CprE 381 – Computer Organization and Assembly-Level Programming

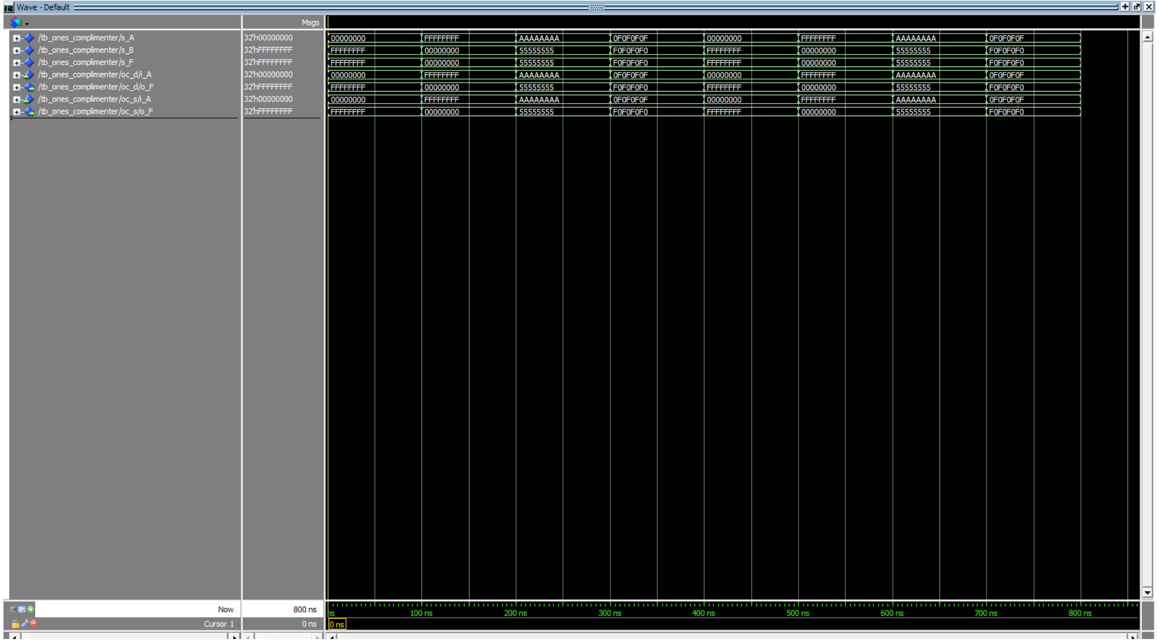
Lab-02 Report

Student Name	Colby McKinley
Section / Lab Time	Section 5/ W6:10 – 8:00p

***Submit a typeset pdf version of this on Canvas by the due date (i.e., the start of your next lab section). Refer to the highlighted language in the Lab-02 instructions for the context of the following questions*.**

a. [Prelab] At the end of Chapter 3, answer questions 4b), 5a), and 6.
See /Prelab2.pdf

b. [Part 1 (c)] Waveform.

