Project2

Verilog
Arithmetic Logic Unit Synopsys

Lin, Shih-Hao

UTD-ID :2021497449

Yang, Wen-Fu

UTD-ID:2021547369

1.Waveform Compare

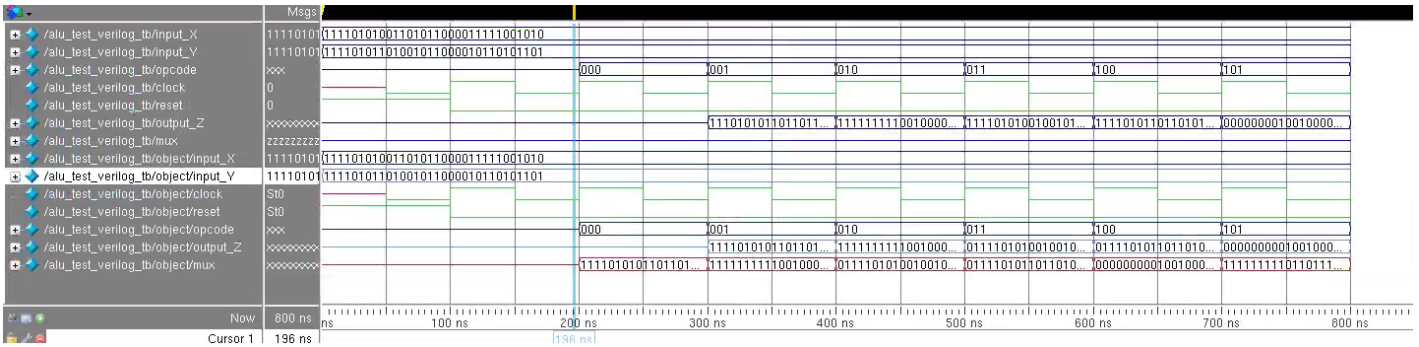


Figure 1 behavioral code waveform

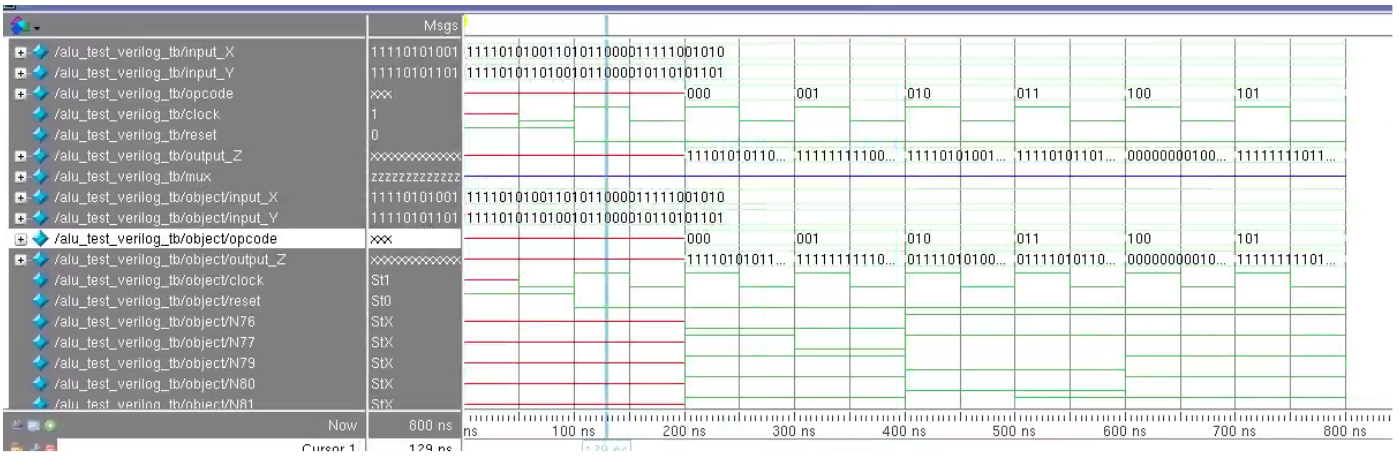


Figure 2 mapped code waveform

By firgure1. When opcode equal to 000, input_X + input_Y= output_Z , delay 100ns.

