임베디드시스템설계및실습

실습 5

VGA Programming on KAU Computer System
(VGAProg01)

Ref: Video IP Cores for Altera DE-Series Boards

Contents

- VGA_Subsystem in KAU_Computer
- An Image Frame
- VGA pixel DMA (Pixel Buffer DMA)
- Exercise 1 : LCD Dead Pixel Test
- Exercise 2 : Displaying Checker-board pattern
- Exercise 3 : Line-Drawing
- Homework 4

VGA_Subsystem in KAU_Computer

- Display Control in KAU_Computer
  - To control VGA DAC and the images on a screen
  - VGA Controller
  - VGA Pixel Buffer

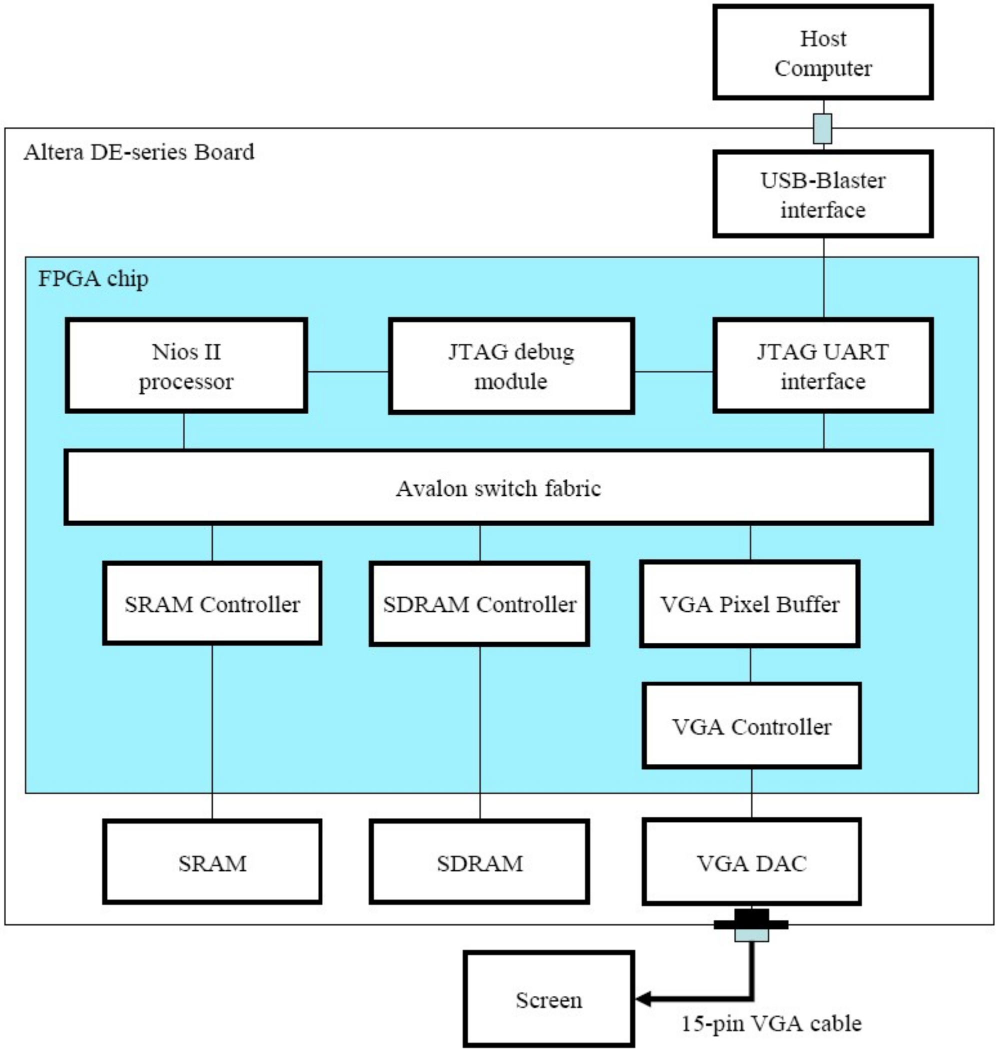
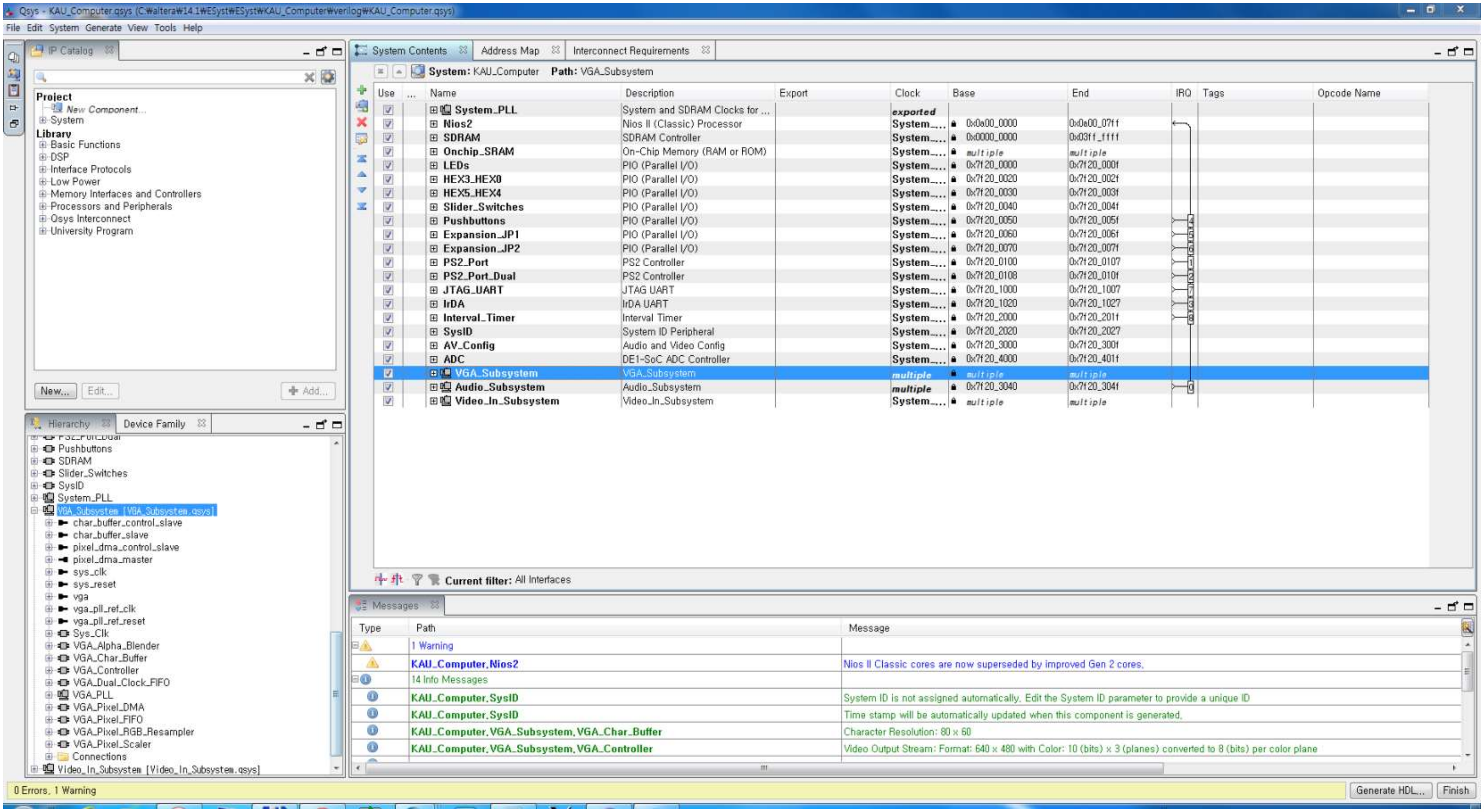


