UNIVERSITI KUALA LUMPUR
MALAYSIAN SPANISH INSTITUTE

SAB36603
ADVANCE DIGITAL DESIGN AND FPGA

LAB 2
BASIC LOGIC DESIGN AND I/O INTERFACE

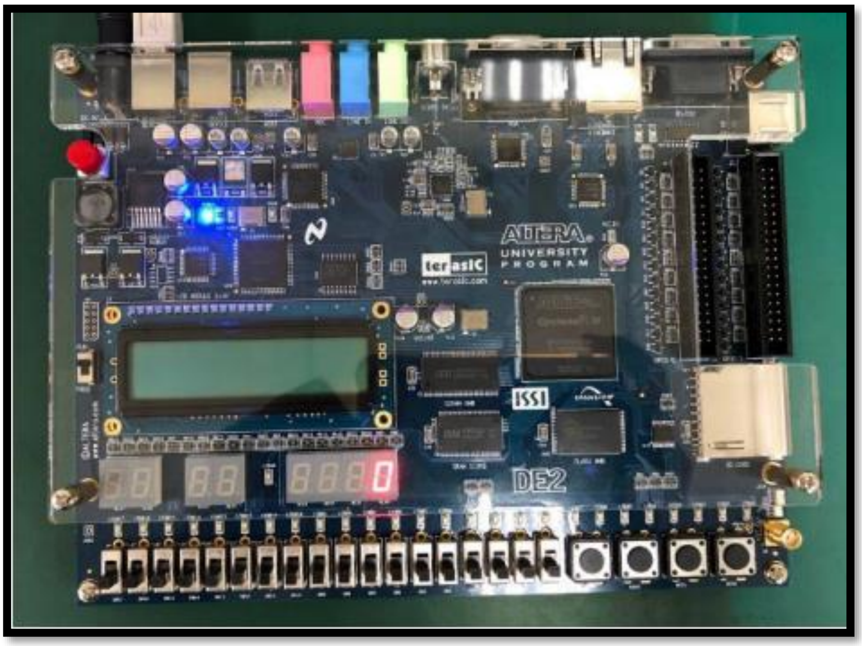
PREPARED BY:

STUDENT'S NAME	ID NUMBER
MUHAMMAD ADIEL ADAM BIN MOHD NOOR	54222121212
KHAIRUL ANWAR BIN KHAIRUL SALLEH	54215121186

LECTURER'S NAME: ENCIK MOHD REZAL BIN MOHAMED

1. Introduction

This lab introduces the Altera Quartus II design environment for implementing FPGA-based designs using the Altera DE2 board. The schematic entry method is employed to create designs. After creating a design, it undergoes compilation to check for functionality and timing correctness. Verification and timing analysis may require iterations to refine the design. Once the design meets requirements, it can be implemented in the FPGA on the Altera DE2 board.



2. Objective

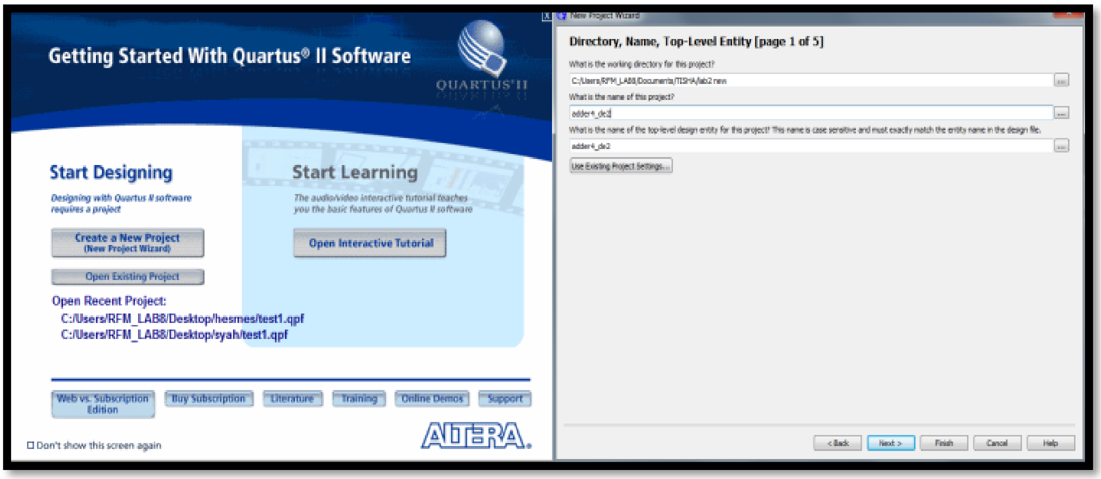
The objectives of this lab are as follows:

- Learn about the schematic-based design flow for FPGA implementation, which involves employing logic gates and hierarchical modeling.
- Create fundamental logic gates and input/output interface.
- Implement the design on the Altera DE2 board using FPGA board.

3. Schematic / HDL language

1) Schematic (BDF) for 4-Bit Adder:

- Open Quartus II.
- Create a new project named "adder4_de2" using the New Project Wizard.



Getting Started With Quartus® II Software

Start Designing

Designing with Quartus II software requires a project

Start Learning

The audio/video interactive tutorial teaches you the basic features of Quartus II software

Open Recent Project:
C:/Users/RFM_LAB8/Desktop/hesmes/test1.qpf
C:/Users/RFM_LAB8/Desktop/syah/test1.qpf

□ Don't show this screen again

New Project Wizard

Directory, Name, Top-Level Entity [page 1 of 5]

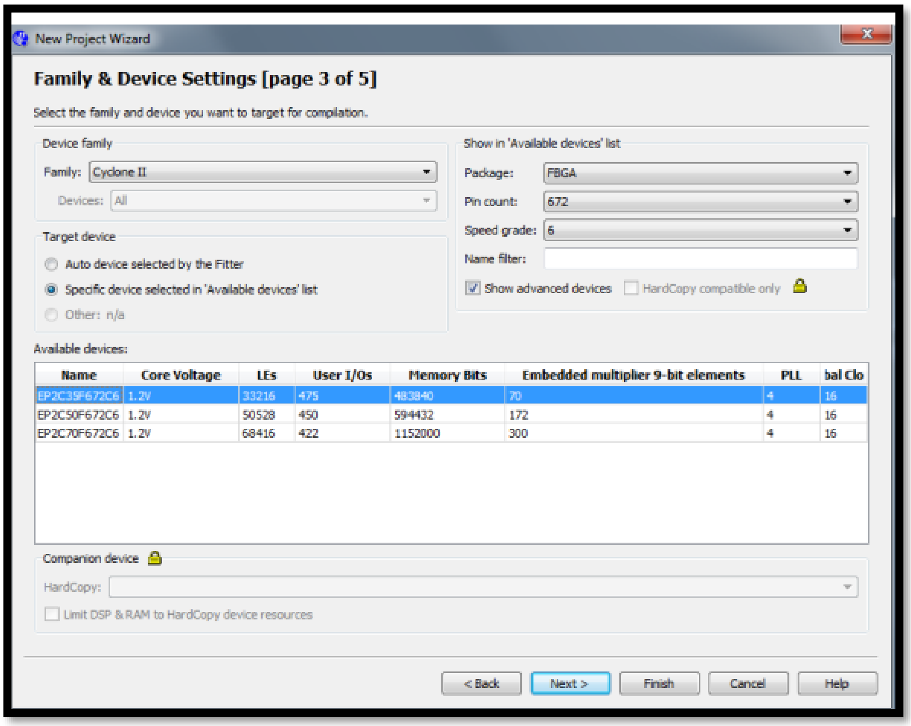
What is the working directory for this project?
C:/Users/RFM_LAB8/Documents/TISHA/lab2 new

What is the name of this project?
adder4_de2

What is the name of the top-level design entity for this project? This name is case sensitive and must exactly match the entity name in the design file.
adder4_de2

QUARTUS II
Create a New Project (New Project Wizard)
Open Existing Project
Open Interactive Tutorial
Web vs. Subscription Edition
Buy Subscription
Literature
Training
Online Demos
Support
ALTERA.
Use Existing Project Settings...
< Back
Next >
Finish
Cancel
Help

- Select Cyclone II in the Family list and EP2C35F672C6 in the Available devices list.



New Project Wizard

Family & Device Settings [page 3 of 5]

Select the family and device you want to target for compilation.

Device family
Family: Cyclone II
Devices: All

Target device
○ Auto device selected by the Fitter
● Specific device selected in 'Available devices' list
○ Other: n/a

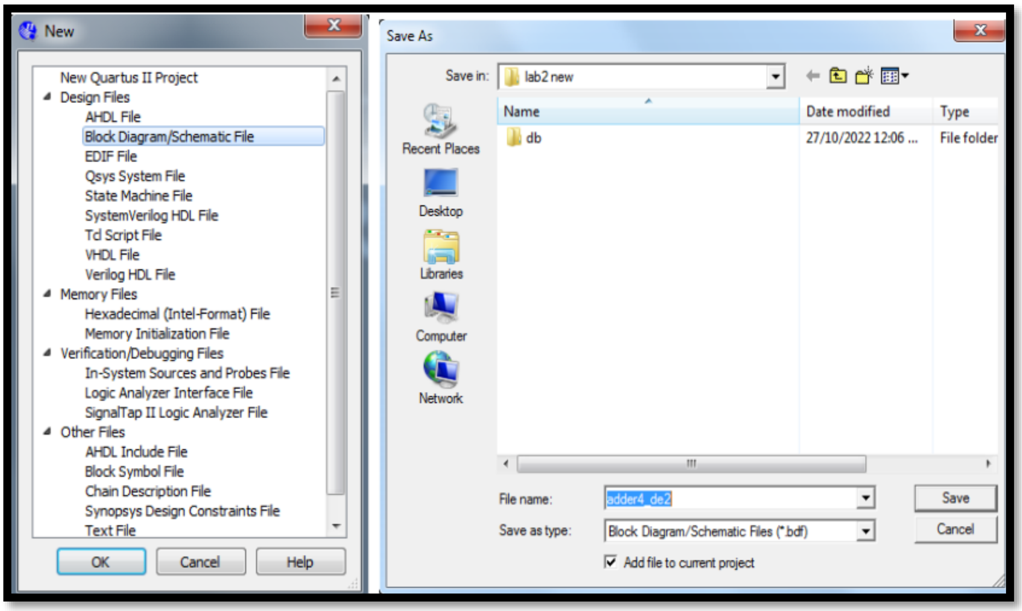
Show in 'Available devices' list
Package: FBGA
Pin count: 672
Speed grade: 6
Name filter:
☑ Show advanced devices □ HardCopy compatible only

Available devices:

Name	Core Voltage	LEs	User I/Os	Memory Bits	Embedded multiplier 9-bit elements	PLL	bal Clo
EP2C35F672C6	1.2V	33216	475	483840	70	4	16
EP2C50F672C6	1.2V	50528	450	594432	172	4	16
EP2C70F672C6	1.2V	68416	422	1152000	300	4	16

Companion device
HardCopy:
□ Limit DSP & RAM to HardCopy device resources

- Create a new Block Diagram File (BDF) called "adder4_de2.bdf" as the top-level design entity of the project.



