Morphologic
Embedded FPGA Solutions
www.morphologic.dk

XPS Video and LCD Controller (V2.00b)

Introduction

The XPS_LCD_CTRL_V2_00_a IP core is a video and LCD controller that interfaces to a PLB V4.6 bus for reading video data and for register access. The core can control both passive and active (TFT) LCD displays as well as provide video data and timing for VGA and DVI interfaced displays. The IP core is intended for Xilinx based embedded FPGA designs.

Features

- Supports PLBv4.6 bus
- PLB bus master port width of 32, 64 or 128 bits
- Selectable pixel FIFO depths of 256, 512 or 1024 words.
- Controls both passive and active LCD displays
- Passive displays can have 4096 colours using random frame rate modulation.
- Controls VGA monitors
- Includes optional PWM and GPIO channels for back light, contrast and display on/off control
- Frame synchronous base address reload to allow gaming displays.
- Frame interrupt signal for software synchronisation.
- Supports from 1 to 32 bits per pixel resolution
- Supports high-resolution display modes
- Separate pixel and system bus clocks
- Configurable small footprint core

Core short specs.	
Supported Device Family	Spartan-3 Spartan-6 Virtex-4 Virtex-5 Virtex-6
Core Version	V2_00_b
Resource utilization	
Slices	
Block RAMs	
Special Features	none
Core deliveries	
Documentation	Data Sheet
Design Files	VHDL & Net list
Constraints File	N/A
Verification	On request
Instantiation Template	N/A
Ref. Design & Application Notes	N/A
Design Tool Requirements	
Xilinx Implementation Tools	ISE® 11.2 or later
Verification	Xilinx BFL toolset
Simulation	Modelsim SE/PE 6.5 or later
Synthesis	XST
Support	
Provided by Morphologic ApS	

1. Functional Description

1.1 Overview

The video controller enables the display of computer graphics on monitors, LCD panels and with the right video timing,- even television sets. Display data (pixels) are fetched from a PLB slave (typically a memory controller) , pushed into a FIFO and then read out from the FIFO and sliced up into the configured number of bits per pixel. The data for each pixel is then sent out on the output port one pixel at a time at the pixel clock rate. For passive LCD display modes, Frame Rate Modulation (FRM) is used to generate up to 16 shades of colours/white and for these modes a pixel clock is needed that is 16 times the LCD pixel clock rate. For active LCD displays and LCD monitors, this FRM is built into the screen logic and the display only needs N bits per colour to indicate which shade is displayed on the screen.

The video controller also generates the horizontal and vertical synchronization signals/clock needed for the display. The video timing is software programmable to suit most display and to be able to change screen size/position. The number of bits per pixel and video mode is however only configurable using generic parameters used at synthesis time. This keeps the size of the core footprint down to a minimum.

The parallel video data can be sent to:

- Directly to a LCD display module with parallel interface.
- An internal serialization core with LVDS output (for LCD display modules)
- An external video DAC chip with parallel inputs (for analog computer monitors)
- An external DVI chip via a DDR interface helper core (for computer monitors with DVI input)

Display data formats include, 8 bit packed formats used in passive LCD modules, where typically 4 pixels are packed in 3 consecutive bytes (colour modules) or 8 pixels in one byte (monochrome modules). FRM modes exist where the generated colour components are modulated to generate 16 shades of each basic colour, resulting in a total of 4096 colours.

Display timing is software programmable with a resolution of 8 pixel clocks (8x8 clocks for modes using FRM) allowing for very flexible timing to suit various display requirements and timing standards such as defined by VESA.

To be able to completely control LCD displays, the core includes up to two optional PWM generators and 3 parallel output bits. These can be used to control back light brightness, display contrast and to turn on/off display, back light and general display power.

An optional 256x18 bit palette RAM can be included, supported , which provides 17 bit 8-bit per pixel colour mode can be selected, where a 256x18 bit palette RAM is used (uses an 18Kbit BRAM).

Video data is read using a PLB bus master module, and into an asynchronous FIFO. The native master bus width and FIFO depth can be configured as required to avoid FIFO under-runs. It is recommended that the bus master port is attached to it's own PLB bus on (typically a MPMC core port).

The figure on the next page, is a block diagram of the video controller IP core together with it's interface signals. The major functional blocks are described in the subsequent sections:

