NIOS I/O

Last updated 7/20/23

NIOS I/O

- NIOS II Embedded Design Suite

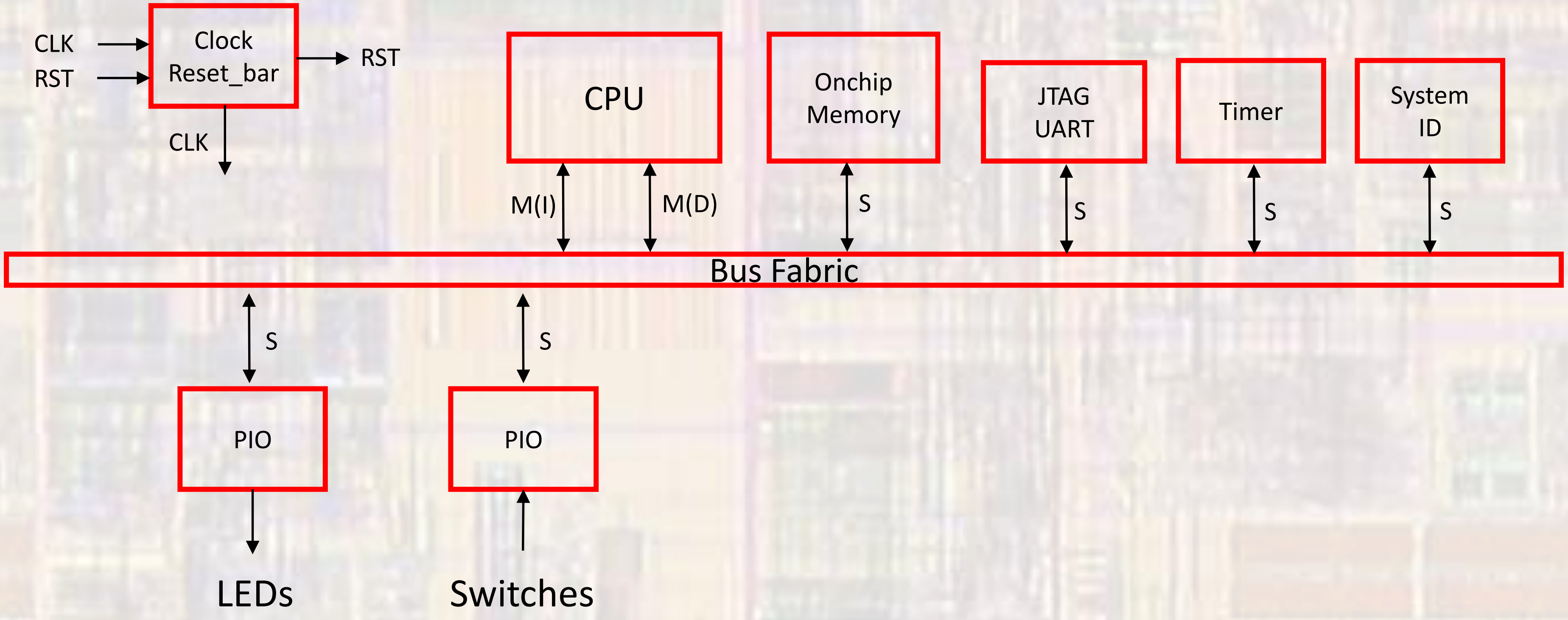
  - Configurable Processor

  - Selection of Peripherals

  - Eclipse based Board Support Package (BSP) for SW development

NIOS I/O

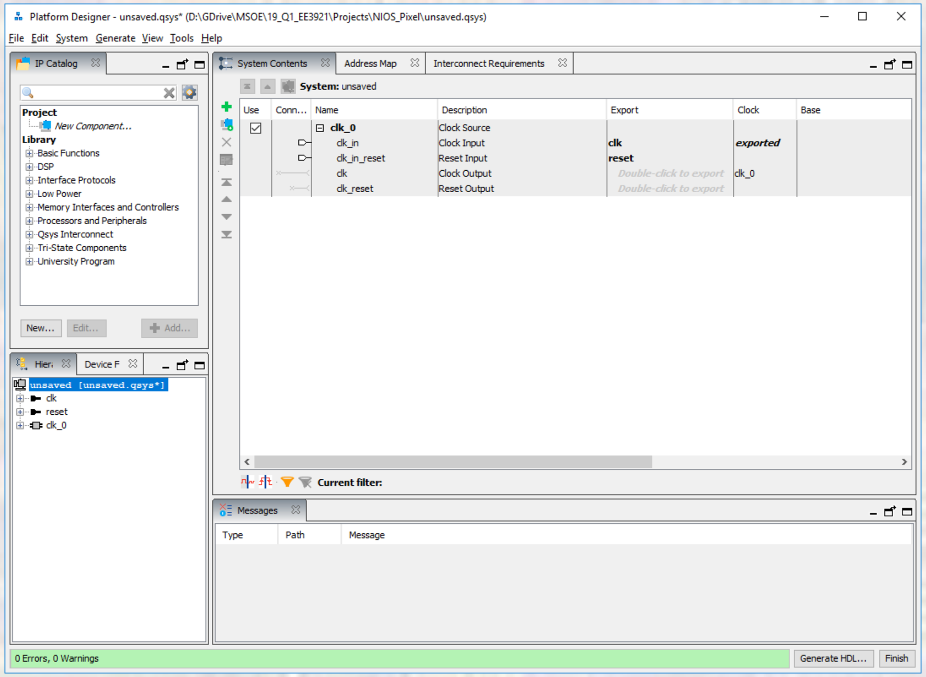
- NIOS I/O System
  - Create a processor system to display a count on LEDs with switch control for up/dn, pause



NIOS I/O

- 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)

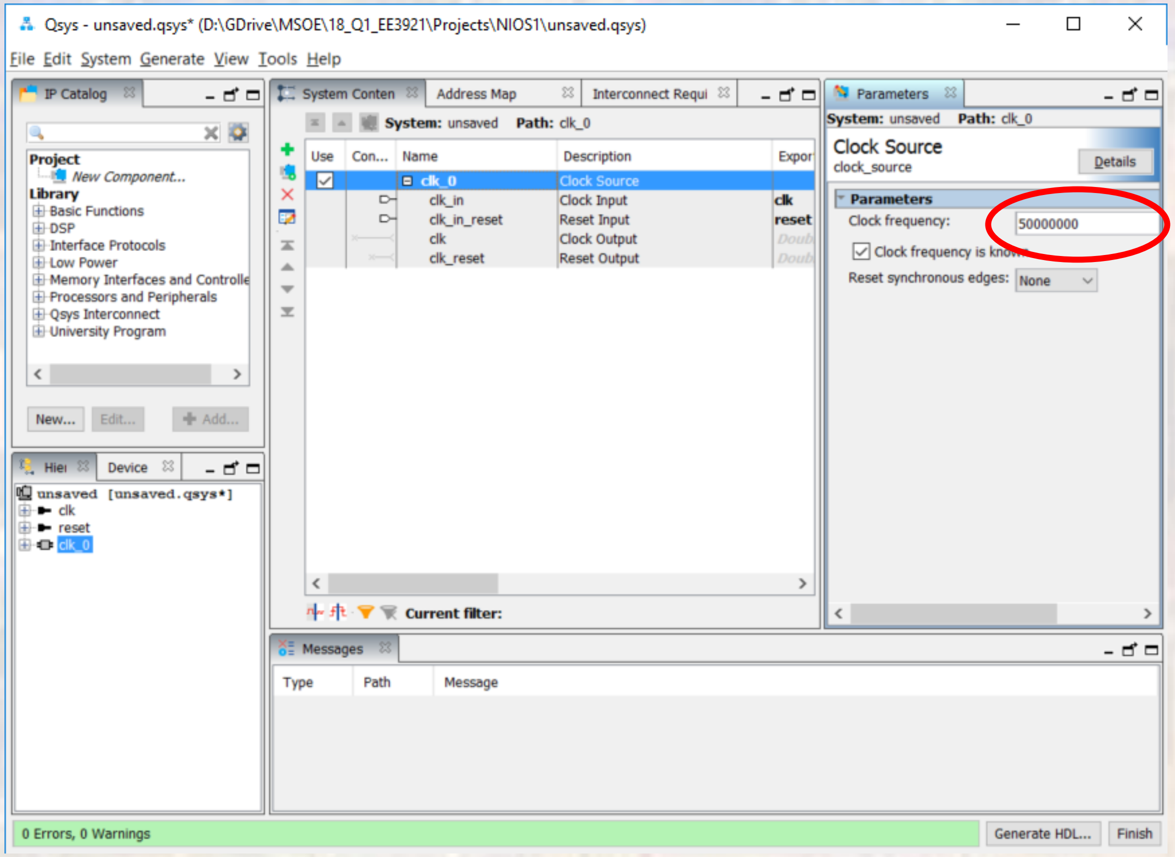
System: unsaved

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		

0 Errors, 0 Warnings

NIOS I/O

- Create NIOS System
  - Double Click on clk_0 - verify clk frequency = 50MHz



Qsys - unsaved.qsys* (D:\GDrive\MSOE\18_Q1_EE3921\Projects\NIOS1\unsaved.qsys)

System: unsaved Path: clk_0

Use	Con...	Name	Description	Expor
✓		clk_0	Clock Source	
		clk_in	Clock Input	clk
		clk_in_reset	Reset Input	reset
		clk	Clock Output	Doub
		clk_reset	Reset Output	Doub

Current filter:

System: unsaved Path: clk_0

Clock Source
clock_source
Details

Parameters
Clock frequency: 50000000
✓ Clock frequency is known
Reset synchronous edges: None

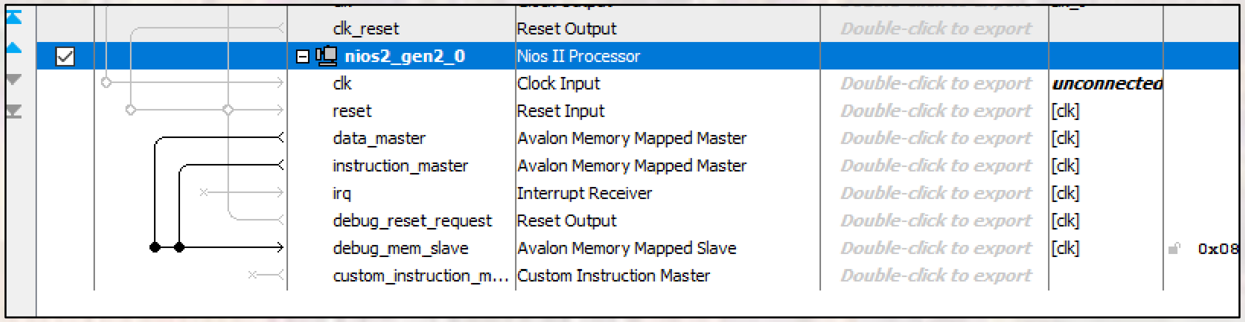
unsaved [unsaved.qsys*]
clk
reset
clk_0

Type	Path	Message

0 Errors, 0 Warnings

NIOS I/O

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

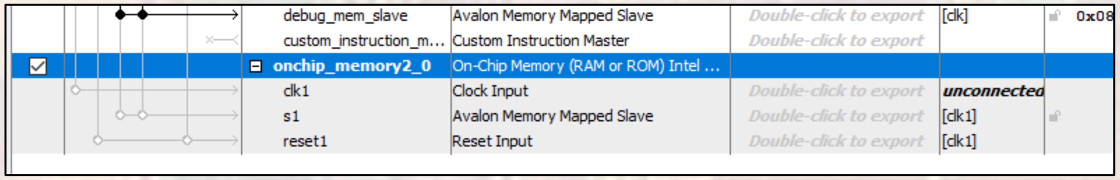


clk_reset	Reset Output	Double-click to export		
✓ nios2_gen2_0	Nios II Processor			
clk	Clock Input	Double-click to export	unconnected	
reset	Reset Input	Double-click to export	[clk]	
data_master	Avalon Memory Mapped Master	Double-click to export	[clk]	
instruction_master	Avalon Memory Mapped Master	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]	0x08
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

