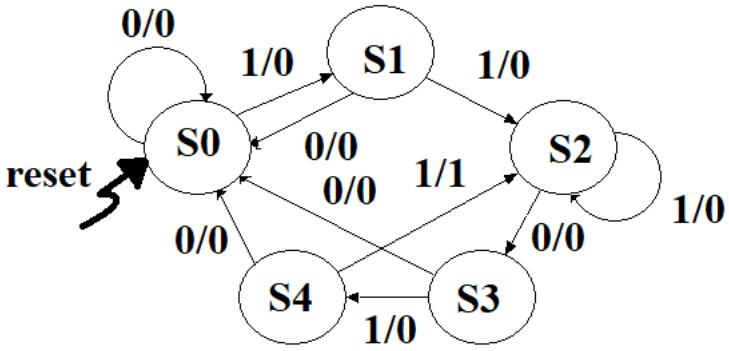
CSUS CPE166 Dummy2 Exercise Student Name: Shammah Thao

You need to upload the solution in multi-page PDF or WORD file format on the canvas course website.
You must write down your name on your solution sheets.

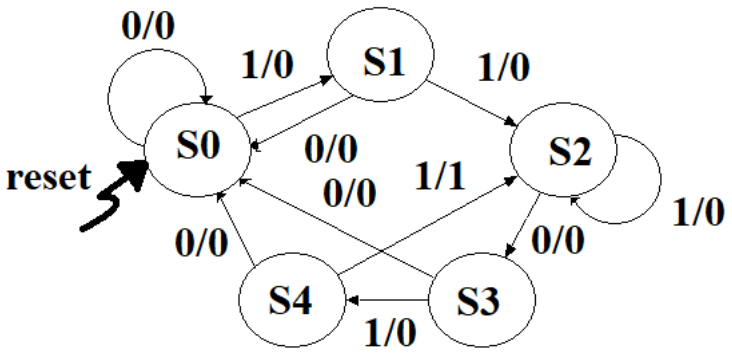
1. Complete the **Verilog** HDL design for the following finite state machine.




```
module   fms ( reset, clk, a, y );

input   reset, clk, a;

output  y;

parameter s0=0,s1=1,s2=2,s3=3,s4=4;
reg[2:0] cu,ns;

always @(posedge clk)

begin
if(rst)
cs <= s0;
else cs <= ns;
end

always@(a,cs)
begin
case(cs)
so: begin
        if(in)
        begin
        ns<= s1;
        y<=0;
        end
```

```
        else
        begin
        ns<=s0;
        y<=0;
        end
end

s1:begin
        if(in)
        begin
        ns<= s2;
        y<=0;
        end
else
        begin
        ns<=s0;
        y<=0;
        end
end

s2:begin
        if(in)
        begin
        ns<= s2;
        y<=0;
        end
else
        begin
        ns<=s3;
        y<=0;
        end
end

s3:begin
        if(in)
        begin
        ns<= s4;
        y<=0;
        end
else
        begin
        ns<=s0;
        y<=0;
        end
end

s4:begin
        if(in)
```

```
        begin
        ns<= s2;
        y<=1;
        end
else
        begin
        ns<=s0;
        y<=0;
        end
end

default :begin
        if(in)
        begin
        ns<= s0;
        y<=0;
        end
endcase
endmodule
```

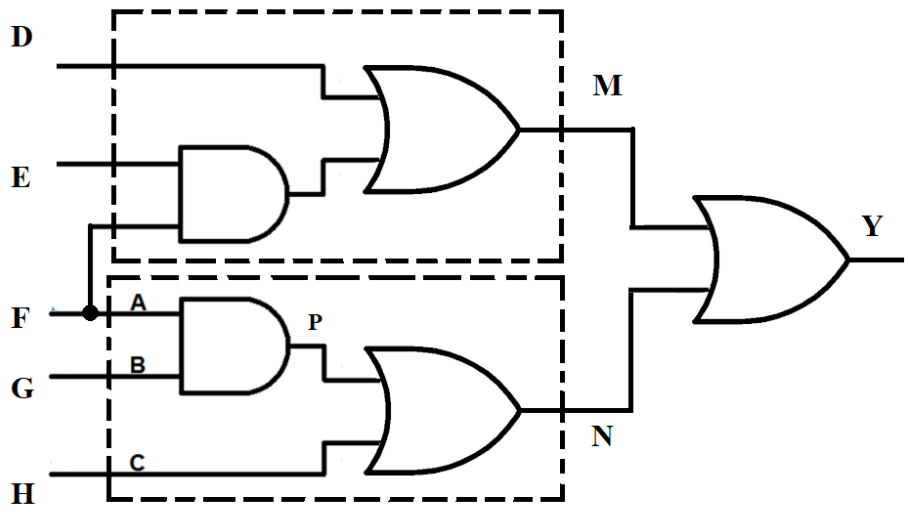


// Continue your Verilog design on this page.

