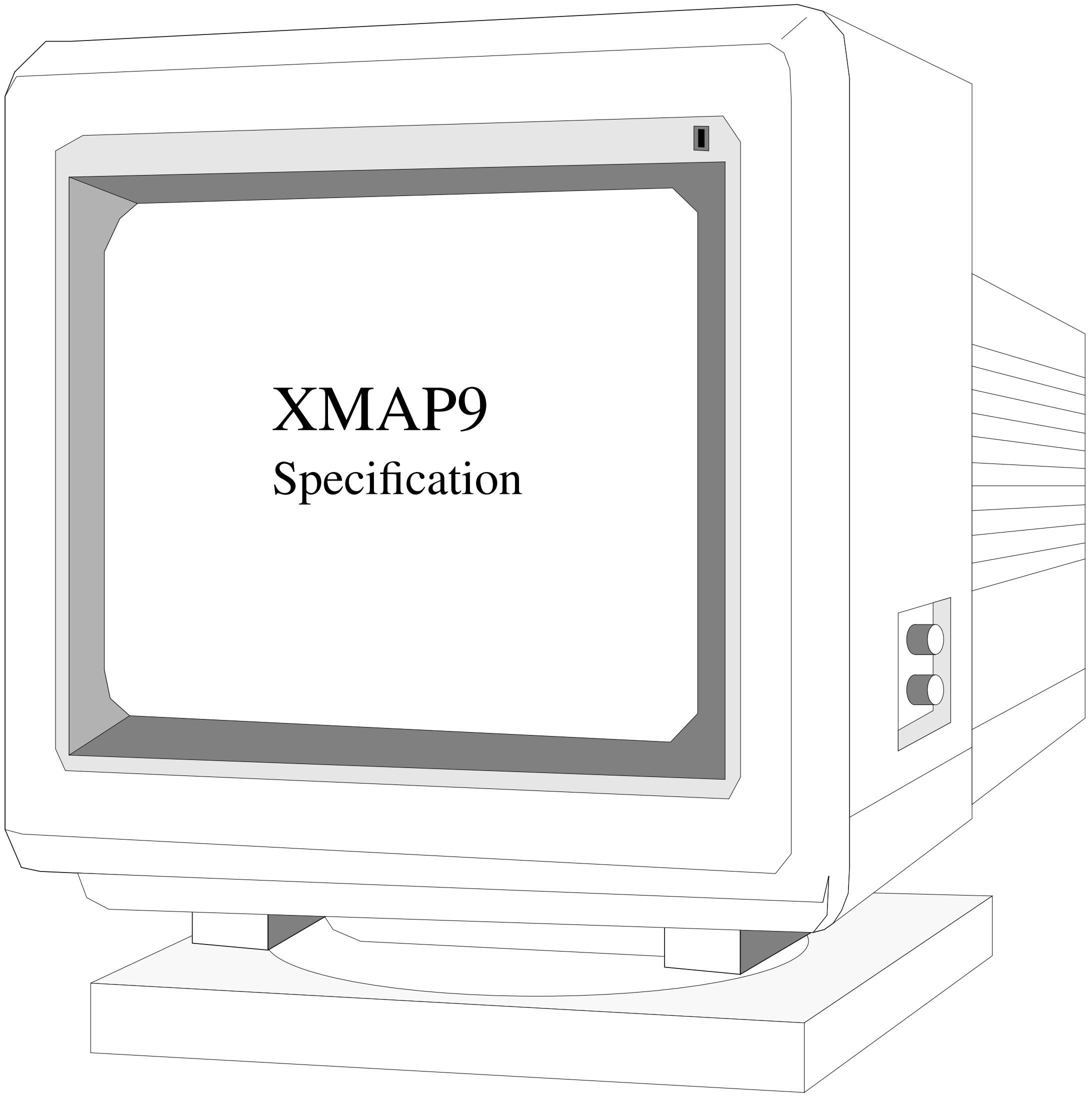


DSS, Silicon Graphics Inc.
Revision 2.1
Justin Chueh

campanile: /d/people/chueh/chips/xmap9/spec/spec

1.0 Introduction

1.1 Part Name and Number

Part Name: XMAP9
SGI Part Number: 099-8913-001
Vendor: LSI Logic Corporation
Vendor Part Number: L1A7726
Process: 1.0 micron CMOS gate array
Base Wafer: LCA100066P
Package:208 PQFP
Gate Count: 19K

1.2 General Description

This document describes the XMAP9, part of the backend of the Newport graphics subsystem. The primary function of the XMAP9 is to select the appropriate pixel information from either the frame buffer or external video data as a function of the display mode. The data will be reorganized into a 24-bit RGB or 13-bit CI address format and sent to the Color Map (CMAP) chip. The XMAP9 also provides the path for data to be sent to the Video Expansion Board from the output of the CMAP chip. Newport graphics can support both 24 and 8-bit plane systems.

In the Newport graphics subsystem, there are 2 XMAP9s each running at one-half the pixel clock rate. For a 1280 x 1024, 76 Hz screen, this works out to 70Mhz. The chip is implemented in LSI's LCA100K (1.0 micron CMOS) technology and has a pin count of 208.

The XMAP9 has 5 primary interface ports. It receives pixel and auxiliary information from the frame buffer via the Reorg (RO1) chips. Cursor and Display IDs as well as timing information come from the Video Controller (VC2) ASIC. The data is selected and reformatted for the CMAP chips. All of these interfaces run at the 70Mhz peak clock rate. The output of the CMAP is also fed back to the XMAP9 at 70Mhz. The interface with the Video Expansion Board is bi-directional and consequently slower, so it runs at 1/4th of the Pixel Clock or 35Mhz peak speed. Finally there is an asynchronous interface without acknowledgement with the REX3/Host via the 8-bit Display Command Bus.

1.3 Features

- 24, 12, 8 and 4-bit RGB as well as 12, 8 and 4-bit color index display modes. (Includes double buffer mode for 4 and 12-bit modes.)
- 8 bits of auxiliary (overlay/underlay) data per pixel.
- 8-bit system with 2-bits of aux.
- Up to 32 different Display IDs.
- Asynchronous host interface (without handshake)
- 70 MHz maximum operating frequency
- Dithering and interface to video expansion board
- 8 bits of pixel data can be replaced by video alpha. (Double buffered mode for RGBA too)
- Overlay Planes can also be double buffered. Overlay buffer control is completely independent of pixel buffer control. (e.g. the pixel plane can be swapped from buffer0 to buffer1 while overlay planes are constantly in buffer0) This mode is supported by the XMAP9, but not by other Newport Graphics chips or software.

- Pop-Up planes. In the 24-bit system there are 2 extra framebuffer bits per pixel which can be used as Pop-Up planes. These planes have a constant color map throughout the entire screen. Current software implementations actually configure the Overlay Planes in this manner, so pop-up planes are unused.

