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!

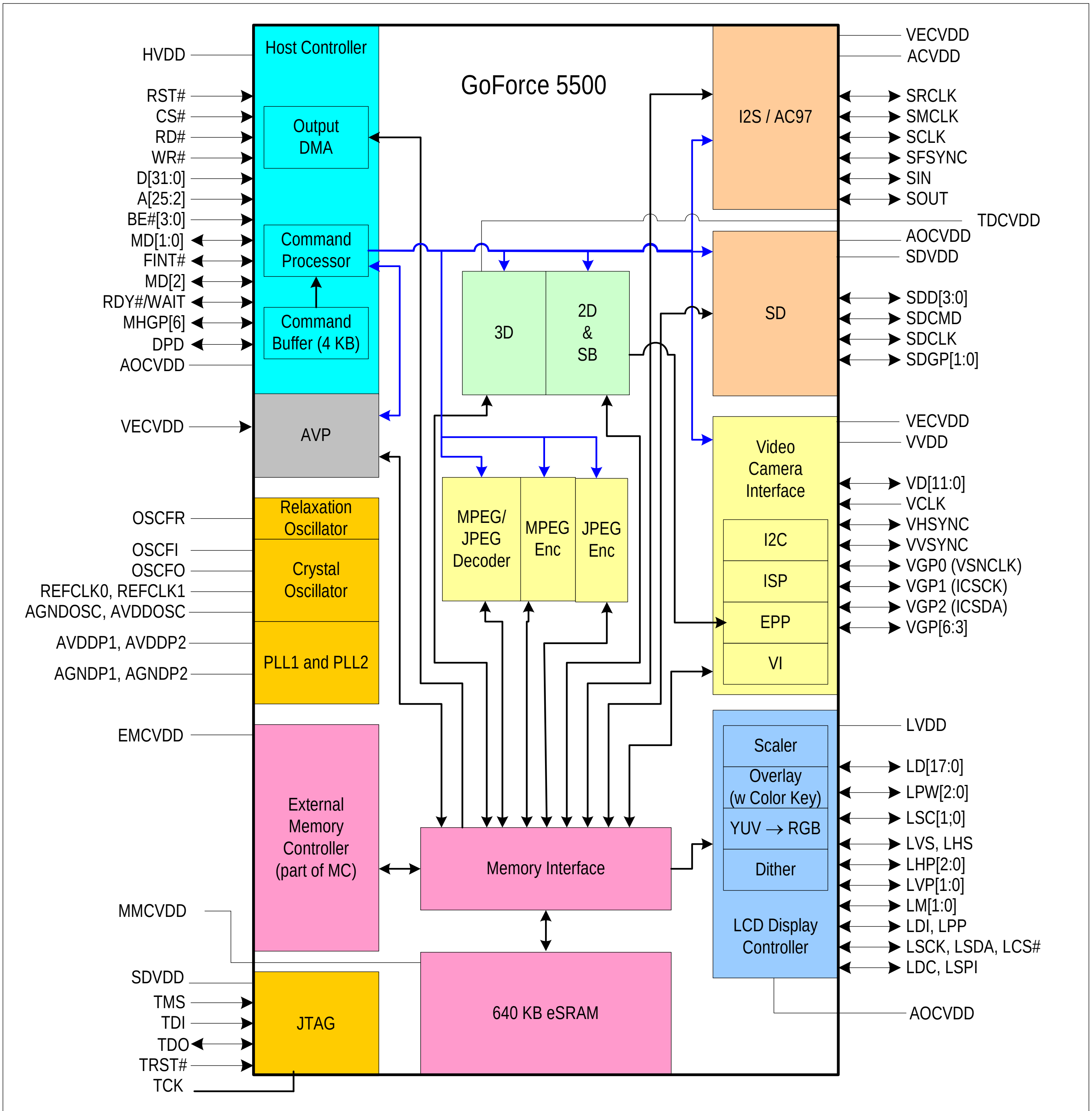
Enhanced alpha-blending modes make it possible to show semi-transparent menus over the top of video.

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
  - I²S/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

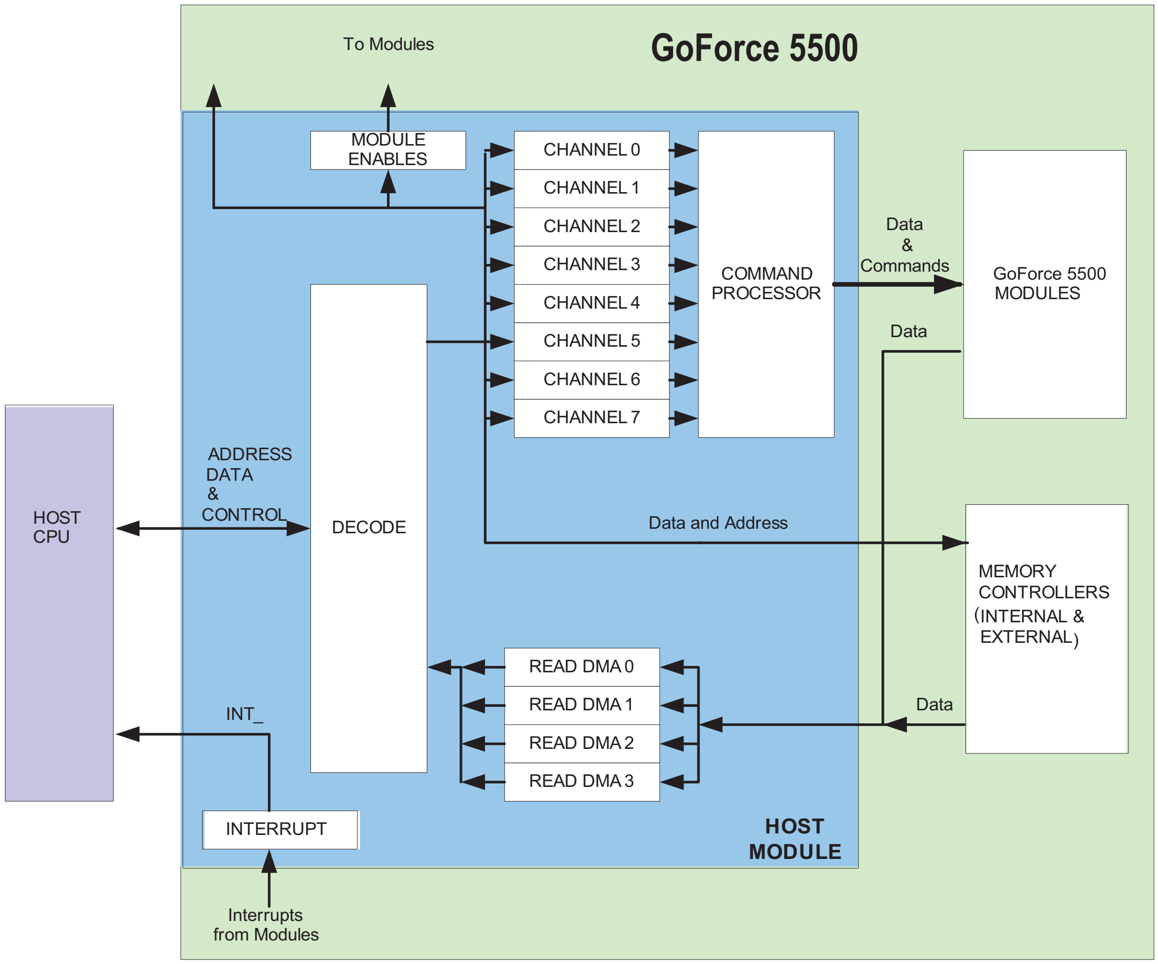
- Asynchronous interfacing to Type A and Type C Host CPUs with
  - 16bit bus width
  - 32bit bus width
- Fixed-cycle interface
  - Does not use the RDY pin (no handshake)
- Handshake interface (uses RDY pin)
- Micro-class interface with smart command processor
  - Eight low-priority channels with deep 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

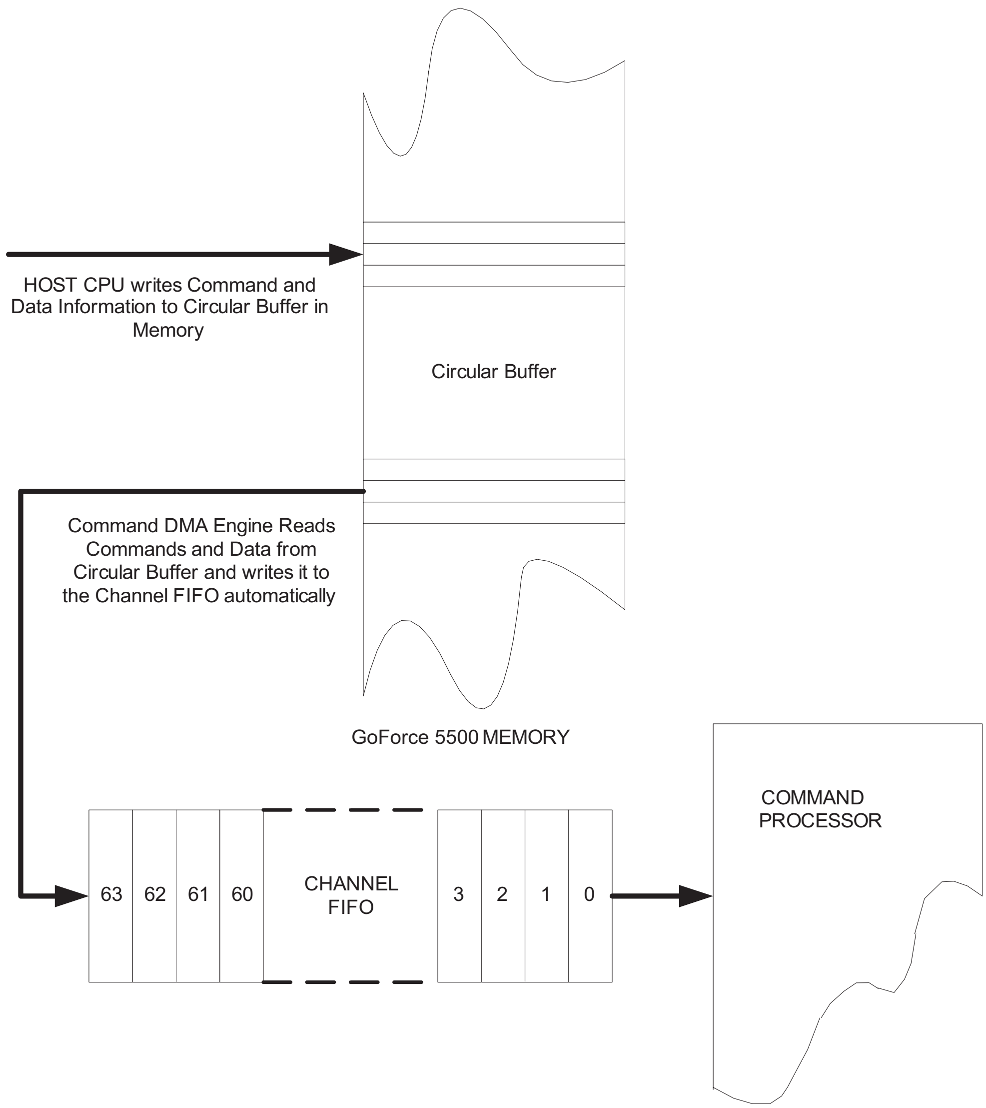
The GoForce 5500 does not have a chip-specific API. Instead, the GoForce 5500 features a Micro-class Interface with a smart-command processor. The programming interface is based on writing to offsets within each class that implements a function, rather than to specific register offsets and formats. Not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

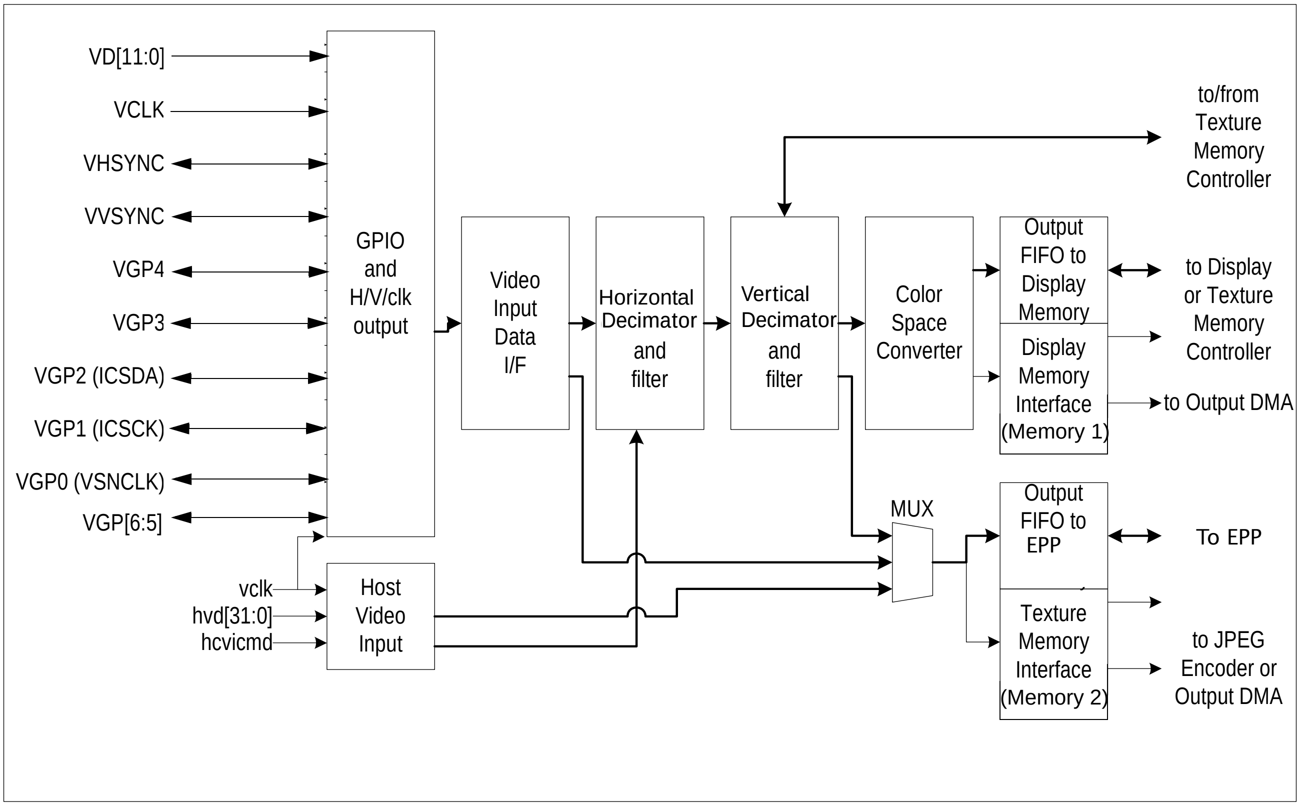
2. Pins VD1 and VD0 should have their inputs disabled. They may be used as general purpose outputs. Bayer[9:0] data is processed by the ISP module.

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)

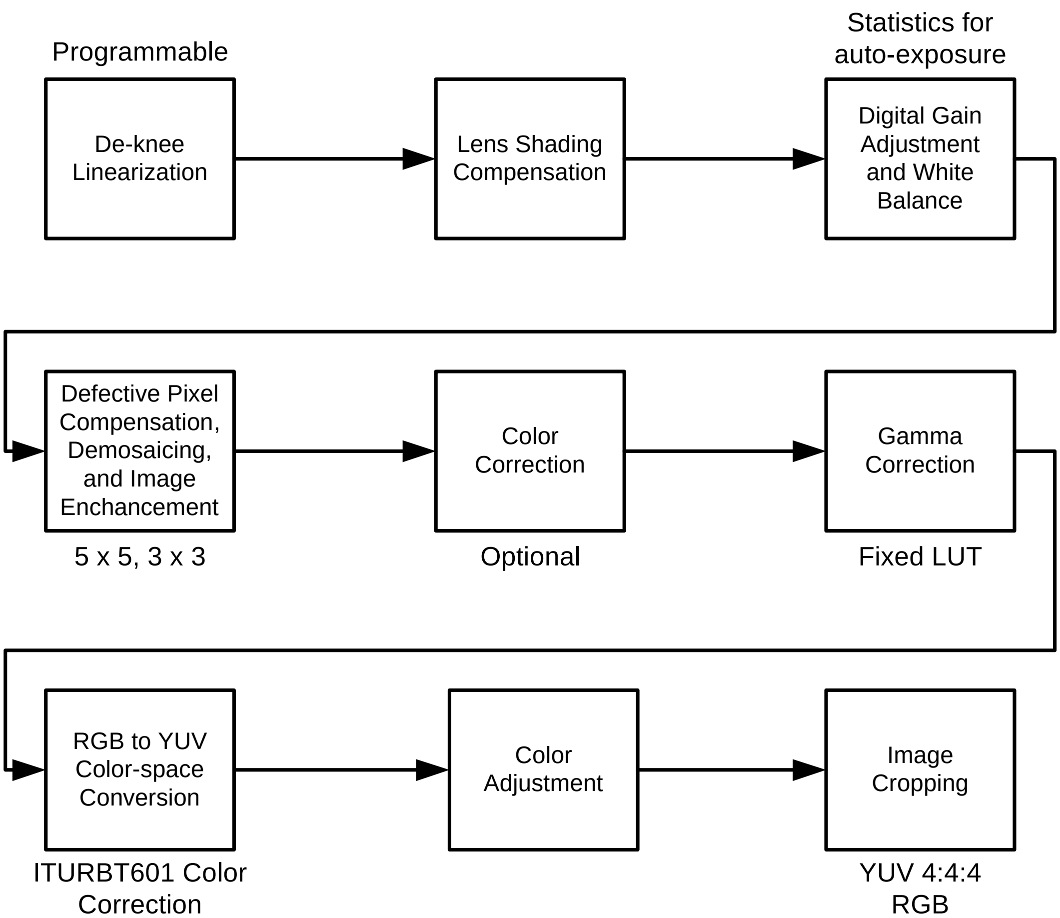
Functional blocks:

- 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.

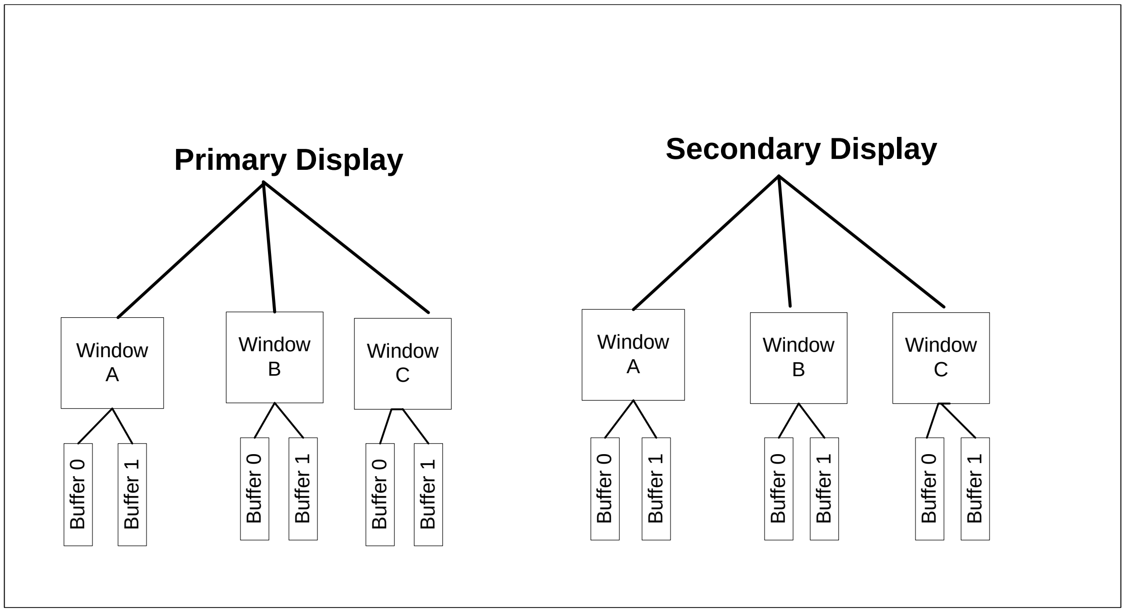
Scaling typically is required for video overlays. So both Window B and Window C support 6-tap, 16-phase horizontal filtering. Window B supports 2-tap 16-phase vertical filtering. Window B can produce better-quality scaling than Window C because of its vertical filter, utilizing twice the memory bandwidth 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.

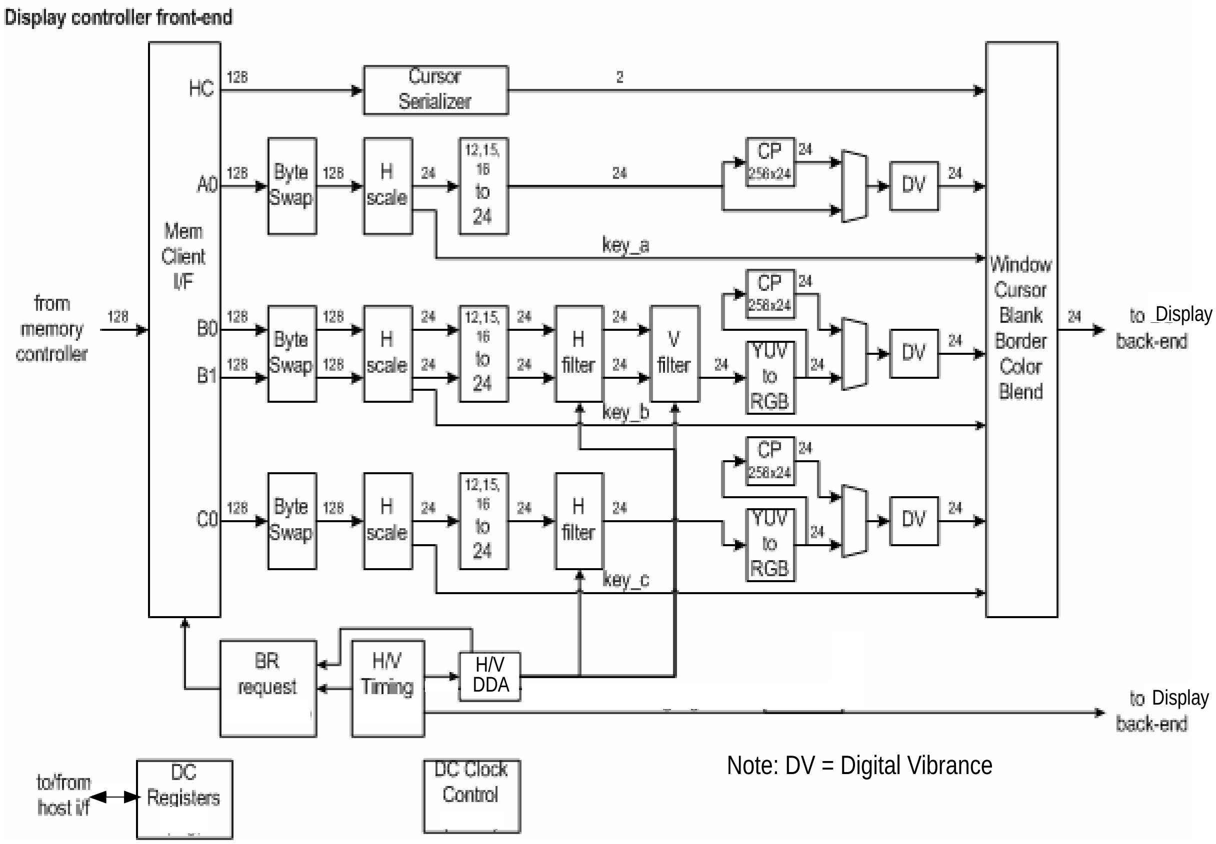
All graphics modes support odd-even byte swapping. Three 256 x 8bit color palettes are used for 1 bpp, 2 bpp, 4 bpp, and 8 bpp to convert them to 24 bpp. The color palette is shared for both graphics image windows in Primary and in Secondary display modes. Also, 4bit, 5bit, and 6bit-to-8bit adaptive 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.

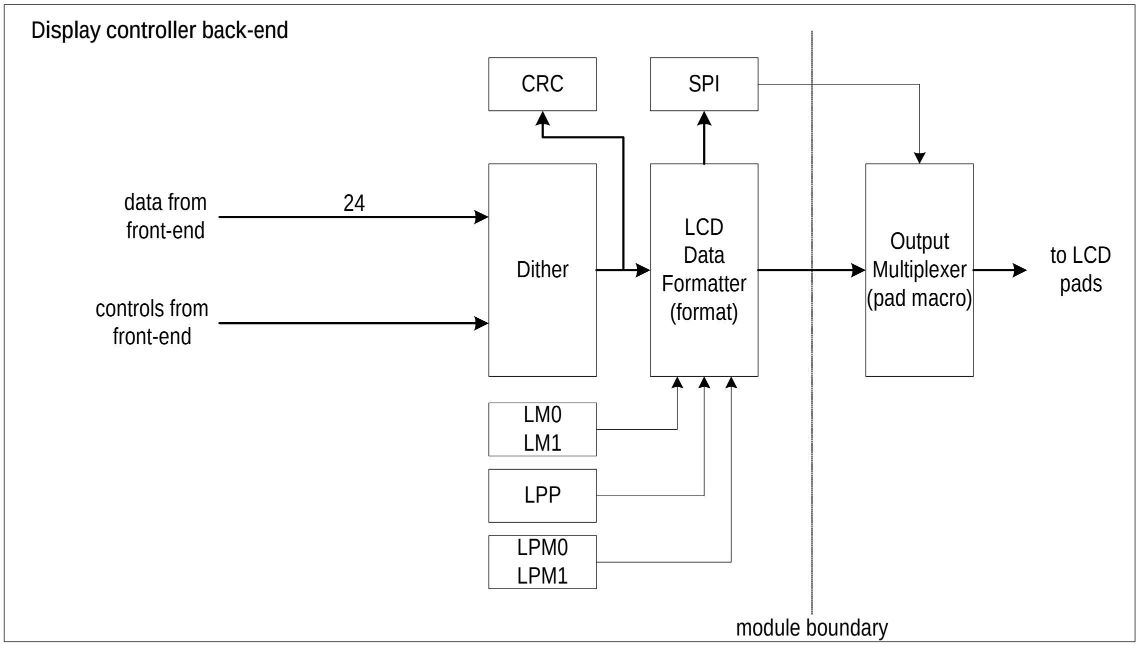
Blended data is the sum of the weighted active window data.

