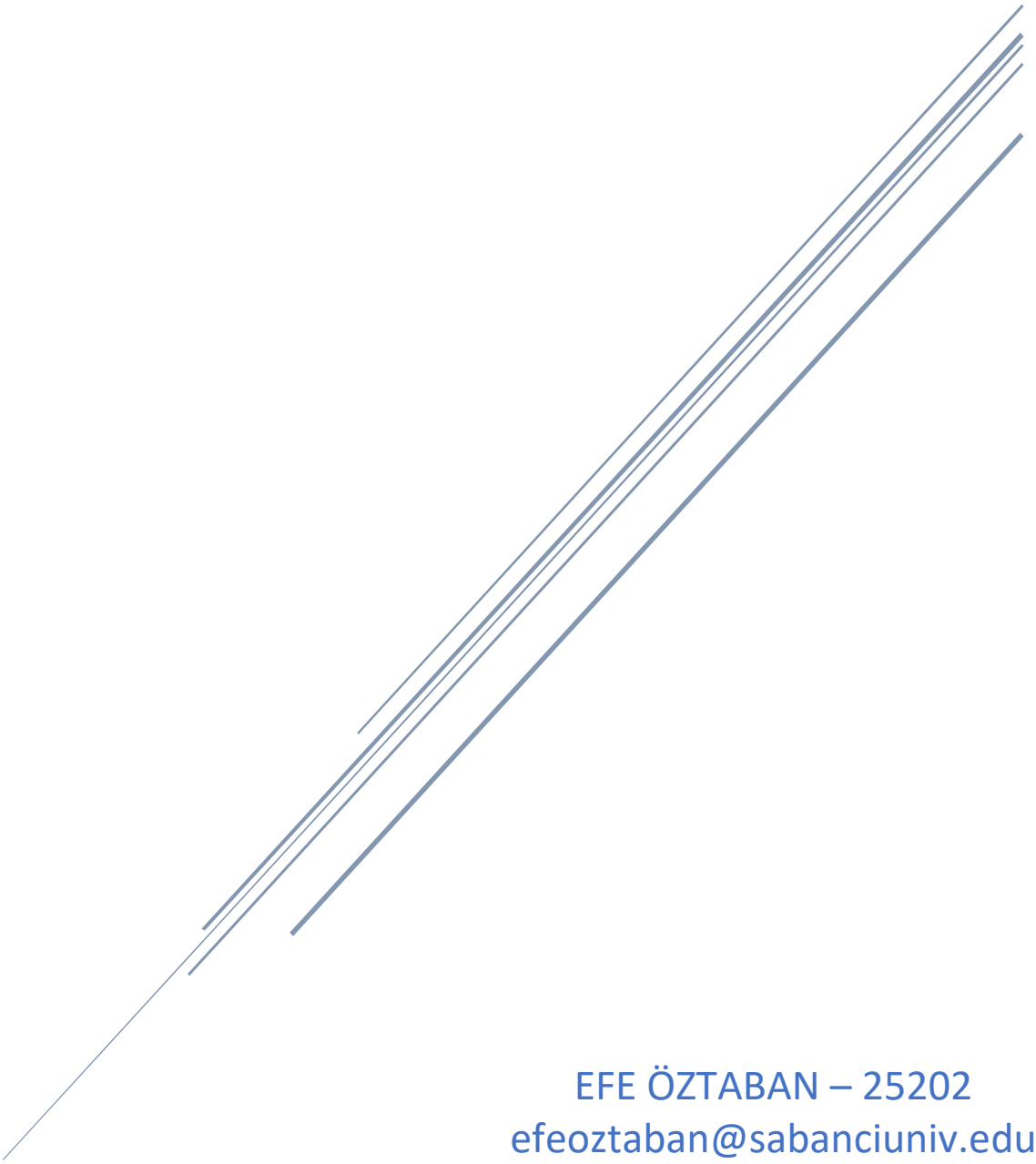
EE 401 VLSI SYSTEM DESIGN I

Lab Report for Lab #2



EFE ÖZTABAN – 25202
efeoztaban@sabanciuniv.edu

KAYRA BİLGİN – 25117
kayrabilgin@sabanciuniv.edu

1. Almost Correct Adder

a. Implemented Code

This is the Verilog implementation of the Almost Correct Adder. Full implementation of the 16-bit Almost Correct Adder is done as shown below:

```
module almost_correct_adder_16bit(clk,a,b,Cout,Sum);

   input [15:0] a,b;
   input clk;

   output reg [15:0] Sum;
   output reg Cout;

   wire [8:0] temp1, temp2, temp3, temp4, temp5, temp6, temp7, temp8, temp9;
   reg [15:0] a_reg, b_reg;

   assign temp1 = a_reg[7:0] + b_reg[7:0];
   assign temp2 = a_reg[8:1] + b_reg[8:1];
   assign temp3 = a_reg[9:2] + b_reg[9:2];
   assign temp4 = a_reg[10:3] + b_reg[10:3];
   assign temp5 = a_reg[11:4] + b_reg[11:4];
   assign temp6 = a_reg[12:5] + b_reg[12:5];
