Graphics Controller Core
- with 2D acceleration functionalities

Product specification

Prevas AB
PO Box 4 (Legeringsgatan 18)
SE-721 03 Västerås, Sweden

Phone: +46 21 360 19 00
Fax: +46 21 360 19 29
Email: johan.ohlsson@prevas.se
URL: www.prevas.com

Features

- Supports any XGA compatible LCD or CRT monitor with analog inputs
- Supports up to 1024x768 pixel resolution at 60Hz refresh rate
- 8/16/24 bit RGB color output
- Interfaces to all standard Video DACs
- Double buffering support (page flipping). Also support high speed copy of data from front to back buffer or wise versa on command
- 2D hardware acceleration functionality
  - Hardware cursor functionality with alpha
  - Line drawing algorithm which draws lines between any given two points at high speed
  - High performance circle drawing algorithm
  - Character printing engine which can draws characters with different sizes. Configurable font table/RAM by host
- Uses external video memory, standard SDRAM interface supported
- Besides host DMA and instruction based access it has separate burst FIFO which utilizes SDRAMs full page burst capabilities. Used for streaming video etc. >30fps achievable at 1024x768 pixel resolution
- Fully synchronous and synthesizable design
- Support a separate host clock domain, host CPU clock frequency can be at any frequency above or equal to 25Mhz
- Altera Avalon slave interface
- Fully embedded in the SOPC Builder environment. Easy SOPC Builder integration, gives rapid and easy integration into Altera-based SoC designs
- Delivered with device drivers and API for Altera Nios II EDS. Designed for minimum RAM usage and for maximum performance
- Simple programming due to small number of control registers.

CORE Facts	
Provided with Core	
Documentation	User's Reference Manual
Design File Formats	VQM netlist; VHDL Source RTL (available at extra cost)
Constraints Files	.qsf
Verification	VHDL test bench
Instantiation templates	VHDL
Reference designs & application notes	HW/SW Demonstartion application
Additional Items	None
Simulation Tool Used	
Modelsim v6.5a	
Support	
Support provided by [Prevas AB]	

Applications

The Prevas Graphics Controller IP core targets 2D applications like:

- Embedded system/process control GUIs
- Other control terminal GUIs
- Security terminals
- Measurement equipment displays
- Medical instruments displays
- Marine navigation systems

General Description

The Graphics Controller is the part of a system that controls the display device, which can for example be a CRT/LCD monitor. It automatically generates the necessary timing signals that are required for the display device. The graphics controller has a separate frame buffer (an external SDRAM Device), in which it stores temporary frames. The controller has a build in SDRAM controller so it automatically handles refresh, reading/writing and initializing the memory device etc. It support linear address mapped direct memory access by the host to/from the frame buffer. It also has a separate instruction based pipeline which uses the HW accelerator implemented in the graphics controller. This HW accelerator is a powerful offload engine that gives embedded systems potent graphics capabilities even if a relatively low performance CPU is employed.

The Prevas Graphics Controller provides a flexible solution, which may be implemented in Altera® Cyclone II/II or Stratix II/III/IV FPGA devices.

Family	Example Device	**Fmax (MHz)** clk_100mhz/ clk_pixel/clk_host	LE/ALM	Total memory bits	IOB	GCLK	Design Tools
Cyclone II	EP2C35F672C6	133/124/420*	2632 (8%)	146304 (30%)	194**	3	Quartus II 8.1
Cyclone III	EP3C16F484C8	116/119/250*	2638 (12%)	146304 (28%)	194**	3	Quartus II 8.1
Stratix III	EP3SL150F720C2	204/182/400*	<1%	146304 (3%)	194**	3	Quartus II 8.1

Table 1: Example implementation statistics

** Note clk_100mhz and clk_pixel should have fixed frequencies. The clk_host clock can be any frequency above or equal to 25 MHz. For more information on this please contact Prevas.*

*** The number of I/Os presented in table above, defines when the controller is not integrated into a system. When integrated the I/O count is max 105, thus all Avalon signals run internally.*

Functional Description

The major blocks that build up the graphics controller are illustrated in the block diagram below. These blocks and there functionality are also described in the upcoming sections.

