DSD LAB

Week3 Assignment Submission

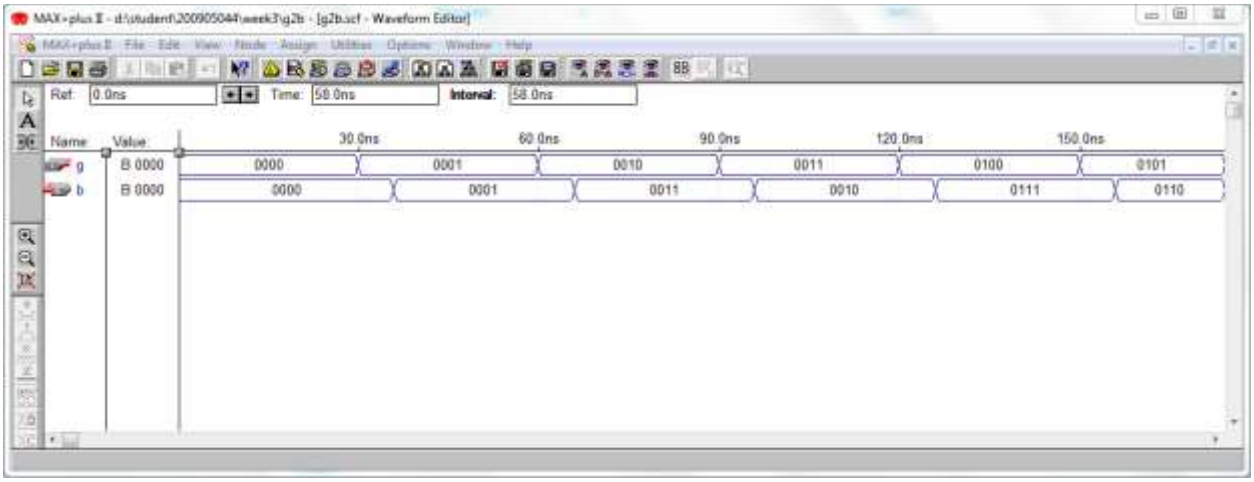
Name: Swamiraju Satya Praveen Varma
Reg No. 200905044
Batch B1
23/11/21
Week 3

1.Using forloop, write behavioral Verilog code to convert an N bit grey code into equivalent binary code.

Verilog code:

```
module g2b(g,b);
parameter n=3;
input[n:0]g;
output[n:0]b;
reg[n:0]b;
integer k;
always@(g)
begin
b[n]=g[n];
for(k=n-1;k>=0;k=k-1)
begin
b[k]=b[k+1]^g[k];
end
end
endmodule
```

Output Waveform:



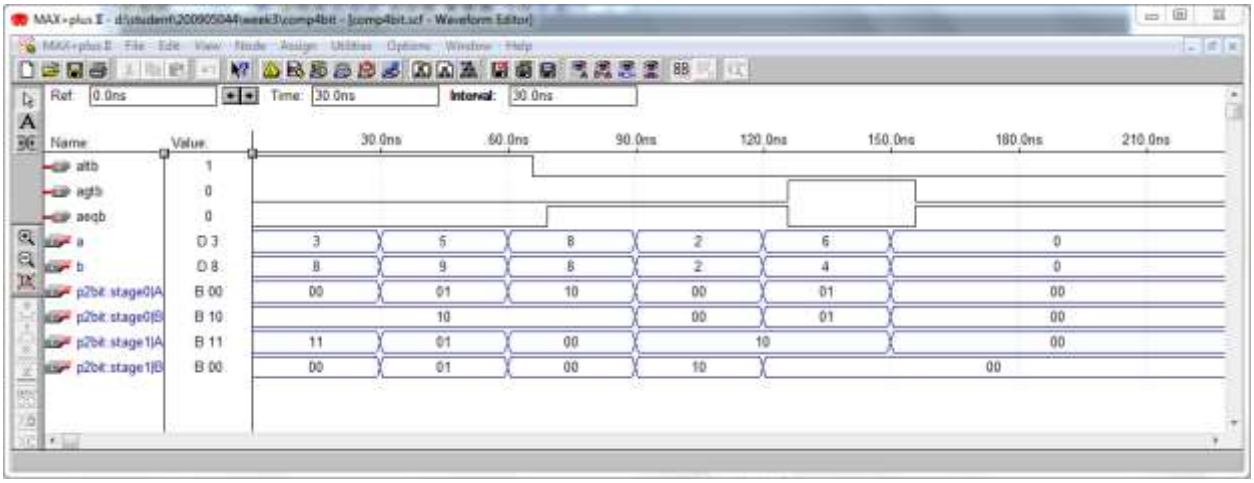
2) Write and simulate the Verilog code for a 4-bitcomparator using 2-bit comparators.

