DDCO ASSIGNMENT 3

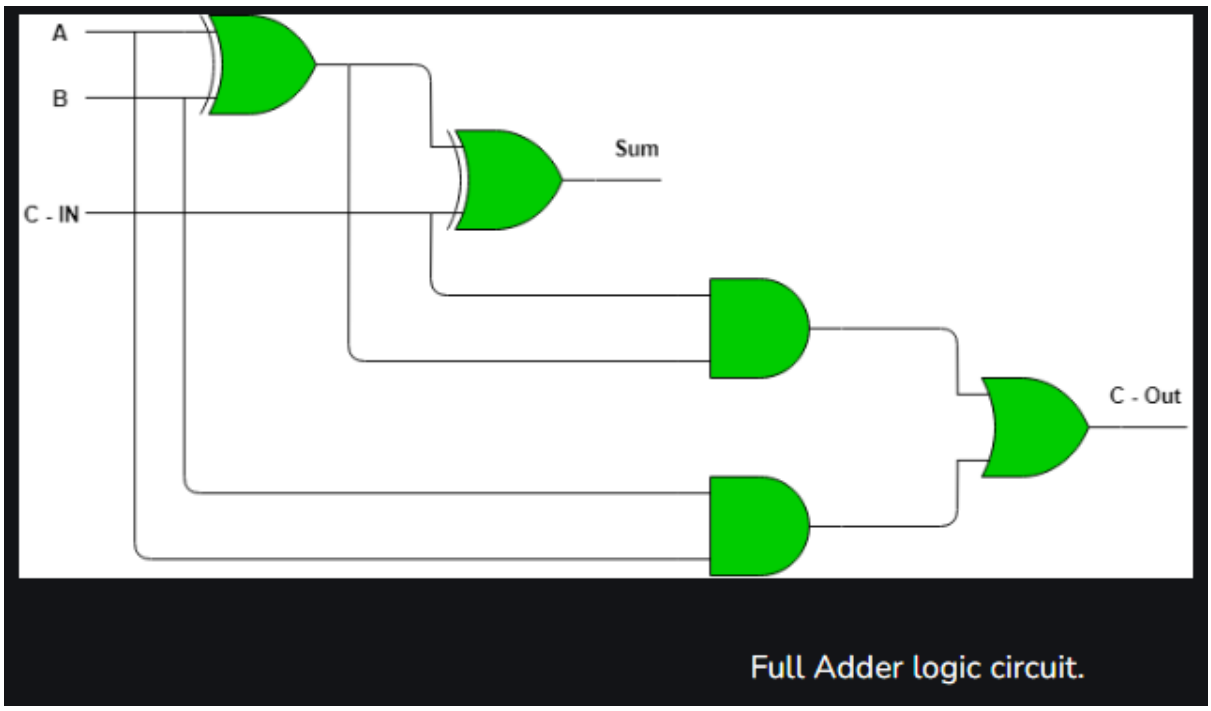
31-08-23

GAUTAM V HANCHINAL

PES1UG22CS215

SECTION D

1) FULL ADDER:



SOURCE : GEEKforGEEKS

Main Code:

fulladder.v

File Edit View

```
module fulladd(input wire a,b,c,output wire sum,cout);
wire [4:0] t;

and a1(t[1],a,b);
and a2(t[2],a,c);
and a3(t[3],b,c);
or o1(t[4],t[1],t[2]);
or o2(cout,t[3],t[4]);
xor x1(t[0],a,b);
xor x2(sum,t[0],c);
endmodule
```

Test Bench:

fulladder_tb.v

File Edit View

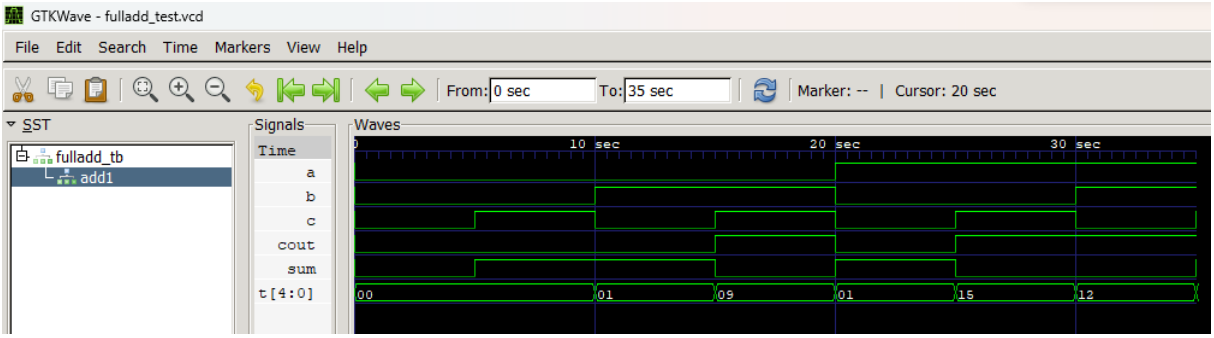
```
module fulladd_tb;
  reg a, b, c;
  wire s, y;
  fulladd add1 (.a(a), .b(b), .c(c), .sum(s), .cout(y));

  initial begin
    $dumpfile("fulladd_test.vcd");
    $dumpvars(0, fulladd_tb);
  end

  initial begin
    $monitor($time, "a=%b, b=%b, c=%b, sum=%b, carry=%b", a, b, c, s, y);
       a = 1'b0; b = 1'b0; c = 1'b0;
    #5 a = 1'b0; b = 1'b0; c = 1'b1;
    #5 a = 1'b0; b = 1'b1; c = 1'b0;
    #5 a = 1'b0; b = 1'b1; c = 1'b1;
    #5 a = 1'b1; b = 1'b0; c = 1'b0;
    #5 a = 1'b1; b = 1'b0; c = 1'b1;
    #5 a = 1'b1; b = 1'b1; c = 1'b0;
    #5 a = 1'b1; b = 1'b1; c = 1'b1;
    $finish;
  end
endmodule
```