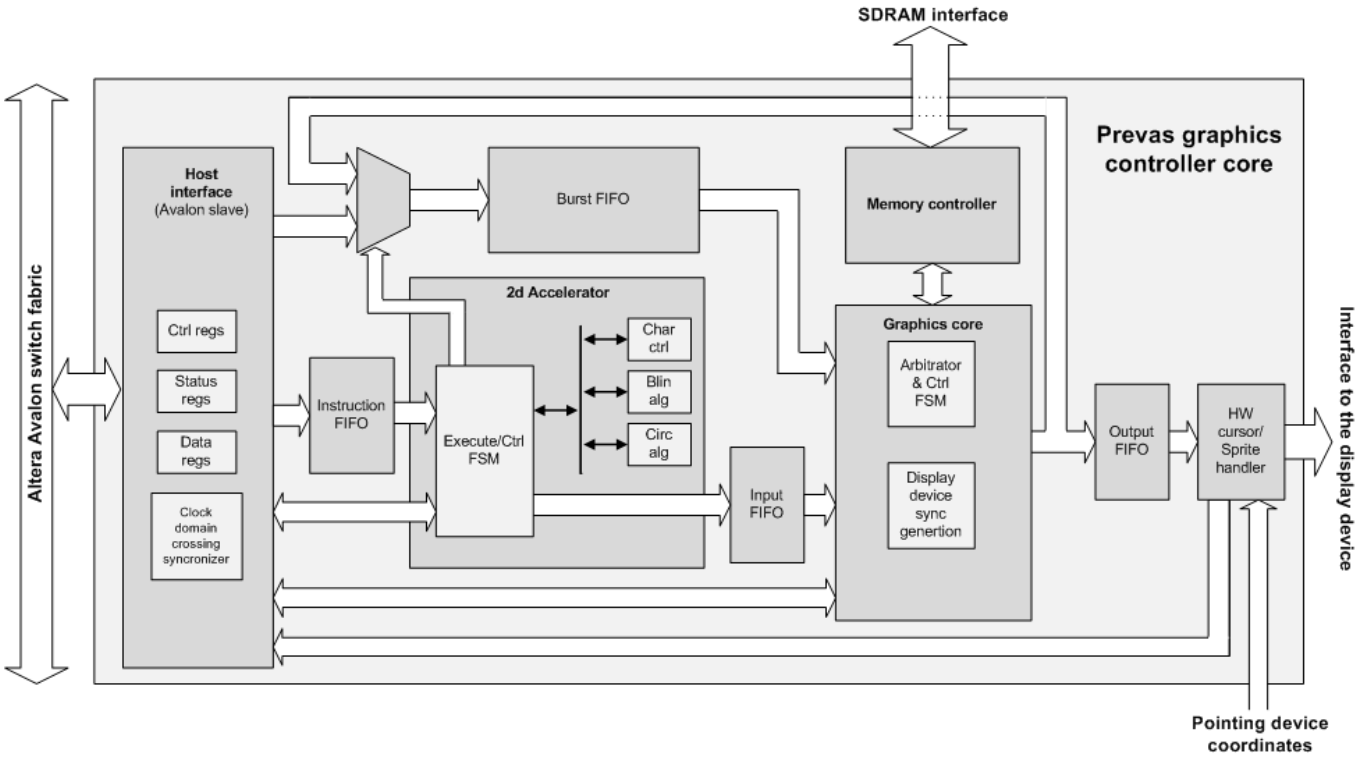

Figure1. Prevas Graphics Controller core block diagram

Host interface (Avalon slave)

This block acts as the interface to the Avalon switch fabric. All internal registers of the core reside in the Host Interface block. The block also provides the rest of the Graphics Controller with control signals. The Avalon interface for the controller employs a single slave port using a small set of Avalon signals, to handle simple and fast read/write transfers to the registers in the block or DMA transfers to/from the frame buffer. The host interface also contains synchronization functionalities when the host clock is not synchronous to the Graphics Controllers 100Mhz core clock.