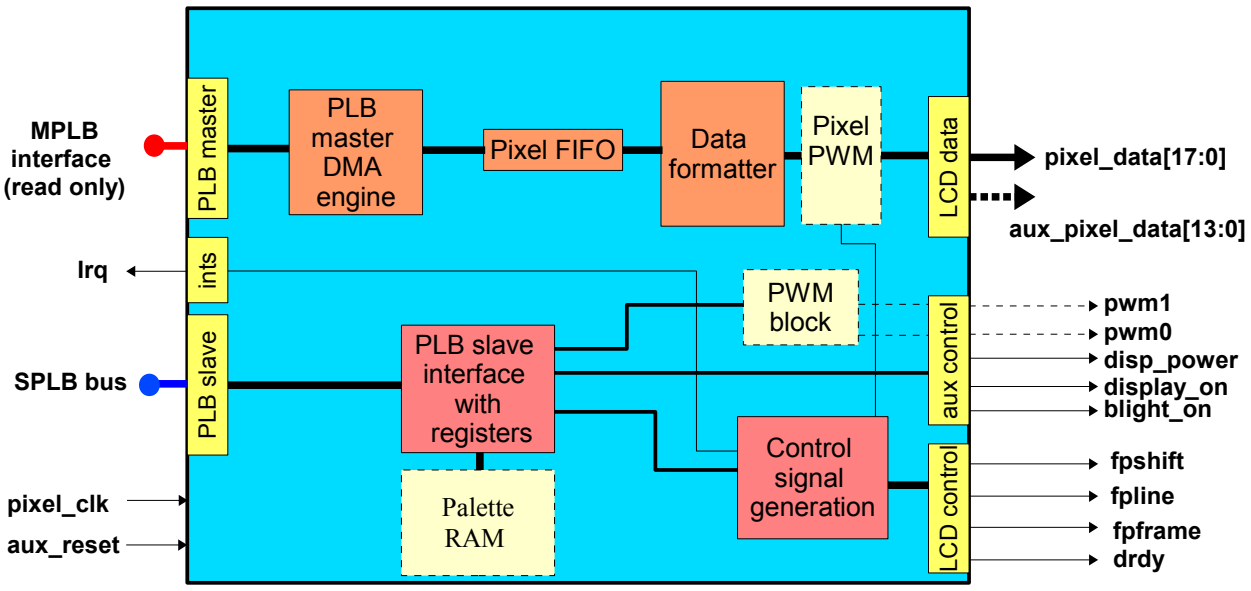

Fig 1. XPS_VIDEO_CTRL IP block diagram. Dotted lines means optional elements.

1.2 PLB Master DMA engine.

The PLB Master DMA engine bus performs periodic burst reads to fetch video data. The native data width of the port can be configured to be 32 or 64 bits. Data is read into an asynchronous FIFO which can be configured to be 512 or 1024 words (32 or 64 bit) wide. The FIFO is initially filled up before readout starts. The bus master requests 128 byte-burst if possible or else 32 byte bursts are used. The PLB master port must use the same clock as the bus slave port, but can be completely asynchronous to the pixel clock. The base address of the display memory must be a multiple of 32 bytes. This is ensured by ignoring the lower 5 bits of the base address register and assuming them to be zero.

1.3 PLB Slave Interface

The PLB slave uses a native 32-bit bus format, but will attach to 32, 64 or 128 bit buses. A block of 8 configuration registers are available through the interface. To simplify logic, the PLB slave only supports 32-bit read and writes. The PLB slave must use the same clock as the bus master port, but can be completely asynchronous to the pixel clock.

1.4 Palette RAM (option)

The optional 256x18 bit palette RAM is accessed via indirect register access. A single 18 Kbit block RAM is used, which allows for 4 of colour sets which can be switched between using the control registers.

1.5 Control Signal Generation

The control signal module contains various counters to generate the programmed timing selected by the configuration registers. The module interfaces to the data formatter to ensure the right pixel data appears at the right time.

1.6 Pixel FIFO

The pixel FIFO acts as a buffer between the PLB bus and the video formatter. It allows for activity on the PLB bus where the mater port is attached, without causing an under-run condition on the FIFO. To allow for various the bus sizes and pixel rates/bus loads, the FIFO size can be configured to be 512 or 1024 32-bit

word and the write port size (native master bus width) can be selected between 32 or 64 bits.

1.7 Data Formatter

The data formatter performs multiplexing of the individual bits of each 32-bit word read out from the pixel FIFO. If for example 1 bit/per pixel mode is selected, each bit is multiplexed into one single pixel bit for each clock cycle of the pixel clock, while for 16 bits/pixel modes, a 32-bit word is multiplexed into two successive pixels etc.

1.8 Pixel PWM (optional)

The optional pixel PWM module (used with passive display modes other than mode 0 and 1), toggles each pixel in the output sequence on and off, controlled by the pixel value (0..15) and the output sequence. The output sequence is a repetitive but pseudo random number to minimise the amount of visible flicker when pixels of the same value is shown next to each other.

The toggle sequence is stored in a look-up table which can be configured as synthesis time to support displays with different colour dynamics. To avoid having 8 look-up tables, the pixel clock must be eight times the byte clock rate, so that each pixel can be looked up in a sequential manner.

The look-up table is a 256-bit ROM, partitioned into 16 32-bit long PWM sequences. The contents of this ROM can be set via the generic C_FRM_ROM. Only half of the levels are specified as the other half is obtained by inverting the pattern. The default sequence is show in the table below:

Shade #	*msb* *PWM sequence* *lsb*
0 / 15	00000000000000000000000000000000
1 / 14	00000000000000010000000000000010
2 / 13	00000000100000010000000100000010
3 / 12	00000100001000010000100001000010
4 / 11	01001000100100010001000100100010
5 / 10	10010010010010010010010010010010
6 / 9	10010100101001010010100101001010
7 / 8	10101010010101010101010010101010

1.9 PWM Block (optional)

The optional PWM block, consists of one or two 8-bit PWM generators. The output of these can be used to control back-light intensity and contrast control of LCD modules.