Figure 1. Portion of the DE-series Media Computer used in this exercise

KAU_Computer



Qsys - KAU_Computer.qsys (C:₩altera₩14.1₩ESyst₩ESyst₩KAU_Computer₩verilog₩KAU_Computer.qsys)

```
PS2_Port_Dual
Pushbuttons
SDRAM
Slider_Switches
SysID
System_PLL
VGA_Subsystem [VGA_Subsystem.qsys]
   char_buffer_control_slave
   char_buffer_slave
   pixel_dma_control_slave
   pixel_dma_master
   sys_clk
   sys_reset
   vga
   vga_pll_ref_clk
   vga_pll_ref_reset
   Sys_Clk
   VGA_Alpha_Blender
   VGA_Char_Buffer
   VGA_Controller
   VGA_Dual_Clock_FIFO
   VGA_PLL
   VGA_Pixel_DMA
   VGA_Pixel_FIFO
   VGA_Pixel_RGB_Resampler
   VGA_Pixel_Scaler
   Connections
Video_In_Subsystem [Video_In_Subsystem.qsys]
```

System: KAU_Computer **Path:** VGA_Subsystem

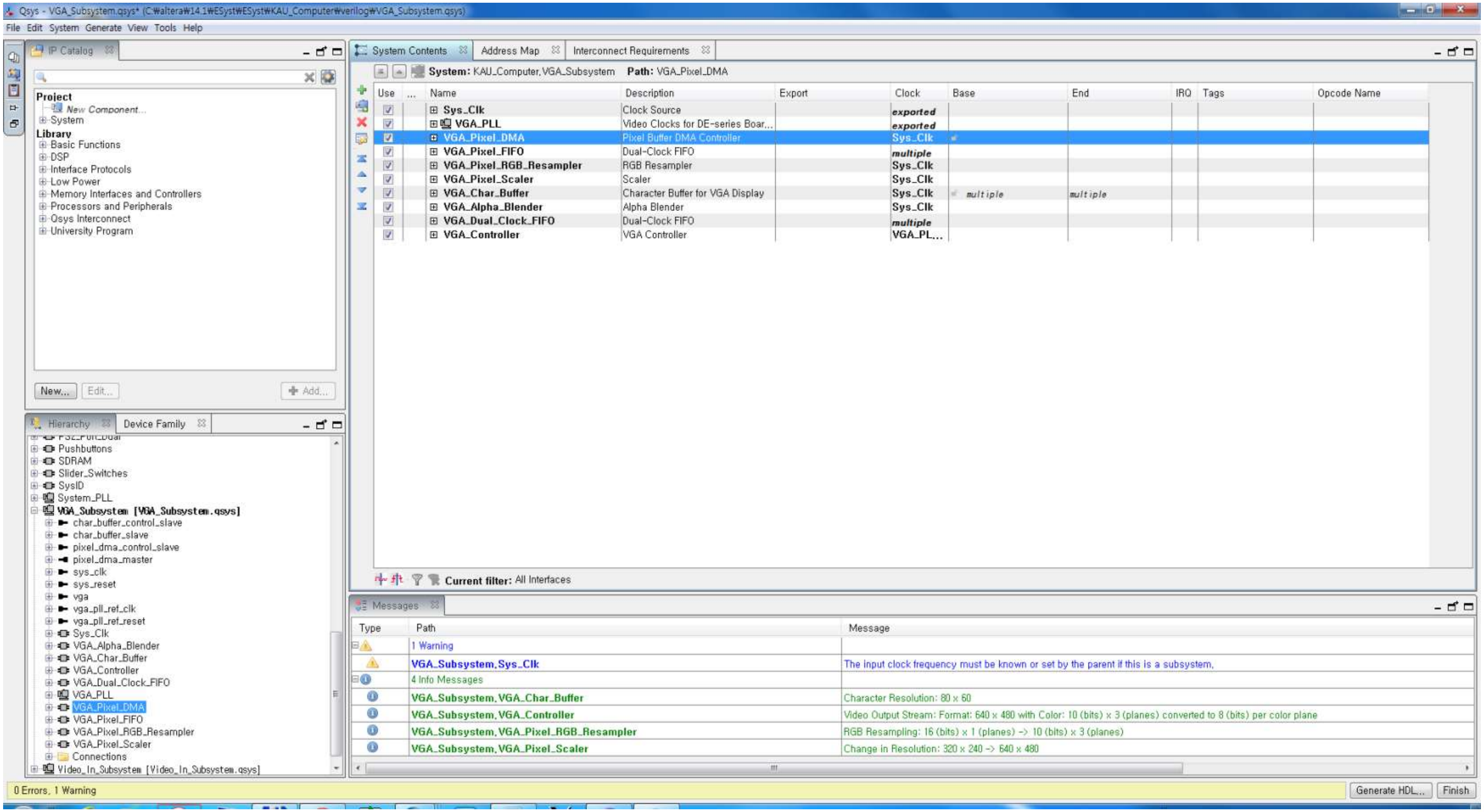
Use	...	Name	Description	Export	Clock	Base	End	IRQ	Tags	Opcode Name
☑		System_PLL	System and SDRAM Clocks for ...		exported					
☑		Nios2	Nios II (Classic) Processor		System_...	0x0a00_0000	0x0a00_07ff			
☑		SDRAM	SDRAM Controller		System_...	0x0000_0000	0x03ff_ffff			
☑		Onchip_SRAM	On-Chip Memory (RAM or ROM)		System_...	multiple	multiple			
☑		LEDs	PIO (Parallel I/O)		System_...	0x7f20_0000	0x7f20_000f			
☑		HEX3_HEX0	PIO (Parallel I/O)		System_...	0x7f20_0020	0x7f20_002f			
☑		HEX5_HEX4	PIO (Parallel I/O)		System_...	0x7f20_0030	0x7f20_003f			
☑		Slider_Switches	PIO (Parallel I/O)		System_...	0x7f20_0040	0x7f20_004f			
☑		Pushbuttons	PIO (Parallel I/O)		System_...	0x7f20_0050	0x7f20_005f	4		
☑		Expansion_JP1	PIO (Parallel I/O)		System_...	0x7f20_0060	0x7f20_006f	5		
☑		Expansion_JP2	PIO (Parallel I/O)		System_...	0x7f20_0070	0x7f20_007f	6		
☑		PS2_Port	PS2 Controller		System_...	0x7f20_0100	0x7f20_0107	1		
☑		PS2_Port_Dual	PS2 Controller		System_...	0x7f20_0108	0x7f20_010f	2		
☑		JTAG_UART	JTAG UART		System_...	0x7f20_1000	0x7f20_1007	7		
☑		IrDA	IrDA UART		System_...	0x7f20_1020	0x7f20_1027	3		
☑		Interval_Timer	Interval Timer		System_...	0x7f20_2000	0x7f20_201f	8		
☑		SysID	System ID Peripheral		System_...	0x7f20_2020	0x7f20_2027			
☑		AV_Config	Audio and Video Config		System_...	0x7f20_3000	0x7f20_300f			
☑		ADC	DE1-SoC ADC Controller		System_...	0x7f20_4000	0x7f20_401f			
☑		VGA_Subsystem	VGA_Subsystem		multiple	multiple	multiple			
☑		Audio_Subsystem	Audio_Subsystem		multiple	0x7f20_3040	0x7f20_304f	0		
☑		Video_In_Subsystem	Video_In_Subsystem		System_...	multiple	multiple			

