CPE 322 EXAM I Name Solution
Spring Semester 2013

Work should be performed systematically and neatly with the final answer being underlined. This exam is closed book, closed notes, closed neighbor. Allowable items on desk include: exam, data manual, pencils, straight edge, and calculator. All other items must be removed from student's desk. Students have approximately 90 minutes (1 1/3 hours) to complete this exam. Best wishes!

1. [3 points] What is the primary difference between sequential logic and combinational logic?

   In combinational logic the output depends only on the current values of the inputs. In sequential logic the output depends on the current value of the inputs and the current state of the system (with the current state itself depending on the past values of the inputs). Thus sequential logic has memory combinational logic does not.

2. [12 points] In general, what are static hazards in a combinational network?

   Static hazards are incorrect transient (temporary) outputs (i.e. glitches) that occur in a static network after one or more inputs change value that are the results of multiple delay paths from one or more inputs to the same output.

   For the network shown below, find any/all static 0-hazards. For any 0-hazard found specify the conditions which will cause the hazard to appear at the output (i.e. specify the logic values of the variables which are constant and the variable which is changing).

transient output function $Z^t = [a'+c][a+c']+c'd$

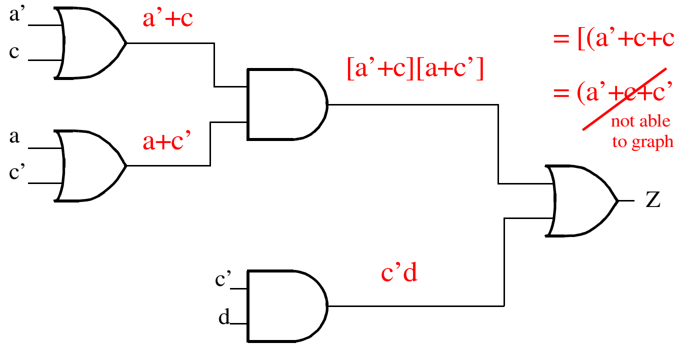
$= [a'+c+c'd][a+c' + c' d]$ distributive law No. 2 $X+YZ = (X+Y)(X+Z)$

$= [(a'+c+c')(a'+c+d)][(a+c'+c')(a+c'+d)]$ distributive law No. 2 $X+YZ = (X+Y)(X+Z)$

$= (a'+\cancel{c+c'}) (a'+c+d) (a+c') (a+\cancel{c'+d})$ distributive law No. 2 $X+YZ = (X+Y)(X+Z)$

not able to graph

simplification theorem $X (X+Y)=X$



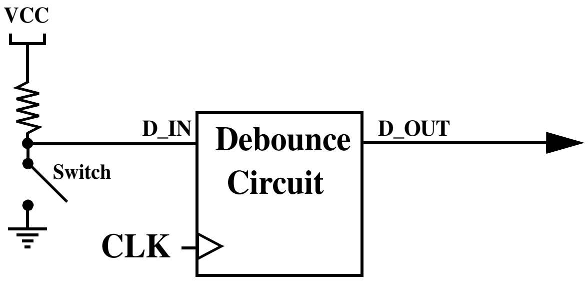
cd \ a	0	1
00		0
01		
11	0	
10	0	

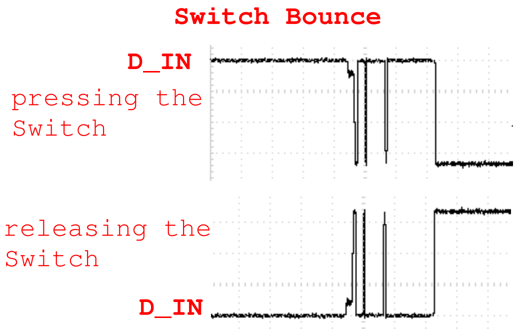
No Static 0-hazards present!

3. [3 points]
In VHDL how many *architecture* sections can be associated with each *entity*? Explain.

Each entity section must have at least one architecture section associated with it but may have many architectures associated with it. The reason for having more than one architecture is to allow one to model at different levels of abstraction the functionality of the module that the entity represents. Often the architecture that is selected by the CAD tool is the one that is last compiled by the user.

