Appendix C2

Video Support

VIDEO SUPPORT

1. INTRODUCTION

An important feature of the CL-GD546X is the ability to capture and display video from either an MPEG decoder, a TV decoder, or a software codec. This appendix describes video capture and VGA feature connector operations.

2. FUNCTIONAL DESCRIPTION

Figure C2-1 shows how video is supported by the CL-GD546X. There are three steps in this process:

- Video capture
- Stretch BitBLTs
- Color space conversion

The following sections describe each step. Not every step is required in each case; for example, video playback using a software codec would not require the video capture step.

Figure C2-1 illustrates an interlaced video frame based on the definitions above. Even fields are solid lines and odd fields are dotted lines.

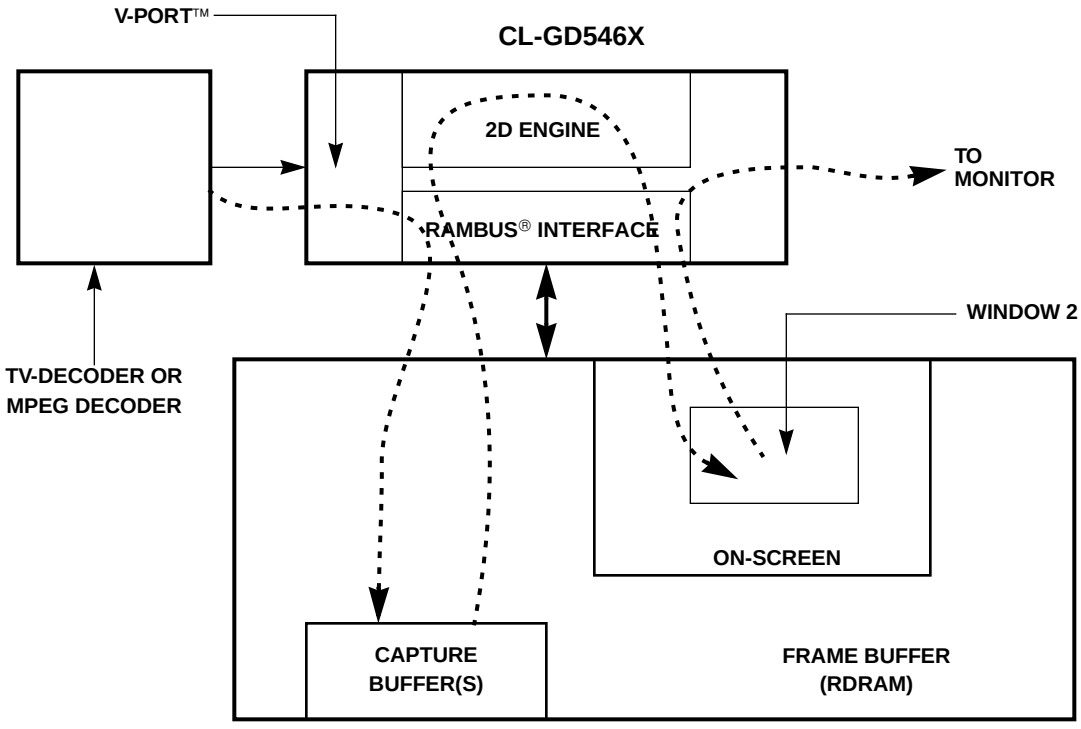


Figure C2-1. Video Overview

2.1 Video Capture

When the source of the video is a hardware video or MPEG decoder, the digital video must first be captured in the frame buffer. The V-Port provides a means to bring this video into the frame buffer asynchronously to the display refresh; the V-Port is compatible with most video and MPEG decoders on the market. The data is normally captured to an off-screen buffer where it is scaled and copied on-screen synchronous to display refresh (see Section 2.2), but it can be captured to on-screen memory (in which case no additional memory is required and the video is immediately visible on the screen).

The video is clocked into the V-Port using the decoder's pixel clock. Horizontal and vertical sync signals are required to frame the data, and an active video signal (if the decoder has one) may be used to prevent capturing data during blanking intervals. The polarities of these signals is programmable (CL-GD5464 only). See Section 3 for details of interfacing to different decoders. The Capture X Start, Capture X Stop, Capture Y Start, and Capture Y Stop registers can be used to restrict capture to a X,Y rectangle within the video frame (CL-GD5464 only). This can be used to prevent capture during blanking intervals if the decoder does not provide an active video signal, and can reduce the data captured if only a portion of the video frame is to be displayed.

Even if the whole frame is to be displayed, Capture X Stop and Capture Y Stop should always be programmed to the size of the video frame to stop capture of a scanline or frame independent of HSYNC or VSYNC, respectively, from the decoder. This is required to prevent writing outside the off-screen buffer if a HSYNC or VSYNC is missing for some reason (an example is a TV tuner that would not produce horizontal and vertical syncs at the proper time if the channel is changed).

