ELEC 4200 Lab#0 Tutorial

AUBURN
UNIVERSITY

SAMUEL GINN
COLLEGE OF ENGINEERING

Objectives(1)

- In this Lab exercise, we will design and implement a 2-to-1 multiplexer (MUX), using Xilinx *Vivado* tools to create a VHDL model of the design, verify the model, and implement the model in a Field Programmable Gate Array (FPGA)
- The behavior of the multiplexer will be implemented in two different ways:

  1) Logic equations, which should be familiar from Digital Logic Circuits.

  2) Behavioral description, in which we specify the desired input/output behavior and allow the *Vivado* synthesis tool to implement the required logic.

Objectives(2)

- Aldec *Active-HDL* tools will be used to simulate and verify the design, with model debugging and correction performed as needed.
- After the design is fully verified, we will use *Vivado* to synthesize the design into an Artix-7 FPGA, generating a configuration data file which will then be downloaded onto a Diligent *Nexus 4 DDR* circuit board, where the implemented design will be verified on the hardware using switches to stimulate the circuit and LEDs to observe the outputs.
- If you have no idea what any of this means, don’t worry. That is why this is a tutorial!

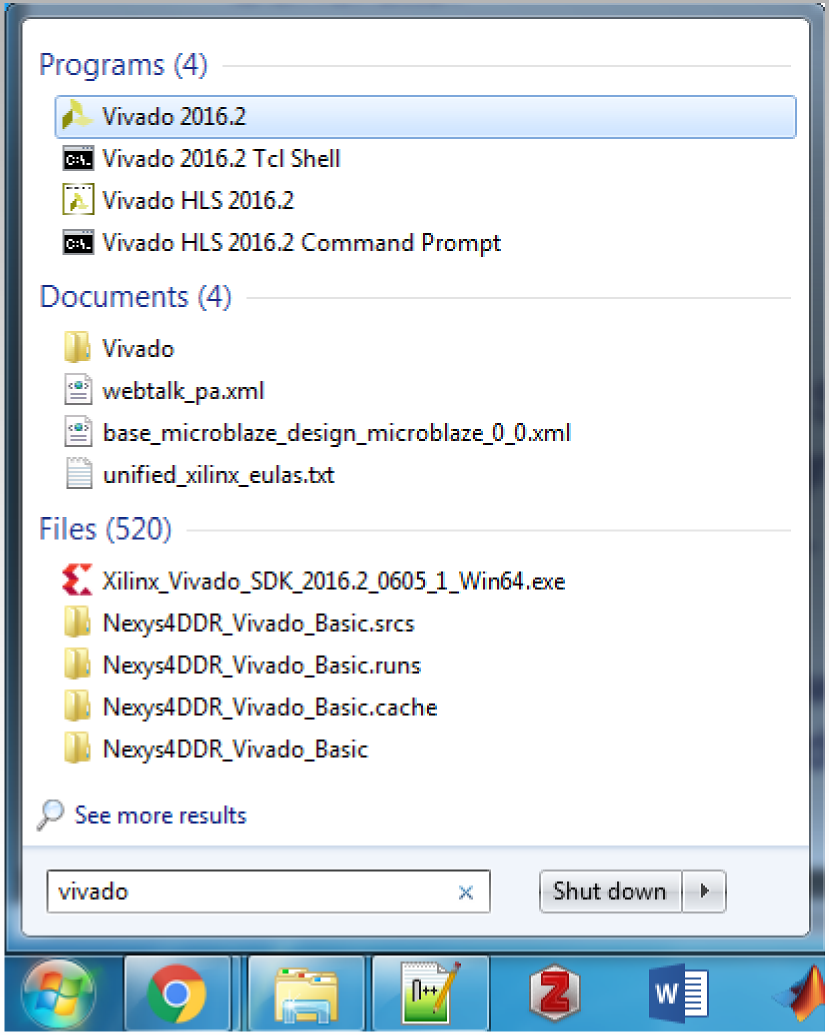
Warning:

- You will need to actually read the instructions.
- If you attempt to just look at the pictures, you will miss steps, make mistakes, encounter errors, etc.
- When you ask for help with problems resulting from not reading the directions, your GTA will make fun of you before telling you to read the directions.
- You have been warned.

AUBURN UNIVERSITY
Samuel Ginn College of Engineering

New Project (1)

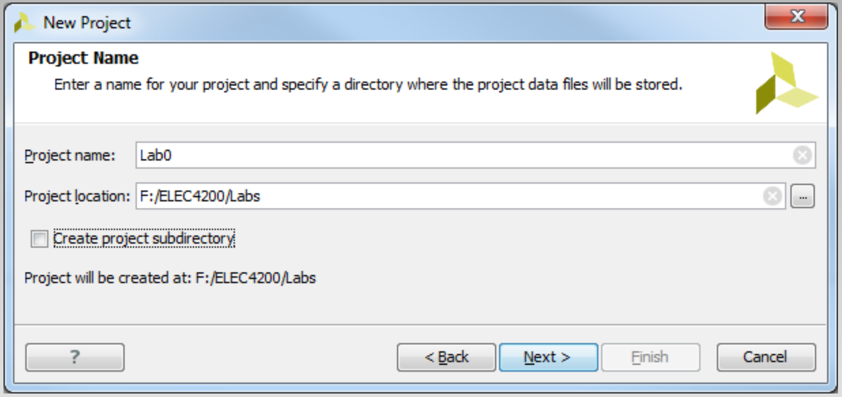
- Click the start menu, and type "Vivado." Then look for "Vivado 2016.2" (or the latest installed version) and click on it.


Programs (4)
Vivado 2016.2
Vivado 2016.2 Tcl Shell
Vivado HLS 2016.2
Vivado HLS 2016.2 Command Prompt
Documents (4)
Vivado
webtalk_pa.xml
base_microblaze_design_microblaze_0_0.xml
unified_xilinx_eulas.txt
Files (520)
Xilinx_Vivado_SDK_2016.2_0605_1_Win64.exe
Nexys4DDR_Vivado_Basic.srcs
Nexys4DDR_Vivado_Basic.runs
Nexys4DDR_Vivado_Basic.cache
Nexys4DDR_Vivado_Basic
See more results
vivado
Shut down

AUBURN UNIVERSITY
SAMUEL GINN COLLEGE OF ENGINEERING

New Project (2)

- Launch the *New Project Wizard* by clicking "Create New Project" on the *Quick Start* page, or "New Project" in the "File" dropdown menu.
- Click "Next" on the first screen to produce the *New Project* window.



New Project

Project Name

Enter a name for your project and specify a directory where the project data files will be stored.

Project name: Lab0

Project location: F:/ELEC4200/Labs

Create project subdirectory

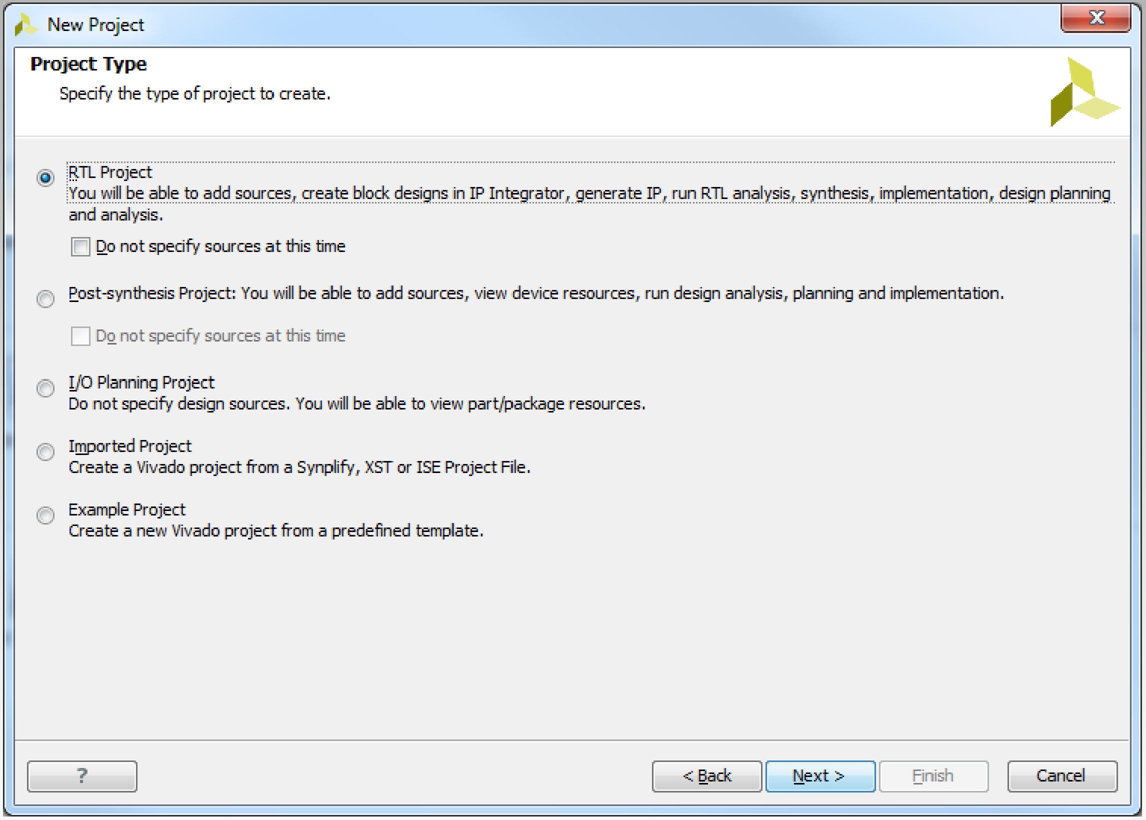
Project will be created at: F:/ELEC4200/Labs

- Enter the project name and location in the corresponding boxes.
- Note: It is HIGHLY recommended that while working in the lab your projects be located on an external flash drive or the C: drive (NOT your H: drive). If your project is located on the H: drive one of the later synthesis steps will ALWAYS fail! If using the C: drive you can use the "C:\TEMP\" directory.
- Give the project a descriptive name, like "LAB0" and click "Next".

AUBURN UNIVERSITY
SAMUEL GINN COLLEGE OF ENGINEERING

New Project (3)

- Select "RTL Project" as the project type, and click "Next".



New Project

Project Type

Specify the type of project to create.

[✓ RTL Project]
You will be able to add sources, create block designs in IP Integrator, generate IP, run RTL analysis, synthesis, implementation, design planning and analysis.

☐ Do not specify sources at this time

Post-synthesis Project: You will be able to add sources, view device resources, run design analysis, planning and implementation.

☐ Do not specify sources at this time

I/O Planning Project
Do not specify design sources. You will be able to view part/package resources.

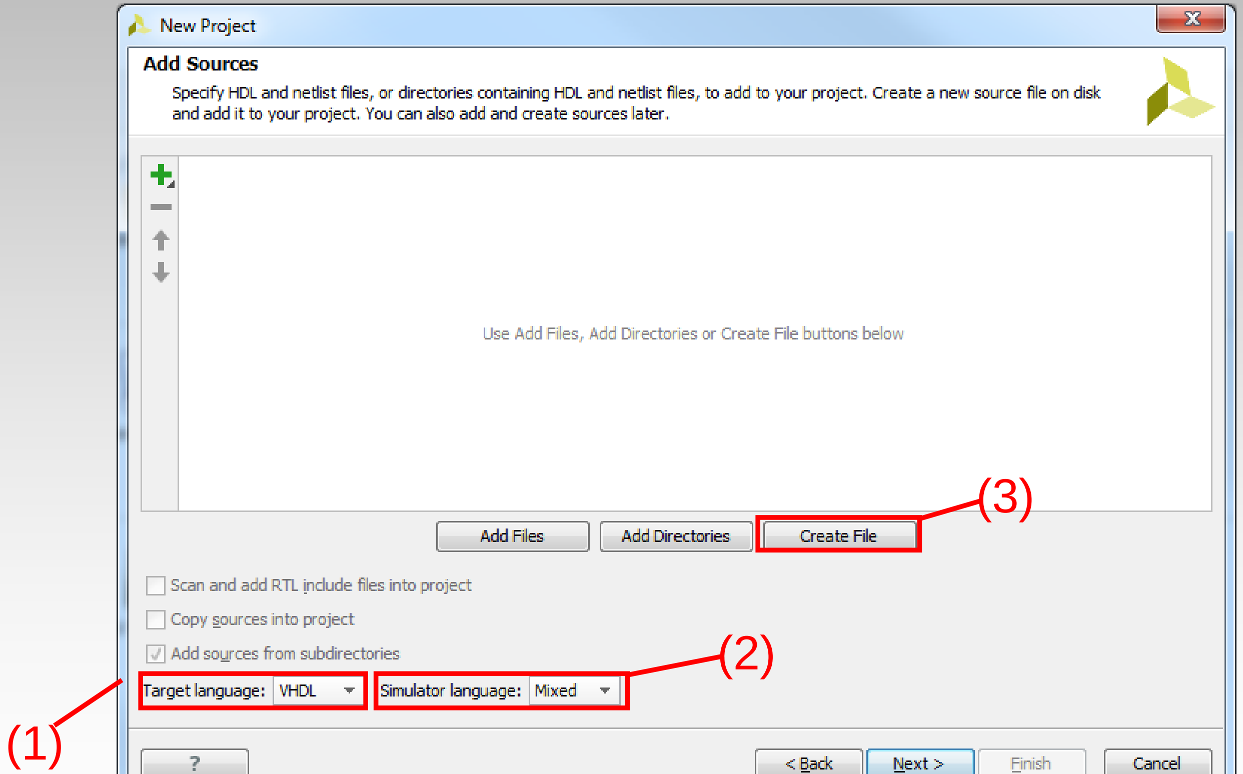
Imported Project
Create a Vivado project from a Synplify, XST or ISE Project File.

Example Project
Create a new Vivado project from a predefined template.

AUBURN UNIVERSITY
SAMUEL GINN COLLEGE OF ENGINEERING

New Project (4)

- In the *Add Sources* window, set the "Target Language" to VHDL and the "Simulator language" to Mixed.
- After setting the language options, click "Create File"



New Project

Add Sources

Specify HDL and netlist files, or directories containing HDL and netlist files, to add to your project. Create a new source file on disk and add it to your project. You can also add and create sources later.

Use Add Files, Add Directories or Create File buttons below

- [ ] Scan and add RTL include files into project
- [ ] Copy sources into project
- [✓] Add sources from subdirectories

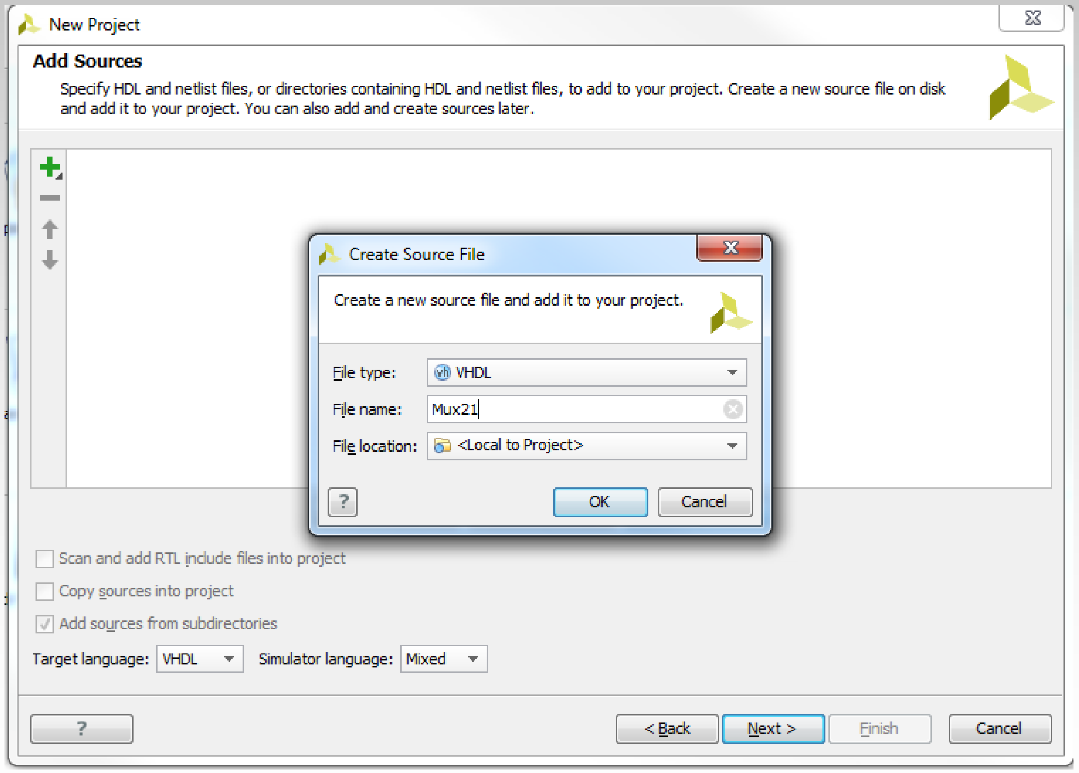
Target language: VHDL

Simulator language: Mixed

AUBURN UNIVERSITY
SAMUEL GINN COLLEGE OF ENGINEERING

New Project (5)

- In the Create Source File dialog box, make sure the "File type" is set to VHDL and enter the name of your file.
  - Again, it is always a good idea to use a descriptive name, in this case *Mux21* because this is a 2-1 multiplexor.
- Click "OK" to close the dialog box and then "Next".



New Project

Add Sources

Specify HDL and netlist files, or directories containing HDL and netlist files, to add to your project. Create a new source file on disk and add it to your project. You can also add and create sources later.

Create Source File

Create a new source file and add it to your project.

