

ECED2200 – Lab #3

STUDENT NAME(s): _____.

STUDENT NUMBER(s): B00 _____.

Pre-Lab Information

It is recommended that you read this entire lab ahead of time. Doing so will save you considerable time during the lab.

There is also several videos showing the use of the software tools – you will save considerable time by watching those now. See <http://colinoflynn.com/teaching/eced2200-intro-to-digital-circuits/> and look at the 'lab 3' information.

Overall Objective

This lab has four main objectives: implementing a design with the 7-segment display, using a K-map to realize a simple design, realizing the design with a mux, and realizing the design with a decoder. All parts will be completed on the physical board.

Deliverables

There is a fill-in-the-blanks lab report due. It will be handed out in class and also available on the website.

Part #1: Using 7-Segment Displays on Digital Explorer Boards

Objective

- Learn about 7-segment displays
- Familiarize yourself with downloading design to Digital Explorer boards

Required Materials

- Computer with Xilinx ISE 13.2 Webpack installed.
 - All computers in the lab have this installed.
 - This is free software so you can install on your own computer if you wish, you can download it from <http://www.xilinx.com/support/download/index.htm> - select '13.2' on the side. The file is very large so you may wish to download at school, and you are required to register to license it.



CC BY-SA

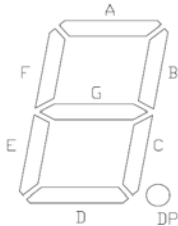
This work by [Colin O'Flynn](#) is licensed under a [Creative Commons Attribution-ShareAlike 3.0 Unported License](#) .

Digital Circuits – Lab #3

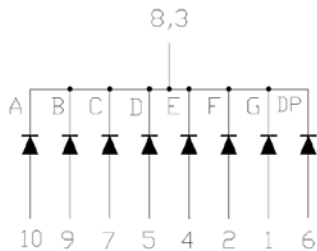
- Example project file DigitalTrainer_Simple.ZIP
 - This file contains an environment which is already setup for your lab.

Background

The 7-segment display is a simple display which can be used for displaying different numbers. Each of the 'segments' shown below is a LED:



The datasheet for example shows how these are connected as LEDs:



By driving a '1' into each LED, it will turn on the associated segment. Note on our board there are TWO 7-segment displays. You can select which one to turn on by driving the 'common' pin LOW to enable that display.

In order to use the display it would be convenient to have some logic which maps from normal binary numbers (e.g.: 0110 = 6) to the corresponding segments which must be illuminated.

We will call this part a BCD-to-7-segment decoder. Remember BCD is Binary Coded Decimal, in which 4 bits are used to represent the values from 0-9. For example '0010 1001' would be 41 in normal decimal, but 29 in BCD. BCD is convenient because we can feed each nibble (group of 4 bits) into a BCD-to-7-segment decoder to directly display a number.



CC BY-SA

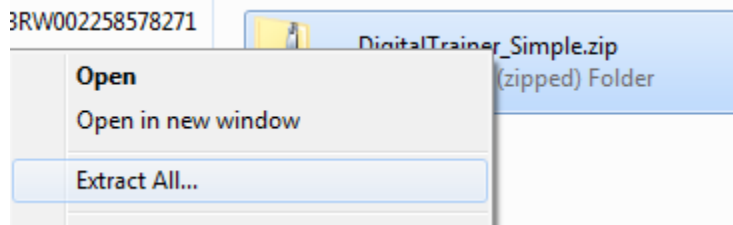
This work by [Colin O'Flynn](#) is licensed under a [Creative Commons Attribution-ShareAlike 3.0 Unported License](#).

Digital Circuits – Lab #3

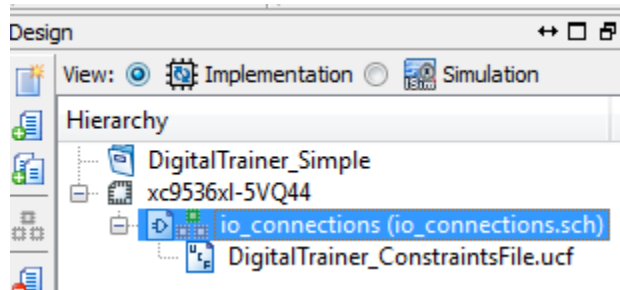
Procedure

NOTE: There is a video version of this procedure at <http://colinoflynn.com/teaching/eced2200-intro-to-digital-circuits/> which will be much easier to follow along.

