Glamo 3362

The Multimedia co-Processor for Mobile

Preliminary

Ver. 0.95

May 24th, 2006

Content

1 Product Overview

1.1 Introduction

As the speed of handheld devices increases with breathtaking pace, the mobile phone has become a vital necessity in our lives, improving and enhancing communication, while also providing a digital camera, game and video recorder. The Glamo 3362 is SMedia's latest mobile multimedia product that gives consumers a dizzying array of options to experience the digital world in all its glory. The Glamo 3362 supports today's most cutting edge multimedia functions including a 2D/3D graphic accelerator, MPEG-4 codec, JPEG codec and LCD controller. With SMedia's unique power management technology, battery life is effectively extended when running 2D/3D games and applications, and the digital camera and video recorder, making the Glamo 3362 chipset ideal choice for those demanding users who like to burn the midnight oil.

1.2 Overview

1.2.1 The High Performance Mobile GPU

The 3D hardware engine in Glamo 3362 is designed for the ultimate in mobile gaming performance, for users who need blistering gaming performance on the bus, train, beach, or anywhere else boredom strikes. Designed on the OpenGL ES 1.1, its performance reaches a smoking 1.54M triangles per second through SMedia's optimizing Transform and Lighting engine. With the capability of 8 lights simultaneously, 3 different lighting types and multiple textures in one cycle, the Glamo 3362 will give users a mind-glowing immersed 3D gaming experience that will make the most die hard gamers weep with joy. The 2D engine features a 50M filling rate per second and full functionality dealing with line, word and picture, making an unprecedented LCD monitor display, ideal for showing off embarrassing, shocking and hilarious photos of family and friends!

1.2.2 Powerful Image Processor

The Glamo 3362 boasts a 5M pixels digital photo resolution image processor. The MPEG-4 codec enables 30 frames per second in CIF resolution, providing an incredibly fluid and life-like image in the phone display. In Video Conference mode, Glamo 3362 can handle two differential images showing on one panel simultaneously. In addition, Glamo 3362 is able to support CIF resolution in video recording, allowing users to see each other with crystal clarity. Of course, the Glamo 3362 is fully compatible with H.263.

The Glamo 3362 supports CSTN and TFT LCD panels, allowing customer to choose the best solution for their particular market. As well, the display also can rotate 90/180/270 mirror mode and dual display mode, depending on user's needs.

1.2.3 Powerful MPEG-4 and JPEG Engine

The MPEG-4 engine in the Glamo 3362 gives users the ability to record/play video at any time, anywhere. Users can record a video clip in H.263 format and then transmit to their friends by MMS, or play received video clip. The MPEG-4 format can achieve higher compression ratio than H.263, so users can use their mobile phone as a digital video recorder to record up to VGA size high quality video as long as they want, the only limitation is the storage space. The advanced rate-control algorithm and the intelligent mode decision algorithm can incredibly increase the visual quality of video while remaining the high compression ratio. The ultimate application of the Glamo 3362 MPEG-4 encoder and decoder engine is to support videophone, gives users and their friends to communicate with not only voice, but also video.

Using the JPEG encoder engine built on Glamo 3362, users can capture images with size over 5M pixels, compress to different quality according to their desire, and then store in the MMC/SD memory card or transmit to friends. With the ISP engine, the JPEG decoder can decode the JPEG file and then scale to any sizes on the fly.

1.2.4 On-chip Frame Buffer and Flexible Clock Scheme

The Glamo 3362 has built-in 16M bits on-chip frame buffer. It is useful for 2D and 3D application. It can support resolution up to 640x480 16bpp for double or triple buffers.

The Glamo 3362 has build-in a clock synthesizer to generate all internal clocks. The clock synthesizer can generate wide range of programmable frequencies. It can provide 1MHz to 90MHz flexible working frequency. The clock synthesizer accepts 32 KHz or 13 MHz reference clock input, which depends on the trapping value of CFG3. And system can even stop the reference clock after the PLL locked the target frequency and phase for power saving.

1.2.5 Optimizes Power Management

Because users often blame battery life of their mobile phones excessively during extreme gaming sessions and heated phone conversations, the Glamo 3362 adds many extra power-saving methods to extend battery life. In power-saving mode, the partial display extends battery life. Compared with software solutions, the Glamo 3362 hardware solution can reduce power consumption by at least 85% while still operating multimedia functions such as watching MPEG-4 videos, photo previewing, and playing 3D/2D games.

1.2.6 Rich Functionality in the Multimedia and Mobile Field

In order to satisfy various customers' demands, the Glamo 3362 can support all types of standard hardware and software specifications in the mobile phone industry. The Glamo 3362 is compatible with products such as CMOS, CCD, LCD, baseband processor or other components, and is well compatible with all products. SMedia not only satisfies all anticipated market demands, but also establishes precedents in mobile phone standards that will set the bar for mobile technologies higher, driving the industry to new heights of technological excellence.

1.3 Feature List

The following are the main features of the Glamo 3362:

Host Interface

- Supports 16 bits SRAM-like with variable latency host interface
- Supports addressing space up to 16 MB
- Supports four types of bus protocol
  - Type 1 direct addressing mode
  - Type 2 direct addressing mode
  - Type 3 iBurst (*Note 1) mode
  - Type 4 indirect addressing mode
- Supports LCD by-pass mode
- Memory read prefetch
- Supports interrupt
- Support burst mode
- Support frequency up to 75 MHz in burst mode

Display Interface

- 640x480 16/18 bpp (RGB565/RGB666)
- Supports 6/9/16/18 bits RGB I/F and 8/9/16/18 bits CPU I/F
- Supports C-STN with frame buffer
- HW 90°/180°/270° rotate and mirror
- Supports TFT or TFD (with or without frame-buffer)
- Supports dual panels
- Supports three channel gamma correction
- Supports hardware cursor
- Supports partial display
- Supports by-pass mode
- Built-in power saving mode

- Support TV encoder interface

High Performance 2D Accelerator

- Fully compliant with the J2ME MIDP 2.0 2D requirement (Support J2ME/JSR-226 special rotation)
- Built-in an 1T pipelined 16bit BitBlt graphics engine
  - ROP3, rectangle fill, font expansion, line-drawing
  - Transparent BitBlt with source destination keys
  - Alpha blended BitBlt, clipping, stretch and shrink
  - Supports mirror/flip mode
  - Alpha Blending: Premultiply
  - AA text
  - Gradient fill
  - NTLine
  - Resolution: Max. 640x480
  - Supports HW 90°/180°/270° rotate
- Max. fill rate: 50 M pixels/sec.

High Performance 3D Accelerator

- Fully compliant with JSR-000184 mobile 3D graphics API for J2ME™
- Fully compliant with OpenGL ES v1.0 and 1.1
- Mobile D3D supported
- Built-in pipelined 3D primitive engine
  - Supports Geometry, lighting, clipping and texture transform
  - 8 active lights (point, directional and spot light) with spectacular and fog
  - Supports Back face culling and texture transform
  - Supports two side lighting
  - Supports linear, EXP and EXP2 vertex fog
  - Supports triangle, line and point primitive type
  - Supports primitive list, strip and fan input
  - Supports multi-stream for VBO
  - Supports user clipping plane

- Built-in pipelined 3D setup engine
  - Supports float-point
  - Supports texture wrap correction
  - Supports polygon offset
- Built-in pipelined 3D graphics engine
  - Supports multiple texture up to two textures
  - Supports per pixel perspective corrected texture mapping
  - Supports MIP structure, MIPMAPLOD bias
  - Supports all of the filtering method (point, linear, bi-linear, NMN, NML, LMN, LML filtering)
  - Texture transparency, blending, wrapping, mirroring
  - Supports palette texture and 8/16 bits ARGB, AL texture
  - Supports rectangle texture
  - Supports non power two texture size
  - Supports Dot Product 3 Bump Mapping
  - Supports texture size up to 256x256
  - Supports flat and Gouraud shading
  - 16 bits Z-buffer, Z test and Z bias
  - 8 bits stencil buffer and stencil test
  - Supports vertex fog and pixel fog (linear, EXP and EXP2)
  - Supports alpha blending and spectacular effects
  - Supports point and line width
  - Supports dithering and ROP
- Supports Max. fill rate: 50 M pixels/sec
- Supports polygon rate: 1.54 M polygons/sec

MPEG–4 Engine

- Fully compliant with ISO/IEC 14496-2 (MPEG-4) video simple profile level 0, 1, 2, 3
- Fully compliant with ITU-T recommendation H.263 profile 0 level 10, 20, 30, 40
- H.263/MPEG-4 decode and encode
- Supports up to CIF 30 fps, 2 Mbps; VGA decoding up to 20fps (*Note 2)

- Supports constant bit rate (CBR) and variable bit rate (VBR)
- Advanced rate-control algorithm with HVS (human visual system) support
- Intelligent mode decision algorithm to increase coding quality
- Deblocking
- Downloadable Huffman table
- Decode non-interleave scan

JPEG Engine

- Fully compliant with Baseline JPEG standard ISO/IEC 10918
- JPEG decode and encode
- Low pass filter for noise reduction
- Supports encode and decode JPEG image larger than 5M pixel

Video Interface

- Supports up to 5M pixels CMOS/CCD sensor input (*Note 3)
- Supports Bayer pattern and YUV422 input format
- 10-bit interface
- Supports 90°/180°/270° rotate & mirror & flip (*Note 4)

Image Signal Processing

- Exposure Control
  - Default modes: Auto, Spot, Center-Weighted, Scenery, Portrait, and Night-Shoot
  - User defined mode: Programmable by the user
  - Configurable exposure table by varying Shutter Speed, Aperture, and Gain
- White balance control
  - Default Modes: Auto, Indoor, Fluorescent, Outdoor
  - User Define Mode: Programmable by the User
  - Configurable white balance table
- Focus Control
  - Default modes: Auto

- User define mode: Programmable by User
- Configurable focus table

- Authentic 16x continuous digital zoom
- Night-shoot mode enhancement
- Image effects
  - Color filter effects: Red, Green, Blue, Cyan, Magenta, Yellow
  - Monochrome
  - Negative
  - Emboss
  - Solarization
- Lens shading correction
  - Configurable shading correction table
  - Four channel
- Bad-pixel removal
  - Configurable bad-pixel position table
- Gamma/Tone-Curve correction
  - Configurable Gamma/Tone-Curve table
- Color enhancement
  - Configurable color correction matrix
  - Supports hue adjustment
  - Supports saturation adjustment
  - Supports brightness adjustment
  - Supports contrast adjustment
- 2-D edge-enhancement
- Image and video scaling engine for scaling up and down
  - 4-Tap scaling filter for both preview and JPEG/MPEG
  - Proprietary 1-input-2-output scaling engine
  - Configurable filter coefficients
- Supports picture overlay
- Dither engine

MMC/SD Interface

- Fully compliant with MMCA v3.3
- Compliant with low-voltage support and 4 bits data of MMCA v4.0
- Compliant with SD

PLL Interface

- Supports two wide range clock frequency synthesizers, one from 1 MHz to 60 MHz and the other from 1MHz to 90 MHz.
- Accepts 32 KHz or 13 MHz optional reference clock input
- Supports power down mode to turn off PLL for power saving

Package

- 160-Ball 8 mm x 8 mm x 1.3 mm LFBGA.

Note:

1. iBurst proprietary interface support Infineon S-GOLD2 series baseband processor.
2. VGA size MPEG4 CODEC can only be implemented on those models stacked with at least 4MB memory.
3. 16 Mb is set to be the default local memory size in Glamo 3362. Be aware of that if you want higher resolution of either image sensor or display device, you will need higher density of memory size.

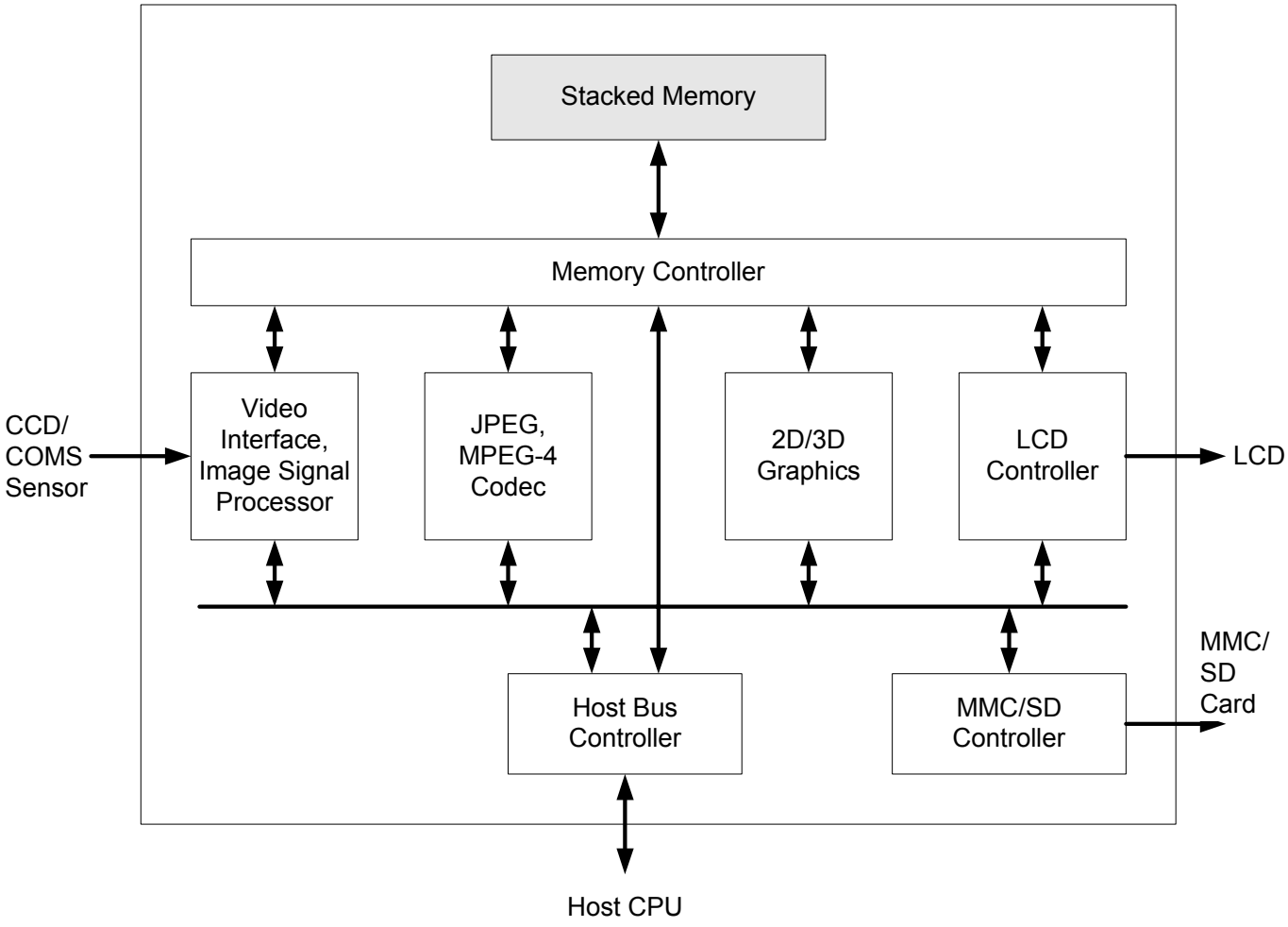
※ Maximum supporting resolution

	Bayer	***YUV***
2MB stacked	2M	<1M
4MB stacked	4M	2M
8MB stacked	5M	4M

4. In JPEG or MPEG mode, Glamo can support 180° & mirror & flip only.

1.4 Glamo 3362 Block Diagram

Figure 1.4-1 Glamo 3362 Block Diagram



2 Signal Descriptions

2.1 Introduction

This chapter describes and lists all of the Glamo 3362 pins. The sign (#) at the end of a signal indicates that the polarity of the signal is active-low. The pin types are classified as:

- P = power pin
- I = input pin
- O = output pin
- I/O = input/output pin

2.1.1 Internal Pull-Up Pins

The following pins are internally pulled up in the Glamo 3362:

- MMCCMD
- MMCDAT
- MMCDAT1
- MMCDAT2
- MMCDAT3

2.1.2 Default Value

The default value of pin is classified as:

- 'I' mark indicates this is input pin and the default value is depends on the input value.
- 'O(0)' mark indicates this is output pin and the default value is L.
- 'O(1)' mark indicates this is output pin and the default value is H.
- 'O(U)' mark indicates this is output pin and the default value is depends on the translated data
- 'Highz' mark indicates the pin with default no driving
- 'Oper' mark indicates this is IO pin and its' I/O can be defined by system, the default value is depends on the system setting.

2.2 Host Interface

Host interface contains total 48 pins. Detail pin information is presented in Table 2.2-1.

Table 2.2-1 Host Interface Pin Descriptions

Ball Location	Pin Name	Type	Default (Reset)	Default (Sleep)	Power	Description
F3	HA1	I	-	-	HOVDD	Host interface address bus and allow up to 16MB addressable space.
F4	HA2	I	-	-	HOVDD	
F5	HA3	I	-	-	HOVDD	
G3	HA4	I	-	-	HOVDD	
G2	HA5	I	-	-	HOVDD	
H5	HA6	I	-	-	HOVDD	
H4	HA7	I	-	-	HOVDD	
H3	HA8	I	-	-	HOVDD	
H2	HA9	I	-	-	HOVDD	
H1	HA10	I	-	-	HOVDD	
J5	HA11	I	-	-	HOVDD	
J4	HA12	I	-	-	HOVDD	
J3	HA13	I	-	-	HOVDD	
J2	HA14	I	-	-	HOVDD	
J1	HA15	I	-	-	HOVDD	
K5	HA16	I	-	-	HOVDD	
K4	HA17	I	-	-	HOVDD	
K3	HA18	I	-	-	HOVDD	
K1	HA19	I	-	-	HOVDD	
L4	HA20	I/O	-	I/Oper	HOVDD	This pin can be programmed as GPIO0 when using type 4 indirect addressing mode.

L3	HA21	I/O	-	I/Oper	HOVDD	This pin can be programmed as GPIO1 when using type 4 indirect addressing mode.
L2	HA22	I/O	-	I/Oper	HOVDD	This pin can be programmed as GPIO2 when using type 4 indirect addressing mode.
L1	HA23	I/O	-	I/Oper	HOVDD	This pin can be programmed as GPIO3 when using type 4 indirect addressing mode.
B3	HD0	I/O	-	-	HOVDD	Host interface bi-directional data bus.
C3	HD1	I/O	-	-	HOVDD	
A2	HD2	I/O	-	-	HOVDD	
B2	HD3	I/O	-	-	HOVDD	
B1	HD4	I/O	-	-	HOVDD	
C1	HD5	I/O	-	-	HOVDD	
C2	HD6	I/O	-	-	HOVDD	
D1	HD7	I/O	-	-	HOVDD	
D3	HD8	I/O	-	-	HOVDD	
D4	HD9	I/O	-	-	HOVDD	
E1	HD10	I/O	-	-	HOVDD	
E2	HD11	I/O	-	-	HOVDD	
E3	HD12	I/O	-	-	HOVDD	
E4	HD13	I/O	-	-	HOVDD	
E5	HD14	I/O	-	-	HOVDD	
F1	HD15	I/O	-	-	HOVDD	
B5	HCS#	I	-	-	HOVDD	Chip select input
C5	HRD#	I	-	-	HOVDD	Read command input
D5	HWR#	I	-	-	HOVDD	Write command input
A4	HWAIT#	O	Highz	Highz	HOVDD	Wait cycle insertion. During a data transfer, HWAIT# is driven active to force the baseband to insert wait state. It is driven inactive to indicate the completion of a data transfer.
C4	HLB#	I	-	-	HOVDD	Byte enable input for lower data byte (HD[7:0])

A3	HUB#	I	-	-	HOVDD	Byte enable input for higher data byte (HD[15:8])
A5	INT#	O	Highz	Highz	HOVDD	Interrupt output
F2	HCLK	I	-	-	HOVDD	Host clock
B4	HADV#	I/O	-	I/Oper	HOVDD	Address/Data valid bit This pin can be programmed as GPIO19.

2.3 LCD Interface

There are two modes supported by LCD interface:

- Mode 0: CPU interface
- Mode 1: RGB interface

LCD interface contains total 28 pins. Detail pin information is presented in Table 2.3-1.

Table 2.3-1 LCD Interface Pin Descriptions

Ball Location	Pin Name	Type	Default (Reset)	Default (Sleep)	Power	Description
M8	LCS0#	I/O	O(1)	O(1)	LOVDD	Mode 0: Chip select output for LCD1 Mode 1: Serial interface enable This pin can be programmed as GPIO4.
H10	LCS1#	I/O	O(1)	O(1)	LOVDD	Mode 0: Chip select output for LCD2 Mode 1: No used This pin can be programmed as GPIO5.
N8	LDCLK	I/O	O(0)	O(0)	LOVDD	Mode 0: No used Mode 1: Clock output This pin can be programmed as GPIO6.
K9	LDE#	I/O	O(1)	O(1)	LOVDD	Mode 0: Read enable for 80CPU Read or Write enable for 68 CPU Mode 1: Data enable output This pin can be programmed as GPIO7.
N10	LD0	O	O(U)	O(U)	LOVDD	B[0] data output

L11	LD1	O	O(U)	O(U)	LOVDD	B[1] data output
M11	LD2	O	O(U)	O(U)	LOVDD	B[2] data output
N11	LD3	O	O(U)	O(U)	LOVDD	B[3] data output
M12	LD4	O	O(U)	O(U)	LOVDD	B[4] data output
N12	LD5	O	O(U)	O(U)	LOVDD	B[5] data output
M13	LD6	O	O(U)	O(U)	LOVDD	G[0] data output
L13	LD7	O	O(U)	O(U)	LOVDD	G[1] data output
L12	LD8	O	O(U)	O(U)	LOVDD	G[2] data output
K13	LD9	O	O(U)	O(U)	LOVDD	G[3] data output
K12	LD10	O	O(U)	O(U)	LOVDD	G[4] data output
K11	LD11	O	O(U)	O(U)	LOVDD	G[5] data output
J13	LD12	O	O(U)	O(U)	LOVDD	R[0] data output
J12	LD13	O	O(U)	O(U)	LOVDD	R[1] data output
J11	LD14	O	O(U)	O(U)	LOVDD	R[2] data output
J10	LD15	O	O(U)	O(U)	LOVDD	R[3] data output
J9	LD16	I/O	O(U)	O(U)	LOVDD	R[4] data output This pin can be programmed as GPIO8.
H11	LD17	I/O	O(U)	O(U)	LOVDD	R[5] data output This pin can be programmed as GPIO9.
L9	LHSYNC	O	O(1)	O(1)	LOVDD	Mode 0: Control register & memory space select Mode 1: Horizontal sync output
M9	LVSYNC	O	O(1)	O(1)	LOVDD	Mode 0: Data write output Mode 1: Vertical sync output
N9	LSCK	I/O	O(1)	O(1)	LOVDD	Mode 0: No used Mode 1: Serial interface clock This pin can be programmed as GPIO10.
L10	LSDA	I/O	O(0)	O(0)	LOVDD	Mode 0: No used Mode 1: Serial interface data input/output

						This pin can be programmed as GPIO11.
M10	LSA0	I/O	O(0)	O(0)	LOVDD	Mode 0: No used Mode 1: Serial interface A0 output This pin can be programmed as GPIO12.
H9	TVCLK	I	-	-	LOVDD	Clock input for TV encoder

2.4 Camera Interface

Camera interface contains total 18 pins. Detail pin information is presented in Table 2.3-2.

Table 2.3-2 Camera Interface Pin Descriptions

Ball Location	Pin Name	Type	Default (Reset)	Default (Sleep)	Power	Description
E13	PCLK	I	-	-	COVDD	Pixel clock from sensor
D11	SCLK	O	highz	highz	COVDD	Output clock to sensor
D12	CSEN	O	highz	highz	COVDD	Serial interface enable
D13	CSCL	O	highz	highz	COVDD	Serial interface clock
C12	CSDA	I/O	-	-	COVDD	Serial interface data input/output
E9	CSGPO0	I/O	O(0)	I/Oper	COVDD	Sensor data input D[0] Serial interface general purpose output This pin can be programmed as GPIO13.
D9	CSGPO1	I/O	O(0)	I/Oper	COVDD	Sensor data input D[1] Serial interface general purpose output This pin can be programmed as GPIO14.
C13	CHSYNC	I	-	-	COVDD	Horizontal sync signal
B12	CVSYNC	I	-	-	COVDD	Vertical sync signal
B13	CD0	I	-	-	COVDD	Sensor data input D[9:2]
A12	CD1	I	-	-	COVDD	
B11	CD2	I	-	-	COVDD	

A11	CD3	I	-	-	COVDD	
D10	CD4	I	-	-	COVDD	
C10	CD5	I	-	-	COVDD	
B10	CD6	I	-	-	COVDD	
A10	CD7	I	-	-	COVDD	
C9	FLCTL	I/O	O(0)	O(0)/Oper	COVDD	Flash light control output This pin can be programmed as GPIO15.

2.5 Dedicated GPIO

Dedicated GPIO contains total 16 pins. Detail pin information is presented in Table 2.3-4.

Table 2.3-4 DGPIO Pin Descriptions

Ball Location	Pin Name	Type	Default (Reset)	Default (Sleep)	Power	Description
N3	DGPIO0	I/O	-	Oper	UOVDD	DGPIO 0 in group A
N4	DGPIO1	I/O	-	Oper	UOVDD	DGPIO 1 in group A
L7	DGPIO2	I/O	-	Oper	UOVDD	DGPIO 2 in group A
M7	DGPIO3	I/O	-	Oper	UOVDD	DGPIO 3 in group A
K8	DGPIO4	I/O	-	Oper	LOVDD	DGPIO 4 in LCD group
L8	DGPIO5	I/O	-	Oper	LOVDD	DGPIO 5 in LCD group
F9	DGPIO6	I/O	-	Oper	COVDD	DGPIO 6 in Sensor group
F10	DGPIO7	I/O	-	Oper	COVDD	DGPIO 7 in Sensor group
E10	DGPIO8	I/O	-	Oper	COVDD	DGPIO 8 in Sensor group
E11	DGPIO9	I/O	-	Oper	COVDD	DGPIO 9 in Sensor group
E12	DGPIO10	I/O	-	Oper	COVDD	DGPIO 10 in Sensor group
B9	DGPIO11	I/O	-	Oper	COVDD	DGPIO 11 in Sensor group
A9	DGPIO12	I/O	-	Oper	COVDD	DGPIO 12 in Sensor group
D8	DGPIO13	I/O	-	Oper	COVDD	DGPIO 13 in Sensor group
C8	DGPIO14	I/O	-	Oper	COVDD	DGPIO 14 in Sensor group

B8	DGPIO15	I/O	-	Oper	COVDD	DGPIO 15 in Sensor group

2.6 Miscellaneous

It contains total 13 miscellaneous pins. Detail pin information is presented in Table 2.3-5.

Table 2.3-5 Miscellaneous Pin Descriptions

Ball Location	Pin Name	Type	Default (Reset)	Default (Sleep)	Power	Description
A6	OSCI	I	-	-	AOVDD	32KHz reference clock for PLL or digital clock input
D6	RST#	I	-	-	HOVDD	Chip asynchronous reset signal
C6	ENTEST	I	-	-	HOVDD	Test mode enables. This pin should be connected with GND in normal operation.
H12	MMCCLK	O	O(0)	O(0)	TOVDD	Multimedia card interface CLK signal output
G12	MMCCMD	I/O	O(1)	O(1)	TOVDD	Multimedia card interface CMD signal
G11	MMCDAT	I/O	-	-	TOVDD	Multimedia card interface DAT signal
F11	MMCDAT1	I/O	-	Oper	TOVDD	Multimedia card interface DAT signal This pin can be programmed as GPIO16.
F12	MMCDAT2	I/O	-	Oper	TOVDD	Multimedia card interface DAT signal This pin can be programmed as GPIO17.
F13	MMCDAT3	I/O	-	Oper	TOVDD	Multimedia card interface DAT signal This pin can be programmed as GPIO18.
M3	CFG0	I	-	-	HOVDD	Configuration trap [3:0]
M2	CFG1	I	-	-	HOVDD	
M1	CFG2	I	-	-	HOVDD	
N2	CFG3	I	-	-	HOVDD	

2.7 Power

It contains total 30 power pins. Detail pin information is presented in Table 2.3-6.

Table 2.3-6 Power Pin Descriptions

Ball Location	Pin Name	Type	Power	Description
A1, A7, A13, N1, N7, N13	MOVDD	P	-	1.8 V power for Memory interface
D2, K2	HOVDD	P	-	1.8/3.3 V IO power for Host interface
C11	COVDD	P	-	1.8/3.3 V IO power for Camera interface
K10	LOVDD	P	-	1.8/3.3 V IO power for LCD interface
H13	TOVDD	P	-	1.8/3.3 V IO power for MMC interface
D7, G4, G10, K7	IVDD	P	-	1.8 V core power
B7	AIVDD1	P	-	1.8 V power for PLL1
C7	AIVDD2	P	-	1.8 V power for PLL2
B6	AOVDD	P	-	1.8/3.3 V IO power for OSCI
L5	UOVDD	P	-	1.8/3.3 V IO power for GPIO group A
E6, E7, E8, G1, G5, G9, G13, J6, J7, J8	VSS	P	-	Ground for core
A8	AIVSS	P	-	Analog ground for PLL

2.8 Pin Assignment

Figure 2.4-1 Top-Down View of Glamo 3362

	1	2	3	4	5	6	7	8	9	10	11	12	13
A	MOVDD	HD2	HUB#	HWAIT#	INT#	OSCI	MOVDD	AIVSS	DGPIO12	CD7	CD3	CD1	MOVDD
B	HD4	HD3	HD0	HADV#	HCS#	AOVDD	AIVDD1	DGPIO15	DGPIO11	CD6	CD2	CVSYNC	CD0
C	HD5	HD6	HD1	HLB#	HRD#	ENTEST	AIVDD2	DGPIO14	FLCTL	CD5	COVDD	CSDA	CHSYNC
D	HD7	HOVDD	HD8	HD9	HWR#	RST#	IVDD	DGPIO13	CSGPO1	CD4	SCLK	CSE#	CSCL
E	HD10	HD11	HD12	HD13	HD14	VSS	VSS	VSS	CSGPO0	DGPIO8	DGPIO9	DGPIO10	P C L K
F	HD15	HCLK	HA1	HA2	HA3				DGPIO6	DGPIO7	MMCDAT1	MMCDAT2	MMCDAT3
G	VSS	HA5	HA4	IVDD	VSS				VSS	IVDD	MMCDAT	MMCCMD	VSS
H	HA10	HA9	HA8	HA7	HA6				TVCLK	LCS1#	LD17	MMCLK	TOVDD
J	HA15	HA14	HA13	HA12	HA11	VSS	VSS	VSS	LD16	LD15	LD14	LD13	LD12
K	HA19	HOVDD	HA18	HA17	HA16	ZSDOUT	IVDD	DGPIO4	LDE#	LOVDD	LD11	LD10	LD9
L	HA23	HA22	HA21	HA20	UOVDD	ZWS	DGPIO2	DGPIO5	LHSYNC	LSDA	LD1	LD8	LD7
M	CFG2	CFG1	CFG0	ACLK	ASDIN	ZCLK	DGPIO3	LCS0#	LVSYNC	LSA0	LD2	LD4	LD6
N	MOVDD	CFG3	DGPIO0	DGPIO1	AWS	AMCLK	MOVDD	LDCLK	LSCK	LD0	LD3	LD5	MOVDD

Total ball: 160 Signal ball: 130 Power ball: 30

3 Interface Descriptions

3.1 Host Interface

3.1.1 Introduction

Mobile phone baseband processors use asynchronous SRAM-like interface to communicate outside device. The outside devices include SRAM, FLASH and the multimedia chip. Most of the interface only support slave mode, that is, CPU driven mode. And some of the interface may support wait state insertion. Glamo 3362 provides an asynchronous SRAM-like interface to communicate with most baseband processors. For getting better data transfer rate, it also supports a proprietary synchronous interface - iBurst.

There are eight types of bus protocol supported by Glamo 3362 host interface. By trapping CFG[1:0] to decide the bus protocol of the host interface. Refer to Table 3.1.1-1.

Table 3.1.1-1 Host Bus Type Table

CFG1	CFG0	HADVN	HA10	HCLK	Host Bus Type Description
0	0	0	A10	0	Direct addressing 16-bit 80 type 1
0	0	1	A10	0	Direct addressing 16-bit 80 type 2
0	0	N/A	A10	1	Direct addressing 16-bit 68 type
1	0	ADV	0	0	Indirect addressing 16-bit 80 type
1	0	ADV	0	1	Indirect addressing 16-bit 68 type
1	0	ADV	1	0	Indirect addressing 8-bit 80 type
1	0	ADV	1	1	Indirect addressing 8-bit 68 type
1	1	ADVN	N/A	Interface Clock	Synchronous iBurst type

3.1.2 Direct Addressing Mode 16 bits 80 type 1 Interface

Figure 3.1.2-1 Direct Addressing Mode 16 bits 80 Type 1 Interface Implementation

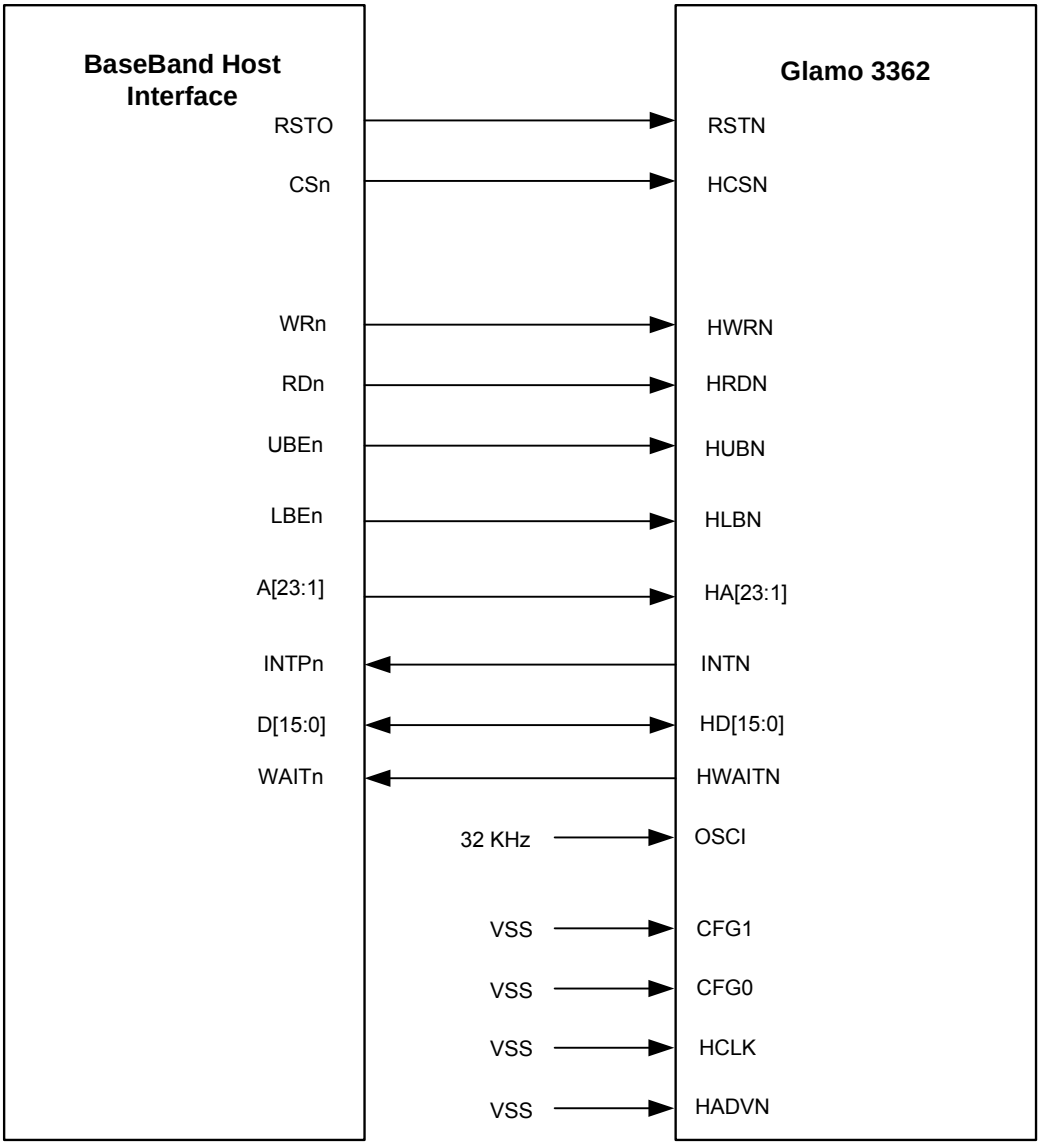


Figure 3.1.2-2 16 bits 80 Type 1 CPU Write Transaction

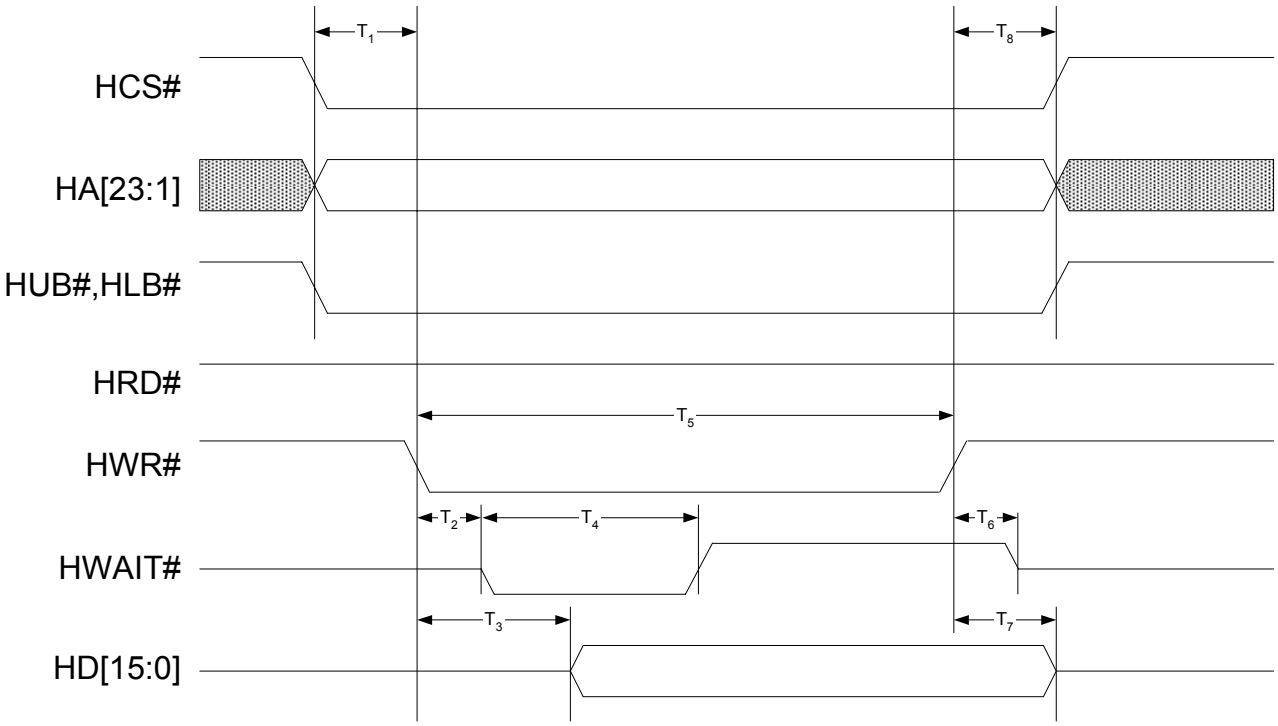


Figure 3.1.2-3 16 bits 80 Type 1 CPU Read Transaction

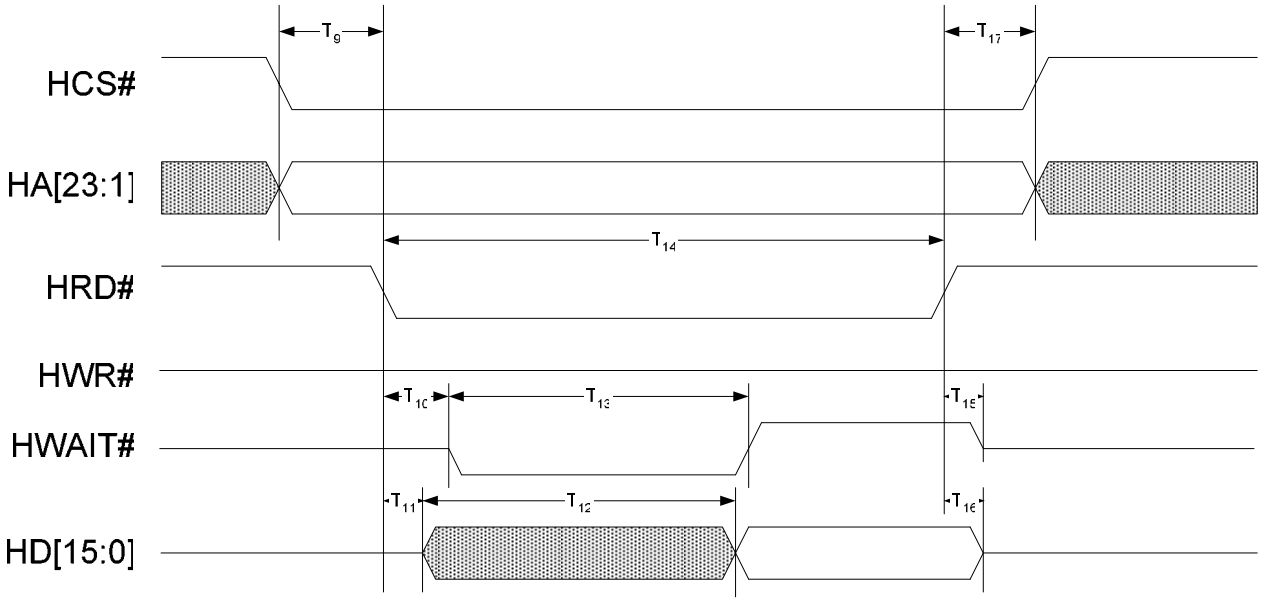


Table 3.1.2-1 Direct Addressing Mode 16 bits 80 Type 1 Interface Timing Table

Symbol	Parameter	Min	Typ	Max	Unit
T_1	Chip select, address and byte enable setup time from write falling edge	5			ns
T_2	Write falling edge to wait driven low	5.0	7.0	11	ns
T_3	Data delay from write falling edge			30	ns
T_4	Wait period during write cycle	(1)Note	(1)Note	(1)Note	
T_5	Write active period	30			ns
T_6	Write rising edge to wait high impedance	3	4	6.5	ns
T_7	Data hold time from write write rising edge	5			ns
T_8	Chip select, address and byte enable hold time from write rising edge	5			ns
T_9	Chip select, address and byte enable setup time from read falling edge	5			ns
T_{10}	Read falling edge to wait driven low	4.5	6.5	10.5	ns
T_{11}	Read falling edge to data driven	3.0	4.0	6.0	ns
T_{12}	Valid data period	(1)Note	(1)Note	(1)Note	-
T_{13}	Wait period during read cycle	(1)Note	(1)Note	(1)Note	-
T_{14}	Read active period	30			ns
T_{15}	Read rising edge to wait high impedance	3.0	3.5	6.5	ns
T_{16}	Data hold time from read rising edge	10.5	11.5	14.0	ns
T_{17}	Chip select, address and byte enable hold time from read rising edge	5			ns

Note:

1. See Table 3.1.2-2.

Table 3.1.2-2 Direct Addressing Mode 16 bits 80 Type 1 Interface Wait Period Table

Description	Min	Typ(2)	Max	Unit
Single Write to Registers		0	3(3)	T(1)
Single Read from Registers except ISP, JPEG, MPEG and 3D Registers		2	5(4)	T
Single Read from ISP Registers		3		T
Single Read from JPEG Registers		5		T
Single Read from MPEG Registers		5	7(5)	T
Single Read from 3D Registers		7		T
Consecutive(6) Write to Registers Penalty		1		T
Consecutive Read from Registers Penalty		1		T
Write to Registers after Read Penalty		1		T
Read from Registers after Write Penalty		3		T

Note:

1. T represents inner clock period.
2. Due to asynchronous timing, all above real timing should add the time from asynchronous asserted to inner clock latch edge.
3. Refer to 0x0200h[5:4] register in programming guide.
4. Refer to 0x0200h[7:6] register in programming guide.
5. Time for the first reading from MPEG registers.
6. It means read/write at the same time in 1T period.

