

Roll No :- 160217235-039

Name :- K. Sai Charan

Design and simulation of HalfAdder in verilog.

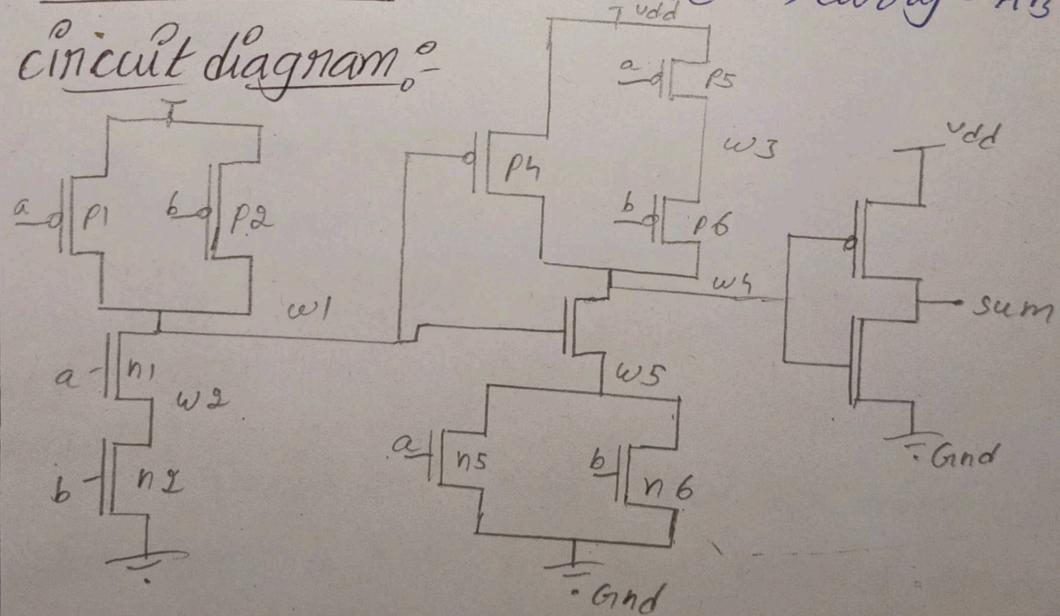
Aim :- Implement and verify the half adder using switch level.

Apparatus Required :- pc, eda playground online simulator.

Theory :- Half adder is a combinational circuit with 2 inputs and 2 outputs. It is designed to add two single bit binary numbers. It is a basic building block of single-bit addition gives 2 output sum and carry.

Boolean equation :- sum = $A \oplus B$, carry = AB .

Circuit diagram :-



Truth table :-

A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Verilog program :-

```
module ha_sw(a,b,sum,carry);  
    input a,b;  
    output sum,carry;  
    supply1 vdd;  
    supply0 gnd;  
    pmos p1 (w1, vdd, b);  
    pmos p2 (w1, vdd, b);  
    nmos n1 (w1, w2, a);  
    nmos n2 (w2, gnd, b);  
    pmos p3 (carry, vdd, w1);  
    nmos n3 (carry, gnd, w1);  
    pmos p4 (w4, vdd, w1);  
    pmos p5 (w3, vdd, a);  
    pmos p6 (w4, w3, b);  
    nmos n4 (w4, w5, w1);  
    nmos n5 (w5, gnd, a);  
    nmos n6 (w5, gnd, b);
```

```

pmos p7 (sum, vdd, w4);
nmos n7 (sum, gnd, w4);

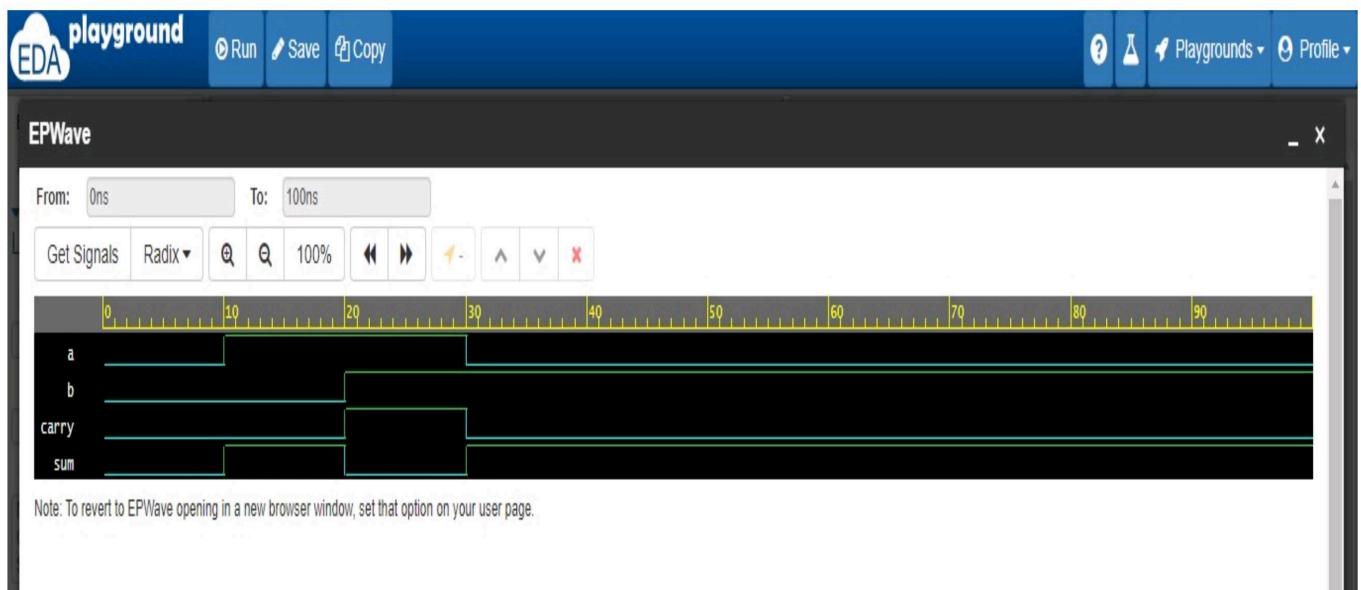
end module

Test bench :
module tb_ha_sw;
reg a,b;
wire sum, carry;
ha_sw ut(a,b,sum,carry);
initial
begin
    a=0;b=0;
    #10 a=1;
    #10 b=1;
    #10 a=0;
end
initial
begin
    $dumpfile("test.vcd");
    $dumpvars(1);
    #100 $finish;
end
endmodule

