NIOS Pixel Display - HW

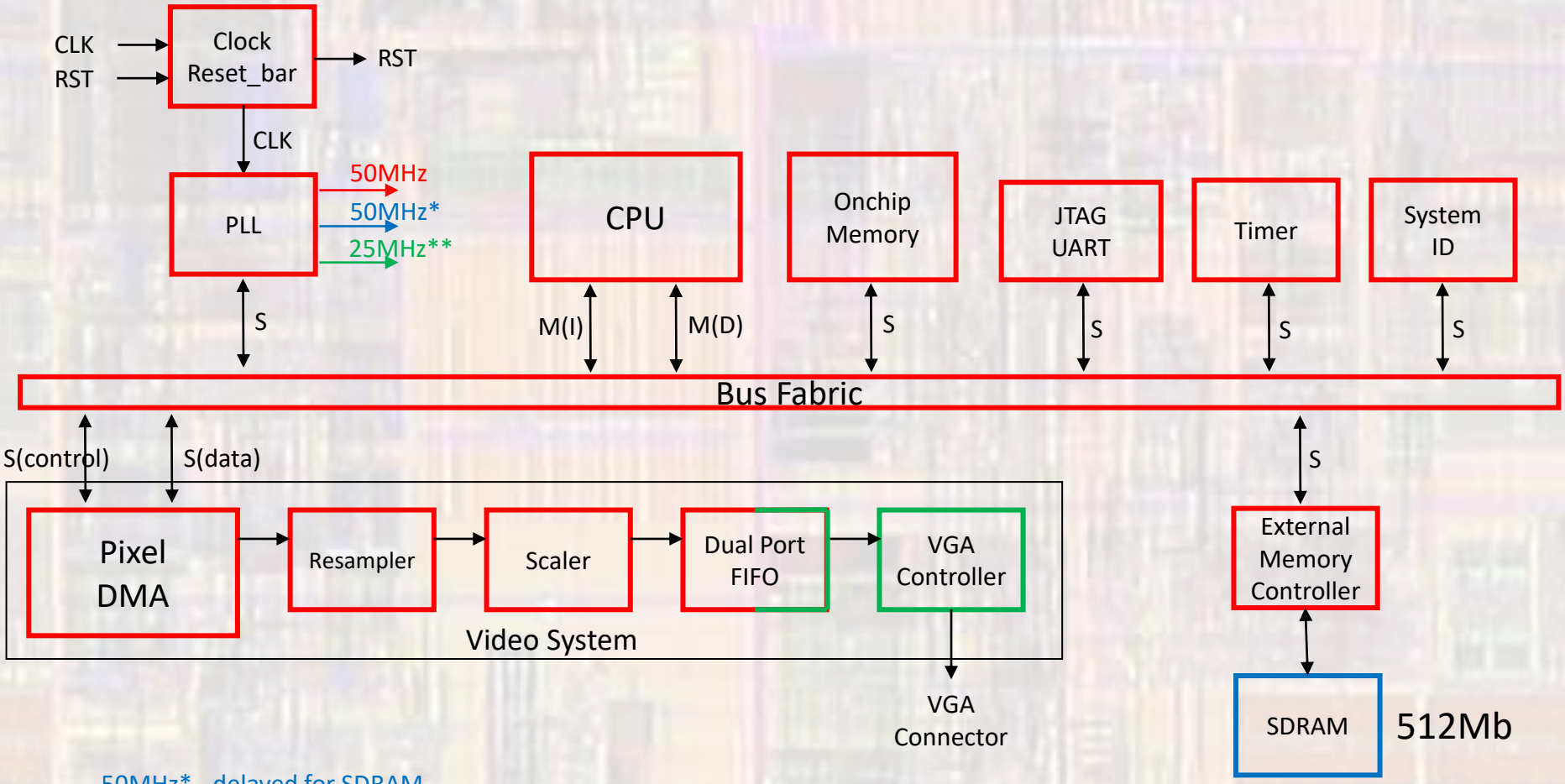
Last updated 10/12/20

NIOS II Pixel Display - HW

These slides describe the development of a moderately complex NIOS Processor using the Pixel Buffer IP

Upon completion: You should be able implement your own NIOS processor using the Pixel Buffer IP