File type: VHDL

File name: Mux21

File location: <Local to Project>

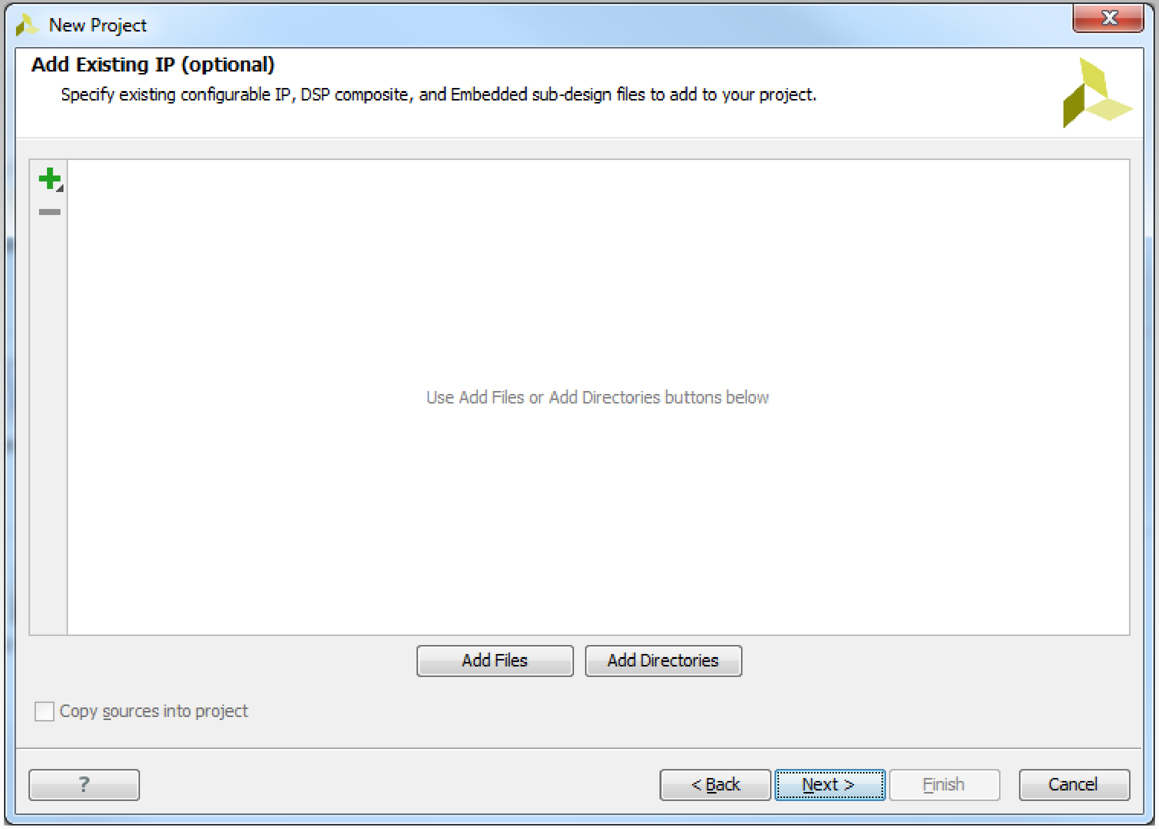
- [ ] Scan and add RTL include files into project
- [ ] Copy sources into project
- [x] Add sources from subdirectories

Target language: VHDL Simulator language: Mixed

AUBURN UNIVERSITY
SAMUEL GINN COLLEGE OF ENGINEERING

New Project (6)

- We are not using any existing IP (intellectual property components), so click “Next” again.



New Project

Add Existing IP (optional)

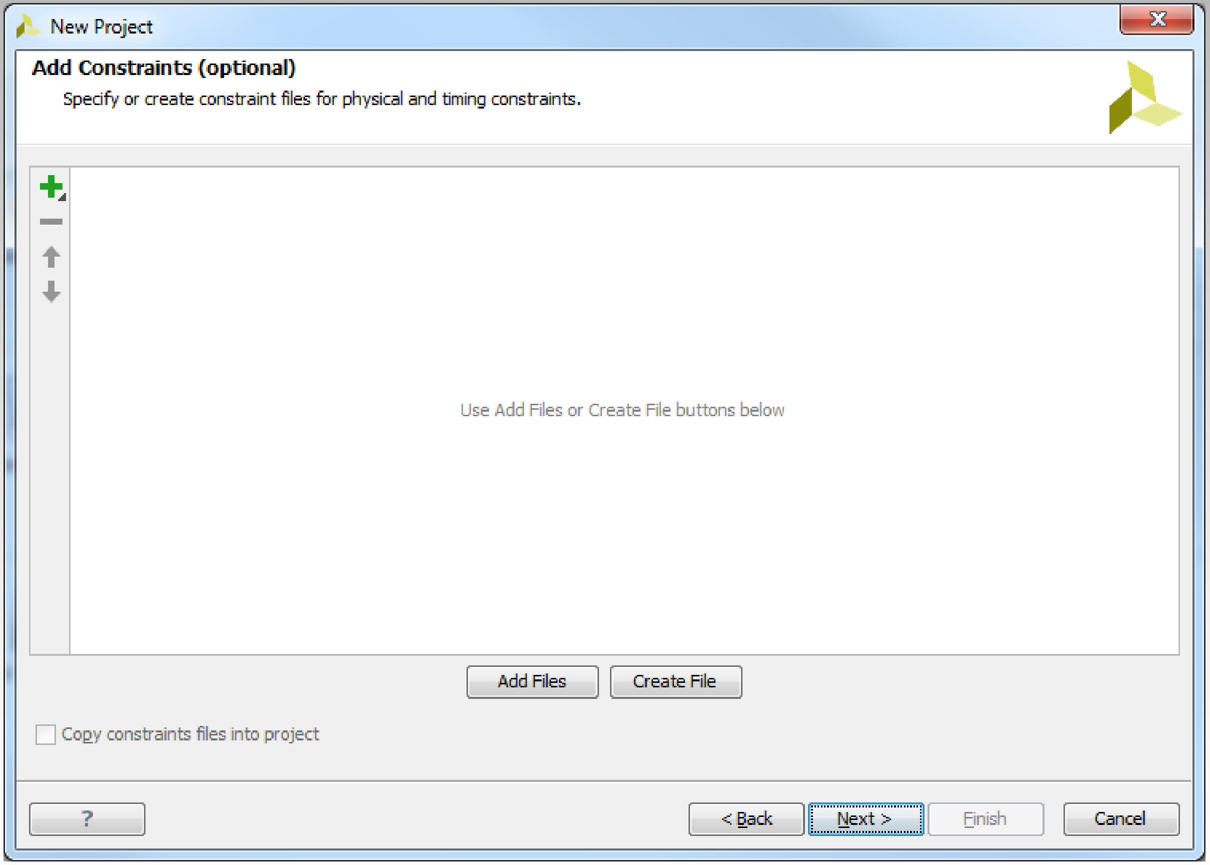
Specify existing configurable IP, DSP composite, and Embedded sub-design files to add to your project.

Use Add Files or Add Directories buttons below

AUBURN UNIVERSITY
SAMUEL GINN COLLEGE OF ENGINEERING

New Project (7)

- We do not have any constraints yet, but in future projects, you may import and edit existing constraints files to save time. So click “Next” for now.


New Project

Add Constraints (optional)

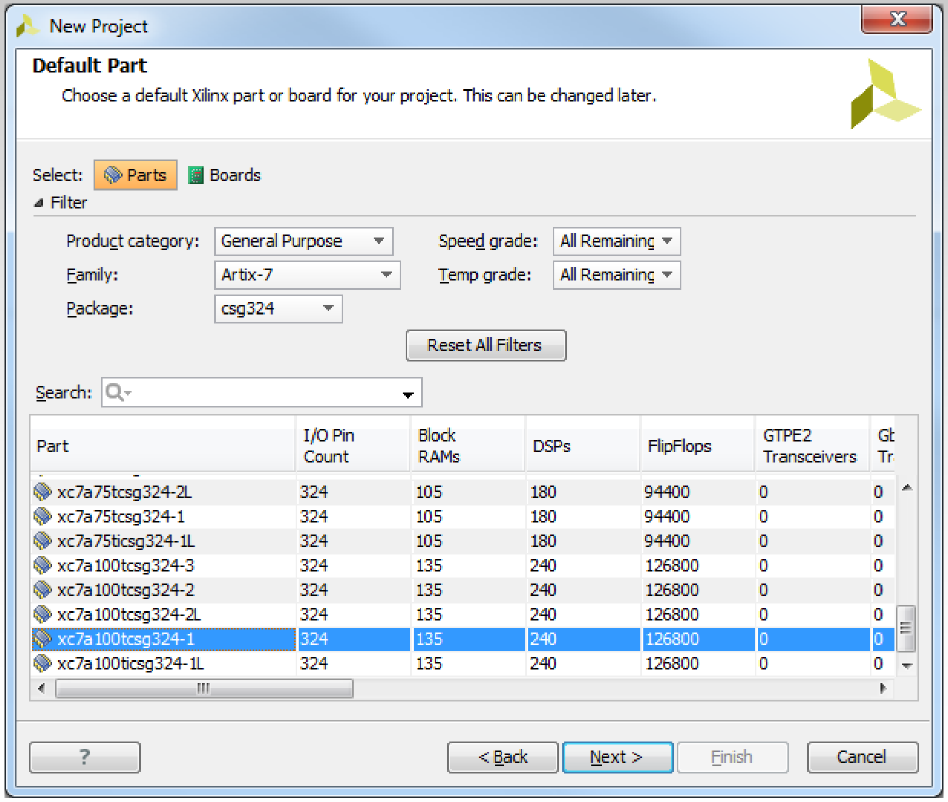
Specify or create constraint files for physical and timing constraints.

Use Add Files or Create File buttons below

AUBURN UNIVERSITY
SAMUEL GINN COLLEGE OF ENGINEERING

New Project (8)

- Here we need to select our device. You can use the search function, filters, or just scroll until you find our device: **XC7A100tcsg324-1**.
  - This FPGA is from the Xilinx Artix-7 family (**XC7A100T**).
  - The device is contained in a 324-pin **csg324** package.
  - The speed grade of the part is "**-1**".
- Click "Next"


New Project

Default Part

Choose a default Xilinx part or board for your project. This can be changed later.

Select: Parts Boards

Filter

Product category: General Purpose
Family: Artix-7
Package: csg324
Speed grade: All Remaining
Temp grade: All Remaining

Reset All Filters

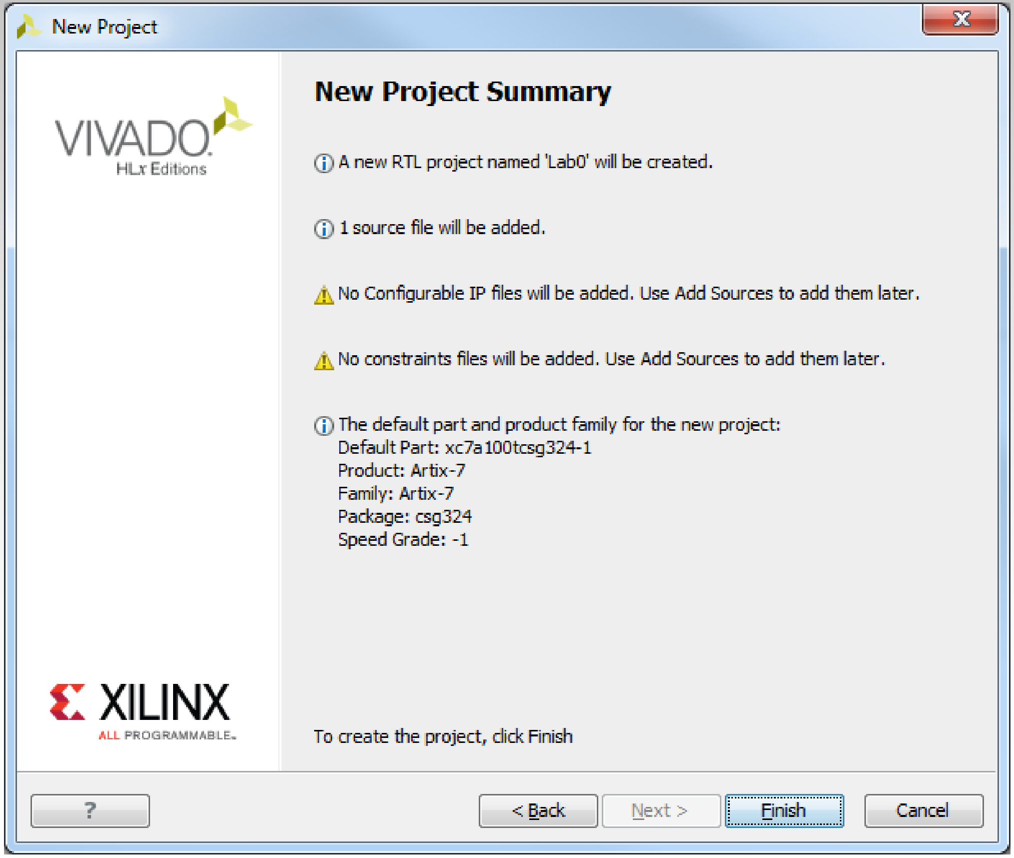
Search:

Part	I/O Pin Count	Block RAMs	DSPs	FlipFlops	GTPE2 Transceivers	Gt Tr
xc7a75tcsg324-2L	324	105	180	94400	0	0
xc7a75tcsg324-1	324	105	180	94400	0	0
xc7a75ticsg324-1L	324	105	180	94400	0	0
xc7a100tcsg324-3	324	135	240	126800	0	0
xc7a100tcsg324-2	324	135	240	126800	0	0
xc7a100tcsg324-2L	324	135	240	126800	0	0
xc7a100tcsg324-1	324	135	240	126800	0	0
xc7a100ticsg324-1L	324	135	240	126800	0	0

AUBURN UNIVERSITY
Samuel Ginn College of Engineering

New Project (9

- The New Project Summary window lists information selected in the previous screens.
    - If necessary, use the “Back” button to return to previous screens to make changes/corrections.
- Click “Finish” to open the Project Manager window.



New Project Summary

A new RTL project named 'Lab0' will be created.

1 source file will be added.

No Configurable IP files will be added. Use Add Sources to add them later.

No constraints files will be added. Use Add Sources to add them later.

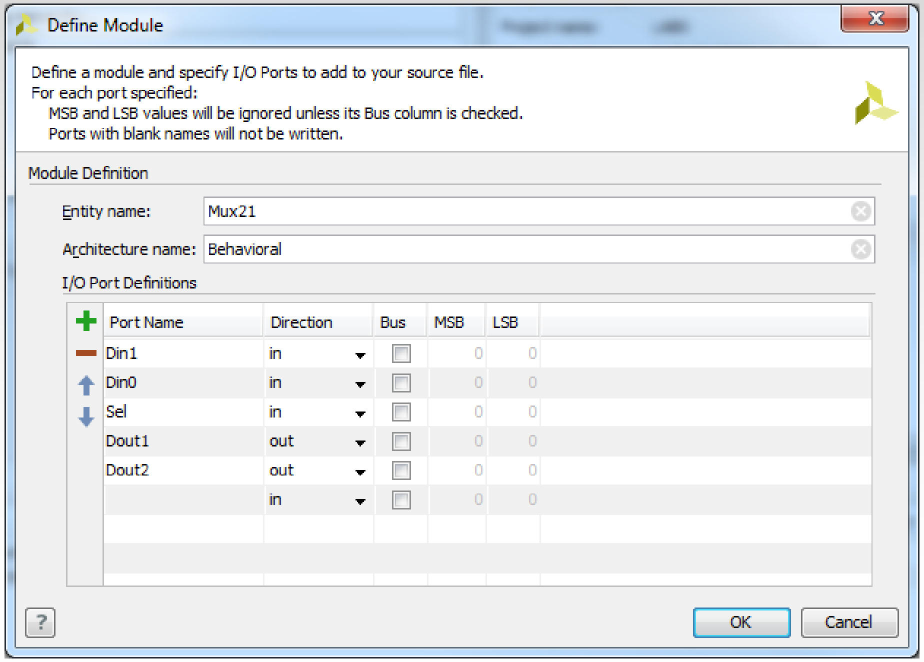
The default part and product family for the new project:
Default Part: xc7a100tcsg324-1
Product: Artix-7
Family: Artix-7
Package: csg324
Speed Grade: -1

To create the project, click Finish

AUBURN UNIVERSITY
SAMUEL GINN COLLEGE OF ENGINEERING

Create the VHDL model

- Since we chose to create a new VHDL file, *Vivado* will automatically launch a wizard to assist in creating the entity and architecture structures that comprise a VHDL model. This can all be done by hand (and you can edit all of this later), but there is no reason not to take advantage of the wizard utility.
- Leave the Entity and Architecture names as their defaults.
- Create three inputs (Direction "in"): *Din0*, *Din1*, *Sel*
- Create two outputs (Direction "out"): *Dout1, Dout2*
- Click "OK"



Define Module

Define a module and specify I/O Ports to add to your source file.
For each port specified:
MSB and LSB values will be ignored unless its Bus column is checked.
Ports with blank names will not be written.