```

Result :- Design and simulation of half adder using switch level is done and also verified the output waveform for the given sample inputs.

HALF ADDER



Roll no: 160217735-039

Name: K. Sairagan

dim :-

To implement and verify the functionality of full adder to switch level in verilog.

Apparatus required:- pc, eda playground online

simulation.

Theory :-

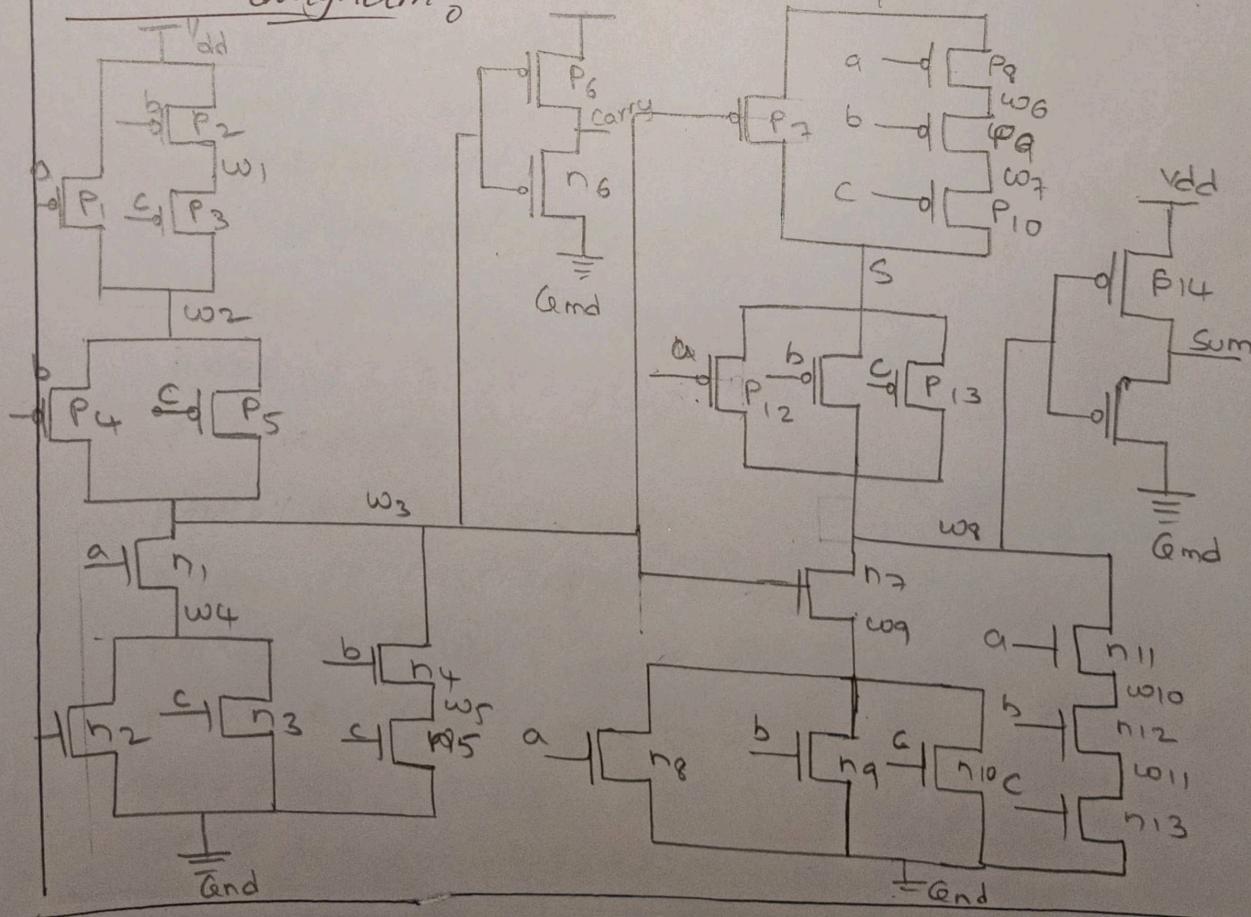
An adder with 3 inputs and 2 outputs.

It is designed to add 3 one bit inputs and gives the sum as output along with carry.