NIOS II Pixel Display - HW



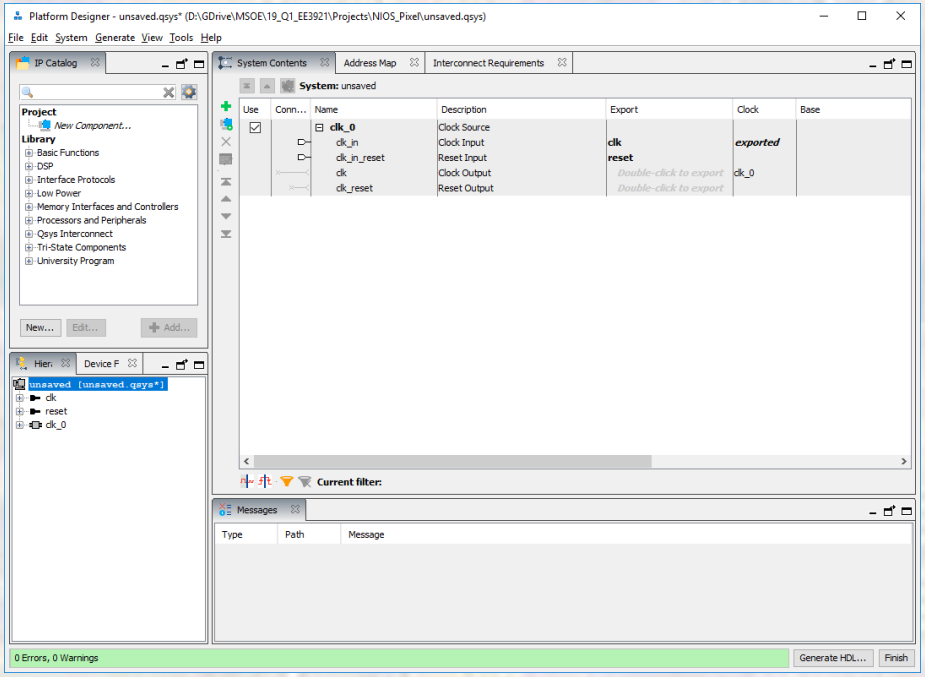
50MHz* - delayed for SDRAM

25MHz** - VGA clk

NIOS II Pixel Display - HW

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

- Open Tools → Platform Designer



NIOS II Pixel Display - HW

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

50MHz input clock

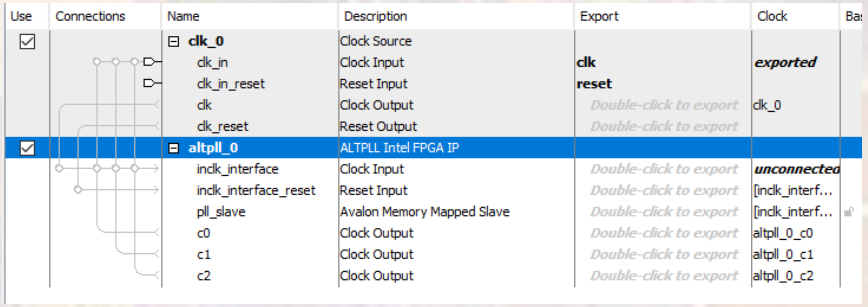
no areset or locked

c0 → 50MHz

c1 → 50MHz

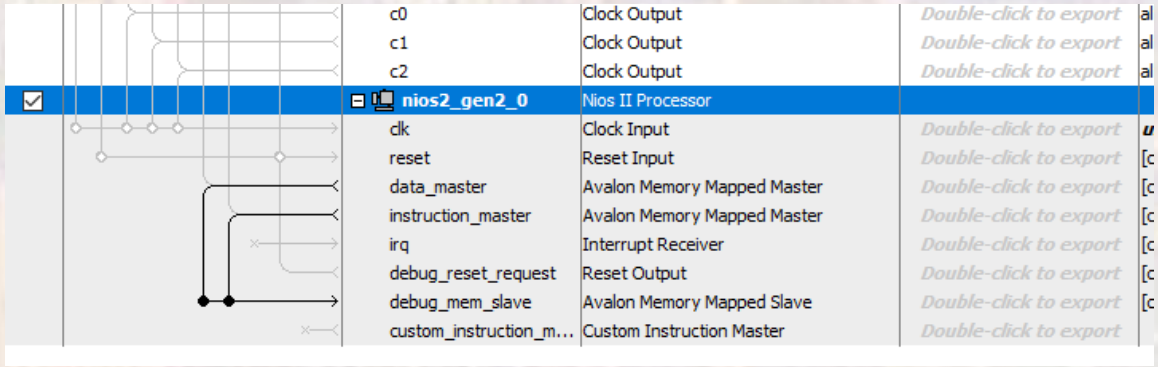
-3ns phase shift (watch for the units)

c2 → 25MHz



NIOS II Pixel Display - HW

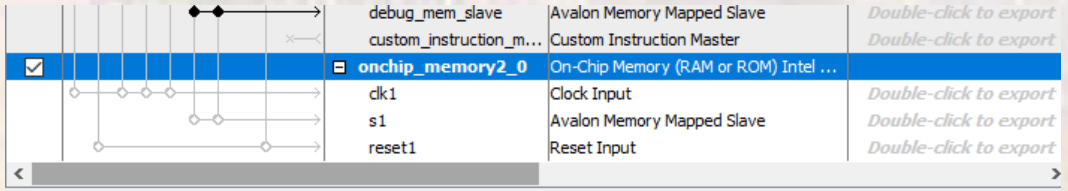
- Add NIOS
  - Processors and Peripherals → Embedded Processors → NIOS II Processor
  - NIOS II/f