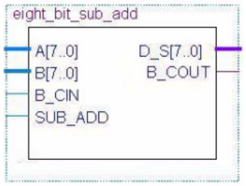
4. [6 points] What is the purpose of the debounce circuit shown below? When might such a circuit be necessary?





Mechanical switches such as this tend to make and break the connection many times each time it is pressed as shown in the figure above. Debounce circuits filter out these variations resulting in a single clean transition from low to high and high to low. If a debounce circuit is not present a device could be triggerred multiple unpredictable number of times for each key press.

5. [6 points] Write an *entity* section for the VHDL model that is represented by the block shown below:



Assume that the signals B_CIN, SUB_ADD, and B_COUT contain single bit values and signals A, B, and D_S each contain 8 bits. You may use VHDL's built-in data types or data types specified in the IEEE 1164 library.

```
entity eight_bit_sub_add is
port (A,B : in STD_LOGIC_VECTOR(7 downto 0);
      B_CIN,SUB_ADD : in STD_LOGIC;
      D_S : out STD_LOGIC_VECTOR (7 downto 0;
      B_COUT : out STD_LOGIC);
end eight_bit_sub_add;
```

Note: many answers possible -- mode ***out*** *could be* ***buffer*** *for example*

6. [6 points] Using VHDL model a Toggle flip-flop with an input *T*, an output *Q*, and an asynchronous reset, *R*, that operates on the rising edge of the clock, *CLK*.

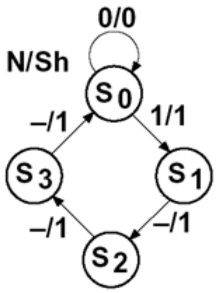
note: allowable to change the mode of Q to buffer if you did not use internal variable or signal.

```
entity TFF is
   port (R, T, CLK: in bit;
            Q: out bit);
end TFF

architecture TFF_ARCH of TFF is
begin
```

note: both CLK and R need to be on sensitivity list

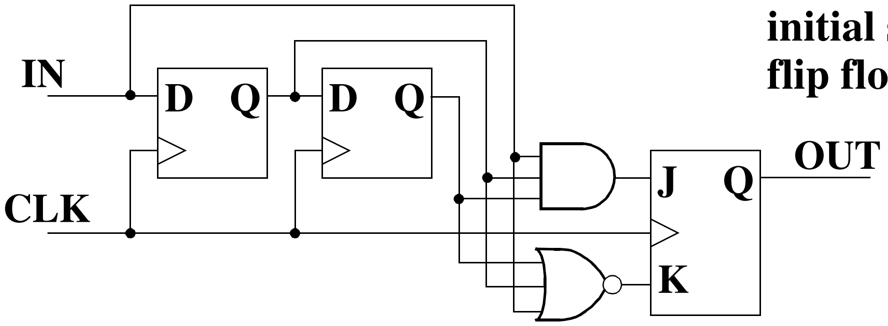
```
   process(CLK,R)
      variable QBUF : bit := ‘0’;
   begin
      if (R = ‘1’) then
         QBUF := 0;
         Q <= QBUF;
      else
         if (CLK=’1’ and CLK’event) then
            QBUF := not QBUF;
            Q <= QBUF;
         end if;
      end if;
   end process;

end TFF_ARCH;
```

7. [10 points] Implement a two-process VHDL behavioral model for the state graph shown below (complete the template):



