module MUX4\_TO\_1 (out,i0,i1,i2,i3,s1,s0);

output out;

input i0, i1, i2, i3, s1, s0;

//Internal wire declaration

wire s0n, s1n, y0, y1, y2;

// Gate instantiations

not ( s1n, s1 );

not ( s0n, s0 );

and ( y0, i0, s1n, s0n );

and ( y1, i1, s1n, s0 );

and ( y2, i2, s1 , s1n );

and ( y3, i3, s1 , s0 );

or ( out, y0, y1, y2, y3 );

endmodule

module MUX4\_TO\_1\_tb;

// Clock and reset signals

reg clk;

reg reset;

// Design Inputs and outputs

reg Ti0,Ti1,Ti2,Ti3,Ts1,Ts0;

wire Y;

// DUT instantiation

MUX4\_TO\_1 dut(

.out(Y),

.i0(Ti0),

.i1(Ti1),

.i2(Ti2),

.i3(Ti3),

.s1(Ts1),

.s0(Ts0)

);

// generate the clock

initial

begin

clk = 1'b0;

forever #1 clk = ~clk;

end

// Generate the reset

initial begin

reset = 1'b1;

#10

reset = 1'b0;

end

//MUX4\_TO\_1 mux1(Y,Ti0,Ti1,Ti2,Ti3,Ts1,Ts0);

initial

begin

Ts1=0;Ts0=0;Ti0=1;Ti1=1;Ti2=1;Ti3=1;

#10 Ts1=0;Ts0=1;Ti0=1;Ti1=1;Ti2=1;Ti3=1;

#20 Ts1=1;Ts0=0;Ti0=1;Ti1=1;Ti2=1;Ti3=1;

#40 Ts1=1;Ts0=1;Ti0=1;Ti1=1;Ti2=1;Ti3=1;

end

initial

$monitor("Select=%b%b Out=%b time=%d",Ts1,Ts0,Y,$time);

endmodule : MUX4\_TO\_1\_tb