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

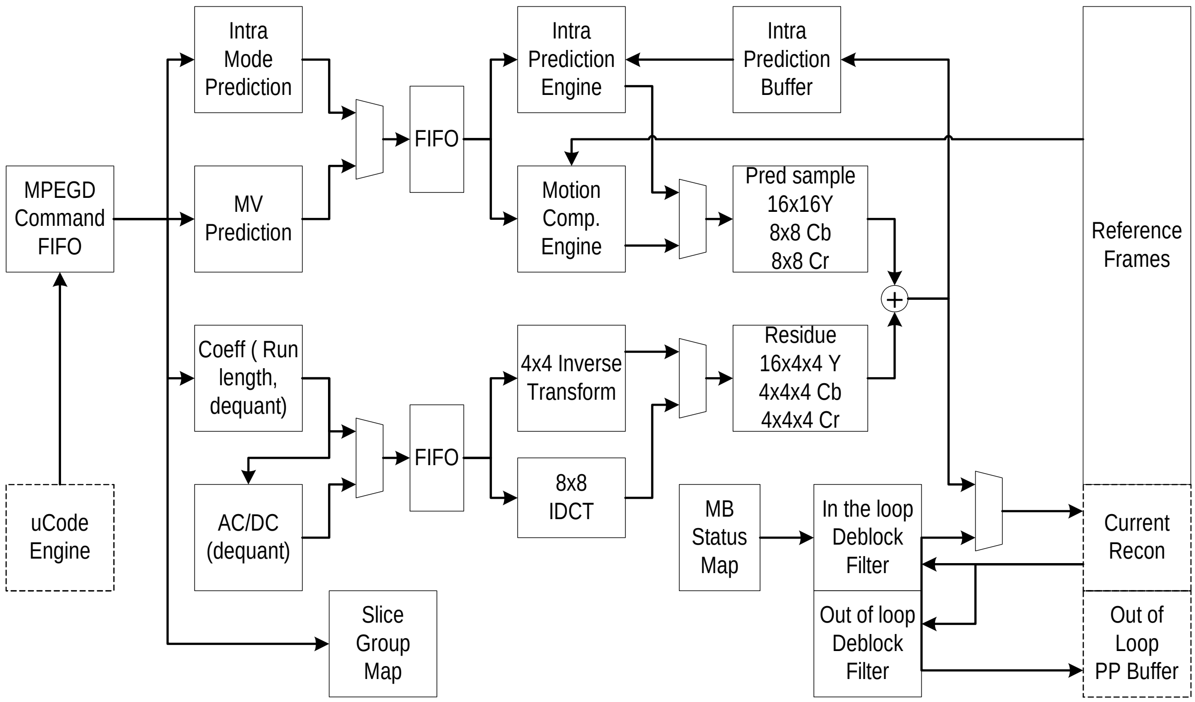
- WMV9 Decode
  - 320 x 240 (QVGA),25 fps, 384 kbps, comparable to simple profile medium level
  - QCIF, 15 fps, 96 kbps, full spec for simple profile, low level

- 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

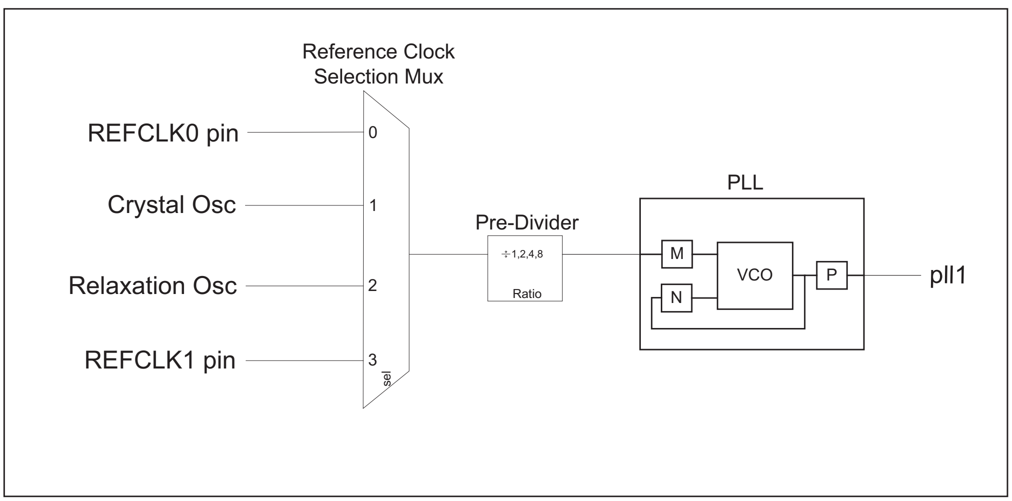
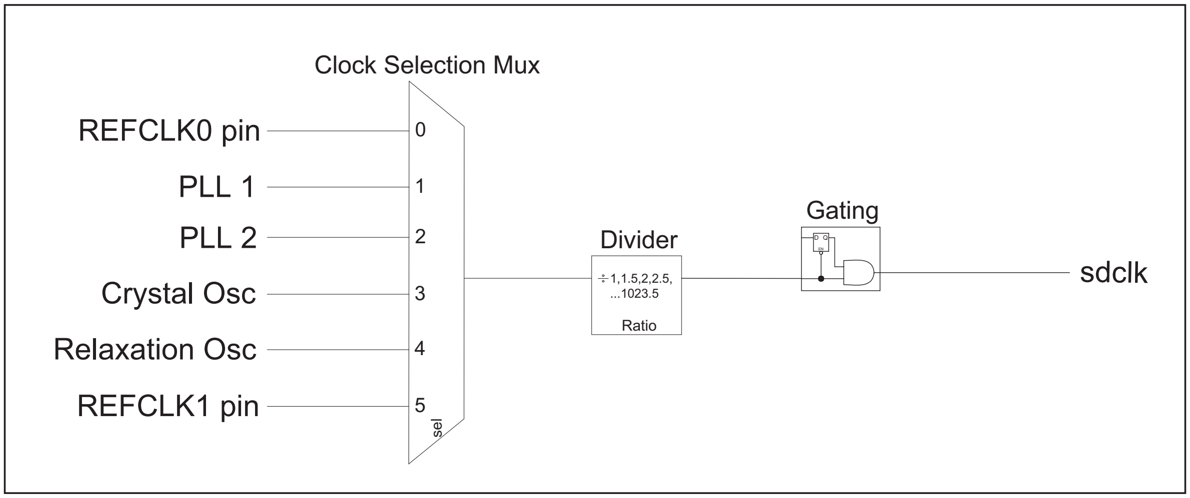


Figure 2.10 depicts a typical module's clock generation; in this case the SD Module. Each module's clock is derived similarly, and separately.

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.

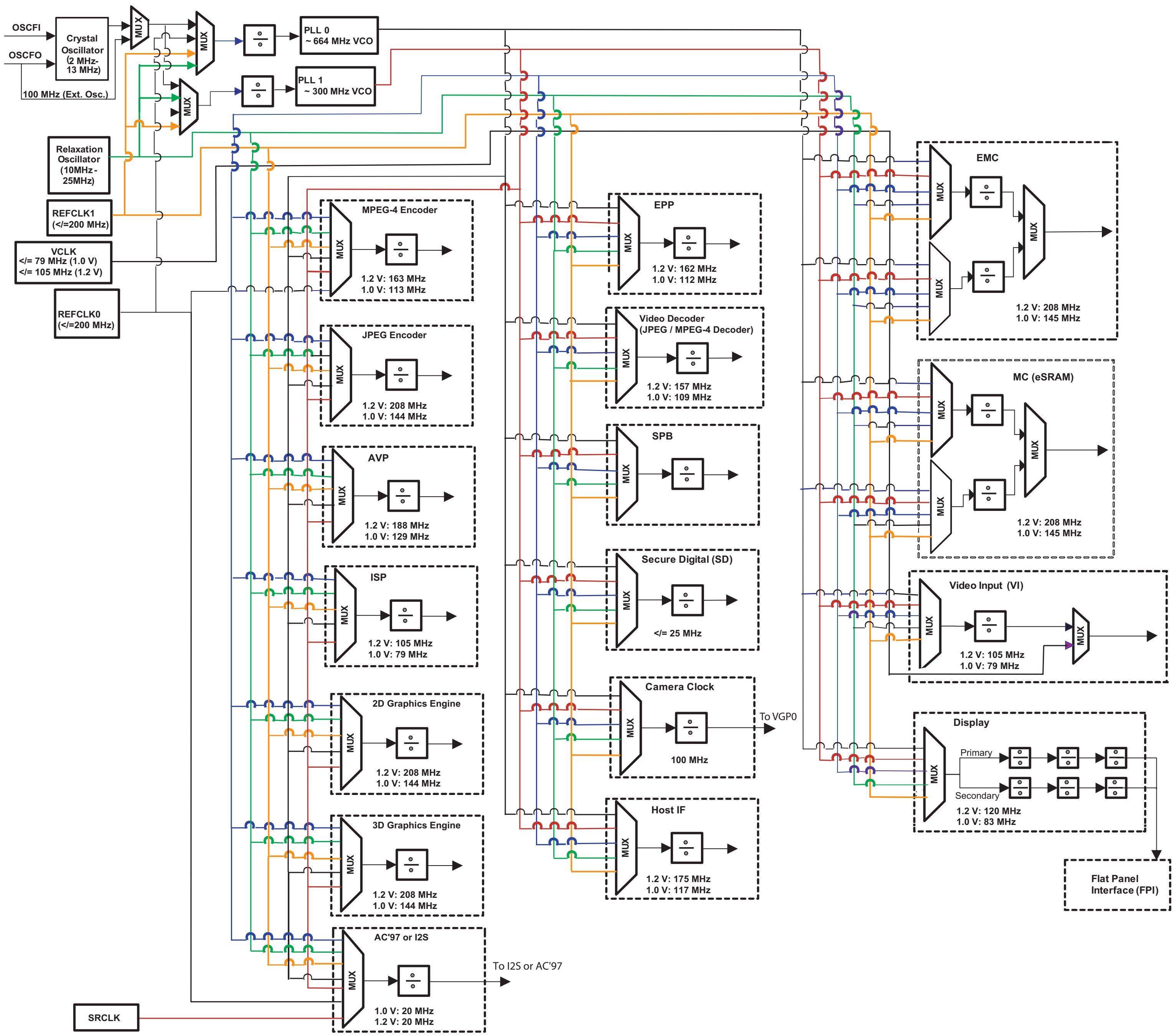
The Relaxation Oscillator can save anywhere from 100 to 400 μA, compared to using the Crystal Oscillator output as the primary clock source. The Relaxation Oscillator should not be used as the source for the camera.

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.

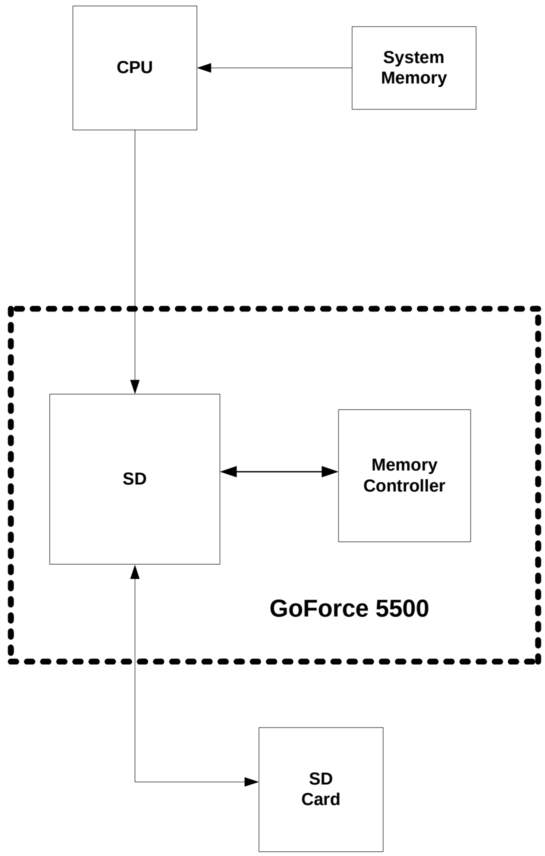
- DMA transfer to/from memory

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 Micro-classes” 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:

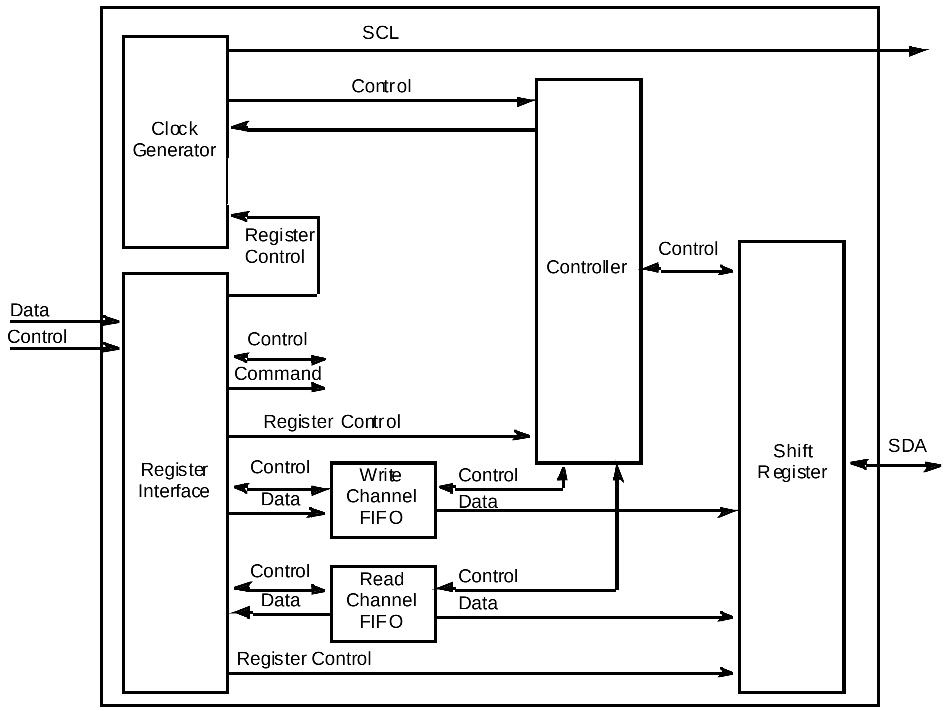
- 7-bit addressing.
- 10-bit addressing.
- Combined 7bit and 10bit addressing.
- Programmable transfer count for transmit and receive.
- Clock synchronization (for multi-master environments).
- Arbitration (for multi-master environments).
- Operation in Standard Mode (100 Kb/second).
- Operation in Fast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

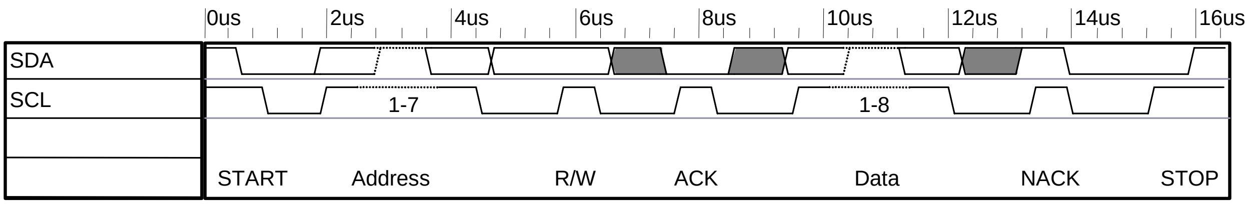


Figure 2.14 shows a typical data transaction. Transfer begins with the transmission of a start condition. Then a seven-bit address is presented, followed by a read/write bit, and the target device acknowledge (ACK). If the master sees a valid acknowledge, data is transferred. A not-acknowledge (NACK) 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 I^{2}S and AC'97 Codec Interface

2.19.1 Introduction

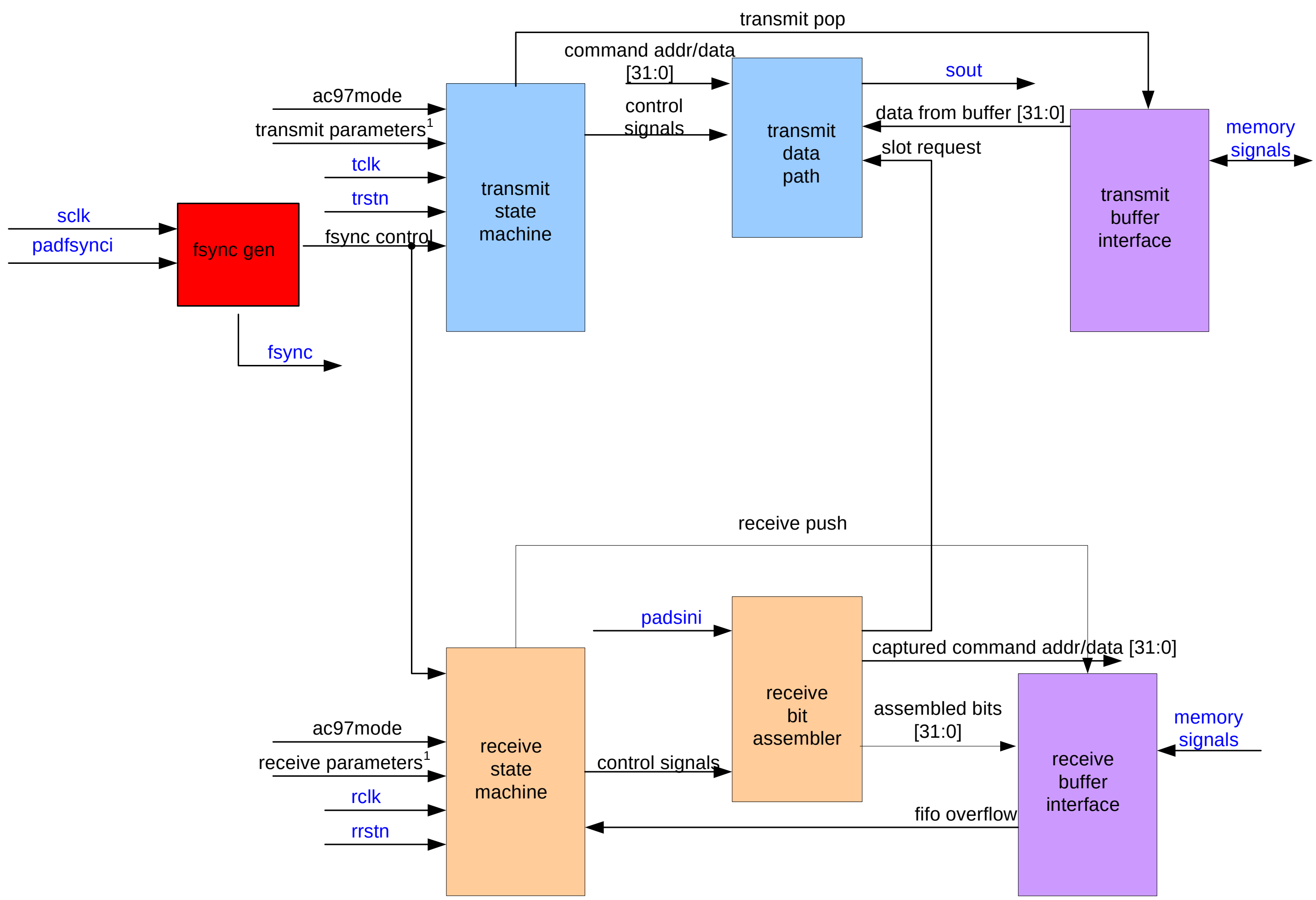
The I^{2}S and AC'97 Codec Interface features the following:

- 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
- I^{2}S 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-I^{2}S 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

<table>
<tr><th>Name</th><th>Modules Powered</th><th>Notes: Requirements</th></tr>
<tr><td rowspan="6">AOCVDD</td><td>All clock cores</td><td rowspan="6">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.</td></tr>
<tr><td>Host Interface</td></tr>
<tr><td>Graphics and Display controllers</td></tr>
<tr><td>Memory controller
(Internal and External)</td></tr>
<tr><td>IO Controls</td></tr>
<tr><td>SD</td></tr>
<tr><td rowspan="5">VECVDD</td><td>Video Codec (DSP/AVP)</td><td rowspan="5">VECVDD must be powered on when any of the modules receiving power from it are in use.</td></tr>
<tr><td>Camera interface</td></tr>
<tr><td>ISP</td></tr>
<tr><td>2D Graphics Engine</td></tr>
<tr><td>Audio</td></tr>
<tr><td>MMCVDD</td><td>SRAMS</td><td>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</td></tr>
<tr><td>TDCVDD</td><td>3D</td><td>TDCVDD must be on when 3D acceleration is needed.</td></tr>
</table>

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
AVDDOSC	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.

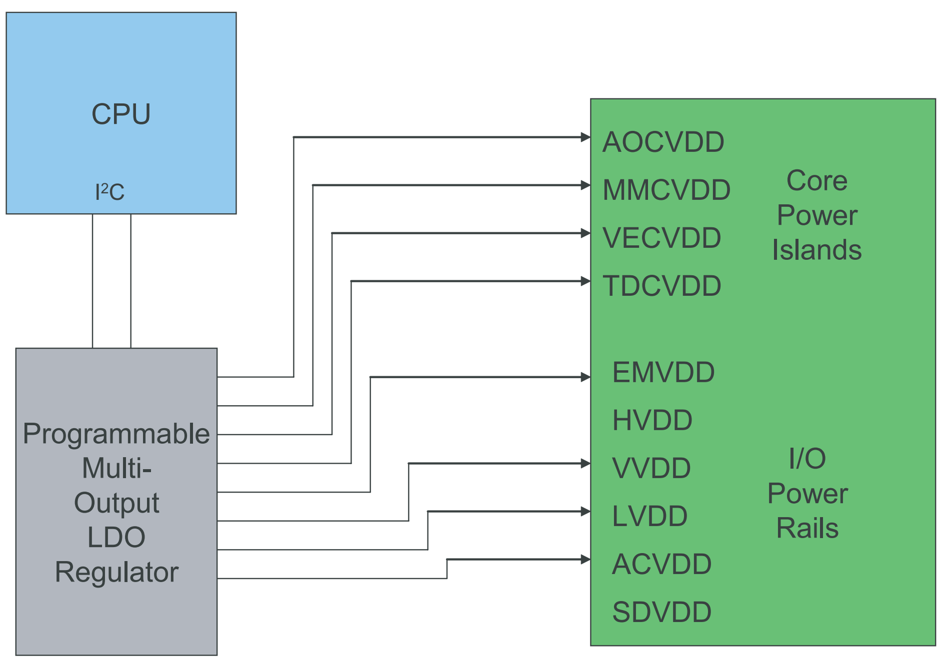
Table 3.6: GoForce 5500 Power Rails

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 I^2S/AC'97 CODEC Interface

The I^2S/AC'97 Codec Interface pins are powered by ACVDD, the I2S/AC'97 core is referenced to VECVDD.

Table 3.17: Description of the I^2S/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

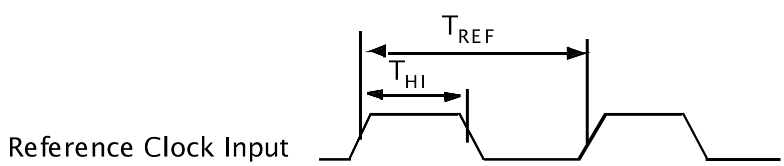


Table 3.6 and Table 3.7 provide the AC timing characteristic values of the input crystal oscillator (OSCFI, used with the internal oscillator clock source) and the external input clock (OSCFO, used when bypassing the internal crystal oscillator.) The values are created by changing the crystal's oscillation 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 x C_{load}Crystal)

Figure 4.2: Reference Clock Timing Test Circuit

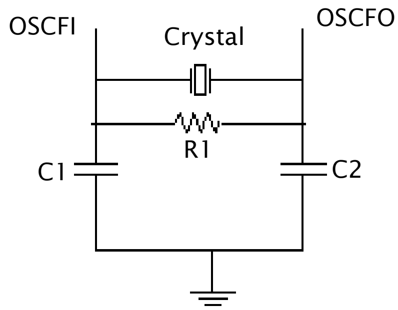


Table 4.8: AC Timing Characteristics - Crystal Input Clock (OSCFI)

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

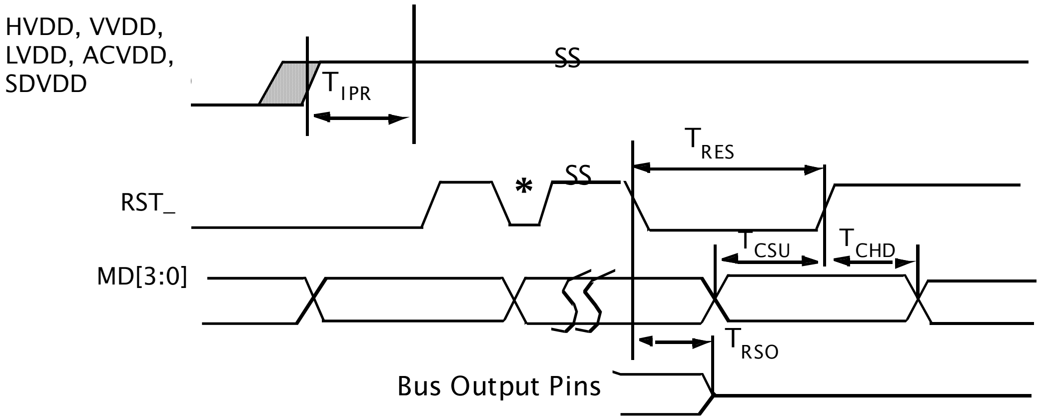


Table 4.10: AC Timing Characteristics - Reset

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.)

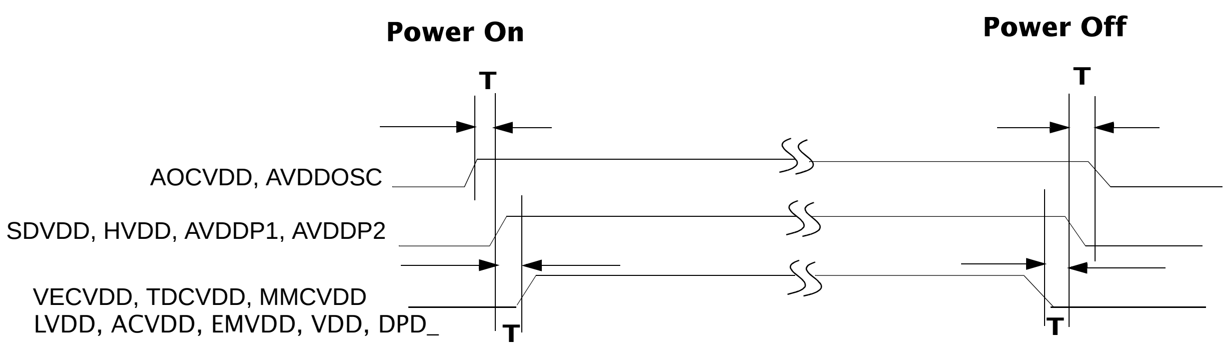
4.4.3.2 Power Down

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

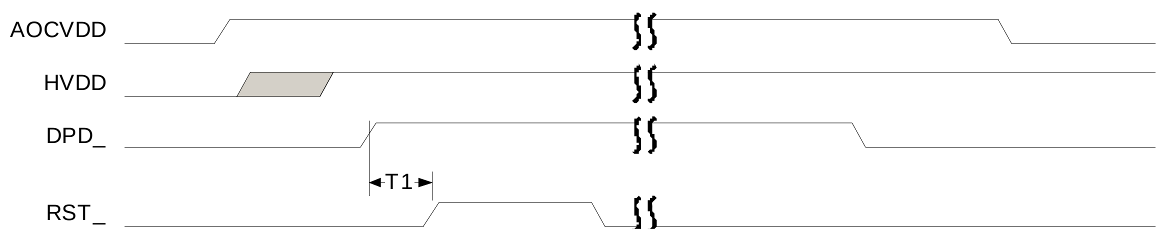
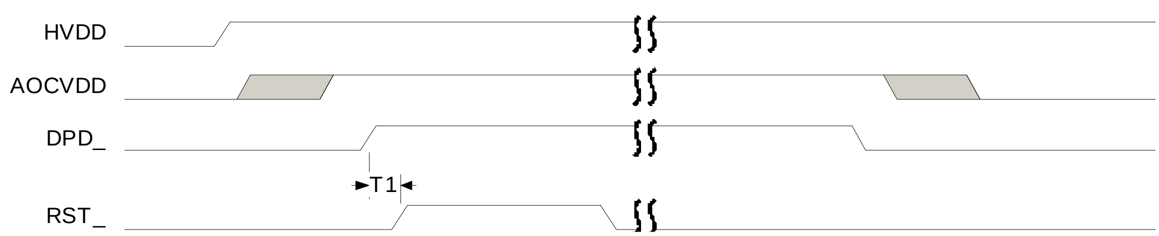