3.1.3 Direct Addressing Mode 16 bits 80 type 2 Interface

Figure 3.1.3-1 Direct Addressing Mode 16 bits 80 Type 2 Interface Implementation

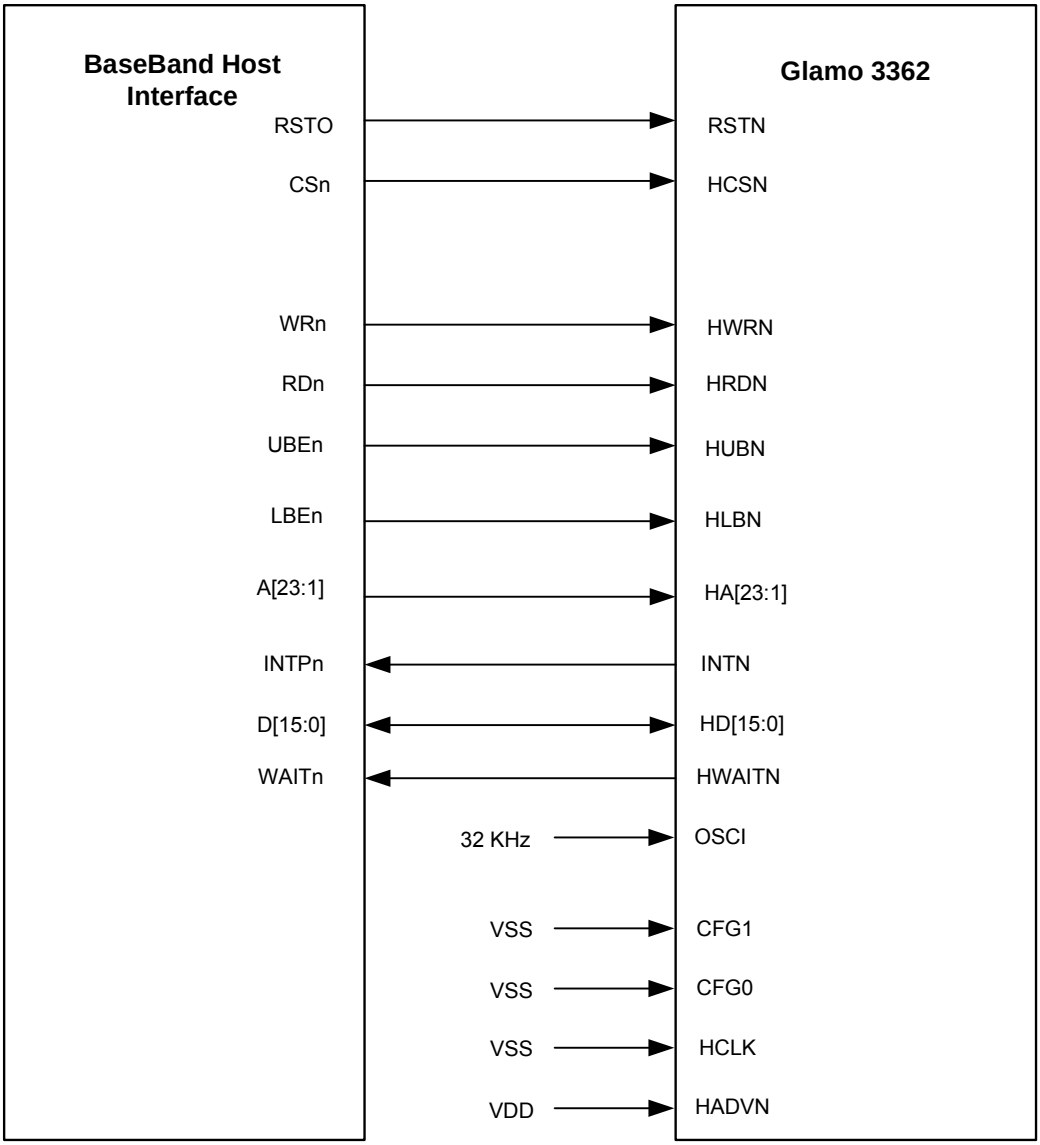


Figure 3.1.3-2 16 bits 80 Type 2 CPU Write Transaction

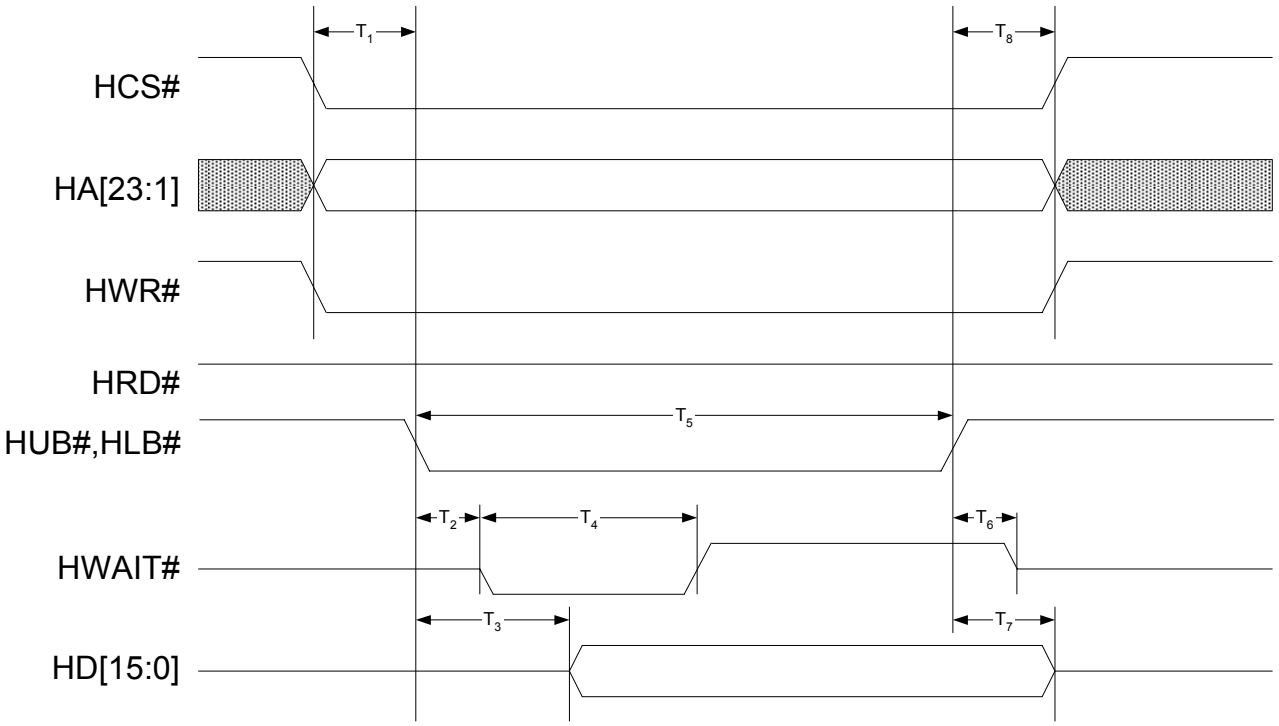


Figure 3.1.3-3 16 bits 80 Type 2 CPU Read Transaction

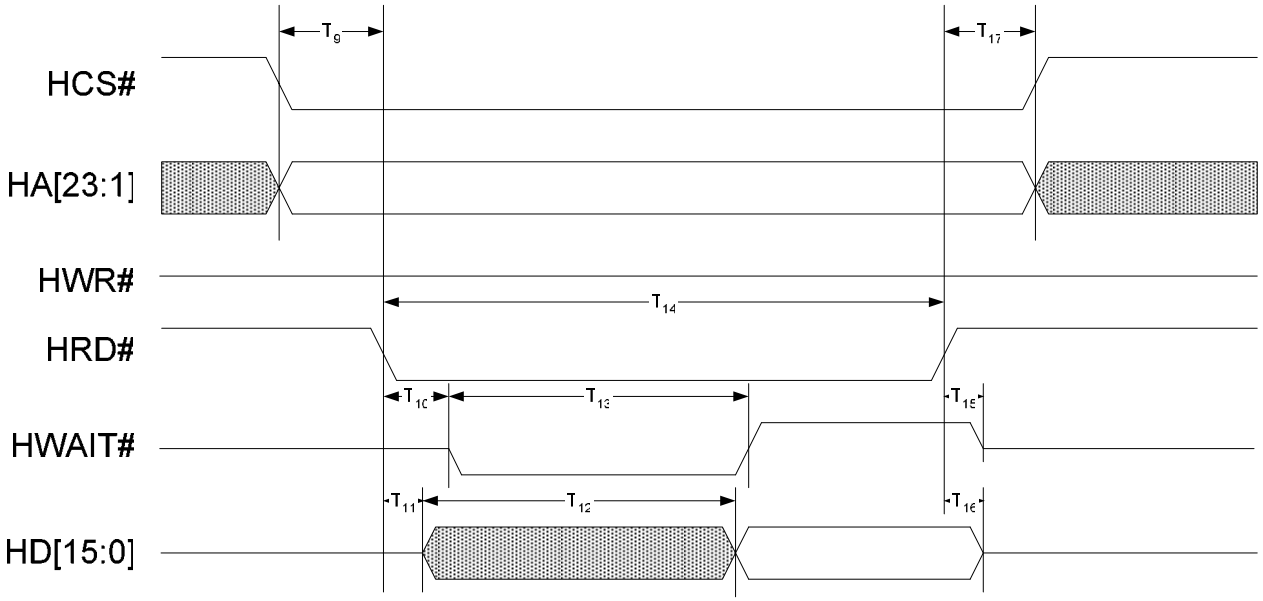


Table 3.1.3-1 Direct Addressing Mode 16 bits 80 Type 2 Interface Timing Table

Symbol	Parameter	Min	Typ	Max	Unit
T_1	Chip select, address and byte enable setup time from write falling edge	5			ns
T_2	Write falling edge to wait driven low	5.0	7.0	11.0	ns
T_3	Data delay from write falling edge			30	ns
T_4	Wait period during write cycle	(1)Note	(1)Note	(1)Note	
T_5	Write active period	30			ns
T_6	Write rising edge to wait high impedance	3	4.0	6.5	ns
T_7	Data hold time from write write rising edge	5			ns
T_8	Chip select, address and byte enable hold time from write rising edge	5			ns
T_9	Chip select, address and byte enable setup time from read falling edge	5			ns
T_{10}	Read falling edge to wait driven low	4.5	6.5	10.5	ns
T_{11}	Read falling edge to data driven	3.0	4	6.0	ns
T_{12}	Valid data period	(1)Note	(1)Note	(1)Note	-
T_{13}	Wait period during read cycle	(1)Note	(1)Note	(1)Note	-
T_{14}	Read active period	30			ns
T_{15}	Read rising edge to wait high impedance	3.0	3.5	6.5	ns
T_{16}	Data hold time from read rising edge	10.5	11.5	14.0	ns
T_{17}	Chip select, address and byte enable hold time from read rising edge	5			ns

Note:

1. See Table 3.1.3-2.

Table 3.1.3-2 Direct Addressing Mode 16 bits 80 Type 2 Interface Wait Period Table

Description	Min	Typ(2)	Max	Unit
Single Write to Registers		0	3(3)	T(1)
Single Read from Registers except ISP, JPEG, MPEG and 3D Registers		2	5(4)	T
Single Read from ISP Registers		3		T
Single Read from JPEG Registers		5		T
Single Read from MPEG Registers		5	7(5)	T
Single Read from 3D Registers		7		T
Consecutive(6) Write to Registers Penalty		1		T
Consecutive Read from Registers Penalty		1		T
Write to Registers after Read Penalty		1		T
Read from Registers after Write Penalty		3		T

Note:

1. T represents inner clock period.
2. Due to asynchronous timing, all above real timing should add the time from asynchronous asserted to inner clock latch edge.
3. Refer to 0x0200h[5:4] register in programming guide.
4. Refer to 0x0200h[7:6] register in programming guide.
5. Time for the first reading from MPEG registers.
6. It means read/write at the same time in 1T period.

3.1.4 Direct Addressing Mode 16 bits 68 Type Interface

Figure 3.1.4-1 Direct Addressing Mode 16 bits 68 Type Interface Implementation

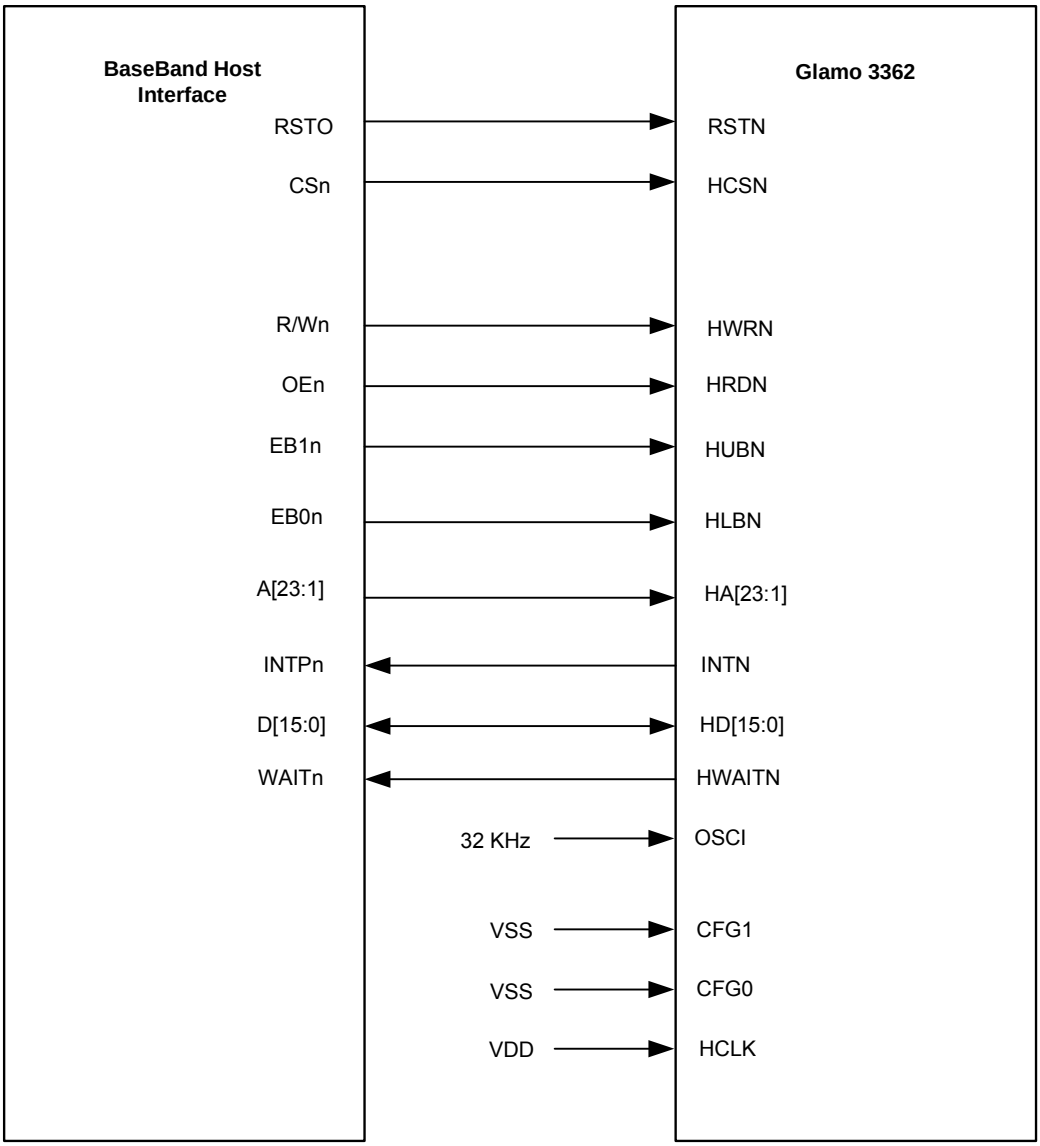


Figure 3.1.4-2 16 bits 68 Type CPU Write Transaction

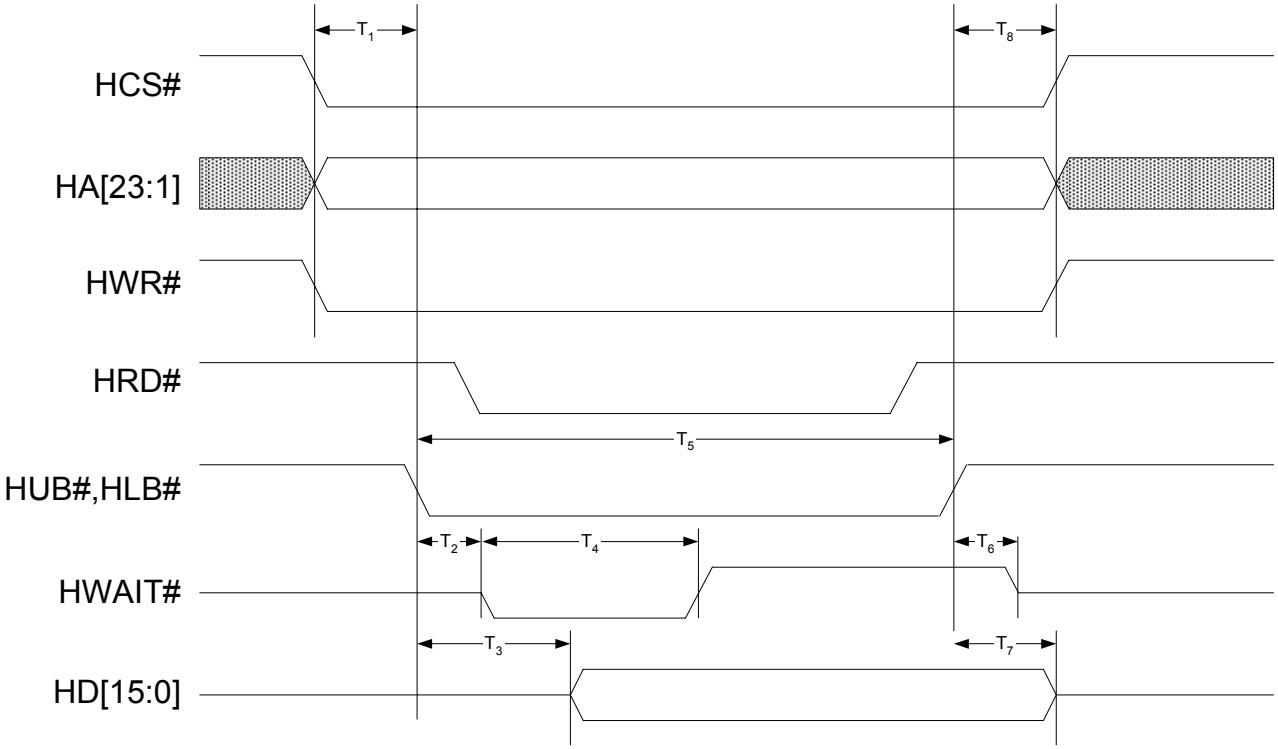


Figure 3.1.4-3 16 bits 68 Type CPU Read Transaction

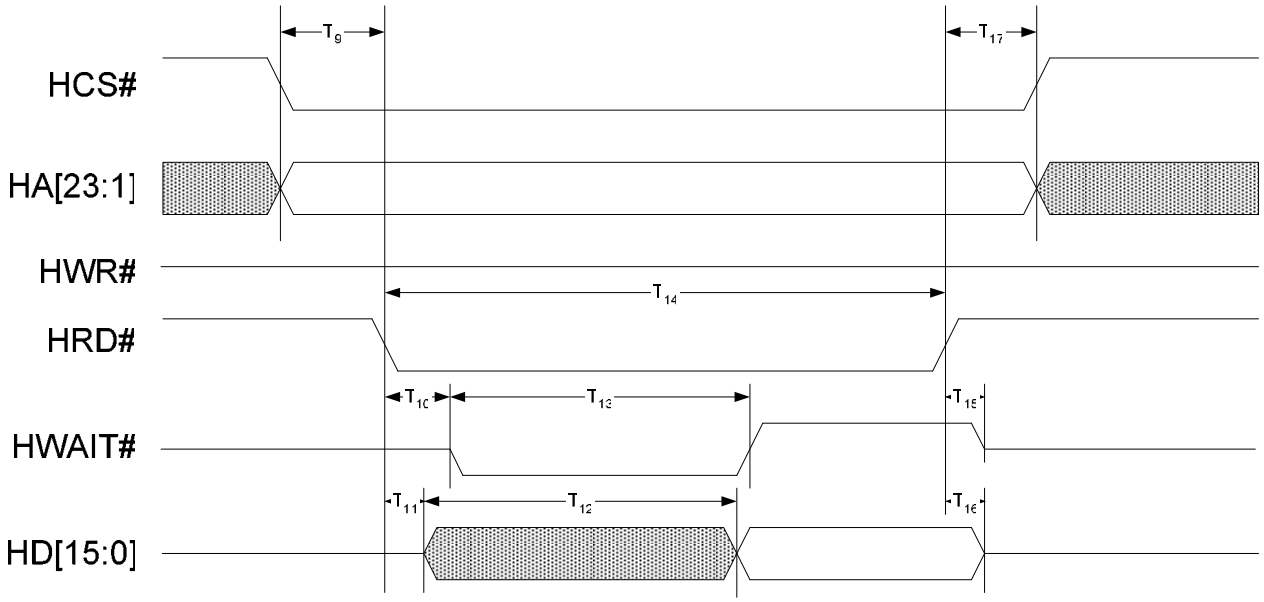


Table 3.1.4-1 Direct Addressing Mode 16 bits 68 Type Interface Timing Table

Symbol	Parameter	Min	Typ	Max	Unit
T_1	Chip select, address and byte enable setup time from write falling edge	5			ns
T_2	Write falling edge to wait driven low	5.0	7.0	11.0	ns
T_3	Data delay from write falling edge			30	ns
T_4	Wait period during write cycle	(1)Note	(1)Note	(1)Note	
T_5	Write active period	30			ns
T_6	Write rising edge to wait high impedance	3.0	4.0	6.5	ns
T_7	Data hold time from write write rising edge	5			ns
T_8	Chip select, address and byte enable hold time from write rising edge	5			ns
T_9	Chip select, address and byte enable setup time from read falling edge	5			ns
T_{10}	Read falling edge to wait driven low	4.5	6.5	10.5	ns
T_{11}	Read falling edge to data driven	3.0	4	6.0	ns
T_{12}	Valid data period	(1)Note	(1)Note	(1)Note	-
T_{13}	Wait period during read cycle	(1)Note	(1)Note	(1)Note	-
T_{14}	Read active period	30			ns
T_{15}	Read rising edge to wait high impedance	3.0	3.5	6.5	ns
T_{16}	Data hold time from read rising edge	10.5	11.5	14.0	ns
T_{17}	Chip select, address and byte enable hold time from read rising edge	5			ns

Note:

1. See Table 3.1.4-2.

Table 3.1.4-2 Direct Addressing Mode 16 bits 68 Type Interface Wait Period Table

Description	Min	Typ(2)	Max	Unit
Single Write to Registers		0	3(3)	T(1)
Single Read from Registers except ISP, JPEG, MPEG and 3D Registers		2	5(4)	T
Single Read from ISP Registers		3		T
Single Read from JPEG Registers		5		T
Single Read from MPEG Registers		5	7(5)	T
Single Read from 3D Registers		7		T
Consecutive(6) Write to Registers Penalty		1		T
Consecutive Read from Registers Penalty		1		T
Write to Registers after Read Penalty		1		T
Read from Registers after Write Penalty		3		T

Note:

1. T represents inner clock period.
2. Due to asynchronous timing, all above real timing should add the time from asynchronous asserted to inner clock latch edge.
3. Refer to 0x0200h[5:4] register in programming guide.
4. Refer to 0x0200h[7:6] register in programming guide.
5. Time for the first reading from MPEG registers.
6. It means read/write at the same time in 1T period.

3.1.5 Indirect Addressing Mode 16 bits 80 Type Interface

Figure 3.1.5-1 Indirect Addressing Mode 16 bits 80 Type Interface Implementation

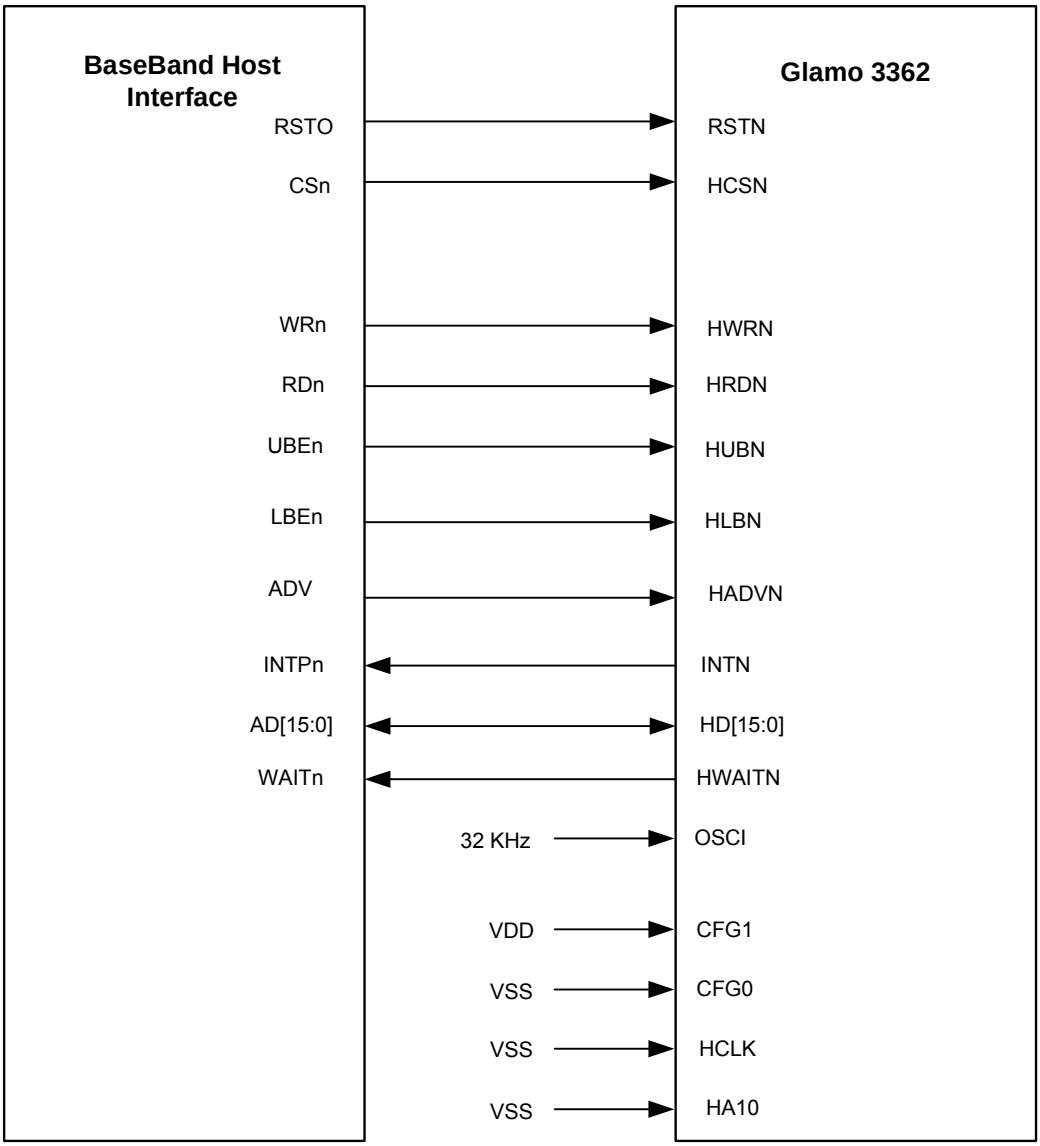


Figure 3.1.5-2 Indirect Addressing Mode 16 bits 80 Type Write Transaction

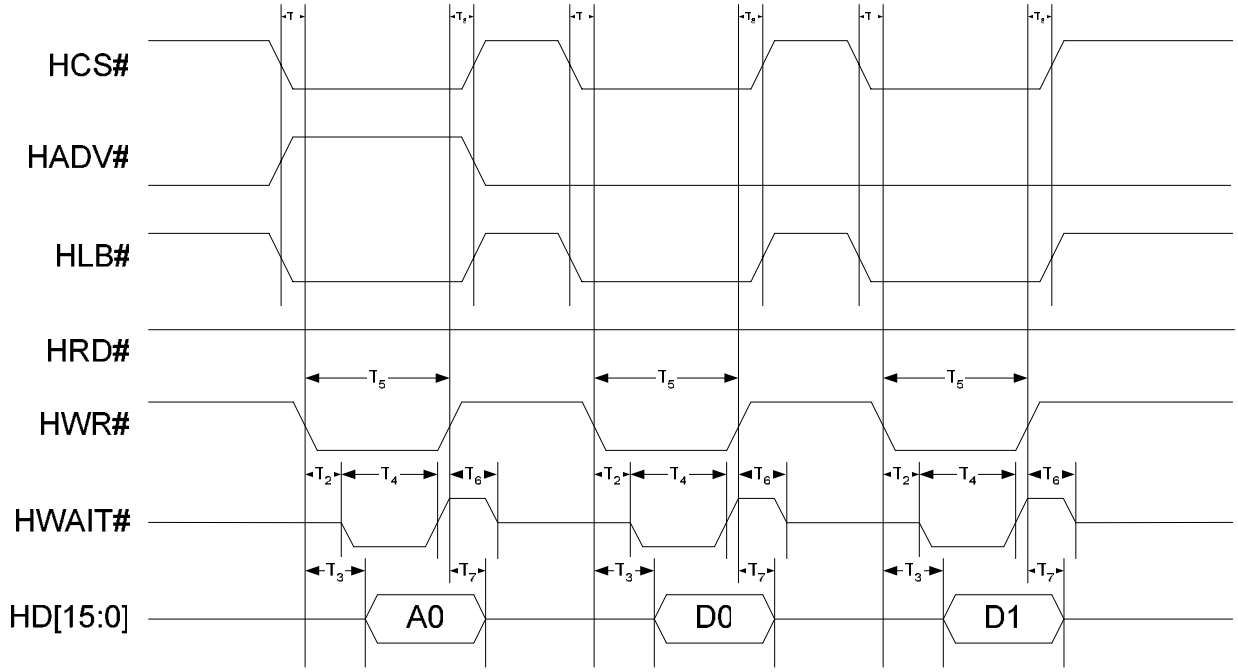


Figure 3.1.5-3 Indirect Addressing Mode 16 bits 80 Type Read Transaction

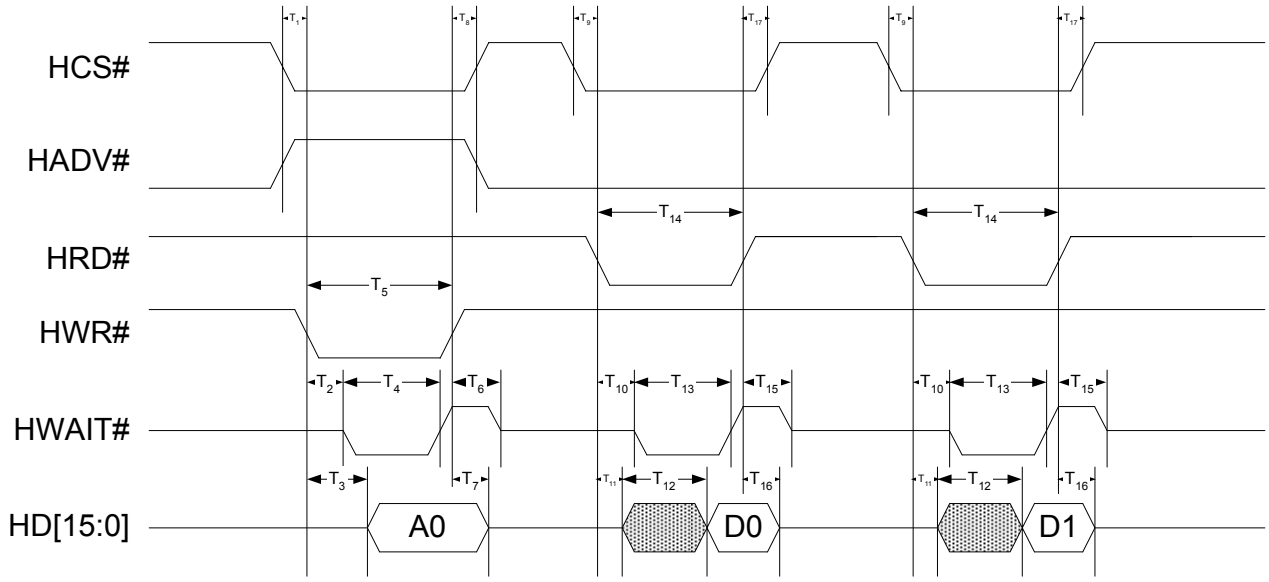


Table 3.1.5-1 Indirect Addressing Mode 16 bits 80 Type Interface Timing Table

Symbol	Parameter	Min	Typ	Max	Unit
T_1	Chip select, address valid and byte enable setup time from write falling edge	5			ns
T_2	Write falling edge to wait driven low	5.0	7.0	11.0	ns
T_3	Data delay from write falling edge			30	ns
T_4	Wait period during write cycle	(1)Note	(1)Note	(1)Note	
T_5	Write active period	30			ns
T_6	Write rising edge to wait high impedance	3.0	4.0	7.0	ns
T_7	Data hold time from write write rising edge	5			ns
T_8	Chip select, address valid and byte enable hold time from write rising edge	0			ns
T_9	Chip select, address valid and byte enable setup time from read falling edge	5			ns
T_{10}	Read falling edge to wait driven low	4.5	6.5	10.5	ns
T_{11}	Read falling edge to data driven	3.0	4.0	6.0	ns
T_{12}	Valid data period	(1)Note	(1)Note	(1)Note	-
T_{13}	Wait period during read cycle	(1)Note	(1)Note	(1)Note	-
T_{14}	Read active period	30.0			ns
T_{15}	Read rising edge to wait high impedance	3.0	3.5	6.5	ns
T_{16}	Data hold time from read rising edge	10.5	11.5	14.0	ns
T_{17}	Chip select, address valid and byte enable hold time from read rising edge	0			ns

Note:

1. See Table 3.1.5-2.

Table 3.1.5-2 Indirect Addressing Mode 16 bits 68 Type Interface Wait Period Table

Description	Min	Typ(2)	Max	Unit
Single Write to Registers		0	3(3)	T(1)
Single Read from Registers except ISP, JPEG, MPEG and 3D Registers		2	5(4)	T
Single Read from ISP Registers		3		T
Single Read from JPEG Registers		5		T
Single Read from MPEG Registers		5	7(5)	T
Single Read from 3D Registers		7		T
Consecutive(6) Write to Registers Penalty		1		T
Consecutive Read from Registers Penalty		1		T
Write to Registers after Read Penalty		1		T
Read from Registers after Write Penalty		3		T

Note:

1. T represents inner clock period.
2. Due to asynchronous timing, all above real timing should add the time from asynchronous asserted to inner clock latch edge.
3. Refer to 0x0200h[5:4] register in programming guide.
4. Refer to 0x0200h[7:6] register in programming guide.
5. Time for the first reading from MPEG registers.
6. It means read/write at the same time in 1T period.

3.1.6 Indirect Addressing Mode 16 bits 68 Type Interface

Figure 3.1.6-1 Indirect Addressing Mode 16 bits 68 Type Interface Implementation

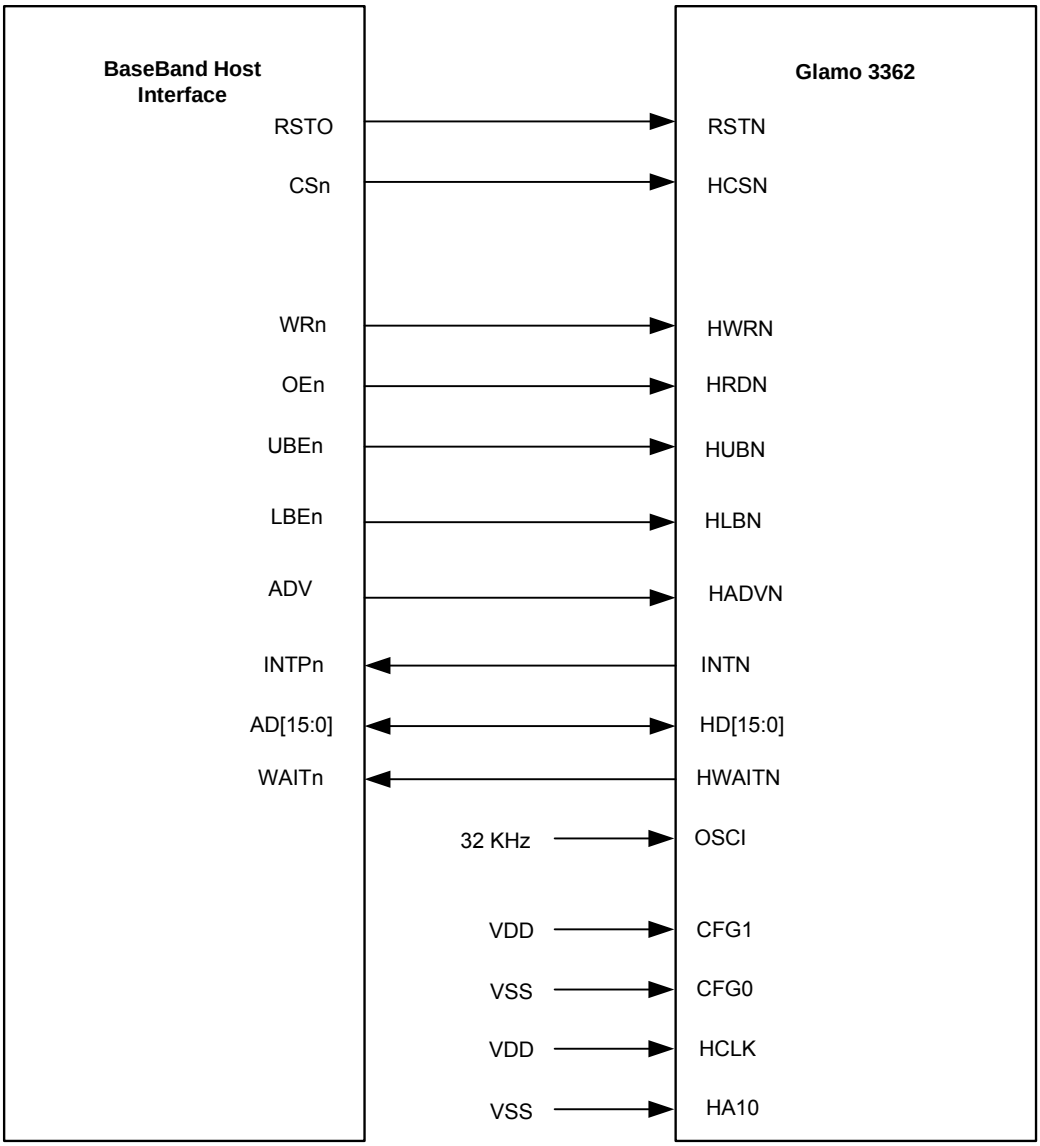


Figure 3.1.6-2 Indirect Addressing Mode 16 bits 68 Type Write Transaction

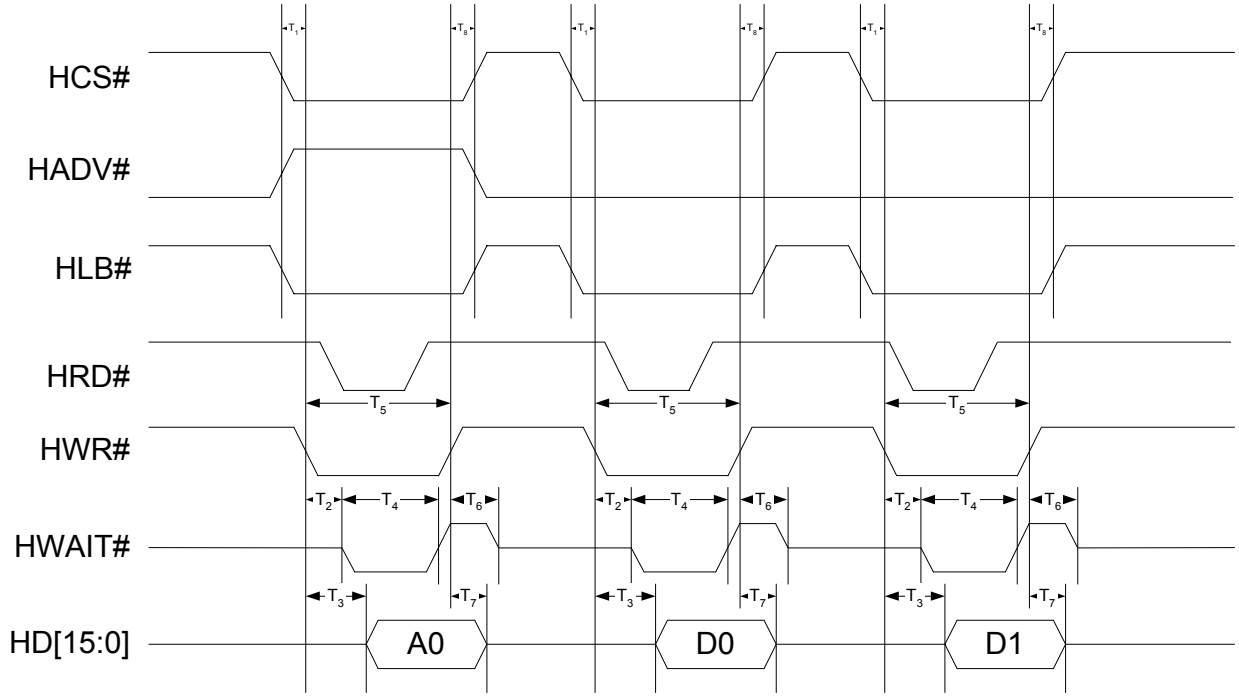


Figure 3.1.6-3 Indirect Addressing Mode 16 bits 68 Type Read Transaction

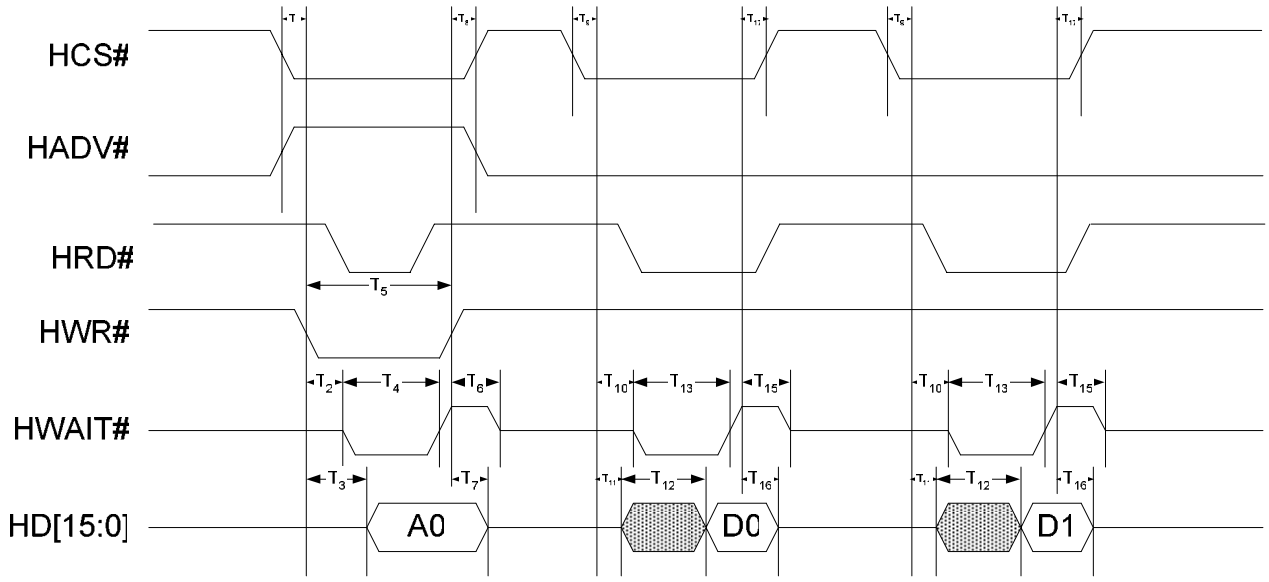


Table 3.1.6-1 Indirect Addressing Mode 16 bits 68 Type Interface Timing Table

Symbol	Parameter	Min	Typ	Max	Unit
T_1	Chip select, address valid and byte enable setup time from write falling edge	5			ns
T_2	Write falling edge to wait driven low	5.0	7.0	11.0	ns
T_3	Data delay from write falling edge			30	ns
T_4	Wait period during write cycle	(1)Note	(1)Note	(1)Note	
T_5	Write active period	30			ns
T_6	Write rising edge to wait high impedance	3.0	4.0	7.0	ns
T_7	Data hold time from write write rising edge	5			ns
T_8	Chip select, address valid and byte enable hold time from write rising edge	0			ns
T_9	Chip select, address valid and byte enable setup time from read falling edge	5			ns
T_{10}	Read falling edge to wait driven low	4.5	6.5	10.5	ns
T_{11}	Read falling edge to data driven	3.0	4.0	6.0	ns
T_{12}	Valid data period	(1)Note	(1)Note	(1)Note	-
T_{13}	Wait period during read cycle	(1)Note	(1)Note	(1)Note	-
T_{14}	Read active period	30			ns
T_{15}	Read rising edge to wait high impedance	3.0	3.5	6.5	ns
T_{16}	Data hold time from read rising edge	10.5	11.5	14.0	ns
T_{17}	Chip select, address valid and byte enable hold time from read rising edge	0			ns

Note:

1. See Table 3.1.6-2.

Table 3.1.6-2 Indirect Addressing Mode 16 bits 68 Type Interface Wait Period Table

Description	Min	Typ(2)	Max	Unit
Single Write to Registers		0	3(3)	T(1)
Single Read from Registers except ISP, JPEG, MPEG and 3D Registers		2	5(4)	T
Single Read from ISP Registers		3		T
Single Read from JPEG Registers		5		T
Single Read from MPEG Registers		5	7(5)	T
Single Read from 3D Registers		7		T
Consecutive(6) Write to Registers Penalty		1		T
Consecutive Read from Registers Penalty		1		T
Write to Registers after Read Penalty		1		T
Read from Registers after Write Penalty		3		T

Note:

1. T represents inner clock period.
2. Due to asynchronous timing, all above real timing should add the time from asynchronous asserted to inner clock latch edge.
3. Refer to 0x0200h[5:4] register in programming guide.
4. Refer to 0x0200h[7:6] register in programming guide.
5. Time for the first reading from MPEG registers.
6. It means read/write at the same time in 1T period.

