a) ,22Specify the size of the output (O) in bits so the overflow can never occur.

Answer

Arthmatic :

100	x NAND y
101	NOT (x)
110	x NOR y
111	x XOR y

LOGIC :

000	(X+Y)/2
001	2*(X+Y)
010	(X/2)+Y
011	X-(Y/2)

X : n bit // x=111

Y : n bit //y=111

X NAND y -> n bit

(111) NAND (111) --> n bit // so output = n bit

Y : n bit //y=111

X : n bit// x=111

NOT (x):

NOT (111) = 000

Output =n bit

X : n bit // x=111

Y : n bit //y=111

x NOR y

(111) NOR (111) = n bit

So output = n bit

Y : n bit //y=111

X : n bit// x=111

x XOR y :

(111) XOR(111)= n bit // so output= n bit

LOGIC

000 (X+Y)/2

001 2*(X+Y)

010 (X/2)+Y

011 X-(Y/2)

(X+Y) / 2 :

IF x = n bit

And y = n bit

Then (x + y) are (n+1) bit because the end carry

So the size of the output from this operation is (n+1)bit

And (x + y) divid by 2 then the size of the output still (n+1)

---

2 * (X + Y) :

IF x = n bit

And y = n bit

Then (x + y) are (n+1) bit because the end carry

And (x + y) multiplay with 2 then the size of the output become (n+2)

(X/2) + Y :

IF x = n bit

And y = n bit

And (x/2) the size is n bit

(x/2) + y the size become (n+1) bit because the end carry

X-(y/2)

IF x = n bit

And y = n bit

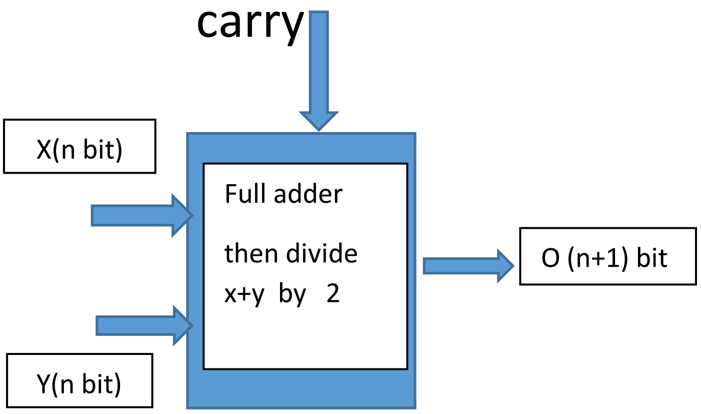
(y/2) the size is n bit

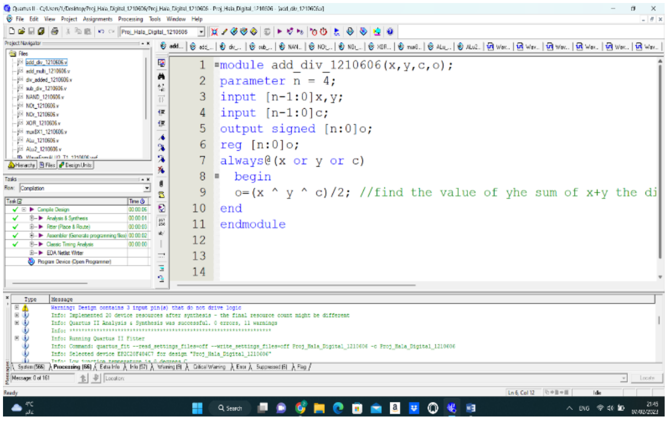
X – (y/2) the size of the output is (n+1)

(SO THE SIZE OF OUTPOUT is (n+2))

b) (10 points) 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).

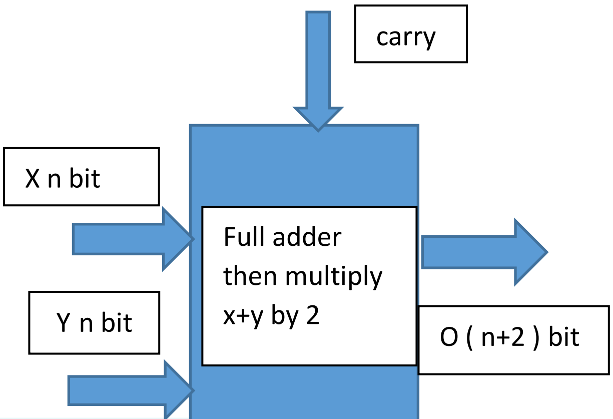
1- (X+Y)/2



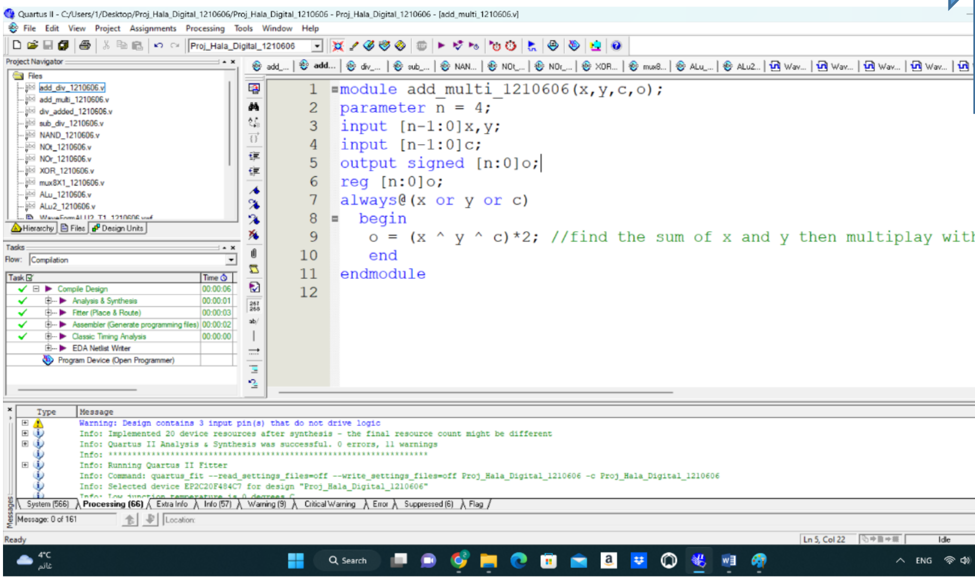


```
module add_div_1210606(x,y,c,o);
parameter n = 4;
input [n-1:0]x,y;
input [n-1:0]c;
output signed [n:0]o;
reg [n:0]o;
always@(x or y or c)
  begin
   o=(x ^ y ^ c)/2; //find the value of yhe sum of x+y the di
end
endmodule
```

```
Warning: Design contains 3 input pin(s) that do not drive logic
Info: Implemented 20 device resources after synthesis - the final resource count might be different
Info: Quartus II Analysis & Synthesis was successful. 0 errors, 11 warnings
Info: *******************************************************************
Info: Running Quartus II Fitter
Info: Command: quartus_fit --read_settings_files=off --write_settings_files=off Proj_Hala_Digital_1210606 -c Proj_Hala_Digital_1210606
Info: Selected device EP2C20F484C7 for design "Proj_Hala_Digital_1210606"
```



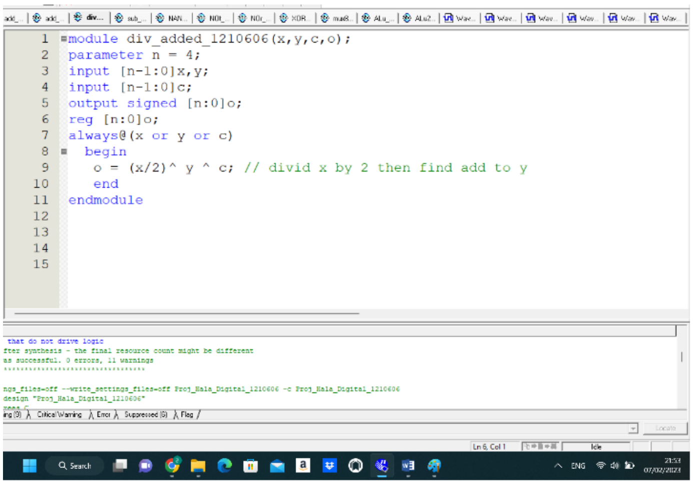
2- 2*(X+Y)



```
module add_multi_1210606(x,y,c,o);
parameter n = 4;
input [n-1:0]x,y;
input [n-1:0]c;
output signed [n:0]o;
reg [n:0]o;
always@(x or y or c)
  begin
    o = (x ^ y ^ c)*2; //find the sum of x and y then multiplay with 2
  end
endmodule
```

```
Warning: Design contains 3 input pin(s) that do not drive logic
Info: Implemented 20 device resources after synthesis - the final resource count might be different
Info: Quartus II Analysis & Synthesis was successful. 0 errors, 11 warnings
Info: *******************************************************************
Info: Running Quartus II Fitter
Info: Command: quartus_fit --read_settings_files=off --write_settings_files=off Proj_Hala_Digital_1210606 -c Proj_Hala_Digital_1210606
Info: Selected device EP2C20F484C7 for design "Proj_Hala_Digital_1210606"
```