1.4 Block Diagrams

FIGURE 1. Newport System Block Diagram

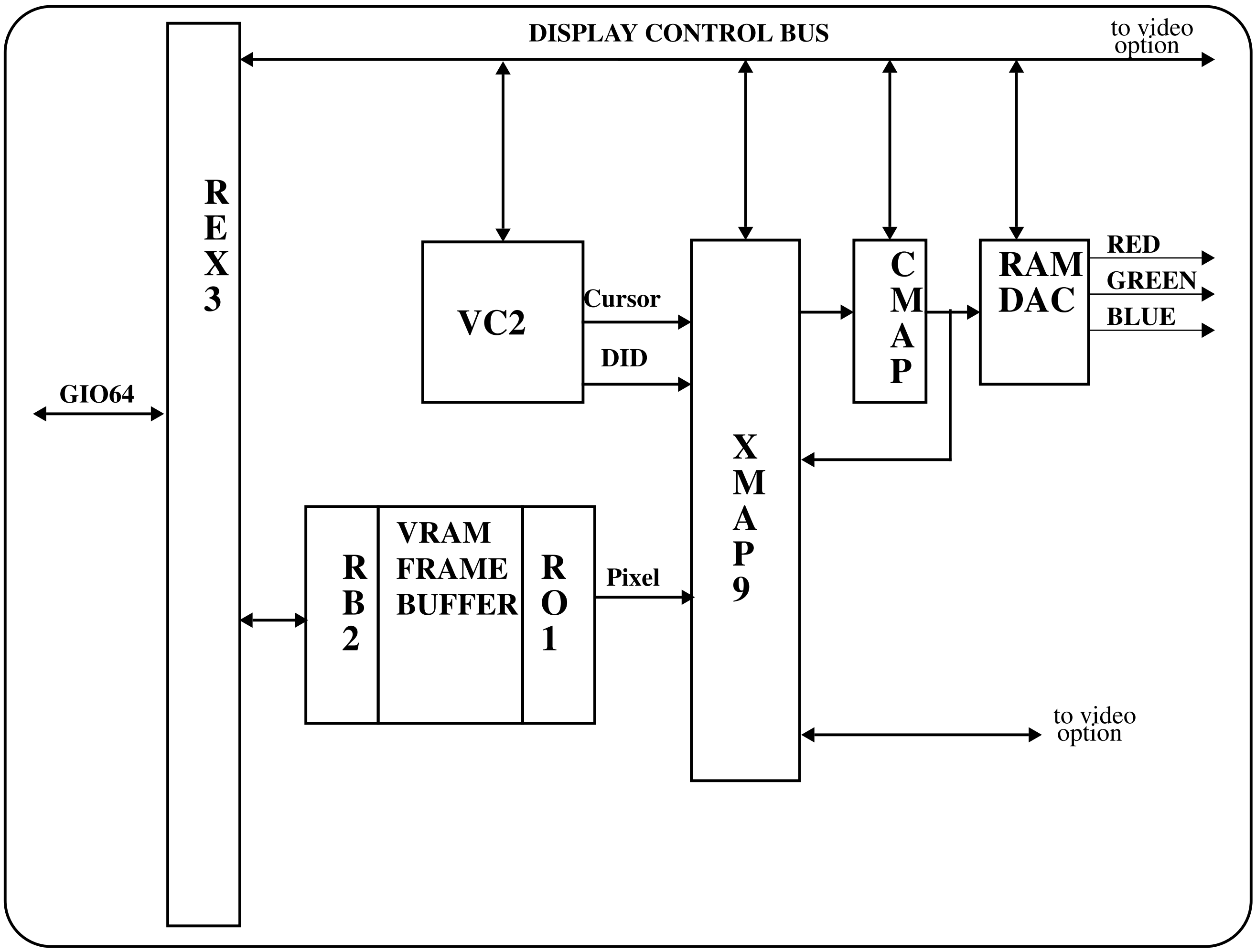
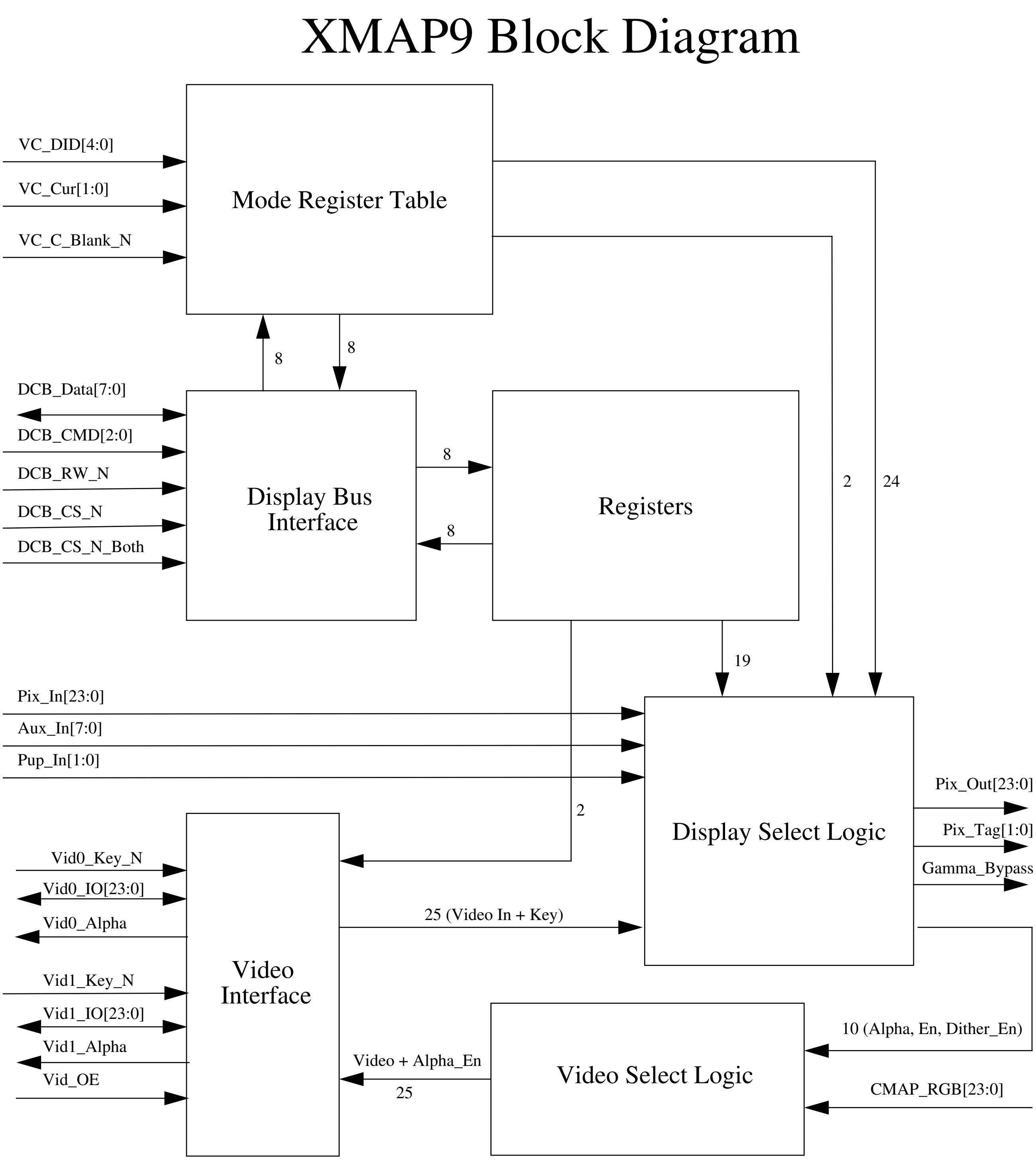


FIGURE 2. XMAP9 Architecture



1.5 Related Documents

For additional information and background, please refer to the following related documents:

Newport Graphics Subsystem Specification, Farhad Fouladi

CMOS Color Map Specification, Ben Hsu, Vitelic Corp.

ReOrganizer 1 (RO1) Specification, Farhad Fouladi

Video Controller 2 (VC2) Specification, Leslie Neft

Display Control Bus Specification, E. Linstadt

Average Buffer 1 (AB1), (Express Video) Specification, Bob Williams, et. al.

Boundary Scan & PLL User's Guide, Anan Nagarajan

2.0 Device Interface

2.1 VHDL Description

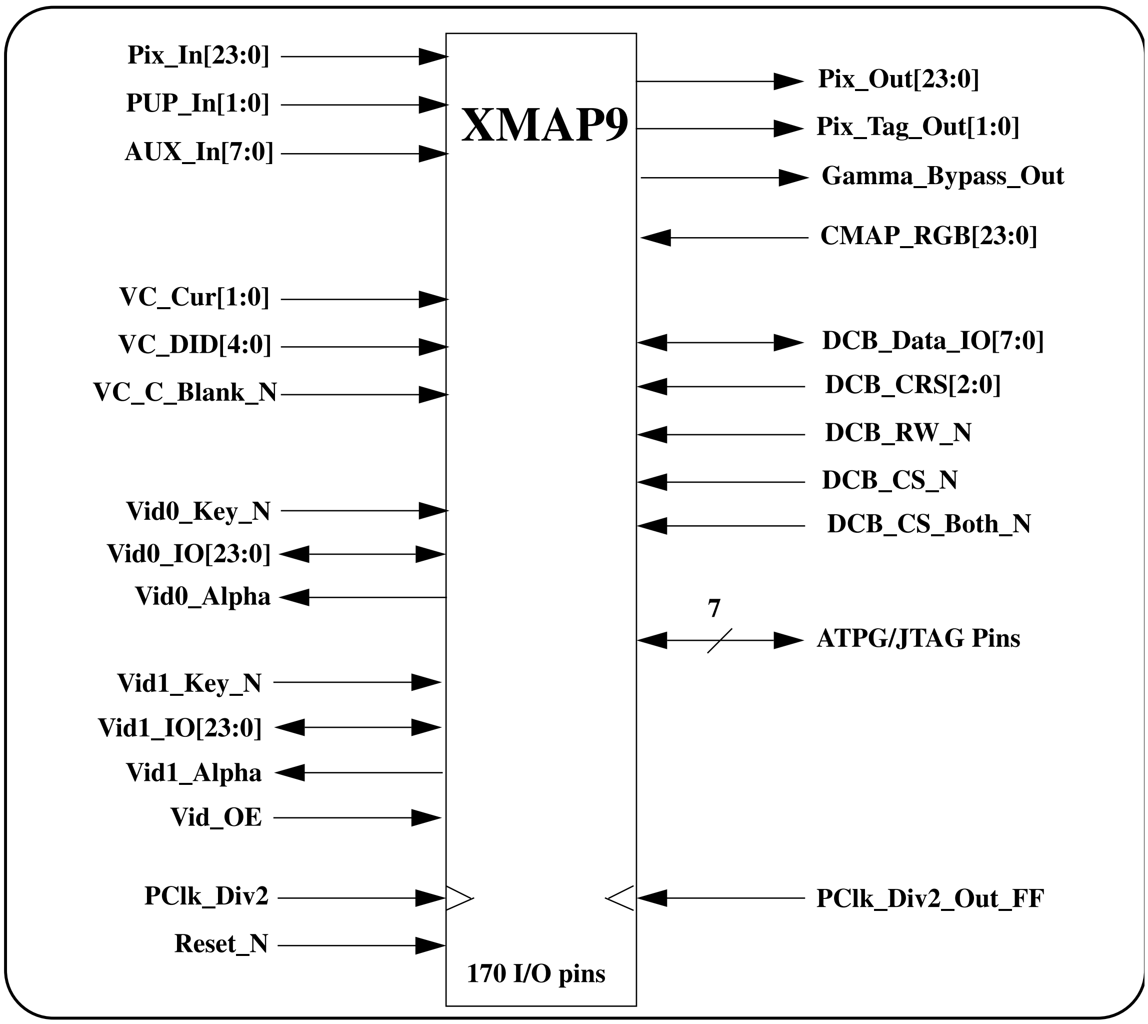
This section describes the device level interface to the XMAP9 as a VHDL entity.

```
entity XMAP9 is
  port(
  --FrameBuffer/Reorg Interface
    P_Pix_In : in mvl7w_vector (23 downto 0);
    P_Pup_In : in mvl7w_vector (1 downto 0);
    P_Aux_In : in mvl7w_vector (7 downto 0);

  --CMAP Interface
    P_Pix_Out : out mvl7w_vector (23 downto 0);
    P_Pix_Tag_Out: out mvl7w_vector (1 downto 0);
    P_Gamma_Bypass_Out : out mvl7w;
    P_CMAP_RGB : in mvl7w_vector (23 downto 0);

  --VC2 Interface
    P_VC_DID : in mvl7w_vector (4 downto 0);
    P_VC_Cur : in mvl7w_vector (1 downto 0);
    P_VC_C_Blank_N : in mvl7w;

  --Video Interface
    P_Vid0_IO : inout mvl7w_vector (23 downto 0);
    P_Vid0_Key_N : in mvl7w;
    P_Vid0_Alpha : out mvl7w;
    P_Vid1_IO : inout mvl7w_vector (23 downto 0);
    P_Vid1_Key_N : in mvl7w;
    P_Vid1_Alpha : out mvl7w;
    P_Vid_OE : in mvl7w;

--Display Bus Interface
    P_DCB_Data_IO : inout mvl7w_vector (7 downto 0);
    P_DCB_CRS:  in mvl7w_vector (2 downto 0);
    P_DCB_RW_N : in mvl7w;
    P_DCB_CS_N : in mvl7w;
    P_DCB_CS_Both_N : in mvl7w;

--Clocks and Reset
    P_PClk_Div2 : in mvl7w;
    P_PClk_Div2_Out_FF : in mvl7w;
    P_Reset_N : in mvl7w;

--Test Pins
    P_ENTEI : in mvl7w;           --External tri-state control
    P_JTDI : in mvl7w;
    P_JTMS : in mvl7w;
    P_JTCK : in mvl7w;
    P_TP0 : in mvl7w;
    P_TP1 : in mvl7w;
    P_JTDO : out mvl7w
```

```
    );
end XMAP9
```

2.2 Pin Diagram

FIGURE 3. XMAP9 Pin Diagram



2.3 Pin List

The following tables list each XMAP9 pin, its assertion level, direction (I, O, I/O), switching levels (TTL or CMOS) and provide a brief functional description.

