Fall Semester 2021-2022

ECE5017 - Digital Design with FPGA ELA

M.Tech VLSI Design

School of Electronics Engineering

Vellore Institute of Technology

Name: Shreyas S Bagi

Register Number: 21MVD0086

Slot: L5+L6

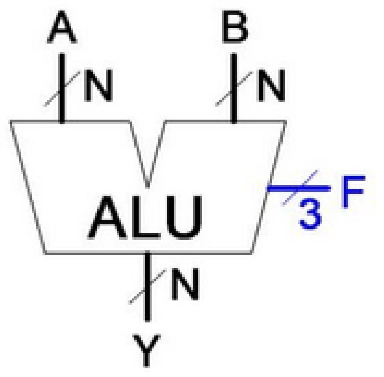
Date:13/10/2021

Lab Task 02

Structural Modelling

Aim: Design a 16-bit ALU in structural modelling using Verilog HDL, which performs the following functions as shown in table. To test the ALU, write a Verilog test-bench code.

F2	F1	F0	FUNCTION
0	0	0	A+B
0	0	1	A+1
0	1	0	A-B
0	1	1	A-1
1	0	X	A*B



Objective:

- To design the ALU circuit using Verilog HDL.
- To write the test bench for generating stimulus.
- To perform functional simulation.
- Understand instantiation of gates, gate symbols, and truth tables.
- Design of ALU circuit using Structural Modelling.
-

Software Details:

For design Functional Simulation: Modelsim-INTEL FPGA STARTER EDITION.

Date:13/10/2021

Verilog Code :

```
// The Keywords of Verilog are highlighted in Bold.
`timescale 10ns/1ns
module alu_F(A,B,F,Y); // Module declaration for ALU
  parameter N=16; // Input lines A and B are of 16bit
  parameter C=3; // 3 bit select line
  input [N-1:0] A,B; // Input lines for ALU
  input [C-1:0] F; // Control lines for ALU or Selection of particular function.
// This can used as Control path line.
  output [(2*N)-1:0] Y; // 32-bit output line because of Multiplication operation.
  wire [(2*N)-1:0] I0, I1,I1B, I2, I3, I4,I5,I6,I7;
  wire F1_bar, F2_bar, F0_bar;
 not G1 (F2_bar, F[2]);
 not G2 (F1_bar, F[1]);
 not G3 (F0_bar, F[0]);
 assign I1B=16'b0000_0000_0000_0001; // Logic 1
// Adder module Instantiation
 add_alu G4 (I0,A,B);        // 000 ----------------- A+B
// Adder module Instantiation
 add_alu G5 (I1,A,I1B);    // 001 ----------------- A+1
// Subtraction  module Instantiation
 sub_alu G6 (I2,A,B);       // 010 ----------------- A-B
// Subtraction  module Instantiation
 sub_alu G7 (I3,A,I1B);   // 011 -----------------  A-1
// Multiplication module Instantiation
 mul_alu G8 (I4,A,B);   // 100  ----------------- A*B
 mul_alu G9 (I5,A,B);   // 101  ----------------- A*B
 mux_8_to_1 G10 (Y,I0,I1,I2,I3,I4,I5,I6,I7,F); // 8 to 1 MUX
endmodule
```

Date:13/10/2021

```
// Adder Module Declaration
module add_alu (Y,A,B);
  parameter N=16;
  input [N-1:0] A,B;
  output [(2*N)-1:0] Y; // MSB bit is Carry and rest bits are Sum part
  assign Y=A+B; // Dataflow modelling of Addition of inputs.
endmodule

// Subtraction Module
module sub_alu (Y,A,B);
  parameter N=16;
  input [N-1:0] A,B;
  output [(2*N)-1:0] Y; // MSB bit is Borrow and rest are difference bits
  assign Y=A-B; // Dataflow modelling of Subtraction of inputs.
endmodule

// Multiplication Module
module mul_alu (Y,A,B);
  parameter N=16;
  input [N-1:0] A,B;
  output [(2*N)-1:0] Y;
  assign Y=A*B; // Dataflow modelling of Multiplication of inputs.
endmodule

// Mux 2 to 1
module mux_2_to_1 (Y,I0,I1,Sel);
  parameter N=32;
  input [N-1:0] I0,I1;
  input Sel;
  output [N-1:0] Y;
```

