Assignment_2

(Under supervision of Eng : Karim Wassem)

Question1]

```
module q1(A,B,C,D,sel,out,out_bar);
input A,B,C,sel;
input [2:0] D;
output reg out,out_bar;
reg and2,or1,xnor1;
always @(*) begin
and2=D[0]&D[1];
or1=and2|D[2];
xnor1=~(A^B^C);
if(sel) begin
 out=xnor1;
 end
else begin
  out=or1;
end
out_bar=~out;
end
endmodule
```

Figure (1) : Verilog code

```
module q1_tb();
reg A,B,C,sel,out_expected;
reg [2:0] D;
wire out,out_bar;
q1 DUT(.A(A),.B(B),.C(C),.D(D),.sel(sel),.out(out),.out_bar(out_bar));
initial
 begin
 D=3'b001; A=1; B=0; C=1; sel=1; out_expected=1;
 #50 D=3'b010; A=1; B=1; C=1; sel=1; out_expected=0;
 #50 D=3'b011; A=1; B=0; C=1; sel=0; out_expected=1;
 #50 D=3'b111; A=0; B=0; C=1; sel=1; out_expected=0;
 #50 D=3'b001; A=1; B=0; C=1; sel=0; out_expected=0;
 #50;
  if(out!=out_expected) begin
  $display("Error,this desin is incorrect!!");
   $stop;
  end
   $stop;
   end    // end of the initial block
initial
 begin
$monitor("D = %b \t A = %b \t B = %b \t C = %b \t sel = %b \t out_expected = %b \t out_dut = %b ",D,A,B,C,sel,out_expected,out);
end
```