   assign temp7 = a_reg[13:6] + b_reg[13:6];
   assign temp8 = a_reg[14:7] + b_reg[14:7];
   assign temp9 = a_reg[15:8] + b_reg[15:8];



   always @ (posedge clk)
   begin

      a_reg <= a;
      b_reg <= b;

      Sum <= {temp9[7], temp8[7], temp7[7], temp6[7], temp5[7], temp4[7], temp3[7], temp2[7], temp1[7:0]};
      Cout <= temp9[8];

   end

endmodule
```

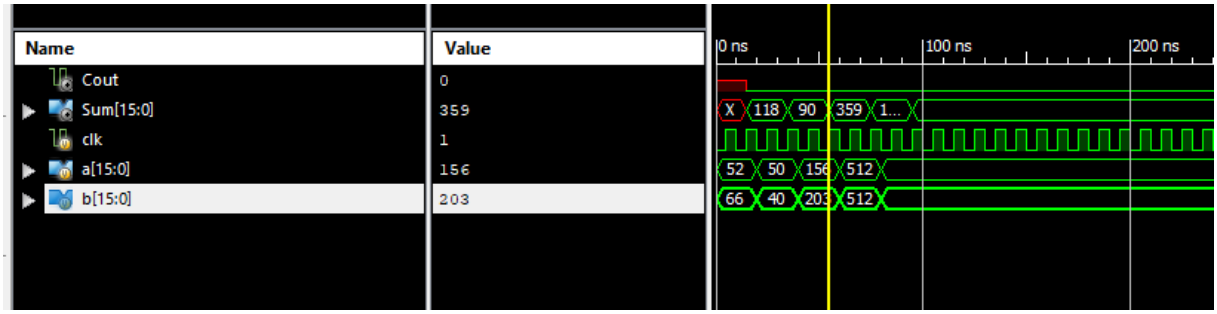
Also, the full implementation of the 32-bit Almost Correct Adder is done as shown below:

```
module almost_correct_adder_32bit(clk,A,B,Cout,S);



   input [31:0] A,B;
   input clk;

   output reg[31:0] S;
   output reg Cout;

   reg [8:0] temp1, temp2, temp3, temp4, temp5, temp6, temp7, temp8, temp9, temp10, temp11, temp12, temp13, temp14,
   reg [8:0] temp15, temp16, temp17, temp18, temp19, temp20, temp21, temp22, temp23, temp24, temp25;

   reg[31:0] a_reg, b_reg;
```

