Assignment2_Extended

(under supervision of Eng: Karim Wassem)

Question1]

```
module q1_ext(A,B);
parameter USE_GRAY=1;
input [2:0] A;
output reg[6:0] B;
always @ (*) begin
if(USE_GRAY==0) begin
case(A)
3'b000: B=7'b000_0000;
3'b001: B=7'b000_0001;
3'b010: B=7'b000_0010;
3'b011: B=7'b000_0100;
3'b100: B=7'b000_1000;
3'b101: B=7'b001_0000;
3'b110: B=7'b010_0000;
3'b111: B=7'b100_0000;
endcase
end
else begin
```

```
module q1_ext_tb();
reg [2:0] a;
wire [6:0] b_hot,b_gray;
integer i;
q1_ext #(.USE_GRAY(0)) DUT1(.A(a),.B(b_hot));
q1_ext #(.USE_GRAY(1)) DUT2(.A(a),.B(b_gray));
initial begin
for (i=0;i<5;i=i+1) begin
a=$random;
#50;
end
$stop;
end
initial begin
$monitor("A = %b \t  out_dut_hot = %b\t
out_dut_gray=%b",a,b_hot,b_gray);
end
endmodule
```

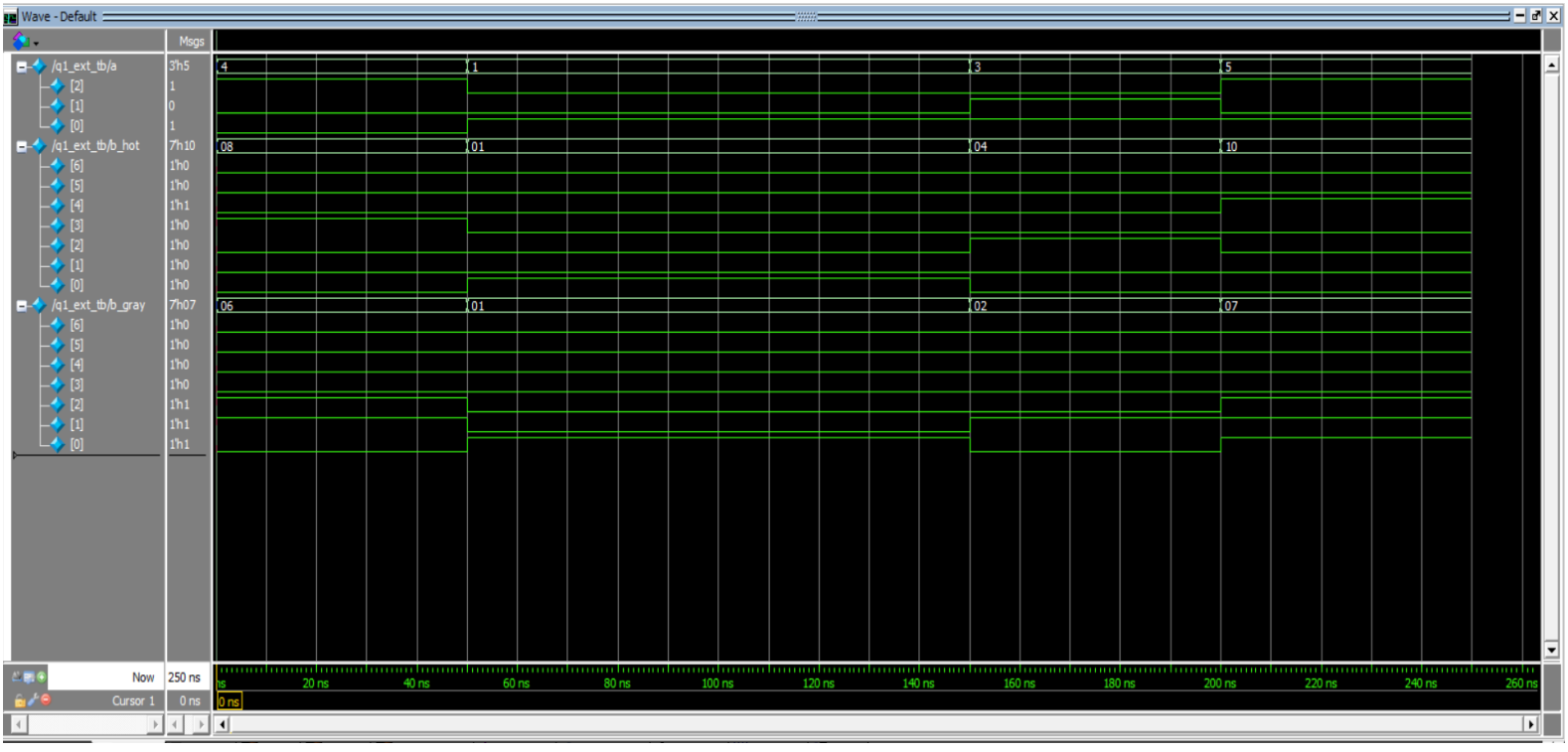
Figure (2) :test_bench_code

```
case(A)
3'b000: B=7'b000_0000;
3'b001: B=7'b000_0001;
3'b010: B=7'b000_0011;
3'b011: B=7'b000_0010;
3'b100: B=7'b000_0110;
3'b101: B=7'b000_0111;
3'b110: B=7'b000_0101;
3'b111: B=7'b000_0100;
endcase
end
end     //end of the always statement
endmodule
```