Boolean equations :-

$$\text{sum} = A \oplus B \oplus C, \text{carry} = AB + BC + CA$$

Circuit diagram :-



Truth table :-

A _{in}	B	C _{in}	sum	carry
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Verilog program :-

```
module fa_sw(a,b,cin,sum,carry);  
    input a,b,cin;  
    output sum,carry;  
    supply1 vdd;  
    supply0 gnd;  
  
    PMOS p1(w2,vdd,a);  
    PMOS p2(w1,vdd,b);  
    PMOS p3(w2,w1,cin);  
    PMOS p4(w4,w3,b);  
    PMOS p5(w3,w2,cin);  
    NMOS n1(w3,w4,a);  
    NMOS n2(w4,gnd,b);  
    NMOS n3(w3,w5,b);  
    NMOS n4(w3,w5,b);
```

```
nmos n5(w5,gnd,cin);  
pmos p6(carrys,vdd,w3);  
nmos n6(carrys,gnd,w3);  
pmos p7(s,vdd,w3);  
pmos p8(w6,vdd,a);  
pmos p9(w7,w6,b);  
pmos p10(s,w7,cin);  
pmos p11(w8,s,a);  
pmos p12(w8,s,b);  
pmos p13(w8,s,cin);  
nmos n7(w8,w9,w3);  
nmos n8(w9,gnd,a);  
nmos n9(w9,gnd,b);  
nmos n10(w9,gnd,cin);  
nmos n11(w8,w10,a);  
nmos n12(w10,w11,b);  
nmos n13(w11,gnd,cin);  
pmos p14(sum,vdd,w8);  
nmos n15(sum,gnd,w8);
```

end module

Test bench

module tb_fa_sw;

reg a,b,cin;

wire sum,carry;

fa_sw uut(a,b,cin,sum,carry);

initial

begin

a=0;b=0;cin=0;

```

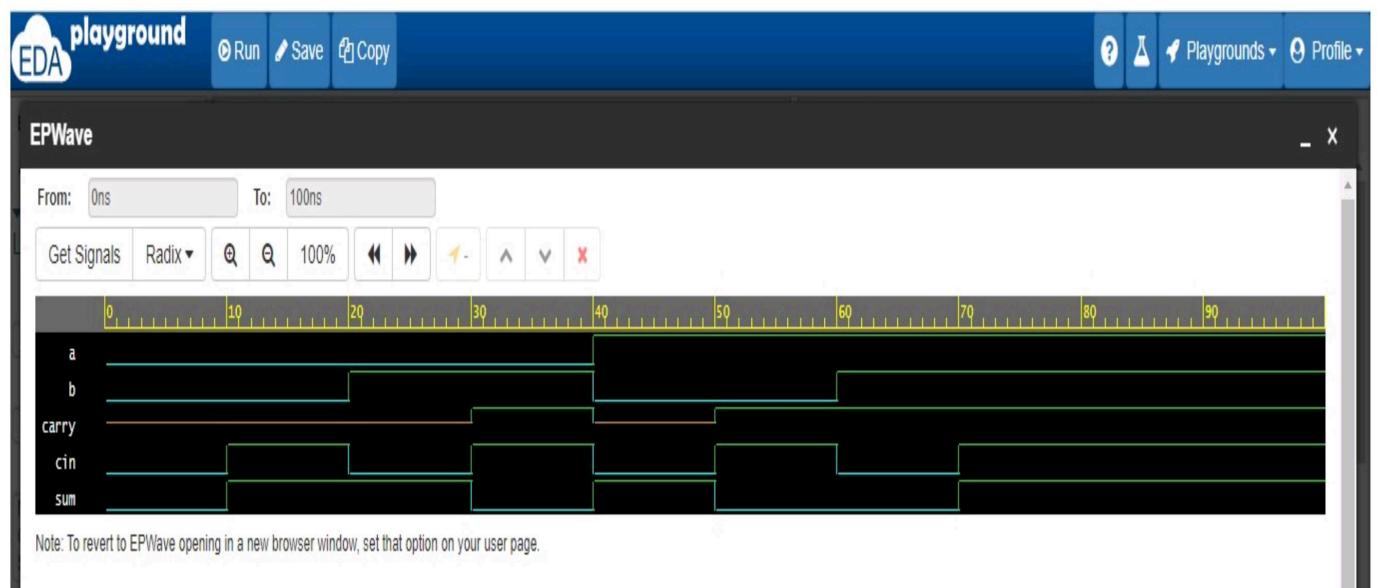
#10 a=0; b=0; cin=1;
#10 a=0; b=1; cin=0;
#10 a=0; b=1; cin=1;
#10 a=1; b=0; cin=0;
#10 a=1; b=0; cin=1;
#10 a=1; b=1; cin=0;
#10 a=1; b=1; cin=1;

initial
begin
    $dumpfile("test.vcd");
    $dumpvars(1);
    #100 $finish;
end
endmodule

```

Result — Design and simulation of full adder combinational circuit is done and also verified the output waveform for the given sampled inputs.

FULL ADDER



Design and verification of 2 to 4 decoder.

Aim:- To implement and verify 2 to 4 decoder functionality using switch level in verilog.

