



First Semester, 2022/2023

Digital Systems ENCS234 – Verilog Project

MULTIFUNCTION ALU

Name : Qusay Taradeh

ID : 1212508

Part A.

Output O has size $n+2$ bits that's the overflow can never occur, such that: Operation 1 the summation needs $n+1$ bits to avoid occur overflow so first operation output $n+1$ bit and output O size is $n+2$ that's accept $n+1$ bit number.

Operation 2 the multiplication needs $n+2$ bits size output to avoid overflow. Hence we notice that large output bits size results from multiplication so we set the output O $n+2$ bits to accept any result of any operation from 8 operations without occur overflow.

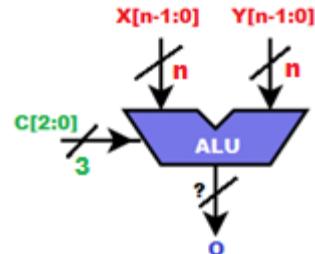
Operation 3 and operation 4 same as operation 1 on the output size hand.

Operations 5,6,7 and 8, these operations outputs same size of input variables; because these are logic operations compares each bit of the variable and set the output bit by bit. Hence the output size is n bits if the input was n bits.

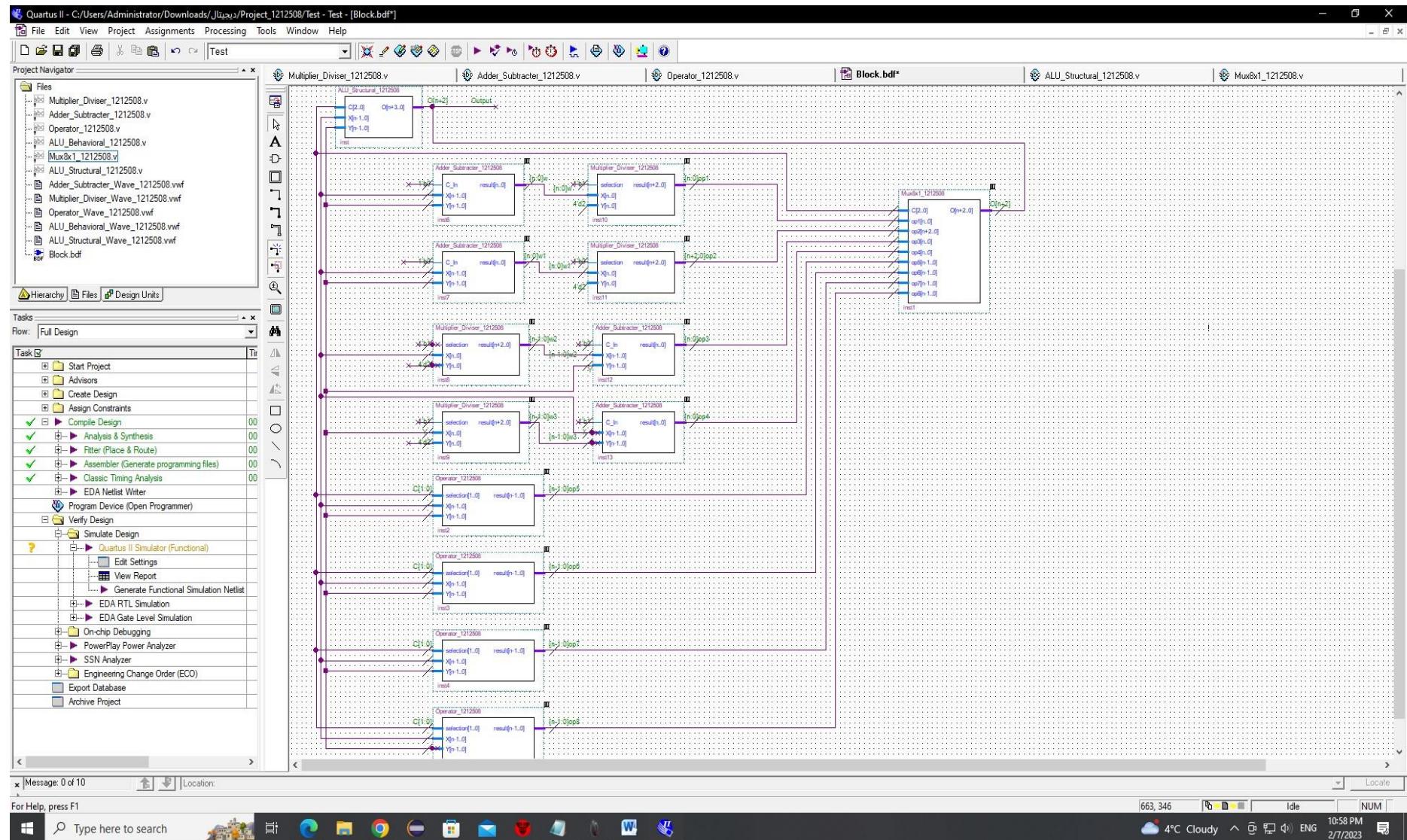
Conclusion : output O must be $n+2$ bit size to accept all operations without occur Overflow in any operation.