VVP Output:

Command Prompt - gtkwave

```
C:\Users\aimem\OneDrive\Desktop\assignment ddco\assignment 3>iverilog -o test1 fulladder.v.txt fulladder_tb.v.txt

C:\Users\aimem\OneDrive\Desktop\assignment ddco\assignment 3>vvp test1
VCD info: dumpfile fulladd_test.vcd opened for output.
                 0a=0, b=0, c=0, sum=0, carry=0
                 5a=0, b=0, c=1, sum=1, carry=0
                10a=0, b=1, c=0, sum=1, carry=0
                15a=0, b=1, c=1, sum=0, carry=1
                20a=1, b=0, c=0, sum=1, carry=0
                25a=1, b=0, c=1, sum=0, carry=1
                30a=1, b=1, c=0, sum=0, carry=1
                35a=1, b=1, c=1, sum=1, carry=1

C:\Users\aimem\OneDrive\Desktop\assignment ddco\assignment 3>gtkwave fulladd_test.vcd

GTKWave Analyzer v3.3.48 (w)1999-2013 BSI

[0] start time.
[35] end time.
```

GTKWAVE output:



Truth Table:

Inputs			Outputs	
A	B	C – IN	Sum	C - Out
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

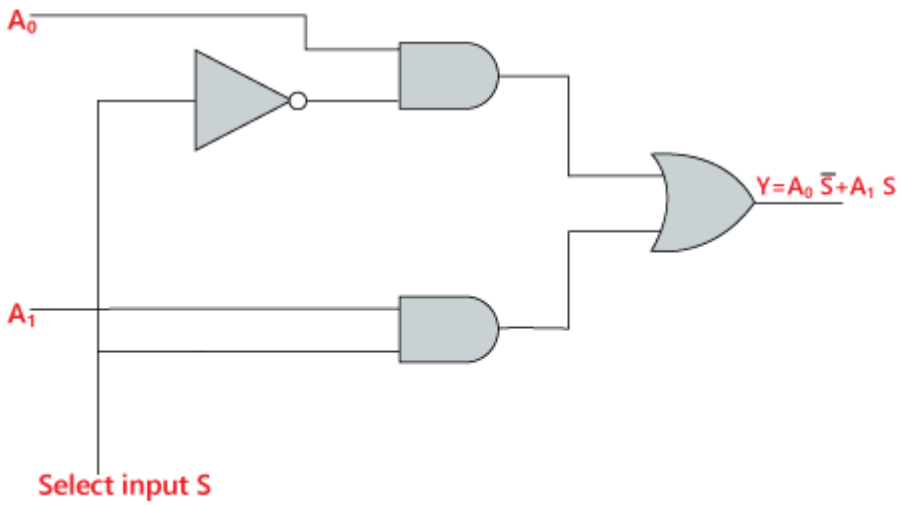
SOURCE : GEEKforGEEKS

2) 2:1 MUX

The logical expression of the term Y is as follows:

$Y=S_0'.A_0+S_0.A_1$

Logical circuit of the above expression is given below:



SOURCE: javatpoint.com/ multiplexer-digital-electronics

Main Code:

2isto1MUX.v | 2isto1MUX_tb.v

File Edit View

```
module mux2 (input wire a, b, x, output wire o);
assign o = (a & (~x))+(b & x);
endmodule
```