Video can be interlaced or non-interlaced (progressive scan), and video can be captured for odd fields only, even fields only, both fields separately, or both fields together (interlaced capture). The pixel data bus can be programmed to be 8- or 16-bits wide, and the data clocked on the rising, falling, or both pixel clock edges.

The following definitions are used in this manual:

1) An interlaced frame consists of an *odd* field followed by an *even* field (that is, odd fields are sent first)

2) An *odd* field has the falling edge of VSYNC *between* two HSYNC pulses; an *even* field has the falling edge of VSYNC *at* an HSYNC pulse. In other words, an odd field starts with a half scanline and ends with a full scanline; an even field starts with a full scanline and ends with a half scanline.

3) *Even* field scanlines are displayed *above* the same scanline of an odd field. The first half-scanline of the odd field is not counted.

The V-Port packs the video into 64-bit words and accumulates them in a 16×64 FIFO. The FIFO is split in two halves and, when each half is full, the V-Port writes the data to the frame buffer. At the end of each scanline any remaining data in the FIFO is written to the frame buffer.

Data may be single- or double-buffered. Buffer 0 X address and Buffer 0 Y address define the location of the first buffer, and Buffer 1 X address and Buffer 1 Y address define the location of the second buffer (double-buffering only).

2.2 Resize BitBLTs and Auto-BitBLTs

Unless video was captured directly to on-screen memory, the next step is to copy the video from the off-screen to on-screen. During the process of copying, three more actions can be performed:

1) Scaling the video,

2) Color space conversion (typically from YUV 4:2:2 to RGB 5:5:5 or RGB 8:8:8), and

3) Synchronization to the display frame rate

The CL-GD5464 Resize BitBLT instruction supports arbitrary X,Y scaling with pixel replication or interpolation in X and Y directions. The Resize BitBLT instruction is used to scale the video while it is being copied from an off-screen buffer to on-screen; if no scaling is required a normal BitBLT instruction is more efficient.

The core of the resize BitBLT engine is a pair of DDA (discrete differential analyzer) or format conversion engines, one each for X and Y. For a complete description see the *Laguna VisualMedia™ Accelerators Family — CL-GD546X Volume II (Software Reference Manual, Second Edition, September 1996)*.

Interpolation is a four-phase linear interpolation in both the X and Y directions.

2.3 Windows

There are three windows in CL-GD546X. Windows 1 and 3 have auto-BitBLT capability. This allows an off-screen area to be BLT'ed into an on-screen area automatically synchronized to screen refresh with scaling. These two windows are intended for software codecs or some other software method of building an image off-screen. If the pixels in Windows 1 and 3 are to be interpreted differently than those in the background, a 9-bit RDRAM must be used.

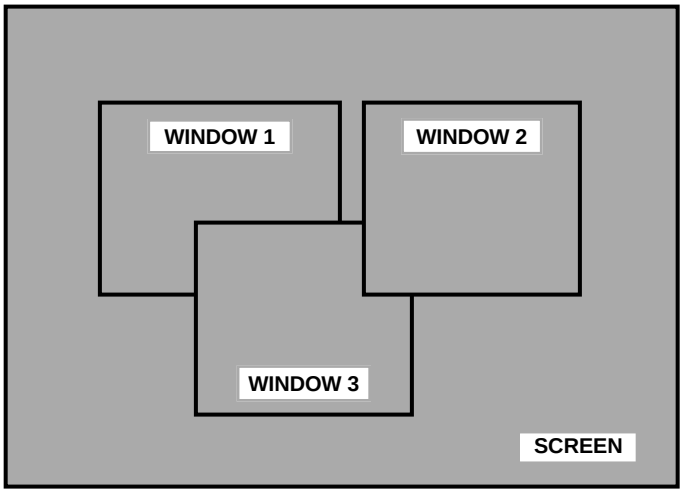
Window 2 has the capabilities of windows 1 and 3 with two important additions. First, an internal signal is available that indicates when window 2 is displayed. Regardless of the RDRAM width, this signal can change pixel interpretation in the video pipeline. Second, window 2 is associated with the V-Port, which allows a hardware capture of video data.

There is no hardware priority between the windows. Software drivers must deal with occlusion between overlapping windows.

Figure C2-2 and Figure C2-3 on page C2-5 illustrate the capabilities that are available with 9- and 8-bit RDRAMs, respectively.

