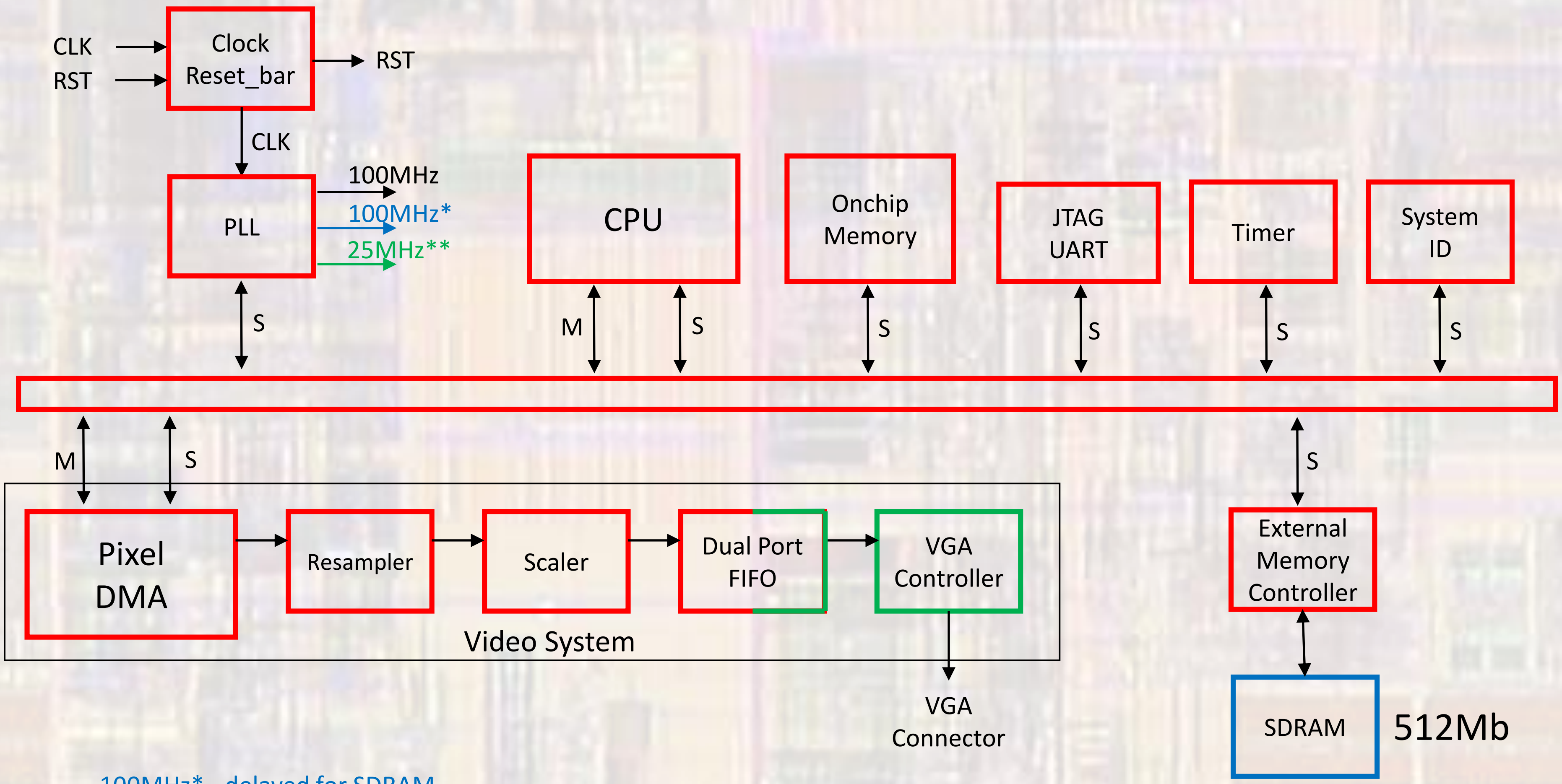
NIOS Pixel

Last updated 7/23/19

NIOS II Pixel Display



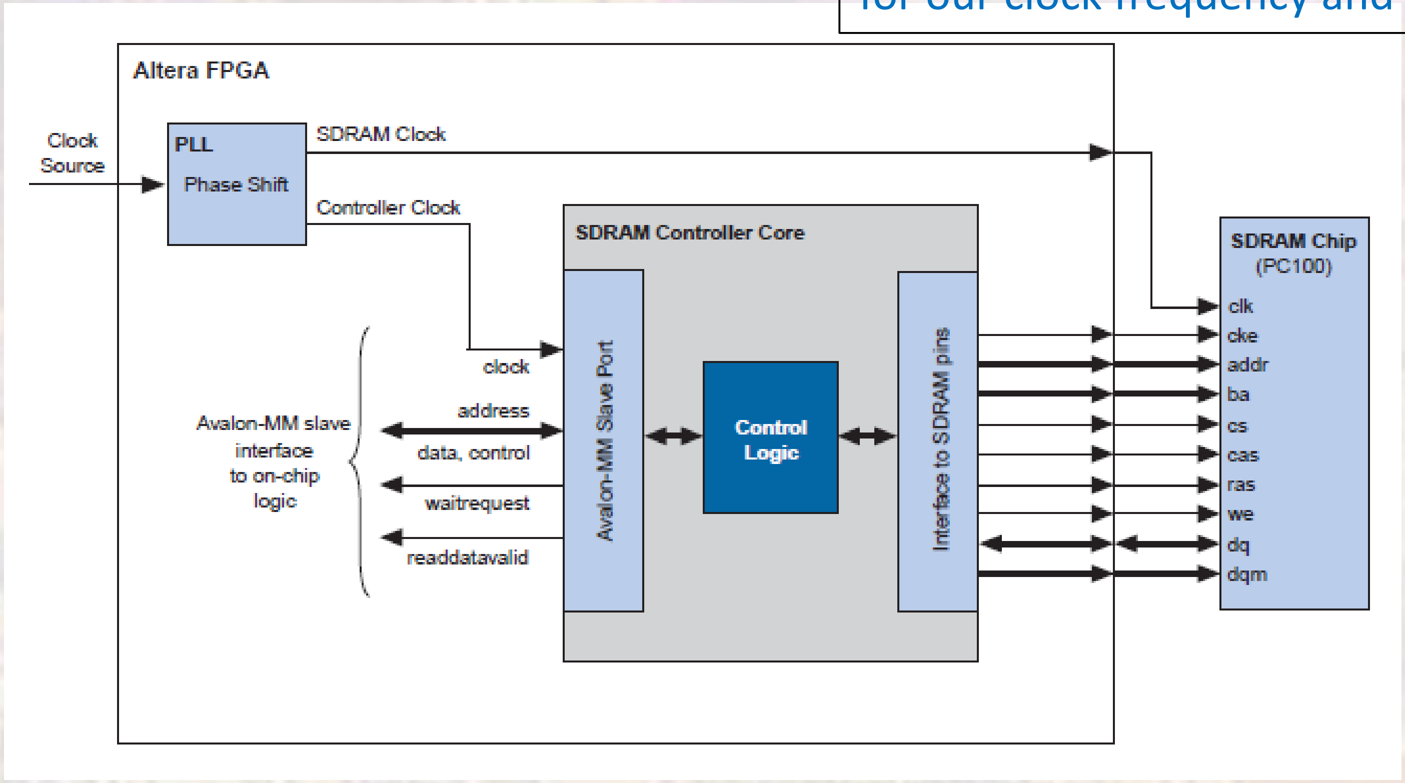
100MHz* - delayed for SDRAM

25MHz** - VGA clk

NIOS II Pixel Display

- SDRAM Controller

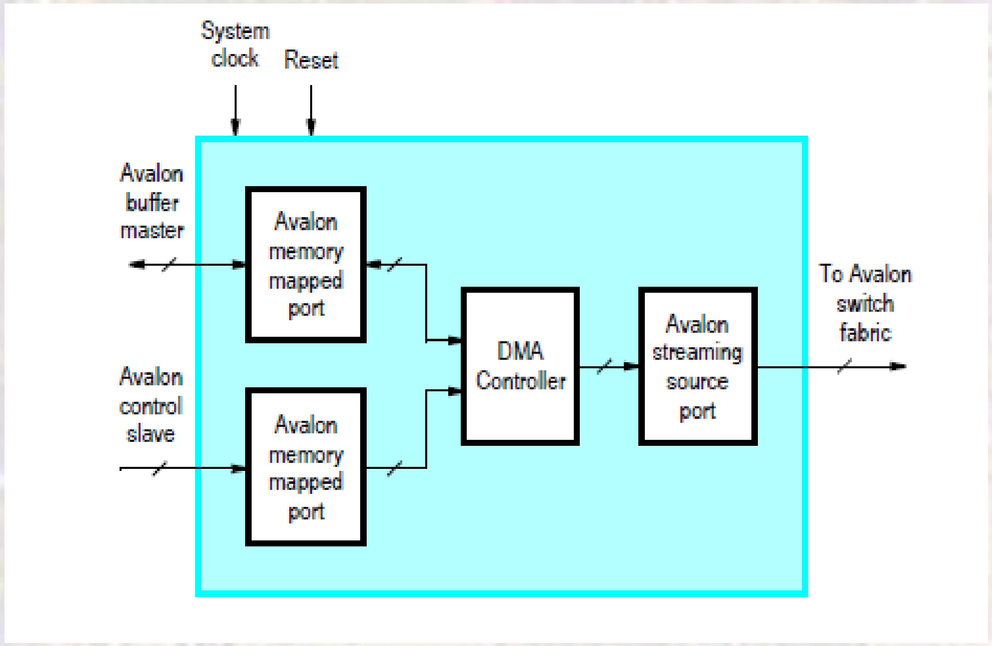
Requires a -3ns clock offset
for our clock frequency and DRAM



Creates the signals required to r/w the SDRAM (does not provide the DRAM clk)

NIOS II Pixel Display

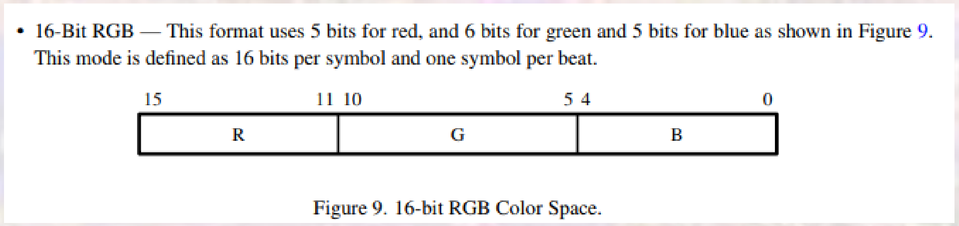
- Pixel Buffer DMA Controller



Reads frames from external memory and passes them into the video processing chain

NIOS II Pixel Display

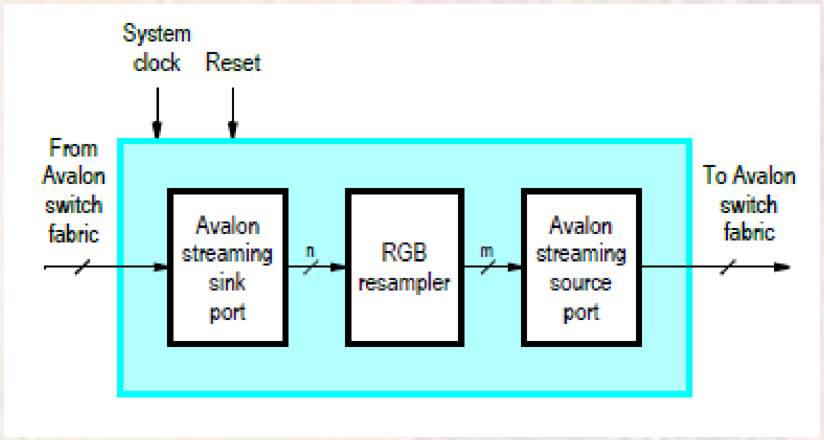
- RGB Resampler



- 16-Bit RGB — This format uses 5 bits for red, and 6 bits for green and 5 bits for blue as shown in Figure 9. This mode is defined as 16 bits per symbol and one symbol per beat.

15 … 11	10 … 5	4 … 0
R	G	B

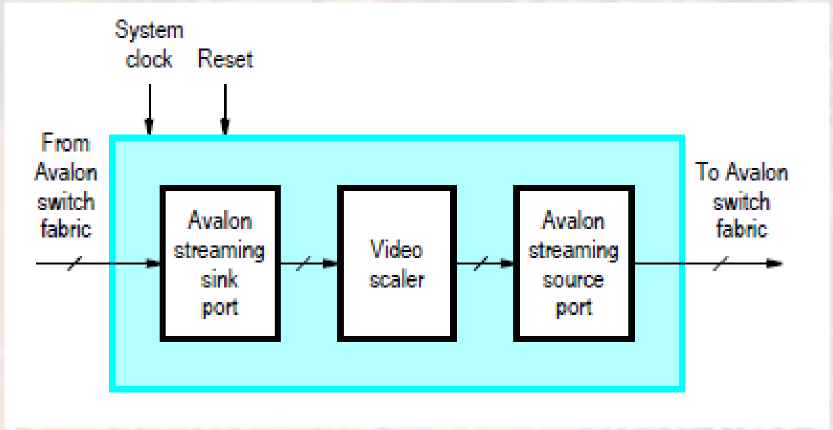
Figure 9. 16-bit RGB Color Space.


System clock
Reset
From Avalon switch fabric
Avalon streaming sink port
n
RGB resampler
m
Avalon streaming source port
To Avalon switch fabric

Converts from 16bit RGB to 30bit RGB

NIOS II Pixel Display

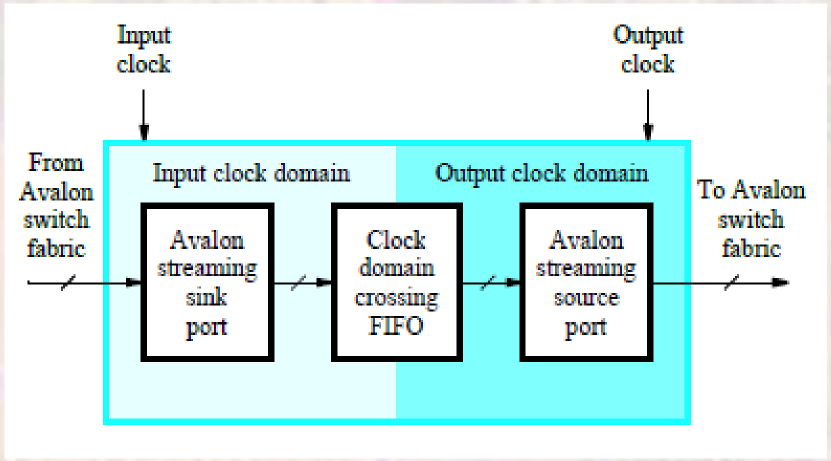
- Scaler


System clock
Reset
From Avalon switch fabric
Avalon streaming sink port
Video scaler
Avalon streaming source port
To Avalon switch fabric

Scales the video by replicating pixels or removing pixels

NIOS II Pixel Display

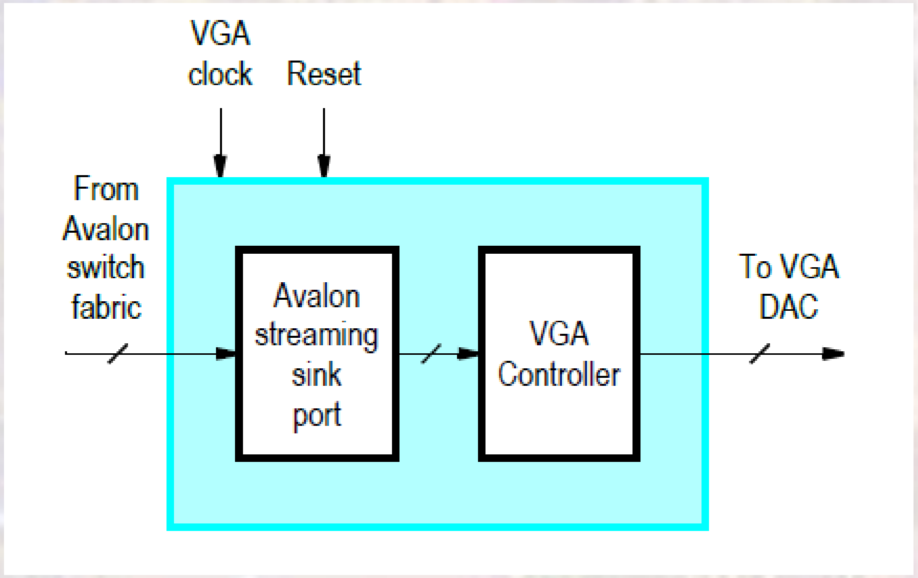
- Dual Clock FIFO


Input clock
Output clock
From Avalon switch fabric
Input clock domain
Output clock domain
To Avalon switch fabric
Avalon streaming sink port
Clock domain crossing FIFO
Avalon streaming source port

Allows different incoming and outgoing data rates

NIOS II Pixel Display

- VGA Controller Block

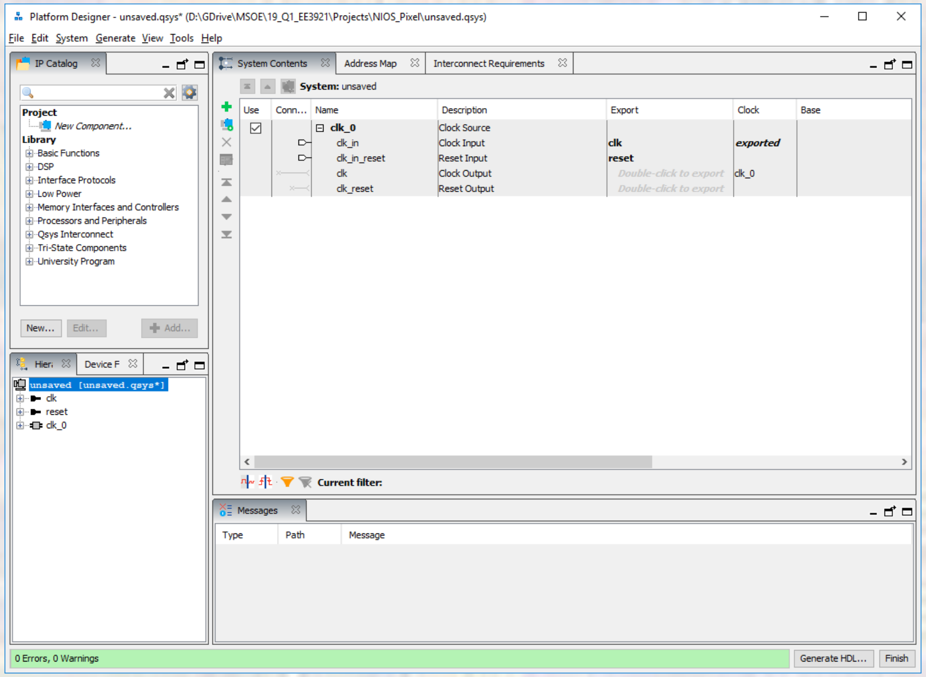


Creates and drives the required VGA signals

NIOS II Pixel Display

- Create a new Quartus project
  - Do not select a Simulation Tool in EDA Tool Settings

- Open Tools → Platform Designer



Platform Designer - unsaved.qsys* (D:\GDrive\MSOE\19_Q1_EE3921\Projects\NIOS_Pixel\unsaved.qsys)

File Edit System Generate View Tools Help

IP Catalog

Project
- New Component...