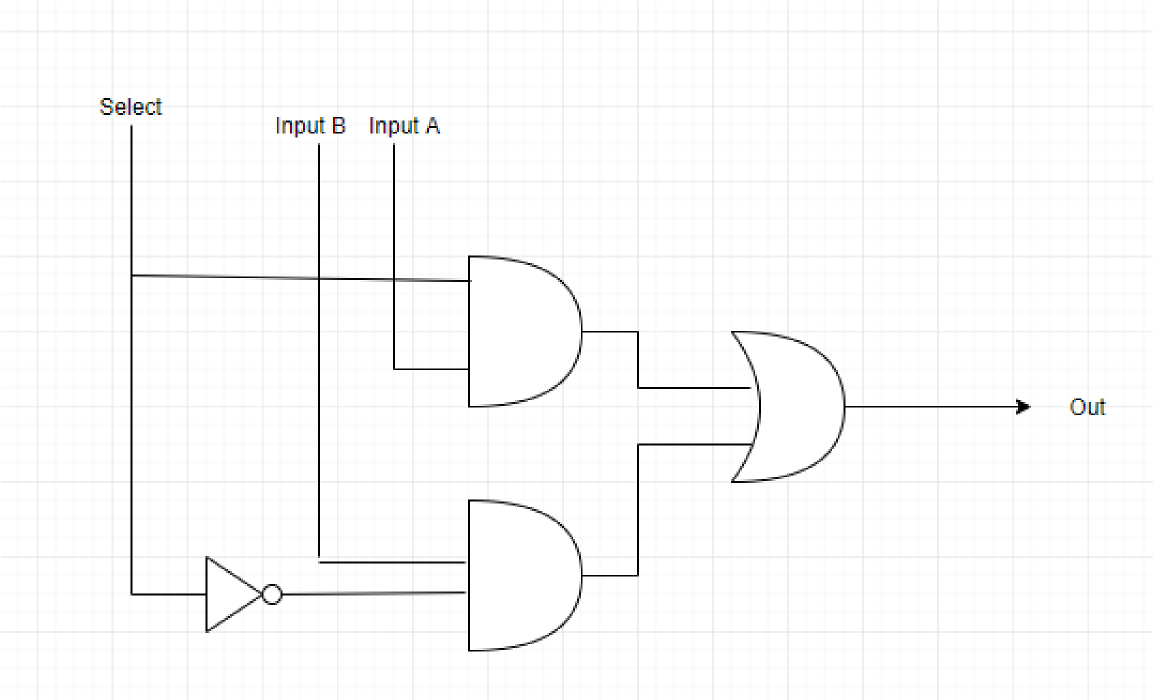


```
/tb_ones_complimenter/s_A        32'h00000000  00000000 FFFFFFFF AAAAAAAA 0F0F0F0F 00000000 FFFFFFFF AAAAAAAA 0F0F0F0F
/tb_ones_complimenter/s_B        32'hFFFFFFFF  FFFFFFFF 00000000 55555555 F0F0F0F0 FFFFFFFF 00000000 55555555 F0F0F0F0
/tb_ones_complimenter/s_F        32'hFFFFFFFF  FFFFFFFF 00000000 55555555 F0F0F0F0 FFFFFFFF 00000000 55555555 F0F0F0F0
/tb_ones_complimenter/oc_d/i_A   32'h00000000  00000000 FFFFFFFF AAAAAAAA 0F0F0F0F 00000000 FFFFFFFF AAAAAAAA 0F0F0F0F
/tb_ones_complimenter/oc_d/o_F   32'hFFFFFFFF  FFFFFFFF 00000000 55555555 F0F0F0F0 FFFFFFFF 00000000 55555555 F0F0F0F0
/tb_ones_complimenter/oc_s/i_A   32'h00000000  00000000 FFFFFFFF AAAAAAAA 0F0F0F0F 00000000 FFFFFFFF AAAAAAAA 0F0F0F0F
/tb_ones_complimenter/oc_s/o_F   32'hFFFFFFFF  FFFFFFFF 00000000 55555555 F0F0F0F0 FFFFFFFF 00000000 55555555 F0F0F0F0

Now       800 ns   ns   100 ns   200 ns   300 ns   400 ns   500 ns   600 ns   700 ns   800 ns
Cursor 1  0 ns     0 ns
```

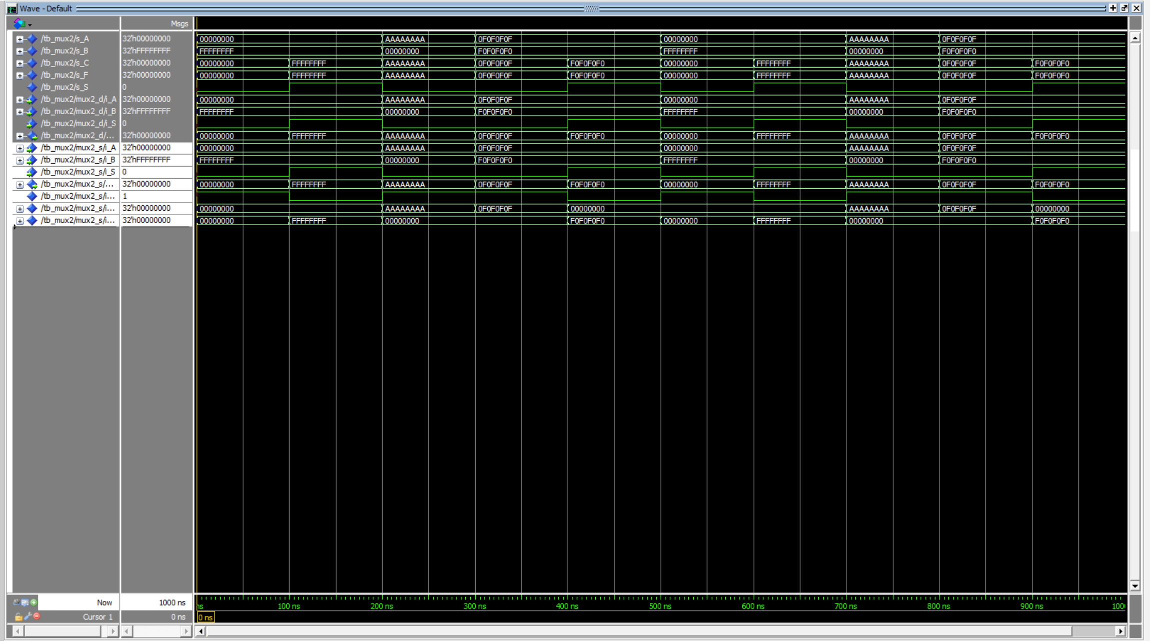
c. [Part 2 (a)] Draw the truth table, Boolean equation, and Boolean circuit equivalent (using only two-input gates) that implements a 2:1 mux.