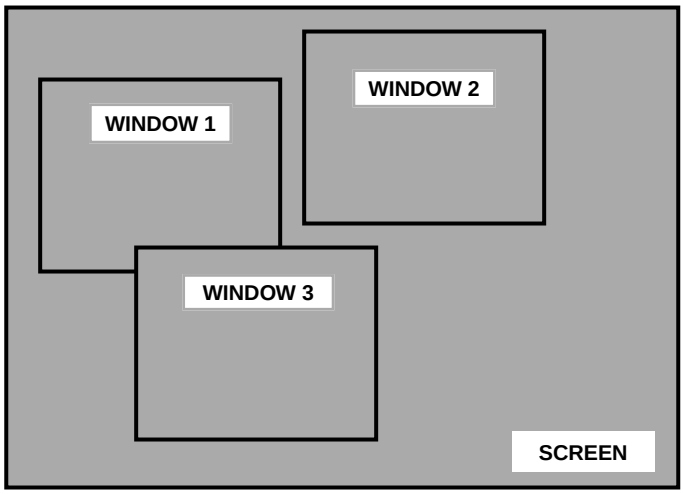
If 9-bit RDRAMs are used, then each pixel has a bit indicating whether it is in a video window. This allows each window to use the graphics pixel format (the background) or the common video pixel format.

If 8-bit RDRAMs are used, then windows 1 and 3 must use the graphics pixel format (the same as the background). Window 2 can use the video pixel format if it is not occluded (that is, if it is a rectangle that can be defined with four corners). If window 2 is occluded, it must use the graphics pixel format.



NOTE: Each window can use the video pixel format or it can use the graphics pixel format. The video pixel format is common for whatever windows use it. The background uses the graphics format.

Figure C2-2. 9-bit RDRAM Windows



NOTE: Windows 1 and 3 must use the graphics pixel format (the background format). Window 2 can use the video pixel format only if it is not occluded. If window 2 is occluded, it must use the graphics pixel format.

Figure C2-3. 8-bit RDRAM Windows

2.4 Mixed Frame Buffer

This section provides an overview of the video capture and overlay capabilities. Programming and hardware details are not discussed.

The CL-GD546X can support a mixed-format frame buffer containing two pixel formats: graphics and video, but any pixel format (including YUV 4:2:2) can be defined as the graphics format, and any format can be defined as the video format. For example, a 'video window' can be defined where the pixels are in YUV 4:2:2 format (the native pixel format of the decoder), but the remainder of frame buffer is palettized. All pixel formats are converted to RGB 8:8:8 format in the video pipeline. In this example, the mixed-format capability allows no loss of video quality. Table C2-1 indicates how the formats can be mixed. A ✔ indicates the video formats that can be displayed with each graphics format.

Table C2-1. Mixed Frame Buffer Formats

Format	8-bpp Palettized	8-bpp Grayscale	8-bpp AccuPak™	16-bpp RGB	16-bpp YUV
8-bpp palettized	✔	✔	✔	✔	✔
8-bpp grayscale	✔	✔	✔	✔	✔
16-bpp RGB				✔	✔

2.5 Pixel Reformatting During Stretch BitBLTs

Figure C2-4 on page C2-7 shows an overview of the display memory as video capture takes place. Typically, the on-screen memory is at the beginning of the frame buffer. Window 2 is positioned somewhere in on-screen memory. The values programmed into the X_START_2, Y_START_2, X_END_2, and Y_END_2 registers are in terms of screen position.

The two capture buffers are off-screen. A third buffer contains a monochrome mask that is used to suppress writing while the auto-BitBLT is taking place. This is for occlusion.

The auto-BitBLT headers are stored off-screen. There is one header for each auto-BitBLT (odd buffer, even buffer). The auto-BitBLT headers contain register values that define the BitBLTs that move the image from the off-screen buffers to window 2 on-screen.