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

New... Edit... Add...

Hier Device F

```
unsaved [unsaved.qsys*]
  dk
  reset
  dk_0
```

System Contents | Address Map | Interconnect Requirements

System: unsaved

Use	Conn...	Name	Description	Export	Clock	Base
✓		**clk_0**	Clock Source			
		clk_in	Clock Input	**clk**	***exported***	
		clk_in_reset	Reset Input	**reset**		
		clk	Clock Output	*Double-click to export*	clk_0	
		clk_reset	Reset Output	*Double-click to export*		

Current filter:

Messages

Type	Path	Message

0 Errors, 0 Warnings

Generate HDL... Finish

NIOS II Pixel Display

- Add a PLL
  - Basic Functions → Clocks; PLLs and Resets → PLL → ALTPLL Intel FPGA IP

50MHz input clock

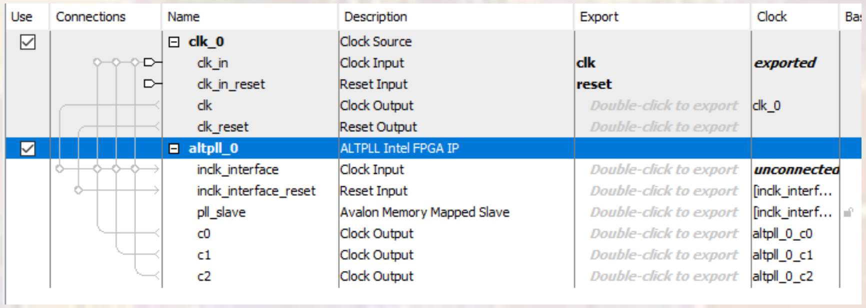
no areset or locked

c0 → 100MHz

c1 → 100MHz

-3ns phase shift (watch for the units)

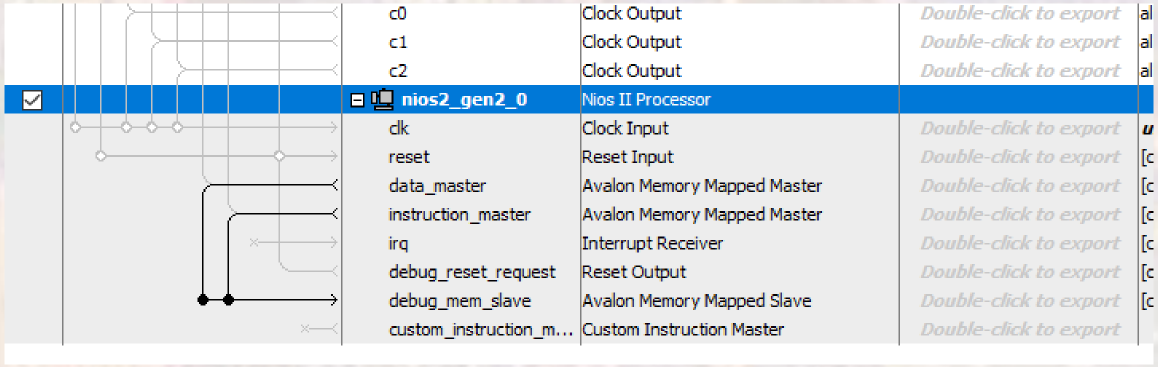
c2 → 25MHz



Use	Connections	Name	Description	Export	Clock	Ba
☑		**clk_0**	Clock Source			
		clk_in	Clock Input	**clk**	***exported***	
		clk_in_reset	Reset Input	**reset**		
		clk	Clock Output	*Double-click to export*	clk_0	
		clk_reset	Reset Output	*Double-click to export*		
☑		**altpll_0**	ALTPLL Intel FPGA IP			
		inclk_interface	Clock Input	*Double-click to export*	***unconnected***	
		inclk_interface_reset	Reset Input	*Double-click to export*	[inclk_interf...	
		pll_slave	Avalon Memory Mapped Slave	*Double-click to export*	[inclk_interf...	
		c0	Clock Output	*Double-click to export*	altpll_0_c0	
		c1	Clock Output	*Double-click to export*	altpll_0_c1	
		c2	Clock Output	*Double-click to export*	altpll_0_c2	

NIOS II Pixel Display

- Add NIOS
  - Processors and Peripherals → Embedded Processors → NIOS II Processor
  - NIOS II/f

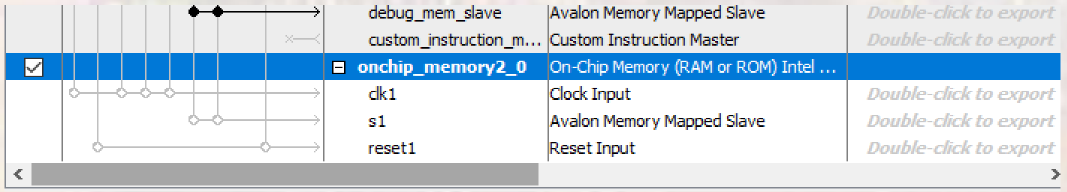


c0	Clock Output	Double-click to export	al
c1	Clock Output	Double-click to export	al
c2	Clock Output	Double-click to export	al
✓ nios2_gen2_0	Nios II Processor		
clk	Clock Input	Double-click to export	u
reset	Reset Input	Double-click to export	[c
data_master	Avalon Memory Mapped Master	Double-click to export	[c
instruction_master	Avalon Memory Mapped Master	Double-click to export	[c
irq	Interrupt Receiver	Double-click to export	[c
debug_reset_request	Reset Output	Double-click to export	[c
debug_mem_slave	Avalon Memory Mapped Slave	Double-click to export	[c
custom_instruction_m...	Custom Instruction Master	Double-click to export	

- Add On-chip Memory
  - Basic Functions→ On Chip Memory → On Chip Memory (RAM or ROM)…

    RAM

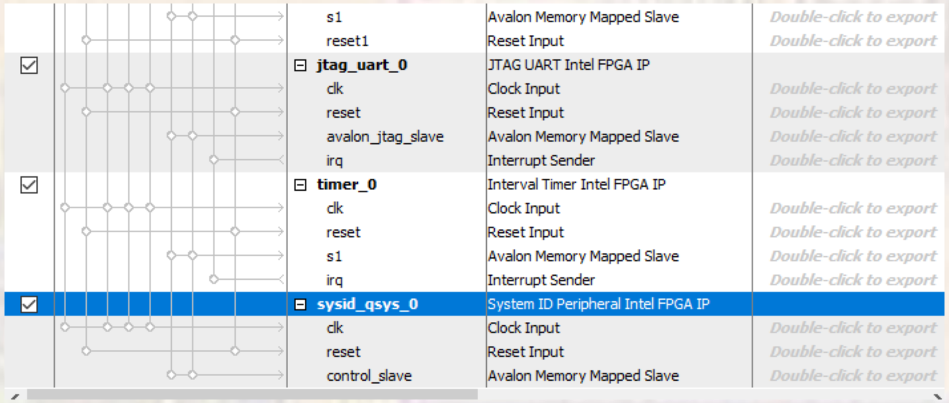
    Size = 100,000 bytes



debug_mem_slave	Avalon Memory Mapped Slave	Double-click to export
custom_instruction_m...	Custom Instruction Master	Double-click to export
✓ onchip_memory2_0	On-Chip Memory (RAM or ROM) Intel ...	
clk1	Clock Input	Double-click to export
s1	Avalon Memory Mapped Slave	Double-click to export
reset1	Reset Input	Double-click to export

NIOS II Pixel Display

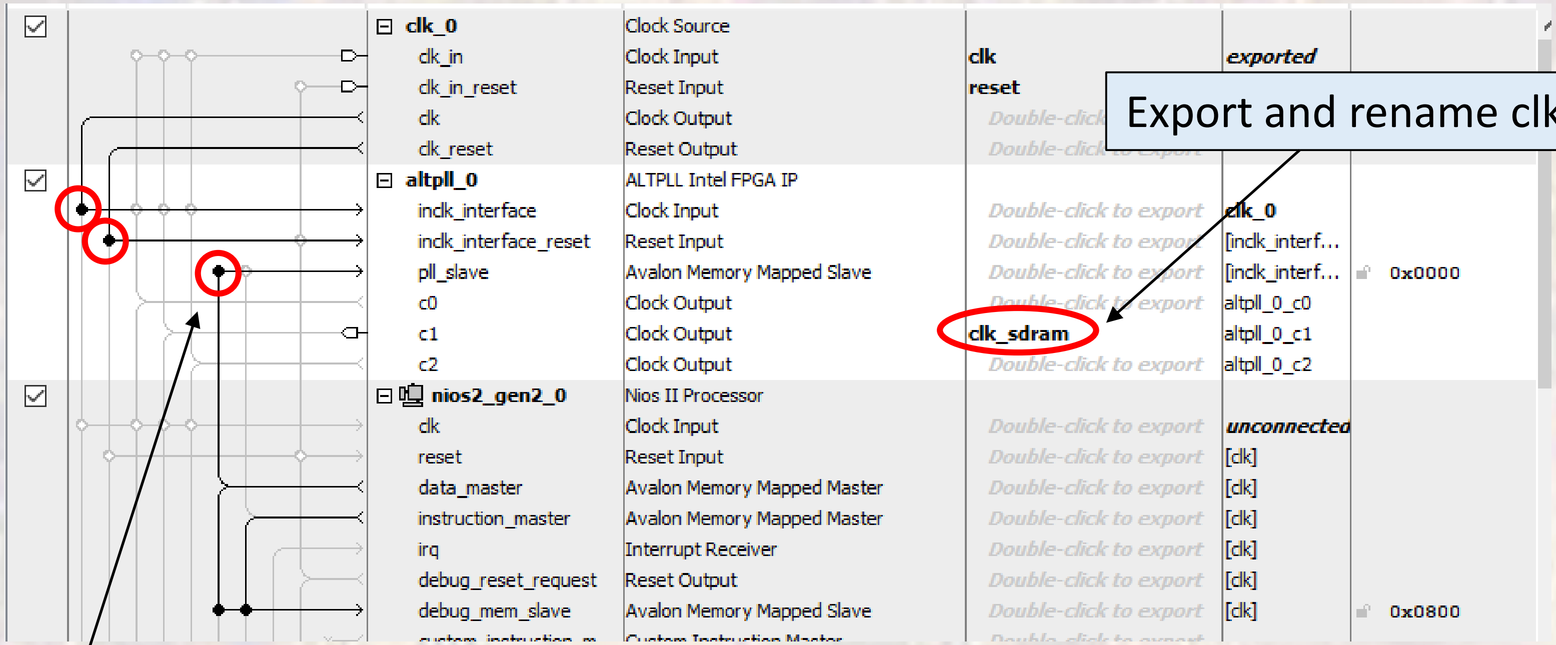
- Add JTAG
  - Interface Protocols → Serial→ JTAG Uart Intel FPGA IP
- Add Timer
  - Processors and Peripherals → Peripherals → Interval Timer Intel FPGA IP
- Add System ID
  - Basic Functions → Simulation; Debug and Verification → Debug and Performance → System ID Peripheral Intel FPGA IP



	s1	Avalon Memory Mapped Slave	Double-click to export
	reset1	Reset Input	Double-click to export
☑	⊟ **jtag_uart_0**	JTAG UART Intel FPGA IP	
	clk	Clock Input	Double-click to export
	reset	Reset Input	Double-click to export
	avalon_jtag_slave	Avalon Memory Mapped Slave	Double-click to export
	irq	Interrupt Sender	Double-click to export
☑	⊟ **timer_0**	Interval Timer Intel FPGA IP	
	clk	Clock Input	Double-click to export
	reset	Reset Input	Double-click to export
	s1	Avalon Memory Mapped Slave	Double-click to export
	irq	Interrupt Sender	Double-click to export
☑	⊟ **sysid_qsys_0**	System ID Peripheral Intel FPGA IP	
	clk	Clock Input	Double-click to export
	reset	Reset Input	Double-click to export
	control_slave	Avalon Memory Mapped Slave	Double-click to export

NIOS II Pixel Display

- Connect up basic NIOS system
  - PLL Inputs



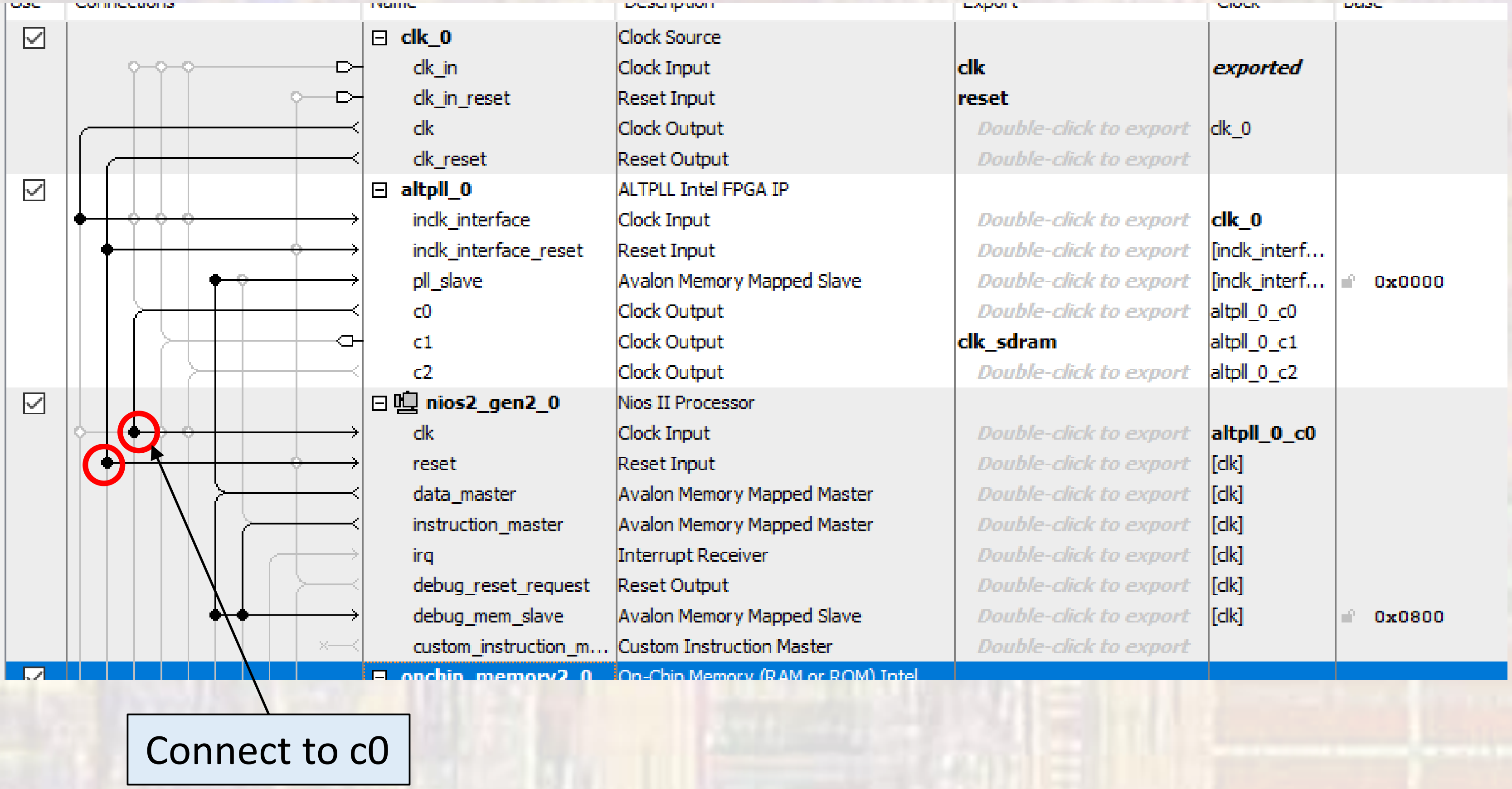
clk_0	Clock Source			
clk_in	Clock Input	clk	*exported*	
clk_in_reset	Reset Input	reset		
clk	Clock Output	*Double-click to export*		
clk_reset	Reset Output	*Double-click to export*		
altpll_0	ALTPLL Intel FPGA IP			
inclk_interface	Clock Input	*Double-click to export*	clk_0	
inclk_interface_reset	Reset Input	*Double-click to export*	[inclk_interf...	
pll_slave	Avalon Memory Mapped Slave	*Double-click to export*	[inclk_interf...	0x0000
c0	Clock Output	*Double-click to export*	altpll_0_c0	
c1	Clock Output	clk_sdram	altpll_0_c1	
c2	Clock Output	*Double-click to export*	altpll_0_c2	
nios2_gen2_0	Nios II Processor			
clk	Clock Input	*Double-click to export*	*unconnected*	
reset	Reset Input	*Double-click to export*	[clk]	
data_master	Avalon Memory Mapped Master	*Double-click to export*	[clk]	
instruction_master	Avalon Memory Mapped Master	*Double-click to export*	[clk]	
irq	Interrupt Receiver	*Double-click to export*	[clk]	
debug_reset_request	Reset Output	*Double-click to export*	[clk]	
debug_mem_slave	Avalon Memory Mapped Slave	*Double-click to export*	[clk]	0x0800
custom_instruction_m	Custom Instruction Master	*Double-click to export*		