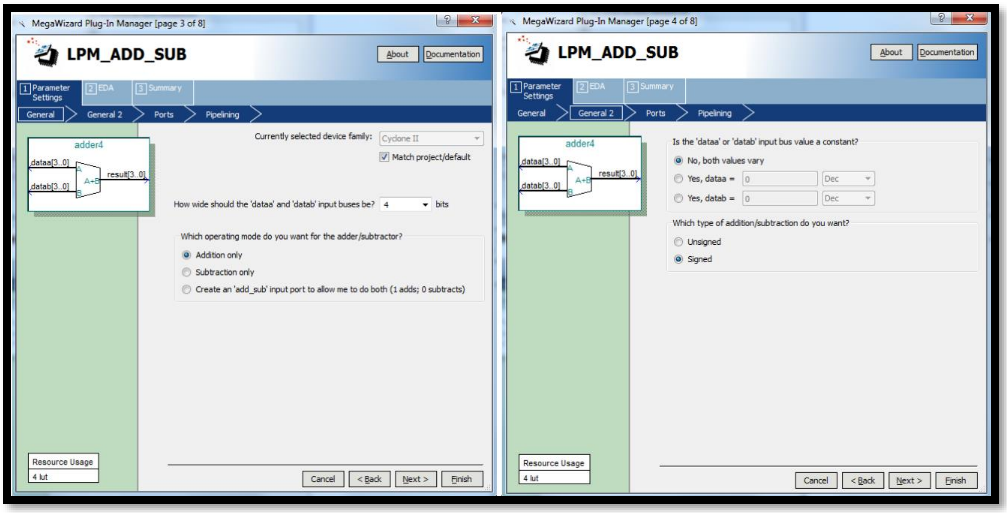
Save in: lab2 new

File name: adder4_de2

Save as type: Block Diagram/Schematic Files (*.bdf)

☑ Add file to current project

- Utilize the MegaWizard Plug-In Manager to create a 4-bit adder variation of the LPM_ADD_SUB megafunction.



Currently selected device family: Cyclone II

☑ Match project/default

How wide should the 'dataa' and 'datab' input buses be? 4 bits

Which operating mode do you want for the adder/subtractor?

● Addition only

○ Subtraction only

○ Create an 'add_sub' input port to allow me to do both (1 adds; 0 subtracts)

Is the 'dataa' or 'datab' input bus value a constant?

● No, both values vary

○ Yes, dataa = 0 Dec

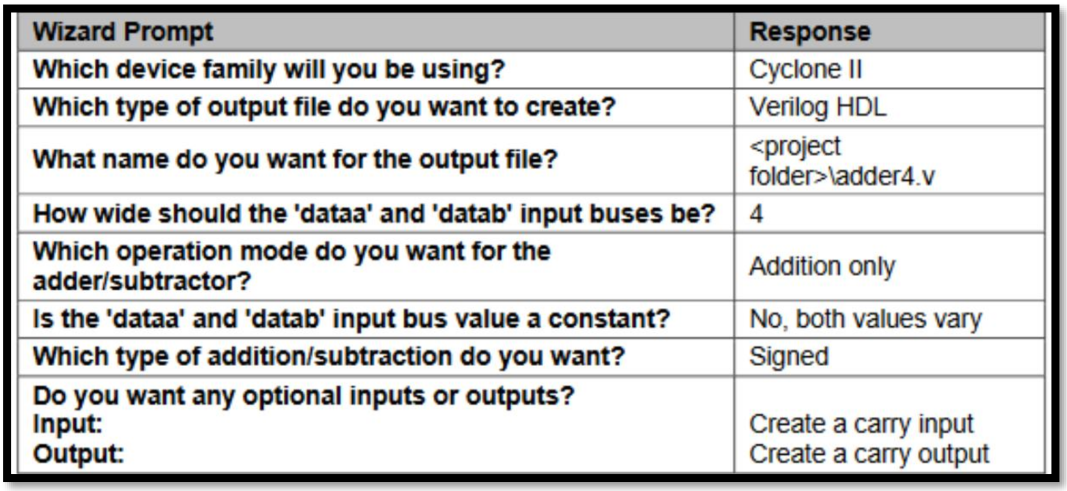
○ Yes, datab = 0 Dec

Which type of addition/subtraction do you want?

○ Unsigned

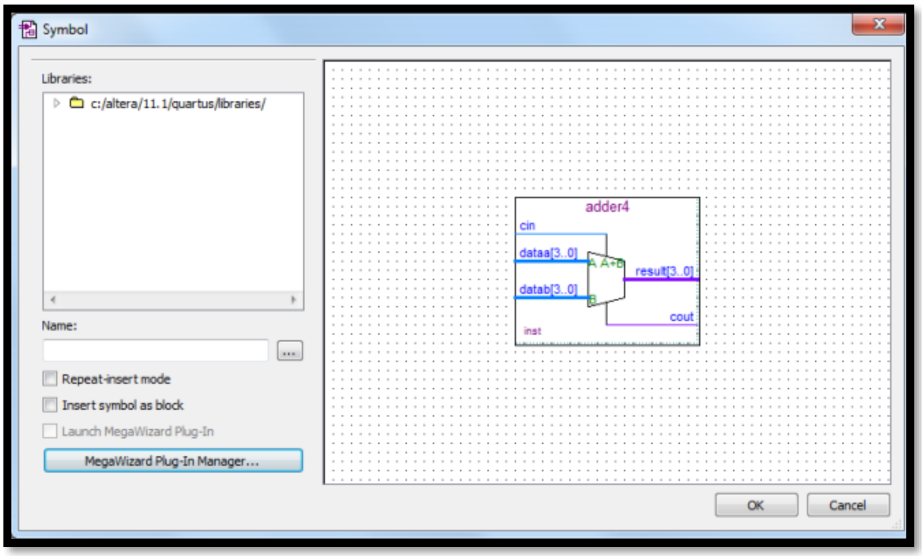
● Signed

- Configure parameters such as operation mode, input bus width, and optional inputs/outputs.



Wizard Prompt	Response
Which device family will you be using?	Cyclone II
Which type of output file do you want to create?	Verilog HDL
What name do you want for the output file?	<project folder>\adder4.v
How wide should the 'dataa' and 'datab' input buses be?	4
Which operation mode do you want for the adder/subtractor?	Addition only
Is the 'dataa' and 'datab' input bus value a constant?	No, both values vary
Which type of addition/subtraction do you want?	Signed
Do you want any optional inputs or outputs? Input: Output:	Create a carry input Create a carry output

- Place the generated symbol in the Block Editor window.

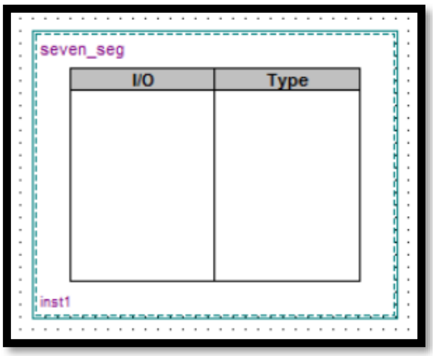


- Then, establish connections between input and output ports of the adder.

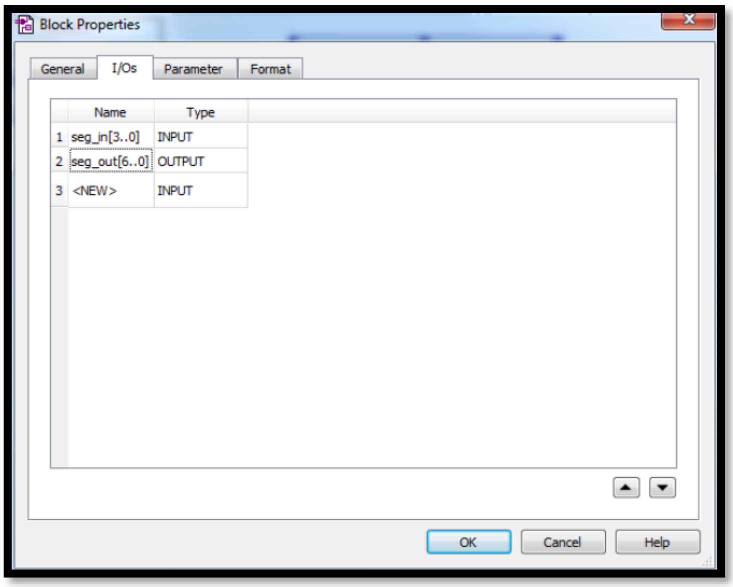
2) Verilog HDL Design File for 7-Segment Decoder:

- Open the schematic window and select the Block Tool.

- Place the block anywhere on the workspace and double-click on it to rename it to "seven_seg".



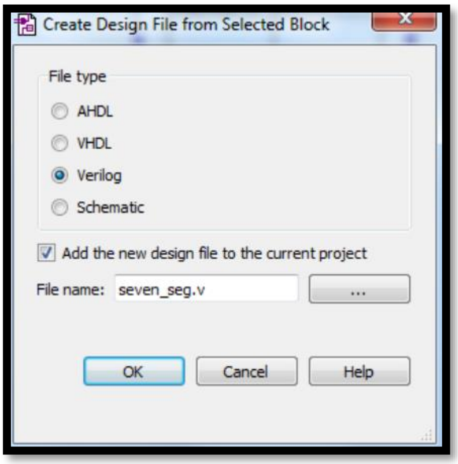
- Right-click the block and select Block Properties. Under the I/Os tab, define the input/output ports.



	Name	Type
1	seg_in[3..0]	INPUT
2	seg_out[6..0]	OUTPUT
3	<NEW>	INPUT

- Right-click the block again and select Create Design File from Selected Block.

