ReOrganizer 1 (RO1) Specification

1.0 Introduction

The RO1 is a 120 pin gate array located between the framebuffer and the XMAPs in the Newport graphics. It is designed in a 1u CMOS Gate Array with about 10K gates. The chip is designed to run at a maximum frequency of 70 mhz. It receives the serial output of the 8 banks of the framebuffer , each eight bits wide. It then performs two main functions on the data before it sends it out to the XMAPs. The first one is to extract the overlay and the normal color for each pixel and combine them into a single stream . The second function is to rotate every 8 pixels by an amount that is decided by the scanline number for the pixels. This is to compensate for the scanline staggering technique that is used by REX3 to draw into the frame buffer. There are about 97 signal pins.It has 1K bits of fifo memory and about 4-5K logic. A block diagram of the RO1 is attached as Figure 1.

2.0 Functional Description

RO1 reads 8 bytes of data from the serial port of the framebuffer every serial clock time. It outputs two 12 bit pixels (odd and even) to the XMAPs every 1/2 dot clock rate. Every 8 bit planes require one RO1 to interface with the XMAPs. Therefore the 24 bit system will require 3 RO1s.

Every 32 bytes of overlay (4 bits per pixel) is followed by 64 bytes of normal color (8 bits per pixel) in the frame buffer. The RO1 starts reading the serial port some time before the horizontal blank time is turned off. It reads 32 bytes of overlay in four serial clock cycles (each cycle brings in 8 bytes) . The data is stored in a fifo. Then it starts reading the following 64 normal colors in 8 serial cycles. It then merges every 4 bits of overlay with its corresponding 8 bits of normal color for every pixel. This section operates on 8 pixel in parallel.

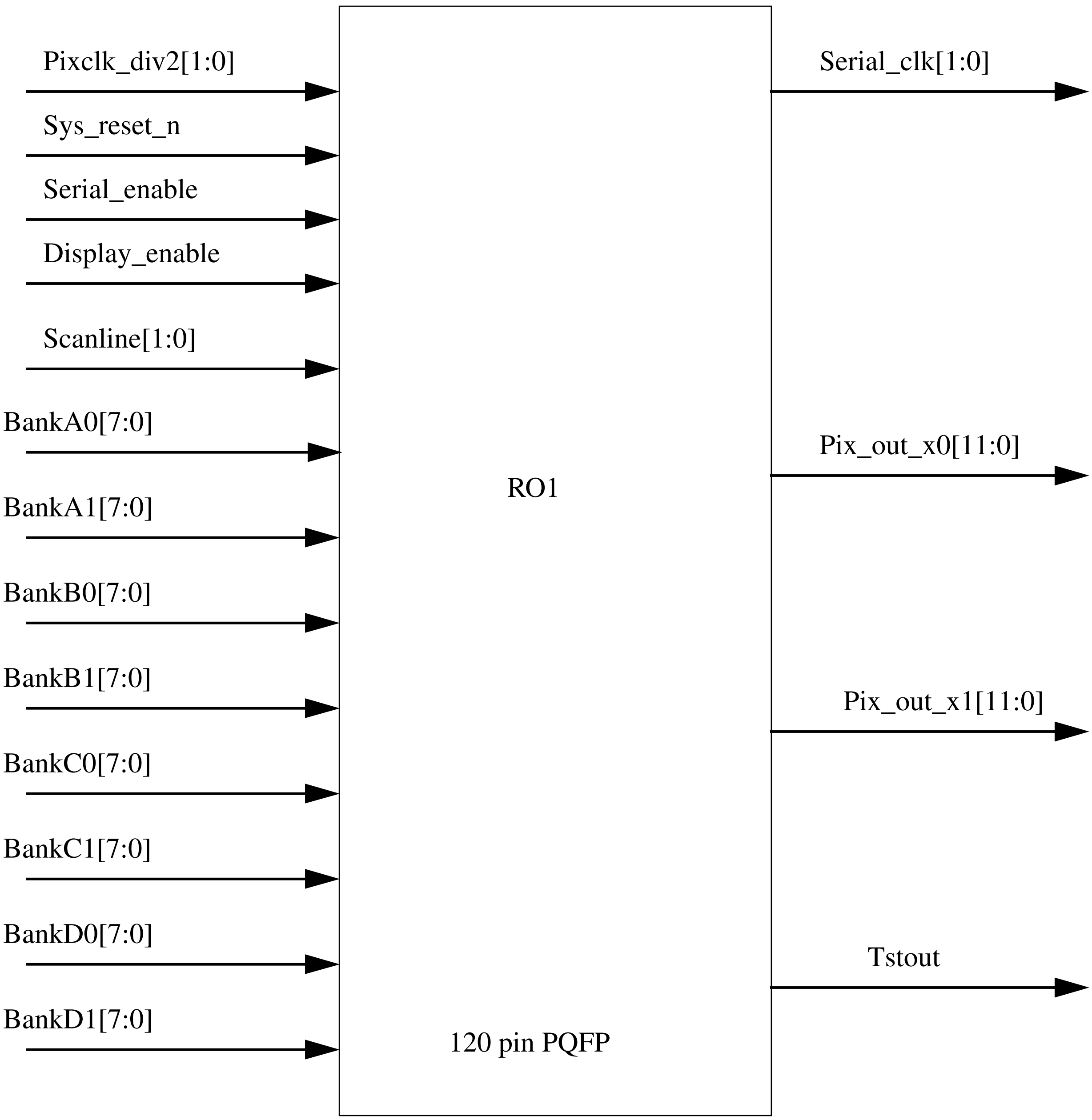
The pixel data is also stored in a fifo before it is merged with the overlay. The reason for this is that at least 20 pixels need to be in the fifo when RO1 is reading the overlay so that there wouldnot be any interruption on the video output in the middle of the scan line.