2. Core Interface Signals

2.1 Xilinx v4.6 PLB Slave Bus Signals

Signal Name	*Direction*	*Type*	*Description*
SPLB_Clk	I	std_logic	PLB main bus clock
SPLB_Rst	I	std_logic	PLB main bus reset
PLB_ABus	I	std_logic_vector(0 to 31)	PLB address bus
PLB_UABus	I	std_logic_vector(0 to 31)	PLB upper address bus
PLB_PAValid	I	std_logic	PLB primary address valid indicator
PLB_SAValid	I	std_logic	PLB secondary address valid indicator
PLB_rdPrim	I	std_logic	PLB secondary to primary read request indicator
PLB_wrPrim	I	std_logic	PLB secondary to primary write request indicator
PLB_masterID	I	std_logic_vector(0 to C_SPLB_MID_WIDTH-1)	PLB current master identifier
PLB_abort	I	std_logic	PLB abort request indicator
PLB_busLock	I	std_logic	PLB bus lock
PLB_RNW	I	std_logic	PLB read/not write
PLB_BE	I	std_logic_vector(0 to C_SPLB_DWIDTH/8-1)	PLB byte enables
PLB_MSize	I	std_logic_vector(0 to 1)	PLB master data bus size
PLB_size	I	std_logic_vector(0 to 3)	PLB transfer size
PLB_type	I	std_logic_vector(0 to 2)	PLB transfer type
PLB_lockErr	I	std_logic	PLB lock error indicator
PLB_wrDBus	I	std_logic_vector(0 to C_SPLB_DWIDTH-1)	PLB write data bus
PLB_wrBurst	I	std_logic	PLB burst write transfer indicator
PLB_rdBurst	I	std_logic	PLB burst read transfer indicator
PLB_wrPendReq	I	std_logic	PLB write pending bus request indicator
PLB_rdPendReq	I	std_logic	PLB read pending bus request indicator
PLB_wrPendPri	I	std_logic_vector(0 to 1)	PLB write pending request priority
PLB_rdPendPri	I	std_logic_vector(0 to 1)	PLB read pending request priority
PLB_reqPri	I	std_logic_vector(0 to 1)	PLB current request priority
PLB_TAttribute	I	std_logic_vector(0 to 15)	PLB transfer attribute
Sl_addrAck	O	std_logic	Slave address acknowledge
Sl_SSize	O	std_logic_vector(0 to 1)	Slave data bus size

Sl_wait	O	std_logic	Slave wait indicator
Sl_rearbitrate	O	std_logic	Slave re-arbitrate bus indicator
Sl_wrDAck	O	std_logic	Slave write data acknowledge
Sl_wrComp	O	std_logic	Slave write transfer complete indicator
Sl_wrBTerm	O	std_logic	Slave terminate write burst transfer
Sl_rdDBus	O	std_logic_vector(0 to C_SPLB_DWIDTH-1)	Slave read data bus
Sl_rdWdAddr	O	std_logic_vector(0 to 3)	Slave read word address
Sl_rdDAck	O	std_logic	Slave read data acknowledge
Sl_rdComp	O	std_logic	Slave read transfer complete indicator
Sl_rdBTerm	O	std_logic	Slave terminate read burst transfer
Sl_MBusy	O	std_logic_vector(0 to C_SPLB_NUM_MASTERS -1)	Slave busy indicator
Sl_MWrErr	O	std_logic_vector(0 to C_SPLB_NUM_MASTERS -1)	Slave write error indicator
Sl_MRdErr	O	std_logic_vector(0 to C_SPLB_NUM_MASTERS -1)	Slave read error indicator
Sl_MIRQ	O	std_logic_vector(0 to C_SPLB_NUM_MASTERS -1)	Slave interrupt indicator

2.2 Xilinx v4.6 PLB Master Bus Signals

Signal Name	*Direction*	*Type*	*Description*
MPLB_Clk	I	std_logic	PLB main bus Clock
MPLB_Rst	I	std_logic	PLB main bus Reset
M_request	O	std_logic	Master request
M_priority	O	std_logic_vector(0 to 1)	Master request priority
M_busLock	O	std_logic	Master buslock
M_RNW	O	std_logic	Master read/nor write
M_BE	O	std_logic_vector(0 to C_MPLB_DWIDTH/8-1)	Master byte enables
M_MSize	O	std_logic_vector(0 to 1)	Master data bus size
M_size	O	std_logic_vector(0 to 3)	Master transfer size
M_type	O	std_logic_vector(0 to 2)	Master transfer type
M_TAttribute	O	std_logic_vector(0 to 15)	Master transfer attribute

M_lockErr	O	std_logic	Master lock error indicator
M_abort	O	std_logic	Master abort bus request indicator
M_UABus	O	std_logic_vector(0 to 31)	Master upper address bus (unused)
M_ABus	O	std_logic_vector(0 to 31)	Master address bus
M_wrDBus	O	std_logic_vector(0 to C_MPLB_DWIDTH-1)	Master write data bus
M_wrBurst	O	std_logic	Master burst write transfer indicator
M_rdBurst	O	std_logic	Master burst read transfer indicator
PLB_MAddrAck	I	std_logic	PLB reply to master for address acknowledge
PLB_MSSize	I	std_logic_vector(0 to 1)	PLB reply to master for slave data bus size
PLB_MRearbitrate	I	std_logic	PLB reply to master for bus re-arbitrate indicator
PLB_MTimeout	I	std_logic	PLB reply to master for bus time out indicator
PLB_MBusy	I	std_logic	PLB reply to master for slave busy indicator
PLB_MRdErr	I	std_logic	PLB reply to master for slave read error indicator
PLB_MWrErr	I	std_logic	PLB reply to master for slave write error indicator
PLB_MIRQ	I	std_logic	PLB reply to master for slave interrupt indicator
PLB_MRdDBus	I	std_logic_vector(0 to (C_MPLB_DWIDTH-1))	PLB reply to master for read data bus
PLB_MRdWdAddr	I	std_logic_vector(0 to 3)	PLB reply to master for read word address
PLB_MRdDAck	I	std_logic	PLB reply to master for read data acknowledge
PLB_MRdBTerm	I	std_logic	PLB reply to master for terminate read burst indicator
PLB_MWrDAck	I	std_logic	PLB reply to master for write data acknowledge
PLB_MWrBTerm	I	std_logic	PLB reply to master for terminate write burst indicator

System signals:

Signal Name	*Direction*	*Type*	*Description*
Irq	O	std_logic	Active high interrupt line. Asserted at the end of video memory transfer for a frame.

