Report ENCS2340

Final Project

Robert Bannoura

1210574

3/2/2023

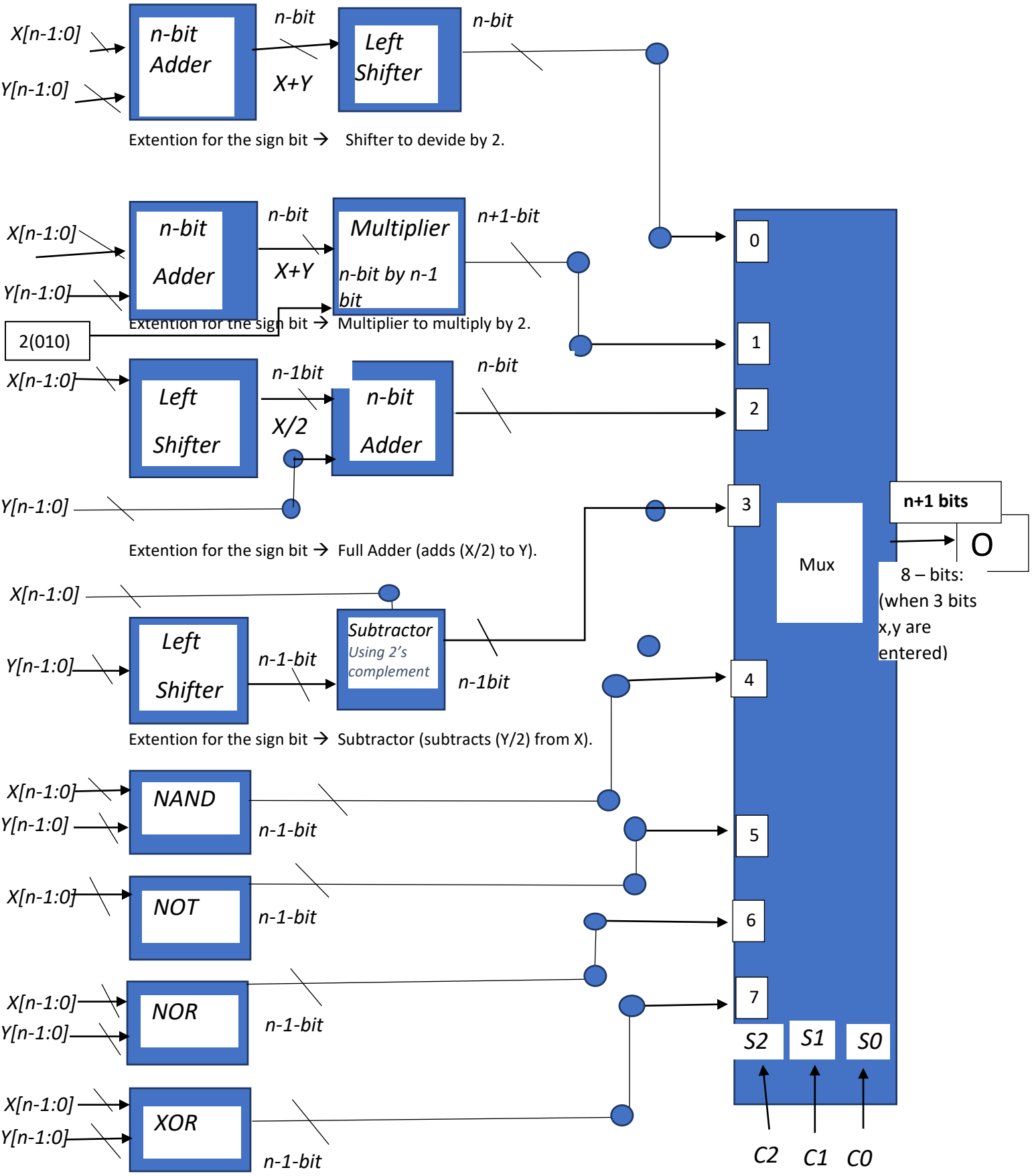
Part 1:-

a) Specify the size of the output (O) in bits so the overflow can never occur.