Once the overlay and normal color is merged the 8 pixels are separate in 2 groups : odd and even. They are then passed through a rotate circuit that can rotate them by 0,1,2 or 3 pixels thus compensating for the scan line staggering that was performed in REX3. The amount of rotation is controlled by 2 external signals (SCANLINE[1:0]) which is generated from REX3 , as it is keeping track of the scan line that is in the serial shift registers of the VRAMs. The rotate section operates at 1/8 of the dot clock rate. Table 1 shows the amount of rotate with respect to the SCANLINE inputs. The rotate block works the same way for odd and even pixels.
Refer to figure 3 for REX3 scanline staggering and RO1 block diagram where pixels A0, B0, C0 and D0 are even pixels and A1, B1 C1 and D1 are odd pixels.

RO1 PINOUT

Table 1: 1

Pin Name	Type	Drive	Function
Pixclk_div2	I	DRVT4	Pixel clock divide by 2 = 70 mhz
Pixclk_div2_1	I	DRVT4	Second clock pin used for output buffers
Serial_enable	I	IBUFN	Signal from VC2 which enables serail data from Vrams and genrates serial_clock
Display_enable	I	IBUFN	Signal from VC2 which enables reading out data from FIFOs.
Serial clock0	O	B4	Serial clock for reading out data from Vrams, pix_clock divide by 5 = 28mhz. This signal goes to 4 banks of Vrams.
Serial clock1	O	B4	Serial clock for reading out data from Vrams, pix_clock divide by 5 = 28mhz. This signal goes to 4 banks of Vrams.
Scanline[1:0]	I	IBUF	2 bits from REX3 for compensating sacn line scattering
Pix_out_x0[11:0]	O	B4	merged 8 bits of normal color and 4 bits of overlay to XMAP9_0 @ 70 mhz
Pix_out_x1[11:0]	O	B4	merged 8 bits of normal color and 4 bits of overlay to XMAP9_1 @ 70 mhz
BANKA0[7:0]	I	IBUF	Serial data from BANKA0 Vram
BANKA1[7:0]	I	IBUF	Serial data from BANKA1 Vram
BANKB0[7:0]	I	IBUF	Serial data from BANKB0 Vram
BANKB1[7:0]	I	IBUF	Serial data from BANKB1 Vram
BANKC0[7:0]	I	IBUF	Serial data from BANKC0 Vram
BANKC1[7:0]	I	IBUF	Serial data from BANKC1 Vram
BANKD0[7:0]	I	IBUF	Serial data from BANKD0 Vram
BANKD1[7:0]	I	IBUF	Serial data from BANKD1 Vram
SYS_RESET_N	I	IBUF	Global System Reset, active low
TSTOUT	O	B2	Parametric nand test out
Power Pins	N/A	N/A	22 total VDD and GND pins



Pixclk_div2[1:0]
Sys_reset_n
Serial_enable
Display_enable
Scanline[1:0]
BankA0[7:0]
BankA1[7:0]
BankB0[7:0]
BankB1[7:0]
BankC0[7:0]
BankC1[7:0]
BankD0[7:0]
BankD1[7:0]
RO1
120 pin PQFP
Serial_clk[1:0]
Pix_out_x0[11:0]
Pix_out_x1[11:0]
Tstout

The table below illustrates how even pixels will be rotated with respect to the scanline[1:0]

SCANLINE [1:0]	PX_A0 [11:0]	PX_B0 [11:0]	PX_C0 [11:0]	PX_D0 [11:0]
00	A0	B0	C0	D0
01	B0	C0	D0	A0
10	C0	D0	A0	B0
11	D0	A0	B0	C0

Table 1:

Table 1. Rotate Encoding

The outputs of the 2 rotate circuits are then sent through a 4 to 1 mux to reduce the number of pins that is required to transmit the pixel to the XMAPs. The odd and even outputs thus run at 1/2 the dot clock rate which translates to 70mhz at 76hz refresh.