In A (A)	In B (B)	Select (S)	Out (O)
X	X	0	In A
X	X	1	In B

Boolean Equation: $O = A \bullet S + B \bullet \overline{S}$



d. [Part 2 (e)] Waveform.



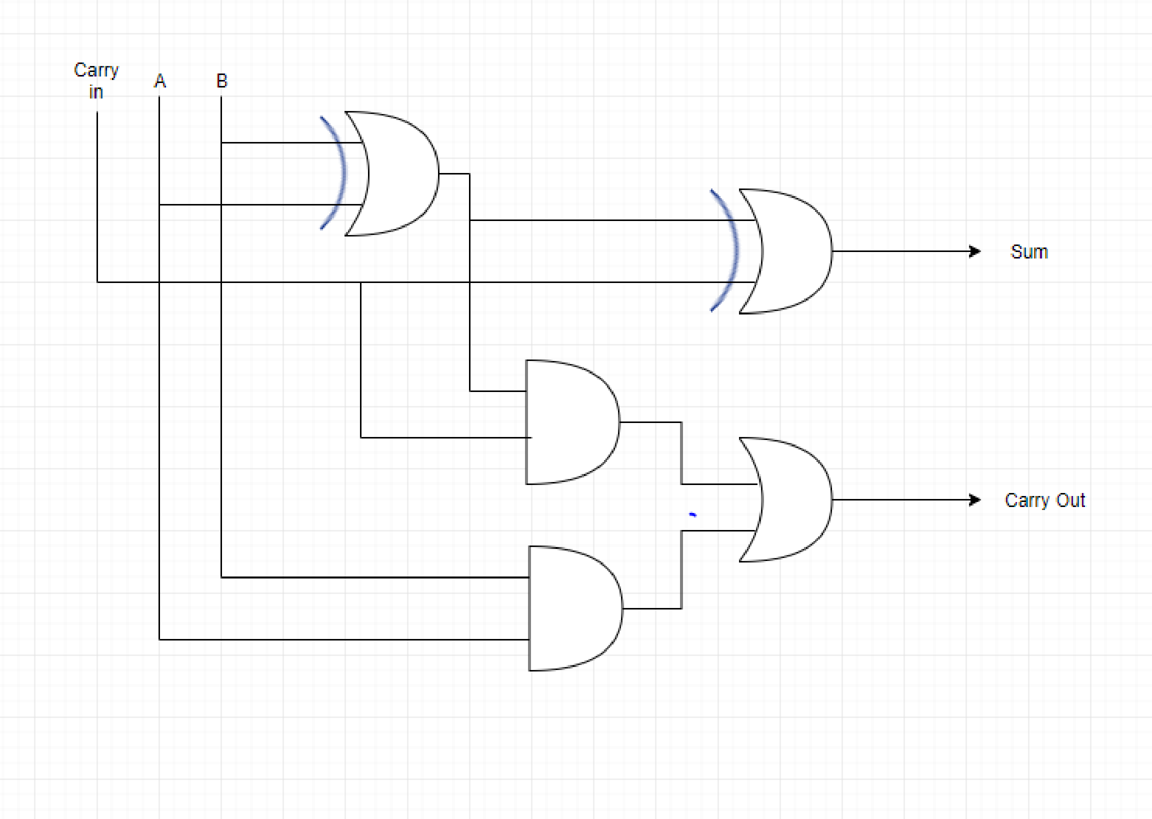
```
Wave - Default
                                   Msgs
/tb_mux2/s_A                       32'h00000000
/tb_mux2/s_B                       32'hFFFFFFFF
/tb_mux2/s_C                       32'h00000000
/tb_mux2/s_F                       32'h00000000
/tb_mux2/s_S                       0
/tb_mux2/mux2_d/i_A                32'h00000000
/tb_mux2/mux2_d/i_B                32'hFFFFFFFF
/tb_mux2/mux2_d/i_S                0
/tb_mux2/mux2_d/...                32'h00000000
/tb_mux2/mux2_s/i_A                32'h00000000
/tb_mux2/mux2_s/i_B                32'hFFFFFFFF
/tb_mux2/mux2_s/i_S                0
/tb_mux2/mux2_s/...                32'h00000000
/tb_mux2/mux2_s/i...               1
/tb_mux2/mux2_s/i...               32'h00000000
/tb_mux2/mux2_s/i...               32'h00000000

Now                                1000 ns
Cursor 1                           0 ns
```