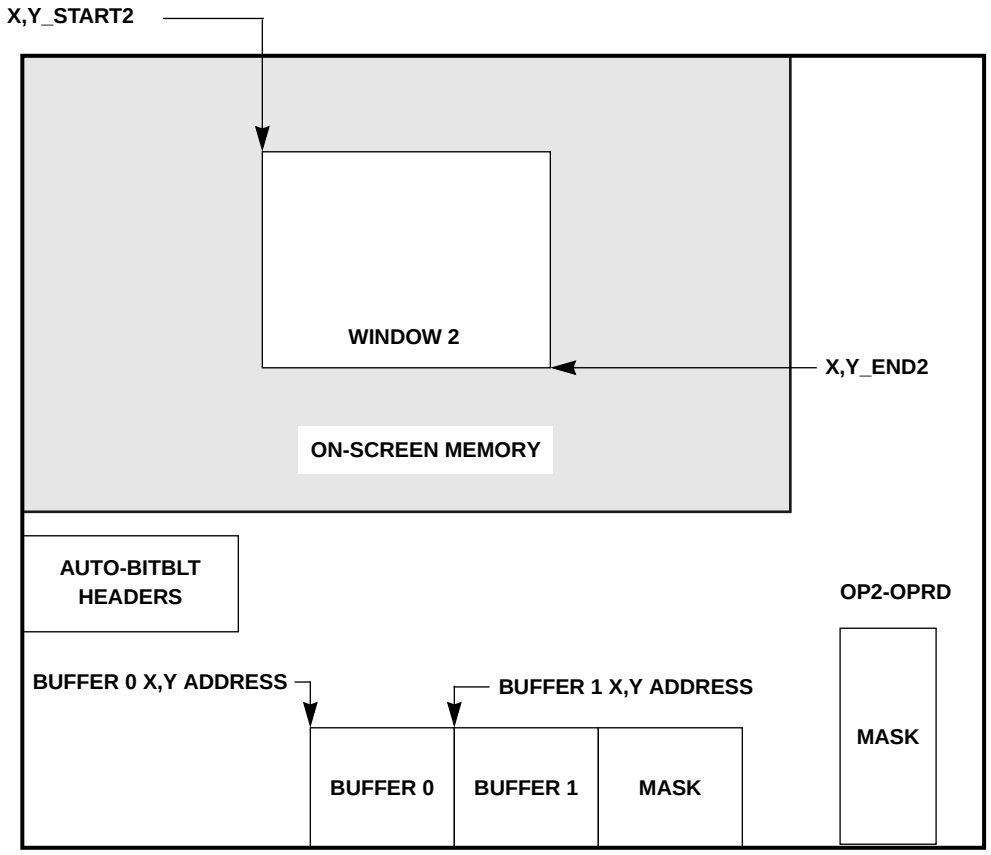


Figure C2-4. Frame Buffer Overview

2.6 Timing Overview

Figure C2-5 shows a sample of the timing involved. If video is being captured from an NTSC source and displayed on a 1024 × 768 display. The video field rate is 60 Hz and alternate fields are captured into alternate off-screen buffers. The display refresh rate is 75 Hz and window 2 is 240 scanlines. Since the NTSC frame rate is different from the CRT frame rate and they are not synchronous, the phase relationship constantly changes.

The top two lines in Figure C2-5 indicate when the odd and even fields are being captured into their respective buffers. Two buffers are necessary since there is no guarantee that the auto-BitBLT is able to occur during the TV retrace time. Auto-BitBLTs must wait until the safe time and also can be delayed by other BitBLTs. Each time a field is complete, the auto-BitBLT is armed. That is, a signal is made active that indicates that an auto-BitBLT can occur. This signal is labeled ‘ARM’ in Figure C2-5 and is shown as being active until the BitBLT is complete.

The movement of the video data into the on-screen area (the auto-BitBLT) must be synchronized with screen refresh to avoid having a portion of field n and a portion of field n + 1 on the screen. If this precaution is not taken, ‘tearing’ can occur when quickly moving objects are displayed. In the CL-GD546X, the times when auto-BitBLTs are prohibited are individually programmable for each of the three windows. As long as each window remains in the same vertical location on the screen, the safe time is fixed with respect to VSYNC.

When the auto-BitBLT is armed and the CRT refresh is in the safe time, the auto-BitBLT occurs. This is shown in the bottom trace of the timing diagram in Figure C2-5.