001, input_X - input_Y= output_Z , delay 100ns.

010, input_X & input_Y= output_Z , delay 100ns.

011, input_X | input_Y= output_Z , delay 100ns.

100, input_X ^ input_Y= output_Z , delay 100ns.

101, input_X ^~ input_Y= output_Z , delay 100ns.

110, input_X % input_Y= output_Z , delay 100ns.

111, input_X / input_Y= output_Z , delay 100ns.

By figure2. When opcode equal to 000, input_X + input_Y= output_Z.

001, input_X - input_Y= output_Z.

010, input_X & input_Y= output_Z.

011, input_X | input_Y= output_Z.

100, input_X ^ input_Y= output_Z.

101, input_X ^~ input_Y= output_Z.

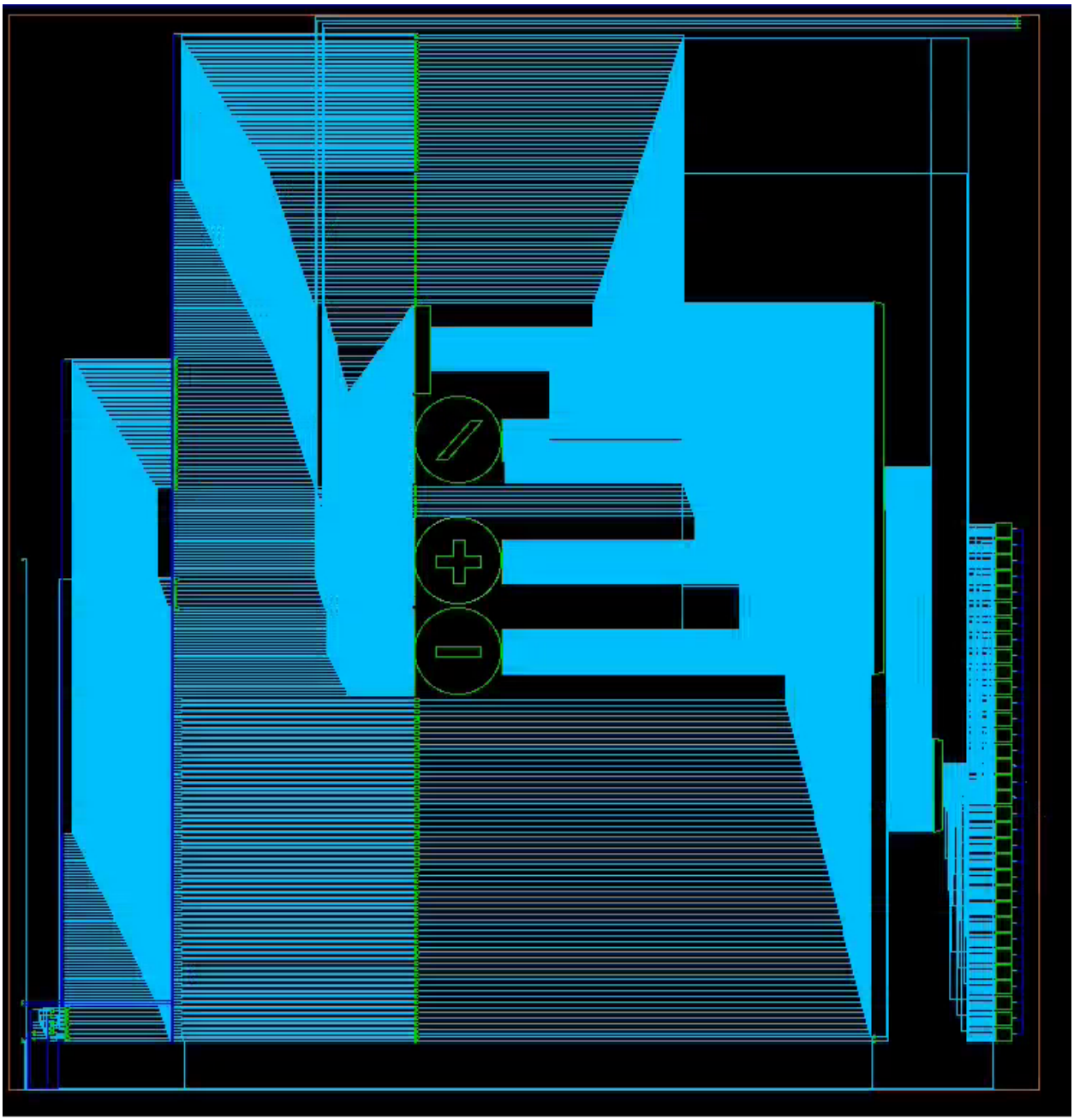
110, input_X % input_Y= output_Z.

