blank is the area outside vertical display active.

Bit	Description
6	**STH_DURATION: STH signal duration.** 0= STH is high for 1 pixel clock in all cases except for 3-bit 2-channel and 6-bit 3-channel where STH is 1.5 pixel clock and for 3-bit 3-channel STH is 3 pixel clocks. 1= STH is high for 2 pixel clock in all cases except for 3-bit 3-channel STH is 4 pixel clocks.
5:2	**SDT_STP_DURATION: SDT and STP signal duration** 0= 1 shift clock 1= 1 pixel clock 2= 1 pixel clock - 1 shift clock 3= 1 pixel clock - 2 shift clock 4= 1 pixel clock - 3 shift clock 5= 1 pixel clock - 4 shift clock : : : F= 1 pixel clock - 14 shift clock STP active duration is same as SDT if next pixel is not the same as current pixel; else, STP active duration is always 1 pixel clock. Maximum valid setting is pixel clock divider - 1 for pixel clock divider > 4. If pixel clock divider is 4 or less, SDT and STP is always low.
1:0	**SDT_STP_MODE: SDT and STP modes.** 0 = SDT and STP disabled 1 = reserved 2 = SDT & STP enabled, duplicate data sent 3 = SDT & STP enabled, duplicate data not sent

DC_DISP_LCD_SPI_OPTIONS_0

Offset: 135h
Read/Write: R/W
Reset: 0000.0000

LCD SPI interface options

LCD SPI interface signals consists of

1. SPI Clock (SCK) which can be output on LSCK pin.
2. SPI Data (SDA) which can be output on LSDA pin.
3. Optional SPI Data/Command (SDC) which can be output on LDC pin.
4. Main-Display SPI Chip Select (Main SCS_) signal which can be output on LCS_ pin.
5. Sub-Display SPI Chip Select (Sub SCS_) signal which can be optionally output on several pins (see pin output select) - this is optional and it is used only if there is a sub display. For LCD SPI, pixel data can only be sent to either Main-Display or Sub-Display but not to both.

Main SCS_ or Sub SCS_ signal is always active low and is typically controlled by SPI logic but can also be forced active one line prior to display active (for SIS SPI) and during vertical display active area (for LCD SPI).

Bit	Description
4	**LCD_SPI_DIRECTION: LCD SPI Data Direction.** 0 = msb to lsb 1 = lsb to msb Note that data direction does not affect the start byte direction (which is always msb to lsb) and position (always first 8-bit of serial data) for SPI16SB mode.

Bit	Description
3:2	**SPI_CS_CONTROL: LCD SPI Chip Select (SCS_) Control for both IS SPI or LCD SPI.** 0 = Main SCS_ or Sub SCS_ is controlled by LCD SPI or by IS SPI 1 = Main SCS_ or Sub SCS_ is controlled by LCD SPI, and depending on LCD SPI Chip Select bit, one of them is forced active for 1-line prior to display active when IS SPI is enabled 2 = Main SCS_ or Sub SCS_ is controlled by IS SPI, and depending on LCD SPI Chip Select bit, one of them is forced active during vertical display active area when LCD SPI is enabled 3 = Main SCS_ or Sub SCS_, depending on LCD SPI Chip Select bit, is forced active 1-line prior to display active when IS SPI is enabled and also during vertical display active area when LCD SPI is enabled
1	**LCD_SPI_DC: LCD SPI Data/Command (SDC).** 0 = SPI Data/Command is low for LCD SPI writes to the display. For PCD6 data format, command byte is sent. 1 = SPI Data/Command is high for LCD SPI writes to the display. For PCD6 data format, data byte is sent.
0	**LCD_SPI_CS: LCD SPI Chip Select (SCS_).** 0 = Send LCD SPI data to Main Display (Main SCS_ is activated) 1 = Send LCD SPI data to Sub Display (Sub SCS_ is activated) This bit is also used when SPI Chip Select Control are NOT LCD_IS_SPI to determine either Main SCS_ or Sub SCS_ to be forced active.

DC_DISP_BORDER_COLOR_0

Offset: 136h
Read/Write: R/W
Reset: 0000.0000

Border Color

Border Color defines the color of areas within the active display area which are outside the defined active windows. This is 24-bit color which is applied after blending.

Bit	Description
23:16	**BORDER_COLOR_B: Blue Border Color**
15:8	**BORDER_COLOR_G: Green Border Color**
7:0	**BORDER_COLOR_R: Red Border Color**

DC_DISP_COLOR_KEY0_LOWER_0

Offset: 137h
Read/Write: R/W
Reset: 0000.0000

Color Key 0 Lower value: Color Key 0 and Color Key 1 Two ranges of color key are defined and they are common for all windows because it is expected that typically only one window will have color key enabled. Because there are two sets of color key, it is possible to have 2 windows each using one color key set. Usage of this color key is described in the Display Color Key and Blending class.

Bit	Description
23:16	**COLOR_KEY0_L_B: Color Key 0 Blue (U) Lower value**
15:8	**COLOR_KEY0_L_G: Color Key 0 Green (Y) Lower value**
7:0	**COLOR_KEY0_L_R: Color Key 0 Red (V) Lower value**

DC_DISP_COLOR_KEY0_UPPER_0

Offset: 138h
Read/Write: R/W
Reset: 0000.0000

Color Key 0 Upper value

Bit	Description
23:16	**COLOR_KEY0_U_B: Color Key 0 Blue (U) Upper value**
15:8	**COLOR_KEY0_U_G: Color Key 0 Green (Y) Upper value**
7:0	**COLOR_KEY0_U_R: Color Key 0 Red (V) Upper value**

DC_DISP_COLOR_KEY1_LOWER_0

Offset: 139h
Read/Write: R/W
Reset: 0000.0000

Color Key 0 Lower value

Bit	Description
23:16	**COLOR_KEY1_L_B: Color Key 0 Blue (U) Lower value**
15:8	**COLOR_KEY1_L_G: Color Key 0 Green (Y) Lower value**
7:0	**COLOR_KEY1_L_R: Color Key 0 Red (V) Lower value**

DC_DISP_COLOR_KEY1_UPPER_0

Offset: 13ah
Read/Write: R/W
Reset: 0000.0000

Color Key 0 Upper value

Bit	Description
23:16	**COLOR_KEY1_U_B: Color Key 0 Blue (U) Upper value**
15:8	**COLOR_KEY1_U_G: Color Key 0 Green (Y) Upper value**
7:0	**COLOR_KEY1_U_R: Color Key 0 Red (V) Upper value**

DC_DISP_G_POSITION_0

Offset: 13bh
Read/Write: R/W
Reset: 0000.0000

Window G Position

Class: Display Window G

Display Window G parameters: Window G defines part of display active area which can be sent to encoder pre-processor (EPP) module. Pixel data sent to the EPP is in RGB888 format and this data can be converted by the EPP to planar YUV format for encoding.

Bit	Description
27:16	**G_V_POSITION: Window G Vertical Position** This is specified with respect to the top edge of active display area.
11:0	**G_H_POSITION: Window G Horizontal Position** This is specified with respect to the left edge of active display area.

DC_DISP_G_SIZE_0

Offset: 13ch
Read/Write: R/W
Reset: 0000.0000

Window G Size

Bit	Description
27:16	**G_V_SIZE: Window G Vertical Size (lines)** This is the vertical size after scaling.
11:0	**G_H_SIZE: Window G Horizontal Size (pixels)** This is the horizontal size after scaling.

DC_DISP_CURSOR_FOREGROUND_0

Offset: 13dh
Read/Write: R/W
Reset: 0000.0000

Cursor Foreground color

Class: Hardware Cursor

Hardware cursor is supported for 32x32 or for 64x64 2-bpp cursor. Cursor start address is aligned to 1 KB boundary. All cursor registers except for cursor foreground and background colors are shadowed twice.

The first shadow registers are written after the last byte of D1PCP register is written. The second shadow registers are written at frame boundary. In this way, cursor registers except for cursor foreground and background colors are not effective until the next frame boundary.

When cursor registers are read by host, the value of the first stage registers are returned. Cursor scaling and flipping are not implemented so this must be done by software if needed. Cursor H/V positions are signed number with respect to one of the display windows or with respect to upper left position of display active area as specified by cursor clipping parameter which also determine cursor clipping boundary. If cursor position is with respect to one of the display window and the corresponding display window is disabled then cursor will also be disabled.

Bit	Description
23:16	**CURSOR_FOREGROUND_B: Cursor Blue Foreground Color**
15:8	**CURSOR_FOREGROUND_G: Cursor Green Foreground Color**
7:0	**CURSOR_FOREGROUND_R: Cursor Red Foreground Color**

DC_DISP_CURSOR_BACKGROUND_0

Offset: 13eh
Read/Write: R/W
Reset: 0000.0000

Cursor Background color

Bit	Description
23:16	**CURSOR_BACKGROUND_B: Cursor Blue Background Color**
15:8	**CURSOR_BACKGROUND_G: Cursor Green Background Color**
7:0	**CURSOR_BACKGROUND_R: Cursor Red Background Color**

DC_DISP_CURSOR_START_ADDR_0

Offset: 13fh
Read/Write: R/W
Reset: 0000.0000

Cursor Start Address

Bit	Description
29:28	**CURSOR_CLIPPING: Cursor Clipping Select.** 0 = Display 1 = Window A 2 = Window B 3 = Window C
24	**CURSOR_SIZE: Cursor Size.** 0 = 32 x 32 1 = 64 x 64
15:0	**CURSOR_START_ADDR: Cursor Start Address bits 19:10**

DC_DISP_CURSOR_POSITION_0

Offset: 140h
Read/Write: R/W
Reset: 0000.0000

Cursor Position

Cursor position is with respect to top-left corner of display active area, or window A, or window B, or window C as specified cursor clipping parameter.

Bit	Description
28:16	**V_CURSOR_POSITION: V cursor position (signed)**
12:0	**H_CURSOR_POSITION: H cursor position (signed)**

DC_DISP_INIT_SEQ_CONTROL_0

Offset: 141h
Read/Write: R/W
Reset: 0000.0000

Initialization Sequence Control

Class: Initialization Sequence (IS)

Display initialization sequence may have to be written to the display if the display has built-in frame buffer. This initialization sequence is used typically to re-initialize the display buffer start address and maybe needed once per frame (frame initialization sequence) and/or once per line (line initialization sequence). Frame initialization sequence is sent during the horizontal active time of the line just before the first active display line.
Line initialization sequence is currently NOT supported.

Initialization sequence can be done through parallel LCD interface or through SPI serial interface. Software is responsible in making sure that the active line time is sufficient to send initialization sequence. For parallel interface initialization, the signals used as chip selects (typically these are one of the vertical signals) must be programmed to be active one line just before the first active display line. Also SC0/SC1 clock divider must be programmed to divide by 1 if initialization sequence is enabled.

Bit	Description
11:8	**FRAME_INIT_SEQ_CYCLES: Frame Initialization Sequence Cycles** This parameter specifies the number of frame initialization sequence cycles to send. If programmed to 0, there is no frame initialization cycle generated.
7	**INIT_SEQ_DC_CONTROL: Initialization Sequence DC Pin** This bit is used only for parallel initialization sequence and it controls how data/command is added to the vertical signal selected by P_INIT_SEQ_DC_SIGNAL. 0= parallel IS DC is inverted and then AND-ed to the vertical signal 1= parallel IS DC is OR-ed to the vertical signal
6:4	**INIT_SEQ_DC_SIGNAL: Frame Initialization Sequence DC Pin.** This parameter is used only for parallel initialization sequence and it specifies which signal carries the data/command signal. 0 = parallel IS DC signal is not needed 1 = parallel IS DC on Vertical Sync 2 = parallel IS DC on Vertical Pulse 0 3 = parallel IS DC on Vertical Pulse 1 4 = parallel IS DC on Vertical Pulse 2 5 = parallel IS DC on Vertical Pulse 3 other = reserved
1	**INIT_SEQUENCE_MODE: Initialization Sequence Mode.** 0= Send init sequence through parallel LCD interface 1= Send init sequence through SPI serial interface
0	**SEND_INIT_SEQUENCE: Send Initialization Sequence (IS).** 0 = Disable 1 = Enable

DC_DISP_SPI_INIT_SEQ_DATA_A_0

Offset:	142h
Read/Write:	R/W
Reset:	0000.0000

SPI Init Sequence Write Data A

For parallel initialization sequence there are two possible data widths: 9 bits or 18 bits. If parallel IS is selected, the number of bits per cycle depend on the DISP_DATA_FORMAT register programming. 18-bit parallel IS cycles are performed for 1-pixel/1-clock parallel interface (DF1P1C). 9-bit parallel IS cycles are performed for non 1-pixel/1-clock parallel interface.

Parallel IS cycles must be completed prior to the end of horizontal active of the line where IS cycles are sent. If all the cycles have been completed prior to the end of horizontal active, control signals are held inactive and last output data is held till end of horizontal active. For 9-bit parallel initialization sequence, the data is output in either LD[8:0] pins or LD[17:9] pins depending on display data alignment.

For serial initialization sequence using SPI (IS SPI) there are six possible data widths:
8 bits, 9 bits, 12 bits, 16 bits, 16 bits data plus start byte (24 bits), 18 bits, or 24 bits.

Parameters in SPI_CONTROL register and SPI_START_BYTE register is also used for serial initialization sequence using SPI. Serial IS cycles must also be completed prior to the end of horizontal active of the line where initialization cycles are sent. The programmer needs to make sure that register programming is such that this is true. If all the cycles have been completed prior to the end of horizontal active, SPI signals will be forced inactive until the next SPI cycles.

The following shows how initialization sequence data bits are used when sending initialization sequence:

For 18-bit parallel initialization - up to 6 initialization cycles can be done.

18Bit Parallel Initialization										
Init Cycle	1	2	3	4	5	6				
Data	17:0	35:18	53:36	71:54	89:72	107:90				
LSC0 Enable	108	111	114	117	120	123				
LSC1 Enable	109	112	115	118	121	124				
Data/ Command	110	113	116	119	122	125				

Serial Initialization using SPI

For serial initialization using SPI, main display SPI chip select (Main SCS_) is always output on LCS_ pin while sub display SPI chip select (Sub SCS_) can be optionally output on several pins (see pin output select definition). Initialization cycle through SPI interface can only be sent to either main or sub display but not to both and the selection bits are specified in the tables below. Note that 0 indicates main display initialization and 1 indicates sub display initialization.

8Bit SPI Initialization: (Up to 12 initialization cycles)										
Init Cycle	1	2	3	4	5	6	7	8	9	10
Data	7:0	15:8	23:16	31:24	39:32	47:40	55:48	63:56	71:64	79:72
Primary/ Secondary SCS_	96	98	100	102	104	106	108	110	112	114
SDC	97	99	101	103	105	107	109	11	113	115
Init Cycle	11	12								
Data	87:80	95:88								
Primary/ Secondary SCS_	116	118								
SDC	117	119								
12Bit SPI Initialization: (Up to 9 initialization cycles)										
Init cycle	1	2	3	4	5	6	7	8	9	
Data	11:0	23:12	35:24	47:36	59:48	71:60	83:72	95:84	108:96	
Primary/ Secondary SCS_	109	111	113	115	117	119	121	123	125	
LDC	110	112	114	116	118	120	122	124	126	
16Bit SPI Initialization: (Up to 7 initialization cycles)										
Init Cycle	1	2	3	4	5	6	7			
Data	15:0	31:16	47:32	63:48	79:64	95:80	111:96			
Primary/ Secondary SCS_	112	114	116	118	120	122	124			
SDC	113	115	117	119	121	123	125			
18Bit SPI Initialization: (Up to 6 initialization cycles)										

Inti Cycle	1	2	3	4	5	6	
Data	17:0	35:18	53:36	71:54	89:72	107:90	
Primary/Secondary SCS_	108	110	112	114	116	118	
SDC	109	111	11	115	117	119	
24Bit SPI Initialization: (Up to 4 initialization cycles)							
Init Cycle	1	2	3	4			
Data	23:0	47:24	71:48	95:72			
Primary/ Secondary SCS_	96	98	100	102			
SDC	97	99	101	103			

Bit	Description
31:0	**SPI_INIT_SEQ_DATA_A: SPI Init Sequence Write Data bits 31-0**

DC_DISP_SPI_INIT_SEQ_DATA_B_0

Offset: 143h
Read/Write: R/W
Reset: 0000.0000

SPI Init Sequence Write Data B

Bit	Description
31:0	**SPI_INIT_SEQ_DATA_B: SPI Init Sequence Write Data bits 63-32**

DC_DISP_SPI_INIT_SEQ_DATA_C_0

Offset: 144h
Read/Write: R/W
Reset: 0000.0000

SPI Init Sequence Write Data C

Bit	Description
31:0	**SPI_INIT_SEQ_DATA_C: SPI Init Sequence Write Data bits 95-64**

DC_DISP_SPI_INIT_SEQ_DATA_D_0

Offset: 145h
Read/Write: R/W
Reset: 0000.0000

SPI Init Sequence Write Data D

Bit	Description
31:0	**SPI_INIT_SEQ_DATA_D: SPI Init Sequence Write Data bits 127-96**

DC_DISP_DC_MCCIF_FIFOCTRL_0

Offset:	146h
Read/Write:	R/W
Reset:	0000.0000

Memory Client Interface Async Fifo Optimization Register A

Memory Client Interface Fifo Control Register. The registers below allow to optimize the synchronization timing in the memory client asynchronous FIFOs. When they can be used depend on the client and memory controller clock ratio.

Additionally, the RDMC_RDFAST/RDCL_RDFAST fields can increase power consumption if the asynchronous FIFO is implemented as a real RAM. There is no power impact on latch-based FIFOs. Flip flop-based FIFOs do not use these fields. See recommended settings below.

Note: The register fields can only be changed when the memory client async FIFOs are empty.

The register field ending with WRCL_MCLE2X (if any) can be set to improve async FIFO synchronization on the write side by one client clock cycle if the memory controller clock frequency is less or equal to twice the client clock frequency:

mcclk_freq <= 2 * clientclk_freq

The register field ending with WRMC_CLLE2X (if any) can be set to improve async FIFO synchronization on the write side by one memory controller clock cycle if the client clock frequency is less or equal to twice the memory controller clock frequency:

clientclk_freq <= 2 * mcclk_freq

The register field ending with RDMC_RDFAST (if any) can be set to improve async FIFO synchronization on the read side by one memory controller clock cycle.

Note: RDMC_RDFAST can be used along with WRCL_MCLE2X only when:

mcclk_freq <= clientclk_freq

The register field ending with RDCL_RDFAST (if any) can be set to improve async FIFO synchronization on the read side by one client clock cycle.

Note: RDCL_RDFAST can be used along with WRMC_CLLE2X only when:

clientclk_freq <= mcclk_freq

RECOMMENDED SETTINGS

Client writing to FIFO, memory controller reading from FIFO
- mcclk_freq <= clientclk_freq
You can enable both RDMC_RDFAST and WRCL_CLLE2X. If one of the FIFOs is a real RAM and power is a concern, you should avoid enabling RDMC_RDFAST.
- clientclk_freq < mcclk_freq <= 2 * clientclk_freq
You can enable RDMC_RDFAST or WRCL_MCLE2X, but because the client clock is slower, you should enable only WRCL_MCLE2X.
- 2 * clientclk_freq < mcclk_freq
You can only enable RDMC_RDFAST. If one of the FIFOs is a real RAM and power is a concern, you should avoid enabling RDMC_RDFAST.

Memory controller writing to FIFO, client reading from FIFO
- clientclk_freq <= mcclk_freq
You can enable both RDCL_RDFAST and WRMC_CLLE2X. If one of the FIFOs is a real RAM and power is a concern, you should avoid enabling RDCL_RDFAST.
- mcclk_freq < clientclk_freq <= 2 * mcclk_freq
You can enable RDCL_RDFAST or WRMC_CLLE2X, but because the memory controller clock is slower, you should enable only WRMC_CLLE2X.
- 2 * mcclk_freq < clientclk_freq
You can only enable RDCL_RDFAST. If one of the FIFOs is a real RAM and power is a concern, you should avoid enabling RDCL_RDFAST.

Bit	Description
3	**DC_MCCIF_RDCL_RDFAST:** 0 = Disable 1 = Enable
2	**DC_MCCIF_WRMC_CLLE2X:** 0 = Disable 1 = Enable
1	**DC_MCCIF_RDMC_RDFAST:** 0 = Disable 1 = Enable
0	**DC_MCCIF_WRCL_MCLE2X:** 0 = Disable 1 = Enable

DISPLAY_OBS_CONTROL_SIGNALS0_0

Offset: 000h
Read/Write: RO
Reset: 0000.0000

Bit	Description
29	**DBG_PP_S:**
28	**DBG_PP_R:**
27	**DBG_PP_START:**
26	**DBG_DCPP_CLKEN:**
25	**DBG_PM1_CNT_DC:**
24	**DBG_PM1_CNT_P:**
23	**DBG_PM1_CLK_CNT_Z:**
22	**DBG_DCPM1_CLKEN:**

Bit	Description
21	**DBG_PM0_CNT_DC:**
20	**DBG_PM0_CNT_P:**
19	**DBG_PM0_CLK_CNT_Z:**
18	**DBG_DCPM0_CLKEN:**
16	**DBG_DCV_CLKEN:**
15	**DBG_DCH_CLKEN:**
14	**DBG_DCPC_CLKEN:**
13	**DBG_DCPB_CLKEN:**
12	**DBG_DCPA_CLKEN:**
11	**DBG_DCP_CLKEN:**
10	**DBG_DCS_CLKEN:**
9	**DCPRSTN:**
8	**D_VACTIVE_E:**
7	**D_VACTIVE:**
6	**VSYNC:**
5	**HSYNC:**
4	**ENDOFFRAME:**
3	**FRAME_START:**
2	**LINE_START:**
1	**DC_STOP:**
0	**DC_RUN:**

DISPLAY_OBS_CONTROL_SIGNALS1_0

Offset: 001h
Read/Write: RO
Reset: 0000.0000

Bit	Description
28	**DBG_ERRDIFF_DP_SRST:**
27	**DBG_ERRDIFF_DP_EN:**
26	**DBG_DITHER_DP_EN:**
25	**DBG_CKMUX_D_ACTIVE:**
22	**LCD_SPI_RUN:**
21	**DBG_SPI_WCNT_DONE:**
20	**DBG_SPICD_CNT_DONE:**
19	**DBG_HOST_SIS_DONE:**
18	**DBG_HOST_SIS_RUN:**
17	**DBG_HOST_SPI_TTRIG:**
16	**DBG_HOST_SPI_TRIG:**
15	**HOST_SPI_SAFE:**
14	**SIS_SPI:**
13	**SIS_SPI_RUN:**
12	**LCD_SPI_RUN:**
11	**LCD_SPI:**
10	**DBG_DCSPI_CLKEN:**

Bit	Description
9:8	**DBG_CURRENT_ACTIVE_DISP:**
5	**DBG_FLDUPDATE_PEND:**
4	**DBG_RAISE_CMD_PEND:**
3	**DBG_BUFUPDATE_PEND:**
2	**DBG_ACTIVEDISP_PEND:**
1	**DBG_SEC_ONESHOT_PEND:**
0	**DBG_PRI_ONESHOT_PEND:**

DISPLAY_OBS_CURSOR_OR_MISC_0

Offset: 002h
Read/Write: RO
Reset: 0000.0000

Bit	Description
14	**DBG_M1_R:**
13	**DBG_DCM1_CLKEN:**
12	**DBG_M0_R:**
11	**DBG_M0_1:**
10	**DBG_DCM0_CLKEN:**
2	**DBG_CKMUX_HC_ACTIVE:**
1:0	**DBG_CURS_FETCH_STATE:**

DISPLAY_OBS_WIN_A_0

Offset: 003h
Read/Write: RO
Reset: 0000.0000

Bit	Description
19	**CBR_DISPLAY0A2MC_DIRDY:**
18	**CBR_DISPLAY0A2MC_DIREQ:**
17:14	**DBG_WA_HSHIFTCOUNT:**
13	**DBG_POP_ALIGNWA_FIFO:**
12:11	**DBG_WA_CK_MATCH:**
10	**DBG_CKMUX_WA_ACTIVE:**
9	**DBG_WA_DV_DP_EN:**
8	**DBG_WA_DV_MUX_EN:**
7	**DBG_WA_CP_DP_EN:**
4	**DBG_WA_HSCALE_DP_EN:**
3	**WA_VACTIVE_E:**
2	**WA_VACTIVE:**
1:0	**DBG_WINA_FETCH_STATE:**

DISPLAY_OBS_WIN_B_0

Offset: 004h
Read/Write: RO
Reset: 0000.0000

Bit	Description
28	**CBR_DISPLAY1B2MC_DIRDY:**
27	**CBR_DISPLAY1B2MC_DIREQ:**
26	**DBG_WB_LINE1_HFILT_DP_EN:**
25	**DBG_WB_LINE1_HSCALE_DP_EN:**
21	**DBG_WB_VFILT_EN:**
20	**DBG_WB_HFILT_EN:**
19	**CBR_DISPLAY0B2MC_DIRDY:**
18	**CBR_DISPLAY0B2MC_DIREQ:**
17:14	**DBG_WB_HSHIFTCOUNT:**
13	**DBG_POP_ALIGNWB_FIFO:**
12:11	**DBG_WB_CK_MATCH:**
10	**DBG_CKMUX_WB_ACTIVE:**
9	**DBG_WB_DV_DP_EN:**
8	**DBG_WB_DV_MUX_EN:**
7	**DBG_WB_CP_DP_EN:**
6	**DBG_WB_VFILT_DP_EN:**
5	**DBG_WB_LINE0_HFILT_DP_EN:**
4	**DBG_WB_LINE0_HSCALE_DP_EN:**
3	**WB_VACTIVE_E:**
2	**WB_VACTIVE:**
1:0	**DBG_WINB_FETCH_STATE:**

DISPLAY_OBS_WIN_C_0

Offset: 005h
Read/Write: RO
Reset: 0000.0000

Bit	Description
20	**DBG_WC_HFILT_EN:**
19	**CBR_DISPLAY0C2MC_DIRDY:**
18	**CBR_DISPLAY0C2MC_DIREQ:**
17:14	**DBG_WC_HSHIFTCOUNT:**
13	**DBG_POP_ALIGNWC_FIFO:**
12:11	**DBG_WC_CK_MATCH:**
10	**DBG_CKMUX_WC_ACTIVE:**
9	**DBG_WC_DV_DP_EN:**
8	**DBG_WC_DV_MUX_EN:**
7	**DBG_WC_CP_DP_EN:**
5	**DBG_WC_LINE0_HFILT_DP_EN:**
4	**DBG_WC_LINE0_HSCALE_DP_EN:**
3	**WC_VACTIVE_E:**
2	**WC_VACTIVE:**
1:0	**DBG_WINC_FETCH_STATE:**

DC_P_P_DISP_SIGNAL_OPTIONS0_0

Offset: 101h
Read/Write: R/W
Reset: 0000.0000

Display Signal Options 0

Bit	Description
26	**P_M1_ENABLE: M1 Enable.** 0 = Disable 1 = Enable
24	**P_M0_ENABLE: M0 Enable.** 0 = Disable 1 = Enable
20	**P_V_PULSE3_ENABLE: V Pulse 3 Enable.** 0 = Disable 1 = Enable
19	**P_V_PULSE2_ENABLE: V Pulse 2 Enable.** 0 = Disable 1 = Enable
18	**P_V_PULSE1_ENABLE: V Pulse 1 Enable.** 0 = Disable 1 = Enable
16	**P_V_PULSE0_ENABLE: V Pulse 0 Enable.** 0 = Disable 1 = Enable
12	**P_H_PULSE2_ENABLE: H Pulse 2 Enable.** 0 = Disable 1 = Enable

Bit	Description
10	**P_H_PULSE1_ENABLE: H Pulse 1 Enable.** 0 = Disable 1 = Enable
8	**P_H_PULSE0_ENABLE: H Pulse 0 Enable.** 0 = Disable 1 = Enable

DC_P_P_DISP_SIGNAL_OPTIONS1_0

Offset: 102h
Read/Write: R/W
Reset: 0000.0000

Display Signal Options 1

Bit	Description
18	**P_PP_ENABLE: PP Enable.** 0 = Disable 1 = Enable
16	**P_DI_ENABLE: DI Enable.** 0 = Disable 1 = Enable

DC_P_P_DISP_WIN_OPTIONS_0

Offset: 103h
Read/Write: R/W
Reset: 0000.0000

Display Window Options

Bit	Description
24	**P_WIN_G_ENABLE: Window G Enable.** 0 = Disable 1 = Enable
16	**P_CURSOR_ENABLE: Cursor Enable.** 0 = Disable 1 = Enable
4	**P_WIN_C_ENABLE: Window C Enable.** 0 = Disable 1 = Enable
2	**P_WIN_B_ENABLE: Window B Enable.** 0 = Disable 1 = Enable
0	**P_WIN_A_ENABLE: Window A Enable.** 0 = Disable 1 = Enable

DC_P_P_MEM_HIGH_PRIORITY_0

Offset: 104h
Read/Write: R/W
Reset: 0000.0000

Memory High Priority request control

Display Memory High Priority Threshold: This controls high priority request for memory read access for each display window and for cursor. High priority request threshold should be increased in scenarios where memory access latency is high.

Bit	Description
26:24	**P_CSR_DISPLAYHC2MC_HPTH: Cursor Memory High Priority enable** 0= memory access for cursor is normal priority 1= memory access for cursor is high priority
22:16	**P_CBR_DISPLAY0C2MC_HPTH: Window C Memory High Priority threshold** Memory access for this window is high priority if the number of empty location in the read data FIFO is less than or equal to this value. Setting this parameter to 0 disables high priority memory request.
14:8	**P_CBR_DISPLAYB2MC_HPTH: Window B Memory High Priority threshold** Memory access for this window is high priority if the number of empty location in the read data FIFO is less than or equal to this value. Setting this parameter to 0 disables high priority memory request. This register is used for both window B) and B1
5:0	**P_CBR_DISPLAY0A2MC_HPTH: Window A Memory High Priority threshold** Memory access for this window is high priority if the number of empty location in the read data FIFO is less than or equal to this value. Setting this parameter to 0 disables high priority memory request.

DC_P_P_MEM_HIGH_PRIORITY_TIMER_0

Offset: 105h
Read/Write: R/W
Reset: 0000.0000

Memory High Priority request control

Bit	Description
29:24	**P_CSR_DISPLAYHC2MC_HPTM: Cursor Memory High Priority timer**
21:16	**P_CBR_DISPLAY0C2MC_HPTM: Window C Memory High Priority timer**
13:8	**P_CBR_DISPLAYB2MC_HPTM: Window B Memory High Priority timer** **This register is used for both window B) and B1**
5:0	**P_CBR_DISPLAY0A2MC_HPTM: Window A Memory High Priority timer**

DC_P_P_DISP_TIMING_OPTIONS_0

Offset: 106h
Read/Write: R/W
Reset: 0000.0000

Display Timing_Options

Class: Display Standard Timings

This register specifies display timing options for HSYNC and VSYNC

Bit	Description
23:16	**P_DISP_VSYNC_OPTIONS: VSYNC Options** **This is reserved for future use.**
7:0	**P_DISP_HSYNC_OPTIONS: HSYNC Options** **This is reserved for future use.**

DC_P_P_REF_TO_SYNC_0

Offset: 107h
Read/Write: R/W
Reset: 0000.0000

H/V Reference to Sync

This register specifies the start position of HSYNC and VSYNC with respect to H and V reference point (line and frame start) correspondingly. The H and V reference points correspond to the time when H and V display timing counter is re-initialized to zero correspondingly.

The H reference point also determines the point where V display timing counter is incremented so this point the horizontal relationship between HSYNC and VSYNC.

Bit	Description
27:16	**P_V_REF_TO_SYNC: V reference to VSYNC** (minimum 1 line clock)
11:0	**P_H_REF_TO_SYNC: H reference to HSYNC** (minimum 0 pixel clock)

DC_P_P_SYNC_WIDTH_0

Offset: 108h
Read/Write: R/W
Reset: 0000.0000

H/V SYNC Pulse Width

This register specifies the width of HSYNC and VSYNC pulses.

Bit	Description
27:16	**P_V_SYNC_WIDTH: VSYNC pulse width** (minimum 1 line clock)
11:0	**P_H_SYNC_WIDTH: HSYNC pulse width** (minimum 1 pixel clock)

DC_P_P_BACK_PORCH_0

Offset: 109h
Read/Write: R/W
Reset: 0000.0000

H/V Back Porch

This register specifies the distance between H/V SYNC trailing edge to beginning of display active area. This is 2's complement value and negative value indicates that H/V SYNC overlaps with the corresponding display active area.

Constraint: P_H_REF + P_H_SYNC_WIDTH + P_H_BACK_PORCH must be larger than 11.
Constraint: P_V_REF + P_V_SYNC_WIDTH + P_V_BACK_PORCH must be larger than 1.