- In the dialog box, select Verilog HDL as the file type.



Create Design File from Selected Block

File type

○ AHDL

○ VHDL

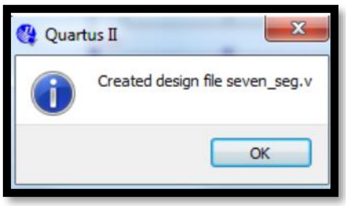
● Verilog

○ Schematic

☑ Add the new design file to the current project

File name: seven_seg.v

- Make sure to add the new design file to the current project and name the file "seven_seg.v".

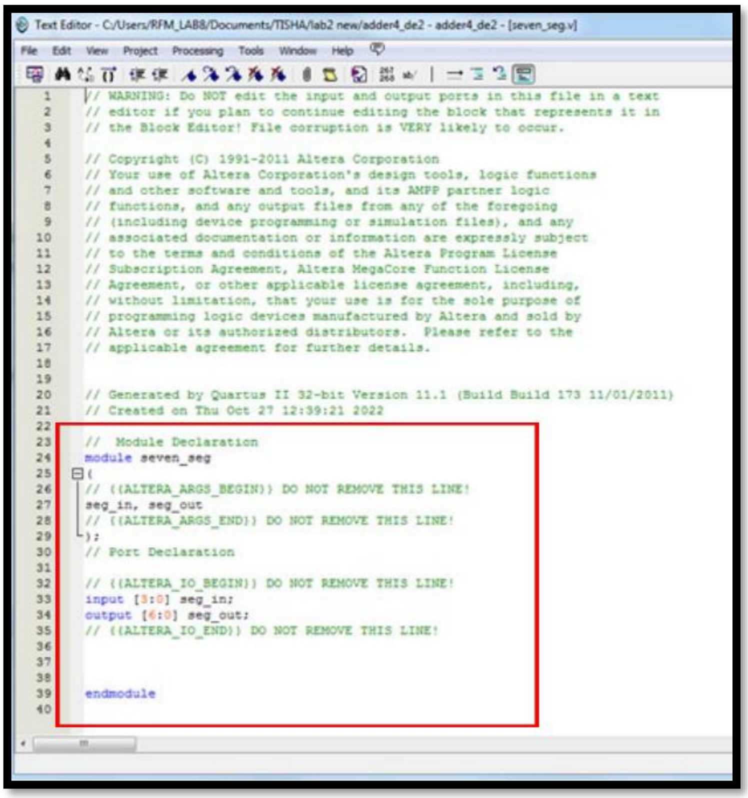


Quartus II

Created design file seven_seg.v

- Click OK to generate the file.

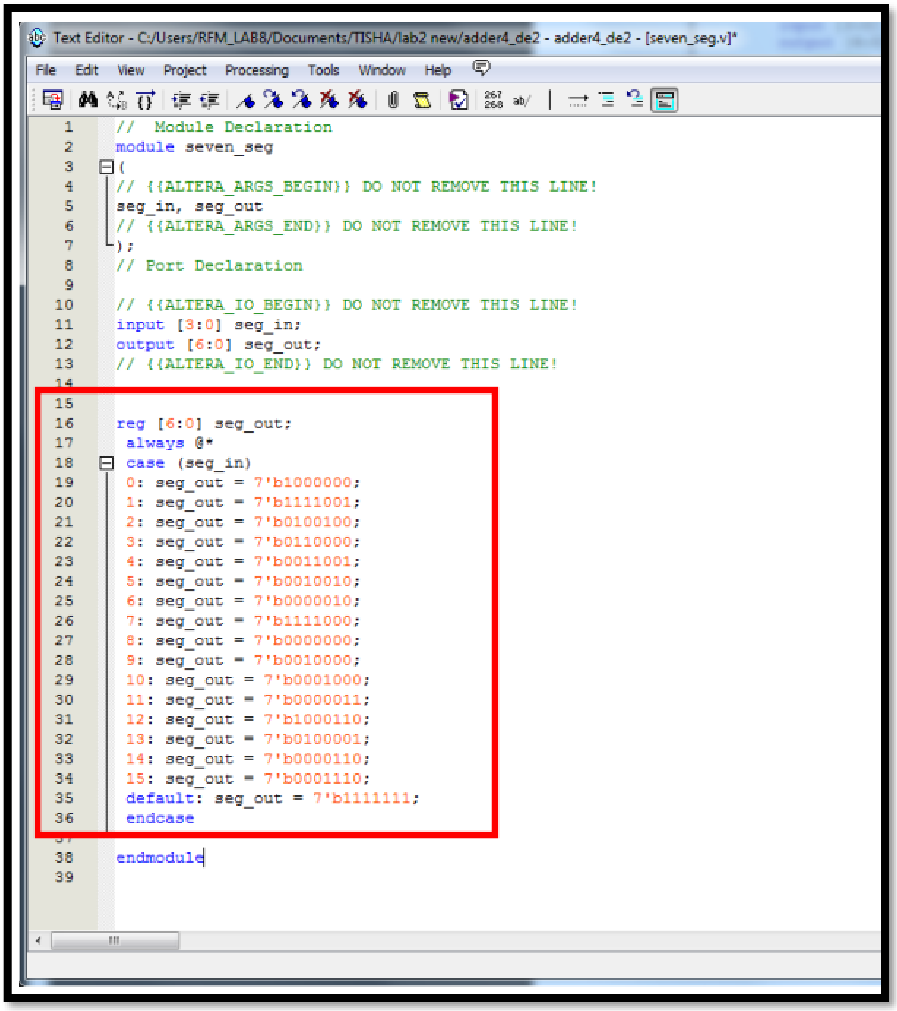
- The Quartus II software will automatically open a text editor window containing the new file.



Text Editor - C:/Users/RFM_LAB8/Documents/TISHA/lab2 new/adder4_de2 - adder4_de2 - [seven_seg.v]

```
// WARNING: Do NOT edit the input and output ports in this file in a text
// editor if you plan to continue editing the block that represents it in
// the Block Editor! File corruption is VERY likely to occur.

// Copyright (C) 1991-2011 Altera Corporation
// Your use of Altera Corporation's design tools, logic functions
// and other software and tools, and its AMPP partner logic
// functions, and any output files from any of the foregoing
// (including device programming or simulation files), and any
// associated documentation or information are expressly subject
// to the terms and conditions of the Altera Program License
// Subscription Agreement, Altera MegaCore Function License
// Agreement, or other applicable license agreement, including,
// without limitation, that your use is for the sole purpose of
// programming logic devices manufactured by Altera and sold by
// Altera or its authorized distributors.  Please refer to the
// applicable agreement for further details.


// Generated by Quartus II 32-bit Version 11.1 (Build Build 173 11/01/2011)
// Created on Thu Oct 27 12:39:21 2022

//  Module Declaration
module seven_seg
(
// {{ALTERA_ARGS_BEGIN}} DO NOT REMOVE THIS LINE!
seg_in, seg_out
// {{ALTERA_ARGS_END}} DO NOT REMOVE THIS LINE!
);
// Port Declaration

// {{ALTERA_IO_BEGIN}} DO NOT REMOVE THIS LINE!
input [3:0] seg_in;
output [6:0] seg_out;
// {{ALTERA_IO_END}} DO NOT REMOVE THIS LINE!



endmodule

```

- Add the necessary Verilog code to implement the design just before the endmodule statement.

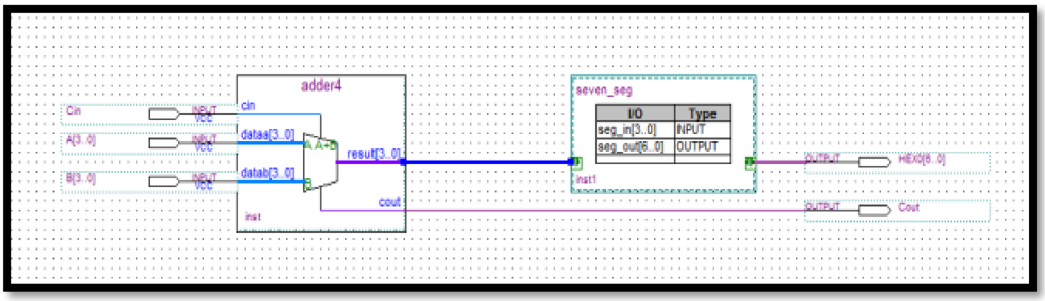

Text Editor - C:/Users/RFM_LAB8/Documents/TISHA/lab2 new/adder4_de2 - adder4_de2 - [seven_seg.v]*
File
Edit
View
Project
Processing
Tools
Window
Help

```verilog
//  Module Declaration
module seven_seg
(
// {{ALTERA_ARGS_BEGIN}} DO NOT REMOVE THIS LINE!
seg_in, seg_out
// {{ALTERA_ARGS_END}} DO NOT REMOVE THIS LINE!
);
// Port Declaration

// {{ALTERA_IO_BEGIN}} DO NOT REMOVE THIS LINE!
input [3:0] seg_in;
output [6:0] seg_out;
// {{ALTERA_IO_END}} DO NOT REMOVE THIS LINE!


reg [6:0] seg_out;
 always @*
 case (seg_in)
 0: seg_out = 7'b1000000;
 1: seg_out = 7'b1111001;
 2: seg_out = 7'b0100100;
 3: seg_out = 7'b0110000;
 4: seg_out = 7'b0011001;
 5: seg_out = 7'b0010010;
 6: seg_out = 7'b0000010;
 7: seg_out = 7'b1111000;
 8: seg_out = 7'b0000000;
 9: seg_out = 7'b0010000;
 10: seg_out = 7'b0001000;
 11: seg_out = 7'b0000011;
 12: seg_out = 7'b1000110;
 13: seg_out = 7'b0100001;
 14: seg_out = 7'b0000110;
 15: seg_out = 7'b0001110;
 default: seg_out = 7'b1111111;
 endcase

endmodule
```

- Save and close the file.