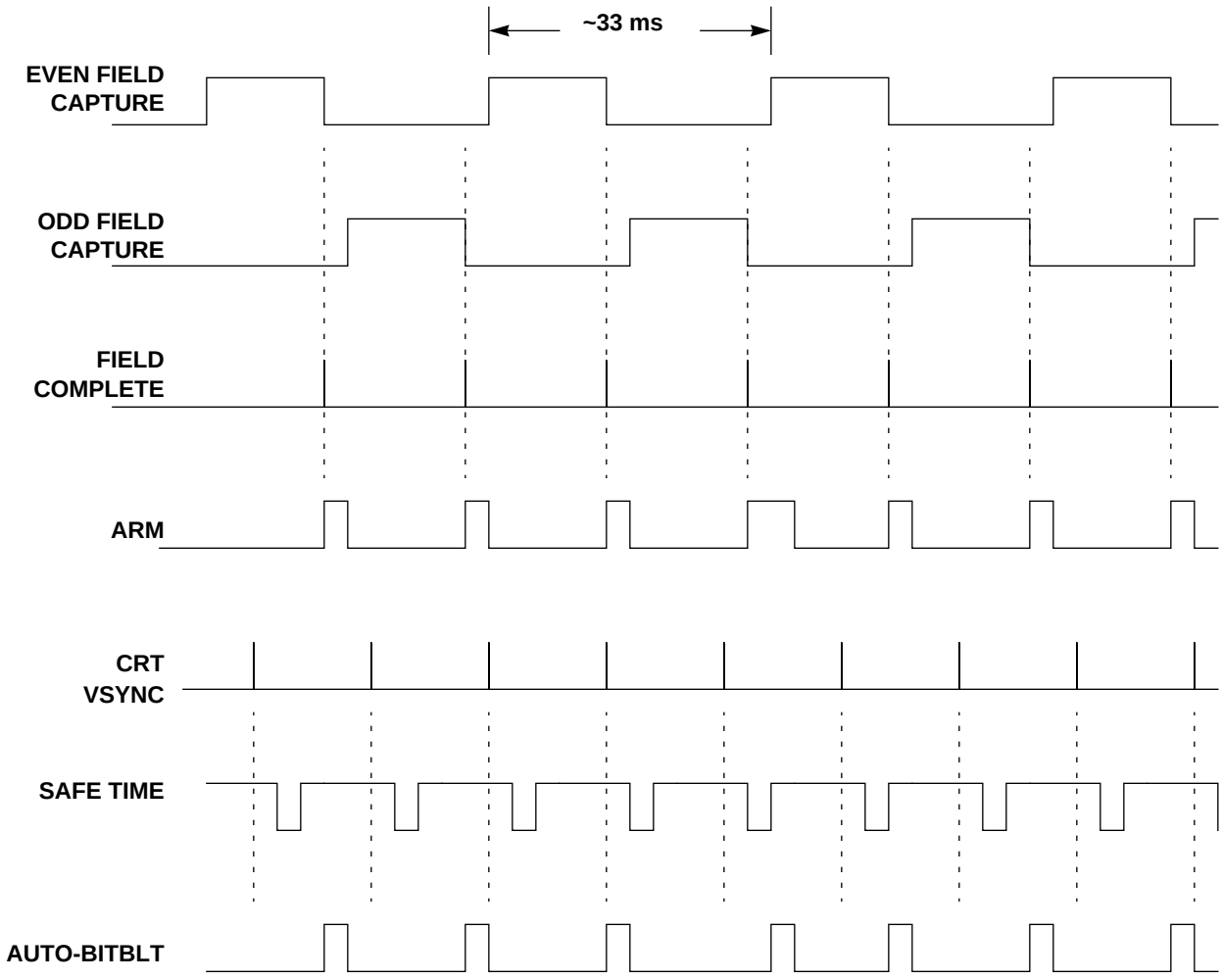


Figure C2-5. NTSC Capture and Display — Gross Timing (Do not scale)

3. VIDEO CAPTURE AND OVERLAY PINS

The V-Port mode is programmed into V-Port Control register (CL-GD5462) and V-Port Mode register (CL-GD5464) at MMI/O offset 10Ch. Table C2-2 shows how the CL-GD546X pins are connected for the overlay modes that are currently defined.

Table C2-2. Video Capture and Overlay Pin Definitions

CL-GD546X Pin		Feature Connector Mode		Capture Mode	
Number	**Name**	**I/O**	**Pin Name**	**I/O**	**Pin Name**
156	P0	I/O	P0	I	P0
155	P1	I/O	P1	I	P1
154	P2	I/O	P2	I	P2
153	P3	I/O	P3	I	P3
152	P4	I/O	P4	I	P4
150	P5	I/O	P5	I	P5
149	P6	I/O	P6	I	P6
148	P7	I/O	P7	I	P7
147	DCLK/PCLK	I/O	DCLK	I	PCLK
145	BLANK#/HREF	I/O	BLANK#	I	HREF
161	HSYNC	I/O	HSYNC		
162	VSYNC	I/O	VSYNC		
144	EVIDEO#/VACT	I	EVIDEO#	I	VACT
143	ESYNC#	I	ESYNC#		
142	EDCLK#/VREF	I	EDCLK#	I	VREF
141	VCLK				
140	P8			I	P8
139	P9			I	P9
138	P10			I	P10
137	P11			I	P11
135	P12			I	P12
134	P13			I	P13
133	P14			I	P14
132	P15			i	P15

3.1 Capture Mode

The V-Port is configured for Capture mode when the V-Port mode field of the V-Port Control register (MMI/O offset 10Ch) is programmed to '010b' (CL-GD5462 only) or when the DCLK mode field is not '00b' (CL-GD5464 only). Table C2-3 and Figure C2-6 illustrate how the CL-GD546X connects to the video decoders.