Date:13/10/2021

```
  wire [N-1:0] Y1,Y2;
  not  G4 (Sel_bar,Sel);
// The following of and gate primitive, or gate primitive are defined for 32bit operation.
  and G1[N-1:0] (Y1,I0,Sel_bar); // Input and output is of 32 bit.
  and G2[N-1:0] (Y2,I1,Sel); // Input and output is of 32 bit.
  or G3[N-1:0] (Y,Y1,Y2); // Input and output is of 32 bit.
endmodule


// Mux 4 to 1
module mux_4_to_1 (Y,I0,I1,I2,I3,Sel);
  parameter N=32;
  input [N-1:0] I0,I1,I2,I3;
  input [1:0] Sel;
  output tri [N-1:0] Y;
  wire [N-1:0] Iout1,Iout2;
  mux_2_to_1 G1 (Iout1,I0,I1,Sel[0]);
  mux_2_to_1 G2 (Iout2,I2,I3,Sel[0]);
  mux_2_to_1 G3 (Y,Iout1,Iout2,Sel[1]);
endmodule


// Mux 8 to 1
module mux_8_to_1 (Y,I0,I1,I2,I3,I4,I5,I6,I7,Sel);
  parameter N=32;
  input [N-1:0] I0,I1,I2,I3,I4,I5,I6,I7;
  input [2:0] Sel;
  output tri [N-1:0] Y;
  wire [N-1:0] Iout1,Iout2;
  mux_4_to_1 G1 (Iout1,I0,I1,I2,I3,Sel[1:0]);
  mux_4_to_1 G2 (Iout2,I4,I5,I6,I7,Sel[1:0]);
  mux_2_to_1 G3 (Y,Iout1,Iout2,Sel[2]);
```

```
endmodule
```

Testbench Code :

```
module test_alu();
 parameter N=16;
 reg [N-1:0] A,B;
 wire [N:0] Y;
 parameter C=3;
 reg [C-1:0] F;
 alu_F G11 (A,B,F,Y);
 initial
 begin
   $monitor($time,"A=%b,B=%b,F=%b,Y=%b",A,B,F,Y);
   #2 A=65000;B=500;F=0; // A+B
   #2 A=6999;B=30;F=1;    // A+1
   #2 A=50;B=40;F=2;       // A-B
   #2 A=10;B=80;F=3;      // A-1
   #2 A=80;B=200;F=4;   // A*B
   #2 A=100;B=90;F=5;    // A*B
   #2 A=100;B=450;F=6;
   #2 A=98;B=1000;F=7;
   #5 $stop;
 end
endmodule
```

Date:13/10/2021

Simulation Waveform :