e. [Part 3 (a)] Draw the truth table, Boolean equation, and Boolean circuit equivalent (using only two-input gates) that implements a full adder.

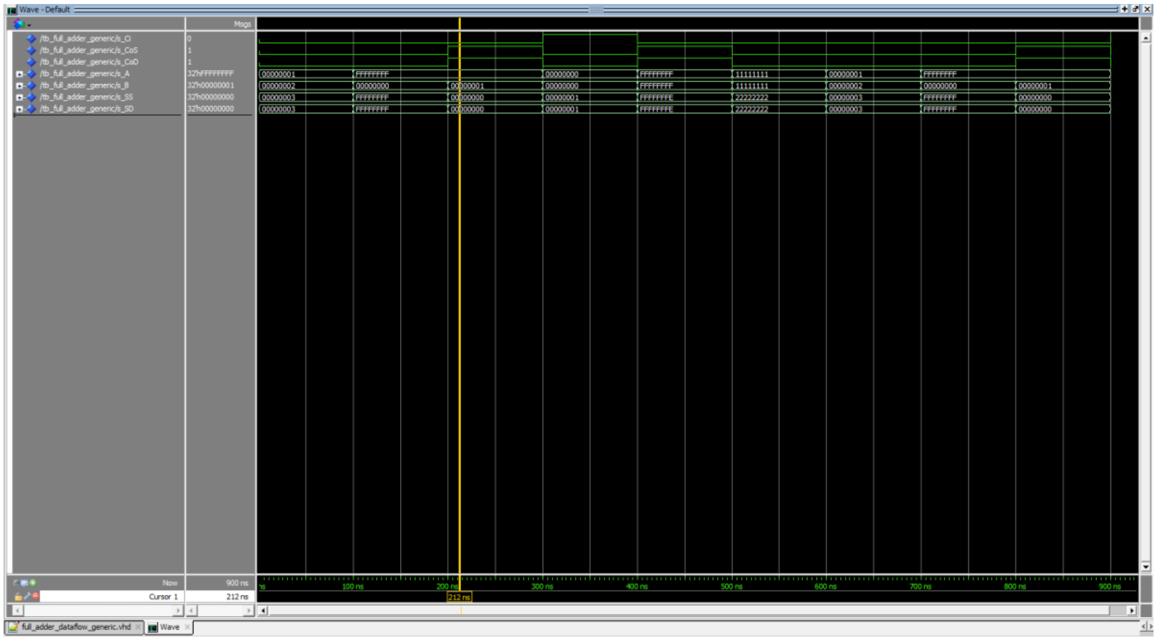
Input A (A)	Input B (B)	Carry in (Cin)	Sum (S)	Carry out (Cout)
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Boolean expression: $S = A \oplus B \oplus Cin$

