

Logic Design 2DI4

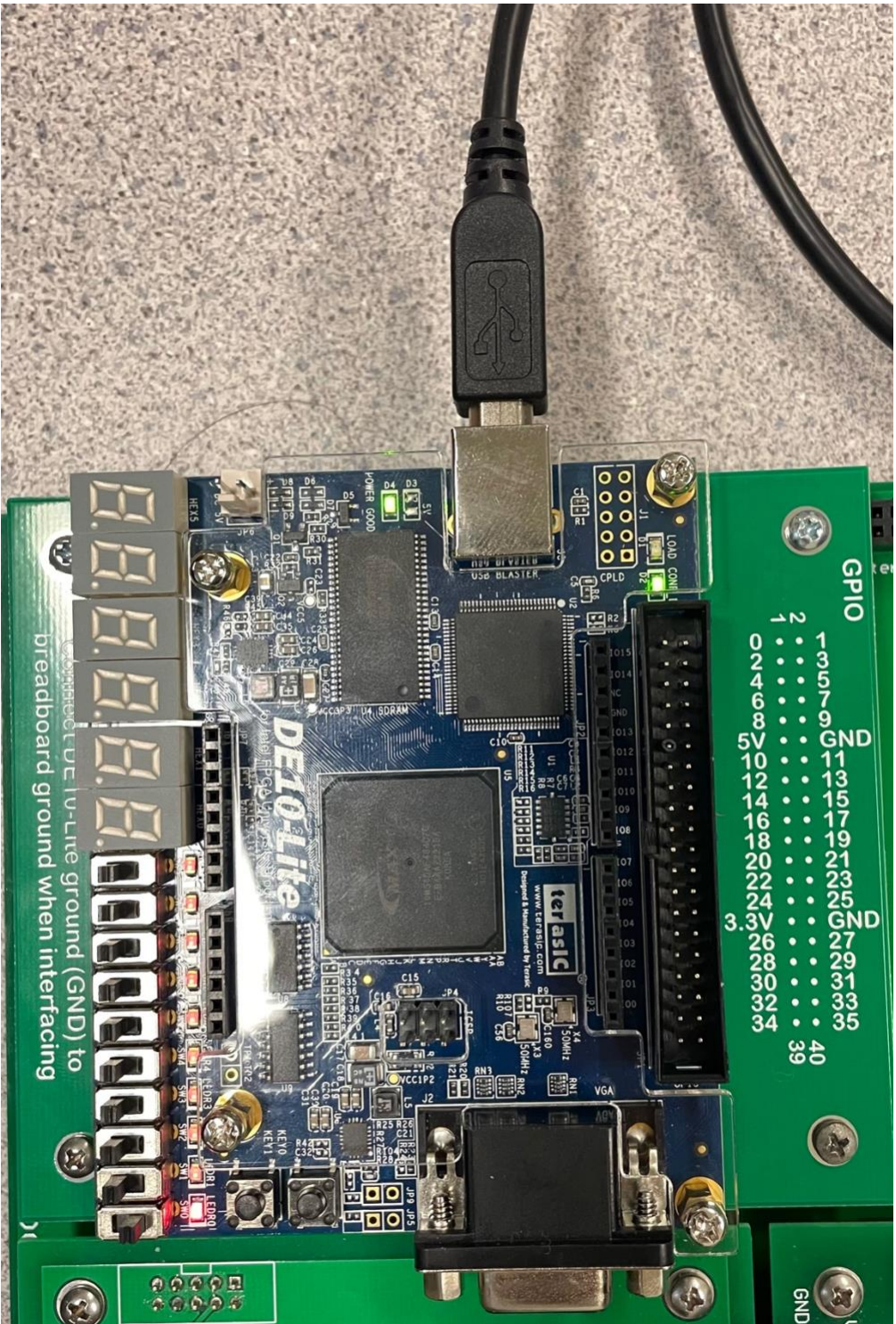
Lab 3

Instructor: Dr. Bauman/Doyle
Lab TAs: Yifei Zhang (zhany217), Nfise
gharbankhani (ghorbn2),
Gaurav.P (prajupg), Connor Rigotta (ricotta)

Mustafa Shahid– L08 – shahim45 - 400440384
Abaan Khan – L08 – khana454-400428399

As a future member of the engineering profession, the student is responsible for performing the required work in an honest manner, without plagiarism and cheating. Submitting this work with my name and student number is a statement and understanding that this work is my own and adheres to the Academic Integrity Policy of McMaster University and the Code of Conduct of the Professional Engineers of Ontario. Submitted by [**Mustafa Shahid, shahim45, 400440384**]

As a future member of the engineering profession, the student is responsible for performing the required work in an honest manner, without plagiarism and cheating. Submitting this work with my name and student number is a statement and understanding that this work is my own and adheres to the Academic Integrity Policy of McMaster University and the Code of Conduct of the Professional Engineers of Ontario. Submitted by [**Abaan Khan, khana454, 400428399**]



breadboard ground when interfacing

GPIO
1 3 5 7 9 GND
2 4 6 8 10 11 13 15 17 19 21 23 25 GND
0 2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32 34
3.3V 27 29 31 33 35
40 39

3.5.2 “Programming” 3-to-8 Line Decoder in physical ICs.