…..

2. Fill out 6 blanks in the following Veirlog design with the given circuit schematic.

```
module ex2(D, E, F, G, H, Y);
input     D, E, F, G, H;
output   Y;
reg       M, N, Y;
always@(D or E or F or G or H)
begin
    M = subcir ( __F_, __E__, __D__);

    N = subcir ( _F__, __G__, ___H_);

    Y =  M  |   N;
end

function subcir;
input A, B, C;
reg    P;
begin
      P = A & B;
     subcir = P | C;
end
endfunction
endmodule
```



VHDL Exercises are for questions below.

Note: VHDL is not case sensitive.

3. Use when … else statement to design 4 to 1 multiplexer.

```
Library ieee;

Use     ____iee.std_logic_1164.all_____________________;

Entity  mux4to1  is
     Port  ( d0, d1, d2, d3:   in std_logic;

               S:          in _____std_logic_____________________ ;

               Y          out _______std_logic_____________________  );
End mux4to1;

Architecture beh1 of mux4to1 is
Begin
          Y <=     d0   when  __S= ‘0’_________________     else

                   d1   when  ___S= ‘1’_________________     else

                   d2  when  _________S= ‘1’__________     else

                   d3;

End beh1;
```

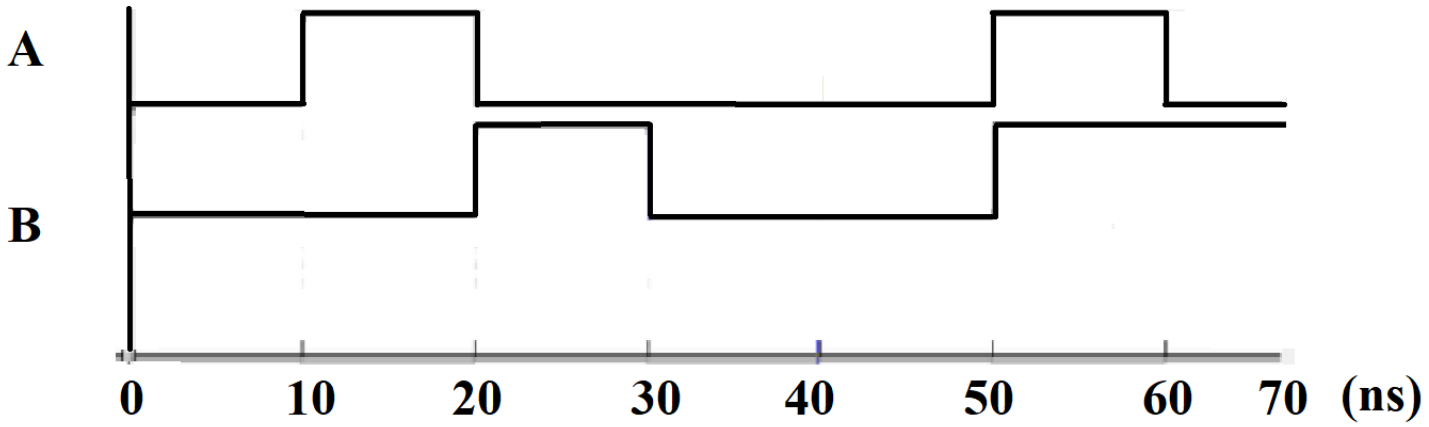
4. Use with … select statement to design 4 to 1 multiplexer.
(Entity part is same with that used in question 3).

```
Architecture beh2 of mux4to1 is
Begin
        With S select

          Y <=     d0   when  _____S=’0’_____________    ,

                   d1   when  __________S=’1’_________    ,

                   d2  when  ________S=’1’___________    ,

                   d3  when   others;

End beh2;
```

5. Use if … else statement to design 4 to 1 multiplexer.
(Entity part is same with that used in question 3).

```
Architecture beh3 of mux4to1 is
Begin
        Process( d0, d1, d2, d3, S)
        begin
            if  (  __S='0'_____________________   ) then

                Y <= d0;

            elsif (    _______S='1'_____________   ) then

                Y <= d1;

            elsif (   _____________S='1'_______     ) then

                Y <= d2;

            else

                 Y <= d3;

            end if;
        End process;

End beh3;
```

6. Complete the VHDL statements to describe waveforms A and B.



A <= ‘0’, ‘1’ after _10____ ns, ‘0’ after __20____ns, ‘1’ after ____50_____ ns, ‘0’ after _____60_____ ns;

Process
Begin
B <= ‘0’;

Wait for _____20_____ ns;

B <= ‘1’;

Wait for _____10_____ ns;

B <= ‘0’;

Wait for _____20_____ ns;

B <= ‘1’;

Wait for _____30_____ ns;

Wait;

End process;