Current filter: All Interfaces

Type	Path	Message
⚠	1 Warning	
⚠	KAU_Computer.Nios2	Nios II Classic cores are now superseded by improved Gen 2 cores.
ℹ	14 Info Messages	
ℹ	KAU_Computer.SysID	System ID is not assigned automatically. Edit the System ID parameter to provide a unique ID
ℹ	KAU_Computer.SysID	Time stamp will be automatically updated when this component is generated.
ℹ	KAU_Computer.VGA_Subsystem.VGA_Char_Buffer	Character Resolution: 80 x 60
ℹ	KAU_Computer.VGA_Subsystem.VGA_Controller	Video Output Stream: Format: 640 x 480 with Color: 10 (bits) x 3 (planes) converted to 8 (bits) per color plane

0 Errors, 1 Warning

VGA_Subsystem in KAU_Computer


Qsys - VGA_Subsystem.qsys* (C:₩altera₩14.1₩ESyst₩ESyst₩KAU_Computer₩verilog₩VGA_Subsystem.qsys)
File Edit System Generate View Tools Help
IP Catalog
Project
New Component...
System
Library
Basic Functions
DSP
Interface Protocols
Low Power
Memory Interfaces and Controllers
Processors and Peripherals
Qsys Interconnect
University Program
New...
Edit...
Add...
Hierarchy
Device Family
System Contents
Address Map
Interconnect Requirements
Current filter: All Interfaces
Messages
Generate HDL...
Finish

- PS2_Port_Dual
- Pushbuttons
- SDRAM
- Slider_Switches
- SysID
- System_PLL
- VGA_Subsystem [VGA_Subsystem.qsys]
  - char_buffer_control_slave
  - char_buffer_slave
  - pixel_dma_control_slave
  - pixel_dma_master
  - sys_clk
  - sys_reset
  - vga
  - vga_pll_ref_clk
  - vga_pll_ref_reset
  - Sys_Clk
  - VGA_Alpha_Blender
  - VGA_Char_Buffer
  - VGA_Controller
  - VGA_Dual_Clock_FIFO
  - VGA_PLL
  - VGA_Pixel_DMA
  - VGA_Pixel_FIFO
  - VGA_Pixel_RGB_Resampler
  - VGA_Pixel_Scaler
  - Connections
- Video_In_Subsystem [Video_In_Subsystem.qsys]

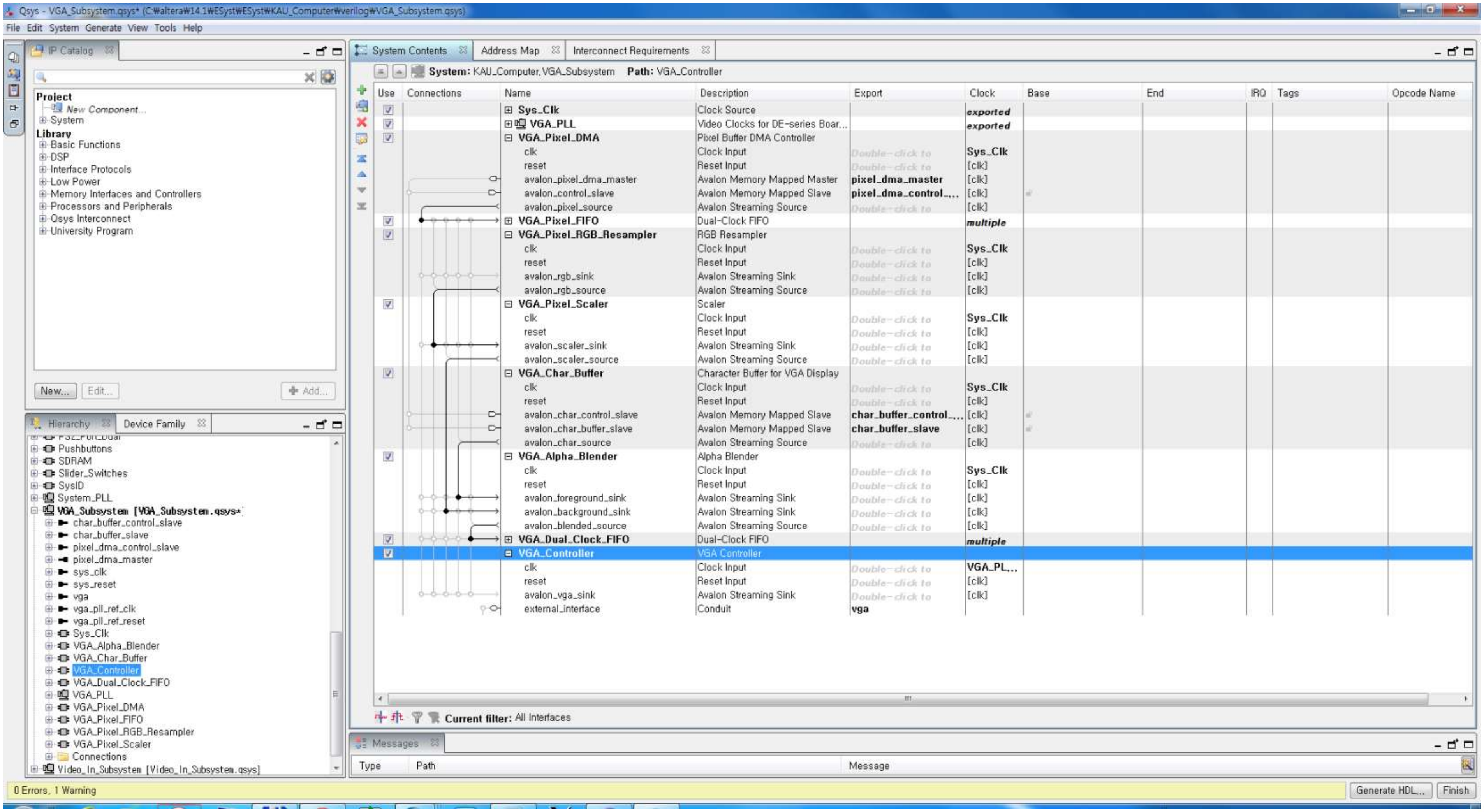
System: KAU_Computer.VGA_Subsystem **Path:** VGA_Pixel_DMA