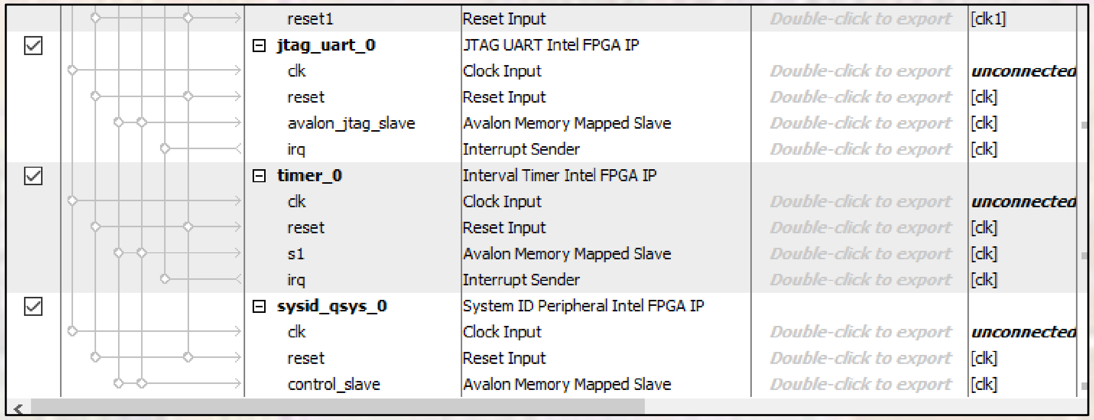
Size = 20,000 bytes



debug_mem_slave	Avalon Memory Mapped Slave	Double-click to export	[clk]	0x08
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	unconnected	
s1	Avalon Memory Mapped Slave	Double-click to export	[clk1]	
reset1	Reset Input	Double-click to export	[clk1]	

NIOS I/O

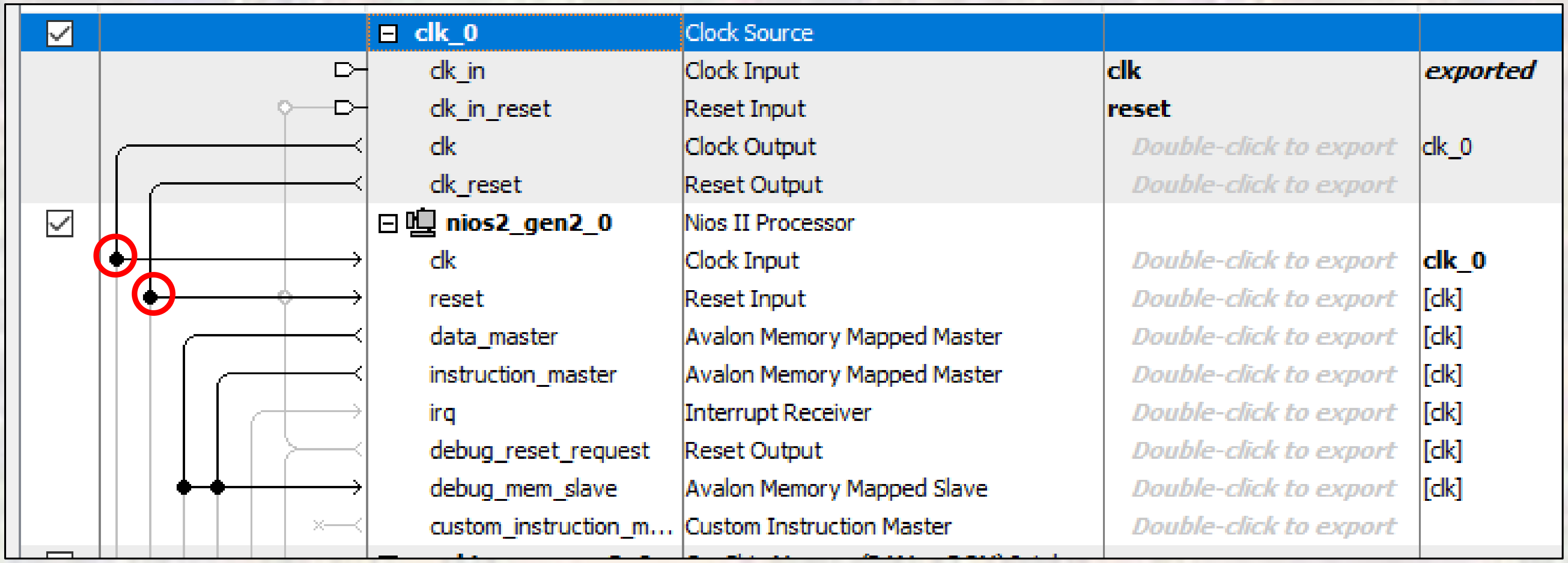
- Add JTAG
  - Interface Protocols → Serial→ JTAG Uart Intel FPGA IP
- Add Timer
  - Processors and Peripherals → Peripherals → Interval Timer Intel FPGA IP
- Add System ID
  - Basic Functions → Simulation; Debug and Verification → Debug and Performance → System ID Peripheral Intel FPGA IP



	reset1	Reset Input	Double-click to export	[clk1]
☑	⊟ **jtag_uart_0**	JTAG UART Intel FPGA IP		
	clk	Clock Input	Double-click to export	*unconnected*
	reset	Reset Input	Double-click to export	[clk]
	avalon_jtag_slave	Avalon Memory Mapped Slave	Double-click to export	[clk]
	irq	Interrupt Sender	Double-click to export	[clk]
☑	⊟ **timer_0**	Interval Timer Intel FPGA IP		
	clk	Clock Input	Double-click to export	*unconnected*
	reset	Reset Input	Double-click to export	[clk]
	s1	Avalon Memory Mapped Slave	Double-click to export	[clk]
	irq	Interrupt Sender	Double-click to export	[clk]
☑	⊟ **sysid_qsys_0**	System ID Peripheral Intel FPGA IP		
	clk	Clock Input	Double-click to export	*unconnected*
	reset	Reset Input	Double-click to export	[clk]
	control_slave	Avalon Memory Mapped Slave	Double-click to export	[clk]

NIOS I/O

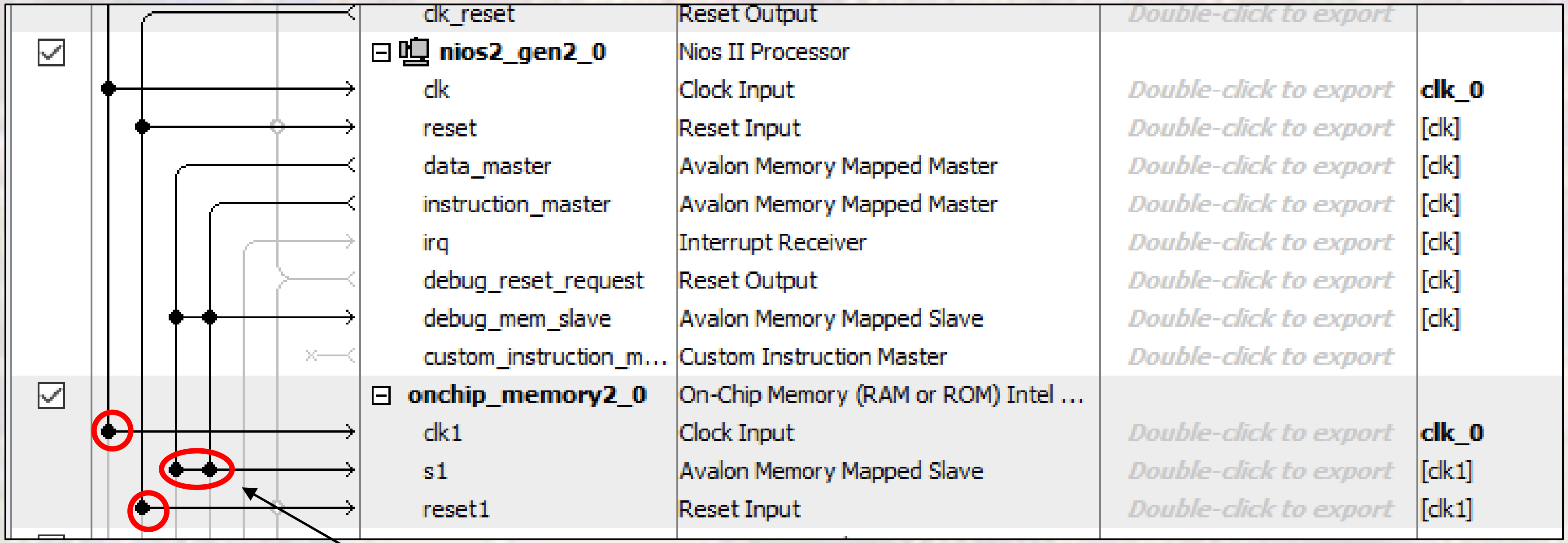
- Connect up NIOS I/O
  - NIOS Inputs



⊟ 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	
⊟ **nios2_gen2_0**	Nios II Processor		
clk	Clock Input	Double-click to export	**clk_0**
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]
custom_instruction_m...	Custom Instruction Master	Double-click to export	

NIOS I/O

- Connect up NIOS I/O
  - On-chip Memory

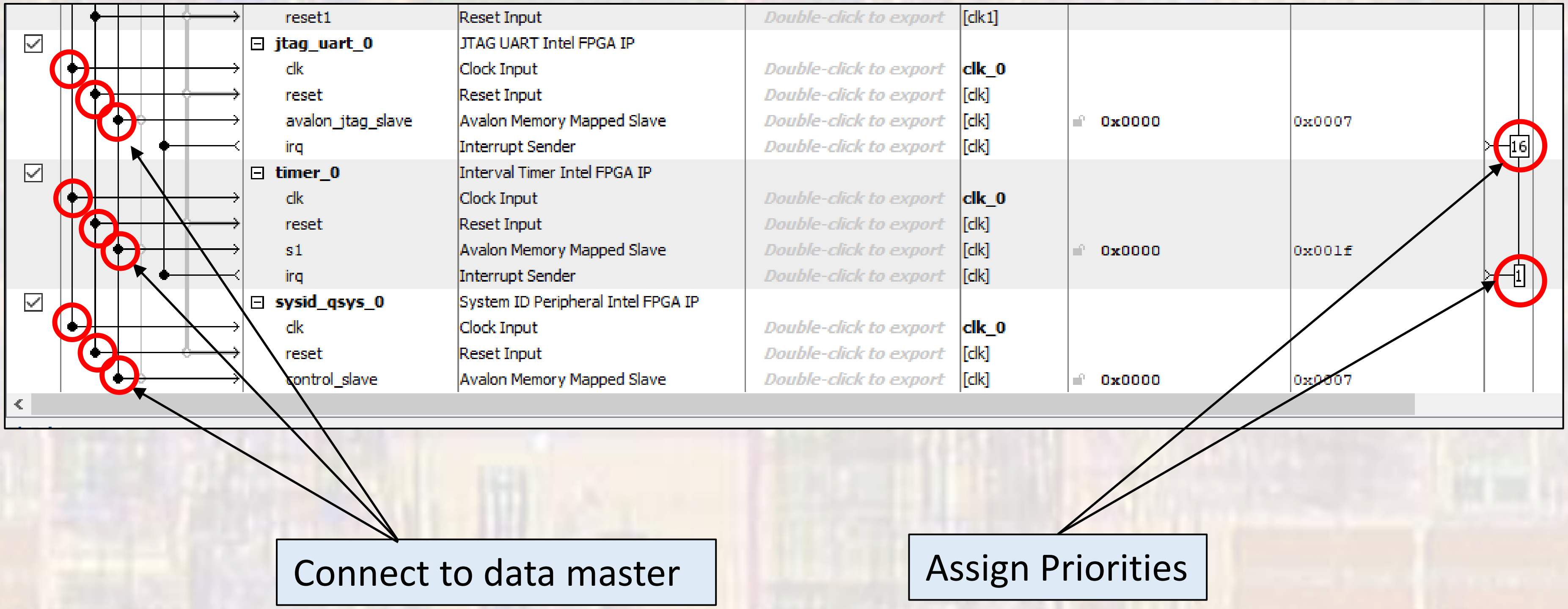


clk_reset	Reset Output	Double-click to export	
☑ nios2_gen2_0	Nios II Processor		
clk	Clock Input	Double-click to export	clk_0
reset	Reset Input	Double-click to export	[clk]
data_master	Avalon Memory Mapped Master	Double-click to export	[clk]
instruction_master	Avalon Memory Mapped Master	Double-click 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]
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	clk_0
s1	Avalon Memory Mapped Slave	Double-click to export	[clk1]
reset1	Reset Input	Double-click to export	[clk1]

