rfFrameBuffer

robfinch<remove>@finitron.ca

Table of Contents

Overview

rfFrameBuffer is a bitmapped display controller circuit supporting multiple display formats. Both the display resolution and color depth may be controlled. The controller acts as a bus master to render a display from memory and as a bus slave to accept display format information from a processing core. The core has pixel plot and pixel fetch capability. The core contains an address mapper allowing non-contiguous memory pages to be used.

The core has an optional internal sync signal generator, if not used then externally generated sync and blanking signals must be supplied.

Features

- controllable horizontal and vertical resolution in terms of video clocks and scanlines.
- scanline buffering
- twelve different pixel encoding formats (6,8,12,16,18,21,24,27, 32, 33, 36 and 40 bits per pixel)
- graphics plane control (z-order)
- 128-bit wide master memory bus
- address mapping, may use up to 64MB for frame memory
- pixels accessed via strips.
- independent video, bus master and bus slave clocks.
- synchronizes to externally supplied horizontal and vertical sync pulses, or may use internal sync
- pixel plot and pixel fetch

Definitions:

ZRGB

The ZRGB format adds bits to the regular RGB format to represent the graphics plane (or the z-order of) the pixel belongs to. The graphics plane is used to determine which pixels appear in the foreground when multiple display devices are competing for the display. For instance, a pixel may appear as part of a background image at plane #15 while sprite data is displayed overtop the background using plane #7. Further imagery may appear in front of the sprite by using a lower numbered plane. The system may use multiple bitmap, sprite or text controllers.

Clocks

The controller uses three independent clocks. These are the video pixel clock, the WISHBONE bus master clock, and the WISHBONE bus slave clock. It is assumed that the slave port will be connected to some sort of processor, and the master port will be connected as a DMA port.

Display Format

The display format is completely programmable. There are register settings that allow the number of horizontal and vertical pixels to be controlled. This controller relies on an external sync generator or the internal sync generator if present. The display generated is relative to the positive edge of the horizontal and vertical synchronization signals. If necessary the position of the display may be altered by adjusting the display window position.

Pixel Plot / Fetch

The controller features pixel plot and pixel fetch capability. Since pixels for some resolutions fit unevenly into a memory strip it can be tricky and time consuming to use a software only solution to pixel plotting and fetching. The core reduces the software overhead involved when displaying a pixel onscreen.

Raster Compare

The controller includes a raster line comparator capable of generating interrupts when the raster scan line matches the value in the compare register.

Interrupts

The controller may generate an interrupt based on a raster line compare. When the raster line matches a value specified in the raster register an interrupt message containing the data specified by the irq_msgdata register is set to the address specified by the irq_msgadr register. The offset of the irq_msgadr register should be configured in the capabilities area of the config space, as part of capability #11h.

Operation

Buffering

The controller ping-pongs between two fifos to buffer information for the display. While a scanline is being displayed from a fifo, data is simultaneously loaded into a second fifo. At low resolutions data for a display line may be fetched across multiple scanlines. Data fetches may occur infrequently which allows the primary cpu to have access to more memory bandwidth.

It is expected that the controller is connected to a system streaming read buffer.

Config Space

A 256-byte config space is supported. The config space is addressed with geographical addressing responding according to the CFG_BUS, CFG_DEVICE, and CFG_FUNC settings in the config address space.

Regno	Width	R/W	Moniker	Description		
000	32	RO	REG_ID	Vendor and device ID		
004	32	R/W				
008	32	RO				
00C	32	R/W				
010	32	R/W	REG_BAR0	Base Address Register		
014	32	R/W	REG_BAR1	Base Address Register		
018	32	R/W	REG_BAR2	Base Address Register		
01C	32	R/W	REG_BAR3	Base Address Register		
020	32	R/W	REG_BAR4	Base Address Register		
024	32	R/W	REG_BAR5	Base Address Register		
028	32	R/W				
02C	32	RO		Subsystem ID		
030	32	R/W		Expansion ROM address		
034	32	RO				
038	32	R/W		Reserved		
03C	32	R/W		Interrupt		
040 to 0FF	32	R/W		Capabilities area		

REG_BAR0 defaults to $FD040001 which is used to specify the address of the controller’s registers in the I/O address space.

The controller will respond with a memory size request of 4MB (0xFFC00000) when BAR0 is written with all ones.

Parameters

CFG_VENDOR_ID	defaults to zero
CFG_DEVICE_ID	defaults to zero
CFG_BUS	defaults to zero
CFG_DEVICE	defaults to zero
CFG_FUNC	defaults to zero

- Config parameters must be set correctly. CFG device and vendors default to zero.