Use	...	Name	Description	Export	Clock	Base	End	IRQ	Tags	Opcode Name
☑		⊞ Sys_Clk	Clock Source		*exported*					
☑		⊞ VGA_PLL	Video Clocks for DE-series Boar...		*exported*					
☑		⊞ VGA_Pixel_DMA	Pixel Buffer DMA Controller		Sys_Clk					
☑		⊞ VGA_Pixel_FIFO	Dual-Clock FIFO		*multiple*					
☑		⊞ VGA_Pixel_RGB_Resampler	RGB Resampler		Sys_Clk					
☑		⊞ VGA_Pixel_Scaler	Scaler		Sys_Clk					
☑		⊞ VGA_Char_Buffer	Character Buffer for VGA Display		Sys_Clk	multiple	multiple			
☑		⊞ VGA_Alpha_Blender	Alpha Blender		Sys_Clk					
☑		⊞ VGA_Dual_Clock_FIFO	Dual-Clock FIFO		*multiple*					
☑		⊞ VGA_Controller	VGA Controller		VGA_PL...					

Type	Path	Message
⊟⚠	1 Warning	
⚠	VGA_Subsystem.Sys_Clk	The input clock frequency must be known or set by the parent if this is a subsystem.
⊟ⓘ	4 Info Messages	
ⓘ	VGA_Subsystem.VGA_Char_Buffer	Character Resolution: 80 x 60
ⓘ	VGA_Subsystem.VGA_Controller	Video Output Stream: Format: 640 x 480 with Color: 10 (bits) x 3 (planes) converted to 8 (bits) per color plane
ⓘ	VGA_Subsystem.VGA_Pixel_RGB_Resampler	RGB Resampling: 16 (bits) x 1 (planes) -> 10 (bits) x 3 (planes)
ⓘ	VGA_Subsystem.VGA_Pixel_Scaler	Change in Resolution: 320 x 240 -> 640 x 480

0 Errors, 1 Warning

VGA_Subsystem in KAU_Computer



Project
- New Component...
- System

Library
- Basic Functions
- DSP
- Interface Protocols
- Low Power
- Memory Interfaces and Controllers
- Processors and Peripherals
- Qsys Interconnect
- University Program

- Pushbuttons
- SDRAM
- Slider_Switches
- SysID
- System_PLL
- VGA_Subsystem [VGA_Subsystem.qsys*]
  - char_buffer_control_slave
  - char_buffer_slave
  - pixel_dma_control_slave
  - pixel_dma_master
  - sys_clk
  - sys_reset
  - vga
  - vga_pll_ref_clk
  - vga_pll_ref_reset
  - Sys_Clk
  - VGA_Alpha_Blender
  - VGA_Char_Buffer
  - VGA_Controller
  - VGA_Dual_Clock_FIFO
  - VGA_PLL
  - VGA_Pixel_DMA
  - VGA_Pixel_FIFO
  - VGA_Pixel_RGB_Resampler
  - VGA_Pixel_Scaler
  - Connections
- Video_In_Subsystem [Video_In_Subsystem.qsys]

System: KAU_Computer, VGA_Subsystem **Path:** VGA_Controller

Use	Connections	Name	Description	Export	Clock	Base	End	IRQ	Tags	Opcode Name
☑		⊞ **Sys_Clk**	Clock Source		***exported***					
☑		⊞ **VGA_PLL**	Video Clocks for DE-series Boar...		***exported***					
☑		⊟ **VGA_Pixel_DMA**	Pixel Buffer DMA Controller							
		clk	Clock Input	*Double-click to*	**Sys_Clk**					
		reset	Reset Input	*Double-click to*	[clk]					
		avalon_pixel_dma_master	Avalon Memory Mapped Master	**pixel_dma_master**	[clk]					
		avalon_control_slave	Avalon Memory Mapped Slave	**pixel_dma_control_...**	[clk]					
		avalon_pixel_source	Avalon Streaming Source	*Double-click to*	[clk]					
☑		⊞ **VGA_Pixel_FIFO**	Dual-Clock FIFO		***multiple***					
☑		⊟ **VGA_Pixel_RGB_Resampler**	RGB Resampler							
		clk	Clock Input	*Double-click to*	**Sys_Clk**					
		reset	Reset Input	*Double-click to*	[clk]					
		avalon_rgb_sink	Avalon Streaming Sink	*Double-click to*	[clk]					
		avalon_rgb_source	Avalon Streaming Source	*Double-click to*	[clk]					
☑		⊟ **VGA_Pixel_Scaler**	Scaler							
		clk	Clock Input	*Double-click to*	**Sys_Clk**					
		reset	Reset Input	*Double-click to*	[clk]					
		avalon_scaler_sink	Avalon Streaming Sink	*Double-click to*	[clk]					
		avalon_scaler_source	Avalon Streaming Source	*Double-click to*	[clk]					
☑		⊟ **VGA_Char_Buffer**	Character Buffer for VGA Display							
		clk	Clock Input	*Double-click to*	**Sys_Clk**					
		reset	Reset Input	*Double-click to*	[clk]					
		avalon_char_control_slave	Avalon Memory Mapped Slave	**char_buffer_control_...**	[clk]					
		avalon_char_buffer_slave	Avalon Memory Mapped Slave	**char_buffer_slave**	[clk]					
		avalon_char_source	Avalon Streaming Source	*Double-click to*	[clk]					
☑		⊟ **VGA_Alpha_Blender**	Alpha Blender							
		clk	Clock Input	*Double-click to*	**Sys_Clk**					
		reset	Reset Input	*Double-click to*	[clk]					
		avalon_foreground_sink	Avalon Streaming Sink	*Double-click to*	[clk]					
		avalon_background_sink	Avalon Streaming Sink	*Double-click to*	[clk]					
		avalon_blended_source	Avalon Streaming Source	*Double-click to*	[clk]					
☑		⊞ **VGA_Dual_Clock_FIFO**	Dual-Clock FIFO		***multiple***					
☑		⊟ **VGA_Controller**	VGA Controller							
		clk	Clock Input	*Double-click to*	**VGA_PL...**					
		reset	Reset Input	*Double-click to*	[clk]					
		avalon_vga_sink	Avalon Streaming Sink	*Double-click to*	[clk]					
		external_interface	Conduit	**vga**						