Figure (2) : test_bench_code

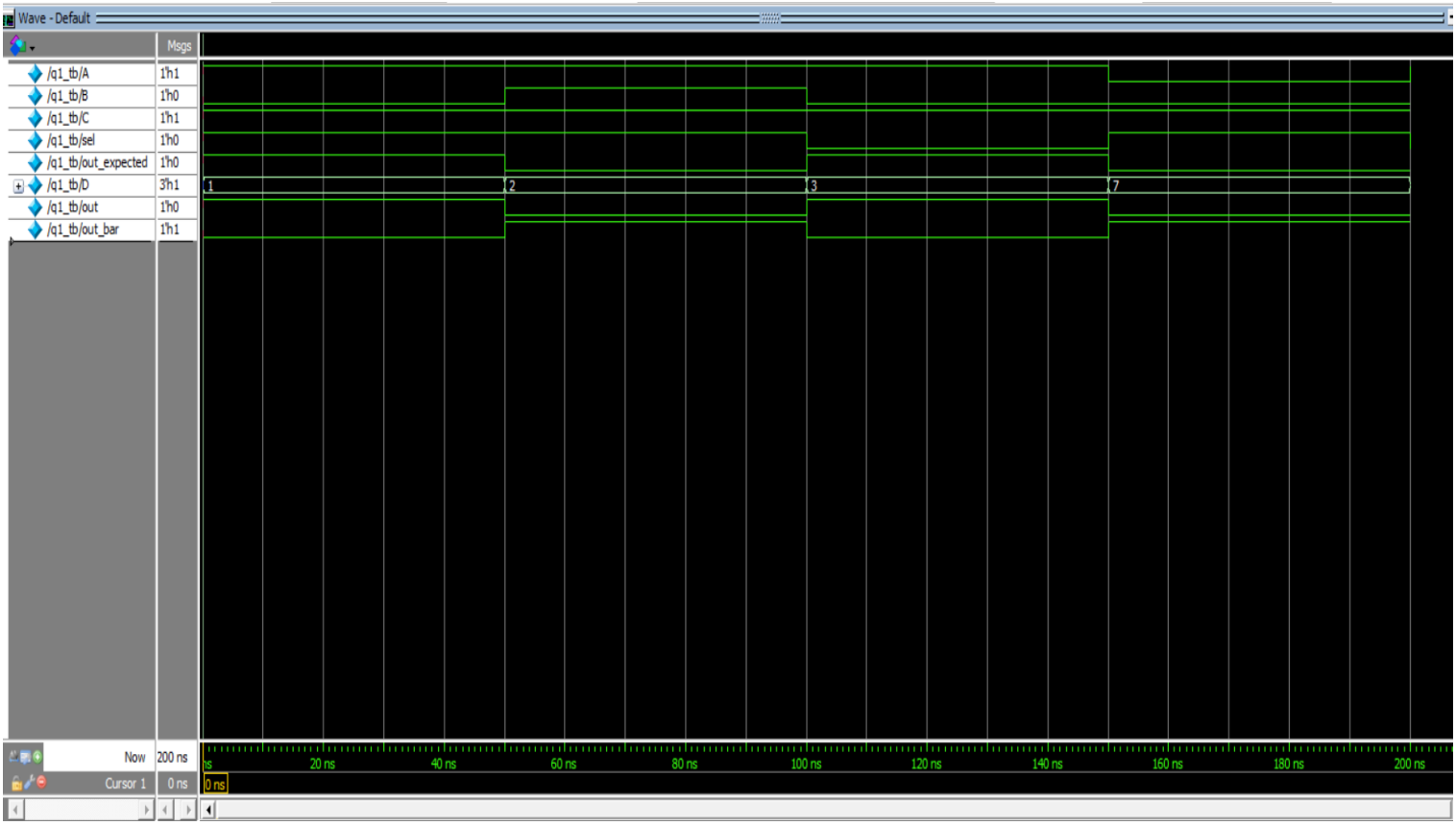

Figure (3) : waveform

-as we notice in figure (2) “testbench” that we use the directed testing method to check the validity of the design and the wave form shown in figure (3) verify the testbench code using .

```
VSIM 24> run -all
# D = 001   A = 1   B = 0   C = 1   sel = 1   out_expected = 1   out_dut = 1
# D = 010   A = 1   B = 1   C = 1   sel = 1   out_expected = 0   out_dut = 0
# D = 011   A = 1   B = 0   C = 1   sel = 0   out_expected = 1   out_dut = 1
# D = 111   A = 0   B = 0   C = 1   sel = 1   out_expected = 0   out_dut = 0
# ** Note: $stop    : D:/VS_code/Verilog codes/Assignment_2/question1/q1_tb(24)
#    Time: 200 ns  Iteration: 0  Instance: /q1_tb
# Break in Module q1_tb at D:/VS_code/Verilog codes/Assignment_2/question1/q1_tb line 24
VSIM 25> quit -sim
```

Figure (4) : transcript

➔ Figure (4) show the transcript & how our design is correct by comparing the expected value with the output value from the DUT.

Question2]

```
module q2(x,y);
input [3:0] x;
output reg [1:0] y;
always @(*) begin
if (x[3]==1)
y=2'b11;
else if(x[2]==1)
y=2'b10;
else if (x[1]==1)
y=2'b01;
else
y=2'b00;
end
endmodule
```

Figure (5) : Verilog code

```
module q2_tb();
reg [3:0] x;
reg [1:0] y_expected;
wire [1:0] y_dut;
q2 DUT (.x(x),.y(y_dut));
initial begin
   x=4'b0001;  y_expected=2'b00;       //start with zero delay
#50 x=4'b0011;  y_expected=2'b01;
#50 x=4'b1011;  y_expected=2'b11;
#50 x=4'b0110;  y_expected=2'b10;
#50 x=4'b1111;  y_expected=2'b11;
#50;
if(y_dut!=y_expected) begin
$display ("Error ,this design is incorrect!!");
$stop;
end
$stop;
end
initial begin
$monitor("input x = %b \t expected_output = %b \t out_dut = %b
",x,y_expected,y_dut);
end
endmodule
```