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

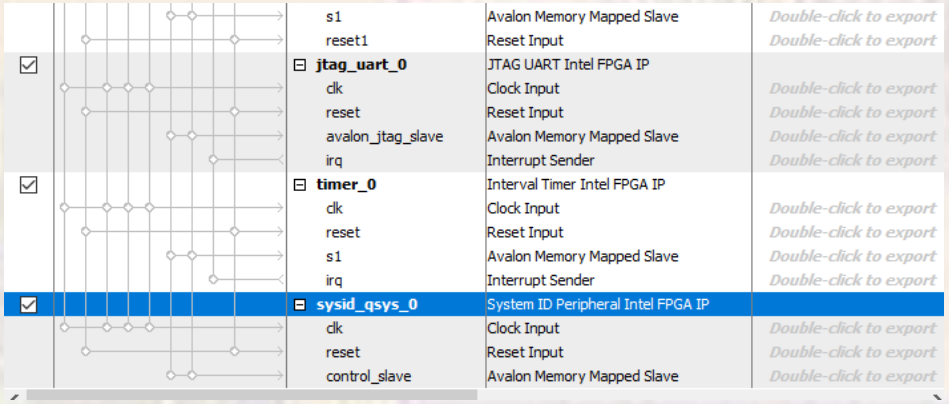
    RAM

    Size = 100,000 bytes



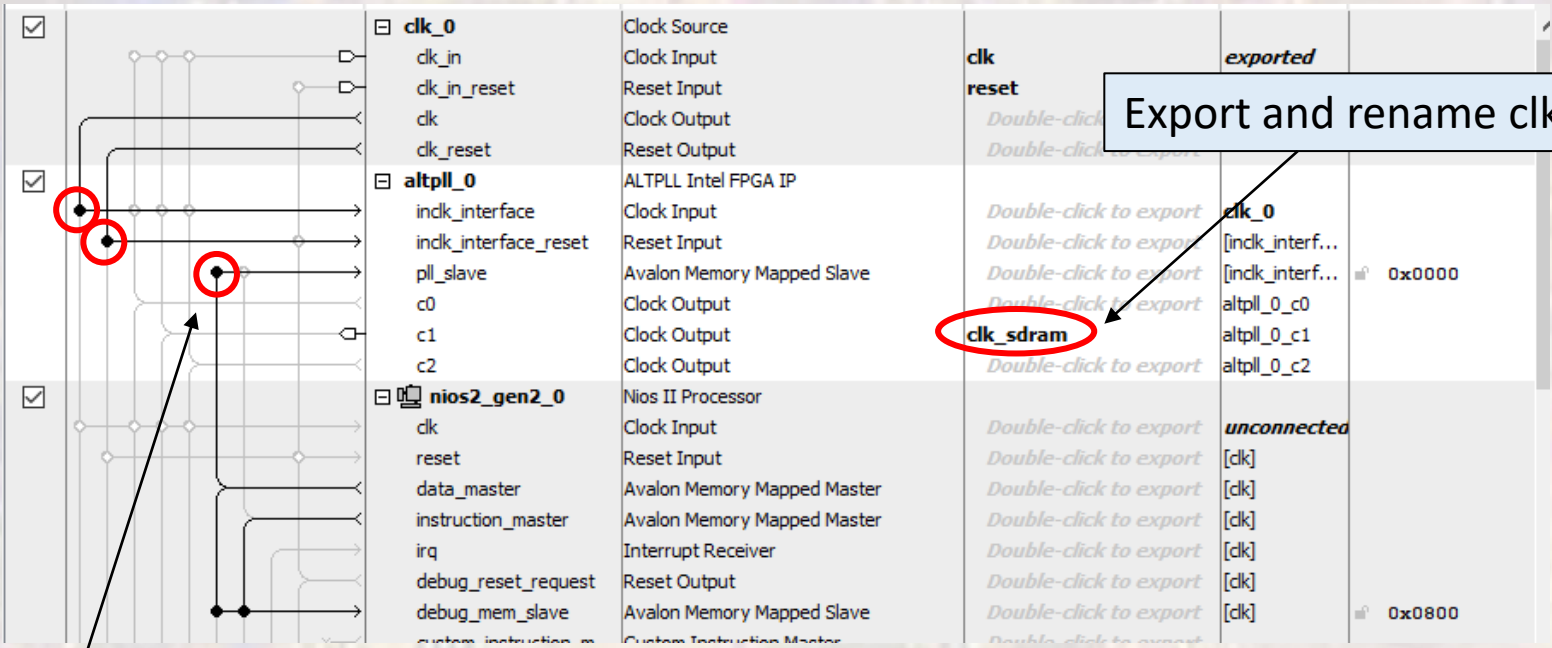
NIOS II Pixel Display - HW

- 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



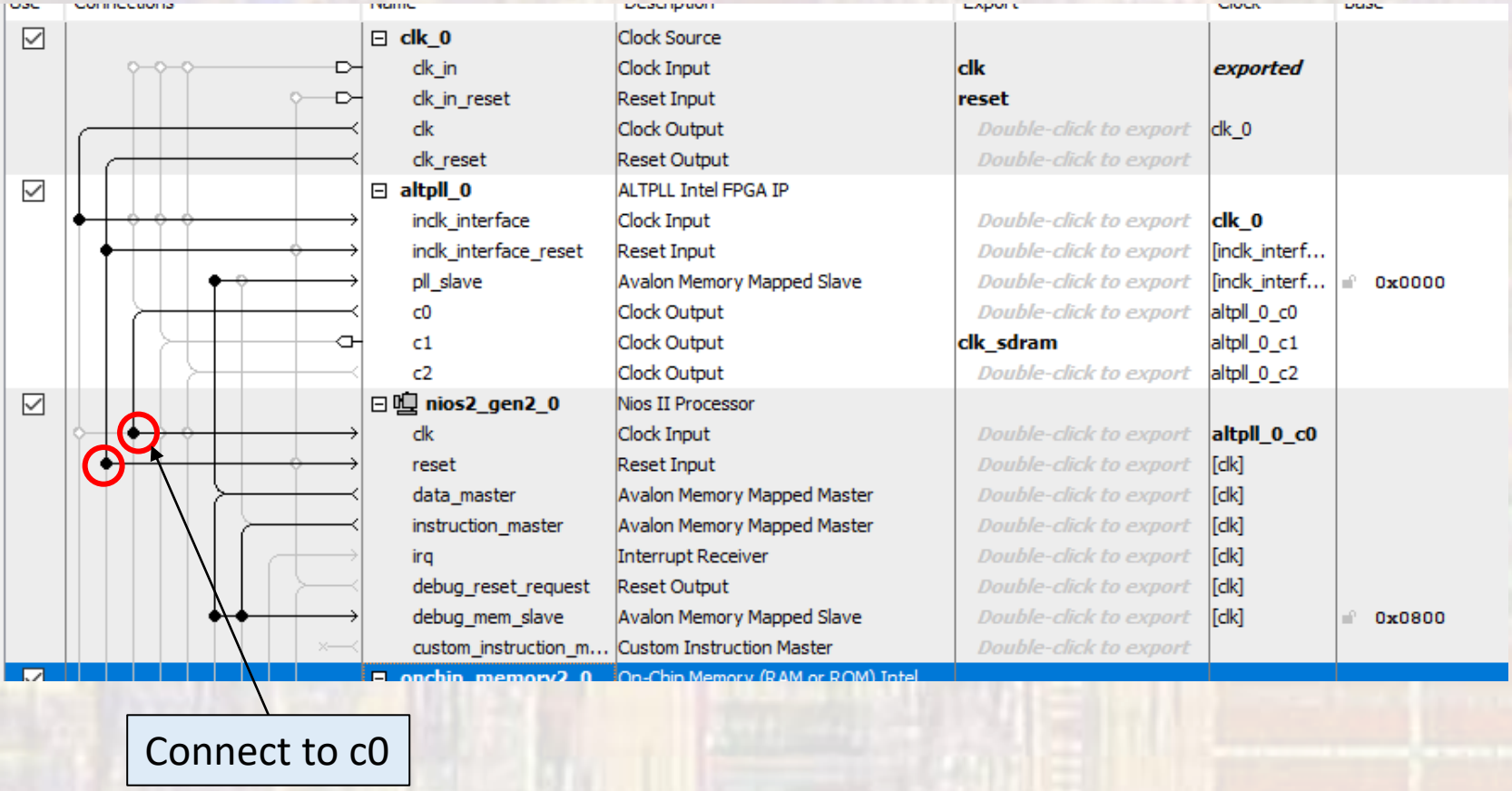
NIOS II Pixel Display - HW

- Connect up basic NIOS system
  - PLL Inputs



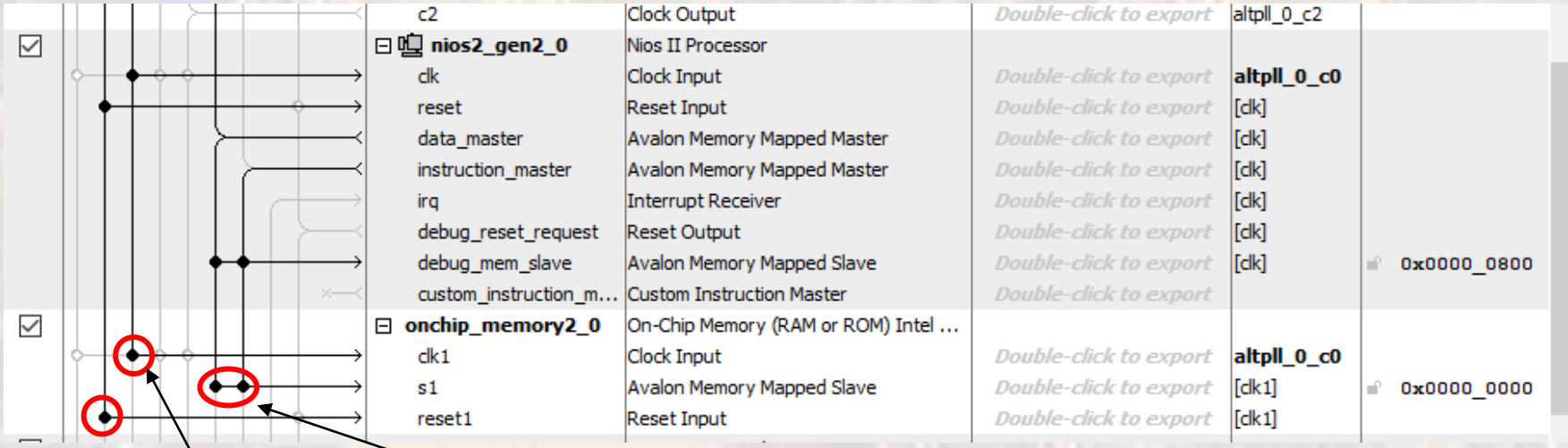
NIOS II Pixel Display - HW

- Connect up basic NIOS system
  - NIOS Inputs



NIOS II Pixel Display - HW

- Connect up basic NIOS system
  - On-chip Memory