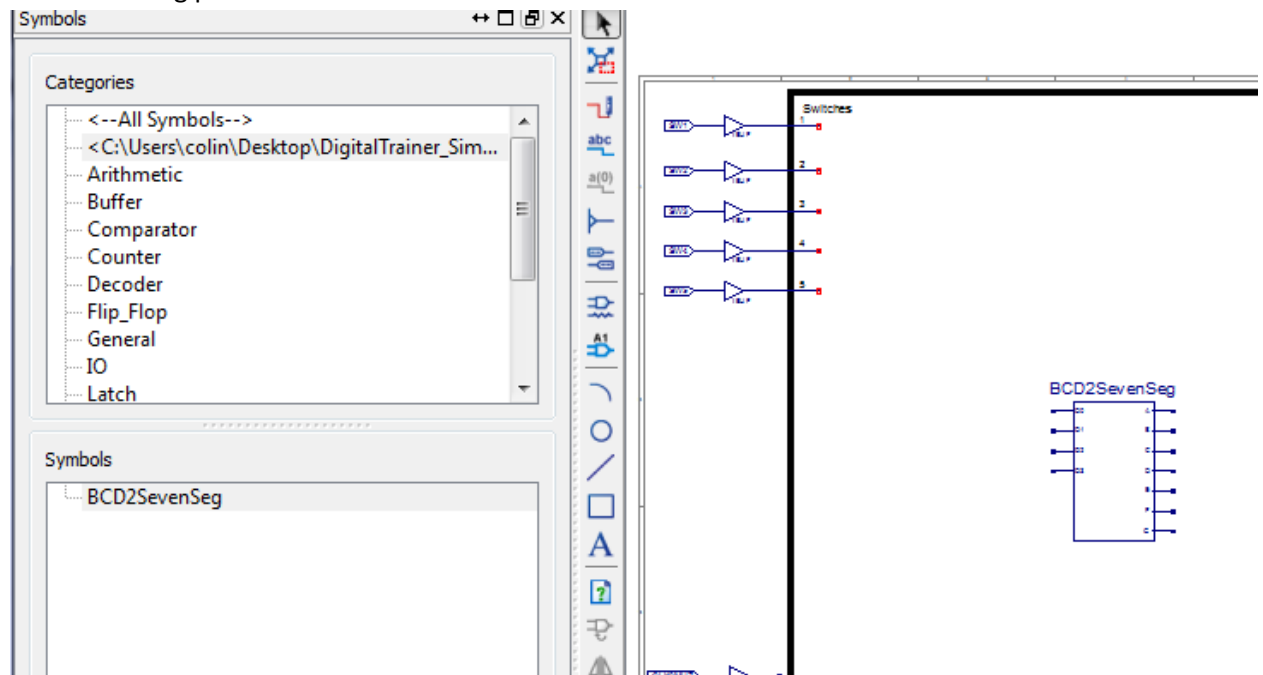
1. Unpack the given ZIP file somewhere (Right-Click on the .ZIP, hit “Extract All”):



2. Open the resulting folder (it may open automatically), double-click on “DigitalTrainer_Simple.xise” which will open the Xilinx Project Navigator
3. From the top left menu select “Implementation” instead of “Simulation” as the view:



4. Double-click on the ‘io_connections’ file.
5. Go to the ‘symbols’ tab in the window that opens, select the first category, then place a BCD2SevenSeg part in the middle of the schematic:

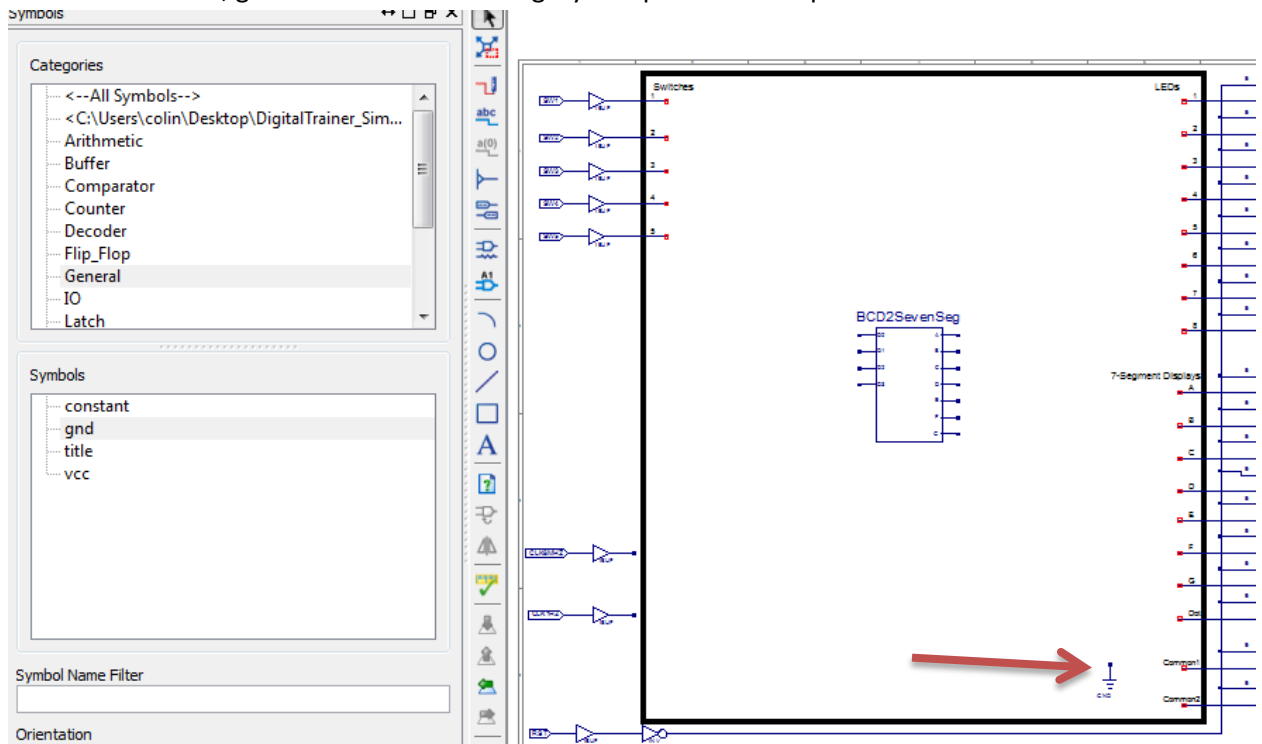


CC BY-SA

This work by [Colin O'Flynn](http://colinoflynn.com/) is licensed under a [Creative Commons Attribution-ShareAlike 3.0 Unported License](https://creativecommons.org/licenses/by-sa/3.0/).

Digital Circuits – Lab #3

6. Hit 'ESC' to cancel, go back to 'General' category and place a 'GND' part:



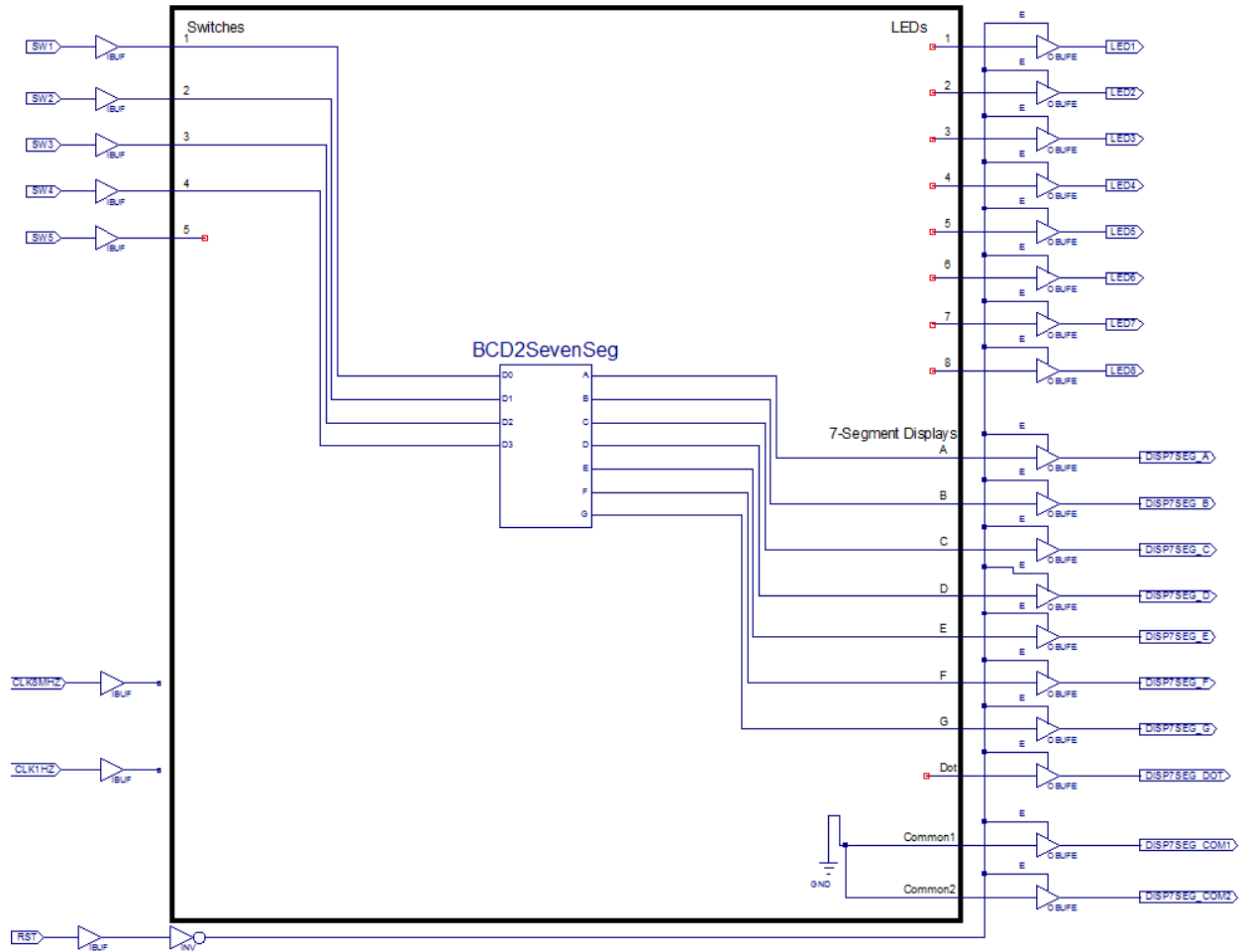
7. Connect the "Common1" and "Common2" of the 7-Segment display to ground. Connect the four inputs of the BCD2SevenSeg to the first four switches. Connect the seven output segments of the BCD2SevenSeg to the segments A through G. The following image shows the circuit you should have:



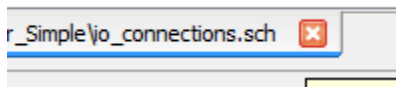
CC BY-SA

This work by [Colin O'Flynn](#) is licensed under a [Creative Commons Attribution-ShareAlike 3.0 Unported License](#).

Digital Circuits – Lab #3



8. Save the schematic. Close that file:



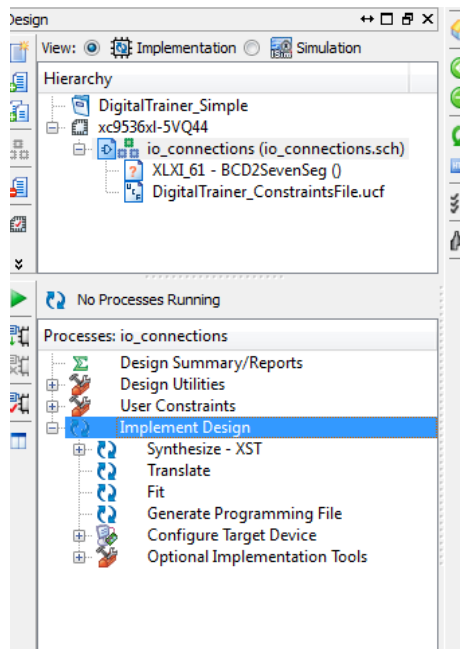
9. Ensure the 'Implementation' view is selected, and select file `io_connections`. Then double-click on 'Implement Design':



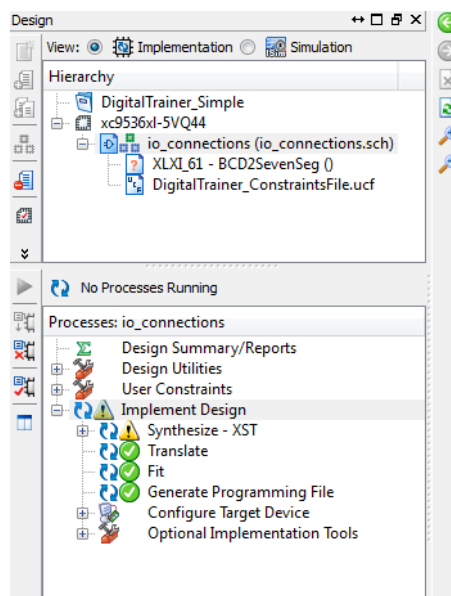
CC BY-SA

This work by [Colin O'Flynn](#) is licensed under a [Creative Commons Attribution-ShareAlike 3.0 Unported License](#).

Digital Circuits – Lab #3



The implementation phase should run. Afterwards your view will look like this, you must have a Green checkmark beside 'Generate Programming File':



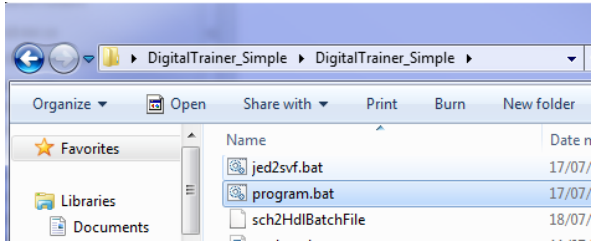
10. Plug in the Digital Explorer board. The power LED should light briefly and the Activity LED should blink on then off. **If the activity LED stays on unplug & replug the board.**
11. Open the folder you created in step 2, the same one with the .xise file in it. Find the file called either **program** or **program.bat** and double-click it:



CC BY-SA

This work by [Colin O'Flynn](#) is licensed under a [Creative Commons Attribution-ShareAlike 3.0 Unported License](#).

