rtfBitmapController4

robfinch@finitron.ca

Table of Contents

Overview

rtfBitmapController4 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 in order to render a display from memory and as a bus slave in order to accept display format information from a processing core. The core has pixel plot and pixel fetch capability.

Features

- controllable horizontal and vertical resolution in terms of video clocks and scanlines.
- scanline buffering
- five different pixel encoding formats (8,12,16,24 and 32 bits per pixel)
- 128 bit wide memory bus
- pixels accessed via 128 bit strips.
- independent video, bus master and bus slave clocks.
- synchronizes to externally supplied horizontal and vertical sync pulses.
- pixel plot and pixel fetch

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. 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 reference counts.

Scanline Buffer

The controller features scan-line buffering whereby once a scanline has been fetched, the data doesn’t need to be re-fetched from memory for subsequent scanline displays where the vertical resolution is more than one scan line per pixel. For a given vertical pixel the data is fetched only a single time into the scanline buffer.

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.

Registers:

The controller responds to the word address range: $FFDC5xxx.

Regno	Width	R/W	Moniker	Description
00	32	R/W	REG_CTRL	Master Control Register
04	32	R/W	REG_CTRL2	Secondary control register / Status register
08	12	R/W	REG_HDISPLAYED	Horizontal displayed
0C	12	R/W	REG_VDISPLAYED	Vertical displayed
14	32	R/W	REG_PAGE1ADDR	Page one memory address
18	32	R/W	REG_PAGE2ADDR	Page two memory address
1C	32	R/W	REG_REFDELAY	sync reference delay register
20	12	R/W	REG_MAP	memory access period
24	12	R/W	REG_PX	pixel x co-ordinate
28	12	R/W	REG_PY	pixel y co-ordinate
2C	32	R/W	REG_COLOR	pixel color
30	32	*R/W	REG_PCMD	pixel command *only the low 2 bits may be read
$800 to $BFF	32	R/W	REG_PALETTE1	Color palette used when the color depth is 00 (eight bits per pixel).
$C00 to $FFF	32	R/W	REG_PALETTE2	Second color palette used when the color depth is 00 (eight bits per pixel).

Master Control Register (REG #00)

This register contains bits that control the bitmap controller.

BitNo		Description		
0	On/off	Turns the display controller on=1 or off=0, default is 1		
10-8	Color Depth	This register identifies the number of bits used per pixel		
		10-8	Color Depth	
		000	not used	
		001	8 bits per pixel (default)	
		010	12 bits per pixel	
		011	16 bits per pixel	
		100	24 bits per pixel	
		101	32 bits per pixel	
		110	not used	
		111	not used	
11	greyscale	This bit enables greyscale mode when the color depth is 8 bpp.		
18-16	hres	Horizontal resolution control		
		18-16		
		000	Not Supported	
		001	1 video clocks per pixel	
		010	2 video clocks per pixel	
		011	3 video clock per pixel	
21-19	vres	Vertical resolution control		
		21-19		
		001	1 scanlines per pixel	
		010	2 scanlines per pixel	
		100	4 scanline per pixel	
		000	Not Supported	

Control Register 2 / Status Register (REG #04)

Bitno		
16	Page	This bit controls which memory page address is used. Default is 0.
17	Pals	This bit controls which palette is in use (0 or 1). Default is 0.

HDisplayed (REG #08)

Bits		
11 to 0	HDISP	The number of pixel displayed horizontally on screen

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 this register should be set to 340. (1366 / 4 rounded).

VDisplayed (REG #0C)

Bits		
11 to 0	VDISP	The number of pixel displayed vertically on screen

The number of pixels displayed depends on both the vertical resolution setting and the video mode used. For example, if a 1366x768 display mode is used and the vertical resolution is set to divide by four, then this register should be set to 192. (768 / 4 rounded).

Page One Address (REG #14)

Bits		
31 to 0	PAGE1ADDR	The memory location of the first bitmap page

Page Two Address (REG #18)

Bits		
31 to 0	PAGE2ADDR	The memory location of the second bitmap page

The memory locations of the bitmap pages should be 16 byte aligned.

Reference Delay Register (REG #1C)

Bits	Name	Description
11 to 0	HRefDelay	Horizontal reference delay (default 218)
27 to 16	VRefDelay	Vertical reference delay (default 27)

The reference delay register 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.