Current filter: All Interfaces

Type	Path	Message

0 Errors, 1 Warning

An Image Frame - 16bit RGB

- An image frame
  - Stored in memory
  - 16-bit RGB
  - Address of pixels
    - X-Y addressing mode
      - (BaseAddr)+(pixel offset)
    - Continuous mode
      - (BaseAddr)+(W*y+x)*2

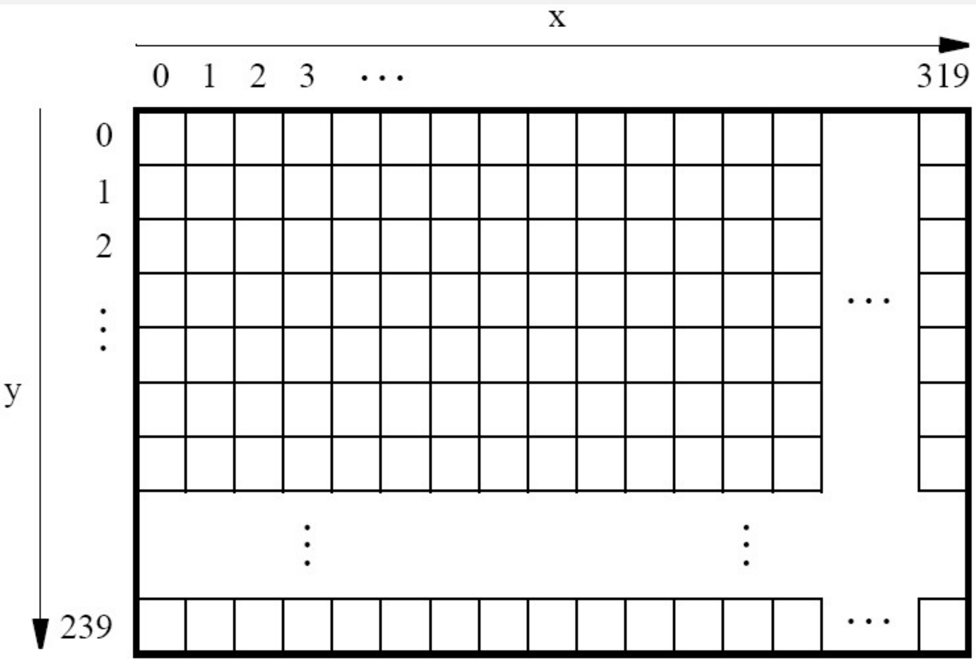
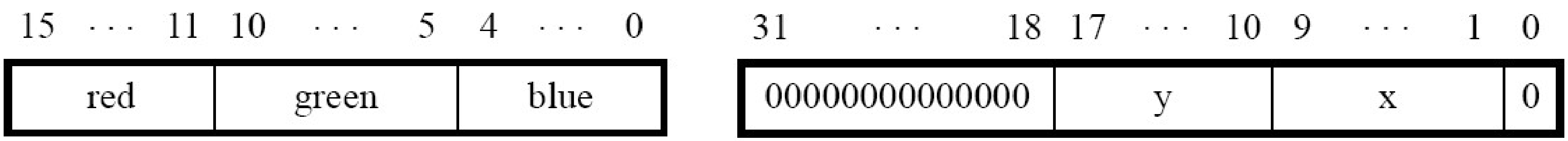


Figure 2. Pixel array.



(a) Pixel color

(b) Pixel (x,y) offset

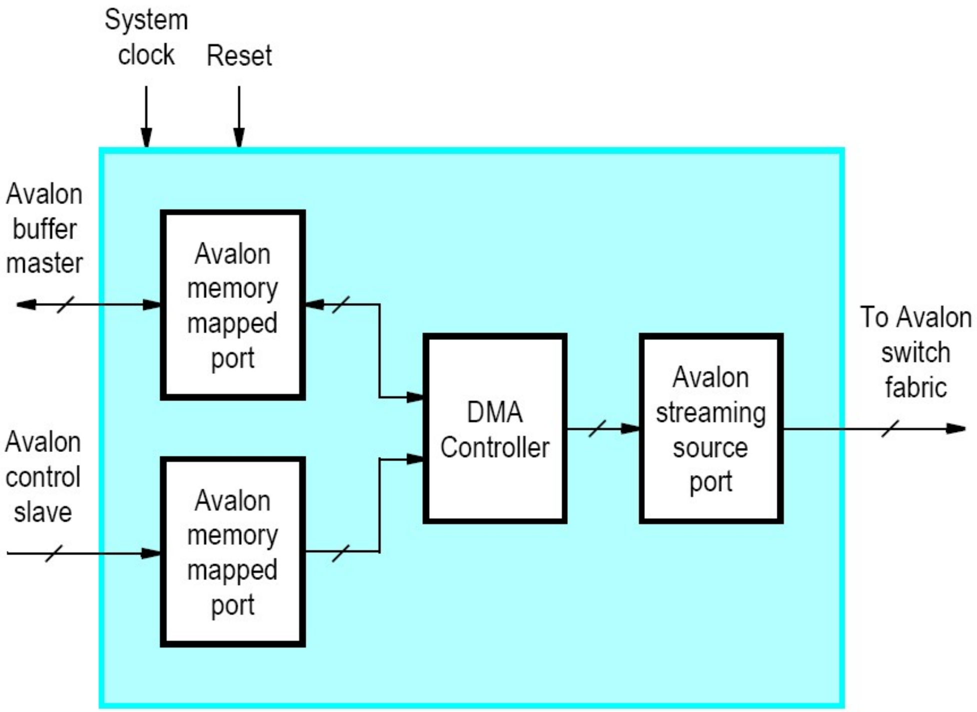
Figure 3. Pixel color and offset.