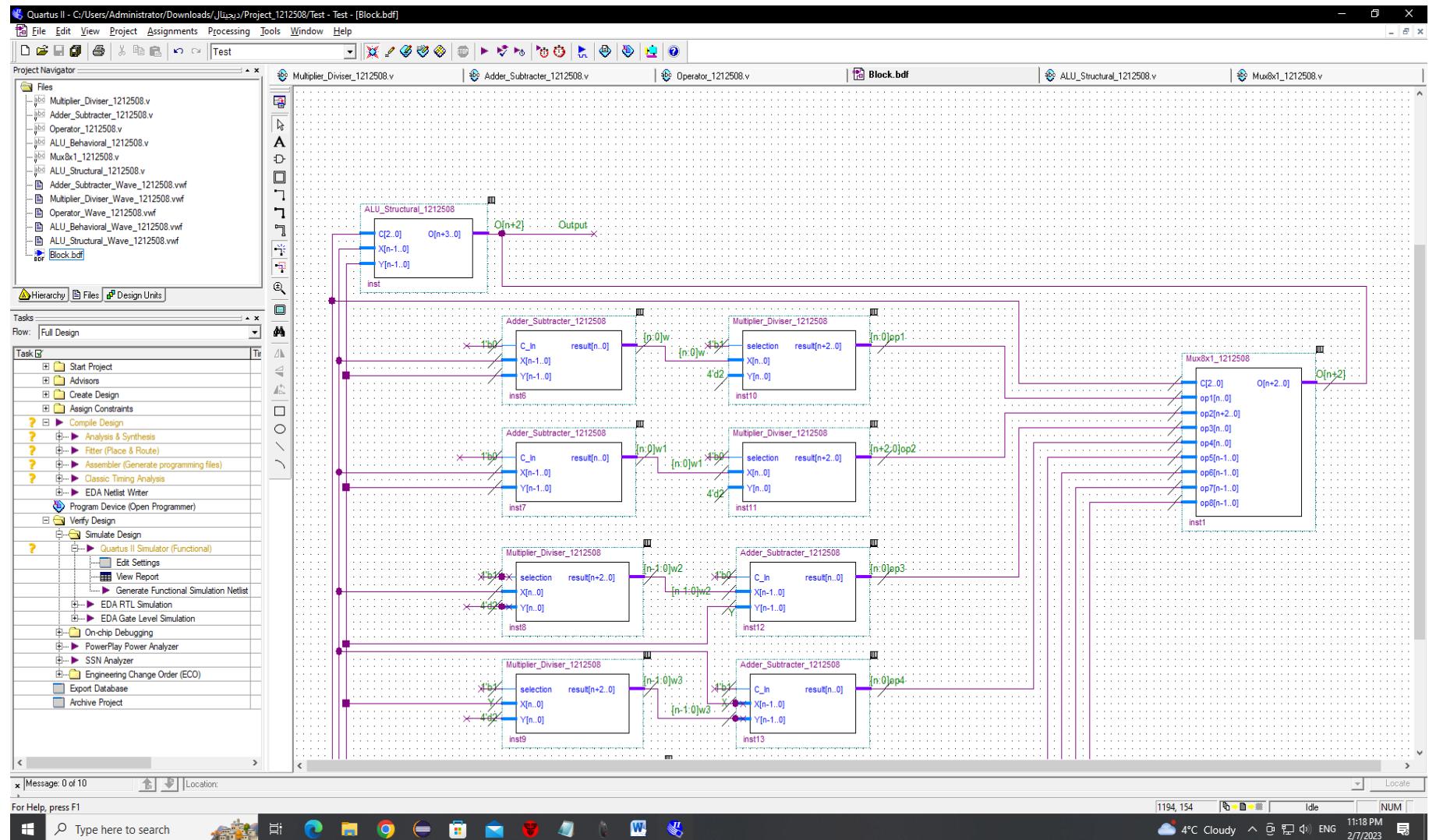
Part B.ALU implementation

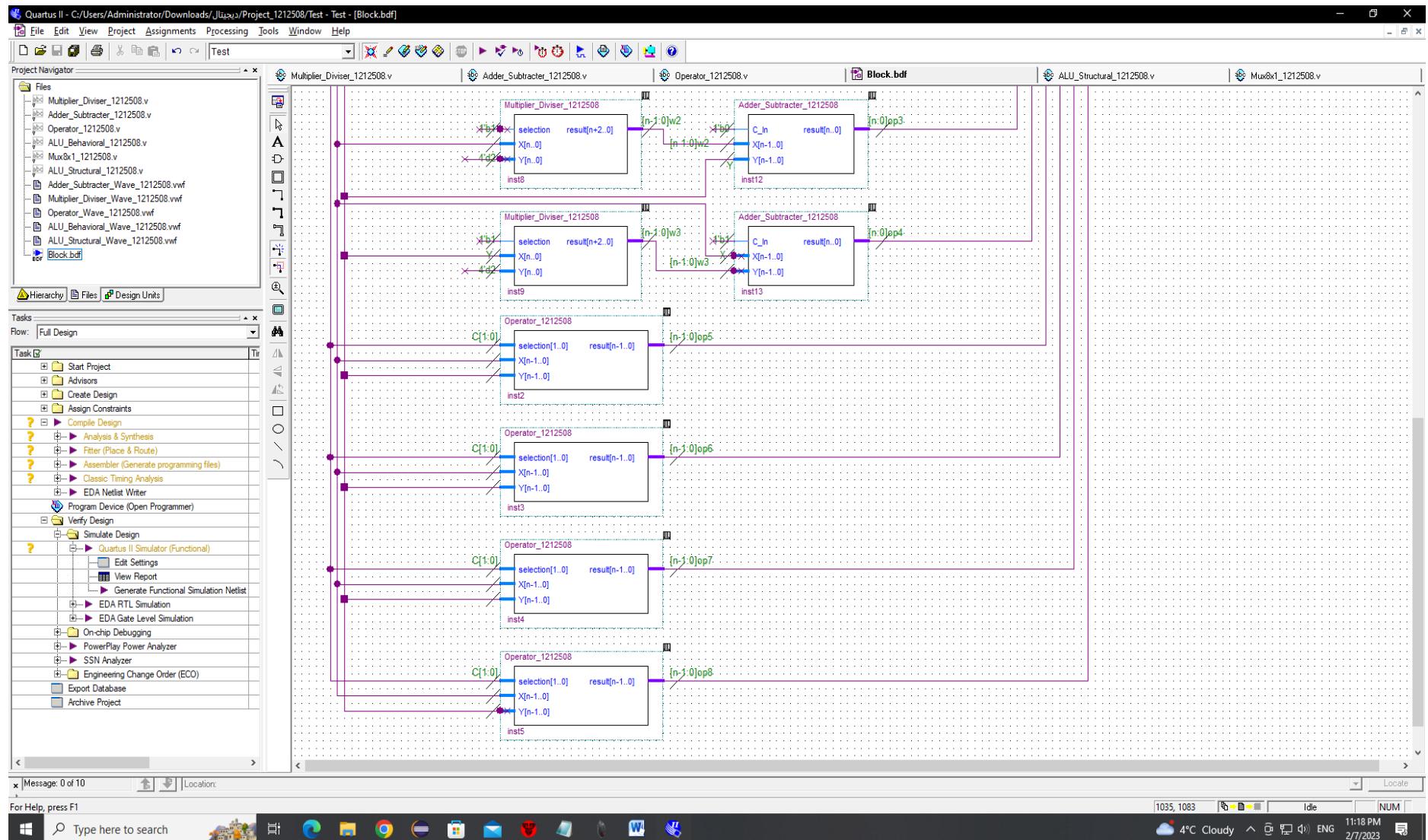
This is the table of ALU followed by ALU implementation :

ALU Function Code (C)	ALU Output (O)	ALU Symbol
000	$(X+Y)/2$	
001	$2*(X+Y)$	
010	$(X/2)+Y$	
011	$X-(Y/2)$	
100	$X \text{ NAND } Y$	
101	$\text{NOT}(X)$	
110	$X \text{ NOR } Y$	
111	$X \text{ XOR } Y$	

Note:First photo is the general MSI ,second and third photos focus on MSI







Part C.

1.Adder_Subtracter_1212508

This module process two operations Summation and Subtraction between two inputs signed numbers X and Y those have n bit , and output the result of size n+1 bit to avoid occur Overflow.

To select the operation input C_In select Summation if it equal zero in binary , also it select Subtraction if it equal one in binary.

Behavioral module of Adder_Subtracter_1212508 in next page ..

Quartus II - C:/Users/Administrator/Downloads/JL4xx3/Project_1212508/Test - Test - [Adder_Subtractor_1212508.v]

File Edit View Project Assignments Processing Tools Window Help

Project Navigator Files Adder_Subtractor_1212508.v Operator_1212508.v Mux8x1_1212508.v ALU_Structural_1212508.v

Hierarchy Files Design Units

Tasks Flow: Full Design

Task

- Start Project
- Advisors
- Create Design
- Assign Constraints
- Compile Design
- Analysis & Synthesis
- Filter (Place & Route)
- Assembler Generate programming files
- Classic Timing Analysis
- EDA Netlist Writer
- Program Device (Open Programmer)
- Verify Design
- Simulate Design
 - Quartus II Simulator (Functional)
 - Edit Settings
 - View Report
 - Generate Functional Simulation Netlist
 - EDA RTL Simulation
 - EDA Gate Level Simulation
 - On-chip Debugging
 - PowerPlay Power Analyzer
 - SSN Analyzer
 - Engineering Change Order (ECO)
 - Export Database
 - Archive Project

Message: 0 of 10 Location:

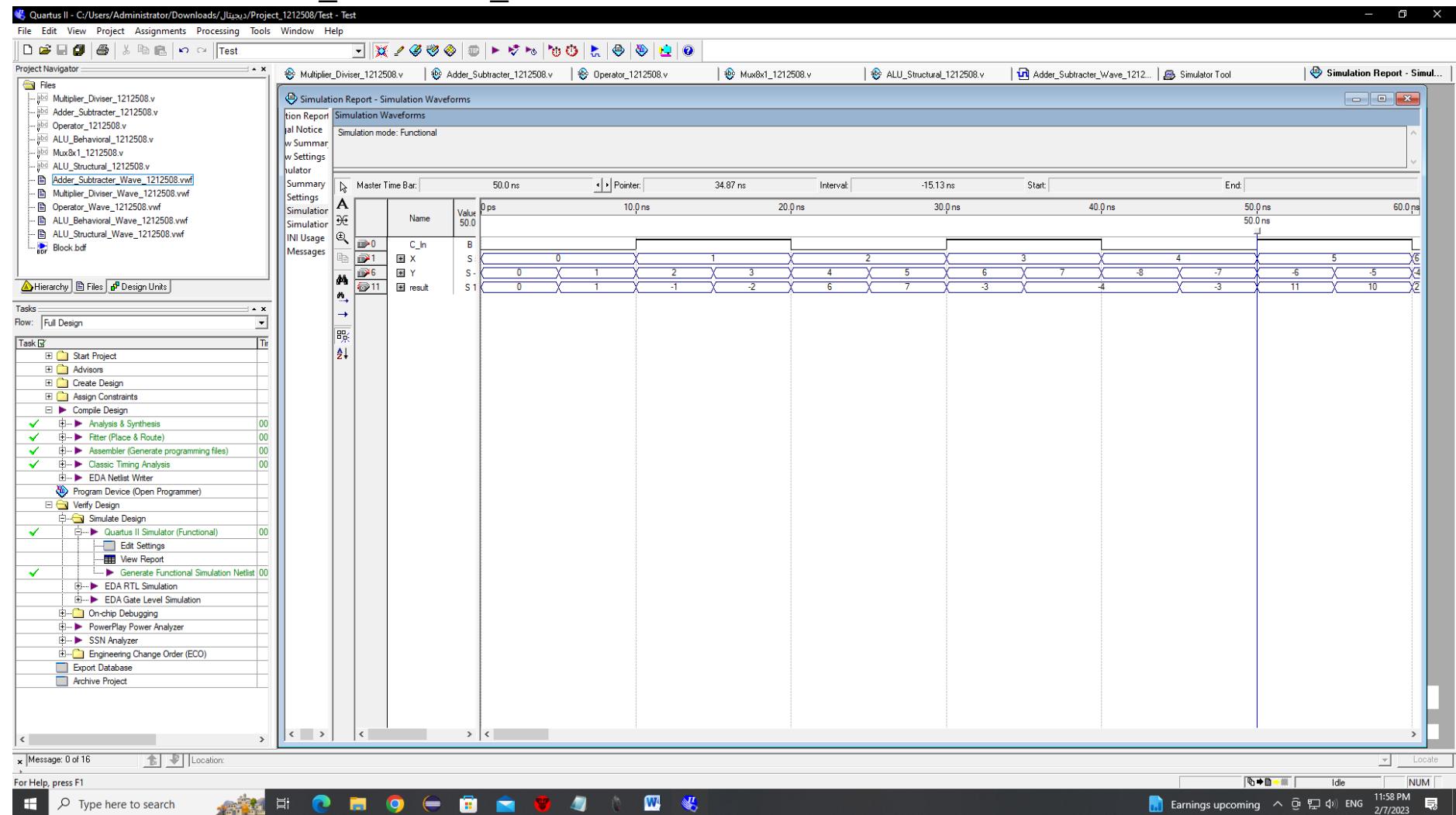
For Help, press F1

Type here to search

Ln 4, Col 53 4°C Cloudy 11:40 PM 2/7/2023

```
1 module Adder_Subtractor_1212508 #(parameter n = 4) (C_In,X,Y,result);
2   input C_In; //determines if the operation sum or sub.
3   input signed [n-1:0]X,Y; //inputs of the system.
4   output reg signed [n:0]result; //result of operation
5
6   always @(*)
7   begin
8     case(C_In)
9       1'b0 : result = X+Y;
10      1'b1 : result = X-Y;
11    endcase
12  end
13 endmodule
14
```

Simulation of Adder_Subtracter_1212508 :



2. Multiplier_Diviser_1212508

This module process two operations Multiplication and Division between two inputs signed numbers X and Y those have $n+1$ bit , and output the result of size $n+2$ bit to avoid occur Overflow.

To select the operation input selection select Multiplication if it equal zero in binary , also it select Division if it equal one in binary.

Behavioral module of Multiplier_Diviser_1212508 in next page ..

Quartus II - C:/Users/Administrator/Downloads/JL4xx/Project_1212508/Test - Test - [Multiplier_Diviser_1212508.v]

File Edit View Project Assignments Processing Tools Window Help

Project Navigator | Multiplier_Diviser_1212508.v | Adder_Subtracter_1212508.v | Operator_1212508.v | Mux8x1_1212508.v | ALU_Structural_1212508.v

Hierarchy Files Design Units

Tasks Full Design

Task

- Start Project
- Advisors
- Create Design
- Assign Constraints
- Compile Design
 - Analysis & Synthesis
 - Filter (Place & Route)
 - Assembler Generate programming files
 - Classic Timing Analysis
 - EDA Netlist Writer
- Program Device (Open Programmer)
- Verify Design
 - Quartus II Simulator (Functional)
 - Edit Settings
 - View Report
 - Generate Functional Simulation Netlist
 - EDA RTL Simulation
 - EDA Gate Level Simulation
 - On-chip Debugging
 - PowerPlay Power Analyzer
 - SSN Analyzer
 - Engineering Change Order (ECO)
 - Export Database
 - Archive Project

Message: 0 of 16 Location:

For Help, press F1

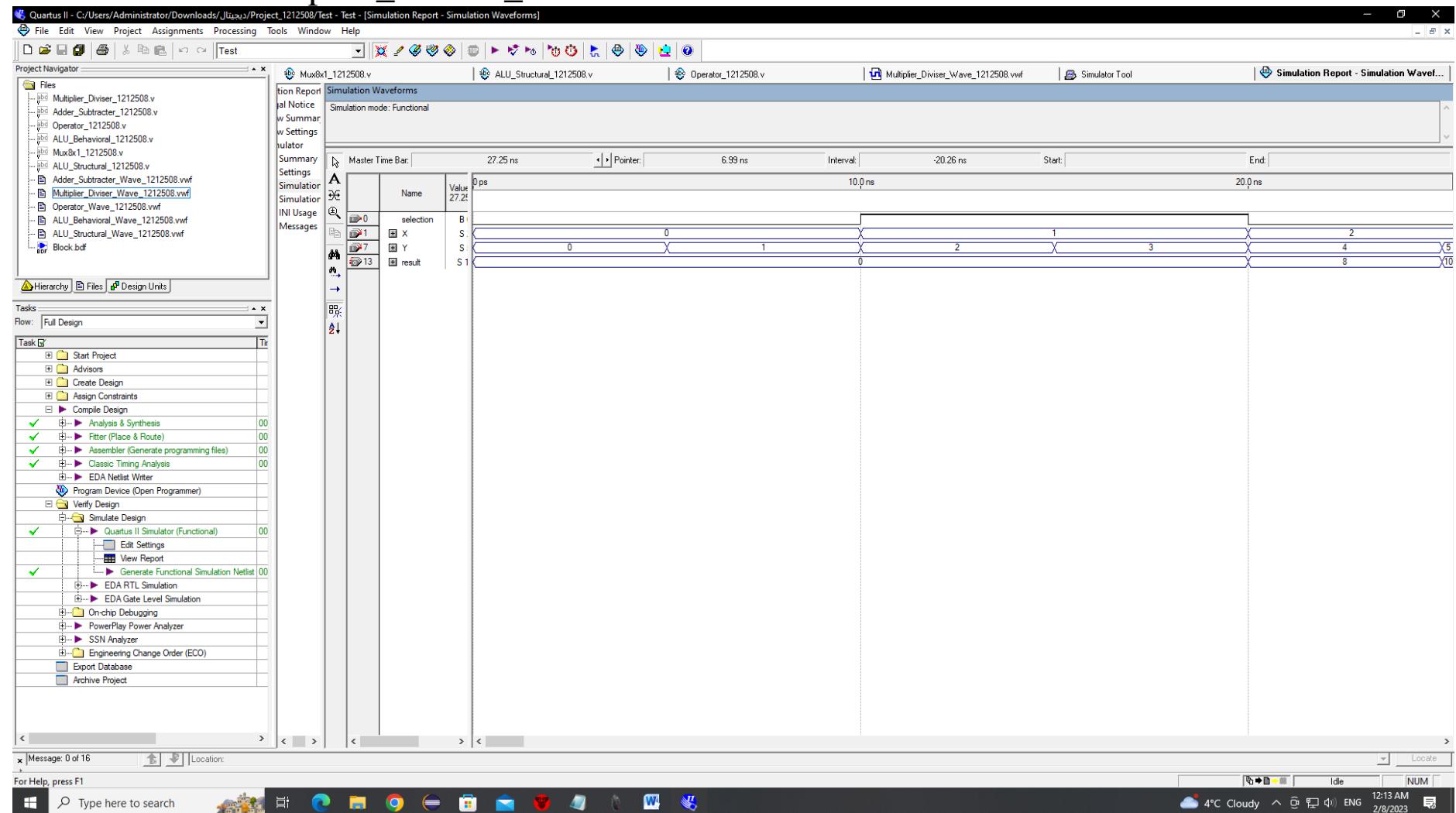
Type here to search

Ln 14, Col 1 Idle NUM

4°C Cloudy 11:59 PM 2/7/2023 ENG

```
1 module Multiplier_Diviser_1212508 #(parameter n = 4)(selection,X,Y,result);
2   input selection; //selection : determines process is multiply if it equals 0 or division if it equals 1.
3   input signed [n:0]X,Y; //X and Y are numbers in nBit.
4   output reg signed [n+2:0]result; //result of multiply or division.
5
6   always@(*)
7   begin
8     case(selection)
9       1'b0 : result = X * Y;
10      1'b1 : result = X / Y;
11    endcase
12  end
13 endmodule
14
```

Simulation of Multiplier_Diviser_1212508:



3. Operator_1212508

This module process four logic operations NAND, NOT, NOR and XOR between two inputs signed numbers X and Y those have n bit , and output the result of size n bit such never occur Overflow permanently.