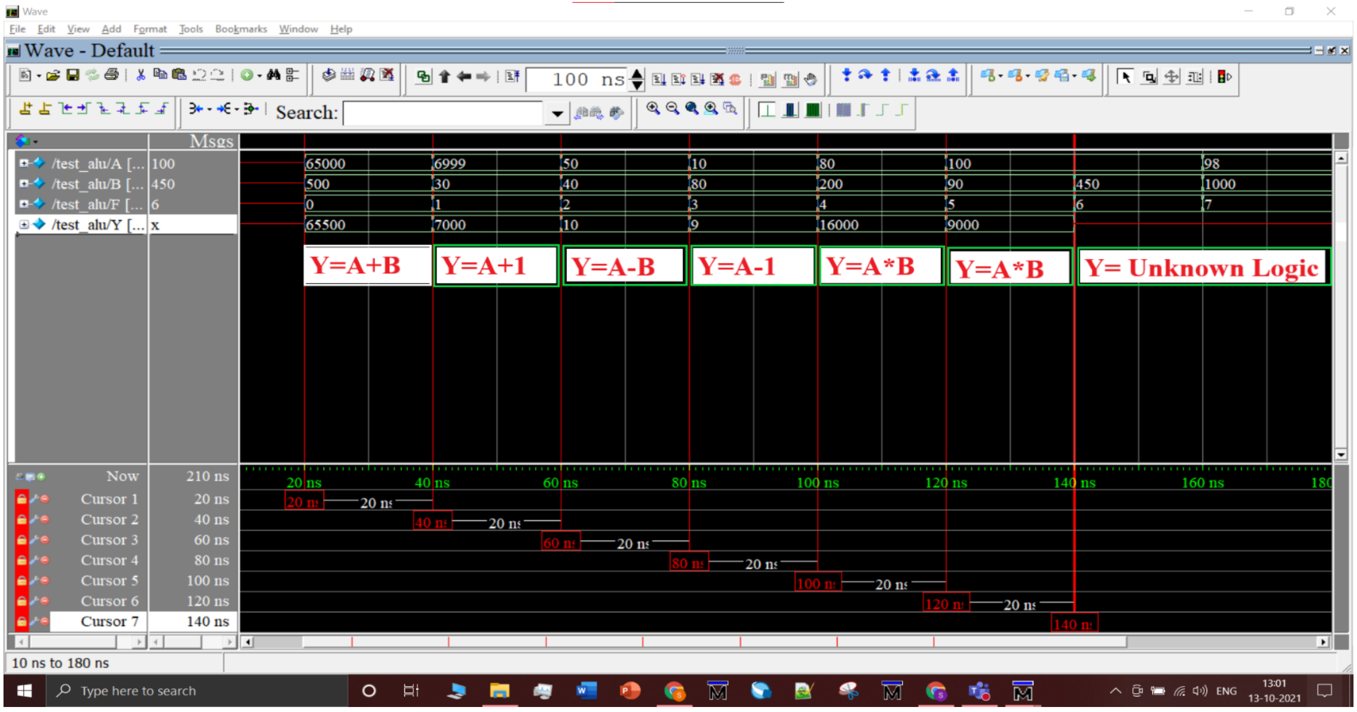