Table 2a: FrameBuffer/Reorg Interface

Pin Name	Level	Type	Description
Pix_In[23:0]	H	IBUFD	Pixel input from RO1 chip. In 8-bit system, bits 23:8 are ignored.
PUP_In[1:0]	H	IBUFD	In the 8-bit system these bits are the 2 bits of Aux Data. In the 24-bit system these bits become 2 bits of global Pop Up.

Table 2a: FrameBuffer/Reorg Interface

Pin Name	Level	Type	Description
Aux_In[7:0]	H	IBUFD	Aux input for 24-bit system.

Table 2b: CMAP Interface

Pin Name	Level	Type	Description
Pix_Out[23:0]	H	B4	Pixel output to CMAP either 24-bit RGB or 13 bit CI
Pix_Tag_Out[1:0]	H	B4	Pix_Out mode. (refer to CMAP spec.) 00 = Color Index Mode 01 = RGB Mode, 11101 + 8-bit address 10 = RGB Mode, 11110 + 8-bit address 11 = RGB Mode, 11111 + 8-bit address
Gamma_Bypass_Out	H	B4	If high, this pin will disable gamma correction in the DAC for this pixel.
CMAP_RGB[23:0]	H	TLCHTD	RGB output of the CMAP (Fed back through XMAP to Video)

Table 2c: VC2 Interface

Pin Name	Level	Type	Description
VC_DID[4:0]	H	IBUFD	Display IDs from VC2
VC_Cur[1:0]	H	IBUFD	2 bit cursor from VC2
VC_C_Blank_N	L	IBUFD	Composite Blanking

Table 2d: Video Interface

Pin Name	Level	Type	Description
Vid0_IO[23:0]	H	BD4CRU	channel 0 data for video board
Vid0_Key_N	L	IBUFNU	Video pixel key for channel 0 0 = Video Data Enabled 1 = Video Data Disabled
Vid0_Alpha	H	BT4R	Video alpha mode enable for channel 0 Only for Express Mode 0 = Alpha Mode Disabled 1 = Alpha Mode Enabled
Vid1_IO[23:0]	H	BD4CRU	channel 1 data for video board
Vid1_Key_N	L	IBUFNU	Video pixel key for channel 1
Vid1_Alpha	H	BT4R	Video alpha mode enable for channel 1

Table 2d: Video Interface

Pin Name	Level	Type	Description
Vid_OE	H	IBUFNU	Video data bus direction control. Only for Express Mode. 0 = Video to Graphics 1 = Graphics to Video

Table 2e: Display Bus Interface

Pin Name	Level	Type	Function
DCB_Data_IO[7:0]	H	BD8TRPU	Data read from (DCB_RW_N = 1) or written (DCB_WR_N = 0) to the Display Control Bus slave devices.
DCB_CRS[2:0]	H	TLCHTU	Display Control Bus command.
DCB_RW_N	L	TLCHTU	Read/Write direction signal.
DCB_CS_N	L	TLCHTU	Display Control Bus command strobe, indicating that valid DCB_Addr, DCB_CRS, DCB_RW_N and, for write transfers, DCB_Data are on the bus
DCB_CS_Both_N	L	TLCHTU	Display Control Bus command strobe, this is a single line hooked up to both XMAP9's in the Newport system.

Table 2f: Clocks and Reset

Pin Name	Level	Type	Function
PClk_Div2	H	TLCHTM 2 x ClkC16	Pixel clock divided by 2, for most of chip. The mux in the pad is used to mux the JTAG clock.
PClk_Div2_Out_FF	H	DDRV + DRVTI8MI	Pixel clock divided by 2. Dedicated to the output FF's to reduce clock skew. The DRVTI8MI converts to CMOS, muxes and drives.
Reset_N	L	TLCHTU	Reset FSMs

Table 2g: Test Pins

Pin Name	Level	Type	Function
ENTEI	H	TLCHN	I/O pin tristate enable
JTDI	H	TLCHTU	JTAG Data In
JTMS	H	TLCHTU	JTAG Mode Select
JTCK	H	TLCHTU	JTAG Clock
TP0	H	TLCHTU	JTAG Control Pin0
TP1	H	TLCHTU	JTAG Control Pin1

Table 2g: Test Pins

Pin Name	Level	Type	Function
JTDO	H	B4	JTAG Data Out

3.0 Programmer Interface

3.1 Registers

3.1.1 Register Summary

The table below shows all the registers in the XMAP9 which are accessible to the host. All registers will be readable by the host for diagnostic purposes.

Table 3a: Register Summary

CRS	Index (Hex)	Type	Name	Bits	Description
0	--	R/W	Configuration Register	8	Bit 7, 0 = Video Option Off, 1= Video Option On Bit 6, 0 = Starter video mode, 1 = Express video mode Bit (5:4), RGB Map for Video Bit 3, Fast/Slow DCB Interface 0=Fast PClk, 1 = Slow Bit 2, 0 = 24-bit System, 1 = 8-bit system Bit 1, 0 = even pixel, 1 = odd pixel path XMAP Bit 0, 0 = PUP Mode Off, 1= PUP Mode on
1	--	R	Revision Register	3	Revision of the XMAP9. Starts at 0.
2	--	R	Fifo Entries Available	3	Number of entries unused in the mode register FIFO
3	--	R/W	Cursor CMAP MSB	8	8 most significant bits of CMAP address for cursor data
4	--	R/W	Pop Up CMAP MSB	8	8 most significant bits of CMAP address for pop up data
5	--	W	Mode Register Table	32	8 bits address plus 24 bits Mode Register entry
5	00-7F	R	Mode Register Table	8	Mode Register entries 0-31, byte 0 - 2
6	--	--	Unused	--	
7	--	R/W	Mode Table Address Register	8	Index register for Mode Register Entries

Note: Writing to the mode register table delivers 32-bits in one atomic operation.
Reading from the mode register table is only used for diagnostics, so only one byte at a time is transferred from the XMAP9 per instruction.

3.1.2 Miscellaneous Control Registers

Reading from the registers is easy in most cases, because the XMAP9 does not modify most of the registers. Thus, the host can treat them as memory locatioins which only the host can change. The reads on the DCB bus are done asynchronously with respect to the rest of the chip (which uses the pixel clock), but since none of these registers are modified by the XMAP9, there are no race conditions. The only exception is the Fifo Entries Available Register which is set by the XMAP9. In order to avoid synchronization errors, this register is grey coded.

Writing to the registers takes an equally simple approach. All the miscellaneous registers are writeable at any time. This means that if the host changes the location of the CMAP Address MSB bits, the change may occur in the middle of a vertical display frame. The only guarantee is that the change will be synchronous and no transitory locations will be selected. Finally, there is no way to change the registers at a particular moment on the screen. Thus, it would be impossible for the host to use one RGB Map for one video window, and change the Video System Register thus using another RGB Map for another video window.

3.1.2.1 Configuration Register

This register contains information that must be set by the system at power-on, but which will not be changed very often during normal operations.

Bit 7: Video Option Enable

This bit enables the video option board. If the bit is set to 0, then XMAP9 will tristate the video option bus. This will help with EMI effects if a video option board is not present. If a board is actually installed this bit should be set to 1.

Bit 6: Video System Mode

This bit determines what mode the video option board is running under. Starter Mode designates 12 bits of data as going from the XMAP9 to the video board and the other 12 bits as video board to graphics. In Express Mode, the video board switches the direction of the 24-bit bus back and forth with the Vid_OE signal.

Bits (5:4): Video RGB Map:

These bits determine which RGB map in the CMAP to use when displaying video. Since all data in Newport Graphics goes through the CMAP chips, video may have a different map than normal graphics RGB. Please refer to CMAP Interface in Section 4.2, or the CMAP specification for more details.

00 = Color Index
01 = RGB Map0
10 = RGB Map1
11 = RGB Map2

Bit 3: Fast/Slow DCB Inteface Control

This bit determines the minimum and maximum time parameters for Writes to the XMAP9 via the DCB Bus. In the following discussion all references to pixel clock periods, are really to the XMAP9 pixel clock div 2 clock, so that 3 clock periods is really 6 pixel (dot) clocks. Also there is some propogation delay and FF setup time, so some extra margin needs to be added to some of the numbers. For example, say the minimum time between 2 writes is 3 clock periods at 14 ns each or 52 ns. A time of 52 ns will not work, but 60 ns will. Chip delays indicate 5 ns need to be added to time between CS_Edges.

If this bit is set to 0 for Fast PClk mode, then the maximum CS low width (active state) for writes is less than 3 times the pixel clock period. Otherwise the same data could be treated as multiple reads. In this mode, the minimum time between falling edges of the CS is 4 pixel clock periods. This mode can be used all the time, but in really slow pixel clock modes like NTSC or PAL, the bit can be set to 1 to increase efficiency by decreasing the time the XMAP9 takes the DCB Bus. This is the default mode on reset of the chip.

If this bit is set to 0 for Slow PClk mode, then the maximum CS low width (active state) for writes is less than 1 pixel clock period. The time between CS Falling Edges is decreased to 2 pixel clock periods. It is important to note, that if the chip is in this Slow PClk DCB Mode, but is actually running on a 70 Mhz clock or something nearly as fast, that the 33 Mhz DCB Bus cannot satisfy the maximum clock period of 1 pixel clock period. A number of things can be done to restore the chip to the proper mode.

1. Write to the configuration register changing the DCB speed. Multiple writes will probably occur, but as long as the same data is written, it doesn’t matter.
2. Slow the graphics backend down, by switching to a differnt monitor type like NTSC. Write to the XMAP9 configuration register back into fast mode. Switch back to a fast pclk monitor.
3. Reset the chip.