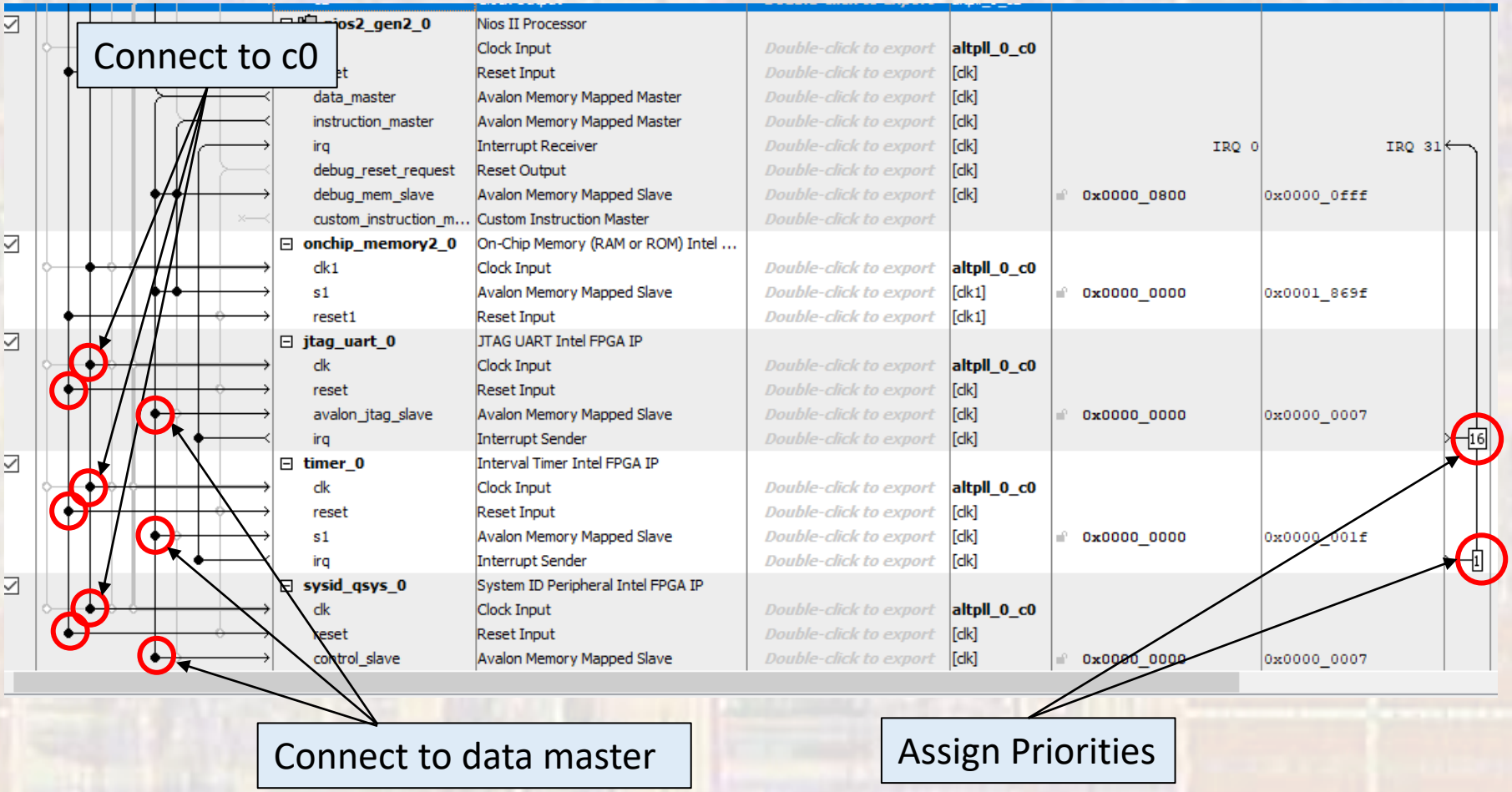


Connect to data and instruction masters

Connect to c0

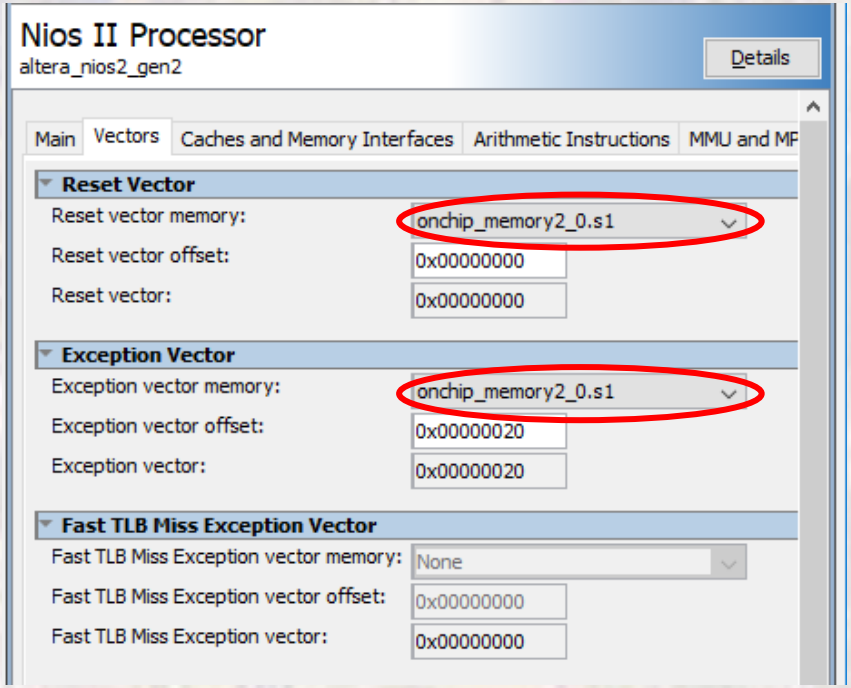
NIOS II Pixel Display - HW

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



NIOS II Pixel Display - HW

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



NIOS II Pixel Display - HW

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

Memory Profile

16 bits

1 chip select

4 banks

13 rows

10 columns

Timing

CAS = 3

2 initialization refresh cycles

7.8125 us refresh command

100us delay after pu

70ns refresh duration

20ns pre-charge duration