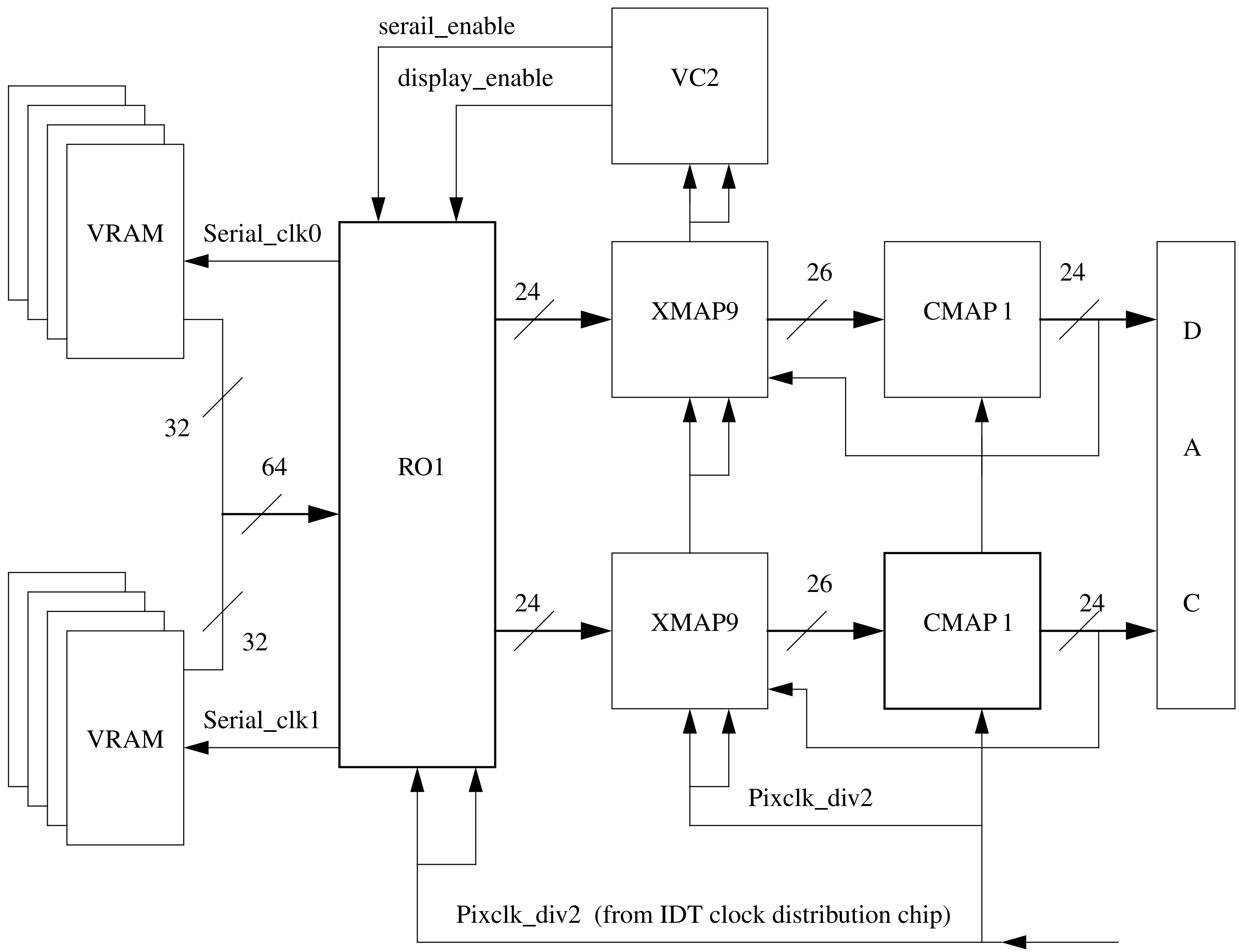
The Bank data from Vrams is latched into the chip by Serail_clk, the same clock which goes to the Vrams.

The dot clock rate at 76 hz is 140mhz (max) and for NTSC, it is 12 mhz. So, the chip supports frequency range between 6mhz and 70 mhz (1/2 dot clock rate).