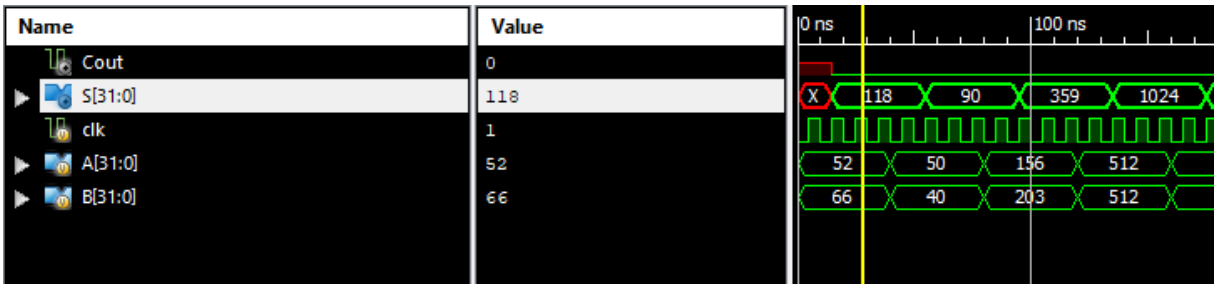
```
    always @(posedge clk) begin
       temp1 <= A[7:0] + B[7:0];
       temp2 <= A[8:1] + B[8:1];
       temp3 <= A[9:2] + B[9:2];
       temp4 <= A[10:3] + B[10:3];
             temp5 <= A[11:4] + B[11:4];
       temp6 <= A[12:5] + B[12:5];
       temp7 <= A[13:6] + B[13:6];
       temp8 <= A[14:7] + B[14:7];
       temp9 <= A[15:8] + B[15:8];
       temp10 <= A[16:9] + B[16:9];
       temp11 <= A[17:10] + B[17:10];
       temp12 <= A[18:11] + B[18:11];
       temp13 <= A[19:12] + B[19:12];
       temp14 <= A[20:13] + B[20:13];
       temp15 <= A[21:14] + B[21:14];
       temp16 <= A[22:15] + B[22:15];
       temp17 <= A[23:16] + B[23:16];
       temp18 <= A[24:17] + B[24:17];
       temp19 <= A[25:18] + B[25:18];
       temp20 <= A[26:19] + B[26:19];
       temp21 <= A[27:20] + B[27:20];
       temp22 <= A[28:21] + B[28:21];
       temp23 <= A[29:22] + B[29:22];
       temp24 <= A[30:23] + B[30:23];
       temp25 <= A[31:24] + B[31:24];
       Cout <= temp25[8];
       S <= {temp25[7],temp24[7],temp23[7],temp22[7],temp21[7],temp20[7],temp19[7],temp18[7],
             temp17[7],temp16[7],temp15[7],temp14[7],temp13[7],temp12[7],temp11[7],temp10[7],
             temp9[7], temp8[7], temp7[7],temp6[7],temp5[7], temp4[7], temp3[7], temp2[7], temp1[7:0]};
    end

endmodule
```

b. Testbench Results

To test the circuit 203+156 = 359 is simulated. This is the testbench result of the 16-bit Almost Correct Adder:



To test the circuit 52+66 = 118 is simulated. This is the testbench result of the 32-bit Almost Correct Adder:



c. Timing Report

i. For 16-bit implementation

Different values are tried, and best time is found as 6.4:

```
------------------------------------------------------------------------
data required time                                              5.69
data arrival time                                              -5.69
------------------------------------------------------------------------
slack (MET)                                                    0.00

------------------------------------------------------------------------
data required time                                              5.69
data arrival time                                              -5.69
------------------------------------------------------------------------
slack (MET)                                                    0.01

------------------------------------------------------------------------
data required time                                              5.69
data arrival time                                              -5.69
------------------------------------------------------------------------
slack (MET)                                                    0.00
```

ii. For 32-bit implementation

Different values are tried, and best time is found as 1.28:

```
-----------------------------------------------------------
data required time                                  0.58
data arrival time                                  -0.58
-----------------------------------------------------------
slack (MET)                                         0.01


-----------------------------------------------------------
data required time                                  0.58
data arrival time                                  -0.58
-----------------------------------------------------------
slack (MET)                                         0.01


-----------------------------------------------------------
data required time                                  0.58
data arrival time                                  -0.58
-----------------------------------------------------------
slack (MET)                                         0.01
```