Connect to data and instruction masters

NIOS I/O

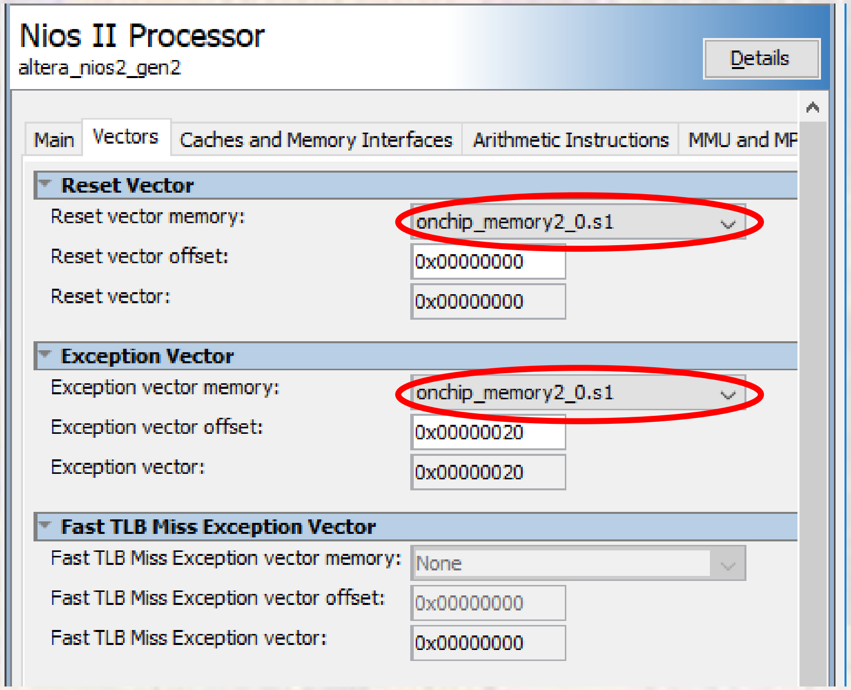
- Connect up NIOS I/O
  - JTAG, Timer, SysID



	reset1	Reset Input	Double-click to export	[clk1]			
☑	⊟ jtag_uart_0	JTAG UART Intel FPGA IP					
	clk	Clock Input	Double-click to export	clk_0			
	reset	Reset Input	Double-click to export	[clk]			
	avalon_jtag_slave	Avalon Memory Mapped Slave	Double-click to export	[clk]	0x0000	0x0007	
	irq	Interrupt Sender	Double-click to export	[clk]			16
☑	⊟ timer_0	Interval Timer Intel FPGA IP					
	clk	Clock Input	Double-click to export	clk_0			
	reset	Reset Input	Double-click to export	[clk]			
	s1	Avalon Memory Mapped Slave	Double-click to export	[clk]	0x0000	0x001f	
	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	clk_0			
	reset	Reset Input	Double-click to export	[clk]			
	control_slave	Avalon Memory Mapped Slave	Double-click to export	[clk]	0x0000	0x0007	

NIOS I/O

- Connect up NIOS I/O
  - Assign the NIOS II Reset and Exception vectors
    - Open the NIOS Processor
    - Select Vectors
    - Select on-chip memory for Reset and Exception



Reset Vector

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

Exception Vector

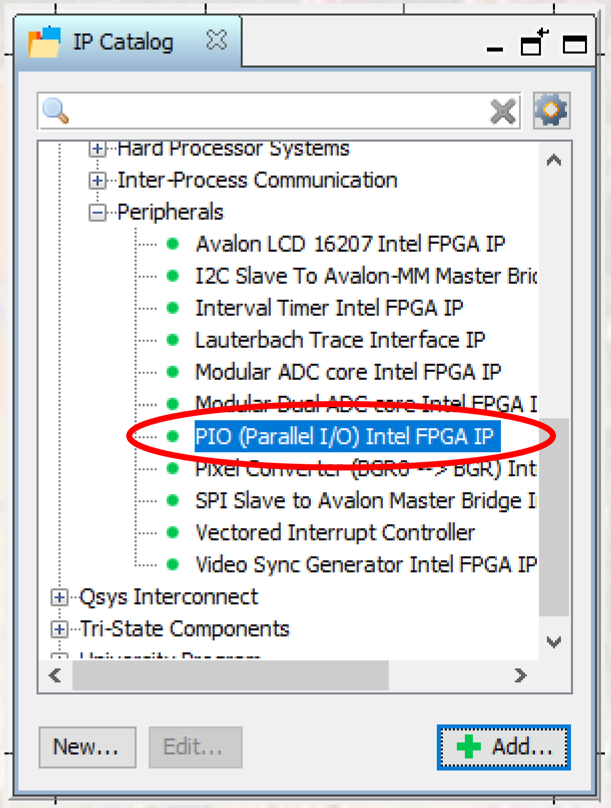
Exception vector memory:	onchip_memory2_0.s1
Exception vector offset:	0x00000020
Exception vector:	0x00000020

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

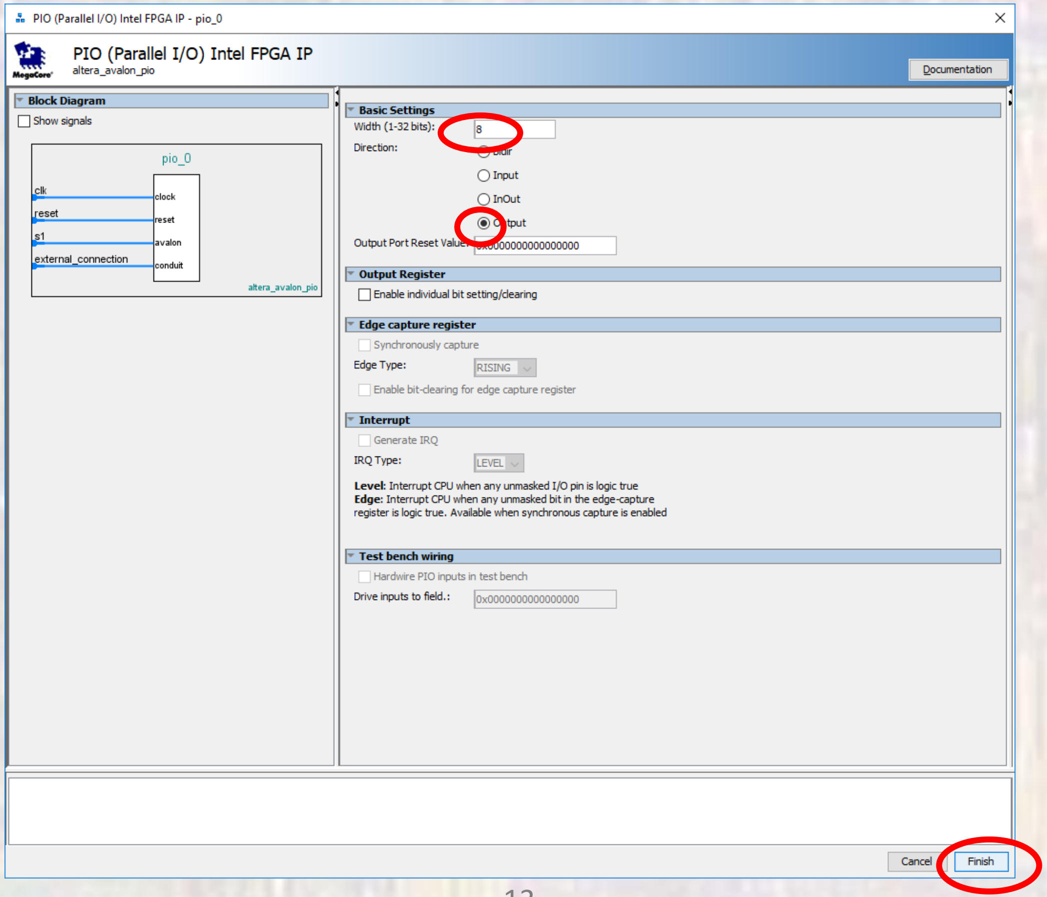
NIOS I/O

- Customize System
  - Create LED output
    - IP Catalog → Library → Processors and Peripherals→ Peripherals → PIO (Parallel I/O)


IP Catalog
Hard Processor Systems
Inter-Process Communication
Peripherals
Avalon LCD 16207 Intel FPGA IP
I2C Slave To Avalon-MM Master Bric
Interval Timer Intel FPGA IP
Lauterbach Trace Interface IP
Modular ADC core Intel FPGA IP
Modular Dual ADC core Intel FPGA I
PIO (Parallel I/O) Intel FPGA IP
Pixel Converter (BGR0 --> BGR) Int
SPI Slave to Avalon Master Bridge I
Vectored Interrupt Controller
Video Sync Generator Intel FPGA IP
Qsys Interconnect
Tri-State Components
New...
Edit...
Add...

NIOS I/O

- Customize System
  - Create LED output
    - Set 8 bit, select output



PIO (Parallel I/O) Intel FPGA IP - pio_0

PIO (Parallel I/O) Intel FPGA IP
altera_avalon_pio

Basic Settings

Width (1-32 bits): 8

Direction: Bidir / Input / InOut / [✓ Output]

Output Port Reset Value: 0x0000000000000000

Output Register

☐ Enable individual bit setting/clearing

Edge capture register

☐ Synchronously capture

Edge Type: RISING

☐ Enable bit-clearing for edge capture register

Interrupt

☐ Generate IRQ

IRQ Type: LEVEL

Level: Interrupt CPU when any unmasked I/O pin is logic true
Edge: Interrupt CPU when any unmasked bit in the edge-capture register is logic true. Available when synchronous capture is enabled