To select the operation input C select NAND if it equal 00 in binary , select NOT if it equal 01 in binary, select NOR if it equal 10 in binary , also select XOR if it equal 11 in binary.

Behavioral module of Operator_1212508 in next page ..

Quartus II - C:/Users/Administrator/Downloads/JL4xx/Project_1212508/Test - Test - [Operator_1212508.v]

File Edit View Project Assignments Processing Tools Window Help

Test

Mux8x1_1212508.v ALU_Structural_1212508.v Operator_1212508.v

Files

- Multiplexer_Divider_1212508.v
- Adder_Subtractor_1212508.v
- Operator_1212508.v
- ALU_Behavioral_1212508.v
- Mux8x1_1212508.v
- ALU_Structural_1212508.v
- Adder_Subtractor_Wave_1212508.wvf
- Multiplexer_Divider_Wave_1212508.wvf
- Operator_Wave_1212508.wvf
- ALU_Behavioral_Wave_1212508.wvf
- ALU_Structural_Wave_1212508.wvf
- Block.bdf

Hierarchy Files Design Units

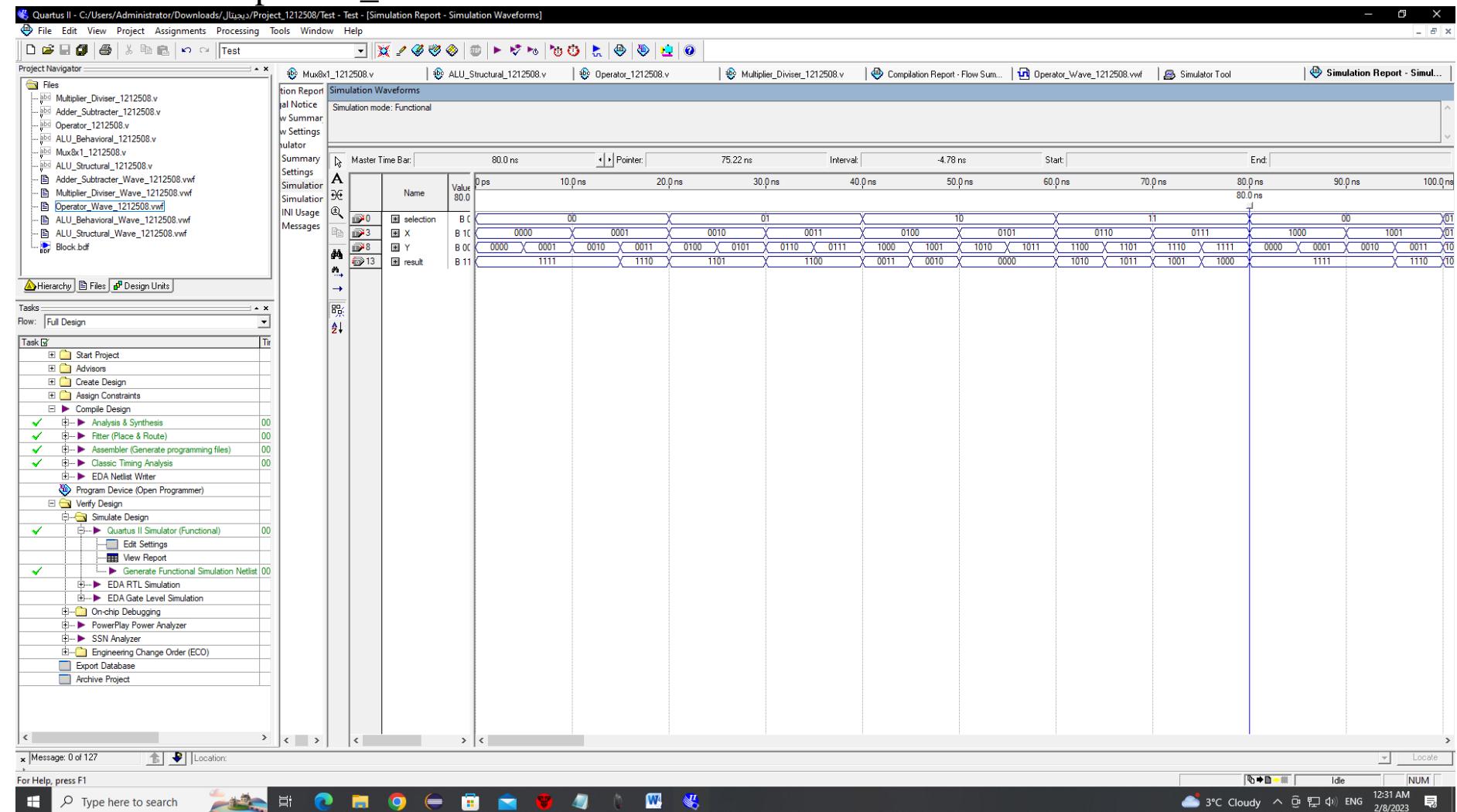
Tasks

Flow: Full Design

Task

- Start Project
- Advisors
- Create Design
- Assign Constraints
- Compile Design
 - Analysis & Synthesis
 - Filter (Place & Route)
 - Assembler Generate programming files
 - Classic Timing Analysis
 - EDA Netlist Writer
- Program Device (Open Programmer)
- Verify Design
 - Simulate Design
 - Quartus II Simulator (Functional)
 - Edit Settings
 - View Report
 - Generate Functional Simulation Netlist
 - EDA RTL Simulation
 - EDA Gate Level Simulation
 - On-chip Debugging
 - PowerPlay Power Analyzer
 - SSN Analyzer
 - Engineering Change Order (ECO)
 - Export Database
 - Archive Project

Simulation of Operator_1212508 :



4. Mux8x1_1212508

This module controls the ALU , that is processes all 8 operations arithmetic and logic, Between two inputs signed numbers X and Y each size is n bit, and output O in n+2 bit that Overflow can't happen.

To select any operation there is a selector in Multiplexer called C it is signed number values of it from 0 to 7 to select any operation from these table:

ALU Function Code (C)	ALU Output (O)
000	$(X+Y)/2$
001	$2*(X+Y)$
010	$(X/2)+Y$
011	$X-(Y/2)$
100	$X \text{ NAND } Y$
101	$\text{NOT}(X)$
110	$X \text{ NOR } Y$
111	$X \text{ XOR } Y$

Behavioral module of Mux8x1_1212508 :