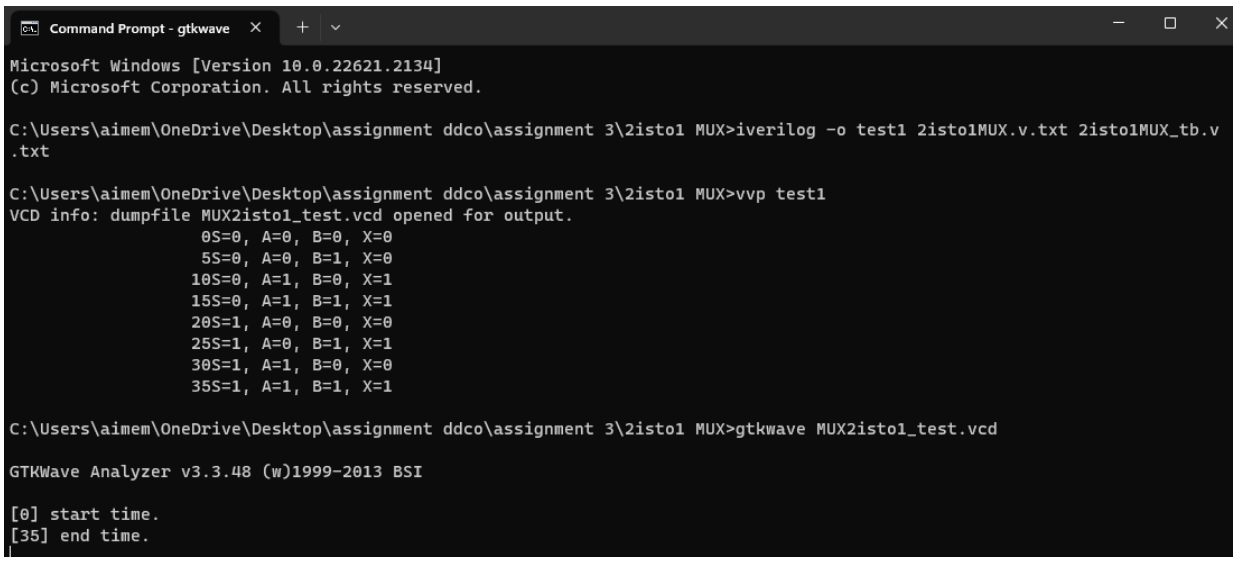
Test Bench:

2isto1MUX.v 2isto1MUX_tb.v

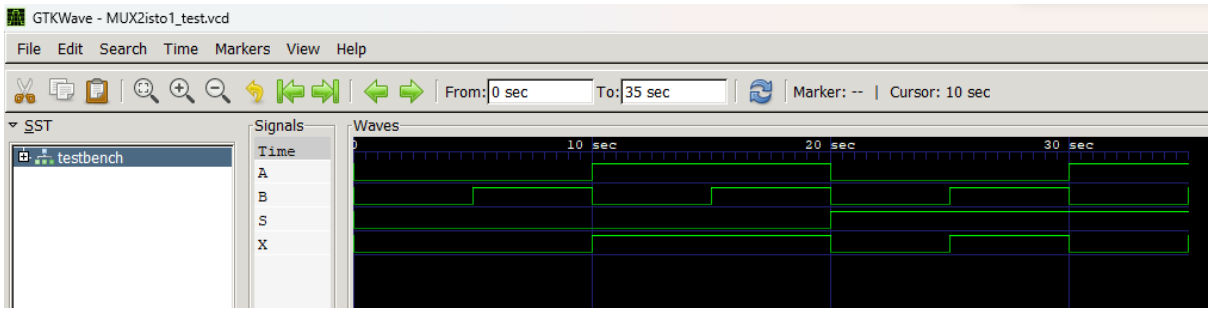
File Edit View

```
module testbench;

reg A,B,S;
wire X;

initial
begin
$dumpfile("MUX2isto1_test.vcd");
$dumpvars(0,testbench);
end

mux2 newMUX(.a(A), .b(B), .x(S), .o(X));

initial
begin
S = 1'b0;A = 1'b0;B = 1'b0;
#5 S = 1'b0;A = 1'b0;B = 1'b1;
#5 S = 1'b0;A = 1'b1;B = 1'b0;
#5 S = 1'b0;A = 1'b1;B = 1'b1;
#5 S = 1'b1;A = 1'b0;B = 1'b0;
#5 S = 1'b1;A = 1'b0;B = 1'b1;
#5 S = 1'b1;A = 1'b1;B = 1'b0;
#5 S=1'b1;A = 1'b1;B = 1'b1;
end

initial
begin
$monitor($time, "S=%b, A=%b, B=%b, X=%b", S,A,B,X);
   S = 1'b0;A = 1'b0;B = 1'b0;
#5 S = 1'b0;A = 1'b0;B = 1'b1;
#5 S = 1'b0;A = 1'b1;B = 1'b0;
#5 S = 1'b0;A = 1'b1;B = 1'b1;
#5 S = 1'b1;A = 1'b0;B = 1'b0;
#5 S = 1'b1;A = 1'b0;B = 1'b1;
#5 S = 1'b1;A = 1'b1;B = 1'b0;
#5 S = 1'b1;A = 1'b1;B = 1'b1;
end

endmodule
```

VVP output:



GTKWAVE output:



Truth Table:

INPUT(s)	*OUTPUT*
S_o	*X*
0	A
1	B