2.3 Display Interface Signals

Pin name	*Type*	*Description*
pixel_clk	In	This is the pixel clock for active displays. For passive displays, this is the byte clock for mode 0 through 2 and for mode 3 and 4 this clock must be 8 times higher than the byte clock (to allow for sequential FRM table lookup)
pixel_data[31:0]	Out	Pixel data output. The format depends on whether a passive or an active display is selected: Active modes except mode 5: pixel_d[17:12] is the Blue colour level pixel_d[11:6] is the Green colour level pixel_d[6:1] is the Red colour level Active mode 5: pixel_d[23:16] is the Blue colour level pixel_d[15:8] is the Green colour level pixel_d[7:0] is the Red colour level Passive modes, except mode 7: pixel_d[7:0] is used for display data, while the rest is unused and may be left open. Passive modes 7: pixel_d[3:0] is used for display data, while the rest is unused and may be left open.

2.4 LCD control signals

This set of signals are used to control the horisontal and vertical timing of the display.

Pin name	*Type*	*Description*
fpshift	Out	Clock signal for both passive and active displays
fpline	Out	Line latch enable signal for passive displays and hsync signal for active displays.
fpframe	Out	Top scanline reset enable signal for passive displays. Vsync signal for active displays.
drdy	Out	Active high data ready signal for active displays.

2.5 Auxillary control signals

This set of signals are used to control to control display power, bias voltage and backlight on/off. Additionally the PWM outputs (if enabled) can be used to control the display contrast (bias voltage) and backlight intensity. If not enabled via the generics, the signals will be ”low” if connected anyway.

Pin name	*Type*	*Description*
display_on	Out	General purpose output signal than can be used as a display enable signals on passive displays, used in the power-up / down sequence.
disp_power	Out	General purpose output signal that can be used to turn display power supply on and off, used in the power-up / down sequence.
blight_on	Out	General purpose output signal that can be used to turn display backlight on/off.
pwm0	Out	Active high or low pulse width modulated signal for controlling display contrast (on passive displays) or backlight light intensity
pwm1	Out	Active high or low pulse width modulated signal for controlling backlight light intensity or display contrast (on passive displays)

3. Core Parameters

Several generics allow customisation of the IP core to suit the type of display being attached to the IP signals, the pixel bitmap format, as well as the back-light and contrast control circuitry if any. Using generics allows for a smaller footprint core, leaving more room for other functions in the FPGA.

XPS_LCD_CTRL Device Design Parameters

Generic name	*type*	*Default value*	*Description*
C_NUM_PWM	integer	0	0=do not include any PWM channels. 1=include PWM channel 0 2=include PWM channel 0 and 1.
C_PWM0_INV	integer	0	0 = PWM channel 0 output is active high. 1 = PWM channel 0 output is active low.
C_PWM1_INV	Integer	0	0 = PWM channel 1 output is active high. 1 = PWM channel 1 output is active low.
C_ACTIVE	integer	1	0=passive display, 1=active display. For passive displays a 8 or 4 bit data interface is assumed

Generic name	*type*	*Default value*	*Description*
C_BITMAP_MODE	integer	3	Bitmap used: 0=1 bits/pixel mode, MSB is leftmost pixel 1=4 bits/pixel XRGBxrgb format (RGB is left) 2=8 bits/pixel, palletized format 3=8 bits/pixel, RRRGGGBB format 4=16 bits/pixel, RRRRRGGGGGGBBBBB 5=32 bits/pixel (only for active displays), RRRRRRRRGGGGGGGGBBBBBBBB 6=4 bits/pixel monochrome format using 8 bit pixel data interface (only for passive displays) 7=4 bits/pixel monochrome format using 4 bit pixel data interface (only for passive displays) Passive modes 0 and 1 will not use the pixel PWM block so the IP will use less logic resources in this configuration.
C_FRM_ROM [1]	bit_vector	x"aa5554aa94a5294a9249249248911122042108420081010200010002000000000"	256-bit pixel PWM sequence. rightmost bit is at location 0. Contents is partitioned into 8 shades and 32-bit long PWM sequence as follows: A7:A5 = Shade grade 0..7 (8 to 15 are obtained by inverting the address and data bit) A4:A0 = PWM sequence index (32 bit long sequence). When a bit is '1' the pixel is turned on.

1. Only used for passive displays in modes other than 0 or 1.

System and PLB parameters.