111, input_X / input_Y= output_Z.

In Mapped code, the inputs didn't pass through the mux (multiplexer), hence the delay did not occur. Beside delay, two designs would be the same function in generally.

2. Design Vision

Schematic View



Total Amount of Cells

Report : cell

Design : alu_test_verilog

Version: L-2016.03-SP3

Date : Tue Sep 8 20:46:05 2020

****

Attributes:

b - black box (unknown)

h - hierarchical

n - noncombinational

r - removable

u - contains unmapped logic

Cell	Reference	Library	Area	Attributes
C7	nand2	library	1.000000	
C10	nand2	library	1.000000	
C14	nand2	library	1.000000	
C23	nand2	library	1.000000	
C33	nand2	library	1.000000	
C36	nand2	library	1.000000	
C277	nand2	library	1.000000	
C278	nand2	library	1.000000	
C279	nand2	library	1.000000	
C280	nand2	library	1.000000	
C281	nand2	library	1.000000	
.				
.				
.				
.				
.				
.				
.				
output_Z_reg[0]	dff	library	7.000000	n
output_Z_reg[1]	dff	library	7.000000	n
output_Z_reg[2]	dff	library	7.000000	n
output_Z_reg[3]	dff	library	7.000000	n
output_Z_reg[4]	dff	library	7.000000	n
output_Z_reg[5]	dff	library	7.000000	n

output_Z_reg[6]	dff	library	7.000000	n
output_Z_reg[7]	dff	library	7.000000	n
output_Z_reg[8]	dff	library	7.000000	n
output_Z_reg[9]	dff	library	7.000000	n
output_Z_reg[10]	dff	library	7.000000	n
output_Z_reg[11]	dff	library	7.000000	n
output_Z_reg[12]	dff	library	7.000000	n
output_Z_reg[13]	dff	library	7.000000	n
output_Z_reg[14]	dff	library	7.000000	n
output_Z_reg[15]	dff	library	7.000000	n
output_Z_reg[16]	dff	library	7.000000	n
output_Z_reg[17]	dff	library	7.000000	n
output_Z_reg[18]	dff	library	7.000000	n
output_Z_reg[19]	dff	library	7.000000	n
output_Z_reg[20]	dff	library	7.000000	n
output_Z_reg[21]	dff	library	7.000000	n
output_Z_reg[22]	dff	library	7.000000	n
output_Z_reg[23]	dff	library	7.000000	n
output_Z_reg[24]	dff	library	7.000000	n
output_Z_reg[25]	dff	library	7.000000	n
output_Z_reg[26]	dff	library	7.000000	n
output_Z_reg[27]	dff	library	7.000000	n
output_Z_reg[28]	dff	library	7.000000	n
output_Z_reg[29]	dff	library	7.000000	n
output_Z_reg[30]	dff	library	7.000000	n
output_Z_reg[31]	dff	library	7.000000	n
output_Z_reg[32]	dff	library	7.000000	n
.				
.				
.				
.				
r76/u_div/I_12	inv	library	1.000000	
r76/u_div/I_13	inv	library	1.000000	
r76/u_div/I_14	inv	library	1.000000	
r76/u_div/I_15	inv	library	1.000000	
r76/u_div/I_16	inv	library	1.000000	
r76/u_div/I_17	inv	library	1.000000	
r76/u_div/I_18	inv	library	1.000000	
r76/u_div/I_19	inv	library	1.000000	
r76/u_div/I_20	inv	library	1.000000	
r76/u_div/I_21	inv	library	1.000000	