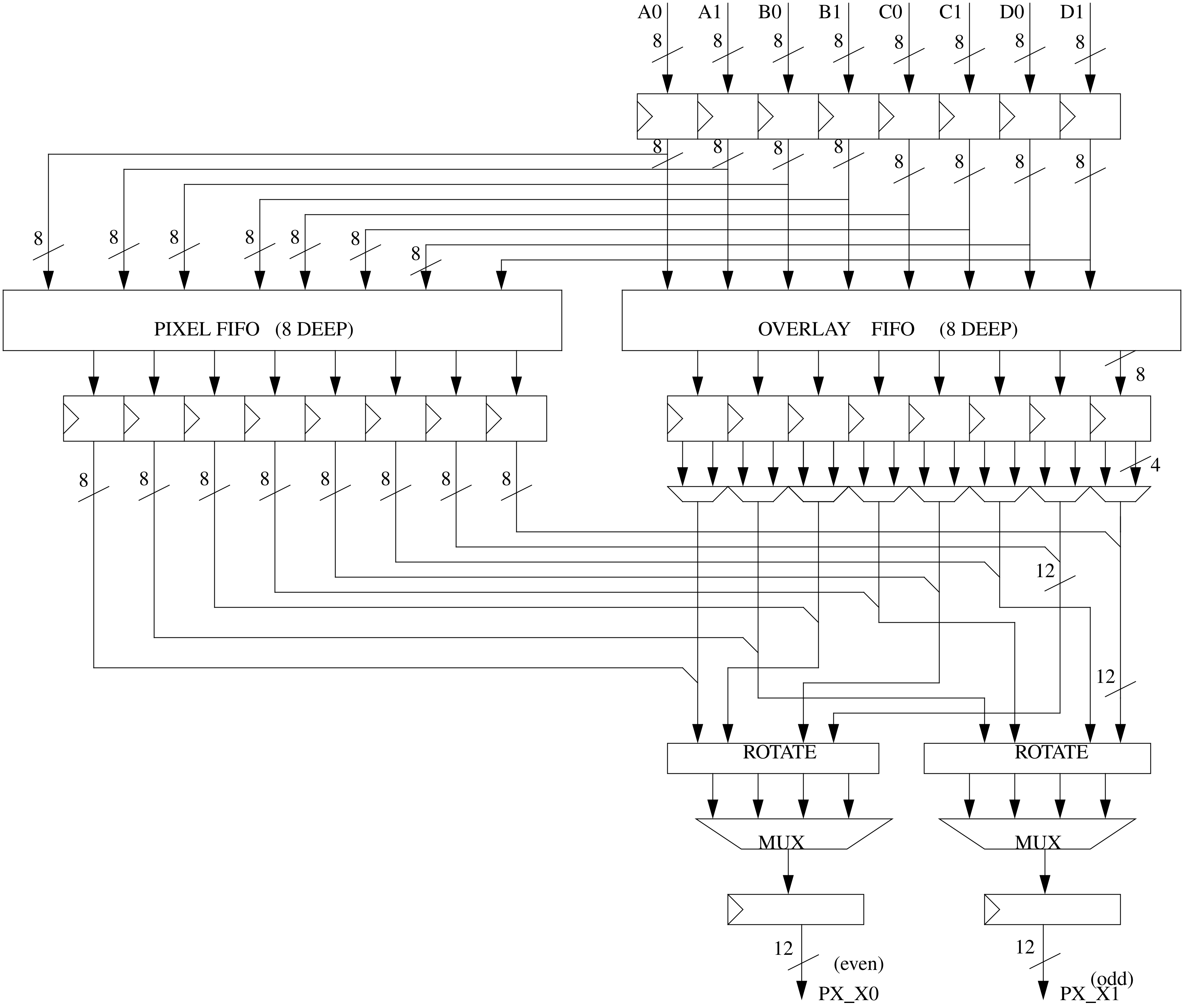
3.0 Serial Clock Generation

Since for every 12 serial reads (4 overlay cycles and 8 normal color cycles) 64 pixels are generated , the ideal rate for serial clock would be 12/64 or 1/5.33 of the dot clock. Obviously this is not an ideal ratio. Therefore RO1 uses 1/5 dot clock as its serial clock. This will result in a 35 nsec serial cycle time which is within reach of the 2M-80 VRAMs. It will generate this clock internally from 1/2 dot clock which it receives as an input. refer to newport.ro1.timing page 1 regarding generation of 1/5dot clock with respect to 1/2 dot clock and Serial_Enable signal from VC2.

The serial clock is enabled by SERIAL_ENABLE signal which is synchronous to 1/2 dot clock. This signal is generated from the VC2 and is activated some time before the active scan line starts. The serial clock to the VRAMs are not active the entire time that the serial_enable is active. The reason is that since the serial clock runs faster than the ideal clock , pixels will start accumulating in the RO1. The RO1 would have read all 1280 pixels across a scan line in ((1280 + 640) / 8) x 5 = 1200 dot clocks . The fifos inside the chip should be able to handle 80 pixels which will increase the die size. Therefore to reduce the fifo size after every 64 pixels that are read the serial clock will not be asserted for one cycle thus reducing the rate of input. This will increase the scanline read time to (1200 + (1280/64) * 5) = 1300 dot clocks. Since the reading starts before horizontal blanking this will be enough time to read the scanline. DISPLAY_ENABLE signal starts unload-



8 BIT MODE BACKEND , RO1 ON THE MOTHER BOARD


A0
A1
B0
B1
C0
C1
D0
D1
8
PIXEL FIFO (8 DEEP)
OVERLAY FIFO (8 DEEP)
4
12
ROTATE
ROTATE
MUX
MUX
(even)
PX_X0
(odd)
PX_X1

ing the fifos and merging the normal color and the overlay and sending out valid pixels to the XMAPs.