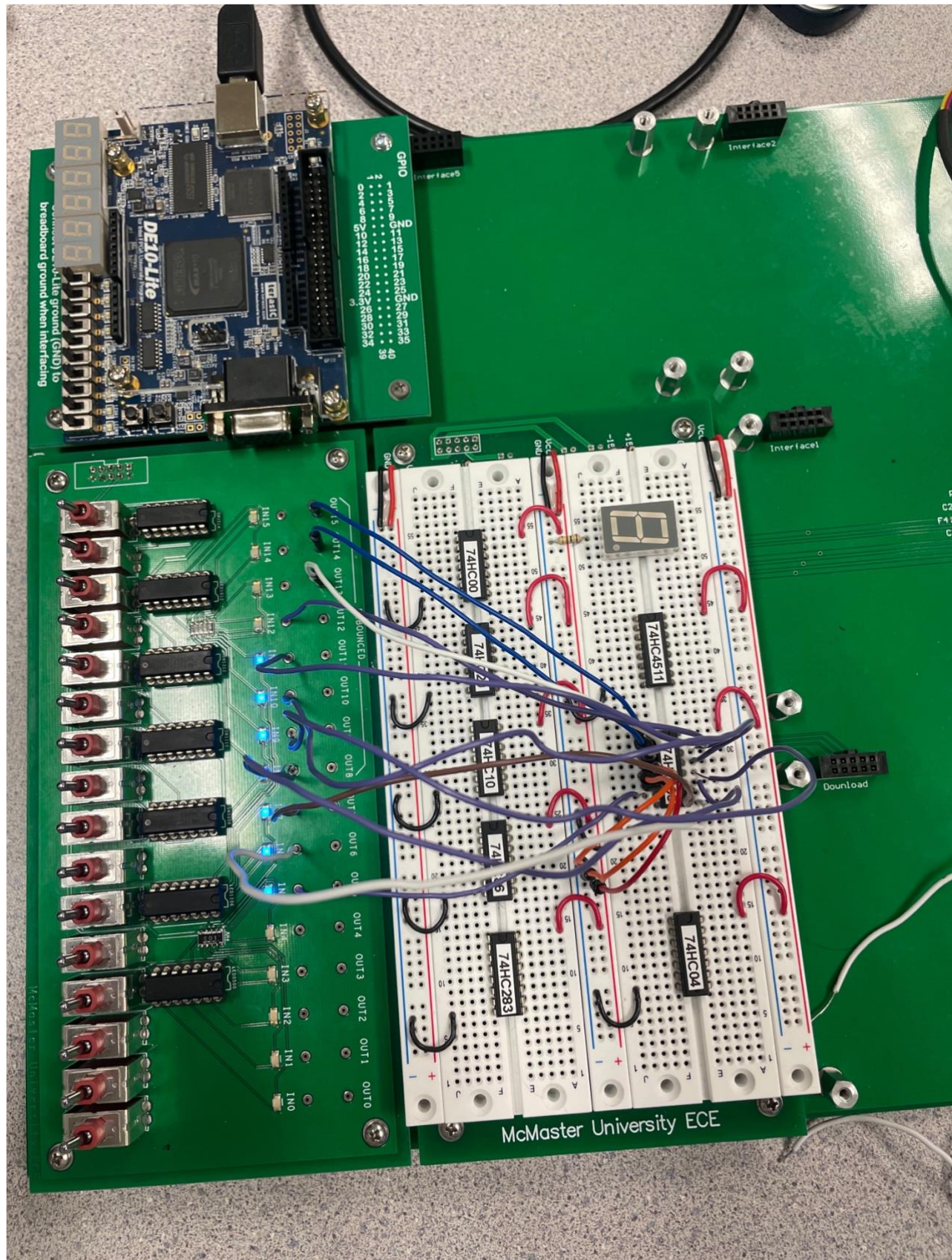
1. Verify the function table for the sn74HC138 is a 3-to-8 line decoder from Table 4.6 of your textbook (Truth Table of a Three-to-Eight-Line Decoder). Use switches for X, Y, Z selects and LEDs for the outputs. Refer to the data sheet for handling the enable pins.