d. Area Report

i. For 16-bit implementation

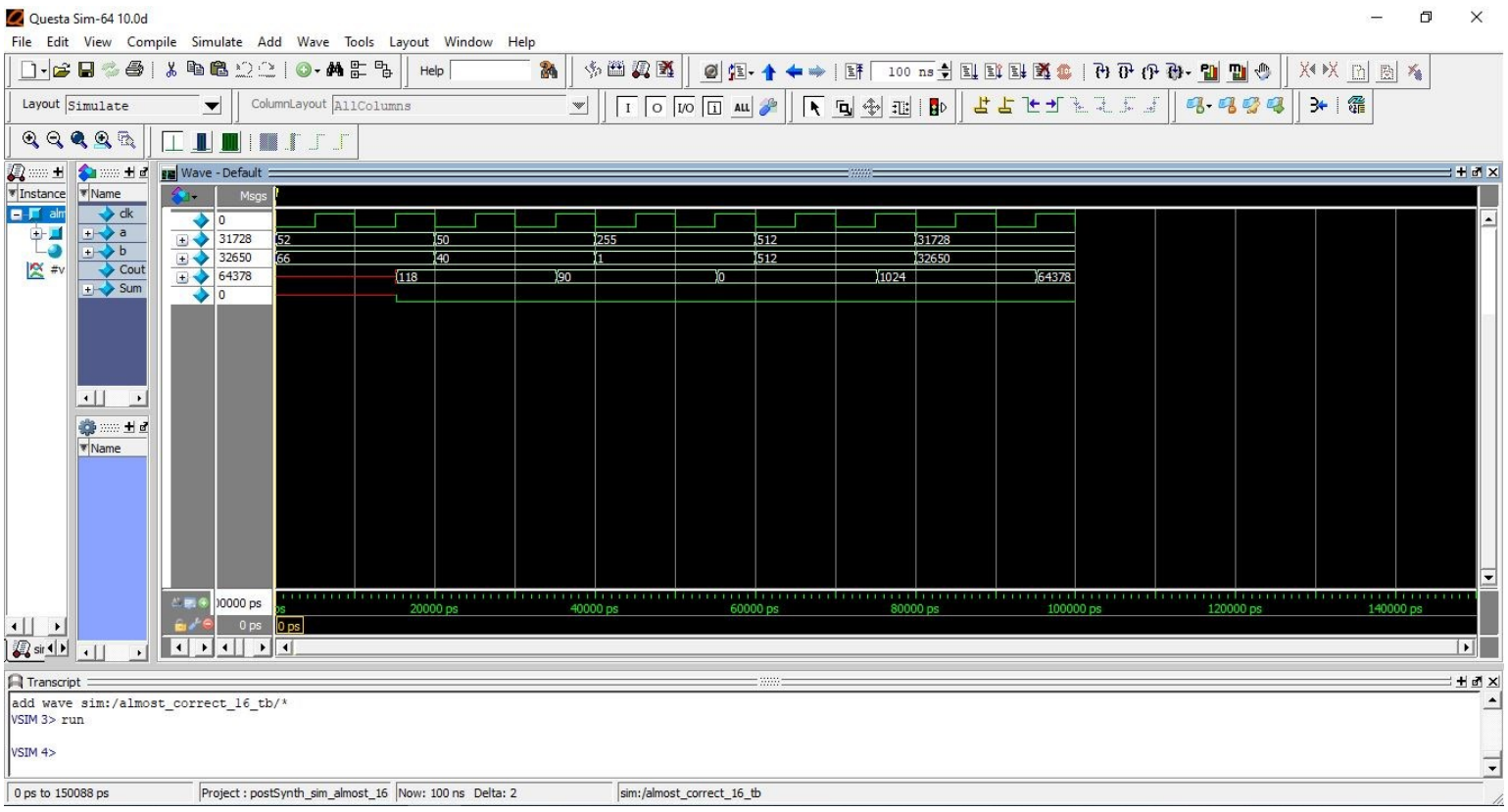
```
Number of ports:                           50
Number of nets:                           195
Number of cells:                           81
Number of combinational cells:             23
Number of sequential cells:                49
Number of macros/black boxes:               0
Number of buf/inv:                         23
Number of references:                      19

Combinational area:                3630.482011
Buf/Inv area:                       795.321002
Noncombinational area:             1493.002007
Macro/Black Box area:                 0.000000
Net Interconnect area:              136.922948

Total cell area:                   5123.484018
Total area:                        5260.406966
```