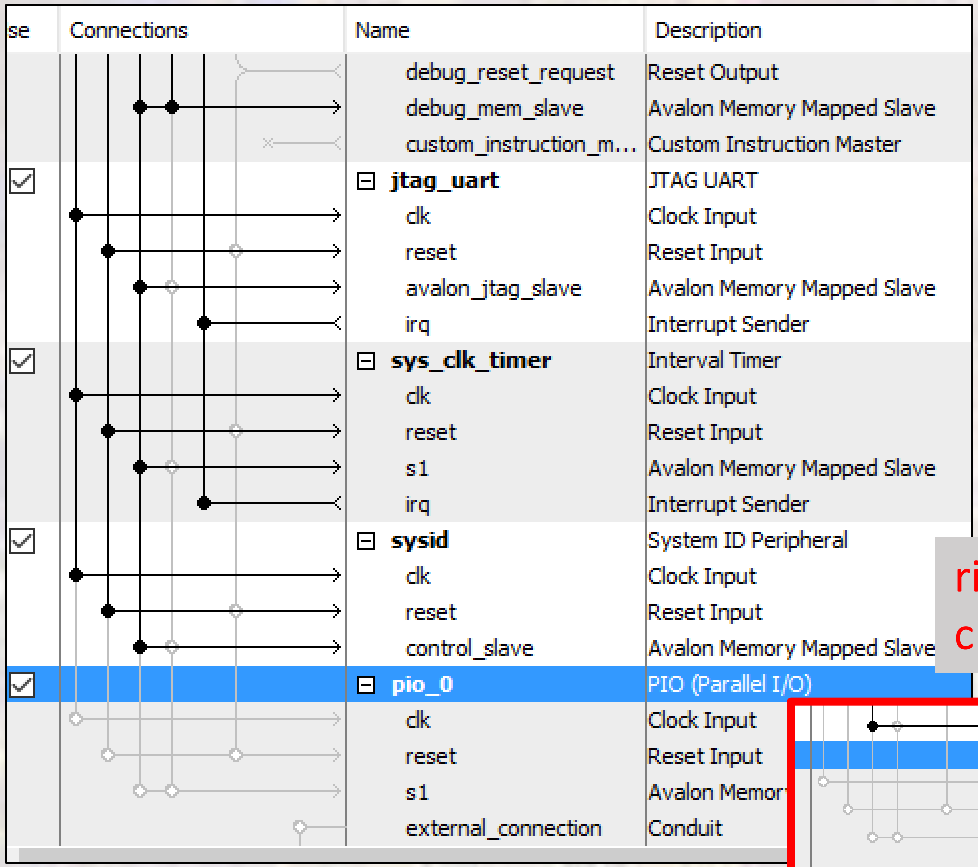
Test bench wiring

☐ Hardwire PIO inputs in test bench

Drive inputs to field.: 0x0000000000000000

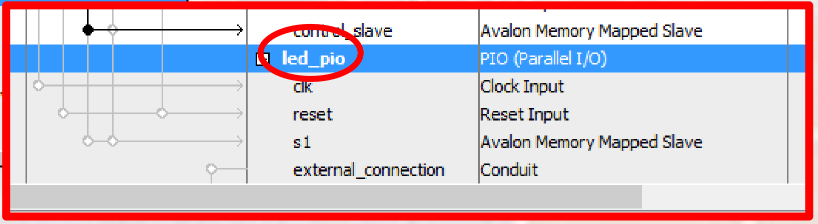
NIOS I/O

- Customize System
  - Hookup LED output



Use	Connections	Name	Description
		debug_reset_request	Reset Output
		debug_mem_slave	Avalon Memory Mapped Slave
		custom_instruction_m...	Custom Instruction Master
✓		⊟ **jtag_uart**	JTAG UART
		clk	Clock Input
		reset	Reset Input
		avalon_jtag_slave	Avalon Memory Mapped Slave
		irq	Interrupt Sender
✓		⊟ **sys_clk_timer**	Interval Timer
		clk	Clock Input
		reset	Reset Input
		s1	Avalon Memory Mapped Slave
		irq	Interrupt Sender
✓		⊟ **sysid**	System ID Peripheral
		clk	Clock Input
		reset	Reset Input
		control_slave	Avalon Memory Mapped Slave
✓		⊟ **pio_0**	PIO (Parallel I/O)
		clk	Clock Input
		reset	Reset Input
		s1	Avalon Memor
		external_connection	Conduit

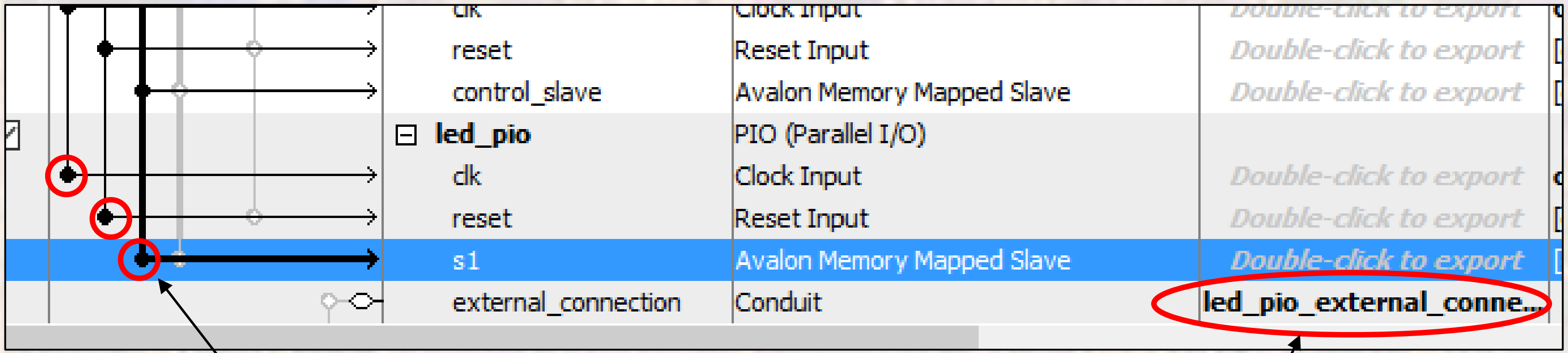
right click the PIO name and change it to led_pio



Name	Description
control_slave	Avalon Memory Mapped Slave
led_pio	PIO (Parallel I/O)
clk	Clock Input
reset	Reset Input
s1	Avalon Memory Mapped Slave
external_connection	Conduit

NIOS I/O

- Customize System
  - Hookup LED output



clk	Clock Input	Double-click to export
reset	Reset Input	Double-click to export
control_slave	Avalon Memory Mapped Slave	Double-click to export
⊟ **led_pio**	PIO (Parallel I/O)	
clk	Clock Input	Double-click to export
reset	Reset Input	Double-click to export
s1	Avalon Memory Mapped Slave	Double-click to export
external_connection	Conduit	**led_pio_external_conne...**

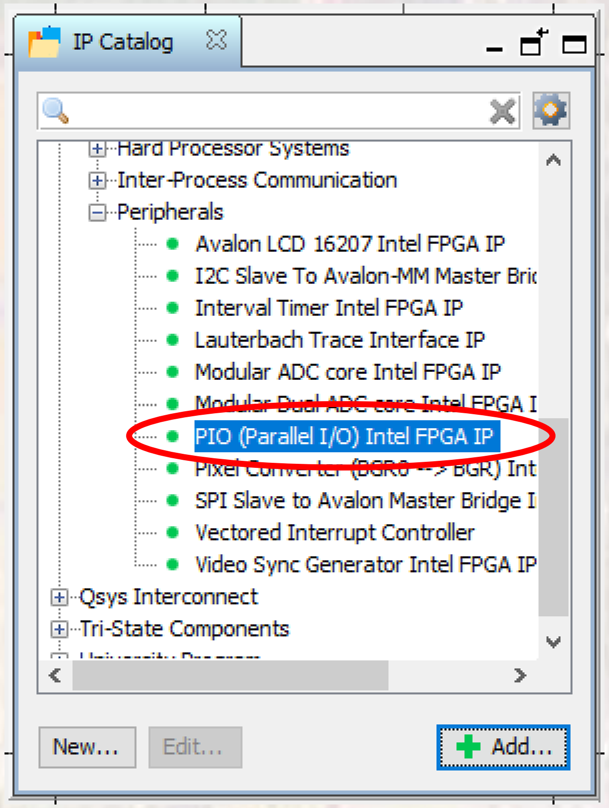
Connect to data master

Double click to export

Makes the connection visible outside the system

NIOS I/O

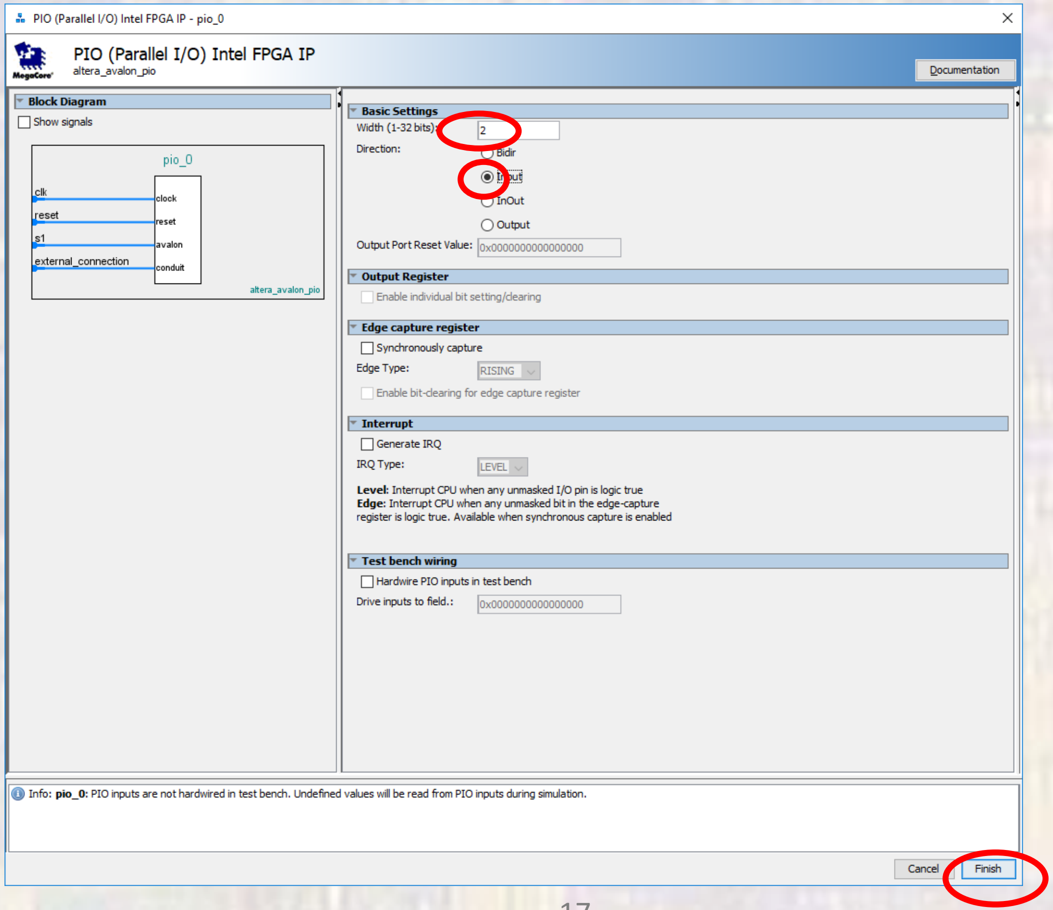
- Customize System
  - Create switch inputs
    - IP Catalog → Library → Processors and Peripherals→ Peripherals → PIO (Parallel I/O)



IP Catalog
Hard Processor Systems
Inter-Process Communication
Peripherals
Avalon LCD 16207 Intel FPGA IP
I2C Slave To Avalon-MM Master Bri
Interval Timer Intel FPGA IP
Lauterbach Trace Interface IP
Modular ADC core Intel FPGA IP
Modular Dual ADC core Intel FPGA I
PIO (Parallel I/O) Intel FPGA IP
Pixel Converter (BGR0 --> BGR) Int
SPI Slave to Avalon Master Bridge I
Vectored Interrupt Controller
Video Sync Generator Intel FPGA IP
Qsys Interconnect
Tri-State Components
New...
Edit...
Add...

NIOS I/O

- Customize System
  - Create switch inputs
    - Set 2 bit, select input



PIO (Parallel I/O) Intel FPGA IP - pio_0

PIO (Parallel I/O) Intel FPGA IP
altera_avalon_pio

Block Diagram
Show signals

Basic Settings
Width (1-32 bits): 2
Direction: Bidir [✓ Input] InOut Output
Output Port Reset Value: 0x0000000000000000

Output Register
Enable individual bit setting/clearing

Edge capture register
Synchronously capture
Edge Type: RISING
Enable bit-clearing for edge capture register

Interrupt
Generate IRQ
IRQ Type: LEVEL

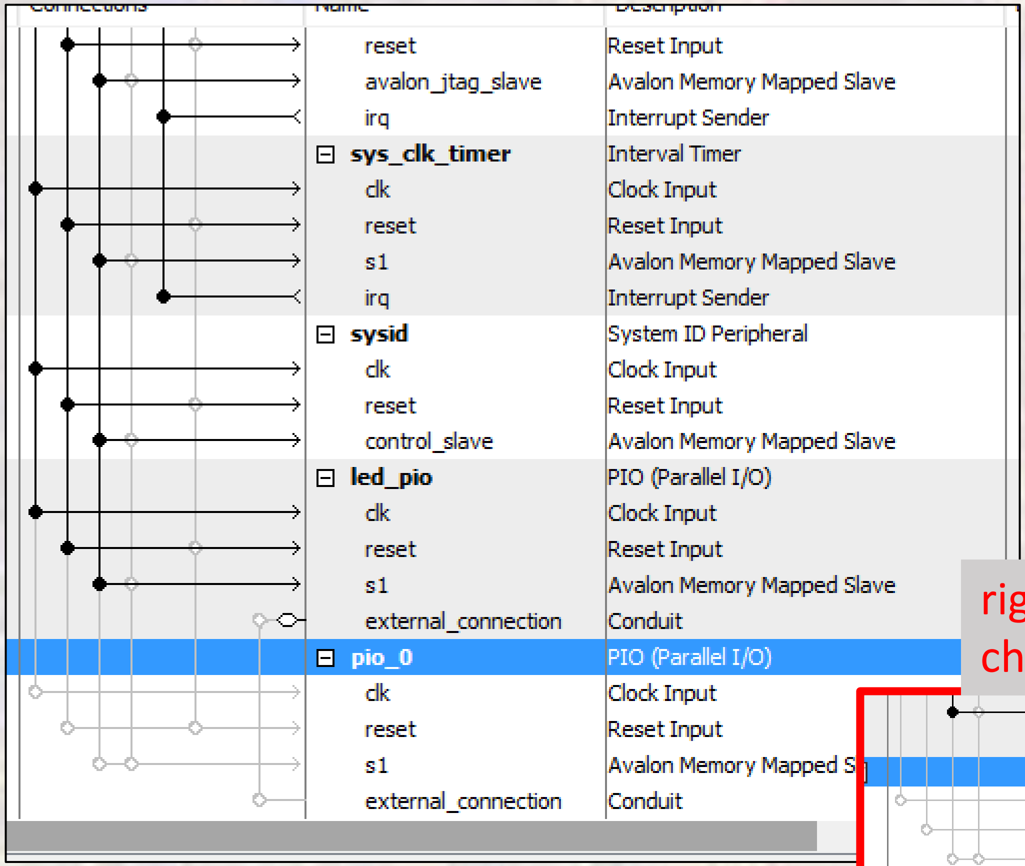
Level: Interrupt CPU when any unmasked I/O pin is logic true
Edge: Interrupt CPU when any unmasked bit in the edge-capture register is logic true. Available when synchronous capture is enabled

Test bench wiring
Hardwire PIO inputs in test bench
Drive inputs to field.: 0x0000000000000000

Info: **pio_0:** PIO inputs are not hardwired in test bench. Undefined values will be read from PIO inputs during simulation.