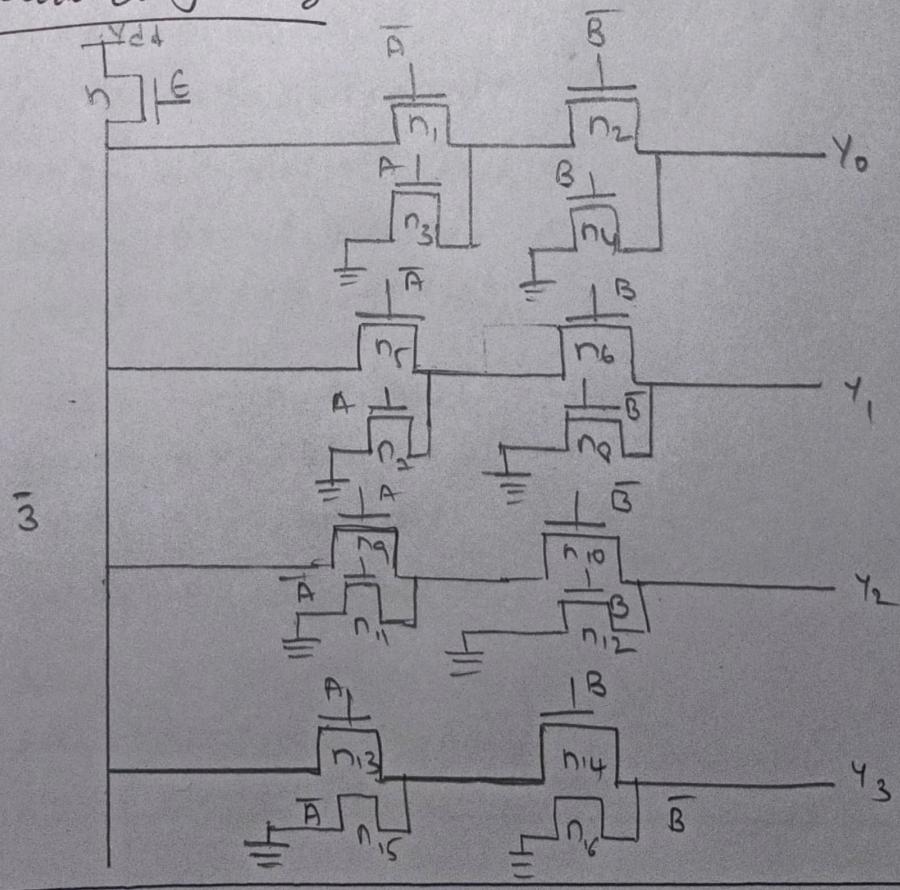
Apparatus:- pc, eda playground online simulator.

Theory:- Decoder is a combinational circuit which has 'n' input lines and maximum 2^n output lines. For given input combinations one of the respective output becomes high for 2 to 4 decoder, it has 2 input and 4 outputs.

Boolean equation:-

$$y_0 = E \cdot \bar{A}_1 \cdot \bar{A}_0, y_1 = E \cdot \bar{A}_1 \cdot A_0, y_2 = E \cdot A_1 \cdot \bar{A}_0, y_3 = E \cdot A_1 \cdot A_0$$

Circuit diagram:-



Truth table:

Enable	Inputs		outputs			
E	A ₁	A ₀	Y ₃	Y ₂	Y ₁	Y ₀
0	x	x	0	0	0	0
1	0	0	0	0	0	1
1	0	1	0	0	1	0
1	1	0	0	1	0	0
1	1	1	1	0	0	0

verilog program:

```

module dec2to4_sw(a,b,en,y);
    input a,b,en;
    output [3:0]y;
    supply1 vdd;
    supply0 gnd;
    nmos n1(w1,vdd,en);
    nmos n1(w2,w1,(na));
    nmos n2(y{0},w2,(nb));
    nmos n3(w2,gnd,a);
    nmos n4(y{0},gnd,b);
    nmos n5(w3,w1,(na));
    nmos n6(y{1},w3,b);
    nmos n7(w3,gnd,a);
    nmos n8(y{1},gnd,(nb));
    nmos n9(w4,w1,a);
    nmos n10(y{2},w4,(nb));
    nmos n11(w4,gnd,(na));