Bit	Description
27:16	**P_V_BACK_PORCH: V back porch**
11:0	**P_H_BACK_PORCH: H back porch**

DC_P_P_DISP_ACTIVE_0

Offset: 10ah
Read/Write: R/W
Reset: 0000.0000

H/V Display Active width

This register specifies the width of H/V display active area.

Bit	Description
27:16	**P_V_DISP_ACTIVE: V display active width** (minimum 16 lines)
11:0	**P_H_DISP_ACTIVE: H display active width** (minimum 16 pixels)

DC_P_P_FRONT_PORCH_0

Offset: 10bh
Read/Write: R/W
Reset: 0000.0000

H/V Front Porch

This register specifies the distance between end of H/V display active area to the leading edge of the corresponding H/V SYNC. Design Note: H/V active end plus the H/V front porch value minus the H/V reference to H/VSYNC determines the H/V total (final H/V count value for the H/V display counter).

Bit	Description
27:16	**P_V_FRONT_PORCH: VSYNC front porch** (minimum P_V_REF_TO_SYNC + 0)
11:0	**P_H_FRONT_PORCH: HSYNC front porch** (minimum P_H_REF_TO_SYNC + 1)

DC_P_P_H_PULSE0_CONTROL_0

Offset: 10ch
Read/Write: R/W
Reset: 0000.0000

H Pulse 0 Control

Class: Display Extended Timings

Horizontal pulse 0 is programmable pulse that repeats every line. In the NORMAL mode, this signal can have several pulses (A to D) per line with programmable width as defined by the pairs of start and end positions. The pulses must not overlap and must occur in sequence: pulse A, then pulse B, etc. In this case, the Enable field must be set to one of the End position. If the Enable field is set to one of the Start position then the pulse generator will stop as if the Enable field is set to the previous End position. If Enable field is set to Start A position then no pulse is generated.

In the ONE_CLOCK mode this signal can have up to twice the number of pulses per line with each pulse having a width of 1 pixel clock. In this mode, the position of the one-clock pulses correspond to the enabled Start and End positions.

Regardless of the mode, the pulse generator processes the pairs of start and end position sequentially in the order of: Start A, End A, Start B, End B, etc. So these start/end positions should be programmed in increasing order. If any of the positions are programmed in non-increasing order (has invalid value) then the pulse generator will stop at the last valid position. Polarity adjustment is made prior to V display qualification.

This register specifies options for Horizontal pulse 0.

Bit	Description
11:8	**P_H_PULSE0_LAST: H Pulse 0 Last point.** 0 = Start_a 1 = End_a 2 = Start_b 3 = End_b 4 = Start_c 5 = End_c 6 = Start_d 7 = End_d
7:6	**P_H_PULSE0_V_QUAL: H Pulse 0 Vertical Qualifier.** 0 = Always 2 = Vactive 3 = Vactive1
4	**P_H_PULSE0_POLARITY: H Pulse 0 Polarity.** 0 = High 1 = Low
3	**P_H_PULSE0_MODE: H Pulse 0 Mode.** 0 = Normal 1 = One_clock

DC_P_P_H_PULSE0_POSITION_A_0

Offset: 10dh
Read/Write: R/W
Reset: 0000.0000

H Pulse 0 Position A

Bit	Description
27:16	**P_H_PULSE0_END_A: H Pulse 0 End A** (minimum P_H_PULSE0_START_A+1)
11:0	**P_H_PULSE0_START_A: H Pulse 0 Start A** (minimum 0)

DC_P_P_H_PULSE0_POSITION_B_0

Offset: 10eh
Read/Write: R/W
Reset: 0000.0000

H Pulse 0 Position B

Bit	Description
27:16	**P_H_PULSE0_END_B: H Pulse 0 End B** (minimum P_H_PULSE0_START_B+1)
11:0	**P_H_PULSE0_START_B: H Pulse 0 Start B** (minimum P_H_PULSE0_END_A+1)

DC_P_P_H_PULSE0_POSITION_C_0

Offset: 10fh
Read/Write: R/W
Reset: 0000.0000

H Pulse 0 Position C

Bit	Description
27:16	**P_H_PULSE0_END_C: H Pulse 0 End C** (minimum P_H_PULSE0_START_C+1)
11:0	**P_H_PULSE0_START_C: H Pulse 0 Start C** (minimum P_H_PULSE0_END_B+1)

DC_P_P_H_PULSE0_POSITION_D_0

Offset: 110h
Read/Write: R/W
Reset: 0000.0000

H Pulse 0 Position D

Bit	Description
27:16	**P_H_PULSE0_END_D: H Pulse 0 End D** (minimum P_H_PULSE0_START_D+1)
11:0	**P_H_PULSE0_START_D: H Pulse 0 Start D** (minimum P_H_PULSE0_END_C+1)

DC_P_P_H_PULSE1_CONTROL_0

Offset: 111h
Read/Write: R/W
Reset: 0000.0000

H Pulse 1 Control

Horizontal pulse 1 is programmable pulse that repeats every line. In the NORMAL mode, this signal can have several pulses (A to D) per line with programmable width as defined by the pairs of start and end positions. The pulses must not overlap and must occur in sequence: pulse A, then pulse B, etc.

In this case, the Enable field must be set to one of the End position. If the Enable field is set to one of the Start position then the pulse generator will stop as if the Enable field is set to the previous End position. If Enable field is set to Start A position then no pulse is generated.

In the ONE_CLOCK mode this signal can have up to twice the number of pulses per line with each pulse having a width of 1 pixel clock. In this mode, the position of the one-clock pulses correspond to the enabled Start and End positions. Regardless of the mode, the pulse generator processes the pairs of start and end position sequentially in the order of: Start A, End A, Start B, End B, etc. So these start/end positions should be programmed in increasing order. If any of the positions are programmed in non-increasing order (has invalid value) then the pulse generator will stop at the last valid position. Polarity adjustment is made prior to V display qualification.
This register specifies options for Horizontal pulse 1.

Bit	Description
11:8	**P_H_PULSE1_LAST: H Pulse 1 Last point.** 0 = Start_a 1 = End_a 2 = Start_b 3 = End_b 4 = Start_c 5 = End_c 6 = Start_d 7 = End_d
7:6	**P_H_PULSE1_V_QUAL: H Pulse 1 Vertical Qualifier.** 0 = ALWAYS 2 = VACTIVE 3 = VACTIVE1

Bit	Description
4	**P_H_PULSE1_POLARITY: H Pulse 1 Polarity.** 0 = High 1 = Low
3	**P_H_PULSE1_MODE: H Pulse 1 Mode.** 0 = Normal 1 = One_clock

DC_P_P_H_PULSE1_POSITION_A_0

Offset: 112h
Read/Write: R/W
Reset: 0000.0000

H Pulse 1 Position A

Bit	Description
27:16	**P_H_PULSE1_END_A: H Pulse 1 End A** (minimum P_H_PULSE1_START_A+1)
11:0	**P_H_PULSE1_START_A: H Pulse 1 Start A** (minimum 0)

DC_P_P_H_PULSE1_POSITION_B_0

Offset: 113h
Read/Write: R/W
Reset: 0000.0000

H Pulse 1 Position B

Bit	Description
27:16	**P_H_PULSE1_END_B: H Pulse 1 End B** (minimum P_H_PULSE1_START_B+1)
11:0	**P_H_PULSE1_START_B: H Pulse 1 Start B** (minimum P_H_PULSE1_END_A+1)

DC_P_P_H_PULSE1_POSITION_C_0

Offset: 114h
Read/Write: R/W
Reset: 0000.0000

H Pulse 1 Position C

Bit	Description
27:16	**P_H_PULSE1_END_C: H Pulse 1 End C** (minimum P_H_PULSE1_START_C+1)
11:0	**P_H_PULSE1_START_C: H Pulse 1 Start C** (minimum P_H_PULSE1_END_B+1)

DC_P_P_H_PULSE1_POSITION_D_0

Offset: 115h
Read/Write: R/W
Reset: 0000.0000

H Pulse 1 Position D

Bit	Description
27:16	**P_H_PULSE1_END_D: H Pulse 1 End D** (minimum P_H_PULSE1_START_D+1)
11:0	**P_H_PULSE1_START_D: H Pulse 1 Start D** (minimum P_H_PULSE1_END_C+1)

DC_P_P_H_PULSE2_CONTROL_0

Offset: 116h
Read/Write: R/W
Reset: 0000.0000

H Pulse 2 Control

Horizontal pulse 2 is programmable pulse that repeats every line. In the NORMAL mode, this signal can have several pulses (A to D) per line with programmable width as defined by the pairs of start and end positions. The pulses must not overlap and must occur in sequence: pulse A, then pulse B, etc.

In this case, the Enable field must be set to one of the End position. If the Enable field is set to one of the Start position then the pulse generator will stop as if the Enable field is set to the previous End position. If Enable field is set to Start A position then no pulse is generated.

In the ONE_CLOCK mode this signal can have up to twice the number of pulses per line with each pulse having a width of 1 pixel clock. In this mode, the position of the one-clock pulses correspond to the enabled Start and End positions.

Regardless of the mode, the pulse generator processes the pairs of start and end position sequentially in the order of: Start A, End A, Start B, End B, etc. So these start/end positions should be programmed in increasing order. If any of the positions are programmed in non-increasing order (has invalid value) then the pulse generator will stop at the last valid position. Polarity adjustment is made prior to V display qualification.

This register specifies options for Horizontal pulse 2.

Bit	Description
11:8	**P_H_PULSE2_LAST: H Pulse 2 Last point.** 0 = Start_a 1 = End_a 2 = Start_b 3 = End_b 4 = Start_c 5 = End_c 6 = Start_d 7 = End_d
7:6	**P_H_PULSE2_V_QUAL: H Pulse 2 Vertical Qualifier.** 0 = Always 2 = Vactive 3 = Vactive1

Bit	Description
4	**P_H_PULSE2_POLARITY: H Pulse 2 Polarity.** 0 = High 1 = Low
3	**P_H_PULSE2_MODE: H Pulse 2 Mode.** 0 = Normal 1 = One_clock

DC_P_P_H_PULSE2_POSITION_A_0

Offset: 117h
Read/Write: R/W
Reset: 0000.0000

H Pulse 2 Position A

Bit	Description
27:16	**P_H_PULSE2_END_A: H Pulse 2 End A** (minimum P_H_PULSE2_START_A+1)
11:0	**P_H_PULSE2_START_A: H Pulse 2 Start A** (minimum 0)

DC_P_P_H_PULSE2_POSITION_B_0

Offset: 118h
Read/Write: R/W
Reset: 0000.0000

H Pulse 2 position B

Bit	Description
27:16	**P_H_PULSE2_END_B: H Pulse 2 End B** (minimum P_H_PULSE2_START_B+1)
11:0	**P_H_PULSE2_START_B: H Pulse 2 Start B** (minimum P_H_PULSE2_END_A+1)

DC_P_P_H_PULSE2_POSITION_C_0

Offset: 119h
Read/Write: R/W
Reset: 0000.0000

H Pulse 2 Position C

Bit	Description
27:16	**P_H_PULSE2_END_C: H Pulse 2 End C** (minimum P_H_PULSE2_START_C+1)
11:0	**P_H_PULSE2_START_C: H Pulse 2 Start C** (minimum P_H_PULSE2_END_B+1)

DC_P_P_H_PULSE2_POSITION_D_0

Offset: 11ah
Read/Write: R/W
Reset: 0000.0000

H Pulse 2 Position D

Bit	Description
27:16	**P_H_PULSE2_END_D: H Pulse 2 End D** (minimum P_H_PULSE2_START_D+1)
11:0	**P_H_PULSE2_START_D: H Pulse 2 Start D** (minimum P_H_PULSE2_END_C+1)

DC_P_P_V_PULSE0_CONTROL_0

Offset: 11bh
Read/Write: R/W
Reset: 0000.0000

V Pulse 0 Control

Vertical pulse 0 is programmable pulse that repeats every frame. This signal can have several pulses (A to C) per frame with programmable width as defined by the pairs of start and end positions. The pulses must not overlap and must occur in sequence: pulse A, then pulse B, etc.

In this case, the Enable field must be set to one of the End position. If the Enable field is set to one of the Start position then the pulse generator will stop as if the Enable field is set to the previous End position. If Enable field is set to Start A position then no pulse is generated.

The pulse generator processes the pairs of start and end position sequentially in the order of: Start A, End A, Start B, End B, etc. So these start/end positions should be programmed in increasing order. If any of the positions are programmed in non-increasing order (has invalid value) then the pulse generator will stop at the last valid position.

This register specifies options for Vertical pulse 0.

Bit	Description
11:8	**P_V_PULSE0_LAST: V Pulse 0 Last point.** 0 = Start_a 1 = End_a 2 = Start_b 3 = End_b 4 = Start_c 5 = End_c
7:6	**P_V_PULSE0_DELAY: V Pulse 0 Delay.** 0 = Nodelay 1 = Delay1 2 = Delay2
4	**P_V_PULSE0_POLARITY: V Pulse 0 Polarity.** 0 = High 1 = Low

DC_P_P_V_PULSE0_POSITION_A_0

Offset: 11ch
Read/Write: R/W
Reset: 0000.0000

V Pulse 0 Position A

Bit	Description
27:16	**P_V_PULSE0_END_A: V Pulse 0 End A** (minimum P_V_PULSE0_START_A+1)
11:0	**P_V_PULSE0_START_A: V Pulse 0 Start A** (minimum 0)

DC_P_P_V_PULSE0_POSITION_B_0

Offset: 11dh
Read/Write: R/W
Reset: 0000.0000

V Pulse 0 Position B

Bit	Description
27:16	**P_V_PULSE0_END_B: V Pulse 0 End B** (minimum P_V_PULSE0_START_B+1)
11:0	**P_V_PULSE0_START_B: V Pulse 0 Start B** (minimum P_V_PULSE0_END_A+1)

DC_P_P_V_PULSE0_POSITION_C_0

Offset: 11eh
Read/Write: R/W
Reset: 0000.0000

V Pulse 0 Position C

Bit	Description
27:16	**P_V_PULSE0_END_C: V Pulse 0 End C** (minimum P_V_PULSE0_START_C+1)
11:0	**P_V_PULSE0_START_C: V Pulse 0 Start C** (minimum P_V_PULSE0_END_B+1)

DC_P_P_V_PULSE1_CONTROL_0

Offset: 11fh
Read/Write: R/W
Reset: 0000.0000

V pulse 1 Control

Vertical pulse 1 is programmable pulse that repeats every frame. This signal can have several pulses (A to C) per frame with programmable width as defined by the pairs of start and end positions. The pulses must not overlap and must occur in sequence: pulse A, then pulse B, etc.

In this case, the Enable field must be set to one of the End position. If the Enable field is set to one of the Start position then the pulse generator will stop as if the Enable field is set to the previous End position. If Enable field is set to Start A position then no pulse is generated.

The pulse generator processes the pairs of start and end position sequentially in the order of: Start A, End A, Start B, End B, etc. So these start/end positions should be programmed in increasing order. If any of the positions are programmed in non-increasing order (has invalid value) then the pulse generator will stop at the last valid position.

This register specifies options for Vertical pulse 1.

Bit	Description
11:8	**P_V_PULSE1_LAST: V pulse 1 Last point.** 0 = Start_a 1 = End_a 2 = Start_b 3 = End_b 4 = Start_c 5 = End_c
7:6	**P_V_PULSE1_DELAY: V pulse 1 Delay.** 0 = Nodelay 1 = Delay1 2 = Delay2
4	**P_V_PULSE1_POLARITY: V pulse 1 Polarity.** 0 = High 1 = Low

DC_P_P_V_PULSE1_POSITION_A_0

Offset: 120h
Read/Write: R/W
Reset: 0000.0000

V Pulse 1 Position A

Bit	Description
27:16	**P_V_PULSE1_END_A: V Pulse 1 End A** (minimum P_V_PULSE1_START_A+1)
11:0	**P_V_PULSE1_START_A: V Pulse 1 Start A** (minimum 0)

DC_P_P_V_PULSE1_POSITION_B_0

Offset: 121h
Read/Write: R/W
Reset: 0000.0000

V Pulse 1 Position B

Bit	Description
27:16	**P_V_PULSE1_END_B: V Pulse 1 End B** (minimum P_V_PULSE1_START_B+1)
11:0	**P_V_PULSE1_START_B: V Pulse 1 Start B** (minimum P_V_PULSE1_END_A+1)

DC_P_P_V_PULSE1_POSITION_C_0

Offset: 122h
Read/Write: R/W
Reset: 0000.0000

V Pulse 1 Position C

Bit	Description
27:16	**P_V_PULSE1_END_C: V Pulse 1 End C** (minimum P_V_PULSE1_START_C+1)
11:0	**P_V_PULSE1_START_C: V Pulse 1 Start C** (minimum P_V_PULSE1_END_B+1)

DC_P_P_V_PULSE2_CONTROL_0

Offset: 123h
Read/Write: R/W
Reset: 0000.0000

V pulse 2 Control

Vertical pulse 2 is programmable pulse that repeats every frame. This signal can have one pulse (A) per frame with programmable width as defined by the pair of start and end positions.

In this case, the Enable field must be set to one of the End position. If the Enable field is set to one of the Start position then the pulse generator will stop as if the Enable field is set to the previous End position. If Enable field is set to Start A position then no pulse is generated.

So these start/end positions should be programmed in increasing order. If any of the positions are programmed in non-increasing order (has invalid value) then the pulse generator will stop at the last valid position.

This register specifies options for Vertical pulse 2.

Bit	Description
8	**P_V_PULSE2_LAST: V pulse 2 Last point.** 0 = Start_a 1 = End_a
4	**P_V_PULSE2_POLARITY: V pulse 2 Polarity.** 0 = High 1 = Low

DC_P_P_V_PULSE2_POSITION_A_0

Offset: 124h
Read/Write: R/W
Reset: 0000.0000

V Pulse 2 Position A

Bit	Description
27:16	**P_V_PULSE2_END_A: V Pulse 2 End A** (minimum P_V_PULSE2_START_A+1)
11:0	**P_V_PULSE2_START_A: V Pulse 2 Start A** (minimum 0)

DC_P_P_V_PULSE3_CONTROL_0

Offset: 125h
Read/Write: R/W
Reset: 0000.0000

V pulse 3 Control

Vertical pulse 3 is programmable pulse that repeats every frame. This signal can have one pulse (A) per frame with programmable width as defined by the pair of start and end positions.

In this case, the Enable field must be set to one of the End position. If the Enable field is set to one of the Start position then the pulse generator will stop as if the Enable field is set to the previous End position. If Enable field is set to Start A position then no pulse is generated.

So these start/end positions should be programmed in increasing order. If any of the positions are programmed in non-increasing order (has invalid value) then the pulse generator will stop at the last valid position.

This register specifies options for Vertical pulse 2.

Bit	Description
8	**P_V_PULSE3_LAST: V pulse 3 Last point.** 0 = Start_a 1 = End_a
4	**P_V_PULSE3_POLARITY: V pulse 3 Polarity.** 0 = High 1 = Low

DC_P_P_V_PULSE3_POSITION_A_0

Offset: 126h
Read/Write: R/W
Reset: 0000.0000

V Pulse 3 Position A

Bit	Description
27:16	**P_V_PULSE3_END_A: V Pulse 3 End A** (minimum P_V_PULSE3_START_A+1)
11:0	**P_V_PULSE3_START_A: V Pulse 3 Start A** (minimum 0)

DC_P_P_M0_CONTROL_0

Offset: 127h
Read/Write: R/W
Reset: 0000.0000

M0 Control

Display M0 signal

M0 signal can be generated either using a line (horizontal) or a frame (vertical) clock and it can be horizontally positioned with respect to H reference point. This signal is typically output on LM0 pin.

This register specifies options for M0 signal.

Bit	Description
27:16	**P_M0_H_POSITION: M0 Horizontal Position** This parameter specifies the position where M0 can toggle with respect to H reference point.
12:8	**P_M0_PERIOD: M0 Period** This should be program to the half of the desired M0 period (in lines) minus 1.
7	**P_M0_POLARITY: M0 Polarity.** 0 = High 1 = Low
6	**P_M0_PHASE_RESET: M0 Phase Reset.** 0 = Not_reset 1 = Reset
5:4	**P_M0_PHASE_CONTROL: M0 Phase Control.** 0 = Free_run 2 = Vactive_restart 3 = Frame_invert
1:0	**P_M0_CLOCK_SELECT: M0 Clock Select.** 0 = PCLK 2 = LCLK 3 = FCLK

DC_P_P_M1_CONTROL_0

Offset: 128h
Read/Write: R/W
Reset: 0000.0000

M1 Control

Display M1 signal

M1 signal can be generated either using a line (horizontal) or a frame (vertical) clock and it can be horizontally positioned with respect to H reference point. This signal is typically output on LM1 pin. This register specifies options for M1 signal.

Bit	Description
27:16	**P_M1_H_POSITION: M1 Horizontal Position** This parameter specifies the position where M0 can toggle with respect to H reference point.
12:8	**P_M1_PERIOD: M1 Period** This should be program to the half of the desired M1 period (in lines) minus 1.

Bit	Description
7	**P_M1_POLARITY: M1 Polarity.** 0 = High 1 = Low
6	**P_M1_PHASE_RESET: M1 Phase Reset.** 0 = Not_reset 1 = Reset
5:4	**P_M1_PHASE_CONTROL: M1 Phase Control.** 0 = Free_run 2 = Vactive_restart 3 = Frame_invert
1:0	**P_M1_CLOCK_SELECT: M1 Clock Select.** 0 = PCLK 1 = M0SYNC 2 = LCLK 3 = FCLK

DC_P_P_DI_CONTROL_0

Offset: 129h
Read/Write: R/W
Reset: 0000.0000

DI Control

Display Data Inversion (DI) signal generation

This signal is typically needed to control data inversion for PWM panels and is typically output on LDI pin. Horizontal position of this signal with respect to horizontal reference point can be programmed.

DI signal together with M0 may also be used to control the actual pixel data inversion. Pixel data may be controlled by either DI only or by (DI ^ M0) as specified by P_PIXDATA_INV_SELECT. The inversion control signal is then used to control pixel data inversion as specified by P_PIXDATA_INV_CONTROL. Note that even if the DI signal is disabled, pixel data inversion could still occur depending on the setting of P_PIXDATA_INV_CONTROL. Data inversion is limited to only active area. For the purpose of pixel data inversion, DI and M0 signals are used before the corresponding horizontal positioning so that these signals are always stable during active area.

In case M0 signal is used to control data inversion then it should be generated using line clock. M0 polarity control is not accounted when M0 is used to generate DI signal or to control pixel data inversion.

This register specifies options for DI signal as well as pixel data inversion.

Bit	Description
27:16	**P_DI_H_POSITION: DI signal Horizontal Position** This parameter specifies the position where DI signal can toggle with respect to H reference point. It should not be programmed larger than P_PP_H_POSITION if DI is used to control PP signal generation.
7:6	**P_PIXDATA_INV_CONTROL: Pixel Data Inversion Control.** 0 = NOINV 1 = EVENINV 2 = ODDINV 3 = ALLINV

Bit	Description
4	**P_PIXDATA_INV_SELECT: Pixel Data Inversion Select.** 0 = DI 1 = DIXORM0
1:0	**P_DI_MODE: DI signal Mode** 00= DI is always low 01= DI is always high 10= DI is forced high every time M0 (before polarity adjustment) toggles from low to high; otherwise then DI toggles every line 11= DI has same frequency (phase) as M0 (before M0 polarity adjustment)

DC_P_P_PP_CONTROL_0

Offset: 12ah
Read/Write: R/W
Reset: 0000.0000

PP Control

Display Programmable Pulse (PP) signal generation

PP signal generation logic can generate up to 128 pulses per line internally and the PP pulse select registers determines which of the 128 pulses will be output. Any of the 128 internally generated pulse can be independently selected as output if they occur within one line time.

PP signal is typically output on LPP pin. Note that DI signal may impact PP generation as controlled by P_PP_REVERSAL_CONTROL. PP signal generation may be delayed (positioned) from H reference point (line start) controlled by P_PP_H_DELAY. Delaying PP may cause the last few internal PP pulses to overflow to the next line.

PP is always generated using the display clock after the shift clock divider. This register specifies options for PP signal.

Bit	Description
15:12	**P_PP_LOW_PULSE: PP Low Pulse width (1 to 16)**
11:8	**P_PP_HIGH_PULSE: PP High Pulse width (1 to 16)**
7:4	**P_PP_H_DELAY: PP signal Horizontal Delay (0 to 15)** This parameter specifies the position where PP signal generation starts with respect to H reference point. If DI is used to generate PP signal then this parameter should not be smaller than P_DI_H_POSITION.
3:2	**P_PP_V_QUALIFIER: PP Vertical Qualifier.** 0 = Free_run 1 = Vpulse1 2 = Vpulse2 3 = Vpulse3
1:0	**P_PP_DIRECTION: PP Direction (incrementing or decrementing).** 0 = Always_inc 1 = Inc_if_di0 2 = Dec_if_di0 3 = Always_dec

DC_P_P_PP_SELECT_A_0

Offset: 12bh
Read/Write: R/W
Reset: 0000.0000

PP Select A

The next 4 registers and P_PP_DIRECTION which of the internal 128 pulses to be output. Each bit in the four registers correspond to one internal pulse.

Bit	Description
31:0	**P_PP_SELECT_A: PP Select bits 31 to 0**

DC_P_P_PP_SELECT_B_0

Offset: 12ch
Read/Write: R/W
Reset: 0000.0000

PP Select B

Bit	Description
31:0	**P_PP_SELECT_B: PP Select bits 63 to 32**

DC_P_P_PP_SELECT_C_0

Offset: 12dh
Read/Write: R/W
Reset: 0000.0000

PP Select C

Bit	Description
31:0	**P_PP_SELECT_C: PP Select bits 95 to 64**

DC_P_P_PP_SELECT_D_0

Offset: 12eh
Read/Write: R/W
Reset: 0000.0000

PP Select D

Bit	Description
31:0	**P_PP_SELECT_D: PP Select bits 127 to 96**

DC_P_P_DISP_CLOCK_CONTROL_0

Offset: 12fh
Read/Write: R/W
Reset: 0000.0006

Display Clock Control

Shift clock divider is used to divide root clock for display controller module to generate internal shift clock for shifting data to the display. Output of this divider is typically used to generate the external shift clock which is sent to the display (SC0 and/or SC1) except for 1-pixel/1-clock parallel display. The output of this divider is also used to generate Programmable Pulse (PP) signal. For 1-pixel/1-clock parallel display, SC0 and SC1 are generated using the output of pixel clock divider which can be set to 1, 2, or 4 for 1-pixel/1-clock parallel display. The reason pixel clock divider 2 and 4 are allowed for 1-pixel/1-clock parallel display interface is so that the clock that generates PP can be generated with 2x or 4x higher frequency than pixel clock and therefore can produce higher resolution PP pulse positions. For all cases of parallel display, SC0 and SC1 can be further divided by 1, 2 or 4.

Class: Display Interface Settings

This register controls generation of shift clock to the display and internal pixel clock. Internal display pipeline runs with pixel clock and processes 1 pixel per clock.

Bit	Description
11:8	**P_PIXEL_CLK_DIVIDER: Pixel Clock Divider.** 0 = PCD1 1 = PCD1H 2 = PCD2 3 = PCD3 4 = PCD4 5 = PCD6 6 = PCD8 7 = PCD9 8 = PCD12 9 = PCD16 10 = PCD18 11 = PCD24

Bit	Description
7:0	**P_SHIFT_CLK_DIVIDER: Shift Clock Divider** 0 = divide by 1 1 = divide by 1.5 2 = divide by 2 3 = divide by 2.5 4 = divide by 3 : : : : 254 = divide by 128 255 = divide by 128.5 Pixel clock divider is used to divide output of internal shift clock divider to generate internal pixel clock which is used to clock the internal horizontal and vertical counters. This divider also determine the output format for parallel interface, serial interface, and LCD SPI interface in conjunction with Display Data Format parameter. For 1-pixel/1-clock parallel display interface, valid settings are PCD1, PCD2, and PCD4. Note that the main reason to use PCD2 and PCD4 is to get higher frequency PP clock because the PP clock is always generated from the output of shift clock divider. For non 1-pixel/1-clock parallel display interface, valid settings are, PCD1H (2-pixel/3-clock), PCD2 (1-pixel/2-clock), and PCD3 (1-pixel/3-clock). For 1-channel serial display interface, valid settings are PCD3 (3-bpp 1-ch), PCD4 (3-bpp 1-ch), PCD6 (6-bpp 1-ch), PCD9 (9-bpp 1-ch), PCD12 (12-bpp 1-ch), PCD16 (16-bpp 1-ch), PCD18 (18-bpp 1-ch). For 2-channel serial display interface, valid settings are PCD2 (3-bpp 2-ch), PCD3 (6-bpp 2-ch), PCD6 (12-bpp 2-ch), PCD8 (16-bpp 2-ch), PCD9 (18-bpp 2-ch). For 3-channel serial display interface, valid settings are PCD1 (3-bpp 3-ch), PCD2 (6-bpp 3-ch), PCD3 (9-bpp 3-ch), PCD4 (12-bpp 3-ch), PCD6 (18-bpp 3-ch). For LCD SPI interface, valid settings are PCD12 (B4G4R4), PCD16 (B5G6R5), PCD18 (B6G6R6), PCD24 (B8G8R8), PCD8 (B5G6R5 with data/command bit), and PCD6 (B5G6R5 with data/command start byte - depending on data/command bit).

DC_P_P_DISP_INTERFACE_CONTROL_0

Offset: 130h
Read/Write: R/W
Reset: 0000.0000

Display Interface Control

This register specifies display interface options

Bit	Description
9	**P_DISP_DATA_ORDER: Display Data Order.** 0 = RED_BLUE 1 = BLUE_RED
8	**P_DISP_DATA_ALIGNMENT: Display Data Alignment.** 0 = MSB 1 = LSB
3:0	**P_DISP_DATA_FORMAT: Display Data Format.** 0 = DF1P1C 1 = DF1P2C24B 2 = DF1P2C18B 3 = DF1P2C16B 4 = DF1S 5 = DF2S 6 = DF3S 7 = DFSPI

DC_P_P_DISP_COLOR_CONTROL_0

Offset: 131h
Read/Write: R/W
Reset: 0000.0000

Display Color Control

Bit	Description
27	**P_LCD_MD3: LCD Mode 3 signal.** 0 = Low 1 = High
26	**P_LCD_MD2: LCD Mode 2 signal.** 0 = Low 1 = High
25	**P_LCD_MD1: LCD Mode 1 signal.** 0 = Low 1 = High
24	**P_LCD_MD0: LCD Mode 0 signal.** 0 = Low 1 = High
18	**P_NON_BASE_COLOR: Non Base Color** 0= zeros 1= ones MD0-3 signals are general purpose mode signals that can be output in various pins (see Pin Output Select) to configure the display device. These bits are effective at start of frame. Typically these can be changed also when changing between display register set (Primary, Secondary, etc).
17	**P_BLANK_COLOR: Blank Color** 0= zeros 1= ones Non Base Color applies to least significant color bits which are not part of base color and it has higher priority over Border Color but lower priority over Blank color.
16	**P_DISP_COLOR_SWAP: Display Color Swap.** 0 = RGB 1 = BGR
13:12	**P_ORD_DITHER_ROTATION: Ordered Dither Frame Rotation** This parameter specifies the rotation frequency of the dither matrix in terms of number of frames. If programmed to 0, there is no dither matrix rotation. If programmed to N where N is larger than 0, the dither matrix is rotated clockwise every N frame.
9:8	**P_DITHER_CONTROL: Dither Control.** 0 = Disable 2 = Ordered 3 = Errdiff
3:0	**P_BASE_COLOR_SIZE: Display Base Color Size.** 0 = Base666 1 = Base111 2 = Base222 3 = Base333 4 = Base444 5 = Base555 6 = Base565 7 = Base332 8 = Base888

DC_P_P_SHIFT_CLOCK_OPTIONS_0

Offset: 132h
Read/Write: R/W
Reset: 0000.0000

Shift Clock options

This register specifies options for both display shift clock 0 (SC0) and display shift clock 1 (SC1). SC0 signal is typically output on LSC0 pin and SC1 signal is typically output on LSC1 pin.

Bit	Description
23:22	**P_SC1_CLK_DIVIDER: SC1 Clock Divider.** 0 = Div1 1 = Div2 2 = Div4
21:19	**P_SC1_V_QUALIFIER: SC1 Vertical Qualifier.** 0 = No_vqual 1 = Reserved 2 = Vactive 3 = Ext_vactive 4 = Vpulse1 5 = Ext_vpulse1
18:16	**P_SC1_H_QUALIFIER: SC1 Horizontal Qualifier.** 0 = Disable 1 = No_hqual 2 = Hactive 3 = Ext_hactive 4 = Hpulse1 5 = Ext_hpulse1
7:6	**P_SC0_CLK_DIVIDER: SC0 Clock Divider.** 0 = Div1 1 = Div2 2 = Div4
5:3	**P_SC0_V_QUALIFIER: SC0 Vertical Qualifier.** 0 = No_vqual 1 = Reserved 2 = Vactive 3 = Ext_vactive 4 = Vpulse0 5 = Ext_vpulse0
2:0	**P_SC0_H_QUALIFIER: SC0 Horizontal Qualifier.** 0 = Disable 1 = No_hqual 2 = Hactive 3 = Ext_hactive 4 = Hpulse0 5 = Ext_hpulse0

DC_P_P_DATA_ENABLE_OPTIONS_0

Offset: 133h
Read/Write: R/W
Reset: 0000.0000

Data Enable options

DE signal is display Data Enable signal which can be used to indicate valid data area and it can be output on LSC1 pin if needed.