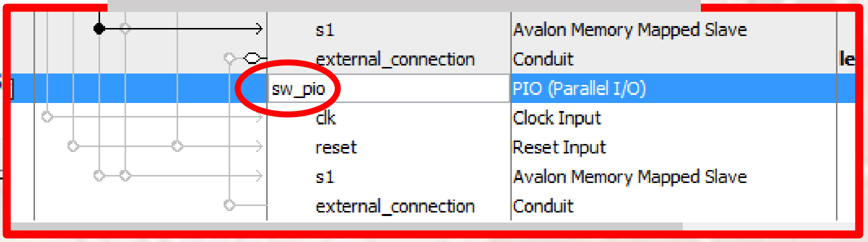
NIOS I/O

- Customize System
  - Hookup switch input



reset	Reset Input
avalon_jtag_slave	Avalon Memory Mapped Slave
irq	Interrupt Sender
sys_clk_timer	Interval Timer
clk	Clock Input
reset	Reset Input
s1	Avalon Memory Mapped Slave
irq	Interrupt Sender
sysid	System ID Peripheral
clk	Clock Input
reset	Reset Input
control_slave	Avalon Memory Mapped Slave
led_pio	PIO (Parallel I/O)
clk	Clock Input
reset	Reset Input
s1	Avalon Memory Mapped Slave
external_connection	Conduit
pio_0	PIO (Parallel I/O)
clk	Clock Input
reset	Reset Input
s1	Avalon Memory Mapped S
external_connection	Conduit

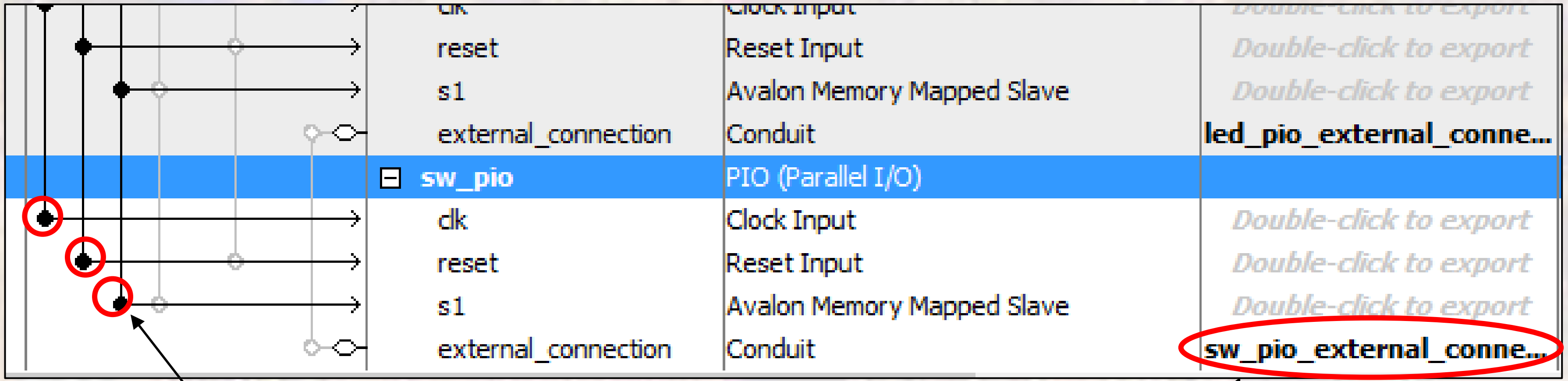
right click the PIO name and change it to sw_pio



s1	Avalon Memory Mapped Slave
external_connection	Conduit
sw_pio	PIO (Parallel I/O)
clk	Clock Input
reset	Reset Input
s1	Avalon Memory Mapped Slave
external_connection	Conduit

NIOS I/O

- Customize System
  - Hookup switch input



reset	Reset Input	Double-click to export
s1	Avalon Memory Mapped Slave	Double-click to export
external_connection	Conduit	led_pio_external_conne...
sw_pio	PIO (Parallel I/O)	
clk	Clock Input	Double-click to export
reset	Reset Input	Double-click to export
s1	Avalon Memory Mapped Slave	Double-click to export
external_connection	Conduit	sw_pio_external_conne...

Connect to data master

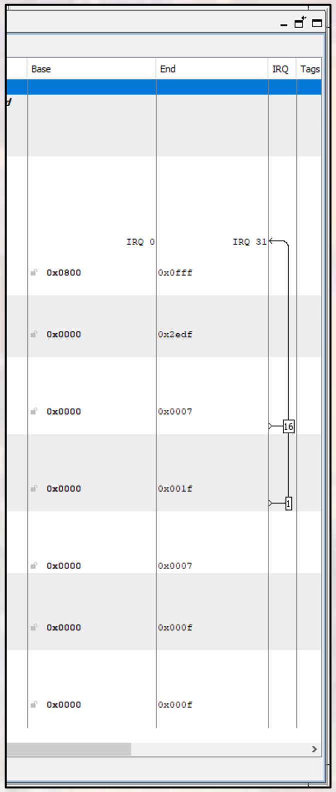
Double click to export

Makes the connection visible outside the system

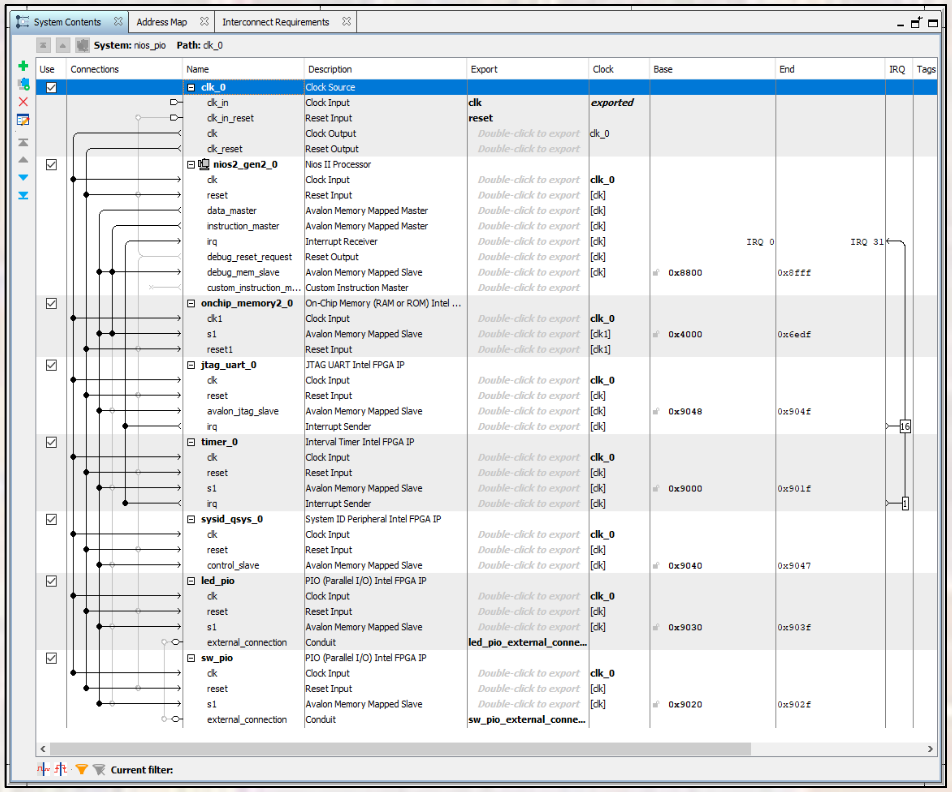
NIOS I/O

- Complete Custom System
  - Assign base addresses

  - System → Assign Base Addresses



Base	End	IRQ	Tags
	IRQ 0	IRQ 31	
0x0800	0x0fff		
0x0000	0x2edf		
0x0000	0x0007	16	
0x0000	0x001f	1	
0x0000	0x0007		
0x0000	0x000f		
0x0000	0x000f		



System Contents | Address Map | Interconnect Requirements

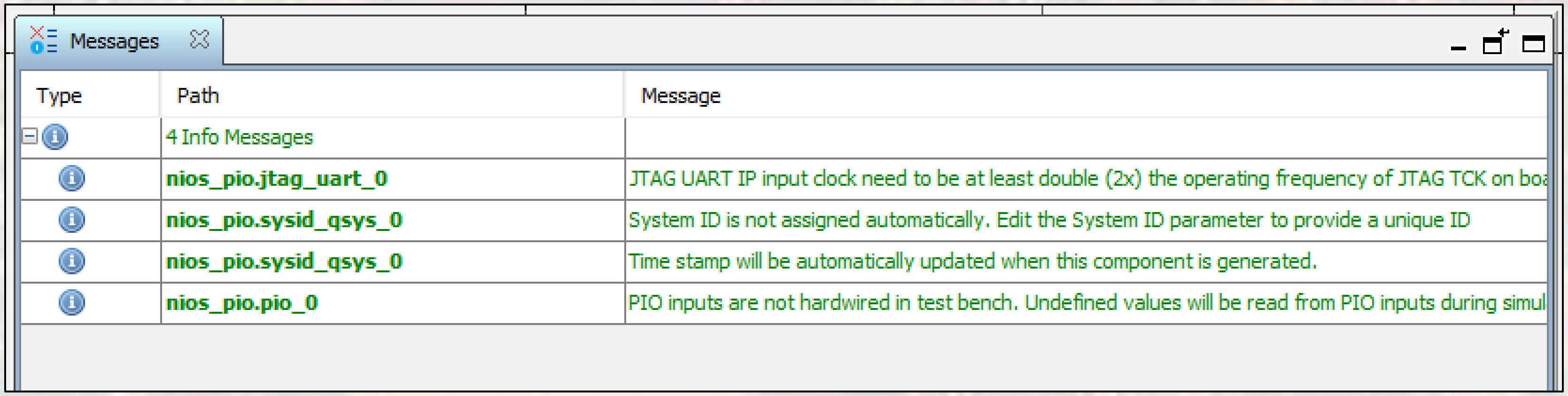
System: nios_pio **Path:** clk_0

Use	Name	Description	Export	Clock	Base	End	IRQ	Tags
☑	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					
☑	nios2_gen2_0	Nios II Processor						
	clk	Clock Input	Double-click to export	clk_0				
	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]	0x8800	0x8fff		
	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	clk_0				
	s1	Avalon Memory Mapped Slave	Double-click to export	[clk1]	0x4000	0x6edf		
	reset1	Reset Input	Double-click to export	[clk1]				
☑	jtag_uart_0	JTAG UART Intel FPGA IP						
	clk	Clock Input	Double-click to export	clk_0				
	reset	Reset Input	Double-click to export	[clk]				
	avalon_jtag_slave	Avalon Memory Mapped Slave	Double-click to export	[clk]	0x9048	0x904f		
	irq	Interrupt Sender	Double-click to export	[clk]			16	
☑	timer_0	Interval Timer Intel FPGA IP						
	clk	Clock Input	Double-click to export	clk_0				
	reset	Reset Input	Double-click to export	[clk]				
	s1	Avalon Memory Mapped Slave	Double-click to export	[clk]	0x9000	0x901f		
	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	clk_0				
	reset	Reset Input	Double-click to export	[clk]				
	control_slave	Avalon Memory Mapped Slave	Double-click to export	[clk]	0x9040	0x9047		
☑	led_pio	PIO (Parallel I/O) Intel FPGA IP						
	clk	Clock Input	Double-click to export	clk_0				
	reset	Reset Input	Double-click to export	[clk]				
	s1	Avalon Memory Mapped Slave	Double-click to export	[clk]	0x9030	0x903f		
	external_connection	Conduit	led_pio_external_conne...					
☑	sw_pio	PIO (Parallel I/O) Intel FPGA IP						
	clk	Clock Input	Double-click to export	clk_0				
	reset	Reset Input	Double-click to export	[clk]				
	s1	Avalon Memory Mapped Slave	Double-click to export	[clk]	0x9020	0x902f		
	external_connection	Conduit	sw_pio_external_conne...					