An Image Frame - 8/24/32 bit RGB

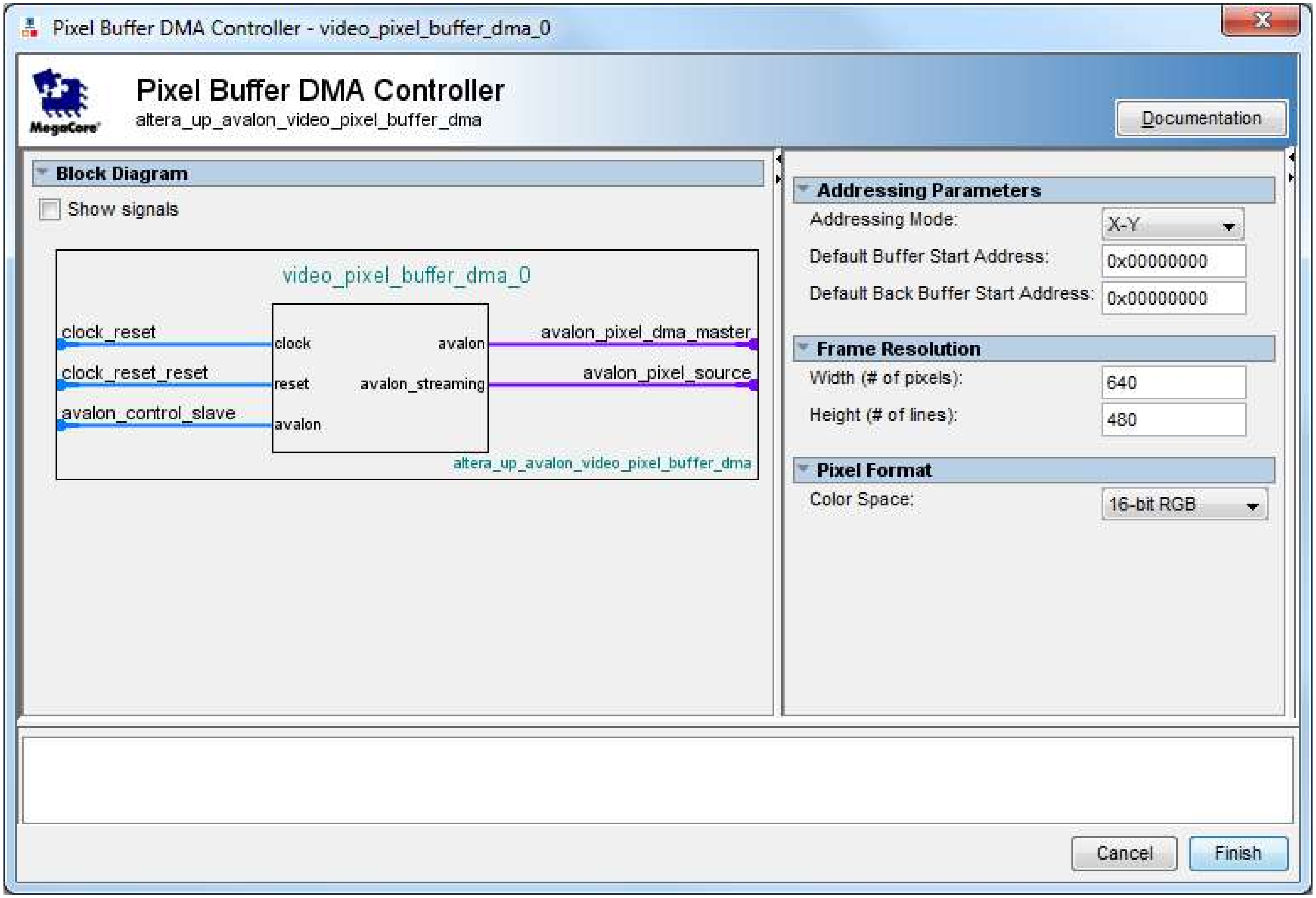
- For other color pixel representation, refer to section 3.1 and 3.2 of *Video IP Cores for Altera DE-Series Boards*

VGA pixel DMA (Pixel Buffer DMA)

- VGA pixel (buffer) DMA
  - An interface between the VGA controller and the memory where the image frames are stored
  - Memory-mapped registers for accessing the VGA pixel buffer control



VGA pixel DMA (Pixel Buffer DMA)



Pixel Buffer DMA Controller
altera_up_avalon_video_pixel_buffer_dma

□ Show signals

Addressing Parameters
Addressing Mode: X-Y
Default Buffer Start Address: 0x00000000
Default Back Buffer Start Address: 0x00000000

Frame Resolution
Width (# of pixels): 640
Height (# of lines): 480

Pixel Format
Color Space: 16-bit RGB

VGA pixel DMA (Pixel Buffer DMA)

- Software Programming Model

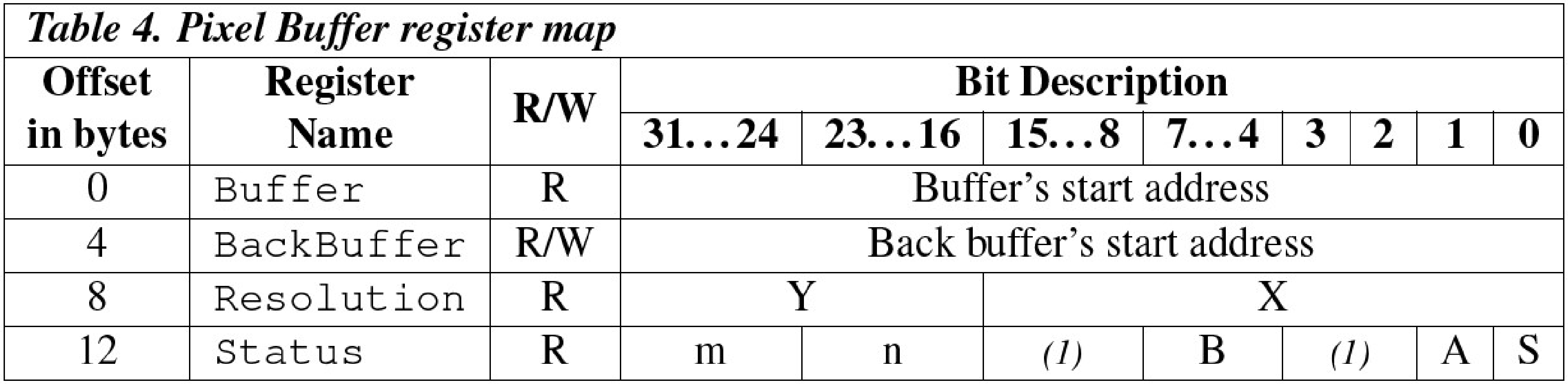


Table 4. Pixel Buffer register map

Offset in bytes	Register Name	R/W	Bit Description							
			31…24	23…16	15…8	7…4	3	2	1	0
0	Buffer	R	Buffer's start address							
4	BackBuffer	R/W	Back buffer's start address							
8	Resolution	R	Y		X					
12	Status	R	m	n	*(1)*	B	*(1)*		A	S