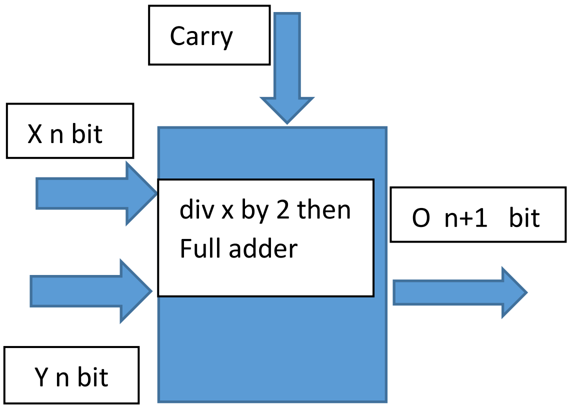
3-(X/2)+Y



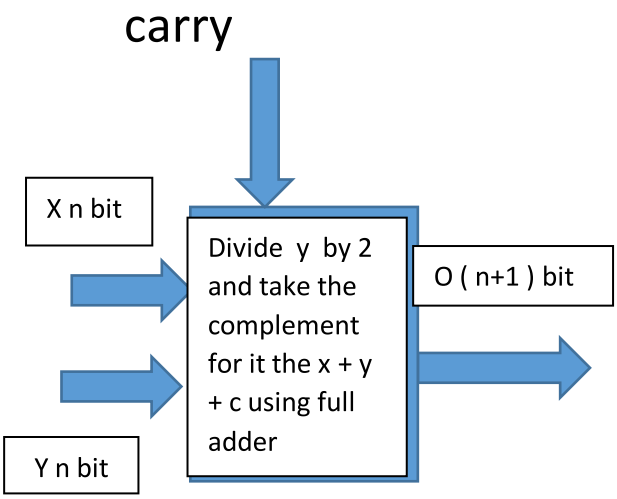
```verilog
module div_added_1210606(x,y,c,o);
parameter n = 4;
input [n-1:0]x,y;
input [n-1:0]c;
output signed [n:0]o;
reg [n:0]o;
always@(x or y or c)
  begin
   o = (x/2)^ y ^ c; // divid x by 2 then find add to y
   end
endmodule
```

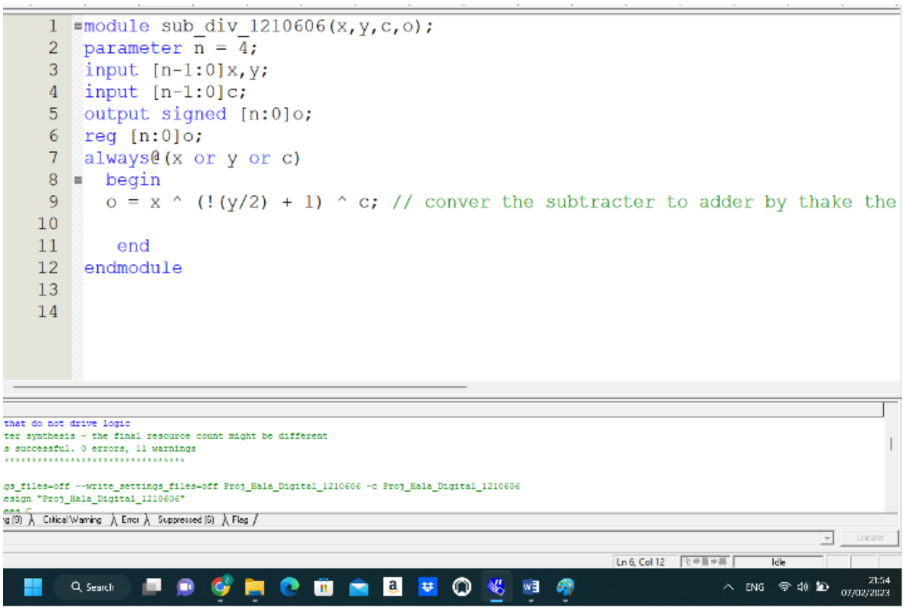
```
 that do not drive logic
fter synthesis - the final resource count might be different
as successful. 0 errors, 11 warnings
********************************

ngs_files=off --write_settings_files=off Proj_Hala_Digital_1210606 -c Proj_Hala_Digital_1210606
design "Proj_Hala_Digital_1210606"
```



4-X-(Y/2)


carry
X n bit
Divide y by 2 and take the complement for it the x + y + c using full adder
O (n+1) bit
Y n bit



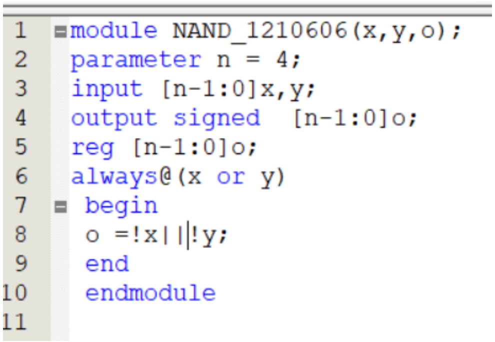
```verilog
module sub_div_1210606(x,y,c,o);
parameter n = 4;
input [n-1:0]x,y;
input [n-1:0]c;
output signed [n:0]o;
reg [n:0]o;
always@(x or y or c)
  begin
  o = x ^ (!(y/2) + 1) ^ c; // conver the subtracter to adder by thake the

   end
endmodule
```

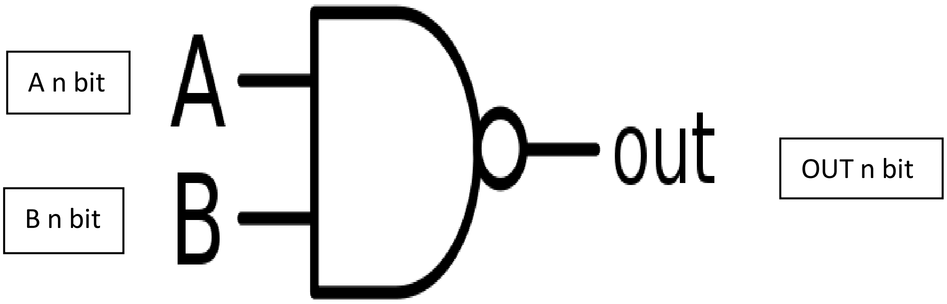
```
that do not drive logic
ter synthesis - the final resource count might be different
s successful. 0 errors, 11 warnings
********************************

gs_files=off --write_settings_files=off Proj_Hala_Digital_1210606 -c Proj_Hala_Digital_1210606
esign "Proj_Hala_Digital_1210606"
```

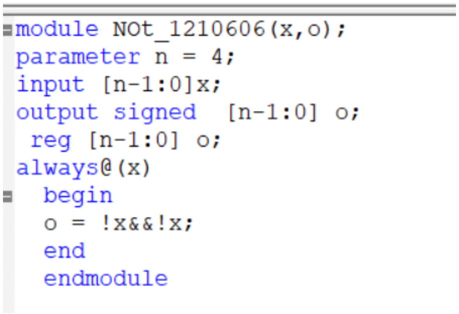
5- x NAND y



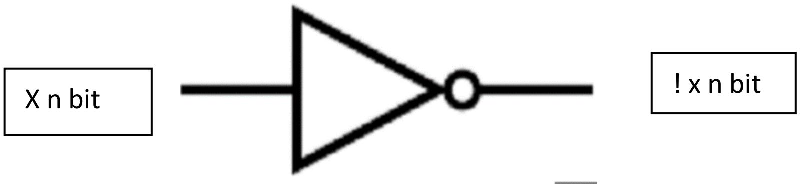
```
module NAND_1210606(x,y,o);
parameter n = 4;
input [n-1:0]x,y;
output signed  [n-1:0]o;
reg [n-1:0]o;
always@(x or y)
 begin
 o =!x||!y;
 end
 endmodule
```



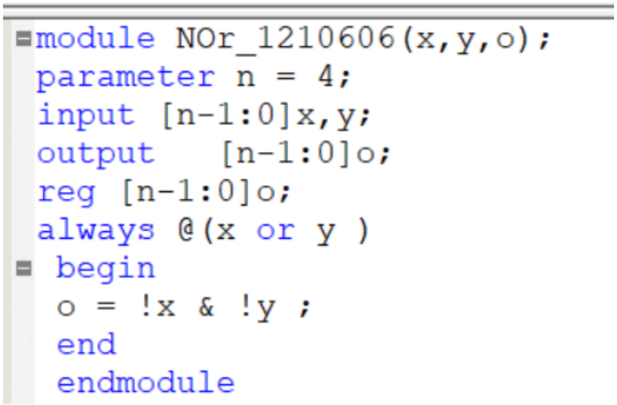
6- NOT(X)



```
module NOt_1210606(x,o);
parameter n = 4;
input [n-1:0]x;
output signed  [n-1:0] o;
 reg [n-1:0] o;
always@(x)
  begin
  o = !x&&!x;
  end
  endmodule
```