Table C2-3. Video Decoder Connections

Video Decoder	CL-GD546X	Direction	Functions	Note
A[7:0]	P[7:0]	to CL-GD546X	Video data (8-bit mode) Y data (16-bit mode)	
B[7:0]	P[15:8]	to CL-GD546X	CrCb data (16-bit mode only)	Optional
DCLK/PCLK	DCLK	to CL-GD546X	Pixel clock	
BLANK#/HREF	BLANK#	to CL-GD546X	Horizontal timing	
EDCLK#/VREF	EDCLK#	to CL-GD546X	Vertical timing	
EVIDEO#/VACT	EDVIDEO#	to CL-GD546X	Video available	
SDA	SDA	to decoder	I^2C data	I^2C bus controls the decoders
SCL	SCL	to decoder	I^2C clock	

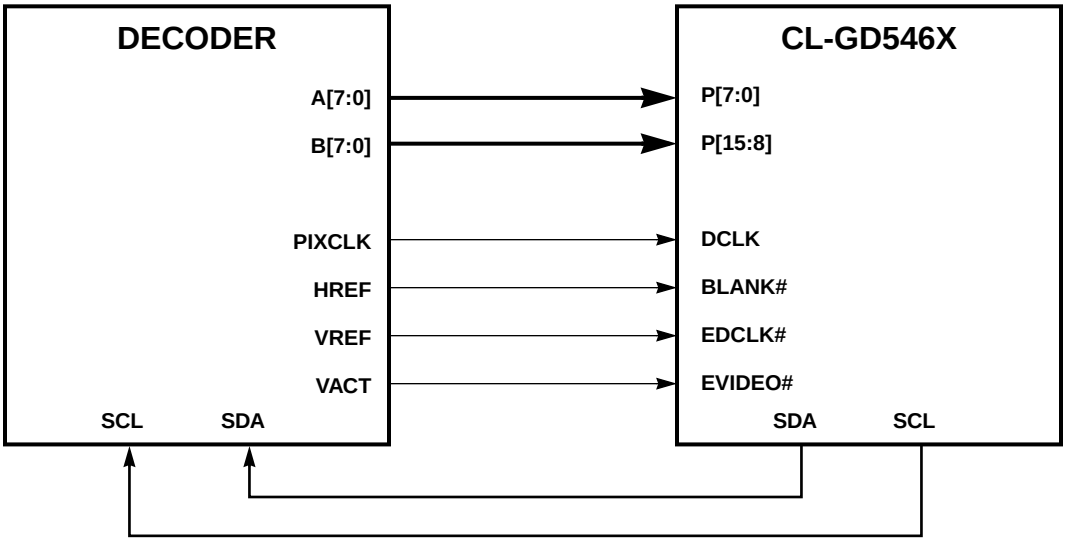


Figure C2-6. Decoder Connections

All capture timing is generated by the decoder. The V-Port portion of the CL-GD546X operates at the video rate. VREF is an indicator that a field is about to begin; HREF is an indicator that a scanline is about to begin. VACT is an indicator that valid data is available on the next falling edge of the DCLK input.

3.2 Feature Connector Mode

The V-Port is configured for standard a feature connector when the V-Port control field is '000b' (CL-GD5462 only), or when the DCLK mode field is '00b' (CL-GD5464 only). This is the power-on configuration. When the V-Port is configured for the feature connector, data is provided directly to the palette. This capability (and its three modes) is provided for compatibility with VGA. In all three cases, this mode is intended for operation up to 25 MHz.

3.2.1 Feature Connector Output Mode

If EVIDEO#, ESYNC#, and EDCLK# are all high, P[7:0], DCLK, BLANK#, HSYNC, and VSYNC are outputs. P[7:0] reflects the pixel data (or the high byte if the CL-GD546X is programmed for a pixel depth of more than one byte). BLANK# reflects the CRTC timing and can be an input to the external RAMDAC. HSYNC and VSYNC reflect the CRTC timing and can control the monitor. All the outputs can be latched to the positive edge of DCLK. EVIDEO#, ESYNC#, and EDCLK# all have internal pull-up resistors.

3.2.2 Feature Connector Input Mode

If EVIDEO#, ESYNC#, and EDCLK# are low, P[7:0], DCLK, and BLANK# are inputs. Data on P[7:0] is clocked into the palette; the palette outputs are clocked into the DACs for conversion to analog. BLANK# is used as the monitor blanking term. HSYNC and VSYNC are high impedance. All the inputs are clocked on the rising edge of DCLK. The RAMDAC is the only part of CL-GD546X used in this mode. The palette must be loaded with the appropriate values.

3.2.3 Feature Connector Dynamic Overlay Mode

External data can be overlaid on the CL-GD546X screen using EVIDEO# as a control. Best results are obtained if the pixel clock is supplied externally (EDCLK# is low). The CRTC timing must be generated by the CL-GD546X.

NOTE: The CL-GD546X does not support GENLOCK.

For each pixel that is to be replaced, EDCLK# must be driven low, and the pixel is driven onto P[7:0]. To avoid a bus crash, allow sufficient time for bus turn-around. This is intended to overlay 8-bpp palettized data onto 8-bpp palettized data.