ii. For 32-bit implementation

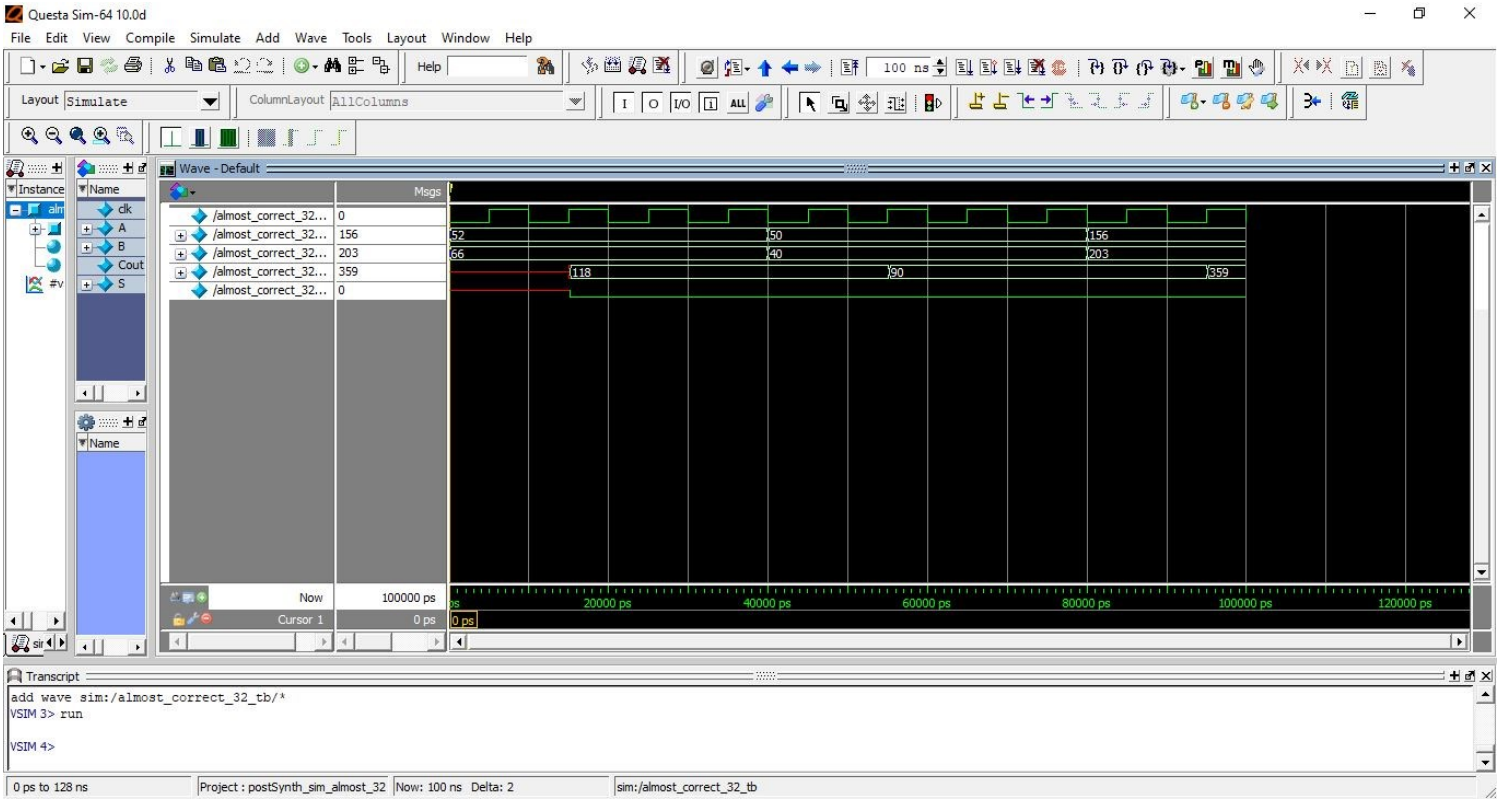
```
Number of ports:                          98
Number of nets:                          288
Number of cells:                         124
Number of combinational cells:            33
Number of sequential cells:               66
Number of macros/black boxes:              0
Number of buf/inv:                        33
Number of references:                     27

Combinational area:                5527.844953
Buf/Inv area:                      1318.802959
Noncombinational area:             2007.258007
Macro/Black Box area:                 0.000000
Net Interconnect area:              286.650441

Total cell area:                   7535.102960
Total area:                        7821.753401
```

e. QuestaSim Simulation Results

i. For 16-bit implementation



ii. For 32-bit implementation



2. Accuracy Configurable Adder

a. Implemented Code

This is the Verilog implementation of the Accuracy Configurable Adder. Full implementation of the 16-bit Accuracy Configurable Adder is done as shown below:

```
module accuracy_configurable_adder_16bit(clk,a,b,Cout,Sum);


   input [15:0] a,b;
   input clk;

   output reg [15:0] Sum;
   output reg Cout;


   wire [8:0] temp1, temp2, temp3;
   reg [15:0] a_reg, b_reg;

   assign temp1 = a_reg[7:0] + b_reg[7:0];
   assign temp2 = a_reg[11:4] + b_reg[11:4];
   assign temp3 = a_reg[15:8] + b_reg[15:8];

   always @ (posedge clk)
   begin

      a_reg <= a;
      b_reg <= b;

      Sum <= {temp3[7:4], temp2[7:4], temp1[7:0]};
      Cout <= temp3[8];

   end




endmodule
```