Memory Access Period (REG #20)

Bits		
11 to 0	MAP	Period of memory requests in bus master clock cycles (default 0)

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 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 = 36.6 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.

Pixel X Co-ordinate Register (REG #24)

Bits	Name	Description
11 to 0	PX	Pixel X Co-ordinate

Pixel Y Co-ordinate Register (REG #28)

Bits	Name	Description
11 to 0	PY	Pixel Y Co-ordinate

Pixel Color Register (REG #2C)

Bits	Name	Description
31 to 0	COLOR	Pixel Color

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.

Pixel Command Register (REG #30)

Bits	Name	Description
1 to 0	PCMD	Pixel command 00 = no command / not busy 01 = fetch pixel color 10 = plot pixel 11 = not used
19 to 16	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

The pixel command register is used to plot or fetch pixels to/from memory. In order to plot a pixel first set the pixel co-ordinates in the PX, PY registers and the pixel color register (REG #28). 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.

Palette Registers (REG $800 to $FFF)

The palette registers map an eight bit color code from memory into a 24 bit RGB (8,8,8) value. There are two color palettes available, which palette is in use is controlled by the pals bit in CTRL2. Note the palette may also be used as a scratchpad memory if not otherwise in use.

Port Signals

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
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_adr_i	32	i	addresses the registers of the core
s_dat_i	32	i	data input for registers
s_dat_o	32	o	data output of registers
m_clk_i	1	i	clock signal for bus master interface
m_bte_o	2	o	Burst type (always 00 for linear burst)
m_cti_o	3	o	Cycle type indicator
m_bl_o	6	o	Burst length (burst length is 8 words)
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_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
vclk	1	i	This is the video clock input
hSync	1	i	This is an externally supplied horizontal sync signal
vSync	1	i	This is an externally supplied vertical sync signal
blank	1	i	video blanking indicator
rgbo	24	o	color output video data in RGB (8,8,8) format
xonoff	1	i	externally supplied on/off signal for core

All bus transfers are 32 bits. The low order two address bits should be fixed at zero.

Pixel Layouts in Memory

The bitmap controller4 reads memory in 128 bit strips. 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 680 pixels, and 6 bpp color depth is chosen then 32.38 strips are needed for the horizontal display. So 33 strips are read, and only 8 pixels from the last strip are displayed. The display will begin the next scanline with the next whole memory strip.

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

0															128
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

12 bits per pixel layout = 10 pixels in an 128 bit strip, with 8 unused bits left over.

0									119	
0	1	2	3	4	5	6	7	8	9	~

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

0							127
0	1	2	3	4	5	6	7

24 bits per pixel layout = 5 pixels in an 128 bit strip, with 8 unused bits left over.

0				119	
0	1	2	3	4	~

32 bits per pixel layout = 4 pixels in an 128 bit strip, with the most significant eight bits of each pixel unused..

0			127
0	1	2	3

WISHBONE Compatibility Datasheet

The rtfBitmapController core may be directly interfaced to a WISHBONE compatible bus.

WISHBONE Datasheet WISHBONE SoC Architecture Specification, Revision B.3		
Description:	Specifications:	
General Description:	Bitmap controller	
Supported Cycles:	SLAVE, READ / WRITE SLAVE, BLOCK READ / WRITE SLAVE, RMW	
Data port, size:	32 bit	
Data port, granularity:	32 bit	
Data port, maximum operand size:	32 bit	
Data transfer ordering:	Little Endian	
Data transfer sequencing	any (undefined)	
Clock frequency constraints:		
Supported signal list and cross reference to equivalent WISHBONE signals	Signal Name:	WISHBONE Equiv.
	rst_i	RST_I
	S_ack_o	ACK_O
	S_adr_i(33:0)	ADR_I()
	S_clk_i	CLK_I
	S_dat_i(31:0)	DAT_I()
	S_dat_o(31:0)	DAT_O()

	S_cyc_i	CYC_I
	S_stb_i	STB_I
	S_we_i	WE_I
	ack_i	ACK_I
	adr_o(33:0)	ADR_O
	clk_i	CLK_I
	dat_i(31:0)	DAT_I
	dat_o(31:0)	DAT_O
	cyc_o	CYC_O
	stb_o	STB_O
	we_o	WE_O
	bte_o	BTE_O
	cti_o	CTI_O
Special Requirements:		