Name	*Type*	*Allowable Values*	*Default value*	*Description*
C_BASEADDR	std_logic_vector	Lower 5 bits must be '0'.	None	Slave register base address.
C_HIGHADDR	std_logic_vector	Lower 5 bits must be '1'.	None	Slave register high address.
C_SPLB_AWIDTH	integer	32	32	PLB Address Bus Width
C_SPLB_DWIDTH	integer	32, 64, 128	32	PLB Data Bus Width
C_SPLB_NUM_MASTERS	integer	1-16	8	Number of PLB masters
C_SPLB_MID_WIDTH	integer	log2(C_SPLB_NUM_MASTERS) with a minimum value of 1	1	Width of the PLB_master ID vector
C_SPLB_NATIVE_DWIDTH	integer	32	32	Width of slave data bus
C_SPLB_P2P	integer	0 : Shared bus topology 1 : Point to Point topology	0	PLB Point-to-Point or shared bus topology

C_SPLB_SUPPORT_BURSTS	integer	0:SPLB does not support bursts 1:SPLB supports bursts	0	unused
C_SPLB_SMALLEST_ MASTER	integer	32, 64, 128	32	Width of the smallest master that will be interacting with this slave. Unused
C_SPLB_CLK_PERIOD_PS	Integer	Any positive value	10000	SPLB clock period in picoseconds. Unused
C_INCLUDE_DPHASE_ TIMER	Integer	0:do include dataphase timer 1:include dataphase timer	1	Select whether a dataphase timer in the slave interface is included or not.
C_FAMILY	String	spartan3e, spartan3a, spartan6, virtex4, virtex5, virtex6	virtex5	XILINX FPGA Family
C_MPLB_AWIDTH	Integer	32	32	Width of master address bus
C_MPLB_DWIDTH	Integer	32, 64, 128	32	Width of master data bus
C_MPLB_NATIVE_DWIDTH	Integer	32	32	Native width of IP master bus
C_MPLB_P2P	Integer	0 : Shared bus topology 1 : Point to Point topology	0	Master bus topology
C_MPLB_SMALLEST_SLAVE	Integer	32,64,128	32	Width of the smallest slave attached to the master bus
C_MPLB_CLK_PERIOD_PS	Integer	Any positive value	10000	MPLB clock period in picoseconds

4. Programming model

The following sections describe the programming model of the IP core. First a register map is given followed by instructions on how to configure the IP for operation with passive and active LCD displays.

4.1 Register map

All of the registers listed below must be accessed as 32-bit wide registers. All fields marked as reserved are read as 0 and written bits are ignored.

Register Name	*Offset (hex)*
BASE_ADDR	0
DISPLAY_LENGTH	4
H_PARAMS	8
V_PARAMS	C
CONTROL	10
PWM	14
PALETTE_IDX	18
PALETTE_DATA	1C

4.1.1 BASE_ADDR (offset 0x0)

This register holds the base address of the memory to display. Programming a full-32 bit base address is allowed. Since the reading of data occurs in multiples of 8 words, the lower 5 address bits are ignored. The software should allocate memory in the DDR SDRAM on a 8-word boundary. The register can be programmed any time, but will only take effect at the end of the frame currently being displayed.

BASE_ADDR																																0	0	0	0	0

Name	*Bit*	*Type*	*Reset value*	*Descripton*
BASE_ADDR	[31:5]	R/W	0	Upper 27 bits if the base address of display memory.
Reserved	[5:0]	R	0	Ignored

4.1.2 DISPLAY_LENGTH (offset 0x4)

This register holds the length of the display in bytes. Since the reading of data occurs in multiples of 8 words, the lower 5 bits are ignored.

										21	DISP_LEN	5	0	0	0	0	0

Name	*Bit*	*Type*	*Reset value*	*Descripton*
Reserved	[4:0]	R	0	Reserved
DISP_LEN	[21:5]	R/W	0	Length of display memory in bytes. It must be a multiple of 32 bytes.
Reserved	[31:22]	R	0	Reserved

4.1.3 H_PARAMS (offset 8)

This register holds all parameters related to the horisontal timing.

			28	H_POS	24	23 H_POL			20	H_PW	16				12	H_BLANK	8	7	H_SIZE	0

Name	*Bit*	*Type*	*Reset value*	*Descripton*
H_SIZE	[7:0]	R/W	0	For active displays:Horisontal size of display in 8 pixel units For passive displays: Horisontal size * 3/8 (the number of pixel clocks it takes to shift out a complete scanline). The required value **minus one** must be programmed.
H_BLANK	[12:8]	R/W	0	Horisontal blanking interval in 8 pixel units. The required value **minus one** must be programmed.

Name	*Bit*	*Type*	*Reset value*	*Descripton*
Reserved	[15:13]	R	0	Reserved
H_PW	[20:16]	R/W	0	Horisontal pulse width in 8 pixel units. The required value **minus one** must be programmed.
Reserved	[22:21]	R	0	Reserved
H_POL	[23]	R/W	0	Polarity of fpline '0' is active low for active displays, active high for passive '1' is active high for active displays, active low for passive
H_POS	[28:24]	R/W	0	Horisontal pulse position, in 8 pixel units. The required value **minus one** must be programmed.
Reserved	[31:29]	R	0	Reserved

4.1.4 V_PARAMS (offset 0xC)

This register holds all parameters related to the vertical timing.