Figure (6) : test_bench_code

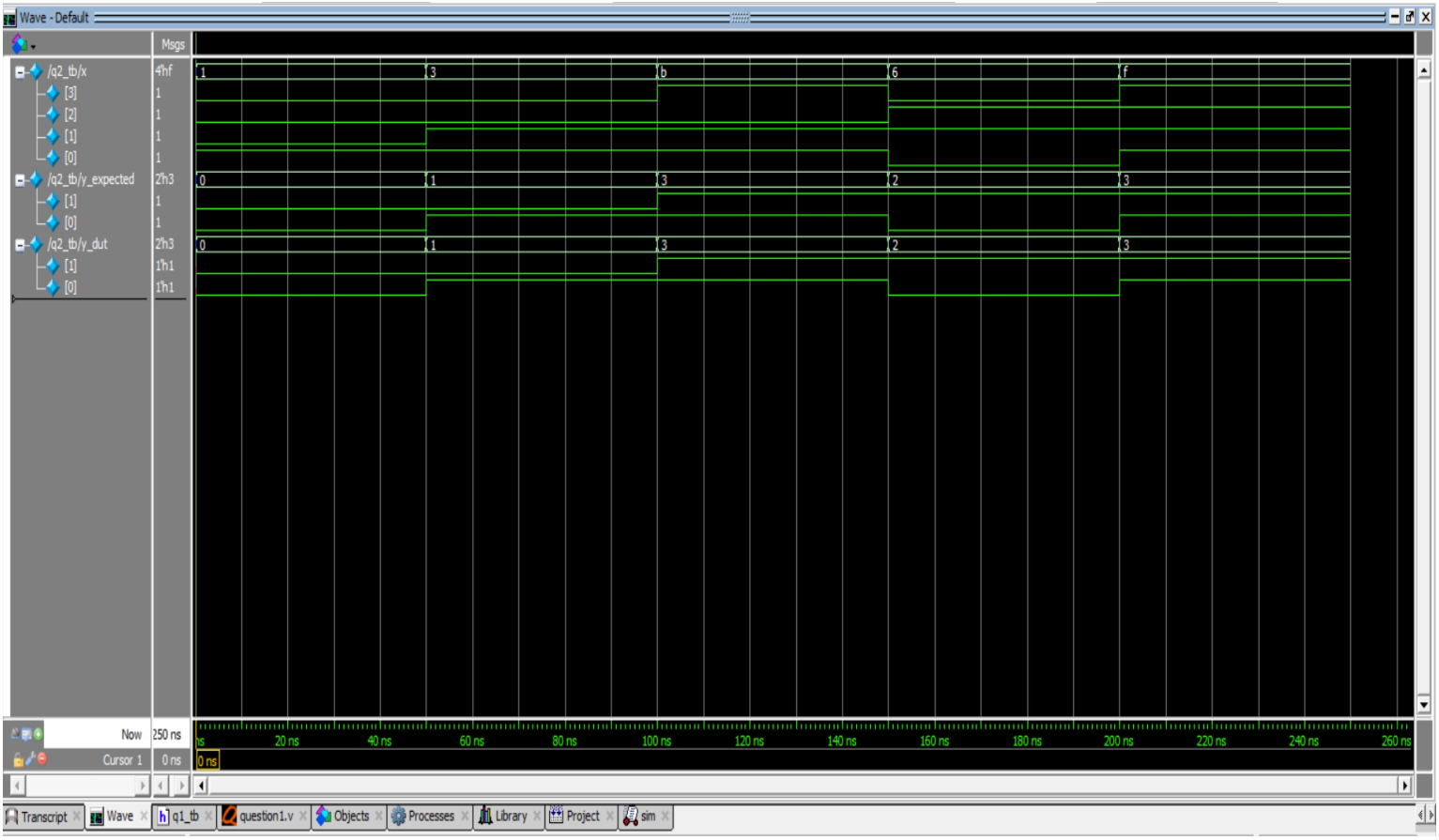

Figure (7) : waveform

```
VSIM 52> run -all
# input x = 0001   expected_output = 00   out_dut = 00
# input x = 0011   expected_output = 01   out_dut = 01
# input x = 1011   expected_output = 11   out_dut = 11
# input x = 0110   expected_output = 10   out_dut = 10
# input x = 1111   expected_output = 11   out_dut = 11
# ** Note: $stop    : D:/VS_code/Verilog codes/Assignment_2/question1/q1_tb(25)
#    Time: 250 ns  Iteration: 0  Instance: /q2_tb
# Break in Module q2_tb at D:/VS_code/Verilog codes/Assignment_2/question1/q1_tb line 25
```

Figure (8) : transcript

➔ Figure (7) & (8) above shows the validity of the design using testbench used in figure (6). Where we use the directed testing with 5 iterations as shown in the waveform.

Question 3]

```
module q3(D,Y);
input [9:0] D;
output reg [3:0] Y;
always @ (*) begin
case (D)
10'b00_0000_0001: Y=4'b0000;
10'b00_0000_0010: Y=4'b0001;
10'b00_0000_0100: Y=4'b0010;
10'b00_0000_1000: Y=4'b0011;
10'b00_0001_0000: Y=4'b0100;
10'b00_0010_0000: Y=4'b0101;
10'b00_0100_0000: Y=4'b0110;
10'b00_1000_0000: Y=4'b0111;
10'b01_0000_0000: Y=4'b1000;
10'b10_0000_0000: Y=4'b1001;
default : Y=4'b0000;
endcase
end
endmodule
```

Figure (9): verilog code

```
module q3_tb();
reg [9:0] d;
reg [3:0] y_expected;
wire [3:0] y_dut;
q3 DUT (.D(d),.Y(y_dut));
initial begin
d=10'b00_1000_0010;  y_expected=4'b0000;
#50 d=10'b00_1000_0000;  y_expected=4'b0111;
#50 d=10'b10_0000_0000;  y_expected=4'b1001;
#50 d=10'b00_0000_0010;  y_expected=4'b0001;
#50 d=10'b00_0001_0000;  y_expected=4'b0100;
#50;
if (y_dut!=y_expected) begin
$display("Error ,this design is incorrect !!");
$stop;
end
$stop;
end
initial begin
$monitor("input D = %b \t expected_output = %b \t out_dut = %b
",d,y_expected,y_dut);
end
endmodule
```