4. REGISTERS

4.1 Introduction

The registers that control the V-Port and auto-BitBLT functions are described in Chapter 5, "Video Pipeline Registers", Chapter 6, "Enhanced V-Port™ Registers", and Chapter 8, "2D Graphics Accelerator Registers".

4.2 Video Pipeline Registers

The registers in this group control overlay, window 2, and the auto-BitBLT timing.

4.2.1 Overlay Control Register

The primary control for overlay is contained in the Overlay Control register (MMI/O B4h). This register contains the following fields:

Table C2-4. Overlay Control Register Bit Fields

Bits	Field	Feature Connector Mode	Memory Attach Mode	Capture Mode
7:6	Overlay control	01b or 10b	01b or 10b	00b or 11b
5:4	Color Key mode	As desired	As desired	(Not used)
0	Rambus ninth bit	(Not used)	(Not used)	If available

When external overlay is used (Feature Connector or Memory Attach mode), the Overlay Control field is programmed to '01b' or '10b' depending on if color keying is enabled. When internal overlay is used (Capture or Memory Attach mode), the Overlay Control field is programmed '00b' or '11b' depending on how occlusion occurs. If 8-bit RDRAMs are used, program this field to '00b'.

Color Key mode is used only for external overlay and controls the comparison. Use this field in conjunction with the Color Key and Color Key Mask registers.

Program the Video Port Mode bit to '1b' if the video is inserted directly into the display pipeline (Feature Connector and Memory Attach mode). Program the Video Port Mode bit to '0b' if the video is being captured (Capture or Memory Attach mode).

The Rambus ninth bit enable can be programmed to '1b' only if 9-bit RDRAMs are being used. If 9-bit RDRAMs are available, the ninth bit is a tag indicating whether a pixel is considered as being in a window. This allows all three windows to have arbitrary occlusion and use the video pixel format.

4.2.2 Color Key and Color Key Mask Registers

The Color Key registers can restrict the external overlay to selected pixels within window 2. These two registers specify the pixel value or range of pixel values used as the color key. These registers are used only when the Overlay Control field of the Overlay Control register is programmed to '10b'. The most-significant (or only) byte of each pixel is compared to the Color Key value, using the Mask value to restrict the bits that are actually compared. The compare is made according to the Color Key Mode. If the compare is true and the screen is in window 2, then external overlay is used.

Table C2-5. Color Key and Color Mask Register Definitions

Register	MMI/O Offset	Size in Bits	Field
Color Key	B8h	8	7:0
Color Key Mask	BCh	8	7:0

4.2.3 Graphics/Video Format Register

This register (MMI/O offset C0h) controls the format of graphics and video pixels. All pixels on the CL-GD546X screen are interpreted as graphics or video pixels according to the following rules.

- Pixels that are not in any of the three windows are interpreted as graphics pixels.
- Pixels in a window that have the format set to video in the respective STOP_BLT register, are interpreted as video pixels.
- The Graphics/Video Format register define these two formats.

There are two identical sets of fields as shown in Table C2-6.

Table C2-6. Pixel Formats

Graphics	Video	Field	Example
13:12	6:5	Pixel Depth	8-, 16-, 24-, 32-bpp
11:9	3:1	Format	LUT, RGB, YUV
8	0	Gamma Correction	(see bit description)

These fields control how the pixels are interpreted in the display pipeline. These particular fields have practically no effect on how data is handled by the 2D engine (although there are fields that contain similar information for the 2D engine).

There are restrictions about how the formats can be mixed. In particular, the two pixel depth fields must be the same, except for a single case. The graphics pixels can be programmed for 16-bpp YUV. While using an inexpensive 8-bpp frame buffer, this provides for high quality video overlay.

4.2.4 Window 2 Descriptor Registers (X,Y_START,END_2)

These four window 2 descriptor registers are programmed in terms of screen coordinates and specify when window 2 is displayed. Overlay control ‘00b’ with 9-bit RDRAMs allows video data anywhere on the display, regardless of the start of these registers.

The X_START_2 and X_END_2 registers are programmed with four-pixel granularity. The Y_START _2 and Y_END_2 registers are programmed in scanlines.

Table C2-7. Window 2 Definition

Register	MMI/O Offset	Size	Field	Defines
X_START_2	CCh	16	11:2	Left edge
Y_START_2	CEh	16	11:0	Top
X_END_2	D0h	16	11:2	Right edge
Y_END_2	D2h	16	11:0	Bottom

4.2.5 Auto-BitBLT Enable Registers (START,STOP_BLT_1,2,3)

These six Auto Enable registers specify when auto-BitBLT for the three windows is enabled. This is the safe-time shown in Figure C2-5 on page C2-9.