Module Definition

Entity name: Mux21

Architecture name: Behavioral

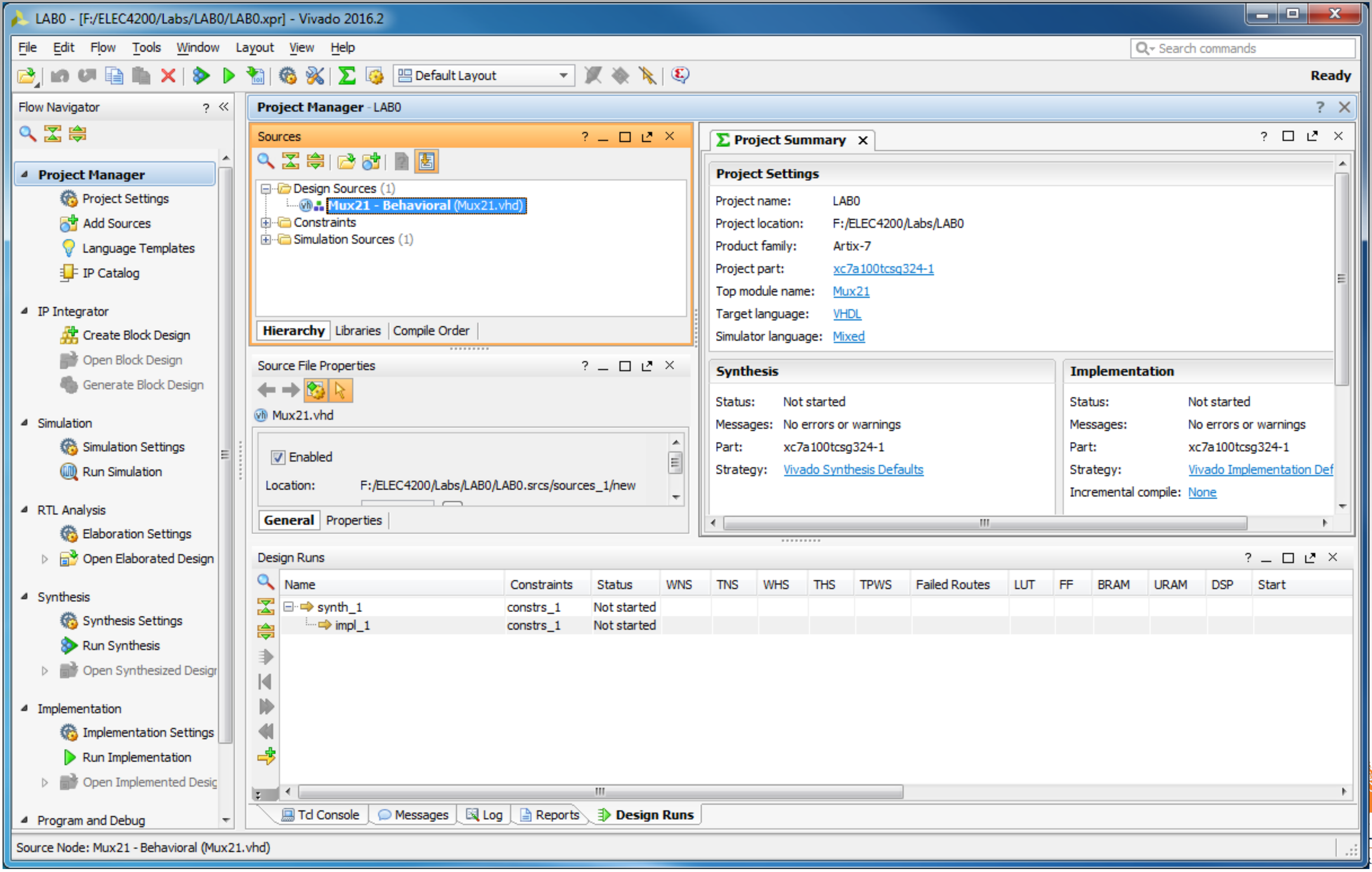
I/O Port Definitions

Port Name	Direction	Bus	MSB	LSB
Din1	in	☐	0	0
Din0	in	☐	0	0
Sel	in	☐	0	0
Dout1	out	☐	0	0
Dout2	out	☐	0	0
	in	☐	0	0

AUBURN UNIVERSITY
SAMUEL GINN COLLEGE OF ENGINEERING

Edit VHDL Model(1)

- Open your new VHDL file from the Sources window
  (double click on the Design Source name, *Mux21*)



LAB0 - [F:/ELEC4200/Labs/LAB0/LAB0.xpr] - Vivado 2016.2

File Edit Flow Tools Window Layout View Help

Default Layout — Ready

Flow Navigator: Project Manager (Project Settings, Add Sources, Language Templates, IP Catalog); IP Integrator (Create Block Design, Open Block Design, Generate Block Design); Simulation (Simulation Settings, Run Simulation); RTL Analysis (Elaboration Settings, Open Elaborated Design); Synthesis (Synthesis Settings, Run Synthesis, Open Synthesized Design); Implementation (Implementation Settings, Run Implementation, Open Implemented Design); Program and Debug

Project Manager - LAB0

Sources:
- Design Sources (1)
  - Mux21 - Behavioral (Mux21.vhd)
- Constraints
- Simulation Sources (1)

Hierarchy | Libraries | Compile Order

Source File Properties: Mux21.vhd

Enabled

Location: F:/ELEC4200/Labs/LAB0/LAB0.srcs/sources_1/new

General | Properties

Project Summary — Project Settings

Project name:	LAB0
Project location:	F:/ELEC4200/Labs/LAB0
Product family:	Artix-7
Project part:	xc7a100tcsg324-1
Top module name:	Mux21
Target language:	VHDL
Simulator language:	Mixed

Synthesis

Status:	Not started
Messages:	No errors or warnings
Part:	xc7a100tcsg324-1
Strategy:	Vivado Synthesis Defaults

Implementation

Status:	Not started
Messages:	No errors or warnings
Part:	xc7a100tcsg324-1
Strategy:	Vivado Implementation Def
Incremental compile:	None

Design Runs

Name	Constraints	Status	WNS	TNS	WHS	THS	TPWS	Failed Routes	LUT	FF	BRAM	URAM	DSP	Start
synth_1	constrs_1	Not started												
impl_1	constrs_1	Not started												

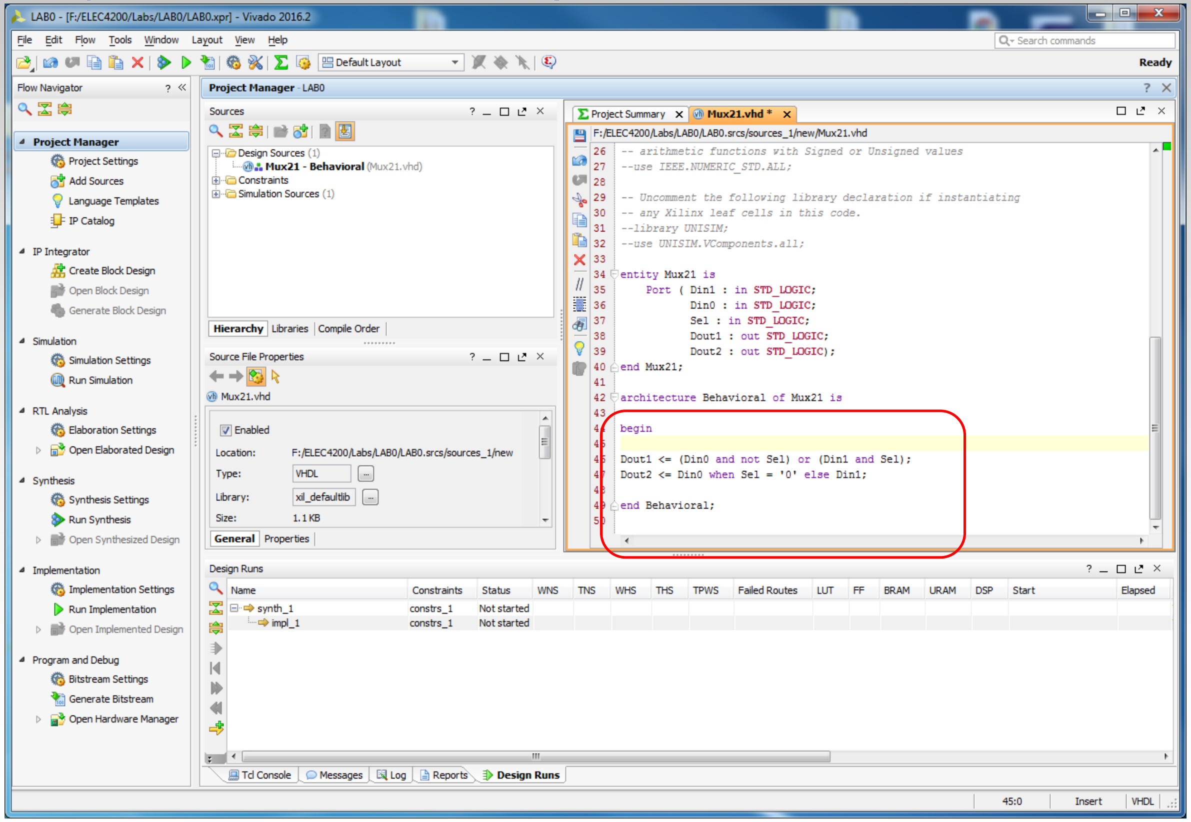
Tcl Console | Messages | Log | Reports | Design Runs

Source Node: Mux21 - Behavioral (Mux21.vhd)

Samuel Ginn College of Engineering

Edit VHDL Model(2)

- Scroll down and add the following two lines of code between the Begin and End statements of the architecture section of the model.
  - *Dout1<=(Din0 and not Sel) or (Din1 and Sel);*
  - *Dout2<=Din0 when Sel = ‘0’ else Din1;*
- After saving, we can then set up the simulator.



F:/ELEC4200/Labs/LAB0/LAB0.srcs/sources_1/new/Mux21.vhd

```vhdl
-- arithmetic functions with Signed or Unsigned values
--use IEEE.NUMERIC_STD.ALL;

-- Uncomment the following library declaration if instantiating
-- any Xilinx leaf cells in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity Mux21 is
    Port ( Din1 : in STD_LOGIC;
           Din0 : in STD_LOGIC;
           Sel : in STD_LOGIC;
           Dout1 : out STD_LOGIC;
           Dout2 : out STD_LOGIC);
end Mux21;

architecture Behavioral of Mux21 is

begin

Dout1 <= (Din0 and not Sel) or (Din1 and Sel);
Dout2 <= Din0 when Sel = '0' else Din1;

end Behavioral;
```

Name	Constraints	Status
synth_1	constrs_1	Not started
impl_1	constrs_1	Not started

Mux models:

1st statement: logic equation

2nd statement: “behavior”

AUBURN UNIVERSITY
SAMUEL GINN COLLEGE OF ENGINEERING

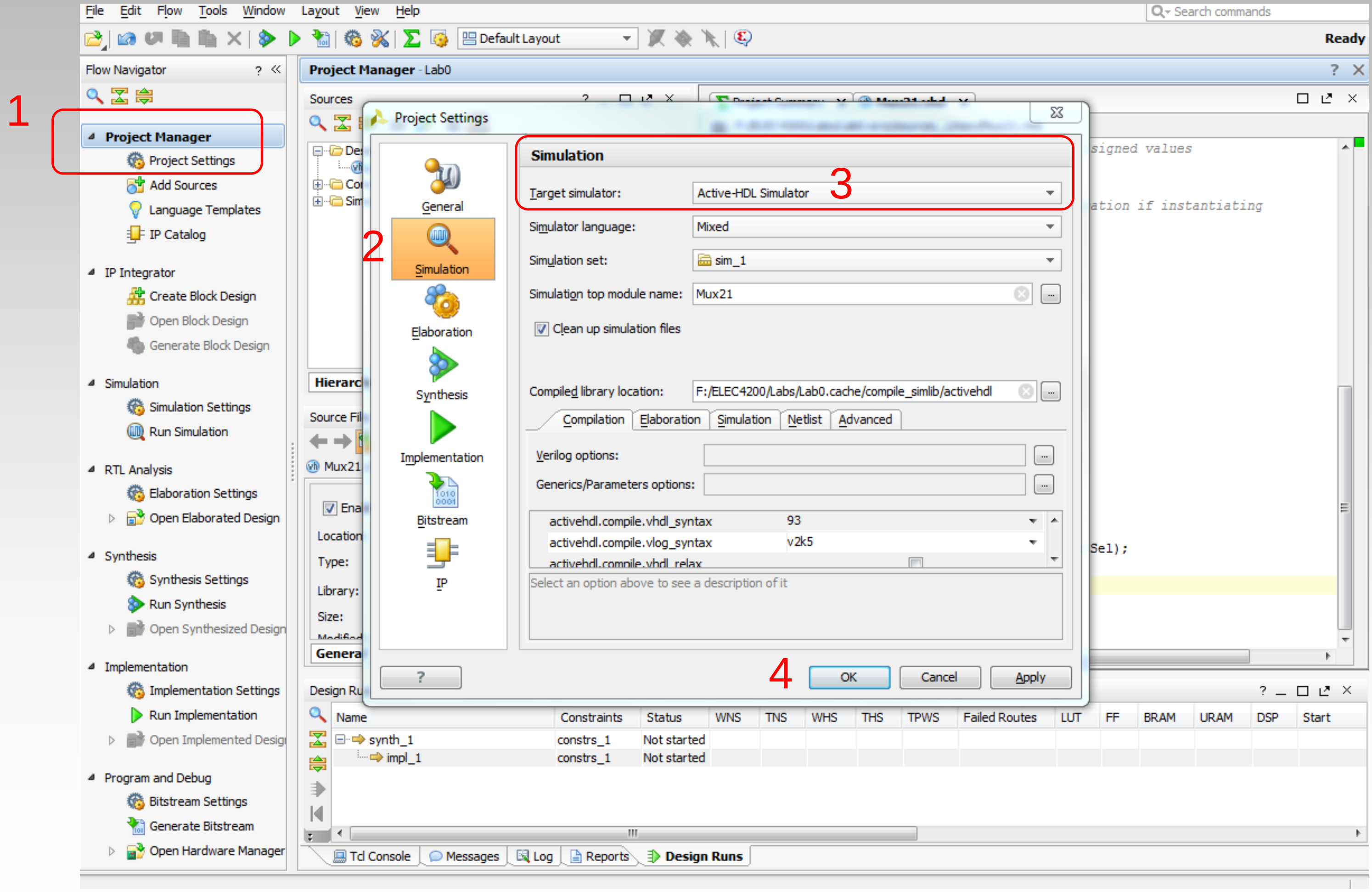
Aldec Active-HDL Setup(1)

- Aldec *Active-HDL* is a commercially available modeling and simulation tool used to verify designs before synthesizing the designs into actual hardware. Xilinx *Vivado* integrates support for *Active-HDL*, as well as several other commercial simulation tools and its own simulator. Therefore, you must select *Active-HDL* from a list of simulators.

- As long as you use the same computer week to week, this setting should only have to be made this one time. However, if you ever get errors from *Vivado* about not being able to find the simulator, check this setting first.

Active-HDL Setup(2)

1. In the Project Manager pane, select *Project Settings*
2. In the Project Settings window, select *Simulation*
3. Select *Active-HDL Simulator* in the Target simulator drop-down menu.
4. Click OK to close the window.



Simulation

Target simulator: Active-HDL Simulator

Simulator language: Mixed

Simulation set: sim_1

Simulation top module name: Mux21

☑ Clean up simulation files

Compiled library location: F:/ELEC4200/Labs/Lab0.cache/compile_simlib/activehdl

Compilation | Elaboration | Simulation | Netlist | Advanced

Verilog options:

Generics/Parameters options:

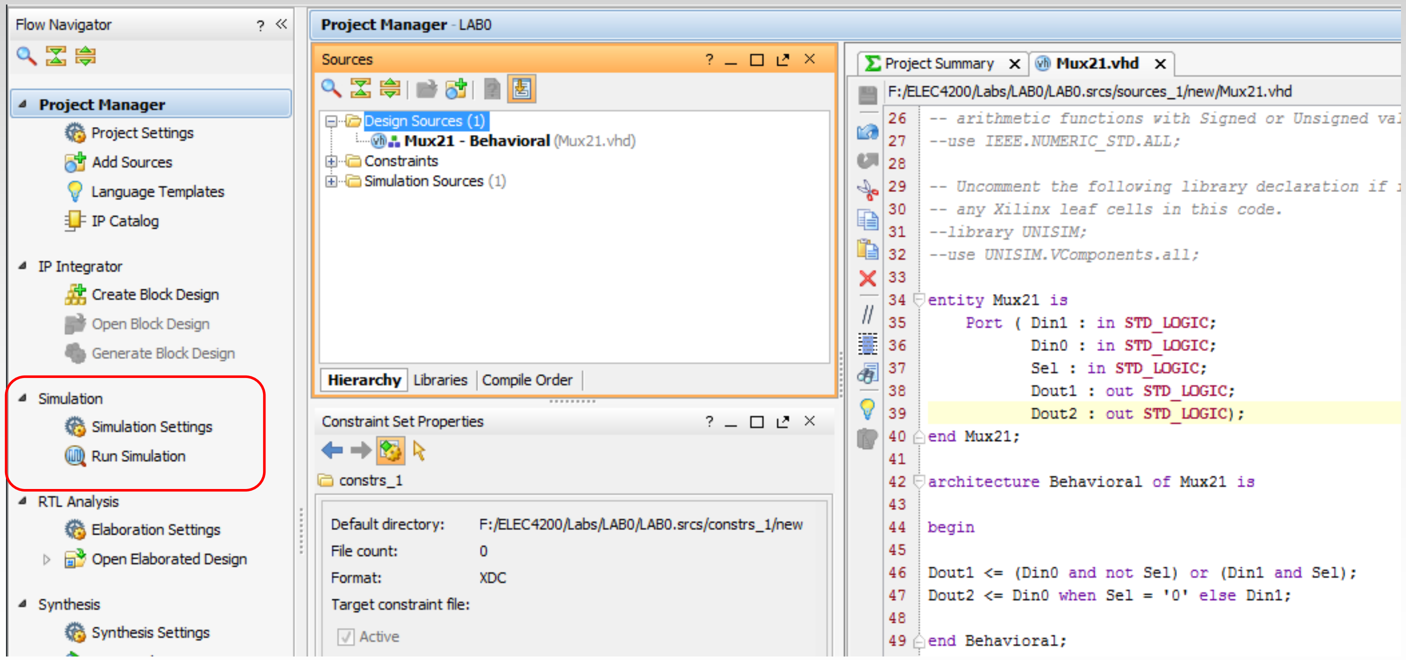
activehdl.compile.vhdl_syntax	93
activehdl.compile.vlog_syntax	v2k5
activehdl.compile.vhdl_relax	

Select an option above to see a description of it

Name	Constraints	Status	WNS	TNS	WHS	THS	TPWS	Failed Routes	LUT	FF	BRAM	URAM	DSP	Start
synth_1	constrs_1	Not started												
impl_1	constrs_1	Not started												

Active-HDL Simulation(1)

- Aldec HDL is packed full of features that help with development, simulation and debugging of FPGA designs. This tutorial describes only how to perform a basic simulation, but its is recommended that you experiment with its many capabilities, as you may find ways it can help you debug in later lab sessions.
- Click on “Run Simulation" in the Flow Navigator, and then Run “Behavioral Simulation” in the popup menu.