Bit	Description
3:2	**P_DE_CONTROL: DE (Data Enable) Control.** 0 = Oneclk 1 = Normal 2 = Early_ext 3 = Early
0	**P_DE_SELECT: DE (Data Enable) Select.** 0 = Active_blank 1 = Active

DC_P_P_SERIAL_INTERFACE_OPTIONS_0

Offset: 134h
Read/Write: R/W
Reset: 0000.0000

Serial display interface options

Serial display interface control signals

Controls signals for the low-voltage differential serial display interface consists of SDT, STP and STH signals. SDT and STP are asserted high if current pixel is same as previous pixel; in this case, SDT is toggled low sometime later but STP is either toggled low at same time as SDT (if next pixel is different than current pixel) or remains high if next pixel is same as current pixel.

When doing pixel comparison, output of dither is used, so pixel comparison depends on the base color (which maybe different than the number of output data bits). Both SDT and STP are always low (disabled) if the pixel clock divider is 4 or less. STH is used to indicate the beginning of line and it is asserted high once at the beginning of each line. The STH pulse exact timing width is dependent on the exact mode. STH is generated from Data Enable therefore Data Enable Select bit also controls STH generation and can be used to generate STH either only for active lines or both for active and blank lines. If STH is sent during blank lines then the blank lines are also transmitted.

Bit	Description
7	**P_STP_CONTROL: STP signal control.** 0 = Normal 1 = Extended
6	**P_STH_DURATION: STH signal duration.** 0 = One_clock 1 = Two_clock

Bit	Description
5:2	**P_SDT_STP_DURATION: SDT and STP signal duration** 0= 1 shift clock 1= 1 pixel clock 2= 1 pixel clock - 1 shift clock 3= 1 pixel clock - 2 shift clock 4= 1 pixel clock - 3 shift clock 5= 1 pixel clock - 4 shift clock : : : F= 1 pixel clock - 14 shift clock STP active duration is same as SDT if next pixel is not the same as current pixel; else, STP active duration is always 1 pixel **c**lock. Maximum valid setting is pixel clock divider - 1 for pixel clock divider > 4. If pixel clock divider is 4 or less, SDT and STP is always low.
1:0	**P_SDT_STP_MODE: SDT and STP modes.** 0 = Disable 1 = Reserved 2 = Enable_dup 3 = Enable

DC_P_P_LCD_SPI_OPTIONS_0

Offset: 135h
Read/Write: R/W
Reset: 0000.0000

LCD SPI interface options

LCD SPI interface signals consists of:
1. SPI Clock (SCK) which can be output on LSCK pin.
2. SPI Data (SDA) which can be output on LSDA pin.
3. Optional SPI Data/Command (SDC) which can be output on LDC pin.
4. Main-Display SPI Chip Select (Main SCS_) signal which can be output on LCS_ pin.
5. Sub-Display SPI Chip Select (Sub SCS_) signal which can be optionally output on several pins (see pin output select) - this is optional and it is used only if there is a sub display.

For LCD SPI, pixel data can only be sent to either Main-Display or Sub-Display but not to both. Main SCS_ or Sub SCS_ signal is always active low and is typically controlled by SPI logic but can also be forced active one line prior to display active (for SIS SPI) and during vertical display active area (for LCD SPI).

Bit	Description
4	**P_LCD_SPI_DIRECTION: LCD SPI Data Direction.** 0 = MSB2LSB 1 = LSB2MSB
3:2	**P_SPI_CS_CONTROL: LCD SPI Chip Select (SCS_) Control for both IS SPI or LCD SPI.** 0 = LCD_IS_SPI 1 = LCD_SPI 2 = IS_SPI 3 = FORCED
1	**P_LCD_SPI_DC: LCD SPI Data/Command (SDC).** 0 = Low 1 = High
0	**P_LCD_SPI_CS: LCD SPI Chip Select (SCS_).** 0 = Main 1 = Sub

DC_P_P_BORDER_COLOR_0

Offset: 136h
Read/Write: R/W
Reset: 0000.0000

Border Color

Border Color defines the color of areas within the active display area which are outside the defined active windows. This is 24-bit color which is applied after blending.

Bit	Description
23:16	**P_BORDER_COLOR_B: Blue Border Color**
15:8	**P_BORDER_COLOR_G: Green Border Color**
7:0	**P_BORDER_COLOR_R: Red Border Color**

DC_P_P_COLOR_KEY0_LOWER_0

Offset: 137h
Read/Write: R/W
Reset: 0000.0000

Color Key 0 Lower value

Color Key 0 and Color Key 1

Two ranges of color key are defined and they are common for all windows because it is expected that typically only one window will have color key enabled. Because there are two sets of color key, it is possible to have 2 windows each using one color key set. Usage of this color key is described in the Display Color Key and Blending class.

Bit	Description
23:16	**P_COLOR_KEY0_L_B: Color Key 0 Blue (U) Lower value**
15:8	**P_COLOR_KEY0_L_G: Color Key 0 Green (Y) Lower value**
7:0	**P_COLOR_KEY0_L_R: Color Key 0 Red (V) Lower value**

DC_P_P_COLOR_KEY0_UPPER_0

Offset: 138h
Read/Write: R/W
Reset: 0000.0000

Color Key 0 Upper Value

Bit	Description
23:16	**P_COLOR_KEY0_U_B: Color Key 0 Blue (U) Upper value**
15:8	**P_COLOR_KEY0_U_G: Color Key 0 Green (Y) Upper value**
7:0	**P_COLOR_KEY0_U_R: Color Key 0 Red (V) Upper value**

DC_P_P_COLOR_KEY1_LOWER_0

Offset: 139h
Read/Write: R/W
Reset: 0000.0000

Color Key 0 Lower Value

Bit	Description
23:16	**P_COLOR_KEY1_L_B: Color Key 0 Blue (U) Lower value**
15:8	**P_COLOR_KEY1_L_G: Color Key 0 Green (Y) Lower value**
7:0	**P_COLOR_KEY1_L_R: Color Key 0 Red (V) Lower value**

DC_P_P_COLOR_KEY1_UPPER_0

Offset: 13ah
Read/Write: R/W
Reset: 0000.0000

Color Key 0 Upper value

Bit	Description
23:16	**P_COLOR_KEY1_U_B: Color Key 0 Blue (U) Upper value**
15:8	**P_COLOR_KEY1_U_G: Color Key 0 Green (Y) Upper value**
7:0	**P_COLOR_KEY1_U_R: Color Key 0 Red (V) Upper value**

DC_P_P_G_POSITION_0

Offset: 13bh
Read/Write: R/W
Reset: 0000.0000

Window G Position

Class: Display Window G

Display Window G parameters: Window G defines part of display active area which can be sent to encoder pre-processor (EPP) module. Pixel data sent to the EPP is in RGB888 format and this data can be converted by the EPP to planar YUV format for encoding.

Bit	Description
27:16	**P_G_V_POSITION: Window G Vertical Position** This is specified with respect to the top edge of active display area.
11:0	**P_G_H_POSITION: Window G Horizontal Position** This is specified with respect to the left edge of active display area.

DC_P_P_G_SIZE_0

Offset: 13ch
Read/Write: R/W
Reset: 0000.0000

Window G Size

Bit	Description
27:16	**P_G_V_SIZE: Window G Vertical Size (lines)** This is the vertical size after scaling.
11:0	**P_G_H_SIZE: Window G Horizontal Size (pixels)** This is the horizontal size after scaling.

DC_P_P_CURSOR_FOREGROUND_0

Offset: 13dh
Read/Write: R/W
Reset: 0000.0000

Cursor Foreground color

Class: Hardware Cursor

Hardware cursor is supported for 32x32 or for 64x64 2-bpp cursor. Cursor start address is aligned to 1 KB boundary. All cursor registers except for cursor foreground and background colors are shadowed twice.

The first shadow registers are written after the last byte of D1PCP register is written. The second shadow registers are written at frame boundary. In this way, cursor registers except for cursor foreground and background colors are not effective until the next frame boundary.

When cursor registers are read by host, the value of the first stage registers are returned. Cursor scaling and flipping are not implemented so this must be done by software if needed. Cursor H/V positions are signed number with respect to one of the display windows or with respect to upper left position of display active area as specified by cursor clipping parameter which also determine cursor clipping boundary. If cursor position is with respect to one of the display window and the corresponding display window is disabled then cursor will also be disabled.

Bit	Description
23:16	**P_CURSOR_FOREGROUND_B: Cursor Blue Foreground Color**
15:8	**P_CURSOR_FOREGROUND_G: Cursor Green Foreground Color**
7:0	**P_CURSOR_FOREGROUND_R: Cursor Red Foreground Color**

DC_P_P_CURSOR_BACKGROUND_0

Offset: 13eh
Read/Write: R/W
Reset: 0000.0000

Cursor Background color

Bit	Description
23:16	**P_CURSOR_BACKGROUND_B: Cursor Blue Background Color**
15:8	**P_CURSOR_BACKGROUND_G: Cursor Green Background Color**
7:0	**P_CURSOR_BACKGROUND_R: Cursor Red Background Color**

DC_P_P_CURSOR_START_ADDR_0

Offset: 13fh
Read/Write: R/W
Reset: 0000.0000

Cursor Start Address

Bit	Description
29:28	**P_CURSOR_CLIPPING: Cursor Clipping Select.** 0 = DISPLAY 1 = WA 2 = WB 3 = WC
24	**P_CURSOR_SIZE: Cursor Size.** 0 = C32X32 1 = C64X64
15:0	**P_CURSOR_START_ADDR: Cursor Start Address bits 19:10**

DC_P_P_CURSOR_POSITION_0

Offset: 140h
Read/Write: R/W
Reset: 0000.0000

Cursor Position
Cursor position is with respect to top-left corner of display active area, or window A, or window B, or window C as specified cursor clipping parameter.

Bit	Description
28:16	**P_V_CURSOR_POSITION: V cursor position (signed)**
12:0	**P_H_CURSOR_POSITION: H cursor position (signed)**

DC_P_P_INIT_SEQ_CONTROL_0

Offset: 141h
Read/Write: R/W
Reset: 0000.0000

Initialization Sequence Control

Class: Initialization Sequence (IS)

Display initialization sequence may have to be written to the display if the display has built-in frame buffer. This initialization sequence is used typically to reinitialize the display buffer start address and maybe needed once per frame (frame initialization sequence) and/or once per line (line initialization sequence). Frame initialization sequence is sent during the horizontal active time of the line just before the first active display line.

Line initialization sequence is currently NOT supported.

Initialization sequence can be done through parallel LCD interface or through SPI serial interface. Software is responsible in making sure that the active line time is sufficient to send initialization sequence. For parallel interface initialization, the signals used as chip selects (typically these are one of the vertical signals) must be programmed to be active one line just before the first active display line. Also SC0/SC1 clock divider must be programmed to divide by 1 if initialization sequence is enabled.

Bit	Description
11:8	**P_FRAME_INIT_SEQ_CYCLES: Frame Initialization Sequence Cycles** This parameter specifies the number of frame initialization sequence cycles to send. If programmed to 0, there is no frame initialization cycle generated.
7	**P_INIT_SEQ_DC_CONTROL: Initialization Sequence DC Pin** This bit is used only for parallel initialization sequence and it controls how data/command is added to the vertical signal selected by P_INIT_SEQ_DC_SIGNAL 0= parallel IS DC is inverted and then AND-ed to the vertical signal 1= parallel IS DC is OR-ed to the vertical signal
6:4	**P_INIT_SEQ_DC_SIGNAL: Frame Initialization Sequence DC Pin.** 0 = NODC 1 = VSYNC 2 = VPULSE0 3 = VPULSE1 4 = VPULSE2 5 = VPULSE3
1	**P_INIT_SEQUENCE_MODE: Initialization Sequence Mode.** 0 = PLCD_INIT 1 = SPI_INIT
0	**P_SEND_INIT_SEQUENCE: Send Initialization Sequence (IS).** 0 = Disable 1 = Enable

DC_P_P_SPI_INIT_SEQ_DATA_A_0

Offset: 142h
Read/Write: R/W
Reset: 0000.0000

SPI Init Sequence Write Data A

For parallel initialization sequence there are two possible data widths: 9 bits or 18 bits. If parallel IS is selected, the number of bits per cycle depend on the DISP_DATA_FORMAT register programming. 18-bit parallel IS cycles are performed for 1-pixel/1-clock parallel interface (DF1P1C). 9-bit parallel IS cycles are performed for non 1-pixel/1-clock parallel interface.

Parallel IS cycles must be completed prior to the end of horizontal active of the line where IS cycles are sent. If all the cycles have been completed prior to the end of horizontal active, control signals are held inactive and last output data is held till end of horizontal active. For 9-bit parallel initialization sequence, the data is output in either LD[8:0] pins or LD[17:9] pins depending on display data alignment. For serial initialization sequence using SPI (IS SPI) there are six possible data widths: 8 bits, 9 bits, 12 bits, 16 bits, 16 bits data plus start byte (24 bits), 18 bits, or 24 bits.

Parameters in SPI_CONTROL register and SPI_START_BYTE register is also used for serial initialization sequence using SPI. Serial IS cycles must also be completed prior to the end of horizontal active of the line where initialization cycles are sent. The programmer needs to make sure that register programming is such that this is true. If all the cycles have been completed prior to the end of horizontal active, SPI signals will be forced inactive until the next SPI cycles.

The following shows how initialization sequence data bits are used when sending initialization sequence:

For 9-bit parallel initialization - up to 10 initialization cycles can be done:

Init cycle	1	2	3	4	5	6	7	8	9	10
data	8-0	17-9	26-18	35-27	44-36	53-45	62-54	71-63	80-72	89-81
LSC0 enable	90	93	96	99	102	105	108	111	114	117
LSC1 enable	91	94	97	100	103	106	109	112	115	118
data/command	92	95	98	101	104	107	110	113	116	119

For 18-bit parallel initialization - up to 6 initialization cycles can be done:

Init cycle	1	2	3	4	5	6
data	17-0	35-18	53-36	71-54	89-72	107-90
LSC0 enable	108	111	114	117	120	123
LSC1 enable	109	112	115	118	121	124
data/command	110	113	116	119	122	125

For serial initialization using SPI, main display SPI chip select (Main SCS_) is always output on LCS_ pin while sub display SPI chip select (Sub SCS_) can be optionally output on several pins (see pin output select definition). Initialization cycle through SPI interface can only be sent to either main or sub display but not to both and the selection bits are specified in the tables below. Note that 0 indicates main display initialization and 1 indicates sub display initialization.

For 8-bit SPI initialization - up to 12 initialization cycles can be done:

Init cycle	1	2	3	4	5	6	7	8	9	10
data	7-0	15-8	23-16	31-24	39-32	47-40	55-48	63-56	71-64	79-72

Main/Sub SCS_ 96 98 100 102 104 106 108 110 112 114
SDC 97 99 101 103 105 107 109 111 113 115

Init cycle 11 12
---
data 87-80 95-88
Main/Sub SCS_ 116 118
SDC 117 119

For 12-bit SPI initialization - up to 9 initialization cycles can be done:
Init cycle 1 2 3 4 5 6 7 8 9
---
data 11-0 23-12 35-24 47-36 59-48 71-60 83-72 95-84 108-96
Main/Sub SCS_ 109 111 113 115 117 119 121 123 125
LDC 110 112 114 116 118 120 122 124 126

For 16-bit SPI initialization - up to 7 initialization cycles can be done:
Init cycle 1 2 3 4 5 6 7
---
data 15-0 31-16 47-32 63-48 79-64 95-80 111-96
Main/Sub SCS_ 112 114 116 118 120 122 124
SDC 113 115 117 119 121 123 125

For 18-bit SPI initialization - up to 6 initialization cycles can be done:
Init cycle 1 2 3 4 5 6
---
data 17-0 35-18 53-36 71-54 89-72 107-90
Main/Sub SCS_ 108 110 112 114 116 118
SDC 109 111 113 115 117 119

For 24-bit SPI initialization - up to 4 initialization cycles can be done:
Init cycle 1 2 3 4
---
data 23-0 47-24 71-48 95-72
Main/Sub SCS_ 96 98 100 102
SDC 97 99 101 103

Bit	Description
31:0	**P_SPI_INIT_SEQ_DATA_A: SPI Init Sequence Write Data bits 31-0**

DC_P_P_SPI_INIT_SEQ_DATA_B_0

Offset: 143h
Read/Write: R/W
Reset: 0000.0000

SPI Init Sequence Write Data B

Bit	Description
31:0	**P_SPI_INIT_SEQ_DATA_B: SPI Init Sequence Write Data bits 63-32**

DC_P_P_SPI_INIT_SEQ_DATA_C_0

Offset: 144h
Read/Write: R/W
Reset: 0000.0000

SPI Init Sequence Write Data C

Bit	Description
31:0	**P_SPI_INIT_SEQ_DATA_C: SPI Init Sequence Write Data bits 95-64**

DC_P_P_SPI_INIT_SEQ_DATA_D_0

Offset: 145h
Read/Write: R/W
Reset: 0000.0000

SPI Init Sequence Write Data D

Bit	Description
31:0	**P_SPI_INIT_SEQ_DATA_D: SPI Init Sequence Write Data bits 127-96**

DC_P_P_DC_MCCIF_FIFOCTRL_0

Offset: 146h
Read/Write: R/W
Reset: 0000.0000

Memory Client Interface Async FIFO Optimization Register A

Memory Client Interface FIFO Control Register: The registers below allow to optimize the synchronization timing in the memory client asynchronous FIFOs. When they can be used depend on the client and memory controller clock ratio. Additionally, the RDMC_RDFAST/RDCL_RDFAST fields can increase power consumption if the asynchronous FIFO is implemented as a real RAM. There is no power impact on latch-based FIFOs. Flipflop-based FIFOs do not use these fields. See recommended settings below.

Note: The register fields can only be changed when the memory client async FIFOs are empty.

The register field ending with WRCL_MCLE2X (if any) can be set to improve async FIFO synchronization on the write side by one client clock cycle if the memory controller clock frequency is less or equal to twice the client clock frequency:

mcclk_freq <= 2 * clientclk_freq

The register field ending with WRMC_CLLE2X (if any) can be set to improve async FIFO synchronization on the write side by one memory controller clock cycle if the client clock frequency is less or equal to twice the memory controller clock frequency:

clientclk_freq <= 2 * mcclk_freq

The register field ending with RDMC_RDFAST (if any) can be set to improve async FIFO synchronization on the read side by one memory controller clock cycle.

Warning: RDMC_RDFAST can be used along with WRCL_MCLE2X only when:
mcclk_freq <= clientclk_freq

The register field ending with RDCL_RDFAST (if any) can be set to improve async FIFO synchronization on the read side by one client clock cycle.

Warning: DCL_RDFAST can be used along with WRMC_CLLE2X only when:

clientclk_freq <= mcclk_freq

RECOMMENDED SETTINGS

Client writing to FIFO, memory controller reading from FIFO
- mcclk_freq <= clientclk_freq

You can enable both RDMC_RDFAST and WRCL_CLLE2X. If one of the FIFOs is a real RAM and power is a concern, you should avoid enabling RDMC_RDFAST.

- clientclk_freq < mcclk_freq <= 2 * clientclk_freq

You can enable RDMC_RDFAST or WRCL_MCLE2X, but because the client clock is slower, you should enable only WRCL_MCLE2X.

- 2 * clientclk_freq < mcclk_freq

You can only enable RDMC_RDFAST. If one of the FIFOs is a real RAM and power is a concern, you should avoid enabling RDMC_RDFAST.

Memory controller writing to FIFO, client reading from FIFO

- clientclk_freq <= mcclk_freq

You can enable both RDCL_RDFAST and WRMC_CLLE2X. If one of the FIFOs is a real RAM and power is a concern, you should avoid enabling RDCL_RDFAST.

- mcclk_freq < clientclk_freq <= 2 * mcclk_freq

You can enable RDCL_RDFAST or WRMC_CLLE2X, but because the memory controller clock is slower, you should enable only WRMC_CLLE2X.

- 2 * mcclk_freq < clientclk_freq

You can only enable RDCL_RDFAST. If one of the FIFOs is a real RAM and power is a concern, you should avoid enabling RDCL_RDFAST.

Bit	Description
3	**P_DC_MCCIF_RDCL_RDFAST:** 0 = Disable 1 = Enable
2	**P_DC_MCCIF_WRMC_CLLE2X:** 0 = Disable 1 = Enable
1	**P_DC_MCCIF_RDMC_RDFAST:** 0 = Disable 1 = Enable
0	**P_DC_MCCIF_WRCL_MCLE2X:** 0 = Disable 1 = Enable

DC_WIN_P_A_WIN_OPTIONS_0

Offset: 201h
Read/Write: R/W
Reset: 0000.0000

Window B Options

Class: Display Window Settings

Display Window B parameters

Bit	Description
20	**P_A_DV_ENABLE: Window B Digital Vibrance Enable.** 0 = Disable 1 = Enable
16	**P_A_CP_ENABLE: Window B Color Palette Enable.** 0 = Disable 1 = Enable

Bit	Description
6	**P_A_COLOR_EXPAND: Window B 12/15/16-to-24 bpp color expansion.** 0 = Disable 1 = Enable
2	**P_A_V_DIRECTION: Window B Vertical (Y) drawing Direction.** 0 = Increment 1 = Decrement
0	**P_A_H_DIRECTION: Window B Horizontal (X) drawing Direction.** 0 = Increment 1 = Decrement

DC_WIN_P_A_BYTE_SWAP_0

Offset: 202h
Read/Write: R/W
Reset: 0000.0000

Window B Byte Swap

Bit	Description
1:0	**P_A_BYTE_SWAP: Window B Byte Swap.** 0 = NOSWAP 1 = SWAP2 2 = SWAP4 3 = SWAP4HW

DC_WIN_P_A_BUFFER_CONTROL_0

Offset: 203h
Read/Write: R/W
Reset: 0000.0000

Window B Buffer Control

Bit	Description
2:0	**P_A_BUFFER_CONTROL: Window B Buffer Control.** 0 = HOST 1 = VI 2 = EPP 3 = MPEGE 4 = SB2D

DC_WIN_P_A_COLOR_DEPTH_0

Offset: 204h
Read/Write: R/W
Reset: 0000.0000

Window B Color Depth

For YCbCr data format, Cb and Cr are 8-bit unsigned values. For YUV data format, U and V are 8-bit signed values. YCbCr422R is similar to YCbCr422P but the Cb and Cr are shared vertically. YUV422R is similar to YUV422P but the U and V are shared vertically. YCbCr422RA is same as YCbCr422R in memory and YUV422RA is same as YUV422R in memory but while reading from memory, for YCbCr422RA and YUV422RA, chroma averaging is applied for each pixel pair so that they can be processed as YUV422 by the display pipeline. For YCbCr422R and YUV422R, every other chroma pixels are not used (discarded) by the display pipeline.

Bit	Description
4:0	**P_A_COLOR_DEPTH: Window B Color Depth.** 0 = P1 1 = P2 2 = P4 3 = P8 4 = B4G4R4A4 5 = B5G5R5A 6 = B5G6R5 12 = B8G8R8A8 13 = R8G8B8A8 16 = YCbCr422 17 = YUV422 18 = YCbCr420P 19 = YUV420P 20 = YCbCr422P 21 = YUV422P 22 = YCbCr422R 23 = YUV422R 24 = YCbCr422RA 25 = YUV422RA

DC_WIN_P_A_POSITION_0

Offset: 205h
Read/Write: R/W
Reset: 0000.0000

Window B Position

This register defines H position and size of window B after scaling (if there is any)

Bit	Description
27:16	**P_A_V_POSITION: Window B V Position** This is specified with respect to the top edge of active display area.
11:0	**P_A_H_POSITION: Window B H Position** This is specified with respect to the left edge of active display area.

DC_WIN_P_A_SIZE_0

Offset: 206h
Read/Write: R/W
Reset: 0000.0000

Window B Size

This register defines V position and size of window B after scaling (if there is any)

Bit	Description
27:16	**P_A_V_SIZE: Window B V Size (lines)** This is the vertical size after scaling.
11:0	**P_A_H_SIZE: Window B H Size (pixels)** This is the horizontal size after scaling.

DC_WIN_P_A_PRESCALED_SIZE_0

Offset: 207h
Read/Write: R/W
Reset: 0000.0000

Window B Pre-scaled Size

This register defines Window B pre-scaled size. The H pre-scaled size is needed to determine how many bytes to fetch from memory per line and this parameter must be programmed exactly as needed taking into account the scaling factor. For planar YUV or YCbCr data formats, this parameter refer to the H pre-scaled of the Y plane.

Bit	Description
27:16	**P_A_V_PRESCALED_SIZE: Window B V Pre-scaled Size (lines)**
13:0	**P_A_H_PRESCALED_SIZE: Window B H Pre-scaled Size (bytes)**

DC_WIN_P_A_H_INITIAL_DDA_0

Offset: 208h
Read/Write: R/W
Reset: 0000.0000

Window B H Initial DDA

Similarly with the V Initial DDA.

Bit	Description
15:0	**P_A_H_INITIAL_DDA: Window B H Initial DDA (4.12)** This is typically programmed to 0.0

DC_WIN_P_A_V_INITIAL_DDA_0

Offset: 209h
Read/Write: R/W
Reset: 0000.0000

Window B V Initial DDA

Bit	Description
31:16	**P_A_VB_INITIAL_DDA: Window B V Bottom Field Initial DDA (4.12)** This is typically programmed to 0.0 for non-interlaced source or 0.5 for interlaced source.
15:0	**P_A_VT_INITIAL_DDA: Window B V Top Field Initial DDA (4.12)** This is typically programmed to 0.0 for both non-interlaced and interlaced sources.

DC_WIN_P_A_DDA_INCREMENT_0

Offset: 20ah
Read/Write: R/W
Reset: 0000.0000

Window B DDA Increment

DDA increment is typically calculated by dividing (Pre-scaled size - 1) by (Post-scaled size - 1). The result should be round-ed up and expressed as 4.12 format (4-bit integer and 12-bit fraction). If the DDA increment is less than 1.0 then image will be up-scaled and if DDA increment is more than 1.0 then image will be down-scaled.

Bit	Description
31:16	**P_A_V_DDA_INCREMENT: Window B Vertical DDA Increment (4.12)** This should be set to 1.0 if there is no scaling. Maximum value is 15.0 regardless of the number of bytes per pixel.
15:0	**P_A_H_DDA_INCREMENT: Window B Horizontal DDA Increment (4.12)** This should be set to 1.0 if there is no scaling. The maximum value for downscaling depends on the number of bytes per pixel. For 4-byte/pixel modes (32-bpp) the maximum value is 4.0 and for all other modes the maximum value is 8.0.

DC_WIN_P_A_LINE_STRIDE_0

Offset: 20bh
Read/Write: R/W
Reset: 0000.0000

Window B Line Stride

Bit	Description
15:0	**P_A_LINE_STRIDE: Window B Line Stride** This is stride (in bytes) for all non-planar data formats. For planar YUV or YCbCr data formats, this is stride (in bytes) for the luma plane. This is valid for all window B buffers.

DC_WIN_P_A_PALETTE_COLOR_EXT_0

Offset: 20ch
Read/Write: R/W
Reset: 0000.0000

Window B Palette Color Extension

Palette extension for 1-bpp, 2-bpp, and 4-bpp. These bits provide the upper most significant bits for indexing the color palette.

Supported for window A only.

Bit	Description
7:1	**P_A_PALETTE_COLOR_EXT: Window B Palette Color Extension** bits 7-1 are used for 1-bpp mode bits 7-2 are used for 2-bpp mode bits 7-4 are used for 4-bpp mode

DC_WIN_P_A_BLEND_NOKEY_0

Offset:	20dh
Read/Write:	R/W
Reset:	0000.0000

Blend Control for this window areas where color key is enabled but the pixel color is not within the color key range (color key not match). This is valid for all overlapping condition but only if there is no overlap with other window with higher priority color key enabled and color key not match.

Class: Display Color Keying and Blending

Color keying and blending of the display windows are done prior to cursor blending. Cursor always go on top of the blended windows. Blending is controlled independently on each possible overlap area of the display windows. If 3 windows are enabled there are 7 possible overlap area combinations. For every window in each overlap area combination, color key can be disabled or enabled. Also, for every window in each overlap area combination, there is a corresponding window blend control parameter and a window blend weight parameter. The window blend control parameter is always effective but the window blend weight is not always used. The window blend weight can also be derived from alpha value or from the reverse of other overlapping windows weight.

Color keying has the highest priority for display window blending. Color key consists of a range of color which is searched independently for each windows. If more than 1 windows color keys are enabled then Window A color key has the highest priority, followed by Window B color key, and then followed by Window C color key.

Two sets of color key range (Color Key 0 and Color Key 1) can be defined and they are shared for all windows. It is possible to use both color key sets for the same window or for two separate windows. The two sets of color key ranges should not overlap and if they do the overlap colors are treated as if they are part of Color Key 0 and not part of Color Key 1.

Assuming that there is no overlap with other higher priority window with color key not matched, if a window color key is enabled for any overlap condition and the window pixel is not within the color key range (key not match), then the window pixel will not be blended with other overlapping window pixels but it will be weighted. The weight is not dependent on the overlap condition it is controlled by the same set of parameters for all overlapping condition.

Assuming that there is no overlap with other higher priority window with color key not matched, if a window color key is enabled for a particular overlap condition and the window pixel is within the color key range (key match), then the window pixel will be weighted and blended with other overlapping pixels and this is controlled separately for each overlap condition.

Assuming that there is no overlap with other higher priority window with color key not matched, if a window color key is disabled then the window pixel will be blended with other overlapping pixels and this is controlled separately for each overlap condition.

Display Color Key parameters

For RGB444, RGB565, RGB565 mode, color key should be compared prior to color conversion to 24-bpp and unused least significant bits of the pixel are filled with zeros. For

palettized mode, color key is compared prior to color palette and the palettized color is compared against the green color key values/mask. For YUV mode, U and V are offset by +128 before performing the color key comparison. In all cases, color key is compared prior to horizontal/vertical scaling filter and prior to digital vibrance control. Both upper and lower values are inclusive.

Bit	Description
23:16	**P_A_BLEND_WEIGHT1_NOKEY: Window blend weight 1 for color key not match areas.** This is used only for alpha weight with 1-bit alpha and alpha value of 1.
15:8	**P_A_BLEND_WEIGHT0_NOKEY: Window blend weight 0 for color key not match areas.** For alpha weight, this is used for 1-bit alpha with value of 0.
0	**P_A_BLEND_CONTROL_NOKEY: Window blend control for color key not match areas.** 0 = FIX_WEIGHT 1 = ALPHA_WEIGHT

DC_WIN_P_A_BLEND_1WIN_0

Offset: 20eh
Read/Write: R/W
Reset: 0000.0000

Blend Control for this window area where it does not overlap with other windows.

Bit	Description
23:16	**P_A_BLEND_WEIGHT1_1WIN: Window blend weight 1 for color key disabled or color key enabled with key matched areas.** This is used only for alpha weight with 1-bit alpha and alpha value of 1.
15:8	**P_A_BLEND_WEIGHT0_1WIN: Window blend weight 0 for color key disabled or color key enabled with key matched areas.** For alpha weight, this is used for 1-bit alpha with value of 0.
2	**P_A_BLEND_CONTROL_1WIN: Window blend control in area where it does not overlap with other windows and either color key disabled or color key enabled with key matched.** 0 = FIX_WEIGHT 1 = ALPHA_WEIGHT
1:0	**P_A_CKEY_ENABLE_1WIN: Window color key enable.** 0 = NOKEY 1 = CKEY0 2 = CKEY1 3 = CKEY01

DC_WIN_P_A_BLEND_2WIN_B_0

Offset: 20fh
Read/Write: R/W
Reset: 0000.0000

Blend Control for this window area that overlaps with window B only.

Bit	Description
23:16	**P_A_BLEND_WEIGHT1_2WIN_B: Window blend weight 1 for color key disabled or color key enabled with key matched areas.** This is used only for alpha weight with 1-bit alpha and alpha value of 1.
15:8	**P_A_BLEND_WEIGHT0_2WIN_B: Window blend weight 0 for color key disabled or color key enabled with key matched areas.** For alpha weight, this is used for 1-bit alpha with value of 0.
3:2	**P_A_BLEND_CONTROL_2WIN_B: Window blend control in area where either color key disabled or color key enabled with key matched.** 0 = FIX_WEIGHT 1 = ALPHA_WEIGHT 2 = DEPENDENT_WEIGHT
1:0	**P_A_CKEY_ENABLE_2WIN_B: Window color key enable.** 0 = NOKEY 1 = CKEY0 2 = CKEY1 3 = CKEY01

DC_WIN_P_A_BLEND_2WIN_C_0

Offset: 210h
Read/Write: R/W
Reset: 0000.0000

Blend Control for this window area that overlaps with window C only.