Table 2: RO1 PINOUT

Pin Name	Type	Drive	Function
P_pixclk_div2	I	DRVT4	Pixel clock divide by 2 = 70 mhz
P_pixclk_div2_1	I	DRVT2	Second clock pin used for output buffers
P_serial_enable	I	IBUFN	Signal from VC2 which enables serail data from Vrams and genrates serial_clock
P_display_enable	I	IBUFN	Signal from VC2 which enables reading out data from FIFOs.
P_serial clock0	O	B4	Serial clock for reading out data from Vrams, pix_clock divide by 5 = 28mhz. This signal goes to 4 banks of Vrams.
P_serial clock1	O	B4	Serial clock for reading out data from Vrams, pix_clock divide by 5 = 28mhz. This signal goes to 4 banks of Vrams.
P_scanline[1:0]	I	IBUF	2 bits from REX3 for compensating sacn line scattering
Pix_out_x0[11:0]	O	B4	merged 8 bits of normal color and 4 bits of overlay to XMAP9_0 @ 70 mhz
Pix_out_x1[11:0]	O	B4	merged 8 bits of normal color and 4 bits of overlay to XMAP9_1 @ 70 mhz
P_BANKA0[7:0]	I	IBUF	Serial data from BANKA0 Vram
P_BANKA1[7:0]	I	IBUF	Serial data from BANKA1 Vram
P_BANKB0[7:0]	I	IBUF	Serial data from BANKB0 Vram
P_BANKB1[7:0]	I	IBUF	Serial data from BANKB1 Vram
P_BANKC0[7:0]	I	IBUF	Serial data from BANKC0 Vram
P_BANKC1[7:0]	I	IBUF	Serial data from BANKC1 Vram
P_BANKD0[7:0]	I	IBUF	Serial data from BANKD0 Vram
P_BANKD1[7:0]	I	IBUF	Serial data from BANKD1 Vram
P_SYS_RESET_N	I	IBUF	Global System Reset, active low

Table 2: RO1 PINOUT

Pin Name	Type	Drive	Function
P_TSTOUT	O	B2	Parametric nand test out
Power Pins	N/A	N/A	22 total VDD and GND pins

There are two fifos in RO1 . One for the overlays and one for the normal color. The data is stuffed into the fifos at 1/5 dot clock rate and is read out of the fifos at 1/8 dot clock rate. Therefore the fifos act as temporary storage for the pixels and also are used to cross the 2 clock boundries. It takes 20 dot clocks to read the overlay data (4 cycles of 1/5 dot clock) , therefore the minimum requirement for the normal color fifo is 20 so that scan line is not interrupted. Since the normal fifo is 64 bits wide , therfore an 8 deep fifo is more than sufficient to hold the normal colors. The overlay fifo is set also 64 bits wide and an 8 deep fifo is chosen for it.

Before the display starts (CBLANK goes inactive) the overlay fifo needs to be loaded with 4 entries. The pixel fifo needs to be loaded with at least 6 entries and no more than 8 entries as there is no control over fifo getting full .. Simulations show that the maximum entries in the overlay fifo will be 6 and maximum entries in the pixel fifo will be 7 during the entire scan line. Therefore 8 deep fifos are chosen for both pixel and overlay.

5.0 Vc2 requirements

After Serial_Enable goes high, the Display_Enable should go high within 60 to 65 pixel clocks. Before Display_Enable can go high, the overlay fifo should have 4 entries and the pixel fifo should have 6 or 7 entries.
This requirement for Vc2 is mostly for the software as the timing for Serial_Enable and Display_Enable is programmable through the External Static Ram.

Loading overlay fifo with 4 entries: 4 1/5 dot clocks = 20 dot clocks
Loading pixel fifo with 6 or 7 entries: (6 or 7) 1/5 dot clocks = 30or 35 dot clocks
Need 7 (1/2 dot) clocks for latency (Serial_Enable) = 14 dot clocks
Latency for Display_Enable to read fifo is 2 (1/2 dot) clocks = -4 dot clocks.

Therefore time required between Serial_Enable and Display_Enable is 60 to 65 dot clocks.

5.0 Timing

The clock (External pixclk_div2) to the D input of XMAP9 F/F is 11.5ns .
The assumptions are:

a). Unidirectional output buffer with 5 pf of loading
b).Maximum trace capacitance of 10 pf.
c).SIMM module capacitance of 5 pf.
d).XMAP9 has ATPG and uses FD1S F/F to latch RO1 data

Worst case operating conditions being Tj = 110C, Vcc = 4.75V which makes the total lumped capacitance on output buffer to be 24 pf.

If the output buffer is made tristatable, then this will add another 1.5 ns to the delay .

Table 3 shows the setup and hold time requirements for the chip.

Table 3:

Name	description	min	max
tpou	pixclk_div2 to output delay of RO1 -XMAP bus	2 ns	9 ns
tpsu	Setup time for signals latched by pixclk_div2	1 ns	
thold	Hold time for signals latched by pixclk_div2	1.3 ns	
tssu	setup time with respect to SCLK for RO1-VRAM	4 ns	
tsh	Hold time with respect to SCLK for RO1-VRAM	1 ns	
tck_pixclk_div2	Clock period	14 ns	
tck_pixclk_div5	Clock period	35 ns	

. 6.0 Testability

There is no ATPG or JTAG support for this chip. The test patterns will be written such that it will cover the full toggle coverage plus it will have enough patterns to cover all boundary conditions.