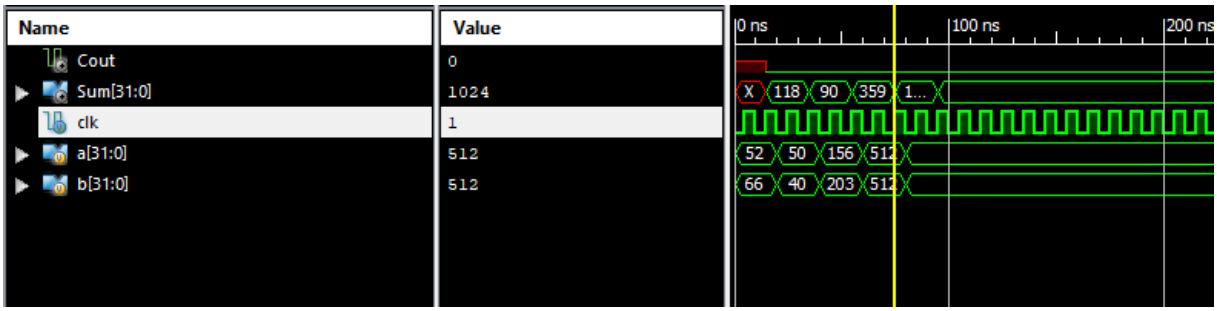
Also, the full implementation of the 32-bit Accuracy Configurable Adder is done as shown below:

```
module accuracy_configurable_adder_32bit(clk,a,b,Cout,Sum);


   input [31:0] a,b;
   input clk;

   output reg [31:0] Sum;
   output reg Cout;


   wire [8:0] temp1, temp2, temp3, temp4, temp5, temp6, temp7;
   reg [31:0] a_reg, b_reg;

   assign temp1 = a_reg[7:0] + b_reg[7:0];
   assign temp2 = a_reg[11:4] + b_reg[11:4];
   assign temp3 = a_reg[15:8] + b_reg[15:8];
   assign temp4 = a_reg[19:12] + b_reg[19:12];
   assign temp5 = a_reg[23:16] + b_reg[23:16];
   assign temp6 = a_reg[27:20] + b_reg[27:20];
   assign temp7 = a_reg[31:24] + b_reg[31:24];

   always @ (posedge clk)
   begin

      a_reg <= a;
      b_reg <= b;

      Sum <= {temp7[7:4], temp6[7:4], temp5[7:4], temp4[7:4], temp3[7:4], temp2[7:4], temp1[7:0]};
      Cout <= temp7[8];


   end


endmodule
```

b. Testbench Results

To test the circuit 32650+31728 = 64378 is simulated. This is the testbench result of the 16-bit Accuracy Configurable Adder:

To test the circuit 512+512 = 1024 is simulated. This is the testbench result of the 32-bit Accuracy Configurable Adder:



c. Timing Report

i. For 16-bit implementation

Different values are tried, and best time is found as 3:

```
------------------------------------------------------------------------------
data required time                                                      2.29
data arrival time                                                      -2.29
------------------------------------------------------------------------------
slack (MET)                                                             0.00

------------------------------------------------------------------------------
data required time                                                      2.29
data arrival time                                                      -2.29
------------------------------------------------------------------------------
slack (MET)                                                             0.00

------------------------------------------------------------------------------
data required time                                                      2.30
data arrival time                                                      -2.29
------------------------------------------------------------------------------
slack (MET)                                                             0.01
```

ii. For 32-bit implementation

Different values are tried, and best time is found as 3.5:

```
---------------------------------------------------------------------------
data required time                                                    2.79
data arrival time                                                    -2.79
---------------------------------------------------------------------------
slack (MET)                                                           0.00


---------------------------------------------------------------------------
data required time                                                    2.79
data arrival time                                                    -2.79
---------------------------------------------------------------------------
slack (MET)                                                           0.00


---------------------------------------------------------------------------
data required time                                                    2.79
data arrival time                                                    -2.79
---------------------------------------------------------------------------
slack (MET)                                                           0.01
```