		29 V_POS 24	23 V_POL		21 V_PW 16	15 V_BLANK 10	9 V_SIZE 0

Name	*Bit*	*Type*	*Reset value*	*Descripton*
V_SIZE	[9:0]	R/W	0	Vertical size of display in scanline units. The required value **minus one** must be programmed.
V_BLANK	[15:10]	R/W	0	Vertical blanking interval in scanline units. The required value **minus one** must be programmed.
V_PW	[21:16]	R/W	0	Vertical pulse width in scanline units. **The required value minus one must be programmed.**
Reserved	[22]	R	0	Reserved
V_POL	[23]	R/W	0	Polarity of fpframe '0' is active low for active displays, active high for passive '1' is active high for active displays, active low for passive
V_POS	[29:24]	R/W	0	Vertical pulse position, in scan-line units. The required value **minus one** must be programmed.
Reserved	[31:30]	R	0	Reserved

4.1.5 CONTROL (offset 0x10)

This register is used for miscellaneous control functions as well as status.

31	30	29	28	27 26	25	24	23 RESERVED 15	14 13	12	11 8	7 0
CTRL_RUN	CTRL_DP	CTRL_BL	CTRL_DE	CTRL_PS	IRQ_EN	IRQ_CLR		STAT_NUM_PWM	STAT_ACTIVE	STAT_BITMAP_MODE	STAT_VERSION

Name	*Bit*	*Type*	*Reset value*	*Descripton*
STAT_VERSION	[7:0]	R	0x10	Version of IP cores. The version is split up into a major and minor version number: xxxxyyyy where xxxx is the major and yyyy is the minor version. Version 1.0 translates to 00010000
STAT_BITMAP_MODE	[11:8]	R	-	Indicates the value of generic C_BITMAP_MODE, represented by a 4-bit binary number.
STAT_ACTIVE	12	R	-	Indicates the value of generic C_ACTIVE
STAT_NUM_PWM	[14:13]	R	-	Indicates the value of generic C_NUM_PWM represented by a 2-bit binary number
Reserved	[23:15]	R	0	Reserved
IRQ_CLR	24	WO	0	Write '1' to clear the Irq line. Read as zero.
IRQ_EN	25	R/W	0	Enables the interrupt output. The Irq line will the asserted high when the internal base address counter is reloaded. The interrupt is cleared by writing to the ***irq_clr*** bit
CTRL_PS	[26:27]	R/W	0	Palette bank currently used for displaying. The palette RAM is split up into 4 banks of 256 entries each.
CTRL_DE	28	R/W	0	Output state of display_on signal
CTRL_DP	29	R/W	0	Output state of disp_power signal
CTRL_BL	30	R/W	0	Output state of blight_on signal
CTRL_RUN	31	R/W	0	Run enable: 1=enable display refresh. 0=disable display refresh

4.1.6 PWM (offset 0x14)

This register contains the output values for the one or two 8-bit PWM channels:

31 RESERVED 16	15 PWM1 8	7 PWM0 0

The output duty cycle is given as X/256, where X is the programmed value. Note that the active level of the PWM outputs may be selected by the generics C_PWM0_INV and C_PWM1_INV for channel 0 and 1 respectively. When no PWM functions are selected, reading it will return 0 and written bits will be ignored.

Name	*Bit*	*Type*	*Reset value*	*Descripton*
PWM0	[7:0]	R/W	0	PWM0 duty cycle. If not implemented it will act as a reserved field.
PWM1	[15:8]	R/W	0	PWM1 duty cycle. If not implemented it will act as a reserved field.
Reserved	[16:31]	R	0	Reserved

4.1.7 PALETTE_IDX (offset 0x18)

This register contains the current index for the palette memory, implemented when bitmap mode 2 is selected. If not implemented, reading it will return 0 and written bits will be ignored.

31	Reserved	10	9	INDEX	0

Name	*Bit*	*Type*	*Reset value*	*Descripton*
PAL_INDEX	[9:0]	R/W	0	Index into the palette RAM. 1024 locations are writeable. But only 256 are used at a time. The palette select field CTRL_PS1 in the control register, selects between four different banks of palette entries: 00 = bank 0 from index 0 to 255 01 = bank 1 from index 256 to 511 10 = bank 2 from index 512 to 767 11 = bank 3 from index 768 to 1024 Having several banks, allows an unused palette bank to be updated while another is being used, after which PS1 and PS0 are programmed to switch immediately to the new bank. This avoids colour flickering.
Reserved	[31:10]	R	0	Reserved

4.1.8 PALETTE_DATA (offset 0x1C)

This register is used for writing palette data into the palette RAM. A write to this register will write the word into the location currently pointed to by the INDEX field of the PALETTE_IDX register. 1024 locations are available. When reading from the register, the palette location currently pointed to by the INDEX field will be returned. The index will auto-increment to the next location, making it easy to fill up or read out the palette RAM contents sequentially.

31	Reserved	18	17	Blue	12	11	Green	6	5	Red	0

Name	*Bit*	*Type*	*Reset value*	*Descripton*
PAL_RED	[5:0]	R/W	unchanged	Red colour value.

Name	*Bit*	*Type*	*Reset value*	*Descripton*
PAL_GREEN	[11:6]	R/W	unchanged	Green colour value.
PAL_BLUE	[17:12]	R/W	unchanged	Blue colour value.

Note that the contents of the palette RAM is unchanged by a OPB reset, but the initial value is zero.

5. IP core connectivity and programming

The following sections describe how to connect the IP core signals to a passive or active display and how to program the IP's registers.