$Cout = A \bullet B + Cin \bullet A \oplus Cin$

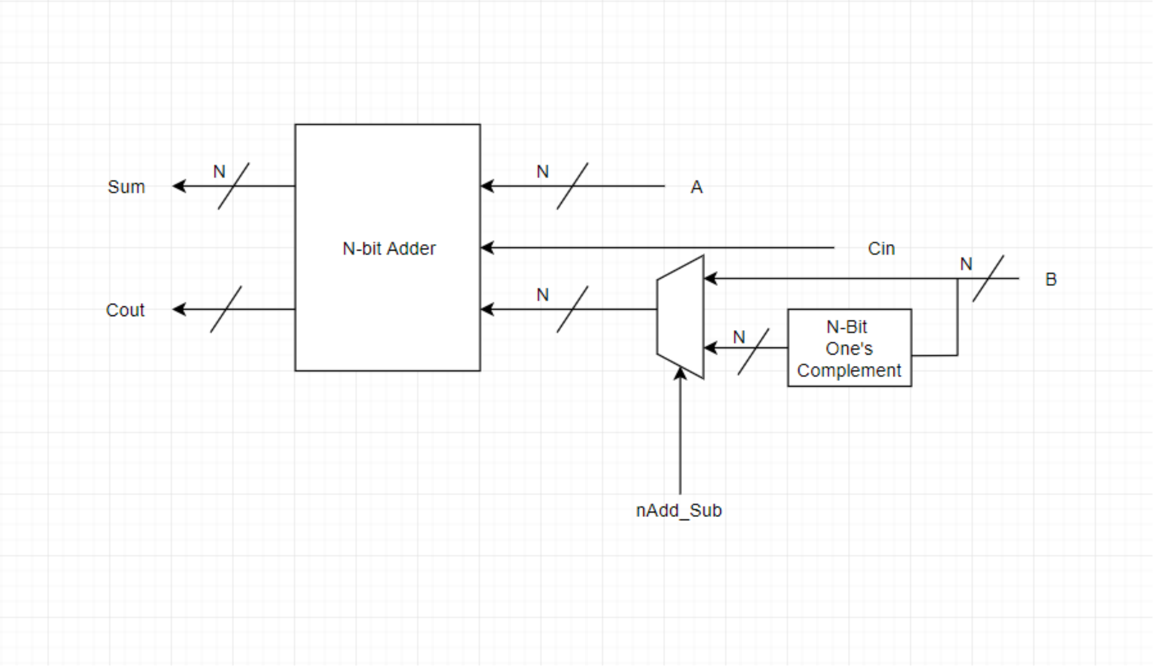


f. [Part 3 (e)] Waveform.

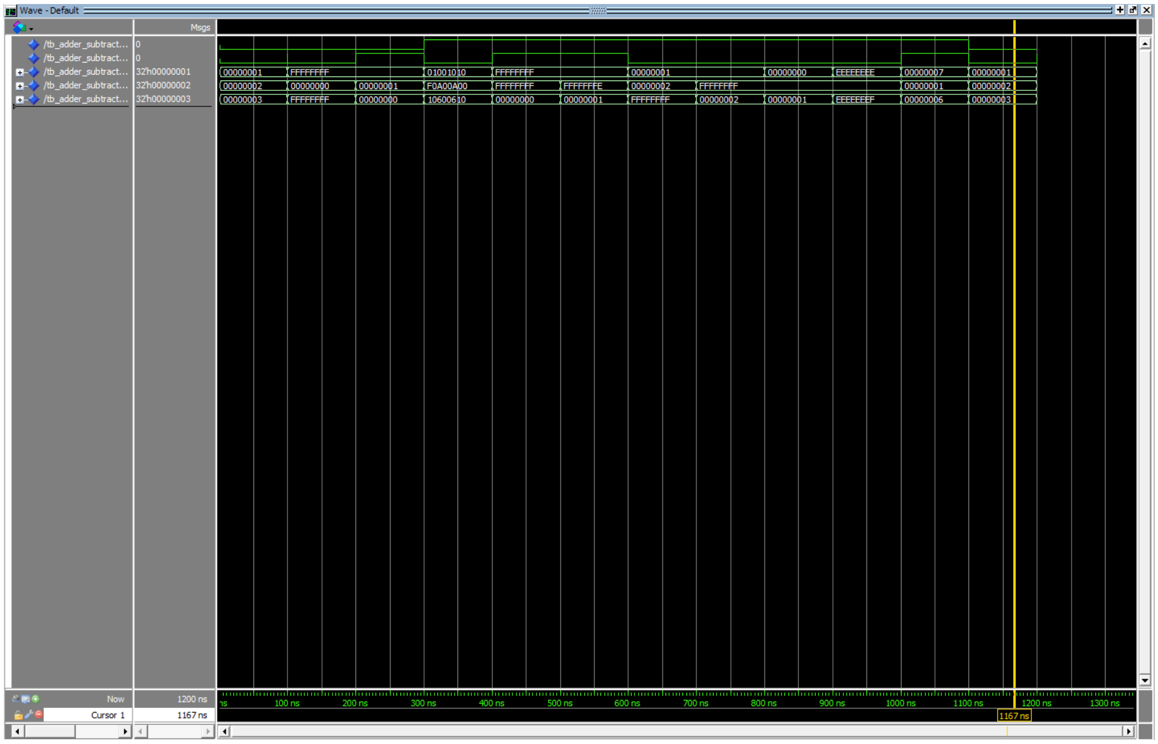


```
Wave - Default
                                   Msgs
/tb_full_adder_generic/s_Ci        0
/tb_full_adder_generic/s_CoS       1
/tb_full_adder_generic/s_CoD       1
/tb_full_adder_generic/s_A         32'hFFFFFFFF   00000001  FFFFFFFF            00000000  FFFFFFFF  11111111  00000001  FFFFFFFF
/tb_full_adder_generic/s_B         32'h00000001   00000002  00000000  00000001  00000000  FFFFFFFF  11111111  00000002  00000000  00000001
/tb_full_adder_generic/s_SS        32'h00000000   00000003  FFFFFFFF  00000000  00000001  FFFFFFFE  22222222  00000003  FFFFFFFF  00000000
/tb_full_adder_generic/s_SD        32'h00000000   00000003  FFFFFFFF  00000000  00000001  FFFFFFFE  22222222  00000003  FFFFFFFF  00000000

Now       900 ns    ns  100 ns  200 ns  300 ns  400 ns  500 ns  600 ns  700 ns  800 ns  900 ns
Cursor 1  212 ns                  212 ns
full_adder_dataflow_generic.vhd   Wave
```