Bit	Description
23:16	**P_A_BLEND_WEIGHT1_2WIN_C: Window blend weight 1 for color key disabled or color key enabled with key matched areas.** This is used only for alpha weight with 1-bit alpha and alpha value of 1.
15:8	**P_A_BLEND_WEIGHT0_2WIN_C: Window blend weight 0 for color key disabled or color key enabled with key matched areas.** For alpha weight, this is used for 1-bit alpha with value of 0.
3:2	**P_A_BLEND_CONTROL_2WIN_C: Window blend control in area where either color key disabled or color key enabled with key matched.** 0 = FIX_WEIGHT 1 = ALPHA_WEIGHT 2 = DEPENDENT_WEIGHT
1:0	**P_A_CKEY_ENABLE_2WIN_C: Window color key enable.** 0 = NOKEY 1 = CKEY0 2 = CKEY1 3 = CKEY01

DC_WIN_P_A_BLEND_3WIN_BC_0

Offset: 211h
Read/Write: R/W
Reset: 0000.0000

Blend Control for this window area that overlaps with windows B and C only.

Bit	Description
23:16	**P_A_BLEND_WEIGHT1_3WIN_BC: Window blend weight 1 for color key disabled or color key enabled with key matched areas.** This is used only for alpha weight with 1-bit alpha and alpha value of 1.
15:8	**P_A_BLEND_WEIGHT0_3WIN_BC: Window blend weight 0 for color key disabled or color key enabled with key matched areas.** For alpha weight, this is used for 1-bit alpha with value of 0.
3:2	**P_A_BLEND_CONTROL_3WIN_BC: Window blend control in area where either color key disabled or color key enabled with key matched.** 0 = FIX_WEIGHT 1 = ALPHA_WEIGHT 2 = DEPENDENT_WEIGHT
1:0	**P_A_CKEY_ENABLE_3WIN_BC: Window color key enable.** 0 = NOKEY 1 = CKEY0 2 = CKEY1 3 = CKEY01

DC_BUF_P_A0_START_ADDR_0

Offset: 300h
Read/Write: R/W
Reset: 0000.0000

Window B Buffer 0 Start Address

In the case where host (software) controls the window buffer, the buffer start addresses are used as byte address to indicate the start address of the buffer.

The least significant bit of the start address is not used for 16-bpp non-planar pixel format and the least two significant bits of the start address are not used for 32-bpp non-planar pixel format.

In the case where window buffer is not controlled by host (software), the buffer 0 start addresses are used as buffer 0 start address sent by the controlling module and buffer 1 start addresses are used as buffer strides. For planar YUV, there are only two buffer strides, one for luma and one for chroma, so Buffer 1 Start Address is used for luma buffer stride and Buffer 1 Start Address for U is used for chroma buffer stride and Buffer 1 Start Address for V is not used.

Buffer 0 start addresses must include necessary buffer offset that is due to H/V flip or cropping. Note that in this case, the controlling module will send buffer index and display module will calculate the actual start addresses of the buffer based on buffer 0 start address, the buffer strides and the buffer index.

In the case where window buffer is not controlled by host (software) then a frame may be stored in multiple buffers. In this case, the buffers must be contiguous in the memory because display will use the same luma or chroma line strides for all lines in the frame. Also buffer wraparound must not occur in the middle of the displayed part of the frame. The controlling module will send frame start and frame end indicators (flags) to display module to indicate the beginning and end of frame. Buffer start address is latched when frame start flag is active but the actual buffer start address is not swithced until frame end flag is active. In the case where one buffer correspond to one frame then frame start and frame end flag are active everytime a buffer index is sent.

Bit	Description
25:0	**P_A0_START_ADDR: Window B Buffer 0 Start Address** This is a byte address. The LSB is not used for 16-bpp non-planar pixel format and the last 2 LSB are not used for 32-bpp non-planar pixel format. For YUV planar pixel format, this specifies start address for the Y plane.

DC_BUF_P_A1_START_ADDR_0

Offset: 300h
Read/Write: R/W
Reset: 0000.0000

Window B Buffer 0 Start Address

In the case where host (software) controls the window buffer, the buffer start addresses are used as byte address to indicate the start address of the buffer.
The least significant bit of the start address is not used for 16-bpp non-planar pixel format and the least two significant bits of the start address are not used for 32-bpp non-planar pixel format.

In the case where window buffer is not controlled by host (software), the buffer 0 start addresses are used as buffer 0 start address sent by the controlling module and buffer 1 start addresses are used as buffer strides. For planar YUV, there are only two buffer strides, one for luma and one for chroma, so Buffer 1 Start Address is used for luma buffer stride and Buffer 1 Start Address for U is used for chroma buffer stride and Buffer 1 Start Address for V is not used. Buffer 0 start addresses must include necessary buffer offset that is due to H/V flip or cropping.

Note that in this case, the controlling module will send buffer index and display module will calculate the actual start addresses of the buffer based on buffer 0 start address, the buffer strides and the buffer index. In the case where window buffer is not controlled by host (software) then a frame may be stored in multiple buffers. In this case, the buffers must be contiguous in the memory because display will use the same luma or chroma line strides for all lines in the frame. Also buffer wraparound must not occur in the middle of the displayed part of the frame. The controlling module will send frame start and frame end indicators (flags) to display module to indicate the beginning and end of frame. Buffer start address is latched when frame start flag is active but the actual buffer start address is not swithced until frame end flag is active. In the case where one buffer correspond to one frame then frame start and frame end flag are active everytime a buffer index is sent.

Bit	Description
25:0	**P_A1_START_ADDR: Window B Buffer 0 Start Address** This is a byte address. The LSB is not used for 16-bpp non-planar pixel format and the last 2 LSB are not used for 32-bpp non-planar pixel format. For YUV planar pixel format, this specifies start address for the Y plane.

DC_WIN_P_B_WIN_OPTIONS_0

Offset: 201h
Read/Write: R/W
Reset: 0000.0000

Window B Options

Class: Display Window Settings

Display Window B parameters

Bit	Description
20	**P_B_DV_ENABLE: Window B Digital Vibrance Enable.** 0 = Disable 1 = Enable
18	**P_B_CSC_ENABLE: Window B Color Space Conversion Enable.** 0 = Disable 1 = Enable
16	**P_B_CP_ENABLE: Window B Color Palette Enable.** 0 = Disable 1 = Enable
12	**P_B_V_FILTER_OPTIMIZE: Window B V Filter Optimization.** 0 = Disable 1 = Enable
10	**P_B_V_FILTER_ENABLE: Window B V Filter Enable.** 0 = Disable 1 = Enable
8	**P_B_H_FILTER_ENABLE: Window B H Filter Enable.** 0 = Disable 1 = Enable
6	**P_B_COLOR_EXPAND: Window B 12/15/16-to-24 bpp color expansion.** 0 = Disable 1 = Enable
2	**P_B_V_DIRECTION: Window B Vertical (Y) drawing Direction.** 0 = Increment 1 = Decrement
0	**P_B_H_DIRECTION: Window B Horizontal (X) drawing Direction.** 0 = Increment 1 = Decrement

DC_WIN_P_B_BYTE_SWAP_0

Offset: 202h
Read/Write: R/W
Reset: 0000.0000

Window B Byte Swap

Bit	Description
1:0	**P_B_BYTE_SWAP: Window B Byte Swap.** 0 = NOSWAP 1 = SWAP2 2 = SWAP4 3 = SWAP4HW

DC_WIN_P_B_BUFFER_CONTROL_0

Offset: 203h
Read/Write: R/W
Reset: 0000.0000

Window B Buffer Control

Bit	Description
2:0	**P_B_BUFFER_CONTROL: Window B Buffer Control.** 0 = HOST 1 = VI 2 = EPP 3 = MPEGE 4 = SB2D

DC_WIN_P_B_COLOR_DEPTH_0

Offset: 204h
Read/Write: R/W
Reset: 0000.0000

Window B Color Depth

For YCbCr data format, Cb and Cr are 8-bit unsigned values. For YUV data format, U and V are 8-bit signed values. YCbCr422R is similar to YCbCr422P but the Cb and Cr are shared vertically. YUV422R is similar to YUV422P but the U and V are shared vertically. YCbCr422RA is same as YCbCr422R in memory and YUV422RA is same as YUV422R in memory but while reading from memory, for YCbCr422RA and YUV422RA, chroma averaging is applied for each pixel pair so that they can be processed as YUV422 by the display pipeline. For YCbCr422R and YUV422R, every other chroma pixels are not used (discarded) by the display pipeline.

Bit	Description
4:0	**P_B_COLOR_DEPTH: Window B Color Depth.** 0 = P1 1 = P2 2 = P4 3 = P8 4 = B4G4R4A4 5 = B5G5R5A 6 = B5G6R5 12 = B8G8R8A8 13 = R8G8B8A8 16 = YCbCr422 17 = YUV422 18 = YCbCr420P 19 = YUV420P 20 = YCbCr422P 21 = YUV422P 22 = YCbCr422R 23 = YUV422R 24 = YCbCr422RA 25 = YUV422RA

DC_WIN_P_B_POSITION_0

Offset: 205h
Read/Write: R/W
Reset: 0000.0000

Window B Position

This register defines H position and size of window B after scaling (if there is any)

Bit	Description
27:16	**P_B_V_POSITION: Window B V Position** This is specified with respect to the top edge of active display area.
11:0	**P_B_H_POSITION: Window B H Position** This is specified with respect to the left edge of active display area.

DC_WIN_P_B_SIZE_0

Offset: 206h
Read/Write: R/W
Reset: 0000.0000

Window B Size

This register defines V position and size of window B after scaling (if there is any)

Bit	Description
27:16	**P_B_V_SIZE: Window B V Size (lines)** This is the vertical size after scaling.
11:0	**P_B_H_SIZE: Window B H Size (pixels)** This is the horizontal size after scaling.

DC_WIN_P_B_PRESCALED_SIZE_0

Offset: 207h
Read/Write: R/W
Reset: 0000.0000

Window B Pre-scaled Size

This register defines Window B pre-scaled size. The H pre-scaled size is needed to determine how many bytes to fetch from memory per line and this parameter must be programmed exactly as needed taking into account the scaling factor. For planar YUV or YCbCr data formats, this parameter refer to the H pre-scaled of the Y plane.

The total number of lines to be fetched from memory is determined by post-scale V size but V pre-scaled size is needed to 'clamp' the last valid line if the vertical DDA is exactly or slightly beyond the specified V pre-scaled size. Design Note: H pre-scaled size ideally should be in terms of pixel but then hardware needs to convert this precisely to bytes to determine the amount of data to request from memory. This could be a risky calculation - maybe this should be made optional on whether we use internal hardware to calculate or left it to software to calculate.

Bit	Description
27:16	**P_B_V_PRESCALED_SIZE: Window B V Pre-scaled Size (lines)**
13:0	**P_B_H_PRESCALED_SIZE: Window B H Pre-scaled Size (bytes)**

DC_WIN_P_B_H_INITIAL_DDA_0

Offset: 208h
Read/Write: R/W
Reset: 0000.0000

Window B H Initial DDA

Design Note: the first pixel of pre-scaled image is always used to output the first pixel so essentially this is the same as forcing the H Initial DDA integer portion to 1 initiallly even though user typically programs this to 0. If it makes the implementation easier, it is possible to force software to program the Initial DDA integer portion to 1.

Similarly with the V Initial DDA.

Bit	Description
15:0	**P_B_H_INITIAL_DDA: Window B H Initial DDA (4.12)** This is typically programmed to 0.0

DC_WIN_P_B_V_INITIAL_DDA_0

Offset: 209h
Read/Write: R/W
Reset: 0000.0000

Window B V Initial DDA

Bit	Description
31:16	**P_B_VB_INITIAL_DDA: Window B V Bottom Field Initial DDA (4.12)** This is typically programmed to 0.0 for non-interlaced source or 0.5 for interlaced source.
15:0	**P_B_VT_INITIAL_DDA: Window B V Top Field Initial DDA (4.12)** This is typically programmed to 0.0 for both non-interlaced and interlaced sources.

DC_WIN_P_B_DDA_INCREMENT_0

Offset: 20ah
Read/Write: R/W
Reset: 0000.0000

Window B DDA Increment

DDA increment is typically calculated by dividing (Pre-scaled size - 1) by (Post-scaled size - 1). The result should be round-ed up and expressed as 4.12 format (4-bit integer and 12-bit fraction). If the DDA increment is less than 1.0 then image will be up-scaled and if DDA increment is more than 1.0 then image will be down-scaled.

Bit	Description
31:16	**P_B_V_DDA_INCREMENT: Window B Vertical DDA Increment (4.12)** This should be set to 1.0 if there is no scaling. Maximum value is 15.0 regardless of the number of bytes per pixel.

Bit	Description
15:0	**P_B_H_DDA_INCREMENT: Window B Horizontal DDA Increment (4.12)** This should be set to 1.0 if there is no scaling. The maximum value for downscaling depends on the number of bytes per pi xel. For 4-byte/pixel modes (32-bpp) the maximum value is 4.0 and for all other modes the maximum value is 8.0.

DC_WIN_P_B_LINE_STRIDE_0

Offset: 20bh
Read/Write: R/W
Reset: 0000.0000

Window B Line Stride

Bit	Description
31:16	**P_B_UV_LINE_STRIDE: Window B Line Stride for Chroma** This is stride (in bytes) for planar YUV or YCbCr data formats for the chroma plane. This is not used (ignored) for other non-planar data formats. This is valid for all window B buffers.
15:0	**P_B_LINE_STRIDE: Window B Line Stride** This is stride (in bytes) for all non-planar data formats. For planar YUV or YCbCr data formats, this is stride (in bytes) for the luma plane. This is valid for all window B buffers.

DC_WIN_P_B_PALETTE_COLOR_EXT_0

Offset: 20ch
Read/Write: R/W
Reset: 0000.0000

Window B Palette Color Extension

Palette extension for 1-bpp, 2-bpp, and 4-bpp. These bits provide the upper most significant bits for indexing the color palette. Supported for window A only.

Bit	Description
7:1	**P_B_PALETTE_COLOR_EXT: Window B Palette Color Extension** bits 7-1 are used for 1-bpp mode bits 7-2 are used for 2-bpp mode bits 7-4 are used for 4-bpp mode

DC_WIN_P_B_BLEND_NOKEY_0

Offset:	20dh
Read/Write:	R/W
Reset:	0000.0000

Blend Control for this window areas where color key is enabled but the pixel color is not within the color key range (color key not match). This is valid for all overlapping condition but only if there is no overlap with other window with higher priority color key enabled and color key not match.

Class: Display Color Keying and Blending

Color keying and blending of the display windows are done prior to cursor blending. Cursor always go on top of the blended windows. Blending is controlled independently on each possible overlap area of the display windows. If 3 windows are enabled there are 7 possible overlap area combinations. For every window in each overlap area combination, color key can be disabled or enabled. Also, for every window in each overlap area combination, there is a corresponding window blend control parameter and a window blend weight parameter. The window blend control parameter is always effective but the window blend weight is not always used. The window blend weight can also be derived from alpha value or from the reverse of other overlapping windows weight.

Color keying has the highest priority for display window blending. Color key consists of a range of color which is searched independently for each windows. If more than 1 windows color keys are enabled then Window A color key has the highest priority, followed by Window B color key, and then followed by Window C color key. Two sets of color key range (Color Key 0 and Color Key 1) can be defined and they are shared for all windows. It is possible to use both color key sets for the same window or for two separate windows.
The two sets of color key ranges should not overlap and if they do the overlap colors are treated as if they are part of Color Key 0 and not part of Color Key 1.

Assuming that there is no overlap with other higher priority window with color key not matched, if a window color key is enabled for any overlap condition and the window pixel is not within the color key range (key not match), then the window pixel will not be blended with other overlapping window pixels but it will be weighted. The weight is not dependent on the overlap condition it is controlled by the same set of parameters for all overlapping condition.

Assuming that there is no overlap with other higher priority window with color key not matched, if a window color key is enabled for a particular overlap condition and the window pixel is within the color key range (key match), then the window pixel will be weighted and blended with other overlapping pixels and this is controlled separately for each overlap condition.

Assuming that there is no overlap with other higher priority window with color key not matched, if a window color key is disabled then the window pixel will be blended with other overlapping pixels and this is controlled separately for each overlap condition.

Display Color Key parameters

For RGB444, RGB565, RGB565 mode, color key should be compared prior to color conversion to 24-bpp and unused least significant bits of the pixel are filled with zeros.
For palettized mode, color key is compared prior to color palette and the palettized color is compared against the green color key values/mask. For YUV mode, U and V are offset by +128 before performing the color key comparison. In all cases, color key is compared prior to horizontal/vertical scaling filter and prior to digital vibrance control. Both upper and lower values are inclusive.

Bit	Description
23:16	**P_B_BLEND_WEIGHT1_NOKEY: Window blend weight 1 for color key not match areas.** This is used only for alpha weight with 1-bit alpha and alpha value of 1.
15:8	**P_B_BLEND_WEIGHT0_NOKEY: Window blend weight 0 for color key not match areas.** For alpha weight, this is used for 1-bit alpha with value of 0.
0	**P_B_BLEND_CONTROL_NOKEY: Window blend control for color key not match areas.** 0 = Fix_weight 1 = Alpha_weight

DC_WIN_P_B_BLEND_1WIN_0

Offset: 20eh
Read/Write: R/W
Reset: 0000.0000

Blend Control for this window area where it does not overlap with other windows.

Bit	Description
23:16	**P_B_BLEND_WEIGHT1_1WIN: Window blend weight 1 for color key disabled or color key enabled with key matched areas.** This is used only for alpha weight with 1-bit alpha and alpha value of 1.
15:8	**P_B_BLEND_WEIGHT0_1WIN: Window blend weight 0 for color key disabled or color key enabled with key matched areas.** For alpha weight, this is used for 1-bit alpha with value of 0.
2	**P_B_BLEND_CONTROL_1WIN: Window blend control in area where it does not overlap with other windows and either color key disabled or color key enabled with key matched.** 0 = Fix_weight 1 = Alpha_weight
1:0	**P_B_CKEY_ENABLE_1WIN: Window color key enable.** 0 = NOKEY 1 = CKEY0 2 = CKEY1 3 = CKEY01

DC_WIN_P_B_BLEND_2WIN_A_0

Offset: 20fh
Read/Write: R/W
Reset: 0000.0000

Blend Control for this window area that overlaps with window A only.

Bit	Description
23:16	**P_B_BLEND_WEIGHT1_2WIN_A: Window blend weight 1 for color key disabled or color key enabled with key matched areas.** This is used only for alpha weight with 1-bit alpha and alpha value of 1.

Bit	Description
15:8	**P_B_BLEND_WEIGHT0_2WIN_A: Window blend weight 0 for color key disabled or color key enabled with key matched areas.** For alpha weight, this is used for 1-bit alpha with value of 0.
3:2	**P_B_BLEND_CONTROL_2WIN_A: Window blend control in area where either color key disabled or color key enabled with key matched.** 0 = Fix_weight 1 = Alpha_weight 2 = Dependent_weight
1:0	**P_B_CKEY_ENABLE_2WIN_A: Window color key enable.** 0 = NOKEY 1 = CKEY0 2 = CKEY1 3 = CKEY01

DC_WIN_P_B_BLEND_2WIN_C_0

Offset: 210h
Read/Write: R/W
Reset: 0000.0000

Blend Control for this window area that overlaps with window C only.

Bit	Description
23:16	**P_B_BLEND_WEIGHT1_2WIN_C: Window blend weight 1 for color key disabled or color key enabled with key matched areas.** This is used only for alpha weight with 1-bit alpha and alpha value of 1.
15:8	**P_B_BLEND_WEIGHT0_2WIN_C: Window blend weight 0 for color key disabled or color key enabled with key matched areas.** For alpha weight, this is used for 1-bit alpha with value of 0.
3:2	**P_B_BLEND_CONTROL_2WIN_C: Window blend control in area where either color key disabled or color key enabled with key matched.** 0 = Fix_weight 1 = Alpha_weight 2 = Dependent_weight
1:0	**P_B_CKEY_ENABLE_2WIN_C: Window color key enable.** 0 = NOKEY 1 = CKEY0 2 = CKEY1 3 = CKEY01

DC_WIN_P_B_BLEND_3WIN_AC_0

Offset: 211h
Read/Write: R/W
Reset: 0000.0000

Blend Control for this window area that overlaps with windows A and C only.

Bit	Description
23:16	**P_B_BLEND_WEIGHT1_3WIN_AC: Window blend weight 1 for color key disabled or color key enabled with key matched areas.** This is used only for alpha weight with 1-bit alpha and alpha value of 1.
15:8	**P_B_BLEND_WEIGHT0_3WIN_AC: Window blend weight 0 for color key disabled or color key enabled with key matched areas.** For alpha weight, this is used for 1-bit alpha with value of 0.
3:2	**P_B_BLEND_CONTROL_3WIN_AC: Window blend control in area where either color key disabled or color key enabled with key matched.** 0 = Fix_weight 1 = Alpha_weight 2 = Dependent_weight
1:0	**P_B_CKEY_ENABLE_3WIN_AC: Window color key enable.** 0 = NOKEY 1 = CKEY0 2 = CKEY1 3 = CKEY01

DC_BUF_P_B0_START_ADDR_0

Offset: 300h
Read/Write: R/W
Reset: 0000.0000

Window B Buffer 0 Start Address

In the case where host (software) controls the window buffer, the buffer start addresses are used as byte address to indicate the start address of the buffer. The least significant bit of the start address is not used for 16-bpp non-planar pixel format and the least two significant bits of the start address are not used for 32-bpp non-planar pixel format.

In the case where window buffer is not controlled by host (software), the buffer 0 start addresses are used as buffer 0 start address sent by the controlling module and buffer 1 start addresses are used as buffer strides. For planar YUV, there are only two buffer strides, one for luma and one for chroma, so Buffer 1 Start Address is used for luma buffer stride and Buffer 1 Start Address for U is used for chroma buffer stride and Buffer 1 Start Address for V is not used.

Buffer 0 start addresses must include necessary buffer offset that is due to H/V flip or cropping. Note that in this case, the controlling module will send buffer index and display module will calculate the actual start addresses of the buffer based on buffer 0 start address, the buffer strides and the buffer index.

In the case where window buffer is not controlled by host (software) then a frame may be stored in multiple buffers. In this case, the buffers must be contiguous in the memory because display will use the same luma or chroma line strides for all lines in the frame. Also buffer wraparound must not occur in the middle of the displayed part of the frame. The controlling module will send frame start and frame end indicators (flags) to display module to indicate the beginning and end of frame. Buffer start address is latched when frame start flag is active but the actual buffer start address is not swithced until frame end flag is active. In the case where one buffer correspond to one frame then frame start and frame end flag are active everytime a buffer index is sent.

Bit	Description
25:0	**P_B0_START_ADDR: Window B Buffer 0 Start Address** This is a byte address. The LSB is not used for 16-bpp non-planar pixel format and the last 2 LSB are not used for 32-bpp non-planar pixel format. For YUV planar pixel format, this specifies start address for the Y plane.

DC_BUF_P_B0_START_ADDR_U_0

Offset: 301h
Read/Write: R/W
Reset: 0000.0000

Window B Buffer 0 Start Address for U plane

Bit	Description
25:0	**P_B0_START_ADDR_U: Window B Buffer 0 Start Address for U plane** This is a byte address.

DC_BUF_P_B0_START_ADDR_V_0

Offset: 302h
Read/Write: R/W
Reset: 0000.0000

Window B Buffer 0 Start Address for V plane

Bit	Description
25:0	**P_B0_START_ADDR_V: Window B Buffer 0 Start Address for V plane** This is a byte address.

DC_BUF_P_B1_START_ADDR_0

Offset: 300h
Read/Write: R/W
Reset: 0000.0000

Window B Buffer 0 Start Address

In the case where host (software) controls the window buffer, the buffer start addresses are used as byte address to indicate the start address of the buffer. The least significant bit of the start address is not used for 16-bpp non-planar pixel format and the least two significant bits of the start address are not used for 32-bpp non-planar pixel format.

In the case where window buffer is not controlled by host (software), the buffer 0 start addresses are used as buffer 0 start address sent by the controlling module and buffer 1 start addresses are used as buffer strides. For planar YUV, there are only two buffer strides, one for luma and one for chroma, so Buffer 1 Start Address is used for luma buffer stride and Buffer 1 Start Address for U is used for chroma buffer stride and Buffer 1 Start Address for V is not used.

Buffer 0 start addresses must include necessary buffer offset that is due to H/V flip or cropping. Note that in this case, the controlling module will send buffer index and display module will calculate the actual start addresses of the buffer based on buffer 0 start address, the buffer strides and the buffer index.

In the case where window buffer is not controlled by host (software) then a frame may be stored in multiple buffers. In this case, the buffers must be contiguous in the memory because display will use the same luma or chroma line strides for all lines in the frame. Also buffer wraparound must not occur in the middle of the displayed part of the frame. The controlling module will send frame start and frame end indicators (flags) to display module to indicate the beginning and end of frame. Buffer start address is latched when frame start flag is active but the actual buffer start address is not swithced until frame end flag is active. In the case where one buffer correspond to one frame then frame start and frame end flag are active everytime a buffer index is sent.

Bit	Description
25:0	**P_B1_START_ADDR: Window B Buffer 0 Start Address** This is a byte address. The LSB is not used for 16-bpp non-planar pixel format and the last 2 LSB are not used for 32-bpp non-planar pixel format. For YUV planar pixel format, this specifies start address for the Y plane.

DC_BUF_P_B1_START_ADDR_U_0

Offset: 301h
Read/Write: R/W
Reset: 0000.0000

Window B Buffer 0 Start Address for U plane

Bit	Description
25:0	**P_B1_START_ADDR_U: Window B Buffer 0 Start Address for U plane** This is a byte address.

DC_BUF_P_B1_START_ADDR_V_0

Offset: 302h
Read/Write: R/W
Reset: 0000.0000

Window B Buffer 0 Start Address for V plane

Bit	Description
25:0	**P_B1_START_ADDR_V: Window B Buffer 0 Start Address for V plane** This is a byte address.

DC_WIN_P_C_WIN_OPTIONS_0

Offset: 201h
Read/Write: R/W
Reset: 0000.0000

Window B Options

Class: Display Window Settings

Display Window B parameters

Bit	Description
20	**P_C_DV_ENABLE: Window B Digital Vibrance Enable.** 0 = Disable 1 = Enable
18	**P_C_CSC_ENABLE: Window B Color Space Conversion Enable.** 0 = Disable 1 = Enable
16	**P_C_CP_ENABLE: Window B Color Palette Enable.** 0 = Disable 1 = Enable
8	**P_C_H_FILTER_ENABLE: Window B H Filter Enable.** 0 = Disable 1 = Enable
6	**P_C_COLOR_EXPAND: Window B 12/15/16-to-24 bpp color expansion.** 0 = Disable 1 = Enable

Bit	Description
2	**P_C_V_DIRECTION: Window B Vertical (Y) drawing Direction.** 0 = Increment 1 = Decrement
0	**P_C_H_DIRECTION: Window B Horizontal (X) drawing Direction.** 0 = Increment 1 = Decrement

DC_WIN_P_C_BYTE_SWAP_0

Offset: 202h
Read/Write: R/W
Reset: 0000.0000

Window B Byte Swap

Bit	Description
1:0	**P_C_BYTE_SWAP: Window B Byte Swap.** 0 = NOSWAP 1 = SWAP2 2 = SWAP4 3 = SWAP4HW

DC_WIN_P_C_BUFFER_CONTROL_0

Offset: 203h
Read/Write: R/W
Reset: 0000.0000

Window B Buffer Control

Bit	Description
2:0	**P_C_BUFFER_CONTROL: Window B Buffer Control.** 0 = Host 1 = VI 2 = EPP 3 = MPEGE 4 = SB2D (Video Scaler, 2D engine)

DC_WIN_P_C_COLOR_DEPTH_0

Offset: 204h
Read/Write: R/W
Reset: 0000.0000

Window B Color Depth For YCbCr data format, Cb and Cr are 8-bit unsigned values. For YUV data format, U and V are 8-bit signed values. YCbCr422R is similar to YCbCr422P but the Cb and Cr are shared vertically. YUV422R is similar to YUV422P but the U and V are shared vertically. YCbCr422RA is same as YCbCr422R in memory and YUV422RA is same as YUV422R in memory but while reading from memory, for YCbCr422RA and YUV422RA, chroma averaging is applied for each pixel pair so that they can be processed as YUV422 by the display pipeline. For YCbCr422R and YUV422R, every other chroma pixels are not used (discarded) by the display pipeline.

Bit	Description
4:0	**P_C_COLOR_DEPTH: Window B Color Depth.** 0 = P1 1 = P2 2 = P4 3 = P8 4 = B4G4R4A4 5 = B5G5R5A 6 = B5G6R5 12 = B8G8R8A8 13 = R8G8B8A8 16 = YCbCr422 17 = YUV422 18 = YCbCr420P 19 = YUV420P 20 = YCbCr422P 21 = YUV422P 22 = YCbCr422R 23 = YUV422R 24 = YCbCr422RA 25 = YUV422RA

DC_WIN_P_C_POSITION_0

Offset: 205h
Read/Write: R/W
Reset: 0000.0000

Window B Position

This register defines H position and size of window B after scaling (if there is any)

Bit	Description
27:16	**P_C_V_POSITION: Window B V Position** This is specified with respect to the top edge of active display area.
11:0	**P_C_H_POSITION: Window B H Position** This is specified with respect to the left edge of active display area.

DC_WIN_P_C_SIZE_0

Offset: 206h
Read/Write: R/W
Reset: 0000.0000

Window B Size

This register defines V position and size of window B after scaling (if there is any)

Bit	Description
27:16	**P_C_V_SIZE: Window B V Size (lines)** This is the vertical size after scaling.
11:0	**P_C_H_SIZE: Window B H Size (pixels)** This is the horizontal size after scaling.

DC_WIN_P_C_PRESCALED_SIZE_0

Offset: 207h
Read/Write: R/W
Reset: 0000.0000

Window B Pre-scaled Size

This register defines Window B pre-scaled size. The H pre-scaled size is needed to determine how many bytes to fetch from memory per line and this parameter must be programmed exactly as needed taking into account the scaling factor. For planar YUV or YCbCr data formats, this parameter refer to the H pre-scaled of the Y plane.

The total number of lines to be fetched from memory is determined by post-scale V size but V pre-scaled size is needed to 'clamp' the last valid line if the vertical DDA is exactly or slightly beyond the specified V pre-scaled size. Design Note: H pre-scaled size ideally should be in terms of pixel but then hardware needs to convert this precisely to bytes to determine the amount of data to request from memory.

This could be a risky calculation - maybe this should be made optional on whether we use internal hardware to calculate or left it to software to calculate.

Bit	Description
27:16	**P_C_V_PRESCALED_SIZE: Window B V Pre-scaled Size (lines)**
13:0	**P_C_H_PRESCALED_SIZE: Window B H Pre-scaled Size (bytes)**

DC_WIN_P_C_H_INITIAL_DDA_0

Offset: 208h
Read/Write: R/W
Reset: 0000.0000

Window B H Initial DDA

Design Note: the first pixel of pre-scaled image is always used to output the first pixel so essentially this is the same as forcing the H Initial DDA integer portion to 1 initially even though user typically programs this to 0. If it makes the implementation easier, it is possible to force software to program the Initial DDA integer portion to 1. Similarly with the V Initial DDA.

Bit	Description
15:0	**P_C_H_INITIAL_DDA: Window B H Initial DDA (4.12)** This is typically programmed to 0.0

DC_WIN_P_C_V_INITIAL_DDA_0

Offset: 209h
Read/Write: R/W
Reset: 0000.0000

Window B V Initial DDA

Bit	Description
31:16	**P_C_VB_INITIAL_DDA: Window B V Bottom Field Initial DDA (4.12)** This is typically programmed to 0.0 for non-interlaced source or 0.5 for interlaced source.
15:0	**P_C_VT_INITIAL_DDA: Window B V Top Field Initial DDA (4.12)** This is typically programmed to 0.0 for both non-interlaced and interlaced sources.

DC_WIN_P_C_DDA_INCREMENT_0

Offset: 20ah
Read/Write: R/W
Reset: 0000.0000

Window B DDA Increment

DDA increment is typically calculated by dividing (Pre-scaled size - 1) by (Post-scaled size - 1). The result should be round-ed up and expressed as 4.12 format (4-bit integer and 12-bit fraction). If the DDA increment is less than 1.0 then image will be up-scaled and if DDA increment is more than 1.0 then image will be down-scaled.