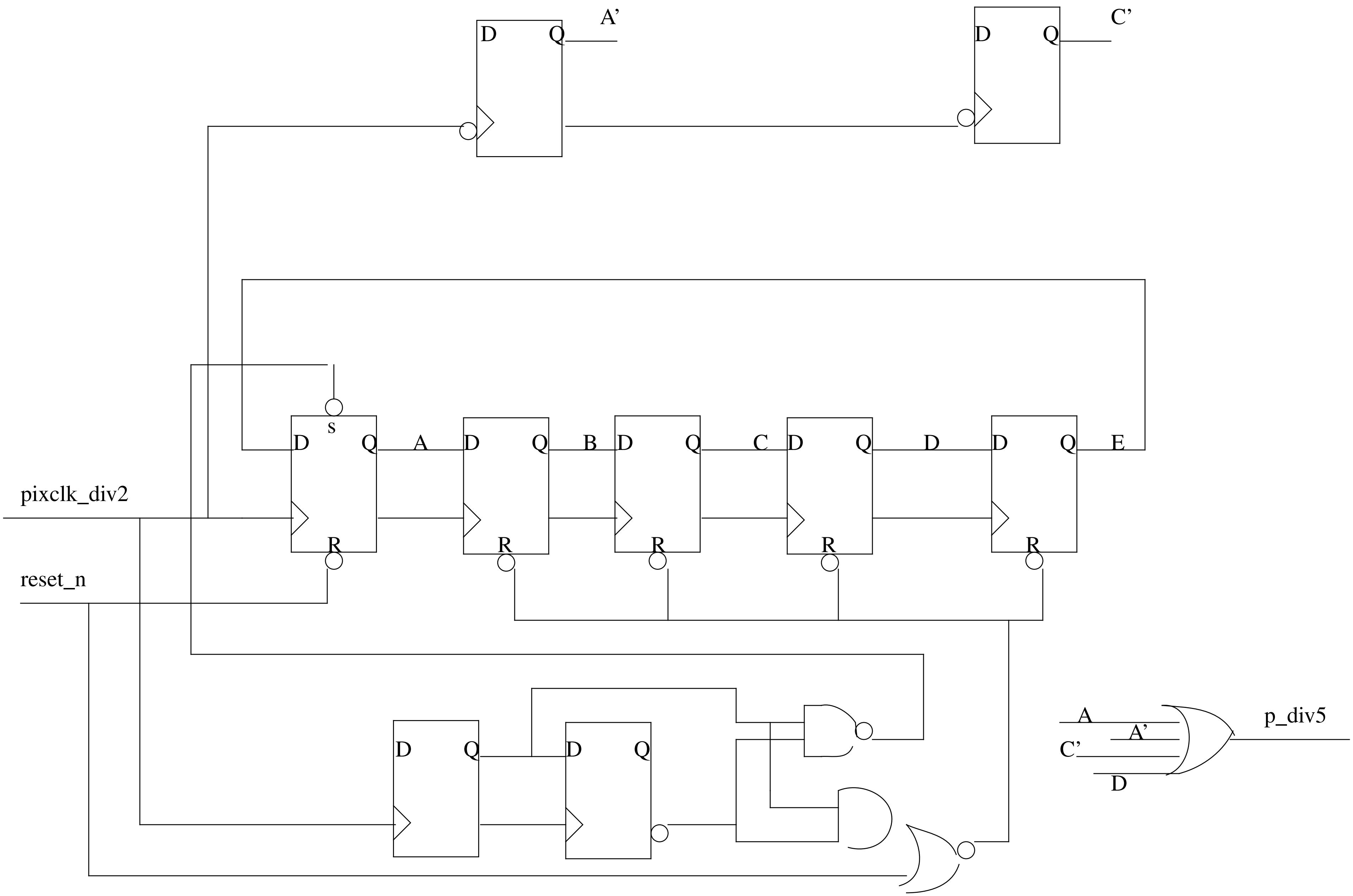
ISSUE:

From the previous page discussions about the clock to output buffer delay, it can be seen that the delay for unidirectional buffer is 11.5 ns. If the buffer is made tristatable, (so that we can do "bed of nails" testing) then the delay increases to 13 ns. and will need one extra pin for tristae enable. (RO1 has extra pin)

By adding this test, the signal traces need to have 'through hole' Vias and this can add another 5 pf of loading . This will not be a problem for the signals in the Mother Board going from RO1 to XMAP9 as there will not be a SIMM module connector loading. But for the signals on the connector, this will be extra loading for RO1 output buffers.

The solution can be that XMAP9 should have 0 ns setup or negative setup and XMAP9 needs to make sure taht it will meet the hold time.