Figure 4.6: Power Sequence with DPD_ Case 2



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

Table 4.12: Type C Host Interface Timing Diagrams List

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

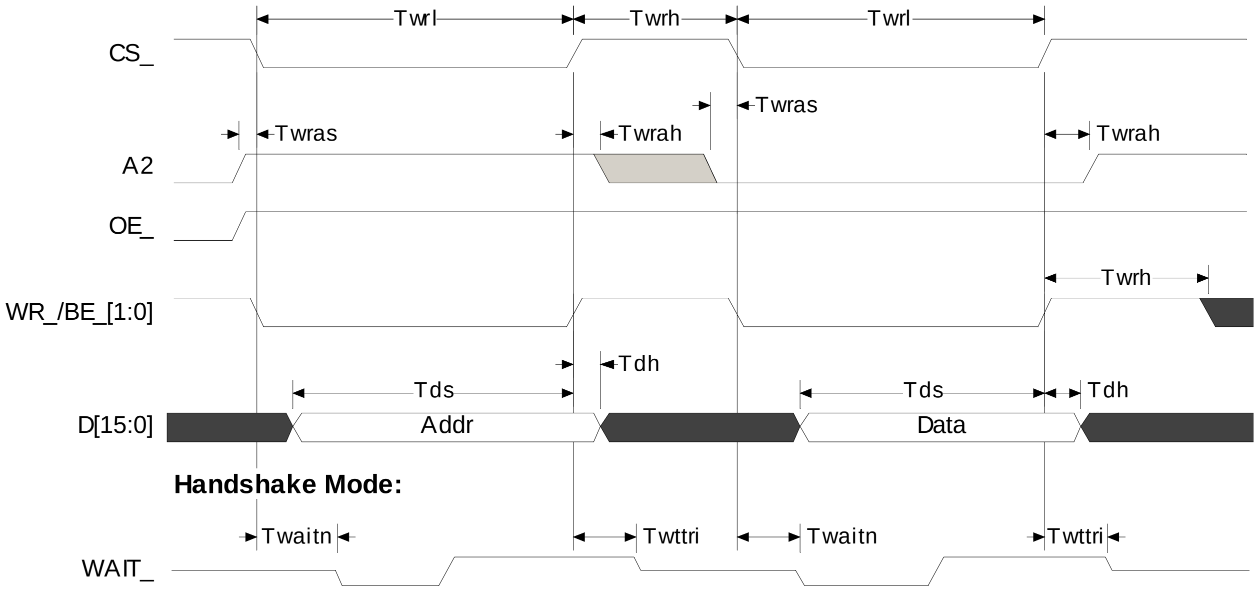


Table 4.14: Register Write D[15:0] Bit Mapping for Addr

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

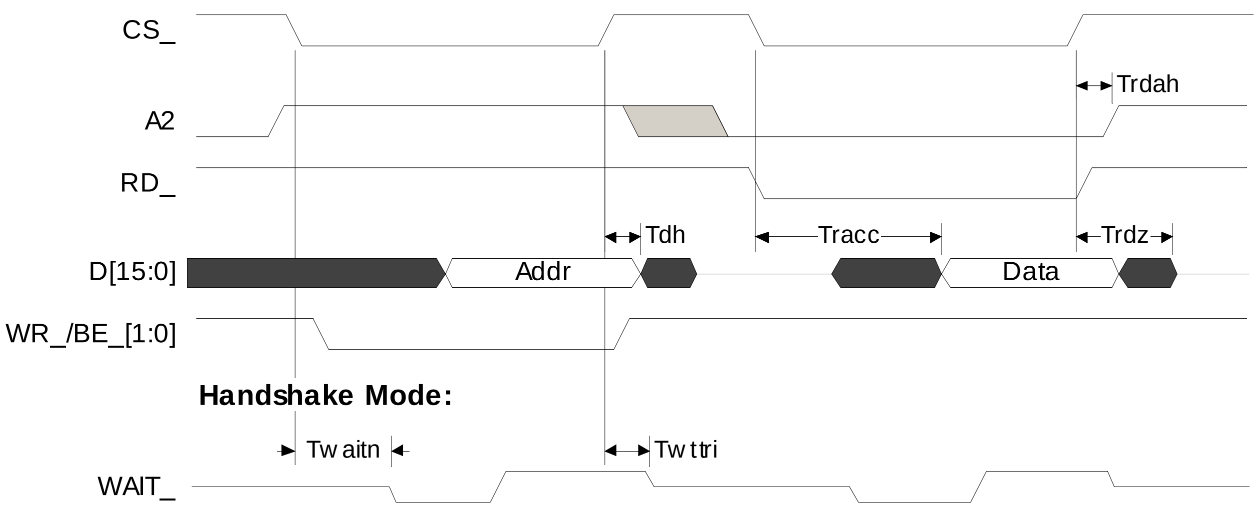


Table 4.15: Register Read D[15:0] Bit Mapping for Addr

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

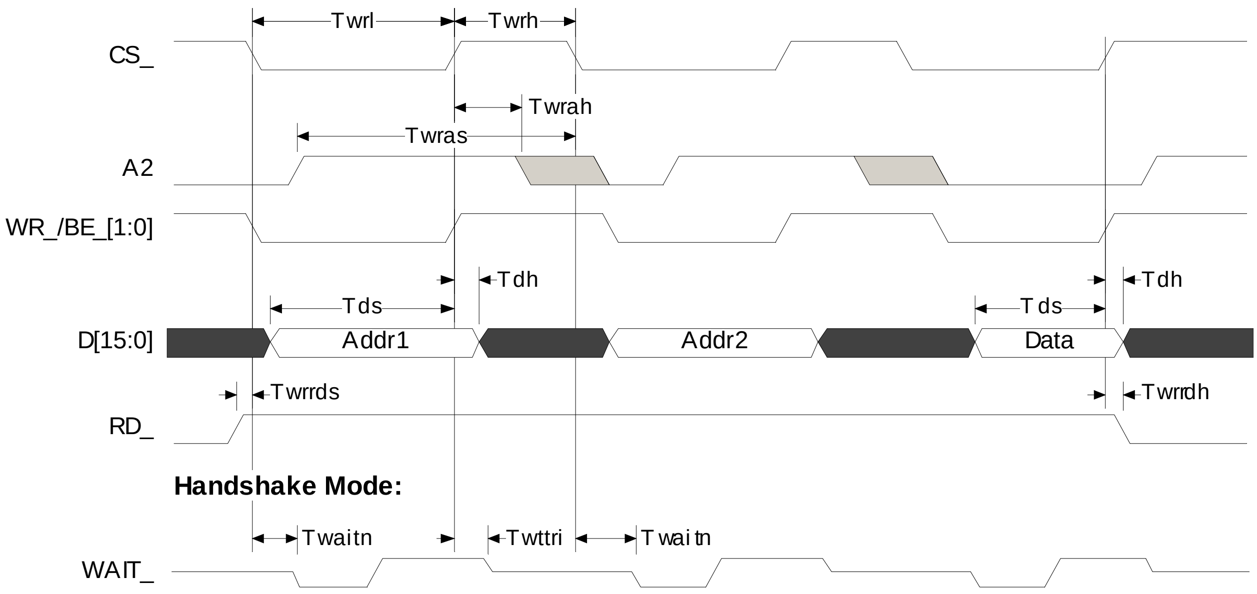


Table 4.16: Memory Write D[15:0] Bit Mapping for Addr1 and Addr2

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

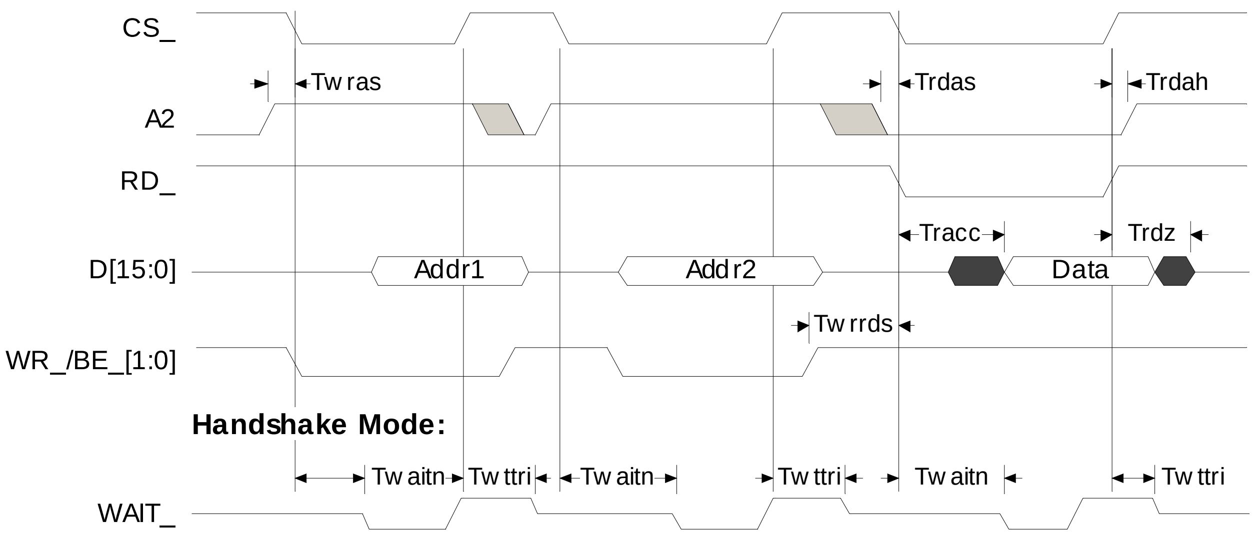


Table 4.17: Memory Read D[15:0] Bit Mapping for Addr1 and Addr2

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

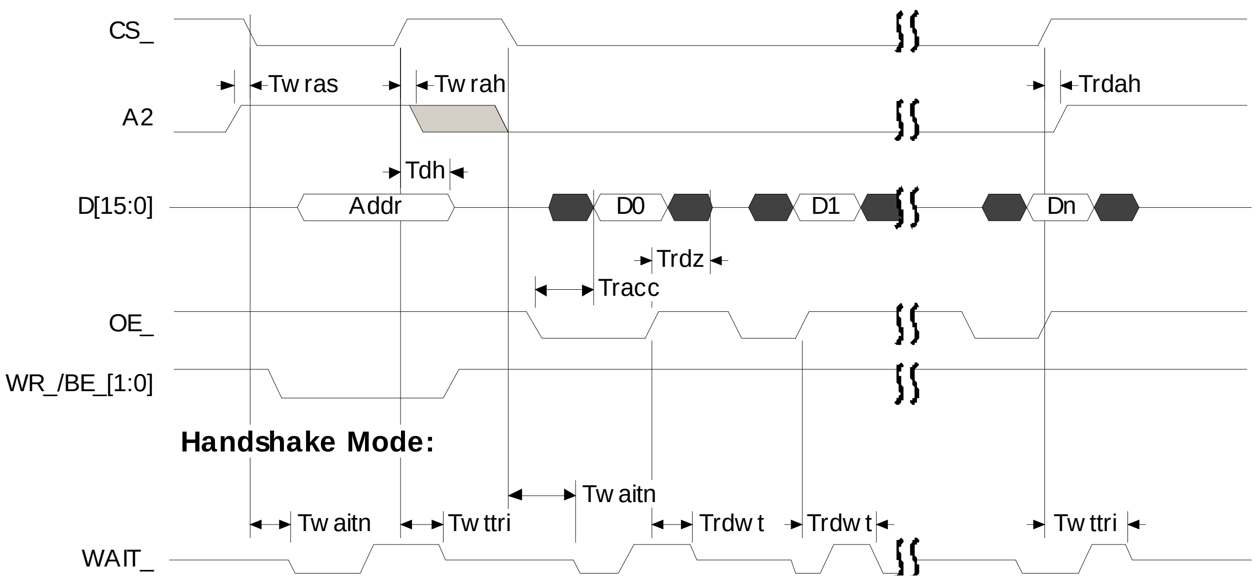


Table 4.18: Register Read D[15:0] Bit Mapping for Addr

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

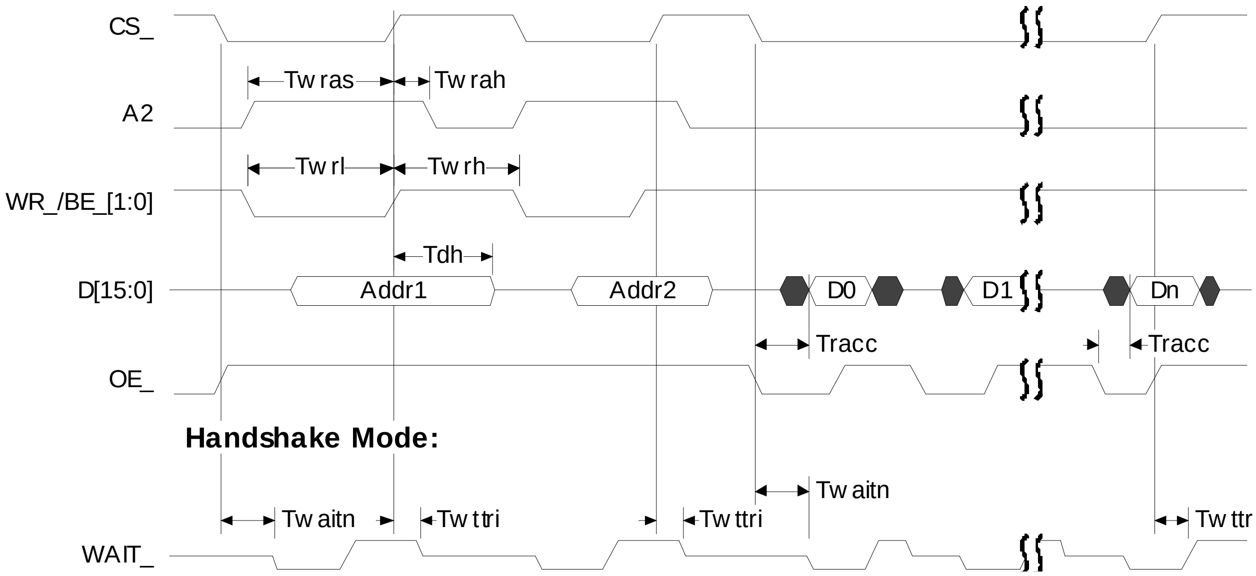


Table 4.19: Memory Read D[15:0] Bit Mapping for Addr1 and Addr2

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

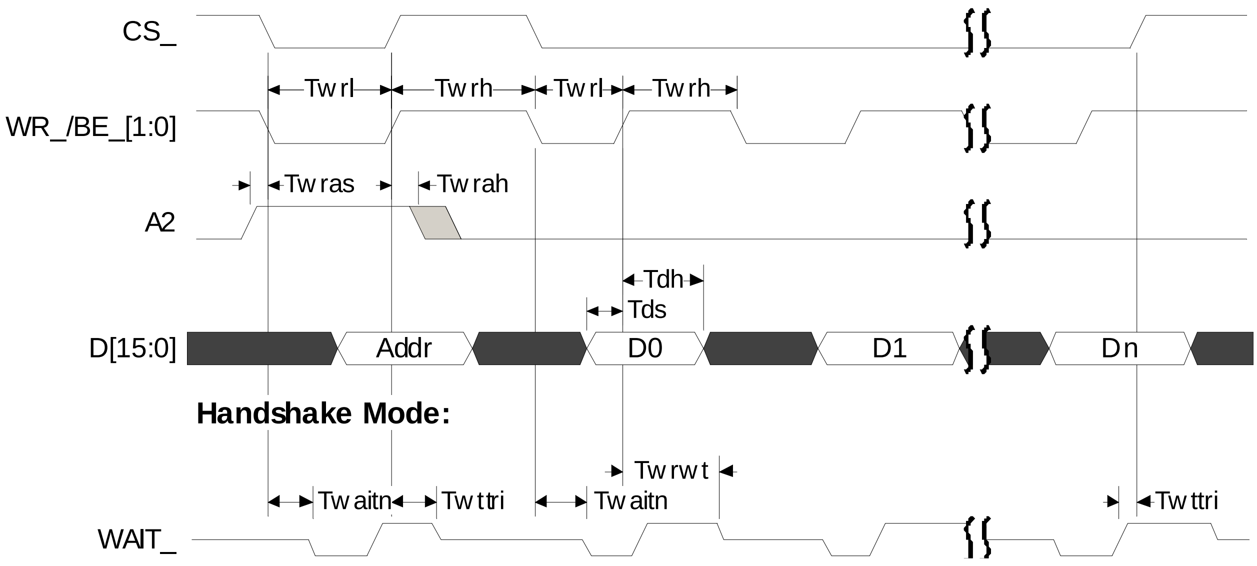


Table 4.20: Register Write D[15:0] Bit Mapping for Addr

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

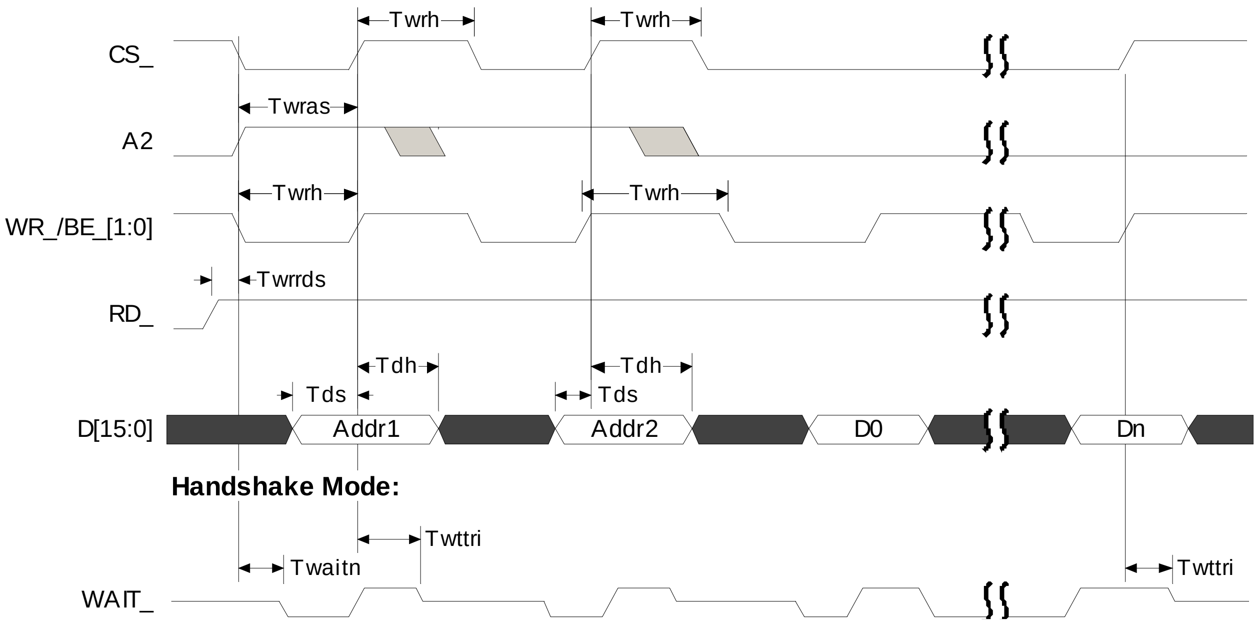


Table 4.21: Memory Write D[15:0] Bit Mapping for Addr1 and Addr2

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

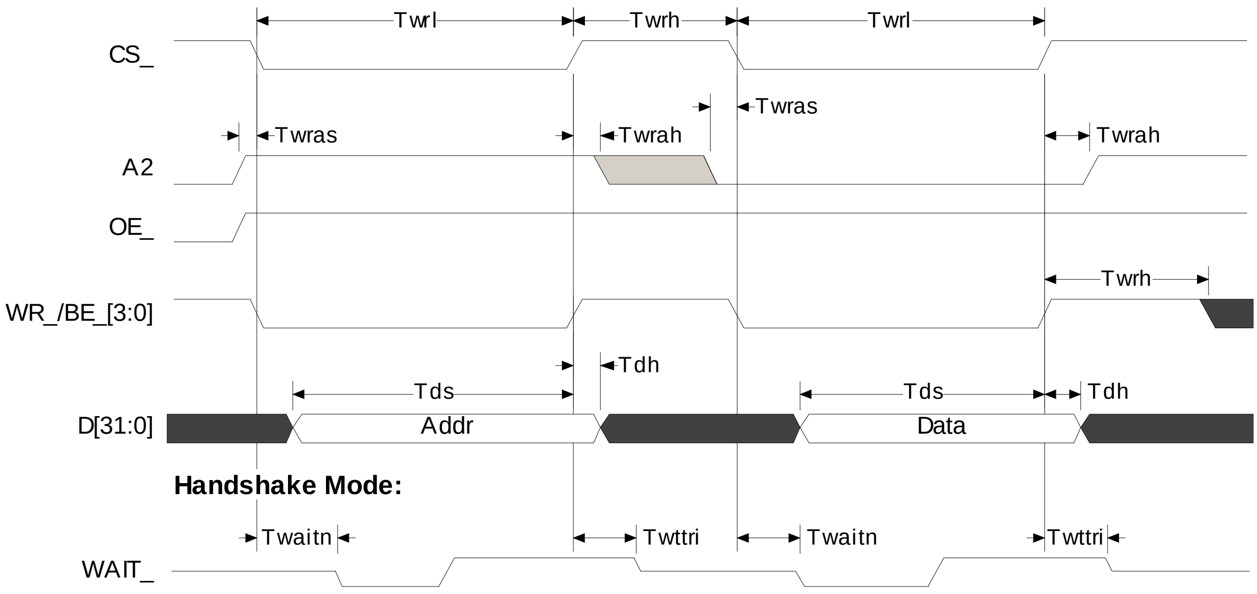


Table 4.22: Register Write D[31:0] Bit Mapping for Addr

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

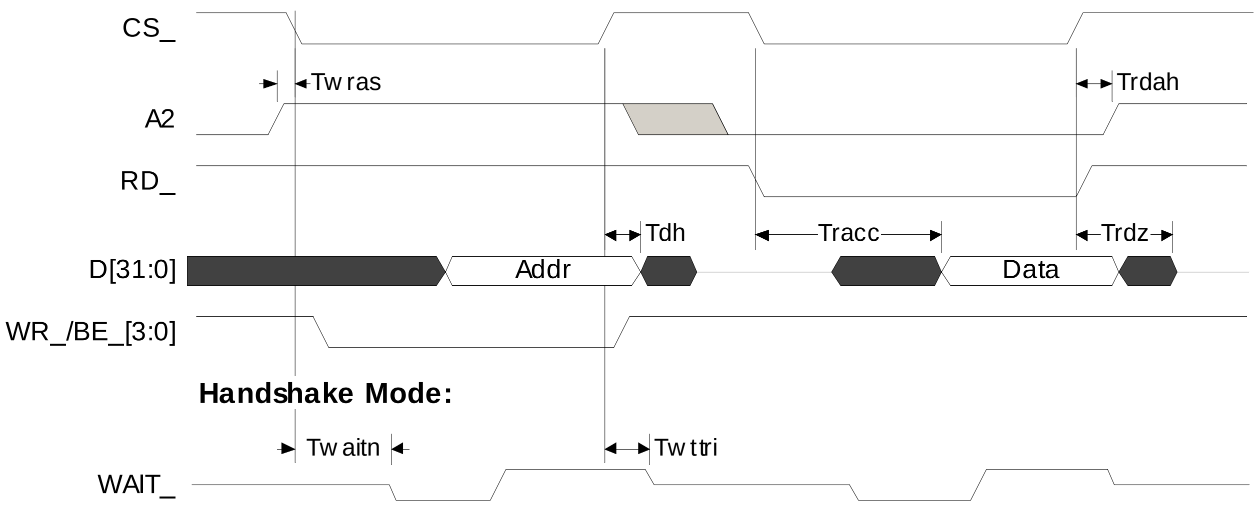