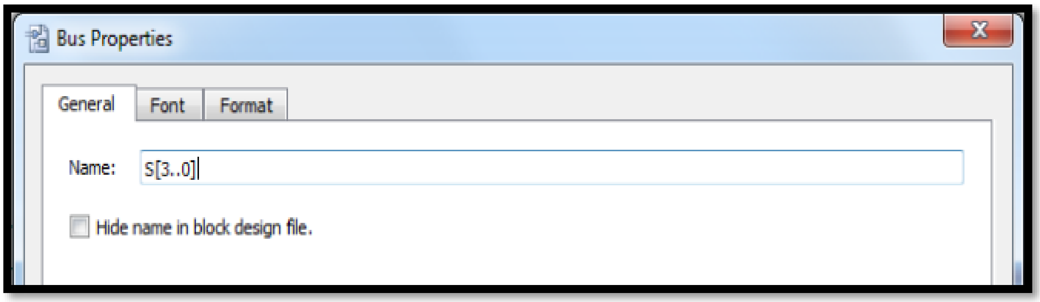
3) Connecting the Adder to the 7-Segment Decoder:

- Use the Orthogonal Bus Tool to connect the sum output of the adder to the input of the 7-segment decoder.



I/O	Type
seg_in[3..0]	INPUT
seg_out[6..0]	OUTPUT

- Right-click on the bus wire and select Properties. Name the bus as S[3..0].

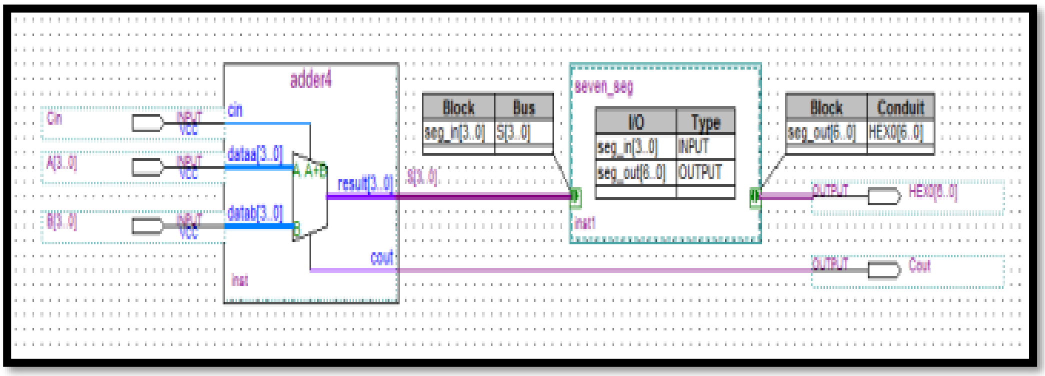


Name: S[3..0]

□ Hide name in block design file.

- Double-click on the mapper symbol and map seg_in[3..0] on the block to the S[3..0] bus.

- Complete the connection as illustrated in the provided figure.

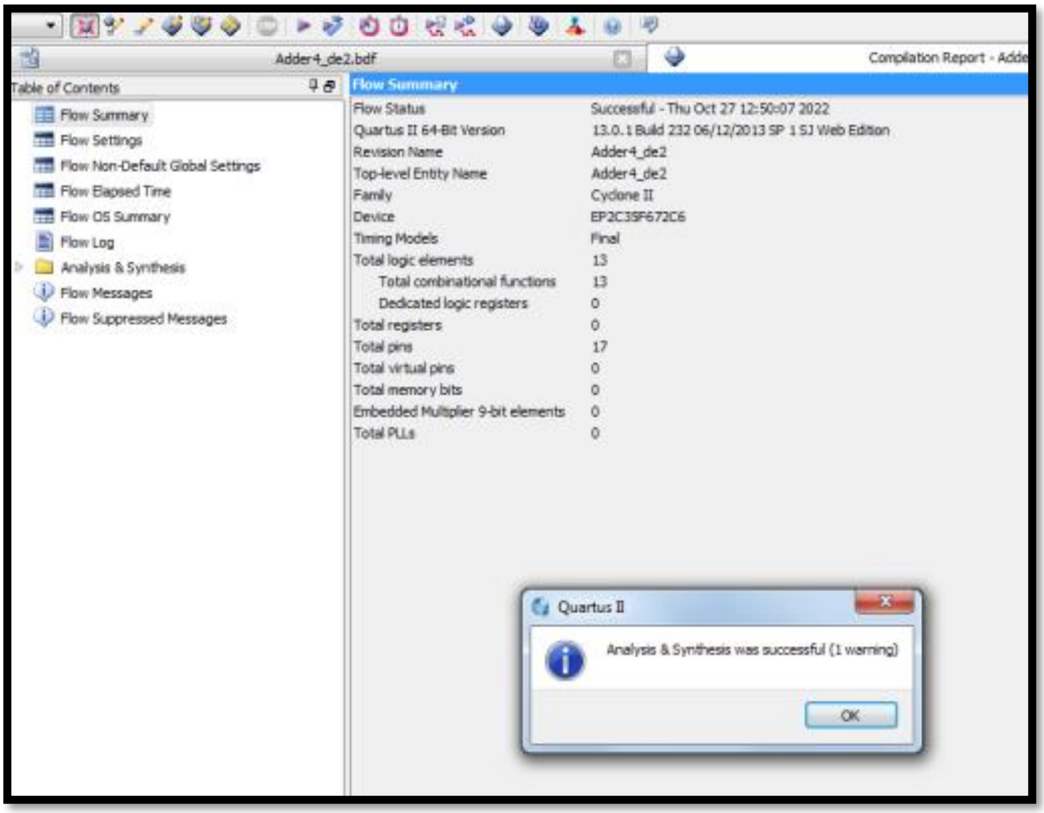


Block	Bus
seg_in[3..0]	S[3..0]

I/O	Type
seg_in[3..0]	INPUT
seg_out[6..0]	OUTPUT

Block	Conduit
seg_out[6..0]	HEX0[6..0]

- Perform simulation to verify the design's functionality.



Flow Summary

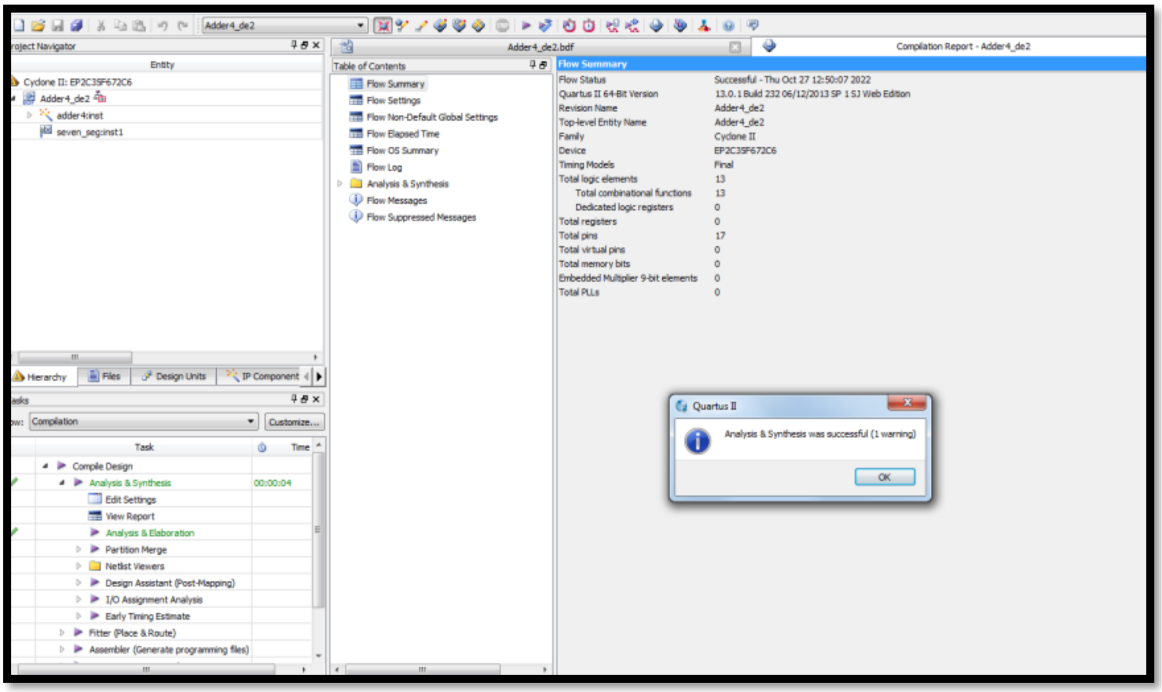
Flow Status	Successful - Thu Oct 27 12:50:07 2022
Quartus II 64-Bit Version	13.0.1 Build 232 06/12/2013 SP 1 SJ Web Edition
Revision Name	Adder4_de2
Top-level Entity Name	Adder4_de2
Family	Cyclone II
Device	EP2C35F672C6
Timing Models	Final
Total logic elements	13
Total combinational functions	13
Dedicated logic registers	0
Total registers	0
Total pins	17
Total virtual pins	0
Total memory bits	0
Embedded Multiplier 9-bit elements	0
Total PLLs	0

Analysis & Synthesis was successful (1 warning)

4. *Simulation Results*

1) Pin Assignment

- Start Analysis & Synthesis to check for errors, synthesize logic, and generate a project database.