PIX_DIV2 TO DIV5 CIRCUIT



VIDEO TIMING CHANNEL REQUIREMENTS

There are two fifos in RO1 . One for the overlays and one for the normal color. The data is stuffed into the fifos at 1/5 dot clock rate and is read out of the fifos at 1/8 dot clock rate. Therefore the fifos act as temporary storage for the pixels and also are used to cross the 2 clock boundries. It takes 20 dot clocks to read the overlay data (4 cycles of 1/5 dot clock) , therefore the minimum requirement for the normal color fifo is 20 so that scan line is not interrupted. Since the normal fifo is 64 bits wide , therfore an 8 deep fifo is more than sufficient to hold the normal colors. The overlay fifo is set also 64 bits wide and an 8 deep fifo is chosen for it.

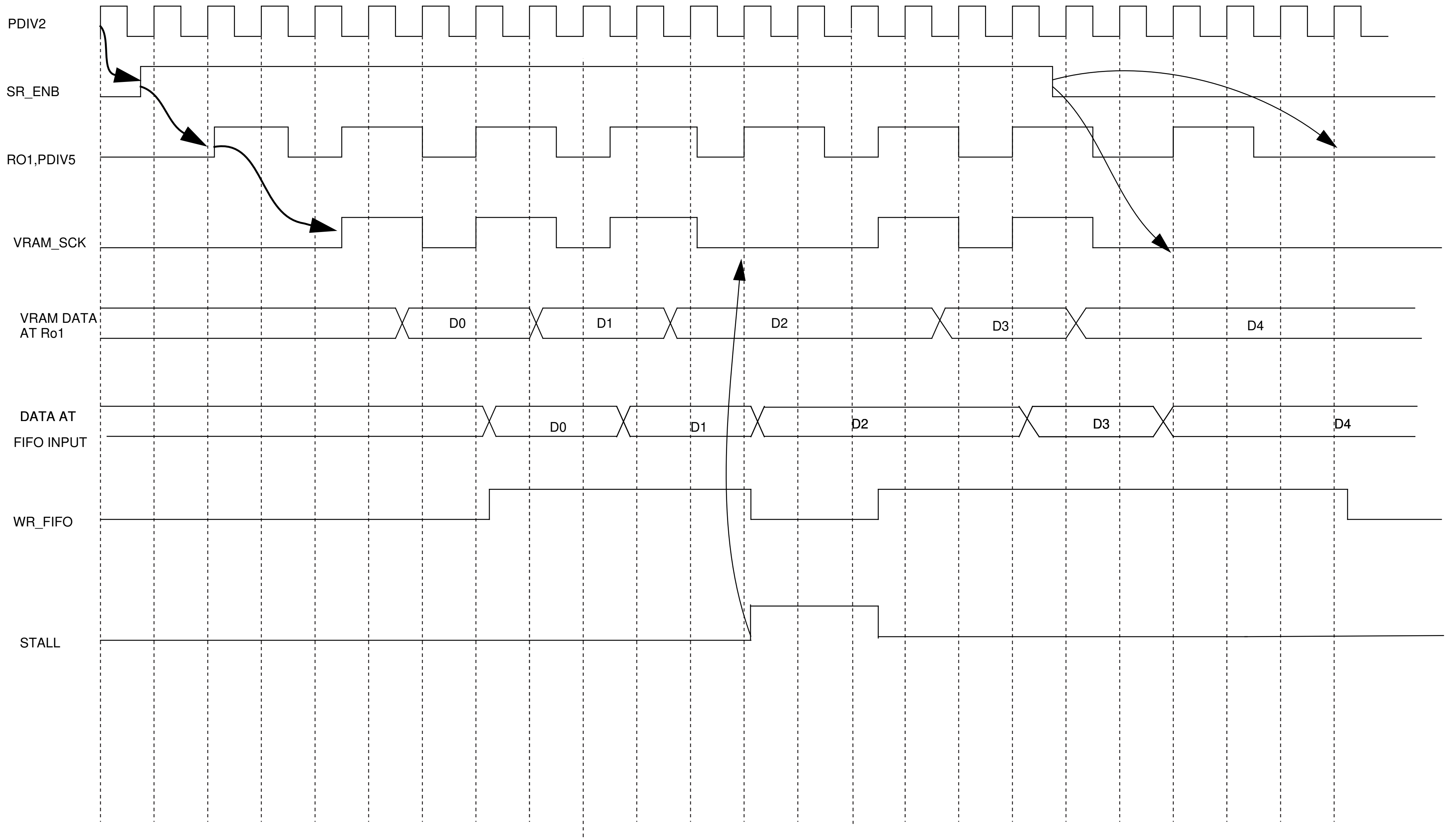
Before the display starts (CBLANK goes inactive) the overlay fifo needs to be loaded with 4 entries. The pixel fifo needs to be loaded with at least 6 entries and no more than 8 entries as there is no control over fifo getting full .. Simulations show that the maximum entries in the overlay fifo will be 6 and maximum entries in the pixel fifo will be 7 during the entire scan line. Therefore 8 deep fifos are chosen for both pixel and overlay.