Figure (10) : test_bench_code

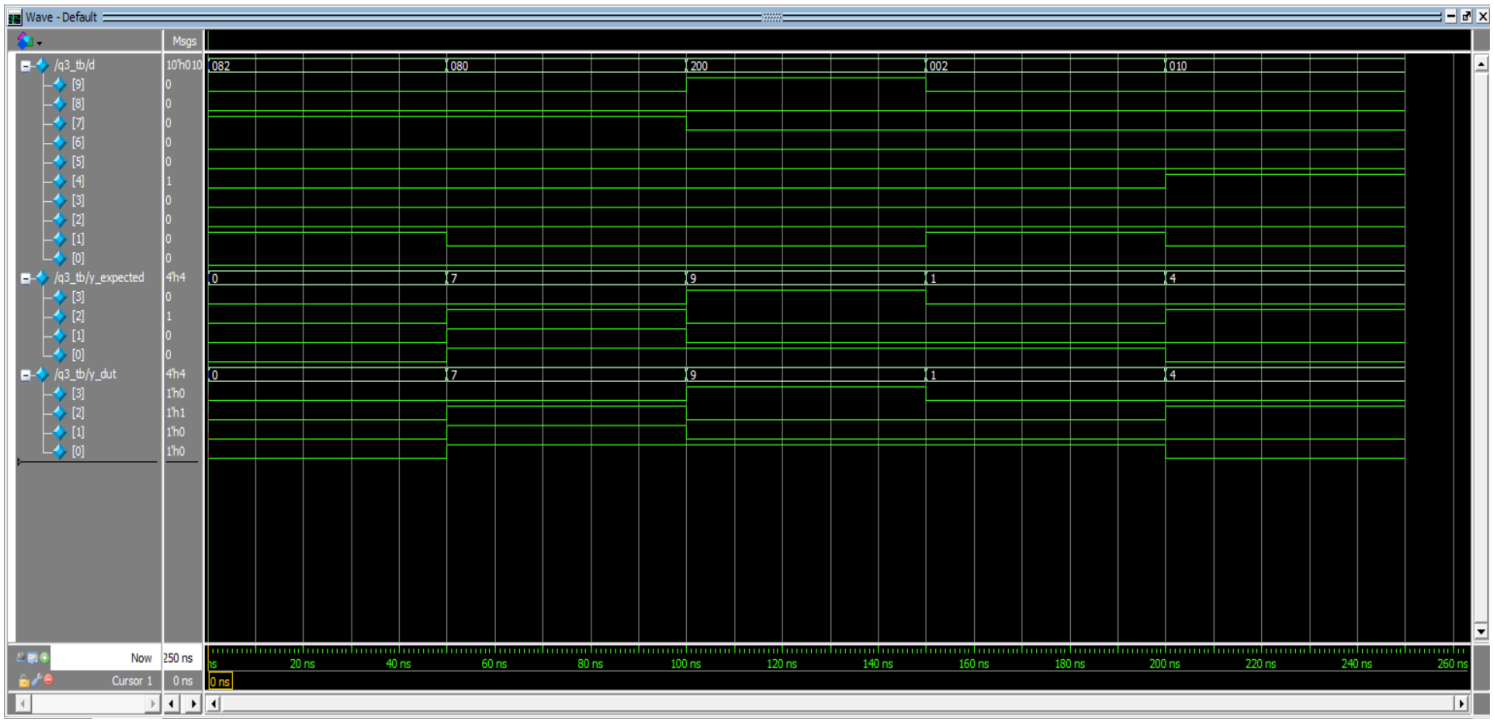

Figure (11) : waveform

```
VSIM 65> run -all
# input D = 0010000010   expected_output = 0000   out_dut = 0000
# input D = 0010000000   expected_output = 0111   out_dut = 0111
# input D = 1000000000   expected_output = 1001   out_dut = 1001
# input D = 0000000010   expected_output = 0001   out_dut = 0001
# input D = 0000010000   expected_output = 0100   out_dut = 0100
# ** Note: $stop    : D:/VS_code/Verilog codes/Assignment_2/question1/q1_tb(26)
#    Time: 250 ns  Iteration: 0  Instance: /q3_tb
# Break in Module q3_tb at D:/VS_code/Verilog codes/Assignment_2/question1/q1_tb line 26
```

Figure (12) :transcript

➔ We test the functionality of the code by using directed testing with 5 iterations where we cover special corner case when input D has two or more bits are active high ⟶ so the output Y=0

Question 4]

```
module q4(A,B,C);
parameter N=1;
input [N-1:0] A , B;
output [N-1:0] C;
assign C= A+B;
endmodule
```