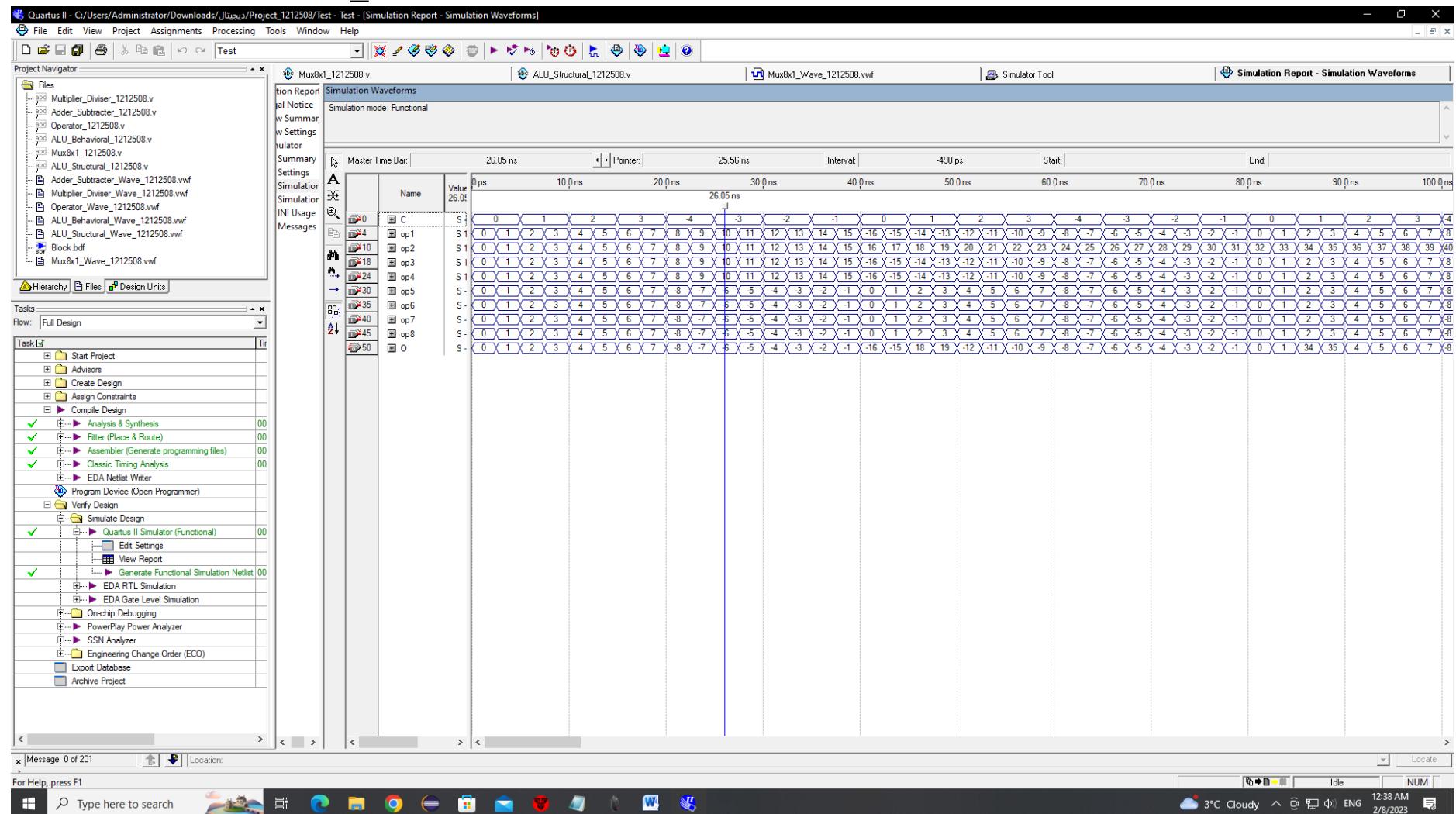
The screenshot shows the Quartus II software interface with the following details:

- Title Bar:** Quartus II - C:/Users/Administrator/Downloads/JL402/Project_1212508/Test - Test - [Mux8x1_1212508.v]
- Menu Bar:** File, Edit, View, Project, Assignments, Processing, Tools, Window, Help
- Toolbars:** Standard toolbar with icons for file operations, zoom, and search.
- Project Navigator:** Shows a tree view of project files including Multiplier_Diviser_1212508.v, Adder_Subracter_1212508.v, Operator_1212508.v, ALU_Behavioral_1212508.v, Mux8x1_1212508.v (selected), ALU_Structural_1212508.v, and various waveform files (Adder_Subracter_Wave_1212508.vwf, Multiplier_Diviser_Wave_1212508.vwf, Operator_Wave_1212508.vwf, ALU_Behavioral_Wave_1212508.vwf, ALU_Structural_Wave_1212508.vwf) and a Block.bdf file.
- Design Unit Editor:** Displays the behavioral module code for Mux8x1_1212508.v. The code defines a module with parameters n=4, inputs C, op1, op2, op3, op4, op5, op6, op7, op8, and output O. It uses an always block with a case statement to map binary values of C to output values op1 through op8.
- Tasks List:** Shows a list of tasks completed (Compile Design, Analysis & Synthesis, Fitter, Assemble, Classic Timing Analysis, EDA Netlist Writer) and pending (Start Project, Create Design, Assign Constraints, Verify Design, Simulate Design, On-chip Debugging, PowerPlay Power Analyzer, SSN Analyzer, ECO, Export Database, Archive Project).
- Status Bar:** Message: 0 of 177, Location: , For Help, press F1, and various system status indicators like NUM, 12:33 AM, 3°C Cloudy, ENG, 2/8/2023.

```
module Mux8x1_1212508 #(parameter n = 4) (C,op1,op2,op3,op4,op5,op6,op7,op8,O);
    input [2:0]C;
    input signed [n:0]op1,op3,op4; //op : operation
    input signed [n+2:0]op2;
    input signed [n-1:0]op5,op6,op7,op8;
    output reg signed [n+2:0]O; //O : output

    always @(*)
    begin
        case(C) //sets the operation.
            3'b000 : O = op1;
            3'b001 : O = op2;
            3'b010 : O = op3;
            3'b011 : O = op4;
            3'b100 : O = op5;
            3'b101 : O = op6;
            3'b110 : O = op7;
            3'b111 : O = op8;
        endcase
    end
endmodule
```

Simulation of Mux8x1_1212508 :



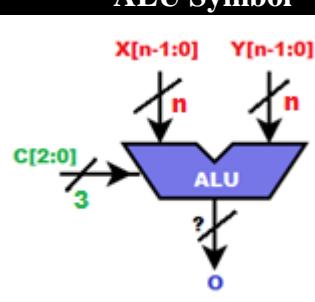
Part D. Structural Verilog model for ALU

ALU_Structural_1212508

ALU (Arithmetic and Logic Unit) is unit processes all operations in any computer , so we build it in this case as a structural model in module called ALU_Structural_1212508 that built from blocks of components that mentioned above and they are Adder_Subtracter_1212508 , Multiplier_Diviser_1212508, Operator_1212508 and Mux8x1_1212508.

ALU take inputs signed numbers X, Y of size n and selector C of size 3 bit the output is signed number O of size n+2.

This is the table of ALU :

ALU Function Code (C)	ALU Output (O)	ALU Symbol
000	$(X+Y)/2$	
001	$2*(X+Y)$	
010	$(X/2)+Y$	
011	$X-(Y/2)$	
100	$X \text{ NAND } Y$	
101	$\text{NOT}(X)$	
110	$X \text{ NOR } Y$	
111	$X \text{ XOR } Y$	

Structural module of ALU_Structural_1212508:

The screenshot shows the Quartus II interface with the project "ALU_Structural_1212508" open. The main window displays the Verilog code for the ALU module. The code defines a module ALU_Structural_1212508 with parameter n = 4, inputs C, X, Y, and output O. It uses various internal wires and components like Adder_Subtractor, Multiplier_Diviser, and Operator to implement different arithmetic and logical operations. The Project Navigator on the left shows other files in the project, and the Tasks pane on the right lists completed design tasks.

```
1 module ALU_Structural_1212508 #(parameter n = 4) (C,X,Y,O);
2   input [2:0]C;
3   input signed [n-1:0]X,Y;
4   output signed [n+2:0]O;
5   wire [n:0]w,w1;
6   wire [n-1:0]w2,w3;
7   wire [n:0]op1,op3,op4;
8   wire [n+2:0]op2;
9   wire [n-1:0]op5,op6,op7,op8;
10
11 //Operation 1 : (X+Y)/2.
12 Adder_Subtracter_1212508 #(4) addOp1(1'b0,X,Y,w);
13 Multiplier_Diviser_1212508 #(4) divOp1(1'b1,w,4'd2,op1);
14 //Operation 2 : 2*(X+Y).
15 Adder_Subtracter_1212508 #(4) addOp2(1'b0,X,Y,w1);
16 Multiplier_Diviser_1212508 #(4) mulOp(1'b0,w1,4'd2,op2);
17 //Operation 3 : (X/2)+Y.
18 Multiplier_Diviser_1212508 #(4) divOp2(1'b1,X,4'd2,w2);
19 Adder_Subtracter_1212508 #(4) addOp3(1'b0,w2,Y,op3);
20 //Operation 4 : X-(Y/2).
21 Multiplier_Diviser_1212508 #(4) divOp3(1'b1,Y,4'd2,w3);
22 Adder_Subtracter_1212508 #(4) subOp(1'b1,X,w3,op4);
23 //Operation 5 : X NAND Y.
24 Operator_1212508 #(4) NAND(C[1:0],X,Y,op5);
25 //Operation 6 : NOT(X).
26 Operator_1212508 #(4) NOT(C[1:0],X,Y,op6);
27 //Operation 7 : X NOR Y.
28 Operator_1212508 #(4) NOR(C[1:0],X,Y,op7);
29 //Operation 8 : X XOR Y.
30 Operator_1212508 #(4) XOR(C[1:0],X,Y,op8);
31 //Processing the selected operation.
32 Mux8x1_1212508 #(4) Operation(C[2:0],op1,op2,op3,op4,op5,op6,op7,op8,O[n+2:0]);
33
34 endmodule
```