Table 4.23: Register Read D[31:0] Bit Mapping for Addr

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

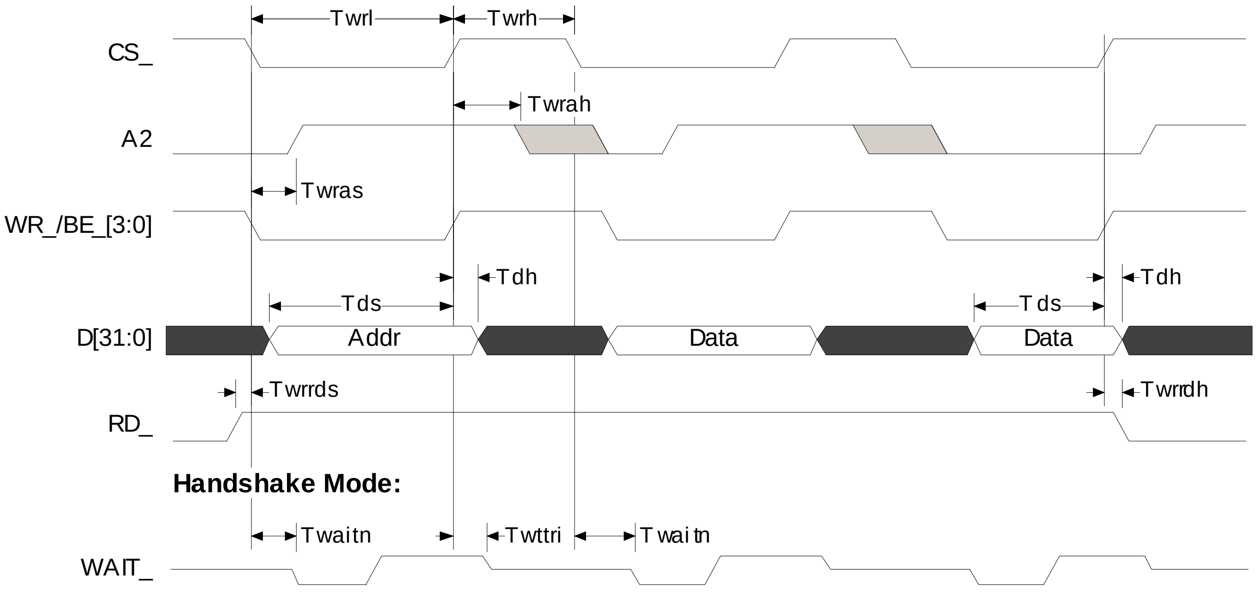


Table 4.24: Memory Write D[31:0] Bit Mapping for Addr

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

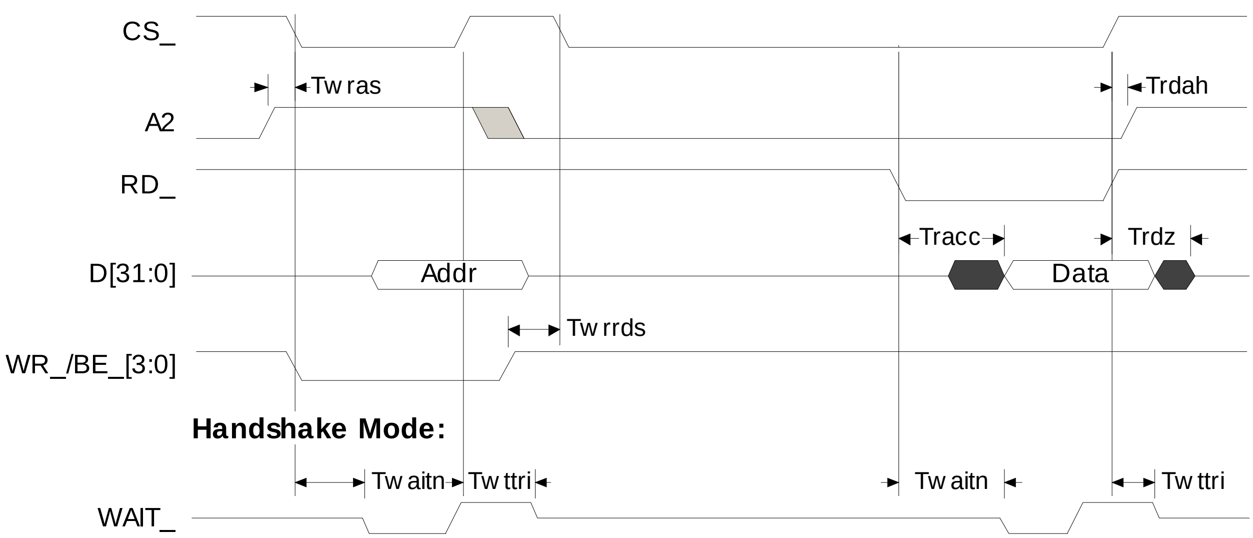


Table 4.25: Memory Read D[31:0] Bit Mapping for Addr

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

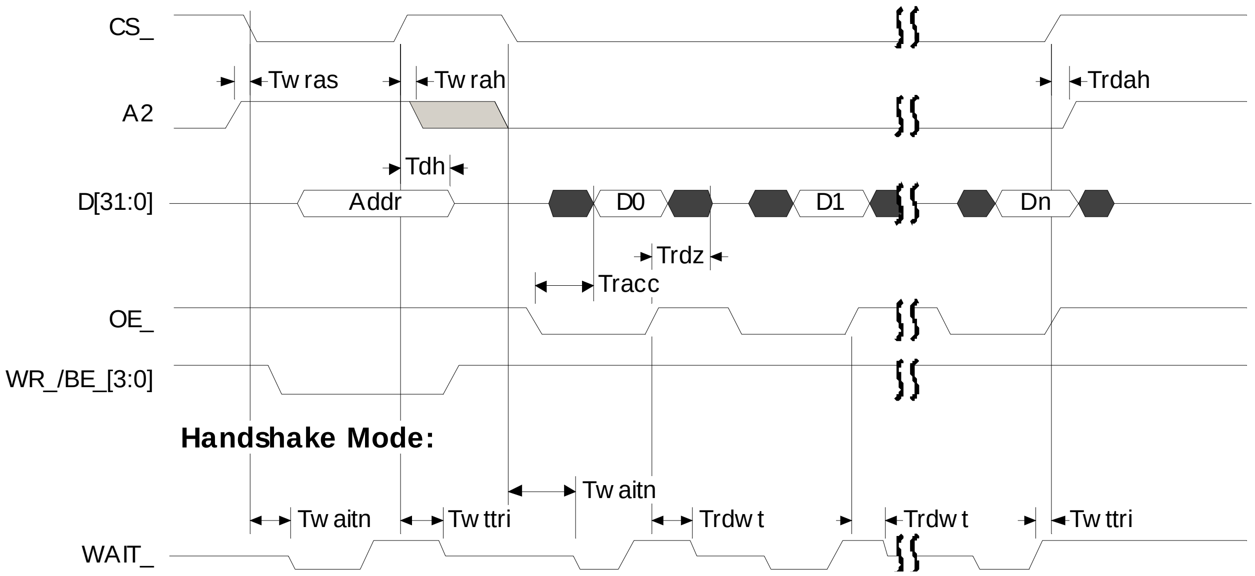


Table 4.26: Register Read D[31:0] Bit Mapping for Addr

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

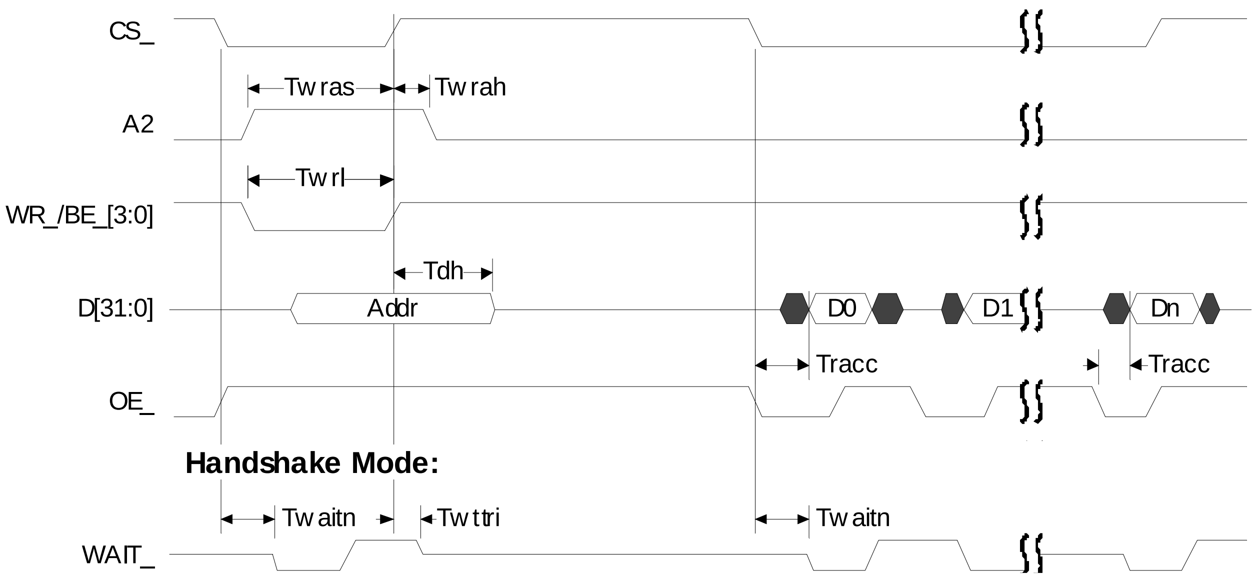


Table 4.27: Memory Read D[31:0] Bit Mapping for Addr

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

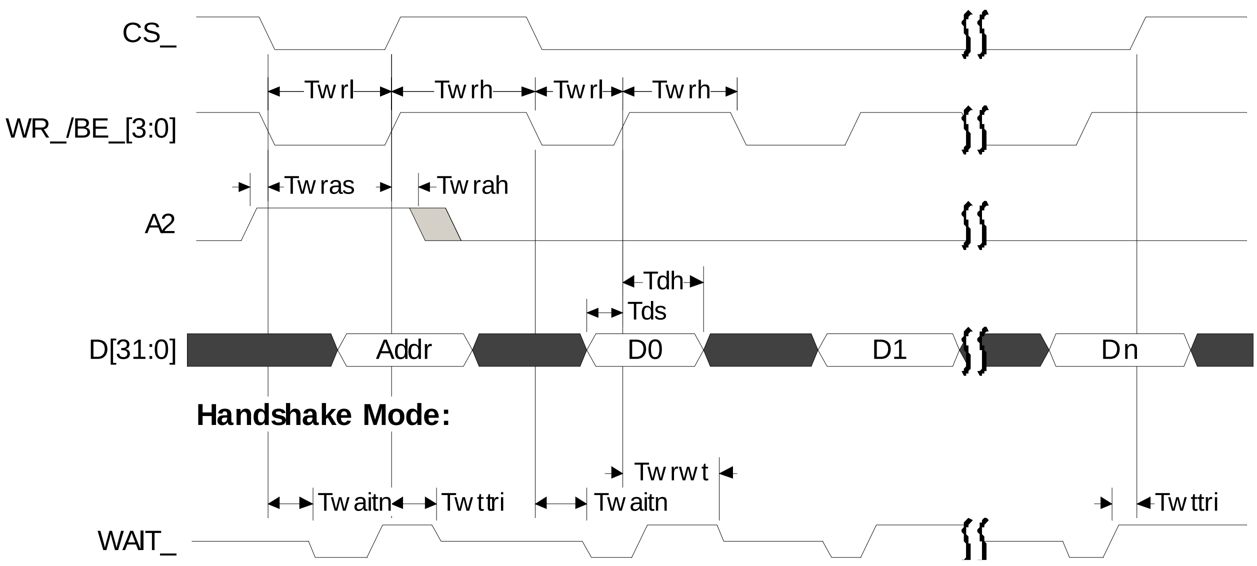


Table 4.28: Register Write D[31:0] Bit Mapping for Addr

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

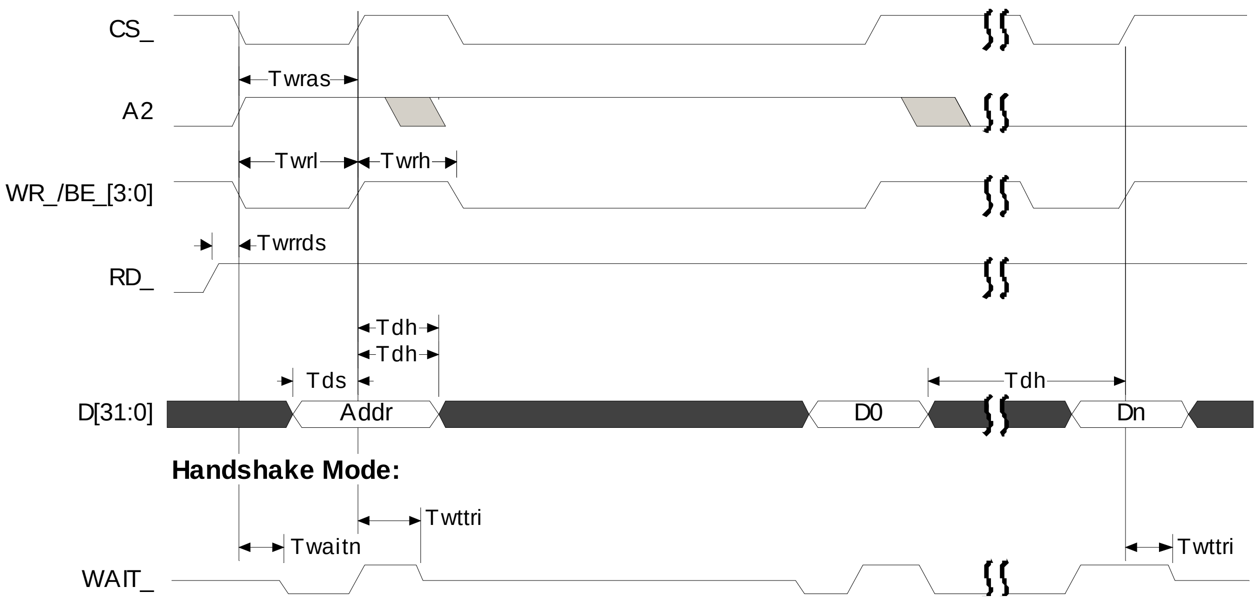
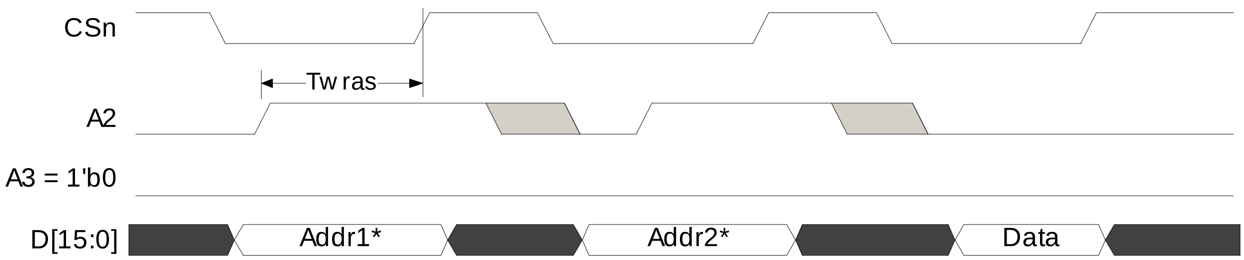


Table 4.29: Memory Write D[31:0] Bit Mapping for Addr

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

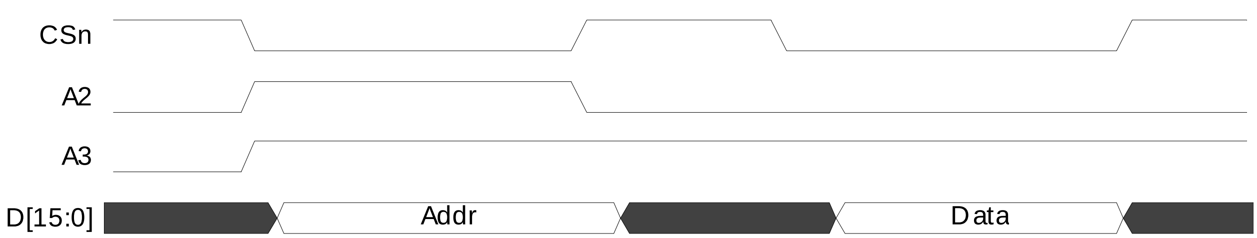
Figure 4.23: One-channel Access, Indirect Addressing



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

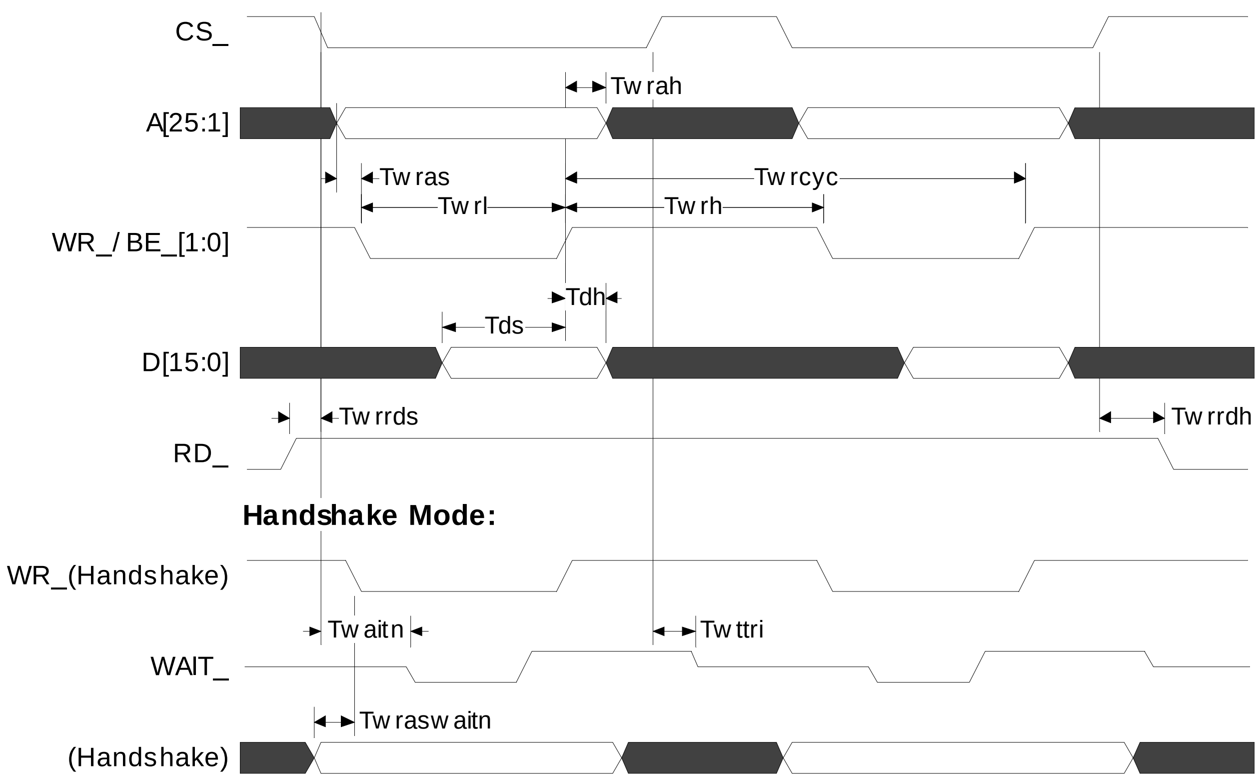


Figure 4.27: CS_-controlled Write: 16Bit Direct Type A

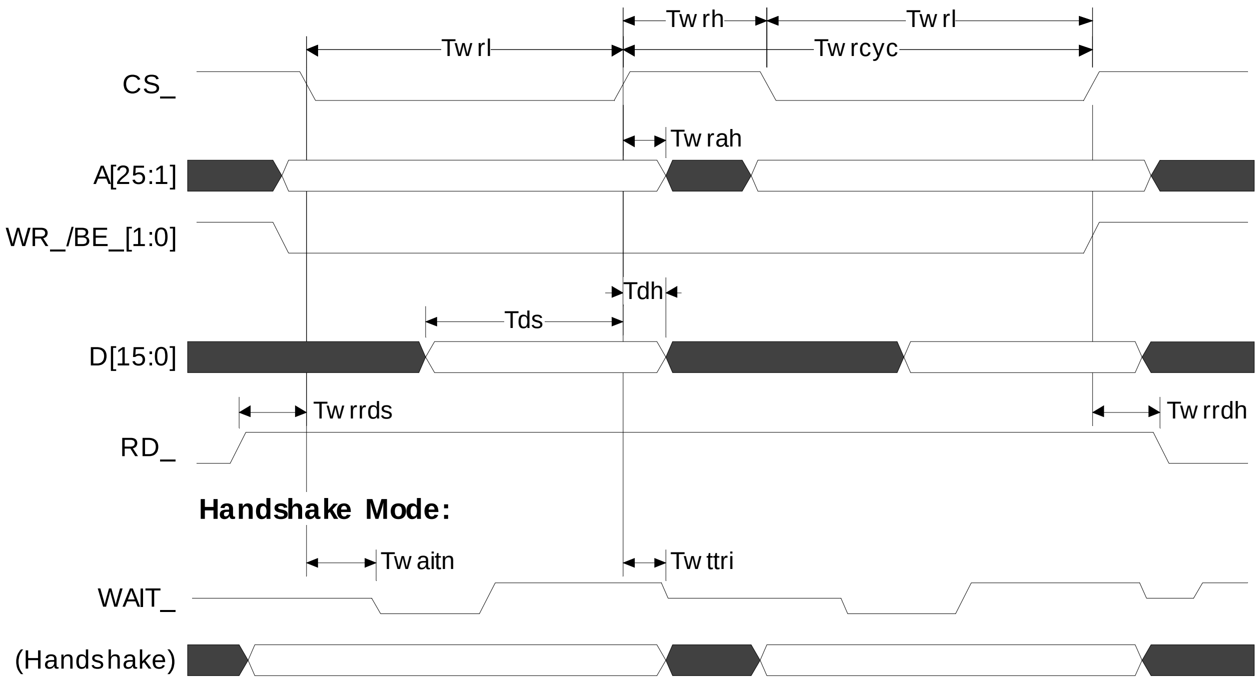


Figure 4.28: RD_-controlled Read: 16Bit Direct Type A

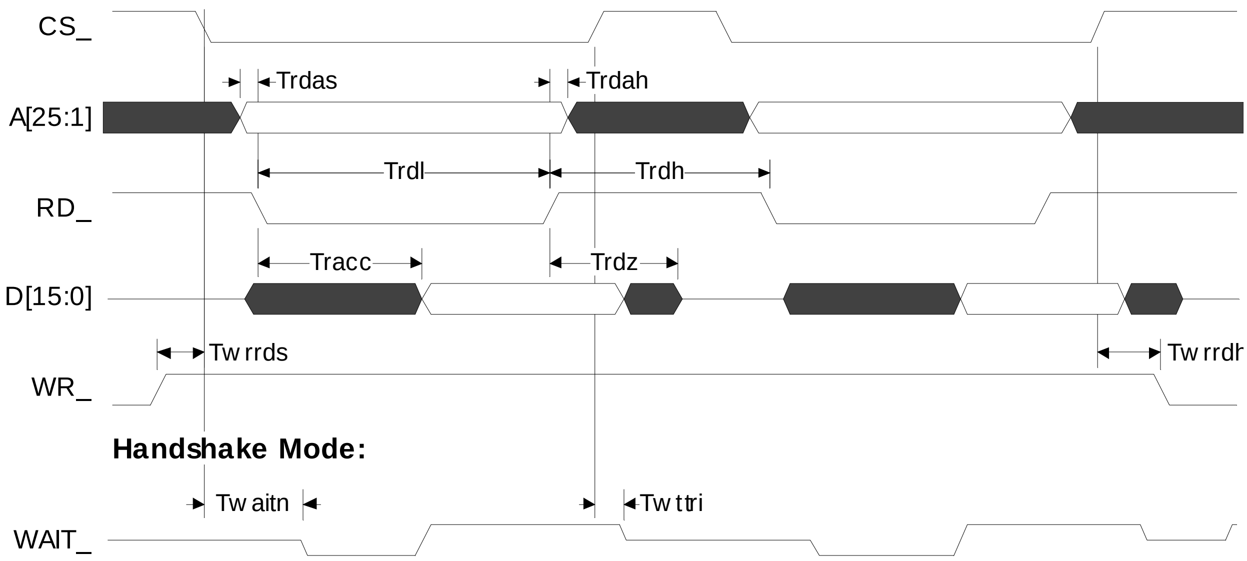


Figure 4.29: CS_-controlled Read: 16Bit Direct Type A

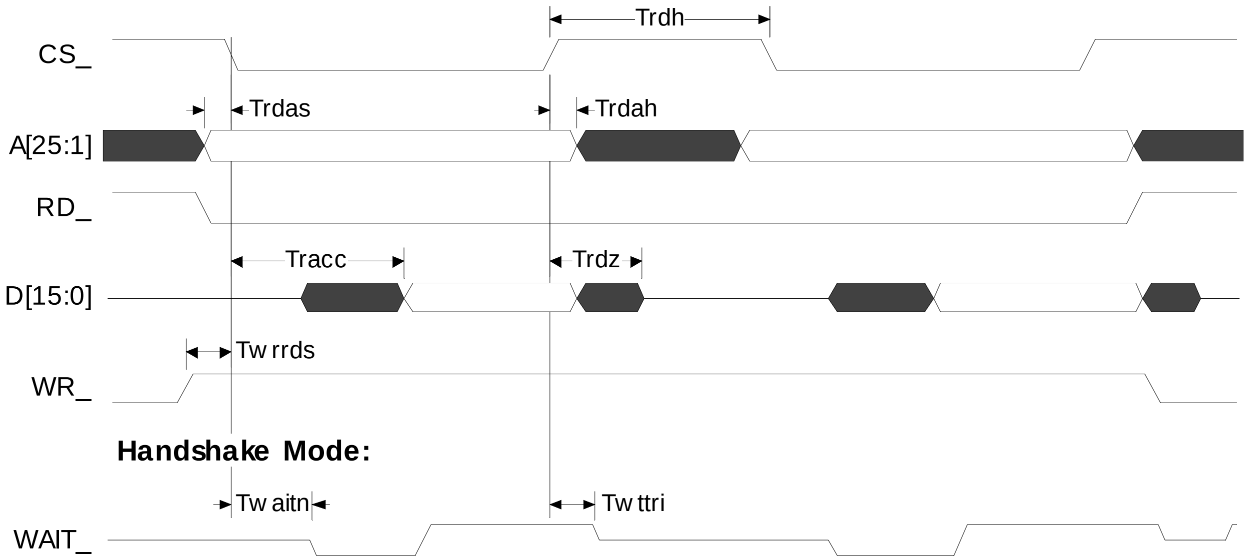


Figure 4.30: Register or Memory Burst Write: 16Bit Direct Type A

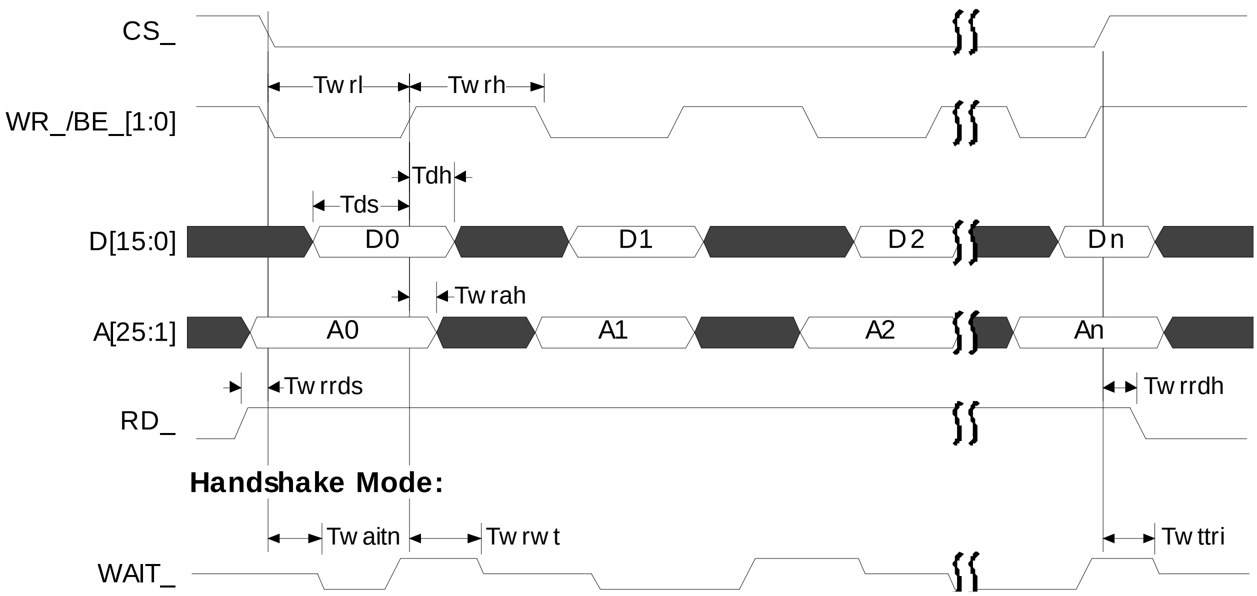
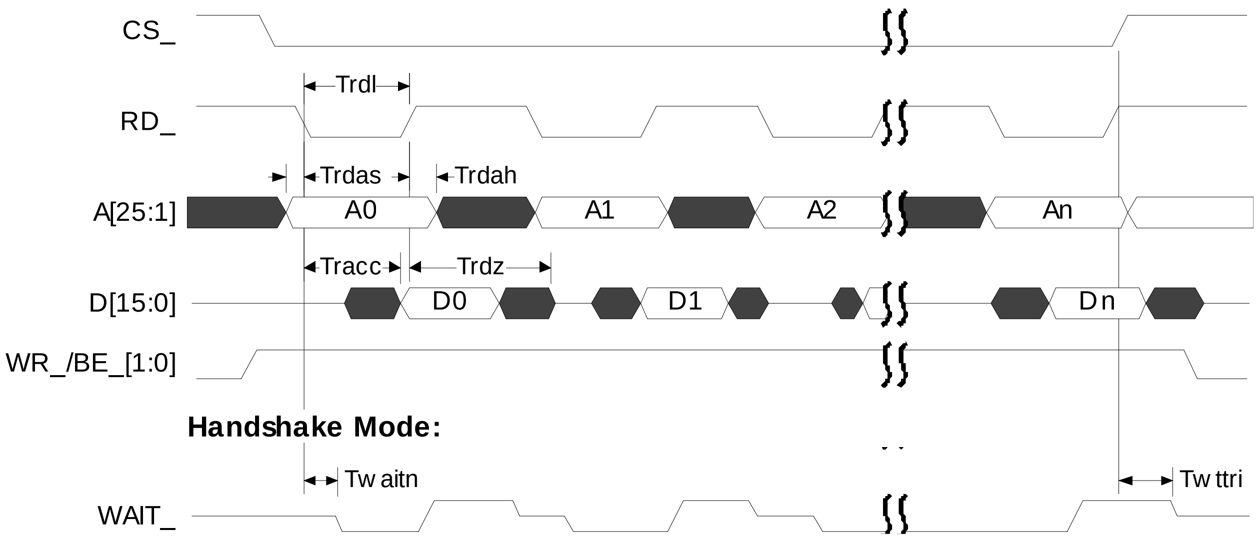