Control Registers:

Regno	Width	R/W	Moniker	Description		
000	64	R/W	REG_CTRL	Master Control Register		
008	64	R/W	REG_REFDLEAY	Reference delay		
010	64	R/W	REG_PAGE1ADDR	Page one memory address		
018	64	R/W	REG_PAGE2ADDR	Page two memory address		
020	64	R/W	REG_PXYZ	pixel x,y,z co-ordinate		
028	64	R/W	REG_PCOLCMD	pixel color and command		
[1]040	64	W	REG_TOTAL	Total horizontal and vertical clocks and scans		
[1]048	64	W	REG_SYNC_ONOFF	vertical and horizontal sync on/off times		
[1]050	64	W	REG_BLANK_ONOFF	vertical and horizontal blank on/off times		
[1]058	64	W	REG_BORDER_ONOFF	vertical and horizontal border on/off times		
060	64	W	REG_RASTER_CMP	raster scanline compare		
068	64	R/W	REG_BITMAP	size of bitmap		
070	64	R/W	REG_OOB_COLOR	out-of-bitmap-bounds color		
078	64	R/W	REG_WINDOW	Window position on bitmap		
080	64	R/W	REG_IRQ_MSGADR	IRQ message address		
088	64	R/W	REG_IRQ_MSGDAT	IRQ message data		
$800 to $9F8	64	R/W	REG_PALETTE1	Color palette used when the color depth is 00 (four or eight bits per pixel).		
$A00 to $BF8	64	R/W	REG_PALETTE2	Second color palette used when the color depth is 00 (four or eight bits per pixel).		

1. Sync generator registers are present only if INTERNAL_SYNC_GEN is defined.

Address Mapping Registers (REG #4000 to 7FF8)

Address mapping registers map a 27-bit frame buffer virtual address to a 64-bit physical address. The maximum memory size supported by the controller is then 128MB. Memory pages are 64kB in size. The lower 16-bits of the frame buffer address are output as the low order 16-bits of the bus master address. The upper 11-bits of the frame buffer virtual address are used as an index into a 2048 entry mapping table. The output of the mapping table is used to drive the upper bits of the physical address. The number of bits for the physical address is specified with the PHYS_ADDR_BITS parameter which defaults to 32.

Regno	Width	R/W	Moniker	Description		
4000	52	R/W		Page #0		
4008	52	R/W		Page #1		
…	52	R/W		Page #2 to 2046		
7FF8	52	R/W		Page #2047		

Master Control Register (REG #00h)

This register contains bits that control the bitmap controller.