FIFOs

These FIFOS has different purposes. The Instruction and Input FIFO is mainly used to buffer write command and higher level instructions. These are useful while the memory is busy outputting data to the display device etc. and then the host can still write/give commands to the graphics controller.

The Burst FIFO is used to temporary buffer and balance the hosts irregularly transfer rate. First the host writes to the Burst FIFO then it gives burst command with start address/coordinate and size. The Burst FIFO fits1024 24bits pixels.

The Output FIFO is used as a line buffer and synchronizing stage between clock domains.

2D Accelerator

This module basically reads the Instruction FIFO containing higher level instructions, interpret and brake down these instructions into simple write commands, which it then writes to the Input FIFO. Generally an instruction contains x/y-coordinate for start/end/mid-point(s) and/or radius, character code,

size, color and instruction type identifier. An instruction can for example be to draw a line with a specific color between two points. Other instructions can be to draw circle, specific character, swap frame buffer (page flip), copy data between buffers, and more.

This block also communicates directly with the host controller and bypasses DMA accesses to the memory controller via the graphics core block.

Graphics core

The graphics core handles the generation of synchronization signals to the display device, depending on the selected resolution mode. This module consists among others of horizontal and vertical counters which form the synchronization signals. It also acts an arbitrator and directly communicates with the memory controller. It provides the Output FIFO with RGB data read from the frame buffer. Other tasks for the graphics core can be to load RGB data from the Input FIFO and save this RGB value at the right addresses, another task is to empty the Burst FIFO by burst writing the data to the frame buffer.

Memory controller

The Memory controller handles reading/writing to the frame buffer. It automatically handles the required access related SDRAM commands, refresh commands and mode register access at startup or when switching between burst and single accesses etc. The Memory controller converts the linear address seen by other blocks into SDRAM column and row addressing. Initialization is also handled by this block. Initialization is made after an active system reset. When data is written to memory from the burst FIFO the SDRAMs full page burst capability is used. Keeping track of page boundaries and such is automatically handled by the Memory controller without the need of host intervention.

Different memory configuration can be used. This is set by generic parameters, see the generic parameters section. For example when the *sdram_data_width* parameter is set to 32, then ether two 16bits or a single 32bits SDRAM device can be used.

The Memory controller is designed and optimized for this graphics controller to give maximum performance.

HW cursor/Sprite handler

The hardware cursor block handles the hardware cursor functionality. It overlay the cursor graphics on the “ordinary” graphics layer/plane at the coordinate provided by a pointing device etc. The current location/coordinate of the cursor is also reflected in a read register located in the Host interface block which the host can read. The cursor color and shape are stored in an internal table/memory, which can be modified. Three colors on the cursor are supported. This block can be enabled and disabled by the host in runtime.

Graphics controller generic parameters

As the Prevas Graphics Controller is fully integrated in the SOPC builder environment, these generic parameters are usually set via a GUI inside SOPC Builder when the core is added to the Avalon switch fabric and customized.