Task	Time
Compile Design	
Analysis & Synthesis	00:00:04
Edit Settings	
View Report	
Analysis & Elaboration	
Partition Merge	
Netlist Viewers	
Design Assistant (Post-Mapping)	
I/O Assignment Analysis	
Early Timing Estimate	
Fitter (Place & Route)	
Assembler (Generate programming files)	

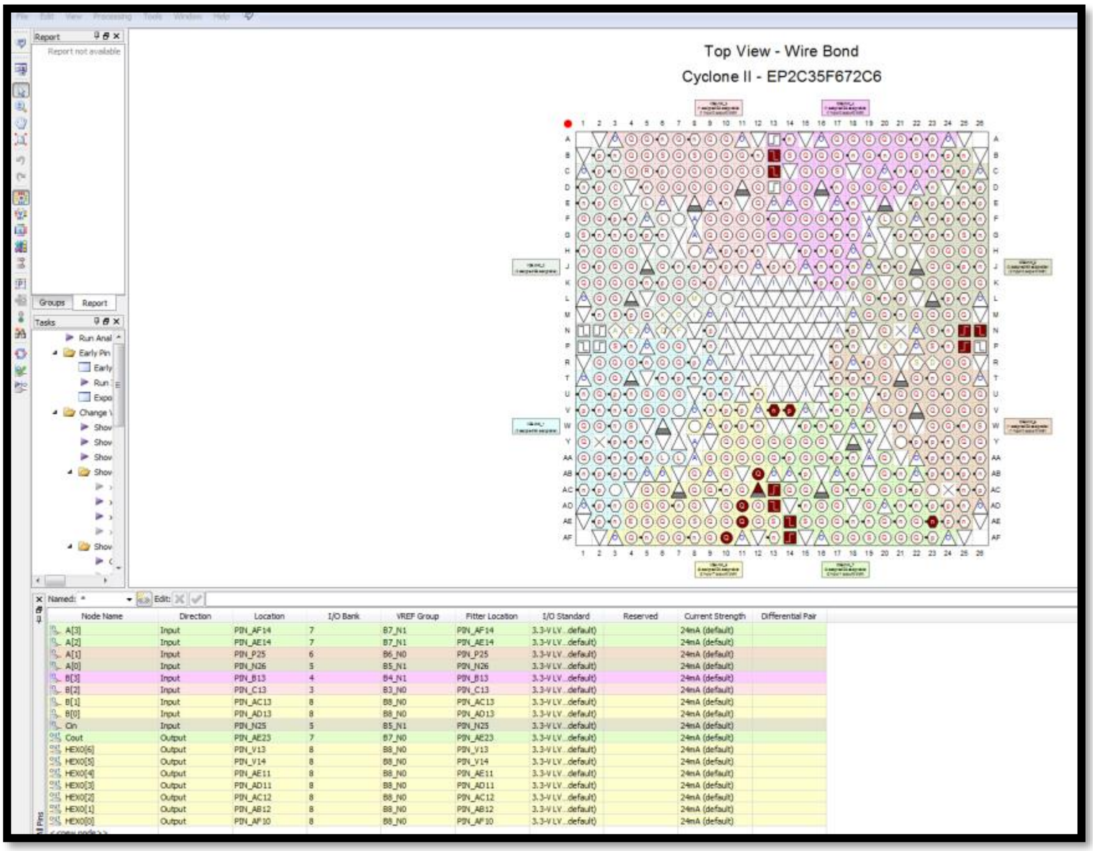
Flow Summary

Flow Status	Successful - Thu Oct 27 12:50:07 2022
Quartus II 64-Bit Version	13.0.1 Build 232 06/12/2013 SP 1 SJ Web Edition
Revision Name	Adder4_de2
Top-level Entity Name	Adder4_de2
Family	Cyclone II
Device	EP2C35F672C6
Timing Models	Final
Total logic elements	13
Total combinational functions	13
Dedicated logic registers	0
Total registers	0
Total pins	17
Total virtual pins	0
Total memory bits	0
Embedded Multiplier 9-bit elements	0
Total PLLs	0

Analysis & Synthesis was successful (1 warning)

- Access Pin Planner via Assignments > Pin Planner.

- View all pins in the Pin Planner.



Top View - Wire Bond

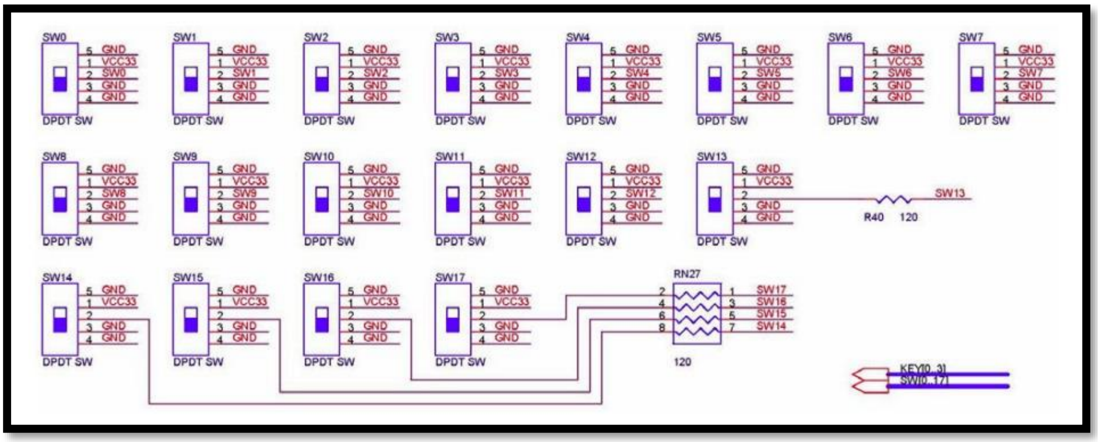
Cyclone II - EP2C35F672C6

Named: * Edit:

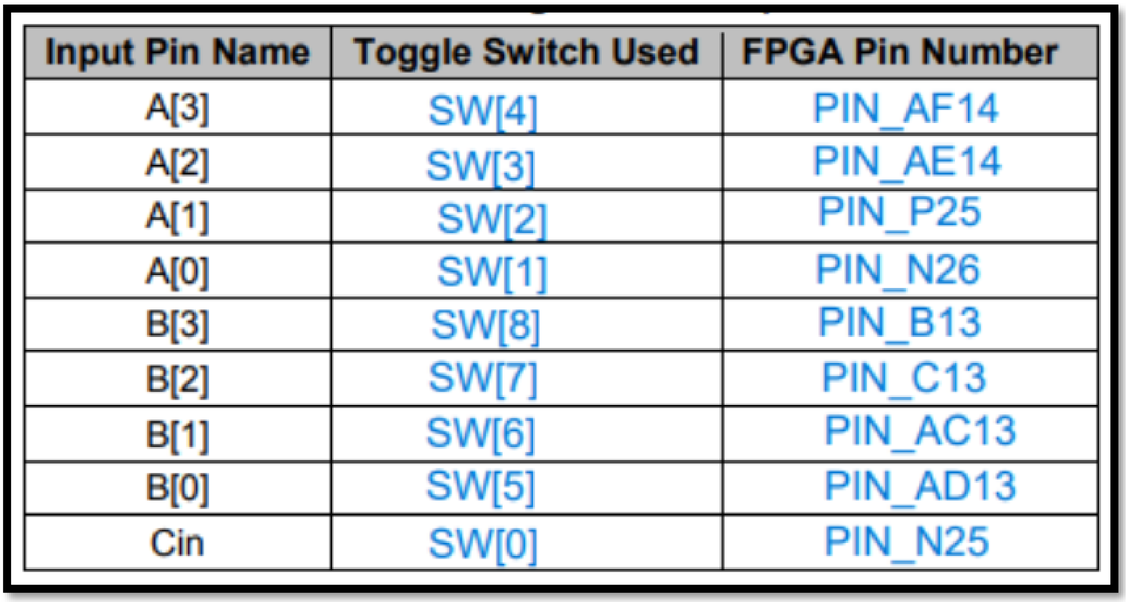
Node Name	Direction	Location	I/O Bank	VREF Group	Fitter Location	I/O Standard	Reserved	Current Strength	Differential Pair
A[3]	Input	PIN_AF14	7	B7_N1	PIN_AF14	3.3-V LV...default)		24mA (default)	
A[2]	Input	PIN_AE14	7	B7_N1	PIN_AE14	3.3-V LV...default)		24mA (default)	
A[1]	Input	PIN_P25	6	B6_N0	PIN_P25	3.3-V LV...default)		24mA (default)	
A[0]	Input	PIN_N26	5	B5_N1	PIN_N26	3.3-V LV...default)		24mA (default)	
B[3]	Input	PIN_B13	4	B4_N1	PIN_B13	3.3-V LV...default)		24mA (default)	
B[2]	Input	PIN_C13	3	B3_N0	PIN_C13	3.3-V LV...default)		24mA (default)	
B[1]	Input	PIN_AC13	8	B8_N0	PIN_AC13	3.3-V LV...default)		24mA (default)	
B[0]	Input	PIN_AD13	8	B8_N0	PIN_AD13	3.3-V LV...default)		24mA (default)	
Cin	Input	PIN_N25	5	B5_N1	PIN_N25	3.3-V LV...default)		24mA (default)	
Cout	Output	PIN_AE23	7	B7_N0	PIN_AE23	3.3-V LV...default)		24mA (default)	
HEX0[6]	Output	PIN_V13	8	B8_N0	PIN_V13	3.3-V LV...default)		24mA (default)	
HEX0[5]	Output	PIN_V14	8	B8_N0	PIN_V14	3.3-V LV...default)		24mA (default)	
HEX0[4]	Output	PIN_AE11	8	B8_N0	PIN_AE11	3.3-V LV...default)		24mA (default)	
HEX0[3]	Output	PIN_AD11	8	B8_N0	PIN_AD11	3.3-V LV...default)		24mA (default)	
HEX0[2]	Output	PIN_AC12	8	B8_N0	PIN_AC12	3.3-V LV...default)		24mA (default)	
HEX0[1]	Output	PIN_AB12	8	B8_N0	PIN_AB12	3.3-V LV...default)		24mA (default)	
HEX0[0]	Output	PIN_AF10	8	B8_N0	PIN_AF10	3.3-V LV...default)		24mA (default)	
<<new node>>									

- Double-click the Location column to select desired pin locations.

- Assign two 4-bit inputs and a 1-bit carry-in to toggle switches on the DE2 board.

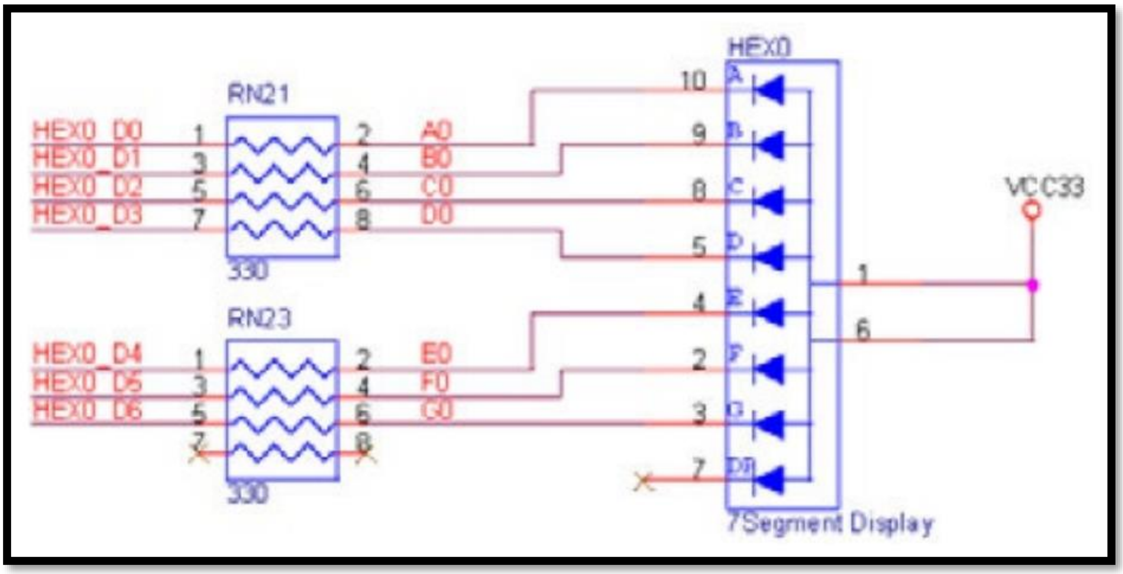


- Consult Appendix A for pin locations and assignments.