Figure 4.31: Register or Memory Burst Read: 16Bit Direct Type A



4.4.4.1.5 Type A Direct Timing Diagrams: 32Bit Interface

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

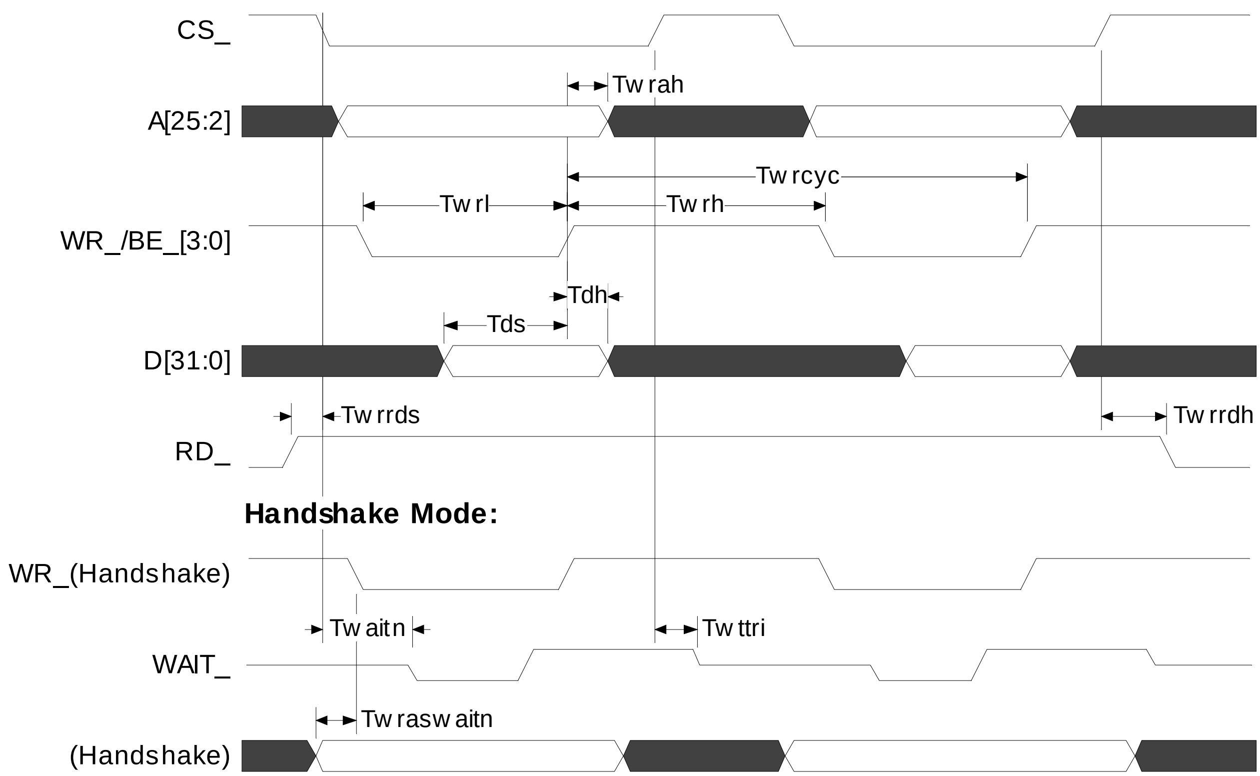


Figure 4.33: CS_-controlled Write: 32Bit Direct Type A

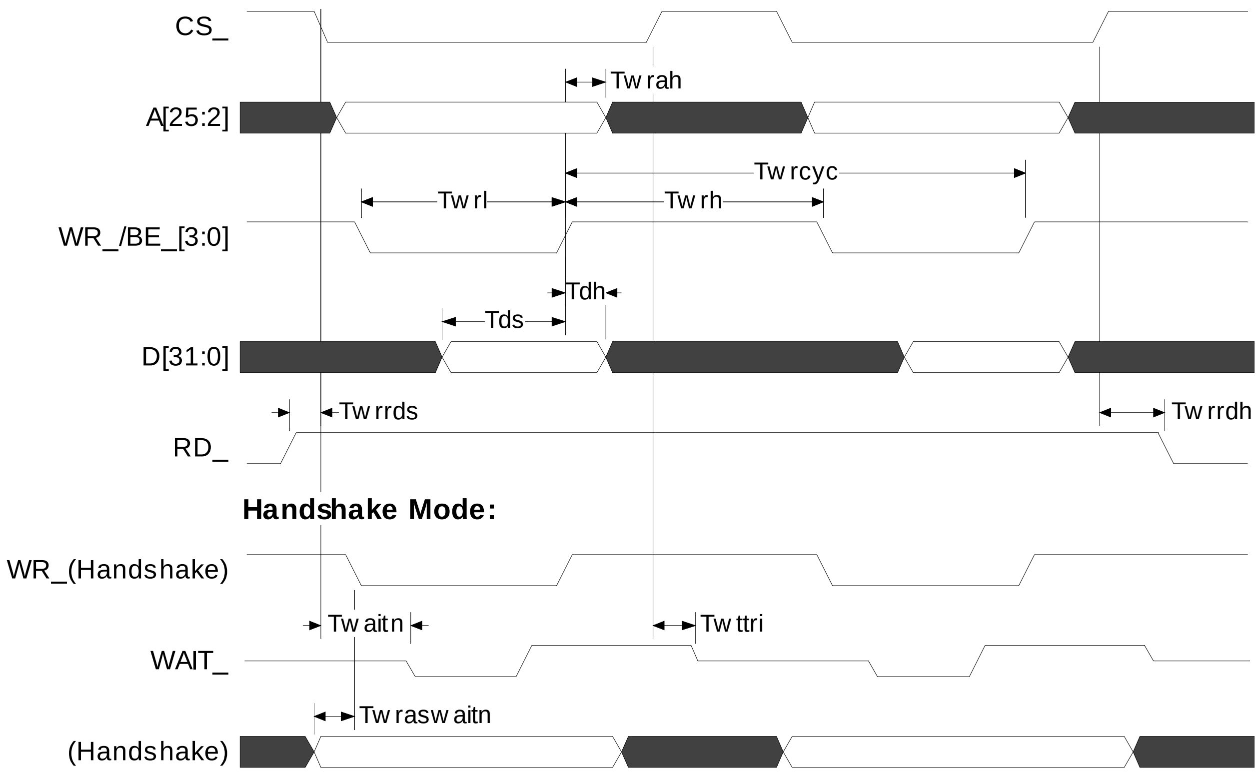


Figure 4.34: RD_-controlled Read: 32Bit Direct Type A

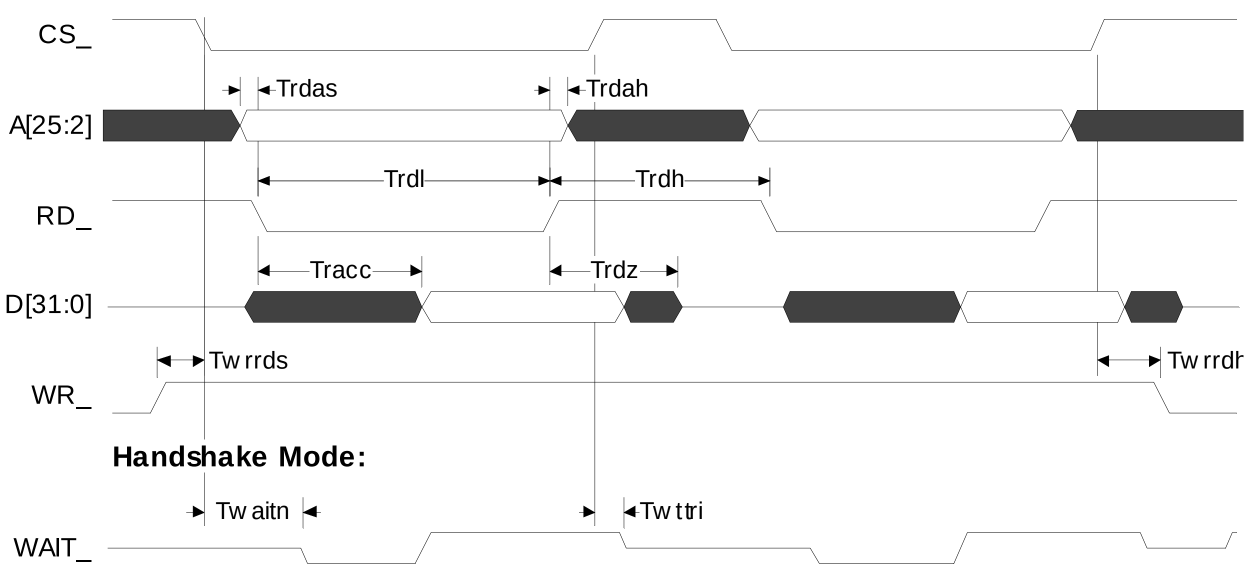


Figure 4.35: CS_-controlled Read: 32Bit Direct Type A

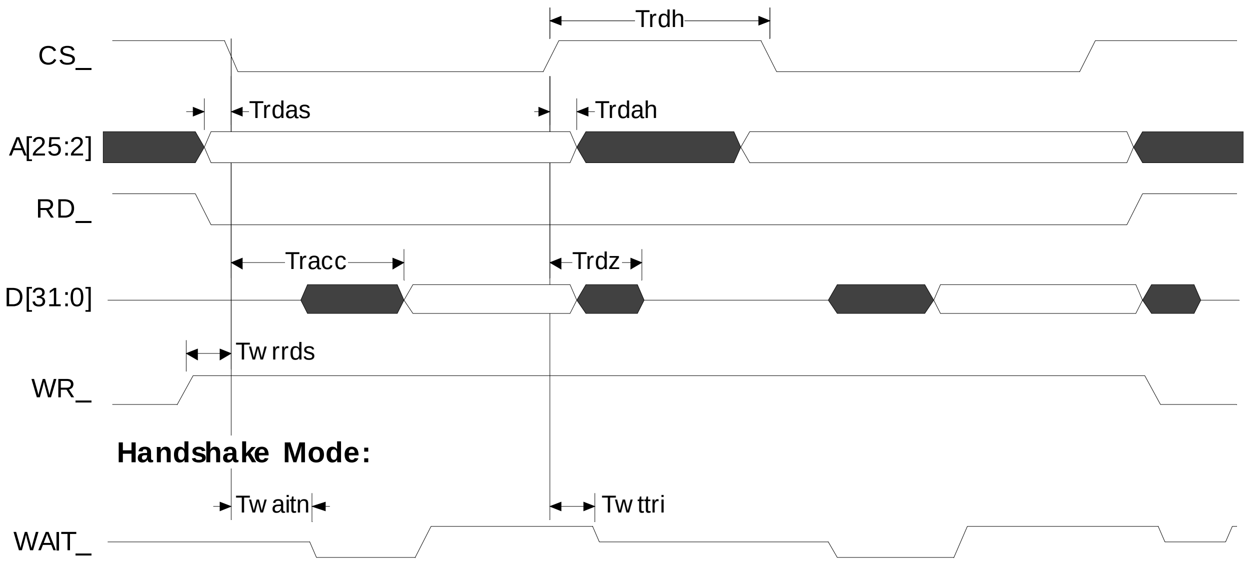


Figure 4.36: Register or Memory Burst Write: 32Bit Direct Type A

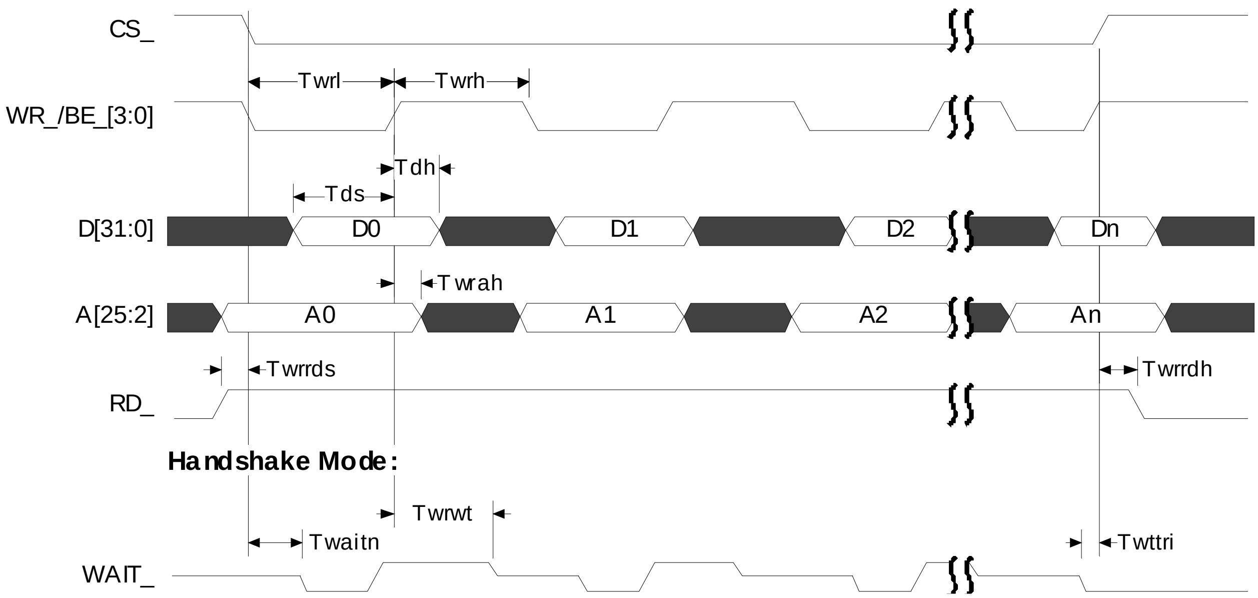
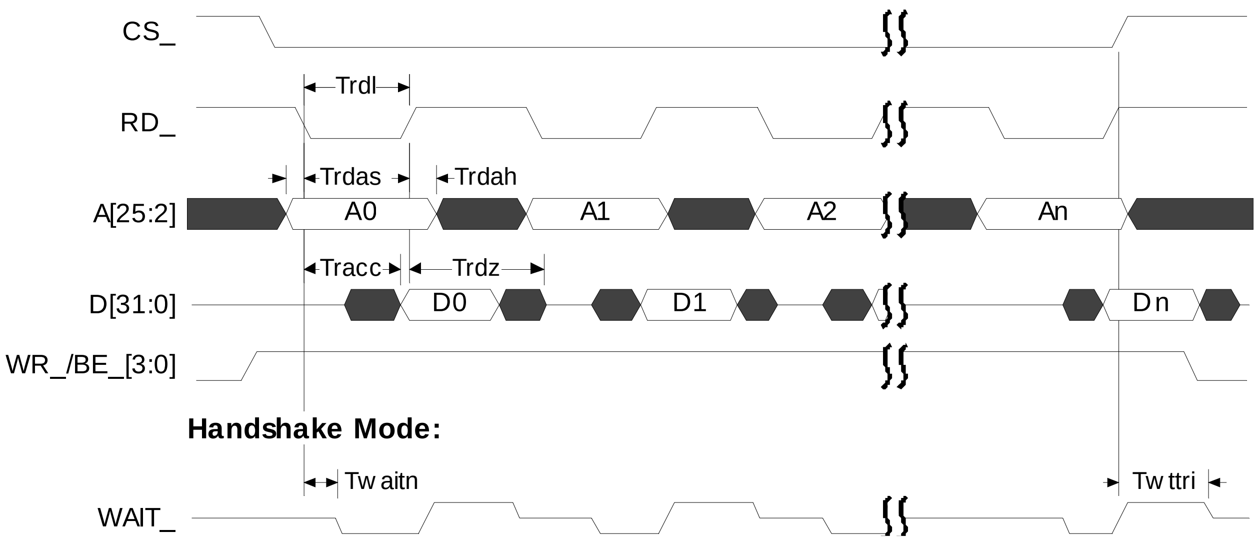


Figure 4.37: Register or Memory Burst Read: 32Bit Direct Type A



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 26 — Handshake	N/A
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 Synchronous Register Access: (5*Host Clock) + 21 SRAM Access: (7*Memory clock) + 21	N/A
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