<table>
<tr><th>BitNo</th><th></th><th>Description</th><th></th></tr>
<tr><td>**0**</td><td>On/off</td><td>Turns the display controller on=1 or off=0, default is 1</td><td></td></tr>
<tr><td>**11-8**</td><td>Color Depth</td><td>This register identifies the number of bits used per pixel</td><td></td></tr>
<tr><td></td><td></td><td><table>
<tr><td>11-8</td><td>Color Depth</td><td></td></tr>
<tr><td>0000</td><td>6 bits per pixel (ZRBG(3,1,1,1)) (8 color)</td><td></td></tr>
<tr><td>0001</td><td>8 bits per pixel (ZRGB(2,2,2,2) (64 color)</td><td></td></tr>
<tr><td>0010</td><td>12 bits per pixel (ZRGB(3,3,3,3) (512 color)</td><td></td></tr>
<tr><td>0011</td><td>16 bits per pixel (ZRGB(4,4,4,4) (4k color)</td><td></td></tr>
<tr><td>0100</td><td>18 bits per pixel (ZRGB(3,5,5,5) (32k color)</td><td></td></tr>
<tr><td>0101</td><td>21 bits per pixel (ZRBG(3,6,6,6) (256k color)</td><td></td></tr>
<tr><td>0110</td><td>24 bits per pixel (ZRBG(3,7,7,7) (2M color)</td><td></td></tr>
<tr><td>0111</td><td>27 bits per pixel (ZRBG(3,8,8,8) (16M color)</td><td></td></tr>
<tr><td>1000</td><td>32 bits per pixel (ZRGB(5,9,9,9) (128M color)</td><td></td></tr>
<tr><td>1001</td><td>33 bits per pixel (ZRGB(3,10,10,10) (1G color)</td><td></td></tr>
<tr><td>1010</td><td>36 bits per pixel (ZRGB(3,11,11,11) (8G color)</td><td></td></tr>
<tr><td>1011</td><td>40 bits per pixel (ZRGB(4,12,12,12) (64G color)</td><td></td></tr>
</table></td><td></td></tr>
<tr><td>**12**</td><td>greyscale</td><td>This bit enables greyscale mode when the color depth is 6 or 8 bpp.</td><td></td></tr>
<tr><td>**18-16**</td><td>hres</td><td>Horizontal resolution control<table>
<tr><td>18-16</td><td></td><td></td></tr>
<tr><td>000</td><td>Not Supported</td><td></td></tr>
<tr><td>001</td><td>1 video clocks per pixel</td><td></td></tr>
<tr><td>010</td><td>2 video clocks per pixel</td><td></td></tr>
<tr><td>011</td><td>3 video clock per pixel</td><td></td></tr>
</table></td><td></td></tr>
<tr><td>**22-20**</td><td>vres</td><td>Vertical resolution control<table>
<tr><td>22-20</td><td></td><td></td></tr>
<tr><td>001</td><td>1 scanlines per pixel</td><td></td></tr>
<tr><td>010</td><td>2 scanlines per pixel</td><td></td></tr>
<tr><td>100</td><td>4 scanline per pixel</td><td></td></tr>
<tr><td>000</td><td>Not Supported</td><td></td></tr>
</table></td><td></td></tr>
<tr><td>**24**</td><td>Page</td><td>This bit controls which memory page address is used. Default is 0.</td><td></td></tr>
<tr><td>**28-25**</td><td>Pals</td><td>This bit controls which palette is in use. Default is 0.</td><td></td></tr>
<tr><td>**59-48**</td><td>Map</td><td>Period of memory requests in bus master clock cycles (default 0)</td><td></td></tr>
</table>

Map

This register allows control over when a pixel strip is requested from memory. It may be used to allow other devices to access memory in between the read of pixel strips. Normally the controller requests one strip after another in a continuous fashion until the number of strips required for the scan-line is met. Setting this register can be used to create space between the accesses. The access period should be set short enough to allow the controller to read all strips before they are required or display problems may occur.

Example:

Using 8 bits per pixel and horizontal resolution of divide by two (683 pixels per line). There are 16 pixels in a 128-bit strip. So, 43 strips must be read from memory during the scanline. Assume there are 1575 memory bus clock cycles per scan line. Then the average rate a pixel strip must be read is 1575 / 43 = 36.9 clocks. Rather than set the period to 36 it’s better to round down a bit so a value of 32 is used. Setting this value would allow other devices to access memory in between the pixel strip reads.

Ref Delay (REG #08h)

Bits		
15 to 0	left	Horizontal reference delay (default 218)
31 to 16	top	Vertical reference delay (default 27)
47 to 32		reserved
63 to 48		reserved

The reference delay registers may be used to control the position of the bitmap on the screen. The horizontal reference delay is relative to the rising edge of the horizontal sync pulse. The vertical reference delay is relative to the rising edge of the vertical sync pulse. Typically, a small negative value is required for these registers to generate a proper display.

Page One Address (REG #10h)

Bits		
63 to 0	PAGE1ADDR	The memory location of the first bitmap page, IO virtual address

Page Two Address (REG #18h)

Bits		
63 to 0	PAGE2ADDR	The memory location of the second bitmap page, IO virtual address

The memory locations of the bitmap pages should be 8 byte aligned.

Pixel X,Y,Z Co-ordinate Register (REG #20h)

Bits	Name	Description
15 to 0	PX	Pixel X Co-ordinate
31 to 16	PY	Pixel Y Co-ordinate
39 to 32	PZ	Pixel Z Co-ordinate

Pixel coordinates allow getting or setting a pixel within the bitmap.

Pixel Color / Command Register (REG #28h)

Bits	Name	Description
1 to 0	PCMD	Pixel command 00 = no command / not busy 01 = fetch pixel color 10 = plot pixel 11 = not used
11 to 8	ROP	Raster Operation 0000 = black – set pixel to black; ignores color register 0001 = copy – set pixel to color register value 0010 = invert – invert pixel color bits; ignores color register 0100 = and – perform bitwise and of target pixel and color 0101 = or 0110 = xor 1111 = white – set pixel to white; ignores color register
55 to 16	COLOR	Pixel Color

The pixel command register is used to plot or fetch pixels to/from memory. To plot a pixel first set the pixel co-ordinates in the PX, PY, PZ registers and the pixel color register (REG #28h). Then plot command bits are set in this register. In order to fetch a pixel set the co-ordinates and fetch command in this register, then read the color register. After a plot or fetch command is issued the register should be polled to ensure that the command has had time to complete. The command bits will read back as 00 if the command has completed. The controller waits until there is an opportunity to perform the command during the scan-line fetch process. The pixel plot operation is performed according to the specified raster operation. The raster operation bits are write-only and read back as zero.