Bit	Description
31:16	**P_C_V_DDA_INCREMENT: Window B Vertical DDA Increment (4.12)** This should be set to 1.0 if there is no scaling. Maximum value is 15.0 regardless of the number of bytes per pixel.
15:0	**P_C_H_DDA_INCREMENT: Window B Horizontal DDA Increment (4.12)** This should be set to 1.0 if there is no scaling. The maximum value for downscaling depends on the number of bytes per pixel. For 4-byte/pixel modes (32-bpp) the maximum value is 4.0 and for all other modes the maximum value is 8.0.

DC_WIN_P_C_LINE_STRIDE_0

Offset: 20bh
Read/Write: R/W
Reset: 0000.0000

Window B Line Stride

Bit	Description
31:16	**P_C_UV_LINE_STRIDE: Window B Line Stride for Chroma** This is stride (in bytes) for planar YUV or YCbCr data formats for the chroma plane. This is not used (ignored) for other non-planar data formats. This is valid for all window B buffers.
15:0	**P_C_LINE_STRIDE: Window B Line Stride** This is stride (in bytes) for all non-planar data formats. For planar YUV or YCbCr data formats, this is stride (in bytes) for the luma plane. This is valid for all window B buffers.

DC_WIN_P_C_PALETTE_COLOR_EXT_0

Offset: 20ch
Read/Write: R/W
Reset: 0000.0000

Window B Palette Color Extension

Palette extension for 1-bpp, 2-bpp, and 4-bpp. These bits provide the upper most significant bits for indexing the color palette.

Supported for window A only.

Bit	Description
7:1	**P_C_PALETTE_COLOR_EXT: Window B Palette Color Extension** bits 7:1 are used for 1-bpp mode bits 7:2 are used for 2-bpp mode bits 7:4 are used for 4-bpp mode

DC_WIN_P_C_BLEND_NOKEY_0

Offset:	20dh
Read/Write:	R/W
Reset:	0000.0000

Blend Control for this window areas where color key is enabled but the pixel color is not within the color key range (color key not match). This is valid for all overlapping condition but only if there is no overlap with other window with higher priority color key enabled and color key not match.

Class: Display Color Keying and Blending

Color keying and blending of the display windows are done prior to cursor blending. Cursor always go on top of the blended windows. Blending is controlled independently on each possible overlap area of the display windows. If 3 windows are enabled there are 7 possible overlap area combinations. For every window in each overlap area combination, color key can be disabled or enabled. Also, for every window in each overlap area combination, there is a corresponding window blend control parameter and a window blend weight parameter. The window blend control parameter is always effective but the window blend weight is not always used. The window blend weight can also be derived from alpha value or from the reverse of other overlapping windows weight.

Color keying has the highest priority for display window blending. Color key consists of a range of color which is searched independently for each windows. If more than 1 windows color keys are enabled then Window A color key has the highest priority, followed by Window B color key, and then followed by Window C color key. Two sets of color key range (Color Key 0 and Color Key 1) can be defined and they are shared for all windows. It is possible to use both color key sets for the same window or for two separate windows. The two sets of color key ranges should not overlap and if they do the overlap colors are treated as if they are part of Color Key 0 and not part of Color Key 1.

Assuming that there is no overlap with other higher priority window with color key not matched, if a window color key is enabled for any overlap condition and the window pixel is not within the color key range (key not match), then the window pixel will not be blended with other overlapping window pixels but it will be weighted. The weight is not dependent on the overlap condition it is controlled by the same set of parameters for all overlapping condition.

Assuming that there is no overlap with other higher priority window with color key not matched, if a window color key is enabled for a particular overlap condition and the window pixel is within the color key range (key match), then the window pixel will be weighted and blended with other overlapping pixels and this is controlled separately for each overlap condition.

Assuming that there is no overlap with other higher priority window with color key not matched, if a window color key is disabled then the window pixel will be blended with other overlapping pixels and this is controlled separately for each overlap condition.

Display Color Key parameters

For RGB444, RGB565, RGB565 mode, color key should be compared prior to color conversion to 24-bpp and unused least significant bits of the pixel are filled with zeros. For palettized mode, color key is compared prior to color palette and the palettized color is compared against the green color key values/mask. For YUV mode, U and V are offset by +128 before performing the color key comparison. In all cases, color key is compared prior to horizontal/vertical scaling filter and prior to digital vibrance control. Both upper and lower values are inclusive.

Bit	Description
23:16	**P_C_BLEND_WEIGHT1_NOKEY: Window blend weight 1 for color key not match areas.** This is used only for alpha weight with 1-bit alpha and alpha value of 1.
15:8	**P_C_BLEND_WEIGHT0_NOKEY: Window blend weight 0 for color key not match areas.** For alpha weight, this is used for 1-bit alpha with value of 0.
0	**P_C_BLEND_CONTROL_NOKEY: Window blend control for color key not match areas.** 0 = Fix_weight 1 = Alpha_weight

DC_WIN_P_C_BLEND_1WIN_0

Offset: 20eh
Read/Write: R/W
Reset: 0000.0000

Blend Control for this window area where it does not overlap with other windows.

Bit	Description
23:16	**P_C_BLEND_WEIGHT1_1WIN: Window blend weight 1 for color key disabled or color key enabled with key matched areas.** This is used only for alpha weight with 1-bit alpha and alpha value of 1.
15:8	**P_C_BLEND_WEIGHT0_1WIN: Window blend weight 0 for color key disabled or color key enabled with key matched areas.** For alpha weight, this is used for 1-bit alpha with value of 0.
2	**P_C_BLEND_CONTROL_1WIN: Window blend control in area where it does not overlap with other windows and either color key disabled or color key enabled with key matched.** 0 = Fix_weight 1 = Alpha_weight
1:0	**P_C_CKEY_ENABLE_1WIN: Window color key enable.** 0 = NOKEY 1 = CKEY0 2 = CKEY1 3 = CKEY01

DC_WIN_P_C_BLEND_2WIN_A_0

Offset: 20fh
Read/Write: R/W
Reset: 0000.0000

Blend Control for this window area that overlaps with window A only.

Bit	Description
23:16	**P_C_BLEND_WEIGHT1_2WIN_A: Window blend weight 1 for color key disabled or color key enabled with key matched areas.** This is used only for alpha weight with 1-bit alpha and alpha value of 1.
15:8	**P_C_BLEND_WEIGHT0_2WIN_A: Window blend weight 0 for color key disabled or color key enabled with key matched areas.** For alpha weight, this is used for 1-bit alpha with value of 0.

Bit	Description
3:2	**P_C_BLEND_CONTROL_2WIN_A: Window blend control in area where either color key disabled or color key enabled with key matched.** 0 = Fix_weight 1 = Alpha_weight 2 = Dependent_weight
1:0	**P_C_CKEY_ENABLE_2WIN_A: Window color key enable.** 0 = NOKEY 1 = CKEY0 2 = CKEY1 3 = CKEY01

DC_WIN_P_C_BLEND_2WIN_B_0

Offset: 210h
Read/Write: R/W
Reset: 0000.0000

Blend Control for this window area that overlaps with window B only.

Bit	Description
23:16	**P_C_BLEND_WEIGHT1_2WIN_B: Window blend weight 1 for color key disabled or color key enabled with key matched areas.** This is used only for alpha weight with 1-bit alpha and alpha value of 1.
15:8	**P_C_BLEND_WEIGHT0_2WIN_B: Window blend weight 0 for color key disabled or color key enabled with key matched areas.** For alpha weight, this is used for 1-bit alpha with value of 0.
3:2	**P_C_BLEND_CONTROL_2WIN_B: Window blend control in area where either color key disabled or color key enabled with key matched.** 0 = FIX_WEIGHT 1 = ALPHA_WEIGHT 2 = DEPENDENT_WE**IGHT**
1:0	**P_C_CKEY_ENABLE_2WIN_B: Window color key enable.** 0 = NOKEY 1 = CKEY0 2 = CKEY1 3 = CKEY01

DC_WIN_P_C_BLEND_3WIN_AB_0

Offset: 211h
Read/Write: R/W
Reset: 0000.0000

Blend Control for this window area that overlaps with windows A & B only.

Bit	Description
23:16	**P_C_BLEND_WEIGHT1_3WIN_AB: Window blend weight 1 for color key disabled or color key enabled with key matched areas.** This is used only for alpha weight with 1-bit alpha and alpha value of 1.

Bit	Description
15:8	**P_C_BLEND_WEIGHT0_3WIN_AB: Window blend weight 0 for color key disabled or color key enabled with key matched areas.** For alpha weight, this is used for 1-bit alpha with value of 0.
3:2	**P_C_BLEND_CONTROL_3WIN_AB: Window blend control in area where either color key disabled or color key enabled with key matched..** 0 = Fix_weight 1 = Alpha_weight 2 = Dependent_weight
1:0	**P_C_CKEY_ENABLE_3WIN_AB: Window color key enable.** 0 = NOKEY 1 = CKEY0 2 = CKEY1 3 = CKEY01

DC_BUF_P_C0_START_ADDR_0

Offset: 300h
Read/Write: R/W
Reset: 0000.0000

Window B Buffer 0 Start Address

In the case where host (software) controls the window buffer, the buffer start addresses are used as byte address to indicate the start address of the buffer. The least significant bit of the start address is not used for 16-bpp non-planar pixel format and the least two significant bits of the start address are not used for 32-bpp non-planar pixel format.

In the case where window buffer is not controlled by host (software), the buffer 0 start addresses are used as buffer 0 start address sent by the controlling module and buffer 1 start addresses are used as buffer strides. For planar YUV, there are only two buffer strides, one for luma and one for chroma, so Buffer 1 Start Address is used for luma buffer stride and Buffer 1 Start Address for U is used for chroma buffer stride and Buffer 1 Start Address for V is not used.

Buffer 0 start addresses must include necessary buffer offset that is due to H/V flip or cropping. Note that in this case, the controlling module will send buffer index and display module will calculate the actual start addresses of the buffer based on buffer 0 start address, the buffer strides and the buffer index.

In the case where window buffer is not controlled by host (software) then a frame may be stored in multiple buffers. In this case, the buffers must be contiguous in the memory because display will use the same luma or chroma line strides for all lines in the frame. Also buffer wraparound must not occur in the middle of the displayed part of the frame. The controlling module will send frame start and frame end indicators (flags) to display module to indicate the beginning and end of frame. Buffer start address is latched when frame start flag is active but the actual buffer start address is not swithced until frame end flag is active. In the case where one buffer correspond to one frame then frame start and frame end flag are active everytime a buffer index is sent.

Bit	Description
25:0	**P_C0_START_ADDR: Window B Buffer 0 Start Address** This is a byte address. The LSB is not used for 16-bpp non-planar pixel format and the last 2 LSB are not used for 32-bpp non-planar pixel format. For YUV planar pixel format, this specifies start address for the Y plane.

DC_BUF_P_C0_START_ADDR_U_0

Offset: 301h
Read/Write: R/W
Reset: 0000.0000

Window B Buffer 0 Start Address for U plane

Bit	Description
25:0	**P_C0_START_ADDR_U: Window B Buffer 0 Start Address for U plane** This is a byte address.

DC_BUF_P_C0_START_ADDR_V_0

Offset: 302h
Read/Write: R/W
Reset: 0000.0000

Window B Buffer 0 Start Address for V plane

Bit	Description
25:0	**P_C0_START_ADDR_V: Window B Buffer 0 Start Address for V plane** This is a byte address**.**

DC_BUF_S_C1_START_ADDR_0

Offset: 300h
Read/Write: R/W
Reset: 0000.0000

Window B Buffer 0 Start Address

In the case where host (software) controls the window buffer, the buffer start addresses are used as byte address to indicate the start address of the buffer. The least significant bit of the start address is not used for 16-bpp non-planar pixel format and the least two significant bits of the start address are not used for 32-bpp non-planar pixel format.

In the case where window buffer is not controlled by host (software), the buffer 0 start addresses are used as buffer 0 start address sent by the controlling module and buffer 1 start addresses are used as buffer strides. For planar YUV, there are only two buffer strides, one for luma and one for chroma, so Buffer 1 Start Address is used for luma buffer stride and Buffer 1 Start Address for U is used for chroma buffer stride and Buffer 1 Start Address for V is not used.

Buffer 0 start addresses must include necessary buffer offset that is due to H/V flip or cropping. Note that in this case, the controlling module will send buffer index and display module will calculate the actual start addresses of the buffer based on buffer 0 start address, the buffer strides and the buffer index.

In the case where window buffer is not controlled by host (software) then a frame may be stored in multiple buffers. In this case, the buffers must be contiguous in the memory because display will use the same luma or chroma line strides for all lines in the frame. Also buffer wraparound must not occur in the middle of the displayed part of the frame. The controlling module will send frame start and frame end indicators (flags) to display module to indicate the beginning and end of frame. Buffer start address is latched when frame start flag is active but the actual buffer start address is not swithced until frame end flag is active. In the case where one buffer correspond to one frame then frame start and frame end flag are active everytime a buffer index is sent.

Bit	Description
25:0	**S_C1_START_ADDR: Window B Buffer 0 Start Address** This is a byte address. The LSB is not used for 16-bpp non-planar pixel format and the last 2 LSB are not used for 32-bpp non-planar pixel format. For YUV planar pixel format, this specifies start address for the Y plane.

DC_BUF_S_C1_START_ADDR_U_0

Offset: 301h
Read/Write: R/W
Reset: 0000.0000

Window B Buffer 0 Start Address for U plane

Bit	Description
25:0	**S_C1_START_ADDR_U: Window B Buffer 0 Start Address for U plane** This is a byte address.

DC_BUF_S_C1_START_ADDR_V_0

Offset: 302h
Read/Write: R/W
Reset: 0000.0000

Window B Buffer 0 Start Address for V plane

Bit	Description
25:0	**S_C1_START_ADDR_V: Window B Buffer 0 Start Address for V plane** This is a byte address.

DC_BUF_S_C0_START_ADDR_0

Offset: 300h
Read/Write: R/W
Reset: 0000.0000

Window B Buffer 0 Start Address

In the case where host (software) controls the window buffer, the buffer start addresses are used as byte address to indicate the start address of the buffer. The least significant bit of the start address is not used for 16-bpp non-planar pixel format and the least two significant bits of the start address are not used for 32-bpp non-planar pixel format.

In the case where window buffer is not controlled by host (software), the buffer 0 start addresses are used as buffer 0 start address sent by the controlling module and buffer 1 start addresses are used as buffer strides. For planar YUV, there are only two buffer strides, one for luma and one for chroma, so Buffer 1 Start Address is used for luma buffer stride and Buffer 1 Start Address for U is used for chroma buffer stride and Buffer 1 Start Address for V is not used.

Buffer 0 start addresses must include necessary buffer offset that is due to H/V flip or cropping. Note that in this case, the controlling module will send buffer index and display module will calculate the actual start addresses of the buffer based on buffer 0 start address, the buffer strides and the buffer index.

In the case where window buffer is not controlled by host (software) then a frame may be stored in multiple buffers. In this case, the buffers must be contiguous in the memory because display will use the same luma or chroma line strides for all lines in the frame.

Also buffer wraparound must not occur in the middle of the displayed part of the frame. The controlling module will send frame start and frame end indicators (flags) to display module to indicate the beginning and end of frame. Buffer start address is latched when frame start flag is active but the actual buffer start address is not swithced until frame end flag is active. In the case where one buffer correspond to one frame then frame start and frame end flag are active everytime a buffer index is sent.

Bit	Description
25:0	**S_C0_START_ADDR: Window B Buffer 0 Start Address** This is a byte address. The LSB is not used for 16-bpp non-planar pixel format and the last 2 LSB are not used for 32-bpp non-planar pixel format. For YUV planar pixel format, this specifies start address for the Y plane.

DC_BUF_S_C0_START_ADDR_U_0

Offset: 301h
Read/Write: R/W
Reset: 0000.0000

Window B Buffer 0 Start Address for U plane

Bit	Description
25:0	**S_C0_START_ADDR_U: Window B Buffer 0 Start Address for U plane** This is a byte address.

DC_BUF_S_C0_START_ADDR_V_0

Offset: 302h
Read/Write: R/W
Reset: 0000.0000

Window B Buffer 0 Start Address for V plane

Bit	Description
25:0	**S_C0_START_ADDR_V: Window B Buffer 0 Start Address for V plane** This is a byte address.

DC_WIN_S_A_WIN_OPTIONS_0

Offset: 201h
Read/Write: R/W
Reset: 0000.0000

Window B Options

Class: Display Window Settings

Display Window B parameters

Bit	Description
20	**S_A_DV_ENABLE: Window B Digital Vibrance Enable.** 0 = Disable 1 = Enable
16	**S_A_CP_ENABLE: Window B Color Palette Enable.** 0 = Disable 1 = Enable
6	**S_A_COLOR_EXPAND: Window B 12/15/16-to-24 bpp color expansion.** 0 = Disable 1 = Enable
2	**S_A_V_DIRECTION: Window B Vertical (Y) drawing Direction.** 0 = Increment 1 = Decrement
0	**S_A_H_DIRECTION: Window B Horizontal (X) drawing Direction.** 0 = Increment 1 = Decrement

DC_WIN_S_A_BYTE_SWAP_0

Offset: 202h
Read/Write: R/W
Reset: 0000.0000

Window B Byte Swap

Bit	Description
1:0	**S_A_BYTE_SWAP: Window B Byte Swap.** 0 = NOSWAP 1 = SWAP2 2 = SWAP4 3 = SWAP4HW

DC_WIN_S_A_BUFFER_CONTROL_0

Offset: 203h
Read/Write: R/W
Reset: 0000.0000

Window B Buffer Control

Bit	Description
2:0	**S_A_BUFFER_CONTROL: Window B Buffer Control.** 0 = Host 1 = VI 2 = EPP 3 = MPEGE 4 = SB2D (Video Scaler, 2D engine)

DC_WIN_S_A_COLOR_DEPTH_0

Offset: 204h
Read/Write: R/W
Reset: 0000.0000

Window B Color Depth

For YCbCr data format, Cb and Cr are 8-bit unsigned values. For YUV data format, U and V are 8-bit signed values. YCbCr422R is similar to YCbCr422P but the Cb and Cr are shared vertically. YUV422R is similar to YUV422P but the U and V are shared vertically. YCbCr422RA is same as YCbCr422R in memory and YUV422RA is same as YUV422R in memory but while reading from memory, for YCbCr422RA and YUV422RA, chroma averaging is applied for each pixel pair so that they can be processed as YUV422 by the display pipeline. For YCbCr422R and YUV422R, every other chroma pixels are not used (discarded) by the display pipeline.

Bit	Description
4:0	**S_A_COLOR_DEPTH: Window B Color Depth.** 0 = P1 1 = P2 2 = P4 3 = P8 4 = B4G4R4A4 5 = B5G5R5A 6 = B5G6R5 12 = B8G8R8A8 13 = R8G8B8A8 16 = YCbCr422 17 = YUV422 18 = YCbCr420P 19 = YUV420P 20 = YCbCr422P 21 = YUV422P 22 = YCbCr422R 23 = YUV422R 24 = YCbCr422RA 25 = YUV422RA

DC_WIN_S_A_POSITION_0

Offset: 205h
Read/Write: R/W
Reset: 0000.0000

Window B Position

This register defines H position and size of window B after scaling (if there is any)

Bit	Description
27:16	**S_A_V_POSITION: Window B V Position** This is specified with respect to the top edge of active display area.
11:0	**S_A_H_POSITION: Window B H Position** This is specified with respect to the left edge of active display area.

DC_WIN_S_A_SIZE_0

Offset: 206h
Read/Write: R/W
Reset: 0000.0000

Window B Size

This register defines V position and size of window B after scaling (if there is any)

Bit	Description
27:16	**S_A_V_SIZE: Window B V Size (lines)** This is the vertical size after scaling.
11:0	**S_A_H_SIZE: Window B H Size (pixels)** This is the horizontal size after scaling.

DC_WIN_S_A_PRESCALED_SIZE_0

Offset: 207h
Read/Write: R/W
Reset: 0000.0000

Window B Pre-scaled Size

This register defines Window B pre-scaled size. The H pre-scaled size is needed to determine how many bytes to fetch from memory per line and this parameter must be programmed exactly as needed taking into account the scaling factor. For planar YUV or YCbCr data formats, this parameter refer to the H pre-scaled of the Y plane.

The total number of lines to be fetched from memory is determined by post-scale V size but V pre-scaled size is needed to 'clamp' the last valid line if the vertical DDA is exactly or slightly beyond the specified V pre-scaled size. Design Note: H pre-scaled size ideally should be in terms of pixel but then hardware needs to convert this precisely to bytes to determine the amount of data to request from memory. This could be a risky calculation - maybe this should be made optional on whether we use internal hardware to calculate or left it to software to calculate.

Bit	Description
27:16	**S_A_V_PRESCALED_SIZE: Window B V Pre-scaled Size (lines)**
13:0	**S_A_H_PRESCALED_SIZE: Window B H Pre-scaled Size (bytes)**

DC_WIN_S_A_H_INITIAL_DDA_0

Offset: 208h
Read/Write: R/W
Reset: 0000.0000

Window B H Initial DDA

Design Note: the first pixel of pre-scaled image is always used to output the first pixel so essentially this is the same as forcing the H Initial DDA integer portion to 1 initially even though user typically programs this to 0. If it makes the implementation easier, it is possible to force software to program the Initial DDA integer portion to 1. Similarly with the V Initial DDA.

Bit	Description
15:0	**S_A_H_INITIAL_DDA: Window B H Initial DDA (4.12)** This is typically programmed to 0.0

DC_WIN_S_A_V_INITIAL_DDA_0

Offset: 209h
Read/Write: R/W
Reset: 0000.0000

Window B V Initial DDA

Bit	Description
31:16	**S_A_VB_INITIAL_DDA: Window B V Bottom Field Initial DDA (4.12)** This is typically programmed to 0.0 for non-interlaced source or 0.5 for interlaced source.
15:0	**S_A_VT_INITIAL_DDA: Window B V Top Field Initial DDA (4.12)** This is typically programmed to 0.0 for both non-interlaced and interlaced sources.

DC_WIN_S_A_DDA_INCREMENT_0

Offset: 20ah
Read/Write: R/W
Reset: 0000.0000

Window B DDA Increment

DDA increment is typically calculated by dividing (Pre-scaled size - 1) by (Post-scaled size - 1). The result should be round-ed up and expressed as 4.12 format (4-bit integer and 12-bit fraction). If the DDA increment is less than 1.0 then image will be up-scaled and if DDA increment is more than 1.0 then image will be down-scaled.

Bit	Description
31:16	**S_A_V_DDA_INCREMENT: Window B Vertical DDA Increment (4.12)** This should be set to 1.0 if there is no scaling. Maximum value is 15.0 regardless of the number of bytes per pixel.
15:0	**S_A_H_DDA_INCREMENT: Window B Horizontal DDA Increment (4.12)** This should be set to 1.0 if there is no scaling. The maximum value for downscaling depends on the number of bytes per pixel. For 4-byte/pixel modes (32-bpp) the maximum value is 4.0 and for all other modes the maximum value is 8.0.

DC_WIN_S_A_LINE_STRIDE_0

Offset: 20bh
Read/Write: R/W
Reset: 0000.0000

Window B Line Stride

Bit	Description
15:0	**S_A_LINE_STRIDE: Window B Line Stride** This is stride (in bytes) for all non-planar data formats. For planar YUV or YCbCr data formats, this is stride (in bytes) for the luma plane. This is valid for all window B buffers.

DC_WIN_S_A_PALETTE_COLOR_EXT_0

Offset: 20ch
Read/Write: R/W
Reset: 0000.0000

Window B Palette Color Extension

Palette extension for 1-bpp, 2-bpp, and 4-bpp. These bits provide the upper most significant bits for indexing the color palette.

Supported for window A only.

Bit	Description
7:1	**S_A_PALETTE_COLOR_EXT: Window B Palette Color Extension** bits 7-1 are used for 1-bpp mode bits 7-2 are used for 2-bpp mode bits 7-4 are used for 4-bpp mode

DC_WIN_S_A_BLEND_NOKEY_0

Offset:	20dh
Read/Write:	R/W
Reset:	0000.0000

Blend Control for this window areas where color key is enabled but the pixel color is not within the color key range (color key not match). This is valid for all overlapping condition but only if there is no overlap with other window with higher priority color key enabled and color key not match.

Class: Display Color Keying and Blending

Color keying and blending of the display windows are done prior to cursor blending. Cursor always go on top of the blended windows. Blending is controlled independently on each possible overlap area of the display windows. If 3 windows are enabled there are 7 possible overlap area combinations. For every window in each overlap area combination, color key can be disabled or enabled. Also, for every window in each overlap area combination, there is a corresponding window blend control parameter and a window blend
weight parameter. The window blend control parameter is always effective but the window blend weight is not always used. The window blend weight can also be derived from alpha value or from the reverse of other overlapping windows weight.

Color keying has the highest priority for display window blending. Color key consists of a range of color which is searched independently for each windows. If more than 1 window's color keys are enabled then Window A color key has the highest priority, followed by Window B color key, and then followed by Window C color key. Two sets of color key range (Color Key 0 and Color Key 1) can be defined and they are shared for all windows. It is possible to use both color key sets for the same window or for two separate windows. The two sets of color key ranges should not overlap and if they do the overlap colors are treated as if they are part of Color Key 0 and not part of Color Key 1.

Assuming that there is no overlap with other higher priority window with color key not matched, if a window color key is enabled for any overlap condition and the window pixel is not within the color key range (key not match), then the window pixel will not be blended with other overlapping window pixels but it will be weighted. The weight is not dependent on the overlap condition it is controlled by the same set of parameters for all overlapping conditions.

Assuming that there is no overlap with other higher priority window with color key not matched, if a window color key is enabled for a particular overlap condition and the window pixel is within the color key range (key match), then the window pixel will be weighted and blended with other overlapping pixels and this is controlled separately for each overlap condition.

Assuming that there is no overlap with other higher priority window with color key not matched, if a window color key is disabled then the window pixel will be blended with other overlapping pixels and this is controlled separately for each overlap condition.

Display Color Key parameters

For RGB444, RGB565, RGB565 mode, color key should be compared prior to color conversion to 24-bpp and unused least significant bits of the pixel are filled with zeros. For palettized mode, color key is compared prior to color palette and the palettized color is compared against the green color key values/mask. For YUV mode, U and V are offset by +128 before performing the color key comparison. In all cases, color key is compared prior to horizontal/vertical scaling filter and prior to digital vibrance control.

Both upper and lower values are inclusive.

Bit	Description
23:16	**S_A_BLEND_WEIGHT1_NOKEY: Window blend weight 1 for color key not match areas.** This is used only for alpha weight with 1-bit alpha and alpha value of 1.
15:8	**S_A_BLEND_WEIGHT0_NOKEY: Window blend weight 0 for color key not match areas.** For alpha weight, this is used for 1-bit alpha with value of 0.
0	**S_A_BLEND_CONTROL_NOKEY: Window blend control for color key not match areas.** 0 = Fix_weight 1 = Alpha_weight

DC_WIN_S_A_BLEND_1WIN_0

Offset: 20eh
Read/Write: R/W
Reset: 0000.0000

Blend Control for this window area where it does not overlap with other windows.

Bit	Description
23:16	**S_A_BLEND_WEIGHT1_1WIN: Window blend weight 1 for color key disabled or color key enabled with key matched areas.** This is used only for alpha weight with 1-bit alpha and alpha value of 1.
15:8	**S_A_BLEND_WEIGHT0_1WIN: Window blend weight 0 for color key disabled or color key enabled with key matched areas.** For alpha weight, this is used for 1-bit alpha with value of 0.
2	**S_A_BLEND_CONTROL_1WIN: Window blend control in area where it does not overlap with other windows and either color key disabled or color key enabled with key matched.** 0 = FIX_WEIGHT 1 = ALPHA_WEIGHT
1:0	**S_A_CKEY_ENABLE_1WIN: Window color key enable.** 0 = NOKEY 1 = CKEY0 2 = CKEY1 3 = CKEY01

DC_WIN_S_A_BLEND_2WIN_

B_0Offset: 20fh
Read/Write: R/W
Reset: 0000.0000

Blend Control for this window area that overlaps with window B only.

Bit	Description
23:16	**S_A_BLEND_WEIGHT1_2WIN_B: Window blend weight 1 for color key disabled or color key enabled with key matched areas.** This is used only for alpha weight with 1-bit alpha and alpha value of 1.
15:8	**S_A_BLEND_WEIGHT0_2WIN_B: Window blend weight 0 for color key disabled or color key enabled with key matched areas.** For alpha weight, this is used for 1-bit alpha with value of 0.
3:2	**S_A_BLEND_CONTROL_2WIN_B: Window blend control in area where either color key disabled or color key enabled with key matched.** 0 = Fix_weight 1 = Alpha_weight 2 = Dependent_weight
1:0	**S_A_CKEY_ENABLE_2WIN_B: Window color key enable.** 0 = NOKEY 1 = CKEY0 2 = CKEY1 3 = CKEY01

DC_WIN_S_A_BLEND_2WIN_C_0

Offset: 210h
Read/Write: R/W
Reset: 0000.0000

Blend Control for this window area that overlaps with window C only.

Bit	Description
23:16	**S_A_BLEND_WEIGHT1_2WIN_C: Window blend weight 1 for color key disabled or color key enabled with key matched areas.** This is used only for alpha weight with 1-bit alpha and alpha value of 1.
15:8	**S_A_BLEND_WEIGHT0_2WIN_C: Window blend weight 0 for color key disabled or color key enabled with key matched areas.** For alpha weight, this is used for 1-bit alpha with value of 0.
3:2	**S_A_BLEND_CONTROL_2WIN_C: Window blend control in area where either color key disabled or color key enabled with key matched.** 0 = Fix_weight 1 = Alpha_weight 2 = Dependent_weight
1:0	**S_A_CKEY_ENABLE_2WIN_C: Window color key enable.** 0 = NOKEY 1 = CKEY0 2 = CKEY1 3 = CKEY01

DC_WIN_S_A_BLEND_3WIN_BC_0

Offset: 211h
Read/Write: R/W
Reset: 0000.0000

Blend Control for this window area that overlaps with windows B & C only.

Bit	Description
23:16	**S_A_BLEND_WEIGHT1_3WIN_BC: Window blend weight 1 for color key disabled or color key enabled with key matched areas.** This is used only for alpha weight with 1-bit alpha and alpha value of 1.
15:8	**S_A_BLEND_WEIGHT0_3WIN_BC: Window blend weight 0 for color key disabled or color key enabled with key matched areas.** For alpha weight, this is used for 1-bit alpha with value of 0.
3:2	**S_A_BLEND_CONTROL_3WIN_BC: Window blend control in area where either color key disabled or color key enabled with key matched.** 0 = Fix_weight 1 = Alpha_weight 2 = Dependent_weight
1:0	**S_A_CKEY_ENABLE_3WIN_BC: Window color key enable.** 0 = NOKEY 1 = CKEY0 2 = CKEY1 3 = CKEY01

DC_BUF_S_A0_START_ADDR_0

Offset: 300h
Read/Write: R/W
Reset: 0000.0000

Window B Buffer 0 Start Address

In the case where host (software) controls the window buffer, the buffer start addresses are used as byte address to indicate the start address of the buffer. The least significant bit of the start address is not used for 16-bpp non-planar pixel format and the least two significant bits of the start address are not used for 32-bpp non-planar pixel format.

In the case where window buffer is not controlled by host (software), the buffer 0 start addresses are used as buffer 0 start address sent by the controlling module and buffer 1 start addresses are used as buffer strides. For planar YUV, there are only two buffer strides, one for luma and one for chroma, so Buffer 1 Start Address is used for luma buffer stride and Buffer 1 Start Address for U is used for chroma buffer stride and Buffer 1 Start Address for V is not used.

Buffer 0 start addresses must include necessary buffer offset that is due to H/V flip or cropping. Note that in this case, the controlling module will send buffer index and display module will calculate the actual start addresses of the buffer based on buffer 0 start address, the buffer strides and the buffer index.

In the case where window buffer is not controlled by host (software) then a frame may be stored in multiple buffers. In this case, the buffers must be contiguous in the memory because display will use the same luma or chroma line strides for all lines in the frame. Also buffer wraparound must not occur in the middle of the displayed part of the frame. The controlling module will send frame start and frame end indicators (flags) to display module to indicate the beginning and end of frame. Buffer start address is latched when frame start flag is active but the actual buffer start address is not swithced until frame end flag is active. In the case where one buffer correspond to one frame then frame start and frame end flag are active everytime a buffer index is sent.

Bit	Description
25:0	**S_A0_START_ADDR: Window B Buffer 0 Start Address** This is a byte address. The LSB is not used for 16-bpp non-planar pixel format and the last 2 LSB are not used for 32-bpp non-planar pixel format. For YUV planar pixel format, this specifies start address for the Y plane.

DC_BUF_S_A1_START_ADDR_0

Offset: 300h
Read/Write: R/W
Reset: 0000.0000

Window B Buffer 0 Start Address

In the case where host (software) controls the window buffer, the buffer start addresses are used as byte address to indicate the start address of the buffer. The least significant bit of the start address is not used for 16-bpp non-planar pixel format and the least two significant bits of the start address are not used for 32-bpp non-planar pixel format.