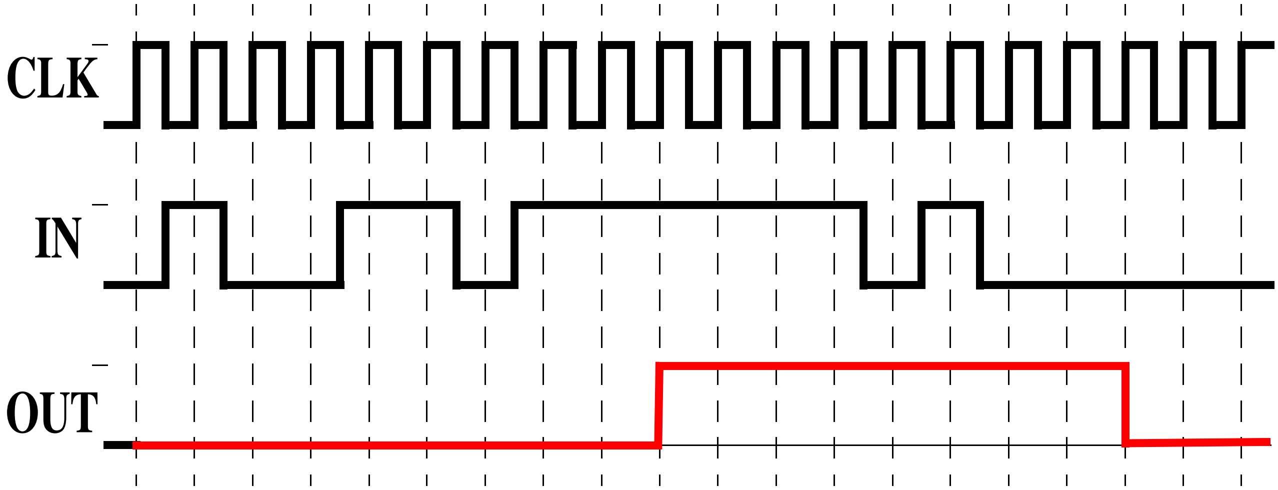
```
entity FSM is
   port (CLK, N : in bit;
              Sh : out bit);
end FSM

architecture BEHAVIORAL of FSM is
   signal STATE,NEXT_STATE : integer range 0 to 3 := 0;

begin                                  Note: a function of both the state
   COMB_LOGIC:process(STATE,N)               and current input N
                                             Should respond to changes in both
   begin                                     or either.
      if (STATE = 0) then
         if (N=0) then
            NEXT_STATE <= 0;
            Sh <= ‘0’;
         else
            NEXT_STATE <= 1;
            Sh <= ‘1’;
         end if;
      else
         -- in cases where next state is not 0 then NEXT STATE will be one
         -- more than the curent state mod 3 regardless of the input N
         if (STATE = 3) then
            NEXT_STATE <= 0);
         else
            NEXT_STATE <= NEXT_STATE + 1;
         end if;
         Sh <= ‘1’; -- in all cases where current state is not 0
                    -- then output is set to ‘1’
      end if;
   end process COMB_LOGIC;


   REG_PART:process(CLK)
   begin
      if (CLK=’1’) then --assuming rising edge clock (falling edge is ok too)
         STATE <= NEXT_STATE;
      end if;
   end process REG_PART;


end TFF_ARCH;
```

8. [10 points total] For the sequential logic network shown below, carefully complete the timing waveform for the **OUT** signal assuming the initial state of all flip flops in the circuit are a Logic 0 and the **IN** signal is driven in the manner that is specified. Neglect gate and flip-flop propagation delay. Also assume that all setup and hold times have been met by the input waveform.



initial states for all flip flops are Logic 0

[8 points]



What function could a circuit such as this be used for? Be specific.[2 points]

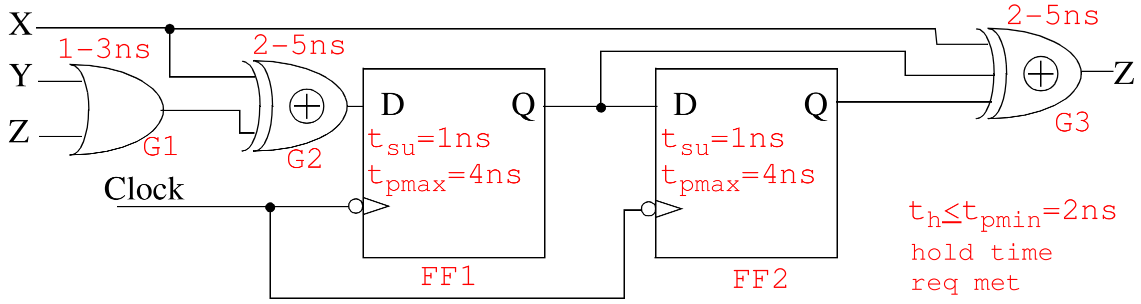
If the clock frequency was chosen correctly this could form the basis of a debounce circuit.