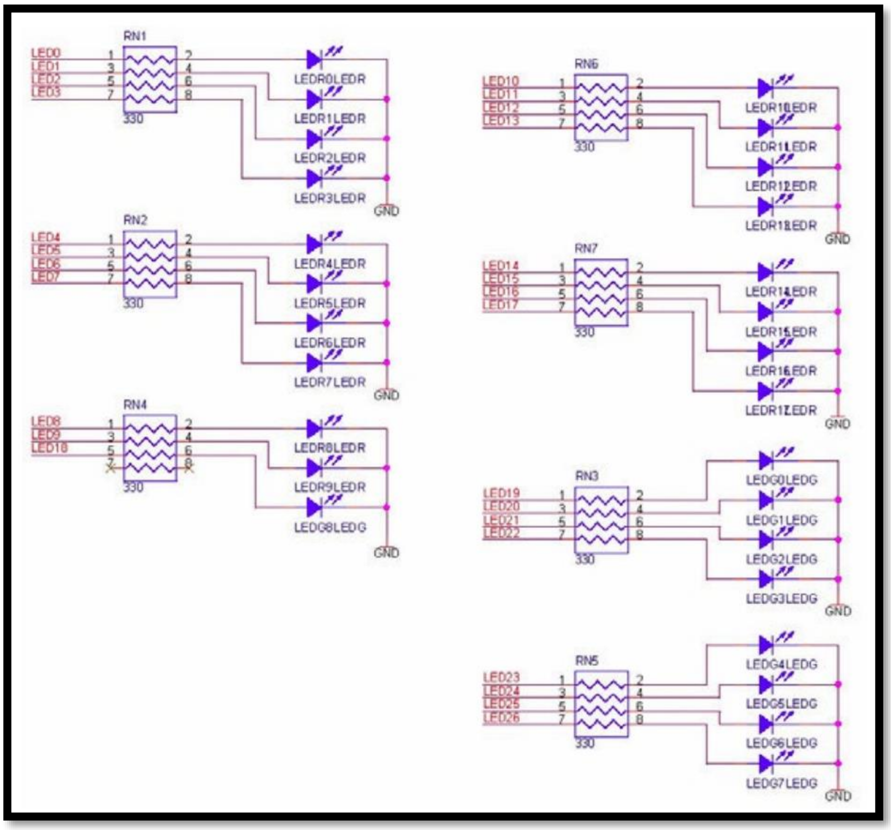


Input Pin Name	Toggle Switch Used	FPGA Pin Number
A[3]	SW[4]	PIN_AF14
A[2]	SW[3]	PIN_AE14
A[1]	SW[2]	PIN_P25
A[0]	SW[1]	PIN_N26
B[3]	SW[8]	PIN_B13
B[2]	SW[7]	PIN_C13
B[1]	SW[6]	PIN_AC13
B[0]	SW[5]	PIN_AD13
Cin	SW[0]	PIN_N25

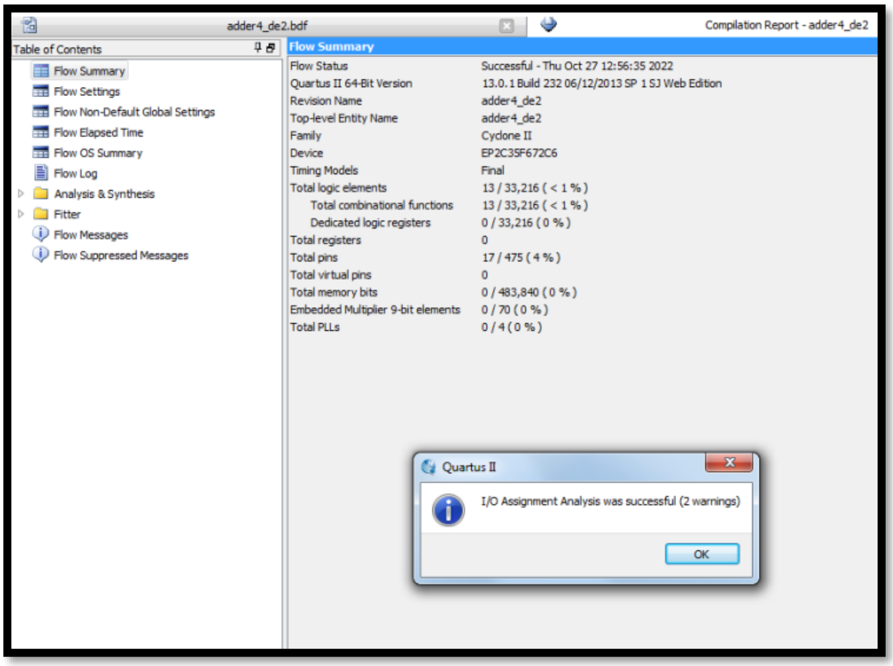
- Assign the output of the 7-segment decoder to one of the 7-segment displays on the DE2 board.



- Assign the carry-out signal to any of the LEDs on the DE2 board.



- Perform I/O Assignment Analysis to ensure the legality of pin assignment.

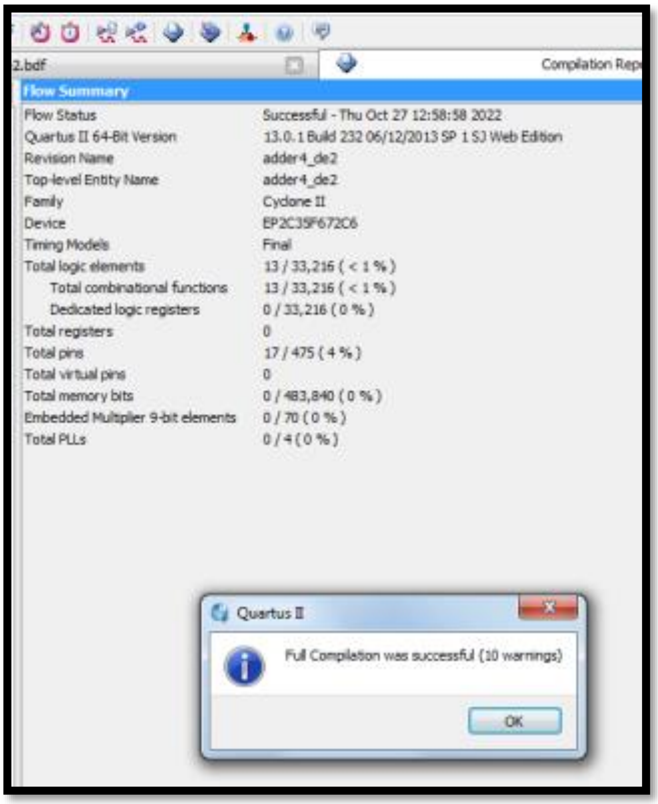


Flow Summary

Flow Status	Successful - Thu Oct 27 12:56:35 2022
Quartus II 64-Bit Version	13.0.1 Build 232 06/12/2013 SP 1 SJ Web Edition
Revision Name	adder4_de2
Top-level Entity Name	adder4_de2
Family	Cyclone II
Device	EP2C35F672C6
Timing Models	Final
Total logic elements	13 / 33,216 (< 1 %)
Total combinational functions	13 / 33,216 (< 1 %)
Dedicated logic registers	0 / 33,216 (0 %)
Total registers	0
Total pins	17 / 475 (4 %)
Total virtual pins	0
Total memory bits	0 / 483,840 (0 %)
Embedded Multiplier 9-bit elements	0 / 70 (0 %)
Total PLLs	0 / 4 (0 %)

I/O Assignment Analysis was successful (2 warnings)

- Compile the design.

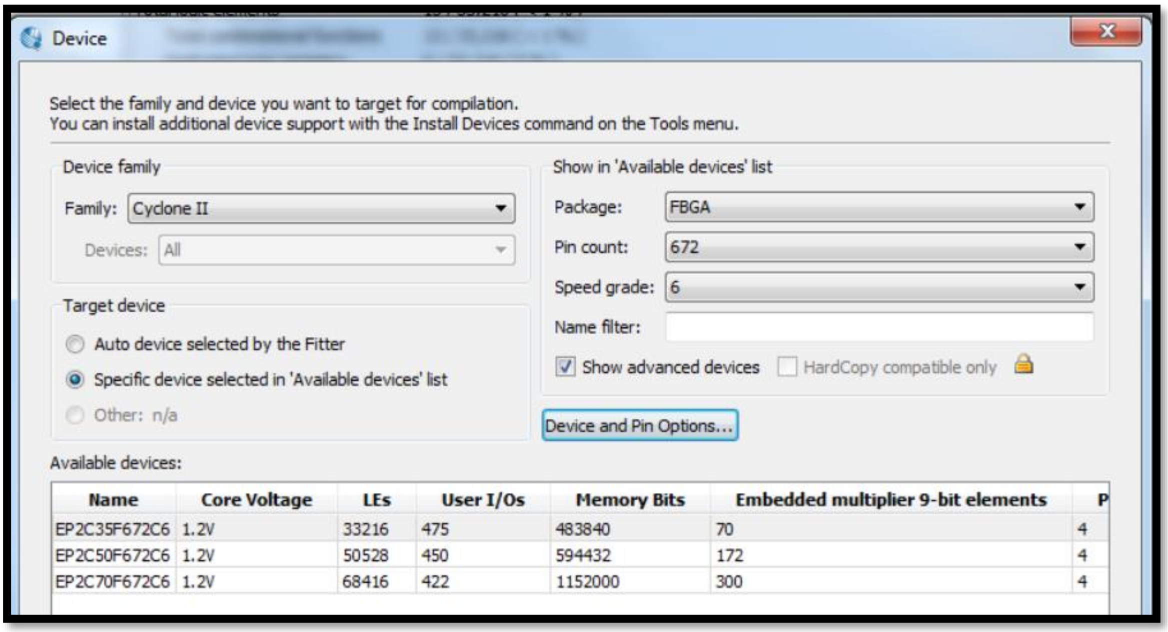


Flow Summary	
Flow Status	Successful - Thu Oct 27 12:58:58 2022
Quartus II 64-Bit Version	13.0.1 Build 232 06/12/2013 SP 1 SJ Web Edition
Revision Name	adder4_de2
Top-level Entity Name	adder4_de2
Family	Cyclone II
Device	EP2C35F672C6
Timing Models	Final
Total logic elements	13 / 33,216 (< 1 %)
Total combinational functions	13 / 33,216 (< 1 %)
Dedicated logic registers	0 / 33,216 (0 %)
Total registers	0
Total pins	17 / 475 (4 %)
Total virtual pins	0
Total memory bits	0 / 483,840 (0 %)
Embedded Multiplier 9-bit elements	0 / 70 (0 %)
Total PLLs	0 / 4 (0 %)

Quartus II

Full Compilation was successful (10 warnings)

- As a safety measure, configure all unused pins to act as input tri-stated to prevent potential short circuits.