```

```

nmos n16(y{3},gnd,(vb));
end module

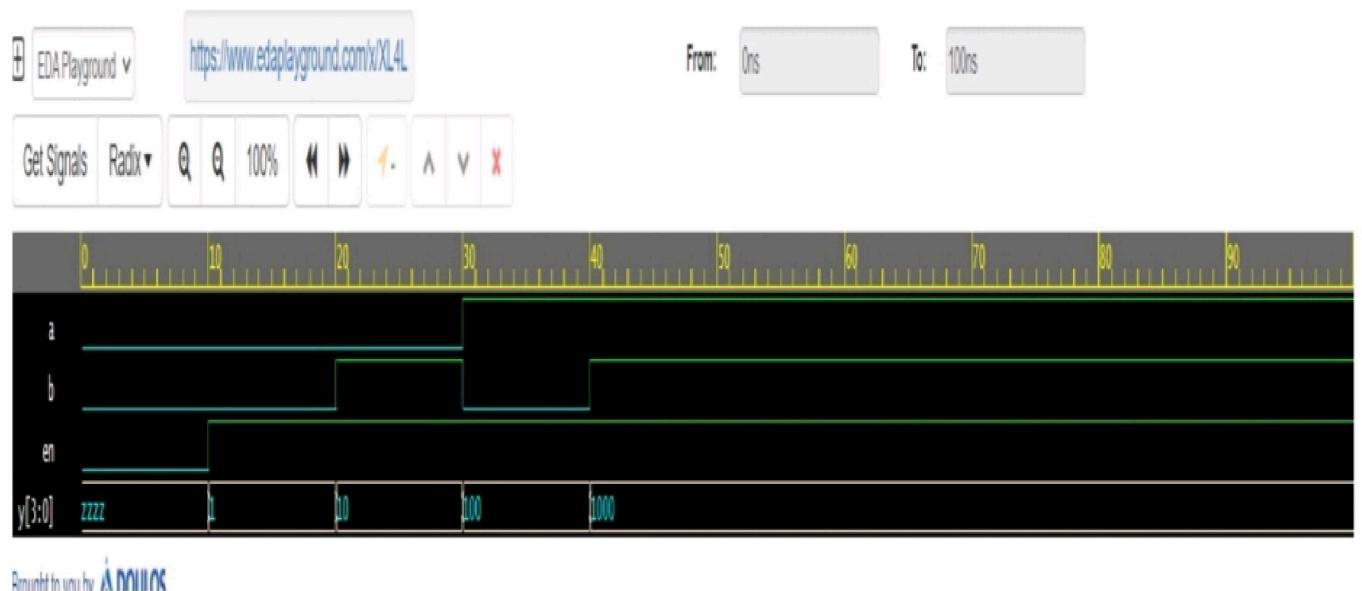
Test bench

module tb-dec2to4-sw;
reg a,b,en;
wire [3:0]y;
dec2to4-sw uut(a,b,en,y);
initial
begin
en=0;a=0;b=0;
#10 en=1;
#10 en=1;a=0;b=1;
#10 en=1;a=1;b=0;
#10 en=1;a=1;b=1;
end
initial
begin
$dumpfile("test.vcd");
$dumpvars(1);
#100 $finish;
end
endmodule

```

Result:- Design and simulation of 2 to 4 decoder is done and also verified the output waveform for the given sample inputs.

2 TO 4 DECODER



roll: 1602-17-735-039

Name: K-Sai charan

Design and simulation of 4x1 multiplexers in verilog

Aim :- To implement and verify the 4x1 multiplex using 2x1 multiplexers using switch level in verilog.

Apparatus required :- pc, eda playground online simulator.

Theory :- Multiplexer is a combinational circuit with '2' input lines & single output line. It has 'n' select lines. Based on the inputs given to select lines, the respective input of input lines appears at output.

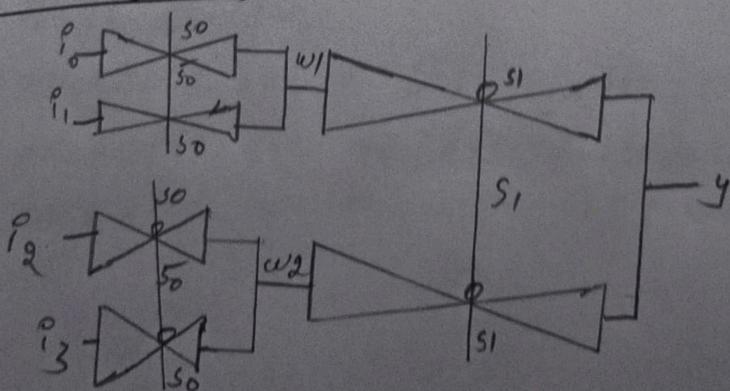
2x1 MUX has 2 input lines, 1 output and 1 selected lines.

4x1 MUX has 4 input, 1 output & 2 selected lines.

Boolean equation :-

$$y = \bar{s}_1 \bar{s}_0 A_0 + \bar{s}_1 s_0 A + s_1 \bar{s}_0 A_2 + s_1 s_0 A_3$$