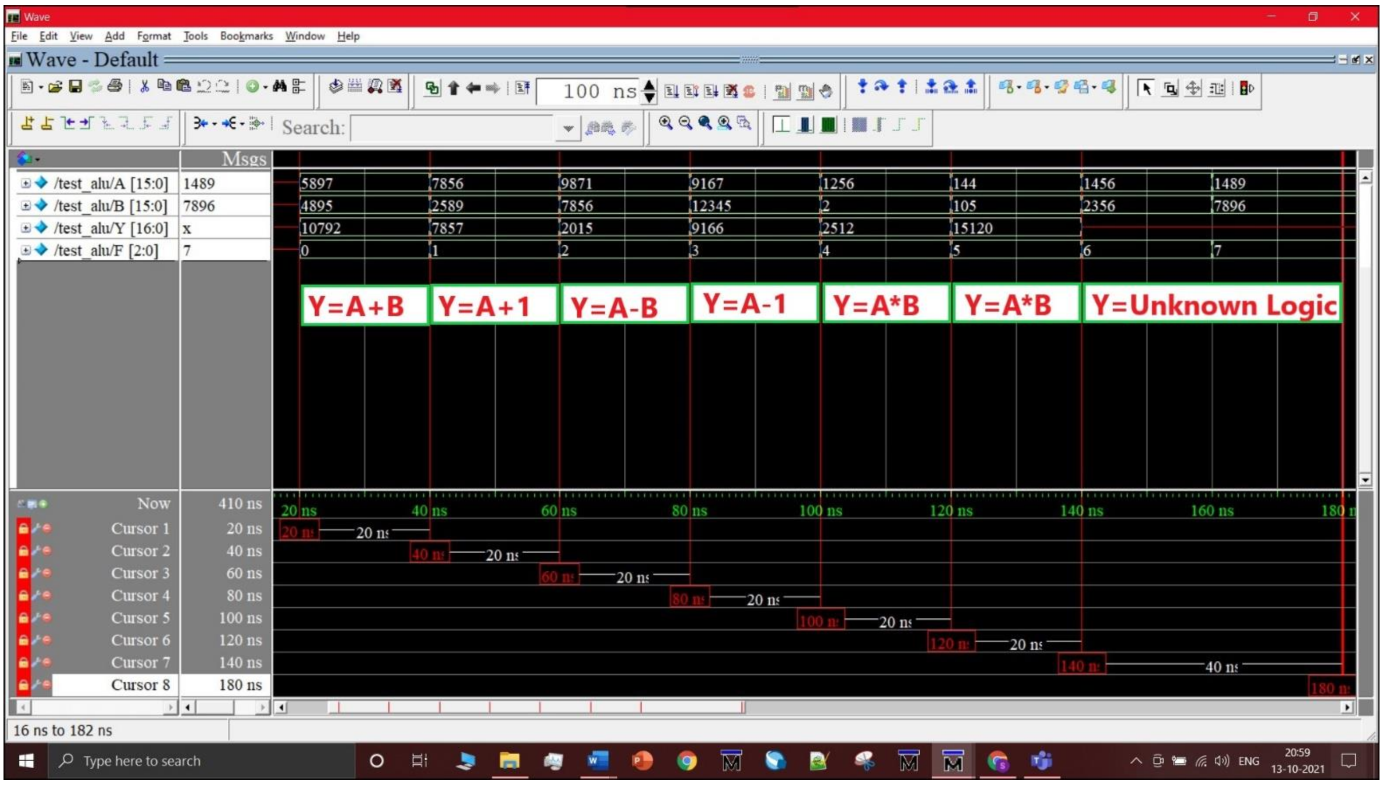