Only as many bits as required to represent the color for a given color depth need to be used in this register. For example, if the color depth is eight bits per pixel only the least significant eight bits of the register should be set.

Palette Registers (REG $800 to $BF8)

The palette registers map a six or eight-bit color code from memory into a 32-bit RGB (5,9,9,9) value. Note that at six bits per pixel only eight colors are available as the most significant bit is used to indicate z-order. Similarly, for eight bits per pixel only 64 colors are available as the top two bits of the value indicate the z-order. There are two color palettes available, which palette is in use is controlled by the pals bit in CTRL.

Horizontal and Vertical Total (REG $040)

These registers may be locked.

Bits		
11 to 0	hTotal	The total number of dot clocks in the horizontal scan line. - default 1056
27 to 16	vTotal	The total number of scan lines in the display – default 628
63 to 32	sgLock	Write code $A1234567 to unlock sync generator registers Write code $7654321A to lock sync generator registers On reset the sync generator registers are locked.

Sync On / Off (REG $048)

These registers may be locked.

Bits		
11 to 0	hSyncOff	The horizontal count at which hSync should be turned off – default 168
27 to 16	hSyncOn	The horizontal count at which hSync should be turned on – default 40
43 to 32	vSyncOff	The scan line count at which vSync should be turned off – default 5
59 to 48	vSyncOn	The scan line count at which vSync should be turned on – default 1

Blank On / Off (REG $050)

These registers may be locked.

Bits		
11 to 0	hBlankOff	The horizontal count at which hBlank should be turned off – default 252
27 to 16	hBlankOn	The horizontal count at which hBlank should be turned on – default 1052
43 to 32	vBlankOff	The scan line count at which vBlank should be turned off – default 28
59 to 48	vBlankOn	The scan line count at which vBlank should be turned on – default 628

Border On / Off (REG $058)

These registers are not subject to locking.

Bits		
11 to 0	hBorderOff	The horizontal count at which hBorder should be turned off – default 256
27 to 16	hBorderOn	The horizontal count at which hBorder should be turned on – default 1056
43 to 32	vBorderOff	The scan line count at which vBorder should be turned off – default 28
59 to 48	vBorderOn	The scan line count at which vBorder should be turned on – default 628

Raster Compare (REG $060)

These registers are not subject to locking.

Bits		
11 to 0	vertical count	The vertical counter value at which a scan-line interrupt should occur.
62 to 11	reserved	not used
63	clear irq	Set this bit to clear an outstanding interrupt. Bit automatically clears.

The vertical count may be set to a value greater than the number of scan-lines produced by the sync generator, in which case the raster compare will always be false. This may be used to disable the compare.

Bitmap Size (REG $068)

Bits		
15 to 0	width	The width of the bitmap in pixels.
47 to 32	height	The height of the bitmap in pixels.

The bitmap size is independent of the display window.

OOB COLOR (REG $070) – Out-of-Bounds Color

Bits		
39 to 0	color	The color displayed when the window area is out of the bounds of the bitmap. Only as many bits as needed for the color depth are used.
63 to 40	reserved	

Window (REG #78h)

Bits		
11 to 0	Width	The number of pixels displayed horizontally on screen (default 400)
27 to 16	Height	The number of pixels displayed vertically on screen (default 300)
47 to 32	Left	First pixel displayed from bitmap relative to left edge (default 0)
63 to 48	Top	First pixel displayed from bitmap relative to top edge (default 0)

The window control register determines what part of the bitmap appears on-screen. The bitmap may be much larger than, or smaller than the window display area.

The number of pixels displayed depends on both the horizontal resolution setting and the video mode used. For example, if a 1366x768 display mode is used and the horizontal resolution is set to divide by four, then the horizontal setting of this register should be set to 340. (1366 / 4 rounded).

IRQ Message Address (REG $080)

Bits		
63 to 0	msg address	Address to write interrupt message to.

IRQ Message Data (REG $088)

Bits		
63 to 0	msg data	Data to write to interrupt message address.