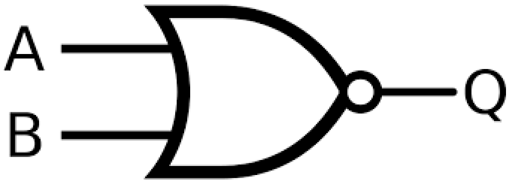
7- X NOR Y



```
module NOr_1210606(x,y,o);
parameter n = 4;
input [n-1:0]x,y;
output    [n-1:0]o;
reg [n-1:0]o;
always @(x or y )
 begin
 o = !x & !y ;
 end
 endmodule
```

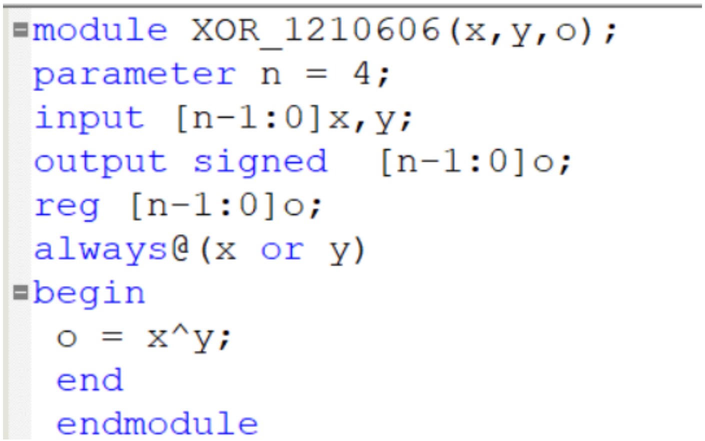
A n bit

B n bit



Q n bit

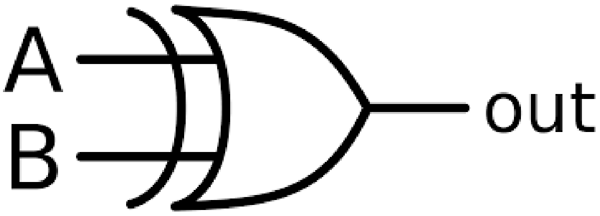
8- x XOR y



```
module XOR_1210606(x,y,o);
parameter n = 4;
input [n-1:0]x,y;
output signed  [n-1:0]o;
reg [n-1:0]o;
always@(x or y)
begin
 o = x^y;
 end
 endmodule
```

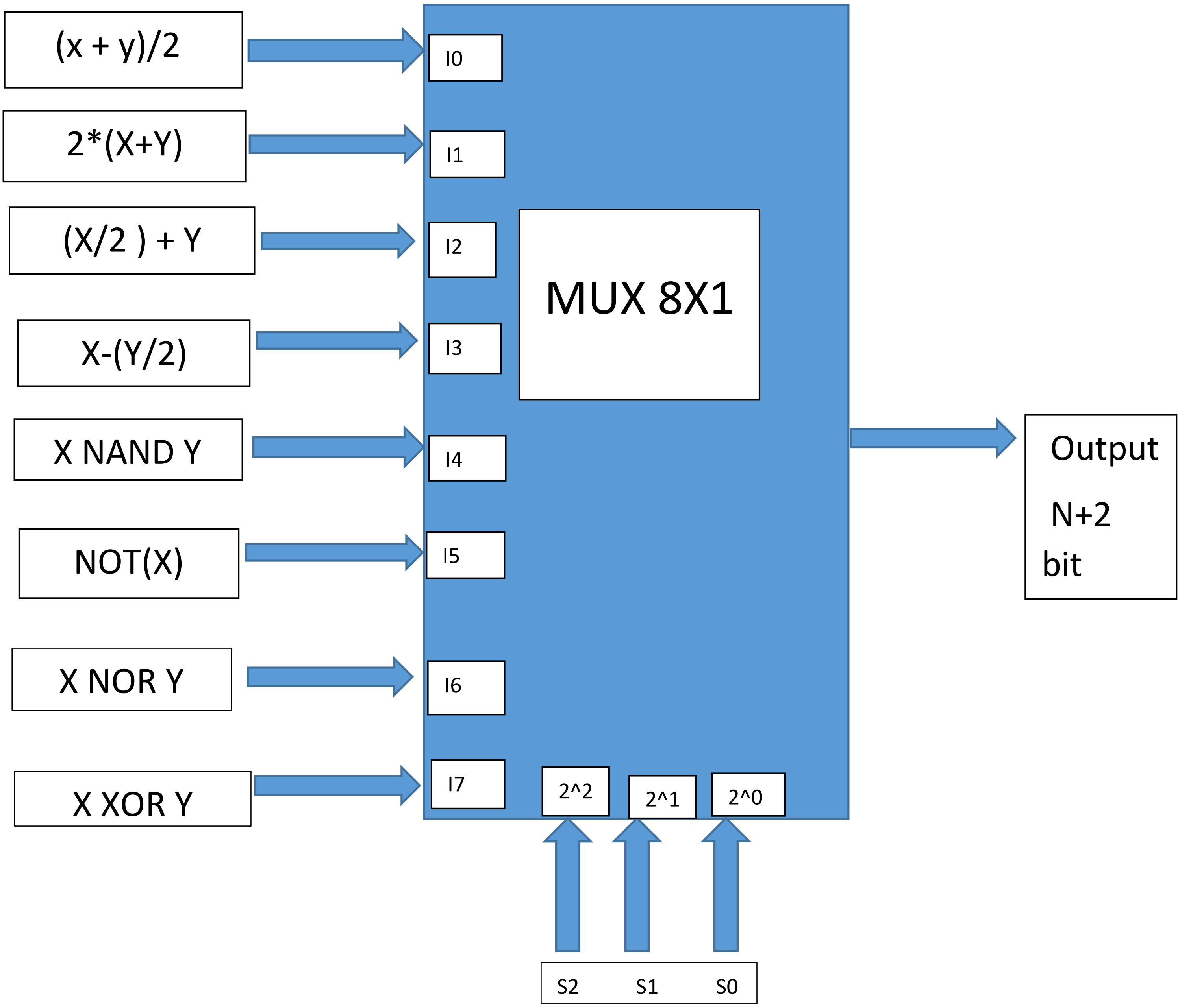
A n bit

B n bit



Out n bit

(9- mux 8 to 1) ***note : the impliment of the ALU in page 13 in this report .**



If the selection is 000 then the mux do the first operation (x + y)/2

If the selection is 001 then the mux do the second operation 2*(x + y)

If the selection is 010 then the mux do the third operation (x/2) + y

If the selection is 011 then the mux do the fourth operation x – (y/2)

If the selection is 100 then the mux do the operation number 5 x NAND y

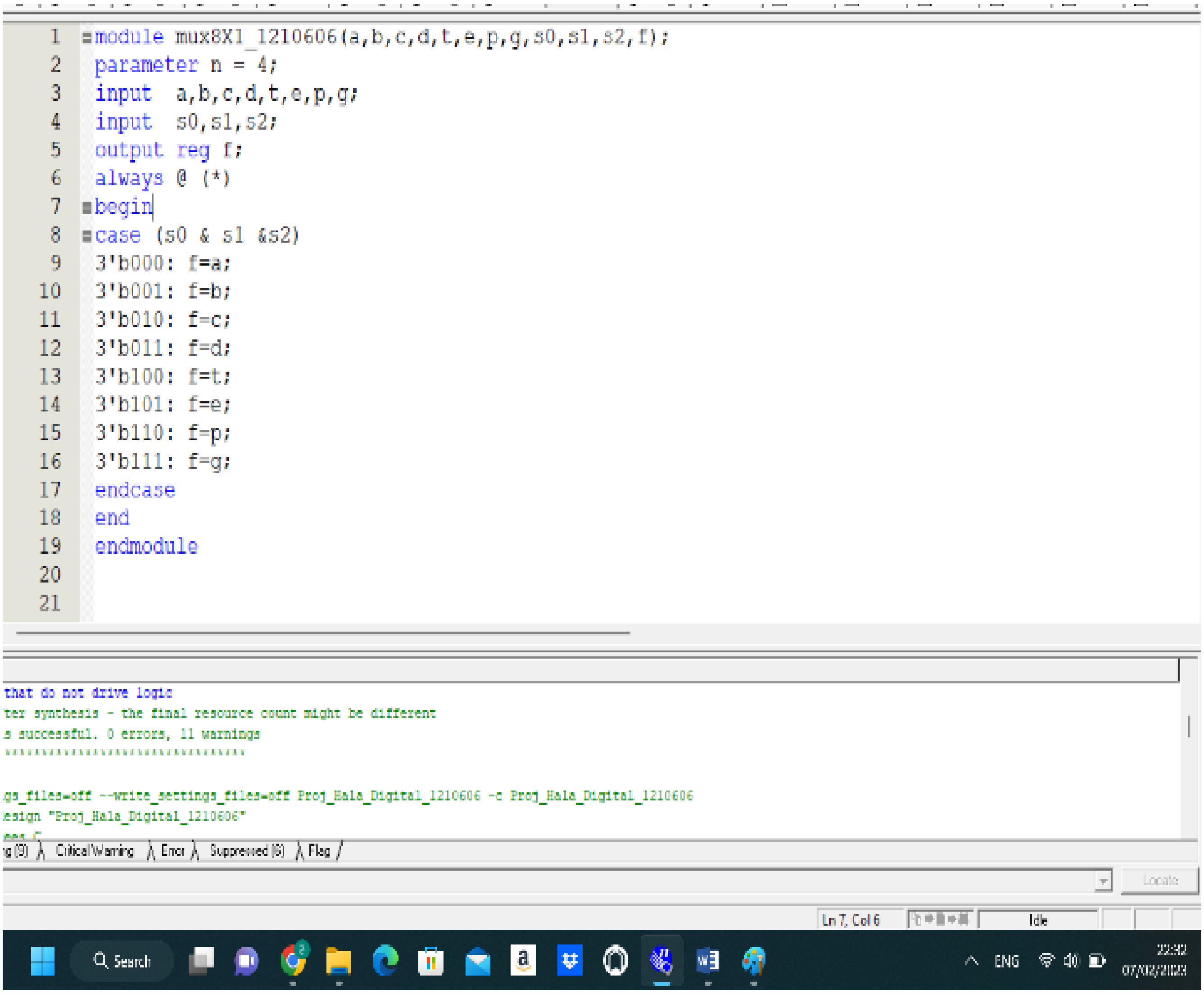
If the selection is 101 then the mux do the operation number 6 not (x)