Export and rename clk_sdram

Connect to the data_master

NIOS II Pixel Display

- Connect up basic NIOS system
  - NIOS Inputs

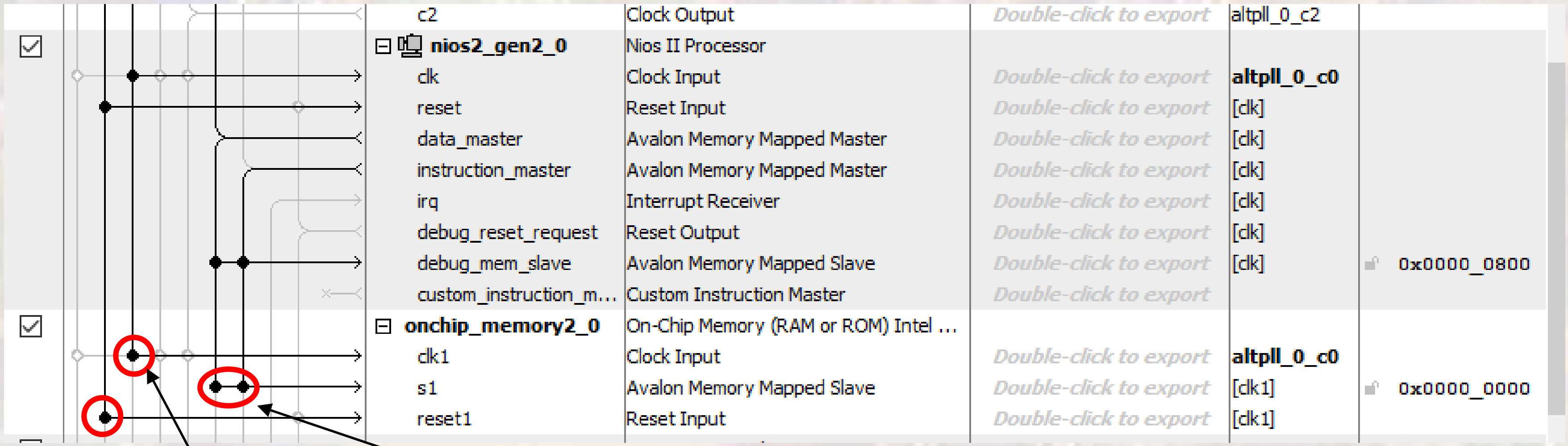


✓	⊟ **clk_0**	Clock Source			
	clk_in	Clock Input	**clk**	***exported***	
	clk_in_reset	Reset Input	**reset**		
	clk	Clock Output	*Double-click to export*	clk_0	
	clk_reset	Reset Output	*Double-click to export*		
✓	⊟ **altpll_0**	ALTPLL Intel FPGA IP			
	inclk_interface	Clock Input	*Double-click to export*	**clk_0**	
	inclk_interface_reset	Reset Input	*Double-click to export*	[inclk_interf...	
	pll_slave	Avalon Memory Mapped Slave	*Double-click to export*	[inclk_interf...	0x0000
	c0	Clock Output	*Double-click to export*	altpll_0_c0	
	c1	Clock Output	**clk_sdram**	altpll_0_c1	
	c2	Clock Output	*Double-click to export*	altpll_0_c2	
✓	⊟ **nios2_gen2_0**	Nios II Processor			
	clk	Clock Input	*Double-click to export*	**altpll_0_c0**	
	reset	Reset Input	*Double-click to export*	[clk]	
	data_master	Avalon Memory Mapped Master	*Double-click to export*	[clk]	
	instruction_master	Avalon Memory Mapped Master	*Double-click to export*	[clk]	
	irq	Interrupt Receiver	*Double-click to export*	[clk]	
	debug_reset_request	Reset Output	*Double-click to export*	[clk]	
	debug_mem_slave	Avalon Memory Mapped Slave	*Double-click to export*	[clk]	0x0800
	custom_instruction_m...	Custom Instruction Master	*Double-click to export*		
✓	⊟ onchip_memory2_0	On-Chip Memory (RAM or ROM) Intel			

Connect to c0

NIOS II Pixel Display

- Connect up basic NIOS system
  - On-chip Memory



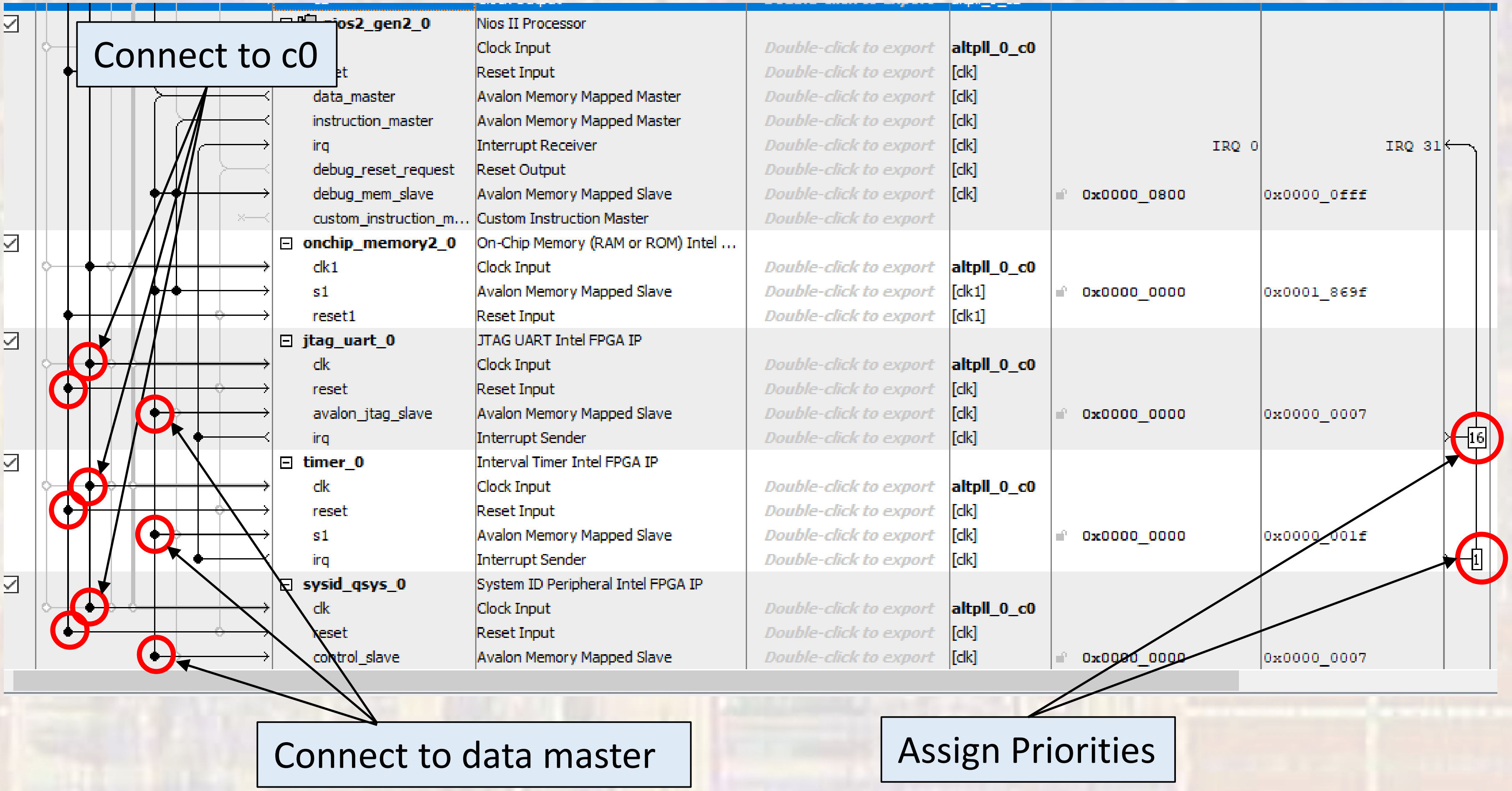
c2	Clock Output	Double-click to export	altpll_0_c2	
⊟ nios2_gen2_0	Nios II Processor			
clk	Clock Input	Double-click to export	**altpll_0_c0**	
reset	Reset Input	Double-click to export	[clk]	
data_master	Avalon Memory Mapped Master	Double-click to export	[clk]	
instruction_master	Avalon Memory Mapped Master	Double-click to export	[clk]	
irq	Interrupt Receiver	Double-click to export	[clk]	
debug_reset_request	Reset Output	Double-click to export	[clk]	
debug_mem_slave	Avalon Memory Mapped Slave	Double-click to export	[clk]	0x0000_0800
custom_instruction_m...	Custom Instruction Master	Double-click to export		
⊟ onchip_memory2_0	On-Chip Memory (RAM or ROM) Intel ...			
clk1	Clock Input	Double-click to export	**altpll_0_c0**	
s1	Avalon Memory Mapped Slave	Double-click to export	[clk1]	0x0000_0000
reset1	Reset Input	Double-click to export	[clk1]	

Connect to data and instruction masters

Connect to c0

NIOS II Pixel Display

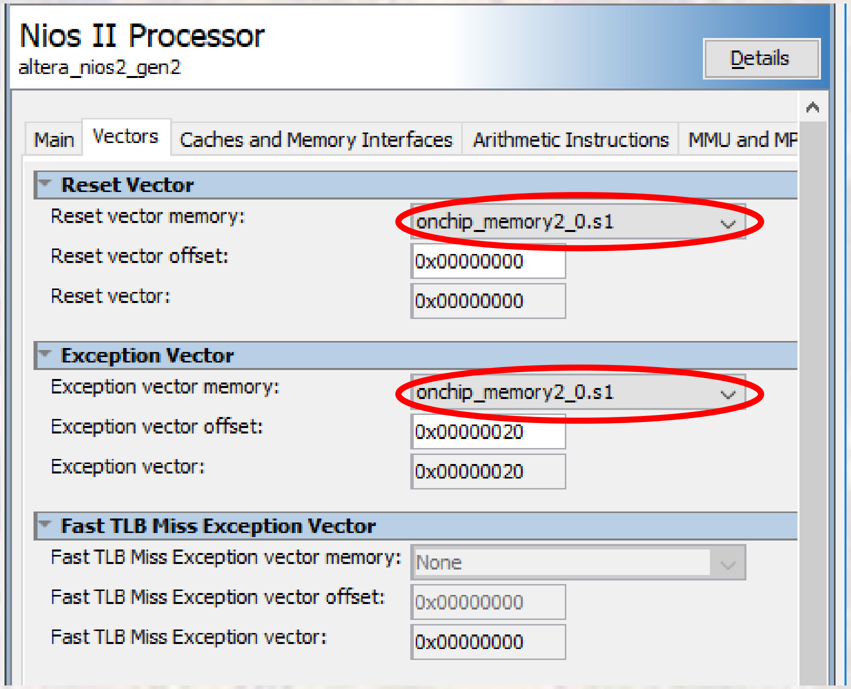
- Connect up basic NIOS system
  - JTAG, Timer, SysID



nios2_gen2_0	Nios II Processor					
	Clock Input	Double-click to export	altpll_0_c0			
et	Reset Input	Double-click to export	[clk]			
data_master	Avalon Memory Mapped Master	Double-click to export	[clk]			
instruction_master	Avalon Memory Mapped Master	Double-click to export	[clk]			
irq	Interrupt Receiver	Double-click to export	[clk]	IRQ 0		IRQ 31
debug_reset_request	Reset Output	Double-click to export	[clk]			
debug_mem_slave	Avalon Memory Mapped Slave	Double-click to export	[clk]	0x0000_0800	0x0000_0fff	
custom_instruction_m...	Custom Instruction Master	Double-click to export				
onchip_memory2_0	On-Chip Memory (RAM or ROM) Intel ...					
clk1	Clock Input	Double-click to export	altpll_0_c0			
s1	Avalon Memory Mapped Slave	Double-click to export	[clk1]	0x0000_0000	0x0001_869f	
reset1	Reset Input	Double-click to export	[clk1]			
jtag_uart_0	JTAG UART Intel FPGA IP					
clk	Clock Input	Double-click to export	altpll_0_c0			
reset	Reset Input	Double-click to export	[clk]			
avalon_jtag_slave	Avalon Memory Mapped Slave	Double-click to export	[clk]	0x0000_0000	0x0000_0007	
irq	Interrupt Sender	Double-click to export	[clk]			16
timer_0	Interval Timer Intel FPGA IP					
clk	Clock Input	Double-click to export	altpll_0_c0			
reset	Reset Input	Double-click to export	[clk]			
s1	Avalon Memory Mapped Slave	Double-click to export	[clk]	0x0000_0000	0x0000_001f	
irq	Interrupt Sender	Double-click to export	[clk]			1
sysid_qsys_0	System ID Peripheral Intel FPGA IP					
clk	Clock Input	Double-click to export	altpll_0_c0			
reset	Reset Input	Double-click to export	[clk]			
control_slave	Avalon Memory Mapped Slave	Double-click to export	[clk]	0x0000_0000	0x0000_0007	

NIOS II Pixel Display

- Connect up basic NIOS system
  - Assign the NIOS II Reset and Exception vectors



Reset Vector

Reset vector memory:	onchip_memory2_0.s1
Reset vector offset:	0x00000000
Reset vector:	0x00000000

Exception Vector

Exception vector memory:	onchip_memory2_0.s1
Exception vector offset:	0x00000020
Exception vector:	0x00000020

Fast TLB Miss Exception Vector

Fast TLB Miss Exception vector memory:	None
Fast TLB Miss Exception vector offset:	0x00000000
Fast TLB Miss Exception vector:	0x00000000

NIOS II Pixel Display

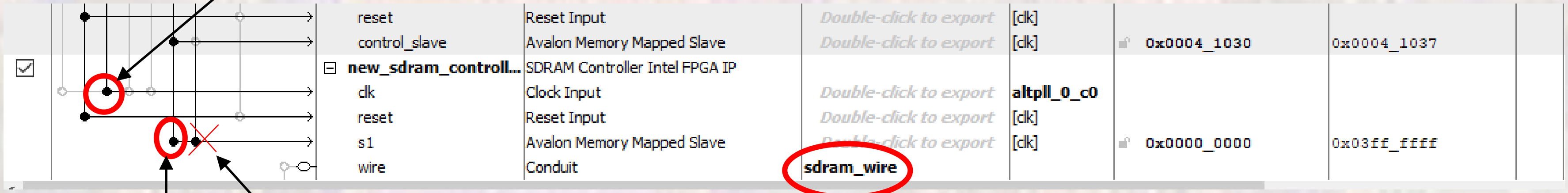
- Create Video System
  - SDRAM Controller
    - Library → Memory Interfaces and Controllers → SDRAM → SDRAM Controller

Memory Profile	Timing
16 bits	CAS = 3
1 chip select	2 initialization refresh cycles
4 banks	7.8125 us refresh command
13 rows	100us delay after pu
10 columns	70ns refresh duration
	20ns pre-charge duration
	20 ns Active R/W delay
	5.5ns access time
	14ns write recovery time

NIOS II Pixel Display

- Create Video System
  - Connect SDRAM Controller

Connect to c0



reset	Reset Input	Double-click to export	[clk]		
control_slave	Avalon Memory Mapped Slave	Double-click to export	[clk]	0x0004_1030	0x0004_1037
new_sdram_controll...	SDRAM Controller Intel FPGA IP				
clk	Clock Input	Double-click to export	**altpll_0_c0**		
reset	Reset Input	Double-click to export	[clk]		
s1	Avalon Memory Mapped Slave	Double-click to export	[clk]	0x0000_0000	0x03ff_ffff
wire	Conduit	**sdram_wire**			