In the case where window buffer is not controlled by host (software), the buffer 0 start addresses are used as buffer 0 start address sent by the controlling module and buffer 1 start addresses are used as buffer strides. For planar YUV, there are only two buffer strides, one for luma and one for chroma, so Buffer 1 Start Address is used for luma buffer stride and Buffer 1 Start Address for U is used for chroma buffer stride and Buffer 1 Start Address for V is not used.

Buffer 0 start addresses must include necessary buffer offset that is due to H/V flip or cropping. Note that in this case, the controlling module will send buffer index and display module will calculate the actual start addresses of the buffer based on buffer 0 start address, the buffer strides and the buffer index.

In the case where window buffer is not controlled by host (software) then a frame may be stored in multiple buffers. In this case, the buffers must be contiguous in the memory because display will use the same luma or chroma line strides for all lines in the frame. Also buffer wraparound must not occur in the middle of the displayed part of the frame. he controlling module will send frame start and frame end indicators (flags) to display module to indicate the beginning and end of frame. Buffer start address is latched when frame start flag is active but the actual buffer start address is not swithced until frame end flag is active. In the case where one buffer correspond to one frame then frame
start and frame end flag are active everytime a buffer index is sent.

Bit	Description
25:0	**S_A1_START_ADDR: Window B Buffer 0 Start Address This is a byte address.** The LSB is not used for 16-bpp non-planar pixel format and the last 2 LSB are not used for 32-bpp non-planar pixel format. For YUV planar pixel format, this specifies start address for the Y plane.

DC_WIN_S_B_WIN_OPTIONS_0

Offset: 201h
Read/Write: R/W
Reset: 0000.0000

Window B Options

Class: Display Window Settings

Display Window B parameters

Bit	Description
20	**S_B_DV_ENABLE: Window B Digital Vibrance Enable.** 0 = Disable 1 = Enable
18	**S_B_CSC_ENABLE: Window B Color Space Conversion Enable.** 0 = Disable 1 = Enable
16	**S_B_CP_ENABLE: Window B Color Palette Enable.** 0 = Disable 1 = Enable
12	**S_B_V_FILTER_OPTIMIZE: Window B V Filter Optimization.** 0 = Disable 1 = Enable
10	**S_B_V_FILTER_ENABLE: Window B V Filter Enable.** 0 = Disable 1 = Enable
8	**S_B_H_FILTER_ENABLE: Window B H Filter Enable.** 0 = Disable 1 = Enable
6	**S_B_COLOR_EXPAND: Window B 12/15/16-to-24 bpp color expansion.** 0 = Disable 1 = Enable
2	**S_B_V_DIRECTION: Window B Vertical (Y) drawing Direction.** 0 = Increment 1 = Decrement
0	**S_B_H_DIRECTION: Window B Horizontal (X) drawing Direction.** 0 = Increment 1 = Decrement

DC_WIN_S_B_BYTE_SWAP_0

Offset: 202h
Read/Write: R/W
Reset: 0000.0000

Window B Byte Swap

Bit	Description
1:0	**S_B_BYTE_SWAP: Window B Byte Swap.** 0 = NOSWAP 1 = SWAP2 2 = SWAP4 3 = SWAP4HW

DC_WIN_S_B_BUFFER_CONTROL_0

Offset: 203h
Read/Write: R/W
Reset: 0000.0000

Window B Buffer Control

Bit	Description
2:0	**S_B_BUFFER_CONTROL: Window B Buffer Control.** 0 = Host 1 = VI 2 = EPP 3 = MPEGE 4 = SB2D (Video Scaler, 2D engine)

DC_WIN_S_B_COLOR_DEPTH_0

Offset: 204h
Read/Write: R/W
Reset: 0000.0000

Window B Color Depth

For YCbCr data format, Cb and Cr are 8-bit unsigned values. For YUV data format, U and V are 8-bit signed values. CbCr422R is similar to YCbCr422P but the Cb and Cr are shared vertically. YUV422R is similar to YUV422P but the U and V are shared vertically. CbCr422RA is same as YCbCr422R in memory and YUV422RA is same as YUV422R in memory but while reading from memory, for YCbCr422RA and YUV422RA, chroma averaging is applied for each pixel pair so that they can be processed as YUV422 by the display pipeline. For YCbCr422R and YUV422R, every other chroma pixels are not used (discarded) by the display pipeline.

Bit	Description
4:0	**S_B_COLOR_DEPTH: Window B Color Depth.** 0 = P1 1 = P2 2 = P4 3 = P8 4 = B4G4R4A4 5 = B5G5R5A 6 = B5G6R5 12 = B8G8R8A8 13 = R8G8B8A8 16 = YCbCr422 17 = YUV422 18 = YCbCr420P 19 = YUV420P 20 = YCbCr422P 21 = YUV422P 22 = YCbCr422R 23 = YUV422R 24 = YCbCr422RA 25 = YUV422RA

DC_WIN_S_B_POSITION_0

Offset: 205h
Read/Write: R/W
Reset: 0000.0000

Window B Position

This register defines H position and size of window B after scaling (if there is any)

Bit	Description
27:16	**S_B_V_POSITION: Window B V Position** This is specified with respect to the top edge of active display area.
11:0	**S_B_H_POSITION: Window B H Position** This is specified with respect to the left edge of active display area.

DC_WIN_S_B_SIZE_0

Offset: 206h
Read/Write: R/W
Reset: 0000.0000

Window B Size

This register defines V position and size of window B after scaling (if there is any)

Bit	Description
27:16	**S_B_V_SIZE: Window B V Size (lines)** This is the vertical size after scaling.
11:0	**S_B_H_SIZE: Window B H Size (pixels)** This is the horizontal size after scaling.

DC_WIN_S_B_PRESCALED_SIZE_0

Offset: 207h
Read/Write: R/W
Reset: 0000.0000

Window B Pre-scaled Size

This register defines Window B pre-scaled size. The H pre-scaled size is needed to determine how many bytes to fetch from memory per line and this parameter must be programmed exactly as needed taking into account the scaling factor. For planar YUV or YCbCr data formats, this parameter refer to the H pre-scaled of the Y plane.

The total number of lines to be fetched from memory is determined by post-scale V size but V pre-scaled size is needed to 'clamp' the last valid line if the vertical DDA is exactly or slightly beyond the specified V pre-scaled size.

Bit	Description
27:16	**S_B_V_PRESCALED_SIZE: Window B V Pre-scaled Size (lines)**
13:0	**S_B_H_PRESCALED_SIZE: Window B H Pre-scaled Size (bytes)**

DC_WIN_S_B_H_INITIAL_DDA_0

Offset: 208h
Read/Write: R/W
Reset: 0000.0000

Window B H Initial DDA

Bit	Description
15:0	**S_B_H_INITIAL_DDA: Window B H Initial DDA (4.12)** This is typically programmed to 0.0

DC_WIN_S_B_V_INITIAL_DDA_0

Offset: 209h
Read/Write: R/W
Reset: 0000.0000

Window B V Initial DDA

Bit	Description
31:16	**S_B_VB_INITIAL_DDA: Window B V Bottom Field Initial DDA (4.12)** This is typically programmed to 0.0 for non-interlaced source or 0.5 for interlaced source.
15:0	**S_B_VT_INITIAL_DDA: Window B V Top Field Initial DDA (4.12)** **This is typically programmed to 0.0 for both non-interlaced and interlaced sources.**

DC_WIN_S_B_DDA_INCREMENT_0

Offset: 20ah
Read/Write: R/W
Reset: 0000.0000

Window B DDA Increment

DDA increment is typically calculated by dividing (Pre-scaled size - 1) by (Post-scaled size - 1). The result should be round-ed up and expressed as 4.12 format (4-bit integer and 12-bit fraction). If the DDA increment is less than 1.0 then image will be up-scaled and if DDA increment is more than 1.0 then image will be down-scaled.

Bit	Description
31:16	**S_B_V_DDA_INCREMENT: Window B Vertical DDA Increment (4.12)** This should be set to 1.0 if there is no scaling. Maximum value is 15.0 regardless of the number of bytes per pixel.
15:0	**S_B_H_DDA_INCREMENT: Window B Horizontal DDA Increment (4.12)** This should be set to 1.0 if there is no scaling. The maximum value for downscaling depends on the number of bytes per pixel. For 4-byte/pixel modes (32-bpp) the maximum value is 4.0 and for all other modes the maximum value is 8.0.

DC_WIN_S_B_LINE_STRIDE_0

Offset: 20bh
Read/Write: R/W
Reset: 0000.0000

Window B Line Stride

Bit	Description
31:16	**S_B_UV_LINE_STRIDE: Window B Line Stride for Chroma** This is stride (in bytes) for planar YUV or YCbCr data formats for the chroma plane. This is not used (ignored) for other non-planar data formats. This is valid for all window B buffers.
15:0	**S_B_LINE_STRIDE: Window B Line Stride** This is stride (in bytes) for all non-planar data formats. For planar YUV or YCbCr data formats, this is stride (in bytes) for the luma plane. This is valid for all window B buffers.

DC_WIN_S_B_PALETTE_COLOR_EXT_0

Offset: 20ch
Read/Write: R/W
Reset: 0000.0000

Window B Palette Color Extension

Palette extension for 1-bpp, 2-bpp, and 4-bpp. These bits provide the upper most significant bits for indexing the color palette. Supported for window A only.

Bit	Description
7:1	**S_B_PALETTE_COLOR_EXT: Window B Palette Color Extension** bits 7-1 are used for 1-bpp mode bits 7-2 are used for 2-bpp mode bits 7-4 are used for 4-bpp mode

DC_WIN_S_B_BLEND_NOKEY_0

Offset:	20dh
Read/Write:	R/W
Reset:	0000.0000

Blend Control for this window areas where color key is enabled but the pixel color is not within the color key range (color key not match). This is valid for all overlapping condition but only if there is no overlap with other window with higher priority color key enabled and color key not match.

Class: Display Color Keying and Blending

Color keying and blending of the display windows are done prior to cursor blending. Cursor always go on top of the blended windows. Blending is controlled independently on each possible overlap area of the display windows. If 3 windows are enabled there are 7 possible overlap area combinations. For every window in each overlap area combination, color key can be disabled or enabled. Also, for every window in each overlap area combination, there is a corresponding window blend control parameter and a window blend weight parameter. The window blend control parameter is always effective but the window blend weight is not always used. The window blend weight can also be derived from alpha value or from the reverse of other overlapping windows weight.

Color keying has the highest priority for display window blending. Color key consists of a range of color which is searched independently for each windows. If more than 1 windows color keys are enabled then Window A color key has the highest priority, followed by Window B color key, and then followed by Window C color key. Two sets of color key range (Color Key 0 and Color Key 1) can be defined and they are shared for all windows. It is possible to use both color key sets for the same window or for two separate windows. The two sets of color key ranges should not overlap and if they do the overlap colors are treated as if they are part of Color Key 0 and not part of Color Key 1.

Assuming that there is no overlap with other higher priority window with color key not matched, if a window color key is enabled for any overlap condition and the window pixel is not within the color key range (key not match), then the window pixel will not be blended with other overlapping window pixels but it will be weighted. The weight is not dependent on the overlap condition it is controlled by the same set of parameters for all overlapping condition.

Assuming that there is no overlap with other higher priority window with color key not matched, if a window color key is enabled for a particular overlap condition and the window pixel is within the color key range (key match), then the window pixel will be weighted and blended with other overlapping pixels and this is controlled separately for each overlap condition.

Assuming that there is no overlap with other higher priority window with color key not matched, if a window color key is disabled then the window pixel will be blended with other overlapping pixels and this is controlled separately for each overlap condition.

Display Color Key parameters

For RGB444, RGB565, RGB565 mode, color key should be compared prior to color conversion to 24-bpp and unused least significant bits of the pixel are filled with zeros. For palettized mode, color key is compared prior to color palette and the palettized color is compared against the green color key values/mask. For YUV mode, U and V are offset by +128 before performing the color key comparison. In all cases, color key is compared prior to horizontal/vertical scaling filter and prior to digital vibrance control. Both upper and lower values are inclusive.

Bit	Description
23:16	**S_B_BLEND_WEIGHT1_NOKEY: Window blend weight 1 for color key not match areas.** This is used only for alpha weight with 1-bit alpha and alpha value of 1.
15:8	**S_B_BLEND_WEIGHT0_NOKEY: Window blend weight 0 for color key not match areas.** For alpha weight, this is used for 1-bit alpha with value of 0.
0	**S_B_BLEND_CONTROL_NOKEY: Window blend control for color key not match areas.** 0 = Fix_weight 1 = Alpha_weight

DC_WIN_S_B_BLEND_1WIN_0

Offset: 20eh
Read/Write: R/W
Reset: 0000.0000

Blend Control for this window area where it does not overlap with other windows.

Bit	Description
23:16	**S_B_BLEND_WEIGHT1_1WIN: Window blend weight 1 for color key disabled or color key enabled with key matched areas.** This is used only for alpha weight with 1-bit alpha and alpha value of 1.
15:8	**S_B_BLEND_WEIGHT0_1WIN: Window blend weight 0 for color key disabled or color key enabled with key matched areas.** For alpha weight, this is used for 1-bit alpha with value of 0.
2	**S_B_BLEND_CONTROL_1WIN: Window blend control in area where it does not overlap with other windows and either color key disabled or color key enabled with key matched.** 0 = Fix_weight 1 = Alpha_weight
1:0	**S_B_CKEY_ENABLE_1WIN: Window color key enable.** 0 = NOKEY 1 = CKEY0 2 = CKEY1 3 = CKEY01

DC_WIN_S_B_BLEND_2WIN_A_0

Offset: 20fh
Read/Write: R/W
Reset: 0000.0000

Blend Control for this window area that overlaps with window A only.

Bit	Description
23:16	**S_B_BLEND_WEIGHT1_2WIN_A: Window blend weight 1 for color key disabled or color key enabled with key matched areas.** This is used only for alpha weight with 1-bit alpha and alpha value of 1.
15:8	**S_B_BLEND_WEIGHT0_2WIN_A: Window blend weight 0 for color key disabled or color key enabled with key matched areas.** For alpha weight, this is used for 1-bit alpha with value of 0.

Bit	Description
3:2	**S_B_BLEND_CONTROL_2WIN_A: Window blend control in area where either color key disabled or color key enabled with key matched.** 0 = Fix_weight 1 = Alpha_weight 2 = Dependent_weight
1:0	**S_B_CKEY_ENABLE_2WIN_A: Window color key enable.** 0 = NOKEY 1 = CKEY0 2 = CKEY1 3 = CKEY01

DC_WIN_S_B_BLEND_2WIN_C_0

Offset: 210h
Read/Write: R/W
Reset: 0000.0000

Blend Control for this window area that overlaps with window C only.

Bit	Description
23:16	**S_B_BLEND_WEIGHT1_2WIN_C: Window blend weight 1 for color key disabled or color key enabled with key matched areas.** This is used only for alpha weight with 1-bit alpha and alpha value of 1.
15:8	**S_B_BLEND_WEIGHT0_2WIN_C: Window blend weight 0 for color key disabled or color key enabled with key matched areas.** For alpha weight, this is used for 1-bit alpha with value of 0.
3:2	**S_B_BLEND_CONTROL_2WIN_C: Window blend control in area where either color key disabled or color key enabled with key matched.** 0 = Fix_weight 1 = Alpha_weight 2 = Dependent_weight
1:0	**S_B_CKEY_ENABLE_2WIN_C: Window color key enable.** 0 = NOKEY 1 = CKEY0 2 = CKEY1 3 = CKEY01

DC_WIN_S_B_BLEND_3WIN_AC_0

Offset: 211h
Read/Write: R/W
Reset: 0000.0000

Blend Control for this window area that overlaps with windows A and C only.

Bit	Description
23:16	**S_B_BLEND_WEIGHT1_3WIN_AC: Window blend weight 1 for color key disabled or color key enabled with key matched areas.** This is used only for alpha weight with 1-bit alpha and alpha value of 1.

Bit	Description
15:8	**S_B_BLEND_WEIGHT0_3WIN_AC: Window blend weight 0 for color key disabled or color key enabled with key matched areas.** For alpha weight, this is used for 1-bit alpha with value of 0.
3:2	**S_B_BLEND_CONTROL_3WIN_AC: Window blend control in area where either color key disabled or color key enabled with key matched..** 0 = Fix_weight 1 = Alpha_weight 2 = Dependent_weight
1:0	**S_B_CKEY_ENABLE_3WIN_AC: Window color key enable.** 0 = NOKEY 1 = CKEY0 2 = CKEY1 3 = CKEY01

DC_BUF_S_B0_START_ADDR_0

Offset: 300h
Read/Write: R/W
Reset: 0000.0000

Window B Buffer 0 Start Address

In the case where host (software) controls the window buffer, the buffer start addresses are used as byte address to indicate the start address of the buffer. The least significant bit of the start address is not used for 16-bpp non-planar pixel format and the least two significant bits of the start address are not used for 32-bpp non-planar pixel format.

In the case where window buffer is not controlled by host (software), the buffer 0 start addresses are used as buffer 0 start address sent by the controlling module and buffer 1 start addresses are used as buffer strides. For planar YUV, there are only two buffer strides, one for luma and one for chroma, so Buffer 1 Start Address is used for luma buffer stride and Buffer 1 Start Address for U is used for chroma buffer stride and Buffer 1 Start Address for V is not used.

Buffer 0 start addresses must include necessary buffer offset that is due to H/V flip or cropping. Note that in this case, the controlling module will send buffer index and display module will calculate the actual start addresses of the buffer based on buffer 0 start address, the buffer strides and the buffer index.

In the case where window buffer is not controlled by host (software) then a frame may be stored in multiple buffers. In this case, the buffers must be contiguous in the memory because display will use the same luma or chroma line strides for all lines in the frame. Also buffer wraparound must not occur in the middle of the displayed part of the frame. The controlling module will send frame start and frame end indicators (flags) to display module to indicate the beginning and end of frame. Buffer start address is latched when frame start flag is active but the actual buffer start address is not switched until frame end flag is active. In the case where one buffer correspond to one frame then frame start and frame end flag are active everytime a buffer index is sent.

Bit	Description
25:0	**S_B0_START_ADDR: Window B Buffer 0 Start Address** This is a byte address. The LSB is not used for 16-bpp non-planar pixel format and the last 2 LSB are not used for 32-bpp non-planar pixel format. For YUV planar pixel format, this specifies start address for the Y plane.

DC_BUF_S_B0_START_ADDR_U_0

Offset: 301h
Read/Write: R/W
Reset: 0000.0000

Window B Buffer 0 Start Address for U plane

Bit	Description
25:0	**S_B0_START_ADDR_U: Window B Buffer 0 Start Address for U plane** This is a byte address.

DC_BUF_S_B0_START_ADDR_V_0

Offset: 302h
Read/Write: R/W
Reset: 0000.0000

Window B Buffer 0 Start Address for V plane

Bit	Description
25:0	**S_B0_START_ADDR_V: Window B Buffer 0 Start Address for V plane** This is a byte address.

DC_BUF_S_B1_START_ADDR_0

Offset: 300h
Read/Write: R/W
Reset: 0000.0000

Window B Buffer 0 Start Address

In the case where host (software) controls the window buffer, the buffer start addresses are used as byte address to indicate the start address of the buffer. The least significant bit of the start address is not used for 16-bpp non-planar pixel format and the least two significant bits of the start address are not used for 32-bpp non-planar pixel format.

In the case where window buffer is not controlled by host (software), the buffer 0 start addresses are used as buffer 0 start address sent by the controlling module and buffer 1 start addresses are used as buffer strides. For planar YUV, there are only two buffer strides, one for luma and one for chroma, so Buffer 1 Start Address is used for luma buffer stride and Buffer 1 Start Address for U is used for chroma buffer stride and Buffer 1 Start Address for V is not used.

Buffer 0 start addresses must include necessary buffer offset that is due to H/V flip or cropping. Note that in this case, the controlling module will send buffer index and display module will calculate the actual start addresses of the buffer based on buffer 0 start address, the buffer strides and the buffer index.

In the case where window buffer is not controlled by host (software) then a frame may be stored in multiple buffers. In this case, the buffers must be contiguous in the memory because display will use the same luma or chroma line strides for all lines in the frame. Also buffer wraparound must not occur in the middle of the displayed part of the frame. The controlling module will send frame start and frame end indicators (flags) to display module to indicate the beginning and end of frame. Buffer start address is latched when frame start flag is active but the actual buffer start address is not swithced until frame end flag is active. In the case where one buffer correspond to one frame then frame start and frame end flag are active everytime a buffer index is sent.

Bit	Description
25:0	**S_B1_START_ADDR: Window B Buffer 0 Start Address** This is a byte address. The LSB is not used for 16-bpp non-planar pixel format and the last 2 LSB are not used for 32-bpp non-planar pixel format. For YUV planar pixel format, this specifies start address for the Y plane.

DC_BUF_S_B1_START_ADDR_U_0

Offset: 301h
Read/Write: R/W
Reset: 0000.0000

Window B Buffer 0 Start Address for U plane

Bit	Description
25:0	**S_B1_START_ADDR_U: Window B Buffer 0 Start Address for U plane** This is a byte address.

DC_BUF_S_B1_START_ADDR_V_0

Offset: 302h
Read/Write: R/W
Reset: 0000.0000

Window B Buffer 0 Start Address for V plane

Bit	Description
25:0	**S_B1_START_ADDR_V: Window B Buffer 0 Start Address for V plane** This is a byte address.

DC_WIN_S_C_WIN_OPTIONS_0

Offset: 201h
Read/Write: R/W
Reset: 0000.0000

Window B Options

Class: Display Window Settings

Display Window B parameters

Bit	Description
20	**S_C_DV_ENABLE: Window B Digital Vibrance Enable.** 0 = Disable 1 = Enable
18	**S_C_CSC_ENABLE: Window B Color Space Conversion Enable.** 0 = Disable 1 = Enable
16	**S_C_CP_ENABLE: Window B Color Palette Enable.** 0 = Disable 1 = Enable
8	**S_C_H_FILTER_ENABLE: Window B H Filter Enable.** 0 = Disable 1 = Enable
6	**S_C_COLOR_EXPAND: Window B 12/15/16-to-24 bpp color expansion.** 0 = Disable 1 = Enable
2	**S_C_V_DIRECTION: Window B Vertical (Y) drawing Direction.** 0 = Increment 1 = Decrement
0	**S_C_H_DIRECTION: Window B Horizontal (X) drawing Direction.** 0 = Increment 1 = Decrement

DC_WIN_S_C_BYTE_SWAP_0

Offset: 202h
Read/Write: R/W
Reset: 0000.0000

Window B Byte Swap

Bit	Description
1:0	**S_C_BYTE_SWAP: Window B Byte Swap.** 0 = NOSWAP 1 = SWAP2 2 = SWAP4 3 = SWAP4HW

DC_WIN_S_C_BUFFER_CONTROL_0

Offset: 203h
Read/Write: R/W
Reset: 0000.0000

Window B Buffer Control

Bit	Description
2:0	**S_C_BUFFER_CONTROL: Window B Buffer Control.** 0 = HOST 1 = VI 2 = EPP 3 = MPEGE 4 = SB2D

DC_WIN_S_C_COLOR_DEPTH_0

Offset: 204h
Read/Write: R/W
Reset: 0000.0000

Window B Color Depth

For YCbCr data format, Cb and Cr are 8-bit unsigned values. For YUV data format, U and V are 8-bit signed values. YCbCr422R is similar to YCbCr422P but the Cb and Cr are shared vertically. YUV422R is similar to YUV422P but the U and V are shared vertically. YCbCr422RA is same as YCbCr422R in memory and YUV422RA is same as YUV422R in memory but while reading from memory, for YCbCr422RA and YUV422RA, chroma averaging is applied for each pixel pair so that they can be processed as YUV422 by the display pipeline. For YCbCr422R and YUV422R, every other chroma pixels are not used (discarded) by the display pipeline.

Bit	Description
4:0	**S_C_COLOR_DEPTH: Window B Color Depth.** 0 = P1 1 = P2 2 = P4 3 = P8 4 = B4G4R4A4 5 = B5G5R5A 6 = B5G6R5 12 = B8G8R8A8 13 = R8G8B8A8 16 = YCbCr422 17 = YUV422 18 = YCbCr420P 19 = YUV420P 20 = YCbCr422P 21 = YUV422P 22 = YCbCr422R 23 = YUV422R 24 = YCbCr422RA 25 = YUV422RA

DC_WIN_S_C_POSITION_0

Offset: 205h
Read/Write: R/W
Reset: 0000.0000

Window B Position

This register defines H position and size of window B after scaling (if there is any)

Bit	Description
27:16	**S_C_V_POSITION: Window B V Position** This is specified with respect to the top edge of active display area.
11:0	**S_C_H_POSITION: Window B H Position** This is specified with respect to the left edge of active display area.

DC_WIN_S_C_SIZE_0

Offset: 206h
Read/Write: R/W
Reset: 0000.0000

Window B Size

This register defines V position and size of window B after scaling (if there is any)

Bit	Description
27:16	**S_C_V_SIZE: Window B V Size (lines)** This is the vertical size after scaling.
11:0	**S_C_H_SIZE: Window B H Size (pixels)** This is the horizontal size after scaling**.**

DC_WIN_S_C_PRESCALED_SIZE_0

Offset: 207h
Read/Write: R/W
Reset: 0000.0000

Window B Pre-scaled Size

This register defines Window B pre-scaled size. The H pre-scaled size is needed to determine how many bytes to fetch from memory per line and this parameter must be programmed exactly as needed taking into account the scaling factor. For planar YUV or YCbCr data formats, this parameter refer to the H pre-scaled of the Y plane.

The total number of lines to be fetched from memory is determined by post-scale V size but V pre-scaled size is needed to 'clamp' the last valid line if the vertical DDA is exactly or slightly beyond the specified V pre-scaled size.

Bit	Description
27:16	**S_C_V_PRESCALED_SIZE: Window B V Pre-scaled Size (lines)**
13:0	**S_C_H_PRESCALED_SIZE: Window B H Pre-scaled Size (bytes)**

DC_WIN_S_C_H_INITIAL_DDA_0

Offset: 208h
Read/Write: R/W
Reset: 0000.0000

Window B H Initial DDA

Design Note: the first pixel of pre-scaled image is always used to output the first pixel so essentially this is the same as forcing the H Initial DDA integer portion to 1 initially even though user typically programs this to 0. If it makes the implementation easier, it is possible to force software to program the Initial DDA integer portion to 1. Similarly with the V Initial DDA.

Bit	Description
15:0	**S_C_H_INITIAL_DDA: Window B H Initial DDA (4.12)** This is typically programmed to 0.0

DC_WIN_S_C_V_INITIAL_DDA_0

Offset: 209h
Read/Write: R/W
Reset: 0000.0000

Window B V Initial DDA

Bit	Description
31:16	**S_C_VB_INITIAL_DDA: Window B V Bottom Field Initial DDA (4.12)** This is typically programmed to 0.0 for non-interlaced source or 0.5 for interlaced source.
15:0	**S_C_VT_INITIAL_DDA: Window B V Top Field Initial DDA (4.12)** This is typically programmed to 0.0 for both non-interlaced and interlaced sources.

DC_WIN_S_C_DDA_INCREMENT_0

Offset: 20ah
Read/Write: R/W
Reset: 0000.0000

Window B DDA Increment

DDA increment is typically calculated by dividing (Pre-scaled size - 1) by (Post-scaled size - 1). The result should be round-ed up and expressed as 4.12 format (4-bit integer and 12-bit fraction). If the DDA increment is less than 1.0 then image will be up-scaled and if DDA increment is more than 1.0 then image will be down-scaled.

Bit	Description
31:16	**S_C_V_DDA_INCREMENT: Window B Vertical DDA Increment (4.12)** This should be set to 1.0 if there is no scaling. Maximum value is 15.0 regardless of the number of bytes per pixel.
15:0	**S_C_H_DDA_INCREMENT: Window B Horizontal DDA Increment (4.12)** This should be set to 1.0 if there is no scaling. The maximum value for downscaling depends on the number of bytes per pixel. For 4-byte/pixel modes (32-bpp) the maximum value is 4.0 and for all other modes the maximum value is 8.0.

DC_WIN_S_C_LINE_STRIDE_0

Offset: 20bh
Read/Write: R/W
Reset: 0000.0000

Window B Line Stride

Bit	Description
31:16	**S_C_UV_LINE_STRIDE: Window B Line Stride for Chroma** This is stride (in bytes) for planar YUV or YCbCr data formats for the chroma plane. This is not used (ignored) for other non-planar data formats. This is valid for all window B buffers.
15:0	**S_C_LINE_STRIDE: Window B Line Stride** This is stride (in bytes) for all non-planar data formats. For planar YUV or YCbCr data formats, this is stride (in bytes) for the luma plane. This is valid for all window B buffers.

DC_WIN_S_C_PALETTE_COLOR_EXT_0

Offset: 20ch
Read/Write: R/W
Reset: 0000.0000

Window B Palette Color Extension
Palette extension for 1-bpp, 2-bpp, and 4-bpp. These bits provide the upper most significant bits for indexing the color palette.

Supported for window A only.

Bit	Description
7:1	**S_C_PALETTE_COLOR_EXT: Window B Palette Color Extension** bits 7-1 are used for 1-bpp mode bits 7-2 are used for 2-bpp mode bits 7-4 are used for 4-bpp mode

DC_WIN_S_C_BLEND_NOKEY_0

Offset: 20dh
Read/Write: R/W
Reset: 0000.0000

Blend Control for this window areas where color key is enabled but the pixel color is not within the color key range color key not match). This is valid for all overlapping condition but only if there is no overlap with other window with higher priority color key enabled and color key not match.

Class: Display Color Keying and Blending

Color keying and blending of the display windows are done prior to cursor blending. Cursor always go on top of the blended windows. Blending is controlled independently on each possible overlap area of the display windows. If 3 windows are enabled there are 7 possible overlap area combinations. For every window in each overlap area combination, color key can be disabled or enabled. Also, for every window in each overlap area combination, there is a corresponding window blend control parameter and a window blend weight parameter. The window blend control parameter is always effective but the window blend weight is not always used. The window blend weight can also be derived from alpha value or from the reverse of other overlapping windows weight.

Color keying has the highest priority for display window blending. Color key consists of a range of color which is searched independently for each windows. If more than 1 windows color keys are enabled then Window A color key has the highest priority, followed by Window B color key, and then followed by Window C color key. Two sets of color key range (Color Key 0 and Color Key 1) can be defined and they are shared for all windows. It is possible to use both color key sets for the same window or for two separate windows. The two sets of color key ranges should not overlap and if they do the overlap colors are treated as if they are part of Color Key 0 and not part of Color Key 1.

Assuming that there is no overlap with other higher priority window with color key not matched, if a window color key is enabled for any overlap condition and the window pixel is not within the color key range (key not match), then the window pixel will not be blended with other overlapping window pixels but it will be weighted. The weight is not dependent on the overlap condition it is controlled by the same set of parameters for all overlapping condition.

Assuming that there is no overlap with other higher priority window with color key not matched, if a window color key is enabled for a particular overlap condition and the window pixel is within the color key range (key match), then the window pixel will be weighted and blended with other overlapping pixels and this is controlled separately for each overlap condition.

Assuming that there is no overlap with other higher priority window with color key not matched, if a window color key is disabled then the window pixel will be blended with other overlapping pixels and this is controlled separately for each overlap condition.

Display Color Key parameters

For RGB444, RGB565, RGB565 mode, color key should be compared prior to color conversion to 24-bpp and unused least significant bits of the pixel are filled with zeros. For palettized mode, color key is compared prior to color palette and the palettized color is compared against the green color key values/mask. For YUV mode, U and V are offset by +128 before performing the color key comparison. In all cases, color key is compared prior to horizontal/vertical scaling filter and prior to digital vibrance control.

Both upper and lower values are inclusive.

Bit	Description
23:16	**S_C_BLEND_WEIGHT1_NOKEY: Window blend weight 1 for color key not match areas. This is used only for alpha weight with 1-bit alpha and alpha value of 1.**
15:8	**S_C_BLEND_WEIGHT0_NOKEY: Window blend weight 0 for color key not match areas. For alpha weight, this is used for 1-bit alpha with value of 0.**
0	**S_C_BLEND_CONTROL_NOKEY: Window blend control for color key not match areas.** 0 = FIX_WEIGHT 1 = ALPHA_WEIGHT

DC_WIN_S_C_BLEND_1WIN_0

Offset: 20eh
Read/Write: R/W
Reset: 0000.0000

Blend Control for this window area where it does not overlap with other windows.

Bit	Description
23:16	**S_C_BLEND_WEIGHT1_1WIN: Window blend weight 1 for color key disabled or color key enabled with key matched areas.** This is used only for alpha weight with 1-bit alpha and alpha value of 1.
15:8	**S_C_BLEND_WEIGHT0_1WIN: Window blend weight 0 for color key disabled or color key enabled with key matched areas.** For alpha weight, this is used for 1-bit alpha with value of 0.
2	**S_C_BLEND_CONTROL_1WIN: Window blend control in area where it does not overlap with other windows and either color key disabled or color key enabled with key matched.** 0 = FIX_WEIGHT 1 = ALPHA_WEIGHT
1:0	**S_C_CKEY_ENABLE_1WIN: Window color key enable.** 0 = NOKEY 1 = CKEY0 2 = CKEY1 3 = CKEY01

DC_WIN_S_C_BLEND_2WIN_A_0

Offset: 20fh
Read/Write: R/W
Reset: 0000.0000

Blend Control for this window area that overlaps with window A only.