If the selection is 110 then the mux do the operation number 7 x nor y

If the selection is 111 then the mux do the operation number 8 x xor y

Picture for (ALU1 , ALU2 , MUX8X1) code :

MUX8X1 code

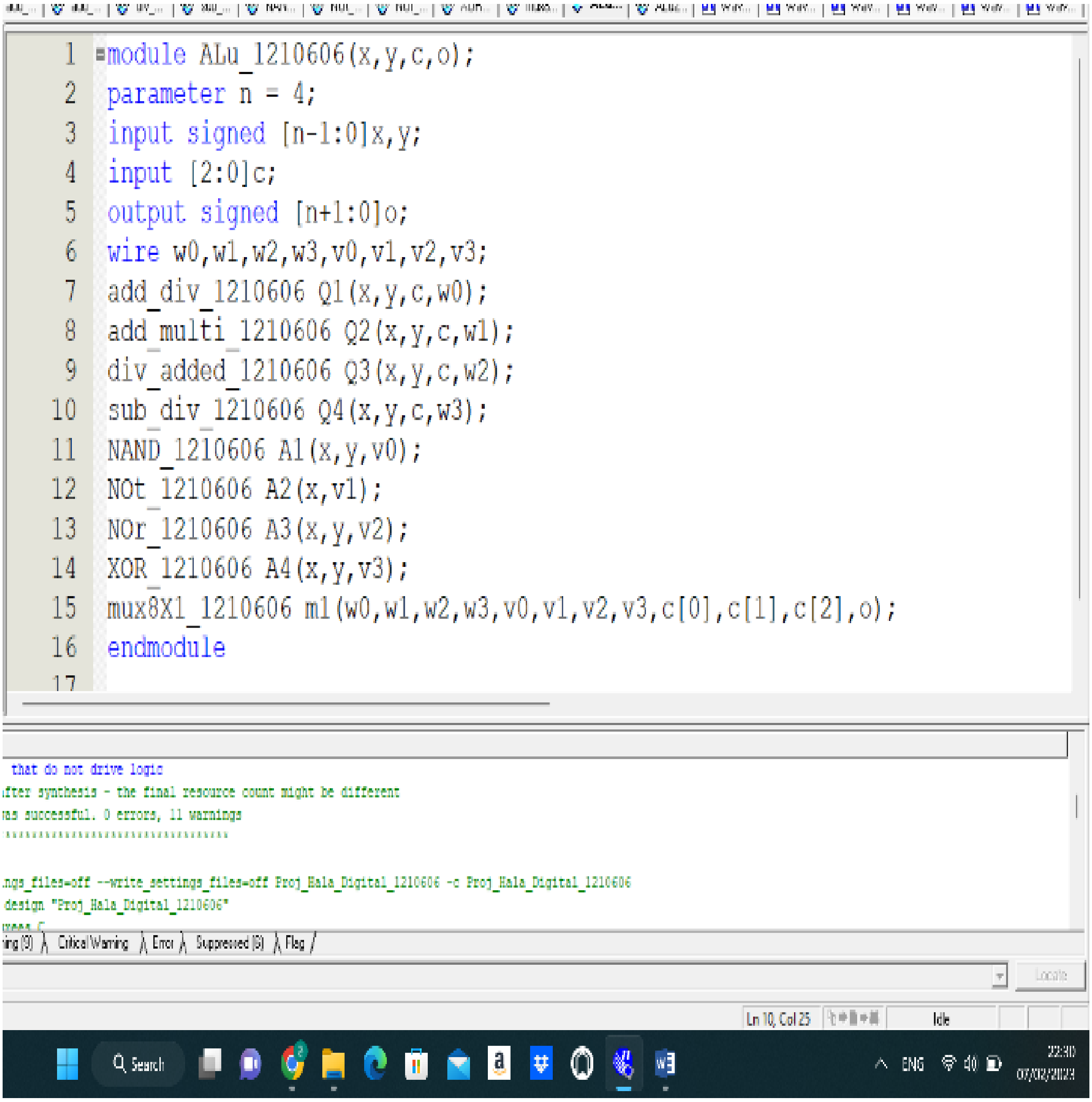


```
module mux8X1_1210606(a,b,c,d,t,e,p,g,s0,s1,s2,f);
parameter n = 4;
input  a,b,c,d,t,e,p,g;
input  s0,s1,s2;
output reg f;
always @ (*)
begin
case  (s0 & s1 &s2)
3'b000: f=a;
3'b001: f=b;
3'b010: f=c;
3'b011: f=d;
3'b100: f=t;
3'b101: f=e;
3'b110: f=p;
3'b111: f=g;
endcase
end
endmodule
```

```
that do not drive logic
ter synthesis - the final resource count might be different
s successful. 0 errors, 11 warnings
*********************************

gs_files=off --write_settings_files=off Proj_Hala_Digital_1210606 -c Proj_Hala_Digital_1210606
esign "Proj_Hala_Digital_1210606"
```

ALU CODE (STRUCTURAL)

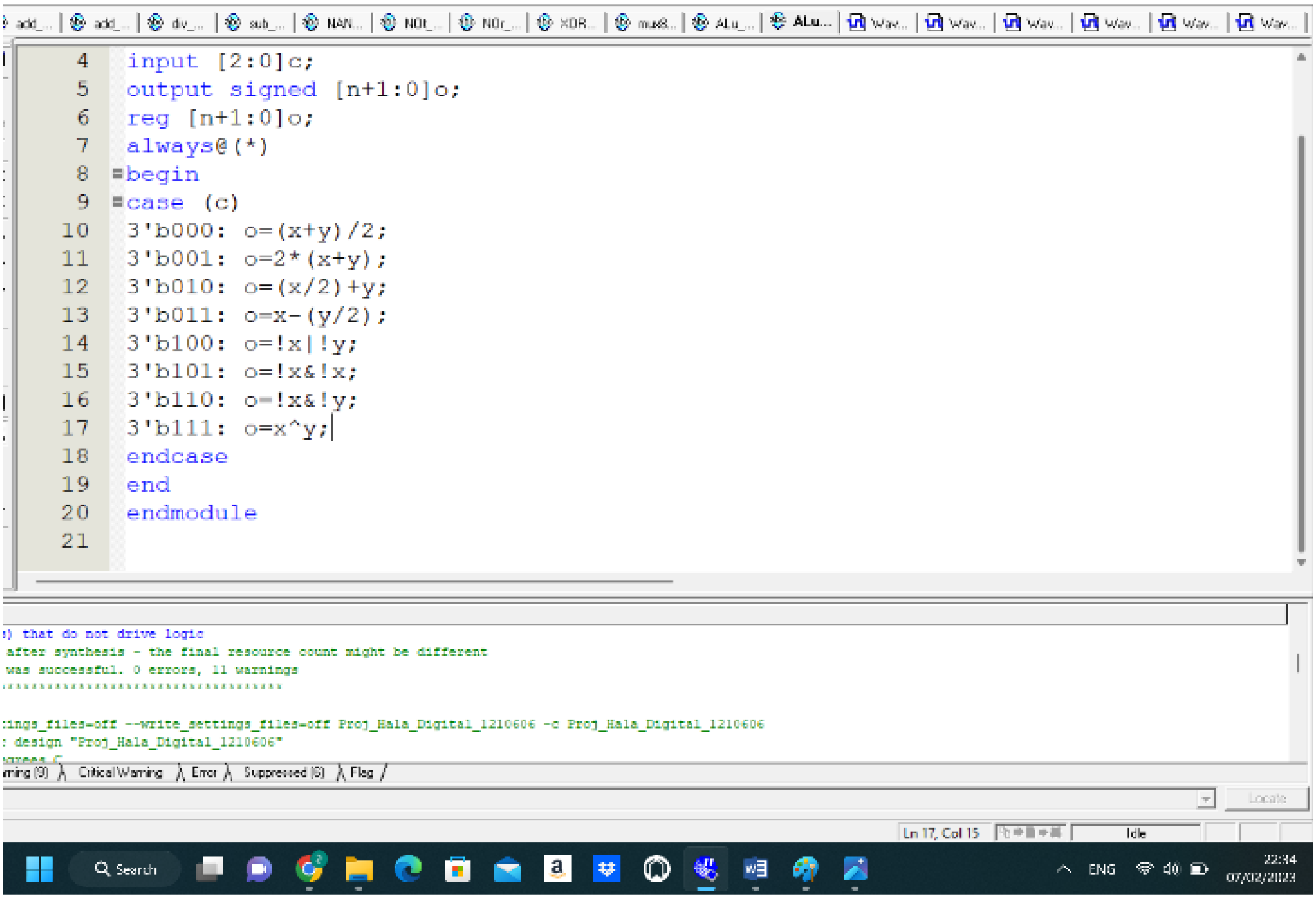


```
module ALu_1210606(x,y,c,o);
parameter n = 4;
input signed [n-1:0]x,y;
input [2:0]c;
output signed [n+1:0]o;
wire w0,w1,w2,w3,v0,v1,v2,v3;
add_div_1210606 Q1(x,y,c,w0);
add_multi_1210606 Q2(x,y,c,w1);
div_added_1210606 Q3(x,y,c,w2);
sub_div_1210606 Q4(x,y,c,w3);
NAND_1210606 A1(x,y,v0);
NOt_1210606 A2(x,v1);
NOr_1210606 A3(x,y,v2);
XOR_1210606 A4(x,y,v3);
mux8X1_1210606 m1(w0,w1,w2,w3,v0,v1,v2,v3,c[0],c[1],c[2],o);
endmodule
```

```
 that do not drive logic
after synthesis - the final resource count might be different
was successful. 0 errors, 11 warnings
*********************************

ngs_files=off --write_settings_files=off Proj_Hala_Digital_1210606 -c Proj_Hala_Digital_1210606
design "Proj_Hala_Digital_1210606"
rees (
```