Part E. Waveform of ALU Structural 1212508

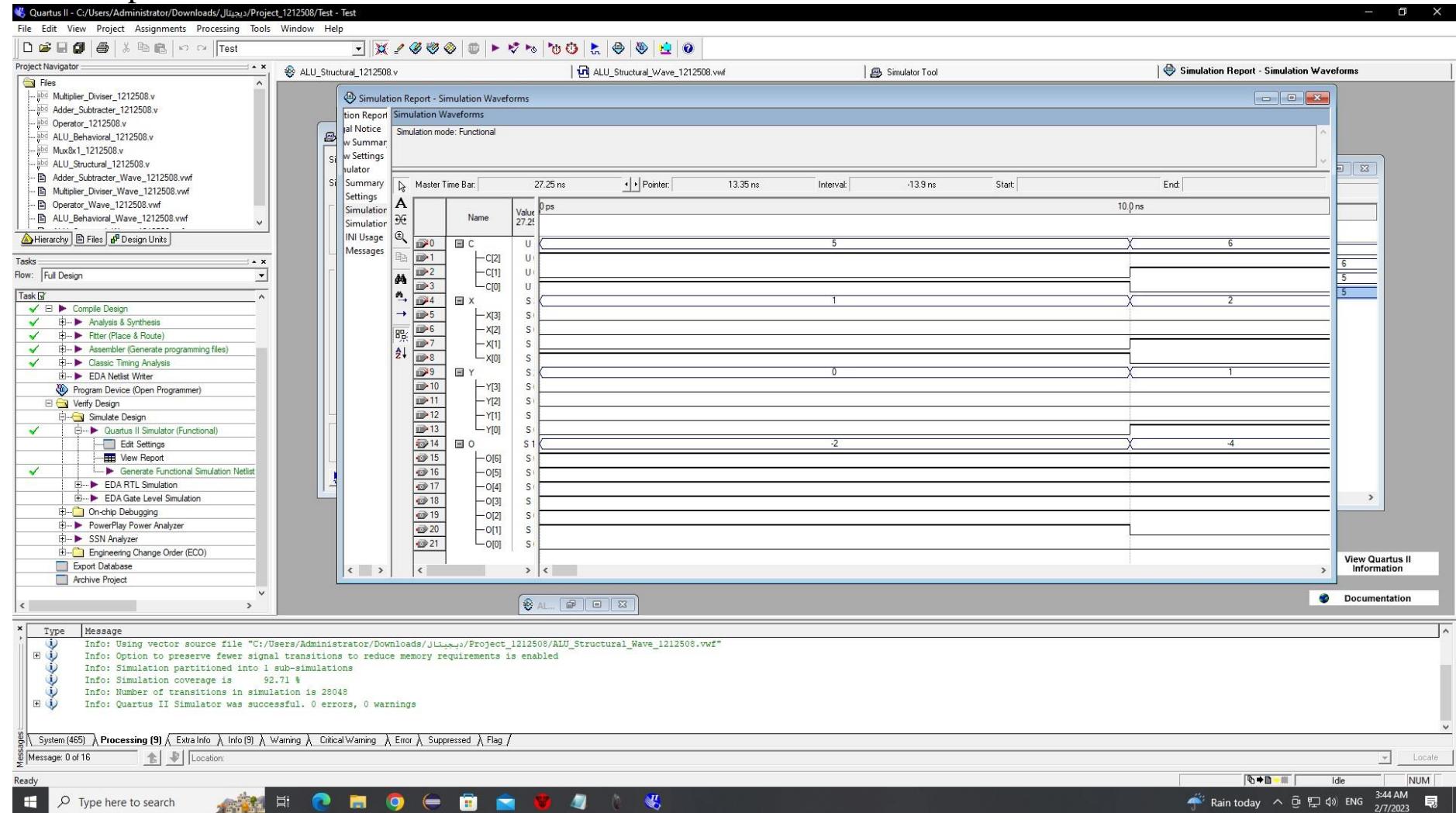
My ID Number is 1212508 that symbols in our ALU is 1C2Y2X2C1Y1X1, and we will replace number 8 by number 1..

This the table of Tests :

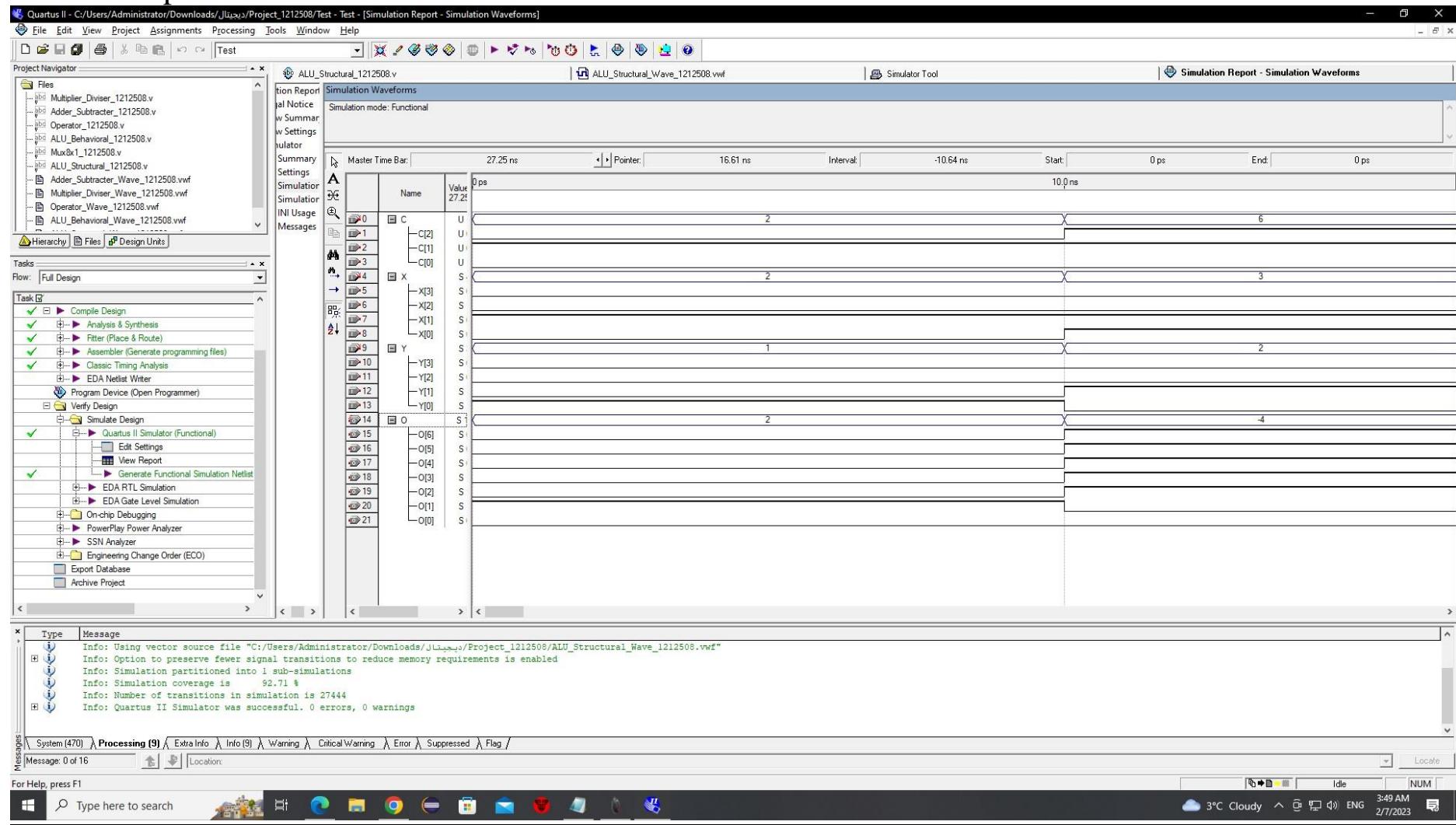
Test	X	Y	C	O
1	$X_1 = 1$	$Y_1 = 0$	$C_1 = 5$	$\text{NOT}(1) = -2$
2	$X_2 = 2$	$Y_2 = 1$	$C_2 = 2$	$((2)/2)+(1) = 2$
3	$X_3 = -X_1 = -1$	$Y_3 = -Y_1 = 0$	$C_3 = C_2 = 2$	$((-1)/2)+(0) = 0$

Now there is snapshots of these 3 tests on ALU_Structural_1212508...