Table 3b: Summary DCB Interface Speeds

Bit 3	Name	Maximum CS Low Width	Minimum time between CS Falling Edges
0	Fast Pixel Clock Mode	3 x Clock Period	4 x Clock Period + 5 ns
1	Slow Pixel Clock Mode	1 x Clock Period	2 x Clock Period + 5 ns

Bit 2: System 24/8

This bit indicates whether the System is in 24-bit or 8-bit mode.

Bit 1: Even/Odd Pixel

This bit tells each XMAP9 in the system whether it is in the even or odd pixel path. The even-odd information is used for spatial dithering.

Bit 0: Pup Enable:

This bit is used to turn off the Pop Up planes. In the 24-bit system, if Pup is turned off, the XMAP9 will ignore these inputs. In the 8-bit system, the Pup bits become Aux bits when Pup is disabled.

This register is set to all zeros during reset. The XMAP9 does not modify any bits in this register after reset.

3.1.2.2 Revision Register

The revision register is used by software to keep track of the spin of the chip. This is a read only register hardwired to a certain value. The revision numbers start at 0.

3.1.2.3 Fifo Entries Available Register

This register keeps a count of the Fifo entries which are still empty for the Mode Register Fifo. The count starts at 3 (Fifo Empty) and decrements as each new 32-bit (8 bits address + 24 bits of data) fifo entry is received from the host via the DCB bus. This register will change relative to the pixel frequency, and consequently, reading from it using the GIO clock on the DCB bus, may cause the register to be read in transition. In order to make sure the data is always valid, this register will use a grey code scheme, such that it can be off by at most one entry. This will be taken care of by having one more entry in the FIFO than the register says there is. Thus if the hosts does a read and sees that there are 3 entries available, there will be 3 or 4 entries available and sending 3 more mode register entries will not overflow the fifo. The grey code scheme is as follows:

000 = No entries available, FIFO full
001 = 1 entry available
011 = 2 entries available
010 = 3 entries available

3.1.2.4 Cursor CMAP MSB Register

The Cursor has only two bits of color data. 13 bits are needed to index into the CMAP. The 8 bits in this register make up the most significant bits of that address, while the rest are zeroed. Please refer to the Cursor Address Format in Table 3k. This register is never modified by the XMAP9.

3.1.2.5 Pop Up CMAP MSB Register

The Pop Up has only two bits of color data. 13 bits are needed to index into the CMAP. The 8 bits in this register make up the most significant bits of that address, while the rest are zeroed. Please refer to the Pop Up Address Format in Table 3l. This register is never modified by the XMAP9.

3.1.2.6 Mode Table Address Register

Since there are not enough CRS modes for all the different registers, some modes must use one level of indirection via the index register. In order to read or write from these registers, first write to the index register, then selecting the particular CRS write over the display bus. This register is never modified by the XMAP9.

3.1.3 Mode Register Table

The mode register controls the XMAP9 display selection. The table has 32 entries each with a 24-bit control word. The control word describes how the pixel, aux. and video data is to be interpreted. Many of the modes of XMAP5 have been eliminated which explains why the control word is only 24-bits instead of 32. A single mode register entry describes the pixel format for windows using that particular display mode ID.

Writes to the Mode Register are fifoed so that the register is only updated during blanking. The host can write to the XMAP9 at any time as long as the fifo is not full. Instead of having a fifo full flag, the XMAP9 provides a FIFO ENTRIES Available register which can be checked before every write to be sure the fifo will not overflow. A more effecient method is to calculate the rate at which the fifo is emptied (relevant data are time between blanking and pixel clock frequency) and to not exceed that rate with writes.

Reads from the Mode Register are done 1 byte at a time. First the address is written into the Mode Table Address Register and then a single byte is read out. Both reads and writes are described in more detail in Section 3.2: Programming Via the DCB Bus.

Table 3c: Mode Register Bit Definition

Bit Field	Mnemonic	Function
Mode[23:19]	Aux_MSB_CMAP[4:0]	Used for aux modes. These modes are always color index.
Mode[18:16]	Aux_Pix_Mode[2:0]	Determines Pixel Mode for Overlay and Underlay. (8-bits Aux in 24-bit system while in 8-bit system 2-bits Aux) 000 = Overlay/Underlay off 001 = Full Underlay (8-bits in 24-bit system. 2-bits in 8-bit system) 010 = Full Overlay (8-bits in 24-bit system. 2-bits in 8-bit system) 110 = Overlay, Double Buffered (4-bit for each buffer or 1,1) 111 = Both Overlay and Underlay (4-bit Overlay, 4-bit Underlay or 1,1)
Mode[15]	Alpha_En	Enables Alpha Modes for Express Video Option. If Alpha_En = 1, then when in 12-Bit RGB or 8-bit DB RGB, there are 8 and 4 bits of alpha respectively.
Mode[14]	Video_Dither_Bypass	Bypass dithering of video data. (Active High)
Mode[13:12]	Video_Mode[1:0]	Video Mode Select 00 = off 01 = video overlay, 10 = video underlay 11 = replace pixel.
Mode[11:10]	Pix_Size[1:0]	Determines Pixel Size for RGB and CI modes. 00 = 4 bits/pixel 01 = 8 bits/pixel 10 = 12 bits/pixel 11 = 24 bits/pixel (Not allowed for CI)

Table 3c: Mode Register Bit Definition

Bit Field	Mnemonic	Function
Mode[9:8]	Pix_Mode	Select CI or one of three RGB Maps. (Refer to section 4.2 or CMAP spec for more details on RGB Maps) 00 = CI 01 = RGB Map0 10 = RGB Map1 11 = RGB Map2
Mode[7:3]	MSB_CMAP[4:0]	Used for color index mode because CMAP needs a 13 bit address. (e.g. 8-bit CI uses 5 bits of this field while 12-bit CI uses only 1 bit for msb)
Mode[2]	Gamma_Bypass	Bypass gamma correction in DAC (Active High)
Mode[1]	OVL_Buf_Sel	Buffer Select for Overlay double buffered modes. (0 = Buf0, 1 = Buf1)
Mode[0]	Buf_Sel	Buffer Select for RGB and CI double buffered modes. 4 and 12-bit RGB and CI double buffered modes are supported. Also 8-bit RGB with 4-bits alpha for video DB is supported. (0 = Buf0, 1 = Buf1)

3.2 Programming via the Display Control Bus

Most DCB transfers to XMAP9 will take 2 GIO operations. First the DCB_MODE register in the REX3 must be set up. (DCB_MODE parameters will be supplied at a later date.) Next a read or write to DCB_Data.

If the transfer is a read from one of the registers which requires an index mode (e.g. Mode Register5, Byte 1), 4 GIO operations are required. First a write to the DCB_MODE register to set up a write to the Index Register. Next a write to the DCB_Data bus which will go to the Index Register. Another write to the DCB_MODE register to set up the read from the mode register, and finally a read from the DCB_Data bus.

The Address Register should have the following format when doing reads from the mode register.

Bit7: Read Port of the RAM. (0 or 1) (Used in chip testing. Either port will give correct results)	Bits 6:2, Mode Register Entry (0 - 32).	Bits 1:0, Byte 0, 1 or 2, of entry. Because the mode register is Big Endian, Byte 0 = Bits23:16. Translation of this field is: 00 Byte 2: Bits 7:0. 01 Byte 1: Bits 15:8. 10 Byte 0: Bits 23:16. 11 Reserved.

A special case is made for writing to the Mode Register Entries because this data goes to a FIFO. After a write sets up the DCB_MODE register, 4 bytes are transferred over the DCB bus in one atomic write. The first byte is the address (5 bits) of the mode register, followed by bytes 0, 1 and 2 of that particular mode register entry. (Note Byte0 is bits 23-16 or Big Endian Notation) Since reads from the Mode Register are not performance critical, they are done one byte at a time.

One final note on Mode Register Reads. If at initialization, the XMAP9 is constantly in Blanked Mode, there is still a 10 cycle latency between the falling edge of CS, to the time the data is actually written into the Mode Register RAM. These are pixel_clock/2 cycles, so if the chip were running in NTSC mode it would take 1.6 microseconds before the data was written into the RAM. Thus a write followed by an immediate read, in these cases, may yield incorrect results. The solution is simply to wait before doing a Read on the Mode Register entry. Writing a different entry or doing some DCB trasfers with another chip, will give the XMAP9 enough time to propogate the data into the RAM.

3.3 Pixel Plane Selection

The selection of the cursor, pop up, overlay, video, pixel or underlay data for display is a function of the data as well as the mode of the particular pixel. Cursor and pop up data is global, and does not have a particular window DID. Planes may not exist depending upon configuration. (E.G. if there is no video option board, then video data cannot be selected) The selection algorithm is summarized below:

```
if Cursor_Data Nonzero then
    select Cursor_Data
else
    if PUP_Data Nonzero then
        select PUP_Data
    else
        if (Overlay_Enabled and Overlay_Data Nonzero) then
            select Overlay_Data
        else
            if (Video_Overlay_Enabled and Video_Key enabled) then
                select Video_Data
            else
                if (Video_Replace_Enabled and Video_Key enabled) then
                    select Video_Data
                else
                    if (not Video_Replace_Enabled and Pixel_Data Nonzero) then
                        select Pixel_Data
                    else
                        if (Video_Underlay_Enabled and Video_Key enabled) then
                            select Video_Data
                        else
                            if Underlay_Enabled then
                                select Underlay_Data
                            else
                                if Video_Replace_Enabled
                                    select Video_Data
                                else
                                    select Pixel_Data
```

Note: In the last Video_Replace Mode, we select video data irregardless of the Key. This is a last resort display in video replace mode. This is consistent with XMAP5/7.

3.4 Display Modes

3.4.1 Frame Buffer Pixel Formats

These are the pixel formats coming out of the frame buffer and into the XMAP9. For the double buffered modes, the right buffer (in this table) is buffer0 and the left buffer is buffer1. For example in 12-Bit DB mode, P[11:0] is buffer0 and P[23:12] is buffer1.