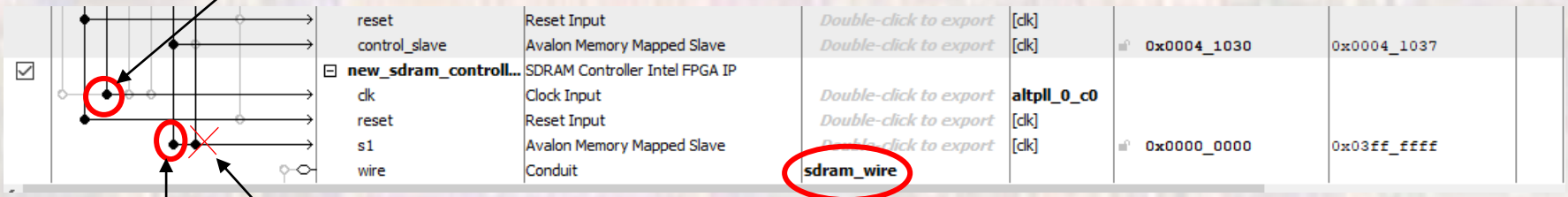
20 ns Active R/W delay

5.5ns access time

14ns write recovery time

NIOS II Pixel Display - HW

- Create Video System
  - Connect SDRAM Controller



Connect to c0

Do not connect to the Instruction master

Connect to data master

Export and Rename

NIOS II Pixel Display - HW

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

Default Buffer Start – 0x01000000

Default Back Buffer Start – 0x01100000

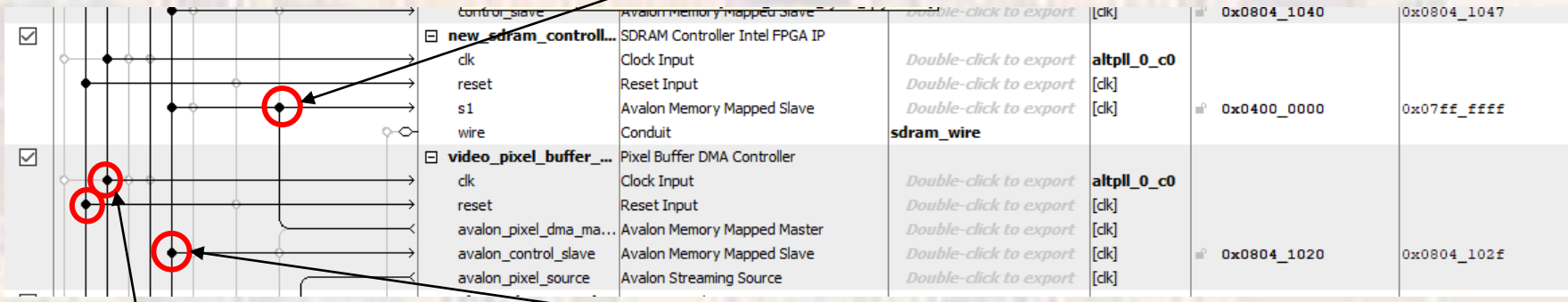
Width – 320

Height – 240

Color space – 16bit RGB

Depends on the final memory map

Connect to SDRAM_Controller MM Slave