Verilog code:

```
module comp2bit (A, B, AeqB, AgtB, AltB);
input [1:0] A;
input [1:0] B;
output AeqB, AgtB, AltB;
wire [1:0] i;
assign i[1] = ~(A[1] ^ B[1]); //A[0] XNOR B[0]
assign i[0] = ~(A[0] ^ B[0]); //A[1] XNOR B[1]
assign AeqB = i[1] & i[0];
assign AgtB = (A[1] & ~B[1]) | (i[1] & A[0] & ~B[0]);
assign AltB = ~(AeqB | AgtB);
```

```
endmodule
module comp4bit (a, b, aeqb, agtb, altb);
input [3:0] a;
input [3:0] b;
output aeqb, agtb, altb;
wire AeqB1,AgtB1,AltB1;
comp2bit stage0 (a[3:2], b[3:2], AeqB1, AgtB1, AltB1);
comp2bit stage1 (a[1:0], b[1:0], AeqB2, AgtB2, AltB2);
assign aeqb = AeqB1 & AeqB2;
assign agtb = AgtB1 | (AeqB1 & AgtB2);
assign altb = ~(aeqb | agtb);
endmodule
```

Output Waveform:



3.Write behavioral Verilog code for

- *an 8 to 1 multiplexer using casestatement*
- *a 2to 1 multiplexer using the if-elsestatement.*

Using the above modules write the hierarchical code for a 16 to 1 multiplexer.

Verilog code:

```
module mux16to1(w,s,f);
input [15:0]w;

input [3:0]s;
wire [1:0]x;
output f;
mux8to1s stage0(w[7:0],s[2:0],x[0]);
mux8to1s stage1(w[15:8],s[2:0],x[1]);
mux2to1s stage2(x[0],x[1],s[3],f);
endmodule

module mux2to1s(w0,w1,s,f);
input w0,w1,s;
output f;
reg f;
always @(w0 or w1 or s)
begin
if(s==0)
assign f = w0;
else
assign f = w1;
end
endmodule

module mux8to1s(w,s,f);
input [7:0]w;
input [2:0]s;
output f;
reg f;
always @(w or s or f)
begin
case(s) 0:f=w[0];
1:f=w[1];

2:f=w[2];

3:f=w[3];
```

4:f=w[4];

5:f=w[5];

6:f=w[6];

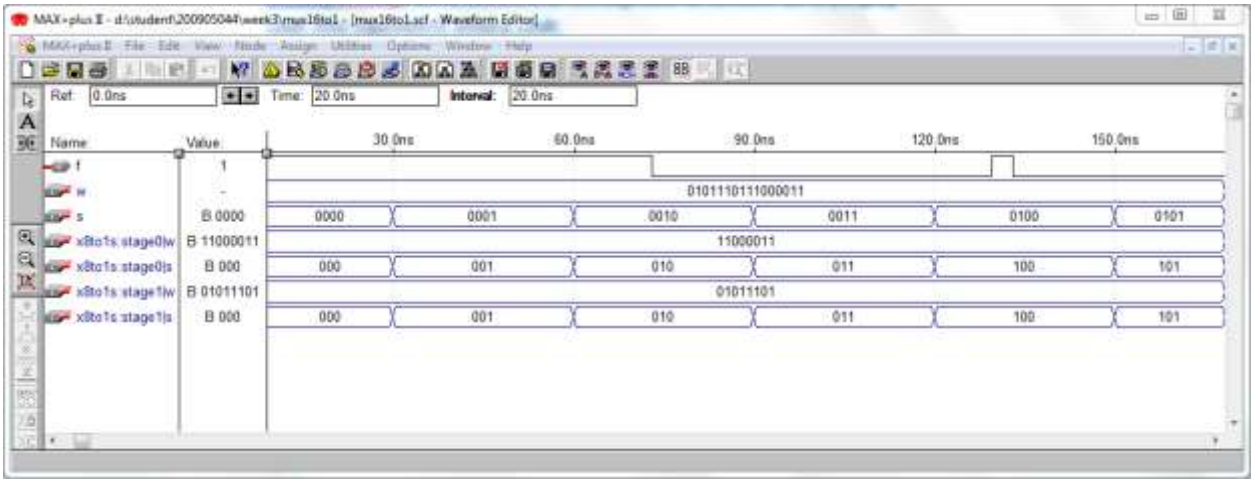
7:f=w[7];

endcase

end

endmodule

Output Waveform:



THANK YOU