ALU Function Code (C)	ALU Output (O)	Size in bits output
000	(X+Y)/2	n-bits
001	2*(X+Y)	n+1 bits
010	(X/2)+Y	n-bits
011	X-(Y/2)	n-1 bits
100	X NAND Y	n-1 bits
101	NOT(X)	n-1 bits
110	X NOR Y	n-1 bits
111	X XOR Y	n-1 bits

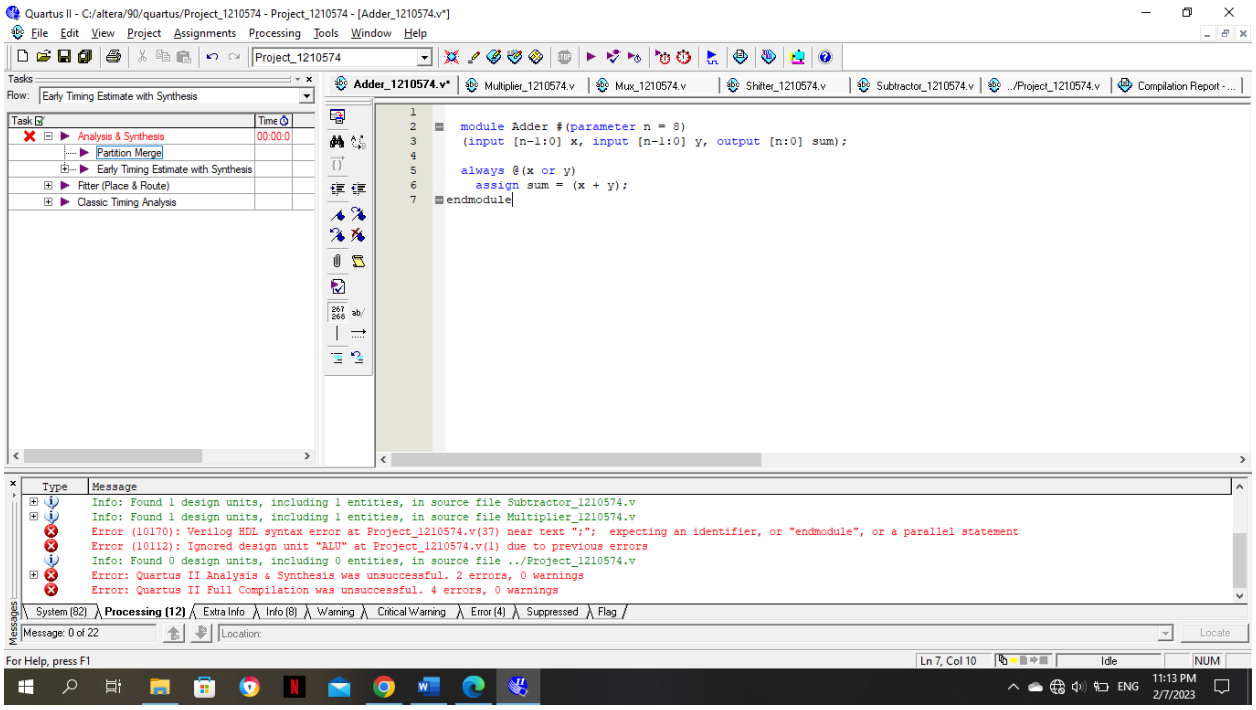
Number of output should be equal to the size of inputs (X and Y) plus 1 bit (for the sign bit). The extra bits is to represent the sign of the output, If the inputs are n-bit signed numbers, the output should be (n+1)-bit signed numbers.

b) Show the ALU implementation using medium-scale integration (MSI) components and minimum number of gates (i.e. in blocks with their sizes). Note that, you might use some kind of extension (sign- or zero-extension).