ALU CODE (BEHAVEIORL)

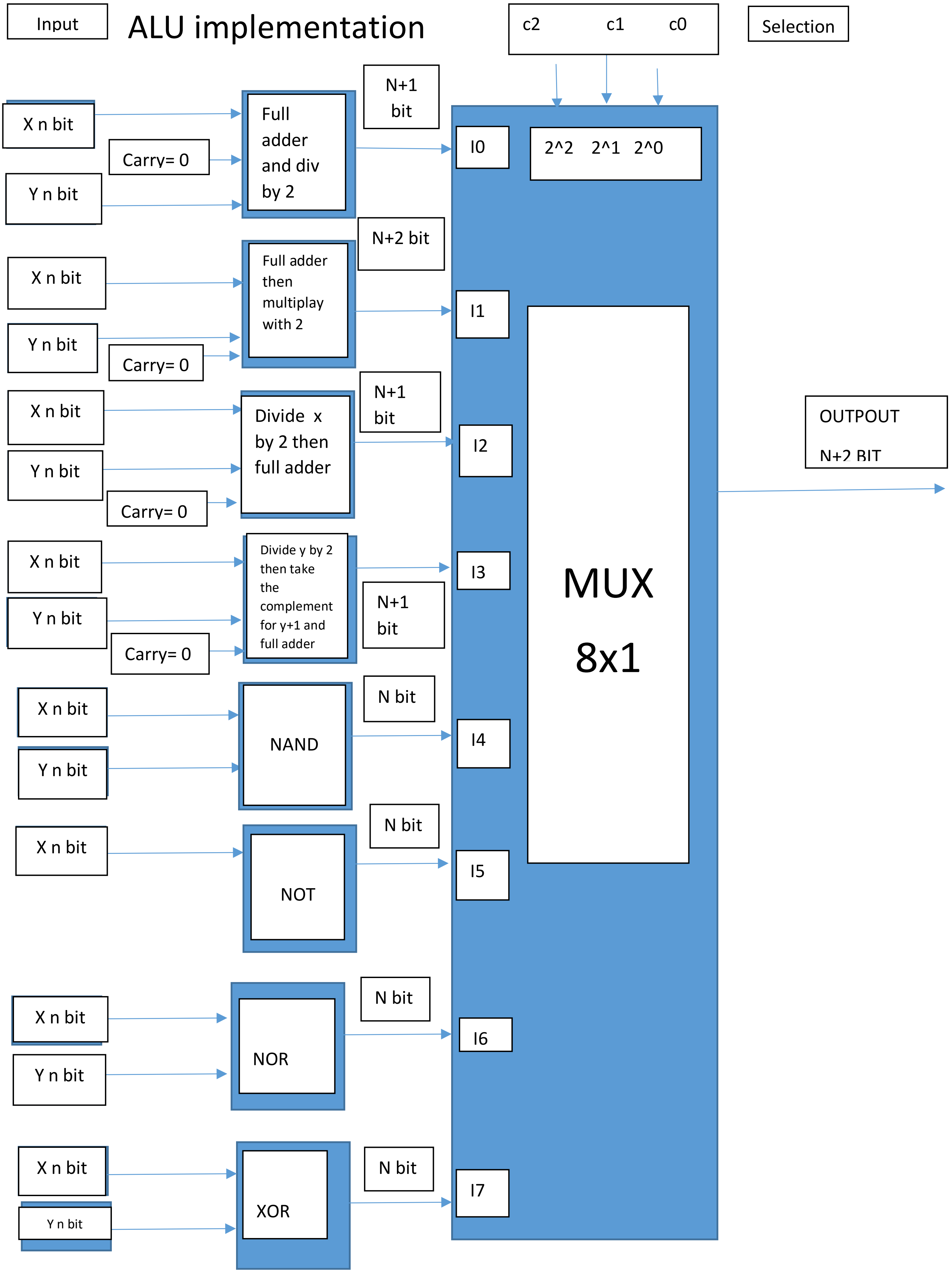


```
input [2:0]c;
output signed [n+1:0]o;
reg [n+1:0]o;
always@(*)
begin
case (c)
3'b000: o=(x+y)/2;
3'b001: o=2*(x+y);
3'b010: o=(x/2)+y;
3'b011: o=x-(y/2);
3'b100: o=!x|!y;
3'b101: o=!x&!x;
3'b110: o=!x&!y;
3'b111: o=x^y;
endcase
end
endmodule
```

```
s) that do not drive logic
 after synthesis - the final resource count might be different
 was successful. 0 errors, 11 warnings
*****************************************

:ings_files=off --write_settings_files=off Proj_Hala_Digital_1210606 -c Proj_Hala_Digital_1210606
: design "Proj_Hala_Digital_1210606"
```

ALU implementation



e

1 2 1 0 6 0 6 (mynumber)

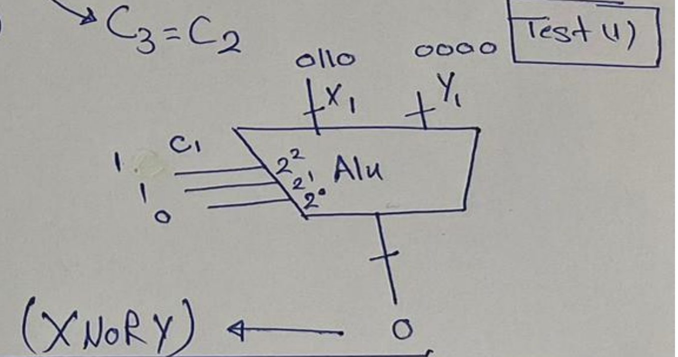
1 C_2 Y_2 X_2 C_1 Y_1 X_1

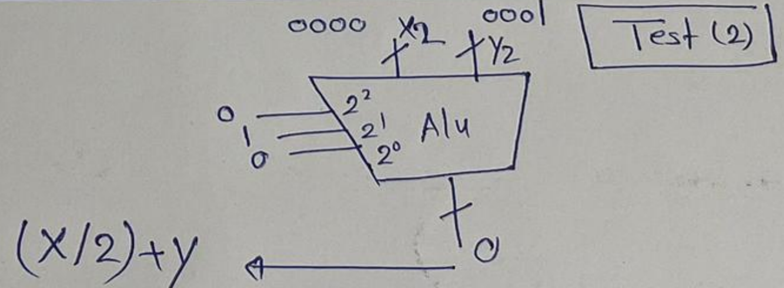
Test	X	Y	C	O
1	$X_1 = 6$	$Y_1 = 0$	$C_1 = 6$	X NoR Y
2	$X_2 = 0$	$Y_2 = 1$	$C_2 = 2$	$(X/2)+Y$
3	$X_3 = -6$	$Y_3 = 0$	$C_3 = 2$	$(X/2)+y$

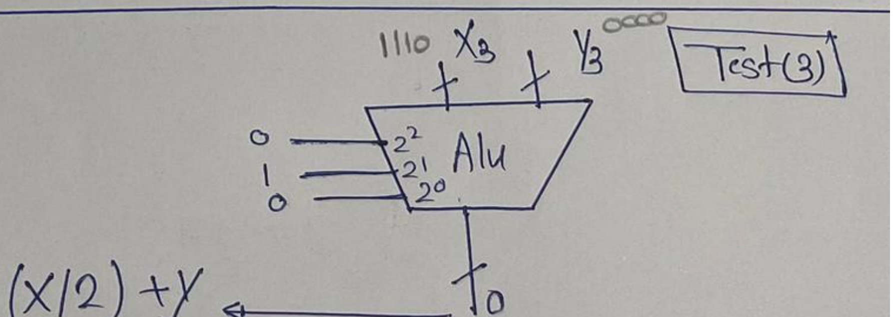
$X_3 = -X_1$ $Y_3 = +0$ $C_3 = C_2$

(mux 8x1)