Table 4.6
Truth Table of a Three-to-Eight-Line Decoder

Inputs			Outputs							
<i>x</i>	<i>y</i>	<i>z</i>	<i>D</i> ₀	<i>D</i> ₁	<i>D</i> ₂	<i>D</i> ₃	<i>D</i> ₄	<i>D</i> ₅	<i>D</i> ₆	<i>D</i> ₇
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

Figure 1: Truth Table of a 3-8 decoder.

Image of our circuit we built in lab :



The above circuit verifies the results in the truth table taken from the textbook.

2. Referring to your textbook, find the truth table and logic expression for a full adder. “Program” this full-adder circuit such that F1 represents the sum (S) and F2 represents the carry (C). Mark every connected row-column intersection with an X. Build the circuit and verify operation.

Milestone 2

Answer:

Table 4.4
Full Adder

<i>x</i>	<i>y</i>	<i>z</i>	C	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Figure 2: Truth table for a full adder.

We have:

$$S = \text{Sum}(m_1, m_2, m_4, m_7)$$

$$C = \text{Sum}(m_3, m_5, m_6, m_7)$$

Therefore, the final circuit using 3:8 decoder:

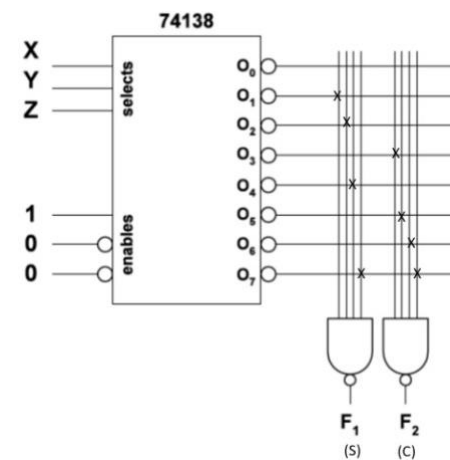
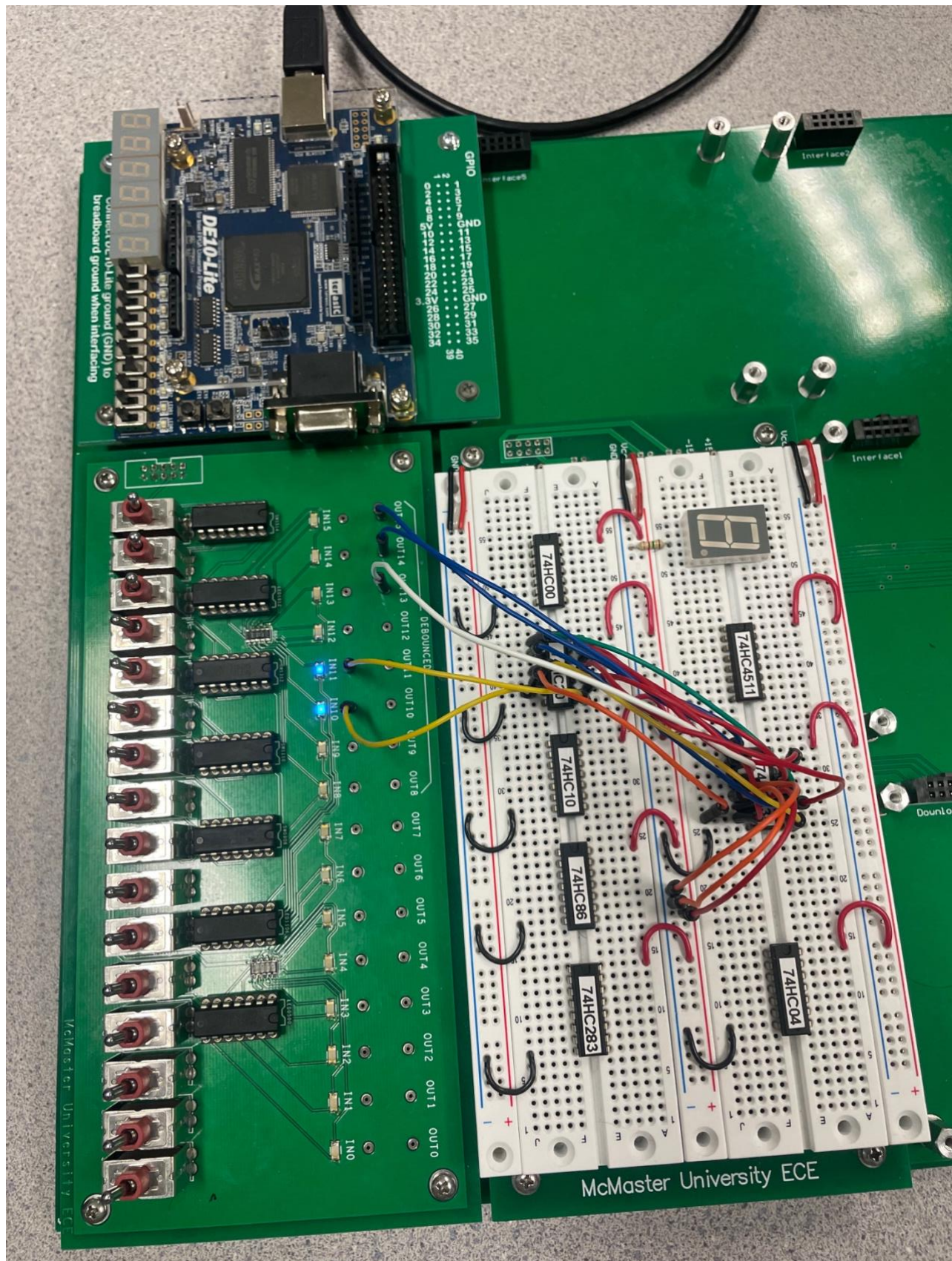