d. Area Report

i. For 16-bit implementation

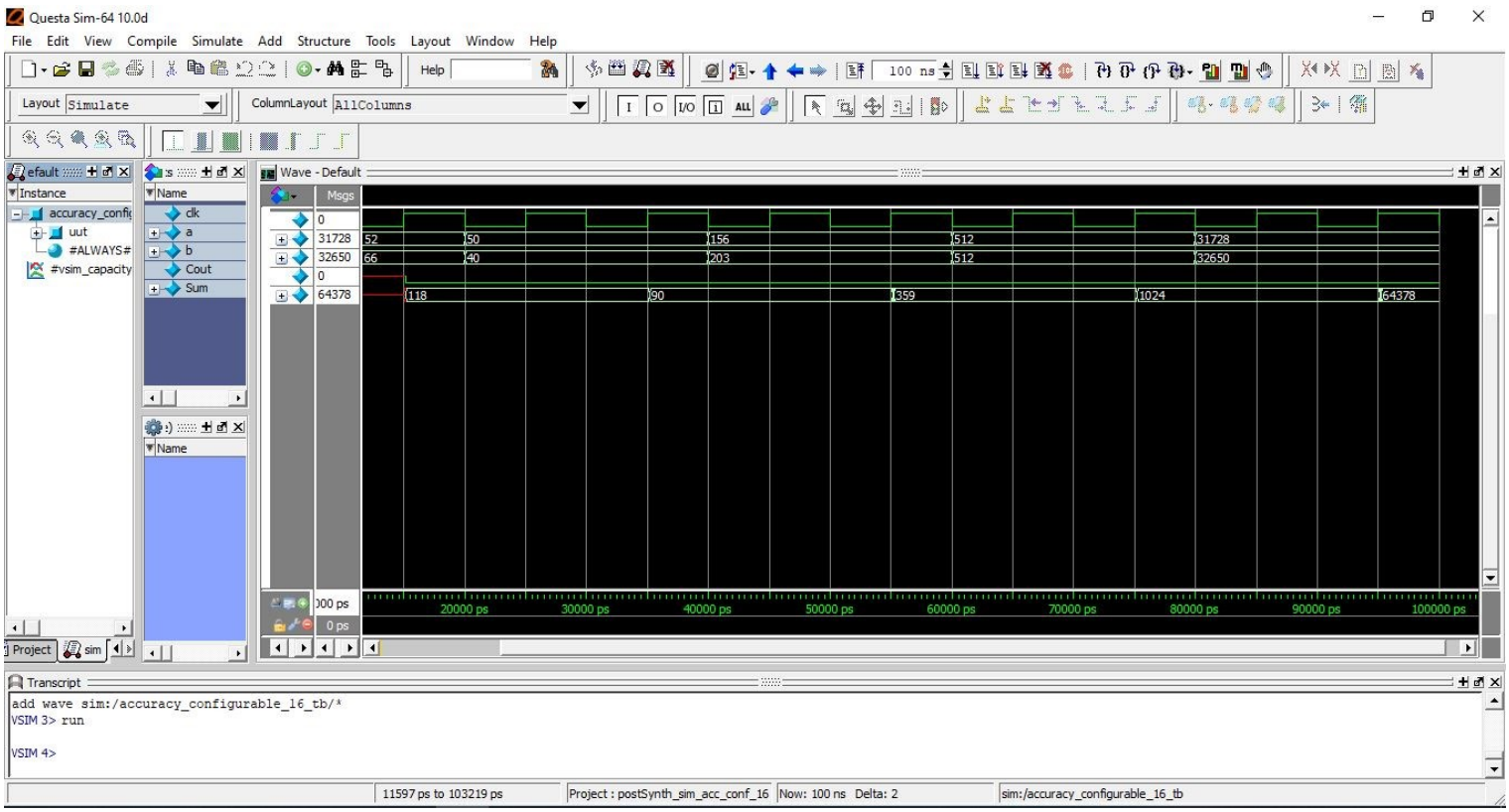
```
Number of ports:                           50
Number of nets:                           185
Number of cells:                           76
Number of combinational cells:             24
Number of sequential cells:                49
Number of macros/black boxes:               0
Number of buf/inv:                         24
Number of references:                       9

Combinational area:               1608.652014
Buf/Inv area:                      356.667011
Noncombinational area:            1498.532009
Macro/Black Box area:                0.000000
Net Interconnect area:             107.291899

Total cell area:                  3107.184023
Total area:                       3214.475922
```