Figure (1) : Verilog code

```
VSIM 150> run -all
# A = 100    out_dut_hot = 0001000  out_dut_gray=0000110
# A = 001    out_dut_hot = 0000001  out_dut_gray=0000001
# A = 011    out_dut_hot = 0000100  out_dut_gray=0000010
# A = 101    out_dut_hot = 0010000  out_dut_gray=0000111
# ** Note: $stop    : D:/VS_code/Verilog codes/Assignment_2/question1/q1_tb(19)
#    Time: 250 ns  Iteration: 0  Instance: /q1_ext_tb
# Break in Module q1_ext_tb at D:/VS_code/Verilog codes/Assignment_2/question1/q1_tb line 19
```

Figure (3) :transcript



Figure(4) : waveform

Question 2]

```
module q2_ext(D,S,Y);
input D;
input [1:0] S;
output reg [3:0] Y;
always @ (*) begin
case(S)
2'b00:Y={3'b000,D};
2'b01:Y={2'b00,D,1'b0};
2'b10:Y={1'b0,D,2'b00};
2'b11:Y={D,3'b000};
default : Y=4'b0000;
endcase
end
endmodule
```

Figure (5) : Verilog code

```
module q2_ext_tb();
reg [1:0] sel;
reg in;
wire [3:0] out_dut;
integer i;
q2_ext DUT(.D(in),.S(sel),.Y(out_dut));
initial begin
for (i=0;i<10;i=i+1) begin
in=$random;
sel[0]=$random;
sel[1]=$random;
#50;
end
$stop;
end
initial begin
$monitor("D= %b \t  sel = %b\t out_dut=%b",in,sel,out_dut);
end
endmodule
```