When switching from single buffered to double buffered mode, one should first draw into buffer0 to ensure a smooth transition. This is because the msb's of the single buffered mode match with those of buffer0 not buffer1. Buffer0 is going to be the lower ordered bits. Please refer to Table 3d for descriptions of Buffer0 and Buffer1 in different modes.

Table 3d: Frame Buffer Pixel Configurations

BIT PLANES	PIXEL TYPE	P23	P22	P21	P20	P19	P18	P17	P16	P15	P14	P13	P12	P11	P10	P9	P8	P7	P6	P5	P4	P3	P2	P1	P0
24	RGB-SB 24BIT	B0	R0	G0	B1	R1	G1	B2	R2	G2	B3	R3	G3	B4	R4	G4	B5	R5	G5	B6	R6	G6	B7	R7	G7
24	RGB-DB 444+444	B4	R4	G4	B5	R5	G5	B6	R6	G6	B7	R7	G7	B4	R4	G4	B5	R5	G5	B6	R6	G6	B7	R7	G7
24	CI-SB 12BIT	-	-	-	-	-	-	-	-	-	-	-	-	I11	I10	I9	I8	I7	I6	I5	I4	I3	I2	I1	I0
24	CI-DB 12+12	I11	I10	I9	I8	I7	I6	I5	I4	I3	I2	I1	I0	I11	I10	I9	I8	I7	I6	I5	I4	I3	I2	I1	I0
8/24	RGB-SB 8BIT 332	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	R5	G5	B6	R6	G6	B7	R7	G7
8/24	RGB-DB 8BIT 121+121	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	G6	B7	R7	G7	G6	B7	R7	G7
8/24	CI-SB 8BIT	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	I7	I6	I5	I4	I3	I2	I1	I0
8/24	CI-DB 4+4	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	I3	I2	I1	I0	I3	I2	I1	I0
24	RGBA - DB 3324 + 3324	A4	A5	A6	A7	R5	G5	B6	R6	G6	B7	R7	G7	A4	A5	A6	A7	R5	G5	B6	R6	G6	B7	R7	G7
24	RGBA - SB 4448	-	-	-	-	A0	A1	A2	A3	A4	A5	A6	A7	B4	R4	G4	B5	R5	G5	B6	R6	G6	B7	R7	G7

NOTES: R - Red, G - Green, B - Blue, I - Color index, A- Alpha

Table 3e: Frame Buffer Aux Configurations

BIT PLANES	PIXEL TYPE	Pup 1	Pup 0	Aux 7	Aux 6	Aux 5	Aux 4	Aux 3	Aux 2	Aux 1	Aux 0
24	2BIT PUP 8BIT AUX	Pup 1	Pup 0	Aux 7	Aux 6	Aux 5	Aux 4	Aux 3	Aux 2	Aux 1	Aux 0

Table 3e: Frame Buffer Aux Configurations

BIT PLANES	PIXEL TYPE	Pup 1	Pup 0	Aux 7	Aux 6	Aux 5	Aux 4	Aux 3	Aux 2	Aux 1	Aux 0
24	2BIT PUP 4+4 AUX	Pup 1	Pup 0	Aux 3	Aux 2	Aux 1	Aux 0	Aux 3	Aux 2	Aux 1	Aux 0
8	2BIT PUP	Pup 1	Pup 0	-	-	-	-	-	-	-	-

3.4.2 Pixel (RGB/CI) Display Modes

The pixel data can be interpreted in a number of ways. 5 bits of a mode register entry determine how the XMAP9 should interpret the 24-bit input pixel data. The following table shows how to set the bits to select each mode. “x” indicates don’t cares where the signal is ignored.

Pix_Tag is used to select RGB or CI.

00 indicates CI mode and any other combination indicates one of the 3 RGB modes.

Pix_Size is 4, 8, 12 or 24.

Buf_Sel, selects buffer 0 or 1, for double buffered modes.

Alpha_En, enables alpha for Video Option Board. If Alpha_En = 1, then the alpha data is mixed with the rgb pixel data returning from the CMAP and the whole package is sent to the video option board.

Table 3f: Pixel (RGB/CI) Display Modes

Pix_Tag[1:0]	Pix_Size[1:0]	Buf_Sel	Alpha_En	Description
00	00	0	0	4-Bit CI, Buf 0
00	00	1	0	4-Bit CI, Buf 1
00	01	0	0	8-Bit CI
00	10	0	0	12-Bit CI, Buf 0
00	10	1	0	12-Bit CI, Buf 1
01,10 or 11	00	0	0	4-Bit RGB, Buf 0
01,10 or 11	00	1	0	4-Bit RGB, Buf 1
01,10 or 11	01	0	0	8-Bit RGB
01,10 or 11	10	0	0	12-Bit RGB, Buf 0
01,10 or 11	10	1	0	12-Bit RGB, Buf 1
01,10 or 11	11	0	0	24-Bit RGB
01,10 or 11	01	0	1	8-Bit RGB, 4-bit Alpha, Buf 0
01,10 or 11	01	1	1	8-Bit RGB, 4-bit Alpha, Buf 1
01,10 or 11	10	0	1	12-Bit RGB, 8-bit Alpha

If the XMAP9 is in Color Index Mode, the following table shows the actual format of the address which is sent to the CMAP. Newport supports an 8K color map, so a 13-bit address is needed to index into the map. The upper 11 bits (Pix_Out[23:13]) are ignored by the CMAP chip in CI mode. For example, in 8-bit CI mode, the 5 most significant bits come from the mode register (MSB_CMAP[4:0]) and the other 8 bits come from the pixel data. One can think of the mode register bits as a location for a 256 entry color map. The pixel data is the index into this color map. In the

4-bit case, 256 entries are allocated, but only the first 16 are used. Multi-map mode which would allow for 16 entry maps is not supported.

Table 3g: Output (To CMAP) Address Formats, Color Index Modes

Mode	CI Pix_Out[12]	CI Pix_Out[11:8]	CI Pix_Out[7:4]	CI Pix_Out[3:0]
4-bit, Buf 0	MSB_CMAP[4]	MSB_CMAP[3:0]	0000	Pix_In[3:0]
4-bit, Buf 1	MSB_CMAP[4]	MSB_CMAP[3:0]	0000	Pix_In[7:4]
8-bit	MSB_CMAP[4]	MSB_CMAP[3:0]	Pix_In[7:4]	Pix_In[3:0]
12-bit, Buf 0	MSB_CMAP[4]	Pix_In[11:8]	Pix_In[7:4]	Pix_In[3:0]
12-bit, Buf 1	MSB_CMAP[4]	Pix_In[23:20]	Pix_In[19:16]	Pix_In[15:12]

If the XMAP9 is in RGB mode, then all 24 bits of pixel data are used by the CMAP. Please refer to section 4.2 or the CMAP spec. to see how this data will be mapped. All XMAP9 outputs go through the CMAP chip, so there is no bypass mode. The CMAP must be properly loaded to get the correct RGB output from the CMAP.

The pixel data coming from the frame buffer is interleaved. The pixel data must be reorganized and also duplicated in most cases to get a full 24-bit RGB pixel value. Take the 8-bit, Buf 0 RGB case. Here we have 3 bits of Green, 3 bits of Red and 2 bits of Blue color input data. The msb of the Blue is bit 2 and the lsb is bit 5. These 2 bits are duplicated, 4 times to get the required 8 bits of color for the CMAP and DAC. If the bits were not duplicated and zeros were inserted instead, a downward shift would be made in the color spectrum and it would be impossible to display a true blue (all 1’s in the blue component).

Table 3h: Output (To CMAP) Pixel Formats, RGB Modes

Mode	Blue Pix_Out[23:16]	Green Pix_Out[15:8]	Red Pix_Out[7:0]
4-bit, Buf 0	Pix_In[2,2,2,2,2,2,2,2]	Pix_In[0,3,0,3,0,3,0,3]	Pix_In[1,1,1,1,1,1,1,1]
4-bit, Buf 1	Pix_In[6,6,6,6,6,6,6,6]	Pix_In[4,7,4,7,4,7,4,7]	Pix_In[5,5,5,5,5,5,5,5]
8-bit, Buf 0	Pix_In[2,5,2,5,2,5,2,5]	Pix_In[0,3,6,0,3,6,0,3]	Pix_In[1,4,7,1,4,7,1,4]
8-bit, Buf 1	Pix_In[14,17,14,17,14,17,14,17]	Pix_In[12,15,18,12,15,18,12,15]	Pix_In[13,16,19,13,16,19,13,16]
12-bit, Buf 0	Pix_In[2,5,8,11,2,5,8,11]	Pix_In[0,3,6,9,0,3,6,9]	Pix_In[1,4,7,10,1,4,7,10]
12-bit, Buf 1	Pix_In[14,17,20,23,14,17,20,23]	Pix_In[12,15,18,21,12,15,18,21]	Pix_In[13,16,19,22,13,16,19,22]
24-bit	Pix_In[2,5,8,11,14,17,20,23]	Pix_In[0,3,6,9,12,15,18,21]	Pix_In[1,4,7,10,13,16,19,22]

3.4.3 Aux (Overlay/Underlay) Display Modes

In the 24-bit Newport System there are 8 bits of Aux. frame buffer memory which can be used as overlay or underlay. These planes are conceptually just above or below the traditional pixel planes. In the 8-bit system, there are only 2 bits of Aux memory. 1 bit of the configuration register (24/8-bit) selects between the 2 system configurations. 3 bits of the mode register (Aux_Pix_Mode) determine what mode the planes will be interpreted as. A buffer select bit supports double buffering of the overlay planes. Doubled buffered underlay is not supported.

If the system is in 8-bit mode and Pop Up planes are enabled, then no aux planes are available and all overlay and underlay modes are off.