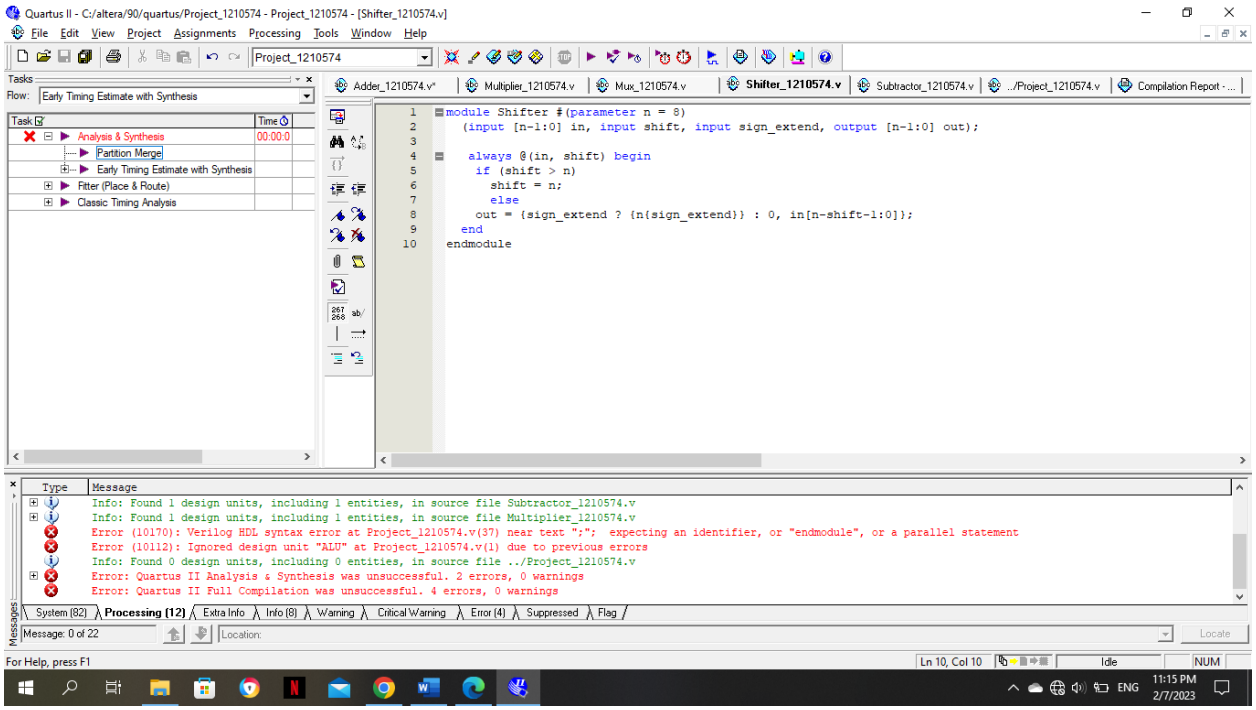

X[n-1:0]
Y[n-1:0]
n-bit Adder
n-bit
X+Y
Left Shifter
n-bit
Extention for the sign bit → Shifter to devide by 2.
X[n-1:0]
Y[n-1:0]
n-bit Adder
n-bit
X+Y
Multiplier
n-bit by n-1 bit
n+1-bit
Extention for the sign bit → Multiplier to multiply by 2.
2(010)
X[n-1:0]
Left Shifter
n-1bit
X/2
n-bit Adder
n-bit
Y[n-1:0]
Extention for the sign bit → Full Adder (adds (X/2) to Y).
X[n-1:0]
Y[n-1:0]
Left Shifter
n-1-bit
Subtractor
Using 2's complement
n-1bit
Extention for the sign bit → Subtractor (subtracts (Y/2) from X).
X[n-1:0]
Y[n-1:0]
NAND
n-1-bit
X[n-1:0]
NOT
n-1-bit
X[n-1:0]
Y[n-1:0]
NOR
n-1-bit
X[n-1:0]
Y[n-1:0]
XOR
n-1-bit
0
1
2
3
4
5
6
7
Mux
S2
S1
S0
C2
C1
C0
n+1 bits
O
8 – bits:
(when 3 bits x,y are entered)

c) (10 points) Write behavioral Verilog modules for your elements you defined in Part (b). Be noted that the size of every element you define should be parameterized, so that you can vary the design during the testing phase.