g. [Part 4 (a)] Draw a schematic (don’t use a schematic capture tool) showing how an N-bit adder/subtractor with control can be implemented using only the three main components designed in problems 1), 2), and 3) (the N-bit inverter, N-bit 2:1 mux, and N-bit adder). How is the ‘nAdd_Sub’ bit used?



h. [Part 4 (c)] Provide multiple waveform screenshots in your write-up to confirm that this component is working correctly. What test-cases did you include and why?


Wave - Default
Msgs
/tb_adder_subtract... 0
/tb_adder_subtract... 0
/tb_adder_subtract... 32'h00000001
/tb_adder_subtract... 32'h00000002
/tb_adder_subtract... 32'h00000003
00000001 FFFFFFFF 01001010 FFFFFFFF 00000001 00000000 EEEEEEEE 00000007 00000001
00000002 00000000 00000001 F0A00A00 FFFFFFFF FFFFFFFE 00000002 FFFFFFFF 00000001 00000002
00000003 FFFFFFFF 00000000 10600610 00000000 00000001 FFFFFFFF 00000002 00000001 EEEEEEEF 00000006 00000003
Now 1200 ns
Cursor 1 1167 ns
ns 100 ns 200 ns 300 ns 400 ns 500 ns 600 ns 700 ns 800 ns 900 ns 1000 ns 1100 ns 1200 ns 1300 ns
1167 ns

I include test cases where overflow or underflow should have occurred, along with a few normal cases. I wanted to test possible paths of the adder/subtractor.

i. [Feedback] You must complete this section for your lab to be graded. Write down the first response you think of; I expect it to take roughly 5 minutes (do not take more than 10 minutes).

   i. How many hours did you spend on this lab?

Task	During lab time	Outside of lab time
Reading lab	15 minutes	0 minutes
Pencil/paper design	20 minutes	45 minutes
VHDL design	35 minutes	3 hr
Assembly coding	0 minutes	0 minutes
Simulation	30 minutes	1 hr
Debugging	30 minutes	2 hr
Report writing	0 minutes	1 hr
Other:		
Total	2 hr	7hr 45 minutes

   ii. If you could change one thing about the lab experience, what would it be? Why? Allow drawing the semantics to be done by hand or Quartus. It takes way to long with the drawing tools.

   iii. What was the most interesting part of the lab?

I think the most interesting part of the lab was seeing when dataflow and structural were easiest to implement.