$2^{③}$ → number of selection







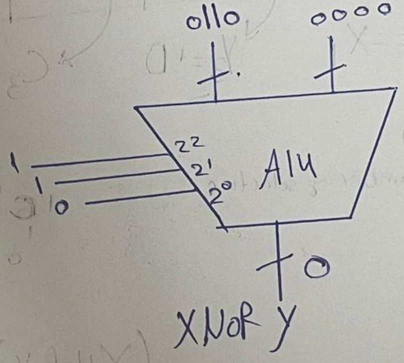
$\boxed{9}$

1210606

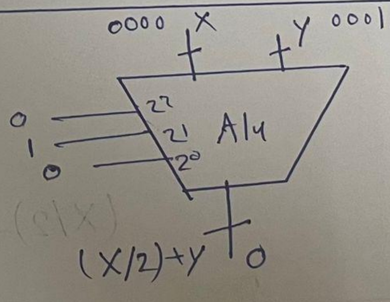
1 C_2 Y_2 X_2 C_1 Y_1 X_1

Test	X	Y	C	O
1	$X_1=6$	$Y_1=0$	$C_1=6$	X NoR Y
2	$X_2=0$	$Y_2=1$	$C_2=2$	(X/2)+Y
3	$X_3=-6$	$Y_3=0$	$C_3=2$	(X/2)+Y

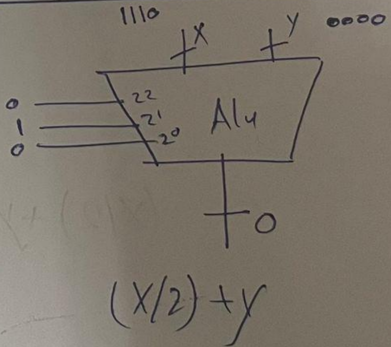
Test (1):



Test $\boxed{2}$



Test $\boxed{3}$



e) (5 points) 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:

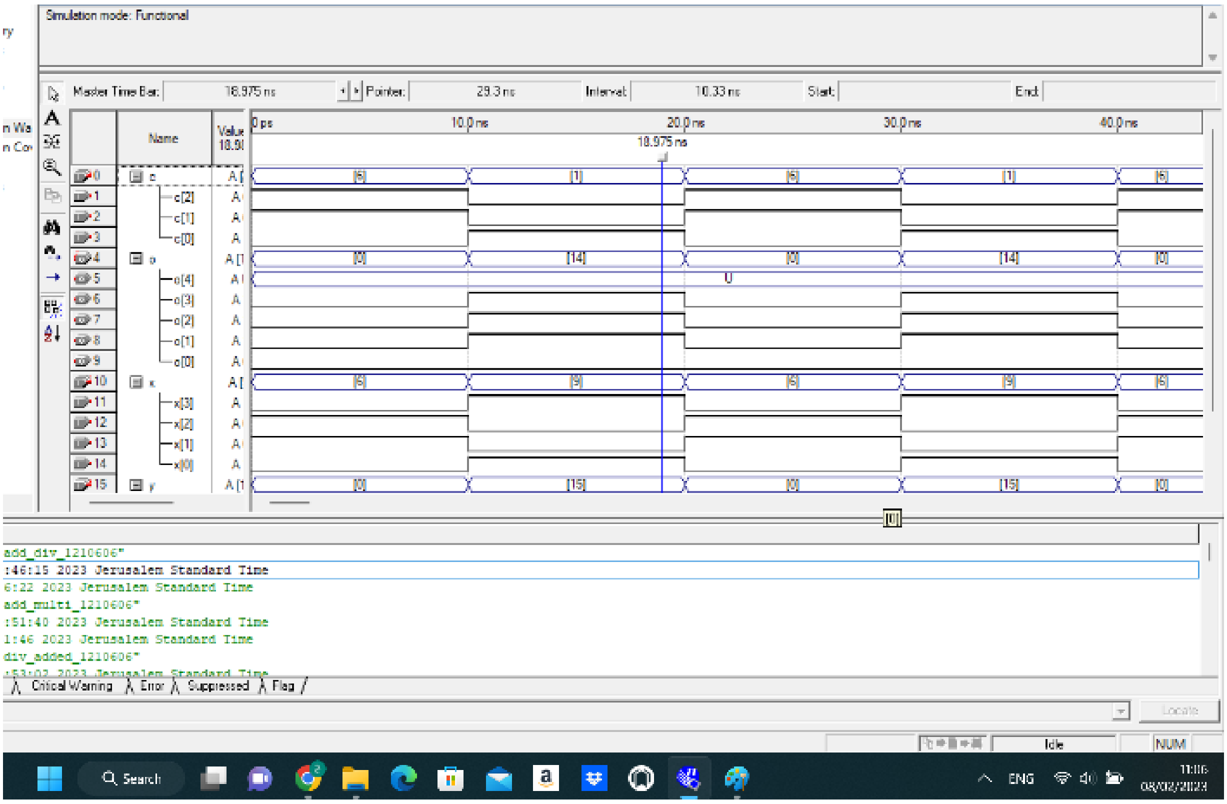
Test 1 :

X1 = 6

Y1 = 0

C1 = 6

Output : X NOR Y



Simulation mode: Functional

Master Time Bar: 18.975 ns

Pointer: 29.3 ns

Interval: 10.33 ns

Start:

End:

```
add_div_1210606"
:46:15 2023 Jerusalem Standard Time
6:22 2023 Jerusalem Standard Time
add_multi_1210606"
:51:40 2023 Jerusalem Standard Time
1:46 2023 Jerusalem Standard Time
div_added_1210606"
:53:02 2023 Jerusalem Standard Time
```

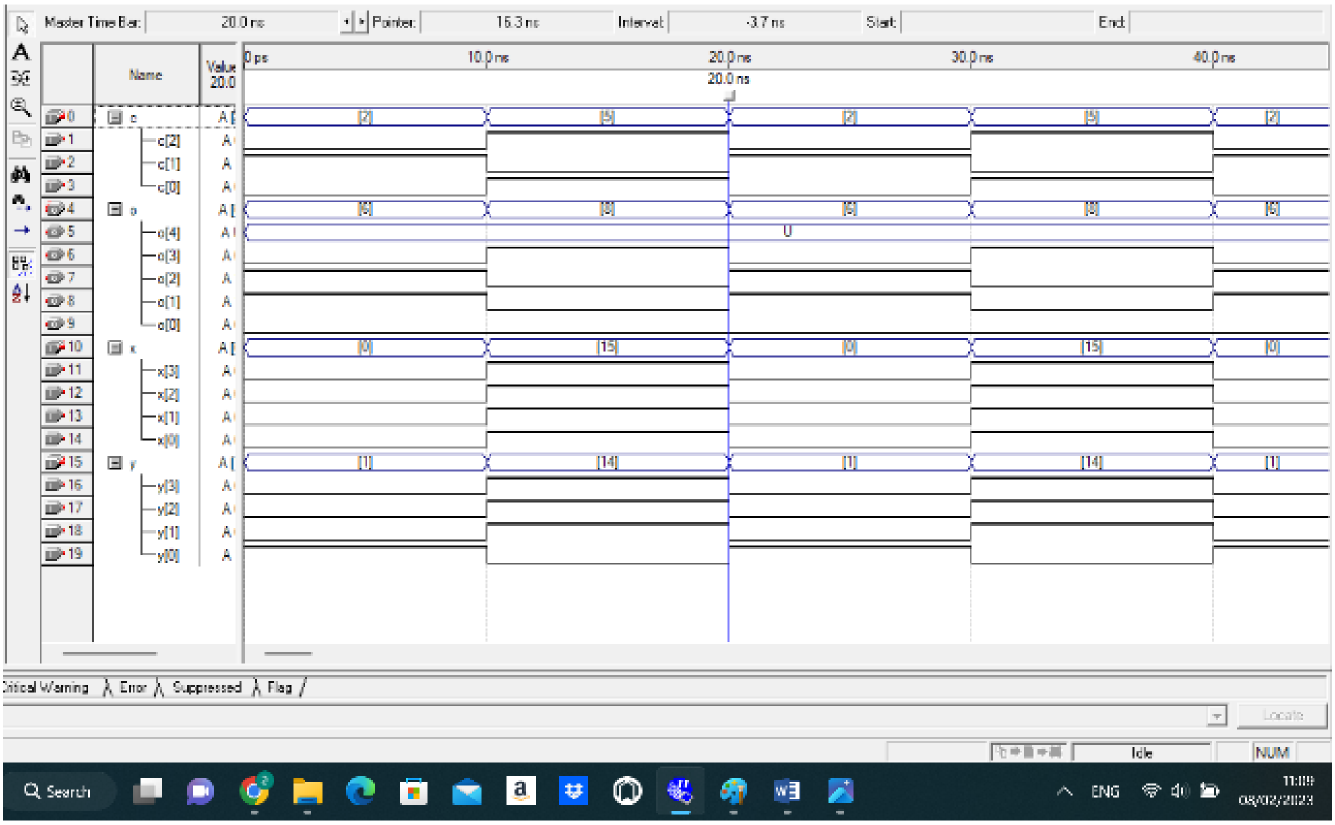
Test 2 :