Do not connect to the Instruction master

Connect to data master

Export and Rename

NIOS II Pixel Display

0x01000000 → 16.7MBytes for the program

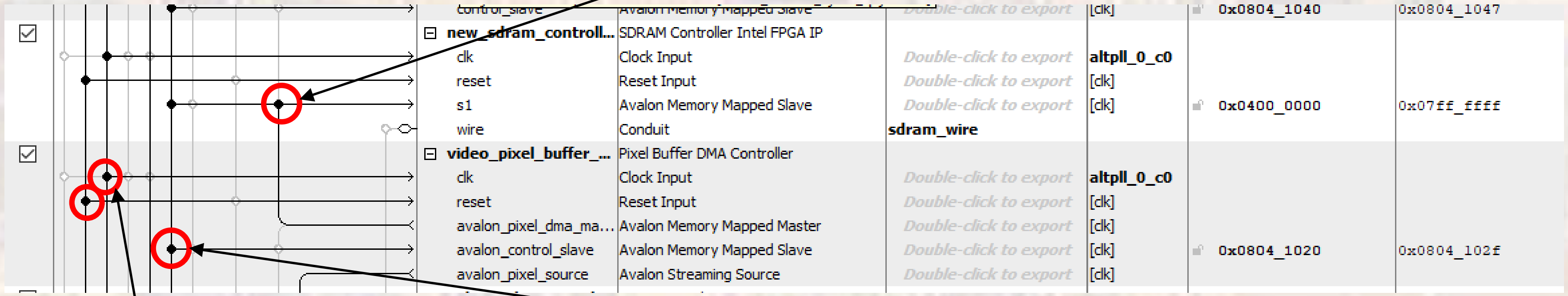
16bits x 320pixels x 240pixels
→ 1,228,800 bits → 153,600 bytes
→ 0x25800 bytes per screen buffer

- Create Video System
  - Pixel Buffer DMA Controller
    - University Program → Audio and Video → Video → Pixel Buffer DMA Controller

x-y mode	Width – 320
Default Buffer Start – 0x01000000	Height – 240
Default Back Buffer Start – 0x01100000	Color space – 16bit RGB

Depends on the final memory map

Connect to SDRAM_Controller MM Slave



	control_slave	Avalon Memory Mapped Slave	Double-click to export	[clk]	0x0804_1040	0x0804_1047
☑	⊟ new_sdram_controll...	SDRAM Controller Intel FPGA IP				
	clk	Clock Input	Double-click to export	altpll_0_c0		
	reset	Reset Input	Double-click to export	[clk]		
	s1	Avalon Memory Mapped Slave	Double-click to export	[clk]	0x0400_0000	0x07ff_ffff
	wire	Conduit	sdram_wire			
☑	⊟ video_pixel_buffer_...	Pixel Buffer DMA Controller				
	clk	Clock Input	Double-click to export	altpll_0_c0		
	reset	Reset Input	Double-click to export	[clk]		
	avalon_pixel_dma_ma...	Avalon Memory Mapped Master	Double-click to export	[clk]		
	avalon_control_slave	Avalon Memory Mapped Slave	Double-click to export	[clk]	0x0804_1020	0x0804_102f
	avalon_pixel_source	Avalon Streaming Source	Double-click to export	[clk]		

Connect to c0

Connect to data master

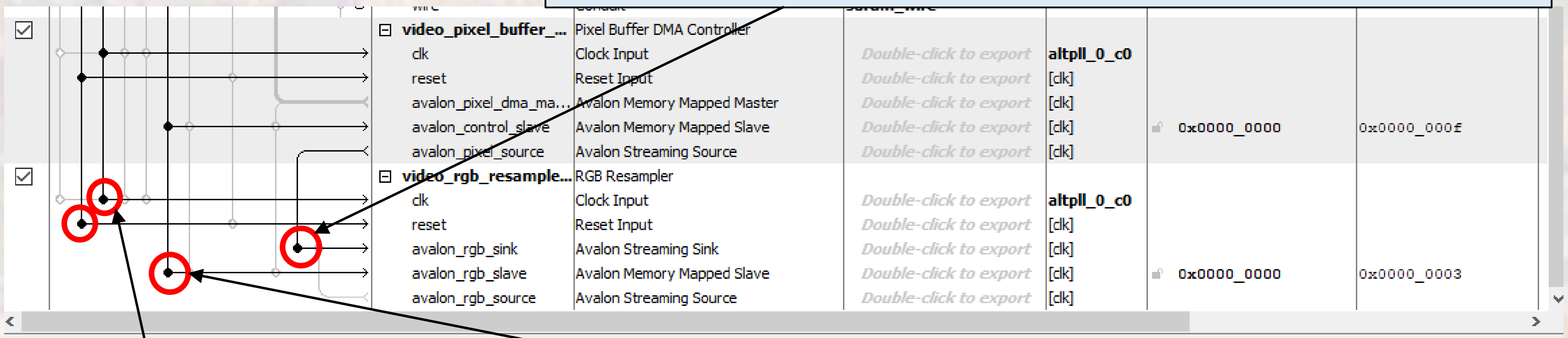
NIOS II Pixel Display

- Create Video System
  - RGB Resampler
    - University Program → Audio and Video → Video → RGB Resampler

16 bit RGB incoming

30 bit RGB outgoing

Connect to video_pixel_buffer_dma streaming source



☑	⊟ **video_pixel_buffer_...**	Pixel Buffer DMA Controller				
	clk	Clock Input	*Double-click to export*	**altpll_0_c0**		
	reset	Reset Input	*Double-click to export*	[clk]		
	avalon_pixel_dma_ma...	Avalon Memory Mapped Master	*Double-click to export*	[clk]		
	avalon_control_slave	Avalon Memory Mapped Slave	*Double-click to export*	[clk]	0x0000_0000	0x0000_000f
	avalon_pixel_source	Avalon Streaming Source	*Double-click to export*	[clk]		
☑	⊟ **video_rgb_resample...**	RGB Resampler				
	clk	Clock Input	*Double-click to export*	**altpll_0_c0**		
	reset	Reset Input	*Double-click to export*	[clk]		
	avalon_rgb_sink	Avalon Streaming Sink	*Double-click to export*	[clk]		
	avalon_rgb_slave	Avalon Memory Mapped Slave	*Double-click to export*	[clk]	0x0000_0000	0x0000_0003
	avalon_rgb_source	Avalon Streaming Source	*Double-click to export*	[clk]		

Connect to c0

Connect to data master

NIOS II Pixel Display

- Create Video System
  - RGB Scaler
    - University Program → Audio and Video → Video → Scaler

      Width Scaling – 2

      Height Scaling – 2

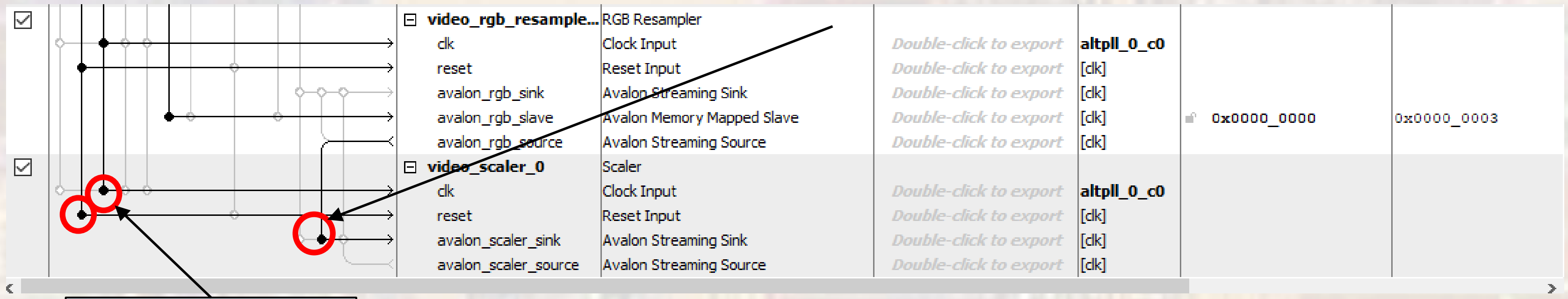
      Width – 320

      Height – 240

      10 bits / symbol

      3 symbols / beat

Connect to rgb_resampler streaming source



⊟ **video_rgb_resample...**	RGB Resampler				
clk	Clock Input	*Double-click to export*	**altpll_0_c0**		
reset	Reset Input	*Double-click to export*	[clk]		
avalon_rgb_sink	Avalon Streaming Sink	*Double-click to export*	[clk]		
avalon_rgb_slave	Avalon Memory Mapped Slave	*Double-click to export*	[clk]	0x0000_0000	0x0000_0003
avalon_rgb_source	Avalon Streaming Source	*Double-click to export*	[clk]		
⊟ **video_scaler_0**	Scaler				
clk	Clock Input	*Double-click to export*	**altpll_0_c0**		
reset	Reset Input	*Double-click to export*	[clk]		
avalon_scaler_sink	Avalon Streaming Sink	*Double-click to export*	[clk]		
avalon_scaler_source	Avalon Streaming Source	*Double-click to export*	[clk]		

Connect to c0

NIOS II Pixel Display

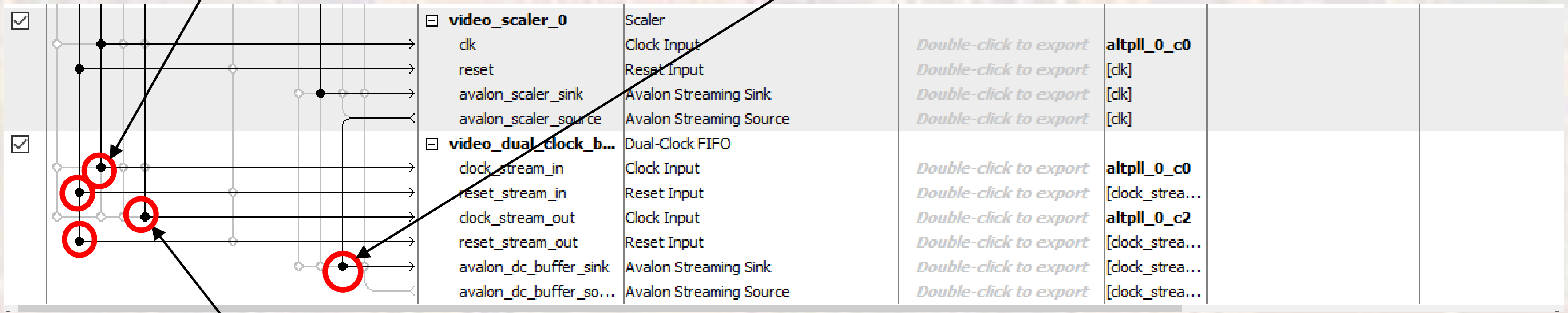
- Create Video System
  - Dual Clock FIFO
    - University Program → Audio and Video → Video → Dual Clock FIFO

      Color Bits – 10

      Color Planes - 3

Connect to c0

Connect to rgb_scaler streaming source



☑ ⊟ **video_scaler_0**	Scaler		
clk	Clock Input	*Double-click to export*	**altpll_0_c0**
reset	Reset Input	*Double-click to export*	[clk]
avalon_scaler_sink	Avalon Streaming Sink	*Double-click to export*	[clk]
avalon_scaler_source	Avalon Streaming Source	*Double-click to export*	[clk]
☑ ⊟ **video_dual_clock_b...**	Dual-Clock FIFO		
clock_stream_in	Clock Input	*Double-click to export*	**altpll_0_c0**
reset_stream_in	Reset Input	*Double-click to export*	[clock_strea...
clock_stream_out	Clock Input	*Double-click to export*	**altpll_0_c2**
reset_stream_out	Reset Input	*Double-click to export*	[clock_strea...
avalon_dc_buffer_sink	Avalon Streaming Sink	*Double-click to export*	[clock_strea...
avalon_dc_buffer_so...	Avalon Streaming Source	*Double-click to export*	[clock_strea...

Connect to c2

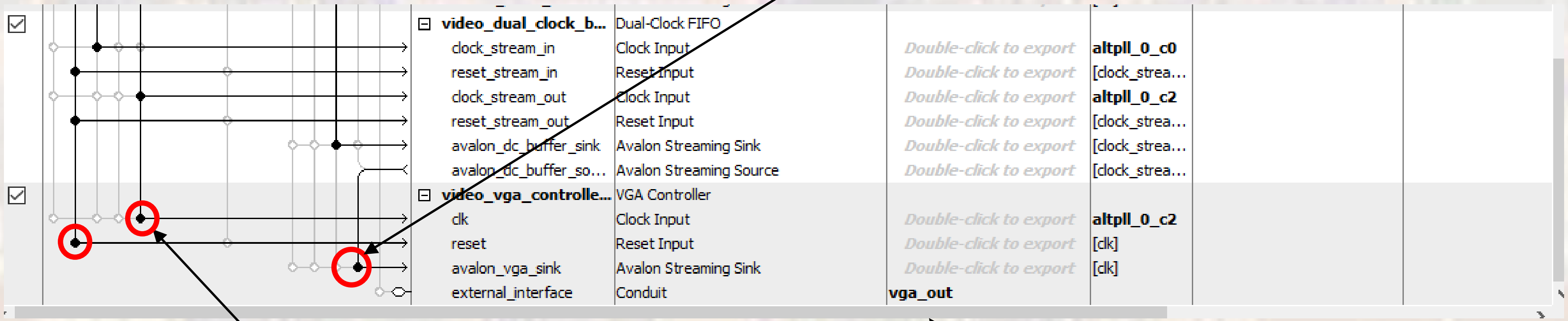
NIOS II Pixel Display

- Create Video System
  - VGA Controller
    - University Program → Audio and Video → Video → VGA Controller

      DE10-Lite

      VGA Connector

      VGA 640x480



video_dual_clock_b...	Dual-Clock FIFO		
clock_stream_in	Clock Input	Double-click to export	altpll_0_c0
reset_stream_in	Reset Input	Double-click to export	[clock_strea...
clock_stream_out	Clock Input	Double-click to export	altpll_0_c2
reset_stream_out	Reset Input	Double-click to export	[clock_strea...
avalon_dc_buffer_sink	Avalon Streaming Sink	Double-click to export	[clock_strea...
avalon_dc_buffer_so...	Avalon Streaming Source	Double-click to export	[clock_strea...
video_vga_controlle...	VGA Controller		
clk	Clock Input	Double-click to export	altpll_0_c2
reset	Reset Input	Double-click to export	[clk]
avalon_vga_sink	Avalon Streaming Sink	Double-click to export	[clk]
external_interface	Conduit	vga_out	

Connect to dual_clock_buffer streaming source

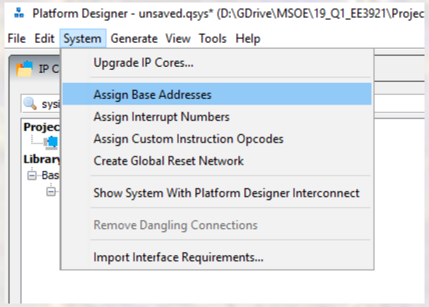
Connect to c2

Export and rename

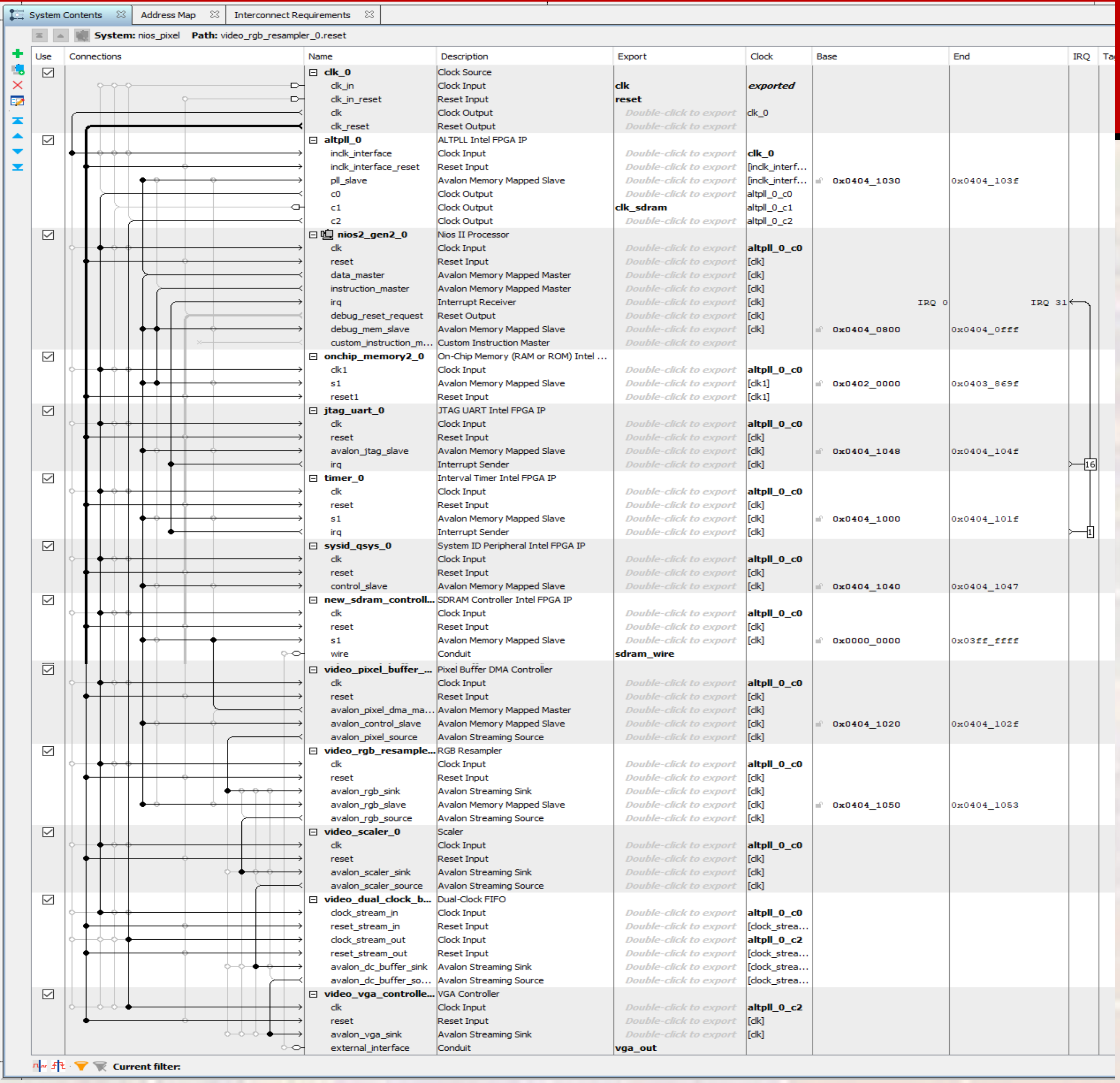
NIOS II P

• Create Vic

Assign Base Addresses


Platform Designer - unsaved.qsys* (D:\GDrive\MSOE\19_Q1_EE3921\Projec

File
Edit
System
Generate
View
Tools
Help
Upgrade IP Cores...
Assign Base Addresses
Assign Interrupt Numbers
Assign Custom Instruction Opcodes
Create Global Reset Network
Show System With Platform Designer Interconnect
Remove Dangling Connections
Import Interface Requirements...