3.1.7 Indirect Addressing Mode 8 bits 80 Type Interface

Figure 3.1.7-1 Indirect Addressing Mode 8 bits 80 Type Interface Implementation

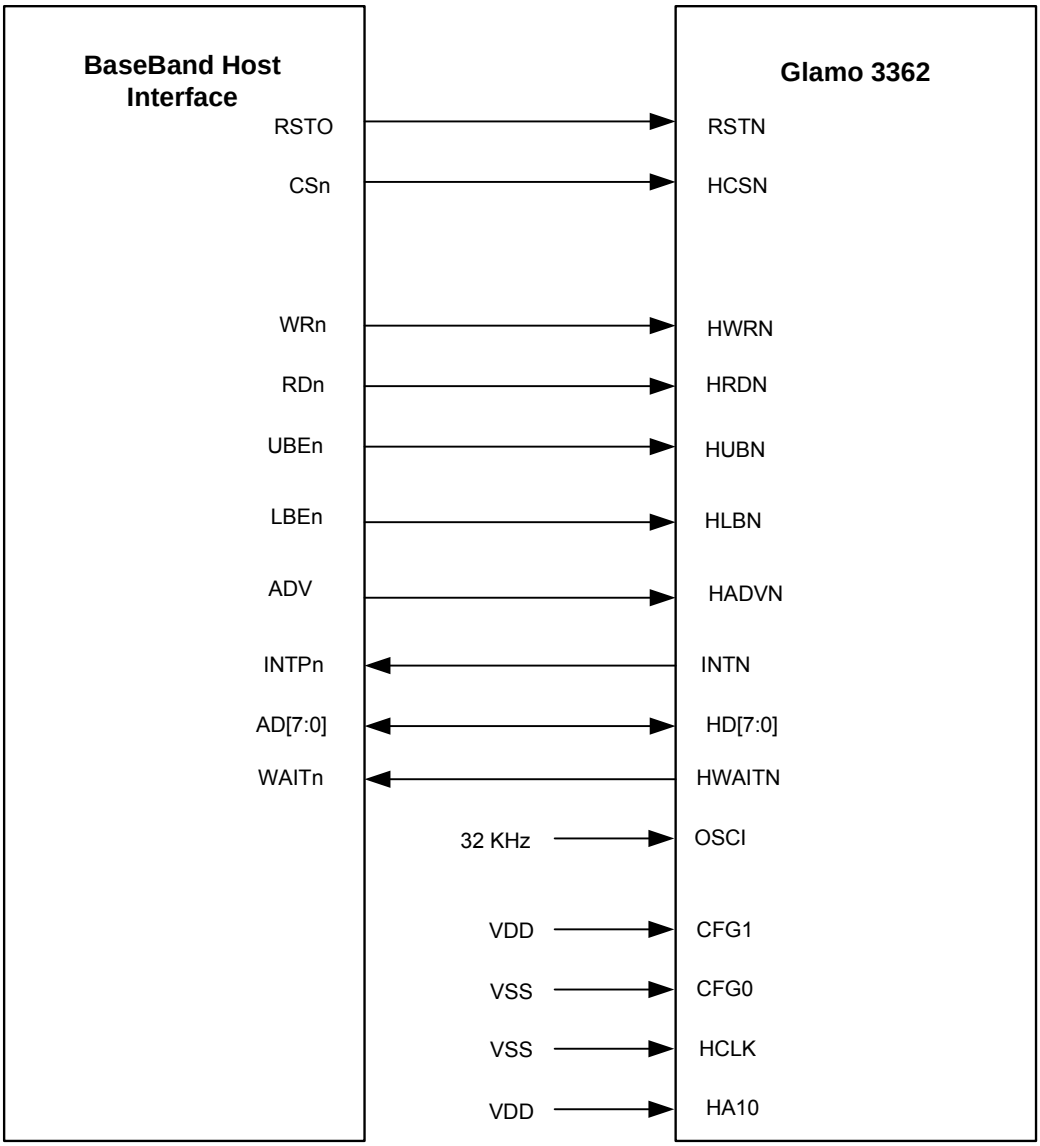


Figure 3.1.7-2 Indirect Addressing Mode 8 bits 80 Type Write Transaction

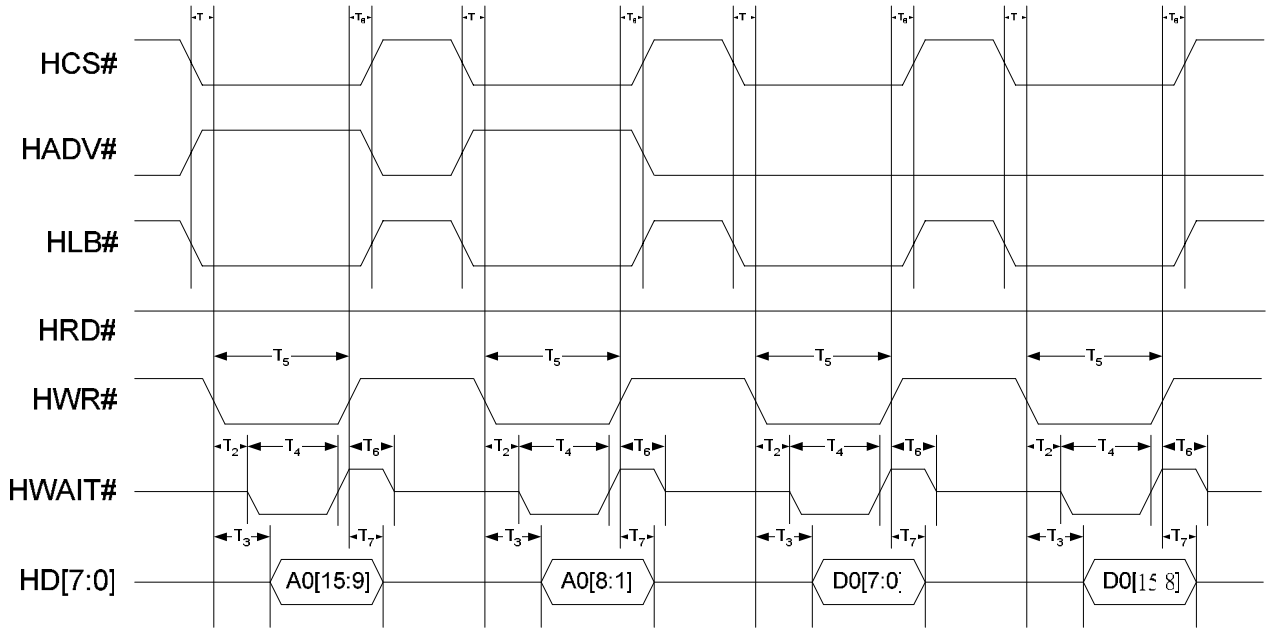
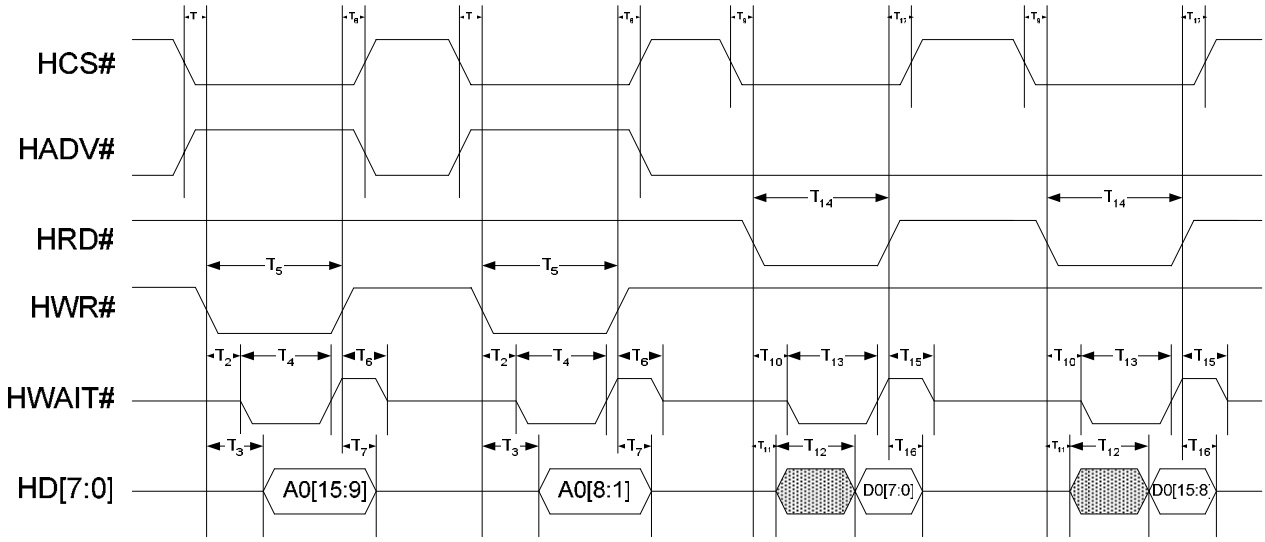


Figure 3.1.7-3 Indirect Addressing Mode



8 bits 80 Type Read Transaction

Table 3.1.7-1 Indirect Addressing Mode 8 bits 80 Type Interface Timing Table

Symbol	Parameter	Min	Typ	Max	Unit
T_1	Chip select, address valid and byte enable setup time from write falling edge	5			ns
T_2	Write falling edge to wait driven low	5.0	7.0	11.5	ns
T_3	Data delay from write falling edge			30	ns
T_4	Wait period during write cycle	(1)Note	(1)Note	(1)Note	
T_5	Write active period	30			ns
T_6	Write rising edge to wait high impedance	3.0	4.0	7.0	ns
T_7	Data hold time from write write rising edge	5			ns
T_8	Chip select, address valid and byte enable hold time from write rising edge	0			ns
T_9	Chip select, address valid and byte enable setup time from read falling edge	5			ns
T_{10}	Read falling edge to wait driven low	4.5	6.5	10.5	ns
T_{11}	Read falling edge to data driven	3.0	4.0	6.0	ns
T_{12}	Valid data period	(1)Note	(1)Note	(1)Note	-
T_{13}	Wait period during read cycle	(1)Note	(1)Note	(1)Note	-
T_{14}	Read active period	30			ns
T_{15}	Read rising edge to wait high impedance	3.0	3.5	6.5	ns
T_{16}	Data hold time from read rising edge	10.5	11.5	14.0	ns
T_{17}	Chip select, address valid and byte enable hold time from read rising edge	0			ns

Note:

1. See Table 3.1.7-2.

Table 3.1.7-2 Indirect Addressing Mode 8 bits 68 Type Interface Wait Period Table

Description	Min	Typ(2)	Max	Unit
Single Write to Registers		0	3(3)	T(1)
Single Read from Registers except ISP, JPEG, MPEG and 3D Registers		2	5(4)	T
Single Read from ISP Registers		3		T
Single Read from JPEG Registers		5		T
Single Read from MPEG Registers		5	7(5)	T
Single Read from 3D Registers		7		T
Consecutive(6) Write to Registers Penalty		1		T
Consecutive Read from Registers Penalty		1		T
Write to Registers after Read Penalty		1		T
Read from Registers after Write Penalty		3		T

Note:

1. T represents inner clock period.
2. Due to asynchronous timing, all above real timing should add the time from asynchronous asserted to inner clock latch edge.
3. Refer to 0x0200h[5:4] register in programming guide.
4. Refer to 0x0200h[7:6] register in programming guide.
5. Time for the first reading from MPEG registers.
6. It means read/write at the same time in 1T period.

3.1.8 Indirect Addressing Mode 8 bits 68 Type Interface

Figure 3.1.8-1 Indirect Addressing Mode 8 bits 68 Type Interface Implementation

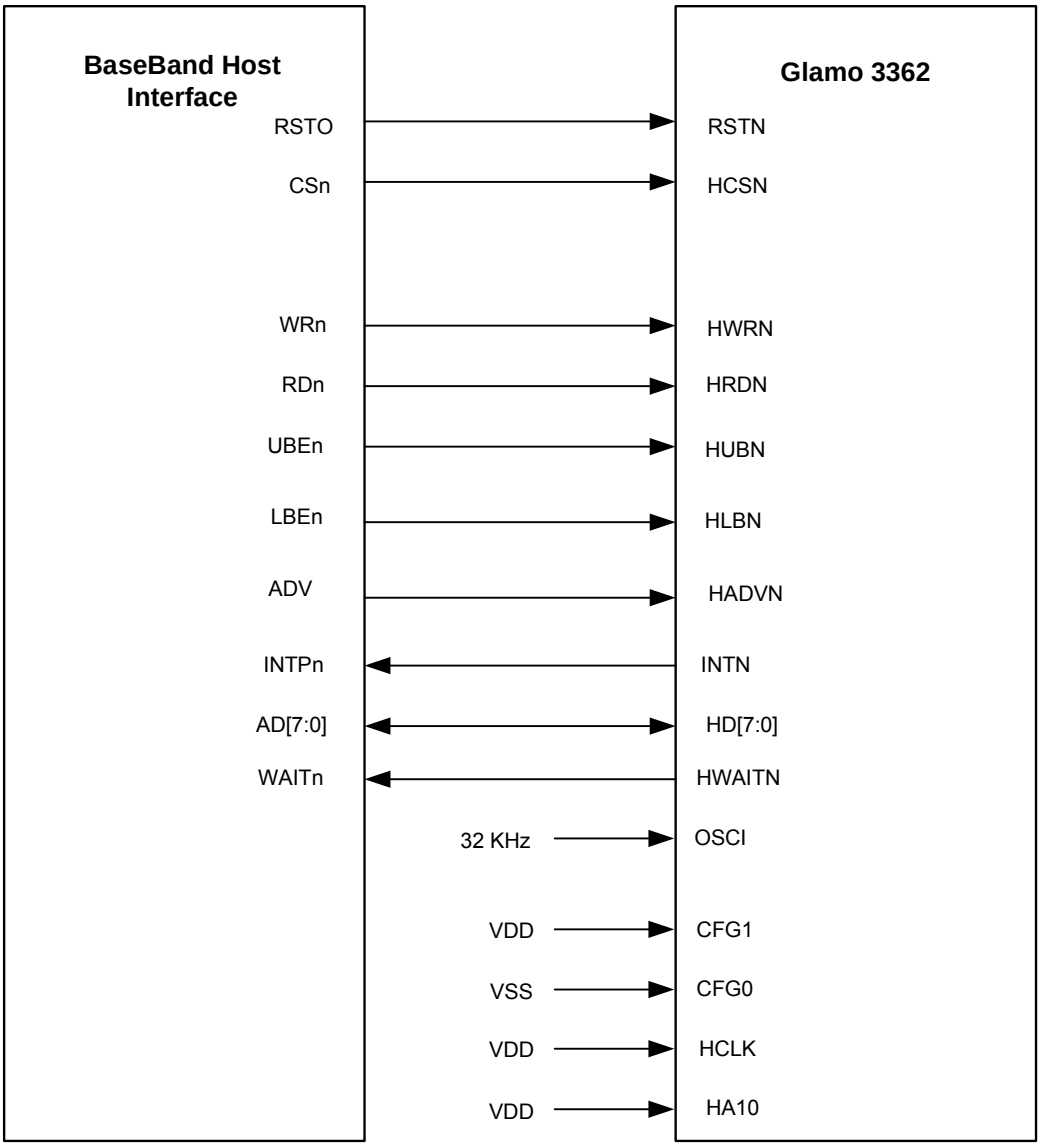


Figure 3.1.8-2 Indirect Addressing Mode 8 bits 68 Type Write Transaction

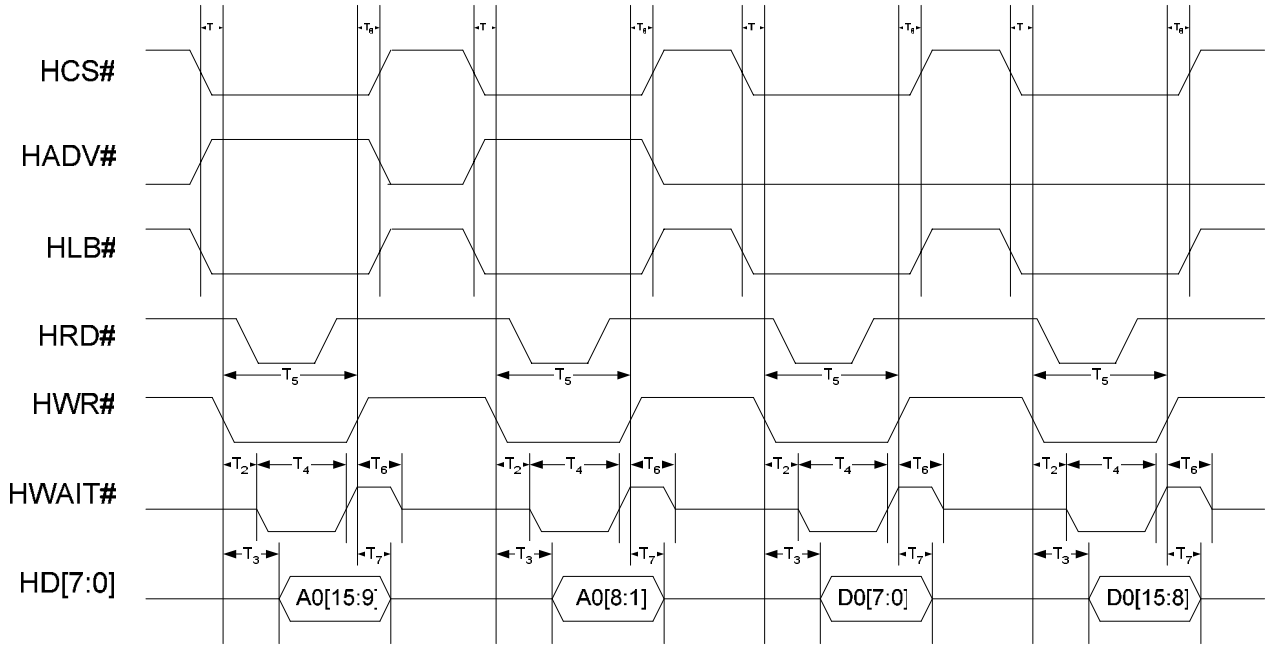


Figure 3.1.8-3 Indirect Addressing Mode 8 bits 68 Type Read Transaction

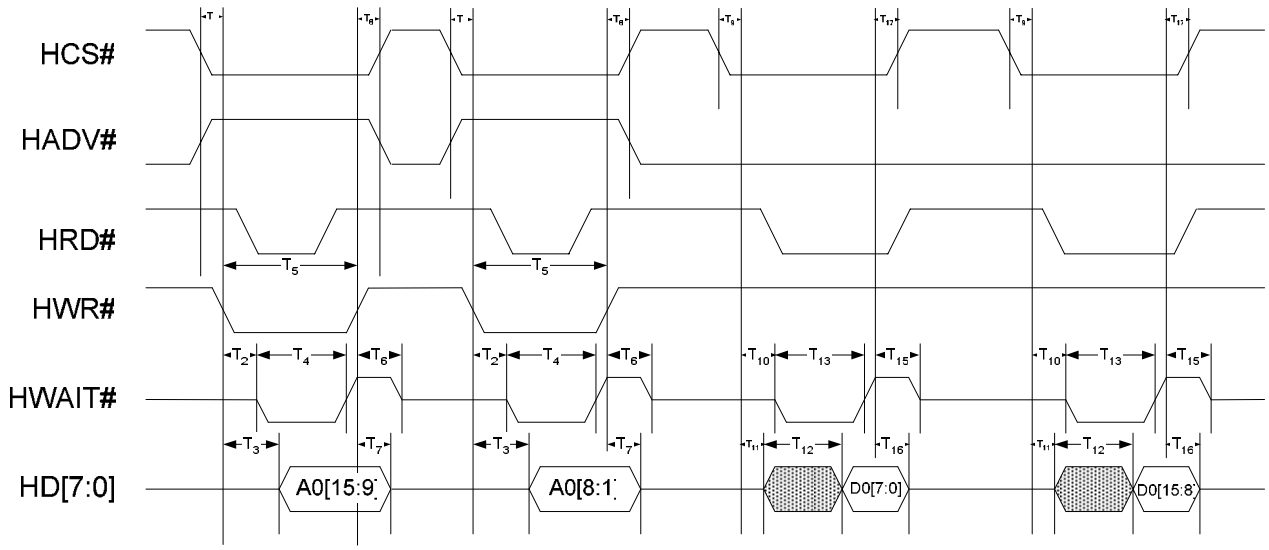


Table 3.1.8-1 Indirect Addressing Mode 8 bits 68 Type Interface Timing Table

Symbol	Parameter	Min	Typ	Max	Unit
T_1	Chip select, address valid and byte enable setup time from write falling edge	5			ns
T_2	Write falling edge to wait driven low	5.0	7.0	11.5	ns
T_3	Data delay from write falling edge			30	ns
T_4	Wait period during write cycle	(1)Note	(1)Note	(1)Note	
T_5	Write active period	30			ns
T_6	Write rising edge to wait high impedance	3.0	4.0	7.0	ns
T_7	Data hold time from write write rising edge	5.0			ns
T_8	Chip select, address valid and byte enable hold time from write rising edge	0			ns
T_9	Chip select, address valid and byte enable setup time from read falling edge	5.0			ns
T_{10}	Read falling edge to wait driven low	4.5	6.5	10.5	ns
T_{11}	Read falling edge to data driven	3.0	4.0	6.0	ns
T_{12}	Valid data period	(1)Note	(1)Note	(1)Note	-
T_{13}	Wait period during read cycle	(1)Note	(1)Note	(1)Note	-
T_{14}	Read active period	30			ns
T_{15}	Read rising edge to wait high impedance	3.0	3.5	6.5	ns
T_{16}	Data hold time from read rising edge	10.5	11.5	14.0	ns
T_{17}	Chip select, address valid and byte enable hold time from read rising edge	0			ns

Note:

1. See Table 3.1.8-2.

Table 3.1.8-2 Indirect Addressing Mode 8 bits 68 Type Interface Wait Period Table

Description	Min	Typ(2)	Max	Unit
Single Write to Registers		0	3(3)	T(1)
Single Read from Registers except ISP, JPEG, MPEG and 3D Registers		2	5(4)	T
Single Read from ISP Registers		3		T
Single Read from JPEG Registers		5		T
Single Read from MPEG Registers		5	7(5)	T
Single Read from 3D Registers		7		T
Consecutive(6) Write to Registers Penalty		1		T
Consecutive Read from Registers Penalty		1		T
Write to Registers after Read Penalty		1		T
Read from Registers after Write Penalty		3		T

Note:

1. T represents inner clock period.
2. Due to asynchronous timing, all above real timing should add the time from asynchronous asserted to inner clock latch edge.
3. Refer to 0x0200h[5:4] register in programming guide.
4. Refer to 0x0200h[7:6] register in programming guide.
5. Time for the first reading from MPEG registers.
6. It means read/write at the same time in 1T period.

3.1.9 Synchronous iBurst Interface

iBurst is a specific interface for synchronous transmission. By now, some baseband suppliers do plan to support it. The following is an example of Infineon's S-Gold2.

Figure 3.1.9-1 iBurst Interface Implementation

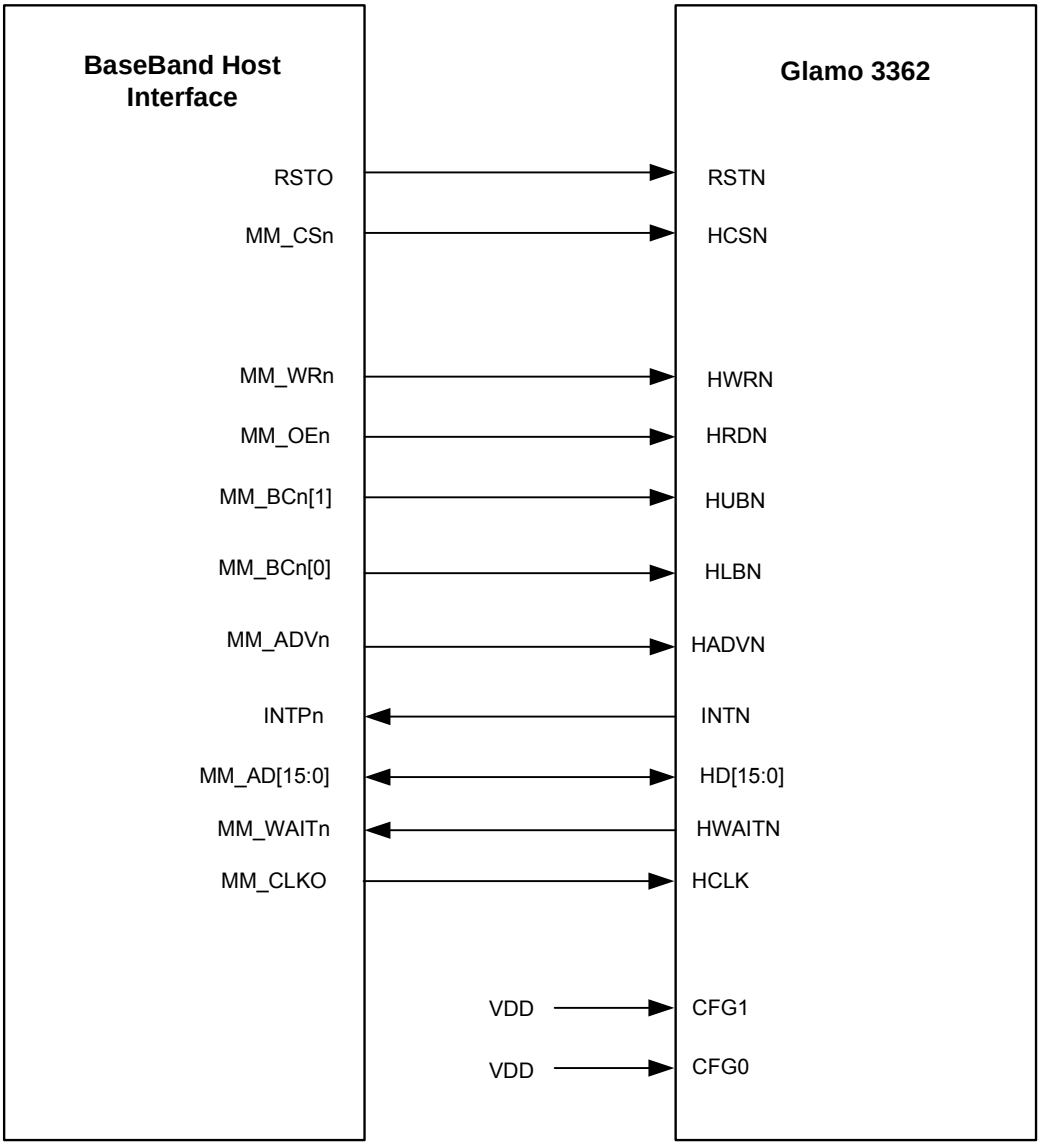


Figure 3.1.9-2 iBurst Single Write Transaction with Wait_Ctrl =1 (High-Low Addressing Multiplex)

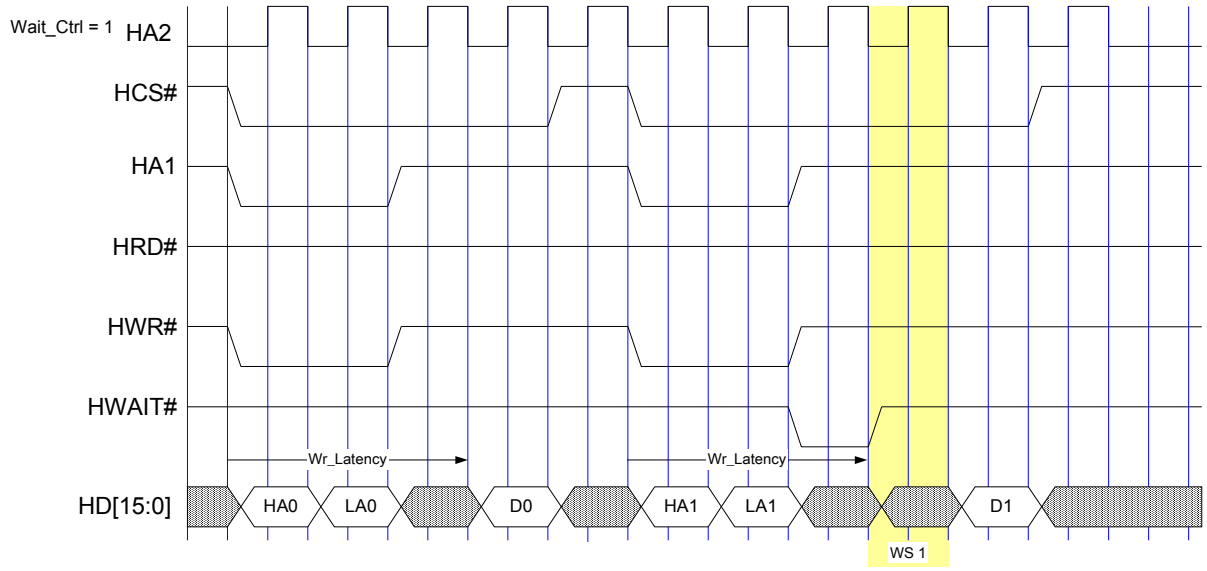


Figure 3.1.9-3 iBurst Single Write Transaction with Wait_Ctrl =0 (High-Low Addressing Multiplex)

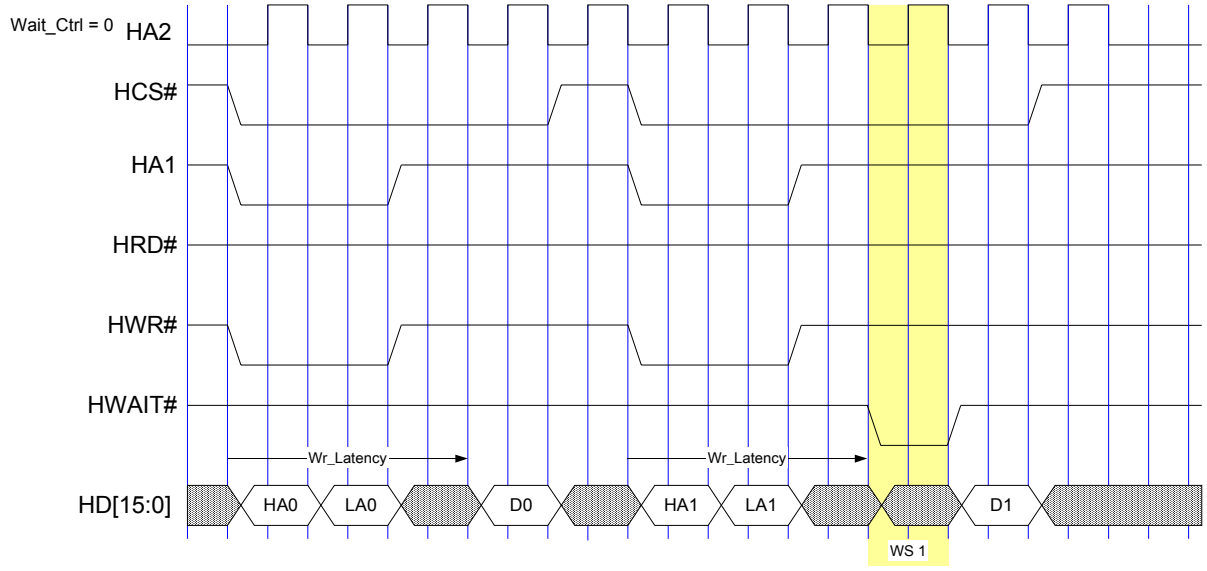


Figure 3.1.9-4 iBurst Burst Write Transaction with Wait_Ctrl =1 (High-Low Addressing Multiplex)

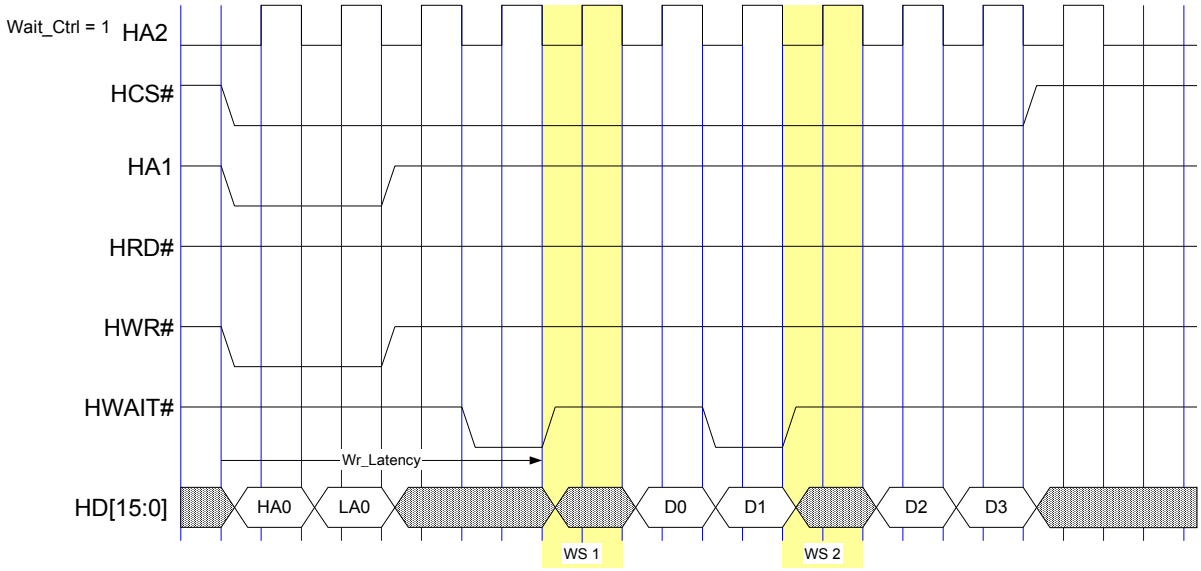


Figure 3.1.9-5 iBurst Burst Write Transaction with Wait_Ctrl =0 (High-Low Addressing Multiplex)

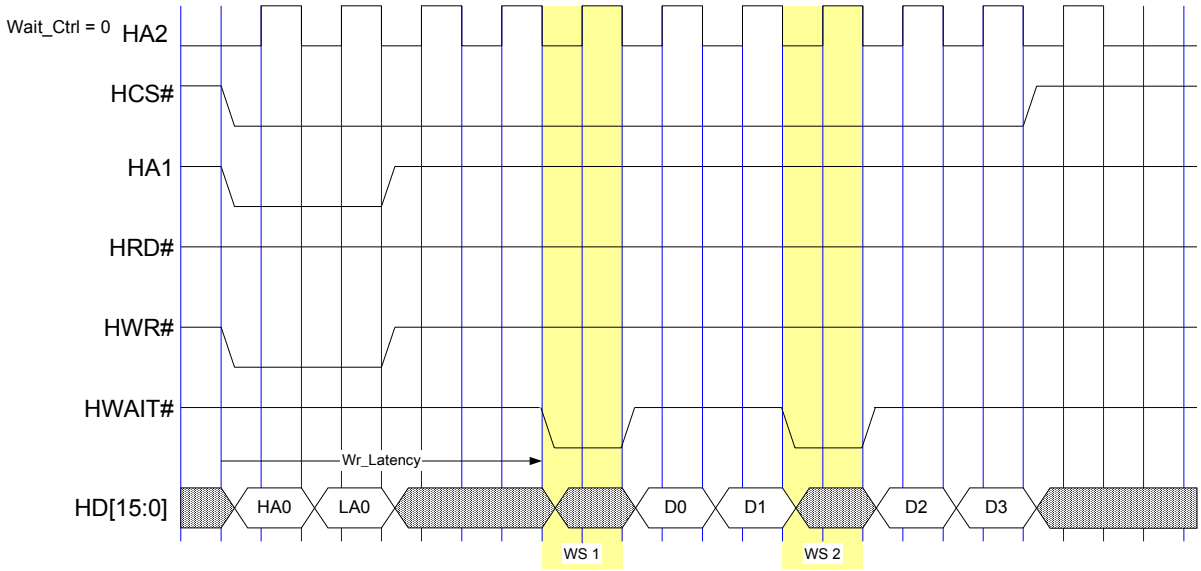


Figure 3.1.9-6 iBurst Single Read Transaction with Wait_Ctrl =1 (High-Low Addressing Multiplex)

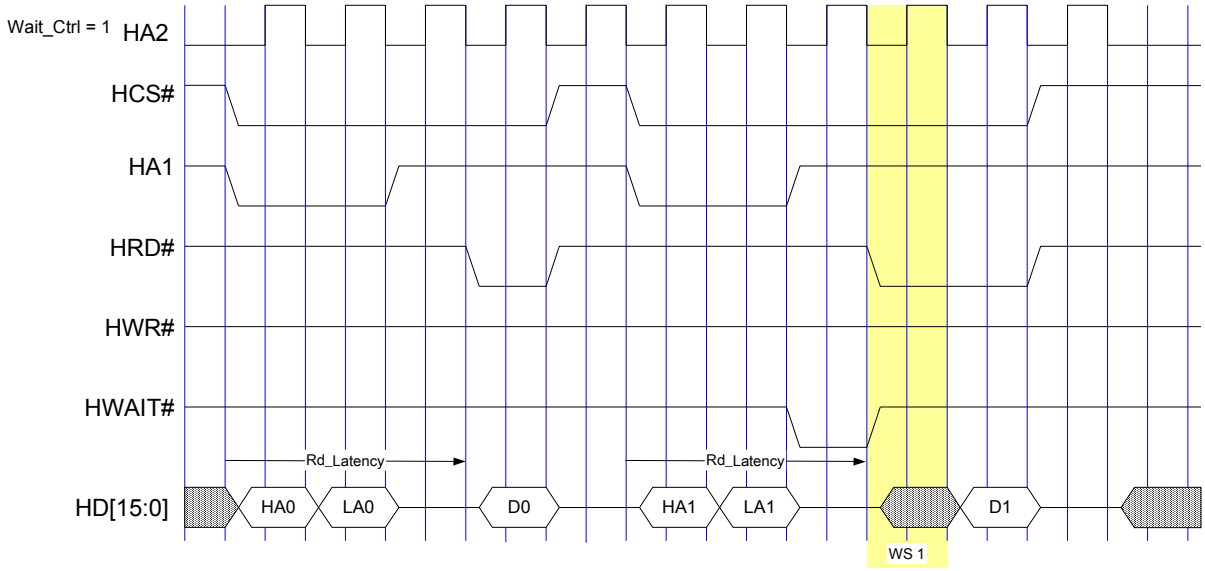


Figure 3.1.9-7 iBurst Single Read Transaction with Wait_Ctrl =0 (High-Low Addressing Multiplex)

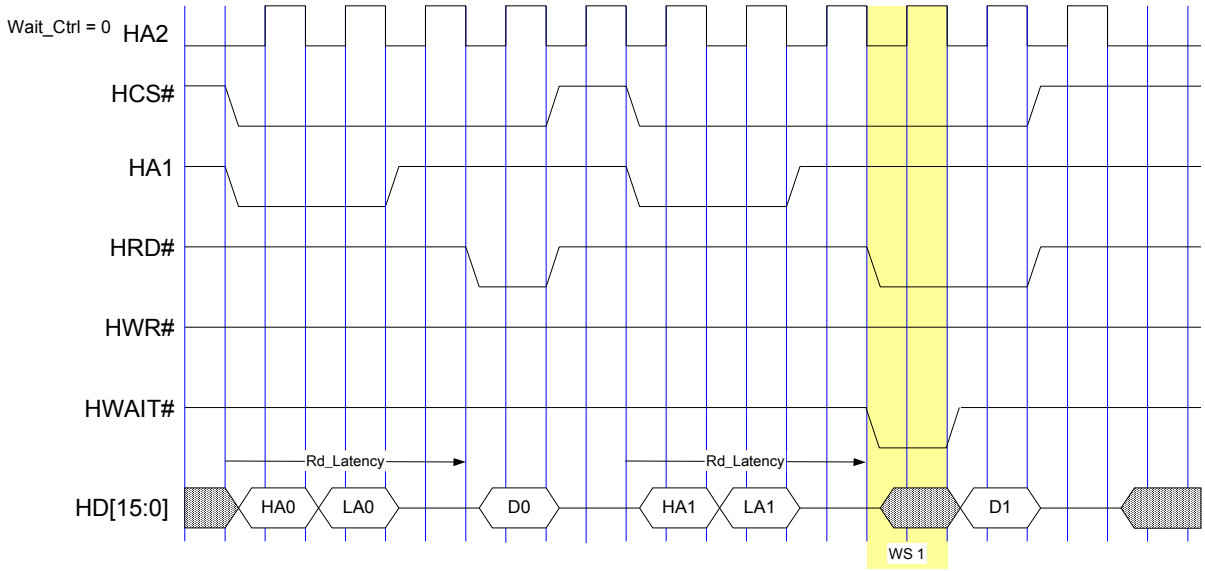


Figure 3.1.9-8 iBurst Burst Read Transaction with Wait_Ctrl =1 (High-Low Addressing Multiplex)

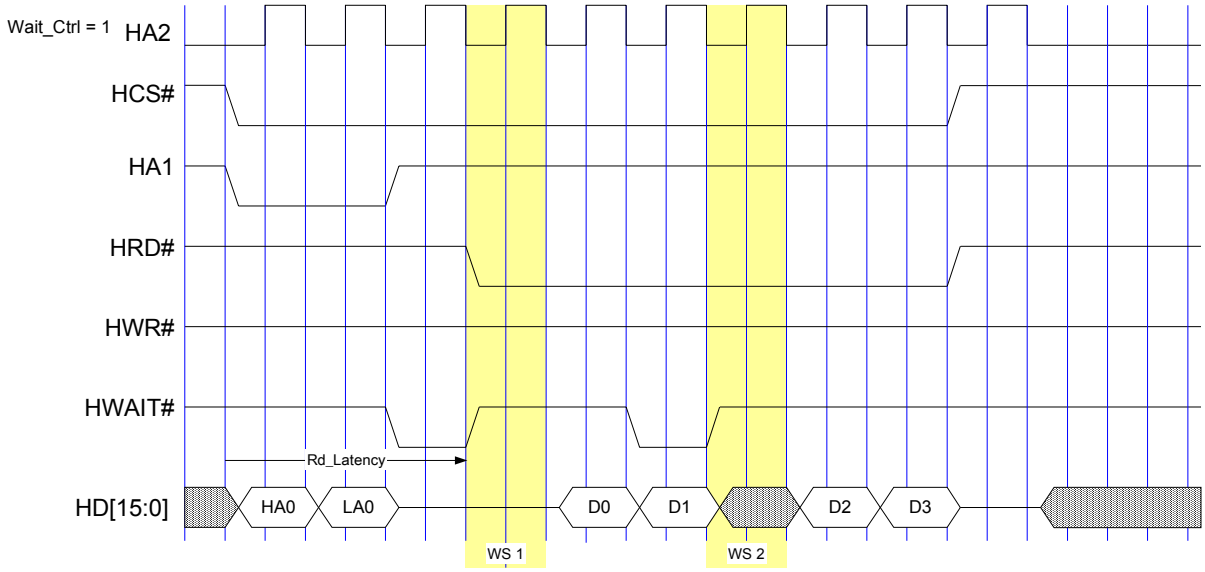


Figure 3.1.9-9 iBurst Burst Read Transaction with Wait_Ctrl =0 (High-Low Addressing Multiplex)

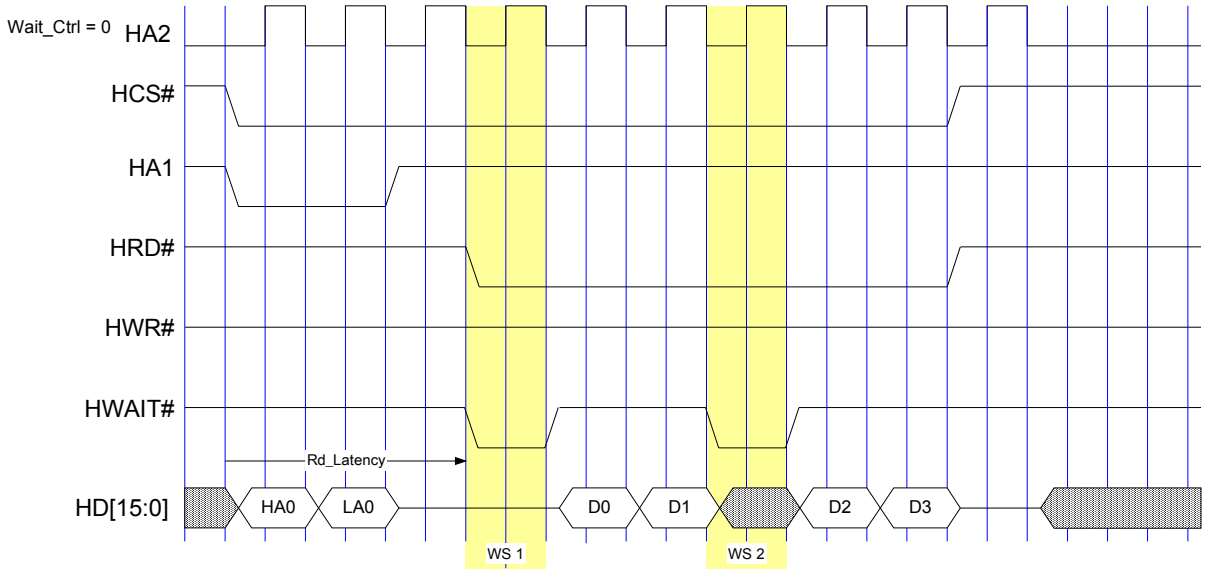


Table 3.1.9-1 iBurst Interface Read/Write Latency Table

Symbol	Parameter	Min	Typ	Max	Unit
Wr_Latency	Write latency cycle	0	4	15	T(1)
Rd_Latency	Read latency cycle	0	4	15	T

Note:

1. T represents one HCLK period from iBurst external clock.

3.2 LCD Interface

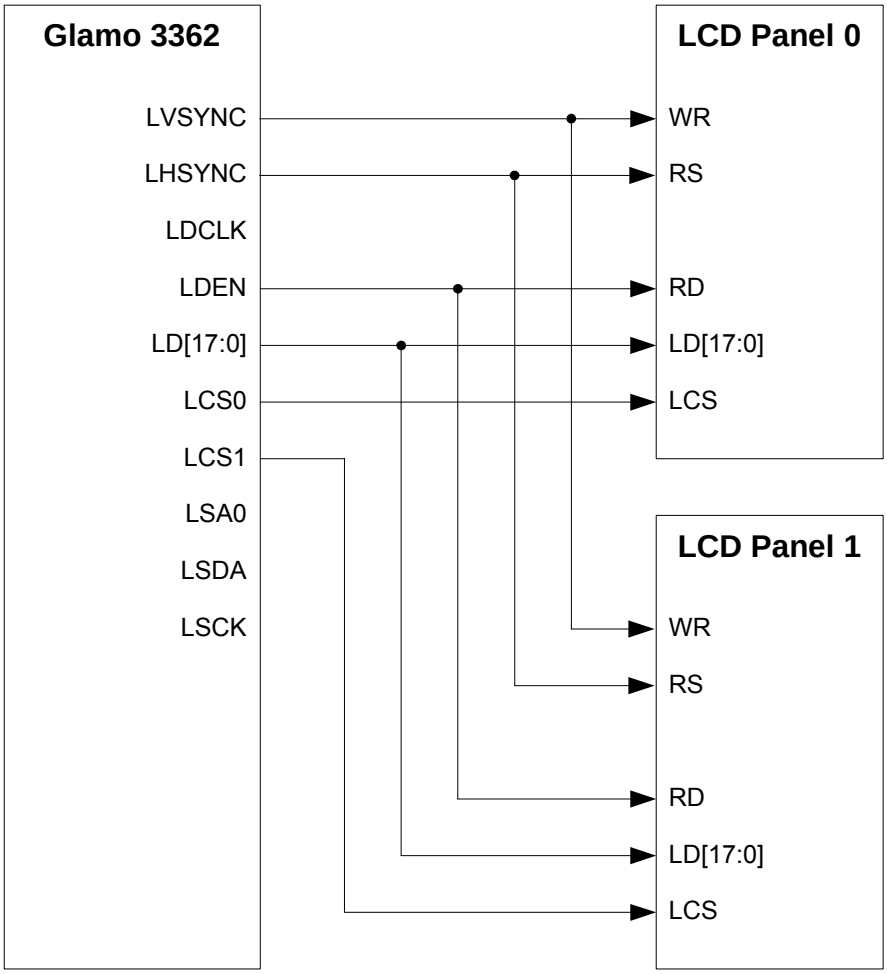
3.2.1 General Description

The Glamo 3362 supports 8/9/16/18 bits CPU interface, 6/9/16/18 bits RGB interface and by-pass mode. The Glamo 3362 supports both that with memory and that without memory LCD modules (LCM). The programmable interface timing is designed to fit most LCMs. The built-in power saving design is useful to increase the battery life.