Figure 3: Full Adder circuit using 3:8 decoder.

Image of our circuit we built in lab:



The above circuit verifies the results of the truth table taken from the textbook.

3.5.3 3-to-8 Line Decoder in HDL

Milestone 3

Refer to your textbook for the design of a 3-to-8 Line Decoder (input: 3-bits binary code. output: 1 of 8 bits). For reference, a 3-to-8 line decoder schematic can be found in Mano Figure 4.18 (Figure 4.19 is a 2-to-4 line decoder with enable). HDL Example 4.1 is also a good starting point. Using Quartus II define a 3-to-8 Line Decoder, compile your model and load it on to the CPLD. For inputs connect wires from toggle switches to CPLD inputs. For outputs connect CPLD outputs to LEDs. For your report, include a screen capture of your modelling – ensure all team member names and student numbers appear in the comments of the Verilog model text.

Answer:

The Verilog model needs to be made based on the circuit shown below:

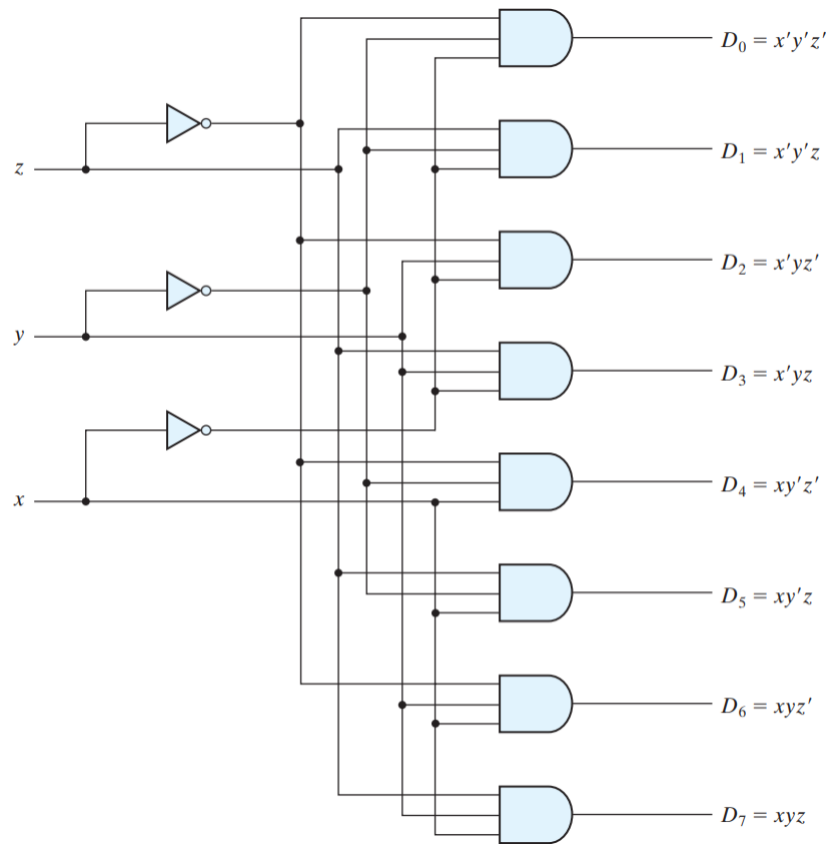
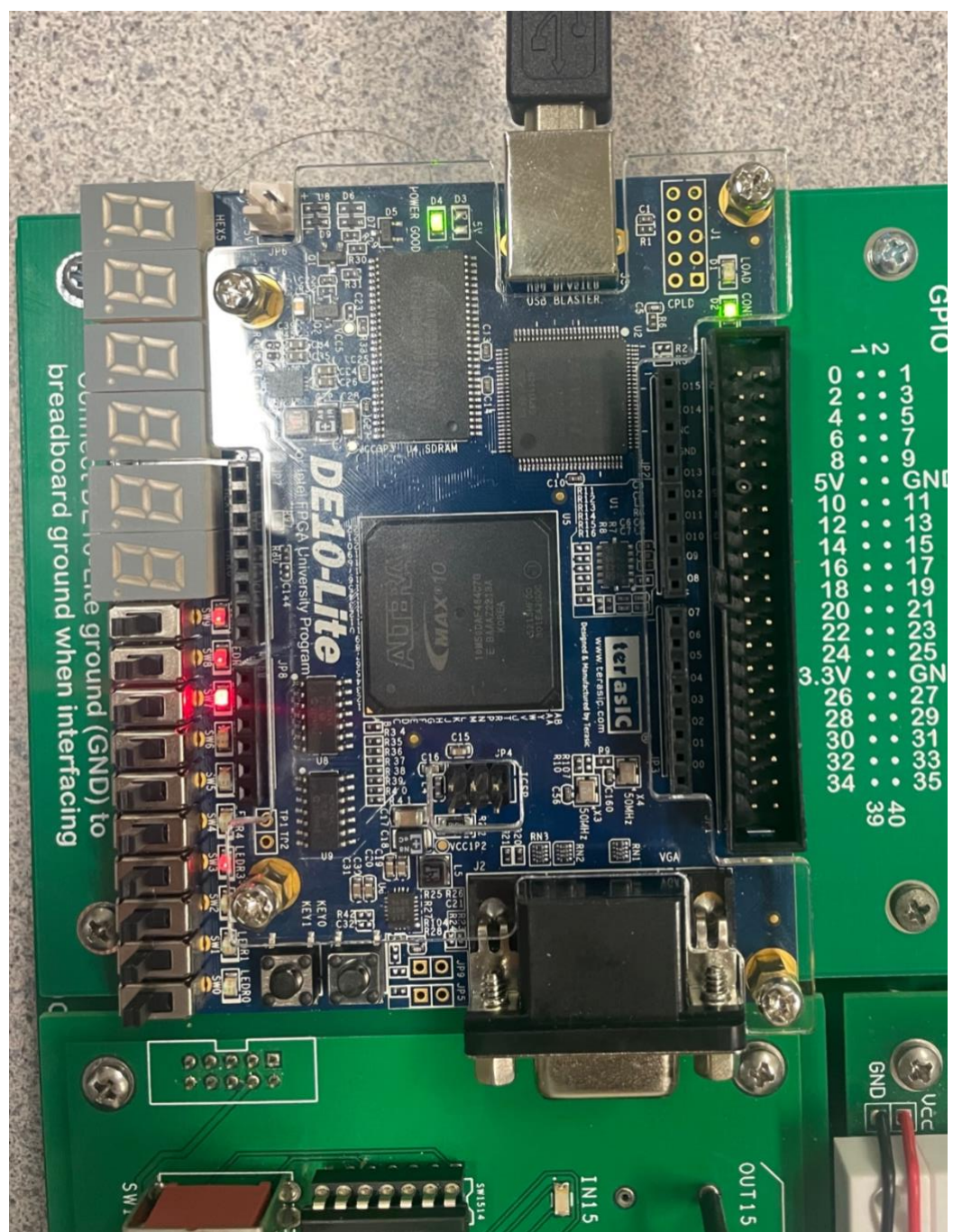
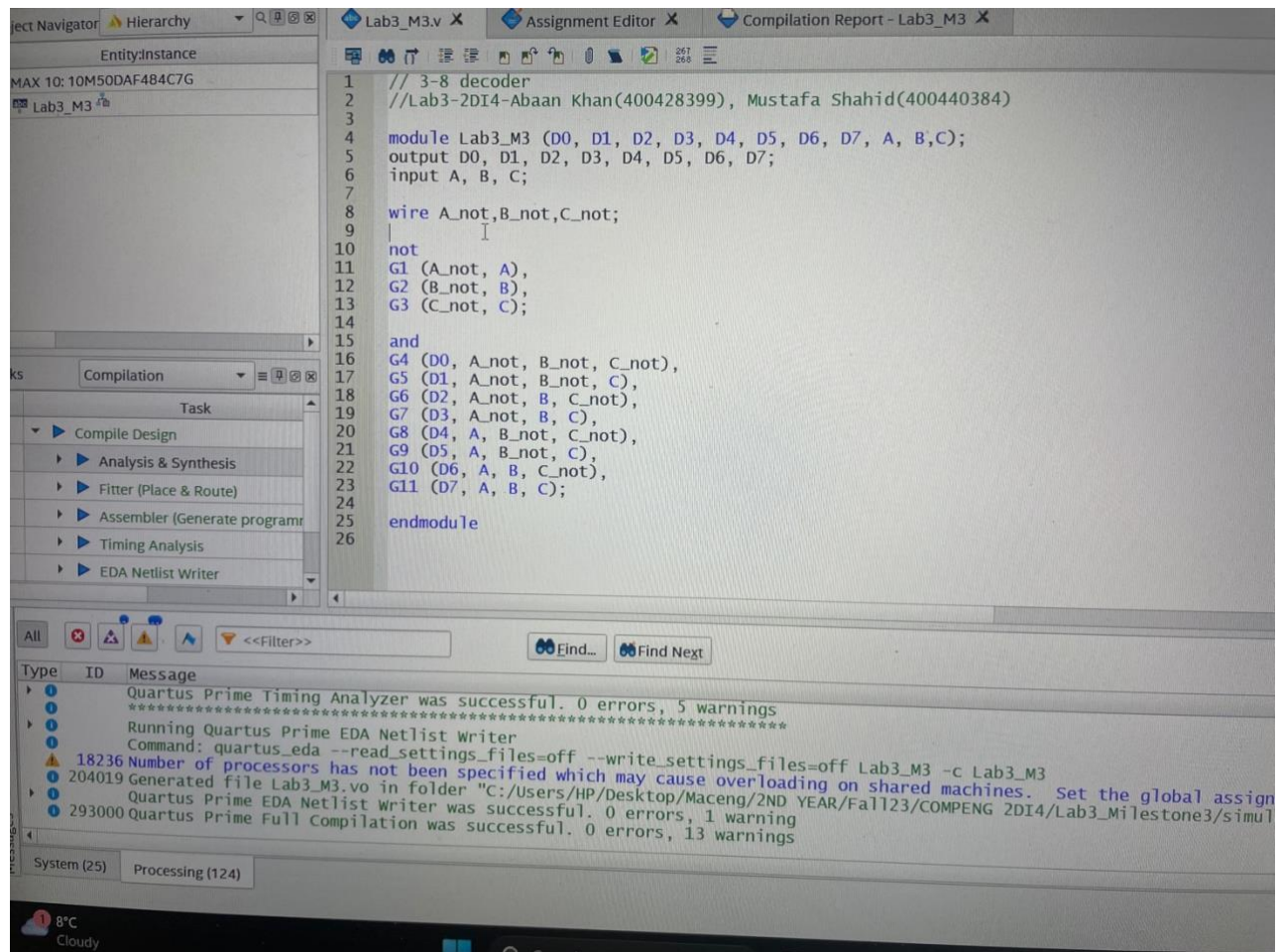