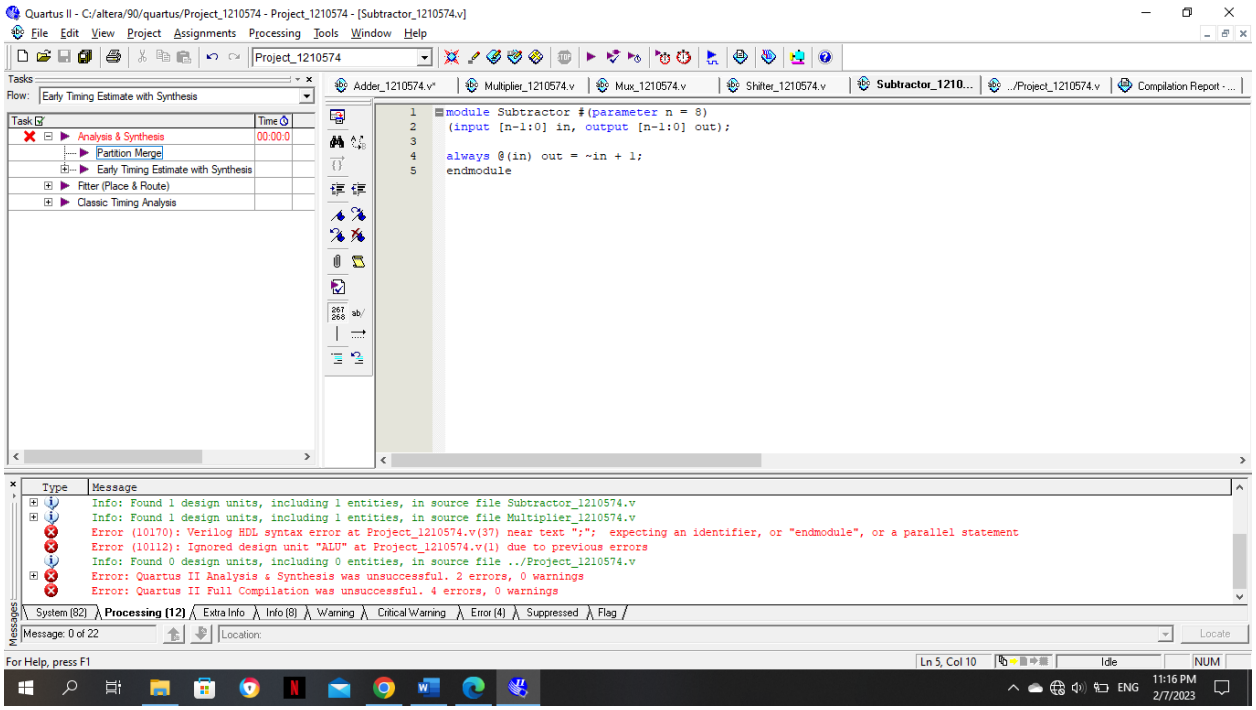
Adder_1210574 module:



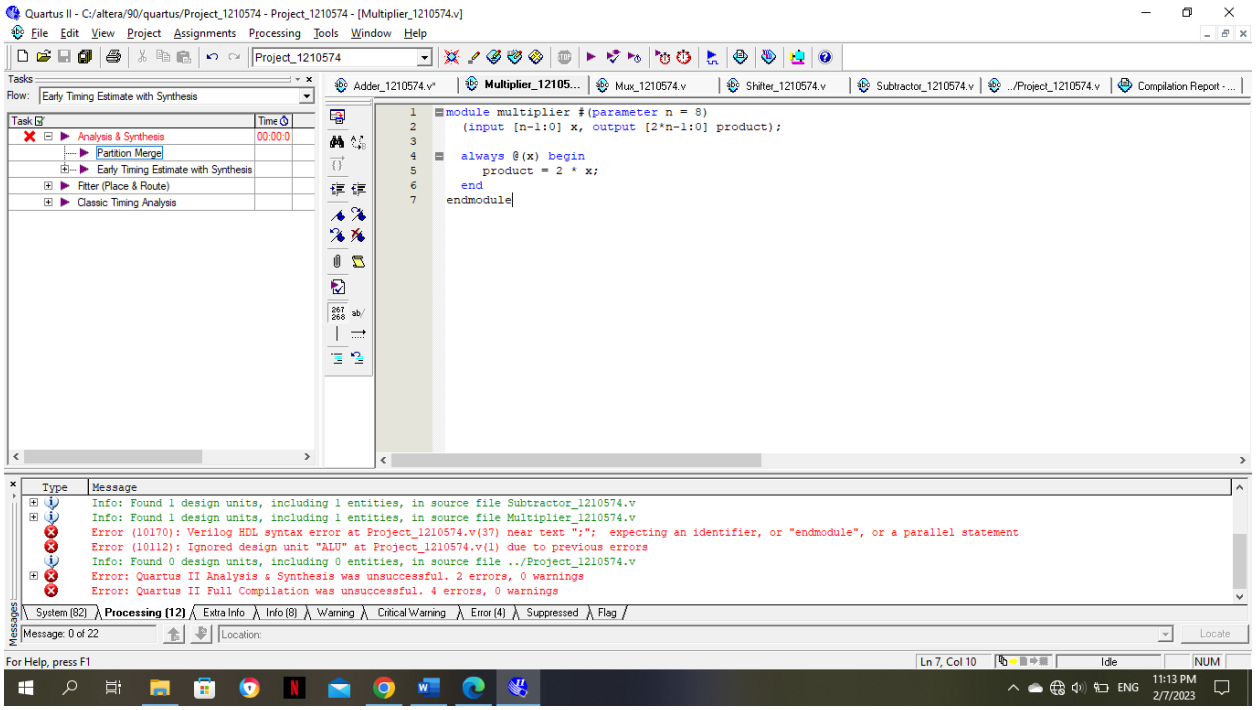
Shifter_1210574 module:



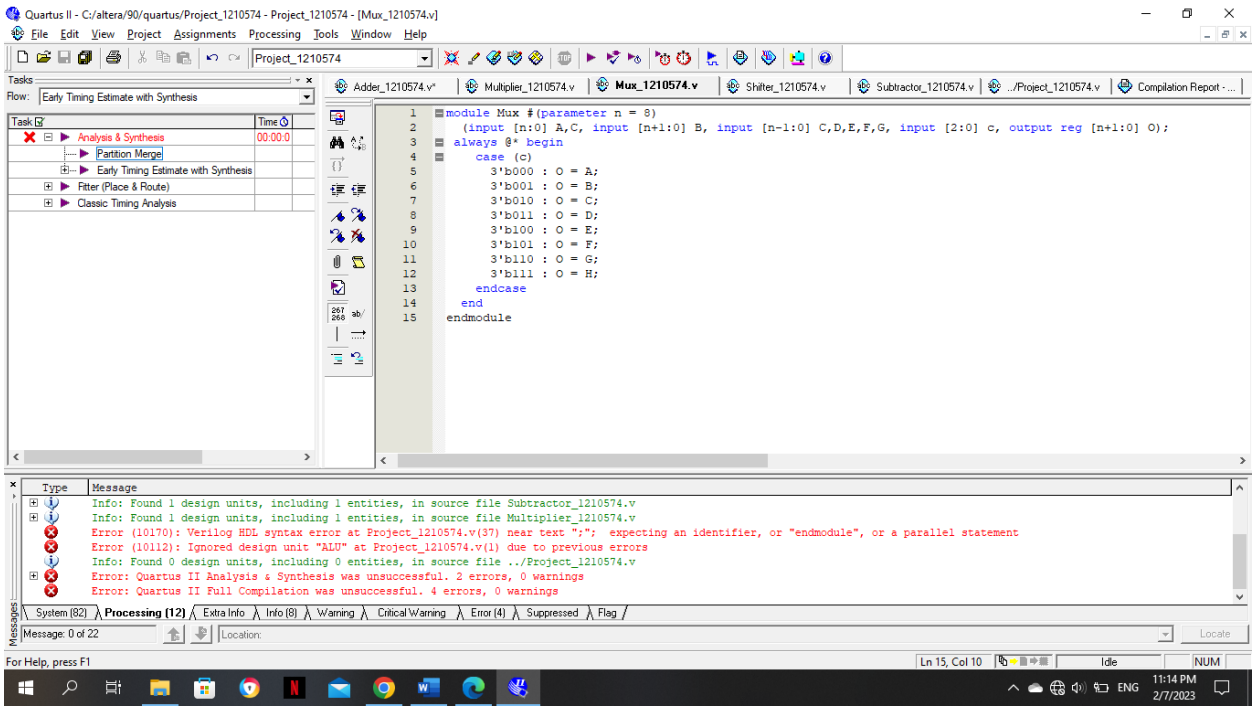
Subtractor_1210574 module:



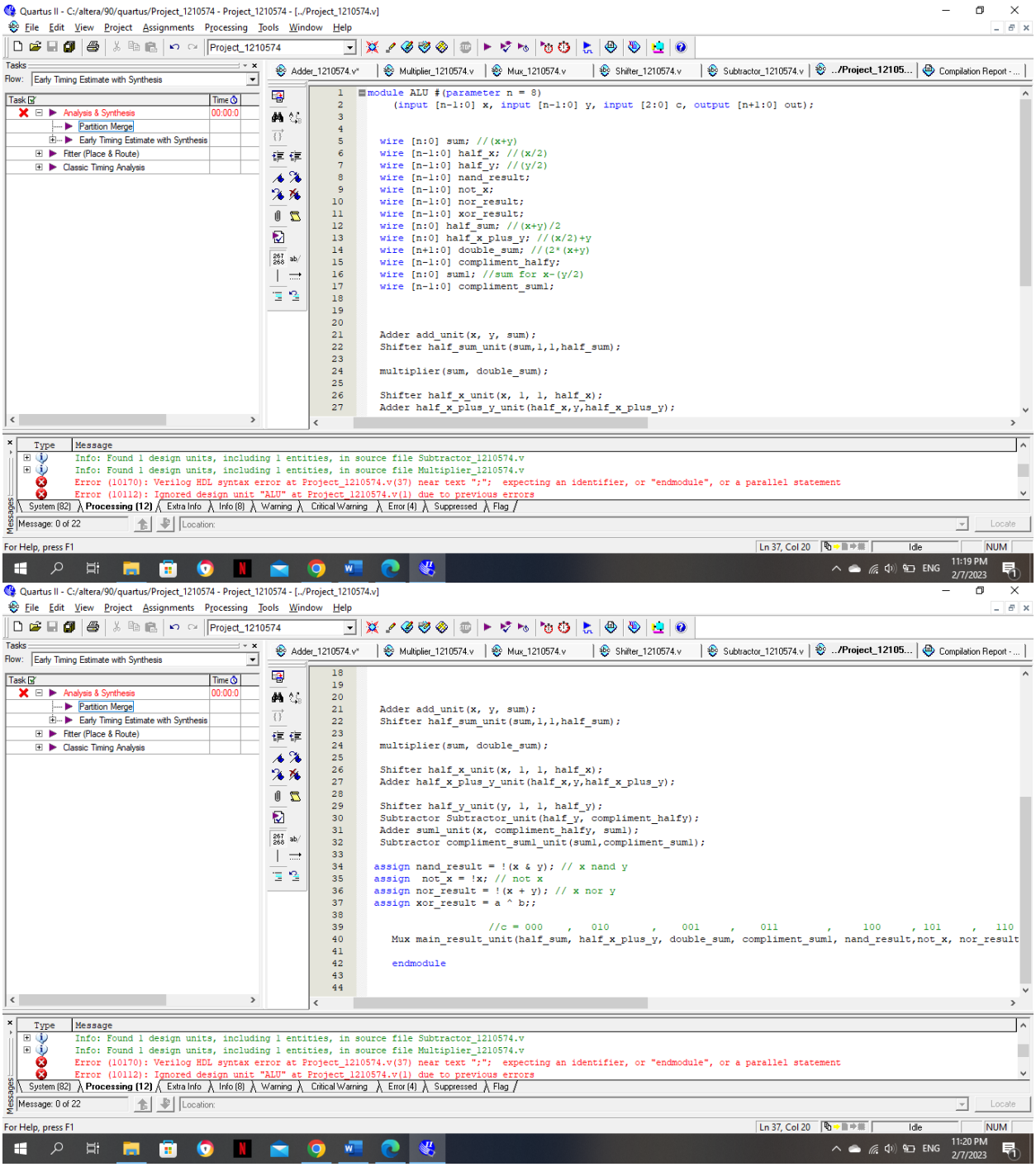
Multiplier_1210574 module:



Mux_1210574 module:



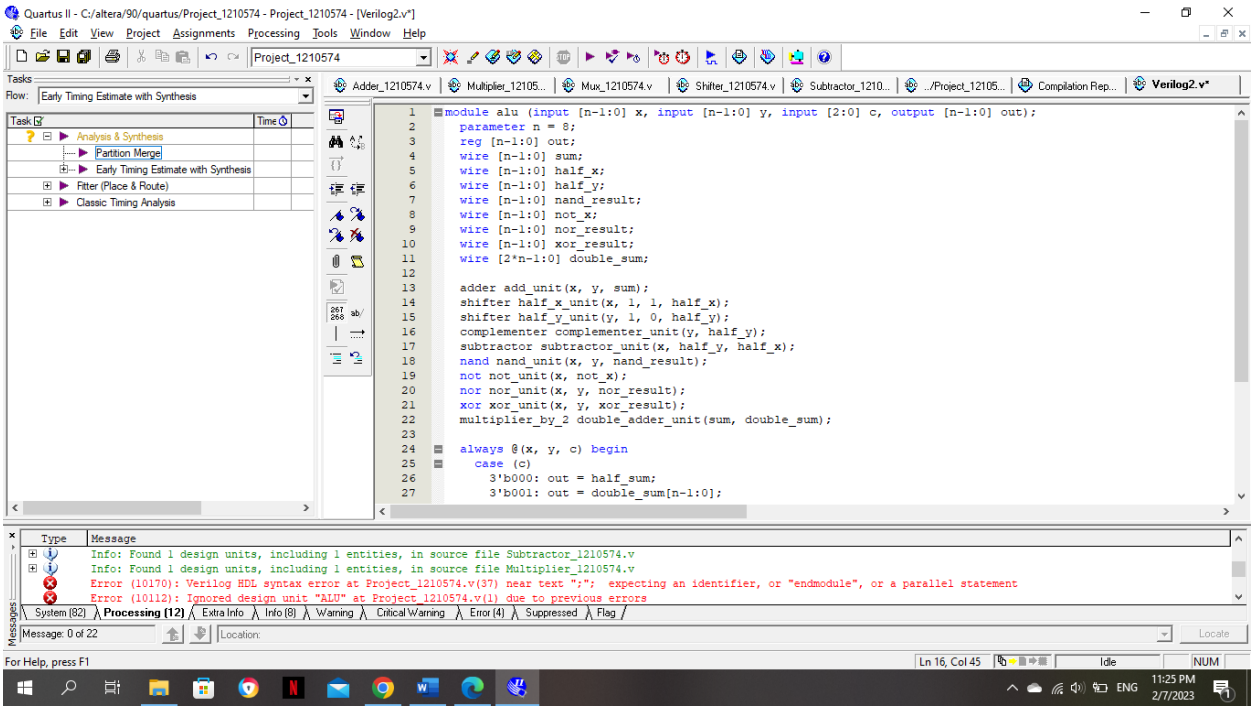
d) Write a structural Verilog model for your ALU designed in Part (b) using the elements you defined in Part (c).