X2 = 0

Y2 = 1

C1=6

Output : (x / 2) + y



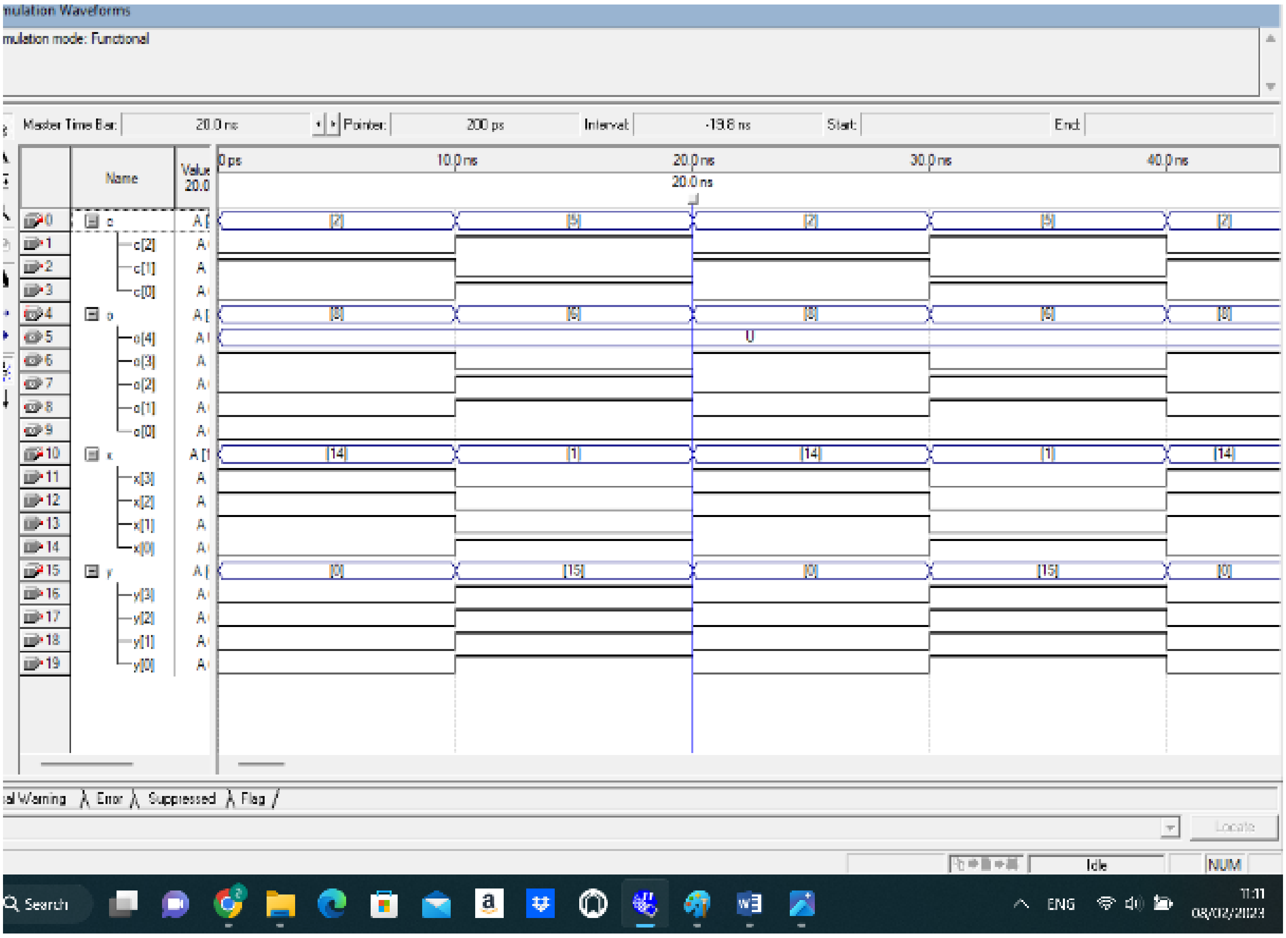
Test 3 :

X3 = -6

Y3 = 0

C3 = 2

Output = (x / 2) + y



g) (5 points) 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:

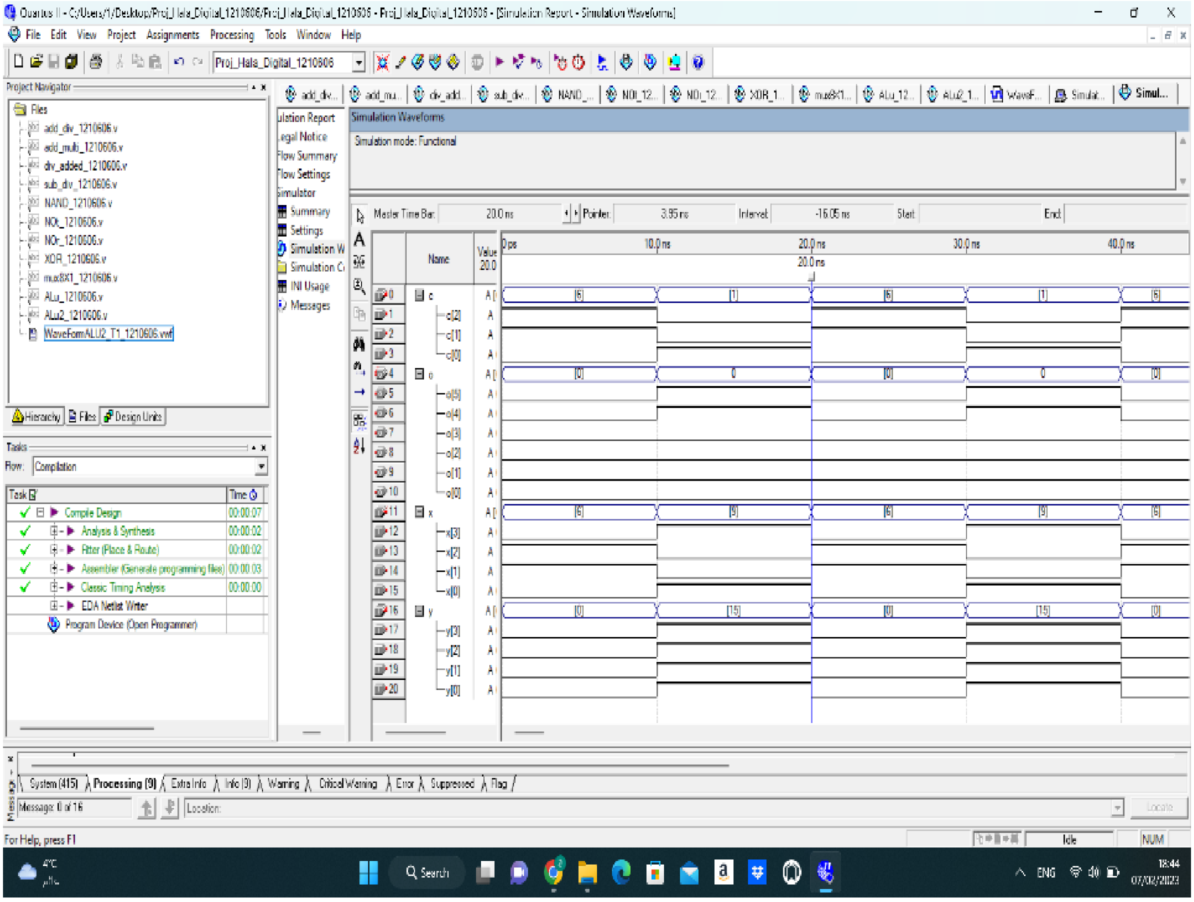
Test 1 :

x1 = 6

y1 = 0

c1 = 6

Output : x NOR y


Quartus II - C:/Users/1/Desktop/Proj_Hala_Digital_1210606/Proj_Hala_Digital_1210606 - Proj_Hala_Digital_1210606 - [Simulation Report - Simulation Waveforms]

Simulation Waveforms

Simulation mode: Functional

Master Time Bar: 20.0 ns Pointer: 3.95 ns Interval: -16.05 ns Start: End:

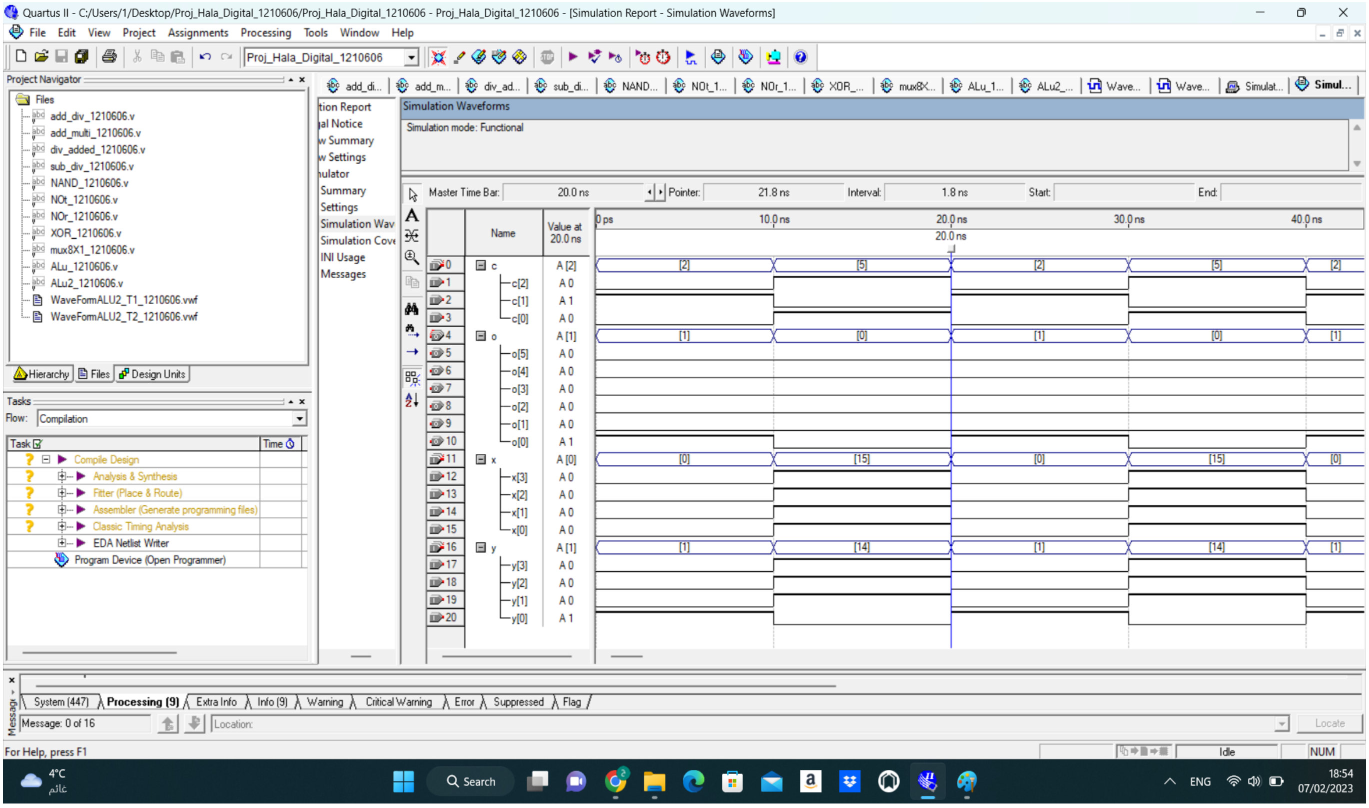
Test 2 :

X2 = 0

Y2=1

C2=2

Output : (x/2) + y



Quartus II - C:/Users/1/Desktop/Proj_Hala_Digital_1210606/Proj_Hala_Digital_1210606 - Proj_Hala_Digital_1210606 - [Simulation Report - Simulation Waveforms]

Simulation Waveforms

Simulation mode: Functional

Master Time Bar: 20.0 ns Pointer: 21.8 ns Interval: 1.8 ns Start: End:

	Name	Value at 20.0 ns
0	c	A [2]
1	c[2]	A 0
2	c[1]	A 1
3	c[0]	A 0
4	o	A [1]
5	o[5]	A 0
6	o[4]	A 0
7	o[3]	A 0
8	o[2]	A 0
9	o[1]	A 0
10	o[0]	A 1
11	x	A [0]
12	x[3]	A 0
13	x[2]	A 0
14	x[1]	A 0
15	x[0]	A 0
16	y	A [1]
17	y[3]	A 0
18	y[2]	A 0
19	y[1]	A 0
20	y[0]	A 1

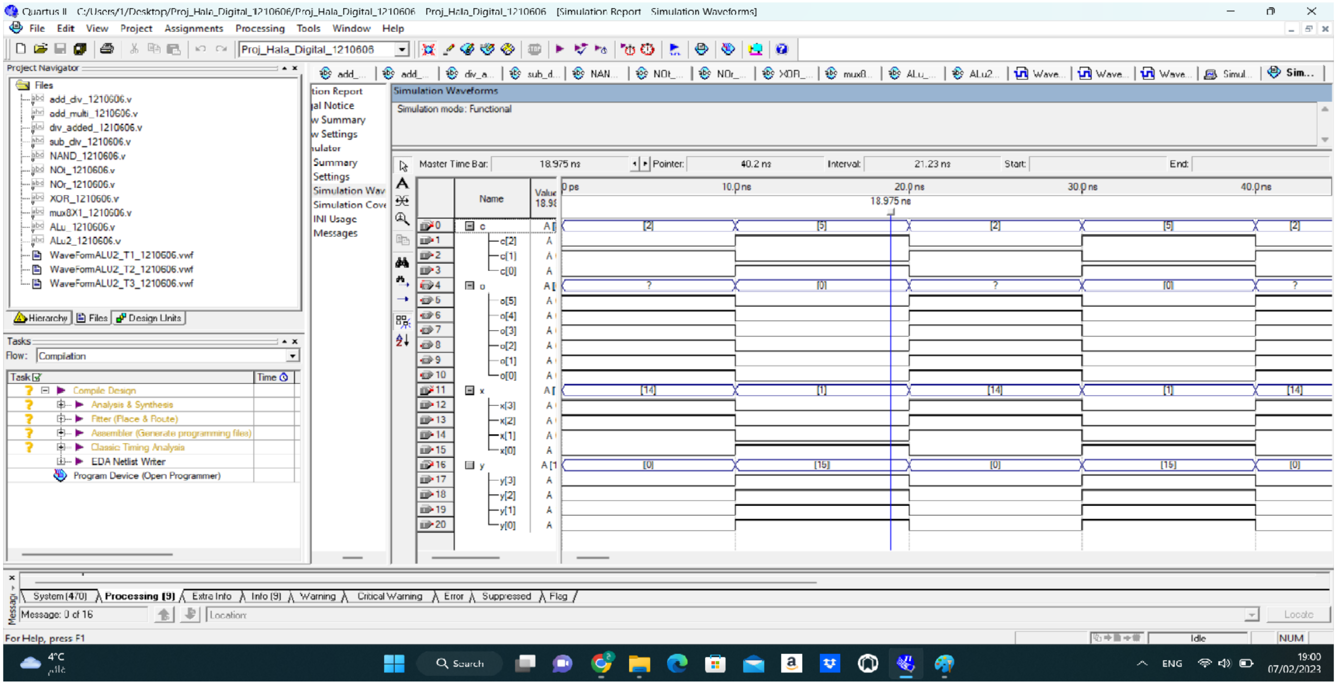
Test 3:

X3=-6

Y3=0

C3=2

Output : (x/2) + y


Quartus II C:/Users/1/Desktop/Proj_Hala_Digital_1210606/Proj_Hala_Digital_1210606 Proj_Hala_Digital_1210606 [Simulation Report Simulation Waveforms]
File Edit View Project Assignments Processing Tools Window Help
Proj_Hala_Digital_1210606
Project Navigator
Files
add_div_1210606.v
add_multi_1210606.v
div_added_1210606.v
sub_div_1210606.v
NAND_1210606.v
NOt_1210606.v
NOr_1210606.v
XOR_1210606.v
mux8X1_1210606.v
ALu_1210606.v
ALu2_1210606.v
WaveFormALU2_T1_1210606.vwf
WaveFormALU2_T2_1210606.vwf
WaveFormALU2_T3_1210606.vwf
Hierarchy Files Design Units
Tasks
Flow: Compilation
Task Time
Compile Design
Analysis & Synthesis
Fitter (Place & Route)
Assembler (Generate programming files)
Classic Timing Analysis
EDA Netlist Writer
Program Device (Open Programmer)
tion Report
al Notice
w Summary
w Settings
ulator
Summary
Settings
Simulation Wav
Simulation Cove
INI Usage
Messages
Simulation Waveforms
Simulation mode: Functional
Master Time Bar: 18.975 ns
Pointer: 40.2 ns
Interval: 21.23 ns
Start:
End:
Name
Value 18.98
0 ps
10.0 ns
20.0 ns
30.0 ns
40.0 ns
18.975 ns
0 c [2] [5] [2] [5] [2]
1 c[2]
2 c[1]
3 c[0]
4 o ? [0] ? [0] ?
5 o[5]
6 o[4]
7 o[3]
8 o[2]
9 o[1]
10 o[0]
11 x [14] [1] [14] [1] [14]
12 x[3]
13 x[2]
14 x[1]
15 x[0]
16 y [0] [15] [0] [15] [0]
17 y[3]
18 y[2]
19 y[1]
20 y[0]
System (470) Processing (9) Extra Info Info (9) Warning Critical Warning Error Suppressed Flag
Message: 0 of 16
Location
Locate
For Help, press F1
Idle
NUM
4°C
Search
ENG
19:00
07/02/2023