Device

Select the family and device you want to target for compilation.
You can install additional device support with the Install Devices command on the Tools menu.

Device family

Family: Cyclone II

Devices: All

Target device

○ Auto device selected by the Fitter

● Specific device selected in 'Available devices' list

○ Other: n/a

Show in 'Available devices' list

Package: FBGA

Pin count: 672

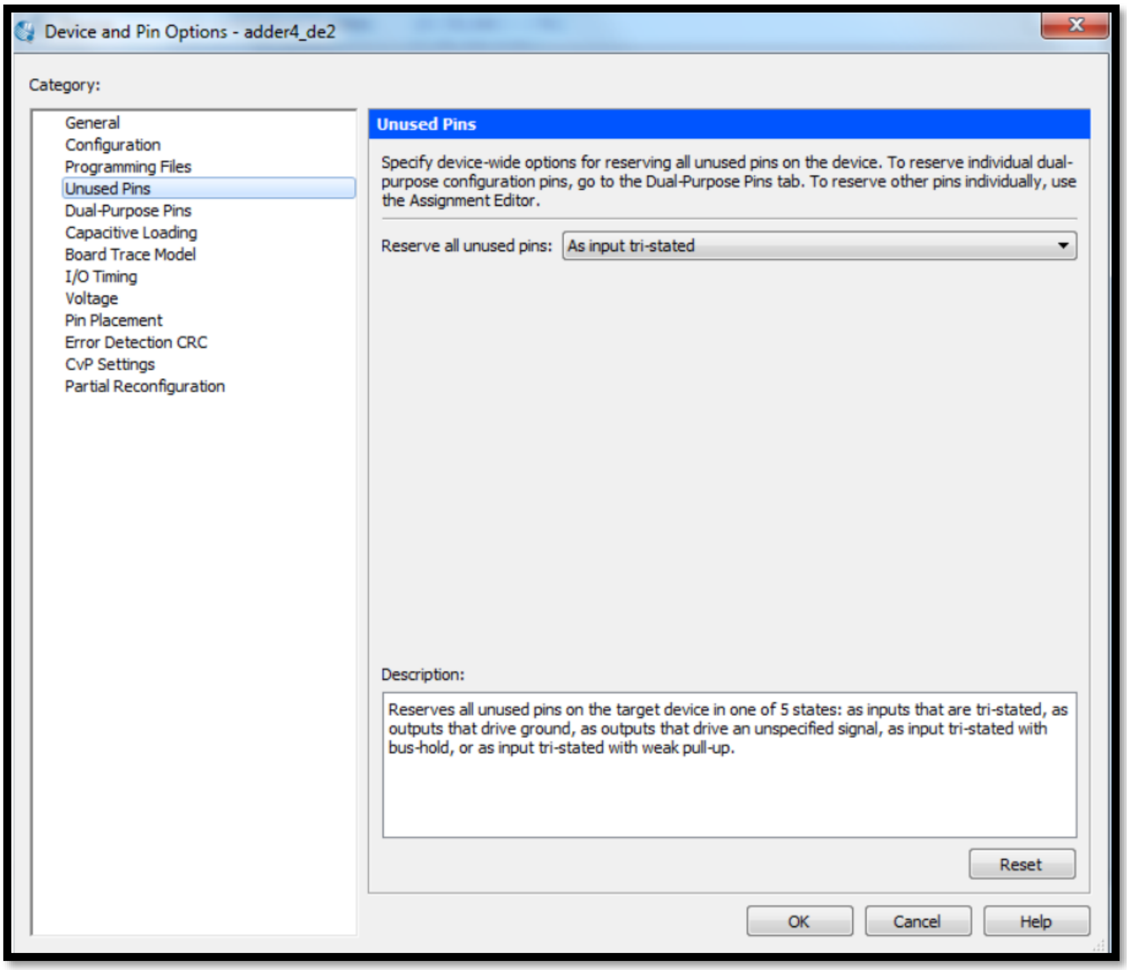
Speed grade: 6

Name filter:

☑ Show advanced devices □ HardCopy compatible only

Available devices:

Name	Core Voltage	LEs	User I/Os	Memory Bits	Embedded multiplier 9-bit elements	P
EP2C35F672C6	1.2V	33216	475	483840	70	4
EP2C50F672C6	1.2V	50528	450	594432	172	4
EP2C70F672C6	1.2V	68416	422	1152000	300	4



Specify device-wide options for reserving all unused pins on the device. To reserve individual dual-purpose configuration pins, go to the Dual-Purpose Pins tab. To reserve other pins individually, use the Assignment Editor.

Reserve all unused pins: As input tri-stated

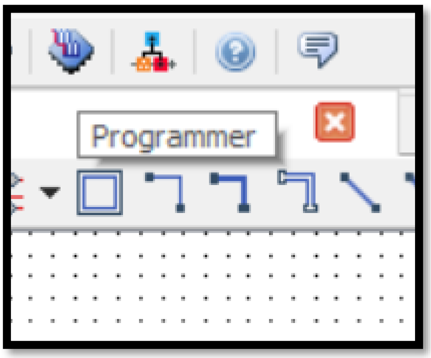
Description:

Reserves all unused pins on the target device in one of 5 states: as inputs that are tri-stated, as outputs that drive ground, as outputs that drive an unspecified signal, as input tri-stated with bus-hold, or as input tri-stated with weak pull-up.

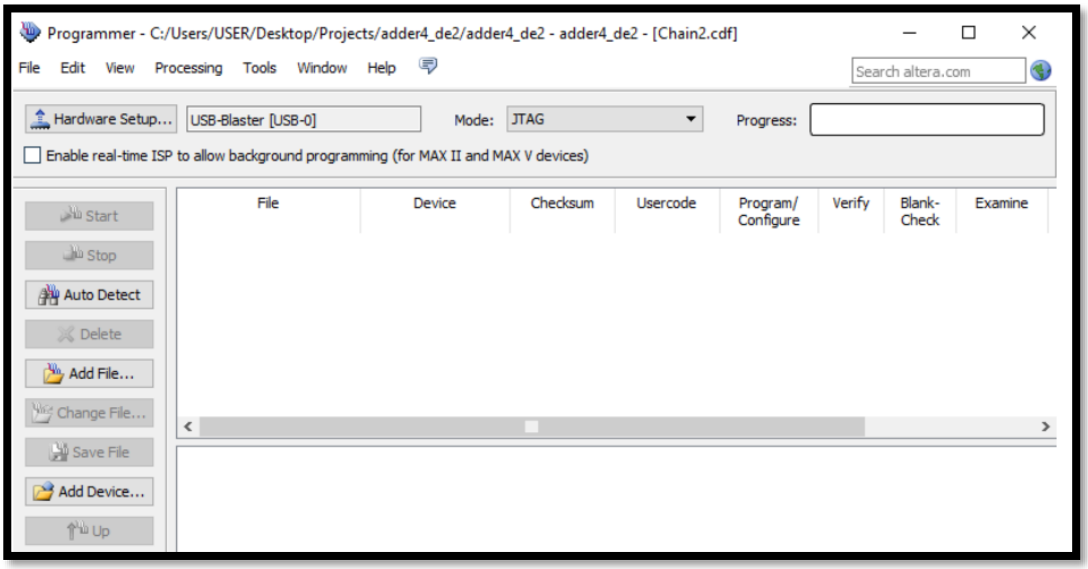
2) Program the Device

- Connect the USB cable from the DE2 board's USB-Blaster Port to the computer.

- Select the Programmer icon from the toolbar to open the Programmer window, showing the sof file.



- Set the Mode in the Programmer window to JTAG.



Programmer - C:/Users/USER/Desktop/Projects/adder4_de2/adder4_de2 - adder4_de2 - [Chain2.cdf]

USB-Blaster [USB-0]

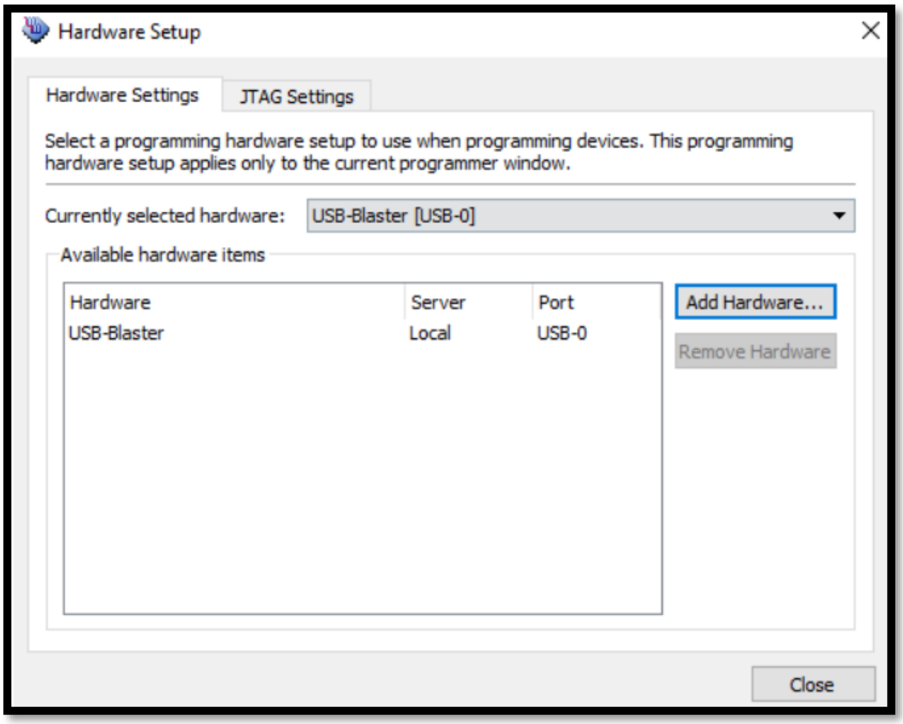
Mode: JTAG

Progress:

□ Enable real-time ISP to allow background programming (for MAX II and MAX V devices)

File	Device	Checksum	Usercode	Program/ Configure	Verify	Blank- Check	Examine

- If no hardware is detected, click the Hardware Setup button, and select USB-Blaster.