- *Buffer* : 32-bit address of the start of the frame buffer
- *BackBuffer* : the start address of the frame buffer to be changed
  - Write the desired frame's start address into the *BackBuffer* register
  - Second write operation on the *Buffer* register
    - A request to swap the contents of the *Buffer* and the *BackBuffer* registers
- *Resolution* : the resolution of the image frame (or the screen)

VGA pixel DMA (Pixel Buffer DMA)

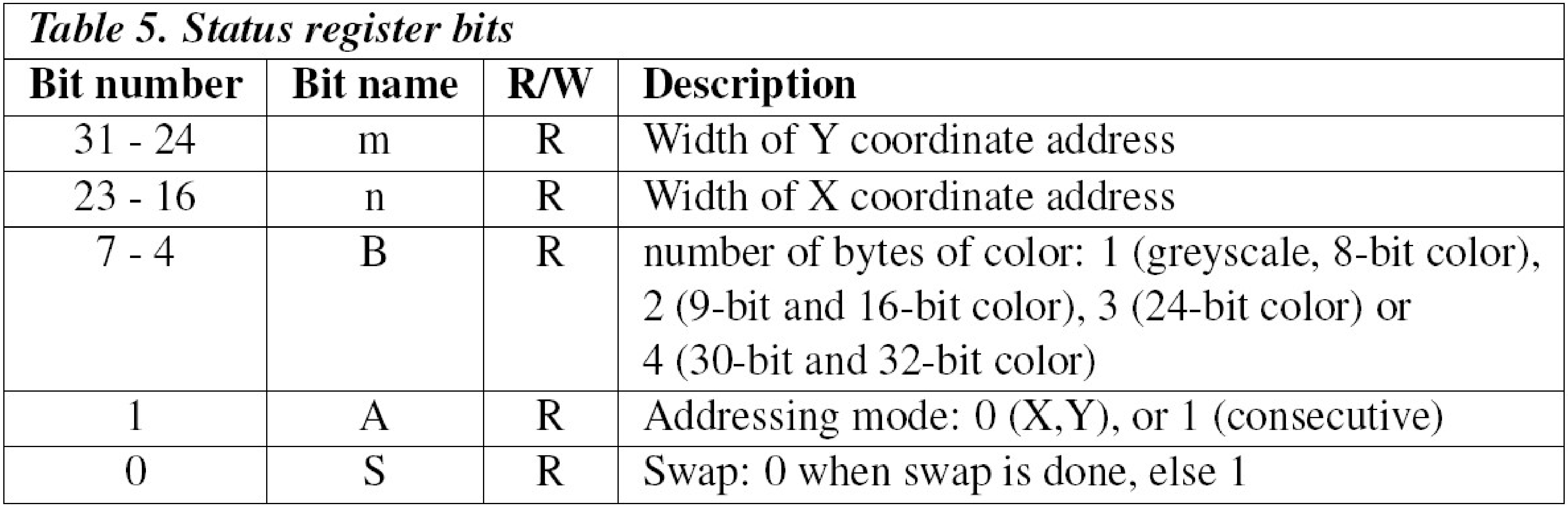


Table 5. Status register bits

Bit number	Bit name	R/W	Description
31 - 24	m	R	Width of Y coordinate address
23 - 16	n	R	Width of X coordinate address
7 - 4	B	R	number of bytes of color: 1 (greyscale, 8-bit color), 2 (9-bit and 16-bit color), 3 (24-bit color) or 4 (30-bit and 32-bit color)
1	A	R	Addressing mode: 0 (X,Y), or 1 (consecutive)
0	S	R	Swap: 0 when swap is done, else 1

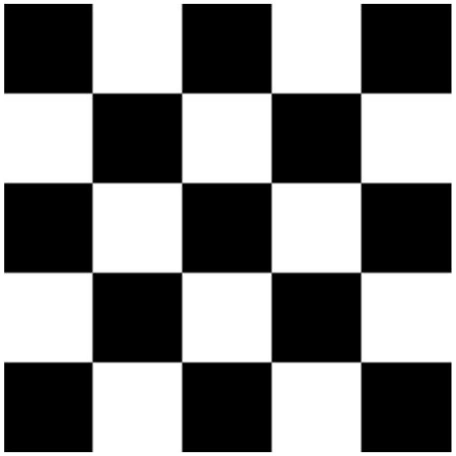
- The Pixel Buffer's Device Drivers for the Nios II Processor
  - Refer to *Video IP Cores for Altera DE-Series Boards*

Exercise 1 : LCD Dead Pixel Test

- This is a simple test designed to make it easier to detect LCD "dead pixels" (pixels that are either **stuck-ON** or **stuck-OFF**) by displaying a series of solid background color pages for contrast.
- Design a function which fill a rectangle with a given 16-bit RGB data
  - *void FillPattern16(alt_u16 *pFB, alt_u16 sx, alt_u16 sy, alt_u16 w, alt_u16 h, alt_u16 data, int addr_mode);*
- LCD Display Test
  - Check if there is any dead pixels on your LCD display
  - Prepare image frames of which pixels are all in R, G, B, White, or Black
  - Write a program to display the images prepared repeatedly with the following transition order R➔G➔B➔White➔Black
  - Can you find any dead pixels?