Table C2-8. Auto Enable Definitions

Register	MMI/O Offset	Size	Field
START_BLT_3	CAh	8	5:0
STOP_BLT_3	CBh	8	5:0
START_BLT_2	D4h	8	5:0
STOP_BLT_2	D5h	8	5:0
START_BLT_1	DEh	8	5:0
STOP_BLT_1	DFh	8	5:0

In addition, the three STOP_BLT registers contain an auto-BitBLT enable bit and the pixel format selection bit for each window.

4.3 Enhanced V-Port™ Registers

These registers define the V-Port and the video capture buffer(s).

4.3.1 V-Port™ Control

The enhanced V-Port has its primary controls in the V-Port Control register (MMI/O offset 10ch). Fields in this register specify the basic V-Port operating mode, the V-Port bus width, and timing parameters.

Table C2-9. V-Port™ Control Register Definitions

Bits	Field
11	Odd Field Detect (read only)
10	Buffer (read only)
9	External VSYNC Interrupt
8	Skip Frame
7:6	DCLK Mode
5	Readback External Interrupt (read only)
4	Bus Width
3	Double Buffer Enable
2:0	V-Port Mode

4.3.2 Capture Buffer Start

Four registers are required to specify the X and Y start of the two capture buffers. Typically, these registers are programmed once and never change. If double-buffering is not enabled, the Even register set is never used.

Table C2-10. Capture Buffer Start Definitions

Register	MMI/O Offset	Field	Granularity
X_START_2 (Odd)	100h	12:8	256 bytes
X_START_2 (Even)	102h	12:8	256 bytes
Y_START_2 (Odd)	104h	13:0	1 scanline
Y_START_2 (Even)	106h	13:0	1 scanline

4.3.3 Capture Buffer Size

Two registers are required to specify the width and height of the capture buffer. These registers are used only when the CL-GD546X is the video timing master (that is, for CL-GD5520 memory attach mode). Typically, these registers are programmed once and never changed.

Table C2-11. Capture Buffer Size Definitions

Register	MMI/O Offset	Size	Field
V-Port Width	108h	16	9:0
V-Port Height	10Ah	8	7:0

4.4 Graphics Accelerator Registers

Only the RSIZE{A–C}_opRDRAM registers are discussed in this appendix because they have controls unique to auto-BitBLTs. Programming of the remaining BitBLT registers is described in Chapter 8.

4.4.1 RSIZE{A–C}_opRDRAM

These three registers control the trigger and arming functions auto-BitBLT for the three windows. In addition, they contain the X,Y address of the auto-BitBLT headers. The auto-BitBLT headers contain the details of the respective auto-BitBLT. The auto-BitBLT headers are shown in Table C2-13.

Table C2-12 shows the offsets of these three registers and indicates the associated window.

Table C2-12. Window Selection

Register	MMI/O Offset	Associated Window
RESIZEA_opRDRAM	408h	1
RESIZEB_opRDRAM	40Ch	2
RESIZEC_opRDRAM	410h	3

4.5 Auto-BitBLT Header

The parameters for an auto-BitBLT are kept in RDRAM and loaded into the appropriate registers when necessary (that is, when the auto-BitBLT is about to begin). The parameter set is called an auto-BLT header. The pointer to the auto-BitBLT header is in the appropriate RSIZE{A–C}_opRDRAM register. The contents of the auto-BitBLT header are shown in Table C2-13.

Table C2-13. Auto-BitBLT Header Contents (CL-GD5464)

Word	Register
0	STRETCH_CNTL
1	SHRINKINC
2	DRAWDEF
3	BLTDEF
4	OP_opFGCOLOR[15:0]
5	OP_opFGCOLOR[31:16]
6	OP_opBGCOLOR[15:0]
7	OP_opBGCOLOR[31:16]
8	OP0_opRDRAM.pt.X
9	OP0_opRDRAM.pt.Y
A	MAJ_Y
B	MIN_Y
C	OP1_opRDRAM.pt.X
D	OP1_opRDRAM.pt.Y
E	ACCUM_Y
F	PATOFF
10	OP2_opRDRAM.pt.X

Word	Register
11	OP2_opRDRAM.pt.Y
12	MAJ_X
13	MIN_X
14	BLTEXT_EX.pt.X
15	BLTEXT_XEX.pt.Y
16	ACCUM_X
17	OP0_opSRAM
18	SRCX
19	OP2_opSRAM
1A	NEXT_HEAD.pt.X
1B	NEXT_HEAD.pt.Y
1C	CHROMA_LOWER[15:0]
1D	CHROMA_LOWER[31:16]
1E	CHROMA_UPPER[15:0]
1F	CHROMA_UPPER[31:16]
20	CHROMA_CNTL