Circuit diagram :-



Truth table :-

Inputs	Output
s ₁ s ₀	y
0 0	A ₀
0 1	A ₁
1 0	A ₂
1 1	A ₃

Verilog program :-

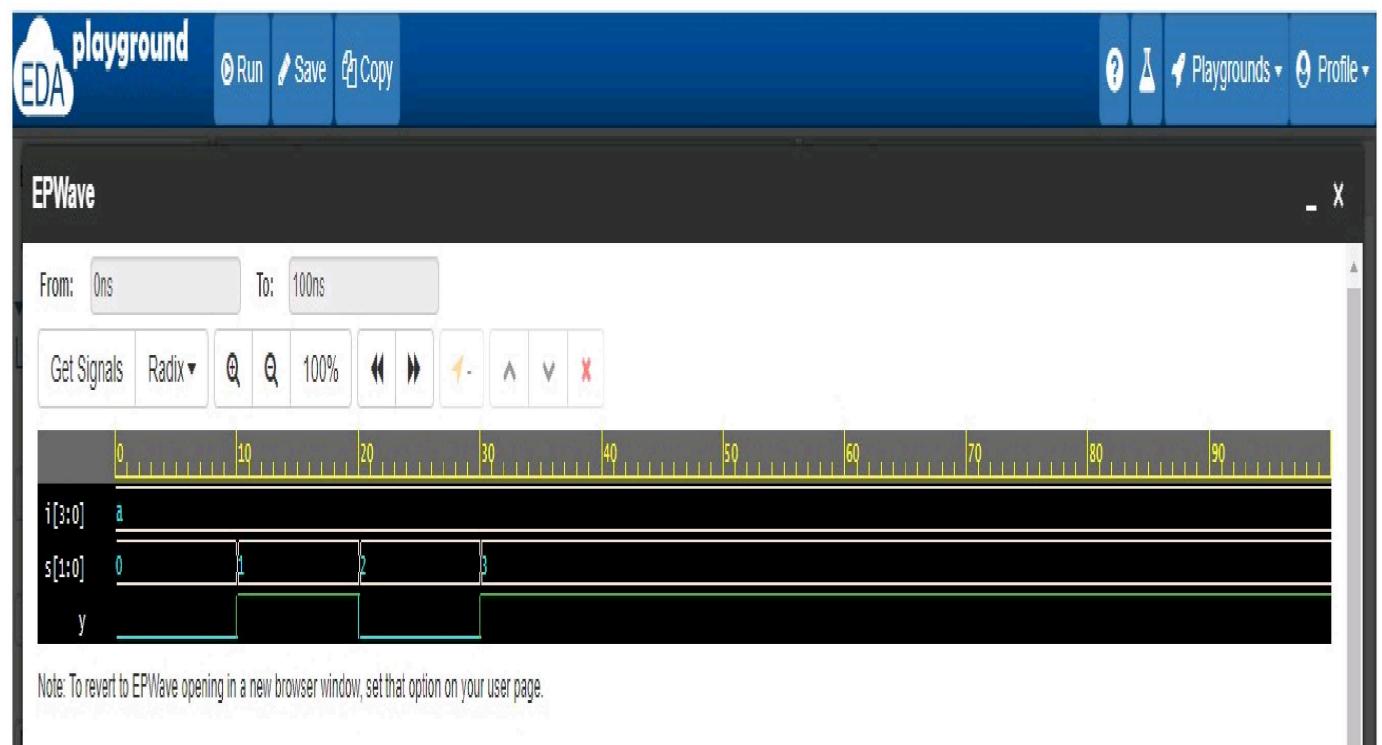
```
module mux2to1_sw(i,s,g);  
    input [1:0] i;  
    input s;  
    output y;  
    CMOS c1(y,i[0];(ns),s);  
    CMOS c2(y,i[1];s,(ns));  
end module  
module mux4to1_sw(i,s,y);  
    input [3:0] i;  
    input [1:0] s;  
    output y;  
    wire [1:0] w;  
    mux2to1_sw m1(i[1:0],s[0],w[0]);  
    mux2to1_sw m2(i[3:2],s[0],w[1]);  
    mux2to1_sw m3(w[1:0],s[1],y);  
endmodule
```

Test bench:

```
module tb_mux4to1_sw;
    reg[3:0] i;
    reg[1:0] s;
    wire y;
    mux4to1_sw uut(i,s,y);
initial
begin
    i[3:0]=4'b1010; # s[1],s[0]=2'b00;
# 10 i[3:0]=4'b1010; # s[1],s[0]=2'b01;
# 10 i[3:0]=4'b0110; # s[1],s[0]=2'b10;
# 10 i[3:0]=4'b1010; # s[1],s[0]=2'b11;
end
initial
begin
    $dumpfile("test.vcd");
    $dumpvars(1);
    # 100 $finish;
end
endmodule
```

Result: Design and simulation of 4x1 multiplexer.
using 2x1 multiplexer is done and also verified
the output waveform for the given sample inputs.

4X1 MUX



Roll no: 1609-12-735-039

Name: K. saicharan

Design and simulation of gates using switch level

dim:- To implement and verify gates using switch level modelling.