Figure r# 4: Circuit for 3:8 Decoder using primitive logic gates.



A screenshot of our Verilog code



Screenshot for the corresponding pin assignments:

The screenshot displays the Quartus Assignment Editor for the project 'Lab3_M3'. The main window shows a table of pin assignments. The table has columns for 'tatu', 'From', 'To', 'Assignment Name', 'Value', 'Enabled', 'Entity', 'Comment', and 'Tag'. The assignments are as follows:

tatu	From	To	Assignment Name	Value	Enabled	Entity	Comment	Tag
1	✓	out D0	Location	PIN_D14	Yes			
2	✓	out D1	Location	PIN_E14	Yes			
3	✓	out D2	Location	PIN_C13	Yes			
4	✓	out D3	Location	PIN_D13	Yes			
5	✓	out D4	Location	PIN_D10	Yes			
6	✓	out D5	Location	PIN_A10	Yes			
7	✓	out D6	Location	PIN_A9	Yes			
8	✓	out D7	Location	PIN_A8	Yes			
9	✓	A	Location	PIN_D12	Yes			
10	✓	B	Location	PIN_C11	Yes			
11	✓	C	Location	PIN_C10	Yes			
12	<<new>>	<<new>>	<<new>>					

The bottom panel shows the compilation messages:

```
Quartus Prime Timing Analyzer was successful. 0 errors, 5 warnings
Running Quartus Prime EDA Netlist Writer
Command: quartus_eda --read_settings_files=off --write_settings_files=off Lab3_M3 -c Lab3_M3
18236 Number of processors has not been specified which may cause overloading on shared machines. Set the global assignment NUM_PARALLEL_PRO
204019 Generated file Lab3_M3.vo in folder "C:/Users/HP/Desktop/Maceng/2ND YEAR/Fall123/COMPENG 2DI4/Lab3_Milestone3/simulation/questa/" for E
293000 Quartus Prime Full Compilation was successful. 0 errors, 13 warnings
```