Flow Navigator

Project Manager
- Project Settings
- Add Sources
- Language Templates
- IP Catalog

IP Integrator
- Create Block Design
- Open Block Design
- Generate Block Design

Simulation
- Simulation Settings
- Run Simulation

RTL Analysis
- Elaboration Settings
- Open Elaborated Design

Synthesis
- Synthesis Settings

Project Manager - LAB0

Sources
- Design Sources (1)
  - **Mux21 - Behavioral** (Mux21.vhd)
- Constraints
- Simulation Sources (1)

Hierarchy | Libraries | Compile Order

Constraint Set Properties

constrs_1

Default directory:	F:/ELEC4200/Labs/LAB0/LAB0.srcs/constrs_1/new
File count:	0
Format:	XDC
Target constraint file:	

Active

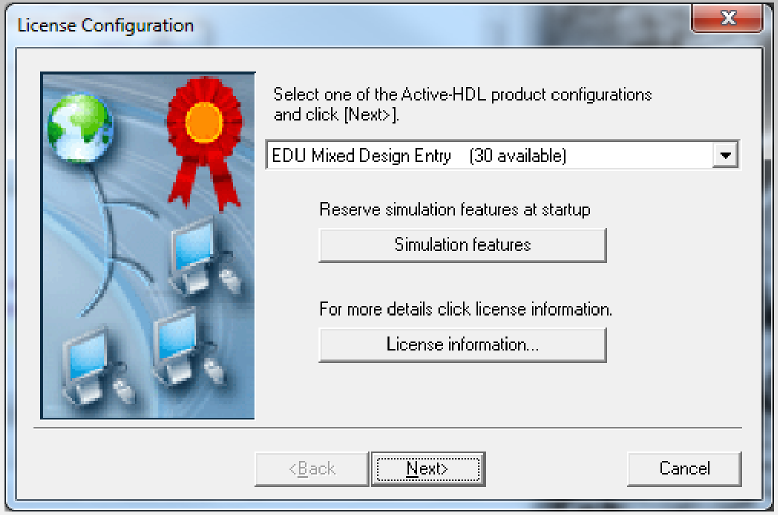
Project Summary | **Mux21.vhd**

F:/ELEC4200/Labs/LAB0/LAB0.srcs/sources_1/new/Mux21.vhd

```
-- arithmetic functions with Signed or Unsigned val
--use IEEE.NUMERIC_STD.ALL;

-- Uncomment the following library declaration if i
-- any Xilinx leaf cells in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity Mux21 is
    Port ( Din1 : in STD_LOGIC;
           Din0 : in STD_LOGIC;
           Sel : in STD_LOGIC;
           Dout1 : out STD_LOGIC;
           Dout2 : out STD_LOGIC);
end Mux21;

architecture Behavioral of Mux21 is

begin

Dout1 <= (Din0 and not Sel) or (Din1 and Sel);
Dout2 <= Din0 when Sel = '0' else Din1;

end Behavioral;
```

Active-HDL Simulation(2)

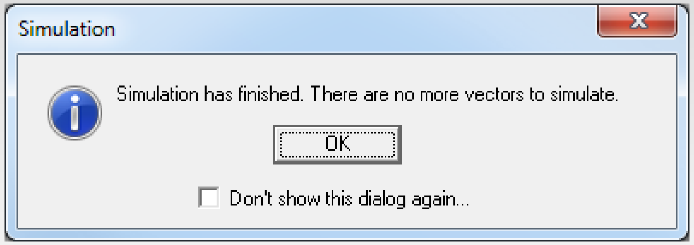
- Active-HDL will run twice
  - It will open to compile the VHDL model(s) and then close.
  - If there are no compilation errors, it will open again to simulate the model and remain open.
  - Each time you must click “Next” in the License Configuration window.


License Configuration

Select one of the Active-HDL product configurations and click [Next>].

EDU Mixed Design Entry (30 available)

Reserve simulation features at startup


Simulation

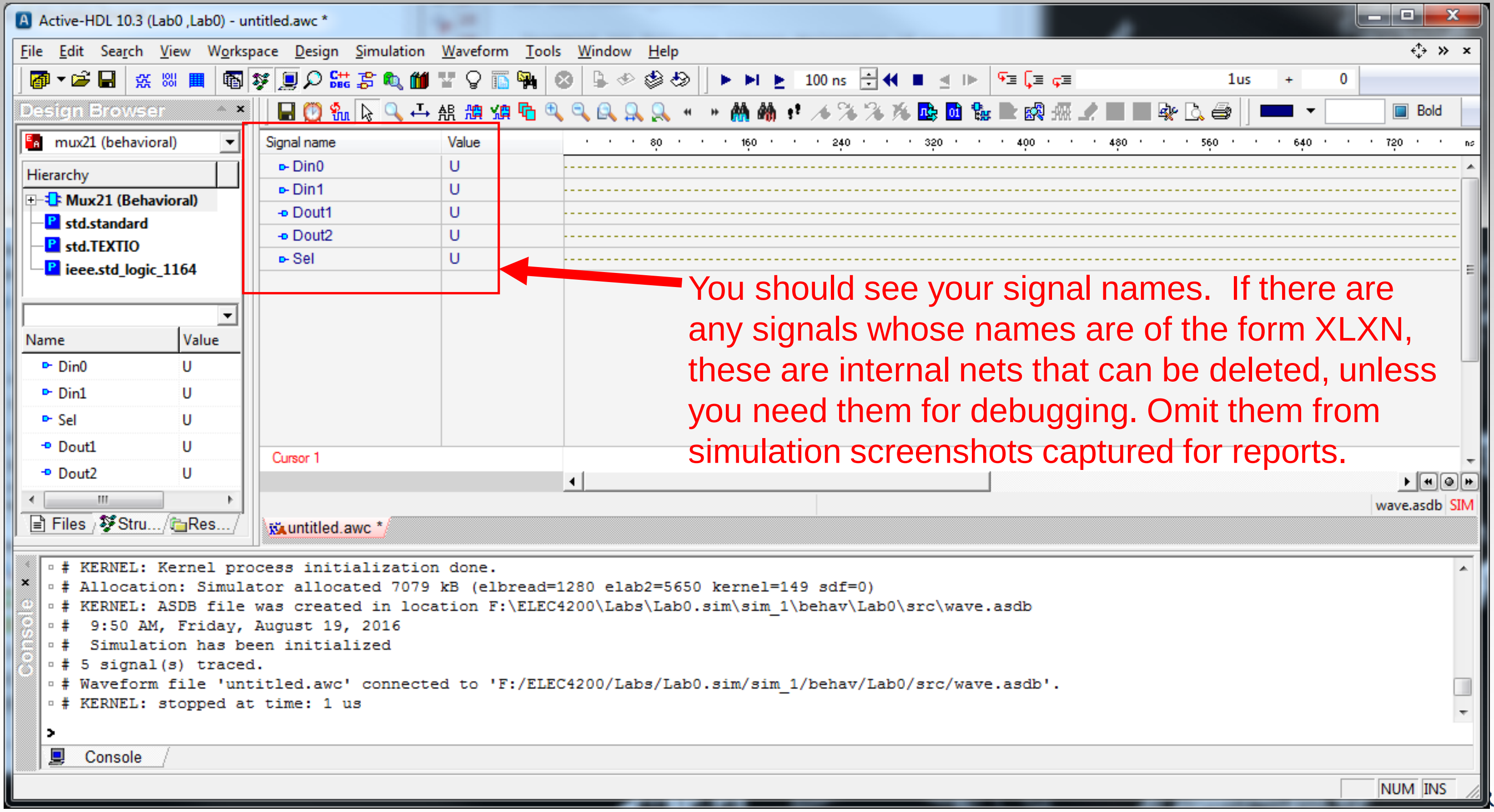
Simulation has finished. There are no more vectors to simulate.

- Click OK on the “Simulation had finished” information window.
  - Note that we have not yet provided any signal inputs, so this “simulation” did not provide any useful information.
  - We will set up the desired simulation in the next steps.

AUBURN UNIVERSITY
SAMUEL GINN COLLEGE OF ENGINEERING

Active-HDL Simulation(3)

- When the simulator is finished launching you should see the following window. If you do not, call the GTA to show you to set up the proper view windows.



Signal name	Value
Din0	U
Din1	U
Dout1	U
Dout2	U
Sel	U

Name	Value
Din0	U
Din1	U
Sel	U
Dout1	U
Dout2	U

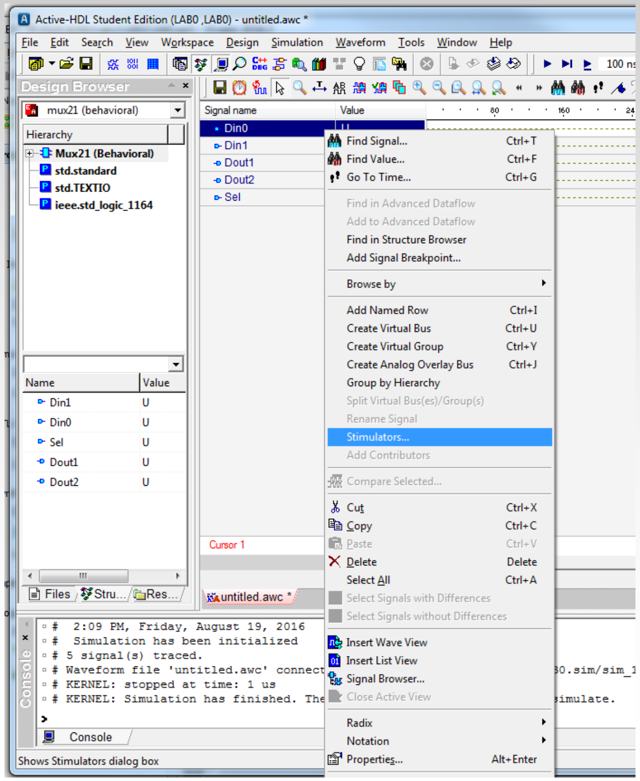
```
# KERNEL: Kernel process initialization done.
# Allocation: Simulator allocated 7079 kB (elbread=1280 elab2=5650 kernel=149 sdf=0)
# KERNEL: ASDB file was created in location F:\ELEC4200\Labs\Lab0.sim\sim_1\behav\Lab0\src\wave.asdb
#  9:50 AM, Friday, August 19, 2016
#  Simulation has been initialized
# 5 signal(s) traced.
# Waveform file 'untitled.awc' connected to 'F:/ELEC4200/Labs/Lab0.sim/sim_1/behav/Lab0/src/wave.asdb'.
# KERNEL: stopped at time: 1 us
```

You should see your signal names. If there are any signals whose names are of the form XLXN, these are internal nets that can be deleted, unless you need them for debugging. Omit them from simulation screenshots captured for reports.

UNIVERSITY
Samuel Ginn College of Engineering

Active-HDL Simulation(4)

- Design verification requires that you stimulate the inputs and observe the outputs. You are to stimulate the inputs with all possible input combinations and observe each output to verify its correctness.
- This is called “exhaustive testing”. It is suitable for a simple circuit (such as the mux) but is not practical for large circuits with many inputs.
- Right click on one of your signals (ex: Din0) and select “Stimulators…” from the context menu.
- You can also shift/control click to select them all at once.



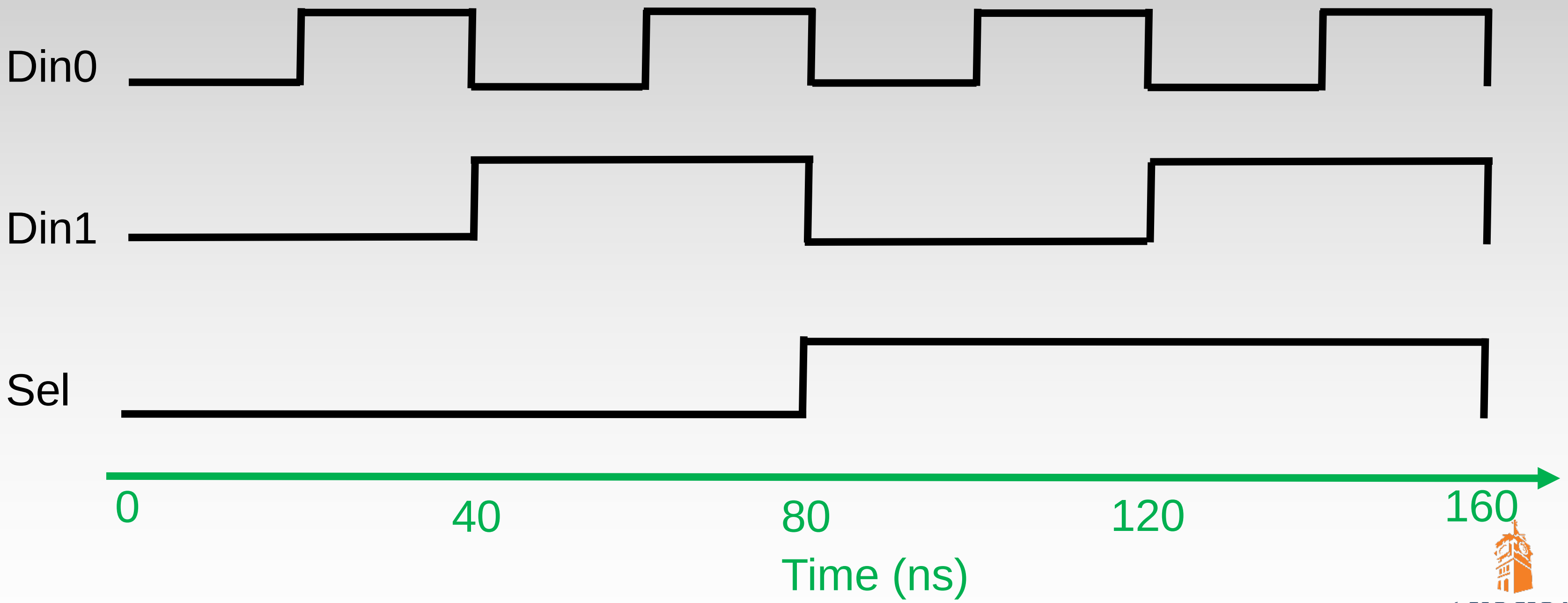
```
# 2:09 PM, Friday, August 19, 2016
# Simulation has been initialized
# 5 signal(s) traced.
# Waveform file 'untitled.awc' connect                    30.sim/sim_1
# KERNEL: stopped at time: 1 us
# KERNEL: Simulation has finished. The                    simulate.
>
```

Shows Stimulators dialog box

AUBURN UNIVERSITY
SAMUEL GINN COLLEGE OF ENGINEERING

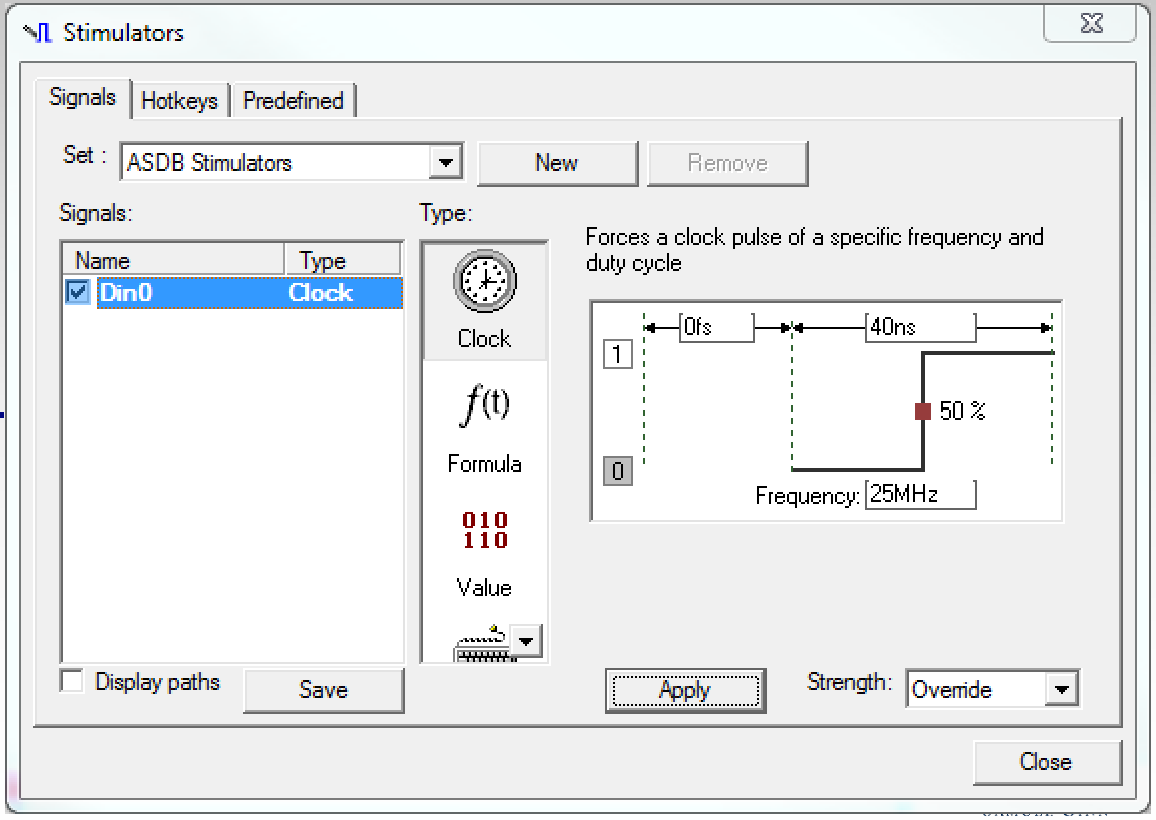
Active-HDL Simulation(5)

- For the 2-1 MUX we have 3 inputs (Din0, Din1, and Sel) where each input can be either logic high (1) or logic low (0).
- There are 2^3=8 possible input combinations. While we could stimulate each combination individually, an easier approach would be to use *Active-HDL's* clock stimulator to generate the input timing diagrams below.