System: nios_pixel Path: video_rgb_resampler_0.reset

Use	Name	Description	Export	Clock	Base	End	IRQ
✓	clk_0	Clock Source					
	clk_in	Clock Input	clk	exported			
	clk_in_reset	Reset Input	reset				
	clk	Clock Output	Double-click to export	clk_0			
	clk_reset	Reset Output	Double-click to export				
✓	altpll_0	ALTPLL Intel FPGA IP					
	inclk_interface	Clock Input	Double-click to export	clk_0			
	inclk_interface_reset	Reset Input	Double-click to export	[inclk_interf...			
	pll_slave	Avalon Memory Mapped Slave	Double-click to export	[inclk_interf...	0x0404_1030	0x0404_103f	
	c0	Clock Output	Double-click to export	altpll_0_c0			
	c1	Clock Output	clk_sdram	altpll_0_c1			
	c2	Clock Output	Double-click to export	altpll_0_c2			
✓	nios2_gen2_0	Nios II Processor					
	clk	Clock Input	Double-click to export	altpll_0_c0			
	reset	Reset Input	Double-click to export	[clk]			
	data_master	Avalon Memory Mapped Master	Double-click to export	[clk]			
	instruction_master	Avalon Memory Mapped Master	Double-click to export	[clk]			
	irq	Interrupt Receiver	Double-click to export	[clk]	IRQ 0	IRQ 31	
	debug_reset_request	Reset Output	Double-click to export	[clk]			
	debug_mem_slave	Avalon Memory Mapped Slave	Double-click to export	[clk]	0x0404_0800	0x0404_0fff	
	custom_instruction_m...	Custom Instruction Master	Double-click to export				
✓	onchip_memory2_0	On-Chip Memory (RAM or ROM) Intel ...					
	clk1	Clock Input	Double-click to export	altpll_0_c0			
	s1	Avalon Memory Mapped Slave	Double-click to export	[clk1]	0x0402_0000	0x0403_869f	
	reset1	Reset Input	Double-click to export	[clk1]			
✓	jtag_uart_0	JTAG UART Intel FPGA IP					
	clk	Clock Input	Double-click to export	altpll_0_c0			
	reset	Reset Input	Double-click to export	[clk]			
	avalon_jtag_slave	Avalon Memory Mapped Slave	Double-click to export	[clk]	0x0404_1048	0x0404_104f	
	irq	Interrupt Sender	Double-click to export	[clk]			16
✓	timer_0	Interval Timer Intel FPGA IP					
	clk	Clock Input	Double-click to export	altpll_0_c0			
	reset	Reset Input	Double-click to export	[clk]			
	s1	Avalon Memory Mapped Slave	Double-click to export	[clk]	0x0404_1000	0x0404_101f	
	irq	Interrupt Sender	Double-click to export	[clk]			1
✓	sysid_qsys_0	System ID Peripheral Intel FPGA IP					
	clk	Clock Input	Double-click to export	altpll_0_c0			
	reset	Reset Input	Double-click to export	[clk]			
	control_slave	Avalon Memory Mapped Slave	Double-click to export	[clk]	0x0404_1040	0x0404_1047	
✓	new_sdram_controll...	SDRAM Controller Intel FPGA IP					
	clk	Clock Input	Double-click to export	altpll_0_c0			
	reset	Reset Input	Double-click to export	[clk]			
	s1	Avalon Memory Mapped Slave	Double-click to export	[clk]	0x0000_0000	0x03ff_ffff	
	wire	Conduit	sdram_wire				
✓	video_pixel_buffer_...	Pixel Buffer DMA Controller					
	clk	Clock Input	Double-click to export	altpll_0_c0			
	reset	Reset Input	Double-click to export	[clk]			
	avalon_pixel_dma_ma...	Avalon Memory Mapped Master	Double-click to export	[clk]			
	avalon_control_slave	Avalon Memory Mapped Slave	Double-click to export	[clk]	0x0404_1020	0x0404_102f	
	avalon_pixel_source	Avalon Streaming Source	Double-click to export	[clk]			
✓	video_rgb_resample...	RGB Resampler					
	clk	Clock Input	Double-click to export	altpll_0_c0			
	reset	Reset Input	Double-click to export	[clk]			
	avalon_rgb_sink	Avalon Streaming Sink	Double-click to export	[clk]			
	avalon_rgb_slave	Avalon Memory Mapped Slave	Double-click to export	[clk]	0x0404_1050	0x0404_1053	
	avalon_rgb_source	Avalon Streaming Source	Double-click to export	[clk]			
✓	video_scaler_0	Scaler					
	clk	Clock Input	Double-click to export	altpll_0_c0			
	reset	Reset Input	Double-click to export	[clk]			
	avalon_scaler_sink	Avalon Streaming Sink	Double-click to export	[clk]			
	avalon_scaler_source	Avalon Streaming Source	Double-click to export	[clk]			
✓	video_dual_clock_b...	Dual-Clock FIFO					
	clock_stream_in	Clock Input	Double-click to export	altpll_0_c0			
	reset_stream_in	Reset Input	Double-click to export	[clock_strea...			
	clock_stream_out	Clock Input	Double-click to export	altpll_0_c2			
	reset_stream_out	Reset Input	Double-click to export	[clock_strea...			
	avalon_dc_buffer_sink	Avalon Streaming Sink	Double-click to export	[clock_strea...			
	avalon_dc_buffer_so...	Avalon Streaming Source	Double-click to export	[clock_strea...			
✓	video_vga_controlle...	VGA Controller					
	clk	Clock Input	Double-click to export	altpll_0_c2			
	reset	Reset Input	Double-click to export	[clk]			
	avalon_vga_sink	Avalon Streaming Sink	Double-click to export	[clk]			
	external_interface	Conduit	vga_out				

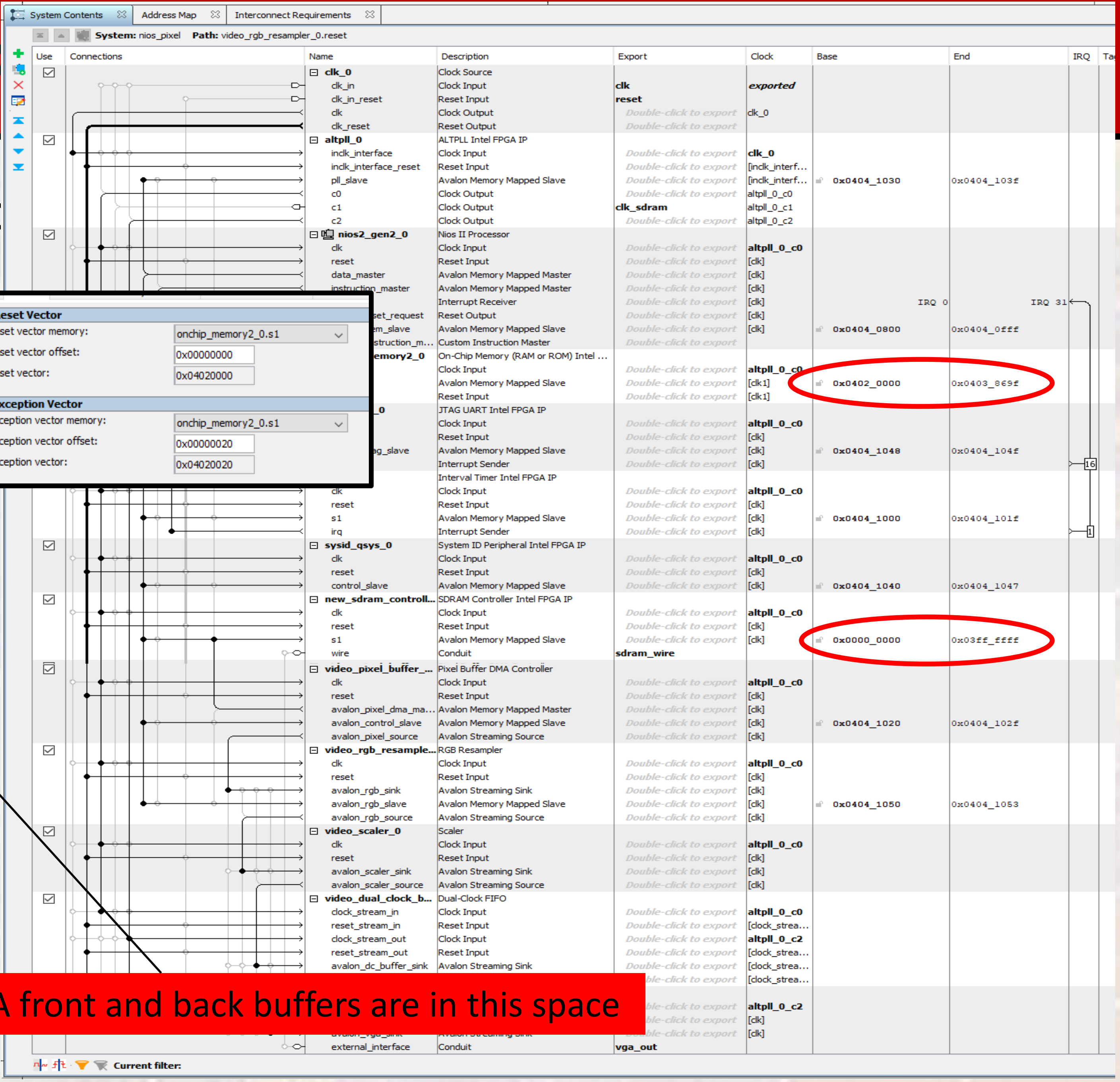
NIOS II P

- Create Vic

On-chip Memory
0x04020000 – 0x0403869F
CPU automatically updated

SDRAM
0x00000000 – 0x03FFFFFF

Peripherals
0x04041000 – 0x04041053



System: nios_pixel **Path:** video_rgb_resampler_0.reset

Reset Vector

Reset vector memory: onchip_memory2_0.s1
Reset vector offset: 0x00000000
Reset vector: 0x04020000

Exception Vector

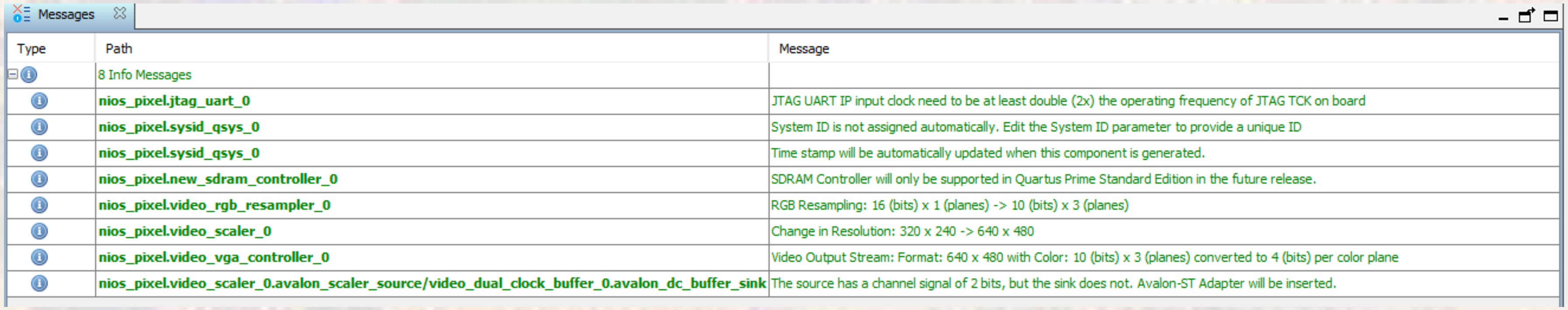
Exception vector memory: onchip_memory2_0.s1
Exception vector offset: 0x00000020
Exception vector: 0x04020020

Name	Description	Export	Clock	Base	End	IRQ
clk_0	Clock Source					
clk_in	Clock Input	**clk**	*exported*			
clk_in_reset	Reset Input	**reset**				
clk	Clock Output	Double-click to export	clk_0			
clk_reset	Reset Output	Double-click to export				
altpll_0	ALTPLL Intel FPGA IP					
inclk_interface	Clock Input	Double-click to export	**clk_0**			
inclk_interface_reset	Reset Input	Double-click to export	[inclk_interf…			
pll_slave	Avalon Memory Mapped Slave	Double-click to export	[inclk_interf…	0x0404_1030	0x0404_103f	
c0	Clock Output	Double-click to export	altpll_0_c0			
c1	Clock Output	**clk_sdram**	altpll_0_c1			
c2	Clock Output	Double-click to export	altpll_0_c2			
nios2_gen2_0	Nios II Processor					
clk	Clock Input	Double-click to export	**altpll_0_c0**			
reset	Reset Input	Double-click to export	[clk]			
data_master	Avalon Memory Mapped Master	Double-click to export	[clk]			
instruction_master	Avalon Memory Mapped Master	Double-click to export	[clk]			
	Interrupt Receiver	Double-click to export	[clk]	IRQ 0	IRQ 31	
…set_request	Reset Output	Double-click to export	[clk]			
…m_slave	Avalon Memory Mapped Slave	Double-click to export	[clk]	0x0404_0800	0x0404_0fff	
…struction_m…	Custom Instruction Master	Double-click to export				
…emory2_0	On-Chip Memory (RAM or ROM) Intel …					
	Clock Input	Double-click to export	**altpll_0_c0**			
	Avalon Memory Mapped Slave	Double-click to export	[clk1]	0x0402_0000	0x0403_869f	
	Reset Input	Double-click to export	[clk1]			
…0	JTAG UART Intel FPGA IP					
	Clock Input	Double-click to export	**altpll_0_c0**			
	Reset Input	Double-click to export	[clk]			
…ag_slave	Avalon Memory Mapped Slave	Double-click to export	[clk]	0x0404_1048	0x0404_104f	
	Interrupt Sender	Double-click to export	[clk]			16
	Interval Timer Intel FPGA IP					
clk	Clock Input	Double-click to export	**altpll_0_c0**			
reset	Reset Input	Double-click to export	[clk]			
s1	Avalon Memory Mapped Slave	Double-click to export	[clk]	0x0404_1000	0x0404_101f	
irq	Interrupt Sender	Double-click to export	[clk]			1
sysid_qsys_0	System ID Peripheral Intel FPGA IP					
clk	Clock Input	Double-click to export	**altpll_0_c0**			
reset	Reset Input	Double-click to export	[clk]			
control_slave	Avalon Memory Mapped Slave	Double-click to export	[clk]	0x0404_1040	0x0404_1047	
new_sdram_controll…	SDRAM Controller Intel FPGA IP					
clk	Clock Input	Double-click to export	**altpll_0_c0**			
reset	Reset Input	Double-click to export	[clk]			
s1	Avalon Memory Mapped Slave	Double-click to export	[clk]	0x0000_0000	0x03ff_ffff	
wire	Conduit	**sdram_wire**				
video_pixel_buffer_…	Pixel Buffer DMA Controller					
clk	Clock Input	Double-click to export	**altpll_0_c0**			
reset	Reset Input	Double-click to export	[clk]			
avalon_pixel_dma_ma…	Avalon Memory Mapped Master	Double-click to export	[clk]			
avalon_control_slave	Avalon Memory Mapped Slave	Double-click to export	[clk]	0x0404_1020	0x0404_102f	
avalon_pixel_source	Avalon Streaming Source	Double-click to export	[clk]			
video_rgb_resample…	RGB Resampler					
clk	Clock Input	Double-click to export	**altpll_0_c0**			
reset	Reset Input	Double-click to export	[clk]			
avalon_rgb_sink	Avalon Streaming Sink	Double-click to export	[clk]			
avalon_rgb_slave	Avalon Memory Mapped Slave	Double-click to export	[clk]	0x0404_1050	0x0404_1053	
avalon_rgb_source	Avalon Streaming Source	Double-click to export	[clk]			
video_scaler_0	Scaler					
clk	Clock Input	Double-click to export	**altpll_0_c0**			
reset	Reset Input	Double-click to export	[clk]			
avalon_scaler_sink	Avalon Streaming Sink	Double-click to export	[clk]			
avalon_scaler_source	Avalon Streaming Source	Double-click to export	[clk]			
video_dual_clock_b…	Dual-Clock FIFO					
clock_stream_in	Clock Input	Double-click to export	**altpll_0_c0**			
reset_stream_in	Reset Input	Double-click to export	[clock_strea…			
clock_stream_out	Clock Input	Double-click to export	**altpll_0_c2**			
reset_stream_out	Reset Input	Double-click to export	[clock_strea…			
avalon_dc_buffer_sink	Avalon Streaming Sink	Double-click to export	[clock_strea…			
		…ble-click to export	[clock_strea…			
		…ble-click to export	**altpll_0_c2**			
		…ble-click to export	[clk]			
		…ble-click to export	[clk]			
external_interface	Conduit	**vga_out**				