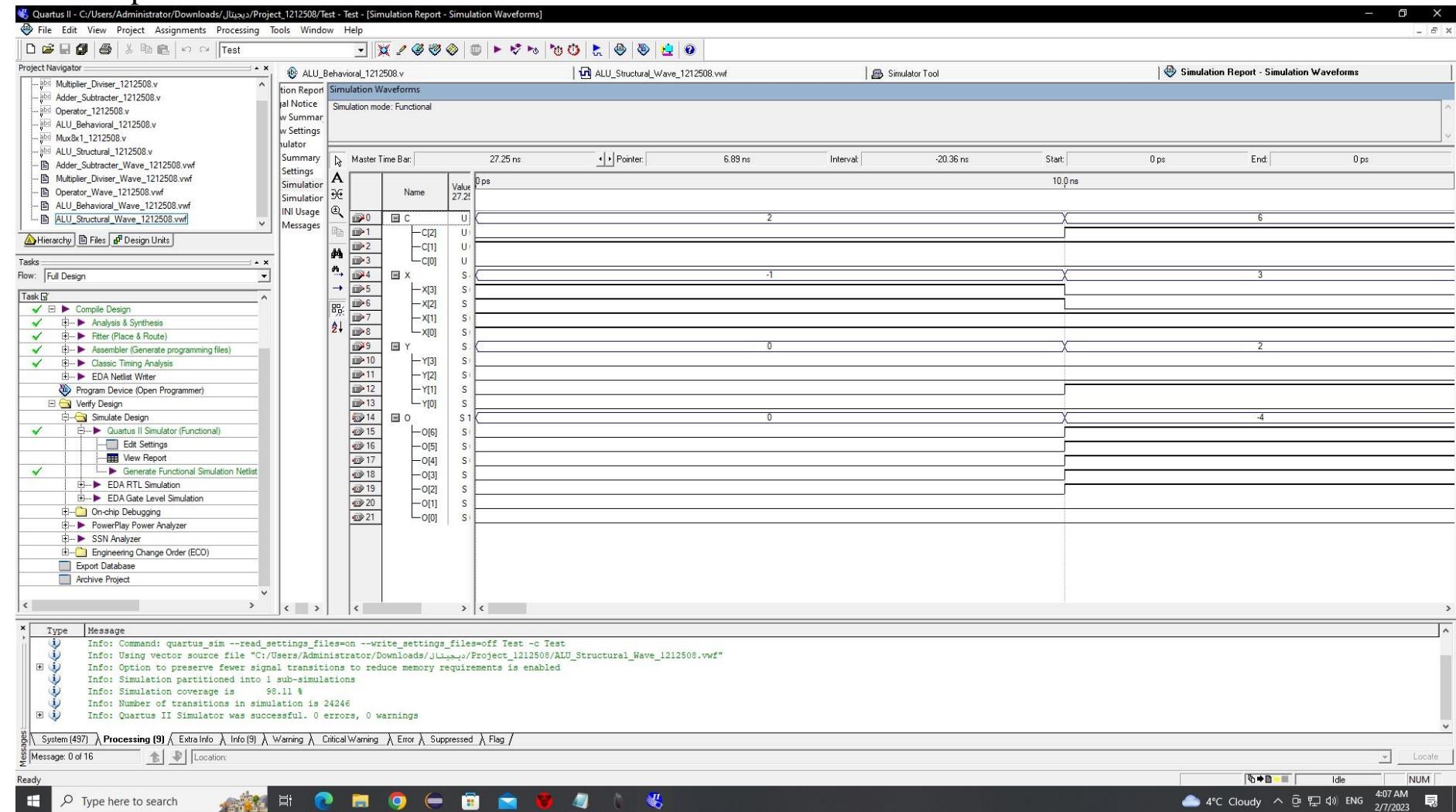
Test 1 report simulation :



Test 2 report simulation :



Test 3 report simulation :



Part F. Behavioral module of our ALU

In this part we build ALU without any component that we mentioned above , but of course this results same results in ALU in Structural module ..

This is the behavioral module code of ALU :

The screenshot shows the Quartus II software interface with the following details:

- Title Bar:** Quartus II - C:/Users/Administrator/Downloads/JL402/Project_1212508/Test - Test - [ALU_Behavioral_1212508.v]
- Toolbar:** File, Edit, View, Project, Assignments, Processing, Tools, Window, Help.
- Project Navigator:** Shows a tree view of files including Multiplier_Diviser_1212508.v, Adder_Subracter_1212508.v, Operator_1212508.v, ALU_Behavioral_1212508.v, Mux&1_1212508.v, ALU_Structural_1212508.v, Adder_Subracter_Wave_1212508.vwf, Multiplier_Diviser_Wave_1212508.vwf, Operator_Wave_1212508.vwf, ALU_Behavioral_Wave_1212508.vwf, ALU_Structural_Wave_1212508.vwf, Block.bdf, and Mux&1_Wave_1212508.vwf.
- Hierarchy Tab:** Selected tab in the Project Navigator.
- Design Navigator:** Shows tasks categorized by flow: Full Design, Top Level, Block Design, and Sub-Block Design.
- Task List:** A detailed list of tasks under "Task List" including Start Project, Advisors, Create Design, Assign Constraints, Compile Design, Analysis & Synthesis, Filter (Place & Route), Assemble (Generate programming files), Classic Timing Analysis, EDA Netlist Writer, Program Device (Open Programmer), Verify Design, Simulate Design (with Quartus II Simulator (Functional) selected), Generate Functional Simulation Netlist, EDA RTL Simulation, EDA Gate Level Simulation, On-chip Debugging, PowerPlay Power Analyzer, SSN Analyzer, Engineering Change Order (ECO), Export Database, and Archive Project.
- Code Editor:** The main window displays the behavioral module code for ALU_Behavioral_1212508.v.
- Code Content:**

```
1 module ALU_Behavioral_1212508 #(parameter n = 4) (C,X,Y,O);
2   input [2:0]C;
3   input signed [n-1:0]X,Y;
4   output reg signed [n+2:0]O;
5
6   always @(*)
7     begin
8       case(C)
9         3'b000 : O = (X + Y)/2;
10        3'b001 : O = 2 * (X+Y);
11        3'b010 : O = (X/2) + Y;
12        3'b011 : O = X - (Y/2);
13        3'b100 : O = ~ (X&Y);
14        3'b101 : O = ~X;
15        3'b110 : O = ~ (X|Y);
16        3'b111 : O = X^Y;
17     endcase
18   end
19 endmodule
```
- System Tray:** Shows the Windows taskbar with the Start button, search bar, and various pinned icons.
- Bottom Status Bar:** Shows the message "Message: 0 of 201", location "Locate", help "For Help, press F1", status "Ln 1, Col 1", idle "Idle", NUM, weather "4°C Cloudy", time "1:25 AM", date "2/8/2023", and language "ENG".