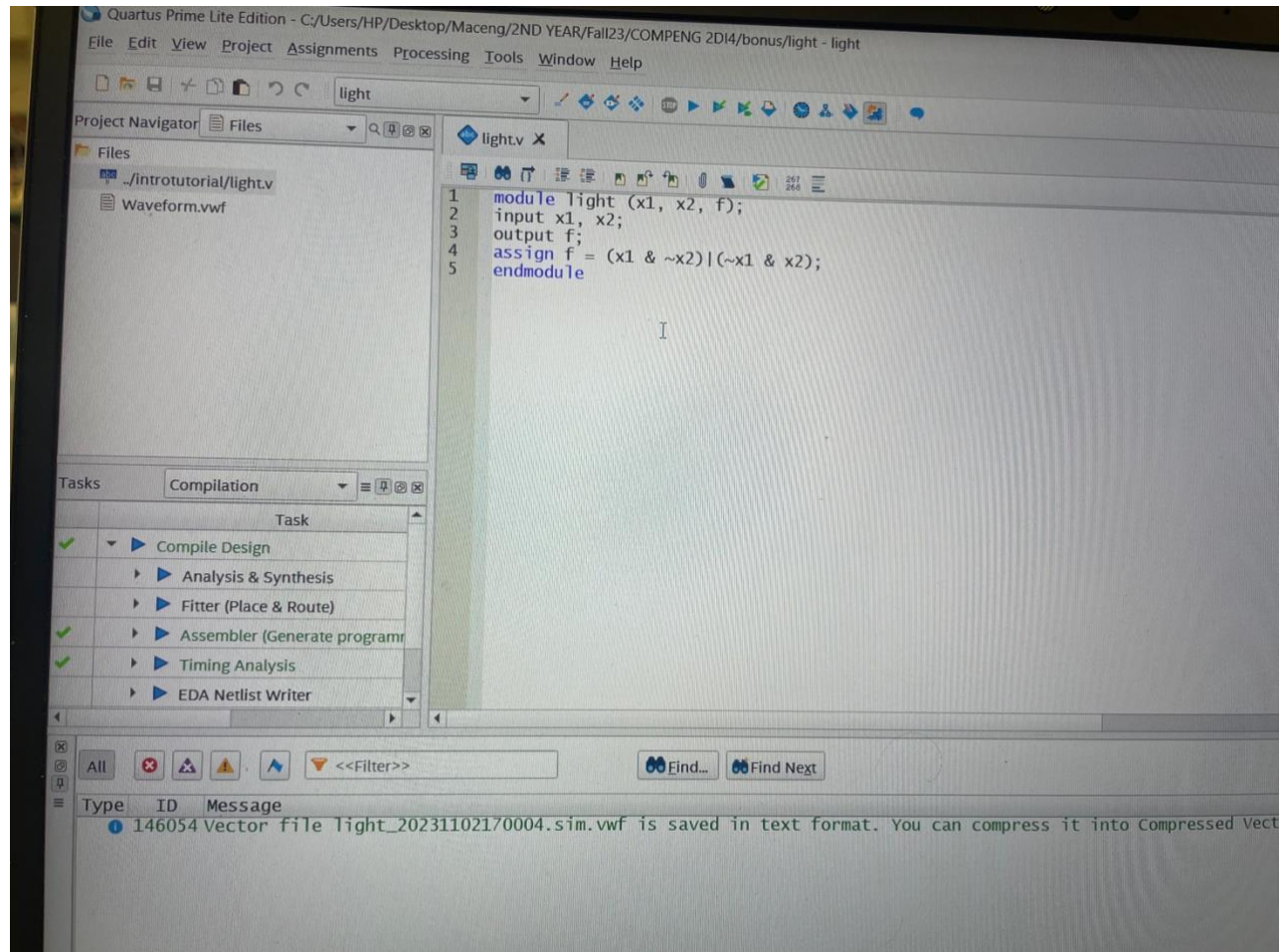

3.5.5 (Bonus) HDL Functional Simulation of Lamp Controller

Bonus Milestone: HDL Simulation of Lamp Controller

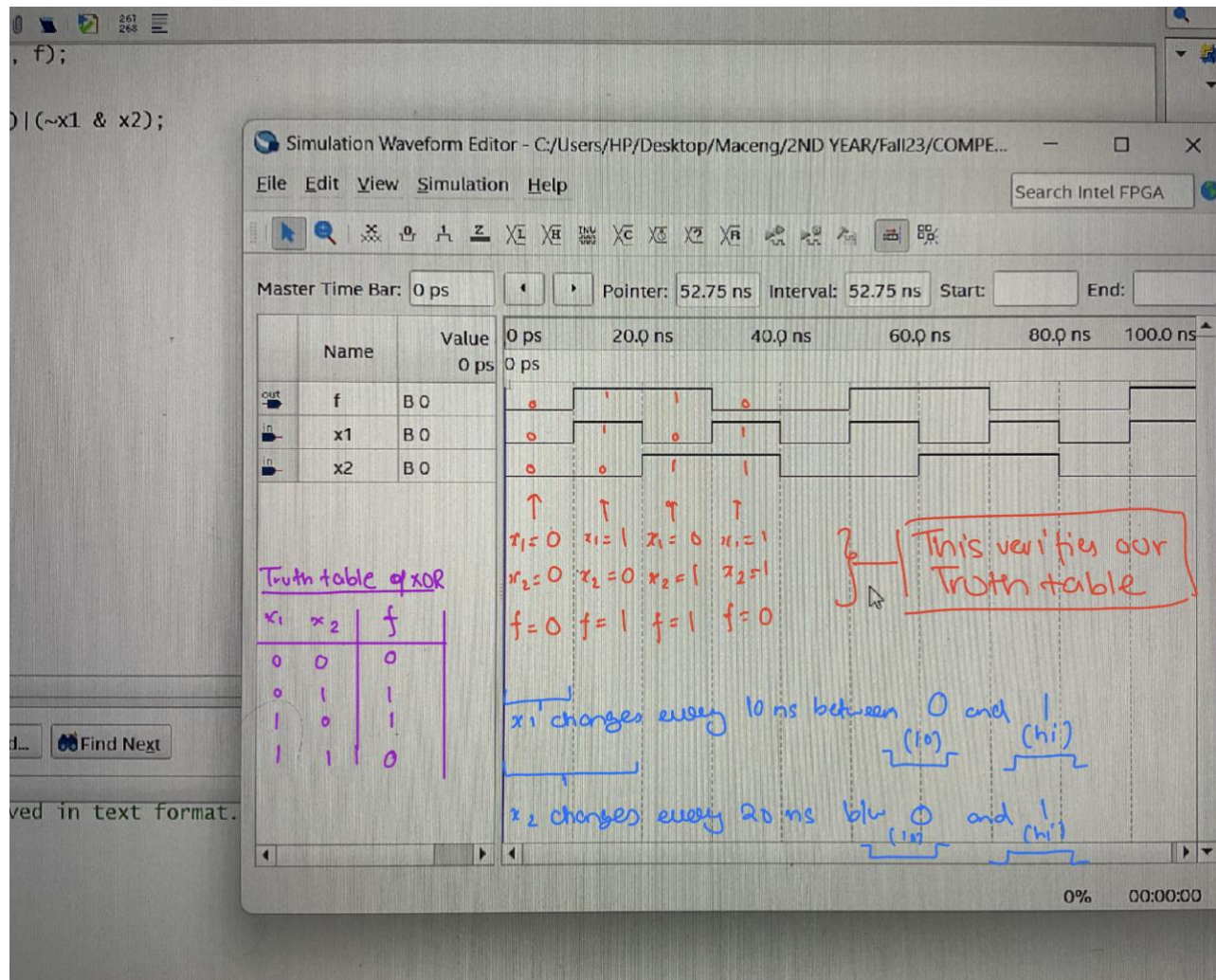
Return to your Lamp Controller, but instead of external switch connections you will run a basic simulation by manually creating a functional simulation with Vector Waveform Files or a testbench module to test all input combinations, and perform a functional simulation. Use Primitive Gates Model (AND, OR, NOT, etc.) for your design entry of the lamp controller. Note section 3.6 has additional information to assist with simulation.

For this bonus include in the report:

1. Copy of Verilog HDL code entry defines the Lamp Controller (place in appendices) [4 marks],



2. An annotated screenshot of functional simulation results with all combinations of bit inputs. The annotation should illustrate your understanding of the connection between the inputs and outputs. [2 marks],



3. A short description of how you tested and verified the circuit [4 marks].

ANSWER-3:

In the manual, we are instructed to use 20ns for the x1&x2. However, we modified the x1 value to 10ns and left x2 at 20ns because we are using an exclusive OR gate and this helps us to differentiate between the output and clearly view the results of our simulation.