Figure (13) : verilog code

```
module q4_tb();
parameter M=1;
reg [M-1:0]a,b,result_expected;
wire [M-1:0] c;
q4 #(.N(M))DUT (.A(a),.B(b),.C(c));
initial begin
   a=0; b=0; result_expected=0;
#50 a=0; b=1; result_expected=1;
#50 a=1; b=0; result_expected=1;
#50 a=1; b=1; result_expected=0;
#50;
if(c!=result_expected) begin
$display("Error,this design is incorrect!!");
$stop;
end
$stop;
end
initial begin
$monitor("A = %b \t B = %b result_expected= %b\t out_dut =
%b",a,b,result_expected,c);
end
endmodule
```

Figure (14) : test_bench_code

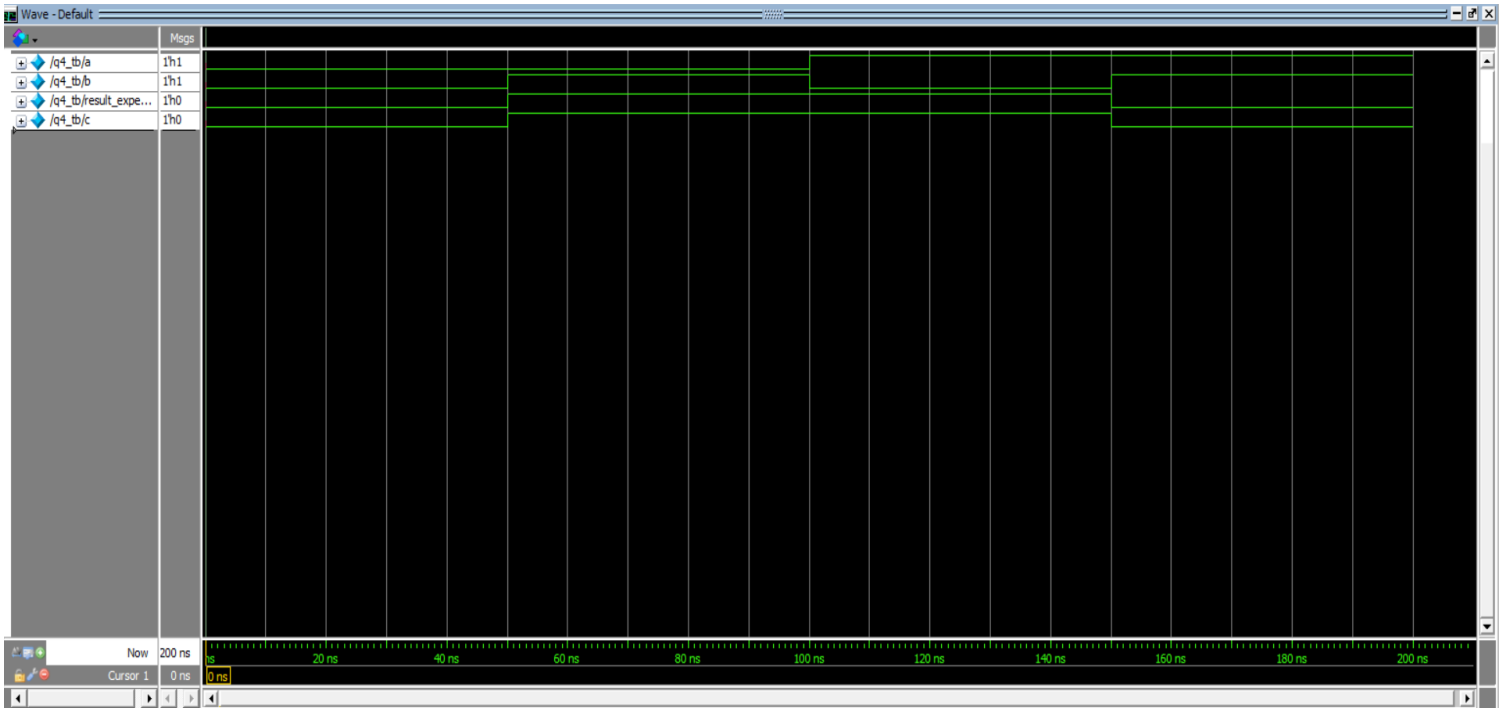

Figure (15) : waveform

```
VSIM 104> run -all
# A = 0   B = 0 result_expected= 0  out_dut = 0
# A = 0   B = 1 result_expected= 1  out_dut = 1
# A = 1   B = 0 result_expected= 1  out_dut = 1
# A = 1   B = 1 result_expected= 0  out_dut = 0
# ** Note: $stop    : D:/VS_code/Verilog codes/Assignment_2/question1/q1_tb(20)
#    Time: 200 ns  Iteration: 0  Instance: /q4_tb
# Break in Module q4_tb at D:/VS_code/Verilog codes/Assignment_2/question1/q1_tb line 20
```