Table 3i: Aux Pixel Display Modes

Aux_Pix_Mode[2:0]	OVL_Buf_Sel	24/8-bit	Pup_En	Description
000	x	x	x	Off
001	x	0	x	8-Bit Underlay
010	x	0	x	8-Bit Overlay
110	0	0	x	4-Bit Overlay, Buf 0
110	1	0	x	4-Bit Overlay, Buf 1
111	x	0	x	4-Bit Overlay, 4-Bit Underlay
001	x	1	0	2-Bit Underlay
010	x	1	0	2-Bit Overrlay
110	0	1	0	1-Bit Overlay, Buf 0
110	1	1	0	1-Bit Overlay, Buf 1
111	x	1	0	1-Bit Overlay, 1-Bit Underlay
xxx	x	1	1	No Overlay Available, Pop Planes instead

The Overlay/Underlay modes are always in Color Index format. No RGB modes are allowed. The formats are similar to pixel CI mode, with the 5 most significant bits coming from the mode register (Aux_MSB_CMAP[4:0]). For 8-bit and 4-bit modes, 256 entry tables are used, with only the first 16 being active in the 4-bit case. In the 2-bit modes (8-bit system), 256 entry maps are also used, even though only 2 or 4 entries will be used. There should not be a problem with CMAP memory space, because even the 8-bit system will have an 8K color map. The Pup[1:0] refer to the 2 Aux planes of the 8-bit system. They function the same as Aux[1:0]. The naming is different because physically they reside in different places.

Table 3j: Output (To CMAP) Address Formats, Overlay or Underlay Modes

Mode	Aux Pix_Out[12:8]	Aux Pix_Out[7:4]	Aux Pix_Out[3:0]
8-bit, Overlay or Underlay	AUX_MSB_CMAP[4:0]	Aux[7:4]	Aux[3:0]
4-bit, Overlay Buf 0	AUX_MSB_CMAP[4:0]	0000	Aux[3:0]
4-bit, Overlay Buf 1	AUX_MSB_CMAP[4:0]	0000	Aux[7:4]
4-bit, Overlay (in the 4bit Overlay, 4bit Underlay mode)	AUX_MSB_CMAP[4:0]	0000	Aux[3:0]
4-bit, Underlay (in the 4bit Overlay, 4bit Underlay mode)	AUX_MSB_CMAP[4:0]	0000	Aux[7:4]
2-bit, Overlay or Underlay	AUX_MSB_CMAP[4:0]	0000	00 & Pup[1:0]
1-bit, Overlay Buf 0	AUX_MSB_CMAP[4:0]	0000	000 & Pup[0]
1-bit, Overlay Buf 1	AUX_MSB_CMAP[4:0]	0000	000 & Pup[1]
1-bit, Overlay (in the 1bit Overlay, 1bit Underlay mode)	AUX_MSB_CMAP[4:0]	0000	000 & Pup[0]

Table 3j: Output (To CMAP) Address Formats, Overlay or Underlay Modes

Mode	Aux Pix_Out[12:8]	Aux Pix_Out[7:4]	Aux Pix_Out[3:0]
1-bit, Underlay (in the 1bit Overlay, 1bit Underlay mode)	AUX_MSB_CMAP[4:0]	0000	000 & Pup[1]

3.4.4 Cursor Display Mode

The cursor pixel format will also be Color Index mode like the Aux planes. Unlike the Aux planes, the cursor CMAP address gets its most significant bits from a register. Thus a single cursor color map (4 entries) is used for the entire screen. 8-bits was chosen as the register length because that is the maximum number that can be sent in one display bus transfer. Theoretically only 3 color map entries need to be used for the cursor, but since all other modes use 256 entries, it may be difficult to partition less than 256 entries for the cursor map.

Table 3k: Output (To CMAP) Address Format, Cursor Mode

Cur Pix_Out[12:5]	Cur Pix_Out[4:2]	Cur Pix_Out[1:0]
Cur_Reg[7:0]	000	Cur_In[1:0]

3.4.5 Pop-Up Display Mode

The pop-up pixel format will be very similar to the cursor pixel format. Unlike the cursor, however, pop up planes are only available if Pup_En is set in the System Configuration register. In the 24-bit system there are dedicated Pop-up bits in the Frame Buffer, so if Pop-ups are disabled, then the bits are wasted. In the 8-bit system, the Pop-up bits are shared witht the Aux bits. Thus if Pop-ups are enabled, no overlay or underlay modes exist. If Pop-ups are disabled, then the 2-bit Aux modes are available. Since the System Configuration register, cannot be written to in a manner to synchornize it with a point on the screen, it is not possible to have one window with Pup planes and another with Aux planes. It is one or the other for the entire screen.

Table 3l: Output (To CMAP) Address Format, Pop-Up Mode

Pup Pix_Out[12:5]	Pup Pix_Out[4:2]	Pup Pix_Out[1:0]
Pup_Reg[7:0]	000	Pup_In[1:0]

3.4.6 Video Display Modes

XMAP9 will support the Express Video Expansion board which has 2 display modes. In the 12-bit, Starter Mode, 12 bits of RGB data come from the video board and 12 bits are sent to the video board. To get a 24-bit RGB value, 4 bits of each color need to be duplicated. In the 24-bit, Express Mode, all 24-bits are used for RGB and the bus is bidirectional. The data is sent directly to the CMAP without modification. The Starter/Express Mode bit is in the configuration register.

Table 3m: Output (To CMAP) Pixel Format, Video Express/Starter Modes

Mode	Video Blue Pix_Out[23:16]	Video Green Pix_Out[15:8]	Video Red Pix_Out[7:0]
Starter Video Mode (12 bits in, 12 bits out)	Vid_In[19:16, 19:16]	Vid_In[11:8, 11:8]	Vid_In[3:0, 3:0]

Table 3m: Output (To CMAP) Pixel Format, Video Express/Starter Modes

Mode	Video Blue Pix_Out[23:16]	Video Green Pix_Out[15:8]	Video Red Pix_Out[7:0]
Express Video Mode (24-bit input)	Vid_In[23:16]	Vid_In[15:8]	Vid_In[7:0]

3.4.7 Video Transfer Modes (Graphics to Video)

The path from graphics to video actually goes through the XMAP9 twice. First the data is taken from the frame buffer and selected normally. The display data is sent to the CMAP and on to the DAC's. The output of the CMAP is also sent back to the XMAP and then repackaged and sent to the video board. Since the output of the CMAP is always 24-bit RGB, selection for video is much simpler than selecting display data.

The first selection is between Express and Starter Video modes. If the system is in Starter Mode, then the 24-bit RGB is dithered (using a 4x4 spatial matrix) down to 12-bits. If it is Express Mode, and alpha mode is disabled, then all 24-bits of RGB are sent to the video board. If alpha is enabled, then the RGB value is sent to the CMAP normally while the alpha value is pipelined in the XMAP9. In the single buffered case, when the 24-bit RGB value is returned from the CMAP, the 16 most significant bits are merged with the 8 bits of alpha. No dithering is done, since there were only 12-bits of color to begin with from the frame buffer. The double buffered case is similar, except there are only 4-bits of alpha which are duplicated also. Alpha is only allowed in Expres Video Mode.

Table 3n: Output (To Video) Pixel Formats, RGB Mode

Mode	Blue Vid_Out [23:20]	Blue Vid_Out [19:16]	Green Vid_Out [15:12]	Green Vid_Out [11:8]	Red Vid_Out [7:4]	Red Vid_Out [3:0]
Starter Video Mode (12 bits in, 12 bits out)	CMAP_RGB [23:20]	Inputs in Starter Video Mode	CMAP_RGB [15:12]	Inputs in Starter Video Mode	CMAP_RGB [7:4]	Inputs in Starter Video Mode
24-bit, Express Video Mode	CMAP_RGB [23:20]	CMAP_RGB [19:16]	CMAP_RGB [15:12]	CMAP_RGB [11:8]	CMAP_RGB [7:4]	CMAP_RGB [3:0]
12-bit RGB with 8-bit Alpha	CMAP_RGB [23:20]	CMAP_RGB [19], Alpha[7:5]	CMAP_RGB [15:12]	CMAP_RGB [11:10], Alpha[4:3]	CMAP_RGB [7:4]	CMAP_RGB [3], Alpha[2:0]
8-bit RGB with 4-bit Alpha *	CMAP_RGB [23:20]	CMAP_RGB [19], Alpha[7:5]	CMAP_RGB [15:12]	CMAP_RGB [11:10], Alpha[4, 7]	CMAP_RGB [7:4]	CMAP_RGB [3], Alpha[6:4]

*Note: 8-bit RGB with 4-bit Alpha is used as a double buffered mode.

4.0 System Interface

4.1 Frame Buffer/Reorg Interface

Unlike previous XMAP chips, XMAP9 does not have a direct interface with the Frame Buffer VRAM. Instead a Reorganization chip (RO1) deals with timing requirements of the VRAM as well as the interleaved data. Consequently, no pixel fifo is needed in XMAP9 and data is inputted at 1/2 the pixel rate. This can reach a peak rate of 70Mhz. The data in the 24-bit system is organized as:

- 24-bits pixel data
- 8-bits aux data
- 2-bits pup data

In the 8- bit system the data from the frame buffer is organized as:

- 8-bits pixel data
- 2-bits aux data

4.2 CMAP Interface

The interface with the CMAP chip has 2 ports, one an input and one an output. Because the interface can run at a peak speed of 70Mhz, all pins are unidirectional and all pixel data including RGB is fed to the CMAP. The output port consists of a 24-bit pixel value and a two bit tag that is sent to the CMAP to indicate whether the data is color index or RGB. The tag also determines which of 3 256x24bit hardcoded RGB color maps is to be used. The format of the tag is:

00 = color index mode
01 = RGB Mode, 11101 + 8-bit color value
10 = RGB Mode, 11110 + 8-bit color value
11 = RGB Mode, 11111 + 8-bit color value

If the pixel is RGB, all 24 bits are used. Actually in RGB mode there are 3 sets of 3 256x8 color maps. For example if the Pix_Tag is 01 then Red is mapped to address: 11101 + Red[7:0] to get an 8-bit Red Color Value. Green is mapped to address: 11101 + Green[7:0] in its 8-bit map, and similarily for Blue[7:0] in its 256x8 map. In color index mode, bits 12 to 0 are used to address the 8k map while the upper 11 bits are ignored.

The output of the CMAP is fed back to the XMAP and then passed onto the video extension board. This interface is also running at pixel_clock/2, so it can reach 70Mhz. The 24-bit return value is always RGB and sent to the Video Interface section of the XMAP chip.