Verify that the DMA front and back buffers are in this space

NIOS II Pixel Display

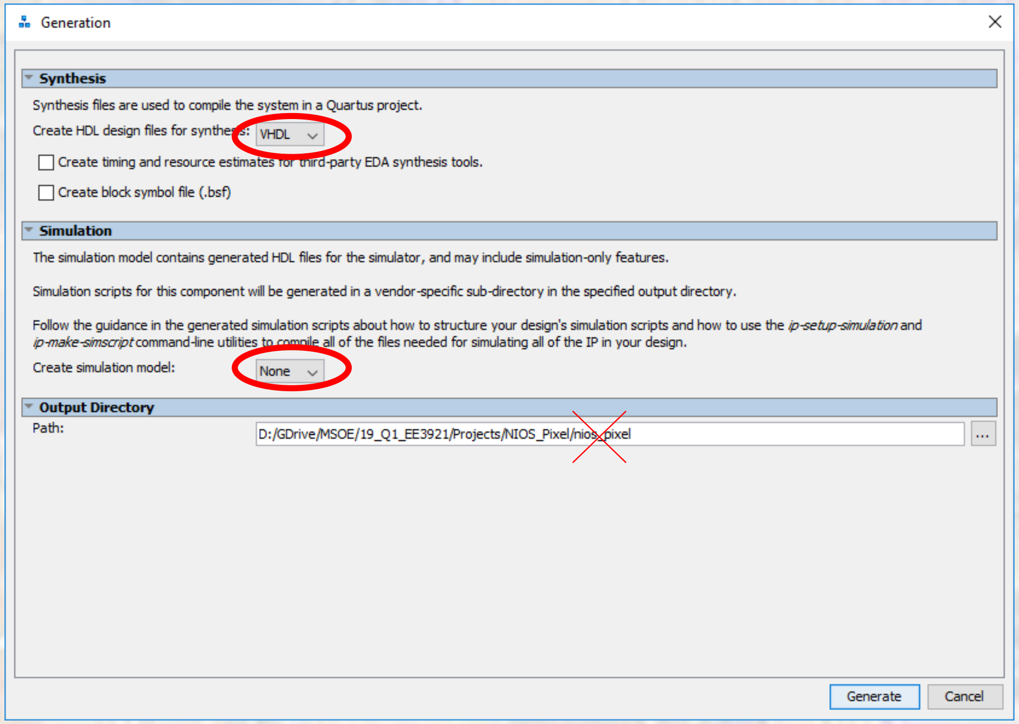
- Create Video System
  - Check for errors



Type	Path	Message
⊟ ⓘ	8 Info Messages	
ⓘ	nios_pixel.jtag_uart_0	JTAG UART IP input clock need to be at least double (2x) the operating frequency of JTAG TCK on board
ⓘ	nios_pixel.sysid_qsys_0	System ID is not assigned automatically. Edit the System ID parameter to provide a unique ID
ⓘ	nios_pixel.sysid_qsys_0	Time stamp will be automatically updated when this component is generated.
ⓘ	nios_pixel.new_sdram_controller_0	SDRAM Controller will only be supported in Quartus Prime Standard Edition in the future release.
ⓘ	nios_pixel.video_rgb_resampler_0	RGB Resampling: 16 (bits) x 1 (planes) -> 10 (bits) x 3 (planes)
ⓘ	nios_pixel.video_scaler_0	Change in Resolution: 320 x 240 -> 640 x 480
ⓘ	nios_pixel.video_vga_controller_0	Video Output Stream: Format: 640 x 480 with Color: 10 (bits) x 3 (planes) converted to 4 (bits) per color plane
ⓘ	nios_pixel.video_scaler_0.avalon_scaler_source/video_dual_clock_buffer_0.avalon_dc_buffer_sink	The source has a channel signal of 2 bits, but the sink does not. Avalon-ST Adapter will be inserted.

NIOS II Pixel Display

- Create Video System
  - Save the Platform Designer system
  - Generate the Platform Designer system
    - The first time you generate you must delete the last directory in the path – don't use the '…'



Generation

Synthesis

Synthesis files are used to compile the system in a Quartus project.

Create HDL design files for synthesis: VHDL

☐ Create timing and resource estimates for third-party EDA synthesis tools.

☐ Create block symbol file (.bsf)

Simulation

The simulation model contains generated HDL files for the simulator, and may include simulation-only features.

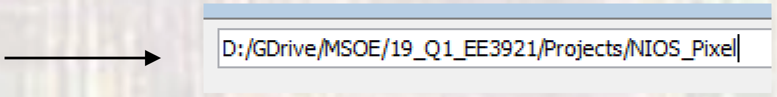
Simulation scripts for this component will be generated in a vendor-specific sub-directory in the specified output directory.

Follow the guidance in the generated simulation scripts about how to structure your design's simulation scripts and how to use the *ip-setup-simulation* and *ip-make-simscript* command-line utilities to compile all of the files needed for simulating all of the IP in your design.

Create simulation model: None

Output Directory

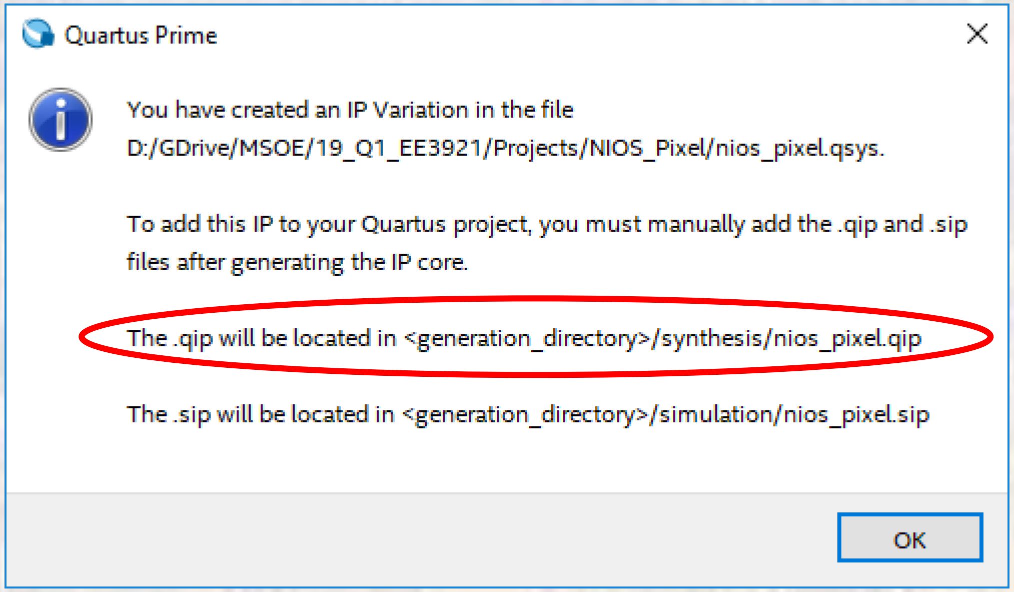
Path: D:/GDrive/MSOE/19_Q1_EE3921/Projects/NIOS_Pixel/~~nios_pixel~~



D:/GDrive/MSOE/19_Q1_EE3921/Projects/NIOS_Pixel

NIOS II Pixel Display

- Create Video System
  - Add the .qip file to the project



You have created an IP Variation in the file D:/GDrive/MSOE/19_Q1_EE3921/Projects/NIOS_Pixel/nios_pixel.qsys.

To add this IP to your Quartus project, you must manually add the .qip and .sip files after generating the IP core.

The .qip will be located in <generation_directory>/synthesis/nios_pixel.qip

The .sip will be located in <generation_directory>/simulation/nios_pixel.sip

NIOS II Pixel Display

- Create DE10 Design
  - Instantiate into a VHDL file
    - Open a new VHDL design
    - In Platform Designer: Generate → Show Instantiation Template
    - Copy and Paste into the new design where appropriate

```
component nios_pixel is
  port (
    clk_clk        : in   std_logic                     := 'X';          -- clk
    clk_sdram_clk    : out  std_logic;                                    -- clk
    reset_reset_n    : in   std_logic                     := 'X';          -- reset_n
    sdram_wire_addr  : out  std_logic_vector(12 downto 0);                     -- addr
    sdram_wire_ba    : out  std_logic_vector(1 downto 0);                      -- ba
    sdram_wire_cas_n : out  std_logic;                                    -- cas_n
    sdram_wire_cke   : out  std_logic;                                    -- cke
    sdram_wire_cs_n  : out  std_logic;                                    -- cs_n
    sdram_wire_dq    : inout std_logic_vector(15 downto 0) := (others => 'X'); -- dq
    sdram_wire_dqm   : out  std_logic_vector(1 downto 0);                      -- dqm
    sdram_wire_ras_n : out  std_logic;                                    -- ras_n
    sdram_wire_we_n  : out  std_logic;                                    -- we_n
    vga_out_CLK      : out  std_logic;                                    -- CLK
    vga_out_HS       : out  std_logic;                                    -- HS
    vga_out_VS       : out  std_logic;                                    -- VS
    vga_out_BLANK    : out  std_logic;                                    -- BLANK
    vga_out_SYNC     : out  std_logic;                                    -- SYNC
    vga_out_R        : out  std_logic_vector(3 downto 0);                      -- R
    vga_out_G        : out  std_logic_vector(3 downto 0);                      -- G
    vga_out_B        : out  std_logic_vector(3 downto 0)                       -- B
  );
end component nios_pixel;
```

```
u0 : component nios_pixel
  port map (
    clk_clk          => CONNECTED_TO_clk_clk,          --      clk.clk
    clk_sdram_clk    => CONNECTED_TO_clk_sdram_clk,    -- clk_sdram.clk
    reset_reset_n    => CONNECTED_TO_reset_reset_n,    --    reset.reset_n
    sdram_wire_addr  => CONNECTED_TO_sdram_wire_addr,  -- sdram_wire.addr
    sdram_wire_ba    => CONNECTED_TO_sdram_wire_ba,    --         .ba
    sdram_wire_cas_n => CONNECTED_TO_sdram_wire_cas_n, --         .cas_n
    sdram_wire_cke   => CONNECTED_TO_sdram_wire_cke,   --         .cke
    sdram_wire_cs_n  => CONNECTED_TO_sdram_wire_cs_n,  --         .cs_n
    sdram_wire_dq    => CONNECTED_TO_sdram_wire_dq,    --         .dq
    sdram_wire_dqm   => CONNECTED_TO_sdram_wire_dqm,   --         .dqm
    sdram_wire_ras_n => CONNECTED_TO_sdram_wire_ras_n, --         .ras_n
    sdram_wire_we_n  => CONNECTED_TO_sdram_wire_we_n,  --         .we_n
    vga_out_CLK      => CONNECTED_TO_vga_out_CLK,      --  vga_out.CLK
    vga_out_HS       => CONNECTED_TO_vga_out_HS,       --         .HS
    vga_out_VS       => CONNECTED_TO_vga_out_VS,       --         .VS
    vga_out_BLANK    => CONNECTED_TO_vga_out_BLANK,    --         .BLANK
    vga_out_SYNC     => CONNECTED_TO_vga_out_SYNC,     --         .SYNC
    vga_out_R        => CONNECTED_TO_vga_out_R,        --         .R
    vga_out_G        => CONNECTED_TO_vga_out_G,        --         .G
    vga_out_B        => CONNECTED_TO_vga_out_B         --         .B
  );
```

NIOS II Pixel Display

- Create DE10 Design
  - Instantiate into a VHDL file

Instantiation template component

```
architecture behavioral of nios_pixel_de10 is
    --
    -- no signals

    component nios_pixel is
        port (
            clk_clk              : in    std_logic            := 'X';       -- clk
            clk_sdram_clk        : out   std_logic;                         -- clk
            reset_reset_n        : in    std_logic            := 'X';       -- reset_n
            sdram_wire_addr      : out   std_logic_vector(12 downto 0);              -- addr
            sdram_wire_ba        : out   std_logic_vector(1 downto 0);               -- ba
            sdram_wire_cas_n     : out   std_logic;                         -- cas_n
            sdram_wire_cke       : out   std_logic;                         -- cke
            sdram_wire_cs_n      : out   std_logic;                         -- cs_n
            sdram_wire_dq        : inout std_logic_vector(15 downto 0) := (others => 'X'); -- dq
            sdram_wire_dqm       : out   std_logic_vector(1 downto 0);               -- dqm
            sdram_wire_ras_n     : out   std_logic;                         -- ras_n
            sdram_wire_we_n      : out   std_logic;                         -- we_n
            vga_out_CLK          : out   std_logic;                         -- CLK
            vga_out_HS           : out   std_logic;                         -- HS
            vga_out_VS           : out   std_logic;                         -- VS
            vga_out_BLANK        : out   std_logic;                         -- BLANK
            vga_out_SYNC         : out   std_logic;                         -- SYNC
            vga_out_R            : out   std_logic_vector(3 downto 0);               -- R
            vga_out_G            : out   std_logic_vector(3 downto 0);               -- G
            vga_out_B            : out   std_logic_vector(3 downto 0)                -- B
        );
    end component;
```

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

entity nios_pixel_de10 is
    port(
        CLOCK_50 :      in std_logic;
        DRAM_ADDR :     out std_logic_vector(12 downto 0);
        DRAM_BA :       out std_logic_vector(1 downto 0);
        DRAM_CAS_N:     out std_logic;
        DRAM_CKE:       out std_logic;
        DRAM_CS_N:      out std_logic;
        DRAM_RAS_N:     out std_logic;
        DRAM_WE_N:      out std_logic;
        DRAM_DQ:        inout std_logic_vector(15 downto 0);
        DRAM_UDQM:      out std_logic;
        DRAM_LDQM:      out std_logic;
        VGA_HS:         out std_logic;
        VGA_VS:         out std_logic;
        VGA_R:          out std_logic_vector(3 downto 0);
        VGA_G:          out std_logic_vector(3 downto 0);
        VGA_B:          out std_logic_vector(3 downto 0);
        DRAM_CLK:       out std_logic
    );
end entity;
```

DE10 pin aliases from .qsf file

NIOS II Pixel Display

- Create DE10 Design
  - Instantiate into a VHDL file

Instantiation template instance mapped to DE10 qsf pin aliases