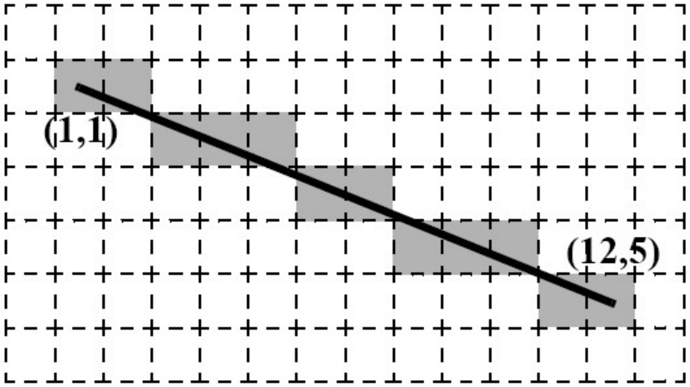
Exercise 2 : Displaying Checker-board pattern

- Draw a checkerboard pattern on the VGA display
  - Write a pattern to the memory buffer
  - Configure the Pixel Buffer DMA to access the memory buffer
- Animation
  - Write three different checkerboard patterns to three separate memory buffers (different sizes or colors at your own)
  - Change the patterns with arbitrary order and interval



Exercise 3 : Line-Drawing

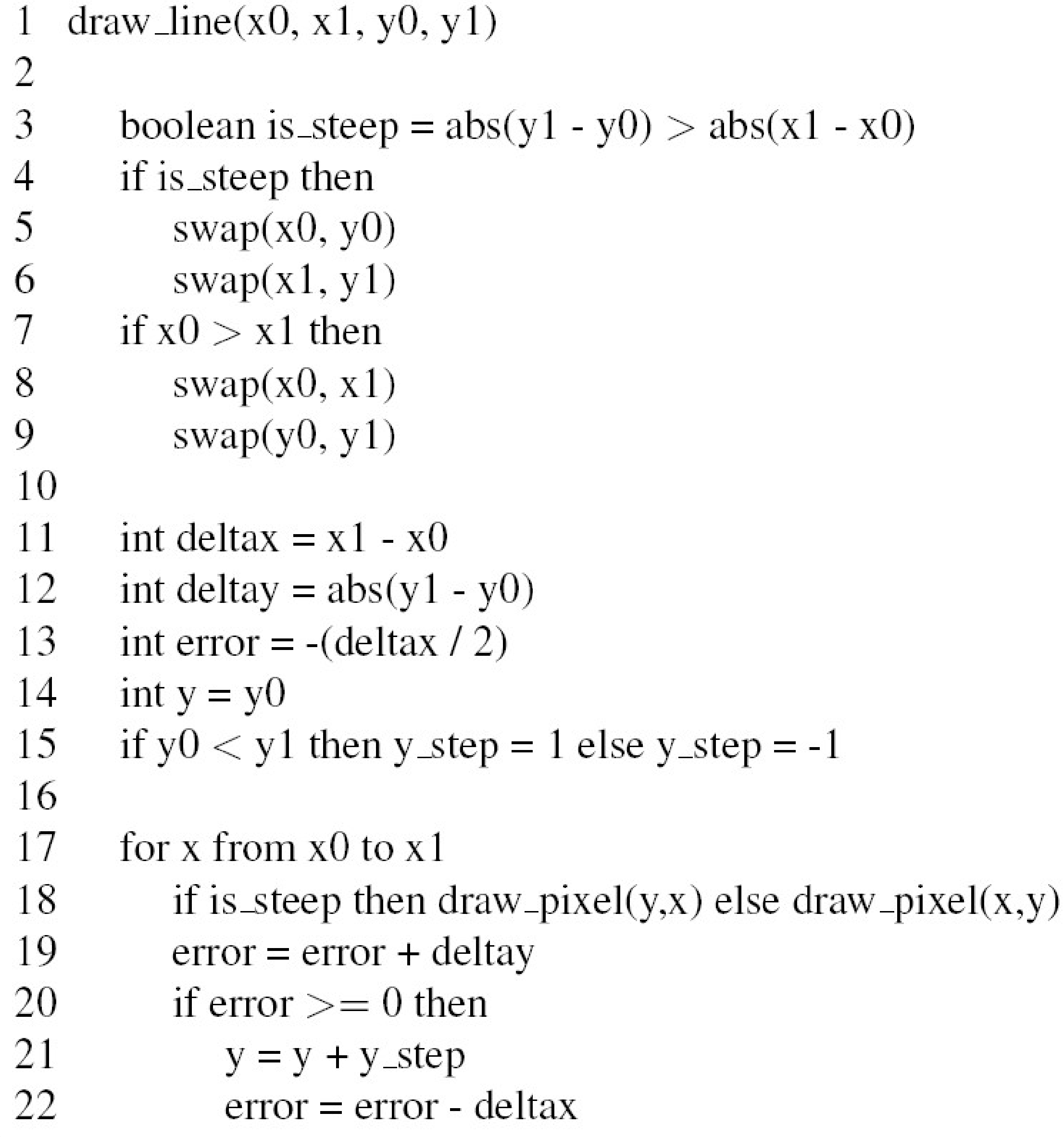
- Drawing a line on a screen
  - Coloring pixels between two points (x1, y1) and (x2, y2) such that they resemble a line as closely as possible
  - Drawing a line between points (1,1) and (12,5)



- An algorithm
  - slope = (y2-y1)/(x2-x1)
  - Move along the x-axis and compute y-coordinate
    - y = slope * (x-x1) + y1
    - moving along the x axis has a drawback when a line is steep
  - Bresenham’s algorithm
    - alter the algorithm to move along the y axis when a line is steep

Exercise 3 : Line-Drawing

- Bresenham’s algorithm
- Write a C-language program that implements the line algorithm
- Download the program into the KAU_Computer
- Connect a 15-pin VGA cable to the VGA connector on the board and th LCD monitor
- Run your program



```
draw_line(x0, x1, y0, y1)

    boolean is_steep = abs(y1 - y0) > abs(x1 - x0)
    if is_steep then
        swap(x0, y0)
        swap(x1, y1)
    if x0 > x1 then
        swap(x0, x1)
        swap(y0, y1)

    int deltax = x1 - x0
    int deltay = abs(y1 - y0)
    int error = -(deltax / 2)
    int y = y0
    if y0 < y1 then y_step = 1 else y_step = -1

    for x from x0 to x1
        if is_steep then draw_pixel(y,x) else draw_pixel(x,y)
        error = error + deltay
        if error >= 0 then
            y = y + y_step
            error = error - deltax
```

Homework 4, Due 5/2

- Inspect the IP modules from the Pixel Buffer DMA to the VGA controller in VGA_Subsystem of KAU_Computer
  - Read the data sheet for the IP modules
    - Dual-Clock FIFO
    - RGB Resampler
    - Scaler
    - Character buffer for VGA display
    - Alpha Blender
  - Summarizes the functions of each modules briefly (2 pages in A4)
- Modify the VGA_Subsystem such that it can handle 8-bit gray-scale image frame
  - Name the system as KAU_Computer_8BGray
  - Generate & Compile & program it to display checker board pattern

끝