4.3 VC2 Interface

The VC2 sends two types of signals. The C_Blank_N timing signal is used to synchronize data transfers to the video option board. The rising edge of the C_Blank_N signal will coincide with the rising edge of the Pixel Clock Divided by 4 rising edge. This way, the 35Mhz clocks at the two sides of the interface will not be out of phase. The C_Blank_N timing signal is also used to indicate when the display is not active or blanked. During this blanking time, data will be read from the FIFO into the Mode Register File. This signal will also be used by the dithering circuitry to set the XY position for spatial dithering. The other signals are the Cursor and Display ID signals which correspond to a particular pixel. Because of this, their transfer is synchronized with that of the pixel data. All signals from the VC2 arrive at Pix_Clk/2 rates.

4.4 Video Interface

The interface with the Video expansion board consists of a 24-bit bidirectional data bus and 3 control signals. Because of board timing issues, the interface has been split into 2 channels each running at Pixel Clock Divided by 4. The Pix_Div4 clocks will be synchronized with the H_Blank_N signal of the AB1 using the XMAP9's own C_Blank_N signal, so that all chips (XMAP9 both even and odd, and video board) will have the same phase clock. For purposes of discussion, the following will apply to both video channels even though only one will be discussed.

The video expansion board has 2 mode of operation: Starter and Express Modes. In both modes, the Video_Key input signal determines for each video pixel, whether or not it should be displayed. Data from the Video Board is always RGB. In Starter Mode, the data bus is split into 12 bits of input and 12 bits of output, so that video and graphics can be mixed continuosly on the screen. Resolution in Starter Mode is limited to 12-bit RGB. The 24-bit RGB data from the CMAP is spatially dithered to 12 bits with each color going from 8-bit to 4-bit resolution.

In Express mode, the direction of the bus is controlled by the XMAP9 pin, Vid_OE. Besides allowing for the 24-bit RGB modes, Express mode also has the capacity for a 16-bit RGB, and 8-bit alpha mode. If this mode is on, then the XMAP9 activiates the Vid_Alpha signal when sending the data to the video option board. This last mode can also be double buffered, giving 8-bit RGB and 4-bit alpha which is then duplicated when sent to the video option board.

4.5 Display Control Bus Interface

The XMAP9 host interface is through the REX3. The interface is through the 8-bit wide display bus using the asynchronous protocol without acknowledge signal. The timing of a Read and Write sequence is shown in Figure 4. The setup and hold times for the data transfer are determined by the Pixel Clock Rate because the Display Bus runs at a constant 33Mhz.

4.6 Tester Interface

The XMAP9 has a limited JTAG implementation which allows for selected boundary scan test functionality.

For the boundary scan chain order, and the order file for the rest of the flip flops in the chip. In LSI scan order format the file is currently in:

campanile:/d/people/chueh/chips/xmap9/lsi/lscan/S_XMAP9.SOF

The coverage of ATPG for the XMAP9 was very good, probably because 75% of the chip is scanable Flip-Flops. According to the LSI ATPG tools, the coverage is 100% of testable faults. Assuming the untestable faults are mostly the RAM, we only subtract the black boxed faults, which give us campanile.esd:/d/people/chueh/chips/xmap9/lsia fault coverage of 96%.

The Parametric and Tri-state (P and Z) tests were done and the results are in the directory:

campanile.esd:/d/people/chueh/chips/xmap9/lsi

In this directory will be the P test control file is S_XMAP9.SCL1P and the vectors are in S_XMAP9.TPT1P. For the Z test the relevent files are S_XMAP9.SCL1Z and S_XMAP9.TPT1Z. The pinout is in the file XMAP9.BDCMD.

Please see me for further instructions on using the JTAG controller for boundary scan testing.

5.0 Architectural Description

Total gate count: 19,000 gates. This includes 4500 gates of RAM. This is well under the 26,400 usable gates in the die size LCA100066P. Array gate usage is 29%.

5.1 Display Select Block

This block formats the data and selects the particular plane that is to be sent to the CMAP. It takes as input Cursor Data, Pixel Data, Aux Data, Pup Data, Video Data as well as 2 8-bit registers that have to location of the Cursor and Pup color maps. It also uses 23 bits of the mode register along with 3 system configuration regsiter bits to determine which plane to select for the display and how the data should be formatted. There are only 3 outpus, Pixel Out, Pixel Tag and Gamma Bypass. There are 4 main blocks: Zero Detect, Pixel Format, Pixel Select1 and Pixel Select2.

Zero Detect just does some of the zero detection that can't be done in Pixel Format, due to timing constraints.

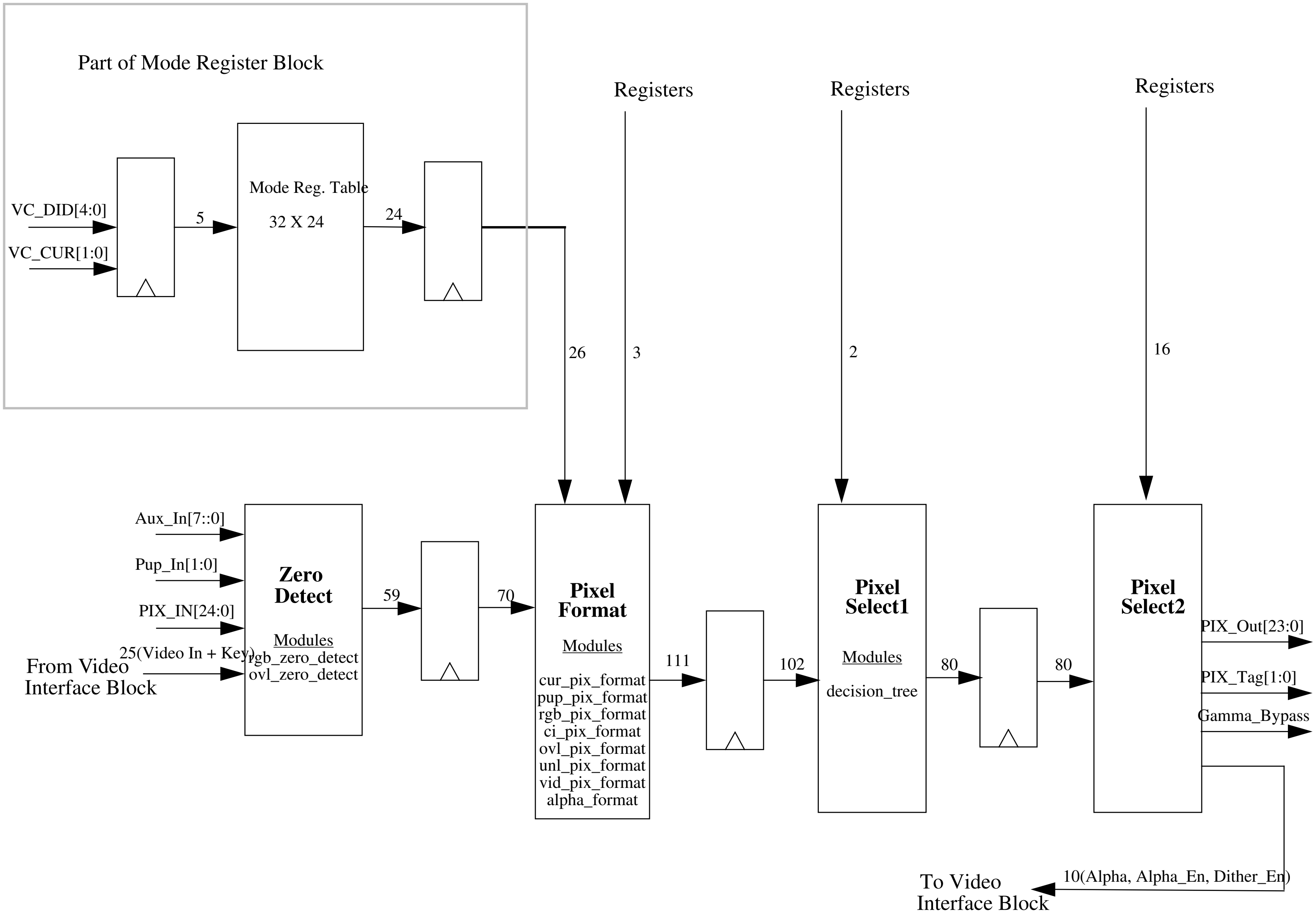
Pixel Format formats each of the planes so their data will look like a CI address or RGB data. It also duplicates RGB data, so that a 4-bit mode will become a 24-bit quantity.

Pixel Select1 goes through the decision tree described in Section 3.3.