Bit	Description
23:16	**S_C_BLEND_WEIGHT1_2WIN_A: Window blend weight 1 for color key disabled or color key enabled with key matched areas.** This is used only for alpha weight with 1-bit alpha and alpha value of 1.
15:8	**S_C_BLEND_WEIGHT0_2WIN_A: Window blend weight 0 for color key disabled or color key enabled with key matched areas.** For alpha weight, this is used for 1-bit alpha with value of 0.
3:2	**S_C_BLEND_CONTROL_2WIN_A: Window blend control in area where either color key disabled or color key enabled with key matched.** 0 = FIX_WEIGHT 1 = ALPHA_WEIGHT 2 = DEPENDENT_WEIGHT
1:0	**S_C_CKEY_ENABLE_2WIN_A: Window color key enable.** 0 = NOKEY 1 = CKEY0 2 = CKEY1 3 = CKEY01

DC_WIN_S_C_BLEND_2WIN_B_0

Offset: 210h
Read/Write: R/W
Reset: 0000.0000

Blend Control for this window area that overlaps with window B only.

Bit	Description
23:16	**S_C_BLEND_WEIGHT1_2WIN_B: Window blend weight 1 for color key disabled or color key enabled with key matched areas.** This is used only for alpha weight with 1-bit alpha and alpha value of 1.
15:8	**S_C_BLEND_WEIGHT0_2WIN_B: Window blend weight 0 for color key disabled or color key enabled with key matched areas.** For alpha weight, this is used for 1-bit alpha with value of 0.
3:2	**S_C_BLEND_CONTROL_2WIN_B: Window blend control in area where either color key disabled or color key enabled with key matched.** 0 = FIX_WEIGHT 1 = ALPHA_WEIGHT 2 = DEPENDENT_WEIGHT
1:0	**S_C_CKEY_ENABLE_2WIN_B: Window color key enable.** 0 = NOKEY 1 = CKEY0 2 = CKEY1 3 = CKEY01

DC_WIN_S_C_BLEND_3WIN_AB_0

Offset: 211h
Read/Write: R/W
Reset: 0000.0000

Blend Control for this window area that overlaps with windows A & B only.

Bit	Description
23:16	**S_C_BLEND_WEIGHT1_3WIN_AB: Window blend weight 1 for color key disabled or color key enabled with key matched areas.** This is used only for alpha weight with 1-bit alpha and alpha value of 1.
15:8	**S_C_BLEND_WEIGHT0_3WIN_AB: Window blend weight 0 for color key disabled or color key enabled with key matched areas.** For alpha weight, this is used for 1-bit alpha with value of 0.
3:2	**S_C_BLEND_CONTROL_3WIN_AB: Window blend control in area where either color key disabled or color key enabled with key matched.** 0 = FIX_WEIGHT 1 = ALPHA_WEIGHT 2 = DEPENDENT_WEIGHT
1:0	**S_C_CKEY_ENABLE_3WIN_AB: Window color key enable.** 0 = NOKEY 1 = CKEY0 2 = CKEY1 3 = CKEY01

DC_BUF_S_C0_START_ADDR_0

Offset: 300h
Read/Write: R/W
Reset: 0000.0000

Window B Buffer 0 Start Address

In the case where host (software) controls the window buffer, the buffer start addresses are used as byte address to indicate the start address of the buffer. The least significant bit of the start address is not used for 16-bpp non-planar pixel format and the least two significant bits of the start address are not used for 32bpp non-planar pixel format.

In the case where window buffer is not controlled by host (software), the buffer 0 start addresses are used as buffer 0 start address sent by the controlling module and buffer 1 start addresses are used as buffer strides. For planar YUV, there are only two buffer strides, one for luma and one for chroma, so Buffer 1 Start Address is used for luma buffer stride and Buffer 1 Start Address for U is used for chroma buffer stride and Buffer 1 Start Address for V is not used.

Buffer 0 start addresses must include necessary buffer offset that is due to H/V flip or cropping. Note that in this case, the controlling module will send buffer index and display module will calculate the actual start addresses of the buffer based on buffer 0 start address, the buffer strides and the buffer index.

In the case where window buffer is not controlled by host (software) then a frame may be stored in multiple buffers. In this case, the buffers must be contiguous in the memory because display will use the same luma or chroma line strides for all lines in the frame. Also buffer wraparound must not occur in the middle of the displayed part of the frame. The controlling module will send frame start and frame end indicators (flags) to display module to indicate the beginning and end of frame. Buffer start address is latched when frame start flag is active but the actual buffer start address is not swithced until frame end flag is active. In the case where one buffer correspond to one frame then frame start and frame end flag are active everytime a buffer index is sent.

Bit	Description
25:0	**S_C0_START_ADDR: Window B Buffer 0 Start Address** This is a byte address. The LSB is not used for 16-bpp non-planar pixel format and the last 2 LSB are not used for 32bpp non-planar pixel format. For YUV planar pixel format, this specifies start address for the Y plane.

DC_BUF_S_C0_START_ADDR_U_0

Offset: 301h
Read/Write: R/W
Reset: 0000.0000

Window B Buffer 0 Start Address for U plane

Bit	Description
25:0	**S_C0_START_ADDR_U: Window B Buffer 0 Start Address for U plane** This is a byte address.

DC_BUF_S_C0_START_ADDR_V_0

Offset: 302h
Read/Write: R/W
Reset: 0000.0000

Window B Buffer 0 Start Address for V plane

Bit	Description
25:0	**S_C0_START_ADDR_V: Window B Buffer 0 Start Address for V plane** This is a byte address.

DC_BUF_S_C1_START_ADDR_0

Offset: 300h
Read/Write: R/W
Reset: 0000.0000

Window B Buffer 0 Start Address

In the case where host (software) controls the window buffer, the buffer start addresses are used as byte address to indicate the start address of the buffer.

The least significant bit of the start address is not used for 16-bpp non-planar pixel format and the least two significant bits of the start address are not used for 32bpp non-planar pixel format.

In the case where window buffer is not controlled by host (software), the buffer 0 start addresses are used as buffer 0 start address sent by the controlling module and buffer 1 start addresses are used as buffer strides. For planar YUV, there are only two buffer strides, one for luma and one for chroma, so Buffer 1 Start Address is used for luma buffer stride and Buffer 1 Start Address for U is used for chroma buffer stride and Buffer 1 Start Address for V is not used.

Buffer 0 start addresses must include necessary buffer offset that is due to H/V flip or cropping. Note that in this case, the controlling module will send buffer index and display module will calculate the actual start addresses of the buffer based on buffer 0 start address, the buffer strides and the buffer index.

In the case where window buffer is not controlled by host (software) then a frame may be stored in multiple buffers. In this case, the buffers must be contiguous in the memory because display will use the same luma or chroma line strides for all lines in the frame. Also buffer wraparound must not occur in the middle of the displayed part of the frame. The controlling module will send frame start and frame end indicators (flags) to display module to indicate the beginning and end of frame. Buffer start address is latched when frame start flag is active but the actual buffer start address is not swithced until frame end flag is active. In the case where one buffer correspond to one frame, then frame start and frame end flag are active everytime a buffer index is sent.

Bit	Description
25:0	**S_C1_START_ADDR: Window B Buffer 0 Start Address** This is a byte address. The LSB is not used for 16-bpp non-planar pixel format and the last 2 LSB are not used for 32bpp non-planar pixel format. For YUV planar pixel format, this specifies start address for the Y plane.

DC_BUF_S_C1_START_ADDR_U_0

Offset: 301h
Read/Write: R/W
Reset: 0000.0000

Window B Buffer 0 Start Address for U plane

Bit	Description
25:0	**S_C1_START_ADDR_U: Window B Buffer 0 Start Address for U plane** This is a byte address.

DC_BUF_S_C1_START_ADDR_V_0

Offset: 302h
Read/Write: R/W
Reset: 0000.0000

Window B Buffer 0 Start Address for V plane

Bit	Description
25:0	**S_C1_START_ADDR_V: Window B Buffer 0 Start Address for V plane** This is a byte address.

DC_WIN_WIN_OPTIONS_0

Offset: 201h
Read/Write: R/W
Reset: 0000.0000

Window B Options

Class: Display Window Settings

Display Window B parameters

Bit	Description
20	**DV_ENABLE: Window B Digital Vibrance Enable** 0 = Disable 1 = Enable
18	**CSC_ENABLE: Window B Color Space Conversion Enable** 0 = Disable 1 = Enable
16	**CP_ENABLE: Window B Color Palette Enable** 0 = Disable 1 = Enable

Bit	Description
12	**V_FILTER_OPTIMIZE: Window B V Filter Optimization** 0 = Disable 1 = Enable
10	**V_FILTER_ENABLE: Window B V Filter Enable** 0 = Disable 1 = Enable
8	**H_FILTER_ENABLE: Window B H Filter Enable** 0 = Disable 1 = Enable
6	**COLOR_EXPAND: Window B 12/15/16-to-24 bpp color expansion** 0 = Disable 1 = Enable
2	**V_DIRECTION: Window B Vertical (Y) drawing Direction** 0 = Increment 1 = Decrement
0	H_**DIRECTION: Window B Horizontal (X) drawing Direction** 0 = Increment 1 = Decrement

DC_WIN_BYTE_SWAP_0

Offset: 202h
Read/Write: R/W
Reset: 0000.0000

Window B Byte Swap

Bit	Description
1:0	**BYTE_SWAP: Window B Byte Swap.** 0 = Noswap 1 = Swap2 2 = Swap4 3 = Swap4hw

DC_WIN_BUFFER_CONTROL_0

Offset: 203h
Read/Write: R/W
Reset: 0000.0000

Window B Buffer Control

Bit	Description
2:0	**BUFFER_CONTROL: Window B Buffer Control.** 0 = Host 1 = VI 2 = EPP 3 = MPEGE 4 = VS2D

DC_WIN_COLOR_DEPTH_0

Offset: 204h
Read/Write: R/W
Reset: 0000.0000

Window B Color Depth

For YCbCr data format, Cb and Cr are 8-bit unsigned values. For YUV data format, U and V are 8-bit signed values. YCbCr422R is similar to YCbCr422P but the Cb and Cr are shared vertically. YUV422R is similar to YUV422P but the U and V are shared vertically. YCbCr422RA is same as YCbCr422R in memory and YUV422RA is same as YUV422R in memory but while reading from memory, for YCbCr422RA and YUV422RA, chroma averaging is applied for each pixel pair so that they can be processed as YUV422 by the display pipeline. For YCbCr422R and YUV422R, every other chroma pixels are not used (discarded) by the display pipeline.

Bit	Description
4:0	**COLOR_DEPTH: Window B Color Depth** 0 = P1 1 = P2 2 = P4 3 = P8 4 = B4G4R4A4 5 = B5G5R5A 6 = B5G6R5 12 = B8G8R8A8 13 = R8G8B8A8 16 = YCbCr422 17 = YUV422 18 = YCbCr420P 19 = YUV420P 20 = YCbCr422P 21 = YUV422P 22 = YCbCr422R 23 = YUV422R 24 = YCbCr422RA 25 = YUV422RA

DC_WIN_POSITION_0

Offset: 205h
Read/Write: R/W
Reset: 0000.0000

Window B Position

This register defines H position and size of window B after scaling (if there is any)

Bit	Description
27:16	**V_POSITION: Window B V Position** This is specified with respect to the top edge of active display area.
11:0	**H_POSITION: Window B H Position** This is specified with respect to the left edge of active display area.

DC_WIN_SIZE_0

Offset: 206h
Read/Write: R/W
Reset: 0000.0000

Window B Size

This register defines V position and size of window B after scaling (if there is any)

Bit	Description
27:16	**V_SIZE: Window B V Size (lines)** This is the vertical size after scaling.
11:0	**H_SIZE: Window B H Size (pixels)** This is the horizontal size after scaling.

DC_WIN_PRESCALED_SIZE_0

Offset: 207h
Read/Write: R/W
Reset: 0000.0000

Window B Pre-scaled Size

This register defines Window B pre-scaled size.

The H pre-scaled size is needed to determine how many bytes to fetch from memory per line and this parameter must be programmed exactly as needed taking into account the scaling factor. For planar YUV or YCbCr data formats, this parameter refer to the H pre-scaled of the Y plane.

The total number of lines to be fetched from memory is determined by post-scale V size but V pre-scaled size is needed to 'clamp' the last valid line if the vertical DDA is exactly or slightly beyond the specified V pre-scaled size.

Bit	Description
27:16	**V_PRESCALED_SIZE: Window B V Pre-scaled Size (lines)**
13:0	**H_PRESCALED_SIZE: Window B H Pre-scaled Size (bytes)**

DC_WIN_H_INITIAL_DDA_0

Offset: 208h
Read/Write: R/W
Reset: 0000.0000

Window B H Initial DDA

Design Note: the first pixel of pre-scaled image is always used to output the first pixel so essentially this is the same as forcing the H Initial DDA integer portion to 1 initially even though user typically programs this to 0. If it makes the implementation easier, it is possible to force software to program the Initial DDA integer portion to 1. Similarly with the V Initial DDA.

Bit	Description
15:0	**H_INITIAL_DDA: Window B H Initial DDA (4.12)** This is typically programmed to 0.0

DC_WIN_V_INITIAL_DDA_0

Offset: 209h
Read/Write: R/W
Reset: 0000.0000

Window B V Initial DDA

Bit	Description
31:16	**VB_INITIAL_DDA: Window B V Bottom Field Initial DDA (4.12)** This is typically programmed to 0.0 for non-interlaced source or 0.5 for interlaced source.
15:0	**VT_INITIAL_DDA: Window B V Top Field Initial DDA (4.12)** This is typically programmed to 0.0 for both non-interlaced and interlaced sources.

DC_WIN_DDA_INCREMENT_0

Offset: 20ah
Read/Write: R/W
Reset: 0000.0000

Window B DDA Increment

DDA increment is typically calculated by dividing (Pre-scaled size - 1) by (Post-scaled size - 1). The result should be round-ed up and expressed as 4.12 format (4-bit integer and 12-bit fraction). If the DDA increment is less than 1.0 then image will be up-scaled and if DDA increment is more than 1.0 then image will be down-scaled.

Bit	Description
31:16	**V_DDA_INCREMENT: Window B Vertical DDA Increment (4.12)** This should be set to 1.0 if there is no scaling. Maximum value is 15.0 regardless of the number of bytes per pixel.
15:0	**H_DDA_INCREMENT: Window B Horizontal DDA Increment (4.12)** This should be set to 1.0 if there is no scaling. The maximum value for downscaling depends on the number of bytes per pixel. For 4-byte/pixel modes (32-bpp) the maximum value is 4.0 and for all other modes the maximum value is 8.0.

DC_WIN_LINE_STRIDE_0

Offset: 20bh
Read/Write: R/W
Reset: 0000.0000

Window B Line Stride

Bit	Description
31:16	**UV_LINE_STRIDE: Window B Line Stride for Chroma** This is stride (in bytes) for planar YUV or YCbCr data formats for the chroma plane. This is not used (ignored) for other non-planar data formats. This is valid for all window B buffers.
15:0	**LINE_STRIDE: Window B Line Stride** This is stride (in bytes) for all non-planar data formats. For planar YUV or YCbCr data formats, this is stride (in bytes) for the luma plane. This is valid for all window B buffers.

DC_WIN_PALETTE_COLOR_EXT_0

Offset: 20ch
Read/Write: R/W
Reset: 0000.0000

Window B Palette Color Extension

Palette extension for 1-bpp, 2-bpp, and 4-bpp. These bits provide the upper most significant bits for indexing the color palette. Supported for window A only. XXX should ifdef for window A, but currently window B spec is assumed to be a superset.

Bit	Description
7:1	**PALETTE_COLOR_EXT: Window B Palette Color Extension** bits 7-1 are used for 1-bpp mode bits 7-2 are used for 2-bpp mode bits 7-4 are used for 4-bpp mode

DC_WIN_BLEND_NOKEY_0

Offset: 20dh
Read/Write: R/W
Reset: 0000.0000

Blend Control for this window areas where color key is enabled but the pixel color is not within the color key range (color key not match). This is valid for all overlapping condition but only if there is no overlap with other window with higher priority color key enabled and color key not match.

Class: Display Color Keying and Blending

Color keying and blending of the display windows are done prior to cursor blending. Cursor always go on top of the blended windows. Blending is controlled independently on each possible overlap area of the display windows. If 3 windows are enabled there are 7 possible overlap area combinations. For every window in each overlap area combination, color key can be disabled or enabled. Also, for every window in each overlap area combination, there is a corresponding window blend control parameter and a window blend weight parameter. The window blend control parameter is always effective but the window blend weight is not always used. The window blend weight can also be derived from alpha value or from the reverse of other overlapping windows weight.

Color keying has the highest priority for display window blending. Color key consists of a range of color which is searched independently for each windows. If more than 1 windows color keys are enabled then Window A color key has the highest priority, followed by Window B color key, and then followed by Window C color key. Two sets of color key range (Color Key 0 and Color Key 1) can be defined and they are shared for all windows. It is possible to use both color key sets for the same window or for two separate windows. The two sets of color key ranges should not overlap and if they do the overlap colors are treated as if they are part of Color Key 0 and not part of Color Key 1.

Assuming that there is no overlap with other higher priority window with color key not matched, if a window color key is enabled for any overlap condition and the window pixel is not within the color key range (key not match), then the window pixel will not be blended with other overlapping window pixels but it will be weighted. The weight is not dependent on the overlap condition it is controlled by the same set of parameters for all overlapping condition.

Assuming that there is no overlap with other higher priority window with color key not matched, if a window color key is enabled for a particular overlap condition and the window pixel is within the color key range (key match), then the window pixel will be weighted and blended with other overlapping pixels and this is controlled separately for each overlap condition.

Assuming that there is no overlap with other higher priority window with color key not matched, if a window color key is disabled then the window pixel will be blended with other overlapping pixels and this is controlled separately for each overlap condition.

Display Color Key parameters

For B4G4R4A4, B5G6R5A, B5G6R5 mode, color key should be compared prior to color conversion to 24bpp and unused least significant bits of the pixel are filled with zeros. For palettized mode, color key is compared prior to color palette and the palettized color is compared against the green color key values/mask. For YUV mode, U and V are offset by +128 before performing the color key comparison. In all cases, color key is compared prior to horizontal/vertical scaling filter and prior to digital vibrance control.

Both upper and lower values are inclusive.

Bit	Description
23:16	**BLEND_WEIGHT1_NOKEY: Window blend weight 1 for color key not match areas.** This is used only for alpha weight with 1-bit alpha and alpha value of 1.
15:8	**BLEND_WEIGHT0_NOKEY: Window blend weight 0 for color key not match areas.** For alpha weight, this is used for 1-bit alpha with value of 0.
0	**BLEND_CONTROL_NOKEY: Window blend control for color key not match areas.** 0 = FIX_WEIGHT 1 = ALPHA_WEIGHT

DC_WIN_BLEND_1WIN_0

Offset: 20eh
Read/Write: R/W
Reset: 0000.0000

Blend Control for this window area where it does not overlap with other windows.

Bit	Description
23:16	**BLEND_WEIGHT1_1WIN: Window blend weight 1 for color key disabled or color key enabled with key matched areas.** This is used only for alpha weight with 1-bit alpha and alpha value of 1.
15:8	**BLEND_WEIGHT0_1WIN: Window blend weight 0 for color key disabled or color key enabled with key matched areas.** For alpha weight, this is used for 1-bit alpha with value of 0.
2	**BLEND_CONTROL_1WIN: Window blend control in area where it does not overlap with other windows and either color key disabled or color key enabled with key matched.** 0 = FIX_WEIGHT 1 = ALPHA_WEIGHT

Bit	Description
1:0	**CKEY_ENABLE_1WIN: Window color key enable.** 0 = Nokey 1 = Ckey0 2 = Ckey1 3 = Ckey01

DC_WIN_BLEND_2WIN_A_0

Offset: 20fh
Read/Write: R/W
Reset: 0000.0000

Blend Control for this window area that overlaps with window A only.

Bit	Description
23:16	**BLEND_WEIGHT1_2WIN_A: Window blend weight 1 for color key disabled or color key enabled with key matched areas.** This is used only for alpha weight with 1-bit alpha and alpha value of 1.
15:8	**BLEND_WEIGHT0_2WIN_A: Window blend weight 0 for color key disabled or color key enabled with key matched areas.** For alpha weight, this is used for 1-bit alpha with value of 0.
3:2	**BLEND_CONTROL_2WIN_A: Window blend control in area where either color key disabled or color key enabled with key matched.** 0 = Fix_weight 1 = Alpha_weight 2 = Dependent_weight
1:0	**CKEY_ENABLE_2WIN_A: Window color key enable.** 0 = Nokey 1 = Ckey0 2 = Ckey1 3 = Ckey01

DC_WIN_BLEND_2WIN_C_0

Offset: 210h
Read/Write: R/W
Reset: 0000.0000

Blend Control for this window area that overlaps with window C only.

Bit	Description
23:16	**BLEND_WEIGHT1_2WIN_C: Window blend weight 1 for color key disabled or color key enabled with key matched areas.** This is used only for alpha weight with 1-bit alpha and alpha value of 1.
15:8	**BLEND_WEIGHT0_2WIN_C: Window blend weight 0 for color key disabled or color key enabled with key matched areas.** For alpha weight, this is used for 1-bit alpha with value of 0.

Bit	Description
3:2	**BLEND_CONTROL_2WIN_C: Window blend control in area where either color key disabled or color key enabled with key matched..** 0 = Fix_weight 1 = Alpha_weight 2 = Dependent_weight
1:0	**CKEY_ENABLE_2WIN_C: Window color key enable.** 0 = Nokey 1 = Ckey0 2 = Ckey1 3 = Ckey01

DC_WIN_BLEND_3WIN_AC_0

Offset: 211h
Read/Write: R/W
Reset: 0000.0000

Blend Control for this window area that overlaps with windows A and C only.

Bit	Description
23:16	**BLEND_WEIGHT1_3WIN_AC: Window blend weight 1 for color key disabled or color key enabled with key matched areas.** This is used only for alpha weight with 1-bit alpha and alpha value of 1.
15:8	**BLEND_WEIGHT0_3WIN_AC: Window blend weight 0 for color key disabled or color key enabled with key matched areas.** For alpha weight, this is used for 1-bit alpha with value of 0.
3:2	**BLEND_CONTROL_3WIN_AC: Window blend control in area where either color key disabled or color key enabled with key matched.** 0 = Fix_weight 1 = Alpha_weight 2 = Dependent_weight
1:0	**CKEY_ENABLE_3WIN_AC: Window color key enable.** 0 = Nokey 1 = Ckey0 2 = Ckey1 3 = Ckey01

DC_WINC_COLOR_PALETTE_0

Offset: 400h:4ffh
Read/Write: WO
Reset: 0000.0000

Window B Color Palette

This is an array of 256 identical register entries; the register fields below apply to each entry.

Color palette

This is used for palettized data format (color depth of 8-bpp or less) or for gamma correction for non-palettized data formats (color depth of more than 8-bpp). Each window has its own color palette which consists of three 256x8 register file which can be written by host and indexed (read) by the window. For palettized data format less than 8-bpp the pixel data is aligned to least significant bits of the palette index (address) and the remaining upper bits are filled with the corresponding bits of the Palette Color Extension. For example, for 4-bpp mode, the pixel data occupies bits 3-0 of the palette index and bits 7-4 of the palette index are set to bits 7-4 of the Palette Color Extension. Note that host read is assumed to be not needed - software can cache the color palette in system memory.

The same color palettes are used for both primary and secondary displays and they can be written as either primary or secondary display color palette.

Bit	Description
23:16	**COLOR_PALETTE_B: Blue Color Palette**
15:8	**COLOR_PALETTE_G: Green Color Palette**
7:0	**COLOR_PALETTE_R: Red Color Palette**

DC_WINC_DV_CONTROL_0

Offset: 500h
Read/Write: R/W
Reset: 0000.0000

Window B Digital Vibrance Control

Digital Vibrance (DV) control

If enabled, Digital Vibrance is applied after H and V scaling and after color palette or color space conversion logic but before color keying multiplexer and before cursor multiplexer.

After DV, new R = R + (2R - G - B) * FR, where FR is fraction from 0 to 7/8
After DV, new G = G + (2G - R - B) * FG, where FG is fraction from 0 to 7/8

After DV, new B = B + (2B - R - G) * FB, where FB is fraction from 0 to 7/8

Bit	Description
18:16	**DV_CONTROL_B: Digital Vibrance control for B**
10:8	**DV_CONTROL_G: Digital Vibrance control for G**
2:0	**DV_CONTROL_R: Digital Vibrance control for R**

DC_WINC_H_FILTER_P00_0

Offset: 501h
Read/Write: R/W
Reset: 0000.0000

Window B Horizontal Filter phase 00

Horizontal scaling filter coefficients.

Horizontal scaling filter is a 6-tap filter with 4-bit positional phase.

Coefficients 0 and 5 are 3-bit signed value ranging from -4 to 3.
Coefficients 1 and 4 are 5-bit signed value ranging from -16 to 15.
Coefficients 2 and 3 are 8-bit unsigned value ranging from 0 to 128.
Coefficient 0 is the multiplier for the earliest pixel (P0) in the group of 6-pixel and coefficient 5 is the multiplier for the latest pixel (P5) in the group. The output pixel positional phase is defined as centered in P2 if the positional phase is 0 or proportionally in between P2 and P3 if the positional phase is larger than 0.

Sum of all coefficients for each phase should be 128 typically and software should never program the the sum of all coefficients for a phase to be more than 128. For each horizontal positional phase, the 6 filter coefficients requires 32 reg bits. Note that color value ranges from 0 to 255 for Y, R, G, B and -128 to 127 for U and V. The same H filter coefficients are used for both primary and secondary displays and they can be written as either primary or secondary display H filter coefficients.

Bit	Description
31:29	**H_FILTER_P00C5: Phase 00 coefficient 5 (typically 0)**
28:24	**H_FILTER_P00C4: Phase 00 coefficient 4 (typically 0)**
23:16	**H_FILTER_P00C3: Phase 00 coefficient 3 (typically 0)**
15:8	**H_FILTER_P00C2: Phase 00 coefficient 2 (typically 128)**
7:3	**H_FILTER_P00C1: Phase 00 coefficient 1 (typically 0)**
2:0	**H_FILTER_P00C0: Phase 00 coefficient 0 (typically 0)**

DC_WINC_H_FILTER_P01_0

Offset: 502h
Read/Write: R/W
Reset: 0000.0000

Window B Horizontal Filter phase 01

Bit	Description
31:29	**H_FILTER_P01C5: Phase 01 coefficient 5 (typically 1)**
28:24	**H_FILTER_P01C4: Phase 01 coefficient 4 (typically -2)**
23:16	**H_FILTER_P01C3: Phase 01 coefficient 3 (typically 8)**
15:8	**H_FILTER_P01C2: Phase 01 coefficient 2 (typically 124)**
7:3	**H_FILTER_P01C1: Phase 01 coefficient 1 (typically -4)**
2:0	**H_FILTER_P01C0: Phase 01 coefficient 0 (typically 1)**

DC_WINC_H_FILTER_P02_0

Offset: 503h
Read/Write: R/W
Reset: 0000.0000

Window B Horizontal Filter phase 02

Bit	Description
31:29	**H_FILTER_P02C5: Phase 02 coefficient 5 (typically 1)**
28:24	**H_FILTER_P02C4: Phase 02 coefficient 4 (typically -5)**
23:16	**H_FILTER_P02C3: Phase 02 coefficient 3 (typically 17)**
15:8	**H_FILTER_P02C2: Phase 02 coefficient 2 (typically 122)**
7:3	**H_FILTER_P02C1: Phase 02 coefficient 1 (typically -8)**
2:0	**H_FILTER_P02C0: Phase 02 coefficient 0 (typically 1)**

DC_WINC_H_FILTER_P03_0

Offset: 504h
Read/Write: R/W
Reset: 0000.0000

Window B Horizontal Filter phase 03

Bit	Description
31:29	**H_FILTER_P03C5: Phase 03 coefficient 5 (typically 2)**
28:24	**H_FILTER_P03C4: Phase 03 coefficient 4 (typically -7)**
23:16	**H_FILTER_P03C3: Phase 03 coefficient 3 (typically 27)**
15:8	**H_FILTER_P03C2: Phase 03 coefficient 2 (typically 115)**
7:3	**H_FILTER_P03C1: Phase 03 coefficient 1 (typically -11)**
2:0	**H_FILTER_P03C0: Phase 03 coefficient 0 (typically 2)**

DC_WINC_H_FILTER_P04_0

Offset: 505h
Read/Write: R/W
Reset: 0000.0000

Window B Horizontal Filter phase 04

Bit	Description
31:29	**H_FILTER_P04C5: Phase 04 coefficient 5 (typically 2)**
28:24	**H_FILTER_P04C4: Phase 04 coefficient 4 (typically -9)**
23:16	**H_FILTER_P04C3: Phase 04 coefficient 3 (typically 37)**
15:8	**H_FILTER_P04C2: Phase 04 coefficient 2 (typically 109)**
7:3	**H_FILTER_P04C1: Phase 04 coefficient 1 (typically -13)**
2:0	**H_FILTER_P04C0: Phase 04 coefficient 0 (typically 2)**

DC_WINC_H_FILTER_P05_0

Offset: 506h
Read/Write: R/W
Reset: 0000.0000

Window B Horizontal Filter phase 05

Bit	Description
31:29	**H_FILTER_P05C5: Phase 05 coefficient 5 (typically 2)**
28:24	**H_FILTER_P05C4: Phase 05 coefficient 4 (typically -11)**
23:16	**H_FILTER_P05C3: Phase 05 coefficient 3 (typically 47)**
15:8	**H_FILTER_P05C2: Phase 05 coefficient 2 (typically 102)**
7:3	**H_FILTER_P05C1: Phase 05 coefficient 1 (typically -15)**
2:0	**H_FILTER_P05C0: Phase 05 coefficient 0 (typically 3)**

DC_WINC_H_FILTER_P06_0

Offset: 507h
Read/Write: R/W
Reset: 0000.0000

Window B Horizontal Filter phase 06

Bit	Description
31:29	**H_FILTER_P06C5: Phase 06 coefficient 5 (typically 3)**
28:24	**H_FILTER_P06C4: Phase 06 coefficient 4 (typically -13)**
23:16	**H_FILTER_P06C3: Phase 06 coefficient 3 (typically 56)**
15:8	**H_FILTER_P06C2: Phase 06 coefficient 2 (typically 94)**
7:3	**H_FILTER_P06C1: Phase 06 coefficient 1 (typically -15)**
2:0	**H_FILTER_P06C0: Phase 06 coefficient 0 (typically 3)**

DC_WINC_H_FILTER_P07_0

Offset: 508h
Read/Write: R/W
Reset: 0000.0000

Window B Horizontal Filter phase 07

Bit	Description
31:29	**H_FILTER_P07C5: Phase 07 coefficient 5 (typically 3)**
28:24	**H_FILTER_P07C4: Phase 07 coefficient 4 (typically -14)**
23:16	**H_FILTER_P07C3: Phase 07 coefficient 3 (typically 67)**
15:8	**H_FILTER_P07C2: Phase 07 coefficient 2 (typically 85)**
7:3	**H_FILTER_P07C1: Phase 07 coefficient 1 (typically -16)**
2:0	**H_FILTER_P07C0: Phase 07 coefficient 0 (typically 3)**

DC_WINC_H_FILTER_P08_0

Offset: 509h
Read/Write: R/W
Reset: 0000.0000

Window B Horizontal Filter phase 08

Bit	Description
31:29	**H_FILTER_P08C5: Phase 08 coefficient 5 (typically 3)**
28:24	**H_FILTER_P08C4: Phase 08 coefficient 4 (typically -15)**
23:16	**H_FILTER_P08C3: Phase 08 coefficient 3 (typically 76)**
15:8	**H_FILTER_P08C2: Phase 08 coefficient 2 (typically 76)**
7:3	**H_FILTER_P08C1: Phase 08 coefficient 1 (typically -15)**
2:0	**H_FILTER_P08C0: Phase 08 coefficient 0 (typically 3)**

DC_WINC_H_FILTER_P09_0

Offset: 50ah
Read/Write: R/W
Reset: 0000.0000

Window B Horizontal Filter phase 09

Bit	Description
31:29	**H_FILTER_P09C5: Phase 09 coefficient 5 (typically 3)**
28:24	**H_FILTER_P09C4: Phase 09 coefficient 4 (typically -16)**
23:16	**H_FILTER_P09C3: Phase 09 coefficient 3 (typically 85)**
15:8	**H_FILTER_P09C2: Phase 09 coefficient 2 (typically 67)**
7:3	**H_FILTER_P09C1: Phase 09 coefficient 1 (typically -14)**
2:0	**H_FILTER_P09C0: Phase 09 coefficient 0 (typically 3)**

DC_WINC_H_FILTER_P0A_0

Offset: 50bh
Read/Write: R/W
Reset: 0000.0000

Window B Horizontal Filter phase 0A

Bit	Description
31:29	**H_FILTER_P0AC5: Phase 0A coefficient 5 (typically 3)**
28:24	**H_FILTER_P0AC4: Phase 0A coefficient 4 (typically -15)**
23:16	**H_FILTER_P0AC3: Phase 0A coefficient 3 (typically 94)**
15:8	**H_FILTER_P0AC2: Phase 0A coefficient 2 (typically 56)**
7:3	**H_FILTER_P0AC1: Phase 0A coefficient 1 (typically -13)**
2:0	**H_FILTER_P0AC0: Phase 0A coefficient 0 (typically 3)**

DC_WINC_H_FILTER_P0B_0

Offset: 50ch
Read/Write: R/W
Reset: 0000.0000

Window B Horizontal Filter phase 0B

Bit	Description
31:29	**H_FILTER_P0BC5: Phase 0B coefficient 5 (typically 3)**
28:24	**H_FILTER_P0BC4: Phase 0B coefficient 4 (typically -15)**
23:16	**H_FILTER_P0BC3: Phase 0B coefficient 3 (typically 102)**
15:8	**H_FILTER_P0BC2: Phase 0B coefficient 2 (typically 47)**
7:3	**H_FILTER_P0BC1: Phase 0B coefficient 1 (typically -11)**
2:0	**H_FILTER_P0BC0: Phase 0B coefficient 0 (typically 2)**

DC_WINC_H_FILTER_P0C_0

Offset: 50dh
Read/Write: R/W
Reset: 0000.0000

Window B Horizontal Filter phase 0C

Bit	Description
31:29	**H_FILTER_P0CC5: Phase 0C coefficient 5 (typically 2)**
28:24	**H_FILTER_P0CC4: Phase 0C coefficient 4 (typically -13)**
23:16	**H_FILTER_P0CC3: Phase 0C coefficient 3 (typically 109)**
15:8	**H_FILTER_P0CC2: Phase 0C coefficient 2 (typically 37)**
7:3	**H_FILTER_P0CC1: Phase 0C coefficient 1 (typically -9)**
2:0	**H_FILTER_P0CC0: Phase 0C coefficient 0 (typically 2)**

DC_WINC_H_FILTER_P0D_0

Offset: 50eh
Read/Write: R/W
Reset: 0000.0000

Window B Horizontal Filter phase 0D

Bit	Description
31:29	**H_FILTER_P0DC5: Phase 0D coefficient 5 (typically 2)**
28:24	**H_FILTER_P0DC4: Phase 0D coefficient 4 (typically -11)**
23:16	**H_FILTER_P0DC3: Phase 0D coefficient 3 (typically 115)**
15:8	**H_FILTER_P0DC2: Phase 0D coefficient 2 (typically 27)**
7:3	**H_FILTER_P0DC1: Phase 0D coefficient 1 (typically -7)**
2:0	**H_FILTER_P0DC0: Phase 0D coefficient 0 (typically 2)**

DC_WINC_H_FILTER_P0E_0

Offset: 50fh
Read/Write: R/W
Reset: 0000.0000

Window B Horizontal Filter phase 0E

Bit	Description
31:29	**H_FILTER_P0EC5: Phase 0E coefficient 5 (typically 1)**
28:24	**H_FILTER_P0EC4: Phase 0E coefficient 4 (typically -8)**
23:16	**H_FILTER_P0EC3: Phase 0E coefficient 3 (typically 122)**
15:8	**H_FILTER_P0EC2: Phase 0E coefficient 2 (typically 17)**
7:3	**H_FILTER_P0EC1: Phase 0E coefficient 1 (typically -5)**
2:0	**H_FILTER_P0EC0: Phase 0E coefficient 0 (typically 1)**

DC_WINC_H_FILTER_P0F_0

Offset: 510h
Read/Write: R/W
Reset: 0000.0000

Window B Horizontal Filter phase 0F

Bit	Description
31:29	**H_FILTER_P0FC5: Phase 0F coefficient 5 (typically 1)**
28:24	**H_FILTER_P0FC4: Phase 0F coefficient 4 (typically -4)**
23:16	**H_FILTER_P0FC3: Phase 0F coefficient 3 (typically 124)**
15:8	**H_FILTER_P0FC2: Phase 0F coefficient 2 (typically 8)**
7:3	**H_FILTER_P0FC1: Phase 0F coefficient 1 (typically -2)**
2:0	**H_FILTER_P0FC0: Phase 0F coefficient 0 (typically 1)**

DC_WINC_CSC_YOF_0

Offset: 511h
Read/Write: R/W
Reset: 0000.0000

Window B CSC Y Offset

Color Space Conversion coefficients.