Parameter	Description
host_clk_freq_mhz	Indicate the host clock frequency in MHz. Minimum value shall be 25. This parameter is used to select proper and optimal syncronization aproach when the host clock is asynchronous to the 100Mhz Graphics Controller core clock (clk_100mhz).
color_depth	Selects the color depth in bits that should be implemented. Valid values are : ■ Value **8**: result in 8bits RGB output (255 colors) ■ Value **16**: result in 16bits RGB output (65k colors) ■ Value **24**: result in 24bits RGB output (~17M colors)
resolution_sel	Selects the display resolution that shall be implemented. Valid values are : ■ Value **1**: result in 640x480 pixels per frame (VGA) (also requires that the clk_pixel is set to 25Mhz) ■ Value **2**: result in 800x600 pixels per frame (SVGA) (also requires that the clk_pixel is set to 40Mhz) ■ Value **3**: result in 1024x768 pixels per frame (XGA) (also requires that the clk_pixel is set to 40Mhz)
sdram_data_bits	Indicates the SDRAM data width. Valid values are **8**, **16** or **32**
sdram_addr_bits	Indicates the SDRAM address width
sdram_row_width	Indicates the row address size in bits
sdram_column_width	Indicates the column address size in bits

Table 2: Prevas Graphics Controller core generic parameters

Graphics controller core I/O signals

Signal	Dir.	Description
System signals		
reset_n	I	Asynchronous global active low reset (initiates all register in the design)
Avalon switch fabric slave signals		
clk_host	I	Host clock
avs_chipselect_n	I	Avalon active low chip select
avs_write_n	I	Avalon active low write enable signal
avs_read_n	I	Avalon active low read enable signal
avs_waitrequest_n	O	Avalon active low wait request. Forces wait states
avs_writedata[31..0]	I	Avalon write data signal
avs_readdata[31..0]	O	Avalon read data signal
avs_address[20..0]	I	Avalon address signal
Graphics Controller core signals		
clk_100mhz	I	100 MHz Graphics Controller core clock
clk_pixel	I	Pixel clock (25/40/65Mhz for VGA/SVGA/XGA timing)
h_sync	O	Horizontal synchronization signal to the display device
v_sync	O	Vertical synchronization signal to the display device
rgb_out[*color depth*..0]	O	RGB output to the display device (Maximum 24bits)
cs_n_sdram	O	SDRAM chip select, active low
cke_sdram	O	SDRAM clock enable
address_sdram[*sdram_addr_bits*..0]	O	SDRAM address
ba_sdram[1..0]	O	SDRAM bank address
ras_n_sdram	O	SDRAM row address strobe, active low
cas_n_sdram	O	SDRAM column address strobe, active low
we_n_sdram	O	SDRAM write enable signal, active low
dqm_sdram[1..0]	O	SDRAM data byte mask
data_sdram[*sdram_data_bits*..0]	I/O	SDRAM data input/output
x_cord_pointer_in[9..0]	I	X-coordinate from pointing device
y_cord_pointer_in[9..0]	I	Y-coordinate from pointing device

Table 3: Prevas Graphics Controller core I/O signals

Core Modifications

The followings are some of the modifications that Prevas can implement on request.

- Adaptation to support smaller STN/TFT displays with different timings.
- DVI or other interface can be added.
- Other HW acceleration functions/algorithms can be implemented.
- Number of colors supported by the HW cursor can be extended.
- The Host interface can be modified to support different CPUs etc.
- Upon request the graphics controller core can be modified to support FPGA devices from other vendors.

Verification Methods

The Graphics Controller core's functionality has been extensively tested with a VHDL test bench and a large number of test cases. The functionality has also been verified in a Nios II environment through implementation in Altera Cyclone II FPGA devices.

A demonstration Nios II based reference design including a PS2 pointing device controller is provided with the core.

Design Services

Prevas AB also offers core integration, core customisation and other design services.

Ordering Information

This product is available from Prevas AB. See www.prevas.com for pricing or contact Prevas for additional information about this product.

Prevas AB
PO Box 4 (Legeringsgatan 18)
SE-721 03 Västerås, Sweden

Phone: +46 21 360 19 00
Fax: +46 21 360 19 29
Email: johan.ohlsson@prevas.se
URL: www.prevas.com

Related Information

Altera Programmable Logic

For information on Altera programmable logic or development system software, contact your local Altera sales office, or visit:

www.altera.com