Active-HDL Simulation(6)

- Select your signal and click the "Clock" type. The right side of the Stimulators dialog box should now show the clock parameters.
- You can set the starting value (0 or 1), adjust the offset value to start the clock, adjust the period, and adjust the duty cycle
- Set the options as shown on the right to create a waveform that starts low and repeats every 40ns with a 50% duty cycle.
- Click "Apply"



Set : ASDB Stimulators

Signals:

Name	Type
☑ Din0	Clock

Type:

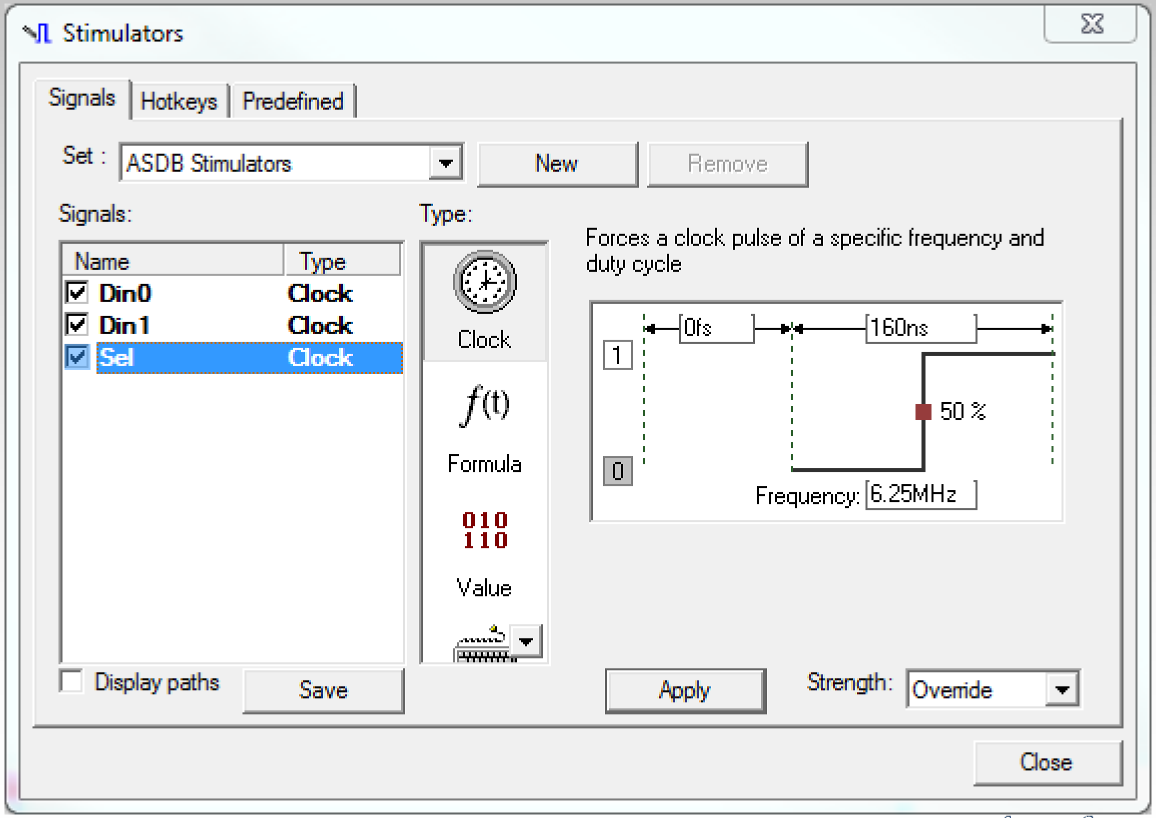
Forces a clock pulse of a specific frequency and duty cycle

1
0
0fs
40ns
50 %
Frequency: 25MHz

Strength: Override

Active-HDL Simulation(6)

- Move the Stimulators dialog box out of the way so you can see the signals in the Waveform Window. Click on a signal in the Waveform Window to add it to the Stimulators dialog box.
- Repeat the clock procedure for Din1 and Sel with the following periods. Click “Apply” after each signal is added.
  - Din1: 80ns period
  - Sel: 160ns period
- Click “Close” when you are finished.



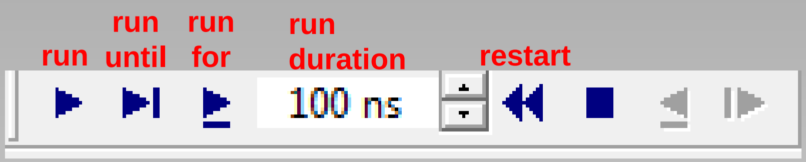
Forces a clock pulse of a specific frequency and duty cycle

Frequency: 6.25MHz

Name	Type
☑ Din0	Clock
☑ Din1	Clock
☑ Sel	Clock

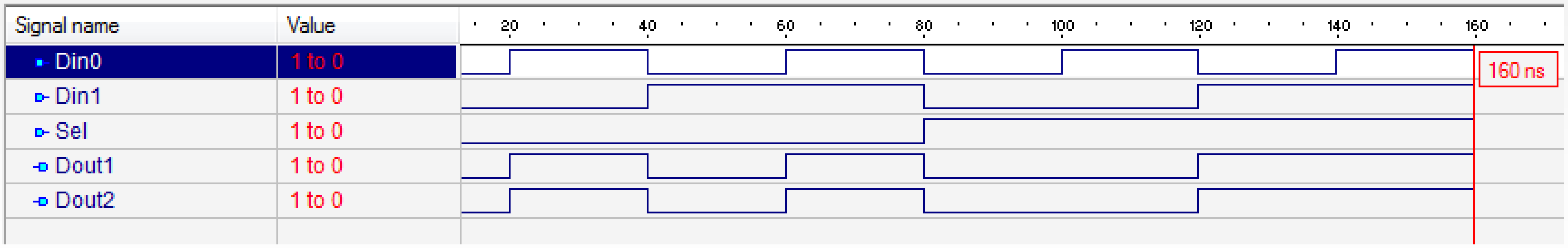
Active-HDL Simulation(7)

- Now we need to run the simulation using the following simulation controls.



- Whenever you begin a new simulation you should clear the waveforms window. Do this by clicking the “Restart Simulation” button or by typing “restart” into the Console Window.

- Run the simulation for 160ns by typing 160 ns into the “Run Duration” box and then click “Run For”. Your output should look similar to this. Use the zoom controls as needed to make the waveform fill the window.



Signal name	Value
Din0	1 to 0
Din1	1 to 0
Sel	1 to 0
Dout1	1 to 0
Dout2	1 to 0

UNIVERSITY
SAMUEL GINN
COLLEGE OF ENGINEERING

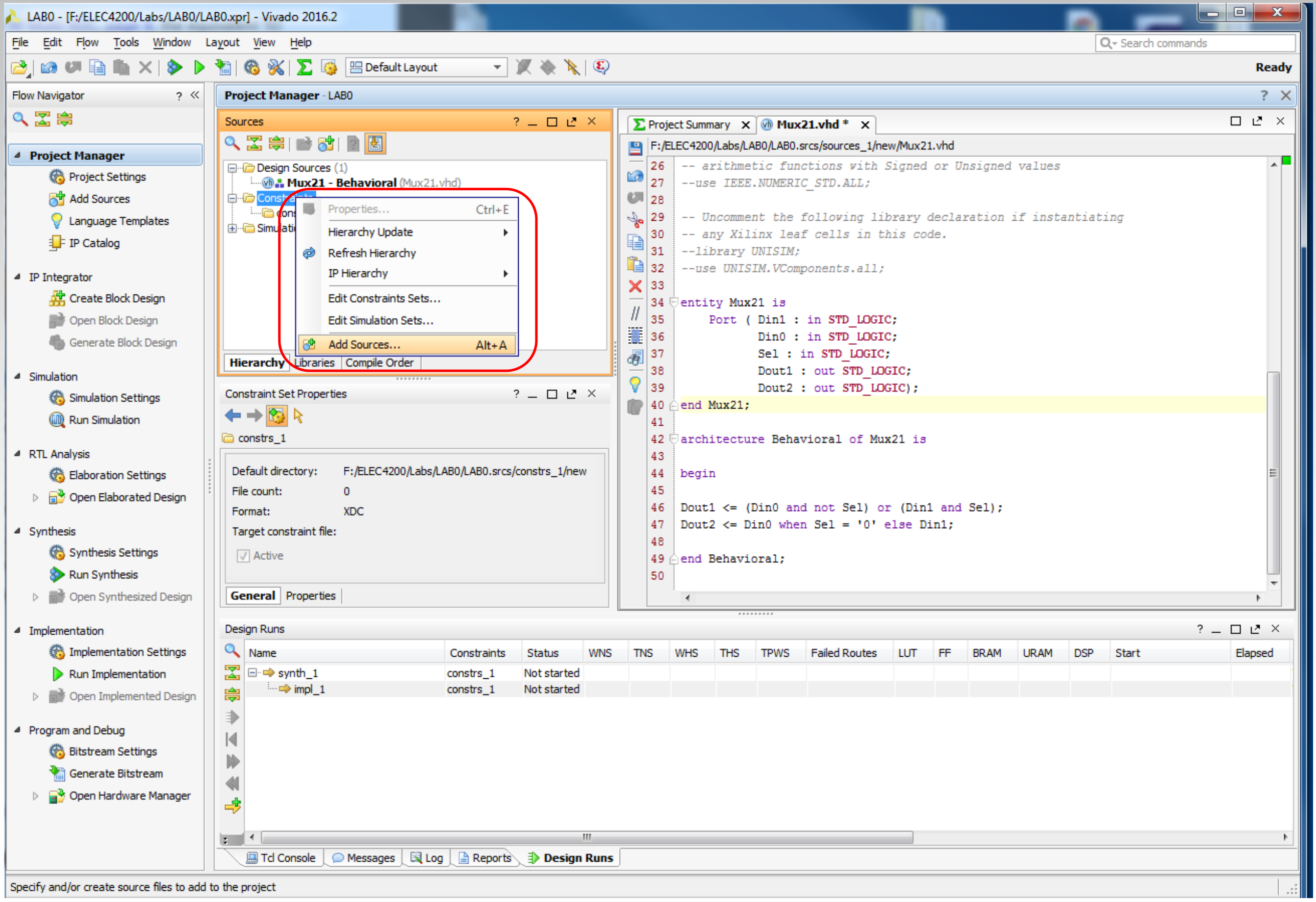
Active-HDL Simulation(8)

- Examine your results and ensure that they accurately match the operation of a 2-1 MUX for both Dout1 and Dout2.
- If you observe any incorrect results during simulation, go back and debug your circuit. Do not proceed any further until your simulation produces the correct results.
- If time allows you may want to experiment with some of the other “types” in the Stimulators window so that you can become familiar with their operation and know how to use them for future labs.
- If your final circuit does not work and it is traced it back to an error in simulation, the GTA reserves the right to laugh at you publicly in front of your classmates.

AUBURN UNIVERSITY
SAMUEL GINN COLLEGE OF ENGINEERING

Add Constraints (1)

- We need to define a “constraints file” that defines which signals (Din, Dout, Sel) go to which pins on the FPGA.
- Right click on the “Constraints” folder in the project manager and select “Add Sources”.



Default directory: F:/ELEC4200/Labs/LAB0/LAB0.srcs/constrs_1/new
File count: 0
Format: XDC
Target constraint file:
Active

```
-- arithmetic functions with Signed or Unsigned values
--use IEEE.NUMERIC_STD.ALL;

-- Uncomment the following library declaration if instantiating
-- any Xilinx leaf cells in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity Mux21 is
    Port ( Din1 : in STD_LOGIC;
           Din0 : in STD_LOGIC;
           Sel : in STD_LOGIC;
           Dout1 : out STD_LOGIC;
           Dout2 : out STD_LOGIC);
end Mux21;

architecture Behavioral of Mux21 is

begin

Dout1 <= (Din0 and not Sel) or (Din1 and Sel);
Dout2 <= Din0 when Sel = '0' else Din1;

end Behavioral;
```

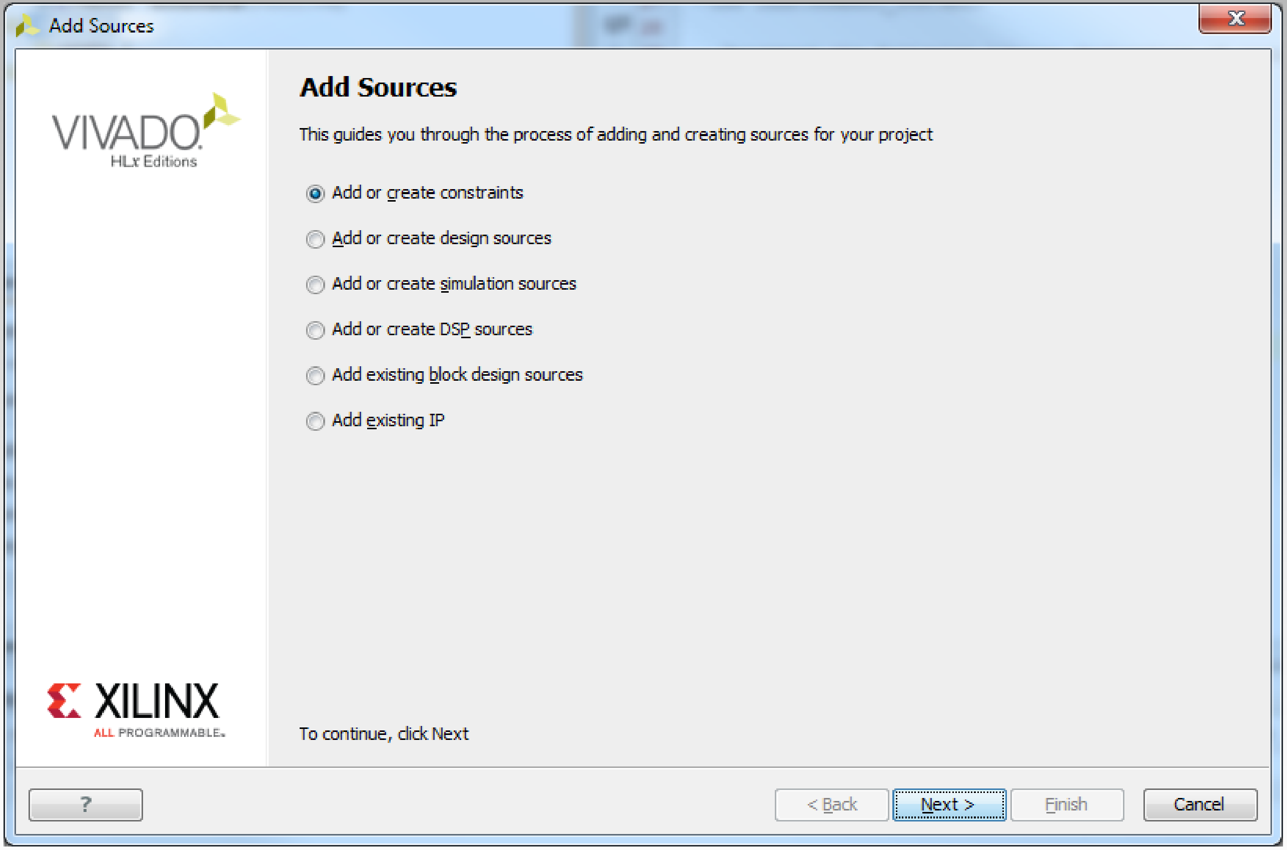
Name	Constraints	Status	WNS	TNS	WHS	THS	TPWS	Failed Routes	LUT	FF	BRAM	URAM	DSP	Start	Elapsed
synth_1	constrs_1	Not started													
impl_1	constrs_1	Not started													

Specify and/or create source files to add to the project

AUBURN UNIVERSITY
SAMUEL GINN COLLEGE OF ENGINEERING

Add Constraints (2)

- This opens the “Add Sources Wizard”. Make sure “Add or create constraints” is selected and click “Next”.