Figure 1.1 Simulation result for ALU Module for Different combination of A,B and Select lines.

In the Figure 1.1 we can observe when select line is 0 the operation performed is addition of A and B input. The addition output is observed from time 20ns to 40ns. Every time after 20ns the Inputs and Select line is getting updated. When Select line is 1 the operation performed is increment input A by one. The increment of Input A is observed from time 40ns to 60ns. When Select line is 2 the operation performed is subtraction of A and B input lines. The subtraction operation is observed from time 60ns to 80ns. When Select line is 3 the operation performed is decrement A by one. The decrement operation is observed from time 80ns to 100ns. When Select line is 4 and 5 the operation performed is multiplication of inputs. The Multiplication operation output can be observed from 100ns to 140ns. When Select line is 6 and 7 the output is at Unknown Logic because in the truth table there is function mentioned for these select lines.

Date:13/10/2021

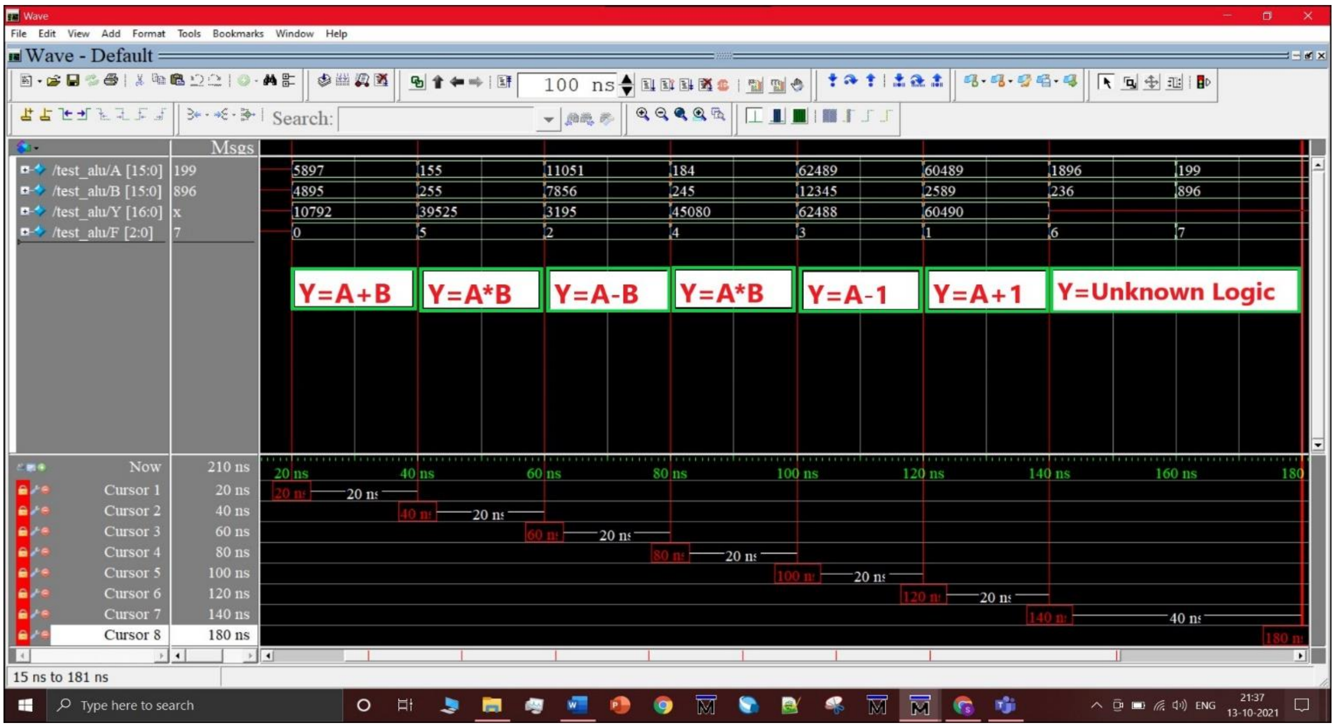


```
/test_alu/A [15:0]  1489  5897   7856  9871  9167   1256  144    1456  1489
/test_alu/B [15:0]  7896  4895   2589  7856  12345  2     105    2356  7896
/test_alu/Y [16:0]  x     10792  7857  2015  9166   2512  15120
/test_alu/F [2:0]   7     0      1     2     3      4     5      6     7

Now       410 ns
Cursor 1   20 ns
Cursor 2   40 ns
Cursor 3   60 ns
Cursor 4   80 ns
Cursor 5  100 ns
Cursor 6  120 ns
Cursor 7  140 ns
Cursor 8  180 ns

16 ns to 182 ns
```

Figure 1.2 Simulation result for ALU Module for Different combination of A,B and Select lines.



```
/test_alu/A [15:0]  199  5897   155    11051  184    62489  60489  1896  199
/test_alu/B [15:0]  896  4895   255    7856   245    12345  2589   236   896
/test_alu/Y [16:0]  x    10792  39525  3195   45080  62488  60490
/test_alu/F [2:0]   7    0      5      2      4      3      1      6     7

Now       210 ns
Cursor 1   20 ns
Cursor 2   40 ns
Cursor 3   60 ns
Cursor 4   80 ns
Cursor 5  100 ns
Cursor 6  120 ns
Cursor 7  140 ns
Cursor 8  180 ns

15 ns to 181 ns
```

Figure 1.3 Simulation result for ALU Module for Different combination of A,B and Select lines.

Date:13/10/2021

Conclusion

Gate level modelling is a low level of abstraction. Design at this level is like describing a design in terms of a gate-level logic diagram.

Dataflow modelling is second highest level of abstraction. The module is implementing the function at a level of abstraction higher than gate level. It provides a powerful way to implement a design. Verilog allows a circuit to be designed in terms of data flow between registers and how a design processes data rather than instantiation of individual gates.

The Structural modelling is a combination of Gate level modelling and Dataflow modelling. In Structural style, a module is described as a set of interconnected components. The components can be modules, gate primitives and switch primitives.

Date:13/10/2021