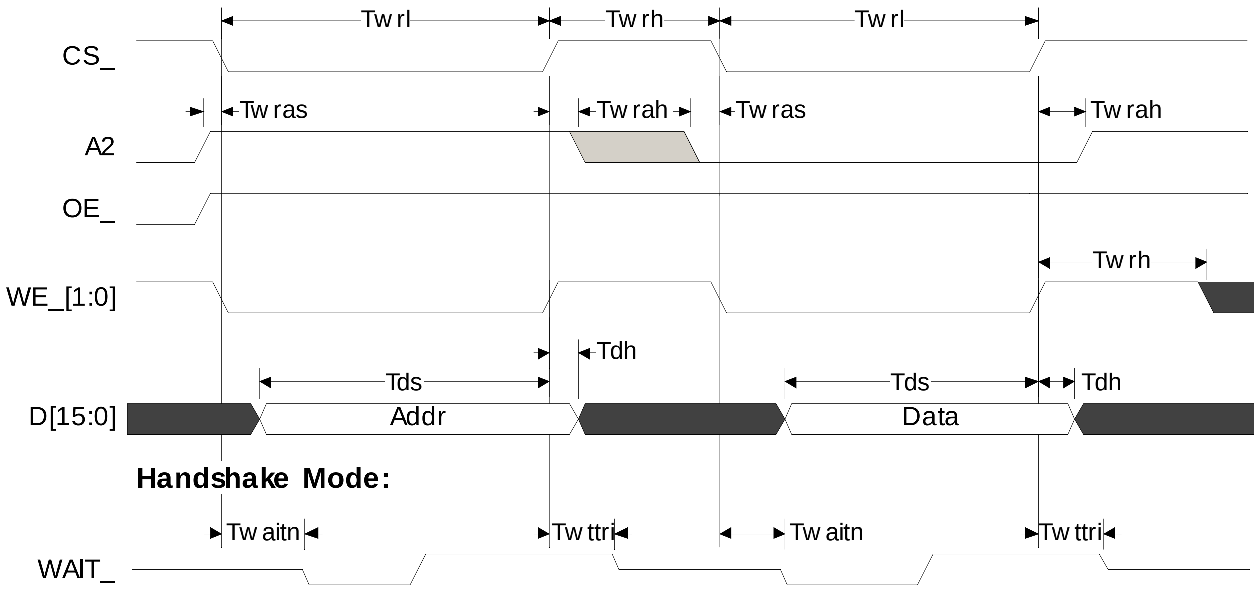


Table 4.32: Register Write D[15:0] Bit Mapping for Addr

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

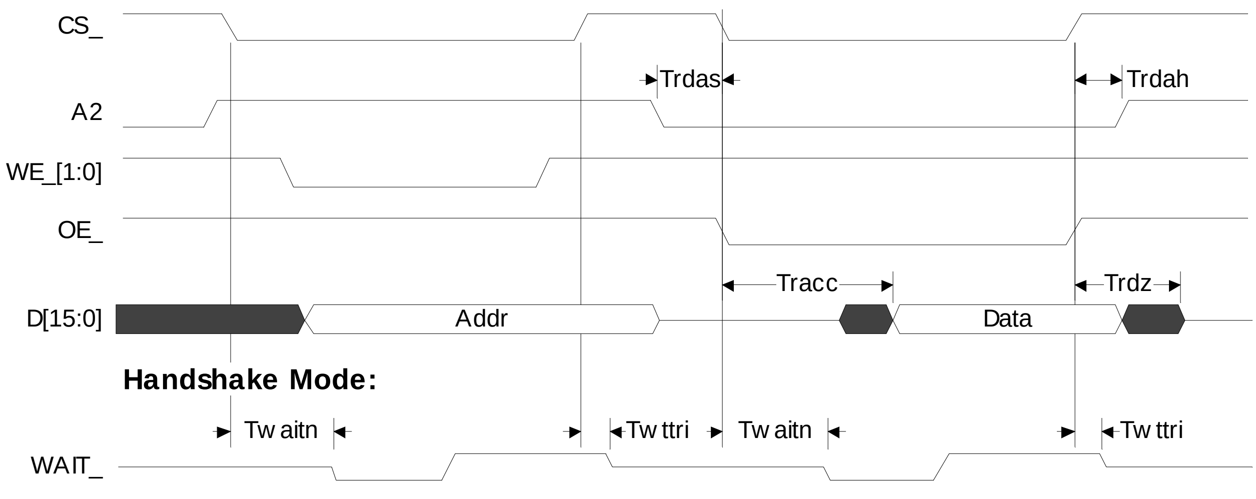


Table 4.33: Register Read D[15:0] Bit Mapping for Addr

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

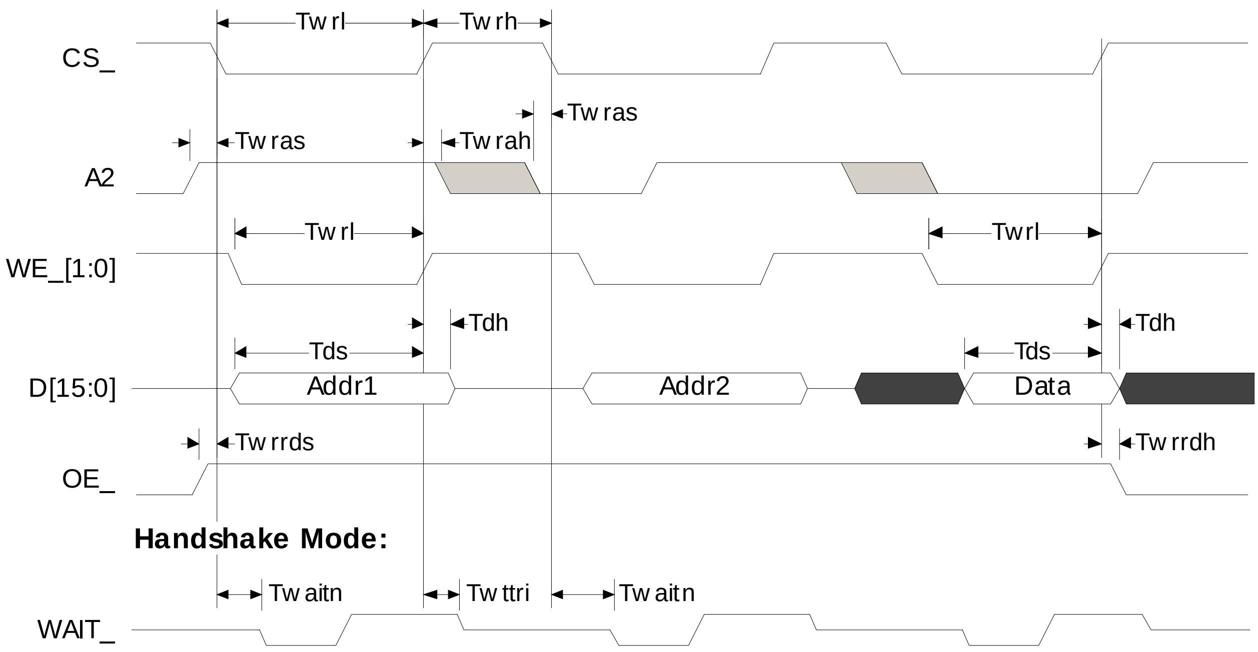


Table 4.34: Memory Write D[15:0] Bit Mapping for Addr1 and Addr2

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

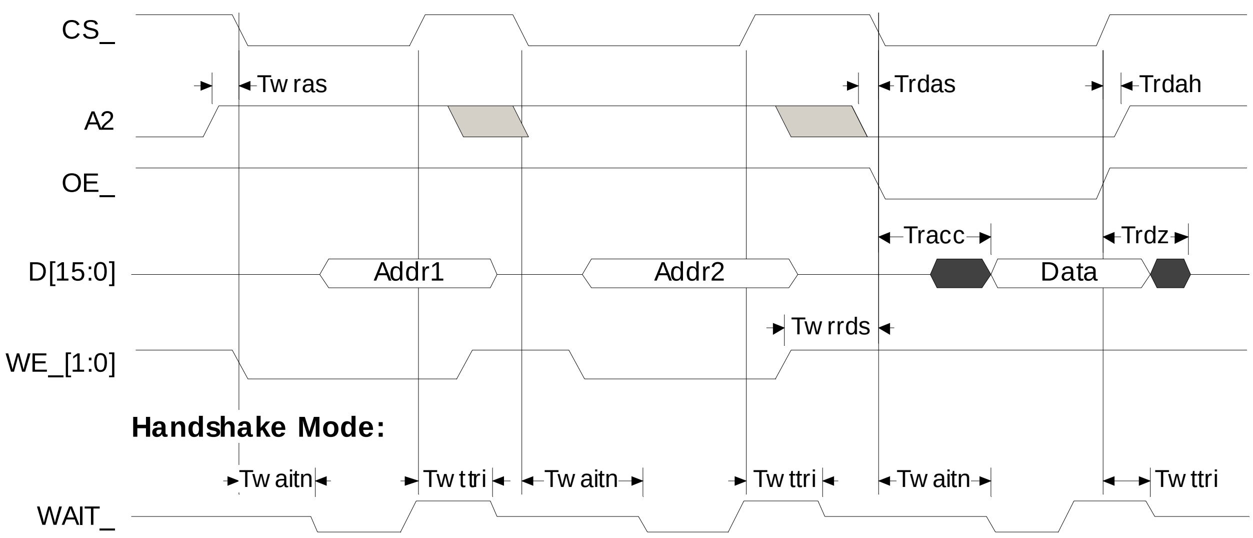


Table 4.35: Memory Read D[15:0] Bit Mapping for Addr1 and Addr2

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

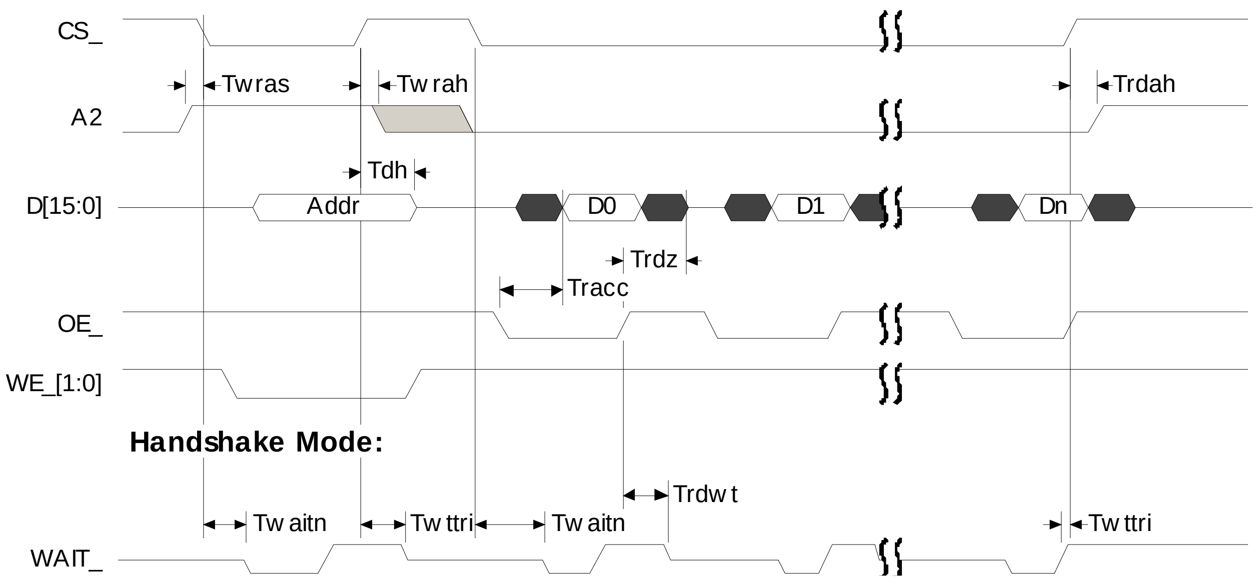


Table 4.36: Register Read D[15:0] Bit Mapping for Addr

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

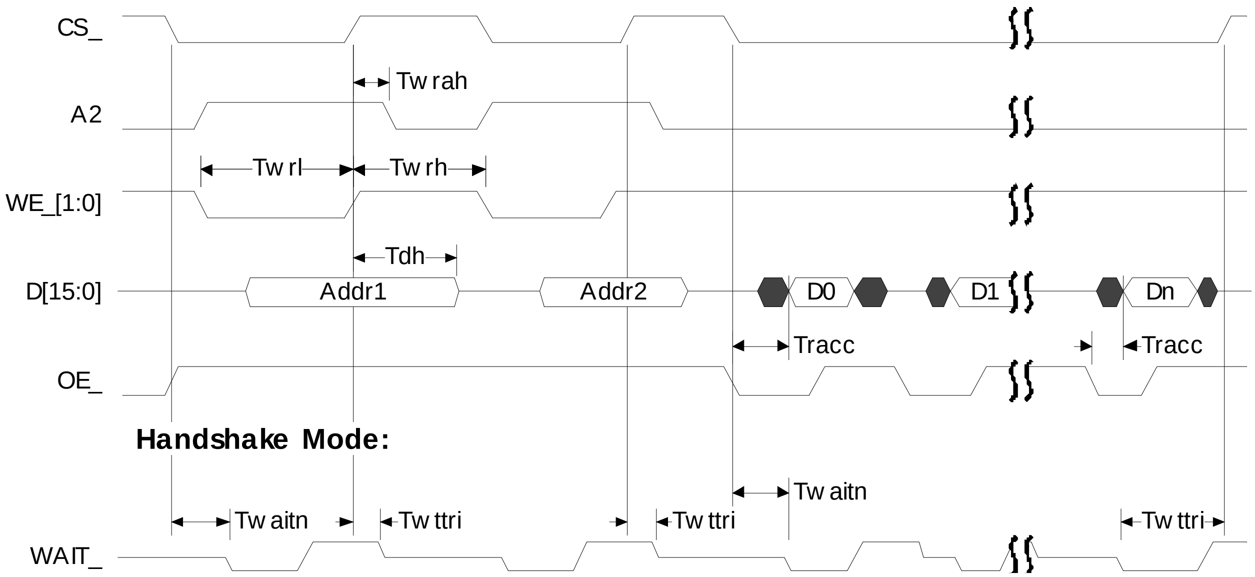


Table 4.37: Memory Read D[15:0] Bit Mapping for Addr1 and Addr2

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

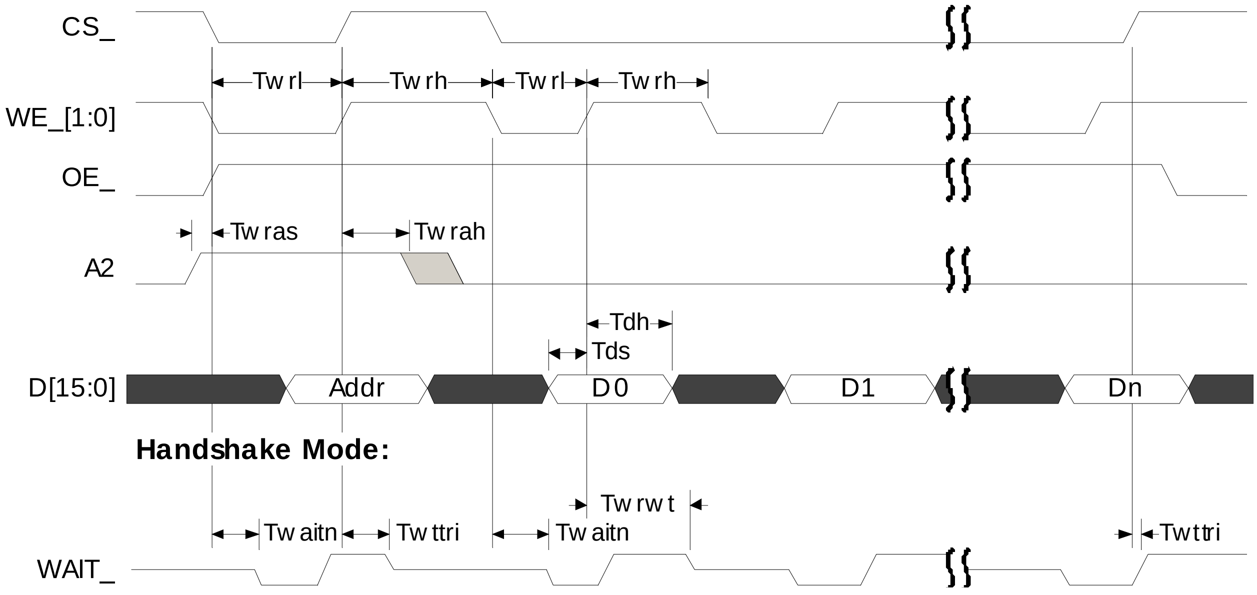


Table 4.38: Register Write D[15:0] Bit Mapping for Addr

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

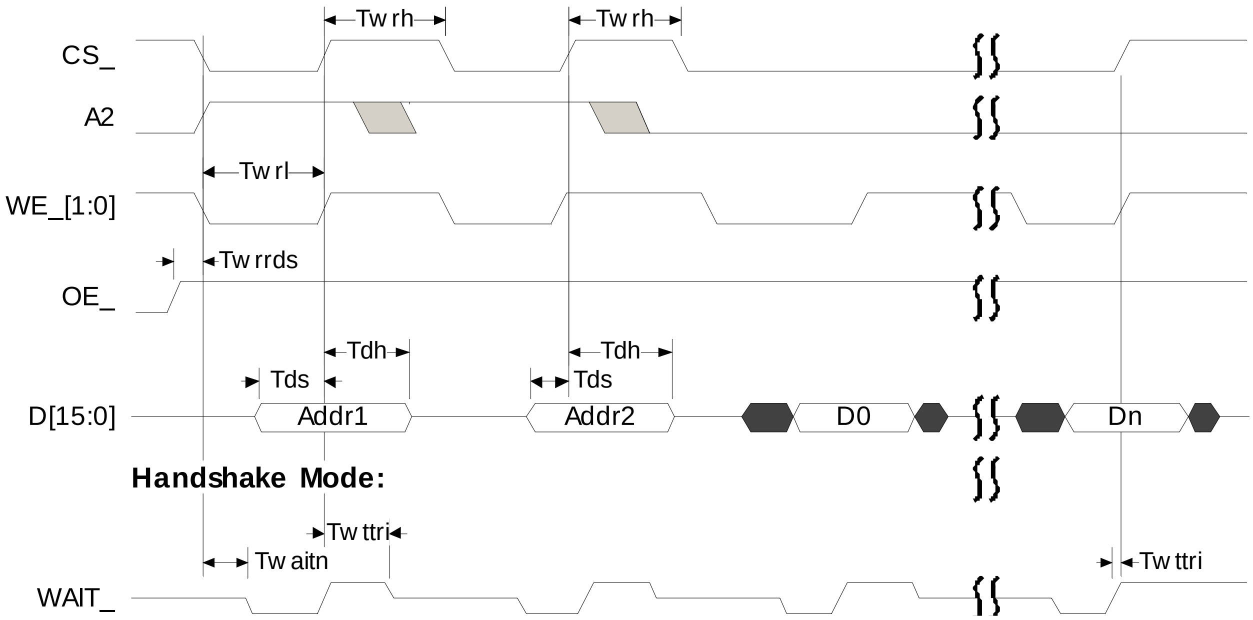


Table 4.39: Memory Write D[15:0] Bit Mapping for Addr1 and Addr2

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

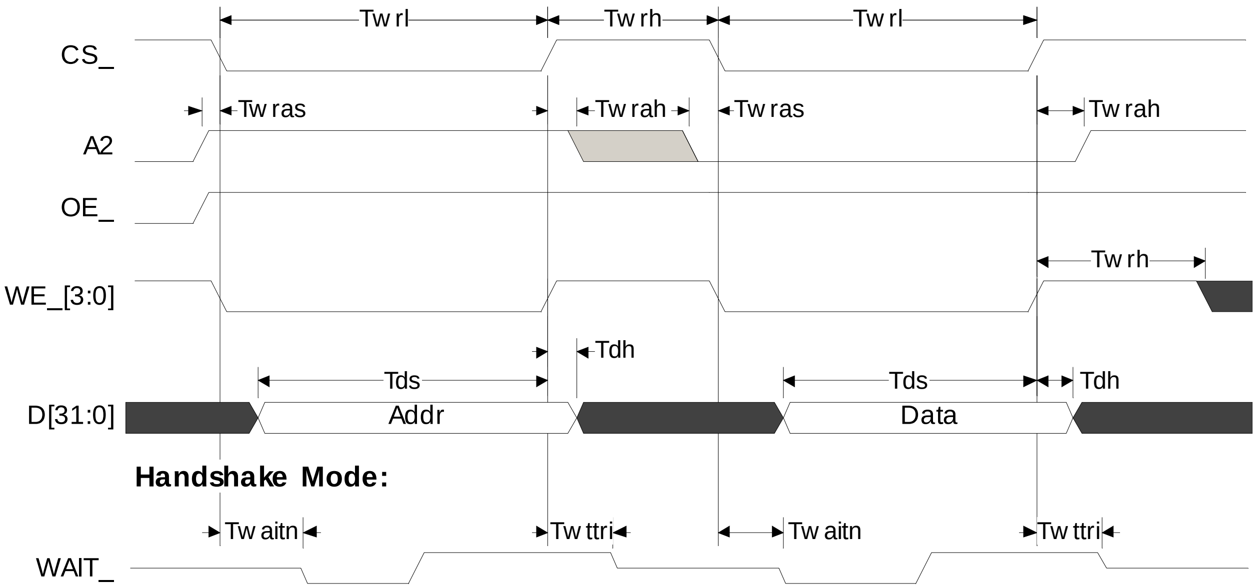


Table 4.40: Register Write D[31:0] Bit Mapping for Addr

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

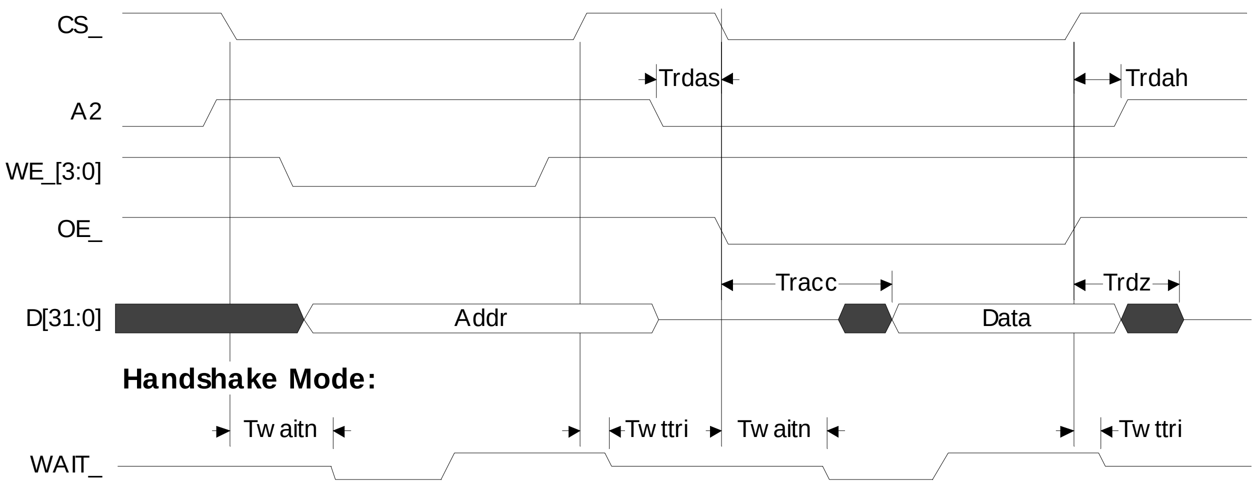


Table 4.41: Register Read D[31:0] Bit Mapping for Addr

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

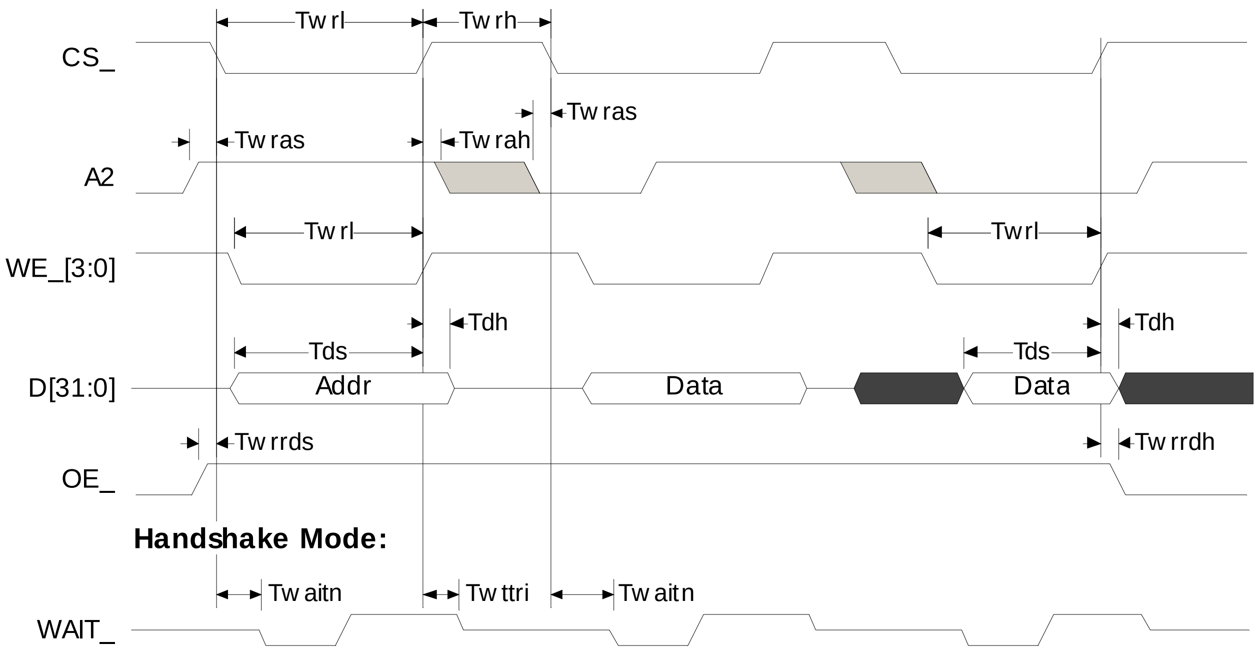


Table 4.42: Memory Write D[31:0] Bit Mapping for Addr

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

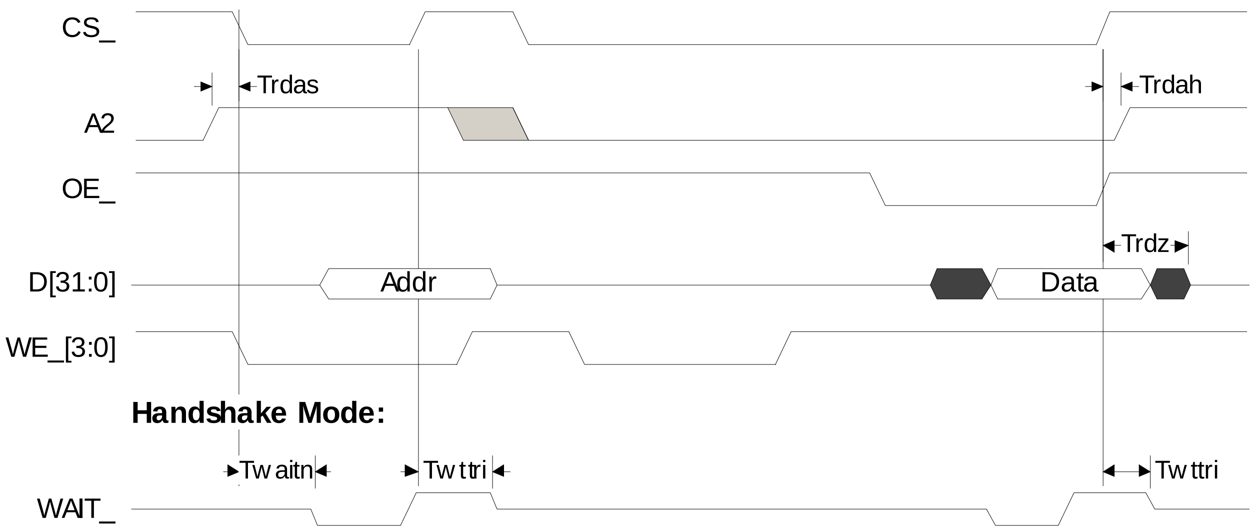


Table 4.43: Memory Read D[31:0] Bit Mapping for Addr and Addr

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

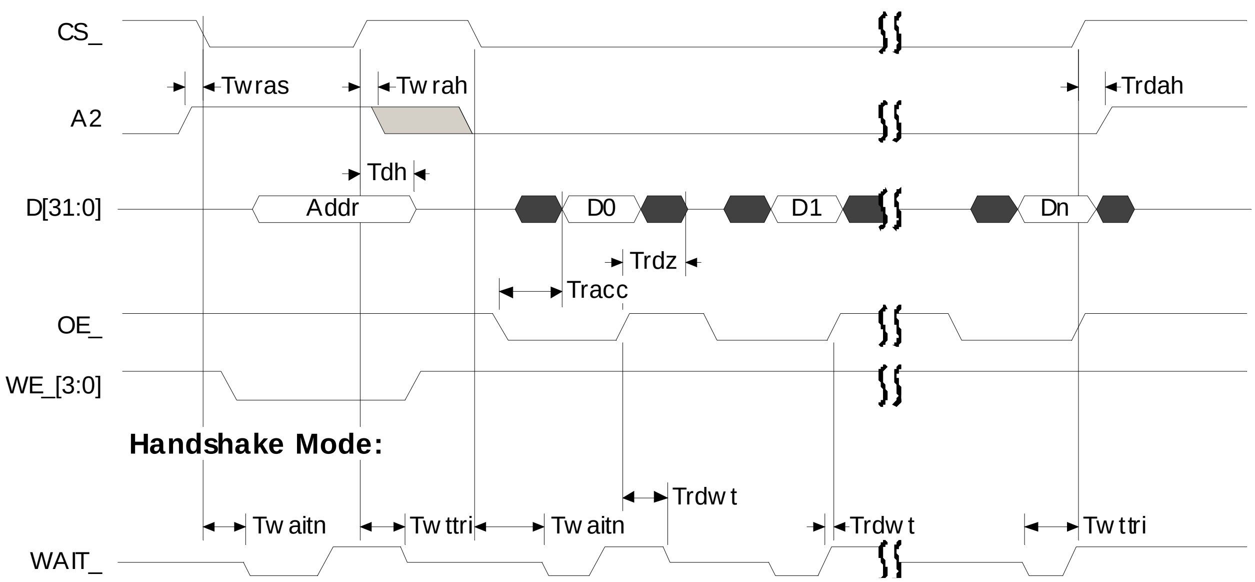


Table 4.44: Register Read D[31:0] Bit Mapping for Addr

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

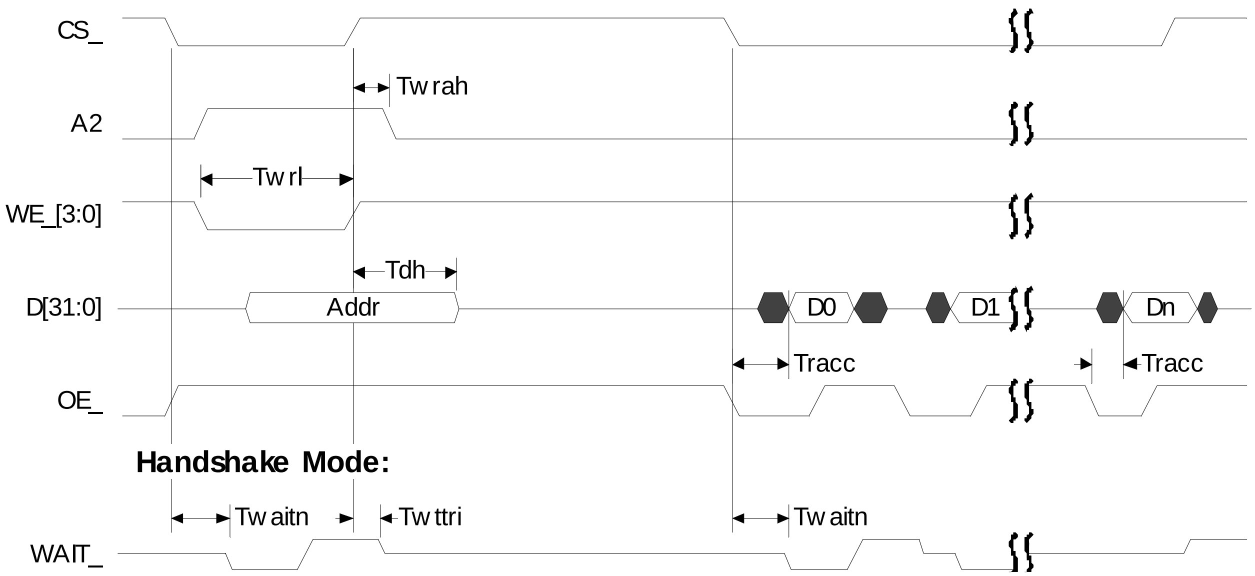


Table 4.45: Memory Read D[31:0] Bit Mapping for Addr

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

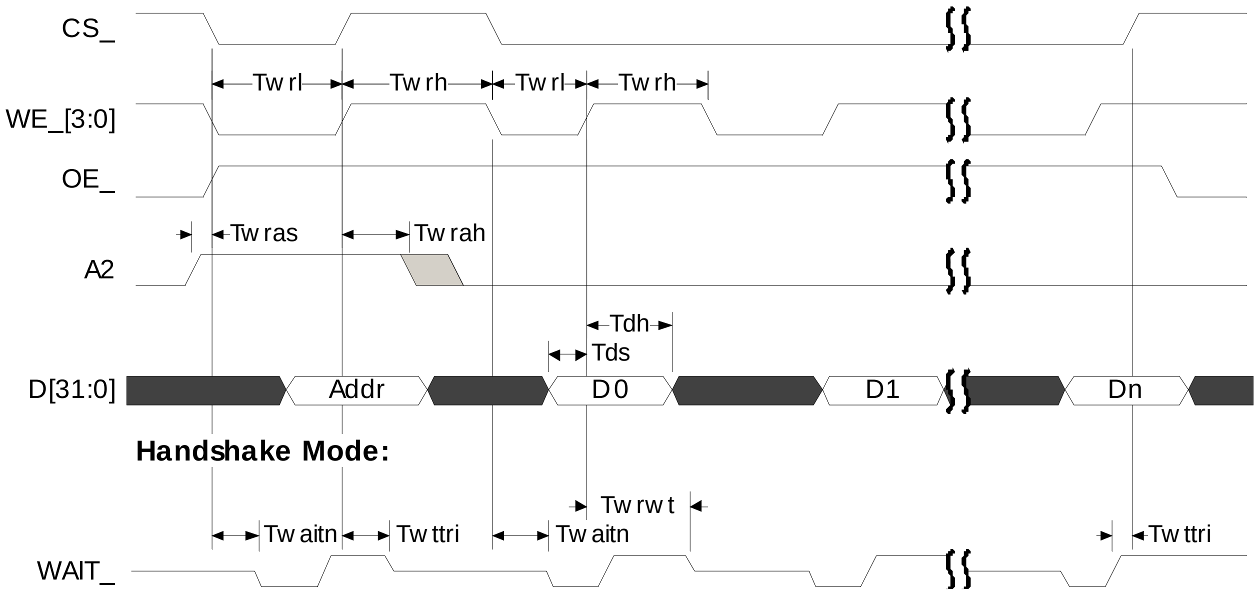


Table 4.46: Register Write D[31:0] Bit Mapping for Addr*

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

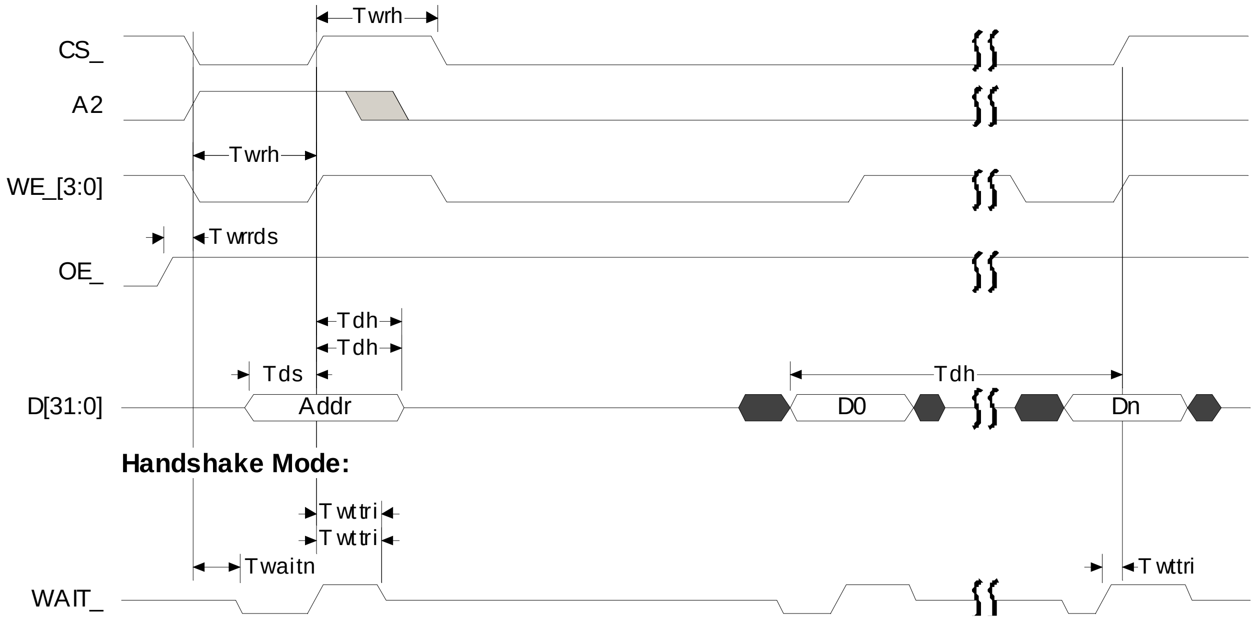
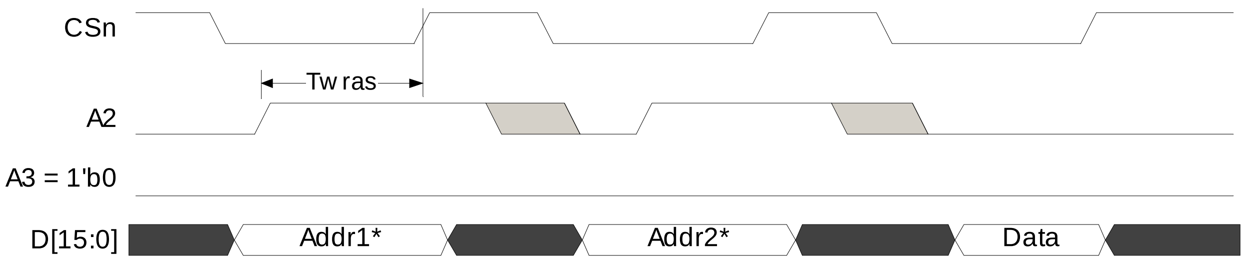


Table 4.47: Memory Write D[31:0] Bit Mapping for Addr1* and Addr2*

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

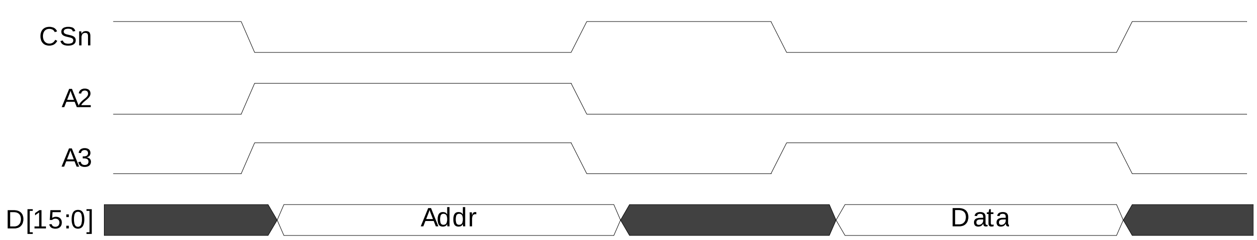
Figure 4.54: One-channel Access, Indirect Addressing