3.2.2 80-Type CPU Interface

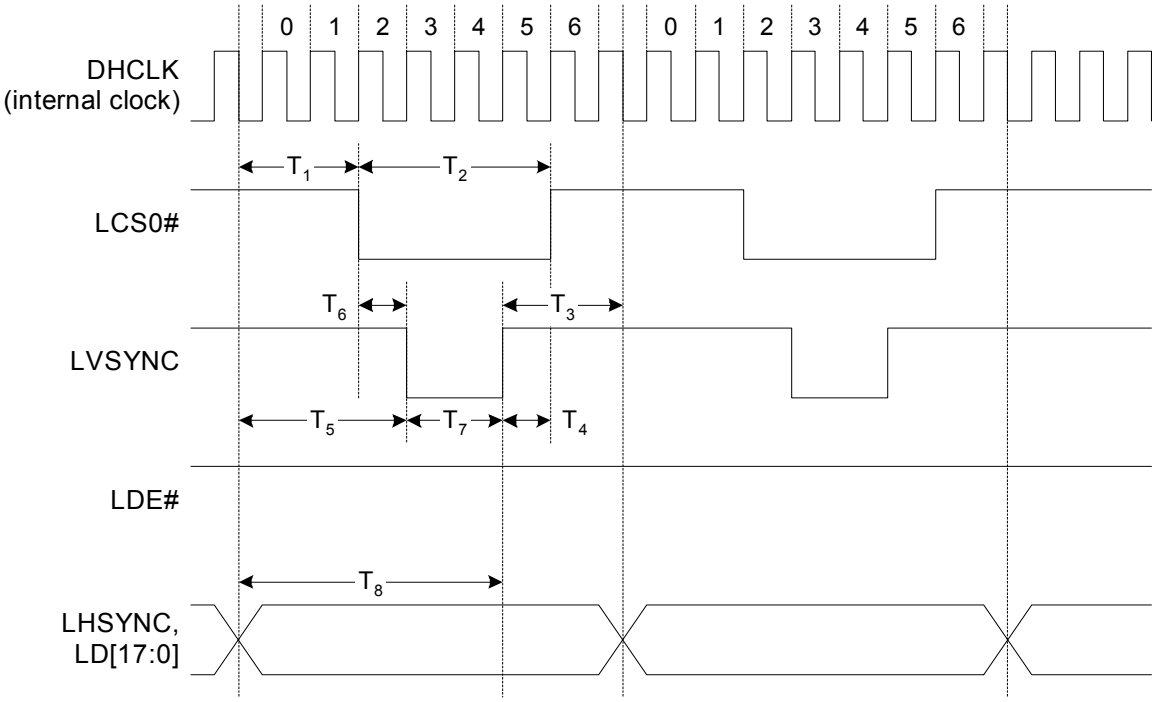
The figure below illustrates the implementation for interfacing the Glamo 3362 to an 80-type CPU interface LCM. Glamo 3362 can support dual display for this interface.

Figure 3.2.2-1 Connection of Glamo 3362 to 80-Type CPU Interface LCM



The timing for 80-type CPU interface is showed in the following figure. There is an internal clock DHCLK. By adjusting the timing parameters T_1 ~ T_8 which are relative to DHCLK, it can satisfy the timing requirement of LCD panels.

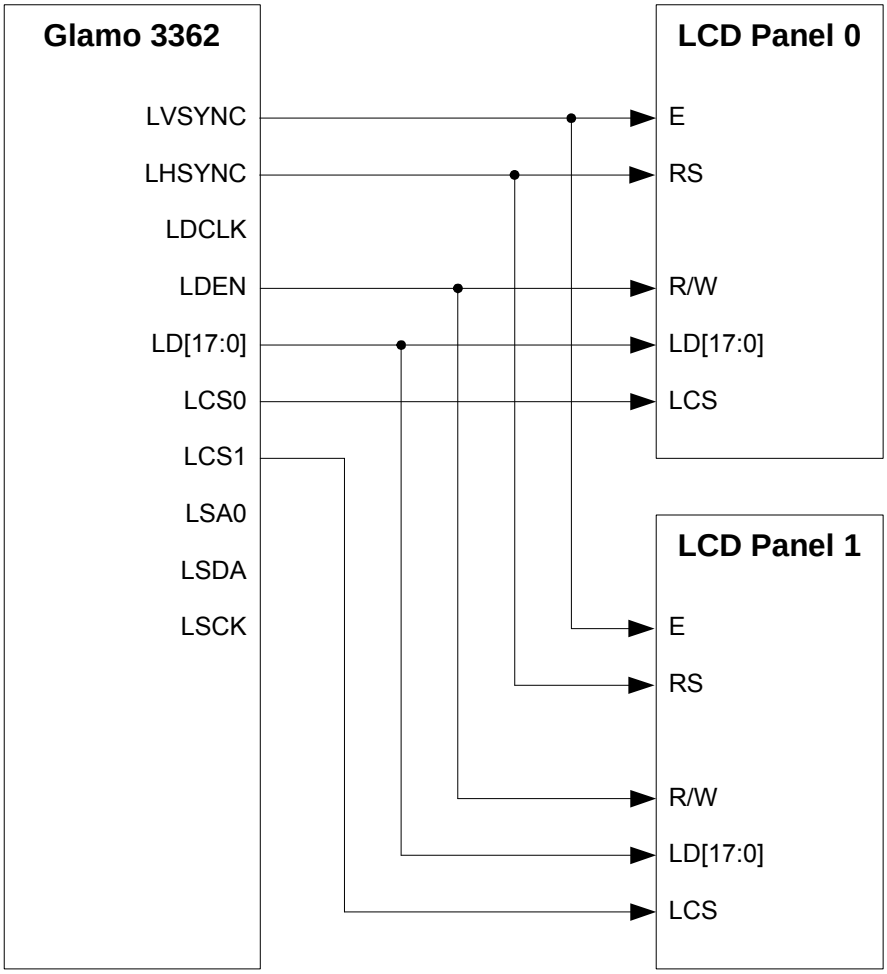
Figure 3.2.2-2 80-Type CPU Interface Timing



3.2.3 68-Type CPU Interface

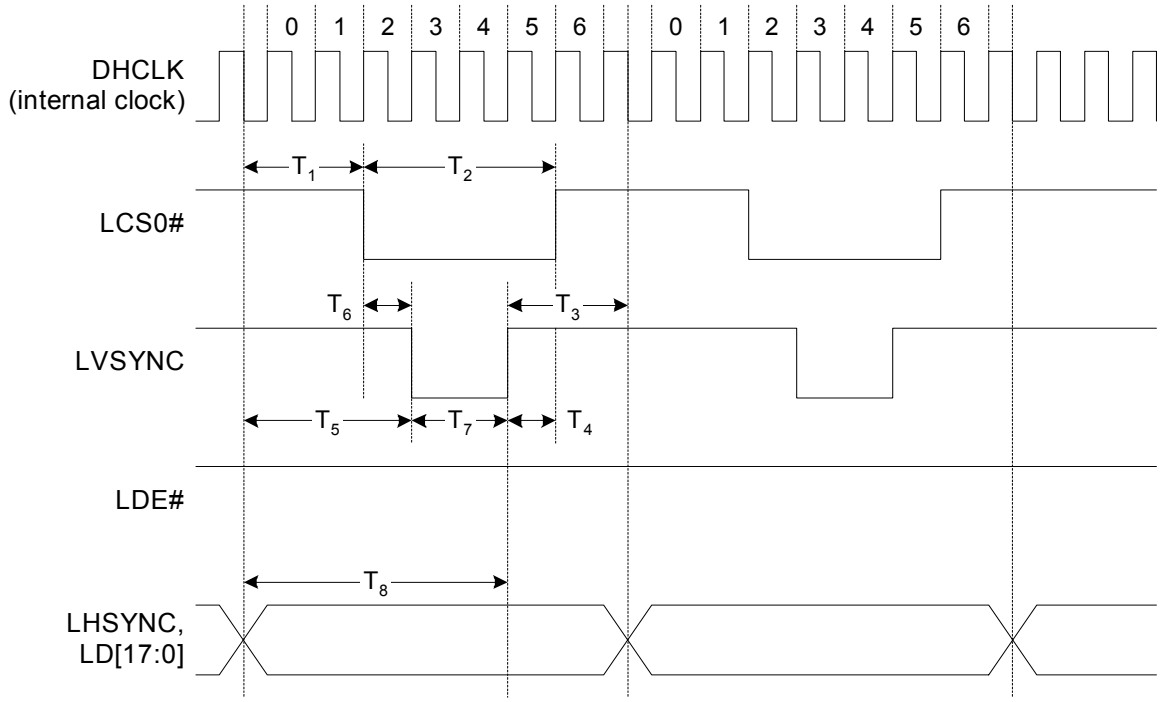
The figure below illustrates the implementation for interfacing the Glamo 3362 to a 68-type CPU interface LCM. Glamo 3362 can support dual display for this interface.

Figure 3.2.3-1 Connection of Glamo 3362 to 68-Type CPU Interface LCM



The timing for 68-type CPU interface is showed in the following figure. There is an internal clock DHCLK. By adjusting the timing parameters T_1 ~ T_8 which are relative to DHCLK, it can satisfy the timing requirement of LCD panels.

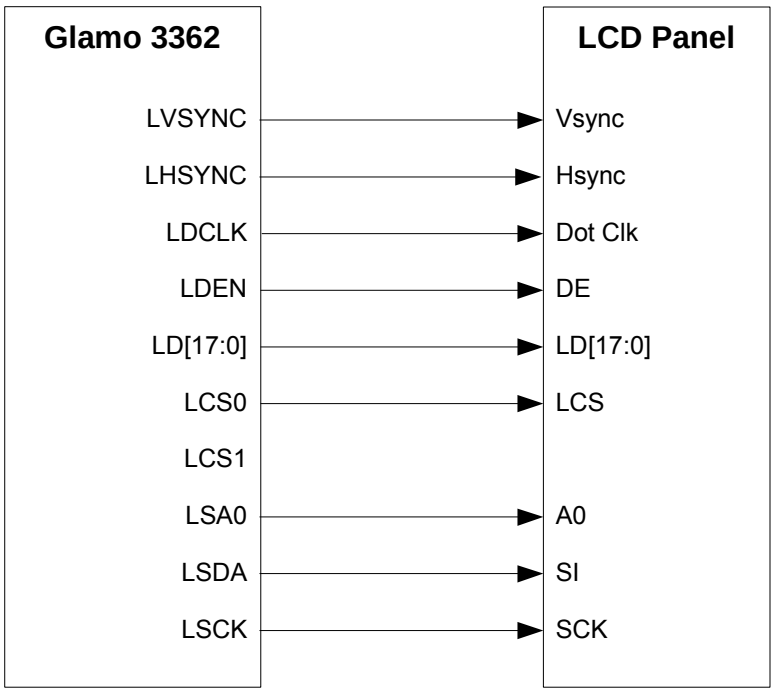
Figure 3.2.3-2 68-Type CPU Interface Timing



3.2.4 RGB with Serial Interface

The figure below illustrates the implementation for interfacing the Glamo 3362 to a RGB with serial interface LCM. Glamo 3362 only support one display for this interface

Figure 3.2.4-1 Connection of Glamo 3362 to RGB with Serial Interface LCM



Glamo 3362 supports 8/9/24 bits serial interface to send LCM command and uses 6/9/16/18 bits parallel RGB interface to send display data. The timing for RGB with serial interface is shown in the following figure.

Figure 3.2.4-2 RGB Timing (for Data)

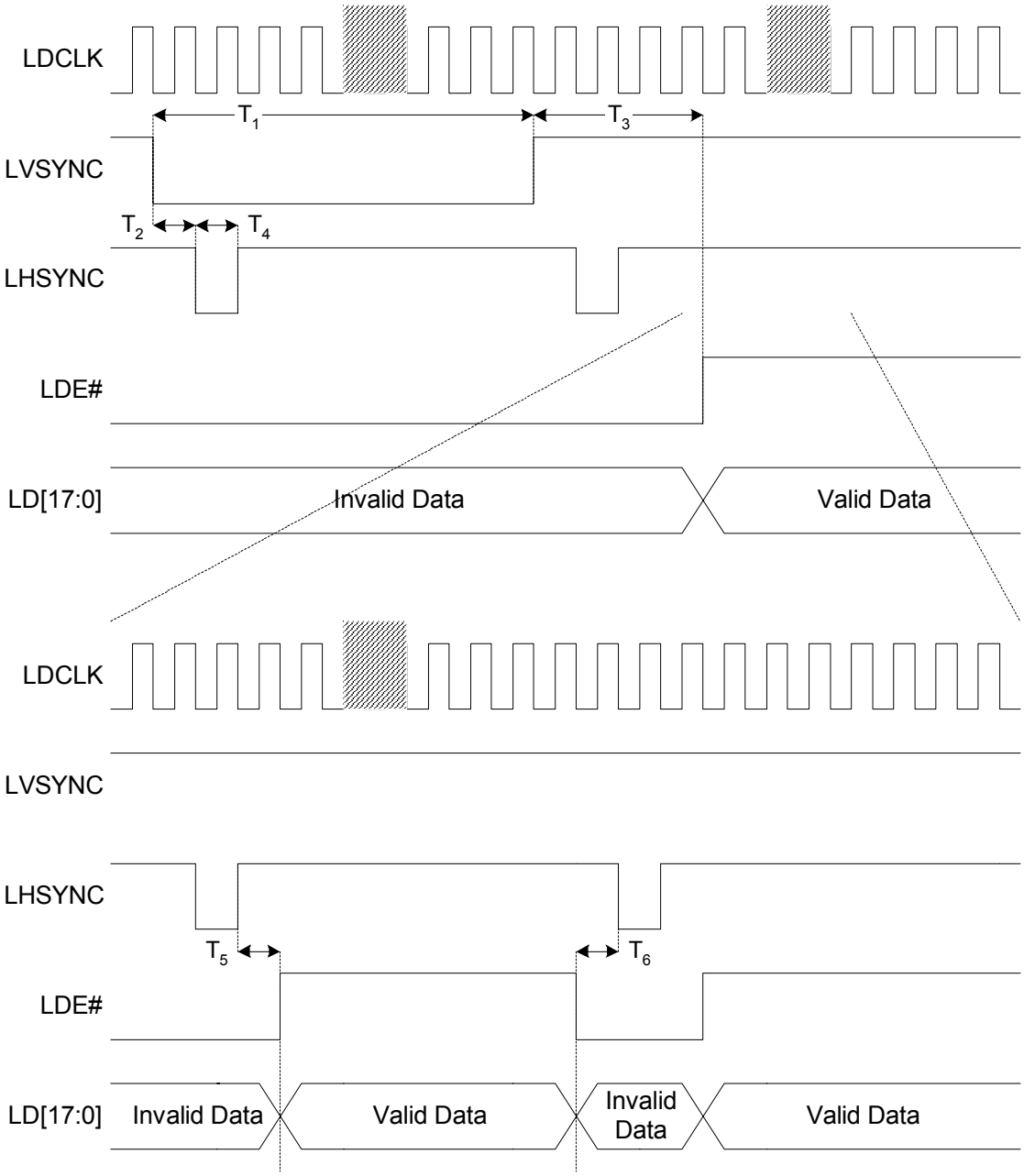
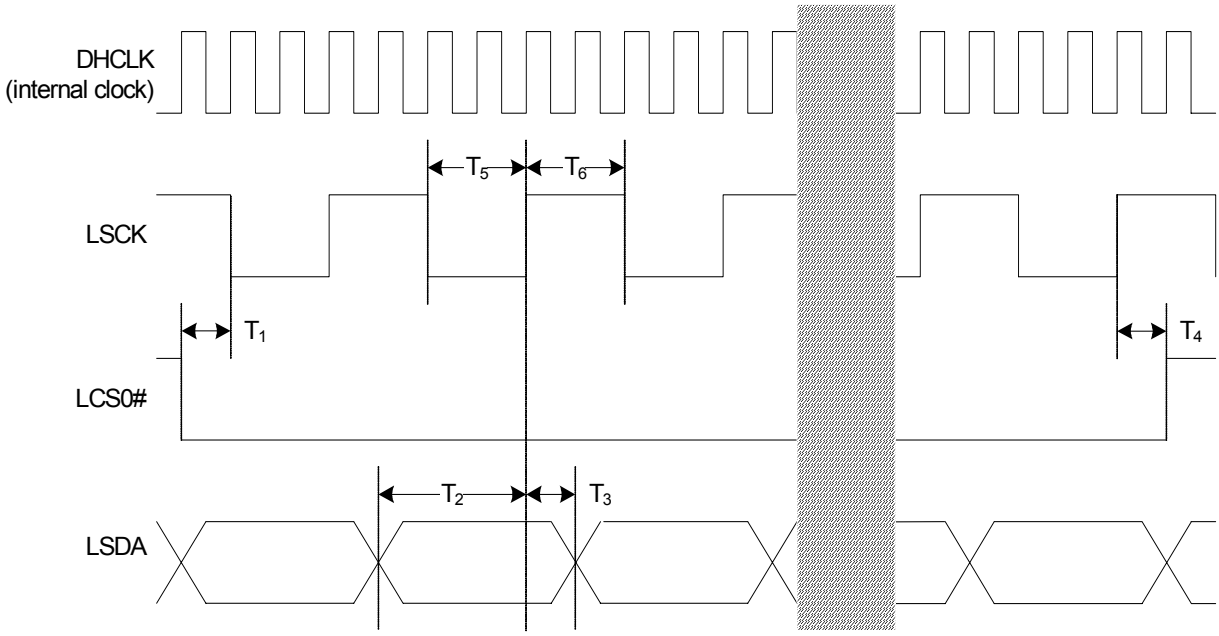


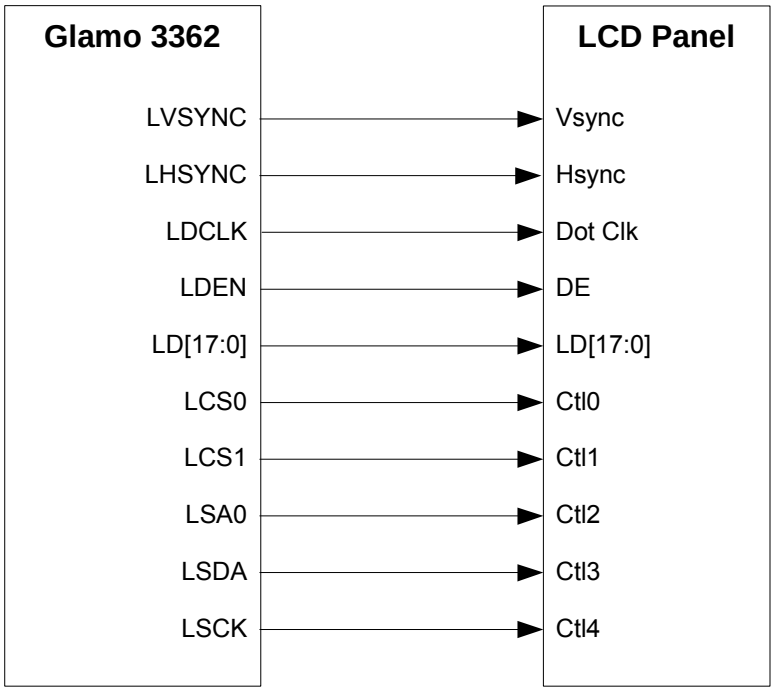
Figure 3.2.4-3 RGB with Serial Interface Timing (for Command)


DHCLK
(internal clock)
LSCK
LCS0#
LSDA
T1
T2
T3
T4
T5
T6

3.2.5 RGB with Direct Control Interface

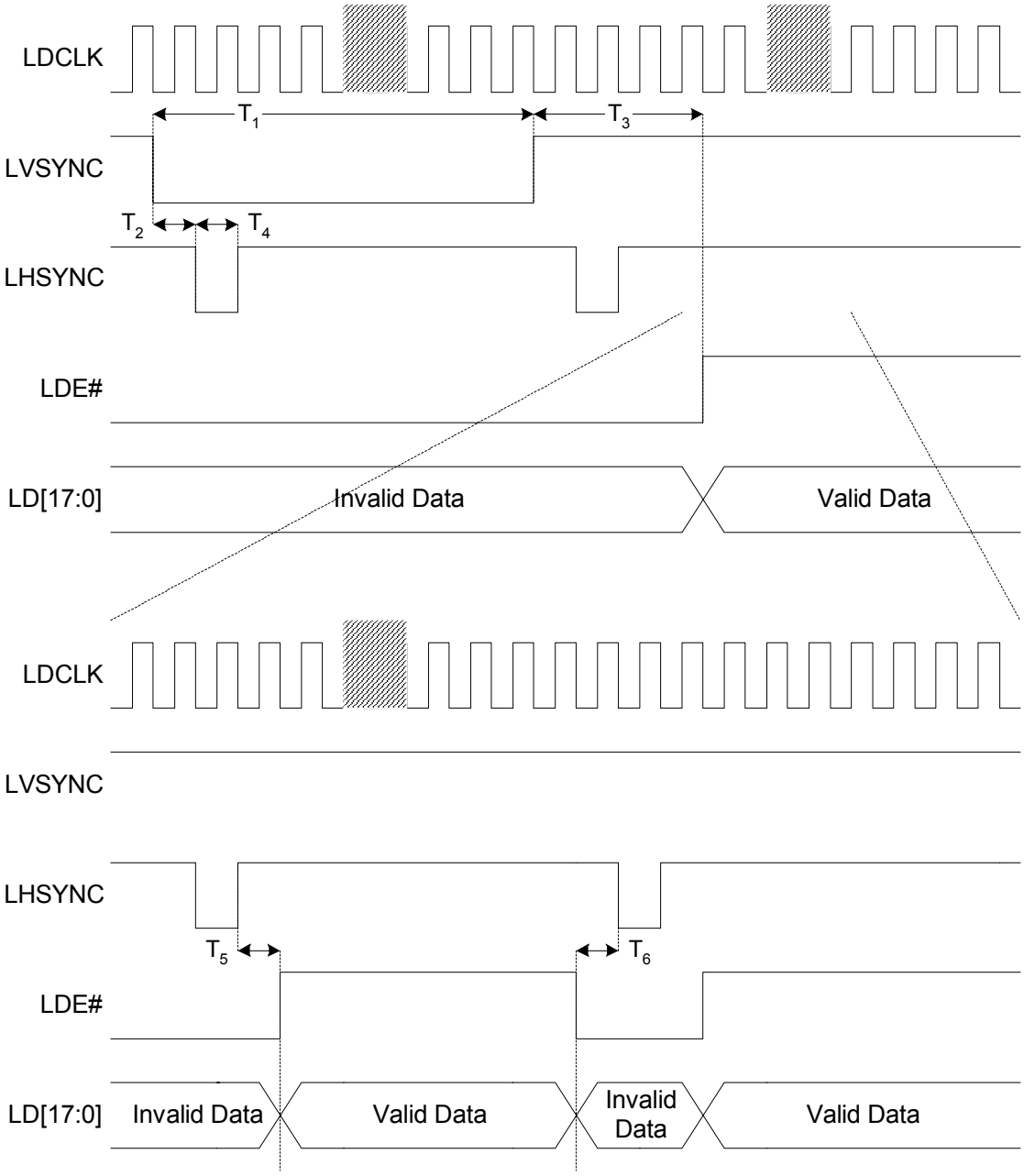
The figure below illustrates the implementation for interfacing the Glamo 3362 to a RGB with direct control interface LCM. Glamo 3362 only supports one display with this interface

Figure 3.2.5-1 Connection of Glamo 3362 to RGB with Direct Control Interface LCM



Some LCMs do not have any serial interface;, but only have parallel RGB interface. In this case, the un-used pins (LCS0#, LCS1#, LSA0, LSDA, LSCK) of the Glamo 3362 can be defined as GPIOs. The timing for RGB parallel interface is showed in the following figure.

Figure 3.2.5-2 RGB with Direct Control Interface Timing (for Data)


LDCLK
T1
T3
LVSYNC
T2
T4
LHSYNC
LDE#
LD[17:0]
Invalid Data
Valid Data
LDCLK
LVSYNC
LHSYNC
T5
T6
LDE#
LD[17:0]
Invalid Data
Valid Data
Invalid Data
Valid Data

3.2.6 By-Pass Mode Implementation

When turn on by-pass mode, some pins of the host interface will connect with the pins of LCD interface internally. In this mode, it supports 80-type CPU or 68-type CPU LCM.

Table 3.2.6-1 Mapping of Host Interface Pins to LCD Interface Pins for 80-Type CPU LCM

Host Interface Pin	LCD Interface Pin
HA1	LHSYNC
HD0	LD0
HD1	LD1
HD2	LD2
HD3	LD3
HD4	LD4
HD5	LD5
HD6	LD6
HD7	LD7
HD8	LD8
HD9	LD9
HD10	LD10
HD11	LD11
HD12	LD12
HD13	LD13
HD14	LD14
HD15	LD15
HCS#	LCS0#
HRD#	LDE#
HWR#	LVSYNC
HA2	LCS1#

Table 3.2.6-2 Mapping of Host Interface Pins to LCD Interface Pins for 68-Type CPU LCM

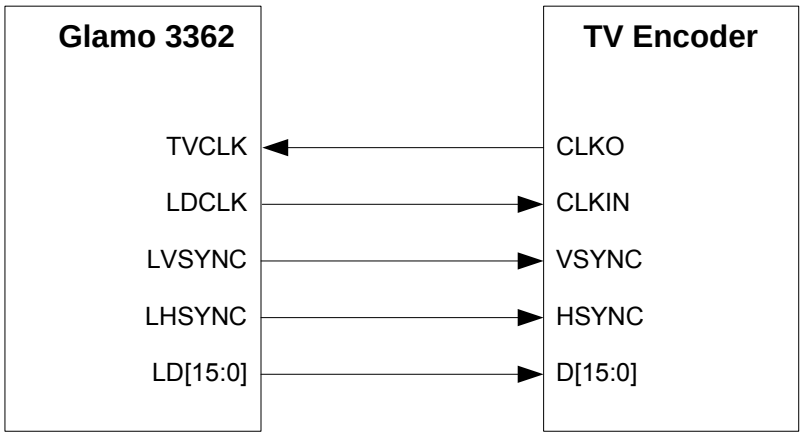
Host Interface Pin	LCD Interface Pin
HA1	LHSYNC
HD0	LD0
HD1	LD1
HD2	LD2
HD3	LD3
HD4	LD4
HD5	LD5
HD6	LD6
HD7	LD7
HD8	LD8
HD9	LD9
HD10	LD10
HD11	LD11
HD12	LD12
HD13	LD13
HD14	LD14
HD15	LD15
HCS#	LCS0#
HRD#	LDE#
HLB#	LVSYNC
HA2	LCS1#

3.2.7 TV encoder Implementation

Glamo 3362 can support TV encoder by LCD interface. Glamo 3362 connect to TV encoder can be configured as CPU mode or RGB mode, it is depends on the TV encoder. If the Glamo 3362 be configured as RGB mode, we recommend using pseudo master mode to connect TV encoder. In pseudo master mode, TV encoder output the Clock signal and input to Glamo 3362, then Glamo 3362 output RGB data, HSYNC, VSYNC signals to TV encode.

The figures below illustrate this implementation.

Figure 3.3.7-1 Connection of Glamo 3362 to TV Encoder



3.3 Video Interface

3.3.1 General Description

The Glamo 3362 provides a powerful video interface for most CCD/CMOS image sensors. It is designed to keep highly elasticity for users to communicate with most digital image sensors. There is a series interface that can be controlled by soft program to fit any series interface protocol. Besides, the Glamo 3362 provides a signal to control the external flashlight.

3.3.2 Video Interface Implementation

Video interface supports two modes for connecting sensor

- Mode1: When sensor provides a main clock input for sensor operation and a pixel clock out for latch pixel data, Glamo 3362 supports mode 1. (Such as Omnivision, Micron)
- Mode2: When sensor provides only one clock input for either sensor operation or latch pixel data, Glamo 3362 supports mode 2. (Such as Sony IU011)

Figure 3.3.2-1 Connection of Glamo 3362 to Sensor (Mode 1)

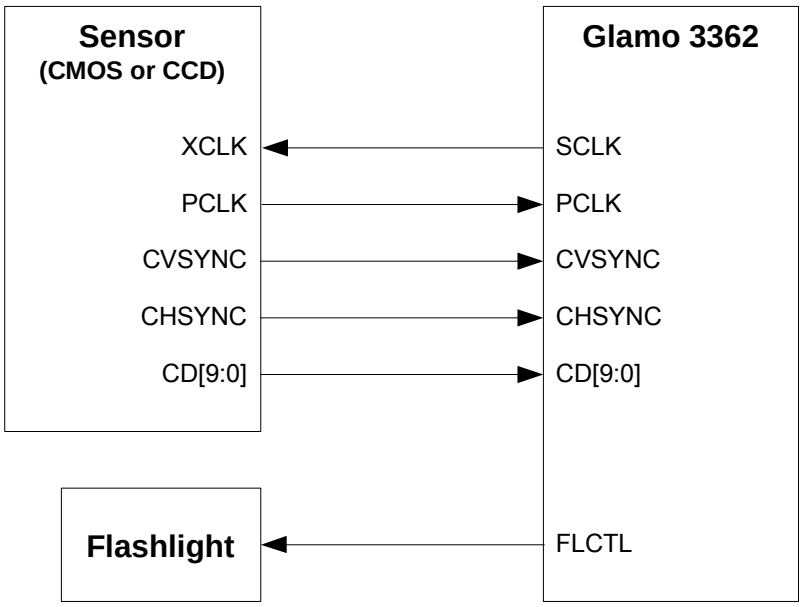
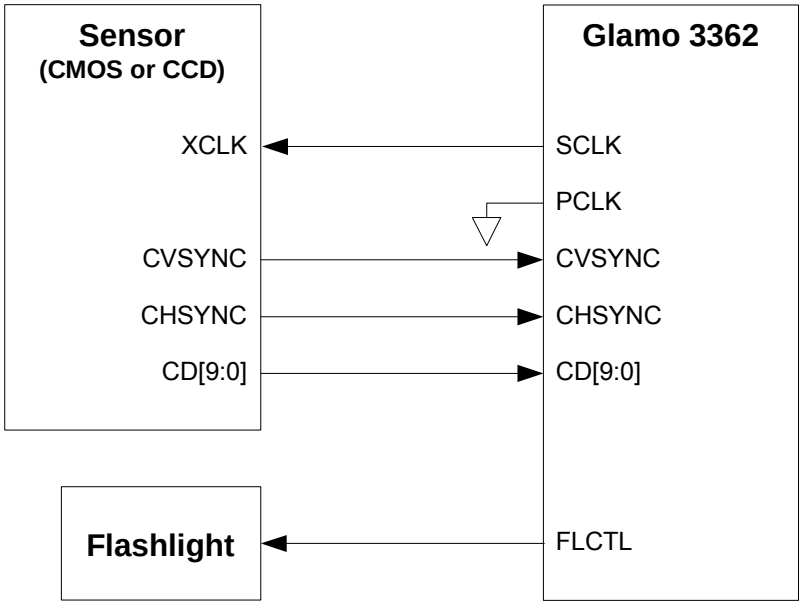


Figure 3.3.2-2 Connection of Glamo 3362 to Sensor (Mode 2)



3.3.3 Video Interface Timing Diagram

The video interface supports 4 input modes. The modes are positive CVSYNC with positive CHSYNC, positive CVSYNC with negative CHSYNC, negative CVSYNC with positive CHSYNC and negative CVSYNC with negative CHSYNC. The timing diagrams are showed as below.

Figure 3.3.3-1 Video Interface: Positive CVSYNC with Positive CHSYNC

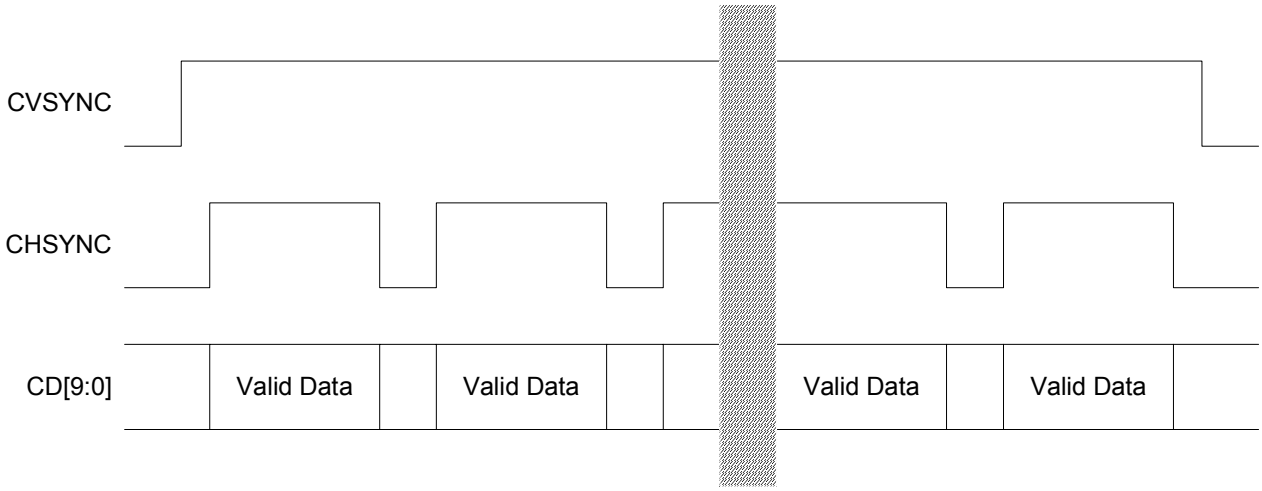


Figure 3.3.3-2 Video Interface: Positive CVSYNC with Negative CHSYNC

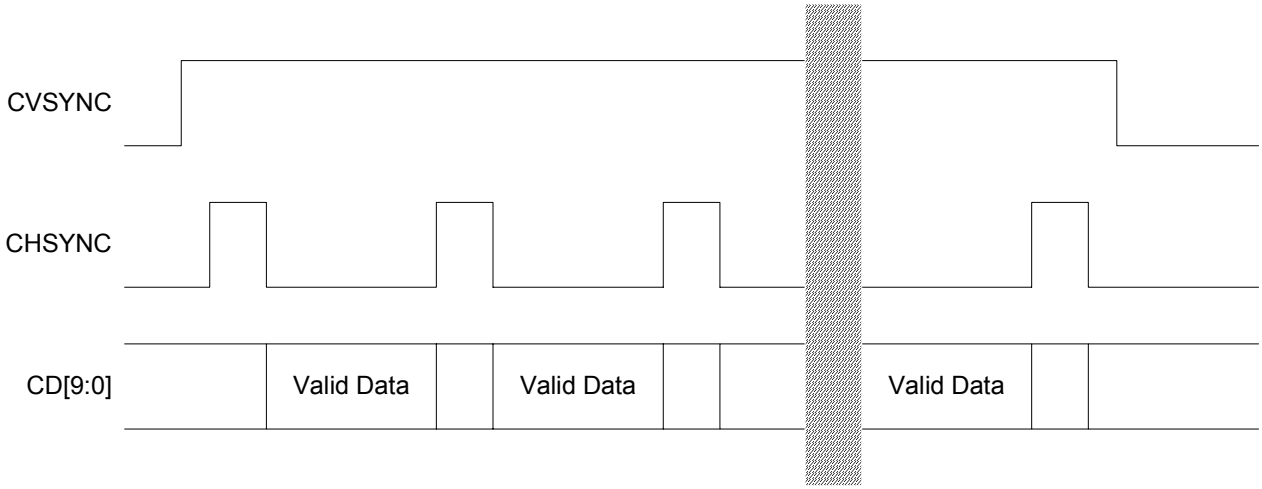


Figure 3.3.3-3 Video Interface: Negative CVSYNC with Positive CHSYNC

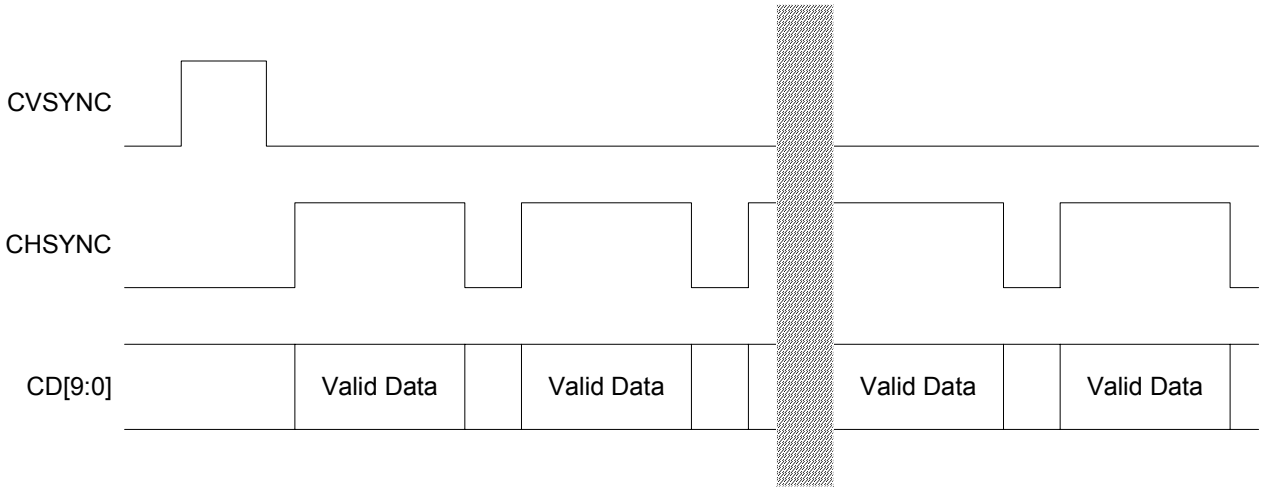
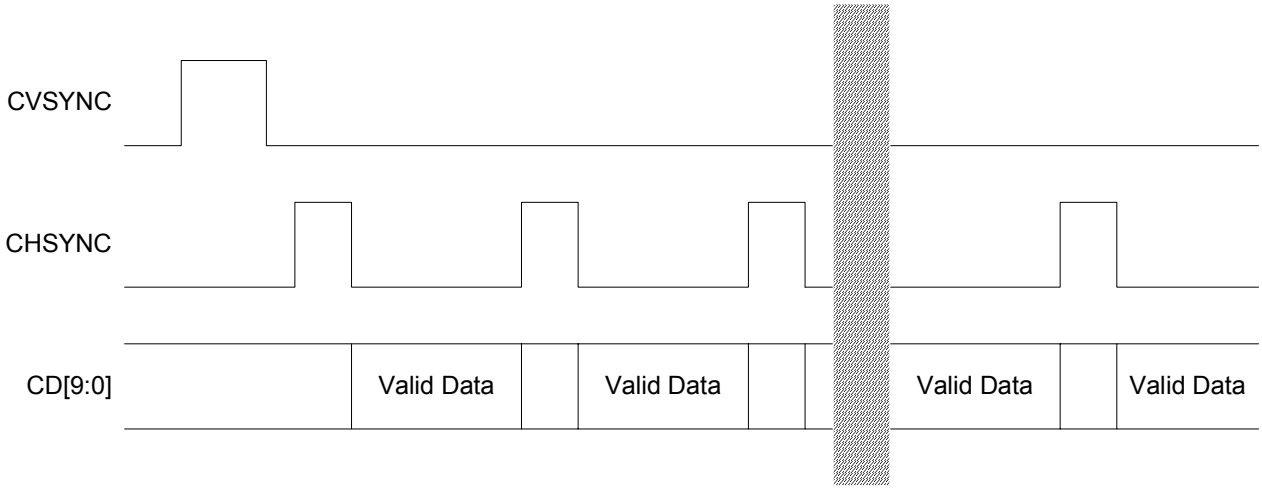
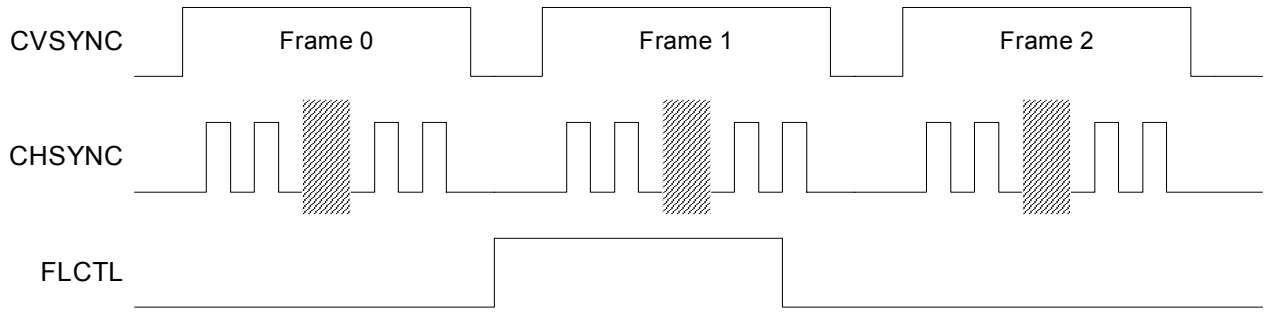


Figure 3.3.3-4 Video Interface: Negative CVSYNC with Negative CHSYNC



The video interface supports a programmable flashlight period. As below example, it can light the flash during frame 1 and take the frame 2 or latter.