Digital Circuits – Lab #3



12. This will open a command-line window, and take a little bit (~30-60 seconds) to finish:

```
C:\Windows\system32\cmd.exe

'1': Erasing device...
'1': Erasure completed successfully.
Elapsed time = 0 sec.

'1': Programming device...
done.
'1': Putting device in ISP mode...done.
'1': Putting device in ISP mode...done.
'1': Programming completed successfully.
Elapsed time = 0 sec.
Could Not Find C:\Users\Colin\Desktop\DigitalTrainer_Simple\DigitalTrainer_Simple\jed2svf.bat
OPENDOUS reading command 0x01 failed (-116)
OPENDOUS reading command 0x02 failed (-116)
IR length: 8
Chain length: 1
Device Id: 01011001011000000010000010010011 <0x0000000059602093>
Manufacturer: Xilinx
Part(0): XC9536XL_UQ44
Stepping: 0
Filename: data/xilinx/xc9536x1_vq44/xc9536x1_vq44
```

At the end it will say “press any key to continue...”. If this happens immediately something is wrong, probably drivers were not loaded for the Digital Explorer board.

```
Manufacturer: Xilinx
Part(0): XC9536XL_UQ44
Stepping: 0
Filename: data/xilinx/xc9536x1_vq44/xc9536x1_vq44
Parsing 1950/1950 (100%)
Scanned device output matched expected TDO values.
'#Pause' is not recognized as an internal or external command,
operable program or batch file.
Press any key to continue . . .
```

During this time your ‘activity’ LED should be on as well.

13. Once it is downloaded, ensure the ‘CPLD Reset’ switch is set to ‘0’ (down towards bottom).
14. By varying the inputs, you should be able to make the 7-segment display show various numbers. Fill out the truth table in the observations section for all possible 16 inputs. I have shown you one line of the truth table to get you started.

Observations

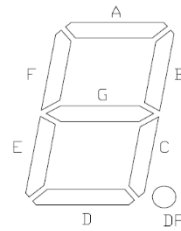


CC BY-SA

This work by [Colin O'Flynn](#) is licensed under a [Creative Commons Attribution-ShareAlike 3.0 Unported License](#).

Digital Circuits – Lab #3

BCD2SevenSegment Truth Table.



SW4	SW3	SW2	SW1	Decimal	7-Seg	SEGA	SEGB	SEGC	SEGD	SEGE	SEGF	SEGG
0	0	0	0									
0	0	0	1									
0	0	1	0									
0	0	1	1									
0	1	0	0	4		0	1	1	0	0	1	1
0	1	0	1									
0	1	1	0									
0	1	1	1									
1	0	0	0									
1	0	0	1									
1	0	1	0									
1	0	1	1									
1	1	0	0									
1	1	0	1									
1	1	1	0									
1	1	1	1									



CC BY-SA

This work by [Colin O'Flynn](#) is licensed under a [Creative Commons Attribution-ShareAlike 3.0 Unported License](#).

Part #2: Three Ways to Implement

Objective

- Learn about different ways of implementing a design with various components
- Familiarize yourself with downloading design to Digital Explorer boards

Required Materials

- Computer with Xilinx ISE 13.2 Webpack installed.
 - All computers in the lab have this installed.
 - This is free software so you can install on your own computer if you wish, you can download it from <http://www.xilinx.com/support/download/index.htm> - select '13.2' on the side. The file is very large so you may wish to download at school, and you are required to register to license it.
- Example project file DigitalTrainer_Simple.ZIP
 - This file contains an environment which is already setup for your lab.

Background

As shown in class (see slide-set: Mux and Demux) we can use both the Mux and Demux (Decoder) parts to design a logic block.

Procedure

1. The following is a truth table for the desired logic function, $Y(A,B,C) =$

A	B	C	Y
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

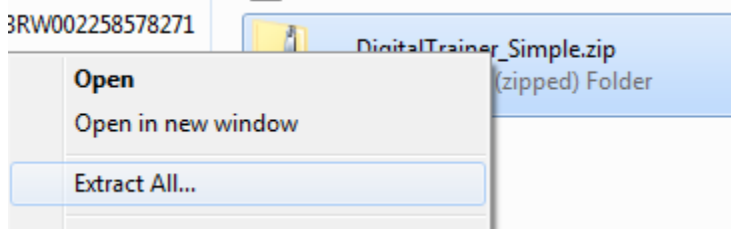
Part 2-A

2. Fill in the K-map in the 'observation' section, and form the simplified logic statements.
3. You will now implement those logic expressions. In the following examples I will be implementing the function $Y=A\cdot\bar{B}+A\cdot\bar{C}+B\cdot C$. **This is not the correct logic expression you should be implementing.** Instead determine the correct expression from the K-Map.
4. Steps 5-9 are as in the previous Part (Part I).

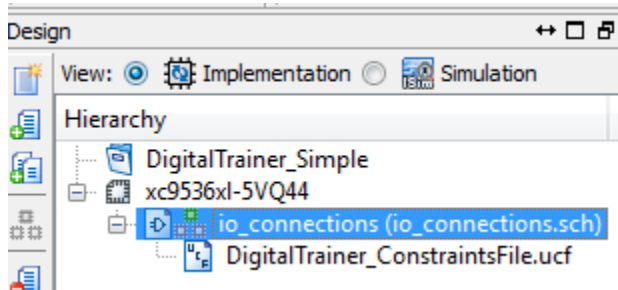


Digital Circuits – Lab #3

- Unpack the given ZIP file somewhere (Right-Click on the .ZIP, hit “Extract All”):