Current filter:

NIOS I/O

- Create Custom System
  - Check for errors

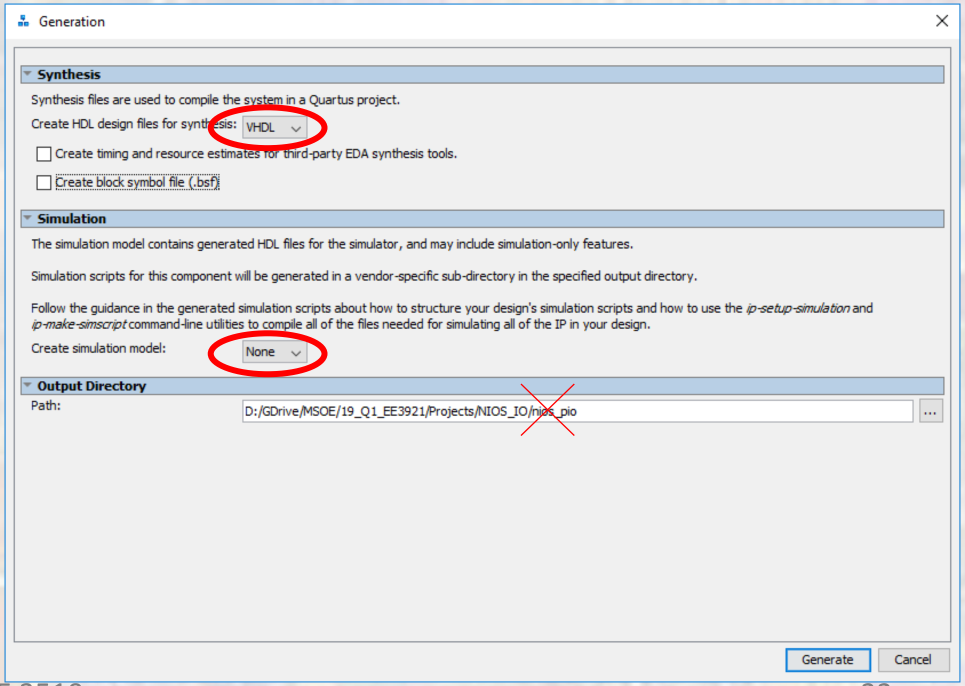


Messages

Type	Path	Message
⊟ ⓘ	4 Info Messages	
ⓘ	nios_pio.jtag_uart_0	JTAG UART IP input clock need to be at least double (2x) the operating frequency of JTAG TCK on boa
ⓘ	nios_pio.sysid_qsys_0	System ID is not assigned automatically. Edit the System ID parameter to provide a unique ID
ⓘ	nios_pio.sysid_qsys_0	Time stamp will be automatically updated when this component is generated.
ⓘ	nios_pio.pio_0	PIO inputs are not hardwired in test bench. Undefined values will be read from PIO inputs during simul

NIOS I/O

- Create Basic System
  - Save the Platform Designer system
  - Generate the Platform Designer system
    - Generate → Generate HDL
    - 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.

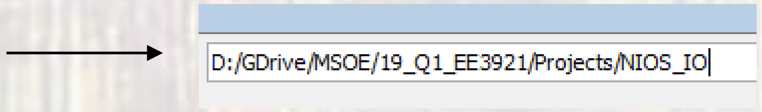
Follow the guidance in the generated simulation scripts about how to structure your design's simulation scripts and how to use the *ip-setup-simulation* and *ip-make-simscript* command-line utilities to compile all of the files needed for simulating all of the IP in your design.

Create simulation model: None

Output Directory

Path: D:/GDrive/MSOE/19_Q1_EE3921/Projects/NIOS_IO/~~nios_pio~~

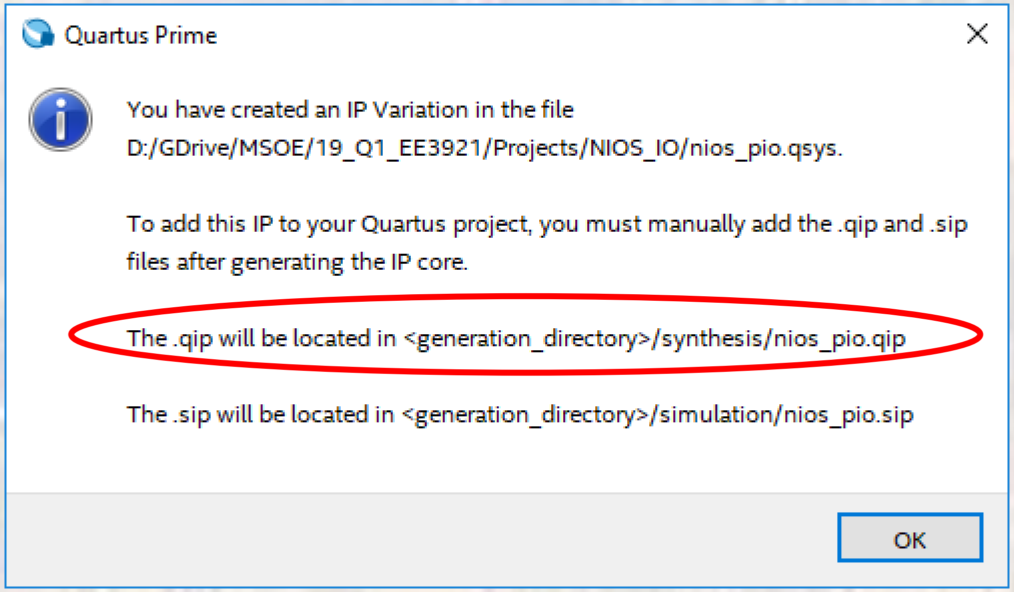
Should point to your project directory



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

NIOS I/O

- Create Custom 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_IO/nios_pio.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_pio.qip

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

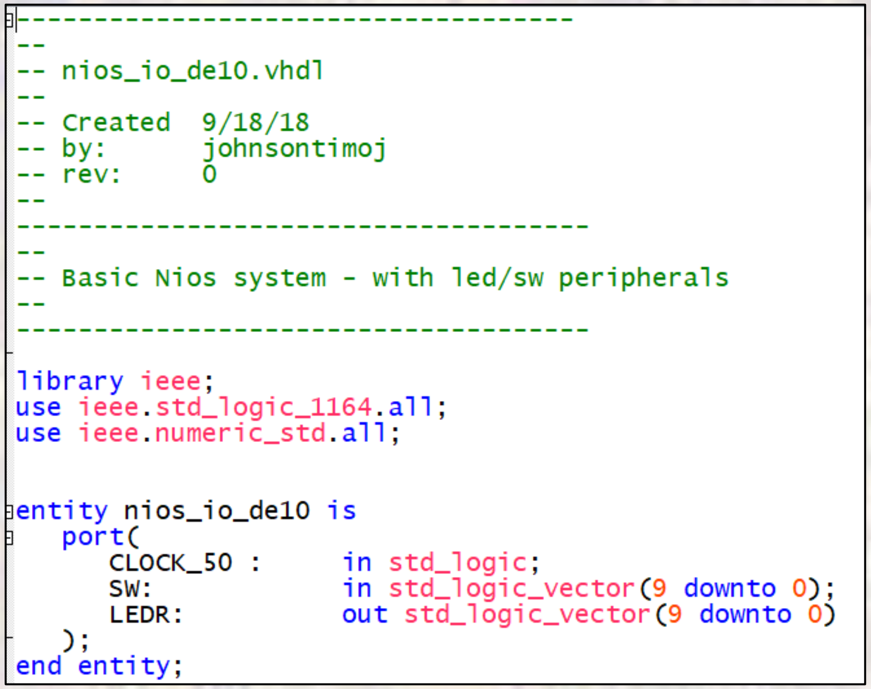
NIOS I/O

- 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_pio is
  port (
    clk_clk                          : in  std_logic                    := 'X';             -- clk
    led_pio_external_connection_export : out std_logic_vector(7 downto 0);                   -- export
    reset_reset_n                    : in  std_logic                    := 'X';             -- reset_n
    sw_pio_external_connection_export  : in  std_logic_vector(1 downto 0) := (others => 'X')  -- export
  );
end component nios_pio;

u0 : component nios_pio
  port map (
    clk_clk                          => CONNECTED_TO_clk_clk,                          --                       clk.clk
    led_pio_external_connection_export => CONNECTED_TO_led_pio_external_connection_export, -- led_pio_external_connection.export
    reset_reset_n                    => CONNECTED_TO_reset_reset_n,                    --                     reset.reset_n
    sw_pio_external_connection_export  => CONNECTED_TO_sw_pio_external_connection_export   --  sw_pio_external_connection.export
  );
```

NIOS I/O

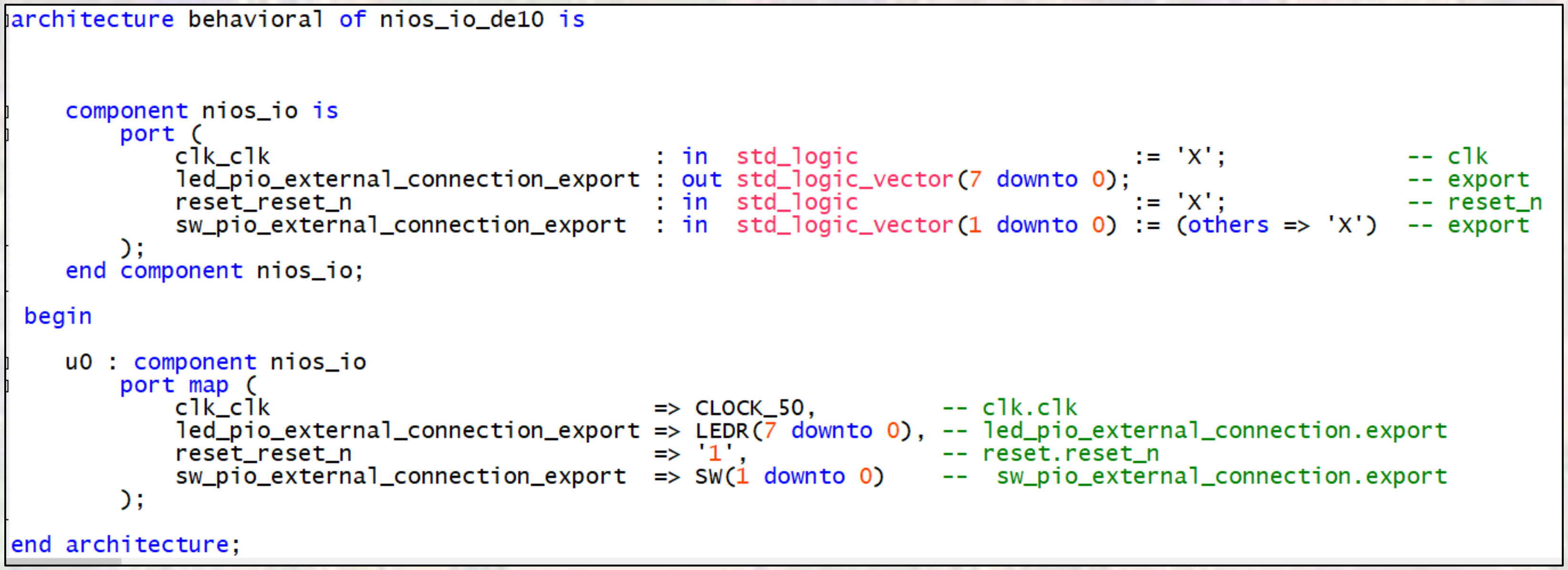
- Create DE10 Design
  - Instantiate into a VHDL file