```
begin

    u0 : component nios_pixel
        port map (
            clk_clk                 => CLOCK_50,                      --    clk.clk
            reset_reset_n            => '1',                          --    reset.reset_n
            --vga_out_CLK           => CONNECTED_TO_vga_out_CLK,      --  vga_out.CLK
            vga_out_HS              => VGA_HS,                        --       .HS
            vga_out_VS              => VGA_VS,                        --       .VS
            --vga_out_BLANK         => CONNECTED_TO_vga_out_BLANK,    --       .BLANK
            --vga_out_SYNC          => CONNECTED_TO_vga_out_SYNC,     --       .SYNC
            vga_out_R               => VGA_R,                         --       .R
            vga_out_G               => VGA_G,                         --       .G
            vga_out_B               => VGA_B,                         --       .B
            clk_sdram_clk           => DRAM_CLK,                      --  clk_sdram.clk
            sdram_wire_addr         => DRAM_ADDR,                     --  sdram_wire.addr
            sdram_wire_ba           => DRAM_BA,                       --       .ba
            sdram_wire_cas_n        => DRAM_CAS_N,                    --       .cas_n
            sdram_wire_cke          => DRAM_CKE,                      --       .cke
            sdram_wire_cs_n         => DRAM_CS_N,                     --       .cs_n
            sdram_wire_dq           => DRAM_DQ,                       --       .dq
            sdram_wire_dqm(1)       => DRAM_UDQM,                     --       .dqm
            sdram_wire_dqm(0)       => DRAM_LDQM,                     --       .dqm
            sdram_wire_ras_n        => DRAM_RAS_N,                    --       .ras_n
            sdram_wire_we_n         => DRAM_WE_N                      --       .we_n
        );

end architecture;
```

Note: these 3 signals are not used
- comment out or remove

NIOS II Pixel Display

- Create DE10 Design

  - Prepare to synthesize
    - If you did not do these when you created the project be sure to do them now
      - assignments → device → device and Pin options
        - Single Uncompressed with memory initialization

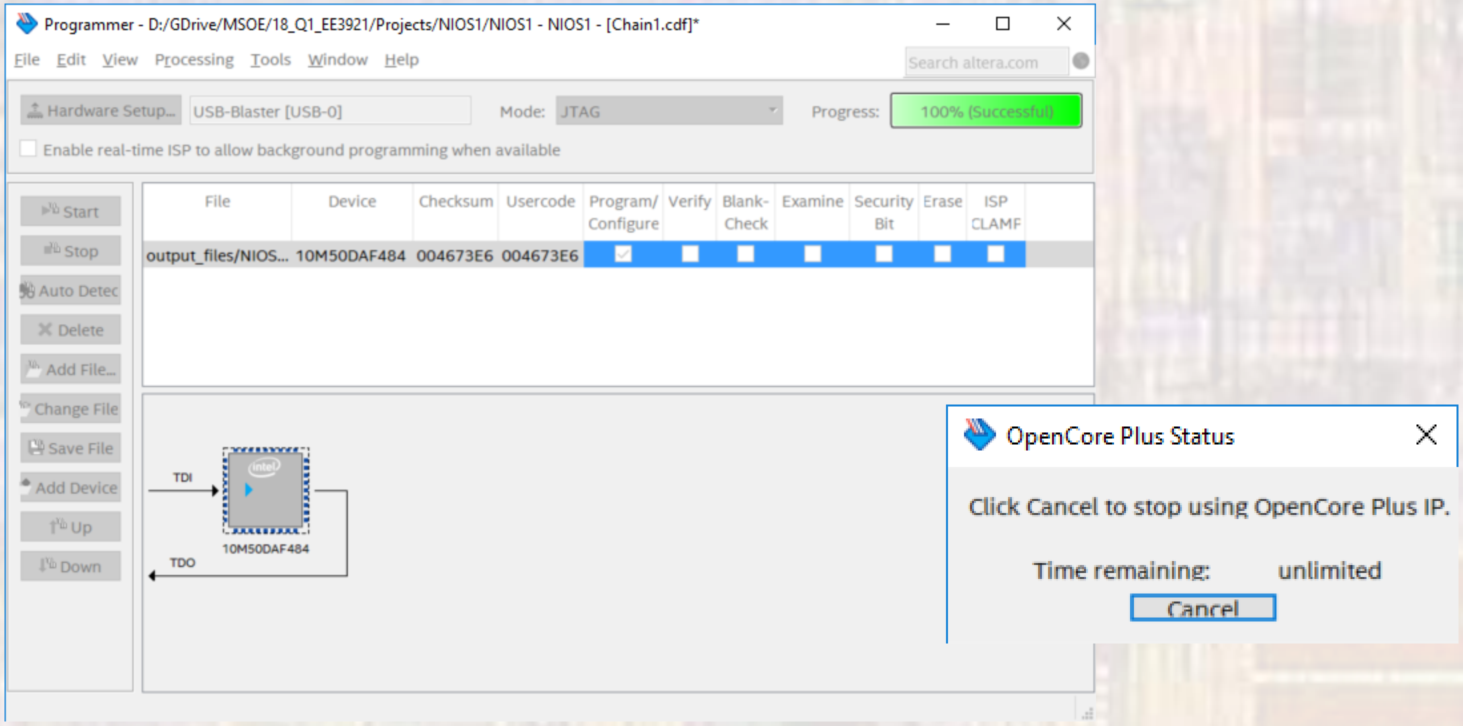
      - Import the pin aliases (qsf file)
      - Setup the SDF file

  - Be sure to set your top level entity

  - Start Compilation

NIOS II Pixel Display

- Create DE10 Design
  - Complete the HW setup
    - Download the HW project onto the board
    - DO NOT CLOSE either of these windows



Programmer - D:/GDrive/MSOE/18_Q1_EE3921/Projects/NIOS1/NIOS1 - NIOS1 - [Chain1.cdf]*

Hardware Setup... USB-Blaster [USB-0] Mode: JTAG Progress: 100% (Successful)

Enable real-time ISP to allow background programming when available

File	Device	Checksum	Usercode	Program/ Configure	Verify	Blank-Check	Examine	Security Bit	Erase	ISP CLAMP
output_files/NIOS...	10M50DAF484	004673E6	004673E6	✓	☐	☐	☐	☐	☐	☐

OpenCore Plus Status

Click Cancel to stop using OpenCore Plus IP.

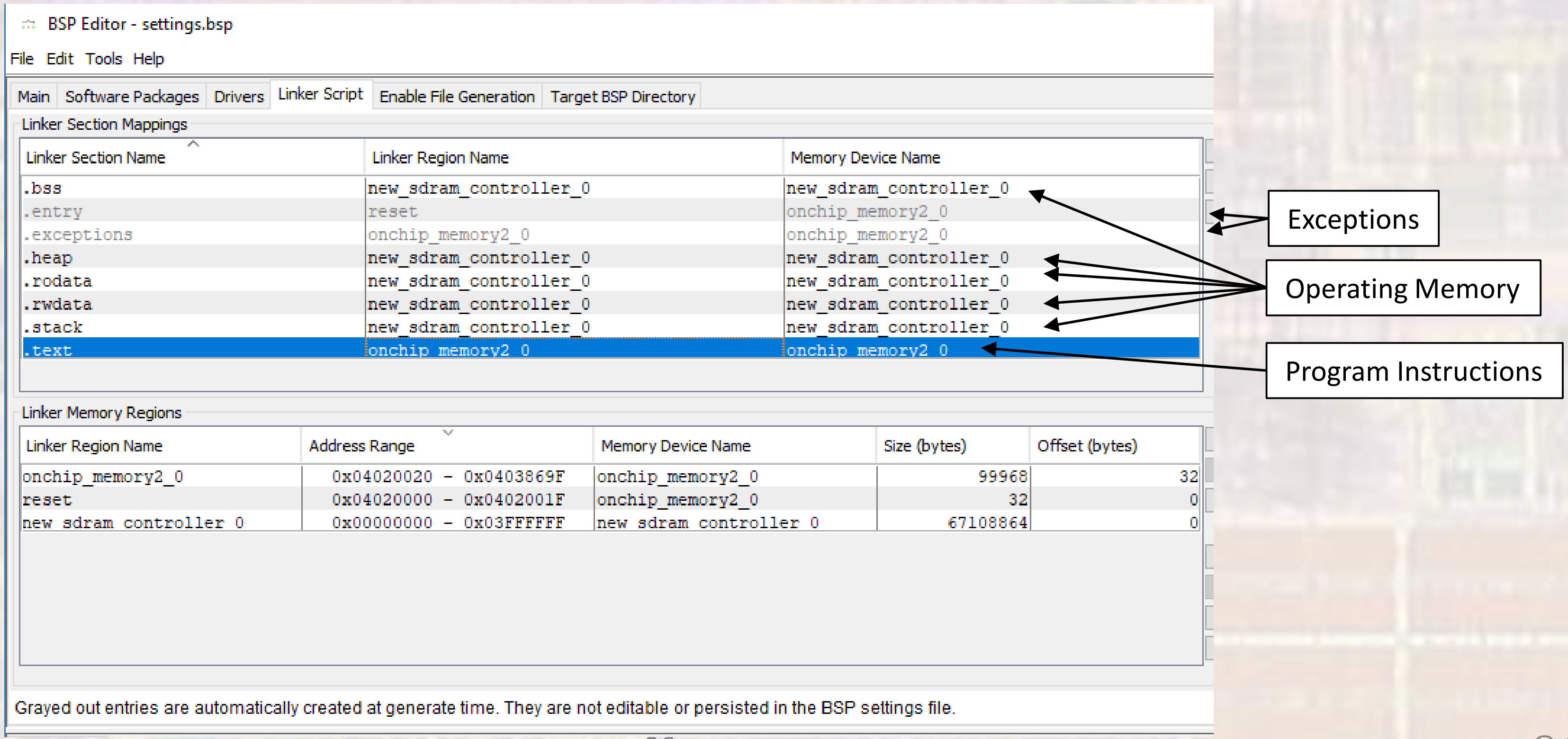
Time remaining: unlimited

NIOS II Pixel Display

- Create Eclipse System
  - Open NIOSII software
    - Tools → NIOSII Software Build Tools for Eclipse
  - Create the BSP
    - File → New → NIOSII Application and BSP from template
    - Blank Template
  - Edit the BSP
    - Right click on the BSP, NIOS II → BSP Editor
    - Change the properties for small systems
      - Small C library
      - Reduced device drivers
  - Re-Generate the BSP

NIOS II Pixel Display

- Create Eclipse System
  - Review the BSP Memory allocations
    - Right click on the BSP, NIOS → BSP Editor → Linker Script Tab
    - Most are in the SDRAM



Linker Section Mappings

Linker Section Name	Linker Region Name	Memory Device Name
.bss	new_sdram_controller_0	new_sdram_controller_0
.entry	reset	onchip_memory2_0
.exceptions	onchip_memory2_0	onchip_memory2_0
.heap	new_sdram_controller_0	new_sdram_controller_0
.rodata	new_sdram_controller_0	new_sdram_controller_0
.rwdata	new_sdram_controller_0	new_sdram_controller_0
.stack	new_sdram_controller_0	new_sdram_controller_0
.text	onchip_memory2_0	onchip_memory2_0

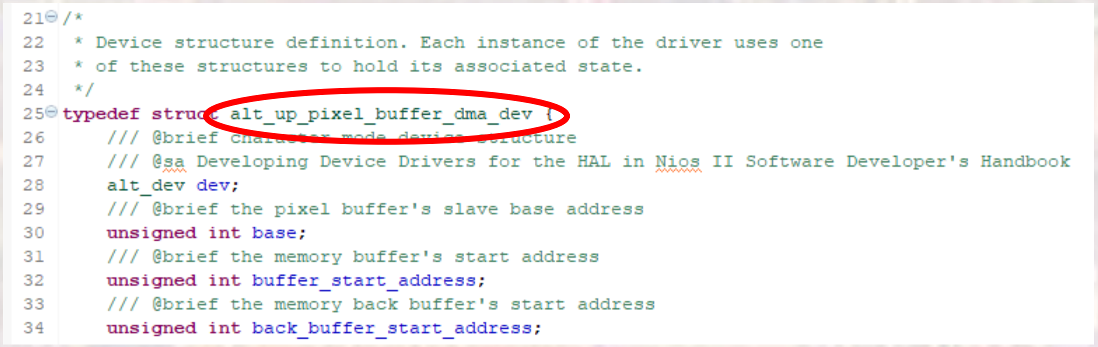
Linker Memory Regions

Linker Region Name	Address Range	Memory Device Name	Size (bytes)	Offset (bytes)
onchip_memory2_0	0x04020020 - 0x0403869F	onchip_memory2_0	99968	32
reset	0x04020000 - 0x0402001F	onchip_memory2_0	32	0
new_sdram_controller_0	0x00000000 - 0x03FFFFFF	new_sdram_controller_0	67108864	0

Grayed out entries are automatically created at generate time. They are not editable or persisted in the BSP settings file.

NIOS II Pixel Display

- Create Eclipse System
  - In the BSP – under drivers/inc
    - Open altera_up_avalon_video_pixel_buffer_dma.h
    - Find the video pixel buffer structure name



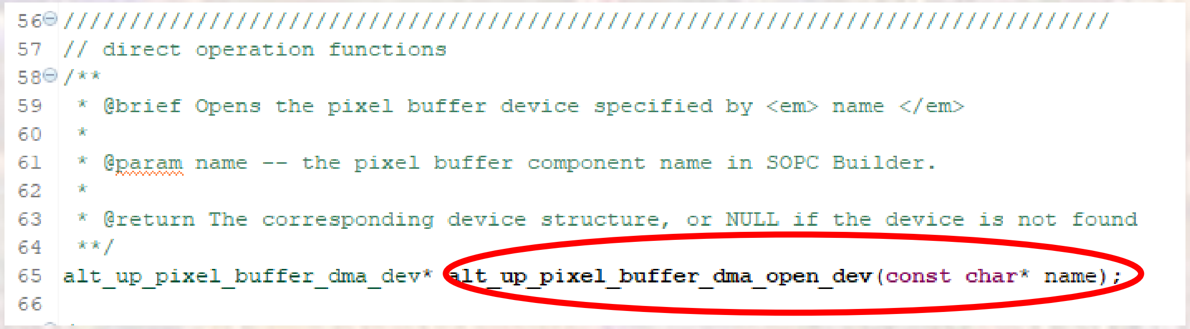
```
/*
 * Device structure definition. Each instance of the driver uses one
 * of these structures to hold its associated state.
 */
typedef struct alt_up_pixel_buffer_dma_dev {
    /// @brief character mode device structure
    /// @sa Developing Device Drivers for the HAL in Nios II Software Developer's Handbook
    alt_dev dev;
    /// @brief the pixel buffer's slave base address
    unsigned int base;
    /// @brief the memory buffer's start address
    unsigned int buffer_start_address;
    /// @brief the memory back buffer's start address
    unsigned int back_buffer_start_address;
```

    - Create a pointer of this type

```
// define a pointer of type pixel_buffer...
// to use as a reference in the dma functions
//
alt_up_pixel_buffer_dma_dev * pixel_buf_dma_dev;
```

NIOS II Pixel Display

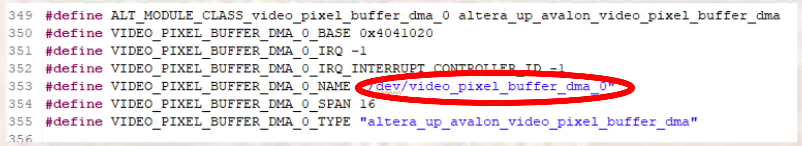
- Create Eclipse System
  - In the BSP – under drivers/inc
    - Open altera_up_avalon_video_pixel_buffer_dma.h
    - Find the function to open the pixel buffer dma device



```
///////////////////////////////////////////////////////////////////////////////
// direct operation functions
/**
 * @brief Opens the pixel buffer device specified by <em> name </em>
 *
 * @param name -- the pixel buffer component name in SOPC Builder.
 *
 * @return The corresponding device structure, or NULL if the device is not found
 **/
alt_up_pixel_buffer_dma_dev* alt_up_pixel_buffer_dma_open_dev(const char* name);
```

    - Open the device and assign it to the previously defined pointer

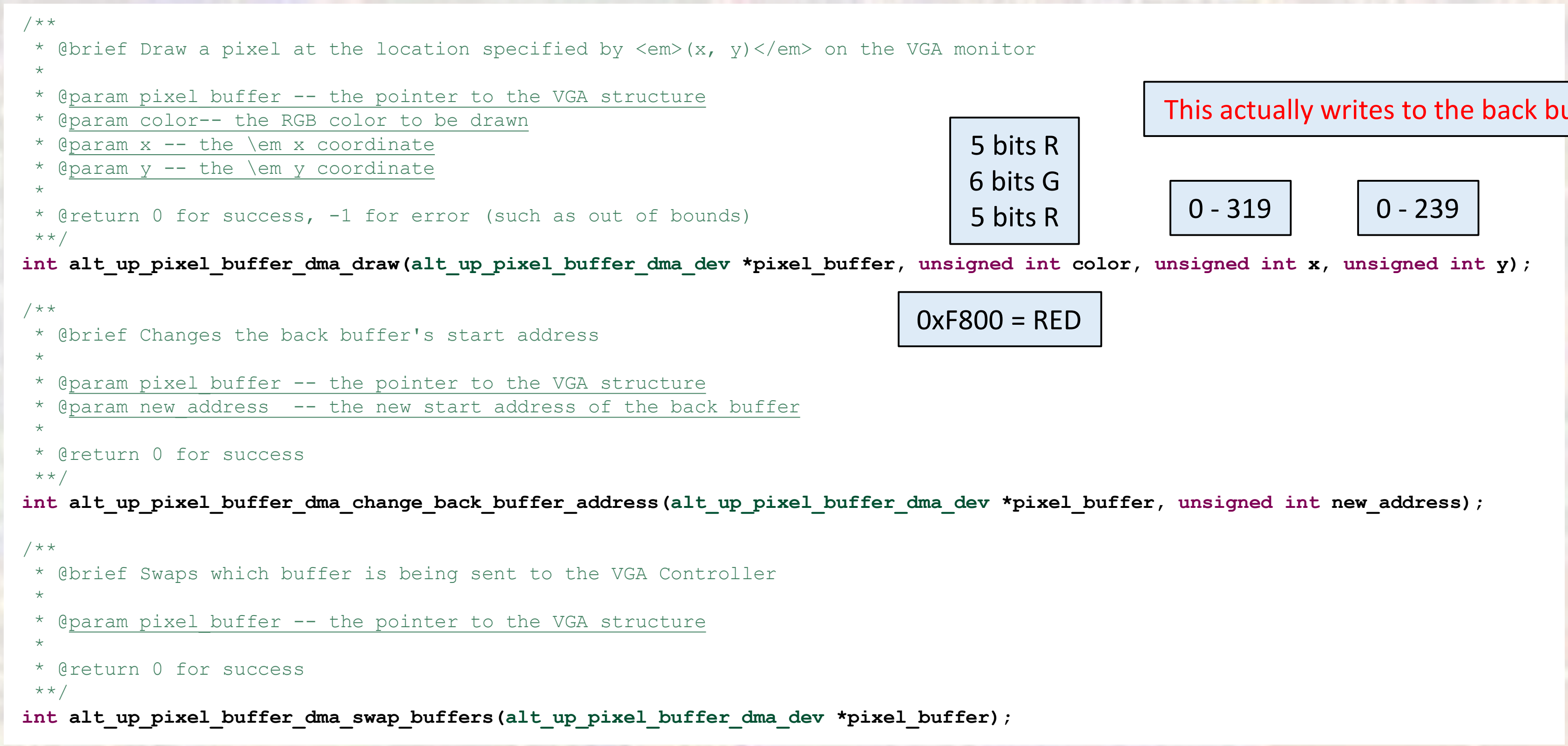
```
// open the Pixel Buffer port
//  - command is in drivers/inc/alter...video_pixel_buffer_dma.h
// name reference is in system.h
//  - "/dev/video_pixel_buffer_dma_0"
//
pixel_buf_dma_dev = alt_up_pixel_buffer_dma_open_dev ("/dev/video_pixel_buffer_dma_0");
```