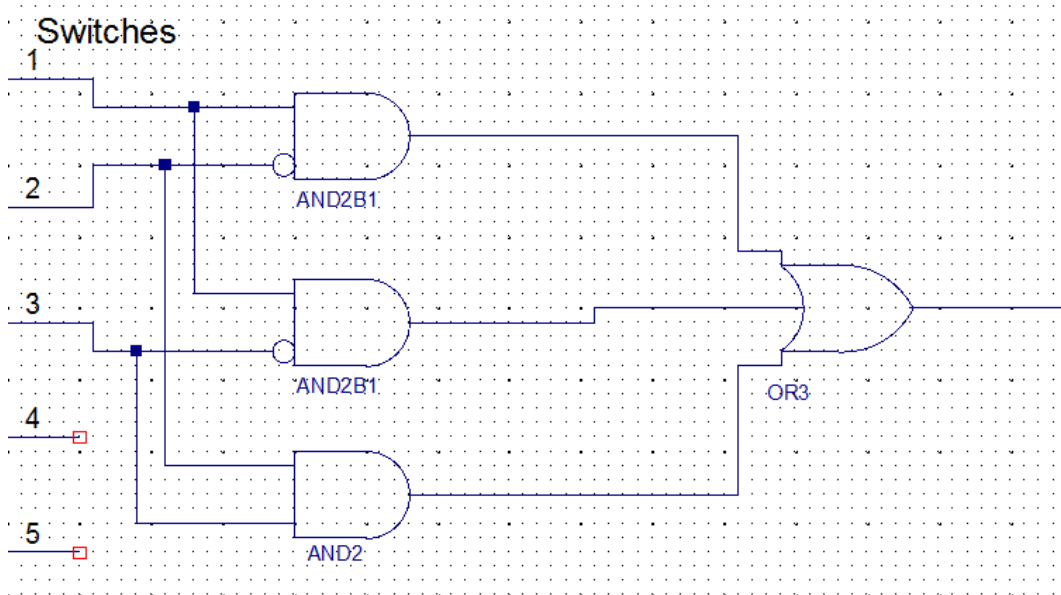


- Open the resulting folder (it may open automatically), double-click on “DigitalTrainer_Simple.xise” which will open the Xilinx Project Navigator
- From the top left menu select “Implementation” instead of “Simulation” as the view:



- Double-click on the ‘io_connections’ file.
- Draw your implemented schematic. You should use the gates such as ‘and2’, ‘and2b1’, ‘and2b2’. These gates contain zero, one, or two inputs which are complemented respectively, and are faster than placing ‘inv’ parts everywhere. Connect the output to LED1.

Again the following schematic is implementing $Y = A \cdot \bar{B} + A \cdot \bar{C} + B \cdot C$ and not the correct equation which you found, so your schematic will differ slightly.



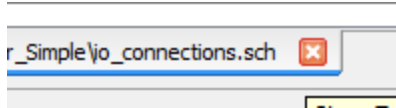
- Save the schematic. Close that file:



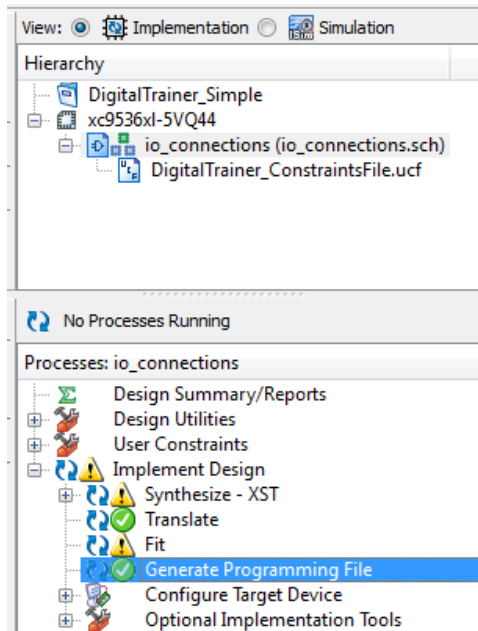
CC BY-SA

This work by [Colin O'Flynn](#) is licensed under a [Creative Commons Attribution-ShareAlike 3.0 Unported License](#).

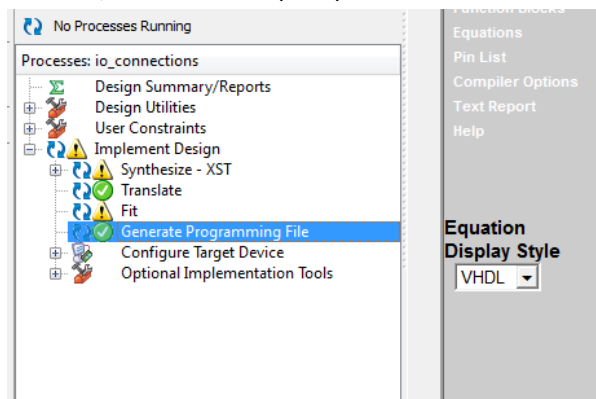
Digital Circuits – Lab #3



- Ensure the 'Implementation' view is selected, and select file `io_connections`. Then double-click on '**Implement Design**', just as in Part 1. The implementation phase should run. Afterwards your view will look like this, you must have a Green checkmark beside 'Generate Programming File':