r76/u_div/l_22 inv library 1.000000

r76/u_div/l_23 inv library 1.000000

r76/u_div/l_24 inv library 1.000000

r76/u_div/l_25 inv library 1.000000

r76/u_div/l_26 inv library 1.000000

r76/u_div/l_27 inv library 1.000000

r76/u_div/l_28 inv library 1.000000

r76/u_div/l_29 inv library 1.000000

r76/u_div/l_30 inv library 1.000000

Total ***4048*** cells 7146.000000

3.Design Code

1)ALU

```
module alu_test_verilog( input_X,  input_Y,  opcode,  clock, reset, output_Z);

        parameter Data_width=32;//building system as 32bits;

        input [Data_width-1:0]  input_X,input_Y;      // define two input;
        input       clock;   //define clock;
        input       reset;                          //define reset;
        input [2:0]   opcode;                       //define operation code;

        output reg [Data_width:0] output_Z;
```

```
reg [Data_width:0] mux ;   //multiplexer;


//designing ALU


        always@( input_X,  input_Y,  opcode,  clock, reset)
        begin
        case (opcode)


                3'b000 : mux <= input_X +  input_Y; //plus;
                3'b001 : mux <= input_X -  input_Y; //minus;
                3'b010 : mux <= input_X &  input_Y; //and;
                3'b011 : mux <= input_X |  input_Y; //or;
                3'b100 : mux <= input_X ^  input_Y; //XOR;
                3'b101 : mux <= input_X ^~ input_Y; //XNOR;
                3'b110 : mux <= input_X %  input_Y; //modules;
                3'b111 : mux <= input_X /  input_Y; //divide;



                endcase
        end

        always@(posedge clock)

                if (reset) begin
```

```
                output_Z=0;
                end


            else begin
             output_Z =mux;

                end

endmodule
```

2) testbench

```
module alu_test_verilog_tb;



    reg [31:0] input_X;//desiding inputs as 32 bits;
    reg [31:0] input_Y;
    reg [2:0] opcode;
    reg clock;
    reg reset;


    wire [31:0] output_Z;
    wire [31:0] mux;
```

```
alu_test_verilog object(
        .input_X (input_X),
        .input_Y (input_Y),
        .opcode  (opcode),
        .clock   (clock),
        .reset   (reset),
        .output_Z (output_Z)
        );

        initial begin

 input_X = 32'b11110101001101011000011111001010;//Input data;

        input_Y = 32'b11110101101001011000010110101101;//Input data;

            reset = 1;
        #100 reset = 0;



        #100 opcode = 3'b000; //plus;

        #100 opcode = 3'b001; //minus;

        #100 opcode = 3'b010; //bit-wise AND

        #100 opcode = 3'b011; //bit-wise OR
```

```
        #100 opcode = 3'b100; //bit-wise XOR;

        #100 opcode = 3'b101; //bit-wise XNOR;

        #100 opcode = 3'b110; //remainder

        #100 opcode = 3'b111; //divide;



        end

//set the cclock

initial begin
  #50 clock = 0;
end

always begin
  #50 clock = 0;
  #50 clock = 1;
end

endmodule
```

3) Synopsys code

```
////////////////////////////////////////////////////////////
// Created by: Synopsys DC Expert(TM) in wire load mode
// Version   : L-2016.03-SP3
// Date      : Tue Sep  8 21:46:26 2020
////////////////////////////////////////////////////////////


module alu_test_verilog ( input_X, input_Y, opcode, clock, reset, output_Z );
  input [31:0] input_X;
  input [31:0] input_Y;
  input [2:0] opcode;
  output [32:0] output_Z;
  input clock, reset;
  wire   N76, N77, N79, N80, N81, N82, N84, N85, N86, N87, N88, N89, N90, N91,
        N92, N93, N94, N95, N130, N131, N132, N133, N134, N135, N136, N137,
        N138, N139, N140, N141, N142, N143, N144, N145, N146, N147, N148,
        N149, N150, N151, N152, N153, N154, N155, N156, N157, N158, N159,
        N160, N161, N162, N163, N164, N165, N166, N167, N168, N169, N170,
        N171, N172, N173, N174, N175, N176, N177, N178, N179, N180, N181,
.
.
.
.
.
inv U3846 ( .in(n3711), .out(N294) );
  inv U3847 ( .in(n3712), .out(N293) );
```

```
  inv U3848 ( .in(n3713), .out(N292) );
  inv U3849 ( .in(n3714), .out(N291) );
endmodule
```