Add Sources

Add Sources

This guides you through the process of adding and creating sources for your project

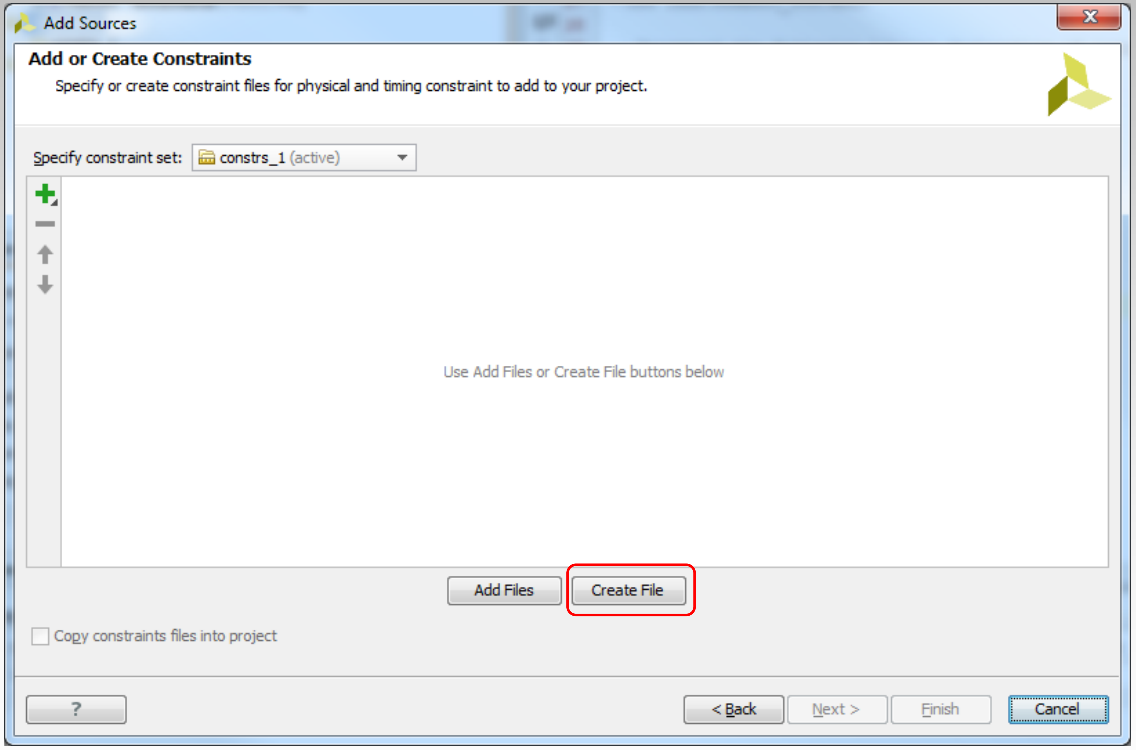
- [✓ Add or create constraints]
- Add or create design sources
- Add or create simulation sources
- Add or create DSP sources
- Add existing block design sources
- Add existing IP

To continue, click Next

AUBURN UNIVERSITY
SAMUEL GINN COLLEGE OF ENGINEERING

Add Constraints (3)

- Click “Create File” and in the dialog box that opens, name your constraints file. As always, use a descriptive name, like “LAB0_Constraints”. Then click “OK”.



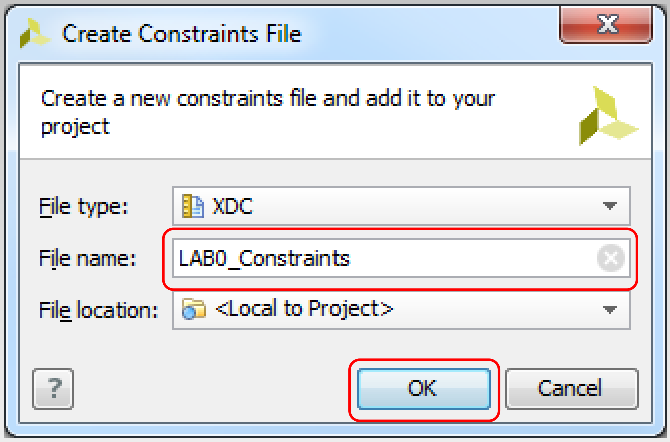
Add Sources

Add or Create Constraints

Specify or create constraint files for physical and timing constraint to add to your project.

Specify constraint set: constrs_1 (active)

Use Add Files or Create File buttons below



Create Constraints File

Create a new constraints file and add it to your project

File type: XDC

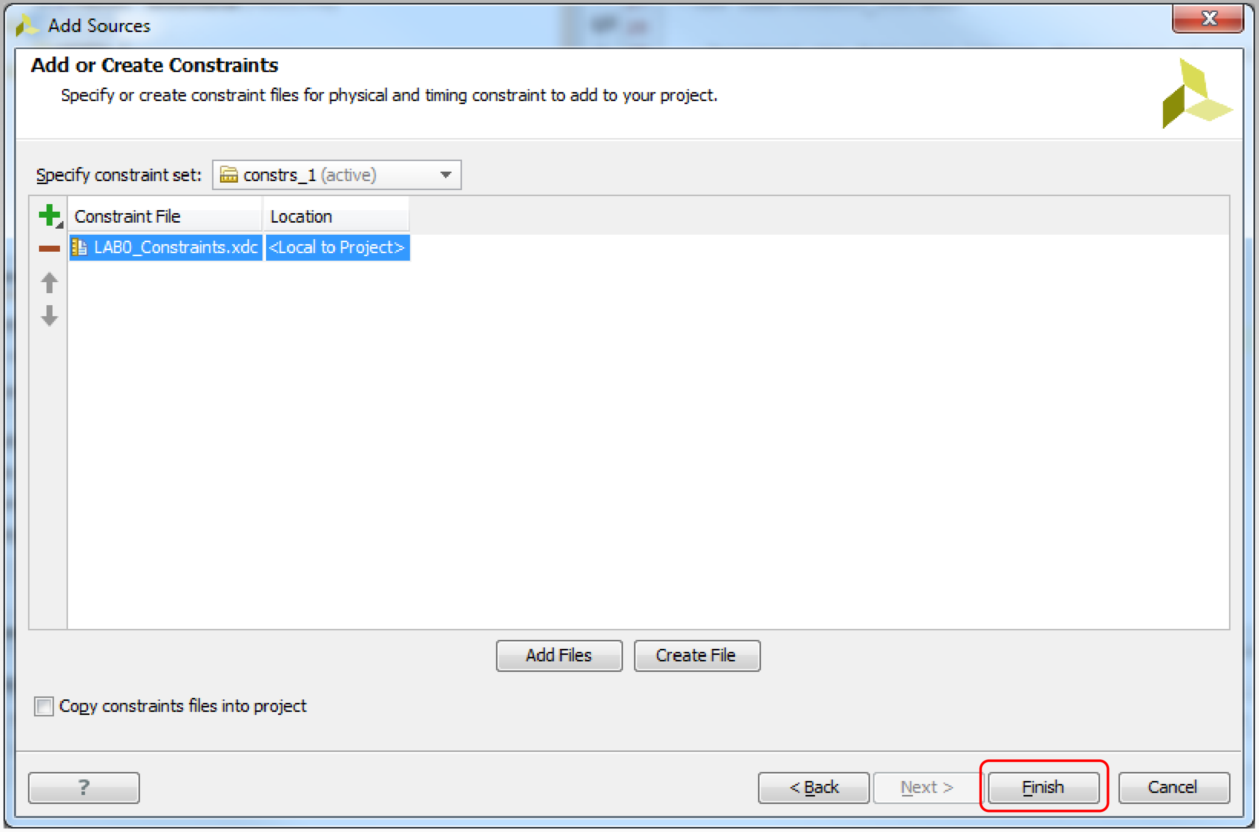
File name: LAB0_Constraints

File location: <Local to Project>

AUBURN UNIVERSITY
SAMUEL GINN COLLEGE OF ENGINEERING

Add Constraints (4)

- Click “Finish” to add your file to the project.


Add Sources

Add or Create Constraints

Specify or create constraint files for physical and timing constraint to add to your project.

Specify constraint set: constrs_1 (active)

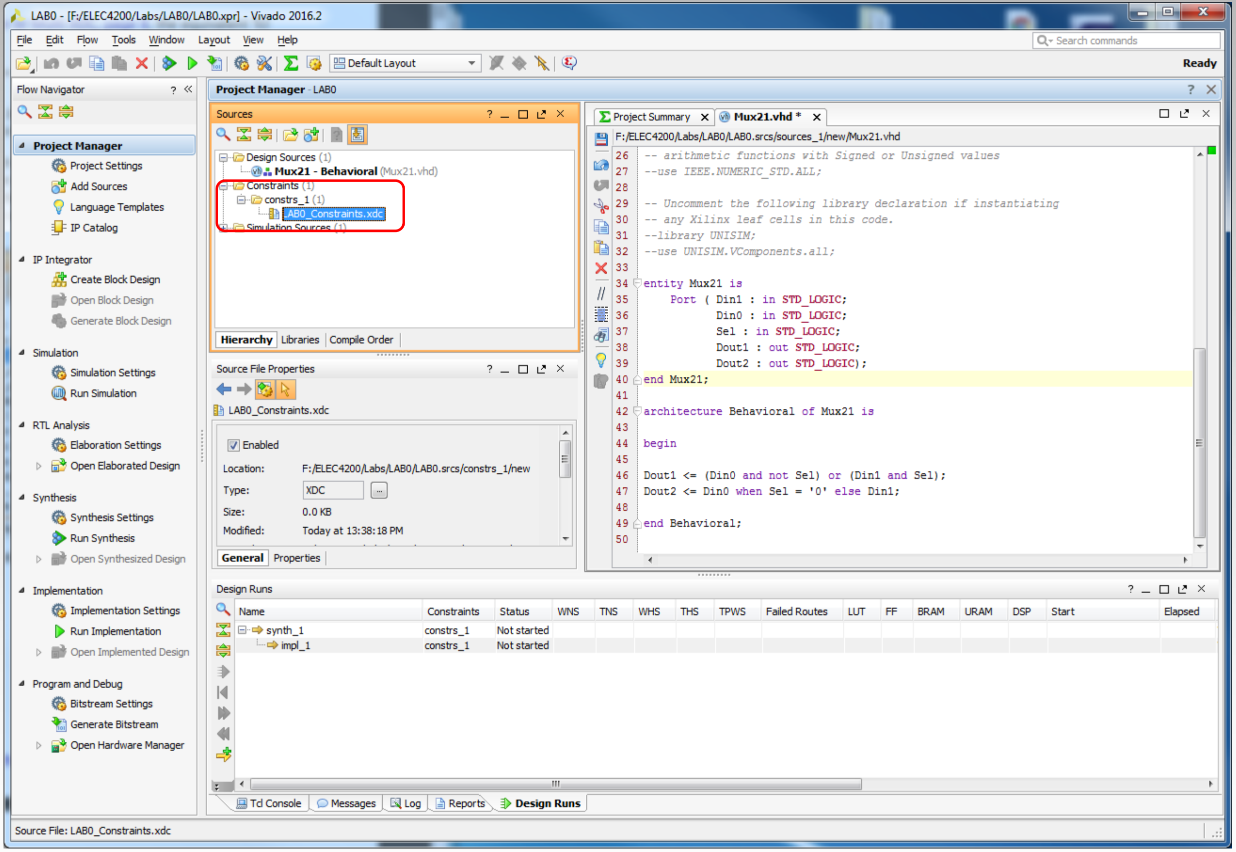
Constraint File	Location
LAB0_Constraints.xdc	<Local to Project>

AUBURN UNIVERSITY

SAMUEL GINN COLLEGE OF ENGINEERING

Add Constraints (5)

- Open your newly added file by double clicking on it from the Project Manager.



F:/ELEC4200/Labs/LAB0/LAB0.srcs/sources_1/new/Mux21.vhd

```
-- arithmetic functions with Signed or Unsigned values
--use IEEE.NUMERIC_STD.ALL;

-- Uncomment the following library declaration if instantiating
-- any Xilinx leaf cells in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity Mux21 is
    Port ( Din1 : in STD_LOGIC;
           Din0 : in STD_LOGIC;
           Sel : in STD_LOGIC;
           Dout1 : out STD_LOGIC;
           Dout2 : out STD_LOGIC);
end Mux21;

architecture Behavioral of Mux21 is

begin

Dout1 <= (Din0 and not Sel) or (Din1 and Sel);
Dout2 <= Din0 when Sel = '0' else Din1;

end Behavioral;

```

LAB0_Constraints.xdc

☑ Enabled

Location:	F:/ELEC4200/Labs/LAB0/LAB0.srcs/constrs_1/new
Type:	XDC
Size:	0.0 KB
Modified:	Today at 13:38:18 PM

Design Runs

Name	Constraints	Status	WNS	TNS	WHS	THS	TPWS	Failed Routes	LUT	FF	BRAM	URAM	DSP	Start	Elapsed
synth_1	constrs_1	Not started													
impl_1	constrs_1	Not started													

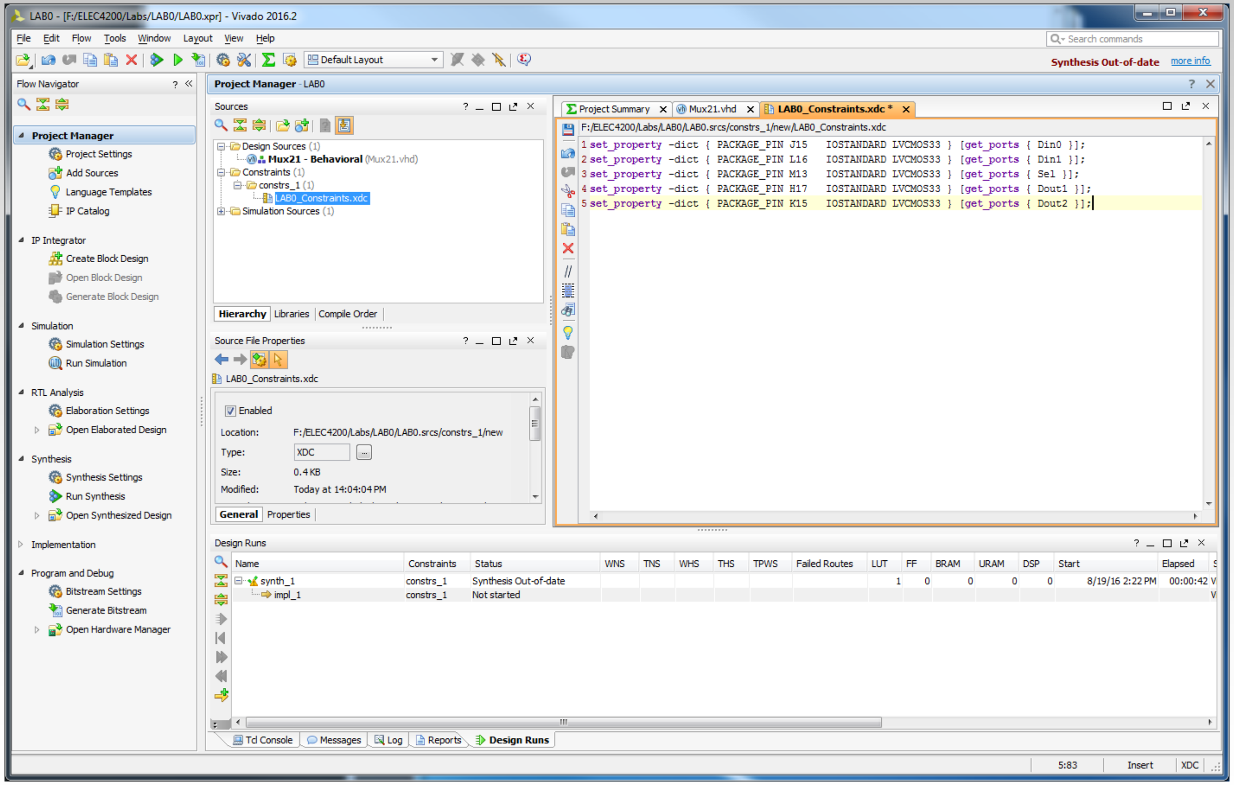
Source File: LAB0_Constraints.xdc

AUBURN UNIVERSITY
SAMUEL GINN COLLEGE OF ENGINEERING

Add Constraints (6)

- Add the following lines in the constraints file, renaming the ports to match those in your design. Save after you are done.

set_property -dict { PACKAGE_PIN J15   IOSTANDARD LVCMOS33 } [get_ports { **Din0** }];
set_property -dict { PACKAGE_PIN L16   IOSTANDARD LVCMOS33 } [get_ports { **Din1** }];
set_property -dict { PACKAGE_PIN M13   IOSTANDARD LVCMOS33 } [get_ports { **Sel** }];
set_property -dict { PACKAGE_PIN H17   IOSTANDARD LVCMOS33 } [get_ports { **Dout1** }];
set_property -dict { PACKAGE_PIN K15   IOSTANDARD LVCMOS33 } [get_ports { **Dout2** }];



F:/ELEC4200/Labs/LAB0/LAB0.srcs/constrs_1/new/LAB0_Constraints.xdc

```
set_property -dict { PACKAGE_PIN J15   IOSTANDARD LVCMOS33 } [get_ports { Din0 }];
set_property -dict { PACKAGE_PIN L16   IOSTANDARD LVCMOS33 } [get_ports { Din1 }];
set_property -dict { PACKAGE_PIN M13   IOSTANDARD LVCMOS33 } [get_ports { Sel }];
set_property -dict { PACKAGE_PIN H17   IOSTANDARD LVCMOS33 } [get_ports { Dout1 }];
set_property -dict { PACKAGE_PIN K15   IOSTANDARD LVCMOS33 } [get_ports { Dout2 }];
```

LAB0_Constraints.xdc

✓ Enabled	
Location:	F:/ELEC4200/Labs/LAB0/LAB0.srcs/constrs_1/new
Type:	XDC
Size:	0.4 KB
Modified:	Today at 14:04:04 PM