Figure (16): transcript

➔ **In this question we test for 1-bit inputs(A,B) and output (C) so, we have four combinations & we cover all of them in the testbench using directed testing.**

➔ **Where the above figures (15,16) shows that the code is valid & perform well.**

Question 5]

```
module q5(A,B,opcode,result);
parameter N1=4;
input [N1-1:0] A,B;
input [1:0] opcode;
output reg [N1-1:0] result;
wire [N1-1:0] add_value;
q4 #(.N(N1)) half_adder(.A(A),.B(B),.C(add_value));
always @(*) begin
case(opcode)
2'b00: result=add_value;
2'b10: result=A-B;
2'b01: result=A|B;
2'b11: result=A^B;
default: result=0;
endcase
end
endmodule
```

Figure (17) : Verilog code

```
module q5_tb();
parameter M=4;
reg [M-1:0]a,b,result_expected;
reg [1:0] opcode;
wire [M-1:0] result;
q5 #(.N1(M)) DUT(.A(a),.B(b),.opcode(opcode),.result(result));
initial begin
 a=5; b=8; opcode=2'b00; result_expected=13;
#50 a=5; b=4; opcode=2'b01; result_expected=5;
#50 a=2; b=10; opcode=2'b11; result_expected=8;
#50 a=15; b=6; opcode=2'b10; result_expected=9;
#50 a=3; b=8; opcode=2'b00; result_expected=11;
#50 a=6; b=6; opcode=2'b10; result_expected=0;
#50;
if(result!=result_expected) begin
$display("Error , this design is incorrect !!");
$stop;
end
$stop;
end
initial begin
$monitor("A = %b \t B = %b \t opcode = %b \t
result_expected= %b\t out_dut =
%b",a,b,opcode,result_expected,result);
end
endmodule
```