- In the resulting view, copy the 'Pterms Used' output to the observations. In this example I've used 20/180 available (12%):



Software Version	O.61xd
Device Used	XC9536X
Date	7-18-201

RESOURCES SUM

Macrocells Used	Pterms Used	Registers Used
18/36 (50%)	20/180 (12%)	0/36 (0%)

PIN RESOURC

Signal Type	Required	Mapped	Pin
	4	4	7/10

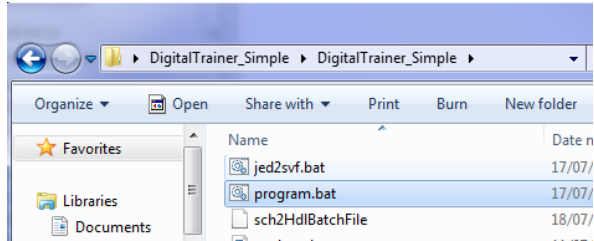
- Open the folder you created in step 2, the same one with the `.xise` file in it. Find the file called either **program** or **program.bat** and double-click it:



CC BY-SA

This work by [Colin O'Flynn](#) is licensed under a [Creative Commons Attribution-ShareAlike 3.0 Unported License](#).

Digital Circuits – Lab #3



14. This will open a command-line window, and take a little bit (~30-60 seconds) to finish:

```
C:\Windows\system32\cmd.exe

'1': Erasing device...
'1': Erasure completed successfully.
Elapsed time = 0 sec.

'1': Programming device...
done.
'1': Putting device in ISP mode...done.
'1': Programming completed successfully.
Elapsed time = 0 sec.
Could Not Find C:\Users\colin\Desktop\DigitalTrainer_Simple\DigitalTrainer_Simple\program.bat
OPENDOUS reading command 0x01 failed (-116)
OPENDOUS reading command 0x02 failed (-116)
IR length: 0
Chain length: 1
Device ID: 01011001011000000010000010010011 <0x0000000059602093>
Manufacturer: Xilinx
Part(0): KC9536XL_UQ44
Stepping: 0
Filename: data/xilinx/xc9536x1_uq44/xc9536x1_uq44
```

At the end it will say “press any key to continue...”. If this happens immediately something is wrong, probably drivers were not loaded for the Digital Explorer board.

```
Manufacturer: Xilinx
Part(0): KC9536XL_UQ44
Stepping: 0
Filename: data/xilinx/xc9536x1_uq44/xc9536x1_uq44
Parsing 1950/1950 (100%)
Scanned device output matched expected TDO values.
'#Pause' is not recognized as an internal or external command,
operable program or batch file.
Press any key to continue . . .
```

During this time your ‘activity’ LED should be on as well.

15. Once it is downloaded, ensure the ‘CPLD Reset’ switch is set to ‘0’ (down towards bottom).
16. Vary the 3 input switches (SW1, SW2, SW3) and record the output (LED1 which is Y) in the truth table.

Part 2-B

17. For this part, we will be using a Multiplexor (MUX) block to implement the same function as before. As discussed in class, we use the ‘select’ line inputs for the A/B/C, and the inputs to the MUX are tied either to ‘1’ or ‘0’ as required.
You will implement the truth table given earlier Y(A,B,C).
18. Setup your schematic as before. You can delete the logic from Part A if you wish, or reopen a new file.
19. Under the ‘Mux’ category place the ‘M8_1E’ part.
20. Place a ‘GND’ and ‘VCC’ part from the ‘General’ category.
21. Connect S0/S1/S2 to SW1/SW2/SW3
22. Connect the ‘E’ line of the M8_1E part to ‘1’ (e.g.: VCC).
23. Tie D0/D1/D2/D3...D6/D7 high (e.g.: to VCC) or low (e.g.: to GND) as required to implement the truth table. The following schematic shows the first three lines of the truth table implemented:

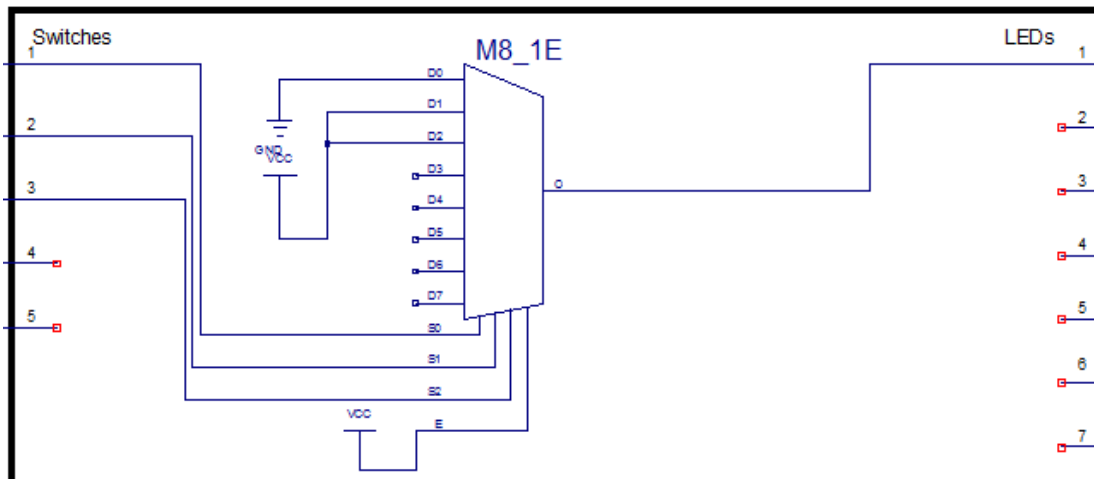


CC BY-SA

This work by [Colin O'Flynn](#) is licensed under a [Creative Commons Attribution-ShareAlike 3.0 Unported License](#).

Digital Circuits – Lab #3

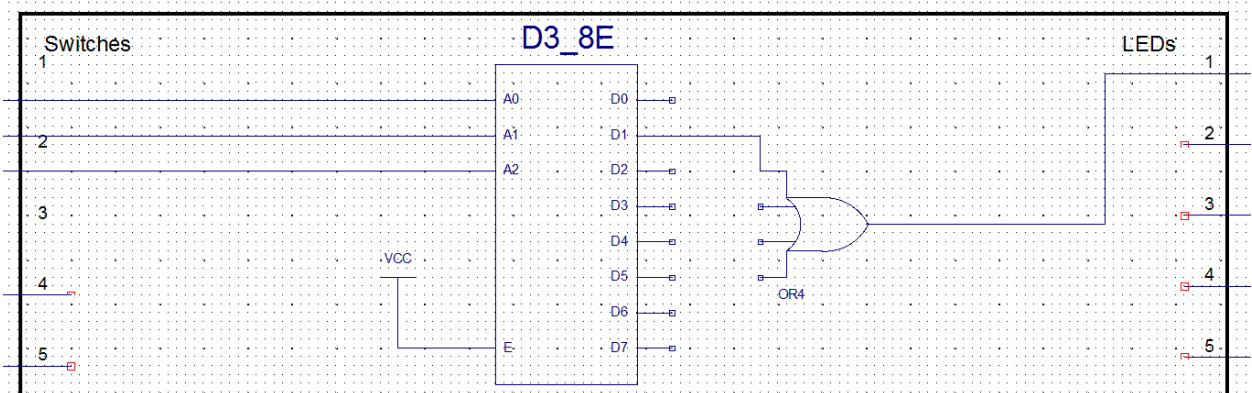
When ABC=000 then Y=0, When ABC=001 then Y=1, When ABC=010 then Y=1. You must decide on the value of D3/D4.../D6/D7 based on the truth table.



24. Repeat steps 10 – 16 of the procedure. Be sure to again record the number of PTerms used in the summary along with confirming the truth table is as expected.

Part 2-C

25. For this part we will be using the decoder to implement the same function as before. As discussed in class, we can OR all the outputs that result in a '1' for our required inputs.
26. Setup the schematic as before. You can delete your schematic from part B, or start a new file.
27. Under the 'decoder' category place a D3_8E (3:8) decoder.
28. Connect A0/A1/A2 to SW1/SW2/SW3. Connect the E pin to VCC. Use a suitably sized OR gate to connect up the required outputs. Remember you connect an output that will be '1' for each '1' in the truth table. The example below shows the connection for ABC=001, Y=1.



29. Repeat steps 10 – 16 of the procedure. Be sure to again record the number of PTerms used in the summary along with confirming the truth table is as expected.



CC BY-SA

This work by [Colin O'Flynn](#) is licensed under a [Creative Commons Attribution-ShareAlike 3.0 Unported License](#).

Digital Circuits – Lab #3

Observations

Part 2A

1. K-Map of desired truth table:

		A B			
		0 0	0 1	1 1	1 0
C	0				
	1				

2. The simplified express from the K-Map is:

Y =

3. The resulting truth table from implementing that design is:

A	B	C	Y
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	
1	1	1	

PTerms Used = _____

Part 2-B

Truth Table:

A	B	C	Y
0	0	0	
0	0	1	
0	1	0	
0	1	1	



CC BY-SA

This work by [Colin O'Flynn](#) is licensed under a [Creative Commons Attribution-ShareAlike 3.0 Unported License](#).

Digital Circuits – Lab #3

1	0	0	
1	0	1	
1	1	0	
1	1	1	

PTerms Used = _____

Part 2-C

Truth Table:

A	B	C	Y
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

PTerms Used = _____



CC BY-SA

This work by [Colin O'Flynn](#) is licensed under a [Creative Commons Attribution-ShareAlike 3.0 Unported License](#).