Figure 3.3.3-5 Video Interface Flashlight Output



The timing diagram is showed in the Figure 4.2-4. The detail timing description is showed in the Table 4.2-2.

Figure 3.3.3-6 Video Capture Interface Timing Diagram

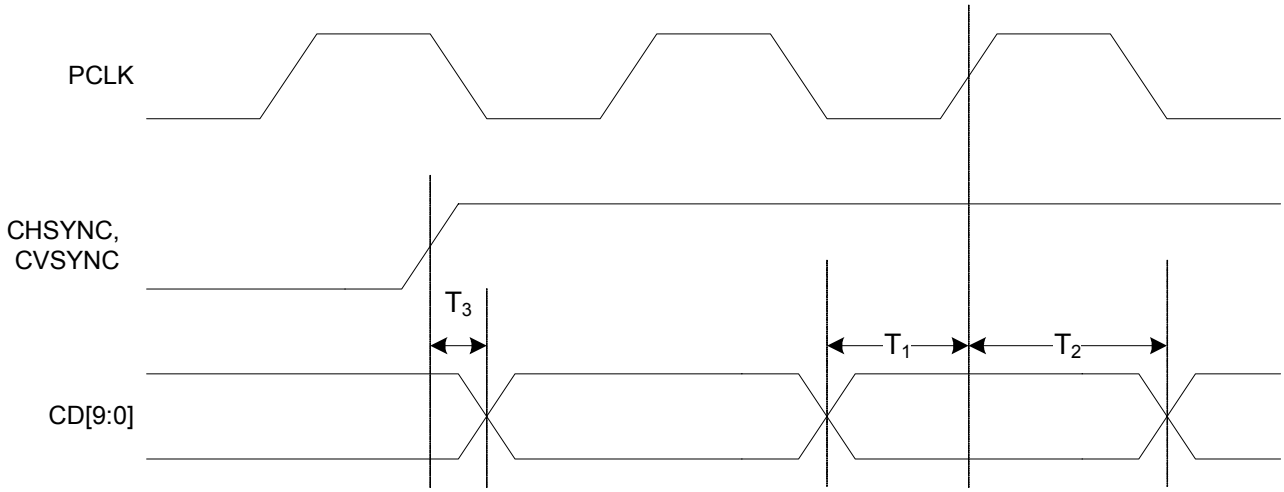


Table 3.3.3-1 Video Interface Timing Table

Symbol	Parameter	Min	Max	Unit
T_1	Pixel data setup time	8	-	ns
T_2	Pixel data hold time	2	-	ns
T_3	SYNC active time before pixel data	0	-	ns

3.4 MMC/SD Interface

3.4.1 General Description

The Glamo 3362 support MMC/SD card for user to storage the 3D games, JPEG image and MPEG-4 movies. It is convenient for user to put these data to the computer or download data from computer. The Glamo 3362 fully compliant with MMCA v3.3 and compliant with low-voltage support and 4 bits data of MMCA v4.0

3.4.2 MMC/SD Protocol

The Glamo 3362 supports both single and multiple block read/write. The figures below illustrate these basic MMC/SD protocol.

Figure 3.4.2-1 MMC/SD Command and Response Protocol

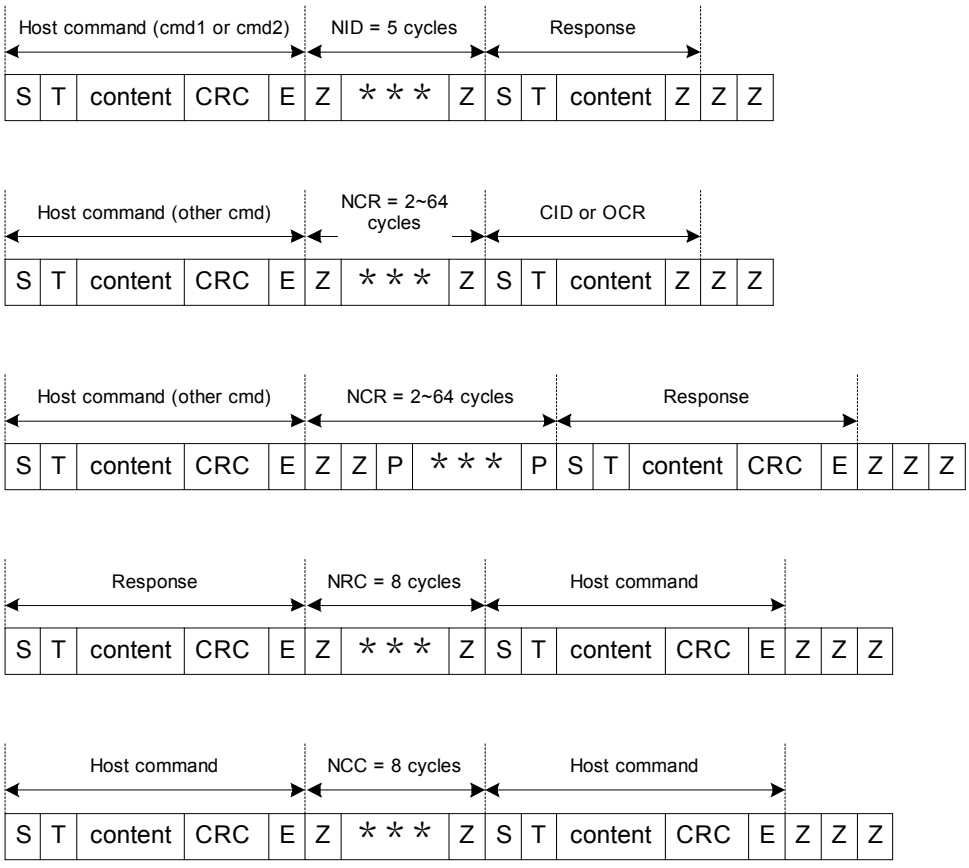


Figure 3.4.2-2 MMC/SD Single Block Read Protocol

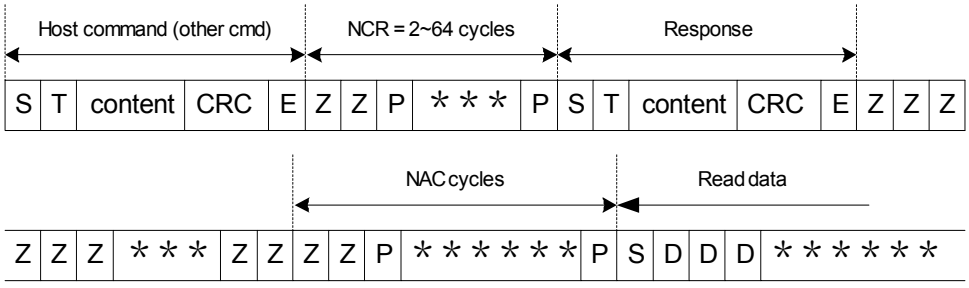


Figure 3.4.2-3 MMC/SD Multiple Block Read Protocol

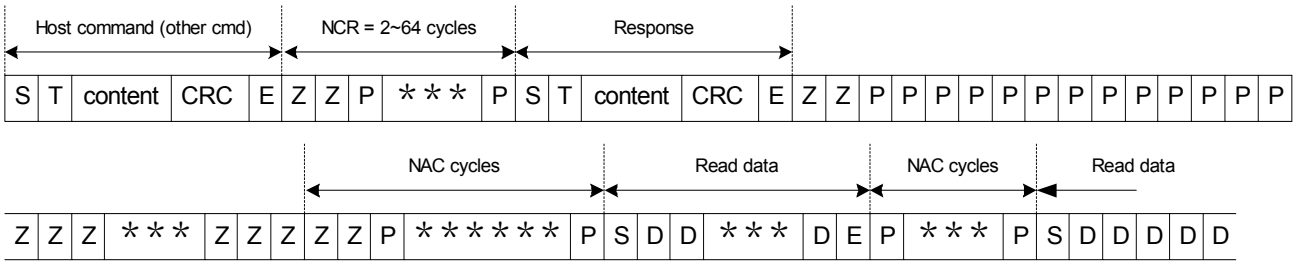


Figure 3.4.2-4 MMC/SD Single Block Write Protocol

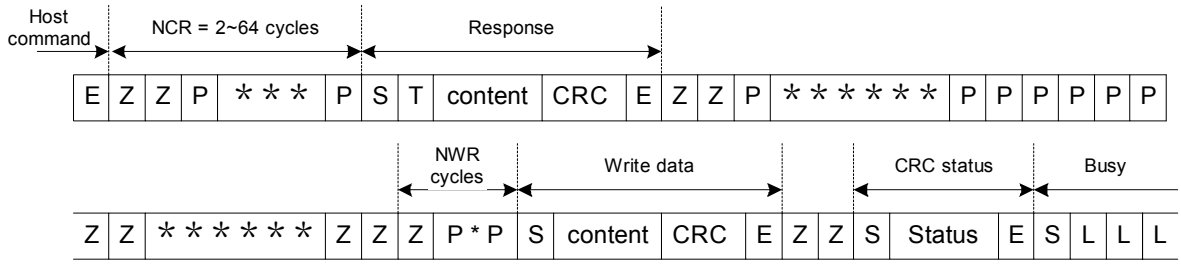
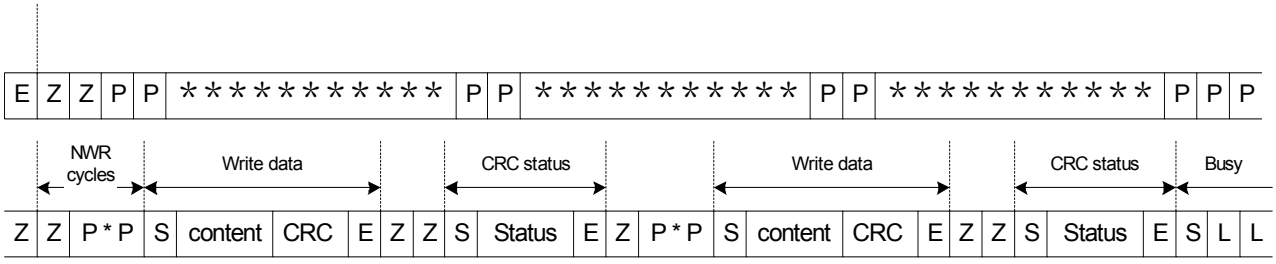


Figure 3.4.2-5 MMC/SD Multiple Block Write Protocol



3.4.3 MMC/SD Interface Timing Diagram

The MMC/SD Interface Timing diagram is showed in the Figure 4.3-6. The detail timing description is showed in the Table 4.3-1.

Figure 3.4.3-1 MMC/SD Interface Timing Diagram

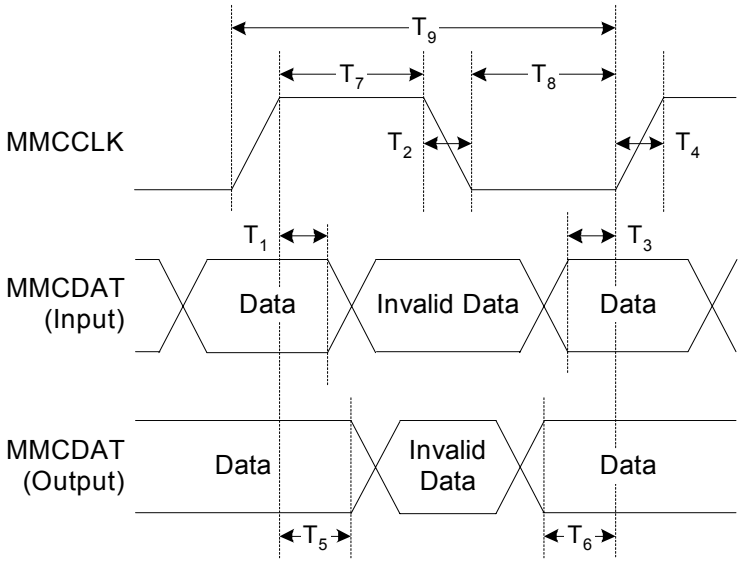


Table 3.4.3-1 MMC/SD Interface Timing Table

Symbol	Parameter	Min	Max	Unit
T_1	Input hold time	5	-	ns
T_2	Clock fall time	-	10	ns
T_3	Input set-up time	5	-	ns
T_4	Clock rise time	-	10	ns
T_5	Output hold time	3	-	ns
T_6	Output set-up time	3	-	ns
T_7	Clock high time	10	-	ns
T_8	Clock low time	10	-	ns
T_9	Clock cycle time	40	-	ns

3.5 PLL Interface

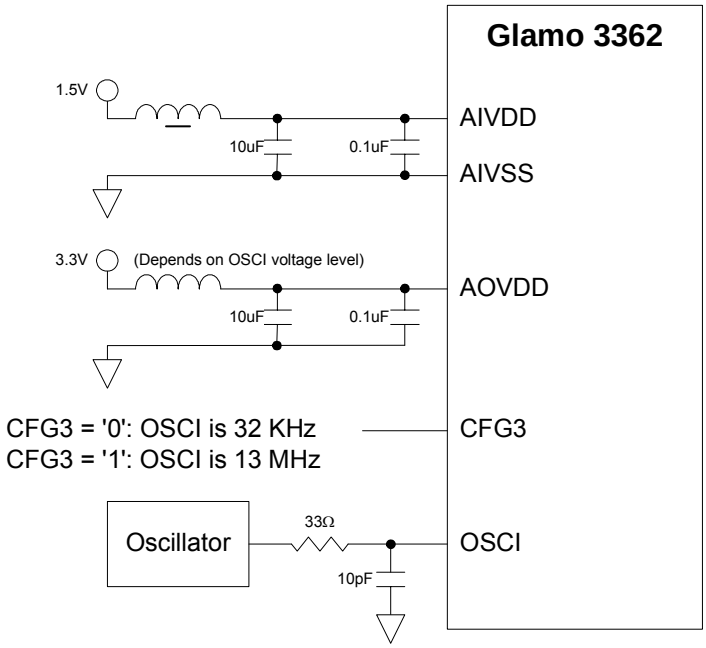
3.5.1 General Description

Glamo 3362 has two clock synthesizers to generate all of the internal clocks. The clock synthesizer can generate wide range of programmable frequencies, one is for up to 60MHz, and the other one is for up to 90MHz. The clock synthesizer accepts 32 KHz or 13 MHz reference clock input, which depends on the trapping value of CFG3. And system can even stop the reference clock after the PLL locked the target frequency and phase for power saving. The locking period is about 5ms. System can resend the reference clock and the PLL will automatically lock the target frequency and phase again. To meet the mobile application, the PLL contains the low power consumption, fast lock period, low clock jitter and flexible programming range features and automatically frequency and phase lock and relock scheme.

3.5.2 PLL Interface Implementation

The figure below illustrates the recommended implementation for PLL interface.

Figure 3.5.2-1 PLL Interface Implementation



3.6 General-Purpose I/O

3.6.1 General Description

The GPIO (general-purpose I/O) signals can be used to control and receive external devices or events. Glamo 3362 support 16 dedicated GPIO. For application flexibility, we divide these DGPIO into three group (group A, LCD and sensor Group). The voltage is set to be the same in the same group. In addition to the DGPIO, there are 21 GPIO pins that are shared with normal pins. Depending on physical usage, these normal pins can be programmed as GPIO pins. All of the GPIO pins are bi-directional; their directions are programmable by setting the internal registers.

3.6.2 GPIO Pins

The following table illustrates the mapping of GPIO pins.

Table 3.7.2-1 Mapping of GPIO Pins

GPIO Pin	Normal Pin
GPIO0	HA20
GPIO1	HA21

GPIO2	HA22
GPIO3	HA23
GPIO4	LCS0#
GPIO5	LCS1#
GPIO6	LDCLK
GPIO7	LDE#
GPIO8	LD16
GPIO9	LD17
GPIO10	LSCK
GPIO11	LSDA
GPIO12	LSA0
GPIO13	CSGPO0
GPIO14	CSGPO1
GPIO15	FLCTL
GPIO16	MMCDAT1
GPIO17	MMCDAT2
GPIO18	MMCDAT3
GPIO19	HADV#

Note: HA20, HA21, HA22, and HA23 can only be programmed as GPIO when using type 4 indirect addressing mode.

4 Electrical and Thermal Characteristics

4.1 Absolute Maximum Ratings

Table 4.1-1 Table of Absolute Maximum Ratings

Parameter	Min.	Max.	Unit
Core power (IVDD)	-0.3	2.0	V
Power for PLL (AIVDD)	-0.3	2.0	V
I/O power (AOVDD, COVDD, HOVDD, LOVDD, MOVDD, TOVDD)	-0.3	4.0	V
Input voltage	-0.3	OVDD+10%	V
Storage temperature	-40	125	^{0}C

NOTE: Violating these above may cause permanent damage on device. Functional operation of this device should be restricted to the conditions described under operating conditions.

4.2 Operating Conditions

Table 4.2-1 Normal Operating Conditions

Symbol	Parameter	Min	Typ	Max	Unit
VDD_{CORE}	Core power (IVDD)	1.7	1.8	1.9	V
VDD_{PLL}	Power for PLL (AIVDD)	1.7	1.8	1.9	V
MOVDD	Memory power	1.71	1.8	1.98	V
OVDD	I/O power (AOVDD, COVDD, HOVDD, LOVDD, MOVDD, TOVDD)	1.71	-	3.6	V
T_A	Operating ambient temperature	-25	-	85	^{0}C
T_J	Operating junction temperature	-25	-	100	^{0}C

4.3 DC Characteristics

Table 4.3-1 DC Characteristics

Symbol	Parameter	Condition	Min	Max	Unit
V_{OL}	Output low voltage	Refer to I_{OL}	OVSS	OVSS+0.4	V
V_{OH}	Output high voltage	Refer to I_{OH}	OVDD-0.4	OVDD	V
I_{OH_S}	Output high current (with maximum driving strength)	OVDD=3.3V, V_{OH}=2.9V	3.03	5.58	mA
		OVDD=2.8V, V_{OH}=2.4V	2.77	5.37	mA
		OVDD=2.5V, V_{OH}=2.1V	2.57	5.17	mA
		OVDD=1.8V, V_{OH}=1.4V	1.91	4.34	mA
I_{OL_S}	Output low current (with maximum driving strength)	OVDD=3.3V, V_{OL}=0.4V	3.60	5.59	mA
		OVDD=2.8V, V_{OL}=0.4V	3.37	5.48	mA
		OVDD=2.5V, V_{OL}=0.4V	3.19	5.37	mA
		OVDD=1.8V, V_{OL}=0.4V	2.51	4.83	mA
I_{OH_W}	Output high current (with minimum driving strength)	OVDD=3.3V, V_{OH}=2.9V	1.20	2.23	mA
		OVDD=2.8V, V_{OH}=2.4V	1.11	2.14	mA
		OVDD=2.5V, V_{OH}=2.1V	1.03	2.07	mA
		OVDD=1.8V, V_{OH}=1.4V	0.76	1.74	mA

I_{OL_W}	Output low current (with minimum driving strength)	OVDD=3.3V, V_{OL}=0.4V	1.44	2.23	mA
		OVDD=2.8V, V_{OL}=0.4V	1.35	2.19	mA
		OVDD=2.5V, V_{OL}=0.4V	1.28	2.14	mA
		OVDD=1.8V, V_{OL}=0.4V	1.00	1.93	mA
I_{OZ}	Tristate leakage current		-	± 5	uA
V_{IL}	Input low voltage	OVDD=3.3V	-	1.35	V
		OVDD=2.8V	-	1.15	V
		OVDD=2.5V	-	1.00	V
		OVDD=1.8V	-	0.70	V
V_{IH}	Input high voltage	OVDD=3.3V	1.95	-	V
		OVDD=2.8V	1.65	-	V
		OVDD=2.5V	1.45	-	V
		OVDD=1.8V	1.05	-	V
I_{IN}	Input leakage current		-	± 1	uA
C_{IN}	Input capacitance		-	3	pF

4.4 AC Characteristics

4.4.1 Reset Timing

Figure 4.4-1 Reset Timing

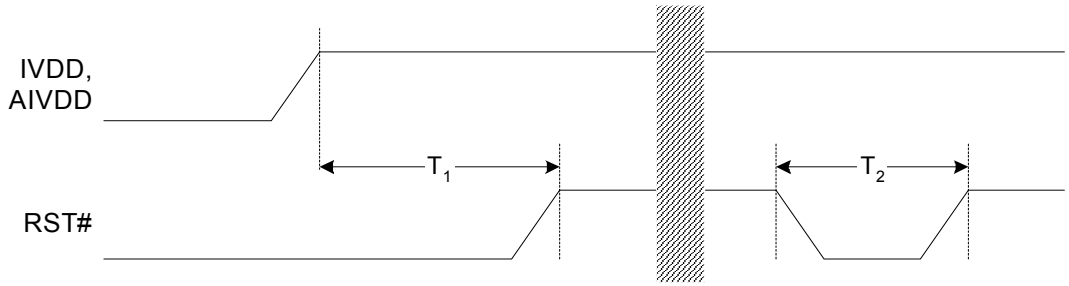


Table 5.4-1 Reset Timing Table

Symbol	Parameter	Min	Max	Unit
T_1	Power valid to reset inactive	1	-	ms
T_2	Minimum reset pulse width	1	-	us

Notes: When the reset process is finished, after 4 ms, you can access the registers.

4.4.2 Power Sequencing

All the I/O power and core power should be turned on or turn off as close as possible. If the power sources of Glamo 3362 can be turned on/off almost at the same time, there will be no specific sequencing requirements.

Figure 4.4.2-1 Power On/Off Sequencing

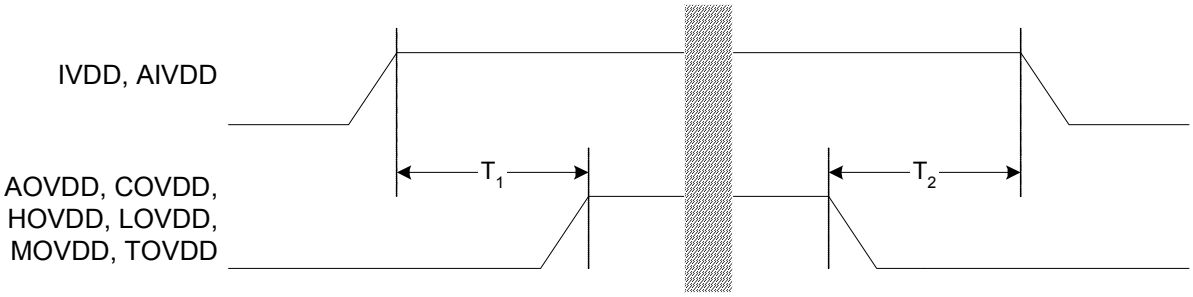
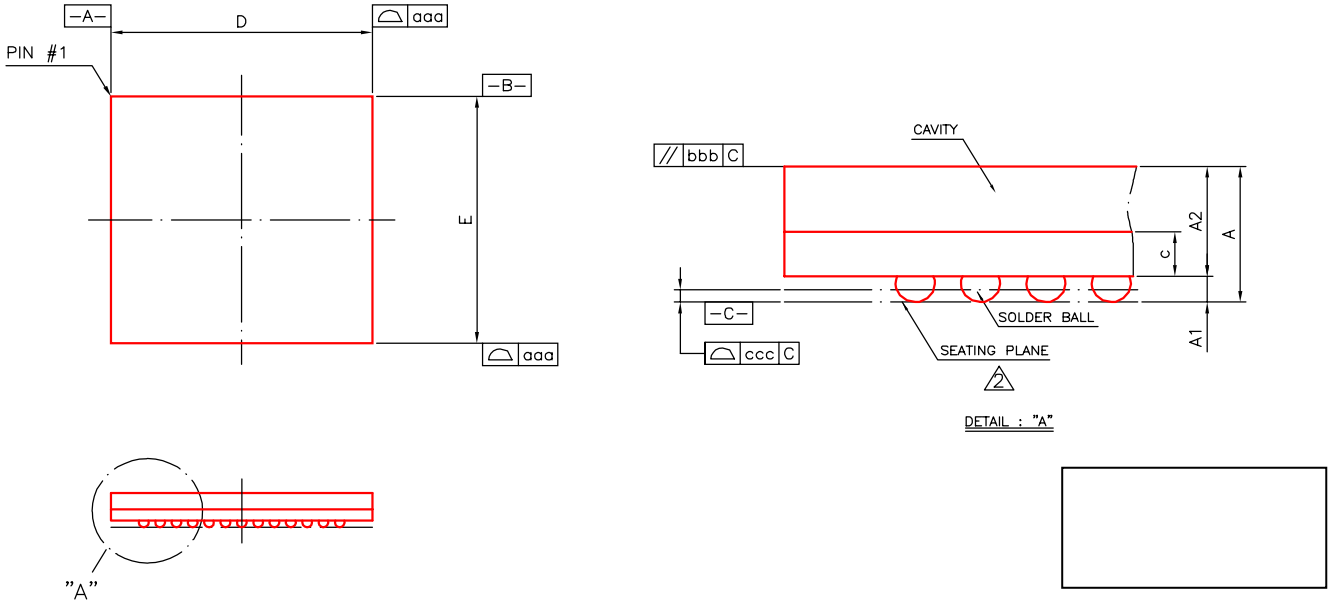


Table 4.4.2-1 Power On/Off Sequencing Table

Symbol	Parameter	Min	Max	Unit
T_1	Core power valid to IO power valid	0	1	us
T_2	IO power invalid to Core power invalid	0	1	us

5 Mechanical Dimension

Figure 5-1 Glamo 3362 8x8x1.3 160 Balls LFBGA Package (Unit: mm)



Bottom View

Top View

Symbol	Dimension in mm			Dimension in Inch		
	MIN	NOM	MAX	MIN	NOM	MAX
A	---	---	1.30	---	---	0.051
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	7.90	8.00	8.10	0.311	0.315	0.319
E	7.90	8.00	8.10	0.311	0.315	0.319
D1	---	6.00	---	---	0.236	---
E1	---	6.00	---	---	0.236	---
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		0.10			0.004	
ddd		0.15			0.006	
eee		0.08			0.003	
MD/ME		13/13			13/13	

6 Memory

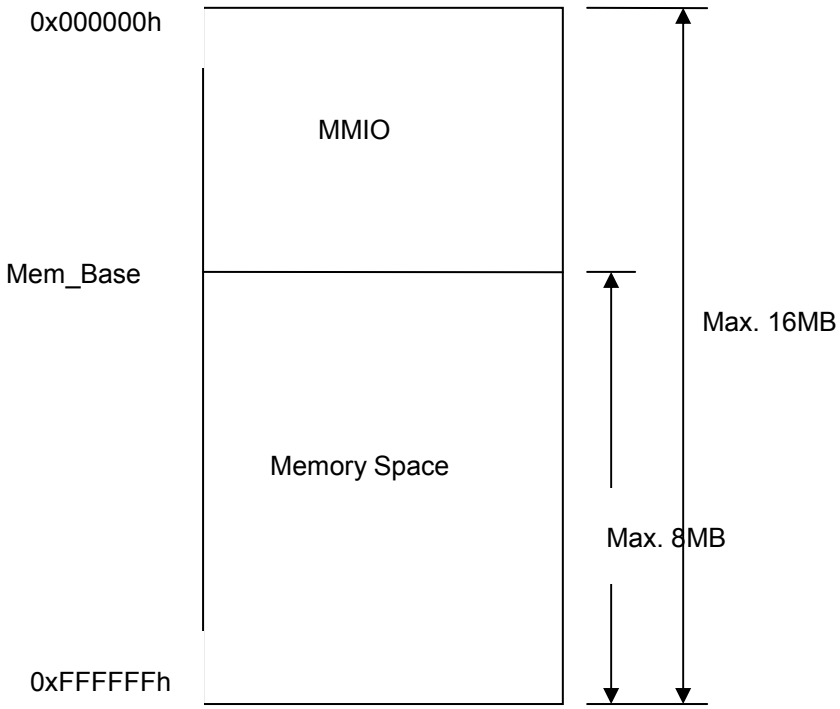
The memory is divided into two parts. One is mapping to the Glamo 3362 internal registers that we call it as Memory Map IO (MMIO) space. And the other is mapping to the stacked memory. The whole memory size is up to 16MBytes and the stacked memory size is maximum 8Mbytes. The physical memory size is depending on the stacked SDRAM or SRAM size.

The starting address of external memory is set on Mem_Base[23:1] which defined in Host Bus Controller Register 0x204h and 0x205h. For example, the default setting in the registers is 0x800000h. It means the starting address of stacked memory is 0x800000h. User can define the space by changing the Mem_Base setting.

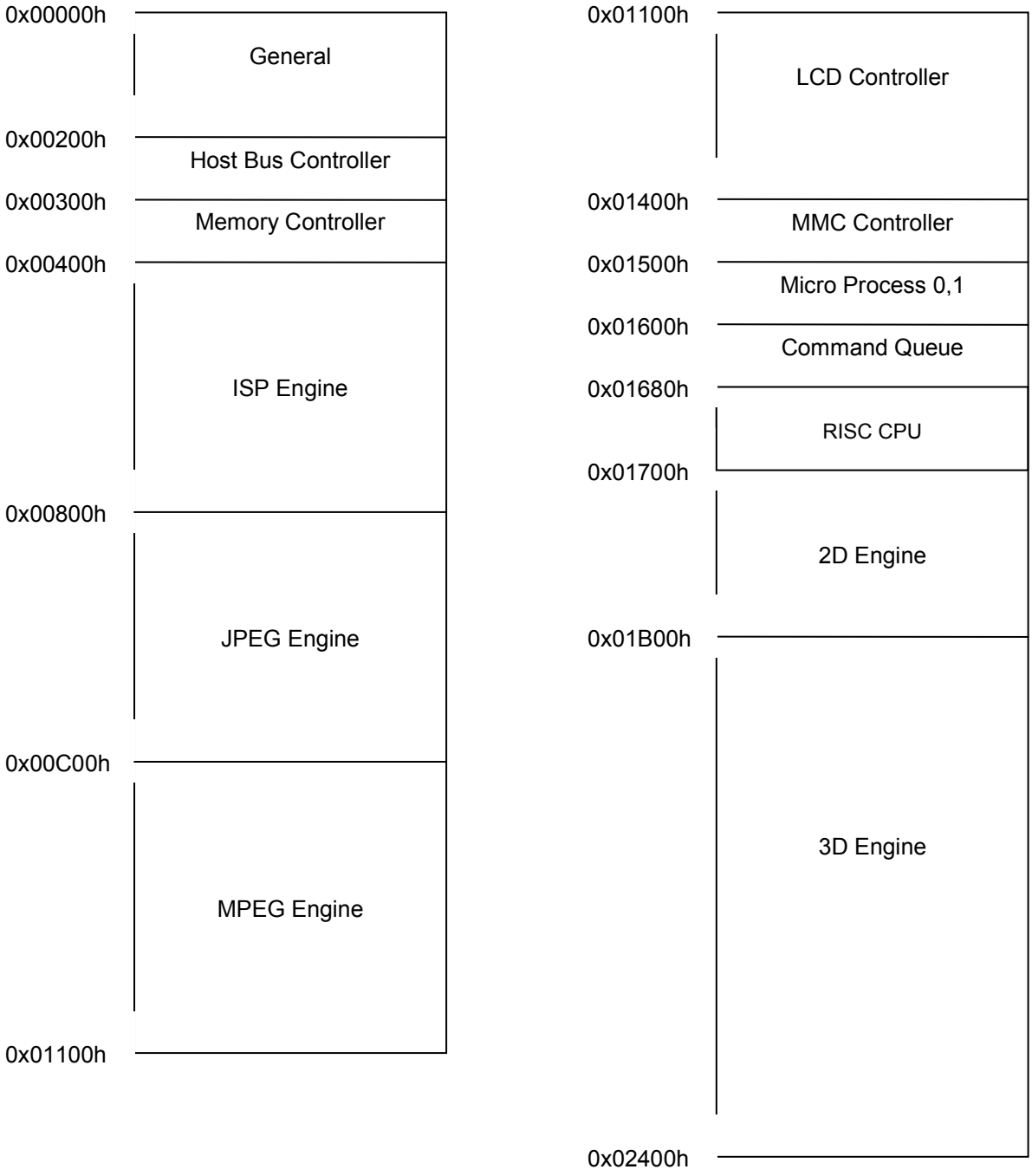
For example, if 2Mbytes SRAM is stacked and Mem_Base = 0x800000h. Then the actual memory space is from 0x800000h to 0x9FFFFFh.

Baseband CPU should set this memory space as non-cacheable space. Otherwise, it will lead to command delay action or data coherence error.

6.1 Memory Map

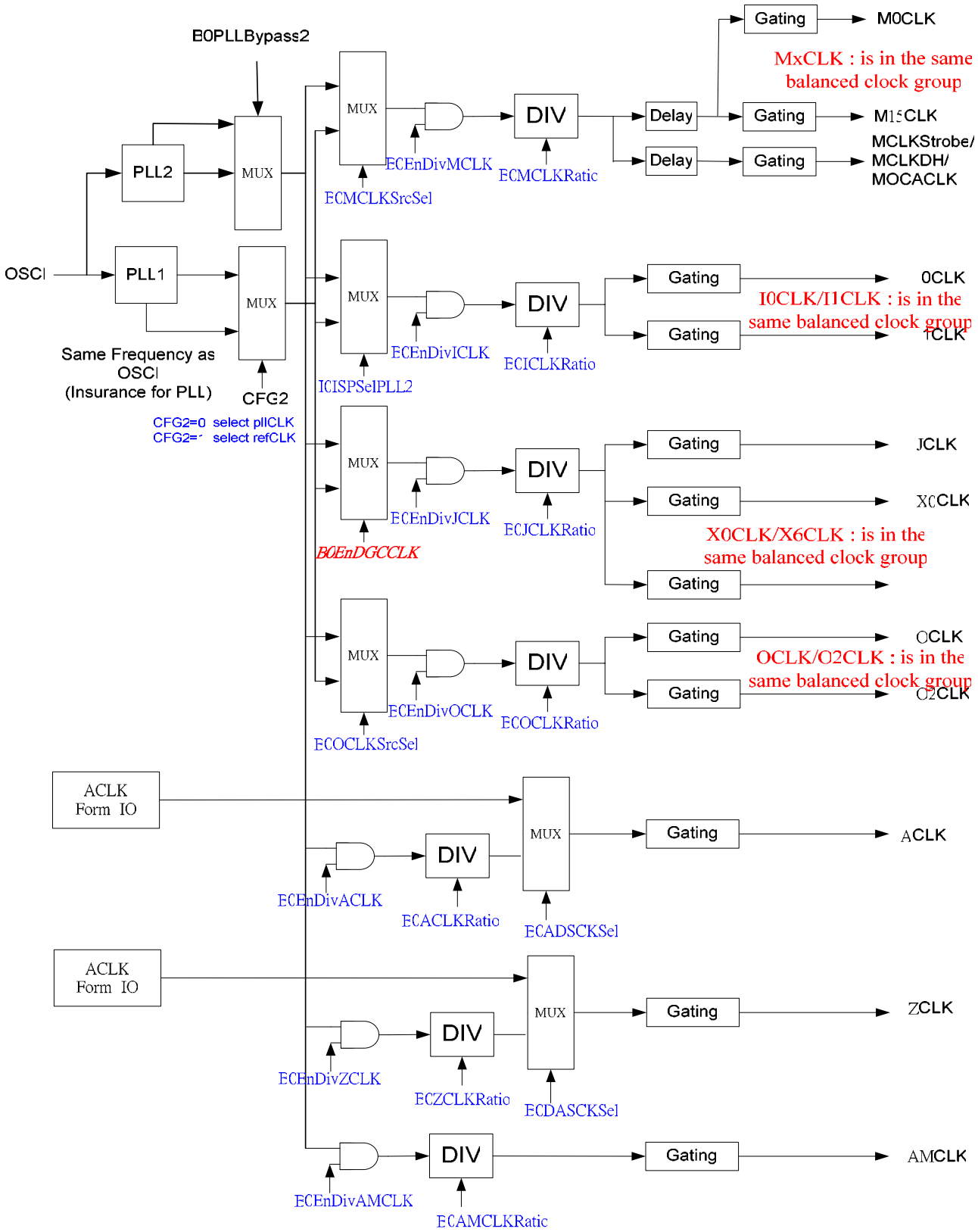


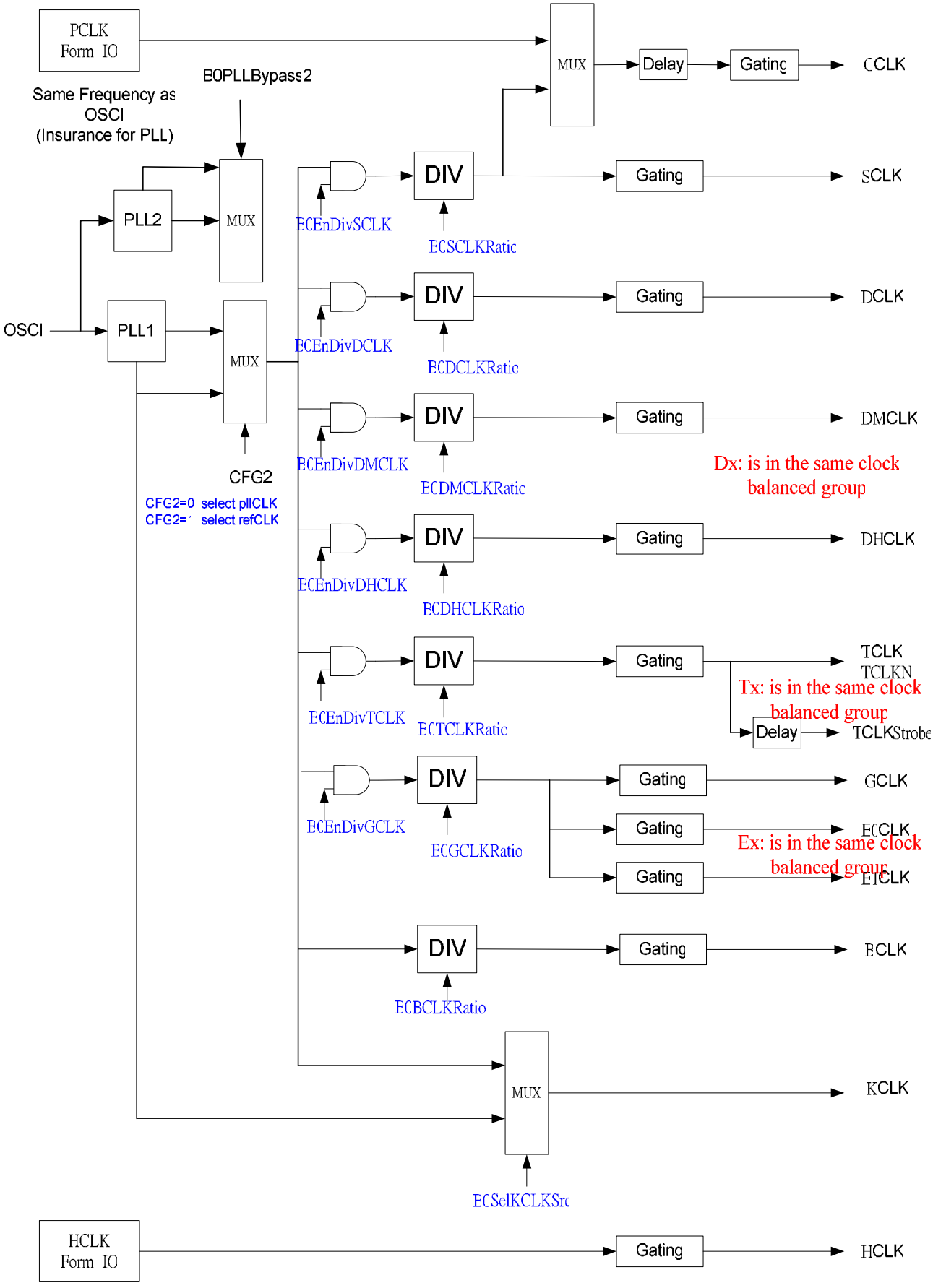
6.2 MMIO Map


0x00000h
General
0x00200h
Host Bus Controller
0x00300h
Memory Controller
0x00400h
ISP Engine
0x00800h
JPEG Engine
0x00C00h
MPEG Engine
0x01100h
0x01100h
LCD Controller
0x01400h
MMC Controller
0x01500h
Micro Process 0,1
0x01600h
Command Queue
0x01680h
RISC CPU
0x01700h
2D Engine
0x01B00h
3D Engine
0x02400h

7 Clocks

7.1 Clock Diagram




PCLK
Form IO
Same Frequency as
OSCI
(Insurance for PLL)
E0PLLBypass2
MUX
Delay
Gating
CCLK
PLL2
MUX
ECEnDivSCLK
DIV
ECSCLKRatic
Gating
SCLK
OSCI
PLL1
MUX
ECEnDivDCLK
DIV
ECDCLKRatic
Gating
DCLK
ECEnDivDMCLK
DIV
ECDMCLKRatic
Gating
DMCLK
Dx: is in the same clock
balanced group
CFG2
CFG2=0 select pllCLK
CFG2=' select refCLK
ECEnDivDHCLK
DIV
ECDHCLKRatio
Gating
DHCLK
ECEnDivTCLK
DIV
ECTCLKRatic
Gating
TCLK
TCLKN
Tx: is in the same clock
balanced group
Delay
TCLKStrobe
ECEnDivGCLK
DIV
ECGCLKRatio
Gating
GCLK
Gating
ECLK
Ex: is in the same clock
balanced group
Gating
EICLK
DIV
Gating
ECBCLKRatio
ECLK
MUX
KCLK
ECSelKCLKSrc
HCLK
Form IO
Gating
HCLK

7.2 Clocks

For power saving, each module has its own clock tree and can be Enable/Disable or divided frequency separately. In General Clock registers 0x0010h~0x0026h define the clock setting of each module.

Host Bus Clock (BCLK)

This clock is used for host bus controller and register banks of all internal modules. It can be divided using BCLK_Ratio. Turn on dynamic gating by En_DG_BCLK for power saving. This clock is always turn on, so there is no enable bit for it.

Image Processor 1 Clock (CCLK)

This clock is used for image signal processor 1. Its clock source is chose by Sel_CCLK_Src. One source is from external input clock PCLK, another source is from SCLK. The clock phase can be programmed by CCLK_Dly. Enable this clock by En_CCLK and turn on dynamic gating by En_DG_CCLK for power saving.

Image Processor 2 Clock (I1CLK)

This clock is used for image signal processor 2. Its clock source is chose by I0ISPSelPLL2. One source is from external input clock PLL1, another source is from PLL2. Set En_Div_ICLK for passing clock to the clock divider. It can be divided using ICLK_Ratio. Enable this clock by En_I1CLK and turn on dynamic gating by En_DG_I1CLK for power saving.