Figure (18) : test_bench_code

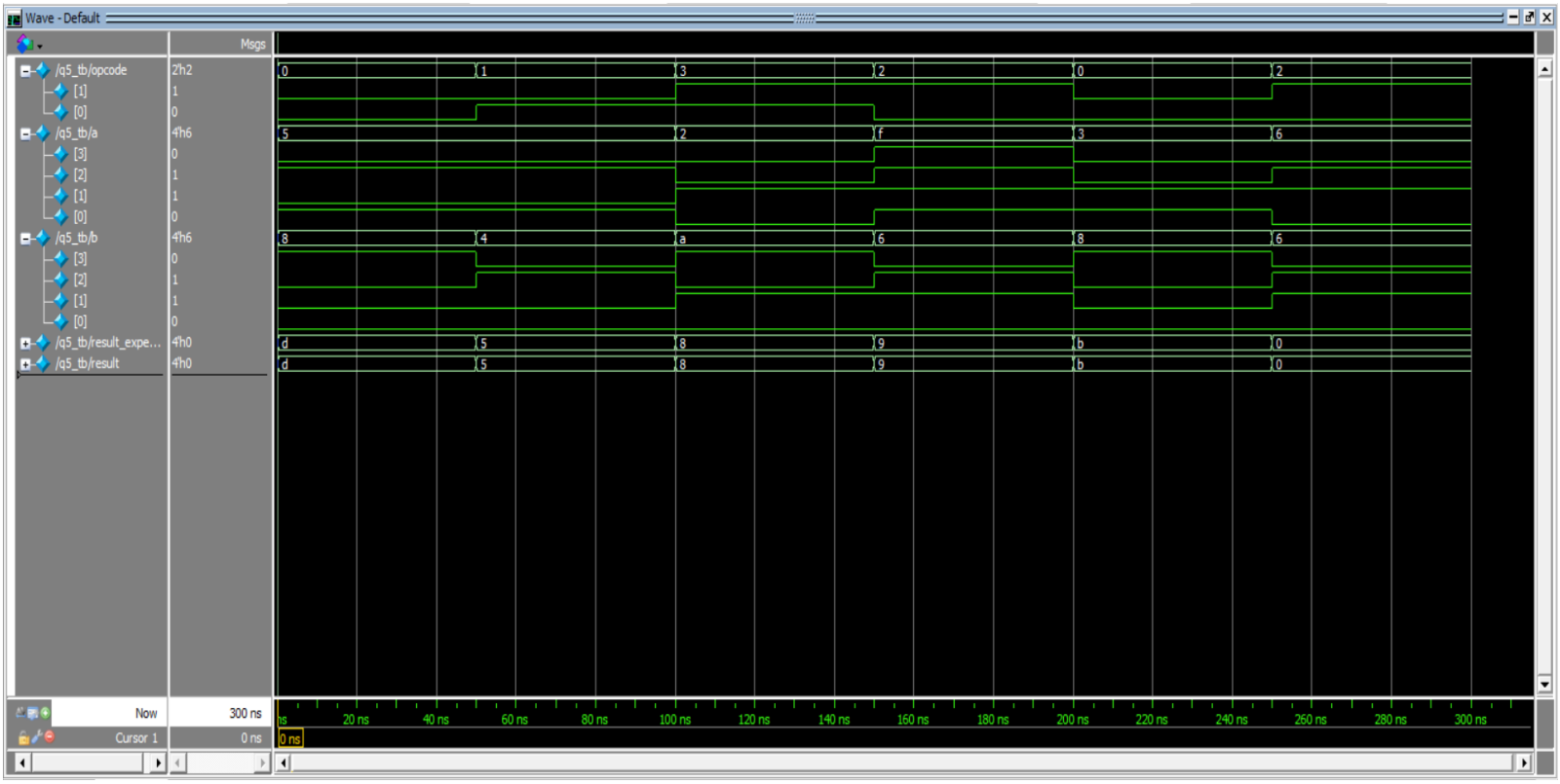

Figure (19) : waveform

```
VSIM 135> run -all
# A = 0101   B = 1000   opcode = 00   result_expected= 1101  out_dut = 1101
# A = 0101   B = 0100   opcode = 01   result_expected= 0101  out_dut = 0101
# A = 0010   B = 1010   opcode = 11   result_expected= 1000  out_dut = 1000
# A = 1111   B = 0110   opcode = 10   result_expected= 1001  out_dut = 1001
# A = 0011   B = 1000   opcode = 00   result_expected= 1011  out_dut = 1011
# A = 0110   B = 0110   opcode = 10   result_expected= 0000  out_dut = 0000
# ** Note: $stop    : D:/VS_code/Verilog codes/Assignment_2/question1/q1_tb(23)
#    Time: 300 ns  Iteration: 0  Instance: /q5_tb
# Break in Module q5_tb at D:/VS_code/Verilog codes/Assignment_2/question1/q1_tb line 23
```

Figure (20): transcript

➔ **Same as previous question we test our code by putting some values (directed testing) and according to the opcode we determine our operation**

➔ **From figure (20) we notice that the expected value equal to the value generated from the design & this indicate that our design is correct.**

Question 6]

```
module q6(A,B,opcode,enable,out);

parameter N2=4;

input [N2-1:0] A,B;

input [1:0] opcode;

input enable;

output reg [6:0] out;

wire [N2-1:0] result;

q5 #(.N1(N2)) ALU
(.A(A),.B(B),.opcode(opcode),.result(result));

always @ (*) begin

if(enable) begin

case(result)

0: out=7'b111_1110;

1: out=7'b011_0000;

2: out=7'b110_1101;

3: out=7'b111_1001;

4: out=7'b011_0011;

5: out=7'b101_1011;

6: out=7'b101_1111;

7: out=7'b111_0000;

8: out=7'b111_1111;

9: out=7'b111_1011;

10:out=7'b111_0111;

11:out=7'b001_1111;

12:out=7'b100_1110;

13:out=7'b011_1101;

14:out=7'b100_1111;

15:out=7'b100_0111;

default : out=0;
```

```
module q6_tb();

parameter M=4;

reg [M-1:0]a,b;

reg [6:0] out_expected;

reg [1:0] opcode;

reg enable;

wire [6:0] out_dut;

q6 #(.N2(M))
DUT(.A(a),.B(b),.opcode(opcode),.enable(enable),.out(out_dut));

initial begin

 a=5; b=8; opcode=2'b00; enable=0;
out_expected=7'b000_0000;

#50 a=5; b=4; opcode=2'b01; enable=1;
out_expected=7'b101_1011;

#50 a=2; b=10; opcode=2'b11; enable=1;
out_expected=7'b111_1111;

#50 a=15; b=6; opcode=2'b10; enable=1;
out_expected=7'b111_1011;

#50 a=3; b=8; opcode=2'b00; enable=1;
out_expected=7'b001_1111;

#50 a=6; b=6; opcode=2'b10; enable=1;
out_expected=7'b111_1110;

#50;

if(out_dut!=out_expected) begin

$display("Error , this design is incorrect !!");

$stop;

end

$stop;

end
```

```
end

else

out=7'b000_0000;

end

endmodule
```

Figure (21) : Verilog code

```
initial begin

$monitor("A = %b \t B = %b \t opcode = %b \t enable = %b\t
out_expected= %b\t out_dut =
%b",a,b,opcode,enable,out_expected,out_dut);

end

endmodule
```