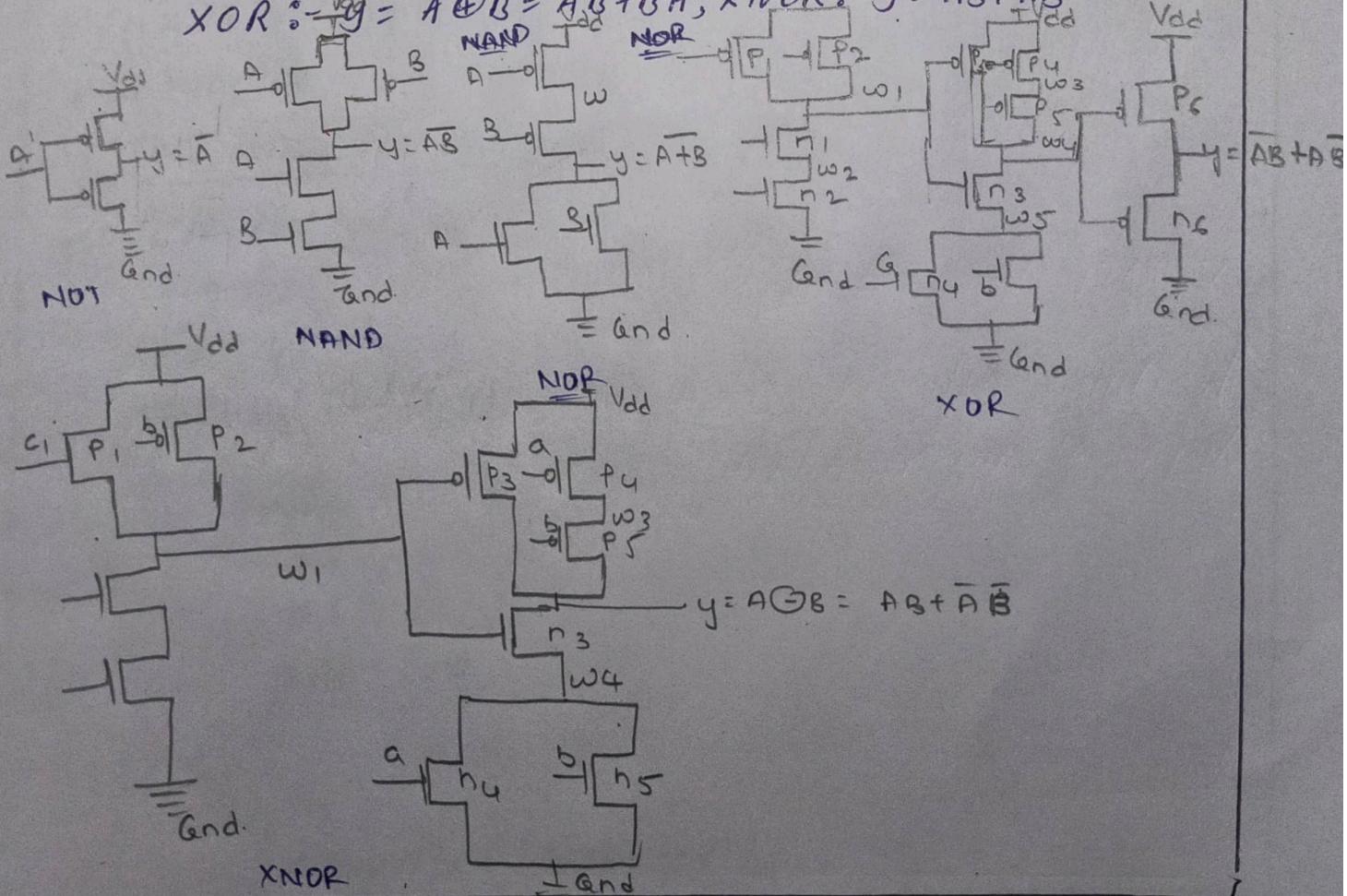
Apparatus:- pc, EDA playground online simulator

Theory:- Logic gates are the basic building blocks of any digital system. It is an electronic circuit having one or more than one input and only one output. The relationship b/w I/P and O/P follows a certain logic.

Boolean equations:-

$$\text{NOT} \Rightarrow y = \bar{A}, \text{NAND} \Rightarrow y = \bar{A}\bar{B}, \text{NOR} \Rightarrow y = \bar{A} + \bar{B}$$

$$\text{XOR} \Rightarrow y = A \oplus B = A\bar{B} + \bar{A}B; \text{XNOR} \Rightarrow y = AB + \bar{A}\bar{B}$$



Truth table :-

Inputs		Outputs				NOT	
a	b	NAND	NOR	XOR	XNOR	i/p	O/p
0	0	1	1	0	1	0	1
0	1	1	0	1	0	1	0
1	0	1	0	1	0	1	0
1	1	0	0	0	1		

VHDL program :- (NOT gate)

```
module nd_sw(i,y);
    input i;
    output y;
    supply1 vdd;
    supply0 gnd;
    pmos p(y,vdd,i);
    nmos n(y,gnd,i);
end module
```

Test bench for NOT gate

```
module tb-not_sw;
    reg i;
    wire y;
    not_sw uut(i,y);
    initial
        begin
            i=0;
            #20 i=1;
            #20 i=0;
    end
    initial
        begin
            $dumpfile("test.vcd");

```

XOR gate

```
module XOR_SW(a,b,y);  
    input a,b;  
    output y;  
    supply1 vdd;  
    supply0 gnd;  
    pmos p1(w1,vdd,a);  
    pmos p2(w1,vdd,b);  
    nmos n1(w1,w2,a);  
    nmos n2(w2,gnd,b);  
    pmos p3(w4,vdd,w1);  
    pmos p4(w3,vdd,a);  
    pmos p5(w5,w3,b);  
    nmos n3(w4,w5,w1);  
    nmos n4(w5,gnd,a);  
    nmos n5(w5,gnd,b);  
    pmos p6(y,vdd,w4);  
    nmos n6(y,gnd,w5);  
end module
```

XNOR gate

```
module XNOR_SW(a,b,y);  
    input a,b;  
    output y;  
    supply1 vdd;  
    supply0 gnd;  
    pmos p1(w1,vdd,a);  
    pmos p2(w1,vdd,b);
```

```
$dumpvar(1);  
#100 $finish;
```

end

endmodule

NAND gate verilog program

```
module nand_sw(a,b,y);  
    input a,b;  
    output y;  
    supply1 vdd;  
    supply0 gnd;  
    pmos p1(y,vdd,a);  
    pmos p2(y,vdd,b);  
    nmos n1(y,w,a);  
    nmos n2(w,gnd,b);  
endmodule
```

NOR gate

```
module nor_sw(a,b,y);  
    input a,b;  
    output y;  
    supply1 vdd;  
    supply0 gnd;  
    pmos p1(w,vdd,a);  
    pmos p2(y,w,b);  
    nmos n1(y,gnd,a);  
    nmos n2(y,gnd,b);  
endmodule
```

```

nmos n1(w1,w2,a);
nmos n2(w2,gnd,b);
pmos p3(y,vdd,w1);
pmos p4(w3,vdd,a);
pmos p5(y,w3,b);
nmos n3(y,w5,w1);
nmos n4(w5,gnd,a);
nmos n5(w5,gnd,b);
endmodule