LCD Controller Clock 1 (DCLK)

This clock is used for LCD controller as the pixel clock. Set En_Div_DCLK for passing clock to the clock divider. It can be divided using DCLK_Ratio. Enable this clock by En_DCLK and turn on dynamic gating by En_DG_DCLK for power saving.

LCD Controller Clock 2 (DHCLK)

This clock is used for LCD controller. Set En_Div_DHCLK for passing clock to the clock divider. It can be divided using DHCLK_Ratio. Enable this clock by En_DHCLK. It does not have dynamic gating ability.

LCD Controller Clock 3 (DMCLK)

This clock is used for LCD controller. Set En_Div_DMCLK for passing clock to the clock divider. It can be divided using DMCLK_Ratio. Enable this clock by En_DMCLK and turn on dynamic gating by En_DG_DMCLK for power saving.

3D Engine Clock 1 (E0CLK)

This clock is used for 3D engine. It shares the same clock divider with E0CLK and GCLK. Set En_Div_GCLK for passing clock to the clock divider. It can be divided using GCLK_Ratio. Enable this clock by En_E0CLK and turn on dynamic gating by En_DG_E0CLK for power saving.

3D Engine Clock 2 (E1CLK)

This clock is used for 3D engine. It shares the same clock divider with E0CLK and GCLK. Set En_Div_GCLK for passing clock to the clock divider. It can be divided using GCLK_Ratio. Enable this clock by En_E1CLK and turn on dynamic gating by En_DG_E1CLK for power saving.

2D Engine Clock (GCLK)

This clock is used for 2D engine. It shares the same clock divider with E0CLK and E1CLK. Set

En_Div_GCLK for passing clock to the clock divider. It can be divided using GCLK_Ratio. Enable this clock by En_GCLK and turn on dynamic gating by En_DG_GCLK for power saving.

Host Bus Clock for iBurst Mode (HCLK)

This clock is used for host bus controller in iBurst mode. When CFG[1:0]=10, the CPU bus uses Type 3 iBurst mode. In this mode the clock comes from the pin HA2.

Micro Processor 1 Clock (I0CLK)

This clock is used for micro processor 0. Its clock source is chose by I0ISPSelPLL2. One source is from external input clock PLL1, another source is from PLL2. It shares the same clock divider with I0CLKand I1CLK. Set En_Div_ICLK for passing clock to the clock divider. It can be divided using ICLK_Ratio. Enable this clock by En_I0CLK and turn on dynamic gating by En_DG_I0CLK for power saving.

Micro Processor 2 Clock (X5CLK)

This clock is used for image signal processor 2. It shares the same clock divider with JCLK and X0CLK~X6CLK. Set En_Div_JCLK for passing clock to the clock divider. It can be divided using JCLK_Ratio. Enable this clock by En_X5CLK and turn on dynamic gating by En_DG_X5CLK for power saving.

JPEG Engine Clock (JCLK)

This clock is used for JPEG engine. Its clock source is chose by B0EnDGCCLK. One source is from external input clock PLL1, another source is from PLL2. It shares the same clock divider with I0CLK, I1CLK and X0CLK~X5CLK. Set En_Div_JCLK for passing clock to the clock divider. It can be divided using JCLK_Ratio. Enable this clock by En_JCLK and turn on dynamic gating by En_DG_JCLK for power saving.

MPEG Engine Clock 1 (X0CLK)

This clock is used for MPEG engine. Its clock source is chose by B0EnDGCCLK. One source is from external input clock PLL1, another source is from PLL2. It shares the same clock divider with JCLK and X0CLK~X6CLK. Set En_Div_JCLK for passing clock to the clock divider. It can be divided using JCLK_Ratio. Enable this clock by En_X0CLK and turn on dynamic gating by En_DG_X0CLK for power saving.

MPEG Engine Clock 2 (X1CLK)

This clock is used for MPEG engine. Its clock source is chose by B0EnDGCCLK. One source is from external input clock PLL1, another source is from PLL2. It shares the same clock divider with JCLK and X0CLK~X6CLK. Set En_Div_JCLK for passing clock to the clock divider. It can be divided using JCLK_Ratio. Enable this clock by En_X1CLK and turn on dynamic gating by En_DG_X1CLK for power saving.

MPEG Engine Clock 3 (X2CLK)

This clock is used for MPEG engine. Its clock source is chose by B0EnDGCCLK. One source is from external input clock PLL1, another source is from PLL2. It shares the same clock divider with JCLK and X0CLK~X6CLK. Set En_Div_JCLK for passing clock to the clock divider. It can be divided using JCLK_Ratio. Enable this clock by En_X2CLK and turn on dynamic gating by En_DG_X2CLK for power saving.

MPEG Engine Clock 4 (X3CLK)

This clock is used for MPEG engine. Its clock source is chose by B0EnDGCCLK. One source is from external input clock PLL1, another source is from PLL2. It shares the same clock divider with JCLK and X0CLK~X6CLK. Set En_Div_JCLK for passing clock to the clock divider. It can be divided using JCLK_Ratio. Enable this clock by En_X3CLK and turn on dynamic gating by En_DG_X3CLK for power saving.

MPEG Engine Clock 5 (X4CLK)

This clock is used for MPEG engine. Its clock source is chose by B0EnDGCCLK. One source is from external input clock PLL1, another source is from PLL2. It shares the same clock divider with JCLK and X0CLK~X6CLK. Set En_Div_JCLK for passing clock to the clock divider. It can be divided using JCLK_Ratio. Enable this clock by En_X4CLK and turn on dynamic gating by En_DG_X4CLK for power saving.

MPEG Engine Clock 6 (X5CLK)

This clock is used for MPEG engine. Its clock source is chose by B0EnDGCCLK. One source is from external input clock PLL1, another source is from PLL2. It shares the same clock divider with JCLK and X0CLK~X6CLK. Set En_Div_JCLK for passing clock to the clock divider. It can be divided using JCLK_Ratio. Enable this clock by En_X5CLK and turn on dynamic gating by En_DG_X5CLK for power saving.

MPEG Engine Clock 7 (X6CLK)

This clock is used for MPEG engine. Its clock source is chose by B0EnDGCCLK. One source is from external input clock PLL1, another source is from PLL2. It shares the same clock divider with JCLK and X0CLK~X6CLK. Set En_Div_JCLK for passing clock to the clock divider. It can be divided using JCLK_Ratio. Enable this clock by En_X6CLK and turn on dynamic gating by En_DG_X6CLK for power saving.

System Timer Clock (KCLK)

This clock is for Accurate timer. This clock may come from two source, one is from external input clock OSCI and the other is from PLL. The clock source is chosed by Sel_KCLK_Src and enable this clock by En_KCLK.

Sensor Clock (SCLK)

This clock is used for sensor interface. Set En_Div_SCLK for passing clock to the clock divider. It can be divided using SCLK_Ratio. Enable this clock by En_SCLK. It does not have dynamic gating ability.

MMC Controller Clock (TCLK)

This clock is used for MMC controller. Set En_Div_TCLK for passing clock to the clock divider. It can be divided using TCLK_Ratio. Enable this clock by En_TCLK and turn on dynamic gating by En_DG_TCLK for power saving.

Memory Clock in Host Bus Controller (M0CLK)

This clock is used for host bus controller. Its clock source is chose by B0MCLKSrcSel. One source is from external input clock PLL1, another source is from PLL2. It shares the same clock divider with M0CLK~M15CLK. Set En_Div_MCLK for passing clock to the clock divider. It can be divided using

MCLK_Ratio. Enable this clock by En_M0CLK and turn on dynamic gating by En_DG_M0CLK for power saving.

Memory Clock in Memory Controller (M1CLK)

This clock is used for memory controller. Its clock source is chose by B0MCLKSrcSel. One source is from external input clock PLL1, another source is from PLL2. It shares the same clock divider with M0CLK~M15CLK. Set En_Div_MCLK for passing clock to the clock divider. It can be divided using MCLK_Ratio. Enable this clock by En_M1CLK and turn on dynamic gating by En_DG_M1CLK for power saving.

Memory Clock in ISP Engine (M2CLK)

This clock is used for image signal processor. Its clock source is chose by B0MCLKSrcSel. One source is from external input clock PLL1, another source is from PLL2. It shares the same clock divider with M0CLK~M15CLK. Set En_Div_MCLK for passing clock to the clock divider. It can be divided using MCLK_Ratio. Enable this clock by En_M2CLK and turn on dynamic gating by En_DG_M2CLK for power saving.

Memory Clock in JPEG Engine (M3CLK)

This clock is used for JPEG engine. Its clock source is chose by B0MCLKSrcSel. One source is from external input clock PLL1, another source is from PLL2. It shares the same clock divider with M0CLK~M15CLK. Set En_Div_MCLK for passing clock to the clock divider. It can be divided using MCLK_Ratio. Enable this clock by En_M3CLK and turn on dynamic gating by En_DG_M3CLK for power saving.

Memory Clock in MPEG Engine (M4CLK)

This clock is used for MPEG engine. Its clock source is chose by B0MCLKSrcSel. One source is from external input clock PLL1, another source is from PLL2. It shares the same clock divider with M0CLK~M15CLK. Set En_Div_MCLK for passing clock to the clock divider. It can be divided using MCLK_Ratio. Enable this clock by En_M4CLK and turn on dynamic gating by En_DG_M4CLK for power saving.

Memory Clock in LCD Controller (M5CLK)

This clock is used for LCD controller. Its clock source is chose by B0MCLKSrcSel. One source is from external input clock PLL1, another source is from PLL2. It shares the same clock divider with M0CLK~M15CLK. Set En_Div_MCLK for passing clock to the clock divider. It can be divided using MCLK_Ratio. Enable this clock by En_M5CLK and turn on dynamic gating by En_DG_M5CLK for power saving.

Memory Clock in Command Queue Controller (M6CLK)

This clock is used for command queue. Its clock source is chose by B0MCLKSrcSel. One source is from external input clock PLL1, another source is from PLL2. It shares the same clock divider with M0CLK~M15CLK. Set En_Div_MCLK for passing clock to the clock divider. It can be divided using MCLK_Ratio. Enable this clock by En_M6CLK and turn on dynamic gating by En_DG_M6CLK for power saving.

Memory Clock in 2D Engine (M7CLK)

This clock is used for 2D engine. Its clock source is chose by B0MCLKSrcSel. One source is from external input clock PLL1, another source is from PLL2. It shares the same clock divider with M0CLK~M15CLK. Set En_Div_MCLK for passing clock to the clock divider. It can be divided using MCLK_Ratio. Enable this clock by En_M7CLK and turn on dynamic gating by En_DG_M7CLK for power saving.

Memory Clock in 3D Engine (M8CLK)

This clock is used for 3D engine. Its clock source is chose by B0MCLKSrcSel. One source is from external input clock PLL1, another source is from PLL2. It shares the same clock divider with M0CLK~M15CLK. Set En_Div_MCLK for passing clock to the clock divider. It can be divided using MCLK_Ratio. Enable this clock by En_M8CLK and turn on dynamic gating by En_DG_M8CLK for power saving.

Memory Clock in MMC Engine (M9CLK)

This clock is used for MMC controller. Its clock source is chose by B0MCLKSrcSel. One source is from external input clock PLL1, another source is from PLL2. It shares the same clock divider with M0CLK~M15CLK. Set En_Div_MCLK for passing clock to the clock divider. It can be divided using MCLK_Ratio. Enable this clock by En_M9CLK and turn on dynamic gating by En_DG_M9CLK for power saving.

Memory Clock in Micro Processor 1 (M10CLK)

This clock is used for Micro Processor 1. Its clock source is chose by B0MCLKSrcSel. One source is from external input clock PLL1, another source is from PLL2. It shares the same clock divider with M0CLK~M15CLK. Set En_Div_MCLK for passing clock to the clock divider. It can be divided using MCLK_Ratio. Enable this clock by En_M10CLK and turn on dynamic gating by En_DG_M10CLK for power saving.

Memory Clock to Sample Write Data (MCLKDH)

This clock is used for adjusting the write data timing of the memory interface. The clock phase can be programmed by MCLKDH_Dly. Enable this clock by En_MCLKDH and turn on dynamic gating by En_DG_MCLKDH for power saving.

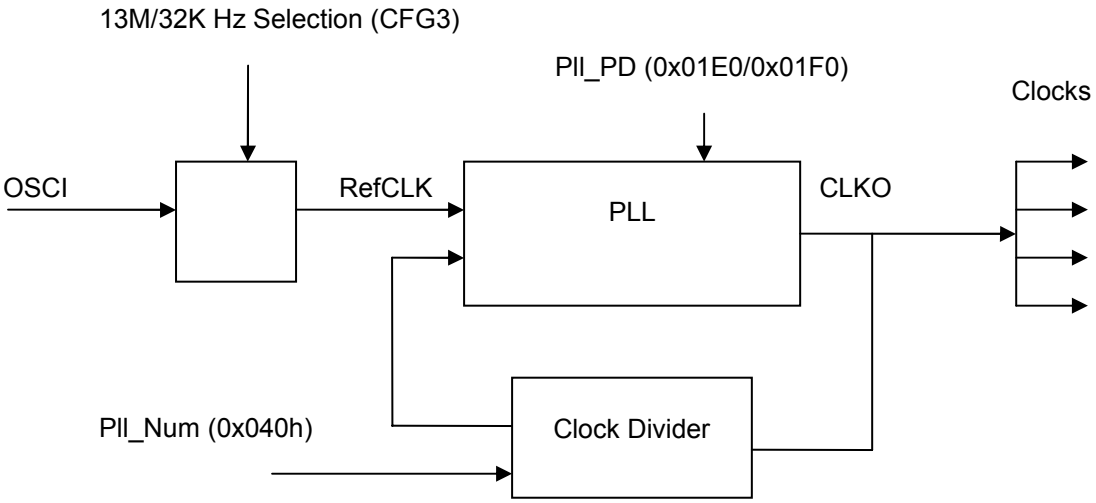
Memory Clock to Strobe Read Data (MCLKStrobe)

This clock is used for adjusting the read data timing of the memory interface. The clock phase can be programmed by MCLKStrobe_Dly. Enable this clock by En_MCLKStrobe and turn on dynamic gating by En_DG_MCLKStrobe for power saving.

Memory Clock to SDRAM (MOCACLK)

This clock is used for memory interface as the output clock to SDRAM. The clock phase can be programmed by MOCACLK_Dly. Enable this clock by En_MOCACLK.

7.3 Phase Lock Loop (PLL)



Glamo 3362 has two clock synthesizers to generate all of the internal clocks. The clock synthesizers can generate wide range of programmable frequencies. It can provide 1 MHz to 90 MHz flexible working frequency. The clock synthesizer accepts 32 KHz or 13 MHz reference clock input, which depends on the trapping value of CFG3 (0:32 KHz, 1:13 MHz). And system can even stop the reference clock after the PLL has locked the target frequency and phase for power saving. The period that system would need to lock the target frequency is about 5ms. System can resend the reference clock and the PLLs will automatically lock the target frequency and phase again. To meet the mobile application requirement, the PLLs have low power consumption, fast lock period, low clock jitter, flexible programming range, automatically frequency, phase lock and relock scheme. The frequency setting for PLL1 is defined in General register 0x0040h. In deep power saving mode, the PLL1 can be power down by setting the Pll_PD to 1 by writing 0x01E0h address with any data. And write 0x01F0h address to return to normal operation. The frequency setting for PLL2 is defined in General register 0x0044h. In deep power saving mode, the PLL2 can be power down by setting the Pll_PD2 to 1 by writing 0x044h D[13]. And disable 0x0044h D[13] to return to normal operation.

8 Power Management

For efficiency control the power consumption, to control the clock gating and divider is very important task for power management. The following table shows the suggestions of clock gating and divider for each operation mode. In addition to the dynamic clock gating control for each module, the Glamo 3362 embeds one hardware clock gating control.

	BCLK, M0CLK	CCLK, I0,I1CLK, M2CLK, M15CLK	DCLK, DHCLK, DMCLK, M5CLK	E0CLK, E1CLK, M6CLK, M8CLK	GCLK, M6CLK, M7CLK	JCLK, M3CLK	X0,X1,X2,X3,X5CLK+M4CLK	X0,X2,X4CLK+M4CLK	SCLK	TCLK, M9CLK	M1CLK	PLL	OCLK, M11~12CLK,ZCLK,AMCLK	OCLK, M13~14CLK,ACLK,AMCLK
HW Dynamic Clock Gating	V	-	-	V	V	-	V	V	-	-	V	-		
Deep Power Down Mode	-	-	-	-	-	-	-	-	-	-	-	-		
LCD Bypass Mode	-	-	-	-	-	-	-	-	-	-	-	-		
Standby Mode	-	-	-	-	-	-	-	-	-	-	-	-		
Partial Display Mode	V	-	V	-	-	-	-	-	-	-	-	V		
Full Static Display	V	-	V	-	-	-	-	-	-	-	V	V		
2D Application	V	-	V	-	V	-	-	-	-	-	V	V		
3D Application	V	-	V	V	-	-	-	-	-	-	V	V		
Preview Mode	V	V	V	-	-	-	-	-	V	-	V	V		
SnapShot Mode	V	V	V	-	-	V	-	-	V	-	V	V		
Video Recording Mode	V	V	V	-	-	-	V	-	V	-	V	V		
Image Viewing Mode	V	V	V	-	-	V	-	-	-	-	V	V		

Video Playback Mode	V	V	V	-	-	-	-	V	-	-	V	V
MMC/SD Access	V	-	V	-	-	-	-	-	-	V	V	V

Note:

1. HW Dynamic Clock Gating means Glamo 3362 supports dynamic clock gating architecture. Hardware module will monitor the engine status. If no new command is received and further staying in idle, the engine will automatically turn off the related clocks. But once any new command is received, the engine will enable the clock in next cycle and further enter normal operation.
2. In this table, we assume the non-memory LCD module is used and Glamo 3362 takes the LCD refresh task.
3. In deep power-down Mode, Glamo 3362 only consumes leakage current. After entering this mode, the data inside the stacked memory will not be guaranteed to keep well.
4. In Standby Mode, all the clocks and PLLs are turned off. After entering this mode, the data inside the stacked memory will be well kept.

9 Function Description

9.1 LCD Display

9.1.1 Display Data Format

Glamo 3362 supports kinds of data formats, and the LCD display related register is defined in LCD Register 0x1104h. Follow explains each data format.

6bits Mode RGB666 (RGB Interface Only)

Table 6bits Mode RGB666

Cycle	0	1	2	……..	3n	3n+1	3n+2
D5	R_{05}	G_{05}	B_{05}		R_{n5}	G_{n5}	B_{n5}
D4	R_{04}	G_{04}	B_{04}		R_{n4}	G_{n4}	B_{n4}
D3	R_{03}	G_{03}	B_{03}		R_{n3}	G_{n3}	B_{n3}
D2	R_{02}	G_{02}	B_{02}		R_{n2}	G_{n2}	B_{n2}
D1	R_{01}	G_{01}	B_{01}		R_{n1}	G_{n1}	B_{n1}
D0	R_{00}	G_{00}	B_{00}		R_{n0}	G_{n0}	B_{n0}

8bits Mode RGB332 (CPU Interface Only)

Table 8bits Mode RG332

Cycle	0	1	……..	n
D7	R_{02}	R_{12}		R_{n2}
D6	R_{01}	R_{11}		R_{n1}
D5	R_{00}	R_{10}		R_{n0}
D4	G_{02}	G_{12}		G_{n2}
D3	G_{01}	G_{11}		G_{n1}
D2	G_{00}	G_{10}		G_{n0}
D1	B_{01}	B_{11}		B_{n1}
D0	B_{00}	B_{10}		B_{n0}

8bits Mode RGB444 (CPU Interface Only)

Table 8bits Mode RGB444 (XR_GB)

Cycle	0	1	……..	2n	2n+1
D7	X	G03		X	Gn3
D6	X	G02		X	Gn2
D5	X	G01		X	Gn1
D4	X	G00		X	Gn0
D3	R03	B03		Rn3	Bn3
D2	R02	B02		Rn2	Bn2
D1	R01	B01		Rn1	Bn1
D0	R00	B00		Rn0	Bn0

Table 8bits Mode RGB444 (RX_GB)

Cycle	0	1	……..	2n	2n+1
D7	R03	G03		Rn3	Gn3
D6	R02	G02		Rn2	Gn2
D5	R01	G01		Rn1	Gn1
D4	R00	G00		Rn0	Gn0
D3	X	B03		X	Bn3
D2	X	B02		X	Bn2
D1	X	B01		X	Bn1
D0	X	B00		X	Bn0

Table 8bits Mode RGB444 (RG_BX)

Cycle	0	1	……..	2n	2n+1
D7	R03	B03		Rn3	Bn3
D6	R02	B02		Rn2	Bn2
D5	R01	B01		Rn1	Bn1

D4	R0 0	B0 0		Rn0	Bn0
D3	G0 3	X		Gn3	X
D2	G0 2	X		Gn2	X
D1	G0 1	X		Gn1	X
D0	G0 0	X		Gn0	X

Table 8bits Mode RGB444 (RG_XB)

Cycle	0	1	……..	2n	2n+1
D7	R0 3	X		Rn3	X
D6	R0 2	X		Rn2	X
D5	R0 1	X		Rn1	X
D4	R0 0	X		Rn0	X
D3	G0 3	B0 3		Gn3	Bn3
D2	G0 2	B0 2		Gn2	Bn2
D1	G0 1	B0 1		Gn1	Bn1
D0	G0 0	B0 0		Gn0	Bn0

Table 8bits Mode RGB444 (RG_BR_GB)

Cycle	0	1	2	……..	3n/2	3n/2+1	3n/2+2
D7	R0 3	B0 3	G1 3		Rn3	Bn3	Gn+1 3
D6	R0 2	B0 2	G1 2		Rn2	Bn2	Gn+1 2
D5	R0 1	B0 1	G1 1		Rn1	Bn1	Gn+1 1
D4	R0 0	B0 0	G1 0		Rn0	Bn0	Gn+1 0
D3	G0 3	R1 3	B1 3		Gn3	Rn+1 3	Bn+1 3
D2	G0 2	R1 2	B1 2		Gn2	Rn+1 2	Bn+1 2
D1	G0 1	R1 1	B1 1		Gn1	Rn+1 1	Bn+1 1
D0	G0 0	R1 0	B1 0		Gn0	Rn+1 0	Bn+1 0

8bits Mode RGB565 (CPU Interface Only)

Table 8bits Mode RGB565

Cycle	0	1	……..	2n	2n+1
D7	R04	G02		Rn4	Gn2
D6	R03	G01		Rn3	Gn1
D5	R02	G00		Rn2	Gn0
D4	R01	B04		Rn1	Bn4
D3	R00	B03		Rn0	Bn3
D2	G05	B02		Gn5	Bn2
D1	G04	B01		Gn4	Bn1
D0	G03	B00		Gn3	Bn0

8bits Mode RGB666 (CPU Interface Only)

Table 8bits Mode RGB666 (XR_XG_XB)

Cycle	0	1	2	……..	3n	3n+1	3n+2
D7	X	X	X	X	X	X	X
D8	X	X	X	X	X	X	X
D5	R05	G05	B05		Rn5	Gn5	Bn5
D4	R04	G04	B04		Rn4	Gn4	Bn4
D3	R03	G03	B03		Rn3	Gn3	Bn3
D2	R02	G02	B02		Rn2	Gn2	Bn2
D1	R01	G01	B01		Rn1	Gn1	Bn1
D0	R00	G00	B00		Rn0	Gn0	Bn0

Table 8bits Mode RGB666 (RX_GX_BX)

Cycle	0	1	2	……..	3n	3n+1	3n+2
D7	R05	G05	B05		Rn5	Gn5	Bn5
D8	R04	G04	B04		Rn4	Gn4	Bn4
D5	R03	G03	B03		Rn3	Gn3	Bn3

D4	R02	G02	B02		Rn2	Gn2	Bn2
D3	R01	G01	B01		Rn1	Gn1	Bn1
D2	R00	G00	B00		Rn0	Gn0	Bn0
D1	X	X	X	X	X	X	X
D0	X	X	X	X	X	X	X

9bits Mode RGB666 (Both in CPU and RGB Interface)

Table 9bits Mode RGB666

Cycle	0	1	……..	2n	2n+1
D8	R05	G02		Rn5	Gn2
D7	R04	G01		Rn4	Gn1
D6	R03	G00		Rn3	Gn0
D5	R02	B05		Rn2	Bn5
D4	R01	B04		Rn1	Bn4
D3	R00	B03		Rn0	Bn3
D2	G05	B02		Gn5	Bn2
D1	G04	B01		Gn4	Bn1
D0	G03	B00		Gn3	Bn0

16bits Mode RGB565 (Both in CPU and RGB Interface)

Table 16bits Mode RGB565

Cycle	0	1	……..	n
D15	R04	R14		Rn4
D14	R03	R13		Rn3
D13	R02	R12		Rn2
D12	R01	R11		Rn1
D11	R00	R10		Rn0
D10	G05	G15		Gn5
D9	G04	G14		Gn4

D8	G_{03}	G_{13}	G_{n3}
D7	G_{02}	G_{12}	G_{n2}
D6	G_{01}	G_{11}	G_{n1}
D5	G_{00}	G_{10}	G_{n0}
D4	B_{04}	B_{14}	B_{n4}
D3	B_{03}	B_{13}	B_{n3}
D2	B_{02}	B_{12}	B_{n2}
D1	B_{01}	B_{11}	B_{n1}
D0	B_{00}	B_{10}	B_{n0}

16bits Mode RGB332 (CPU Interface Only)

Table 16bits Mode RGB332

Cycle	0	1	……..	n/2
D15	R_{02}	R_{22}		R_{n4}
D14	R_{01}	R_{21}		R_{n3}
D13	R_{00}	R_{20}		R_{n2}
D12	G_{02}	G_{22}		R_{n1}
D11	G_{01}	G_{21}		R_{n0}
D10	G_{00}	G_{20}		G_{n5}
D9	B_{01}	B_{21}		G_{n4}
D8	B_{00}	B_{20}		G_{n3}
D7	R_{12}	R_{32}		G_{n+12}
D6	R_{11}	R_{31}		G_{n+11}
D5	R_{10}	R_{30}		G_{n+10}
D4	G_{12}	G_{32}		B_{n+14}
D3	G_{11}	G_{31}		B_{n+13}
D2	G_{10}	G_{30}		B_{n+12}
D1	B_{11}	B_{31}		B_{n+11}
D0	B_{10}	B_{30}		B_{n+10}

16bits Mode RGB444 (CPU Interface Only)

Table 16bits Mode RGB444 (XRGB)

Cycle	0	1	……..	n
D15	X	X	X	X
D14	X	X	X	X
D13	X	X	X	X
D12	X	X	X	X
D11	R_{03}	R_{13}		R_{n3}
D10	R_{02}	R_{12}		R_{n2}
D9	R_{01}	R_{11}		R_{n1}
D8	R_{00}	R_{10}		R_{n0}
D7	G_{03}	G_{13}		G_{n3}
D6	G_{02}	G_{12}		G_{n2}
D5	G_{01}	G_{11}		G_{n1}
D4	G_{00}	G_{10}		G_{n0}
D3	B_{03}	B_{13}		B_{n3}
D2	B_{02}	B_{12}		B_{n2}
D1	B_{01}	B_{11}		B_{n1}
D0	B_{00}	B_{10}		B_{n0}

Table 16bits Mode RGB444 (RGBX)

Cycle	0	1	……..	n
D15	R_{03}	R_{13}		R_{n3}
D14	R_{02}	R_{12}		R_{n2}
D13	R_{01}	R_{11}		R_{n1}
D12	R_{00}	R_{10}		R_{n0}
D11	G_{03}	G_{13}		G_{n3}
D10	G_{02}	G_{12}		G_{n2}

D9	G01	G11		Gn1
D8	G00	G10		Gn0
D7	B03	B13		Bn3
D6	B02	B12		Bn2
D5	B01	B11		Bn1
D4	B00	B10		Bn0
D3	X	X	X	X
D2	X	X	X	X
D1	X	X	X	X
D0	X	X	X	X

18bits Mode RGB666 (Both in CPU and RGB Interface)

Table 18bits Mode RGB666

Cycle	0	1	……..	n
D17	R05	R15		Rn5
D16	R04	R14		Rn4
D15	R03	R13		Rn3
D14	R02	R12		Rn2
D13	R01	R11		Rn1
D12	R00	R10		Rn0
D11	G05	G15		Gn5
D10	G04	G14		Gn4
D9	G03	G13		Gn3
D8	G02	G12		Gn2
D7	G01	G11		Gn1
D6	G00	G10		Gn0
D5	B05	B15		Bn5
D4	B04	B14		Bn4
D3	B03	B13		Bn3

D2	B02	B12	Bn2
D1	B01	B11	Bn1
D0	B00	B10	Bn0

9.1.2 Full Display Mode

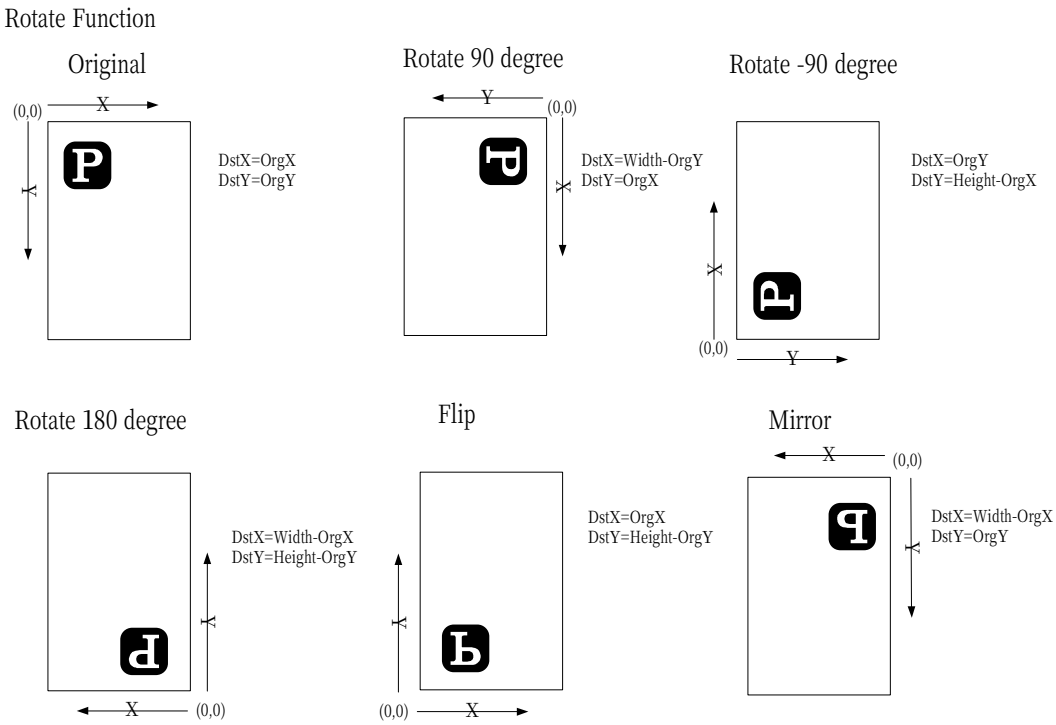
Glamo 3362 embeds display memory, LCD timing control and both CPU and RGB interface. Therefore Glamo 3362 can support both memory embedded LCM module and non-memory LCD module. The maximum display size is 640x480. The active period, front-porch, back-porch, retrace period are all programmable.

Refresh Rate

Only the video application or playing game needs to use full refresh rate for smooth motion display quality. But most of time, the LCD display is in static. While this happens, the display refresh rate can be reduced. With this, I/O power in the LCD interface and power consumption of the LCD module can be saved greatly. Glamo 3362 display controller refresh rate can be reduced to 10 Hz, even lower, by reducing the clock frequency of LCD controller module. Be noticed that the refresh rates supported by the LCD should be confirmed by the LCD vendor.

Rotation

Glamo 3362 LCD controller supports the following kinds of rotations.



Dithering

When viewing image of higher color depth on lower color depth LCD display, we tend to ignore low bits. This causes block effect on otherwise smooth surface of the image. Glamo 3362 provides special dithering architecture to eliminate this effect.

OSD

Glamo 3362 provides 2bits up to full screen size On Screen Display function. 2bits plain can define four operations:

00: Background color (defined in LCD Register 0x1164h)

10: Foreground color (defined in LCD Register 0x1160h)

01: Source color (or third color defined in LCD Register 0x1168h, if 0x1100h bit 6 set to 1)

11: Inverse source color

With this function, we can apply to pop-up menu on any plain, such as graphic or video.

Flipping

When motion video clip or graphic animation is shown on LCD, user can see the discomfort tearing image if there is only one frame displayed on the LCD. To solve this, Glamo 3362 implemented two display buffers, one is for LCD display and the other is for video or graphic engine drawing for the following frame. With that, we named the feature “flipping function”. When the following frame is well drawn, engine will issue a flip command to LCD controller. After receiving the flip command, LCD controller will not flip to the new frame until the vertical retracing starts. With this scheme, LCD display will always show the complete frame.

Gamma Correction

The character of each type of LCD panel may be different and with different gamma curve. Glamo 3362 supports fully programmable Gamma table to fine tune the display quality. The related registers are defined at LCD Register 0x1200h~0x125Fh.

9.1.3 Partial Display Mode

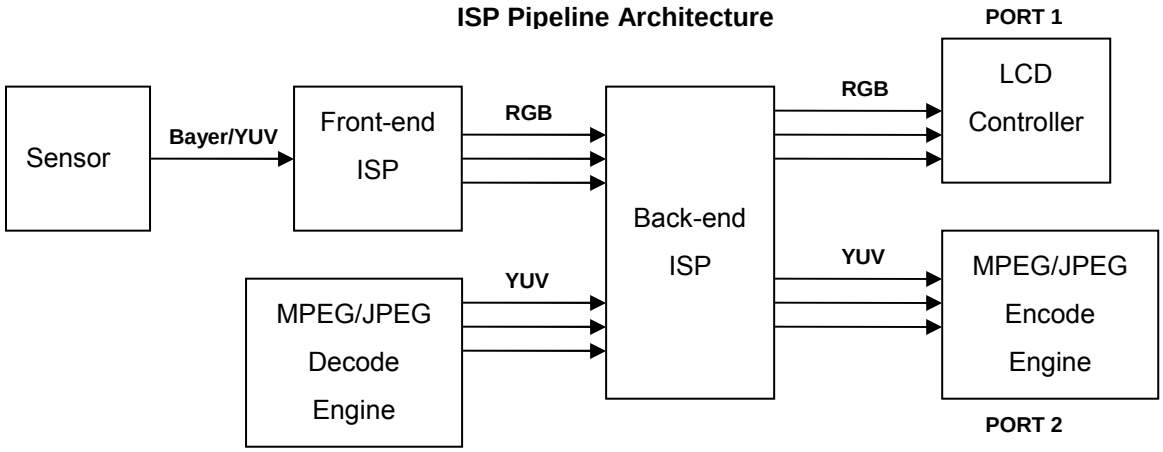
In full display mode, for display refresh, LCD controller in Glamo 3362 needs to read the frame buffer data from external memory periodically. The power consumption includes the memory operation power, the memory interface IO power, LCD controller power and the LCD IO power. Glamo 3362 supports one 128x64 pixels (2bits per pixel) internal SRAM for partial display. It will save the power consumption of memory operation and IO. The internal SRAM can be defined as any kind of dimension smaller than 128x64. For example, user can define the partial display screen as 256x32, 128x64, 64x128, and 32x256. The color of partial display uses the same definition as that of OSD function.

9.1.4 Bypass Mode

Glamo 3362 supports the LCD bypass mode. When this mode is enabled, baseband CPU will directly control the LCD panel and bypass Glamo 3362's LCD controller. Because it is hardware link from Host IO to LCD IO, Glamo 3362 can turn off all the clocks and PLL. At this moment, only Host IO and LCD IO consume power.

9.2 Image Signal Processor

The Image Signal Processor (ISP) is a hardware pipeline engine in Glamo 3362. It receives the sensor captured data or JPEG/MPEG decoding engine, doing the image processing and scaling and then output to the LCD monitor or JPEG/MPEG encoding engine. The input data format can be Bayer or YUV, and output format can be RGB for LCD and YUV for JPEG/MPEG engine. All the functions are controlled by the Capture Registers and ISP registers. Besides, there is s micro processor in Glamo 3362 to control the sensor programming and 3A (auto exposure, auto white balance and auto focus). That's to say, with Glamo 3362, baseband will just need to do handful things on DSC functions. The following diagram shows the pipeline architecture of ISP.



9.2.1 Sensor Programming

Each sensor needs to be configured before it can work properly, and most of sensors provide various modes for preview and snapshot. Additionally, based on various environment conditions, we need real-time change the sensor parameters, for example, the gain value for auto white balance control and exposure time for auto exposure control. Sensor contains one serial bus to program the internal registers. Glamo 3362 provides programmable micro processor scheme to handle these real-time control task. With this scheme, baseband CPU only needs to send commands or get the result without any extra loading, so system will be very easy to implement the DSC function into their mobile solution.

9.2.2 Front-End Image Processing

Front-End Image Processing module receives the data from Sensor and output to the back-end Image Processing module. The function of front-end ISP includes bad pixel removal, shading correction, black pixel

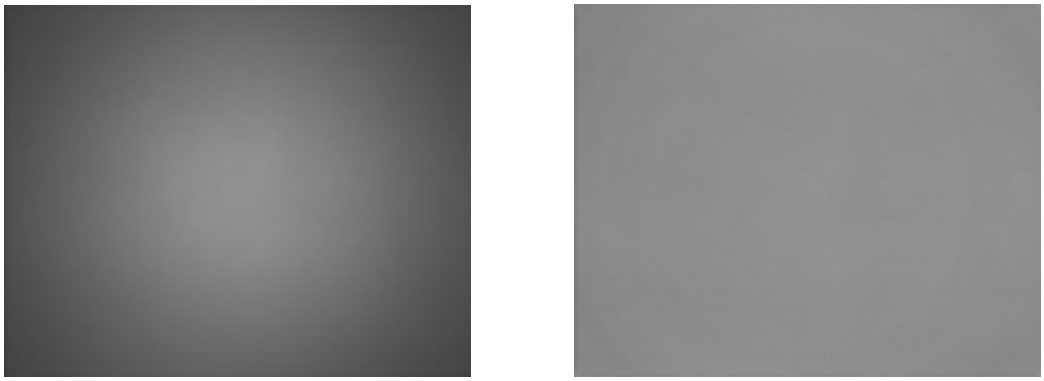
compensation, gain compensation, gamma Correction and Image Decimation.

Bad Pixel Removal

Every sensor may contain bad pixels. The bad pixels may be dark points or bright points, and it will let user feel discomfort because of the abruptly color noise. Glamo 3362 supports the bad pixel removal by recording the bad pixel location in advance and remove the bad pixels at run time. Glamo 3362 will use adjacent information to regenerate the defect pixel.

Lens Shading Correction

In mobile phone, because of the camera module's mechanism limitation, the lens size will be limited. Therefore the higher resolution the sensor is, the worse the lens distortion will be. Glamo 3362 supports hardware real-time lens shading compensation, and shading shape and shading location can be programmed. The related registers are defined in Video Capture registers 0x46Eh~0x497h.



Example of Shading Correction (a) before shading correction (b) after shading correction

Black Level Compensation

Each sensor may define an optically masked region for the black level compensation. It is because the sensor in the masked region may still contain a small value larger than 0. Therefore, the image acquired by the sensor needs to be compensated (subtracted) by the black level value first.

Gain Compensation

The spectrum sensitivities of Gr, R, Gb, B channels in a Bayer pattern image sensor vary depending on the ambient light sources. Hence, Glamo 3362 supports the gain compensation for each color channel to compensate the sensitivity differences.

Gamma Correction

Glamo 3362 supports a fully programmable gamma correction which allows users to define specific gamma curves. The gamma curves of Gr, R, Gb, and B channels can be defined independently.

Example of Gamma correction – (a) before gamma (b) after gamma correction

Image Decimation

In preview or video recording mode, the LCD or MPEG record size is much smaller than the sensor size. Glamo 3362 supports pre-step scaling down the image size. The function is useful for power saving, because it can reduce the data throughput of the ISP pipeline. If use simply drop line decimation algorithm, it may cause the jaggy edge and high frequency image. It will let user feels discomfort and raises the MPEG bits rate and also impact the video quality. Glamo 3362 support high quality 2-taps horizontal and 2-taps vertical decimation. It can improve the video quality display on LCD display and the MPEG encoding quality.

Image source

Example of Decimation - (a) Direct Drop (b) High quality decimation

9.2.3 Back-End Image Processing

Back-End Image Processing module receives data from Front-End Image Processing module (Encoding mode) or data from JPEG/MPEG decode engine (Playback mode) and output to the LCD controller and JPEG/MPEG encode engine. The functions of back-end Image Processing include Color Interpolation, Color Correction, Image Enhancement, Color space Transformation, 4-by-4 taps Continuous Scaling, Image Effects, Rotation and etc.. Back-End Image Processing module also collects the image statistic information for Auto Exposure, Auto White Balance and Auto Focus judgment.

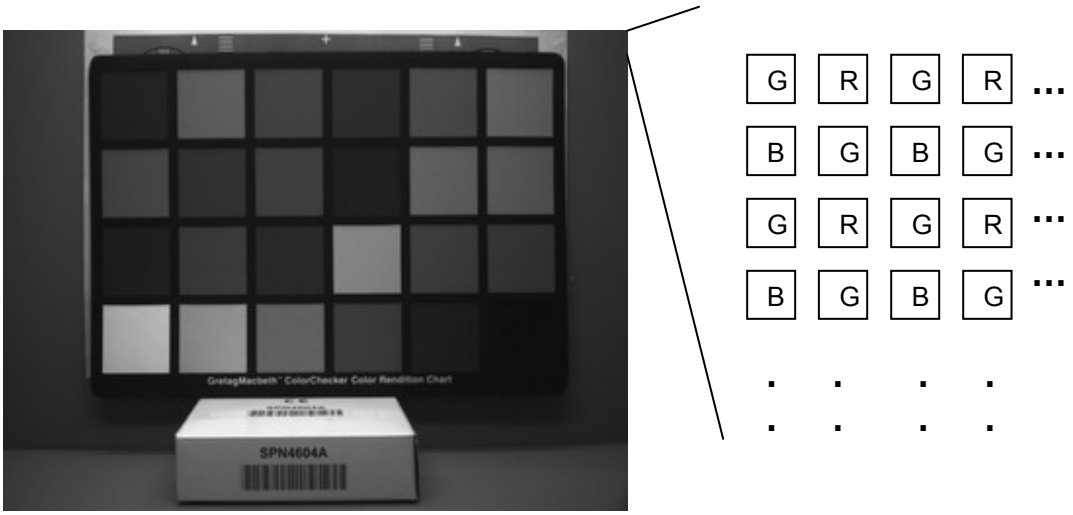
Auto Exposure/Auto White Balance/Auto Focus

Glamo 3362 provides the related statistic information to achieve the “3A” control. In order to reduce the baseband CPU loading, Glamo 3362 also builds in a micro processor to handle these tasks. So Glamo 3362 can achieve the “3A” control without baseband CPU involved. Auto Exposure is to dynamically changes the senor exposure time. Glamo 3362 ISP module provides the luminance statistic information every frame. Micro processor uses the information to change the exposure time of the sensor by programming the sensor registers. Auto White Balance is to dynamically change the gain value of the four-channel of raw data (Gr/R/Gb/B). Glamo 3362 provides the white area detection and white value statistic information every frame. Micro processor uses the information to change the gain value of the Gr, R, Gb, B channels on Glamo 3362 or sensor. Auto Focus is to dynamically change the focus-motor to focus on the target object. Glamo 3362 provides the Modulation Transfer Function (MTF) information every frame. Micro processor watches the changing MTF information of series frames and modifies the focus-motor value by GPIO pins.

Example of Auto white balance (a) before AWB (b) after AWB

Color Interpolation

For Bayer data format input, Glamo 3362 provides color interpolation architecture to generate from Gr, R, Gb, B channels per pixel per byte to RGB channel per pixel 3 bytes.