5.0 Vc2 requirements

After Serial_Enable goes high, the Display_Enable should go high within 60 to 65 pixel clocks. Before Display_Enable can go high, the overlay fifo should have 4 entries and the pixel fifo should have 6 or 7 entries.
This requirement for Vc2 is mostly for the software as the timing for Serial_Enable and Display_Enable is programmable through the External Static Ram.

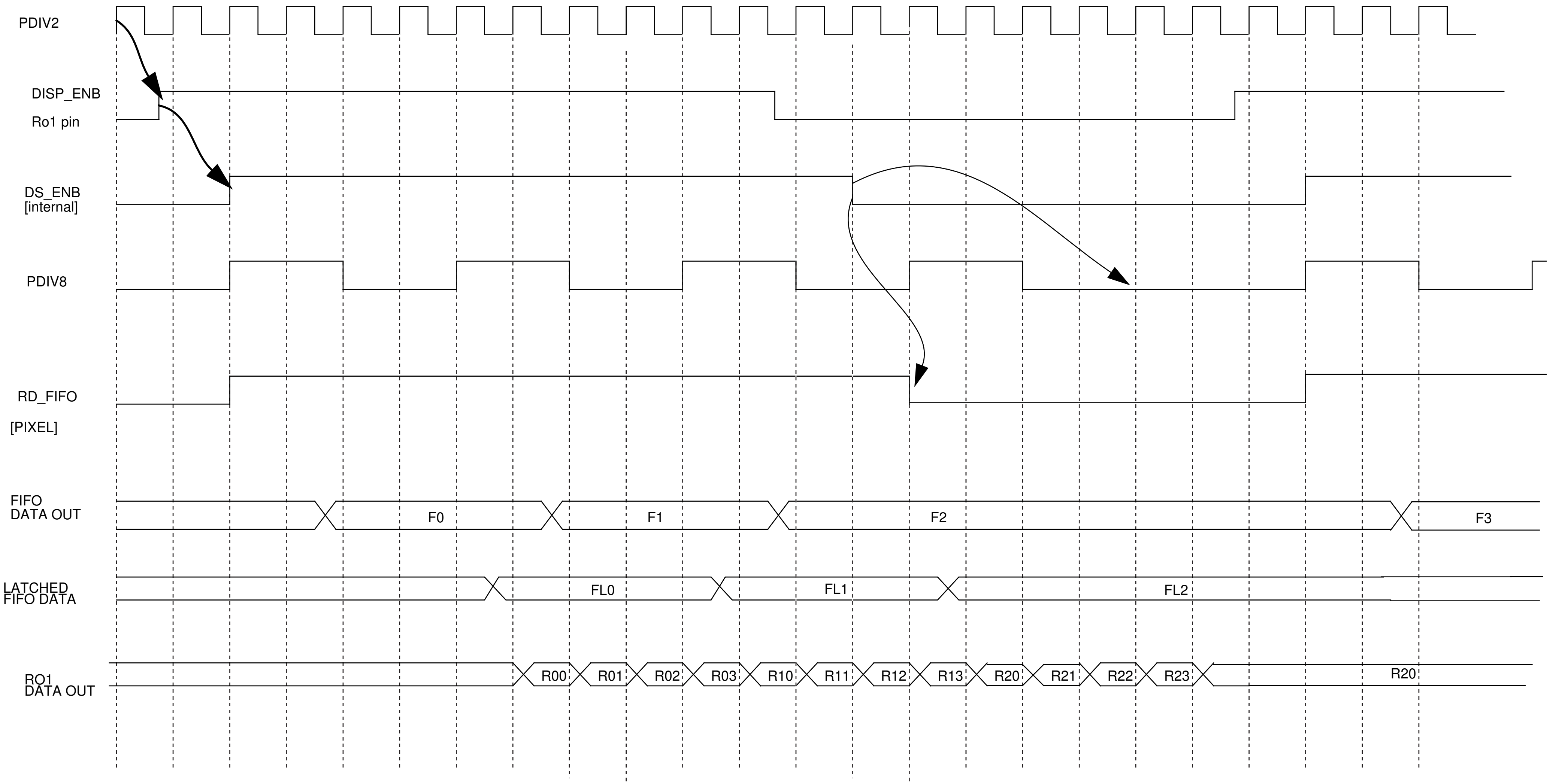
Loading overlay fifo with 4 entries: 4 1/5 dot clocks = 20 dot clocks
Loading pixel fifo with 6 or 7 entries: (6 or 7) 1/5 dot clocks = 30or 35 dot clocks
Need 7 (1/2 dot) clocks for latency (Serial_Enable) = 14 dot clocks
Latency for Display_Enable to read fifo is 2 (1/2 dot) clocks = -4 dot clocks.

Therefore time required between Serial_Enable and Display_Enable is 60 to 65 dot clocks.



SERIAL ENABLE FROM VC2, RO1 LOADS DATA INTO FIFO

STALL FOR ONE PDIV5 CLOCK WHEN 64 PIXELS ARE LOADED



DISPLAY_ENABLE FROM VC2, RO1 OUTPUTS DATA TO XMAP9 AT PIXDIV2 CLOCK