9. [14 points] In the following VHDL model fragment (which corresponds to a portion of the architecture section of a VHDL file) A, B, C, and D are all integers that have a value of 0 at time = 10ns. If E changes from ‘0’ to ‘1’ at time 20 ns, specify the times (s) at which each signal will change and the value to which it will change. List these changes in chronological order. List any delta delays as a separate entry.

```
P1: process
    begin
       wait on E;
       B <= A + 2;
       A <= transport 2 after 5 ns;
       wait for 0 ns;
        A <= transport A + 5 after 5 ns;
       B <= B + 1;
     end process P1;

P2: process(D)
    begin
       C <= B after 5 ns;
    end process P2;

D <= B after 10 ns;
```

Time	A	B	C	D	E
10 ns	0	0	0	0	0
20 ns	0	0	0	0	1
20 ns Δ	0	2	0	0	1
20 ns + 2Δ	0	3	0	0	1
25 ns	5	3	0	0	1
30 ns	5	3	0	3	1
35 ns	5	3	3	3	1

10. [10 points] A sequential network is implemented using an OR gate, two XOR gates and two D flip-flops as shown below. Assume that the inputs *X*, *Y*, and Z always change at the same time as the rising edge of the 50% duty cycle clock (clock is high 1/2 cycle then low 1/2 cycle). Also assume the following delay parameters: XOR gate delays range 2 to 5 ns, OR Gate delay ranges from 1 to 3 ns, flip-flop propagation delays range from 2 to 4 ns, setup time requirement are 1 ns, hold time requirements are 2 ns.



$t_h \le t_{pmin}$=2ns
hold time
req met

a) What general type of sequential network is this? Explain why you came to this conclusion [2 points].

Mealy -- because the output Z depends on both the current state (i.e. the output of the two FLIP-FLOPs and at least one Input (input X)

b) Determine the maximum clock rate for proper synchronous operation. Show all steps [8 points].

Case 1:

Considering the impact of the register to register path first

$t_{ck} \geq t_{pmax} + t_{cmax} + t_{su}$ where

where, t_{pmax} = 4 ns, t_{su} = 1 ns -- from flip-flop specifications

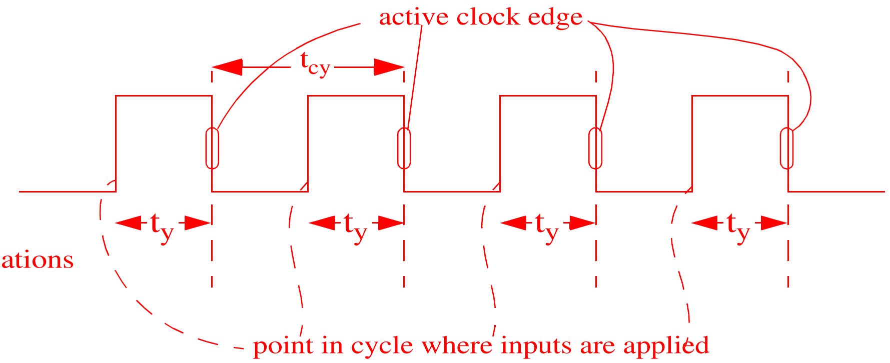
and

t_{cmax} = 0 ns -- Because only a wire is present in feedback path between flip-flop stages and in this problem the wiring delay is assumed to be negligible

$t_{ck} \geq 4ns + 0ns + 1ns = 5ns$

Case 2:

Considering the impact of the inputs X,Y, and Z being supplied at a fixed point relative to the active edge of the clock



$$t_y \geq t_{cxmax} + t_{su}$$

where,

$t_{su} = 1$ ns

from flip-flop specifications

and

$$t_{cxmax} = \overset{\substack{\text{OR gate delay}\\ \text{G1}}}{3ns} + \overset{\substack{\text{XOR gate delay}\\ \text{G2}}}{5ns} = 8ns$$ Max delay from input to registers

Note: XOR gate, G3, delay does not impact maximum clock frequency

Relationship between t_y and t_{ck}

$t_y = 1/2\ t_{ck}$ Because of 50% duty cycle and inputs being applied on non-active edge.