Example of Bayer data input

Example of after color interpolation

Color Correction

Various sensor types may contain various color sensitivity curves. It needs processes color correction to get the real color. Glamo 3362 provides fully programmable color correction matrix to fit various color correction matrix requests from various vendors. The color correction formula is as follows:

$$\begin{bmatrix} R_{DST} \\ G_{DST} \\ B_{DST} \end{bmatrix} = \begin{bmatrix} CC_{11} & CC_{12} & CC_{13} \\ CC_{21} & CC_{22} & CC_{23} \\ CC_{31} & CC_{32} & CC_{33} \end{bmatrix} \cdot \begin{bmatrix} R_{SRC} \\ G_{SRC} \\ B_{SRC} \end{bmatrix} + \begin{bmatrix} R_{Delta} \\ G_{Delta} \\ B_{Delta} \end{bmatrix}$$

Example of Color Correction – (a) before color correction (b) after color correction

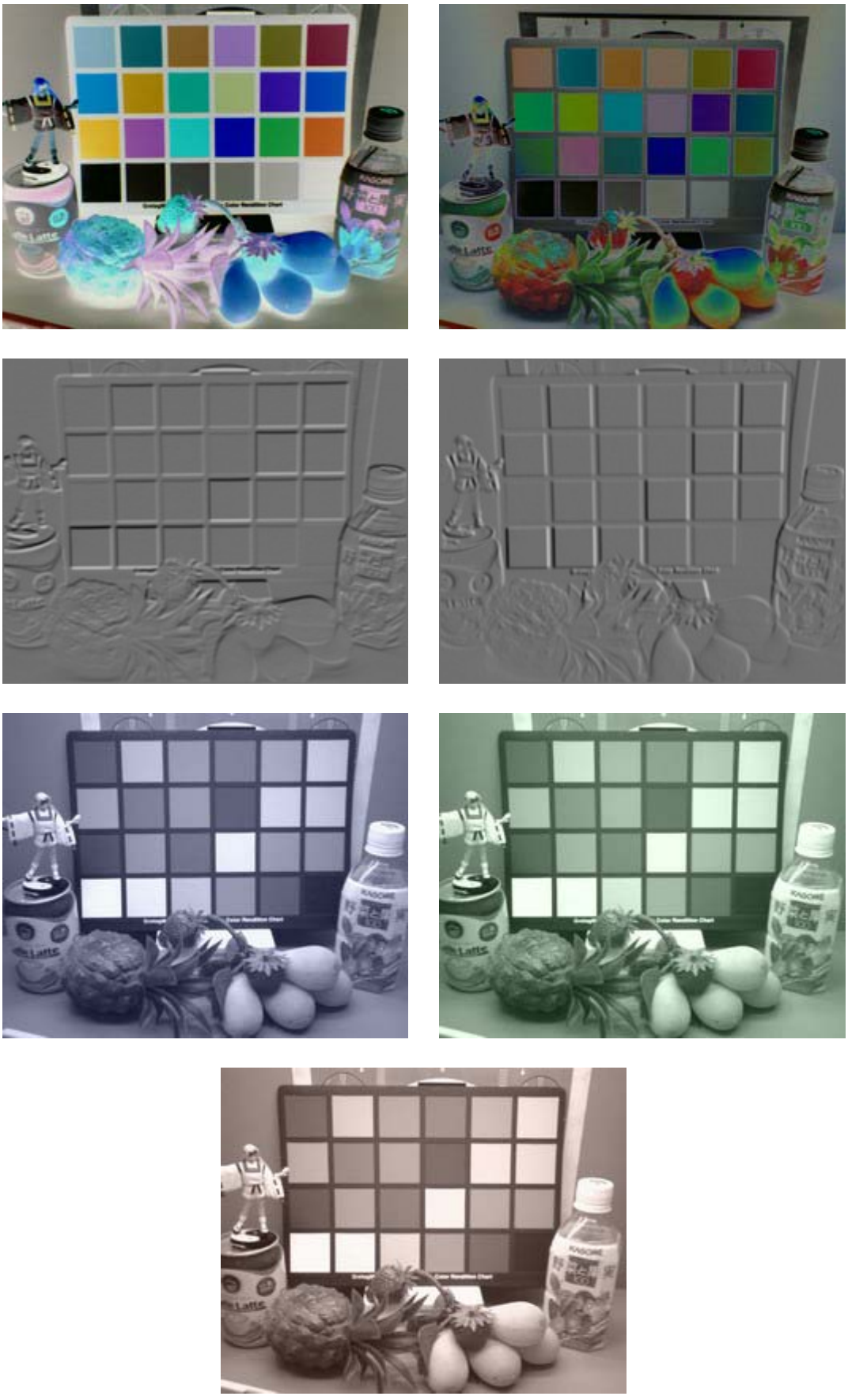
Hue/Saturation/Brightness/Contrast Image Enhancement

By using different color correction matrix, Glamo 3362 can support Hue/Saturation/Brightness/Contrast image enhancement and Black/White, solarization and a variety of color filter effects. All the effects can be applied to preview mode as well as JPEG image and MPEG video.

Exampe of Image Enhancement – (a) Origin (b) Brightness (c) Contrast (d) Hue (e) Saturation

Image Effects

Glamo 3362 provides hardware night shot compensation, negative picture, solarization, emboss and color filter effects. All the effects can be applied to preview mode as well as JPEG image and MPEG video.



Example of Image Effects – (a) Origin (b) Negative (c) Solarization (d) Emboss 1 (e) Emboss 2 (f) Color Filter with Blue (g) Color Filter with Green (h) Color Filter with Red

2D Edge Enhancement

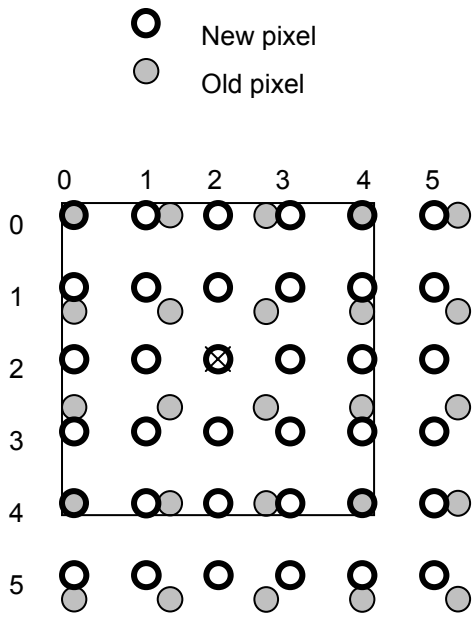
Glamo 3362 supports hardware 2D edge enhancement and the enhance strength can be programmed.

Example of 2D edge enhancement (a) without edge enhancement (b) with edge enhancement

Continuous Image Scaling

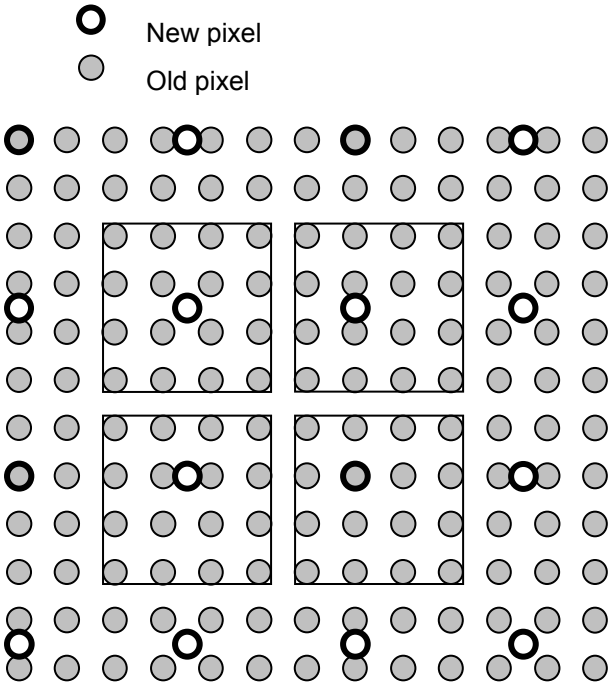
Glamo 3362 supports 4-tap horizontal scaler and 4-tap vertical scaler to improve the image scaling quality. The image can be scaled up to 16X without discomfort block effect and can be scaled down to 16X without discomfort jaggy effect. Glamo 3362 also supports continuous image scaling. That is to say, customer can zoom in or zoom out to set the scene he would like to snap smoothly. The following case shows the 4-tap scaling methodology. All the weighted parameters are user programmable.

Example 1: 5/4 Scaling Up:



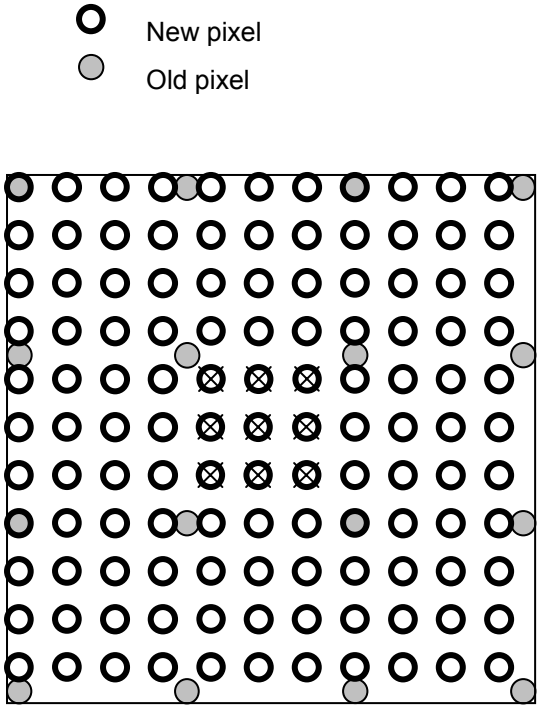
New pixel is calculated by using the 16 old pixels inside the rectangle

Example 2: 7/2 Scaling Down:



New pixel is calculated by using the 16 old pixels inside the rectangle

Example 3: 7/2 Scaling Up:



Marked new pixel are calculated by using the 16 old pixels inside the rectangle

Color Space Conversion

In video encoding mode or snapshot mode, ISP module provides RGB to YUV color space conversion in the data path. In video clip playback mode or photo browsing mode, ISP module provides YUV to RGB color space conversion. These conversions are 3x3 matrix operation. The parameter of the matrix is fully

programmable.

Rotation

Glamo 3362 supports two kinds of options for rotation. For display, that is, browsing video clips or images on the LCD monitor, Glamo 3362 supports +90, +180, -90, mirror and flip rotation options. For MPEG/JPEG file, Glamo 3362 supports +180, mirror, flip rotation options. The related register is defined in ISP register 0x0502h.

Frame Function

User may want to take pictures by using beautiful scene as his background or add flower frame around his picture. We call this application as frame function. Glamo 3362 supports frame function in snapshot. The background picture or frame image can be defined by user. Glamo 3362 uses overlay with color key technology to merge the picture that user took with pre-defined frame image. In the frame function mode, Glamo 3362 still can support continuous image scaling. That is, user can do continuous zoom in to take a picture.

Clipping

Glamo 3362 supports rectangle overlay to display the video or image on LCD monitor. User can define the rectangle size and position on the LCD by setting the clipping window. User also can define clipping inside or outside. With this function, video can be updated within this defined window without redrawing the original graphic outside the defined window.

9.2.4 ISP Operation Mode

According to the applications, ISP module supports the following operation modes:

Preview mode

In this mode, MPEG codec, JPEG codec are idle and the related clocks are off. The host controller is idle and related clock is off because micro processor take care all the auto exposure, auto white balance and auto focus jobs. The 2D/3D engines are idle and related clocks are off because of rectangle overlay, only video updated. Memory controller is enabled. LCD Controller is enabled. Micro processor is enabled. The initial sequence is as follows:

Turn on the related clocks

Program Sensor to preview mode.

Program the ISP registers for data input from sensor and output to LCD (port 1).

Program the video capture registers.

Enable micro processor.

Fire ISP module by setting 0x0502 bit0 to 1

Fire Video Capture module by setting 0x04c0 bit0 to 1

Snapshot mode

In this mode, JPEG encode will be enabled and the other settings will be similar to preview mode. The initial sequence is as follows:

Turn on the related clocks

Program Sensor to snapshot mode.

Program the JPEG encode registers.

Program the ISP registers for data input from sensor and output to LCD (port 1) and JPEG encoder (port 2).

Program the video capture registers.

Fire JPEG encoder module

Fire ISP module by setting 0x0502 bit0 to 1

Fire Video Capture module by setting 0x04c0 bit0 to 1

Video Recoding mode

In this mode, MPEG encode will be enable and Host Controller is enabled to read back the encoded bitstream. The other setting is similar to preview mode. The initial sequence is as follows:

Turn on the related clocks

Program Sensor to preview mode.

Program the MPEG encode registers.

Program the ISP registers for data input from sensor and output to LCD (port 1) and MPEG encoder (port 2).

Program the video capture registers.

Fire MPEG encoder module

Fire ISP module by setting 0x0502 bit0 to 1

Fire Video Capture module by setting 0x04c0 bit0 to 1

Photo browsing mode

In this mode, the JPEG Decoder, Memory Controller, LCD Controller and Host Controller are enabled.

Turn on the related clocks

Program the JPEG decode registers.

Program the ISP registers for data input from JPEG decoder and output to LCD (port 1).

Fire JPEG Decoder module

Fire ISP module by setting 0x0500 bit0 to 1

Video Playback mode

In this mode, the MPEG Decoder, Memory Controller, LCD Controller and Host Controller are enabled.

Turn on the related clocks

Program the MPEG decode registers.

Program the ISP registers for data input from MPEG decoder and output to LCD (port 1).

Fire MPEG Decoder module

Fire ISP module by setting 0x0500 bit0 to 1

Video Conference mode

Turn on the related clocks

Program Sensor to preview mode.

Enable Micro Processor.

Program the MPEG decode registers.

Program the MPEG encode registers.

Program the video capture registers.

Loop

Program the ISP registers for data input from sensor and output to LCD (port 1) and MPEG encoder (port 2).

Fire MPEG Encoder module

Fire ISP module by setting 0x0502 bit0 to 1

Fire Video Capture module by setting 0x04c0 bit1 to 1

Program the ISP registers for data input from MPEG decoder and output to LCD (port 1).

Fire MPEG Decoder module

Fire ISP module by setting 0x0500 bit0 to 1

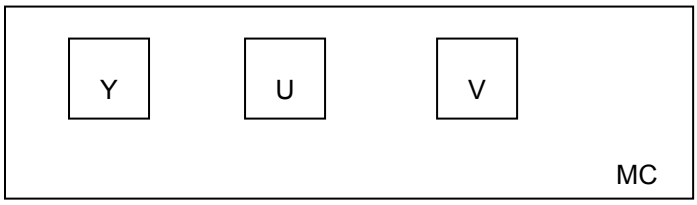
Go to Loop

9.3 JPEG Engine

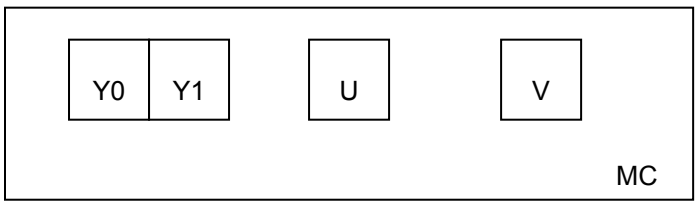
9.3.1 Support Formats

The Glamo 3362 JPEG codec is based on the JPEG baseline standard and the arithmetic accuracy satisfies the requirement of the compatibility test of JPEG Part-2 (ISO/IEC10918-2). The maximum image size is 5M pixels (2592x1944). The image that can be encoded/decoded must be the alignment of MCU base. The parameters of luminance and chrominance quantization table are fully programmable. The Huffman tables use the default tables that are suggested by specification. The decoding process supports YUV 4:4:4, 4:2:2, 4:1:1 and 4:2:0 with interleaved format, but the encoding process only supports YUV 4:2:2 interleaved format. The progressive mode is not supported, but sequential mode. The following diagram shows the definition of each formation.

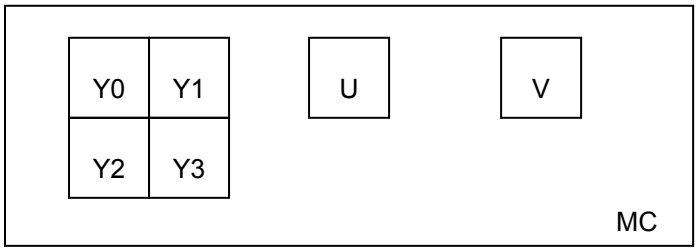
YUV 4:4:4 Format



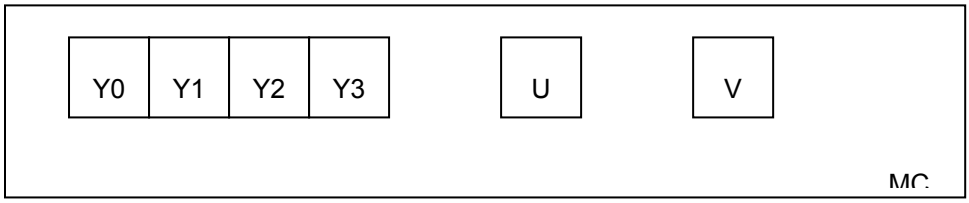
YUV 4:2:2 Format



YUV 4:2:0 Format



YUV 4:1:1 Format



9.3.2 Encode/Decode Time

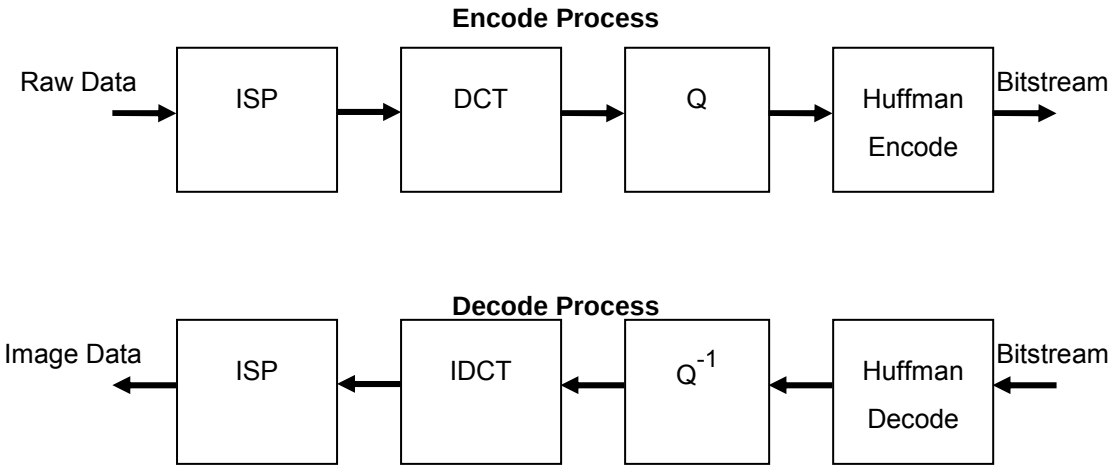
The encoding and decoding time of an image is about 70 clock cycle every block,. For example, when 1.3M-pixel(1280x1028) 4:2:2 format is processed under 33MHz frequency. The total numbers of block are 20480(Y) + 10240(U) + 10240(V) = 40960 blocks. The encode/decode time is about 0.08 second.

9.3.3 JPEG Engine Pipe Line Architecture

The following diagram shows the JPEG engine pipe line.

In encoding process, JPEG engine receives the image raw data from frame buffer and output the bit-stream to bit stream buffer. The handshaking with the ISP module is slice base, and can be triggered by hardware automatically. JPEG encoding engine also support software trigger mode. System puts the raw data from host interface to frame buffer and then driver will command JPEG engine to be triggered. JPEG encoding engine passes the raw data to DCT engine block, quantization block and then Huffman entropy encode block. All the flow is pipelined and controlled by hardware. The output bit-stream needs to add file header and marker to become JFIF or EXIF file format. The packaging job should be done by CPU.

In decoding process, JPEG engine receives the bit-stream from bit-stream buffer and output the constructed image to ISP module for further operations (scaling and rotation). The handshaking with the ISP module is slice-based and is triggered by software command. JPEG decode engine passes the bit-stream to Huffman entropy decode block, Dequantiaztion block and Inverse-DCT block. All the flow is pipelined and controlled by hardware.



9.3.4 Thumbnail Image

There are two ways to generate the thumbnail image. One is to use Port 1 of ISP module to output the RGB raw image as thumbnail image. With this method, the thumbnail image will be RGB 888 raw data format. The image size can be programmed by changing the scaling factors of Port 1 of ISP module. The other way is to use two passes to generate another small size image as thumbnail image. After the first pass is gone, we can change the scaling factors of ISP module and fire for JPEG encode again by software trigger command. With this method, the thumbnail image will be JPEG data format.

9.3.5 Engine Status Report

Glamo 3362 provides two schemes to inform baseband CPU the JPEG engine status. One is register flag. CPU is poling the JPEG Register 0xAB2 bit 0 or bit 1 to check the encoding or decoding task finished or not. The other is use interrupt to inform baseband CPU the JPEG engine status.

9.3.6 JPEG Encoding Process

The JPEG encoding process is as follows:

1. Initial the sensor and ISP module to snapshot mode.
2. Initial the JPEG encoding related register.
3. Fire ISP for snapshot
4. Interrupt signal to inform CPU for JPEG encoding ready
5. Check the JPEG engine status
6. Read back the encoded bitstream
7. Append the header and marker as JFIF or EXIF file format.

9.3.7 JPEG Decoding Process

The JPEG encoding process is as follows:

1. Initial ISP module to image viewing mode.
2. Initial the JPEG decoding related register.
3. Send bitstream to frame buffer for decoding
4. Fire ISP for decode mode
5. Interrupt signal to inform CPU for JPEG decoding ready
6. Check the JPEG engine status
7. Show the decoded image on the LCD display

9.4 MPEG Engine

9.4.1 Support Formats

The Glamo 3362 MPEG codec is based on the ISO/IEC 14496-2 (MPEG-4) simple profile standard. The maximum size can up to CIF (352x288) 30 fps or VGA (640x480) 12 fps. Glamo 3362 also supports the VOP

with short header (H.263). Glamo 3362 hardware decoder and encoder are fully pipelined architecture. It supports to VOP layer, the baseband only need to packaging the file header and AV synchronization for MPEG encoding and parsing the header for MPEG decoding.

9.4.2 MPEG Codec Architecture

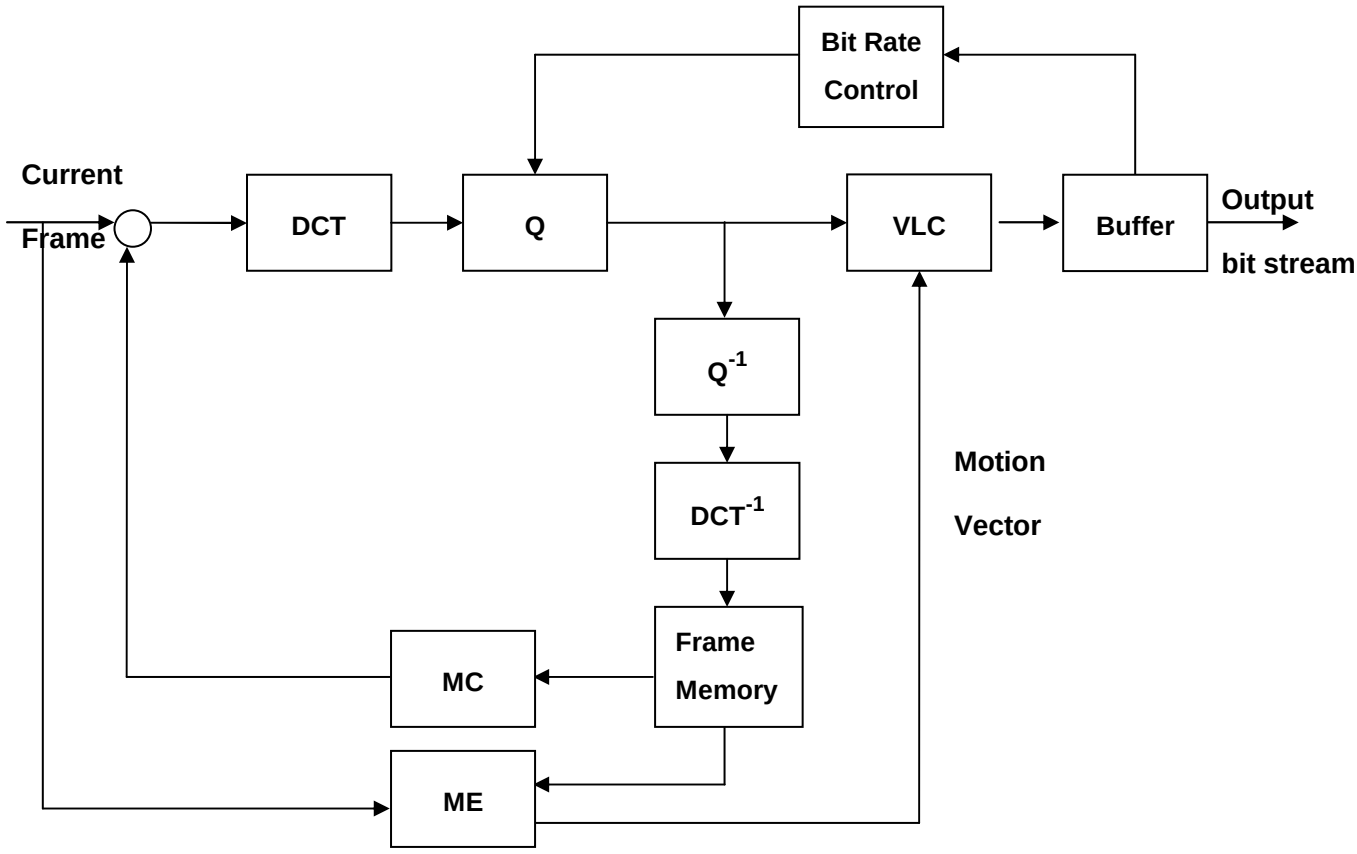
The following diagram shows the hardware architecture of MPEG encoder and MPEG decoder of Glamo 3362. They are all fully pipelined design and hardware flow control.

In MPEG decoder, the bit-stream sends to the VLD (Variable Length Decoding) block, Inverses Quantization block, inverses DCT transform and adds with data of motion compensation to generate the decoded MPEG video and then sends to the ISP and LCD module to display.

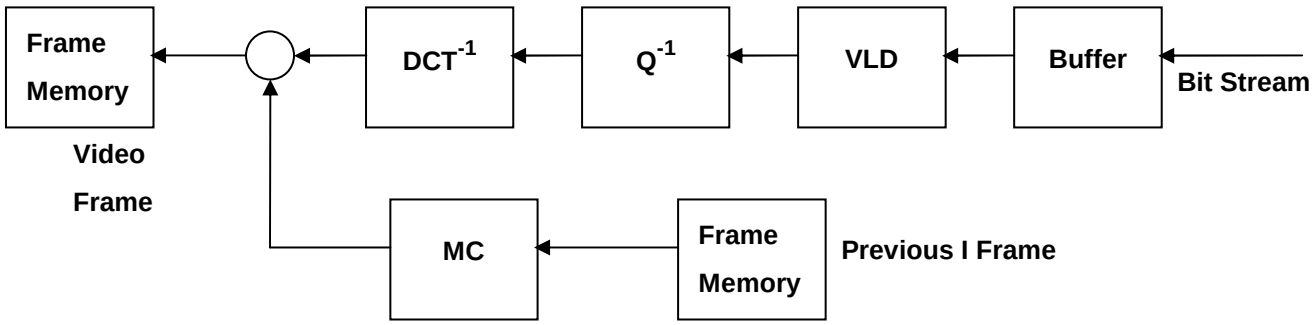
In MPEG encoder, there is one reconstructed path to generate the reconstructed video frame, and one motion estimation block to estimate the motion vector. The video frame subtracts the reconstructed video frame and sends to DCT block, Quantization block, Variable Length Coding and output the encoded bit-stream the bit-stream buffer.

The motion estimation of Glamo 3362 supports full search algorithm with ±16 search range in both X and Y direction. The resolution of motion vector is half-pixel. If the encoder is configured to encode MPEG-4 bitstream, the 4-MV (four motion vectors per macroblock) and unrestricted MV (motion vectors are allowed to point outside picture boundary) functions are also fully supported.

MPEG Encode Architecture



MPEG Decoder Architecture



Rate Control

Base on different applications, the rate control function of Glamo 3362 supports both CBR (constant bit rate) and VBR (variable bit rate) mode. For network transmission with constant bandwidth, the CBR mode can fulfill the allocated bandwidth with maximum quality. When recording video into local storage like flash card or hard disk, the VBR mode can adjust the bit-consumption rate according to the complexity variation of the incoming video, achieve a more uniform visual quality. The rate control function also includes a adaptive quantization scheme to support the HVS (human visual system). The quantization level of each macroblock is varied according to its visual complexity. With this scheme, the encoded video quality can be improved comparing to the video without rate control with same bit rate.

Handshaking with ISP

In the MPEG encoding mode, MPEG encoder receives the video data from ISP module with triple buffer scheme. The handshaking control is done by hardware, baseband CPU do not need to take the flow control efforts to reduce the CPU loading. Baseband CPU only needs to read back the encoded bit-stream and package the header of MPEG file format. In the MPEG decoding mode, MPEG decoder provided the decoded video data to ISP module every frame. And then software flips the video to LCD display according to the time stamp.

9.4.3 MPEG Encoding Process

The MPEG encoding process is as follows:

1. Initial the sensor and ISP module to video recording mode.
2. Initial the MPEG encoding related register.
3. Fire ISP for video recording.
4. Read back the encoded bitstream
5. Append the header and synchronous with audio bit-stream to became 3GPP file format.

9.4.4 MPEG Decoding Process

The MPEG decoding process is as follows:

8. Initial ISP module to video playback mode.
9. Initial the MPEG decoding related register.
10. De-multiplex the audio and video bit-streams
11. Send video bitstream to frame buffer for decoding
12. Fire ISP for MPEG decode mode
13. Show the decoded video on the LCD display

9.5 Micro Processor

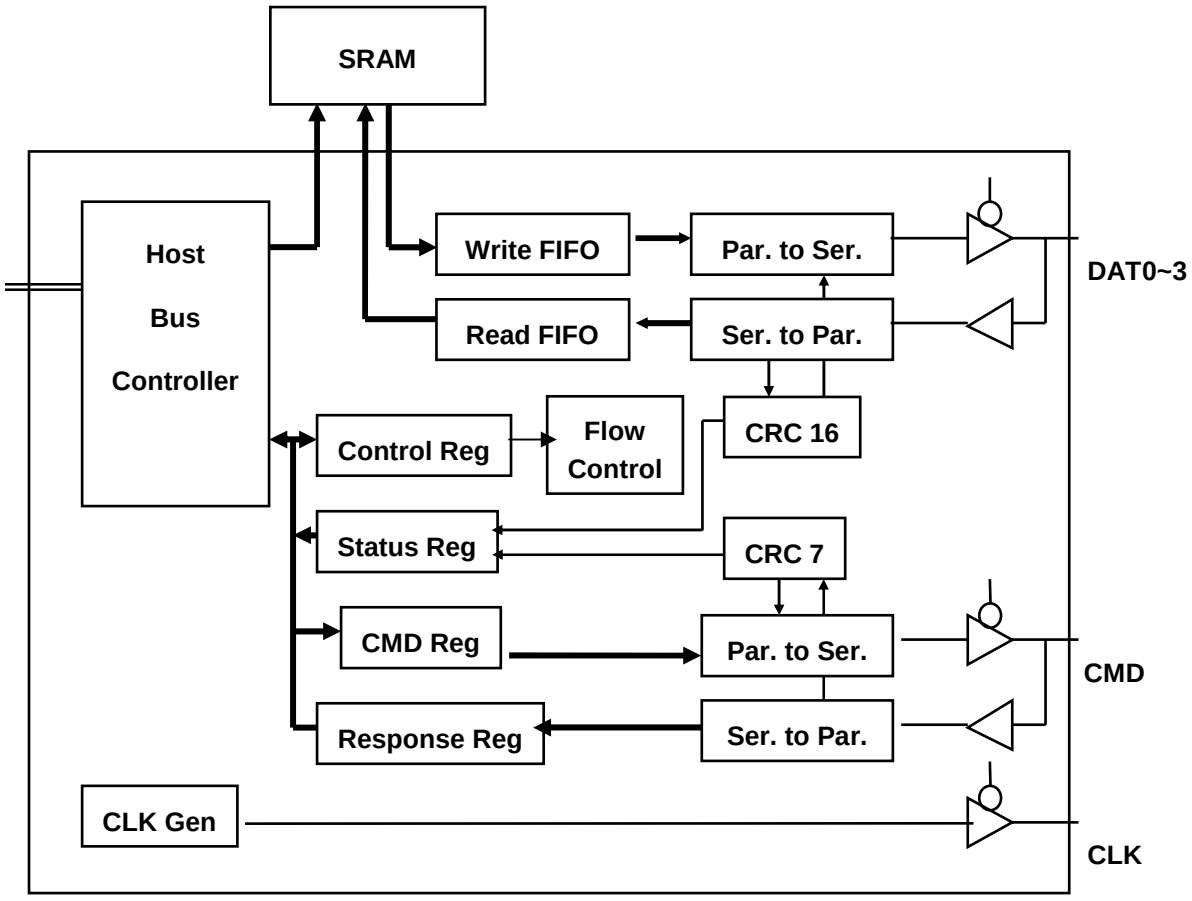
Glamo 3362 embeds two RISC processors for operational control. One is used for ISP function flow control and the other is used for rate control of MPEG encoding. Two processors can be worked independently. The purpose of the micro processors is to release the baseband CPU loading. When system process the camera function or video recording function, the baseband CPU is almost idle. Micro processor will take cares all the flow control and simple calculation efforts.

9.6 MMC/SD Controller

The Glamo 3362 support MMC/SD memory card and fully compliant with MMC 3.3 specification and compliant with low-voltage support of MMCA v4.0 and SD specification. The supported features as follows:

Support 4-bit DAT
Support variable clock rates : up to 50MHz supported
Support variable transfer block size
Support Single block READ/WRITE
Support open-ended Multiple block READ
Support Multiple block READ with pre-defined block count
Support Multiple block WRITE with pre-defined block count
Support Stream READ

9.6.1 Block Diagram



9.6.2 MMC/SD Read Process

1. Initialize the MMC/SD card
2. Prepare the read control information
3. Issue read CMD to CMD register
4. Wait response status from MMC/SD card.
5. Ready to receive data and save the received data back to SRAM
6. Check the CRC to make sure the data valid
7. Update the status register for CPU read back
8. CPU read back the data
9. Loop back to 3

9.6.3 MMC/SD Write Process

1. Initialize the MMC/SD card
2. Prepare the write control Information and write data into SRAM
3. Issue write CMD to CMD register
4. Wait response status
5. Send write data from SRAM to MMC DAT port block by block
6. Check the CRC/Busy status on DAT Line for valid write and wait programming complete
7. Loop Back to 5
8. Update the status register for write complete

9.7 2D Graphics Engine

Mobile Multimedia Processor (MMP) supports a powerful 2D graphics engine to enhance the performance. It only supports high color (16bpp) mode, and following functions:

1. BitBlt with ROP3
2. Color expansion
3. Transparent BitBlt with source and destination key
4. Line Drawing
5. Stretch
6. Alpha Blending, it supports constant alpha, ARGB8888, ARGB1555, and ARGB4444 source bitmap for per-pixel alpha value.

For each function, it can rotate by 90^{o}, -90^{o}, 180^{o}, flip and mirror the coordinate. The fill rate depends on the clock rate of 2D engine. For example, if the clock rate is 33MHz, the fill rate is 33M pixels per second.

9.7.1 ROP3

This section lists the ternary raster-operation codes used by the **BitBlt**, **Color Expansion**, and **Line Draw** functions. The line drawing function just supports ROP2 operation. It's same as ROP3 Boolean truth table but without Source components. The Pattern components comes from line style, the Destination components comes from destination bitmap. Ternary raster-operation codes define how hardware combines the bits in a source bitmap with the bits in the destination bitmap.

Each raster-operation code represents a Boolean operation in which the values of the pixels in the source, the selected brush, and the destination are combined. Following are the three operands used in these operations:

Operand	**Meaning**
D	Destination bitmap
P	Selected brush (also called pattern)
S	Source bitmap

Boolean operators used in these operations follow:

Operator	**Meaning**
a	Bitwise AND
n	Bitwise NOT (inverse)
o	Bitwise OR
x	Bitwise exclusive OR (XOR)

All Boolean operations are presented in reverse Polish notation. For example, the following operation replaces the values of the pixels in the destination bitmap with a combination of the pixel values of the source and brush:

PSo

The following operation combines the values of the pixels in the source and brush with the pixel values of the destination bitmap (there are alternative spellings of the same function, so although a particular spelling may not be in the list, an equivalent form would be):

DPSoo

For example, the operation indexes for the PSo and DPSoo operations are shown in the following list:

P	S	D	PSo	DPSoo
0	0	0	0	0
0	0	1	0	1
0	1	0	1	1
0	1	1	1	1
1	0	0	1	1
1	0	1	1	1
1	1	0	1	1
1	1	1	1	1
Operation index:			FCh	FEh

In this case, PSo has the operation index FC (read from the bottom up); DPSoo has the operation index FE. These values define the location of the corresponding raster-operation codes, as shown in Table A.1, "Raster-Operation Codes." The PSo operation is in line 252 (FCh) of the table; DPSoo is in line 254 (FEh).

Table A.1 Raster-Operation Codes

Boolean function (hexadecimal)	Boolean function in reverse Polish	Common name
00	0	**BLACKNESS**
01	DPSoon	
02	DPSona	
03	PSon	
04	SDPona	
05	DPon	
06	PDSxnon	
07	PDSaon	

08	SDPnaa	
09	PDSxon	
0A	DPna	
0B	PSDnaon	
0C	SPna	
0D	PDSnaon	
0E	PDSonon	
0F	Pn	
10	PDSona	
11	DSon	**NOTSRCERASE**
12	SDPxnon	
13	SDPaon	
14	DPSxnon	
15	DPSaon	
16	PSDPSanaxx	
17	SSPxDSxaxn	
18	SPxPDxa	
19	SDPSanaxn	
1A	PDSPaox	
1B	SDPSxaxn	
1C	PSDPaox	
1D	DSPDxaxn	
1E	PDSox	
1F	PDSoan	
20	DPSnaa	
21	SDPxon	
22	DSna	
23	SPDnaon	
24	SPxDSxa	

25	PDSPanaxn	
26	SDPSaox	
27	SDPSxnox	
28	DPSxa	
29	PSDPSaoxxn	
2A	DPSana	
2B	SSPxPDxaxn	
2C	SPDSoax	
2D	PSDnox	
2E	PSDPxox	
2F	PSDnoan	
30	PSna	
31	SDPnaon	
32	SDPSoox	
33	Sn	**NOTSRCCOPY**
34	SPDSaox	
35	SPDSxnox	
36	SDPox	
37	SDPoan	
38	PSDPoax	
39	SPDnox	
3A	SPDSxox	
3B	SPDnoan	
3C	PSx	
3D	SPDSonox	
3E	SPDSnaox	
3F	PSan	
40	PSDnaa	
41	DPSxon	

42	SDxPDxa	
43	SPDSanaxn	
44	SDna	**SRCERASE**
45	DPSnaon	
46	DSPDaox	
47	PSDPxaxn	
48	SDPxa	
49	PDSPDaoxxn	
4A	DPSDoax	
4B	PDSnox	
4C	SDPana	
4D	SSPxDSxoxn	
4E	PDSPxox	
4F	PDSnoan	
50	PDna	
51	DSPnaon	
52	DPSDaox	
53	SPDSxaxn	
54	DPSonon	
55	Dn	**DSTINVERT**
56	DPSox	
57	DPSoan	
58	PDSPoax	
59	DPSnox	
5A	DPx	**PATINVERT**
5B	DPSDonox	
5C	DPSDxox	
5D	DPSnoan	
5E	DPSDnaox	

5F	DPan	
60	PDSxa	
61	DSPDSaoxxn	
62	DSPDoax	
63	SDPnox	
64	SDPSoax	
65	DSPnox	
66	DSx	**SRCINVERT**
67	SDPSonox	
68	DSPDSonoxxn	
69	PDSxxn	
6A	DPSax	
6B	PSDPSoaxxn	
6C	SDPax	
6D	PDSPDoaxxn	
6E	SDPSnoax	
6F	PDSxnan	
70	PDSana	
71	SSDxPDxaxn	
72	SDPSxox	
73	SDPnoan	
74	DSPDxox	
75	DSPnoan	
76	SDPSnaox	
77	DSan	
78	PDSax	
79	DSPDSoaxxn	
7A	DPSDnoax	
7B	SDPxnan	

7C	SPDSnoax	
7D	DPSxnan	
7E	SPxDSxo	
7F	DPSaan	
80	DPSaa	
81	SPxDSxon	
82	DPSxna	
83	SPDSnoaxn	
84	SDPxna	
85	PDSPnoaxn	
86	DSPDSoaxx	
87	PDSaxn	
88	DSa	**SRCAND**
89	SDPSnaoxn	
8A	DSPnoa	
8B	DSPDxoxn	
8C	SDPnoa	
8D	SDPSxoxn	
8E	SSDxPDxax	
8F	PDSanan	
90	PDSxna	
91	SDPSnoaxn	
92	DPSDPoaxx	
93	SPDaxn	
94	PSDPSoaxx	
95	DPSaxn	
96	DPSxx	
97	PSDPSonoxx	
98	SDPSonoxn	

99	DSxn	
9A	DPSnax	
9B	SDPSoaxn	
9C	SPDnax	
9D	DSPDoaxn	
9E	DSPDSaoxx	
9F	PDSxan	
A0	DPa	
A1	PDSPnaoxn	
A2	DPSnoa	
A3	DPSDxoxn	
A4	PDSPonoxn	
A5	PDxn	
A6	DSPnax	
A7	PDSPoaxn	
A8	DPSoa	
A9	DPSoxn	
AA	D	
AB	DPSono	
AC	SPDSxax	
AD	DPSDaoxn	
AE	DSPnao	
AF	DPno	
B0	PDSnoa	
B1	PDSPxoxn	
B2	SSPxDSxox	
B3	SDPanan	
B4	PSDnax	
B5	DPSDoaxn	

B6	DPSDPaoxx	
B7	SDPxan	
B8	PSDPxax	
B9	DSPDaoxn	
BA	DPSnao	
BB	DSno	**MERGEPAINT**
BC	SPDSanax	
BD	SDxPDxan	
BE	DPSxo	
BF	DPSano	
C0	PSa	**MERGECOPY**
C1	SPDSnaoxn	
C2	SPDSonoxn	
C3	PSxn	
C4	SPDnoa	
C5	SPDSxoxn	
C6	SDPnax	
C7	PSDPoaxn	
C8	SDPoa	
C9	SPDoxn	
CA	DPSDxax	
CB	SPDSaoxn	
CC	S	**SRCCOPY**
CD	SDPono	
CE	SDPnao	
CF	SPno	
D0	PSDnoa	
D1	PSDPxoxn	
D2	PDSnax	

D3	SPDSoaxn	
D4	SSPxPDxax	
D5	DPSanan	
D6	PSDPSaoxx	
D7	DPSxan	
D8	PDSPxax	
D9	SDPSaoxn	
DA	DPSDanax	
DB	SPxDSxan	
DC	SPDnao	
DD	SDno	
DE	SDPxo	
DF	SDPano	
E0	PDSoa	
E1	PDSoxn	
E2	DSPDxax	
E3	PSDPaoxn	
E4	SDPSxax	
E5	PDSPaoxn	
E6	SDPSanax	
E7	SPxPDxan	
E8	SSPxDSxax	
E9	DSPDSanaxxn	
EA	DPSao	
EB	DPSxno	
EC	SDPao	
ED	SDPxno	
EE	DSo	**SRCPAINT**
EF	SDPnoo	

F0	P	**PATCOPY**
F1	PDSono	
F2	PDSnao	
F3	PSno	
F4	PSDnao	
F5	PDno	
F6	PDSxo	
F7	PDSano	
F8	PDSao	
F9	PDSxno	
FA	DPo	
FB	DPSnoo	**PATPAINT**
FC	PSo	
FD	PSDnoo	
FE	DPSoo	
FF	1	**WHITENESS**

Line: ROP2

01	0	**BLACKNESS**
02	DPon	
03	DPna	
04	Pn	
05	PDna	
06	Dn	
07	DPx	
08	DPan	
09	Dpa	
0A	PDxn	
0B	D	
0C	DPno	

0D	P	
0E	PDno	
0F	DPo	
10	1	**WHITENESS**

9.7.2 BitBlt

Actually, BitBlt with ROP3 can represent all of the Bitblt functions. But most of them are never used. Here list some frequently used bitblt.

Rectangle Fill

Rectangle Fill is to fill the destination with solid color. The color is specified in the foreground color register. The fill size and position are also specified in registers. Rectangle Fill operation is the same as BitBlt with ROP3 and the code 0xF0. This function always used in fast clean the display screen

Pattern Copy

Pattern Copy is to fill the destination with 8x8 pattern. The pattern is specified in the pattern registers. The fill size and position are also specified in registers. The 8x8 pattern will be repeatedly tiled to the pre-defined rectangle. Pattern Copy operation is the same as BitBlt with ROP3 and the code 0xF0.