Hardware Setup

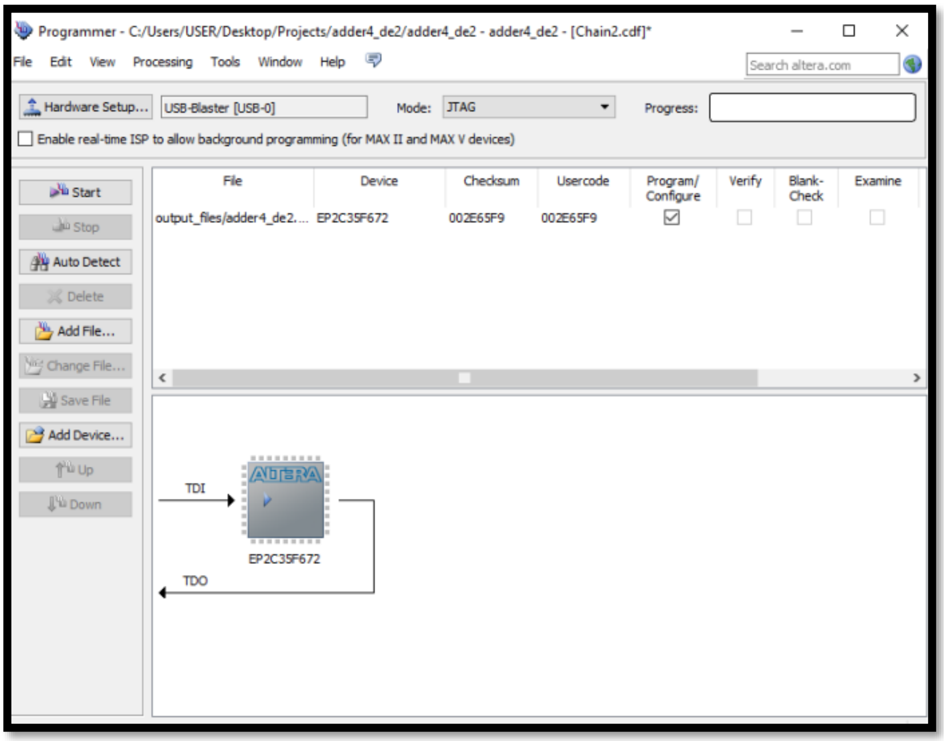
Select a programming hardware setup to use when programming devices. This programming hardware setup applies only to the current programmer window.

Currently selected hardware: USB-Blaster [USB-0]

Available hardware items

Hardware	Server	Port
USB-Blaster	Local	USB-0

- Mark the Program/Configure checkbox.



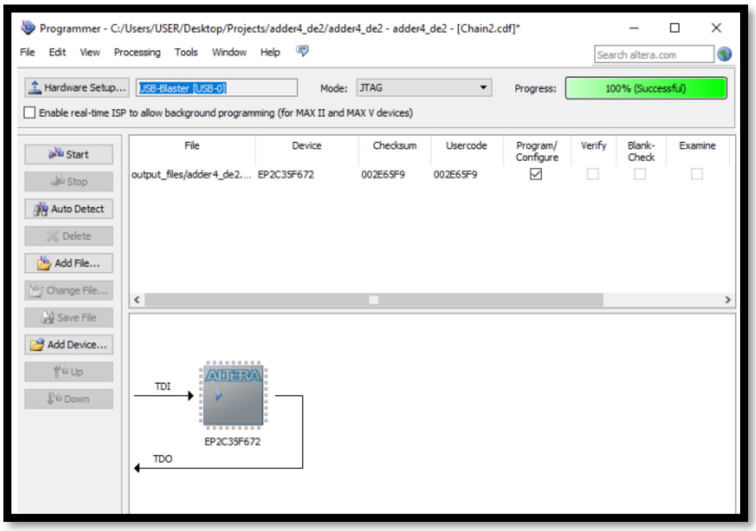
Programmer - C:/Users/USER/Desktop/Projects/adder4_de2/adder4_de2 - adder4_de2 - [Chain2.cdf]*

Hardware Setup... USB-Blaster [USB-0] Mode: JTAG Progress:

□ Enable real-time ISP to allow background programming (for MAX II and MAX V devices)

File	Device	Checksum	Usercode	Program/ Configure	Verify	Blank- Check	Examine
output_files/adder4_de2....	EP2C35F672	002E65F9	002E65F9	☑	□	□	□

- Click the Start button in the Programmer window to begin the configuration process.
- After receiving a confirmation message indicating completion, click OK.



Programmer - C:/Users/USER/Desktop/Projects/adder4_de2/adder4_de2 - adder4_de2 - [Chain2.cdf]*

Hardware Setup... USB-Blaster [USB-0]

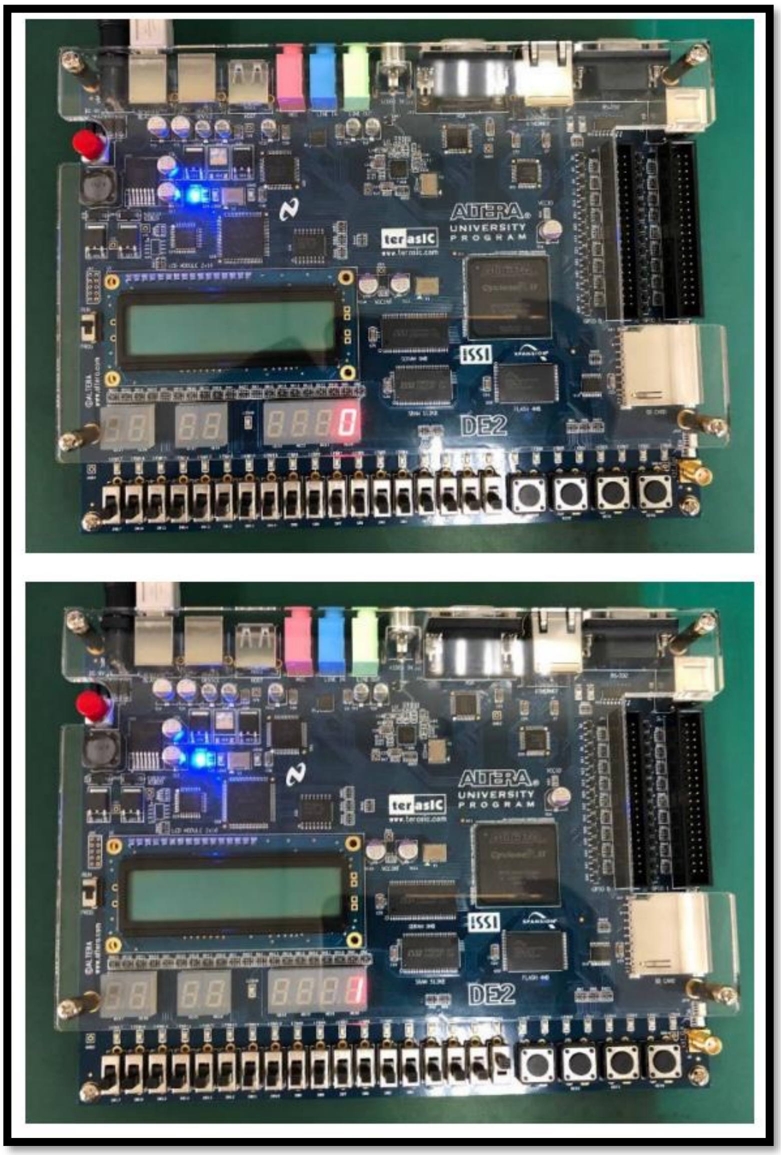
Mode: JTAG

Progress: 100% (Successful)

□ Enable real-time ISP to allow background programming (for MAX II and MAX V devices)

File	Device	Checksum	Usercode	Program/ Configure	Verify	Blank-Check	Examine
output_files/adder4_de2....	EP2C35F672	002E65F9	002E65F9	☑	□	□	□

- Verify the design by checking if the 7-segment display shows the correct sum for input values from toggle switches, and if the LED is lit for a carry-out signal.



5. *Analysis*

- **Blinking LED.**
  Design a binary counter to control the blinking frequency of an LED. Utilize the Megawizard Plug-In Manager to create the counter, ensuring appropriate clock inputs and output connections.

- **Running LEDs.**
  Implement unidirectional and bidirectional running LEDs using shift registers and counters. Design the logic to sequentially light up LEDs from left to right and vice versa.

- **Design process.**
  Create a 4-bit adder using the MegaWizard Plug-In Manager, establish connections between input and output ports. Develop a Verilog HDL design file for a 7-segment decoder, defining input/output ports and implementing functionality. Connect the adder output to the 7-segment decoder using the Orthogonal Bus Tool and verify the connection through simulation.

- **Pin assignment.**
  Use the Pin Planner to specify physical pin locations for inputs, outputs, and other signals. Configure unused pins as input tri-stated for safety.

- **Device programming.**
  Connect USB cable from the DE2 board to the computer, configure the FPGA using the Programmer window. Verify the design by checking if the 7-segment display shows the correct sum and if the LED indicates a carry-out signal.

6. *Conclusion*

This lab provided valuable insights into Altera FPGA design flow. By navigating the schematic design-entry environment, compiling, debugging, testing, simulating, and executing a simple circuit on the Altera FPGA DE2 board, we gained hands-on experience. The correct functionality of the design was confirmed by observing the output on the 7-segment display and LED. Overall, this lab enhanced our understanding of FPGA implementation and digital logic design.