Pixel Select2 actually muxes the data, and sets the Pix_Tag bits

Display Select Block Diagram

Justin Chueh
6-2-92



5.2 DCB Interface

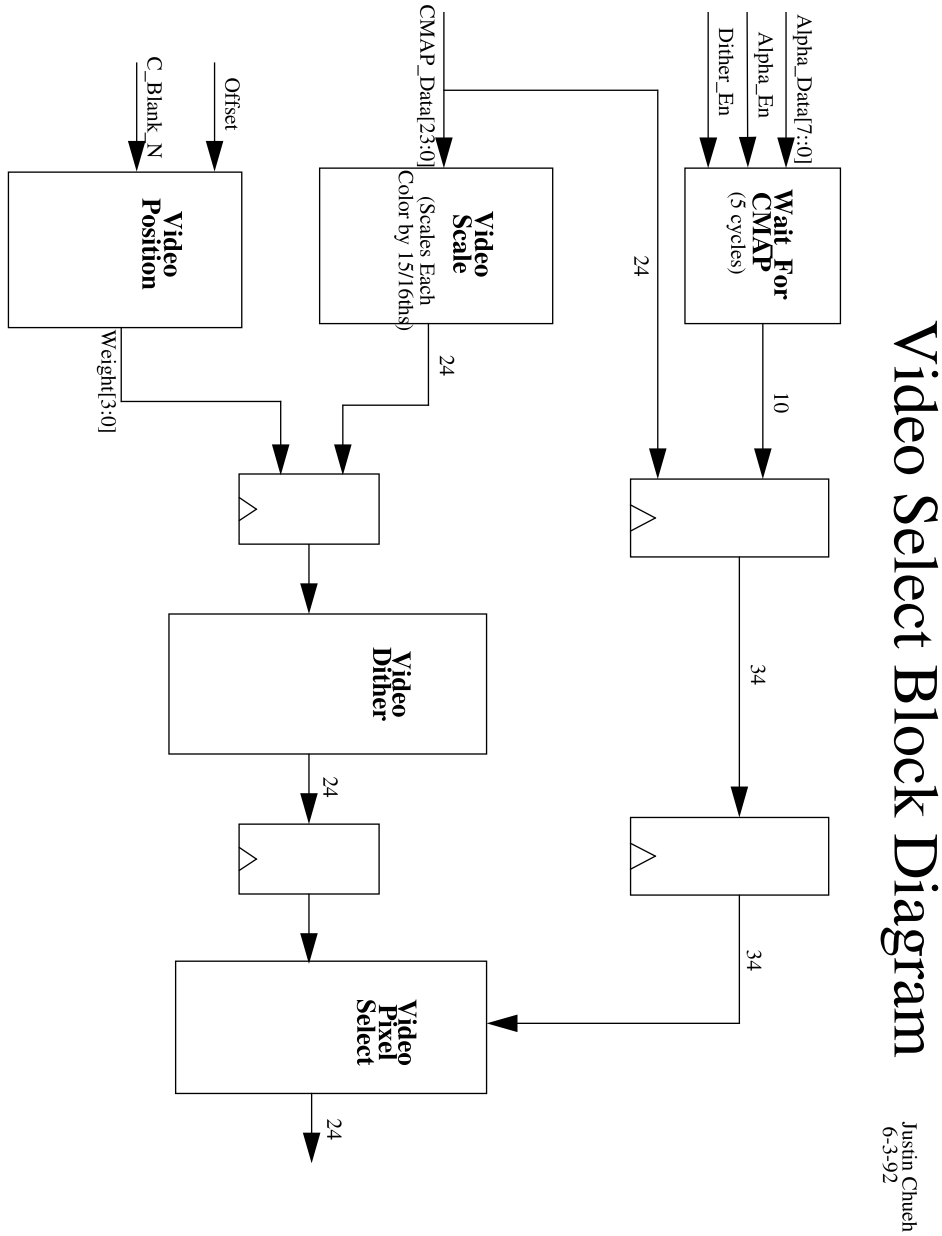
The DCB interface synchronizes the CS select signal which runs in a gio clock domain to the XMAP9 clock domain which is running at pixel clock div2 rates. During this synchronization the Data and CRS bits are latched into a level sensitive negative edge latch. Because the pixel clock rate can vary from 12 Mhz to 140 Mhz, there are restrictions on the width and time between CS pulses based upon the pixel frequency. These are documented in section 3.1.2 and 3.2. The DCB interface uses 460 gates.

5.3 Video Interface

The video interface block switches from a pixel div 2 clock domain to a pixel div 4 clock domain. The clocks are aligned with each other. The phase of the pixel div 4 clock is determined by the C_Blank_N signal. This signal is programmed in a way such that the phase of the pixel div 4 clock is the same as the clock on the AB1 chip. This block requires 2440 gates.

5.4 Video Select Block

The video select block, selects the pixel type to be sent to video. Since the output is always RGB there are a lot fewer choices than in the Display Select Block. The only choices are complete 24-bit RGB, Dithered 12-bit RGB, and Truncated 12-Bit RGB. Synthesized gate count: 1778



5.5 Mode Register Block

The Mode Register Block takes 8-bit bytes and reformats them as 29-bit Words (5 bits of RAM address and a 24-bit Mode Register Word). These words are put in a FIFO which is only read from during monitor blanking. The block also controls mode register RAM reads. The RAM is triple ported (2 Read, One Write). Two Read ports are necessary because of AC timing concerns. The access time of the RAM is too slow to run at 70 Mhz without ping-ponging the reads to give twice the cycle time.

Synthesized gate count = 2,712

Mode Register RAM gate count = 4484. (32x24T, 768 bits)

5.6 Registers

These are registers other than the mode register. Most of these are static signals which are only written once during power up routines.

Synthesized gate count: 398.

5.7 IO Flip Flop

Synthesized gate count: 920

6.0 System Implementation Issues

6.1 Double Buffered Overlay Planes

These are currently not supported by the X window system. Implementing them is difficult and will force a signficant performance penalty on the software. Unless the market nessitates a significant need for this feature, future versions of the XMAP may want to forgo implementing it.

6.2 Overlay versus Pop Up

In the current windowing system, the overlay planes do not change color maps on a per window basis. Thus any distinction between overlay and pop-up is really non-existent. Putting the overlay bits in the Mode Register is again a large amount of wasted gates with the current software.

6.3 Extra Frame-Buffer Bits

In the 24-bit system, there are 10 Aux bits (8 + 2 left over from the 8-bit configuration). Since only 8 bits are used, the extra 2 bits are configured as Pop-Up. This feature is also not used, so effectively, these 2 frame-buffer bits are wasted. Possible uses for these bits are:

Graphics Key Bits, more CID bits etc.

6.4 Cost Reduction

Removing the bits to support both these features from the mode register would reduce its size to 16 bits or possibly even 12 bits depending upon some of the video features like dithering control, or alpha control. This would save 6000 gates in the mode register alone, and probably reduce the total gate count of the XMAP to under 10,000 gates.

6.5 Video Timing Channels

6.5.1 RO1 Pixel Port

The number of pipeline stages in pix/2 clocks for the pixel, aux and pup data coming from the RO1 chips is 5 FF's through the XMAP9. The CMAP has 3 delays so from the input of the XMAP9 to the input of the DAC is 8 pipeline delays.

6.5.2 VC2 Did and Cursor Port

The number of pipeline stages in pix/2 clocks for the DID and Cursor data coming from the VC2 ASIC chips is 7 FF's through the XMAP9. The CMAP has 3 delays so the total delay from the VC2 through the XMAP9 to the input of the DAC is 10 pipeline delays.

6.5.3 AB1 Video Data Port

The video arrives in 2 channels based on a pix/4 clock. The pipeline delays for Channel 0 through the XMAP9 are 6. Video data in channel1 takes one cycle longer, but the AB1 knows this and will send it earlier by 1 cycle. The delay for Channel 1 is 7 FF's. Adding the 3 delays for the CMAP, we get delays from the AB1 output to the DAC of 9 and 10 pipeline stages for Video Channels 0 and 1 respectively.

6.5.4 CMAP RGB Data Port

RGB data from the CMAP outputs is sent through the XMAP9 and on to the Video System. The XMAP9 to AB1 (Video Option Card) transfers occur on a pix/4 clock so each channel will have a one pipe diffenence in delay. For channel 0 the data takes 6 cycles to be processed through the XMAP9. For Video data in channel1 the delay is 5 Flip Flop delays.

6.5.5 Composite Blank (from VC2)

This signal is the only video timing channel the XMAP9 uses so it has many restrictions on it's programming.

Rising Edge (end of blanking)

1. Restriction due to RAM Access

We can't write the XMAP9 Mode Registers during visible time, otherwise the screen will sparkle and flash during the changes. In order to guarantee that the last write to the mode register ram has finished before visible pixels are sent through the XMAP9, the C_Blank_N signal must be deactivated (rising edge) 13 cycles before the RAMDAC C_Blank_N signal.

2. Restriction due to synchronization with Video Option Board

Transfers with the video option board (express video) occur at a pix/4 clock rate. Since both chips only receive a pix/2 clock, the state machines are synchronized with the rising edge of C_Blank_N. The rising edge of C_Blank_N must coincide with the rising edge of the pix/4 clock. Because of this restriction, the XMAP9 C_Blank_N signal rising edge must be 14 cycles before video data is needed at the RAMDAC. This may not be 14 cycles earlier than the RAMDAC C_Blank_N because video windows have to be a certain distance from the edge of visible screen.

3. Restriction due to Dithering

The dithering matrix is reset based upon the rising edge of C_Blank_N each line. Thus in order to provide a consistent dither pattern, the rising edge of C_Blank_N has to occur the same number of cycles before RAMDAC C_Blank_N every line and every frame.

Falling Edge (start of blanking)

Only Restriction due to RAM Access.

We don't want to write to the RAM while visible pixels are still passing through, so the earliest the XMAP9 C_Blank_N can fall is 10 cycles before the RAMDAC C_Blank_N falling edge.

Optimization

In order to allow the highest bandwidth of Mode Register Writes, it is advantageous to have C_Blank_N low for as long as possible. Thus the earliest Falling Edge and latest Rising Edge are optimum.

7.0 Device Characteristics

7.1 Recommended Operating Conditions

Parameter	Symbol	Min	Typ	Max	Units
Power Supply	Vdd	4.75	5.00	5.25	Volts
Ambient Operating Temperature	TA	0		45	Deg. C
Pixel (To CMAP) Output Load	CL			30.0	pF
Video Bus Output Load	CL			37.0	pF
DCB Output Load	CL			50.0	pF
Junction Temperature	Tjmax	0.0		110	Deg. C

7.2 AC Characteristics

Parameter	From	To	Min	Max	Units
Max Freq.		Pixel Bus and Internal	0	70	Mhz
Max Freq.		Video Bus	0	35	Mhz
tplh	CLK	Pix_Out (To CMAP, TTL Levels)	2.2	9.6	ns
tphl	CLK	Pix_Out	2.7	11.5	ns
tplh	CLK	Video Bus (CMOS Levels)	5.0	24.6	ns
tphl	CLK	Video Bus	5.0	24.6	ns
tplh	Vid_OE	Video Bus	6.0	26.4	ns
tphl	Vid_OE	Video Bus	6.0	26.4	ns

Parameter	Signal	Min	Max	Units
Clock Freq.	PCLK_Div2, PCLK_Div2_Out_FF	0	70	Mhz
Duty Cycle	PCLK_Div2, PCLK_Div2_Out_FF	0	100	%
Setup	Reset_N	0.55		ns
Setup	Pix_In (Data from RO1)	0.45		ns
Setup	CMAP_RGB (Pixel from CMAP)	1.6		ns
Setup	Video Bus	1.4		ns
Hold	Reset_N	1.5		ns

Parameter	Signal	Min	Max	Units
Hold	Pix_In (Data from RO1)	2.6		ns
Hold	CMAP_RGB (Pixel from CMAP)	1.6		ns
Hold	Video Bus	2.0		ns

8.0 Revision History

1.0 First Released Version

1.1 Minor changes including some changes with underlay selection, and register numbering.

2.0 Post Signoff.

2.1 Almost complete. Missing hardware descriptions of blocks for next XMAP designer and explanation of test interface.