```
#define ALT_MODULE_CLASS_video_pixel_buffer_dma_0 altera_up_avalon_video_pixel_buffer_dma
#define VIDEO_PIXEL_BUFFER_DMA_0_BASE 0x4041020
#define VIDEO_PIXEL_BUFFER_DMA_0_IRQ -1
#define VIDEO_PIXEL_BUFFER_DMA_0_IRQ_INTERRUPT_CONTROLLER_ID -1
#define VIDEO_PIXEL_BUFFER_DMA_0_NAME "/dev/video_pixel_buffer_dma_0"
#define VIDEO_PIXEL_BUFFER_DMA_0_SPAN 16
#define VIDEO_PIXEL_BUFFER_DMA_0_TYPE "altera_up_avalon_video_pixel_buffer_dma"
```

NIOS II Pixel Display

- Create Eclipse System
  - In the BSP – under drivers/inc
    - Open altera_up_avalon_video_pixel_buffer_dma.h
    - The remainder of the pixel buffer dma commands are in this file



```
/**
 * @brief Draw a pixel at the location specified by <em>(x, y)</em> on the VGA monitor
 *
 * @param pixel_buffer -- the pointer to the VGA structure
 * @param color-- the RGB color to be drawn
 * @param x -- the \em x coordinate
 * @param y -- the \em y coordinate
 *
 * @return 0 for success, -1 for error (such as out of bounds)
 **/
int alt_up_pixel_buffer_dma_draw(alt_up_pixel_buffer_dma_dev *pixel_buffer, unsigned int color, unsigned int x, unsigned int y);

/**
 * @brief Changes the back buffer's start address
 *
 * @param pixel_buffer -- the pointer to the VGA structure
 * @param new_address  -- the new start address of the back buffer
 *
 * @return 0 for success
 **/
int alt_up_pixel_buffer_dma_change_back_buffer_address(alt_up_pixel_buffer_dma_dev *pixel_buffer, unsigned int new_address);

/**
 * @brief Swaps which buffer is being sent to the VGA Controller
 *
 * @param pixel_buffer -- the pointer to the VGA structure
 *
 * @return 0 for success
 **/
int alt_up_pixel_buffer_dma_swap_buffers(alt_up_pixel_buffer_dma_dev *pixel_buffer);
```

This actually writes to the back buffer

5 bits R
6 bits G
5 bits R

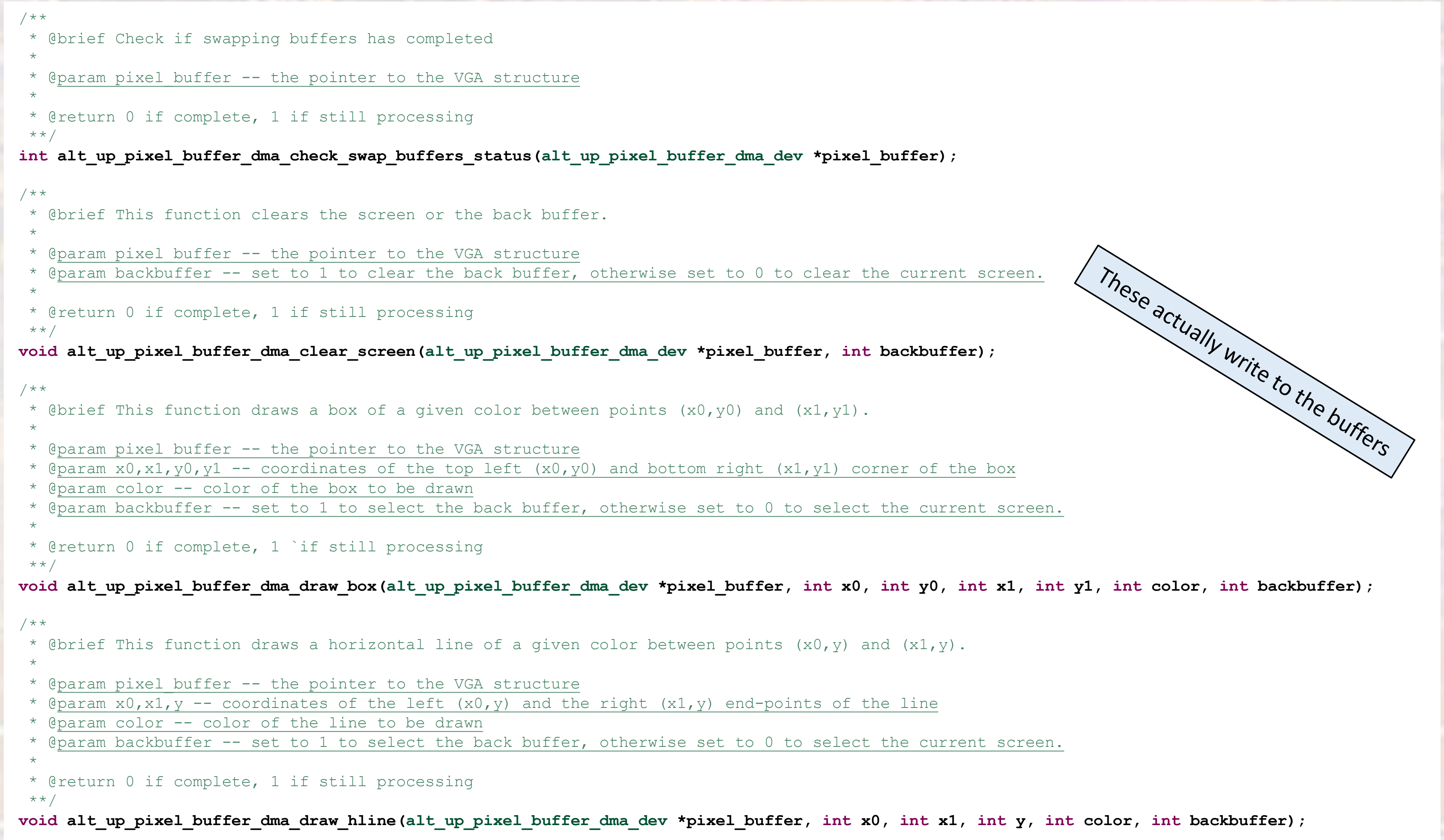
0 - 319

0 - 239

0xF800 = RED

NIOS II Pixel Display

- Create Eclipse System



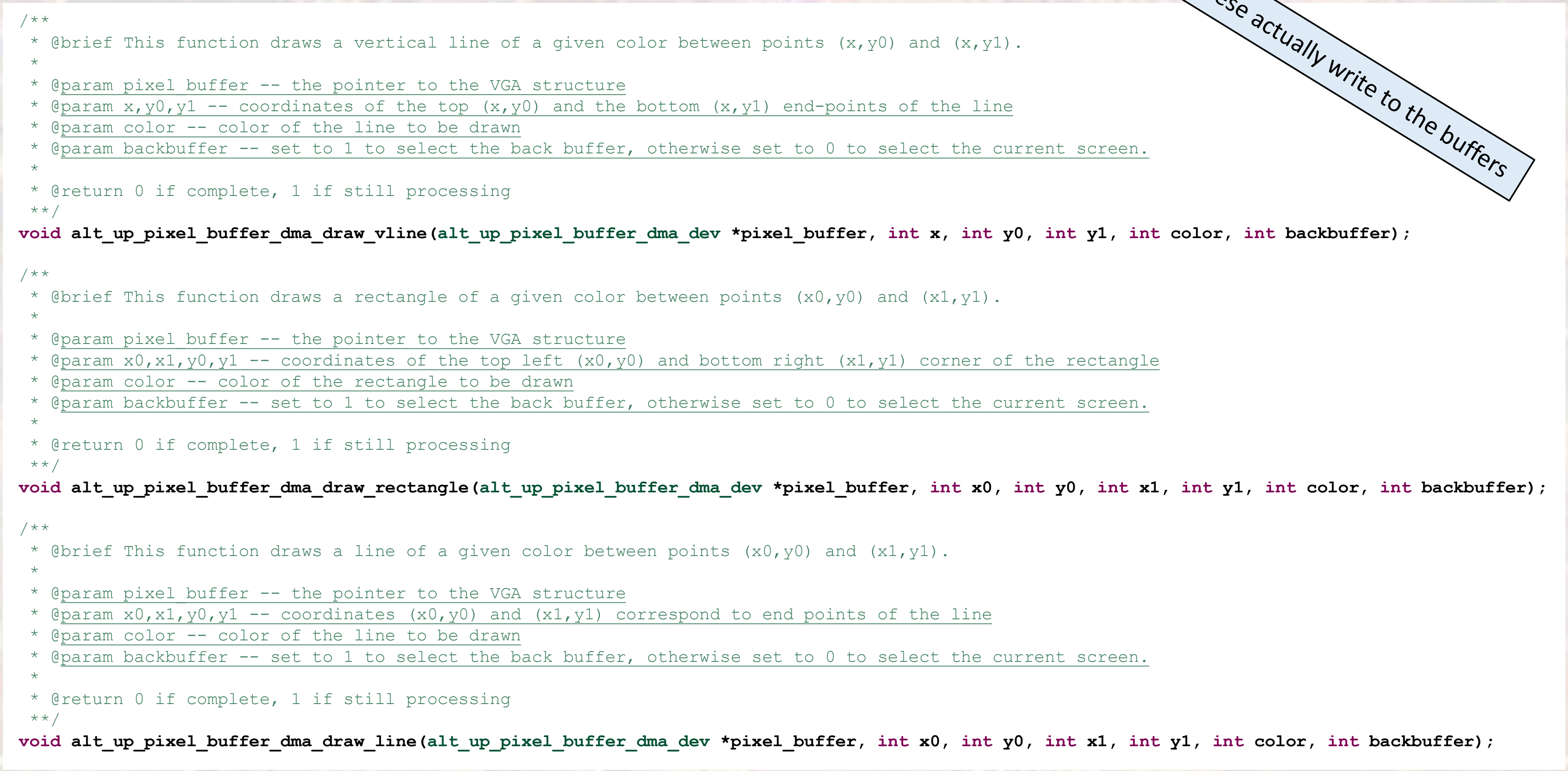
```
/**
 * @brief Check if swapping buffers has completed
 *
 * @param pixel_buffer -- the pointer to the VGA structure
 *
 * @return 0 if complete, 1 if still processing
 **/
int alt_up_pixel_buffer_dma_check_swap_buffers_status(alt_up_pixel_buffer_dma_dev *pixel_buffer);

/**
 * @brief This function clears the screen or the back buffer.
 *
 * @param pixel_buffer -- the pointer to the VGA structure
 * @param backbuffer -- set to 1 to clear the back buffer, otherwise set to 0 to clear the current screen.
 *
 * @return 0 if complete, 1 if still processing
 **/
void alt_up_pixel_buffer_dma_clear_screen(alt_up_pixel_buffer_dma_dev *pixel_buffer, int backbuffer);

/**
 * @brief This function draws a box of a given color between points (x0,y0) and (x1,y1).
 *
 * @param pixel_buffer -- the pointer to the VGA structure
 * @param x0,x1,y0,y1 -- coordinates of the top left (x0,y0) and bottom right (x1,y1) corner of the box
 * @param color -- color of the box to be drawn
 * @param backbuffer -- set to 1 to select the back buffer, otherwise set to 0 to select the current screen.
 *
 * @return 0 if complete, 1 `if still processing
 **/
void alt_up_pixel_buffer_dma_draw_box(alt_up_pixel_buffer_dma_dev *pixel_buffer, int x0, int y0, int x1, int y1, int color, int backbuffer);

/**
 * @brief This function draws a horizontal line of a given color between points (x0,y) and (x1,y).
 *
 * @param pixel_buffer -- the pointer to the VGA structure
 * @param x0,x1,y -- coordinates of the left (x0,y) and the right (x1,y) end-points of the line
 * @param color -- color of the line to be drawn
 * @param backbuffer -- set to 1 to select the back buffer, otherwise set to 0 to select the current screen.
 *
 * @return 0 if complete, 1 if still processing
 **/
void alt_up_pixel_buffer_dma_draw_hline(alt_up_pixel_buffer_dma_dev *pixel_buffer, int x0, int x1, int y, int color, int backbuffer);
```

These actually write to the buffers

NIOS II Pixel Display

- Create Eclipse System

These actually write to the buffers



```
/**
 * @brief This function draws a vertical line of a given color between points (x,y0) and (x,y1).
 *
 * @param pixel_buffer -- the pointer to the VGA structure
 * @param x,y0,y1 -- coordinates of the top (x,y0) and the bottom (x,y1) end-points of the line
 * @param color -- color of the line to be drawn
 * @param backbuffer -- set to 1 to select the back buffer, otherwise set to 0 to select the current screen.
 *
 * @return 0 if complete, 1 if still processing
 **/
void alt_up_pixel_buffer_dma_draw_vline(alt_up_pixel_buffer_dma_dev *pixel_buffer, int x, int y0, int y1, int color, int backbuffer);

/**
 * @brief This function draws a rectangle of a given color between points (x0,y0) and (x1,y1).
 *
 * @param pixel_buffer -- the pointer to the VGA structure
 * @param x0,x1,y0,y1 -- coordinates of the top left (x0,y0) and bottom right (x1,y1) corner of the rectangle
 * @param color -- color of the rectangle to be drawn
 * @param backbuffer -- set to 1 to select the back buffer, otherwise set to 0 to select the current screen.
 *
 * @return 0 if complete, 1 if still processing
 **/
void alt_up_pixel_buffer_dma_draw_rectangle(alt_up_pixel_buffer_dma_dev *pixel_buffer, int x0, int y0, int x1, int y1, int color, int backbuffer);

/**
 * @brief This function draws a line of a given color between points (x0,y0) and (x1,y1).
 *
 * @param pixel_buffer -- the pointer to the VGA structure
 * @param x0,x1,y0,y1 -- coordinates (x0,y0) and (x1,y1) correspond to end points of the line
 * @param color -- color of the line to be drawn
 * @param backbuffer -- set to 1 to select the back buffer, otherwise set to 0 to select the current screen.
 *
 * @return 0 if complete, 1 if still processing
 **/
void alt_up_pixel_buffer_dma_draw_line(alt_up_pixel_buffer_dma_dev *pixel_buffer, int x0, int y0, int x1, int y1, int color, int backbuffer);
```

NIOS II Pixel Display

- Create Eclipse System
  - Write a program to print some Pixels to the screen

Include the DMA functions

```
//////////////////
// Includes
//////////////////
#include "altera_up_avalon_video_pixel_buffer_dma.h"
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
#include <stdint.h>

int main(void){
       // define a pointer of type pixel_buffer...
       // to use as a reference in the dma functions
       //
       alt_up_pixel_buffer_dma_dev * pixel_buf_dma_dev;

       // open the Pixel Buffer port
       //   - command is in drivers/inc/alter...video_pixel_buffer_dma.h
       // name reference is in system.h
       //   - "/dev/video_pixel_buffer_dma_0"
       //
       pixel_buf_dma_dev = alt_up_pixel_buffer_dma_open_dev ("/dev/video_pixel_buffer_dma_0");

       // Check for error and output to the console
       //
       if ( pixel_buf_dma_dev == NULL)
           printf ("Error: could not open pixel buffer device \n");
       else
           printf ("Opened pixel buffer device \n");
```

Debug code

NIOS II Pixel Display

- Create Eclipse System
  - Write a program to print some Pixels to the screen

```
 // Clear the screen
 // - command is in drivers/inc/alter...video_pixel_buffer_dma.h
 // - wait until done before continuing
 //
 alt_up_pixel_buffer_dma_clear_screen (pixel_buf_dma_dev, 0);
 usleep(1000000);// 1sec

// Draw a box
// - command is in drivers/inc/alter...video_pixel_buffer_dma.h
//
alt_up_pixel_buffer_dma_draw_box (pixel_buf_dma_dev, 100,  50, 149,  99, 0xF800, 0);
alt_up_pixel_buffer_dma_draw_box (pixel_buf_dma_dev, 150, 100, 199, 149, 0x07E0, 0);
alt_up_pixel_buffer_dma_draw_box (pixel_buf_dma_dev, 200, 150, 249, 199, 0x001F, 0);
```

NIOS II Pixel Display

- Create Eclipse System
  - Compile the software
    - Select the code file (box.c)
    - Project → Build Project
    - Right Click on the project → run as → Nios II Hardware