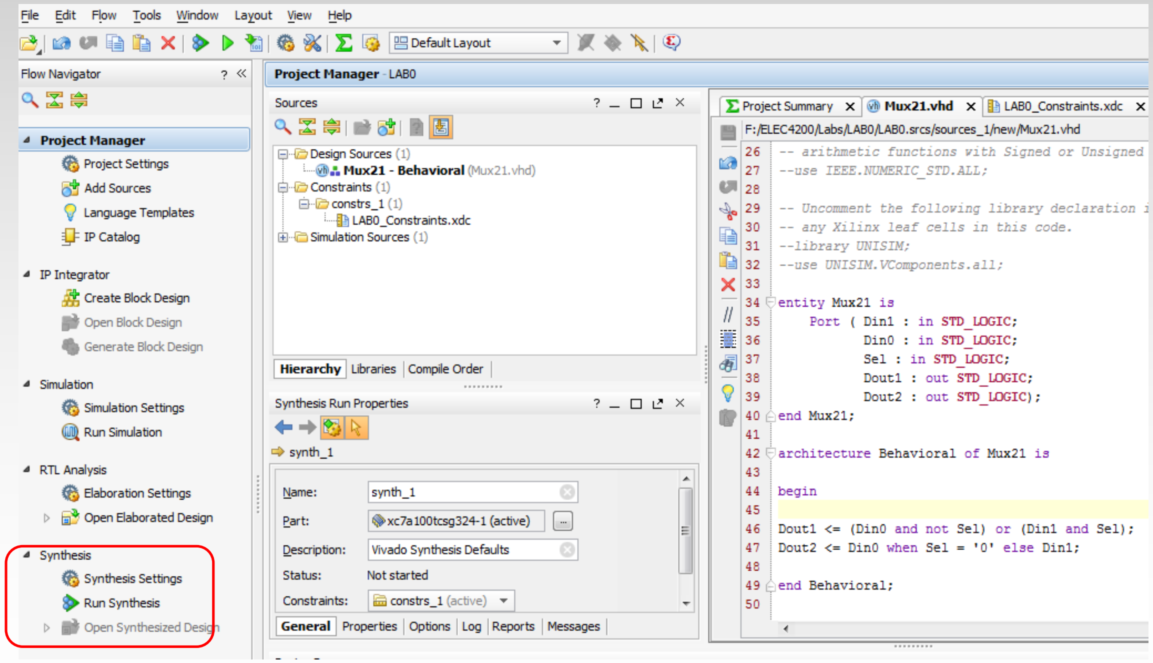
Design Runs

Name	Constraints	Status	WNS	TNS	WHS	THS	TPWS	Failed Routes	LUT	FF	BRAM	URAM	DSP	Start	Elapsed
synth_1	constrs_1	Synthesis Out-of-date							1	0	0	0	0	8/19/16 2:22 PM	00:00:42
impl_1	constrs_1	Not started													

AUBURN UNIVERSITY
SAMUEL GINN COLLEGE OF ENGINEERING

Implement the design(1)

- Now you are going to have *Vivado* “synthesize” your design into generic digital logic components.
- In the Flow Navigator, click “Run Synthesis”.
- This may take several minutes to run, so be patient.
- A dialog box will open on completion, either select “Run Implementation” or hit “Cancel”.



Name: synth_1
Part: xc7a100tcsg324-1 (active)
Description: Vivado Synthesis Defaults
Status: Not started
Constraints: constrs_1 (active)

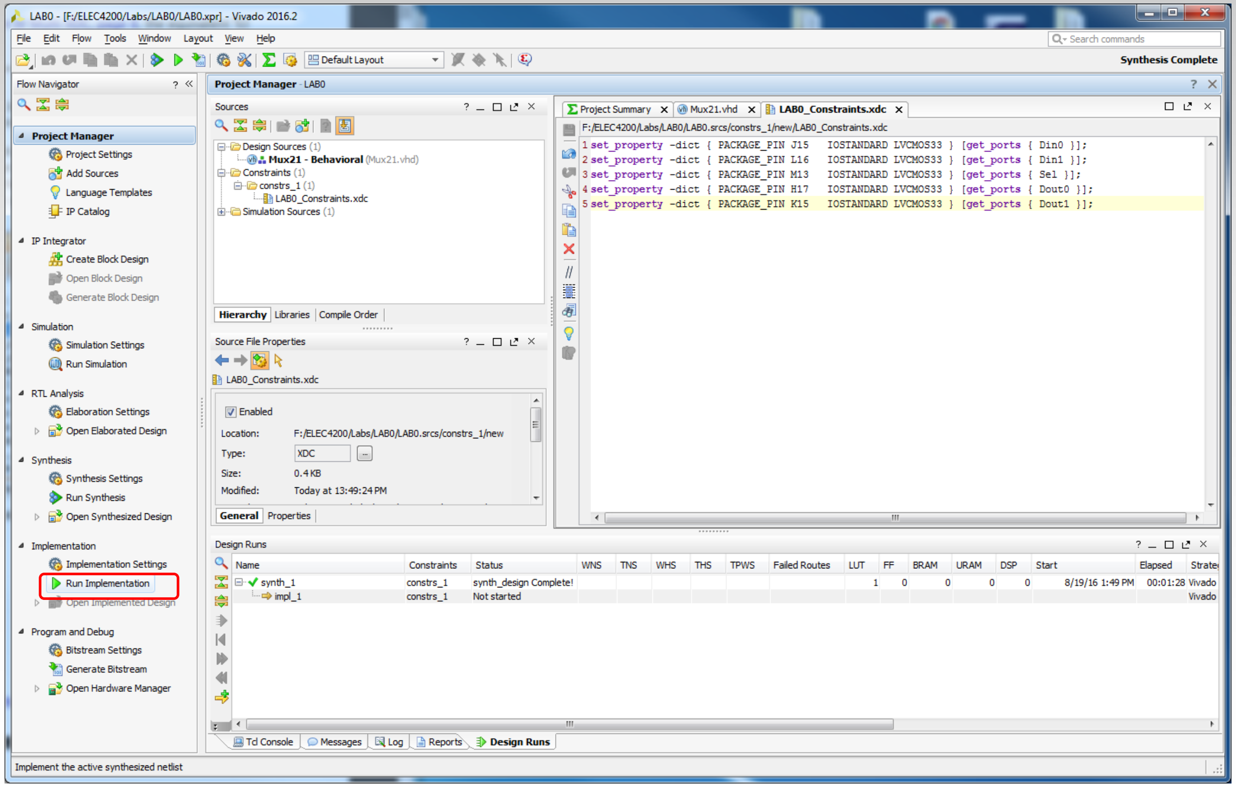
F:/ELEC4200/Labs/LAB0/LAB0.srcs/sources_1/new/Mux21.vhd

```
-- arithmetic functions with Signed or Unsigned
--use IEEE.NUMERIC_STD.ALL;

-- Uncomment the following library declaration i
-- any Xilinx leaf cells in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity Mux21 is
    Port ( Din1 : in STD_LOGIC;
           Din0 : in STD_LOGIC;
           Sel : in STD_LOGIC;
           Dout1 : out STD_LOGIC;
           Dout2 : out STD_LOGIC);
end Mux21;

architecture Behavioral of Mux21 is

begin

Dout1 <= (Din0 and not Sel) or (Din1 and Sel);
Dout2 <= Din0 when Sel = '0' else Din1;

end Behavioral;

```

AUBURN UNIVERSITY
SAMUEL GINN COLLEGE OF ENGINEERING

Implement the design (2)

- Click on “Implement Design”. This maps the synthesized design to the hardware, and routes I/O ports to the pins specified in the constraints file.
- Again, this will take several minutes to run.
- A dialog box will open on completion, either select “Generate Bitstream” or hit “Cancel”.



Enabled
Location: F:/ELEC4200/Labs/LAB0/LAB0.srcs/constrs_1/new
Type: XDC
Size: 0.4 KB
Modified: Today at 13:49:24 PM

F:/ELEC4200/Labs/LAB0/LAB0.srcs/constrs_1/new/LAB0_Constraints.xdc

```
set_property -dict { PACKAGE_PIN J15   IOSTANDARD LVCMOS33 } [get_ports { Din0 }];
set_property -dict { PACKAGE_PIN L16   IOSTANDARD LVCMOS33 } [get_ports { Din1 }];
set_property -dict { PACKAGE_PIN M13   IOSTANDARD LVCMOS33 } [get_ports { Sel }];
set_property -dict { PACKAGE_PIN H17   IOSTANDARD LVCMOS33 } [get_ports { Dout0 }];
set_property -dict { PACKAGE_PIN K15   IOSTANDARD LVCMOS33 } [get_ports { Dout1 }];
```

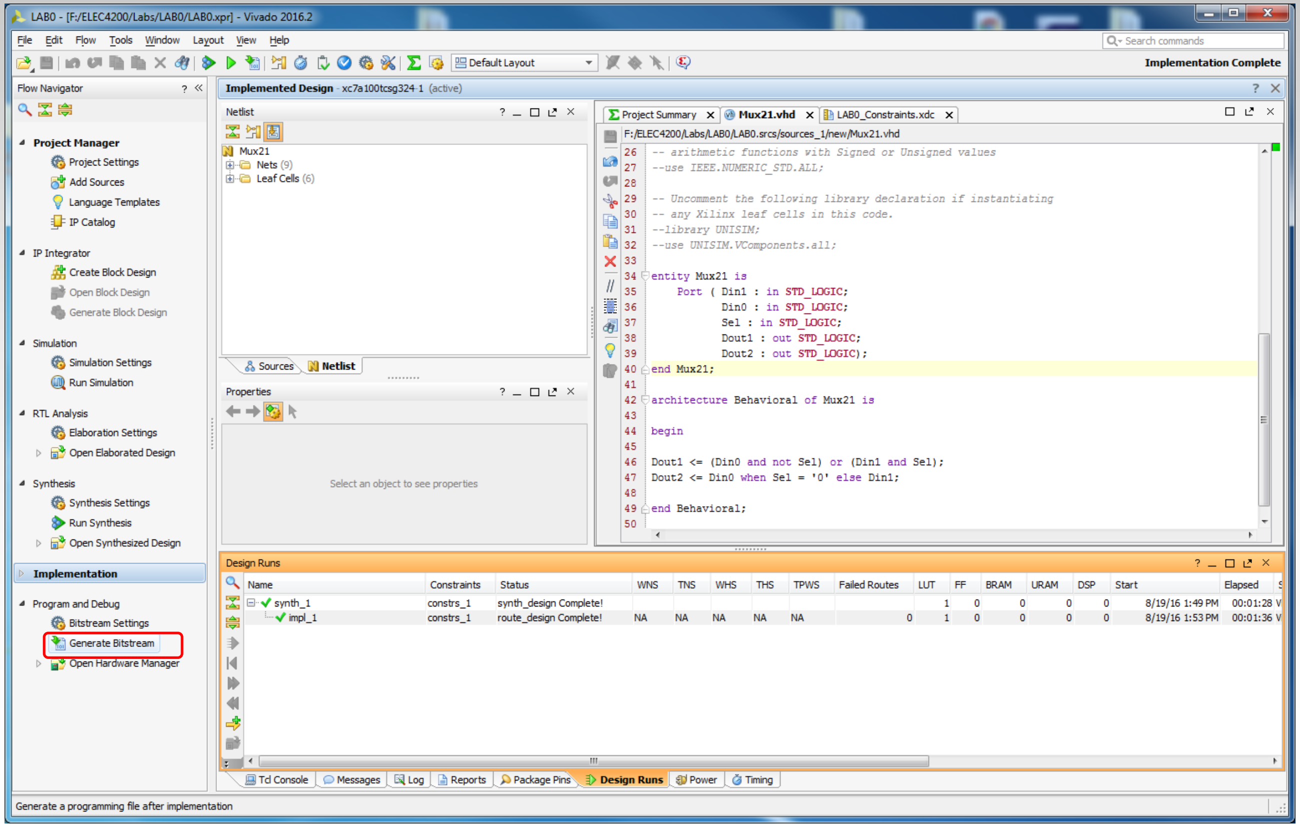
Design Runs

Name	Constraints	Status	WNS	TNS	WHS	THS	TPWS	Failed Routes	LUT	FF	BRAM	URAM	DSP	Start	Elapsed	Strate
synth_1	constrs_1	synth_design Complete!							1	0	0	0	0	8/19/16 1:49 PM	00:01:28	Vivado
impl_1	constrs_1	Not started														Vivado

AUBURN
UNIVERSITY
SAMUEL GINN
COLLEGE OF ENGINEERING

Generate Bitstream

- With the design now implemented, it is time to generate the bitstream, or the file that will be downloaded to the FPGA.
- Click “Generate Bitstream”
- A dialog box will open on completion, either select “Open Hardware Manager” or hit “Cancel”.



```
-- arithmetic functions with Signed or Unsigned values
--use IEEE.NUMERIC_STD.ALL;

-- Uncomment the following library declaration if instantiating
-- any Xilinx leaf cells in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity Mux21 is
    Port ( Din1 : in STD_LOGIC;
           Din0 : in STD_LOGIC;
           Sel : in STD_LOGIC;
           Dout1 : out STD_LOGIC;
           Dout2 : out STD_LOGIC);
end Mux21;

architecture Behavioral of Mux21 is

begin

Dout1 <= (Din0 and not Sel) or (Din1 and Sel);
Dout2 <= Din0 when Sel = '0' else Din1;

end Behavioral;
```

Design Runs

Name	Constraints	Status	WNS	TNS	WHS	THS	TPWS	Failed Routes	LUT	FF	BRAM	URAM	DSP	Start	Elapsed
✓ synth_1	constrs_1	synth_design Complete!							1	0	0	0	0	8/19/16 1:49 PM	00:01:28
✓ impl_1	constrs_1	route_design Complete!	NA	NA	NA	NA	NA	0	1	0	0	0	0	8/19/16 1:53 PM	00:01:36

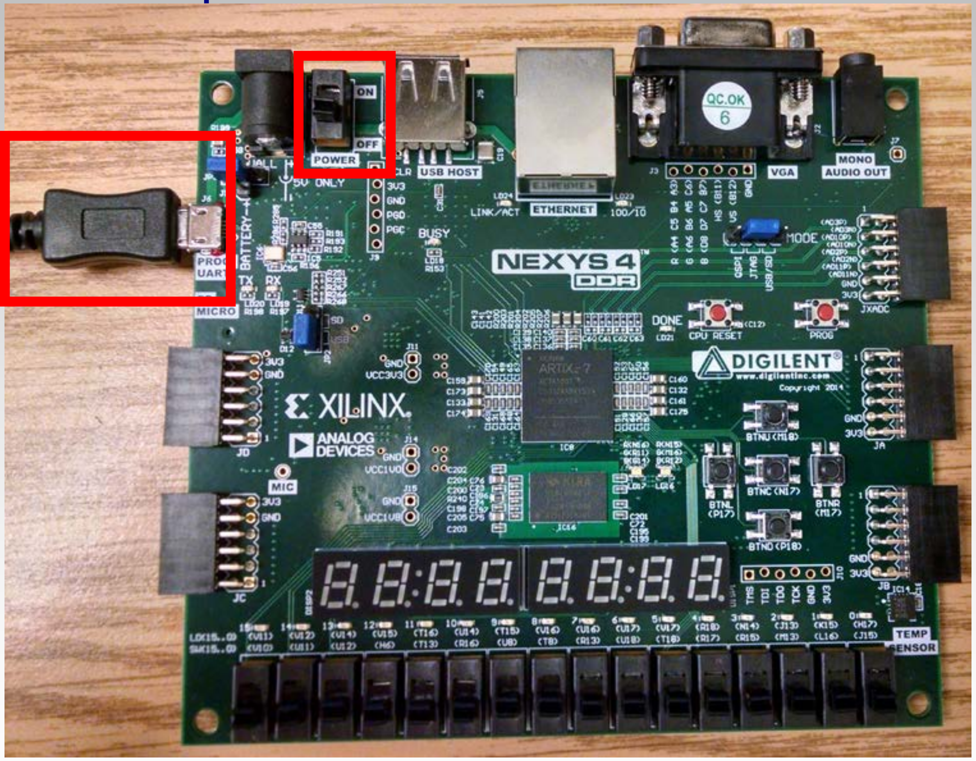
Generate a programming file after implementation

AUBURN UNIVERSITY
Samuel Ginn College of Engineering

Download to Hardware(1)

- Plug in the USB cable between a PC USB port and the USB port on the FPGA board to access the JTAG programming module.