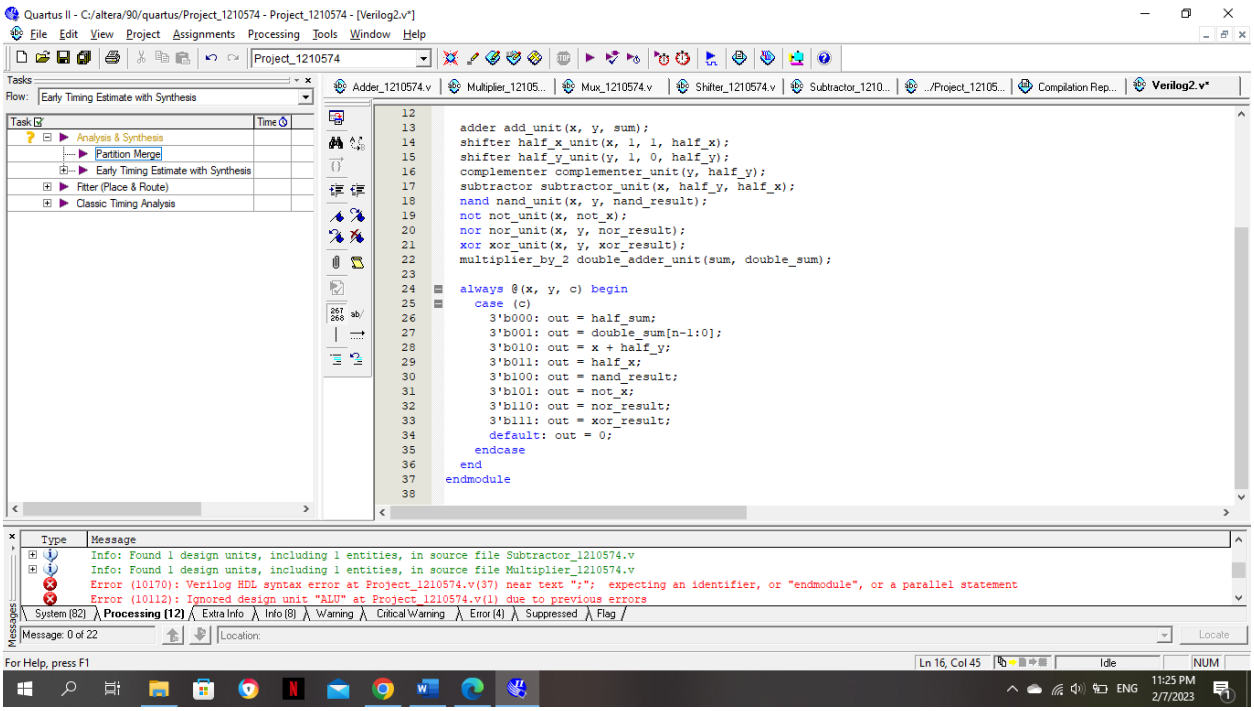


e) Generate the waveforms of the ALU defined in Part (d), assumes that X and Y are 4-bits and their values based on your student ID should be set as follows:

Test	X	Y	C	O
1	4	7	5	X nand y
2	0	1	2	2*(x+y)=2
3	-4	-7	2	2*(x+y)=-22

e) Write a single behavioral Verilog module that models the designed ALU.





f) Generate the waveforms of the behavioral ALU defined in Part (f), assumes that X and Y are 4-bits and their values based on your student ID should be set as follows: The general representation of the student ID is 1C2Y2X2C1Y1X1, so, if your student ID is 1220520, then X, Y, and C values for the three test cases as follows:

Test	X	Y	C	O
1	4	7	5	X nand y
2	0	1	2	2*(x+y)=2
3	-4	-7	2	2*(x+y)=-22