Figure (22) : test_bench_code

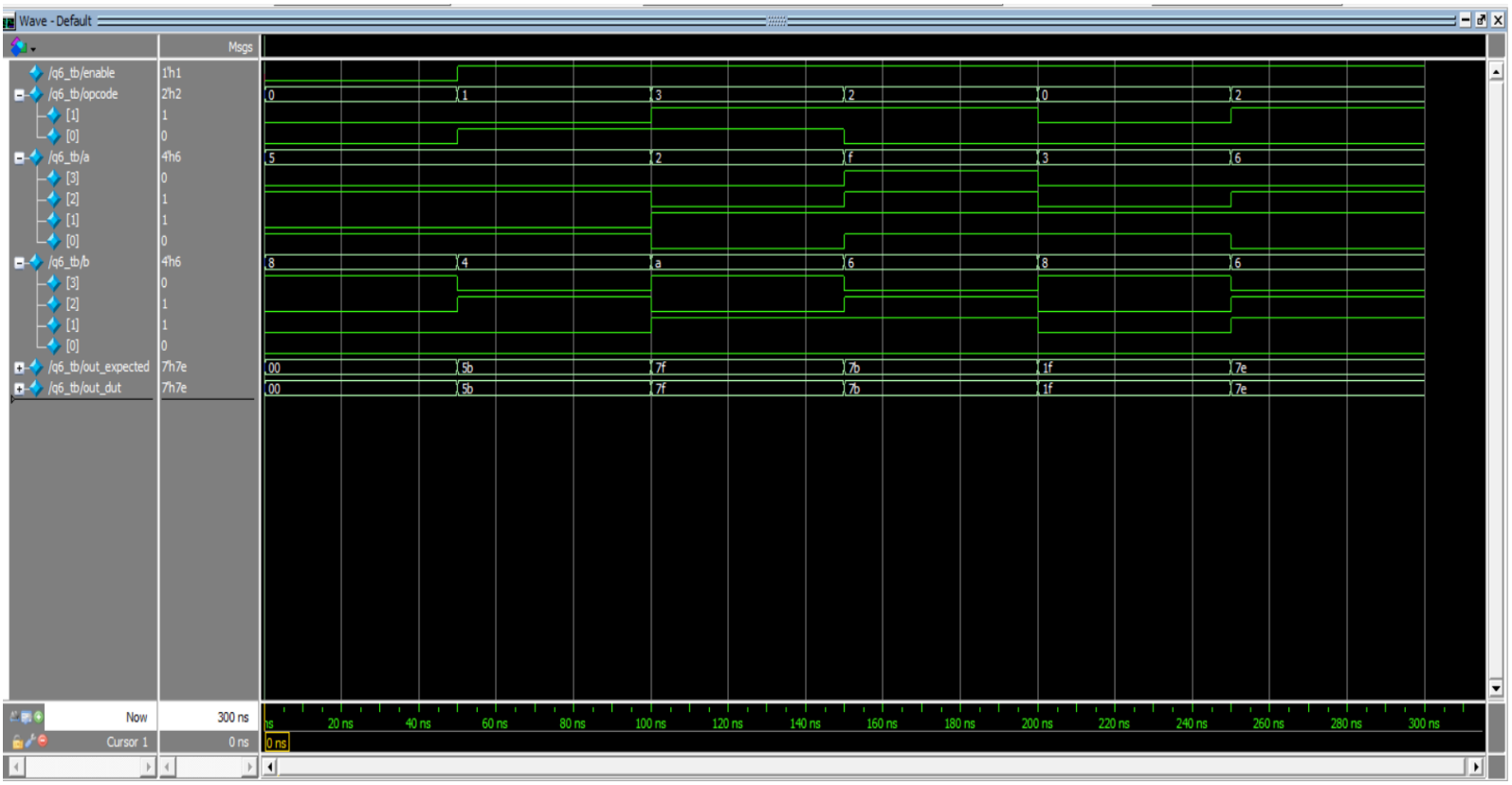

Figure (23): waveform

```
VSIM 142> run -all
# A = 0101   B = 1000   opcode = 00   enable = 0  out_expected= 0000000  out_dut = 0000000
# A = 0101   B = 0100   opcode = 01   enable = 1  out_expected= 1011011  out_dut = 1011011
# A = 0010   B = 1010   opcode = 11   enable = 1  out_expected= 1111111  out_dut = 1111111
# A = 1111   B = 0110   opcode = 10   enable = 1  out_expected= 1111011  out_dut = 1111011
# A = 0011   B = 1000   opcode = 00   enable = 1  out_expected= 0011111  out_dut = 0011111
# A = 0110   B = 0110   opcode = 10   enable = 1  out_expected= 1111110  out_dut = 1111110
# ** Note: $stop    : D:/VS_code/Verilog codes/Assignment_2/question1/q1_tb(25)
#    Time: 300 ns  Iteration: 0  Instance: /q6_tb
# Break in Module q6_tb at D:/VS_code/Verilog codes/Assignment_2/question1/q1_tb line 25
```

Figure (24) : transcript