5.1 Connections to a passive display

The IP core supports single scan passive displays with an 8 bit data interface. The control signals are typically connected as shown in the table below:

IP signal name	*LCD panel name*	*Function*
fpshift	CP or CL2	Pixel clock. Pixel data is output on the rising edge of this clock
fpline	LOAD or CL1	Line latch enable. Data shifted into the shift registers in the display is latched and output to the LCD pixels in the current scanline
fpframe	FRM or FLM	Scanline reset signal, enabled by fpline.
drdy	Not used	
display_on	DISP or /DOFF	LCD bias voltage generator enable
disp_power	-	Usually connected to an external MOSFET switch, which switches the displays power on and off
blight_on	-	Usually connected to an input on the backlight power supply module.
pwm0	Indirectly to VCON	Optionally connected via a RC-circuit (or similar) to the contrast input of the LCD display, controlling the LCD bias voltage.
pwm1	-	Optionally connected to the LCD backlight supply to control it's intensity.
D0..7	D0..7	Parallel data sent in the following format: byte +0: RGBRGBRG, byte +1: BRGBRGBR, byte +2: GBRGBRGB

5.2 Configuring registers for operation with passive LCD displays.

Given a display of size, X by Y, the following values need to be programmed:

Register.field	*Value*
H_PARAMS.H_SIZE	(X * 3 / 8) - 1
H_PARAMS.H_BLANK	2
H_PARAMS.H_PW	0
H_PARAMS.H_POL	0 (typical, results in active high sync pulse)
H_PARAMS.H_POS	0
V_PARAMS.V_SIZE	Y-1
V_PARAMS.V_BLANK	1
V_PARAMS.V_PW	0
V_PARAMS.V_POL	0 (typical, results in active high sync pulse)
V_PARAMS.V_POS	0
DISPLAY_LENGTH	X*Y/8 for mode 0. X*Y/2 for mode 1. X*Y for mode 2 and 3. X*Y*2 for mode 4.
BASE_ADDR	Allocated base address, must be 8-word aligned
CONTROL.CTRL_RUN	1

This will result in the sequence of display signals depicted below (fpframe part shown only).

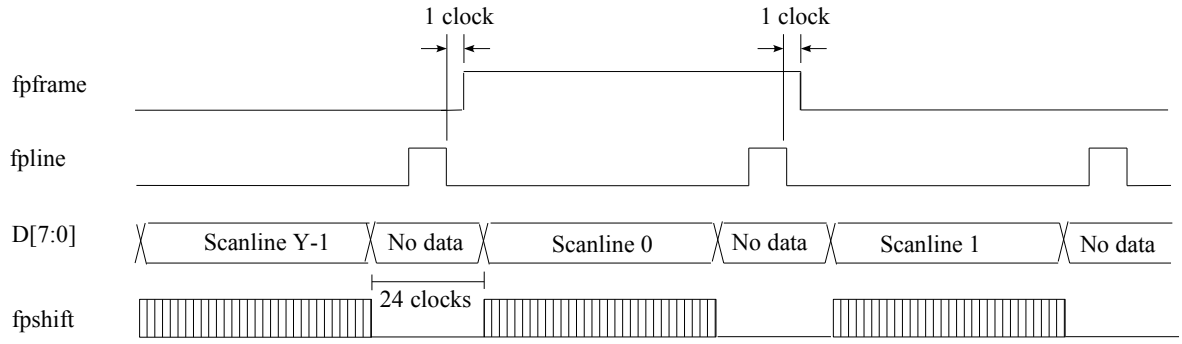

Fig. 2: Passive display timing

All signals are clocked out on the rising edge of fpshift (see active display section for detail).

5.3 Configuring operation with active LCD displays.

Since there can be considerable variation in the timing requirements for active displays, the following information is given as a guideline only for the typical display. Consult the display data sheet for specific requirements of the display.

Given a display with the following timing parameters:

Parameter	***Unit (pixel clocks or lines)***
Horizontal active time	*X* clocks
Total horizontal	*TH* clocks
Horizontal front porch (before hsync)	*HFP* clocks
Horizontal sync pulse width	*HPW*
Vertical active time	*Y* lines
Total vertical	*TV* lines
Vertical front porch (before vsync)	*VFP* lines
Vertical sync pulse width (before vsync)	*VPW* lines

The following values need to be programmed:

Register.field	***Value***
H_PARAMS.H_SIZE	(*X*/8)-1
H_PARAMS.H_BLANK	(*TH*-*X*)/8-1
H_PARAMS.H_PW	(*HPW*/8)-1
H_PARAMS.H_POL	0 (typical, results in active low sync pulse)
H_PARAMS.H_POS	(*HFP*/8)-1
V_PARAMS.V_SIZE	*Y*-1
V_PARAMS.V_BLANK	*TV*-*Y*-1
V_PARAMS.V_PW	*VPW*-1
V_PARAMS.V_POL	0 (typical, results in active low sync pulse)
V_PARAMS.V_POS	*VFP*-1
DISPLAY_LENGTH	X*Y/8 for mode 0. X*Y/2 for mode 1. X*Y for mode 2 and 3. X*Y*2 for mode 4.
BASE_ADDR	Allocated base address, must be 8-word aligned
CONTROL.CTRL_RUN	1

This will result in the sequence of display signals depicted below (example is shown with *Y*=5, *VT*=12, *VFP*=2)