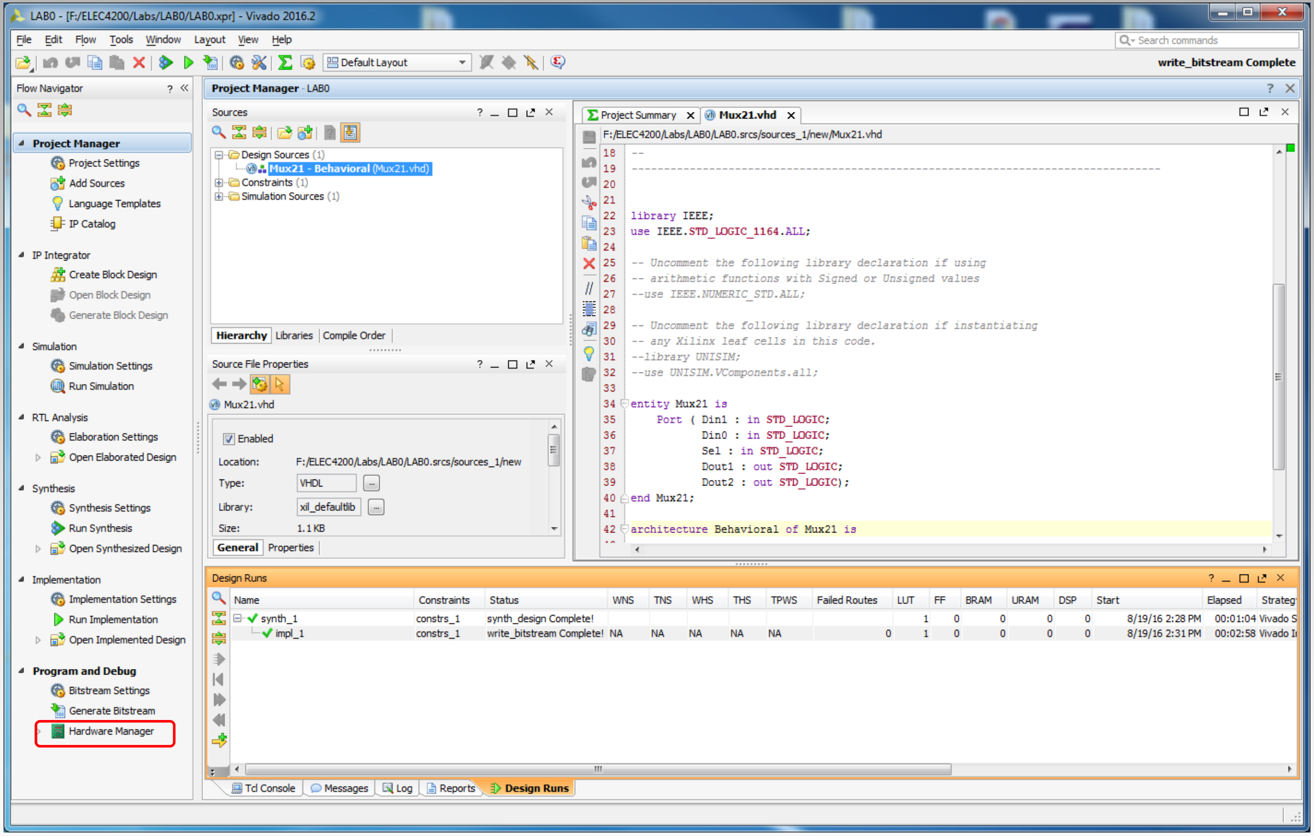
- Flip the board power switch from OFF to ON



AUBURN UNIVERSITY
SAMUEL GINN COLLEGE OF ENGINEERING

Download to Hardware (2)

- Click on “Hardware Manager”



Source File Properties:

Enabled	✓
Location:	F:/ELEC4200/Labs/LAB0/LAB0.srcs/sources_1/new
Type:	VHDL
Library:	xil_defaultlib
Size:	1.1 KB

F:/ELEC4200/Labs/LAB0/LAB0.srcs/sources_1/new/Mux21.vhd

```
--
------------------------------------------------------------------------------


library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--use IEEE.NUMERIC_STD.ALL;

-- Uncomment the following library declaration if instantiating
-- any Xilinx leaf cells in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity Mux21 is
    Port ( Din1 : in STD_LOGIC;
           Din0 : in STD_LOGIC;
           Sel : in STD_LOGIC;
           Dout1 : out STD_LOGIC;
           Dout2 : out STD_LOGIC);
end Mux21;

architecture Behavioral of Mux21 is
```

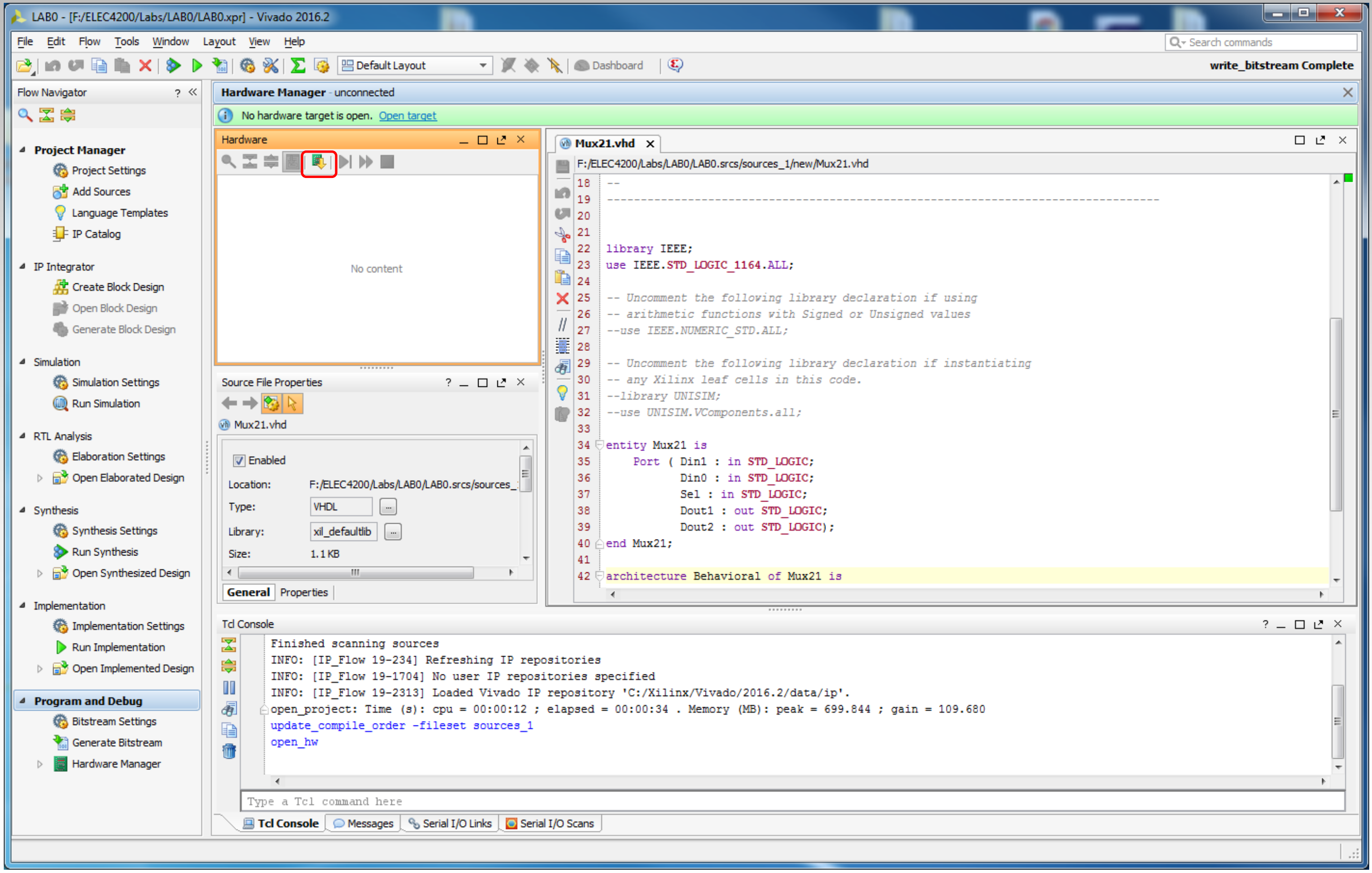
Design Runs

Name	Constraints	Status	WNS	TNS	WHS	THS	TPWS	Failed Routes	LUT	FF	BRAM	URAM	DSP	Start	Elapsed	Strateg
✓ synth_1	constrs_1	synth_design Complete!							1	0	0	0	0	8/19/16 2:28 PM	00:01:04	Vivado S
✓ impl_1	constrs_1	write_bitstream Complete!	NA	NA	NA	NA	NA	0	1	0	0	0	0	8/19/16 2:31 PM	00:02:58	Vivado I

AUBURN UNIVERSITY
Samuel Ginn College of Engineering

Download to Hardware (3)

- Click “Auto Connect” to connect to your Nexys 4 board.
- If you encounter any errors, verify that the board is plugged in and powered on.



Enabled
Location: F:/ELEC4200/Labs/LAB0/LAB0.srcs/sources_
Type: VHDL
Library: xil_defaultlib
Size: 1.1 KB

Mux21.vhd
F:/ELEC4200/Labs/LAB0/LAB0.srcs/sources_1/new/Mux21.vhd

```
--
--------------------------------------------------------------------------------


library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--use IEEE.NUMERIC_STD.ALL;

-- Uncomment the following library declaration if instantiating
-- any Xilinx leaf cells in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity Mux21 is
    Port ( Din1 : in STD_LOGIC;
           Din0 : in STD_LOGIC;
           Sel : in STD_LOGIC;
           Dout1 : out STD_LOGIC;
           Dout2 : out STD_LOGIC);
end Mux21;

architecture Behavioral of Mux21 is
```

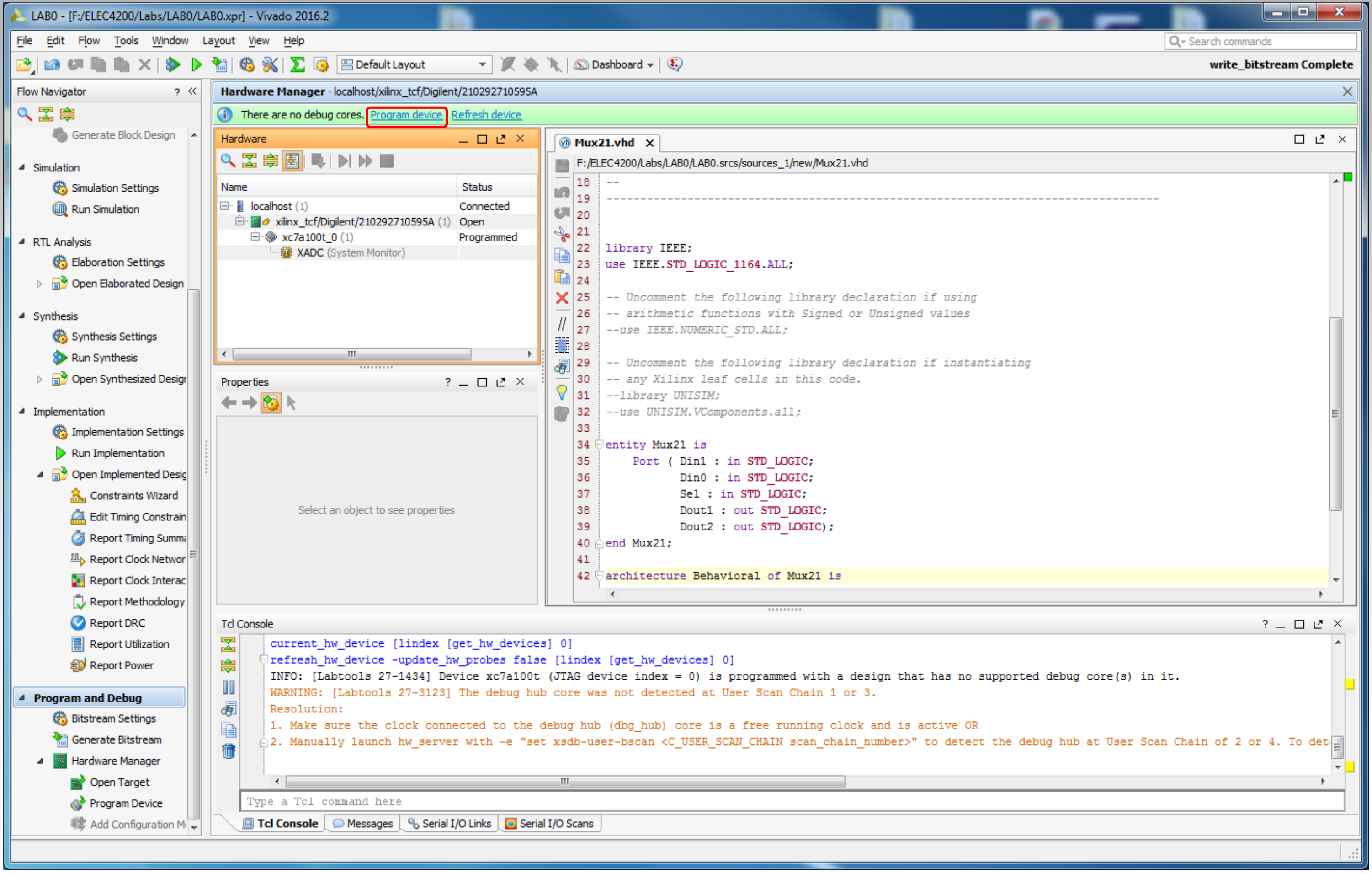
Tcl Console

```
Finished scanning sources
INFO: [IP_Flow 19-234] Refreshing IP repositories
INFO: [IP_Flow 19-1704] No user IP repositories specified
INFO: [IP_Flow 19-2313] Loaded Vivado IP repository 'C:/Xilinx/Vivado/2016.2/data/ip'.
open_project: Time (s): cpu = 00:00:12 ; elapsed = 00:00:34 . Memory (MB): peak = 699.844 ; gain = 109.680
update_compile_order -fileset sources_1
open_hw
```

Type a Tcl command here

Download to Hardware(4)

- Click “Program Device” and select xc7a100t_0 from the list.



There are no debug cores. Program device Refresh device

Name	Status
localhost (1)	Connected
xilinx_tcf/Digilent/210292710595A (1)	Open
xc7a100t_0 (1)	Programmed
XADC (System Monitor)	

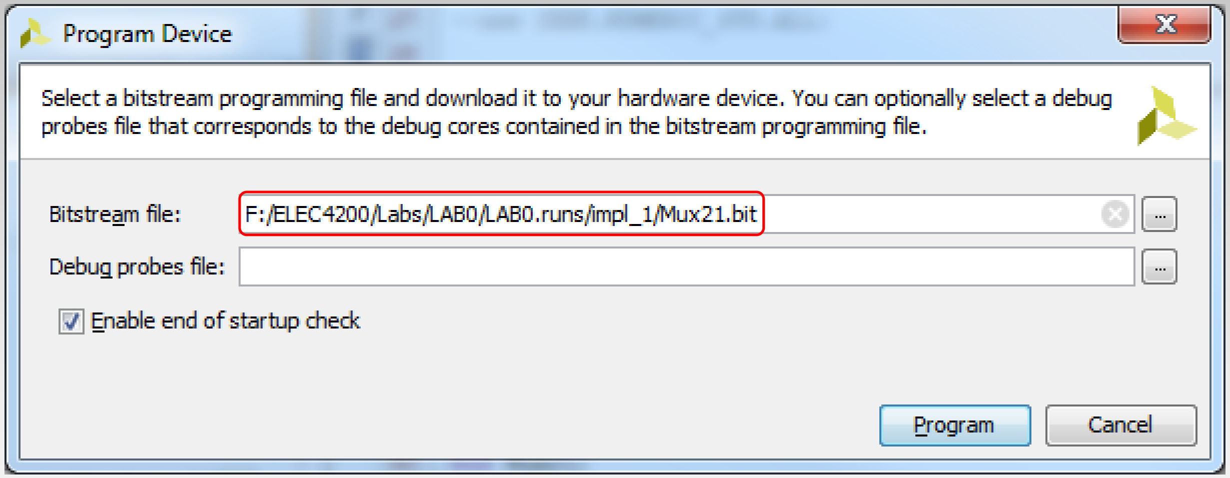
```
--
------------------------------------------------------------------------------------


library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--use IEEE.NUMERIC_STD.ALL;

-- Uncomment the following library declaration if instantiating
-- any Xilinx leaf cells in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity Mux21 is
    Port ( Din1 : in STD_LOGIC;
           Din0 : in STD_LOGIC;
           Sel : in STD_LOGIC;
           Dout1 : out STD_LOGIC;
           Dout2 : out STD_LOGIC);
end Mux21;

architecture Behavioral of Mux21 is
```

```
current_hw_device [lindex [get_hw_devices] 0]
refresh_hw_device -update_hw_probes false [lindex [get_hw_devices] 0]
INFO: [Labtools 27-1434] Device xc7a100t (JTAG device index = 0) is programmed with a design that has no supported debug core(s) in it.
WARNING: [Labtools 27-3123] The debug hub core was not detected at User Scan Chain 1 or 3.
Resolution:
1. Make sure the clock connected to the debug hub (dbg_hub) core is a free running clock and is active OR
2. Manually launch hw_server with -e "set xsdb-user-bscan <C_USER_SCAN_CHAIN scan_chain_number>" to detect the debug hub at User Scan Chain of 2 or 4. To det
```

Type a Tcl command here

Download to Hardware(5)

- In the dialog box, make sure the correct bit file is selected.
- Leave the "Debug probes file" box empty for now.
- Click "Program"



Program Device

Select a bitstream programming file and download it to your hardware device. You can optionally select a debug probes file that corresponds to the debug cores contained in the bitstream programming file.

Bitstream file: F:/ELEC4200/Labs/LAB0/LAB0.runs/impl_1/Mux21.bit

Debug probes file:

☑ Enable end of startup check

AUBURN UNIVERSITY
SAMUEL GINN COLLEGE OF ENGINEERING

Download to Hardware(6)

- Verify correct operation of the circuit using the switches and observe the output on the LEDs. If you encounter any bugs, attempt to figure out what went wrong before asking for help.
- Ensure that you apply all possible input combinations, as you did in simulation, and verify that the outputs match the simulation results.
- After verifying the circuit is correct, call the GTA over and demonstrate the circuit.

Clean up

- Turn off the board, unplug the USB cable, put them back in their box, and return the box to the shelf.
- Close Vivado, any other open programs, and save all files.
- Don’t forget to log out of your machine and take any USB drives with you.

AUBURN UNIVERSITY
SAMUEL GINN COLLEGE OF ENGINEERING