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

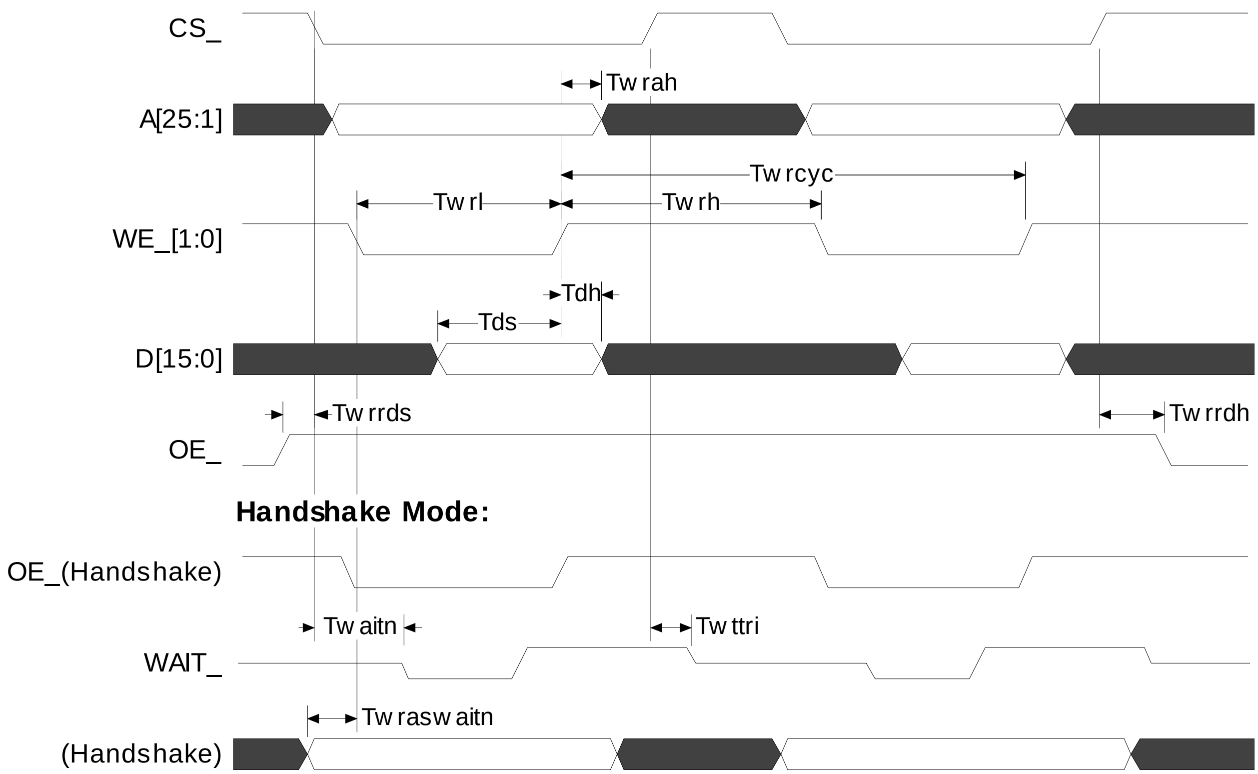


Figure 4.58: CS_-controlled Write: 16Bit Direct Type C

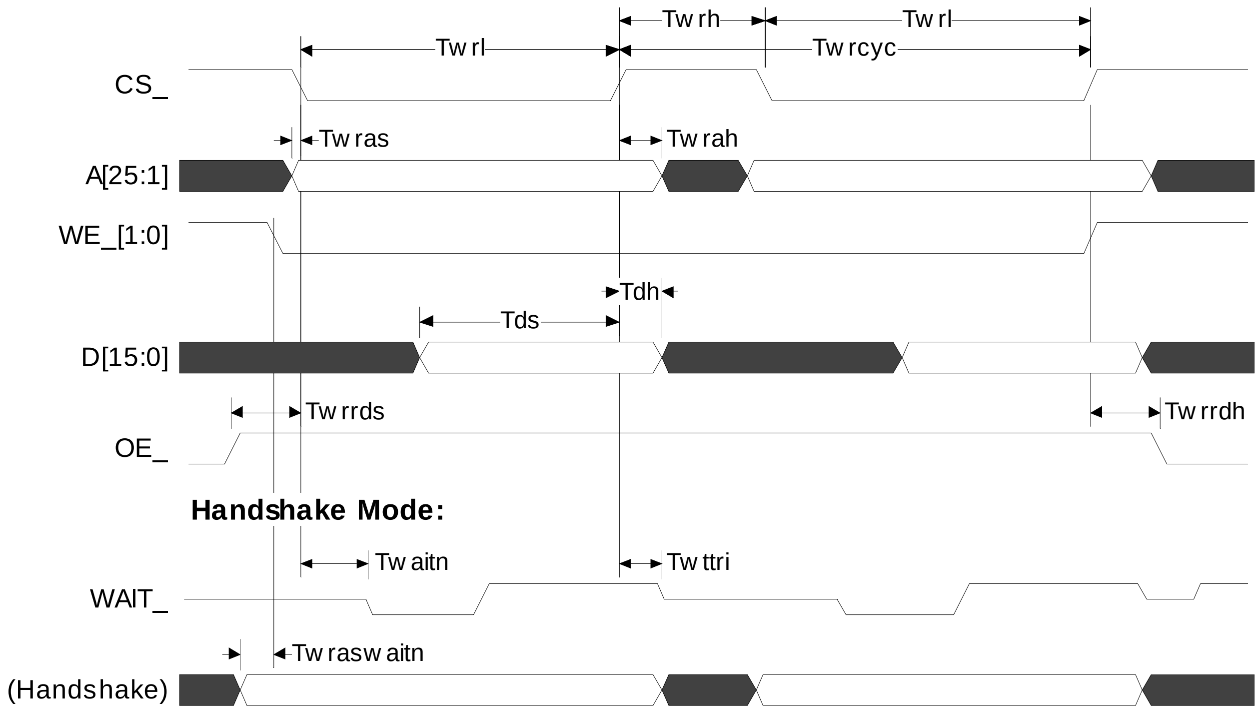


Figure 4.59: OE_-controlled Read: 16Bit Direct Type C

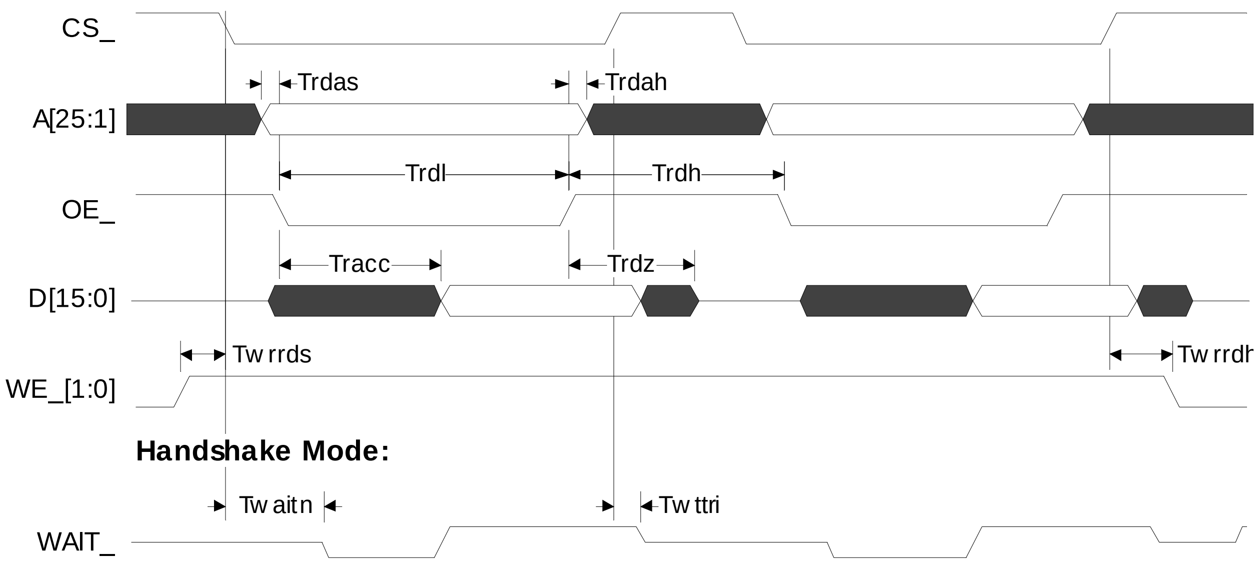


Figure 4.60: CS_-controlled Read: 16Bit Direct Type C

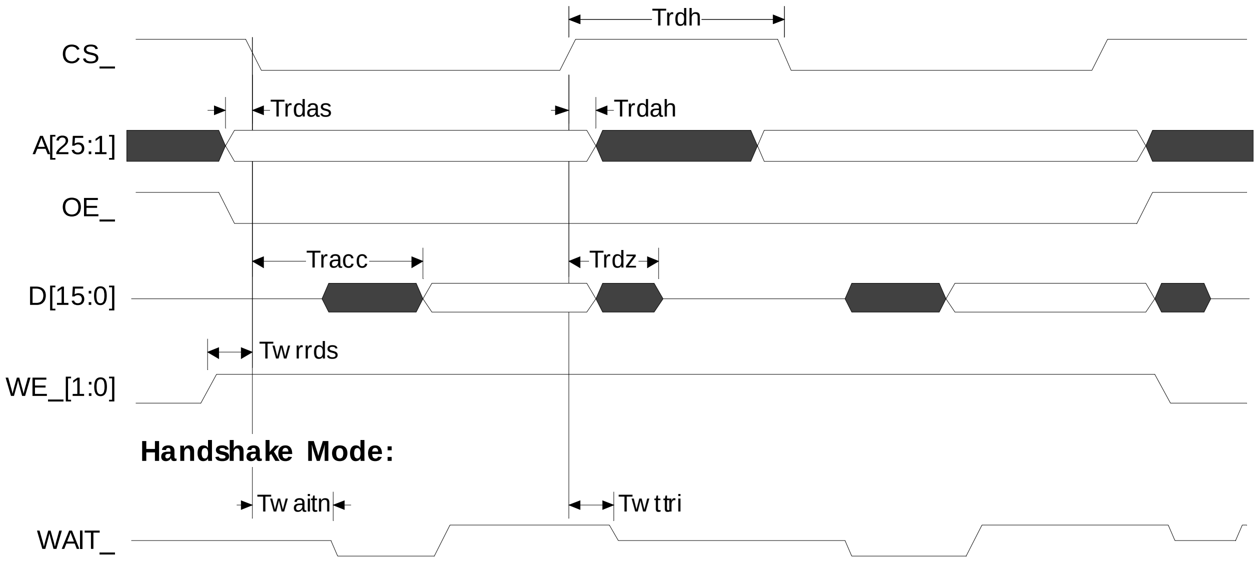


Figure 4.61: Register or Memory Burst Write: 16Bit Direct Type C

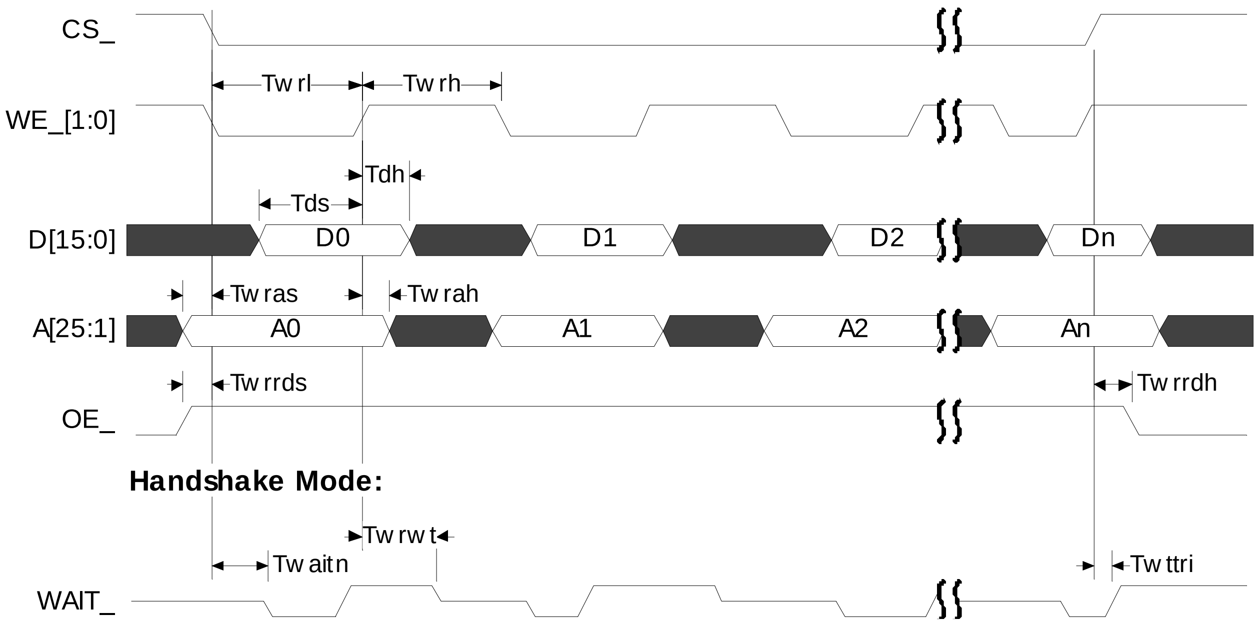
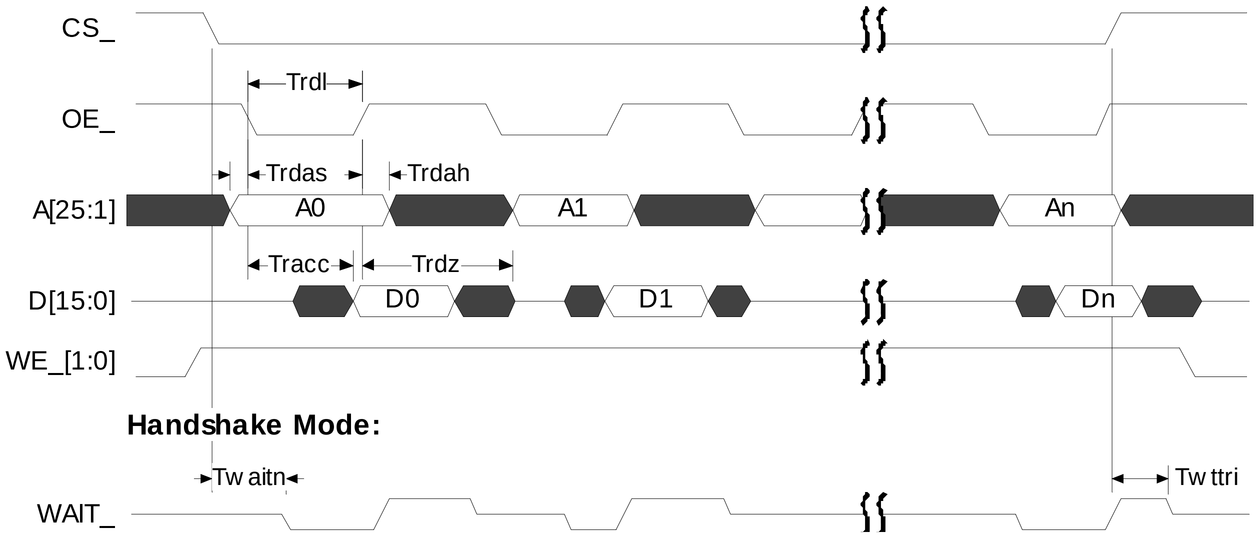


Figure 4.62: Register or Memory Burst Read: 16Bit Type C



4.4.4.3.5 Type C Host Interface Timings: 32Bit Direct

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

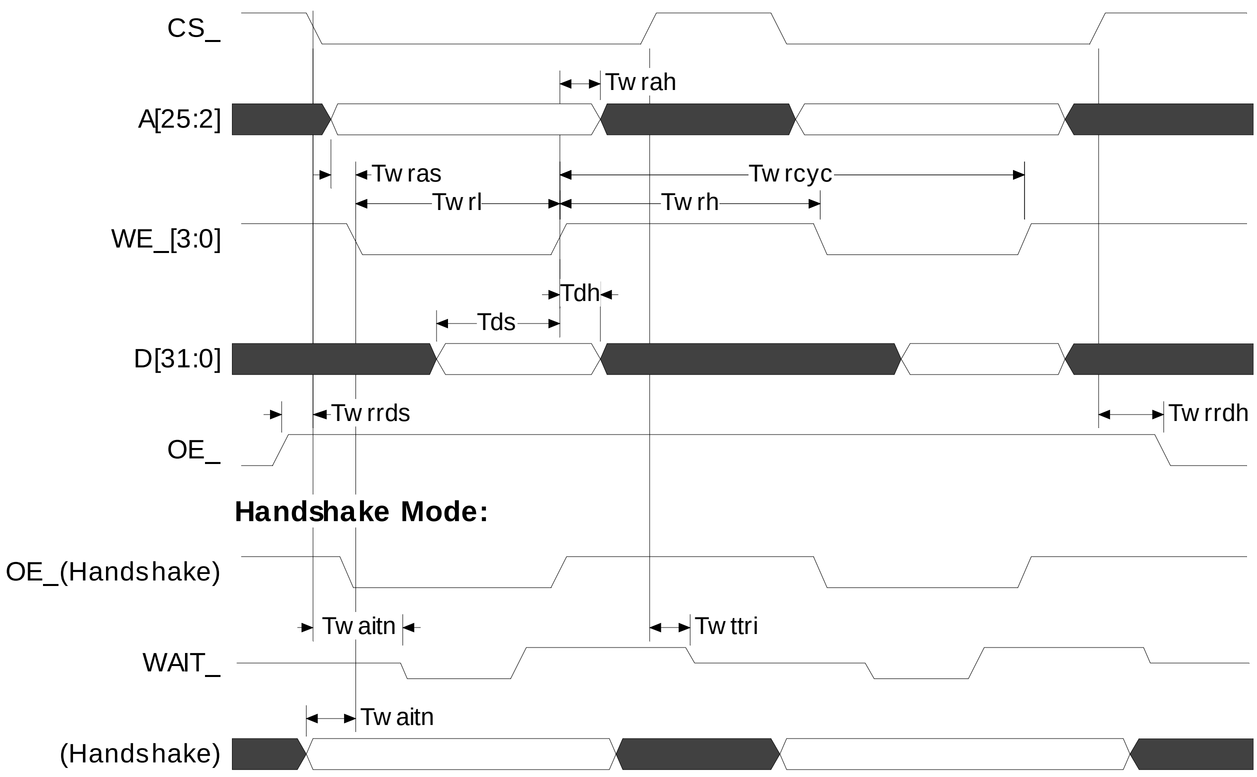


Figure 4.64: CS_-controlled Write: 32Bit Direct Type C

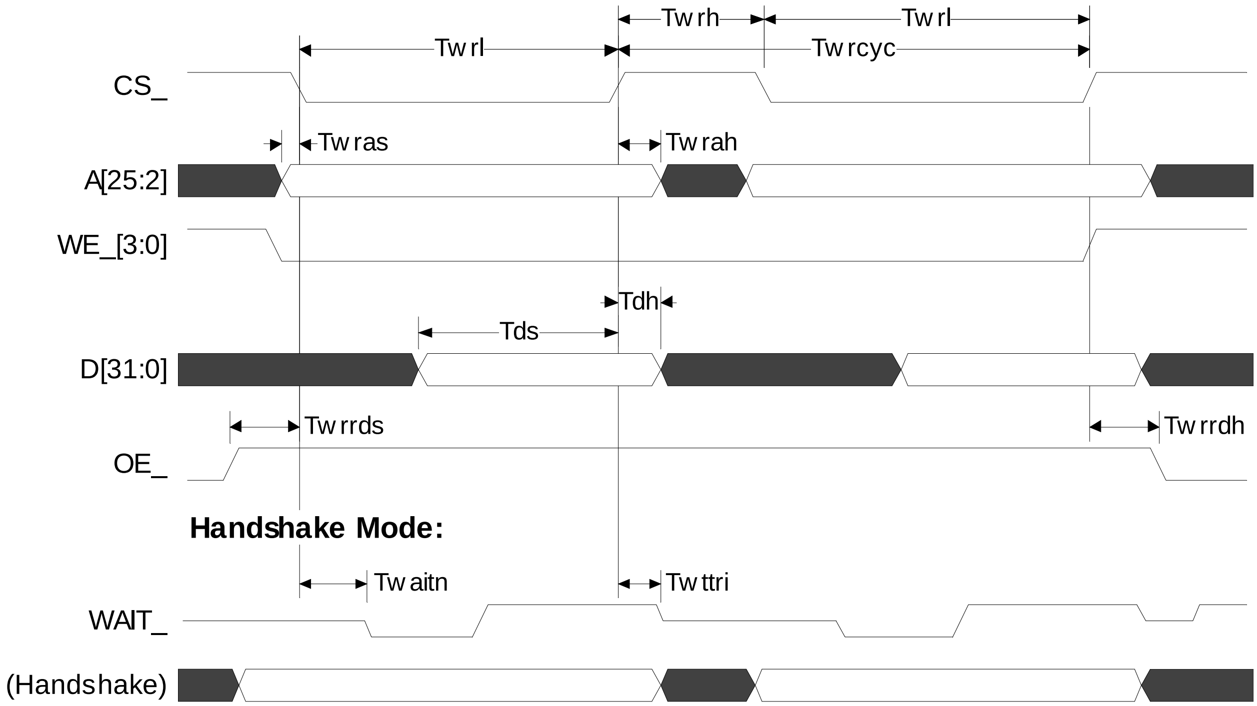


Figure 4.65: OE_-controlled Read: 32Bit Direct Type C

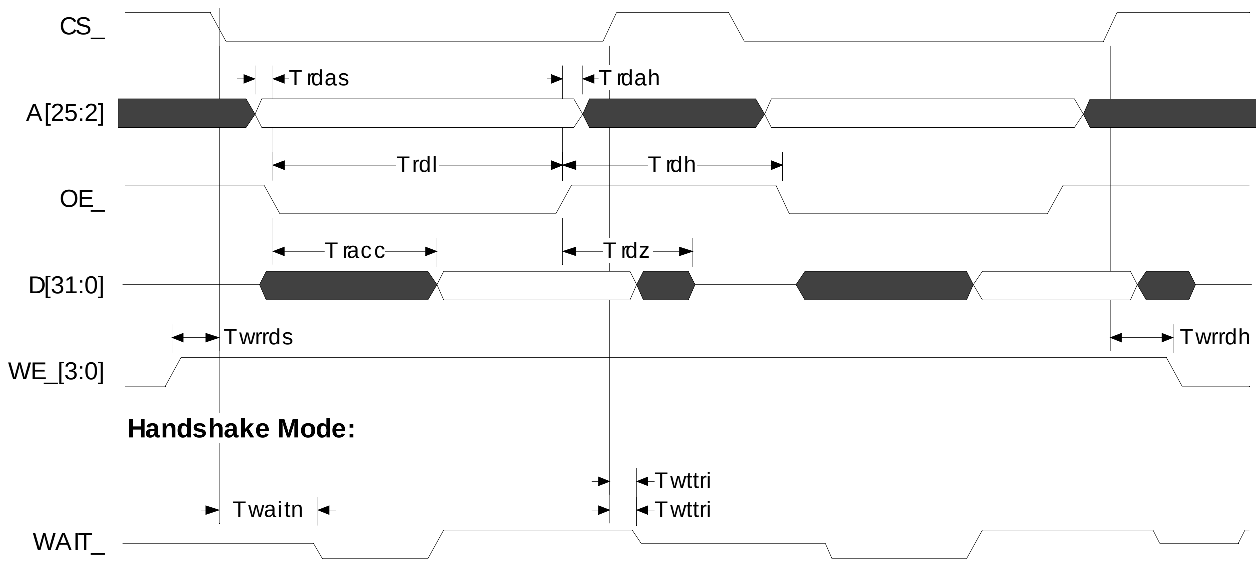


Figure 4.66: CS_-controlled Read: 32Bit Direct Type C

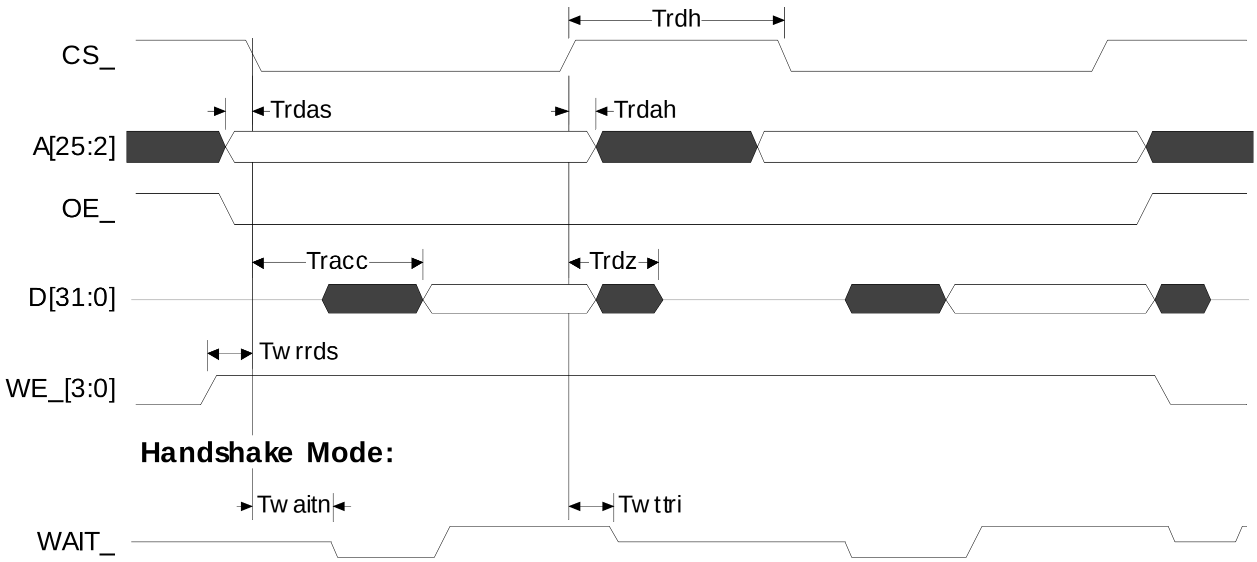


Figure 4.67: Register or Memory Burst Write: 32Bit Direct Type C

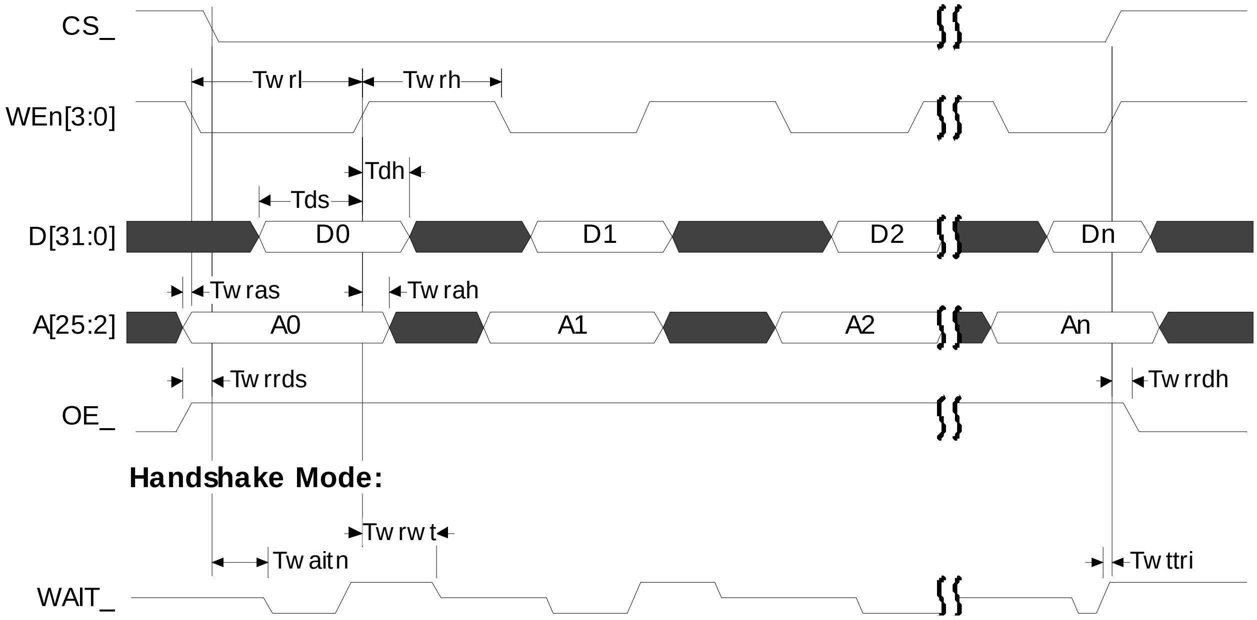
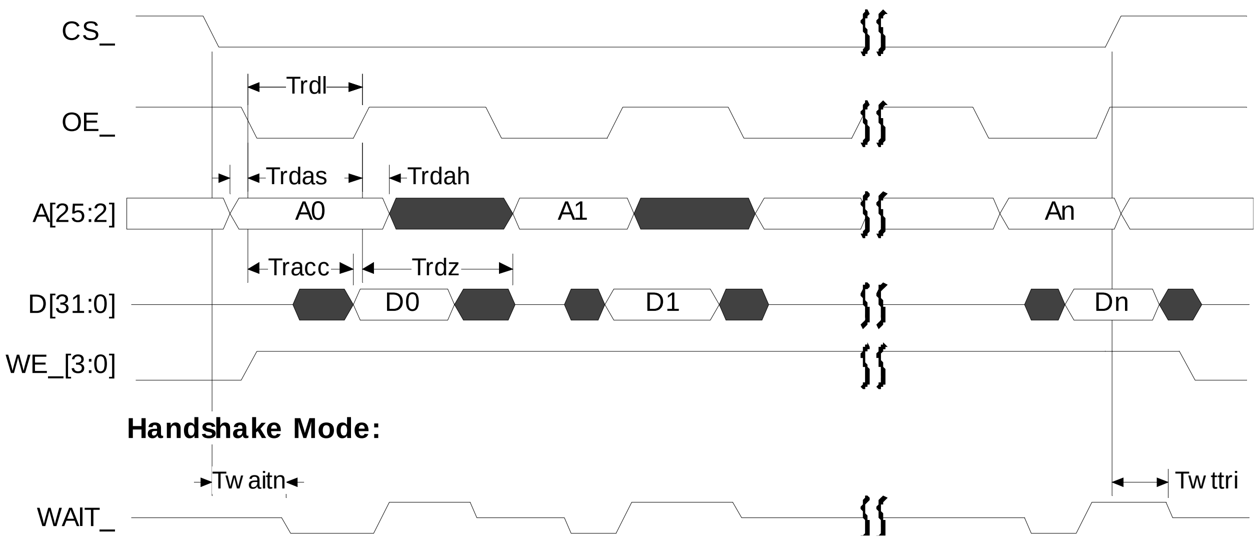