```

Test bench for NAND,NOR,XNOR,XOR gates:-

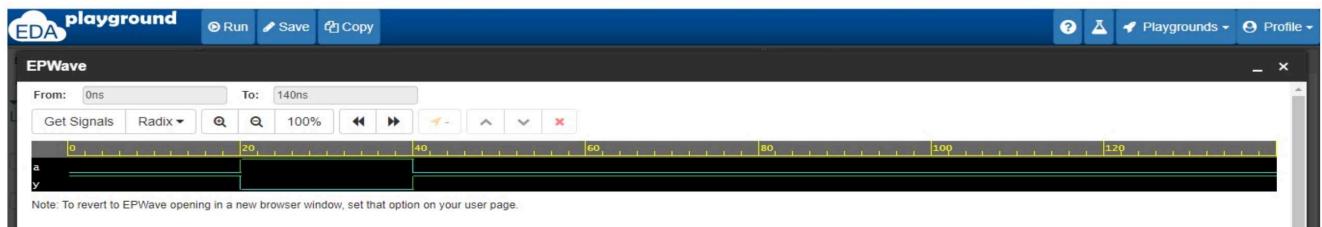
```

module tb_nand_sw;
reg a,b;
wire y;
nand_sw uut(a,b,y);
initial
begin
    a=0; b=0;
    # 10 a=0; b=1;
    # 10 a=1; b=0;
    # 10 a=1; b=1;
end
initial
begin
    $dumpfile ("test.vcd");
    $dumpvars(1);
    # 100 $finish;
end
endmodule

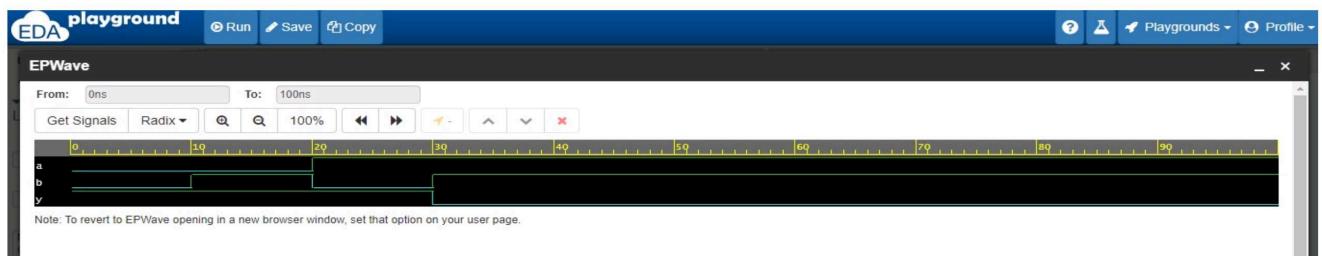
```

Result: Design and simulation of logic using switch level are done and verified the o/p waveforms for given inputs.

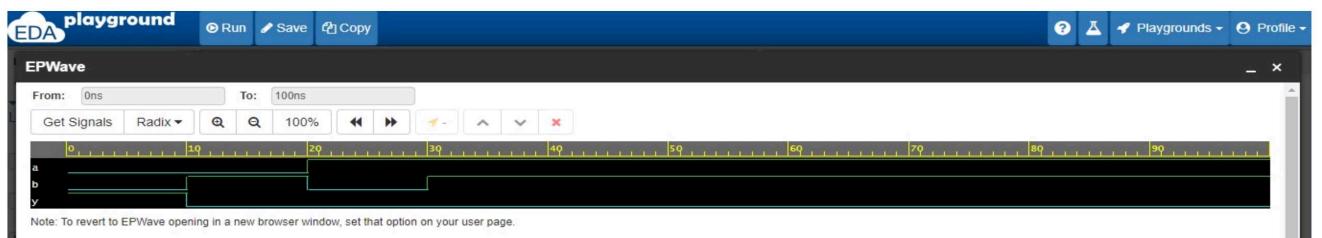
NOT GATE



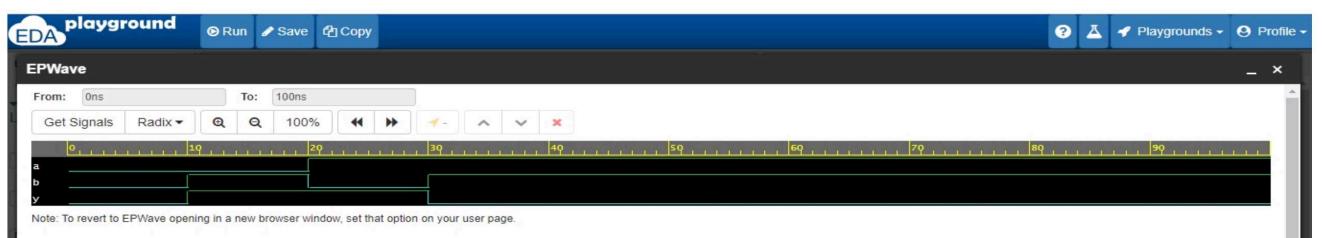
NAND GATE



NOR GATE



XOR GATE



XNOR GATE