ii. For 32-bit implementation

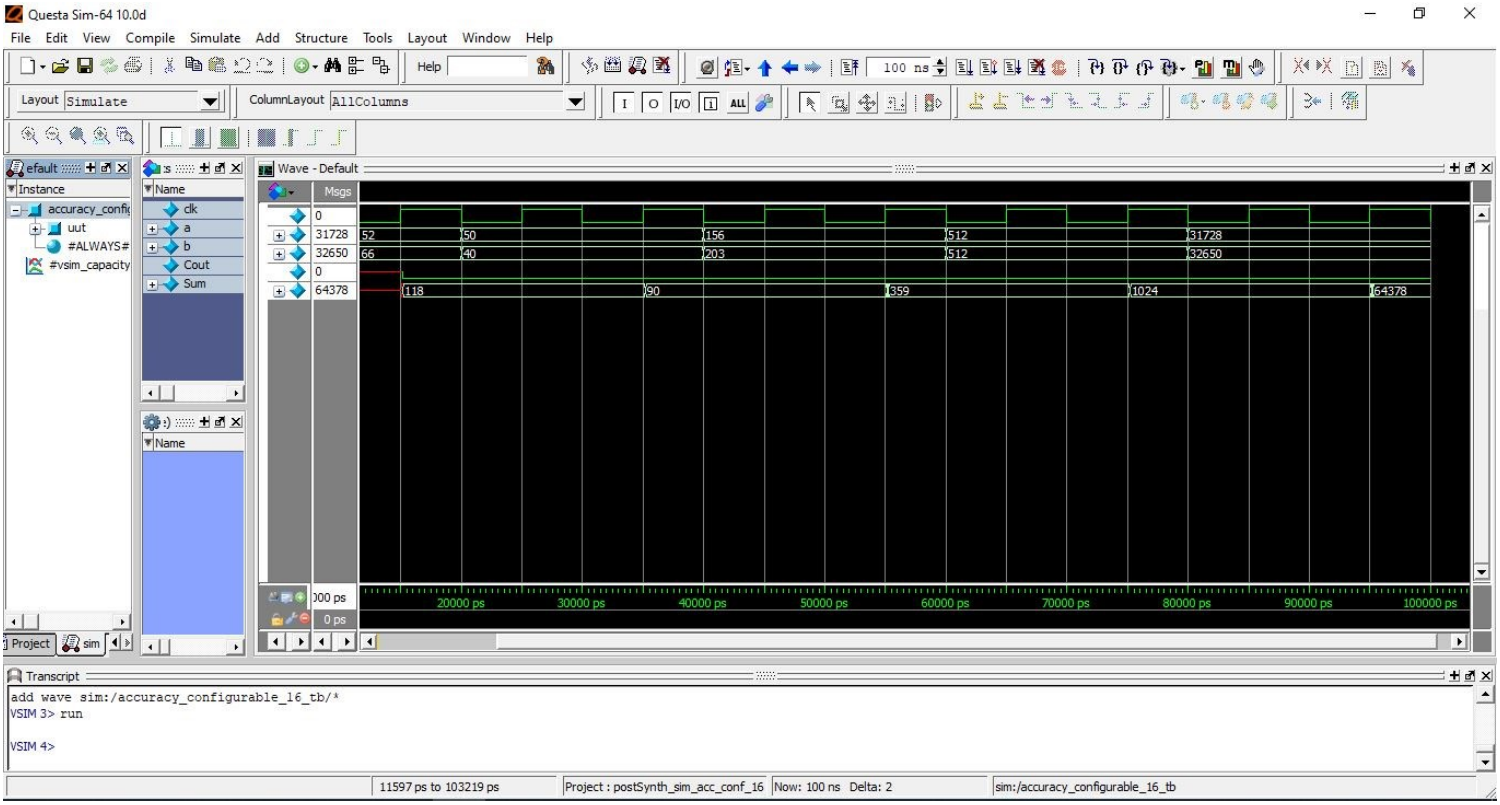
```
Number of ports:                          98
Number of nets:                          329
Number of cells:                         119
Number of combinational cells:            15
Number of sequential cells:               97
Number of macros/black boxes:              0
Number of buf/inv:                        15
Number of references:                      9

Combinational area:                2392.501013
Buf/Inv area:                       298.620011
Noncombinational area:             2950.061010
Macro/Black Box area:                 0.000000
Net Interconnect area:              213.784863

Total cell area:                   5342.562024
Total area:                        5556.346886
```

e. QuestaSim Simulation Results

i. For 16-bit implementation



ii. For 32-bit implementation



3. Discussion about comparison of circuits about timing and area

In this lab assignment 4 different versions of approximate adders are implemented in Verilog. Then, they are synthesised to achieve netlist and timing & area reports. Finally, designs are simulated for verification in QuestaSim as post-synthesis simulations.

This is the table for best timing and area results for each approximate adder design:

Timing of;

- 16-bit Almost Correct Adder: 5.69
- 32-bit Almost Correct Adder: 0.58
- 16-bit Accuracy Configurable Adder: 2.29
- 32-bit Accuracy Configurable Adder: 2.79

Area of;

- 16-bit Almost Correct Adder: 5260.40
- 32-bit Almost Correct Adder: 7821.75
- 16-bit Accuracy Configurable Adder: 3214.47
- 32-bit Accuracy Configurable Adder: 5556.34

- With the designs and the simulation results following discussions can be made:

    - Accuracy Configurable Adder works faster than the Almost Correct Adder. In addition to that Accuracy Configurable Adder requires less circuit area than Almost Correct Adder. These are because of the design of the approximate adders. Accuracy Configurable Adder perform the adding operation with using more approximation. This decreases the timing and area requirements. However, the chance to have a result with error in Accuracy Configurable Adder is higher because of its design. Even if Almost Correct Adder also approximates the result, the chance to have a result with error is very low.