```
-------------------------------------
--
-- nios_io_de10.vhdl
--
-- Created  9/18/18
-- by:      johnsontimoj
-- rev:     0
--
--------------------------------------
--
-- Basic Nios system - with led/sw peripherals
--
--------------------------------------

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


entity nios_io_de10 is
   port(
      CLOCK_50 :     in std_logic;
      SW:            in std_logic_vector(9 downto 0);
      LEDR:          out std_logic_vector(9 downto 0)
   );
end entity;
```

NIOS I/O

- Create DE10 Design
  - Instantiate into a VHDL file



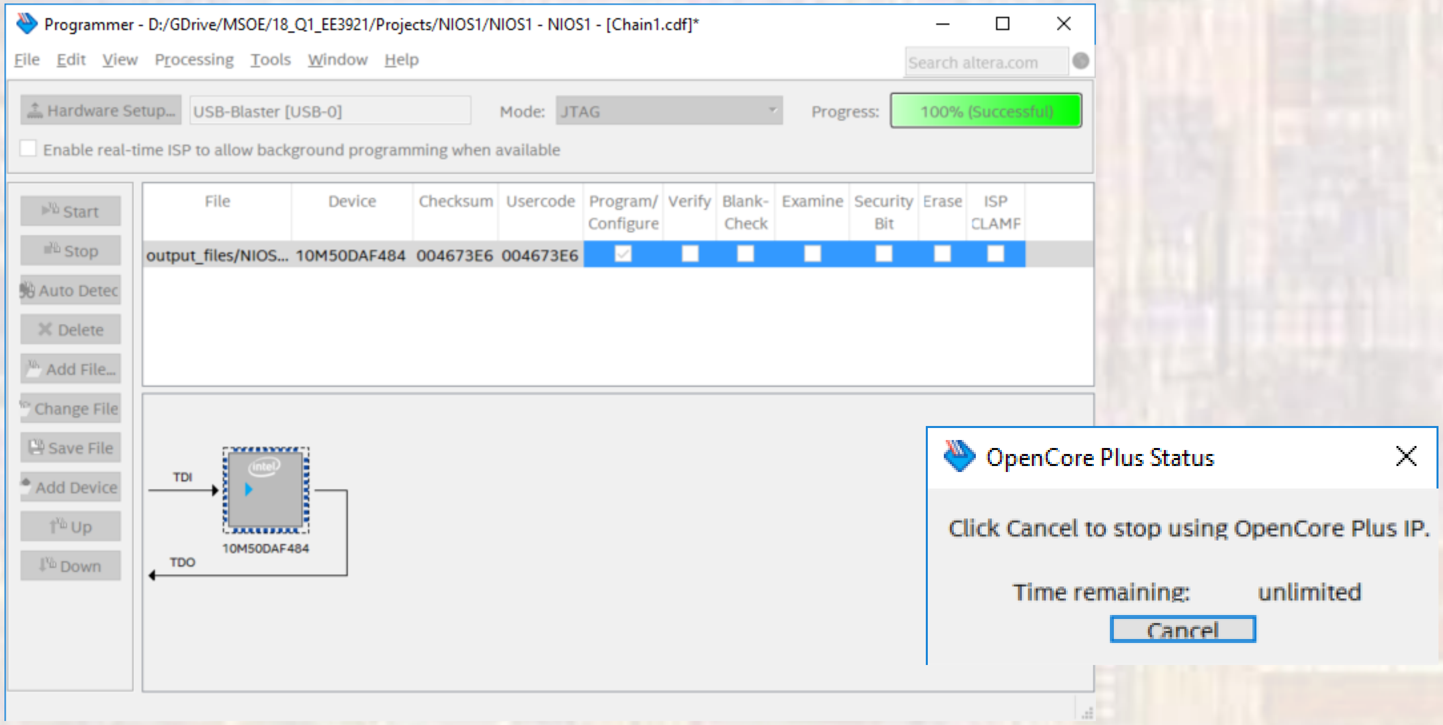
```
architecture behavioral of nios_io_de10 is


    component nios_io is
        port (
            clk_clk                            : in  std_logic                    := 'X';             -- clk
            led_pio_external_connection_export : out std_logic_vector(7 downto 0);                    -- export
            reset_reset_n                      : in  std_logic                    := 'X';             -- reset_n
            sw_pio_external_connection_export  : in  std_logic_vector(1 downto 0) := (others => 'X')  -- export
        );
    end component nios_io;

 begin

    u0 : component nios_io
        port map (
            clk_clk                            => CLOCK_50,          -- clk.clk
            led_pio_external_connection_export => LEDR(7 downto 0),  -- led_pio_external_connection.export
            reset_reset_n                      => '1',               -- reset.reset_n
            sw_pio_external_connection_export  => SW(1 downto 0)     --  sw_pio_external_connection.export
        );