Figure 4.68: Register or Memory Burst Read: 32Bit Direct Type C



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

<table>
<tr><th>Symbol</th><th>Description</th><th colspan="2">Min (ns):
Time and Conditions</th><th>Max (ns):
Time and Conditions</th></tr>
<tr><td>Tdh</td><td>Write cycles:
Data hold time from rising edge of CS_ or WE_, whichever comes first</td><td colspan="2">0</td><td>N/A</td></tr>
<tr><td>Tds</td><td>Write cycles:
Data setup time to rising edge of either CS_ or WE_, whichever comes first.</td><td colspan="2">7</td><td>N/A</td></tr>
<tr><td rowspan="3">Tracc</td><td rowspan="3">Read cycles:
Maximum read access time from the beginning of the read cycle to the first valid data access.</td><td colspan="2" rowspan="3">N/A</td><td>Asynchronous Register Access: 26</td></tr>
<tr><td>Synchronous Register Access: (5*Host Clock) + 21</td></tr>
<tr><td>SRAM Access: (7*Memory clock) + 21</td></tr>
<tr><td>Trdah</td><td>Read cycles:
Address hold time from rising edge of CS_ or RD_, whichever comes first.</td><td colspan="2">0</td><td>N/A</td></tr>
<tr><td>Trdas</td><td>Read cycles:
Address setup time to falling edge of CS_ or OE_, whichever comes last.</td><td colspan="2">0</td><td>N/A</td></tr>
<tr><td rowspan="2">Trdh</td><td rowspan="2">Read cycles:
Read enable Inactive time measured from the end of one read cycle to the beginning of the next read cycle.</td><td>5</td><td>No hand-shake</td><td rowspan="2">N/A</td></tr>
<tr><td>26</td><td>Handshake</td></tr>
<tr><td rowspan="3">Trdl</td><td rowspan="3">Read cycles:
Read enable active low time.
CS_-controlled read cycles: CS_ low time
OE_-controlled read cycles: OE_ low time</td><td colspan="2">Asynchronous Register Access: 26</td><td rowspan="3">N/A</td></tr>
<tr><td colspan="2">Synchronous Register Access: (5*Host Clock) + 21</td></tr>
<tr><td colspan="2">SRAM Access: (7*Memory clock) + 21</td></tr>
<tr><td>Trdwt</td><td>Time from rising edge of OE_ to falling edge of WAIT_</td><td colspan="2">N/A</td><td>20</td></tr>
<tr><td>Trdz</td><td>Time from rising edge of CS_ or OE_, whichever comes first, to the data bus floating state</td><td colspan="2">3</td><td>15</td></tr>
<tr><td>Twaitn</td><td>WAIT_/RDY_ assertion time from the falling edge of CS_.</td><td colspan="2">N/A</td><td>20</td></tr>
<tr><td>Twrah</td><td>Write cycles:
Address hold time from the rising edge of CS_ or WE_, whichever comes first.</td><td colspan="2">0</td><td>N/A</td></tr>
<tr><td>Twras</td><td>Write cycles:
Address valid setup time to the falling edge of CS_ or WE_, whichever comes first.</td><td colspan="2">0</td><td>N/A</td></tr>
</table>

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 <u>*OR*</u> Host Clock Period <u>*OR*</u> 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

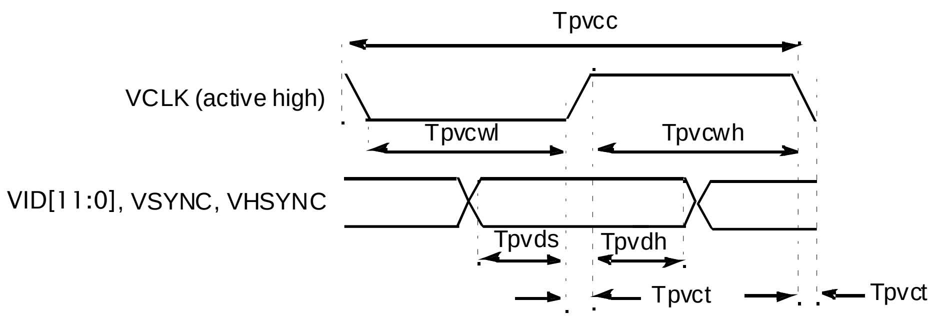


Table 4.49: Video Parallel Input Clock Timing

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)

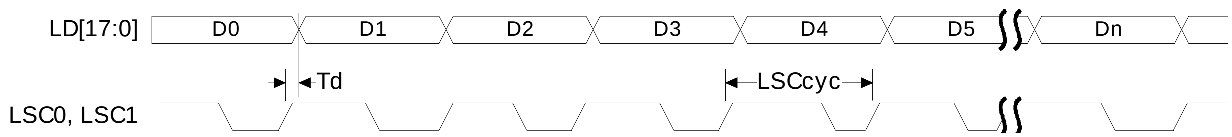


Table 4.50: Parallel Mode Divide-by-one Timing Parameters

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)

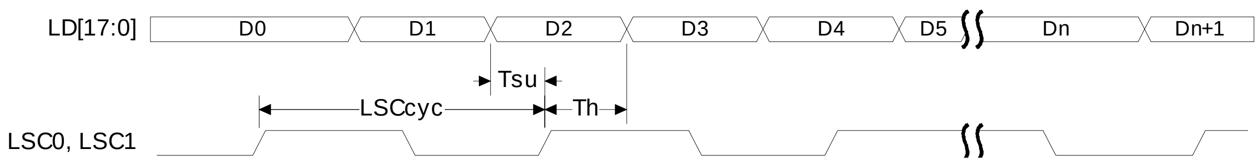


Table 4.51: Parallel Mode Divide-by-two Timing Parameters

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)

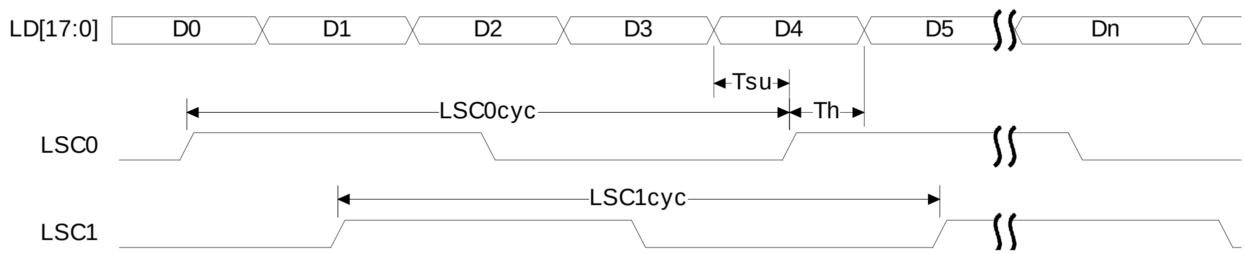


Table 4.52: Parallel Mode Divide-by-four Timing Parameters

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)

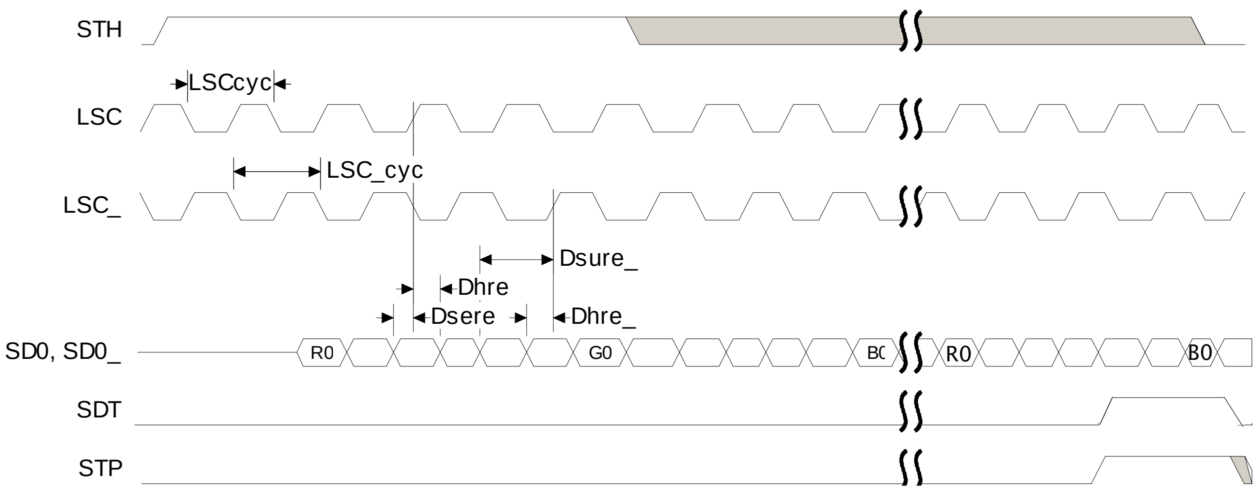


Table 4.53: Serial Mode (10 pF Load) Timing Parameters

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

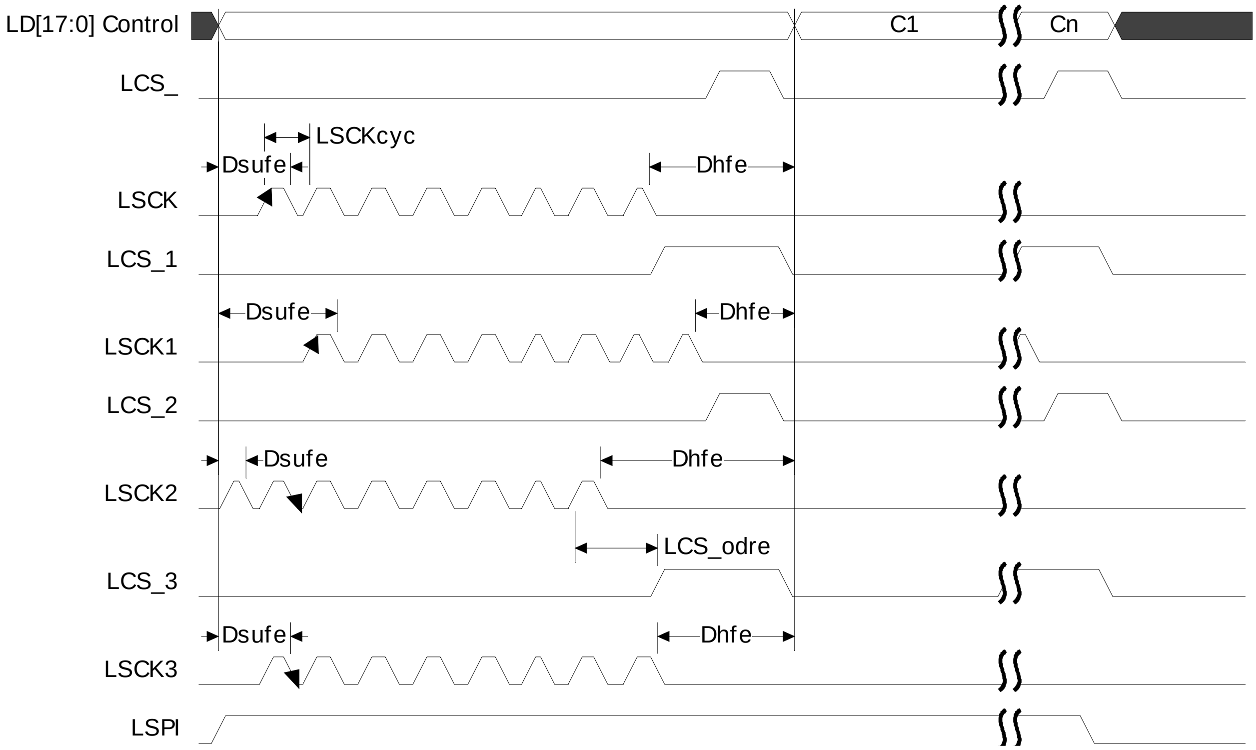


Table 4.54: SPI Timing Parameters

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**	**OSCFO**
K	LD10	LD6	LD11	**GND**	LD9	LD16	TDCVDD	TDCVDD	**GND**	**GND**	MMCVDD	MMCVDD	VGP1	VGP4	**GND**	NC	**AGNDOSC**	**OSCFI**
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	

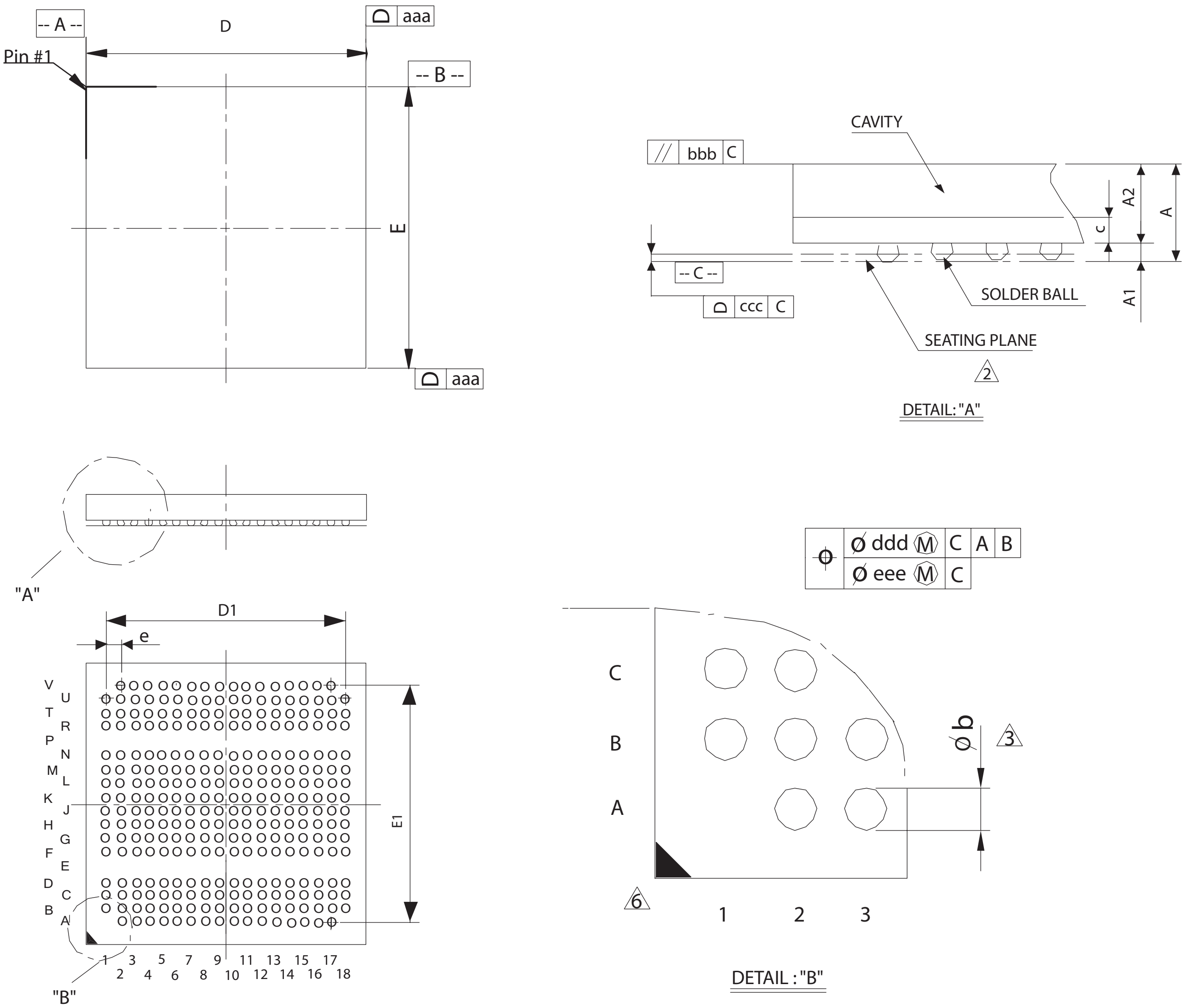
4.5 Mechanical Drawing

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

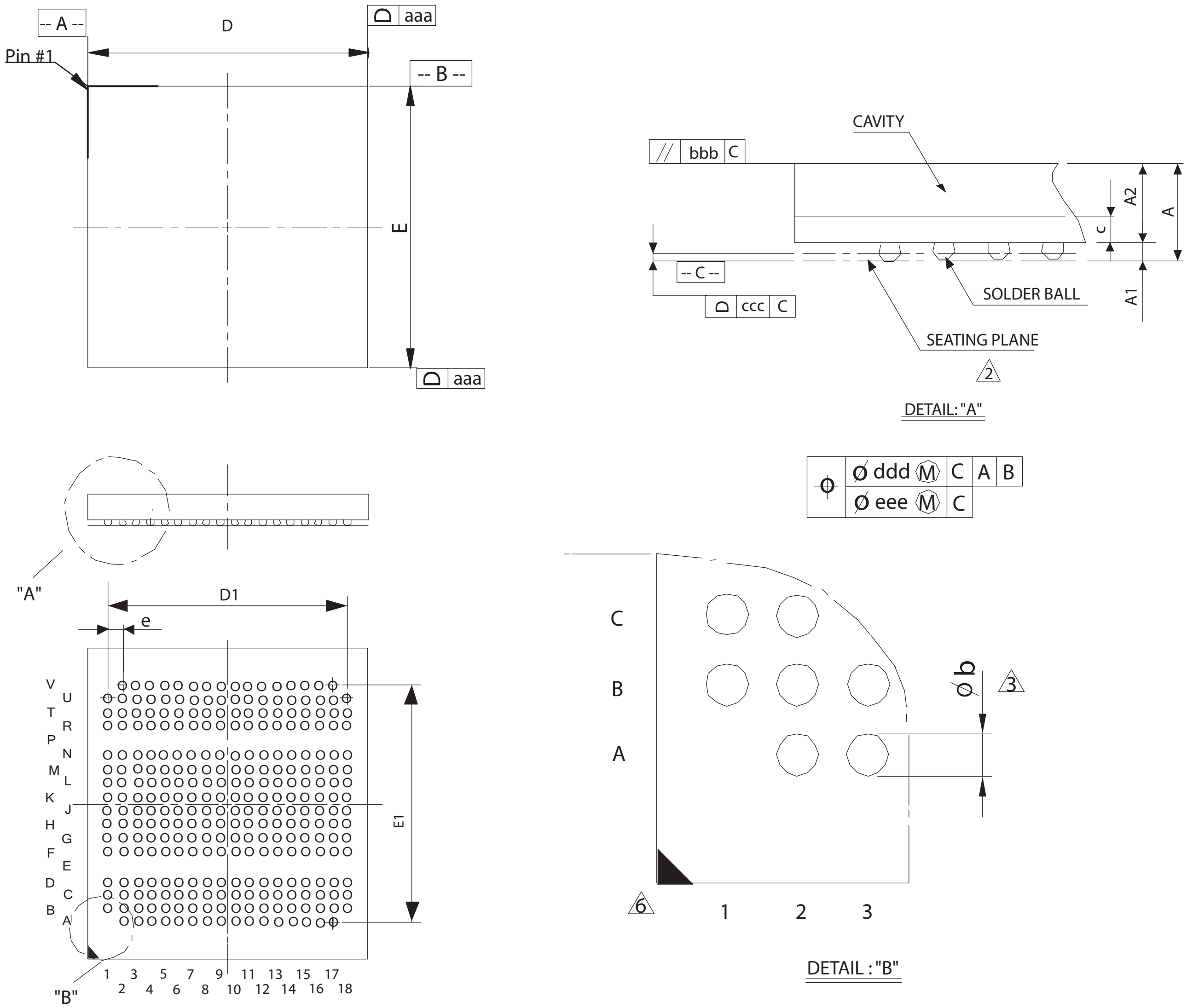


Symbol	Dimension in mm			Dimension in inch		
	MIN	NOM	MAX	MIN	NOM	MAX
A	1.10	1.20	1.40	0.043	0.047	0.055
A1	0.16	0.21	0.26	0.006	0.008	0.010
A2	0.91	0.96	1.01	0.036	0.038	0.040
c	0.22	0.26	0.30	0.009	0.010	0.012
D	9.90	10.00	10.10	0.390	0.394	0.398
E	9.90	10.00	10.10	0.390	0.394	0.398
D1	- - -	8.50	- - -	- - -	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

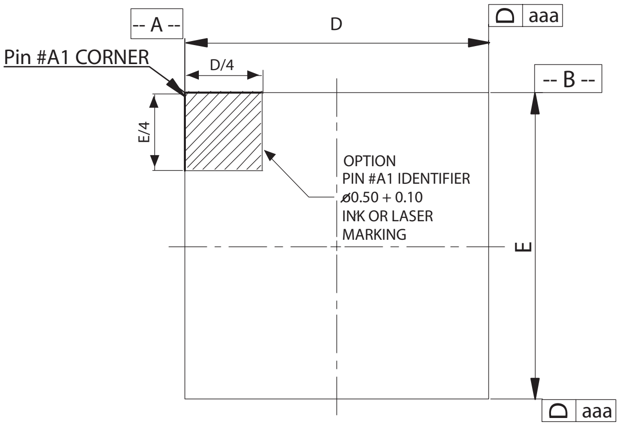


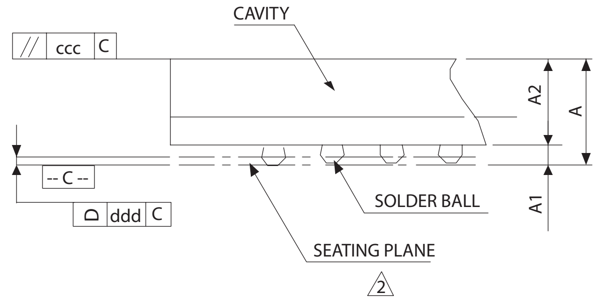
Symbol	Dimension in mm			Dimension in inch		
	MIN	NOM	MAX	MIN	NOM	MAX
A	1.10	1.20	1.40	0.043	0.047	0.055
A1	0.16	0.21	0.26	0.006	0.008	0.010
A2	0.91	0.96	1.01	0.036	0.038	0.040
c	0.22	0.26	0.30	0.009	0.010	0.012
D	9.90	10.00	10.10	0.390	0.394	0.398
E	11.90	12.00	12.10	0.469	0.472	0.476
D1	- - -	8.50	- - -	- - -	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:

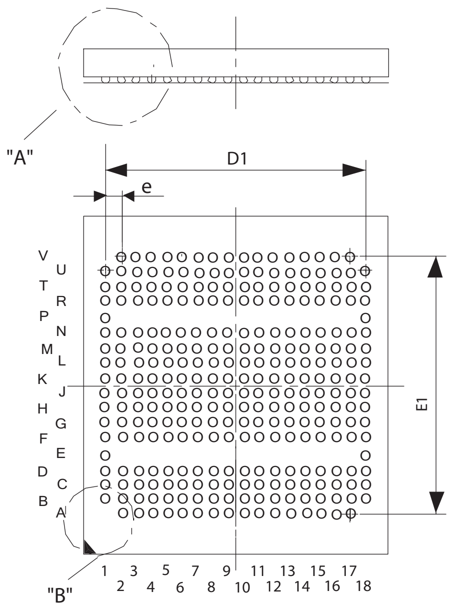
1. CONTROLLING DIMENSION: MILLIMETER
2. (△) PRIMARY DATUM C AND SEATING PLANE ARE DEFINED BY THE SPHERICAL CROWNS OF THE SOLDER BALLS.
3. (△) DIMENSION b IS MEASURED AT THE MAXIMUM SOLDER BALL DIAMETER, PARALLEL TO PRIMARY DATUM C.
4. THERE SHALL BE A MINIMUM CLEARANCE OF 0.25 mm BETWEEN THE EDGE OF 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

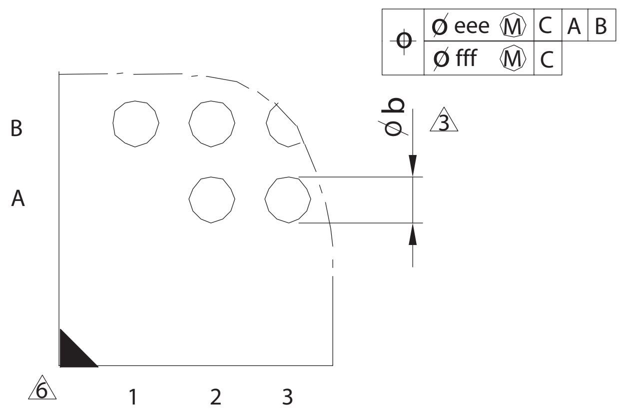
Figure 4.79: GoForce 5500-XT 10 x 12 mm Package Mechanical Drawing





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. (2) PRIMARY DATUM C AND SEATING PLANE ARE DEFINED BY THE SPHERICAL CROWNS OF THE SOLDER BALLS.
3. (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. (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

CPU's Point-of-View

Address (A[23:0])	
000000h-003FFFh	Channel 0 (16 KB)
004000h-007FFFh	Channel 1 (16 KB)
008000h-00BFFFh	Channel 2 (16 KB)
	. . .
01C000h-01FFFFh	Channel 7 (16 KB)
020000h-023FFFh	Protected Channel (16 KB)
024000h-3FFFFFh	Reserved (3 MB, 880 KB)
400000h-49FFFFh	Internal Memory (640 KB)
4A0000h-7FFFFFh	Reserved (3 MB, 384 KB)
800000h-FFFFFFh	External Memory (8 MB)

16 MB Memory Footprint

GoForce 5500's Point-of-View

Address (A[25:0])	
0000000h-0003FFFh	Channel 0 (16 KB)
0004000h-0007FFFh	Channel 1 (16 KB)
0008000h-000BFFFh	Channel 2 (16 KB)
	. . .
001C000h-001FFFFh	Channel 7 (16 KB)
0020000h-0023FFFh	Protected Channel (16 KB)
0024000h-03FFFFFh	Reserved (3 MB, 880 KB)
0400000h-049FFFFh	Internal Memory (640 KB)
04A0000h-1FFFFFFh	Reserved (27 MB, 384 KB)
2000000h-27FFFFFh	External Memory (8 MB)
2800000h-3FFFFFFh	Reserved (24 MB)

64 MB Memory Footprint

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

CPU's Point-of-View

Address (A[22:0])	
000000h-003FFFh	Channel 0 (16 KB)
004000h-007FFFh	Channel 1 (16 KB)
008000h-00BFFFh	Channel 2 (16 KB)
	...
01C000h-01FFFFh	Channel 7 (16 KB)
020000h-023FFFh	Protected Channel (16 KB)
024000h-1FFFFFh	Reserved (1 MB, 880 KB)
200000h-29FFFFh	Internal Memory (640 KB)
2A0000h-3FFFFFh	Reserved (1 MB, 384 KB)
400000h-5FFFFFh	External Memory (2 MB)
600000h-7FFFFFh	Reserved (2 MB)

8 MB Memory Footprint

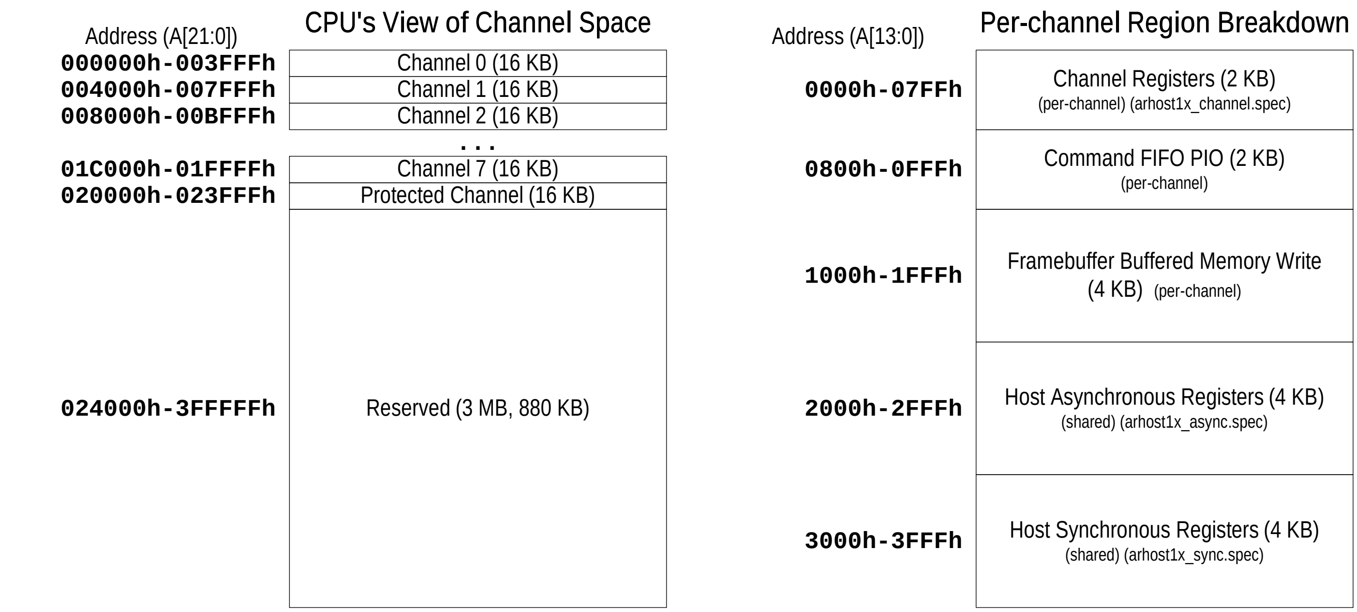
GoForce 5500's Point-of-View

Address (A[25:0])	
0000000h-0003FFFh	Channel 0 (16 KB)
0004000h-0007FFFh	Channel 1 (16 KB)
0008000h-000BFFFh	Channel 2 (16 KB)
	...
001C000h-001FFFFh	Channel 7 (16 KB)
0020000h-0023FFFh	Protected Channel (16 KB)
0024000h-03FFFFFh	Reserved (3 MB, 880 KB)
0400000h-049FFFFh	Internal Memory (640 KB)
04A0000h-1FFFFFFh	Reserved (27 MB, 384 KB)
2000000h-21FFFFFh	External Memory (2 MB)
2200000h-3FFFFFFh	Reserved (30 MB)

64 MB Memory Footprint

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.