Part G. Waveform of ALU Behavioral 1212508

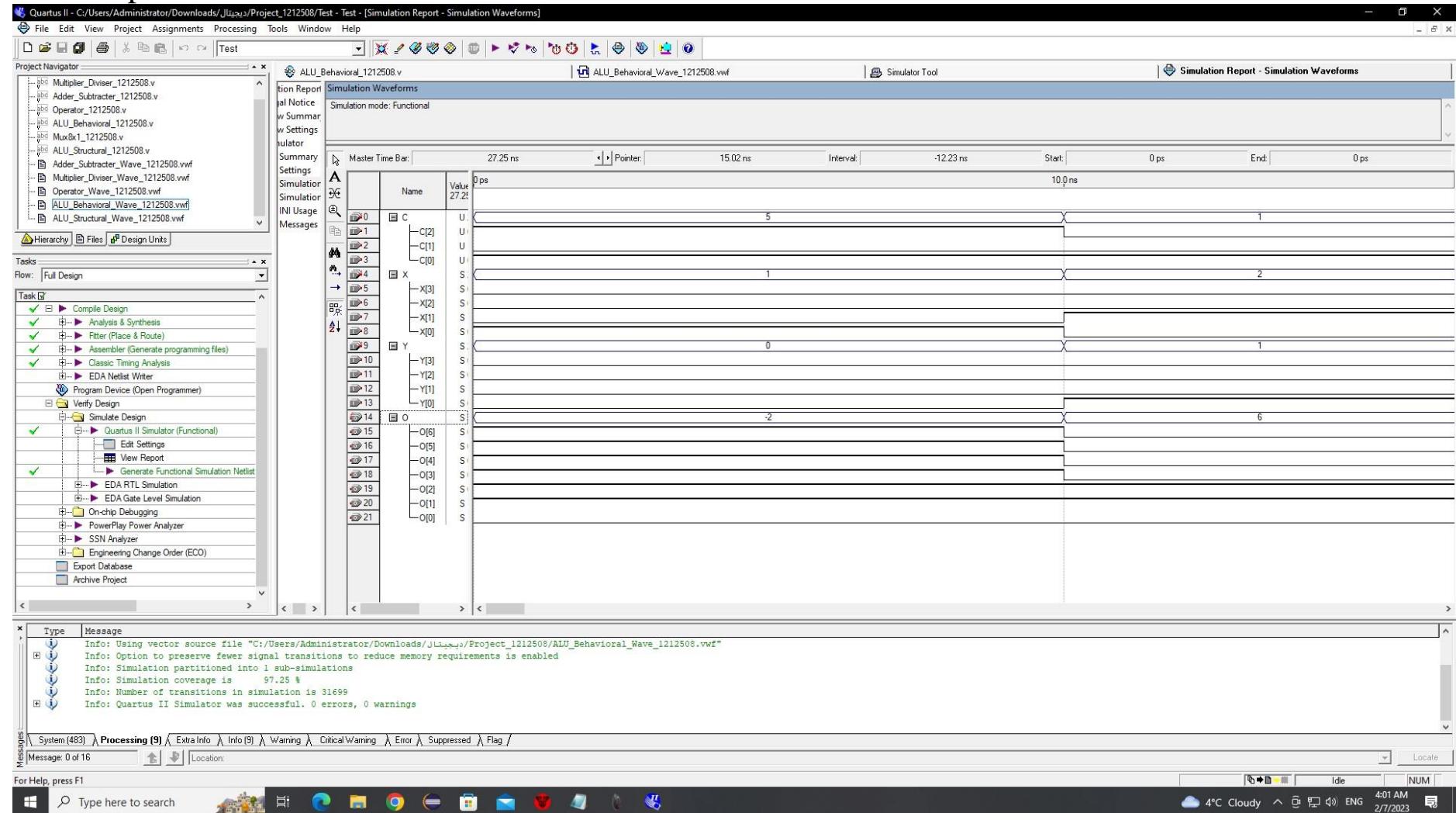
My ID Number is 1212508 that symbols in our ALU is 1C2Y2X2C1Y1X1, and we will replace number 8 by number 1..

This the table of Tests :

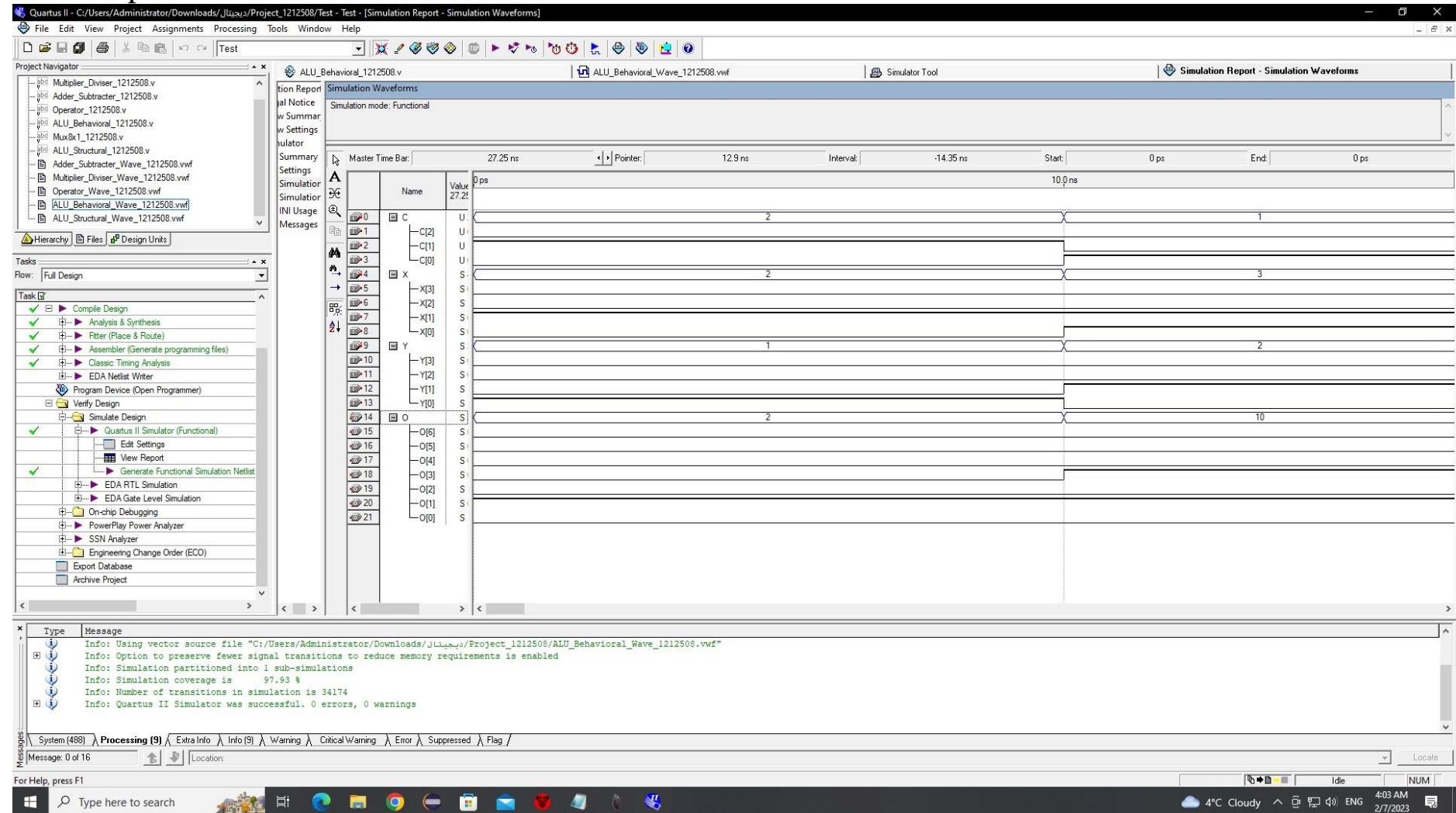
Test	X	Y	C	O
1	$X_1 = 1$	$Y_1 = 0$	$C_1 = 5$	$\text{NOT}(1) = -2$
2	$X_2 = 2$	$Y_2 = 1$	$C_2 = 2$	$((2)/2)+(1) = 2$
3	$X_3 = -X_1 = -1$	$Y_3 = -Y_1 = 0$	$C_3 = C_2 = 2$	$((-1)/2)+(0) = 0$

Now there is snapshots of these 3 tests on ALU_Behavioral_1212508...

Test 1 report simulation :



Test 2 report simulation :



Test 3 report simulation :