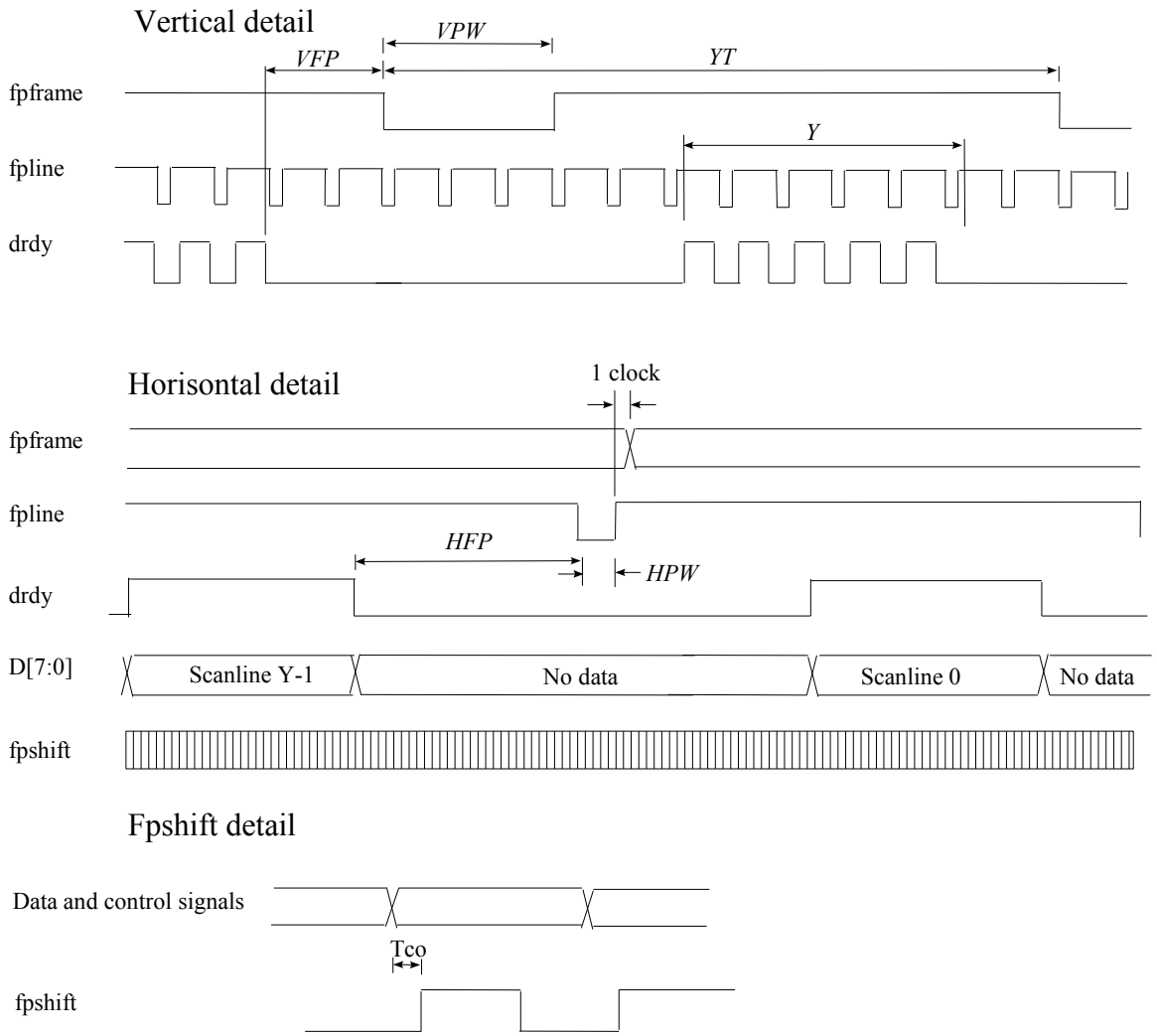

Fig. 3: Active (TFT) display timing

Refer to the FPGA datasheet for values of Tco, which will depend on the clocking strategy and whether the output signals flip-flops are located in the IO blocks or not.

6. Device Utilization and Performance Benchmarks

The resource utilization and performance benchmark table below is a snapshot of reports from actual implementations and gives an estimate of the amount of resources used by the core and the maximum speed obtainable.

Performance and Resource Utilization Benchmarks on the Spartan-3E (xc3s1600efg320-5)

Parameter Values	Device Resources				Bus speed MHz	Pixel clock Fmax MHz
	Slices	Slice FFs	LUTs	BRAM		
C_ACTIVE=1 C_BITMAP_MODE=4 C_MPLB_NATIVE_DWIDTH=32 C_MPLB_P2P=1 C_FIFO_DEPTH => 512	634	751	887	1	66.67	149

Performance and Resource Utilization Benchmarks on the Spartan-6 (xc6slx45tfgg484-3)

Parameter Values	Device Resources			Bus speed MHz	Pixel clock Fmax MHz
	Slices	Slice LUTs	BRAM		
C_ACTIVE=1 C_BITMAP_MODE=4 C_MPLB_NATIVE_DWIDTH=32 C_MPLB_P2P=0 C_FIFO_DEPTH => 1024	723	692	2	100	281

Performance and Resource Utilization Benchmarks on the Virtex-5 (5vfx70tff1136-2)

Parameter Values	Device Resources			Bus speed MHz	Pixel clock Fmax MHz
	Slices	Slice LUTs	BRAM		
C_ACTIVE=1 C_BITMAP_MODE=4 C_MPLB_NATIVE_DWIDTH=64 C_MPLB_P2P=0 C_FIFO_DEPTH => 1024	753	673	2	133.3	296