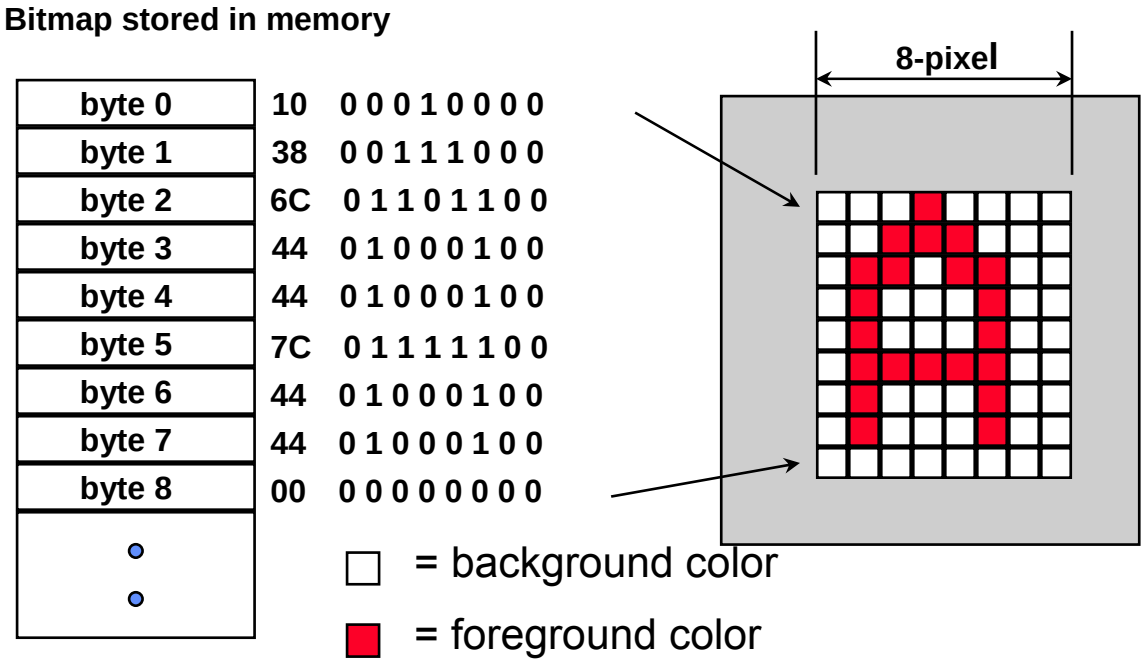
Source Copy

Source Copy is to copy a source rectangle to the destination. Source Copy operation is the same as BitBlt with ROP3 and the code 0xCC.

Because the fill rate is one pixel per cycle, when the clock frequency is 48MHz, Glamo 3362 can provide 48M pixels per second. For example, with VGA LCD display (640x480). Glamo 3362 2D engine can draw full screen more than one hundred times per second. The BitBlt function is always used in smoothly scrolling up and down or graphic animation.

9.7.3 Color Expansion

Color expansion also called font expansion. It used to expand a monochrome bitmap to color bitmap. With this function, user can save the font pattern size to 1 bit per pixel and run-time expanse to two colors font character. User can define his own font pattern with any size and use the color expansion function to expand to the display screen.



9.7.4 Transparent BitBlt

It copies a rectangular region of one bitmap into another, with some transparent pixels depends on the source and destination key. With this function user can perform the multi-layers of graphic overlay. Glamo 3362 provides 16 ROP.

ROP	Source	Destination	Read Destination
0000	Never	Always	No
0001	SRC key and DST key	Otherwise	Yes
0010	Not SRC key and DST key	Otherwise	Yes
0011	DST key	Otherwise	Yes
0100	SRC key and not DST key	Otherwise	Yes
0101	SRC key	Otherwise	No
0110	SRC key xor DST key	Otherwise	Yes
0111	SRC key or DST key	Otherwise	Yes
1000	Not SRC key and not DST key	Otherwise	Yes
1001	SRC key xnor DST key	Otherwise	Yes
1010	Not SRC key	Otherwise	No

1011	Not SRC key or DST key	Otherwise	Yes
1100	Not DST key	Otherwise	Yes
1101	SRC key or not DST key	Otherwise	Yes
1110	Not SRC key or not DST key	Otherwise	Yes
1111	Always	Never	No

The function is always used to merge one character's image into the background image for game application or animation.

9.7.5 Line Drawing

Glamo 3362 uses Bresenham's algorithm to draw a line. The drawing line could be either a solid line or a dashed line. The dashed line pattern can be defined in registers. The line color is define in foreground color register

9.7.6 Stretch

The stretch function is to scales up or down a rectangular region of one bitmap into another. The scaling factor of X axis and Y axis are independent. That is, user can scale up in X axis while scale down in Y axis. This function can be used in the display effect of pop-up menu and window creation or close.

9.7.7 Alpha Blending

Alpha blending means that you can copy a rectangular region of one bitmap into another. Glamo 3362 supports two kinds of alpha blending effects. One is constant alpha value. The formula is as follows:

Dst.R = (Src.R * Ac + (255 – Ac) * Dst.R) / 255

Dst.G = (Src.G * Ac + (255 – Ac) * Dst.G) / 255

Dst.B = (Src.B * Ac + (255 – Ac) * Dst.B) / 255

For constant alpha value, the color format of source supports ARGB8888, ARGB1555, ARGB4444 and RGB565. The color format of destination supports RGB565 and RGB555. Constant alpha value can be used in the display effect of fan-in and fan-out

The other is per-pixel alpha value in the source bitmap. source format The formula is as follows:

Dst.R = (Src.R * Src.A + (255 – Src.A) * Dst.R) / 255

Dst.G = (Src.G * Src.A + (255 – Src.A) * Dst.G) / 255

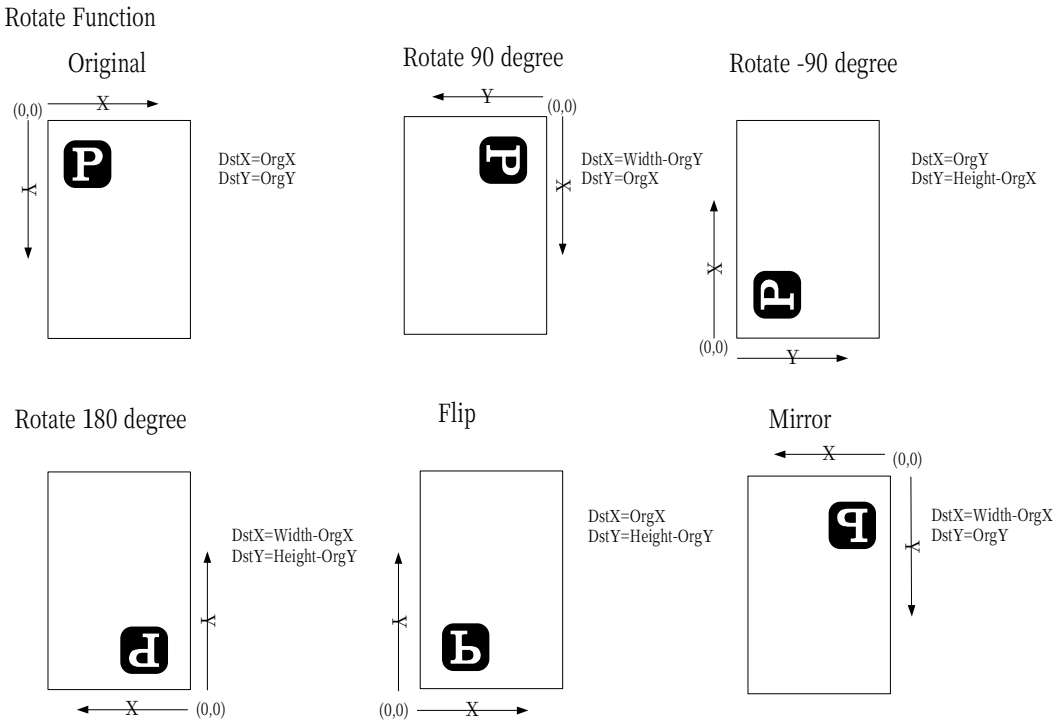
Dst.B = (Src.B * Src.A + (255 – Src.A) * Dst.B) / 255

For per-pixel alpha value, the color format of source supports ARGB8888, ARGB1555 and ARGB4444. The color format of destination supports RGB565 and RGB555. Per-pixel alpha value can be used in the display effect of merging two image and enhance the central part.

9.7.8 Rotation

Glamo 3362 2D engine supports five kinds of rotation, 90^0, -90^0, 180^0, Mirror, Flip

The definition of coordinate rotates as follows:



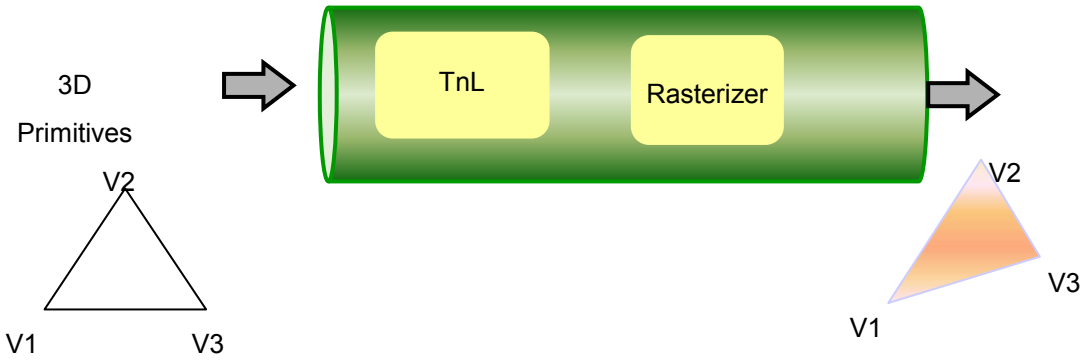
9.8 3D Graphics Engine

Glamo embeds 3D hardware accelerator which fully compliant with OpenGL ES v1.0, v1.1 and Mobile D3D. Glamo supports wide range of data types (8-bit, 16-bit, 32-bit, fixed-point and floating-point) to reduce the computing load of processor. Glamo supports vertex buffering technology which can reduce the bus loading. Glamo supports all of 3D pipeline (Transform, cull, lighting, clipping, setup, and Rasterizer), it is very important in mobile application. Only the hardware accelerator can give the high quality and high performance 3D application in mobile device.

9.8.1 3D Engine Pipeline

The following diagram shows the schematic diagram of the 3D accelerator. Commands are fed into the 3D engine on the left. The data formats of the commands are geometric primitives which described by vertices: points, line

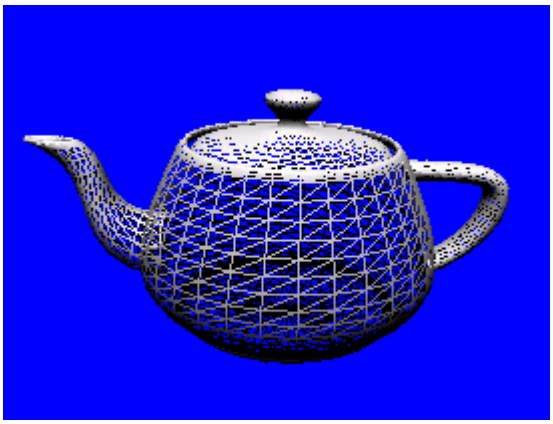
segments, and triangles. The first stage of 3D pipeline is vertices transform, lighting and clipped to a viewing volume. The second stage of 3D pipeline is rasterization the primitives. At this stage, the primitive will be converted to a two-dimensional image. Each pixel of this image contains color and depth information. Rasterizing a primitive consists of two parts. The first is to determine which pixels are occupied by the primitive. The second is assigning a depth value and color values to each pixel. The color value of pixels are determined by the shading operation, texture mapping, fog or alpha blending.



3D Pipeline Architecture

9.8.2 3D primitives

A 3D primitive is a collection of vertices to form a 3D object. The simplest primitive is 3D coordinate point set which is called a point list. Often, 3D primitives are polygons. The simplest polygon is a triangle. You can combine the triangles to form large, complex object. Following Figure shows a teapot triangle mesh.

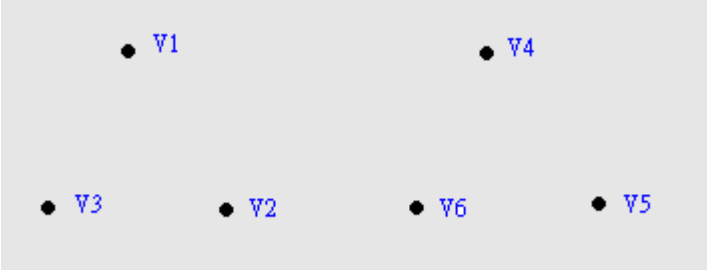


3D objects – teapot triangle mesh

Glamo can support following types of primitives.

Point lists

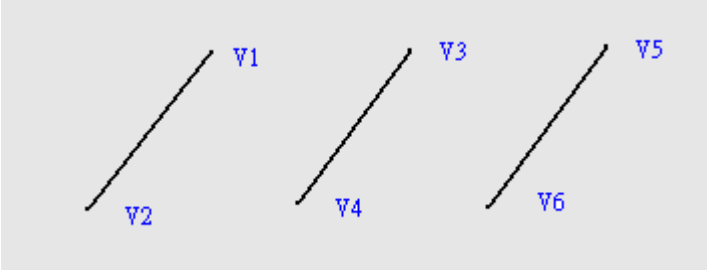
A point list is a collection of vertices that are rendered as isolated points. You can apply the materials, lights and textures to a point list. Following figure shows the rendered point list.



Example of point lists

Line Lists

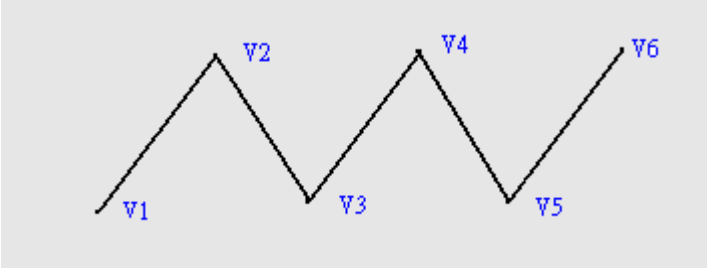
A line list is a list of isolated, straight line segments. You can apply materials, lights and textures to a line list. The colors in the material or texture appear only along the lines drawn, not at any point in between the lines. Following figure shows the rendered line list.



Example of line lists

Line Strips

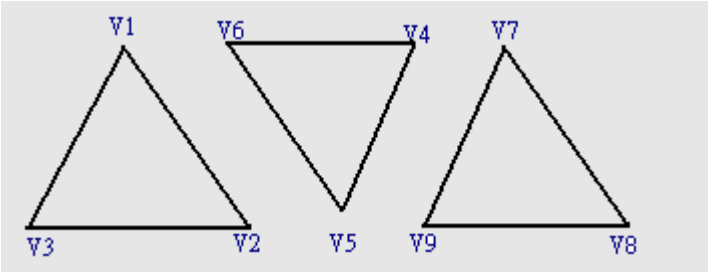
A line strip is a primitive which composed of connected line segments. Following figure shows the rendered Line Strips.



An example of line lists

Triangle Lists

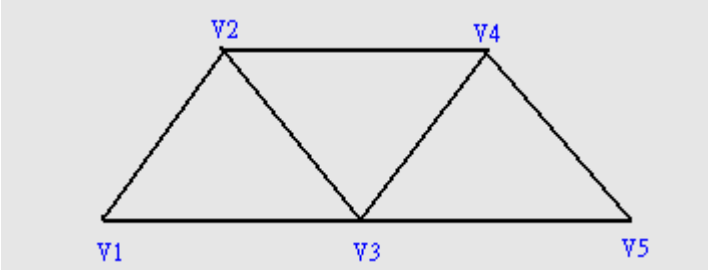
A triangle list is a list of isolated triangles. They may or may not be near each other. A triangle list must have at least three vertices and the total number of vertices must be divisibled by three. Following figure shows the rendered Triangle Lists.



An example of triangle lists

Triangle Strips

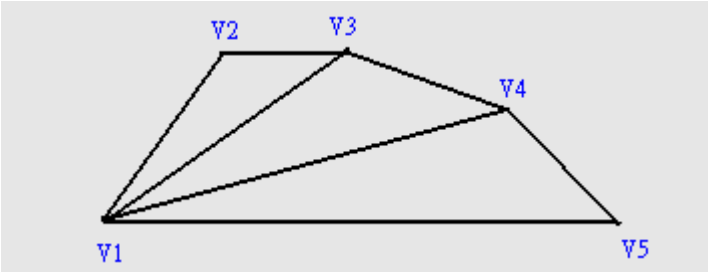
A triangle strip is a series of connected triangles. Because the triangles are connected, the application does not need to repeatedly specify all three vertices for each triangle. Most objects in 3D scenes are composed by triangle strips to reduce the vertices number and save memory sapce and improve processing time. Following figure shows the rendered Triangle Strips.



An example of triangle strips

Triangle Fans

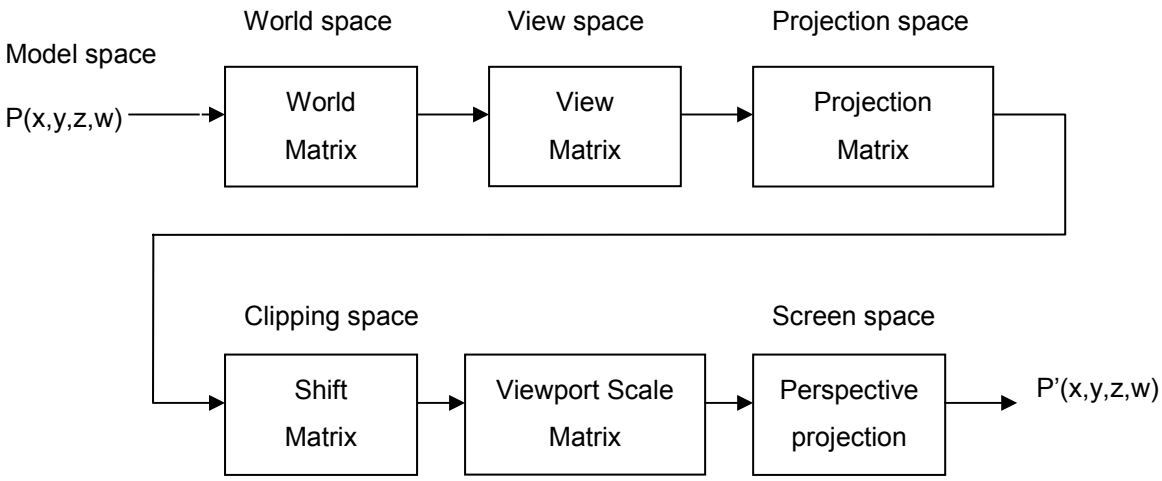
A triangle fan is similar to a triangle strip, excepting that all the triangles share one vertex. Following figure shows the rendered Triangle Fans.



An example of triangle fans

9.8.3 Transform

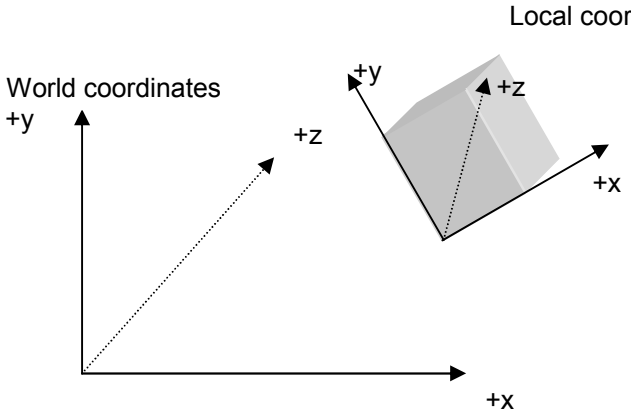
The transformation engine applies the world, view, and projection transformation and clips vertices to the view port. Following figure illustrates the transformation pipeline.



Transformation pipeline

World Space

The world transformation changes coordinates from model space to world space. The vertices are defined with respect to a common origin to all objects in the scene. World matrix includes the combination of translation, scaling and rotation operation. Following figure shows the relationship between the world coordinate system and a model's local coordinate system.



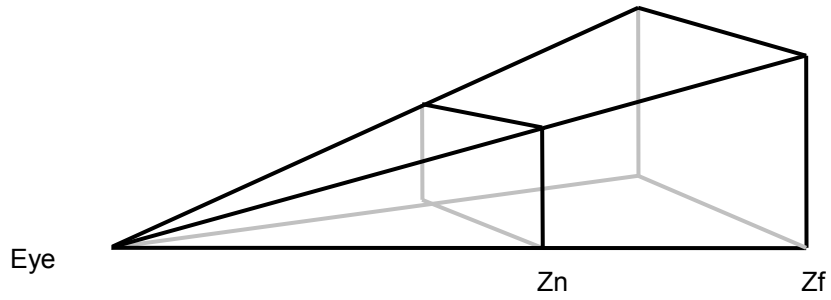
Relationship of world space and model's local space

View Space

View space is also known as camera space or eye space. The view transformation puts the viewer in world space and transforms vertices into view space. In view space, the viewer is at the origin and looks into the positive z-direction.

Projection space

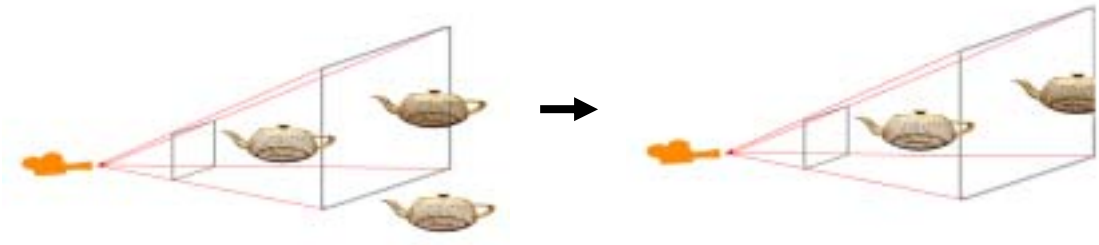
Projection space transform is to project the 3D view space onto a 2D space with depth. Projection space is a homogeneous space in which all vertices have x- and y-coordinates that range from –1.0 to 1.0, and a z-coordinate that ranges from 0.0 to 1.0. Following figure shows the view frustum. A viewing frustum is 3-D volume in a scene positioned relative to the view point. The shape of the volume affects how models are projected from view space onto the screen. The most common type of projection, a perspective projection, is responsible for making objects near the viewer appear bigger than objects in the distance.



View frustum

Clipping Volumes

D3DM and OpenGL ES define the clipping volume in clipping space. Any vertices that have an x, y, or z component outside these ranges are clipped. Following figure shows an example of clipping. After the clipping stage, we can use the view-port location and dimensions to scale the vertices to fit a rendered scene into the appropriate location on the target surface.



Example of clipping

Screen Space

After passing through the geometry pipeline, vertices have been transformed, clipped, and scaled to fit in the view-port on the render-target surface, After the perspective projection transform, we get the coordinate in screen space.

9.8.4 Lighting

Glamo support all the lighting type defined in D3DM and OpenGL ES (Point Light, Directional Light and Spot Light). Glamo can calculate all the Ambient Light, Diffuse Light, Specular Light and Emissive Light terms to determine the diffuse and specular color. Glamo supports up to 8 active light, and two side lighting in OpenGL.

Ambient

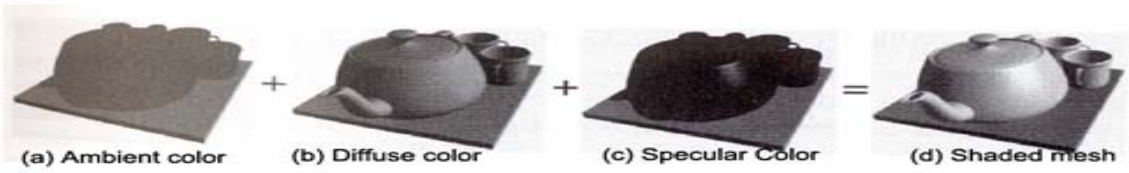
Ambient illumination is light that has been scattered by the environment, its direction is impossible to determine.

Diffuse

A diffuse light comes from one direction. Once it hits a surface, it is scattered equally in all directions. The diffuse color is decided not only by the diffuse color of the light and diffuse material of the object, but also by the surface normal and the light direction. However, the eye position will not affect the diffuse color.

Specular

Specular light comes from one direction, and it tends to bounce off the surface in a specific direction. It is decided not only by the specular color of the light and specular material of the object, but also by the surface normal, the light direction and the eye vector.

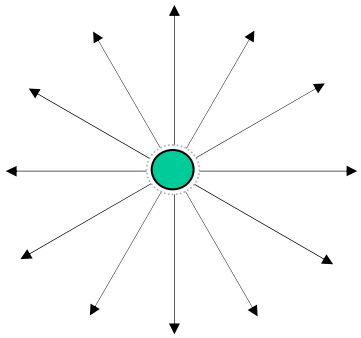


Light Type

Glamo support three light type: point light, directional light and spot light.

Point Light

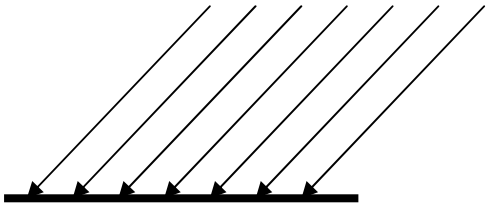
Point lights have to define color and position, but no single direction. The light illumination is equally in all directions. A good example of point light is a light bulb. Point light is affected by attenuation and range. Following figure shows an example of point light.



Example of point light

Directional Light

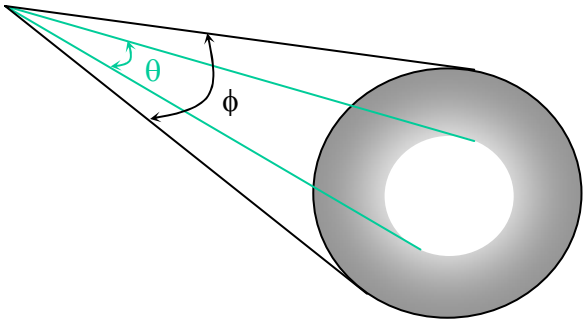
Directional lights have only color and direction, not position. All light generated by directional light travels through scene in the same direction. You can imagine a directional light as a light source at near infinite distance, such as the sun. Directional lights are not affected by attenuation or range. Following figure shows an example of directional light.



An example of direction light

Spot Lights

Spotlights have color, position, and direction in which they emit light. Spotlights are affected by falloff, attenuation, and range. Following figure shows the relationship between the value and how they can affect a spot light's inner and outer cones of light. The "θ" value is the radian angle of the spotlight's inner cone, and the "Φ" value is the angle for the outer cone of light. The falloff value controls how light intensity decreases between the outer edge of the inner cone and the inner edge of the outer cone.



An example of spot light

9.8.5 Shading

Shading modes determine the intensity of color and lighting at any pixels on a polygon face. Glamo supports two shading modes: Flat shading and Gouraud shading.

Flat shading

In the flat shading mode, render using the color of the polygon material at its first vertex as the color for the entire polygon. 3D objects that are rendered with flat shading have visibly sharp edges between polygons if they are not coplanar. Following figure shows an example of flat shading.

An example of flat shading

Gouraud shading

In Gouraud shading mode, render all the pixels in the triangle using the color interpolation of three vertices color. Followin figure shows an example of Gouraud shading. Gouraud shading makes the surface of the object appear curved and smooth.

An example of Gouraud shading

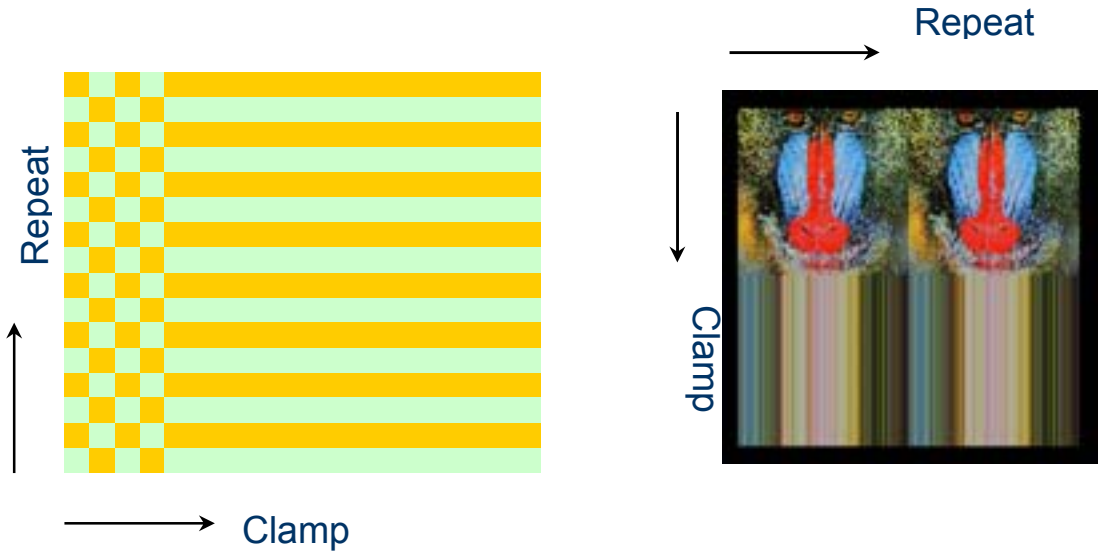
9.8.6 Textures Mapping

As we know texture mapping is a important technology in today's 3D game and 3D graphic, without texture mapping technology the world and model that are rendered would be far from realistic. Glamo supports an extensive texturing feature to provide developers easily accessing to advanced texturing techniques. Glamo supports maximum texture up to 256x256. Glamo also supports none-power-of-two texture and multiple textures. Following figure shows an example of texture mapping.

An example of texture mapping

Texture address mode

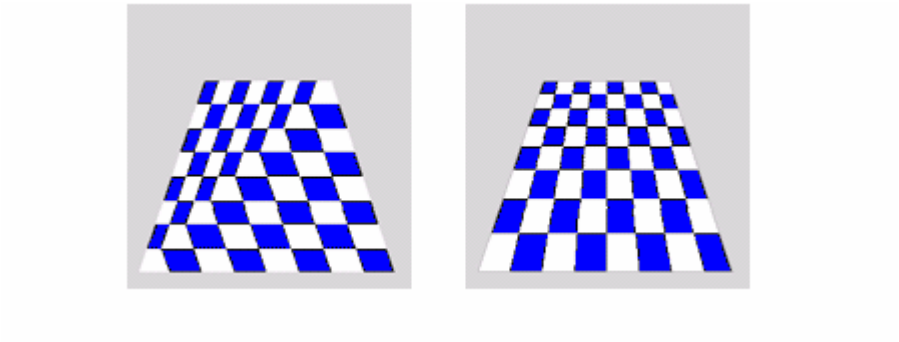
Typically, the texture coordinates that we assign to a vertex will be in the range of 0.0 to 1.0 inclusive. However, by assigning texture coordinates outside that range, we can create certain special texturing effects. We can control the texture coordinates that are outside the [0.0, 1.0] range by setting the texture addressing mode. Glamo supports all the addressing mode defined in D3DM and OpenGL ES (Repeat, Clamp, Mirror and Border). Following figure shows an example of texture address mode.



An example of texture address mode

Perspective Correction

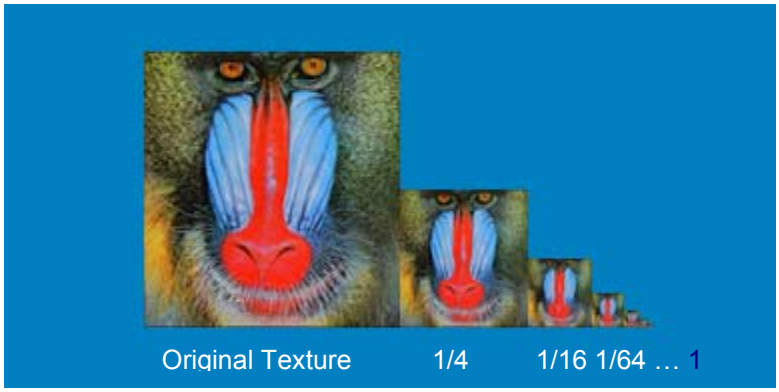
Without the perspective correction, the texture mapping will cause the incorrect in the 3D scene as shows in the following figure. Glamo supports per pixel perspective correction. Application can also enable or disable perspective correction. D3DM and OpenGL ES perspective correction are enabled by default.



Example of without perspective correction and with perspective correction

Texture MIPMAP

A mipmap is a sequence of textures, each of which is a progressively lower resolution representation of the same image. Mipmap textures are popularly used in 3D scenes to decrease the rendering time. They also improve the object realism. Glamo supports mipmap structure from 1x1 to 256x256. Following figure shows an example of mipmap texture.

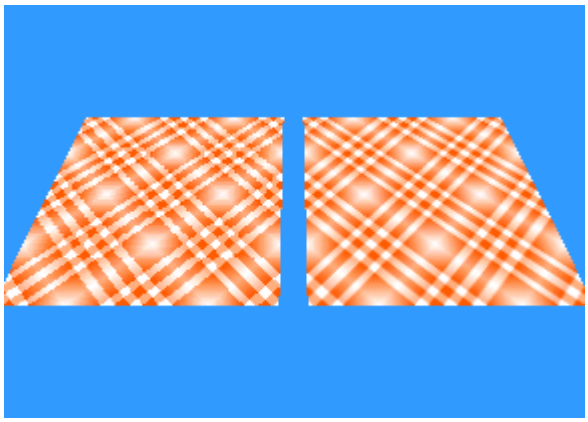


An example of mipmap texture

Texture filtering

The pixels in the texture map are called texel. When a texture filter operation is performed, the texture can be magnified or minified. In other words, it is mapped onto a primitive image that is larger or smaller than itself. Magnification of a texture will cause that many pixels are mapped to one texel. The result is chunky appearance. Minification of a texture will cause that a single pixel is mapped to many texels. The result is blurry or aliased appearance. To resolve these problems, some blending of the texel colors must be filtered before mapping to the object. Glamo supports three types of texture filtering—linear filtering, nearest filtering and mipmap filtering. For

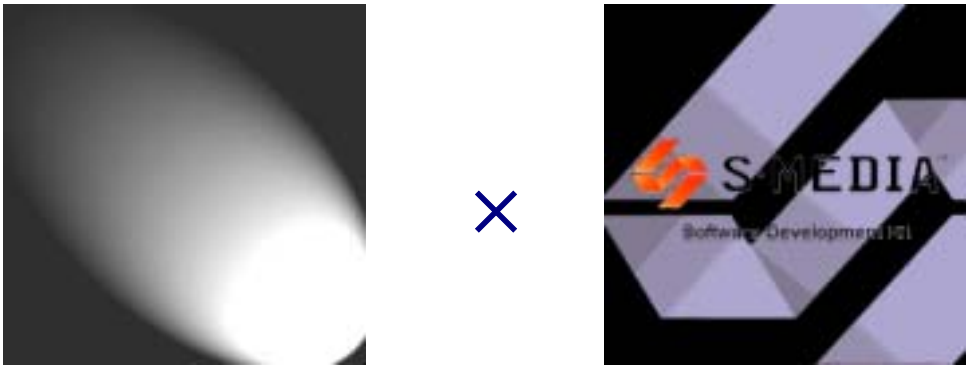
mipmap filtering, Glamo can support NMN (Nearest-Mipmap-Nearest), NML (Nearest-Mipmap-Linear), LMN (Linear-Mipmap-Nearest), and LML (tri-linear) filtering. Following figure shows an example of texture filtering. The texture filter mode for left plane is nearest, the right plane is bi-linear. You can see the different between two planes.



An example of texture filtering

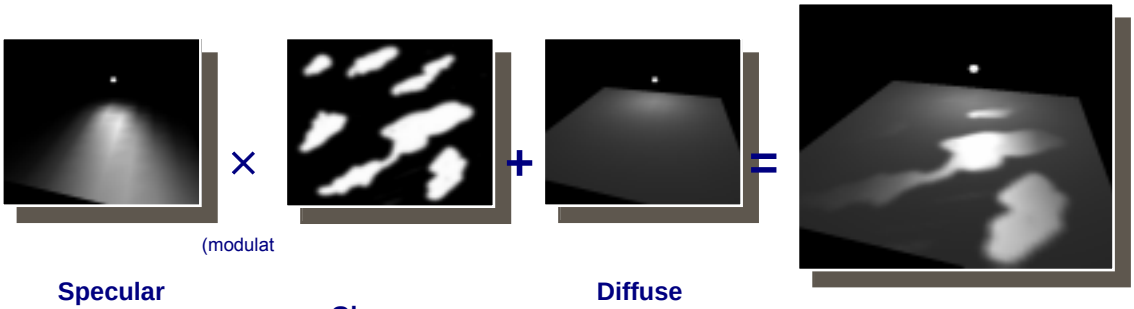
Texture Blending

D3DM and OpenGL ES can produce transparency effects by blending a texture with a primitive's color. It can also blends multiple textures onto a primitive. Glamo supports a texture combiner which can blend diffuse color, specular color and multiple textures. Glamo can supports all the texture blending mode defined in D3DM and OpenGL ES. Following figure shows an example of multiple textures and example of texture blending.



Light Map Combine with Texture

Example of multiple textures

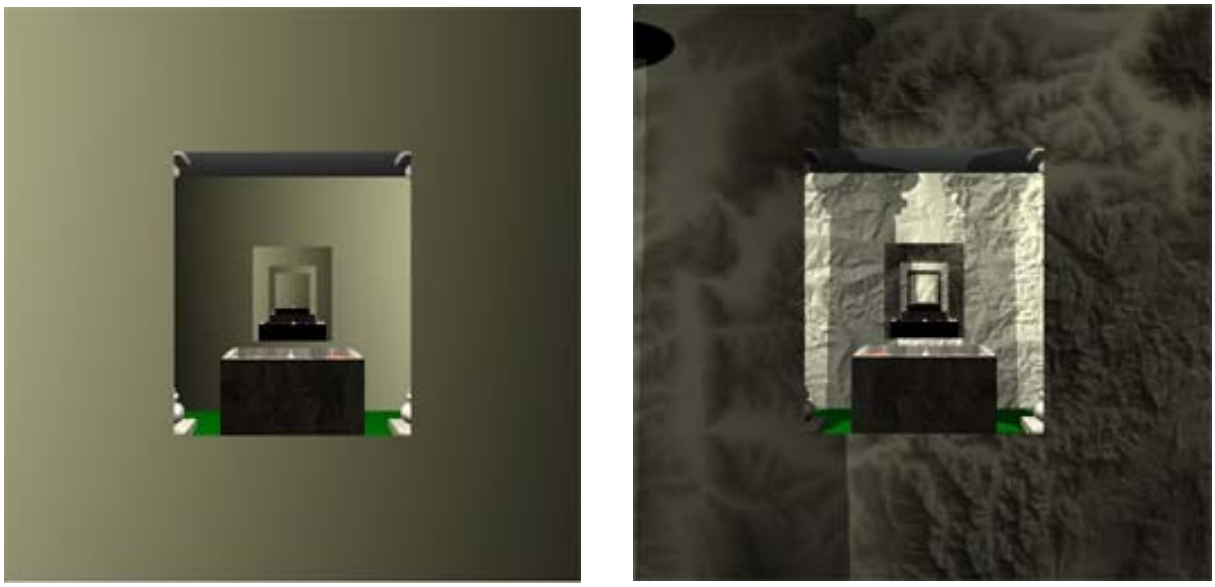


Example of texture blending

Bump Mapping

Bump mapping is a special form of specular or diffuse environment mapping that simulates the reflections of finely tessellated objects without requiring extremely high polygon counts. Bump mapping relies on blending multiple

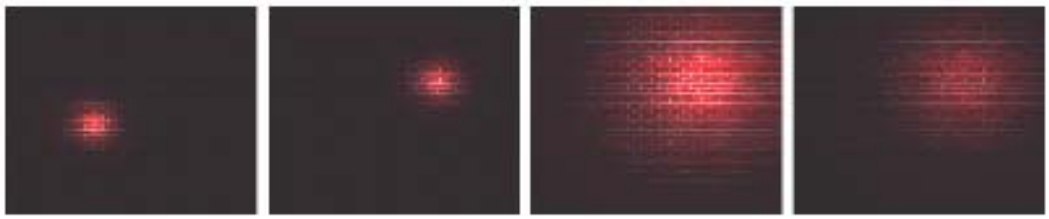
textures. Glamo supports multiple textures up to two and also have a programmable combiner can do such as dp3 or other operation. Following figure shows an example of bump mapping.



Example of bump mapping (a)without bumpmap (b) with bumpmap

Texture Transform

Glamo provides transform the texture coordinates of vertices by applying a 4×4 matrix. It is called the texture transform. The texture coordinate can be scaled, rotated, translated, projected, sheared or any bind of them. Texture coordinate transformations are useful to produce special effects without modifying the texture coordinates of existing geometry. You can use simple translation or rotation matrices to animate textures on an object. Following figure shows an example of texture transform.



Origina

Translate

Spotlight

Texture

Scale

Spotlight

Texture

Change Base

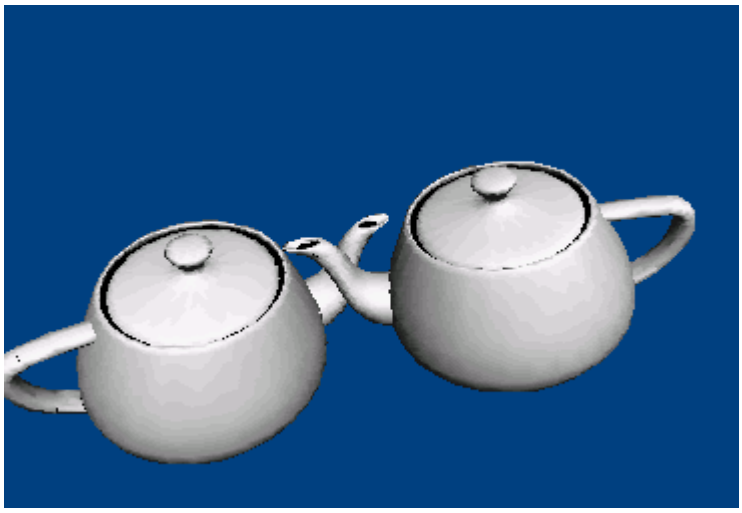
Polygon

Use texture matrix to perform spotlight texture coordinates transformations.

Example of texture transform

9.8.7 Z Buffer

Z buffer is a surface that stores depth information used in rending. When a 3D scene is rasterized with depth buffering enabled, each point on the rendering surface is tested. If the depth test is passed, means the new object is near to the viewer. Then the color can be updated to the target surface. The new depth value also can update to the Z buffer. Glamo supports the Z test and Z buffer. Glamo also supports polygon offset. Flolowing figure shows the result of two 3D objects with Z test.



The result of two 3D objects with z test.

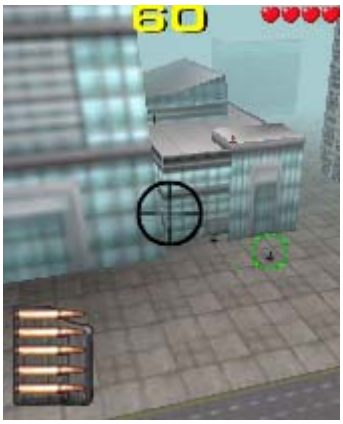
9.8.8 Stencil buffer

Applications use the stencil buffer to mask pixels on the image. The mask controls whether the pixel is drawn or not. Using stencil buffer can do some special effects. You can use the stencil buffer to composite 2D or 3D images onto a 3D scene. You also can do the special effects that are commonly used in movies, such as dissolves, swipes, and fades. For more abstract effects, such as outlining and silhouetting, you also can use stencil buffer to do it. Even the shadow effect also can be done by stencil buffer. Following figure shows an example of shadow which is done by stencil buffer.

Example of shadow

9.8.9 Fog

Adding fog to a 3D scene can enhance realism. Fog is implemented by blending the color of objects in a scene with a chosen fog color based on the depth of an object in a scene, or its distance from the viewpoint. Glamo provides two fog calculation, pixel fog and vertex fog. Pixel fog is implemented in the render pipeline and calculated the fog by pixel. Vertex fog is implemented in the light engine and calculated the fog by vertex and then, calculated each pixel fog value by shading. For different fog effect, Glamo supports the three kinds of fog types, linear, EXP and EXP2. Following figures shows a scene with fog effect.



The 3D scene with fog effect

9.8.10 Alpha Blending

Alpha blending is a combination of two colors with transparency effects in computer graphics. The alpha channel is additional bits that can present the level of translucency. The value of alpha in the color range from 0.0

to 1.0, where 0.0 represents a fully transparent, and 1.0 represents a fully opaque. Glamo fully supports all the blending modes define in the OpenGL ES and D3DM. Following figure shows an example of alpha blending.

Example of alpha blending