Port Signals

The slave port may be either 32 or 64-bit.

Name	Width	I/O	
rst_i	1	i	This active high signal resets the core and WISHBONE bus interfaces
s_clk_i	1	i	Clock signal for slave peripheral interface
cs_config_i	1	i	circuit select for configuration space
cs_io_i	1	i	circuit select for io address space
s_cyc_i	1	i	cycle is valid
s_stb_i	1	i	data transfer in progress
s_ack_o	1	o	data transfer acknowledge
s_we_i	1	i	write enable to register set
s_sel_i	4/8	i	byte lane selects
s_adr_i	32	i	addresses the registers of the core
s_dat_i	32/64	i	data input for registers
s_dat_o	32/64	o	data output of registers
m_clk_i	1	i	clock signal for bus master interface
m_cyc_o	1	o	cycle is valid
m_stb_o	1	o	data transfer is taking place
m_ack_i	1	i	data transfer acknowledge
m_we_o	1	o	write enable
m_sel_o	16	o	byte lane select
m_adr_o	32	o	Memory address for bitmap data read
m_dat_i	128	i	data input from bitmap memory
m_dat_o	128	o	data output to bitmap memory
dot_clk_i	1	i	This is the video clock input
hsync_i	1	i	This is an externally supplied horizontal sync signal
vsync_i	1	i	This is an externally supplied vertical sync signal
blank_i	1	i	video blanking indicator
zrgb_o	40	o	color output video data in ZRGB (4,12,12,12) format
xonoff_i	1	i	externally supplied on/off signal for core

Pixel Layouts in Memory

The bitmap controller reads memory in 128-bit strips on its bus master port. A number of whole pixels are fit into each strip. The number of pixels in a strip does not always work out evenly, in which case there are left over bits in the strip.

The bitmap controller always reads whole 128-bit strips of memory. If the number of strips for a scanline does not work out evenly, a whole strip is still read for the last set of pixels. However only the pixels required to meet the number of pixels on the scan line are displayed. For instance, if the horizontal resolution is 676 pixels, and 8 bpp color depth is chosen then 42.25 strips are needed for the horizontal display. So, 43 strips are read, and only 4 pixels from the last strip are displayed. The display will begin the next scanline with the next whole memory strip.

6 bits per pixel layout = 21 pixels in a 64-bit strip, with two unused bits left over.

0																					127
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	2

8 bits per pixel layout = 16 pixels in a 128-bit strip, with no unused bits left over.

0															127
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

12 bits per pixel layout = 10 pixels in a 128-bit strip, with eight unused bits left over.

0												127
0	1	2	3	4	5	6	7	8	9	10	11	8

16 bits per pixel layout = 8 pixels in a 128-bit strip, with no unused bits left over.

0							127
0	1	2	3	4	5	6	7

32 bits per pixel layout = 4 pixels in a 128-bit strip, with no unused bits left over.

0			127
0	1	2	3

Other layouts follow the same pattern.

ZRGB Formats

6 or 8 bits per pixel

Six and eight bits per pixel modes read the color RGB value from the color palette. Color palette values are stored as 32-bit RGB (10,10,10) values. The most significant three bits of a six-bit value or the two most significant bits of an eight-bit value are passed through to the output as the most significant bits of the ‘Z’ portion of the ZRGB output value.

12 bits per pixel - ZRGB (3,3,3,3)

At 12 bits per pixel color depth only eight graphics planes may be specified. The least significant bit of the plane number is assumed to be zero.

Z (2 to 0)	R (2 to 0)	G (2 to 0)	B (2 to 0)
11 9	8 6	5 3	2 0

16 bits per pixel – ZRGB (4,4,4,4)

Z (3 to 0)	R (3 to 0)	G (3 to 0)	B (3 to 0)
15 12	11 8	7 4	3 0

18 bits per pixel – ZRGB (3,5,5,5)

Z (2 to 0)	R (4 to 0)	G (4 to 0)	B (4 to 0)
17 15	14 10	9 5	4 0

24 bits per pixel – ZRGB (3,7,7,7)

24 bits per pixel represents close to the maximum number of colors the human eye can see.

Z (2 to 0)	R (6 to 0)	G (6 to 0)	B (46 to 0)
23 21	20 14	13 7	6 0

32 bits per pixel – ZRGB (5,9,9,9)

At 32 bits per pixel the least significant bit of the Z component is ignored.

Z (4 to 0)	R (8 to 0)	G (8 to 0)	B (8 to 0)
31 27	26 18	17 9	8 0