

MUX 4X1 + TEST BENCH

```
module MUX_4X1(output MUX, input S0,S1,D0,D1,D2,D3);
assign MUX = ((D0&~S1&~S0)|(D1&~S1&S0)|(D2&S1&~S0)|(D3&S1&S0));
endmodule

module TB_MUX_4X1;
reg d0,d1,d2,d3,s0,s1;
wire mux;

MUX_TB
M41(.MUX(mux),.D0(d0),.D1(d1),.D2(d2),.D3(d3),.S0(s0),.S1(s1));
initial begin
d0=1'b0;
d1=1'b0;
d2=1'b0;
d3=1'b0;
s0=1'b0;
s1=1'b0;
#128 $finish;
end
always #8 d0=~d0;
always #16 d1=~d1;
always #32 d2=~d2;
always #64 d3=~d3;
always #2 s0=~s0;
always #4 s1=~d1;

always @(mux)
$display("Time=%0t \tOutput:\t%b \tInput:\tS1:%b S0:%b \tD3:%b D2:%b
D1:%b D0:%b", $time,mux,s0,s1,d3,d2,d1,d0);
endmodule
```

DEMUX 1X4 + TEST BENCH

```
module DEMUX_1X4(output Y0,Y1,Y2,Y3, input A,S0,S1);
assign Y0 = A&~S1&~S0;
assign Y1 = A&~S1&S0;
assign Y2 = A&S1&~S0;
assign Y3 = A&S1&S0;
endmodule

module TB_DEMUX_1X4;
reg a,s0,s1;
wire y0,y1,y2,y3;

DEMUX_1X4 M14(.A(a),.Y0(y0),.Y1(y1),.Y2(y2),.Y3(y3),.S0(s0),.S1(s1));
initial begin
a=1'b0;
s0=1'b0;
s1=1'b0;
#45 $finish;
end
always #8 a = ~a;
always #2 s0=~s0;
always #4 s1=~s1;

always @(y0 or y1 or y2 or y3)
$display("Time:%0t\t Y3:%b\t Y2:%b\t Y1:%b\t Y0:%b\t S1:%b\t S0:%b\t
A:%b\t",$time,y3,y2,y1,y0,s1,s0,a);
endmodule
```

Full Subtractor + Test Bench

```
module full_subtractor(input a,b,c,output reg diff,cout);
always @(*)begin
diff =a^b^c;
cout=(~a&b)|(b&c)|(c&~a);
end
endmodule
```

```
module tbfullsub ;
reg A,B,C;
wire DIFF , COUT ;
full_subtractor fs(.a(A),.b(B),.diff(DIFF),.c(C),.cout(COUT));

initial begin
A=1'b0;
B=1'b0;
C=1'b0;
#16 $finish;
end

always #2 A=~A;
always #4 B=~B;
always #8 C=~C;

always @(DIFF or COUT)
$display("time =%0t \tINPUT VALUES:\t A=%b B=%b C=%b \t output value
DIFF=%b COUT =%b", $time,A,B,C,DIFF,COUT);

endmodule
```

Full Adder + Test Bench

```
module full_adder(input a,b,c,output reg sum,cout);
always @(*)begin
sum=a^b^c;
cout=(a&b)|(b&c)|(c&a);
end
endmodule
```

```
module tbfulladder ;
reg A,B,C;
wire SUM, COUT ;
full_adder fa(.a(A),.b(B),.sum(SUM),.c(C),.cout(COUT));

initial begin
A=1'b0;
B=1'b0;
C=1'b0;
#16 $finish;
end

always #2 A=~A;
always #4 B=~B;
always #8 C=~C;

always @(SUM or COUT)
$display("time =%0t \tINPUT VALUES:\t A=%b B=%b C=%b \t output value
SUM=%b COUT =%b", $time,A,B,C,SUM,COUT);

endmodule
```