The CSC can be used for YUV to RGB conversion with brightness and hue/saturation control. The CSC can only be enabled for Window B controlled by (P/S)_B_CSC_ENABLE register bits. For Y color, the Y offset is applied first and saturation (clipping) is performed immediately after the Y offset is applied.

R = sat(KYRGB * sat(Y + YOF) + KUR * U + KVR * V)
G = sat(KYRGB * sat(Y + YOF) + KUG * U + KVG * V)
B = sat(KYRGB * sat(Y + YOF) + KUB * U + KVB * V)

Saturation and rounding is performed in the range of 0 to 255 for the above equations.

Typical values are:
YOF = -16.000, KYRGB = 1.1644
KUR = 0.0000, KVR = -1.5960
KUG = -0.3918, KVG = -0.8130
KUB = 2.0172, KVB = 0.0000
KUR and KVB are typically 0.0000 but they may be programmed non-zero for hue rotation.

The CSC can also take RGB input, in which case YOF, KVB, KUG, KUR should be programmed to 0 and KYRGB will be forced to 0 by the hardware for generating R and B. KYRGB will not be forced to 0 for generating G. KVR, KYRGB, and KUB can be programmed to 1.0 or used as gain control for R, G, B correspondingly.

Note that color value ranges from 0 to 255 for Y, R, G, B and -128 to 127 for U and V.

Bit	Description
7:0	**CSC_YOF: Y Offset in s.7.0 format**

DC_WINC_CSC_KYRGB_0

Offset: 512h
Read/Write: R/W
Reset: 0000.0000

Window B CSC Y Coefficient (gain) for RGB

Bit	Description
9:0	**CSC_KYRGB: Y Gain for R, G, B colors in 2.8 format**

DC_WINC_CSC_KUR_0

Offset: 513h
Read/Write: R/W
Reset: 0000.0000

Window B CSC U coefficent for R

Bit	Description
10:0	**CSC_KUR: U coefficients for R in s.2.8 format**

DC_WINC_CSC_KVR_0

Offset: 514h
Read/Write: R/W
Reset: 0000.0000

Window B CSC V coefficent for R

Bit	Description
10:0	**CSC_KVR: V coefficients for R in s.2.8 format**

DC_WINC_CSC_KUG_0

Offset: 515h
Read/Write: R/W
Reset: 0000.0000

Window B CSC U coefficent for G

Bit	Description
9:0	**CSC_KUG: U coefficients for G in s.1.8 format**

DC_WINC_CSC_KVG_0

Offset: 516h
Read/Write: R/W
Reset: 0000.0000

Window B CSC V coefficent for G

Bit	Description
9:0	**CSC_KVG: V coefficients for G in s.1.8 format**

DC_WINC_CSC_KUB_0

Offset: 517h
Read/Write: R/W
Reset: 0000.0000

Window B CSC U coefficent for B

Bit	Description
10:0	**CSC_KUB: U coefficients for B in s.2.8 format**

DC_WINC_CSC_KVB_0

Offset: 518h
Read/Write: R/W
Reset: 0000.0000

Window B CSC V coefficent for B

Bit	Description
10:0	**CSC_KVB: V coefficients for B in s.2.8 format**

DC_WINC_V_FILTER_P00_0

Offset: 519h
Read/Write: R/W
Reset: 0000.0000

Window B Vertical Filter phase 00

Vertical scaling filter coefficients.

Vertical scaling filter is a 2-tap filter with 4-bit positional phase.

Coefficients 0 and 1 are 8-bit unsigned value ranging from 0 to 128.
Coefficient 0 is the multiplier for the earlier pixel (P0) in the group of 2-pixel and coefficient 1 is the multiplier for the later pixel (P1) in the group. The output pixel positional phase is defined as centered in P0 if the positional phase is 0 or proportionally in between P0 and P1 if the positional phase is larger than 0. Sum of all coefficients for each phase should be 128 typically therefore coefficient 1 can be calculated from (1 - coefficient 0) and therefore only coefficient 0 is programmed.

For each vertical positional phase, the filter coefficient requires 8 reg bits. Note that color value ranges from 0 to 255 for Y, R, G, B and -128 to 127 for U and V. The same V filter coefficients are used for both primary and secondary displays and they can be written as either primary or secondary display V filter coefficients.

Bit	Description
7:0	**V_FILTER_P00C0: Phase 00 coefficient 0 (typically 128)**

DC_WINC_V_FILTER_P01_0

Offset: 51ah
Read/Write: R/W
Reset: 0000.0000

Window B Vertical Filter phase 01

Bit	Description
7:0	**V_FILTER_P01C0: Phase 01 coefficient 0 (typically 120)**

DC_WINC_V_FILTER_P02_0

Offset: 51bh
Read/Write: R/W
Reset: 0000.0000

Window B Vertical Filter phase 02

Bit	Description
7:0	**V_FILTER_P02C0: Phase 02 coefficient 0 (typically 112)**

DC_WINC_V_FILTER_P03_0

Offset: 51ch
Read/Write: R/W
Reset: 0000.0000

Window B Vertical Filter phase 03

Bit	Description
7:0	**V_FILTER_P03C0: Phase 03 coefficient 0 (typically 104)**

DC_WINC_V_FILTER_P04_0

Offset: 51dh
Read/Write: R/W
Reset: 0000.0000

Window B Vertical Filter phase 04

Bit	Description
7:0	**V_FILTER_P04C0: Phase 04 coefficient 0 (typically 96)**

DC_WINC_V_FILTER_P05_0

Offset: 51eh
Read/Write: R/W
Reset: 0000.0000

Window B Vertical Filter phase 05

Bit	Description
7:0	**V_FILTER_P05C0: Phase 05 coefficient 0 (typically 88)**

DC_WINC_V_FILTER_P06_0

Offset: 51fh
Read/Write: R/W
Reset: 0000.0000

Window B Vertical Filter phase 06

Bit	Description
7:0	**V_FILTER_P06C0: Phase 06 coefficient 0 (typically 80)**

DC_WINC_V_FILTER_P07_0

Offset: 520h
Read/Write: R/W
Reset: 0000.0000

Window B Vertical Filter phase 07

Bit	Description
7:0	**V_FILTER_P07C0: Phase 07 coefficient 0 (typically 72)**

DC_WINC_V_FILTER_P08_0

Offset: 521h
Read/Write: R/W
Reset: 0000.0000

Window B Vertical Filter phase 08

Bit	Description
7:0	**V_FILTER_P08C0: Phase 08 coefficient 0 (typically 64)**

DC_WINC_V_FILTER_P09_0

Offset: 522h
Read/Write: R/W
Reset: 0000.0000

Window B Vertical Filter phase 09

Bit	Description
7:0	**V_FILTER_P09C0: Phase 09 coefficient 0 (typically 56)**

DC_WINC_V_FILTER_P0A_0

Offset: 523h
Read/Write: R/W
Reset: 0000.0000

Window B Vertical Filter phase 0A

Bit	Description
7:0	**V_FILTER_P0AC0: Phase 0A coefficient 0 (typically 48)**

DC_WINC_V_FILTER_P0B_0

Offset: 524h
Read/Write: R/W
Reset: 0000.0000

Window B Vertical Filter phase 0B

Bit	Description
7:0	**V_FILTER_P0BC0: Phase 0B coefficient 0 (typically 40)**

DC_WINC_V_FILTER_P0C_0

Offset: 525h
Read/Write: R/W
Reset: 0000.0000

Window B Vertical Filter phase 0C

Bit	Description
7:0	**V_FILTER_P0CC0: Phase 0C coefficient 0 (typically 32)**

DC_WINC_V_FILTER_P0D_0

Offset: 526h
Read/Write: R/W
Reset: 0000.0000

Window B Vertical Filter phase 0D

Bit	Description
7:0	**V_FILTER_P0DC0: Phase 0D coefficient 0 (typically 24)**

DC_WINC_V_FILTER_P0E_0

Offset: 527h
Read/Write: R/W
Reset: 0000.0000

Window B Vertical Filter phase 0E

Bit	Description
7:0	**V_FILTER_P0EC0: Phase 0E coefficient 0 (typically 16)**

DC_WINC_V_FILTER_P0F_0

Offset: 528h
Read/Write: R/W
Reset: 0000.0000

Window B Vertical Filter phase 0F

Bit	Description
7:0	**V_FILTER_P0FC0: Phase 0F coefficient 0 (typically 8)**

7.9 EMC Registers

EMC_CTXSW_0

Offset: 000h
Read/Write: R/W
Reset: fffff000.f000

Context switch register. Should be common to all modules. Includes the current channel/class (which is writable by SW) and the next channel/class (which the hardware sets when it receives a context switch.)

Context switch works like this:

Any context switch request triggers an interrupt to the host and causes the new channel/class to be stored in NEXT_CHANNEL/NEXT_CLASS (see vmod/chexample). SW sees that there is a context switch interrupt and does the necessary operations to make the module ready to receive traffic from the new context. It clears the context switch interrupt and writes CURR_CHANNEL/CLASS to the same value as NEXT_CHANNEL/CLASS, which causes a context switch acknowledge packet to be sent to the host. This completes the context switch and allows the host to continue sending data to the module.

Bit	Description
31:28	**NEXT_CHANNEL: Next requested channel**
25:16	**NEXT_CLASS: Next requested class**
15:12	**CURR_CHANNEL: Current working channel, reset to 'invalid'**
9:0	**CURR_CLASS: Current working class**

EMC_INTSTATUS_0

Offset: 001h
Read/Write: RO
Reset: 0000.0000

Context Switch Interrupt Status Register. Clear on 1-write.

Bit	Description
0	**CTXSW_INT** 0 = CLEAR 1 = SET

EMC_DBG_0

Offset: 002h
Read/Write: R/W
Reset: 0100.0000

Debug Register

The DBG register is used to reconfigure the EMC during debug or chip testing.

The MRS register should be enabled when a non-mobile DRAM is used because they require a 200 cycle delay between the DLL reset and any read commands. The PERIODIC_QRST field specifies whether or not to periodic reset the FBIO read-data fifo during normal operation. The periodic resets can be used for graceful recovery from an intermittent failure condition; only the initial reset is absolutely required.

The CFG_PRIORITY field determines the priority of cfg accesses to the DRAM. Setting this register to ENABLED gives DRAM config cycles (refresh, mrs, emrs, etc.) higher priority over real time requestors. The DISABLED setting gives the real time requestors higher priority than DRAM config cycles.

Bit	Description
24	**CFG_PRIORITY** 0 = DISABLED 1 = ENABLED
5	**PERIODIC_QRST** 0 = DISABLED 1 = ENABLED
4	**MRS_WAIT** 0 = MRS_2 1 = MRS_256

EMC_CFG_0

Offset: 003h
Read/Write: R/W
Reset: 0300.ff00

Configuration Register

The CFG registers are used to configure the external memory interface.

When enabled, PRE_IDLE_EN preemptively closes all of the banks after the EMC has been idle for PRE_IDLE_CYCLES cycles and there are banks open. PRE_IDLE_EN can be enabled if violating tRAS max is an issue. Note that 0 is an illegal setting for PRE_IDLE_CYCLES.

The BYPASS_FIFOS signals allows the synchronization fifos between the MC and EMC to be bypassed. The AUTO_PRE_RD, AUTO_PRE_WR bits enable auto-precharge in the EMC. These bits, when set to DISABLE, will override the settings in the MC register. Otherwise, they permit clients to make auto-precharge requests as specified by the Memory Controller.

DRAM_ACPD allows the DRAM controller to perform opportunistic active powerdown control using the CKE pin on the DRAM. The behavior of the powerdown control logic is controlled by the config bits in TIMING5.

Bit	Description
29	**DRAM_ACPD:** 0 = No_powerdown 1 = Active_powerdown
25	**AUTO_PRE_WR:** 0 = Disabled 1 = Enabled
24	**AUTO_PRE_RD:** 0 = Disabled 1 = Enabled
16	**BYPASS_FIFOS**
15:8	**PRE_IDLE_CYCLES**
0	**PRE_IDLE_EN:** 0 = Disabled 1 = Enabled

EMC_REFCTRL_0

Offset: 004h
Read/Write: R/W
Reset: 0000.0000

Refresh Control Register

The REFCTRL register allows the bios to enable or disable refresh requests. It should be enabled after the initialization sequence is completed.

Bit	Description
31	**REF_VALID:** 0 = Disabled 1 = Enabled

EMC_PIN_0

Offset: 005h
Read/Write: R/W
Reset: 0000.0001

Controls state of selected DRAM pins The PIN register allows the bios to control the state of the selected external DRAM pins.

PIN_CKE selects the level of the CKE pin. This can be used to place the DRAM in power down state. PIN_DQM is used to always mask DRAM writes. This pin should only be used for initialization. Certain DRAM vendors (e.g., Samsung), require the DQM to be high during initialization. The register value should be set to NORMAL after the initialization sequence.

Bit	Description
4	**PIN_DQM** 0 = Normal 1 = Inactive
0	**PIN_CKE** 0 = Powerdown 1 = Normal

EMC_TIMING0_0

Offset: 006h
Read/Write: R/W
Reset: 3f3f.3f3f

Timing Control Register 0

The TIMING0 register specifies parameters for the DRAM device.

The RC field specifies the row cycle time. This is the minimum number of cycles between activate commands to the same bank. The RFC field specifies the auto refresh cycle time. This is the minimum number of cycles between an auto refresh command and a subsequent auto refresh or activate command.

The RAS field specifies the row active time. This is the minimum number of cycles between an activate command and a precharge command to the same bank. The RP field specifies the row precharge time. This is the minimum number of cycles between a precharge command and an activate command to
the same bank.

Bit	Description
29:24	**RP**
21:16	**RAS**
13:8	**RFC**
5:0	**RC**

EMC_TIMING1_0

Offset: 007h
Read/Write: R/W
Reset: 1f1f.1f1f

Timing Control Register 1

The TIMING1 register specifies parameters for the DRAM devices.

R2something and W2something registers are independent of burst length, and are relative to the last virtual CAS_ of a command. We start counting from the last data transfer as related to where CAS_ would be if BL = 4.

The R2W field specifies the minimum number of cycles from any read command to any write command, irrespective of bank. This parameter guarantees the read->write turn-around time on the bus. Set to ((CL+1)-WL + R2W_bus_turnaround_clks). The W2R field specifies the minimum number of cycles from a write command to a read command. Set to ((WL+1) + tWTR). The R2P field specifies the minimum number of cycles from a read command to a precharge command for the same bank. Set to 1 clk.

The W2P field specifies the minimum number of cycles from a write command to a precharge command for the same bank. Set to ((WL+1) + tWR).

Bit	Description
28:24	**W2P**
20:16	**R2P**
12:8	**W2R**
4:0	**R2W**

EMC_TIMING2_0

Offset: 008h
Read/Write: R/W
Reset: 000f.1f1f

Timing Control Register 2

The TIMING2 register specifies parameters for the DRAM devices.

The RCD field specifies the ras to cas delay. RCD_RD is the minimum number of cycles between an activate command and a read command to the same bank. RCD_WR is the minimum number of cycles between an activate command and a write command to the same bank.

The RRD field specifies the Bank X Act to Bank Y Act command delay. WDV is the number of cycles to post (delay) write data from being asserted to the rams. Set to 0 for SDR, DDR1 and some DDR2 operation; set to "read latency" (CL - 1) on JEDEC DDR2 operation. For SDR, the delay obtained is programmed value. For other types, the delay obtained is programmed value + 1.

Bit	Description
27:24	**WDV**
19:16	**RRD**
12:8	**WR_RCD**
4:0	**RD_RCD**

EMC_TIMING3_0

Offset: 009h
Read/Write: R/W
Reset: 0008.7102

Timing Control Register 3

The TIMING3 register specifies parameters for the DRAM devices.

The QRST,QUSE,RDV fields specify the delays from read to internal timing signals. These fields should be set as follows:

QUSE = CAS_LATENCY - 1 for SDR and non-mobile DDR = CAS_LATENCY - 2 for mobile DDR
QRST = CAS_LATENCY - 2
QSAFE >= RDV - QRST
RDV = CAS_LATENCY + 5

Because DDR uses a tristating clock, the DQS, a method is needed to deal with the ambiguity of when DQS is tristate. For SDR, a method is needed to determine which read clocks are valid. GoForce 5500 only supports one such method: QUSE. This signal tells the chip when to look for read return data clocked by DQS.

QSAFE is the time from a read command to when it is safe to issue a QRST (delayed by the QRST parameter). When PERIODIC_QRST is enabled, the QSAFE parameter is intended to guarantee that the QRSTs will not interfere with pending reads (e.g. the queue is empty). This field must be set to at least (RDV - QRST). RDV is the time from read command to latching the read data from the pad macros. Thus it is the read latency register. This register value is not negotiable, it will work with the right value and will not with the wrong value.

It is sequential timing, not combinational (i.e., maybe the silicon is fast enough type timing.

Bit	Description
20:16	**RDV: 23 = MAX**
15:12	**QSAFE**
11:8	**QRST**
3:0	**QUSE**

EMC_TIMING4_0

Offset: 00ah
Read/Write: R/W
Reset: 0000.001f

Timing Control Register 4

The TIMING4 register specifies parameters for the DRAM devices.

The REFRESH field specifies the interval between refresh requests. This time is represented (((REFRESH + 1) * 32) * clk_period). The DRAM refresh clock in this case is the EMC clock. For example, if our clock frequency is 100MHz and the refresh interval for our DRAM is 32ms, REFRESH works out to be 99 according to the formula above, which gives a programmed value of 0x63.

Bit	Description
15:5	**REFRESH**
4:0	**REFRESH_LO: 31 = MAX**

EMC_TIMING5_0

Offset: 00bh
Read/Write: R/W
Reset: 01ff.ffff

Timing Control Register 5

The TIMING5 register specifies parameters for the DRAM devices.

The PDEX2WR field is used to specify the timing delay from exit of powerdown mode to a write command. The PDEX2RD field is used to specify the timing delay from exit of powerdown mode to a read command. The PCHG2PDEN field is used to specify the timing delay from a precharge command to powerdown entry. The RW2PDEN field is used to specify the timing delay from a read/write command to powerdown entry. The ACT2PDEN field is used to specify the timing delay from an activate, mrs or emrs command to powerdown entry.

The AR2PDEN field is used to specify the timing delay from an autorefresh command to powerdown entry.

Bit	Description
24:20	**AR2PDEN**
19:16	**ACT2PDEN**
15:12	**RW2PDEN**
11:8	**PCHG2PDEN**
7:4	**PDEX2RD**
3:0	**PDEX2WR**

EMC_MRS_0

Offset: 00ch
Read/Write: R/W
Reset: 0000.0000

MRS value

The MRS register is used to configure the frame buffer memory DRAM on both external banks.

BA0, BA1 are used to address MRS or EMRS registers in DRAM. Drive both BA0 and BA1 low for MRS. Although this register can also program EMRS, use the EMRS register so that the HW registers can shadow what is in the DRAM.

Bit	Description
21:20	**MRS_BA**
12:0	**MRS_ADR**

EMC_EMRS_0

Offset: 00dh
Read/Write: R/W
Reset: 0000.0000

EMRS value

The EMRS register is used to send an EMRS command to both external banks.

Bit	Description
21:20	**EMRS_BA**
12:0	**EMRS_ADR**

EMC_REF_0

Offset: 00eh
Read/Write: R/W
Reset: 0000.0000

Refresh command register

The REF register allows the bios to issue refresh commands. This is done to ensure proper DRAM initialization.

The REF_CMD field causes the hardware to perform (REF_NUM + 1) refresh cycles.

Bit	Description
15:8	**REF_NUM**
0	**REF_CMD**

EMC_PRE_0

Offset: 00fh
Read/Write: R/W
Reset: 0000.0000

Precharge command register

The PRE register allows the bios to issue a precharge all command. This command may be used to ensure proper DRAM initialization.

The PRE_CMD field causes the hardware to perform a PRECHARGE to all DRAM banks.

Bit	Description
0	**PRE_CMD**

EMC_NOP_0

Offset: 010h
Read/Write: R/W
Reset: 0000.0000

NOP command register

The NOP register allows the bios to issue an explicit nop command. This command may be used to ensure proper DRAM initialization.

The NOP_CMD field causes the hardware to perform a NOP to all DRAM banks.

Bit	Description
0	**NOP_CMD**

EMC_SELF_REF_0

Offset: 011h
Read/Write: R/W
Reset: 0000.0000

SELF REFRESH command register

The SELF_REF register allows the bios to issue self refresh commands.

The SELF_REF_CMD field issues a self refresh command to the DRAM. While CMD:ENABLED, the CKE pin is held deasserted. The DRAM will ignore all accesses until CMD:DISABLED. Note. The CMD:ENABLED state will override the PIN:CKE setting.

Bit	Description
0	**SELF_REF_CMD:** 0 = Disabled 1 = Enabled

EMC_DPD_0

Offset: 012h
Read/Write: R/W
Reset: 0000.0000

Deep Power Down command register

The DPD register allows the bios to issue a deep power down command. The DPD_CMD field causes the hardware to issue the deep power down command. (Burst Terminate with cke low).

Bit	Description
0	**DPD_CMD:** 0 = Disabled 1 = Enabled

EMC_CMDQ_0

Offset: 013h
Read/Write: R/W
Reset: 0000.1304

Command Queue Depth register

The CMDQ register controls the depth of the bank command queues. There are 3 queues. One for R/W, one for precharge, and one for activate. They are read by the DRAM controller. The hardware has queue depths of 8, 4, 4 respectively. Restricting queue depth improves the latency time of the real time requestor.

Bit	Description
14:12	**PRE_DEPTH**
10:8	**ACT_DEPTH**
3:0	**RW_DEPTH**

EMC_FBIO_CFG1_0

Offset: 014h
Read/Write: R/W
Reset: 0000.0000

FBIO configuration register

The FBIO_CFG1 register controls the width of the DQ enable window when driving write data. DEN_EARLY determines whether the output enable is the same width as data (DEN_EARLY=0) or 1/2 bit time wider on either end (DEN_EARLY=1)

Bit	Description
16	**CFG_DEN_EARLY:** 0 = Disable 1 = Enable

EMC_FBIO_DQSIB_DLY_0

Offset: 015h
Read/Write: R/W
Reset: 0000.0000

FBIO configuration register

The FBIO_DQSIB_DELAY register sets the trimmers for inbound dqs delay on each of the inbound byte lanes.

Bit	Description
30:24	**CFG_DQSIB_DLY_BYTE_3**
22:16	**CFG_DQSIB_DLY_BYTE_2**
14:8	**CFG_DQSIB_DLY_BYTE_1**
6:0	**CFG_DQSIB_DLY_BYTE_0**

EMC_FBIO_SPARE_0

Offset: 016h
Read/Write: R/W
Reset: 0000.0000

FBIO spare register

The FBIO_SPARE register provides extra bits for future needs. They are connected to FBIO at the "common" pad macro, which is near the center of each fbio partition.

Bit	Description
31:0	**CFG_FBIO_SPARE**

EMC_FBIO_CFG5_0

Offset: 017h
Read/Write: R/W
Reset: 0000.0000

FBIO configuration Register

The FBIO_CFG5 register controls the FBIO I/O cells, configuring them for SDR or DDR1 operation. It also must be programmed with the location of the DRAM device, whether internal or external. The DRAM_TYPE specifies whether the DRAM is SDR or DDR1. The DRAM_LOCATION specifies whether the DRAM is internal (in the same package or external (off-package).

Bit	Description
4	**DRAM_LOCATION:** 0 = Internal 1 = External
0	**DRAM_TYPE:** 0 = Sdr 1 = Ddr1

EMC_FBIO_WRPTR_EQ_2_0

Offset: 018h
Read/Write: RO
Reset: 0000.0000

FBIO wrptr register

The FBIO_WRPTR_EQ_2 read-only register provides a way to observe the state of the write pointers to each of the fifos in the fbio byte lanes. Each byte lane contains it's own 6 entry inbound fifo, clocked by the incoming DQS strobe. Each burst of 4 should advance the write pointer by 2 positions. After resetting all the fifo pointers via qrst, if we issue a single read, then all the write pointers should have advanced from 0 -> 2. By reading back this register, we can see if this has happened correctly or not. An algorithm can be developed that identifies the optimal setting for QUSE and QUSE_LATE that maximizes the window for proper quse gating of the incoming strobes. The FB_DYN_RD_VAL provides a way to observe the values generated by the dynamic trimmer circuit for the inbound dqs trimmer.

Bit	Description
23:4	**FB_DYN_RD_VAL**
3:0	**FB_WRPTR_EQ_2**

EMC_FBIO_QUSE_DLY_0

Offset: 019h
Read/Write: R/W
Reset: 0000.0000

QUSE delay register

QUSE_LATE and QUSE_DLY determine how much added delay fbio should add to the QUSE path. QUSE needs to align (approximately) with the incoming DQS (DDR1) or feedback clock (SDR) in order to qualify that signal, since the incoming DQS/feedback clock is not always valid.

DRAMC can position QUSE with m2clk granularity (2 bit times). For DDR1 only, QUSE_LATE provides finer granularity of 1/2 an m4clk cycle (1/2 bit time). The amount of delay we add will be primarily a function of the round trip wire delay to/from the DRAM. Other portions of the delay (driver and receiver delay) are compensated for by delay through a non-bonded QUSE pad cell.

QUSE_DLY controls trimmers in each byte lane. This also allows for variation in the propagation delay from the "common" pad macro where the QUSE logic lives out to each of the 4 byte lanes.

Bit	Description
28:24	**CFG_QUSE_DLY_BYTE_3**
20:16	**CFG_QUSE_DLY_BYTE_2**
12:8	**CFG_QUSE_DLY_BYTE_1**
4:0	**CFG_QUSE_DLY_BYTE_0**

EMC_FBIO_CFG6_0

Offset: 01ah
Read/Write: R/W
Reset: 0000.0002

FBIO configuration register

Bit	Description
2:0	**CFG_QUSE_LATE**

EMC_OBS_FBIO_SIGNALS_0

Offset: 000h
Read/Write: RO
Reset: 0000.0000

Bit	Description
31:28	**FB_DQM1_N**
27:24	**FB_DQM0_N**
23:12	**FB_ADR_N: 12 LSB's**
11:10	**FB_BANK_N**
9	**FB_WDQS_N**
8	**FB_QRST_N_**
7	**FB_QPOP_N_**
6	**FB_QUSE_RESET_N_**
5	**FB_QUSE_N_**
4	**FB_WE_N_**
3	**FB_CAS_N_**
2	**FB_RAS_N_**
1	**FB_CS_N_**
0	**FB_CKE_N**

Notice

ALL NVIDIA DESIGN SPECIFICATIONS, REFERENCE BOARDS, FILES, DRAWINGS, DIAGNOSTICS, LISTS, AND OTHER DOCUMENTS (TOGETHER AND SEPARATELY, "MATERIALS") ARE BEING PROVIDED "AS IS." NVIDIA MAKES NO WARRANTIES, EXPRESSED, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE MATERIALS, AND EXPRESSLY DISCLAIMS ALL IMPLIED WARRANTIES OF NONINFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE.

Information furnished is believed to be accurate and reliable. However, NVIDIA Corporation assumes no responsibility for the consequences of use of such information or for any infringement of patents or other rights of third parties that may result from its use. No license is granted by implication or otherwise under any patent or patent rights of NVIDIA Corporation. Specifications mentioned in this publication are subject to change without notice. This publication supersedes and replaces all information previously supplied. NVIDIA Corporation products are not authorized for use as critical components in life support devices or systems without express written approval of NVIDIA Corporation.

NVIDIA and its licensors retain all intellectual property and proprietary rights in and to the Materials provided herein. Any use, reproduction, disclosure or distribution of the Materials without an express license agreement from NVIDIA, or without NVIDIA's prior written consent, is strictly prohibited

IN NO EVENT SHALL NVIDIA BE LIABLE FOR ANY SPECIAL, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THE MATERIALS PROVIDED HEREIN.

Trademarks

NVIDIA, the NVIDIA logo, and GoForce 5500 are trademarks or registered trademarks of NVIDIA Corporation. Other company and product names may be trademarks of the respective companies with which they are associated.

Copyright

© 2005 - 2006 by NVIDIA Corporation. All rights reserved