Figure (6) : test_bench_code

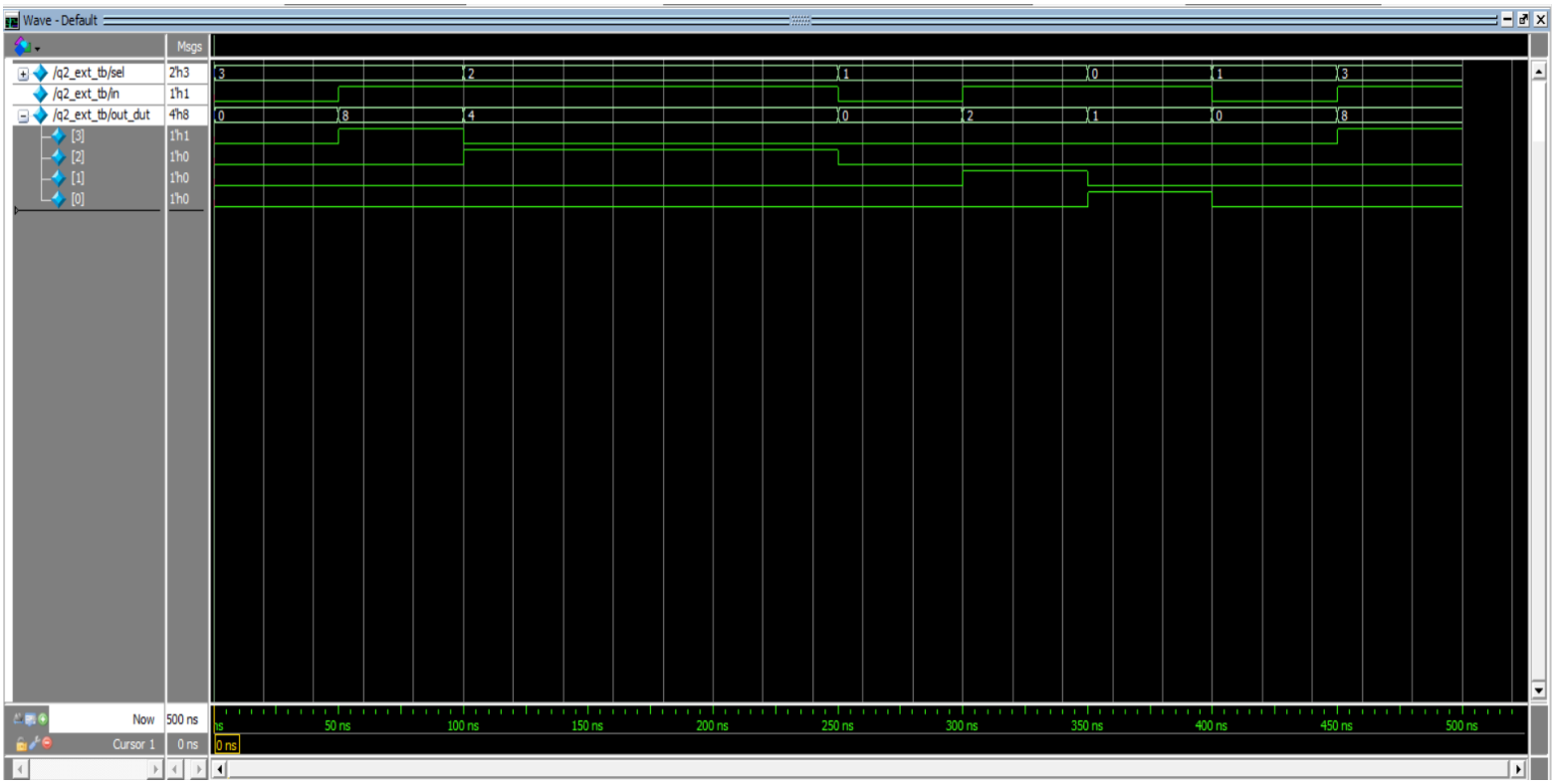


Figure (7) :waveform

```
VSIM 161> run -all
# D= 0    sel = 11  out_dut=0000
# D= 1    sel = 11  out_dut=1000
# D= 1    sel = 10  out_dut=0100
# D= 0    sel = 01  out_dut=0000
# D= 1    sel = 01  out_dut=0010
# D= 1    sel = 00  out_dut=0001
# D= 0    sel = 01  out_dut=0000
# D= 1    sel = 11  out_dut=1000
# ** Note: $stop    : D:/VS_code/Verilog codes/Assignment_2/question1/q1_tb(21)
#    Time: 500 ns  Iteration: 0  Instance: /q2_ext_tb
# Break in Module q2_ext_tb at D:/VS_code/Verilog codes/Assignment_2/question1/q1_tb line 21
```

Figure (8) :transcript

➔ **We test the code by applying some random combination as shown in figure (8) and we deduce that the code perform well.**

➔ **Above waveform plot the random combinations that we generated in the code "fig.(6)"**