Connect to data master

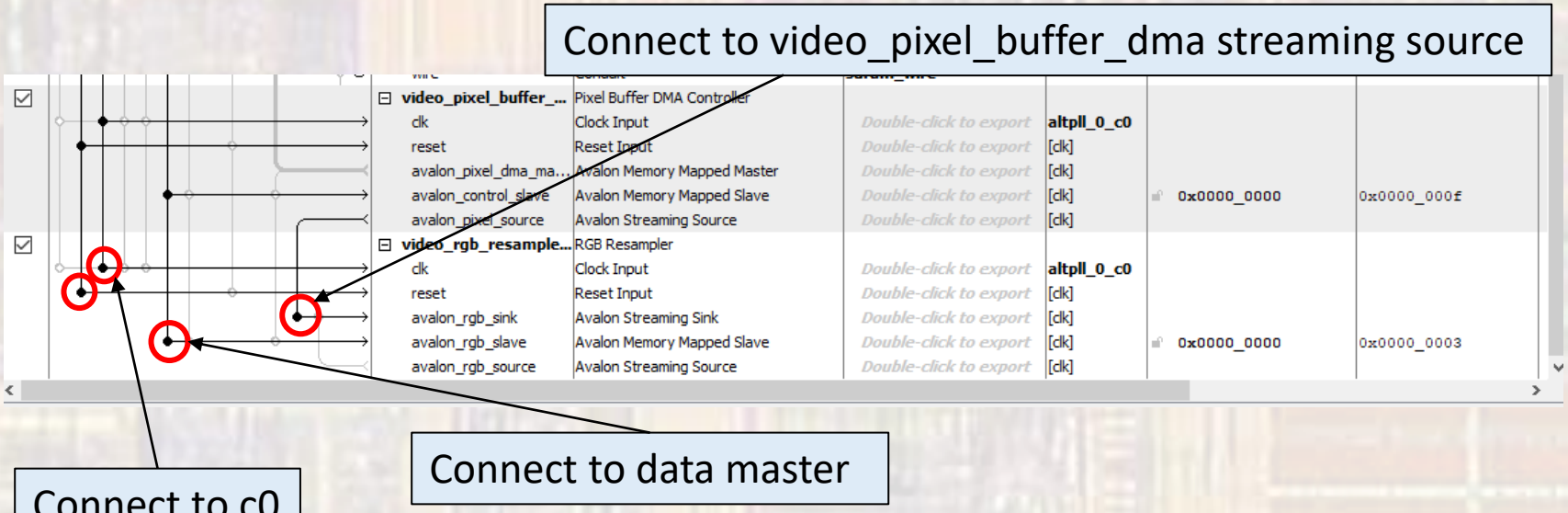
Connect to c0

NIOS II Pixel Display - HW

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

16 bit RGB incoming

30 bit RGB outgoing



NIOS II Pixel Display - HW

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

      Width Scaling – 2

      Height Scaling – 2

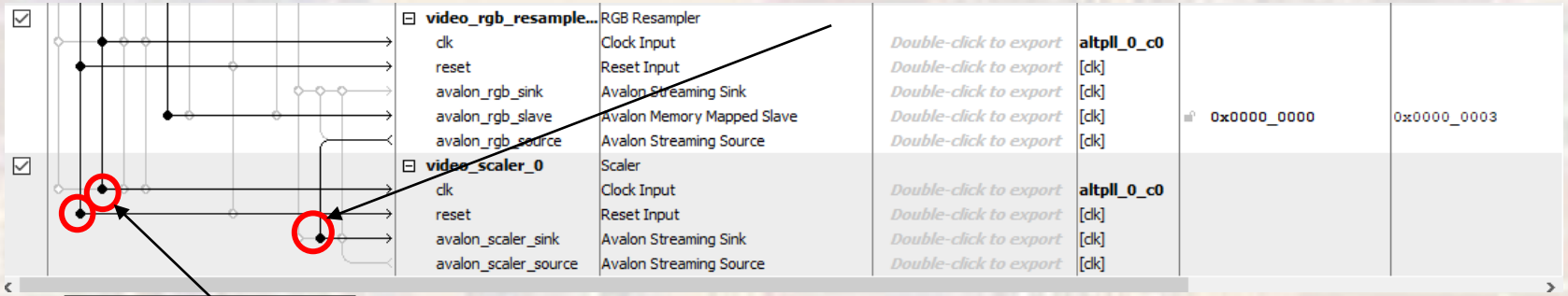
      Width – 320

      Height – 240

      10 bits / symbol

      3 symbols / beat

Connect to rgb_resampler streaming source



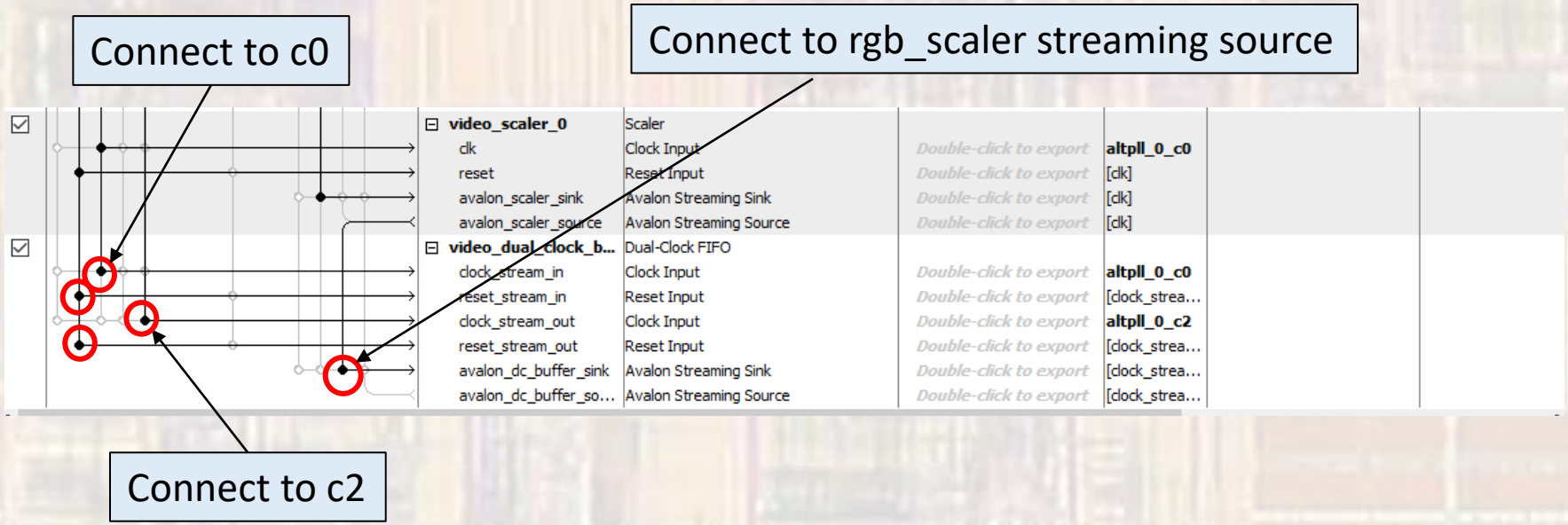
Connect to c0