end architecture;
```

NIOS I/O

- 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 I/O

- 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.

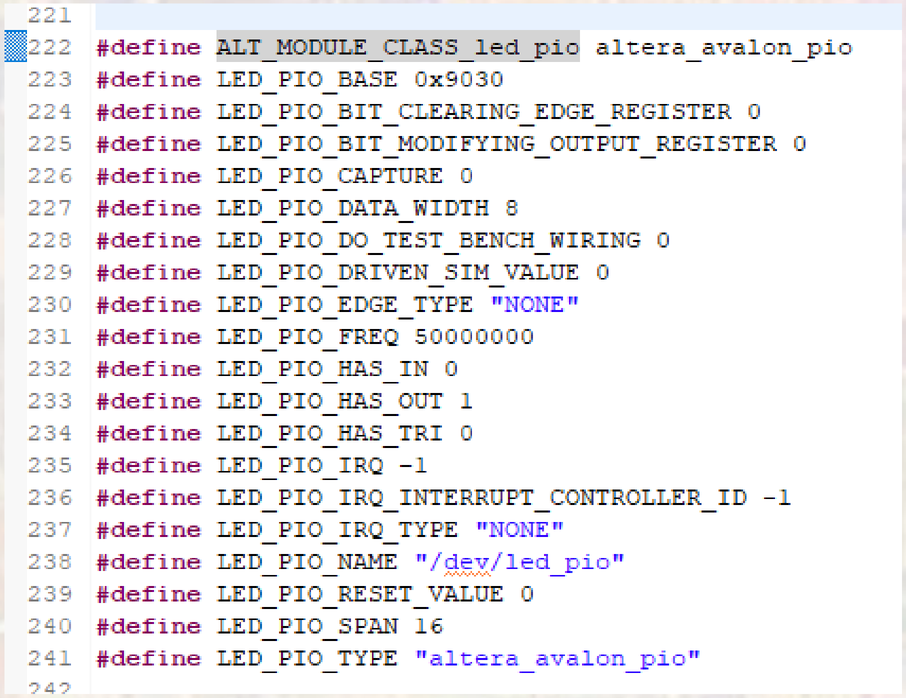
Time remaining: unlimited

NIOS I/O

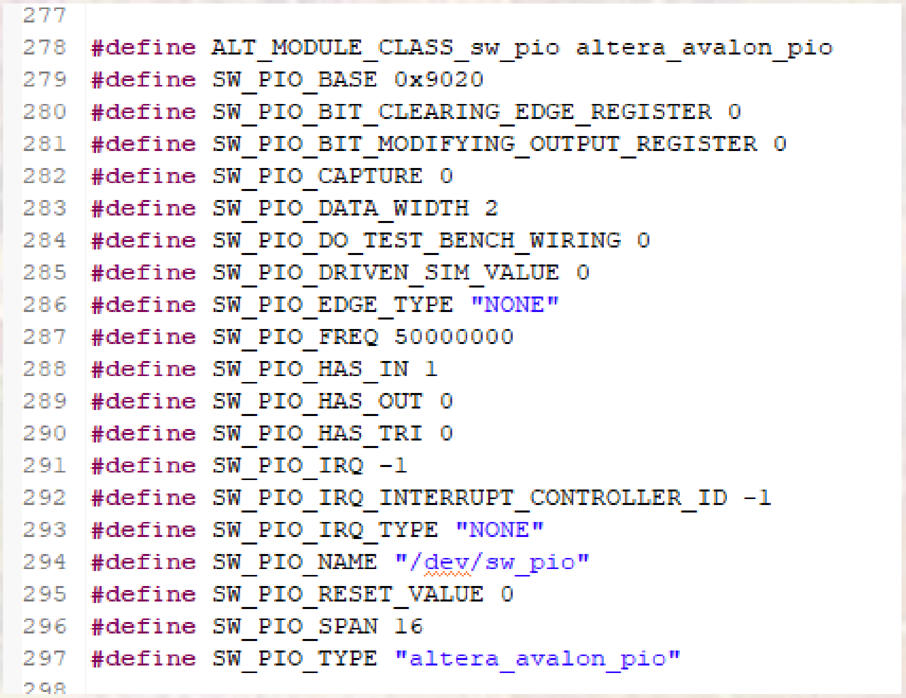
- Create Eclipse System
  - Open NIOSII software
    - Tools → NIOSII Software Build Tools for Eclipse
    - Select the project directory for the workspace
  - Create the BSP
    - File → New → NIOSII Application and BSP from template
    - Select the SOPCinfo file in the project directory
    - Provide a name for the sw project (I use 'project_name_sw')
    - Blank Project
  - Edit the BSP
    - Right click on the BSP, NIOS II → BSP Editor
    - Change the properties for small systems
      - Small C library
      - Reduced device drivers
  - Generate the BSP (bottom of window)

NIOS I/O

- Software setup
  - Open system.h in the bsp
    - scroll down to the PIOs
    - These define software parameters for the PIOs



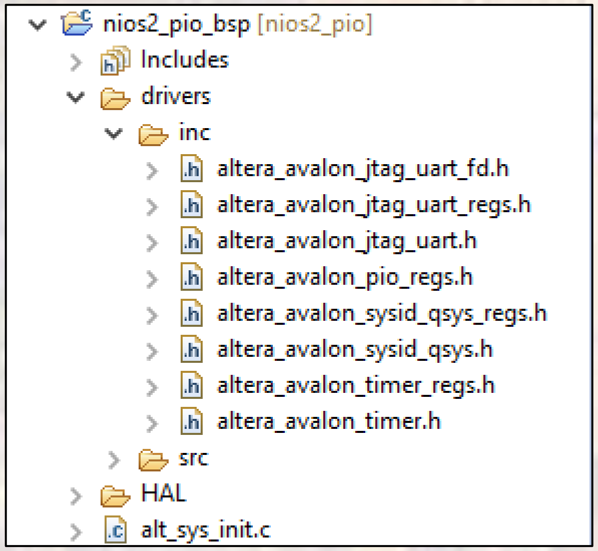
```
#define ALT_MODULE_CLASS_led_pio altera_avalon_pio
#define LED_PIO_BASE 0x9030
#define LED_PIO_BIT_CLEARING_EDGE_REGISTER 0
#define LED_PIO_BIT_MODIFYING_OUTPUT_REGISTER 0
#define LED_PIO_CAPTURE 0
#define LED_PIO_DATA_WIDTH 8
#define LED_PIO_DO_TEST_BENCH_WIRING 0
#define LED_PIO_DRIVEN_SIM_VALUE 0
#define LED_PIO_EDGE_TYPE "NONE"
#define LED_PIO_FREQ 50000000
#define LED_PIO_HAS_IN 0
#define LED_PIO_HAS_OUT 1
#define LED_PIO_HAS_TRI 0
#define LED_PIO_IRQ -1
#define LED_PIO_IRQ_INTERRUPT_CONTROLLER_ID -1
#define LED_PIO_IRQ_TYPE "NONE"
#define LED_PIO_NAME "/dev/led_pio"
#define LED_PIO_RESET_VALUE 0
#define LED_PIO_SPAN 16
#define LED_PIO_TYPE "altera_avalon_pio"
```



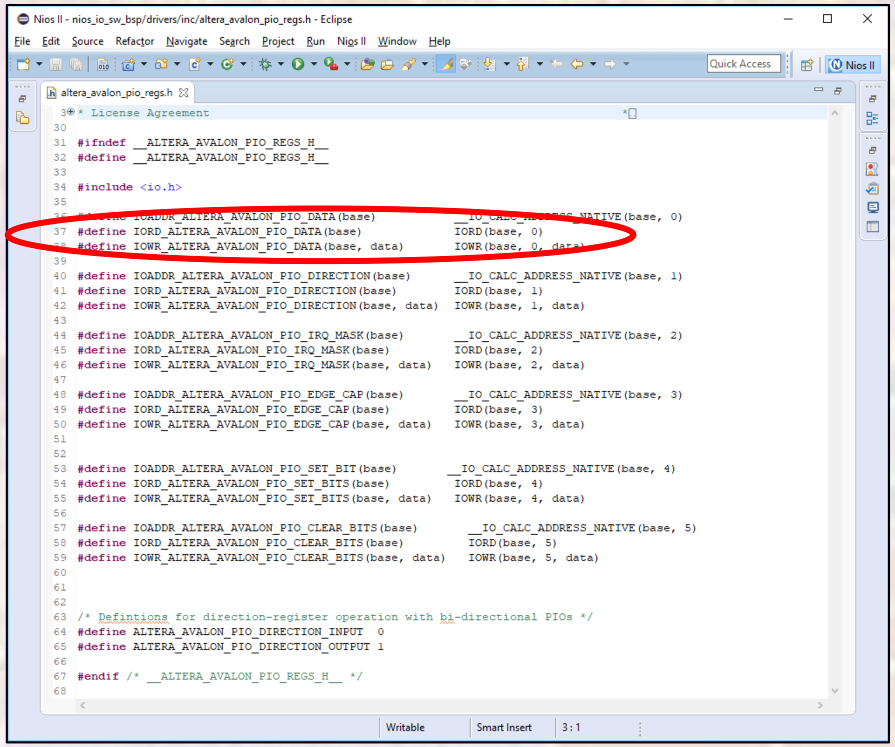
```
#define ALT_MODULE_CLASS_sw_pio altera_avalon_pio
#define SW_PIO_BASE 0x9020
#define SW_PIO_BIT_CLEARING_EDGE_REGISTER 0
#define SW_PIO_BIT_MODIFYING_OUTPUT_REGISTER 0
#define SW_PIO_CAPTURE 0
#define SW_PIO_DATA_WIDTH 2
#define SW_PIO_DO_TEST_BENCH_WIRING 0
#define SW_PIO_DRIVEN_SIM_VALUE 0
#define SW_PIO_EDGE_TYPE "NONE"
#define SW_PIO_FREQ 50000000
#define SW_PIO_HAS_IN 1
#define SW_PIO_HAS_OUT 0
#define SW_PIO_HAS_TRI 0
#define SW_PIO_IRQ -1
#define SW_PIO_IRQ_INTERRUPT_CONTROLLER_ID -1
#define SW_PIO_IRQ_TYPE "NONE"
#define SW_PIO_NAME "/dev/sw_pio"
#define SW_PIO_RESET_VALUE 0
#define SW_PIO_SPAN 16
#define SW_PIO_TYPE "altera_avalon_pio"
```

NIOS I/O

- Software setup
  - Expand drivers → inc in the bsp
    - Open altera_Avalon_pio_regs.h
    - PIO command are defined here



- nios2_pio_bsp [nios2_pio]
  - Includes
  - drivers
    - inc
      - altera_avalon_jtag_uart_fd.h
      - altera_avalon_jtag_uart_regs.h
      - altera_avalon_jtag_uart.h
      - altera_avalon_pio_regs.h
      - altera_avalon_sysid_qsys_regs.h
      - altera_avalon_sysid_qsys.h
      - altera_avalon_timer_regs.h
      - altera_avalon_timer.h
    - src
  - HAL
  - alt_sys_init.c

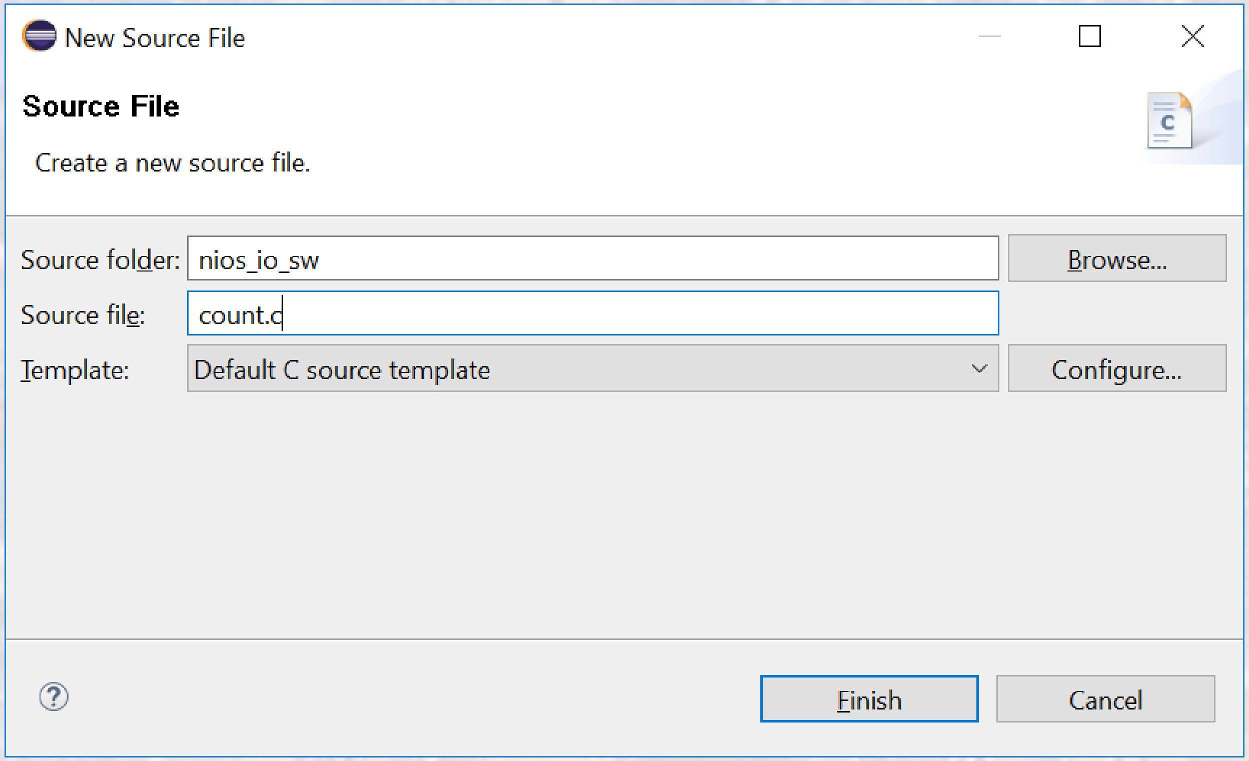


Nios II - nios_io_sw_bsp/drivers/inc/altera_avalon_pio_regs.h - Eclipse
File Edit Source Refactor Navigate Search Project Run Nios II Window Help
Quick Access
Nios II
altera_avalon_pio_regs.h
Writable
Smart Insert
3 : 1

```
* License Agreement

#ifndef __ALTERA_AVALON_PIO_REGS_H__
#define __ALTERA_AVALON_PIO_REGS_H__

#include <io.h>

#define IOADDR_ALTERA_AVALON_PIO_DATA(base)           __IO_CALC_ADDRESS_NATIVE(base, 0)
#define IORD_ALTERA_AVALON_PIO_DATA(base)             IORD(base, 0)
#define IOWR_ALTERA_AVALON_PIO_DATA(base, data)       IOWR(base, 0, data)

#define IOADDR_ALTERA_AVALON_PIO_DIRECTION(base)      __IO_CALC_ADDRESS_NATIVE(base, 1)
#define IORD_ALTERA_AVALON_PIO_DIRECTION(base)        IORD(base, 1)
#define IOWR_ALTERA_AVALON_PIO_DIRECTION(base, data)  IOWR(base, 1, data)

#define IOADDR_ALTERA_AVALON_PIO_IRQ_MASK(base)       __IO_CALC_ADDRESS_NATIVE(base, 2)
#define IORD_ALTERA_AVALON_PIO_IRQ_MASK(base)         IORD(base, 2)
#define IOWR_ALTERA_AVALON_PIO_IRQ_MASK(base, data)   IOWR(base, 2, data)

#define IOADDR_ALTERA_AVALON_PIO_EDGE_CAP(base)       __IO_CALC_ADDRESS_NATIVE(base, 3)
#define IORD_ALTERA_AVALON_PIO_EDGE_CAP(base)         IORD(base, 3)
#define IOWR_ALTERA_AVALON_PIO_EDGE_CAP(base, data)   IOWR(base, 3, data)


#define IOADDR_ALTERA_AVALON_PIO_SET_BIT(base)       __IO_CALC_ADDRESS_NATIVE(base, 4)
#define IORD_ALTERA_AVALON_PIO_SET_BITS(base)         IORD(base, 4)
#define IOWR_ALTERA_AVALON_PIO_SET_BITS(base, data)   IOWR(base, 4, data)

#define IOADDR_ALTERA_AVALON_PIO_CLEAR_BITS(base)       __IO_CALC_ADDRESS_NATIVE(base, 5)
#define IORD_ALTERA_AVALON_PIO_CLEAR_BITS(base)         IORD(base, 5)
#define IOWR_ALTERA_AVALON_PIO_CLEAR_BITS(base, data)   IOWR(base, 5, data)



/* Defintions for direction-register operation with bi-directional PIOs */
#define ALTERA_AVALON_PIO_DIRECTION_INPUT  0
#define ALTERA_AVALON_PIO_DIRECTION_OUTPUT 1

#endif /* __ALTERA_AVALON_PIO_REGS_H__ */

```

NIOS I/O

- Software setup
  - Create a new c file in the nios_io_sw project



New Source File

Source File

Create a new source file.

Source folder: nios_io_sw

Source file: count.c

Template: Default C source template

NIOS I/O

- Software setup
  - Write a program to read the switches and display a count to the LEDs

```c
/*
/*
 * count.c
 *
 *  Created on: Sep 19, 2018
 *      Author: johnsontimoj
 */
/////////////////////////////
//
// Example file using
// NIOS with LEDs and switches
//
//////////////////////////////

#include "altera_avalon_pio_regs.h"
#include "system.h"
#include <stdio.h>
#include <unistd.h>

int main(){
  printf("My count program!\n");
  alt_u8 count = 0;
  alt_u8 sw;

  while(1){
        // output the count to the LEDs
        IOWR_ALTERA_AVALON_PIO_DATA(LED_PIO_BASE, count);

        // read the switches
        sw = IORD_ALTERA_AVALON_PIO_DATA(SW_PIO_BASE);

        // count up/dn/pause
        if(sw & 0x01){          // pause
              if(!(sw & 0x02))        // up/dn
                    count++;
              else
                    count--;
              printf("%02x,  ", count);
              usleep(500000);
        }// end if

        // delay before restarting
        if( count == 0xff ){
              printf("\nWaiting...");
              int i;
              for (i = 0; i<6; ++i){
                  usleep(500000); /* Sleep for 3s. */
              }
        }// end if
  }// end while
  return 0;
}// end main
```

NIOS I/O

- Software setup
  - Compile the software
    - Select the code file (count.c)
    - Project → Build Project
    - Right Click on the project → run as → Nios II Hardware