NIOS II Pixel Display - HW

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

      Color Bits – 10

      Color Planes - 3



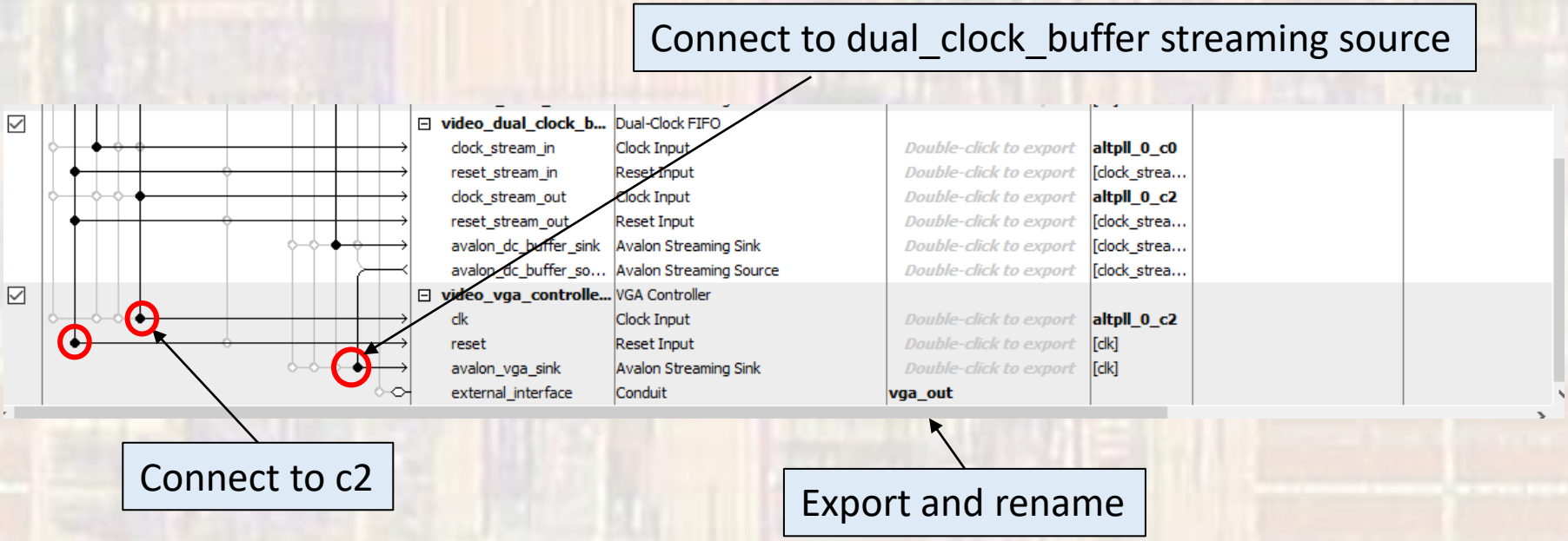
NIOS II Pixel Display - HW

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

      DE10-Lite

      VGA Connector

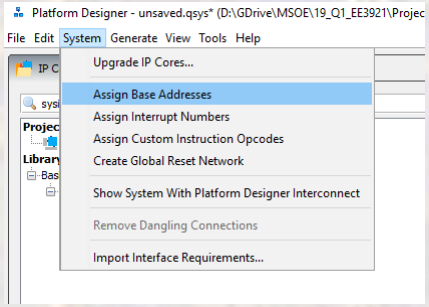
      VGA 640x480



NIOS II P

- Create Vic

Assign Base Addresses



System Contents | Address Map | Interconnect Requirements

System: nios_pixel Path: video_rgb_resampler_0.reset

Use	Connections	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				

Current filter:

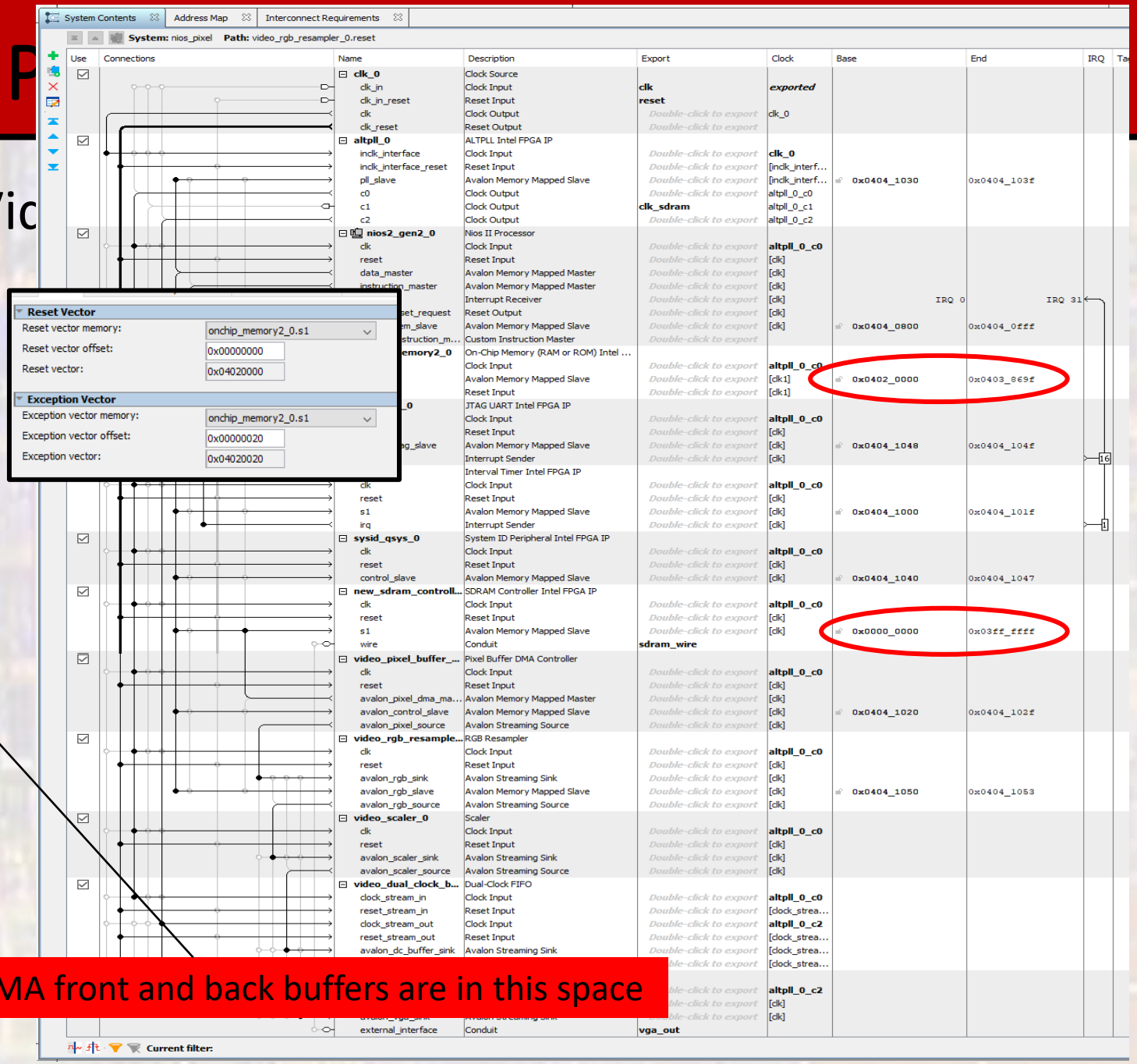
NIOS II P

- Create Vic

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

SDRAM
0x00000000 – 0x03FFFFFF

Peripherals
0x04041000 – 0x04041053



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

NIOS II Pixel Display - HW

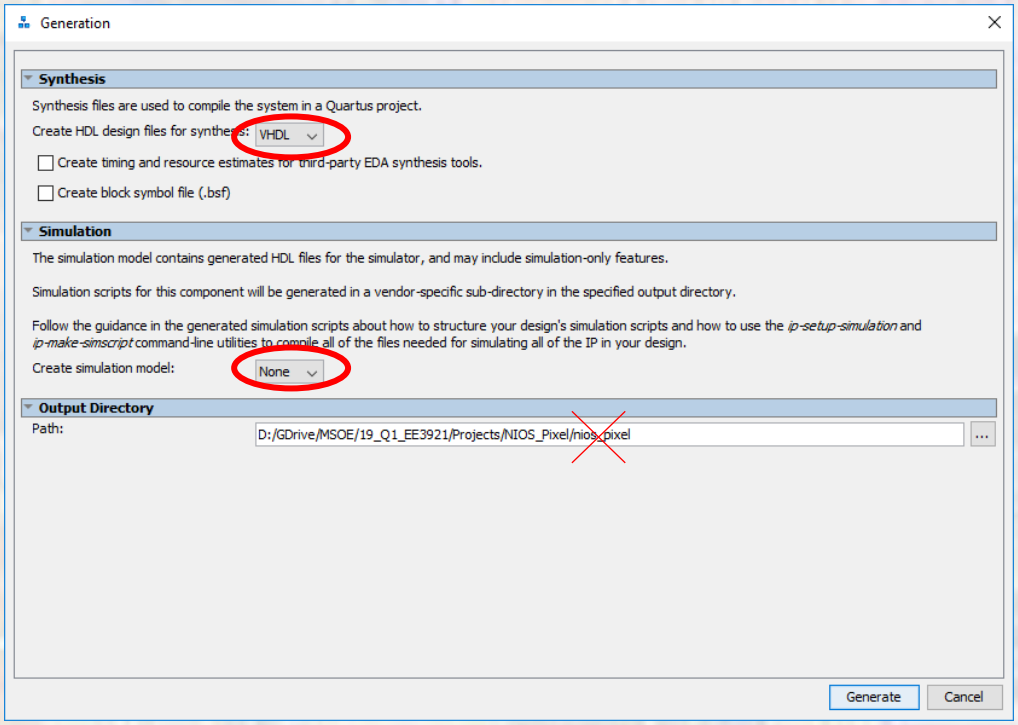
- Create Video System
  - Check for errors

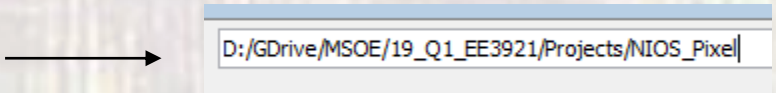
Messages

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 - HW

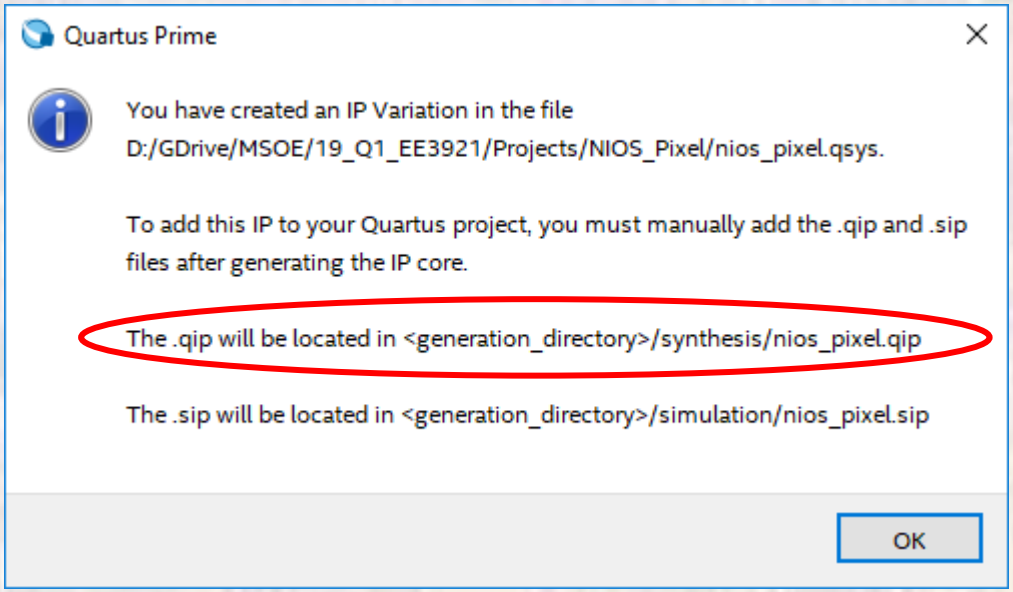
- Create Video System
  - Save the Platform Designer system
  - Generate the Platform Designer system
    - The first time you generate you must delete the last directory in the path – don't use the '…'





NIOS II Pixel Display - HW

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



NIOS II Pixel Display - HW

- 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 - HW

- Create DE10 Design
  - Instantiate into a VHDL file

Instantiation template component

```
----------------------------------
--
-- nios_pixel_de10.vhdl
--
-- Created  9/18/18
-- by:      johnsontimoj
-- rev:     0
--
----------------------------------
--
-- Nios pixal system - vga driver with pixel buffer syst
--
----------------------------------

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;
```

```
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;
```

DE10 pin aliases from .qsf file

NIOS II Pixel Display - HW

- 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 - HW

- 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 - HW

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