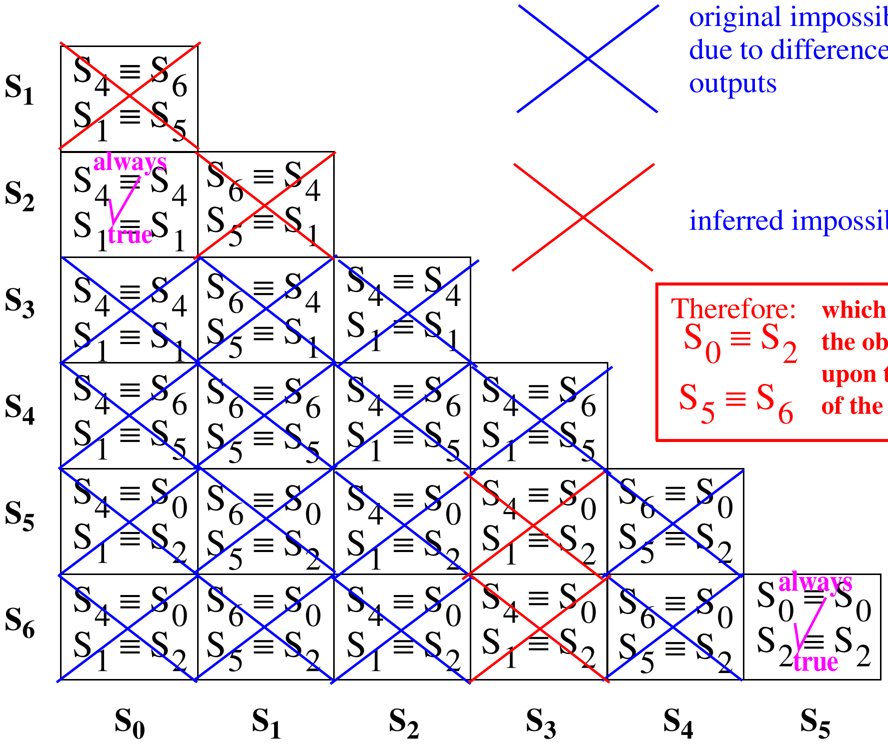
Since $t_y \geq t_{cxmax} + t_{su} \xrightarrow{\text{then}} 1/2\ t_{ck} \geq t_{cxmax} + t_{su}$ and

$t_{ck} \geq 2(t_{cxmax} + t_{su)} = 2(8ns + 1ns) = \boxed{18ns}$ which is a larger min period than in Case 1 so this is the limiting case

Max Clock Frequency = 1/min(t_{ck}) = 1/18ns = 55.56 Mhz

11. [10 points] Reduce the following state table to a minimum number of states.

Present State	Next State		Output	
	X=0	X=1	X=0	X=1
S_0	S_4	S_1	1	0
S_1	S_6	S_5	1	0
S_2	S_4	S_1	1	0
S_3	S_4	S_1	0	1
S_4	S_6	S_5	1	1
S_5	S_0	S_2	0	1
S_6	S_0	S_2	0	1



	S_0	S_1	S_2	S_3	S_4	S_5
S_1	$S_4 \equiv S_6$ $S_1 \equiv S_5$					
S_2	always $S_4 \equiv S_4$ $S_1 \equiv S_1$ true	$S_6 \equiv S_4$ $S_5 \equiv S_1$				
S_3	$S_4 \equiv S_4$ $S_1 \equiv S_1$	$S_6 \equiv S_4$ $S_5 \equiv S_1$	$S_4 \equiv S_4$ $S_1 \equiv S_1$			
S_4	$S_4 \equiv S_6$ $S_1 \equiv S_5$	$S_6 \equiv S_6$ $S_5 \equiv S_5$	$S_4 \equiv S_6$ $S_1 \equiv S_5$	$S_4 \equiv S_6$ $S_1 \equiv S_5$		
S_5	$S_4 \equiv S_0$ $S_1 \equiv S_2$	$S_6 \equiv S_0$ $S_5 \equiv S_2$	$S_4 \equiv S_0$ $S_1 \equiv S_2$	$S_4 \equiv S_0$ $S_1 \equiv S_2$	$S_6 \equiv S_0$ $S_5 \equiv S_2$	
S_6	$S_4 \equiv S_0$ $S_1 \equiv S_2$	$S_6 \equiv S_0$ $S_5 \equiv S_2$	$S_4 \equiv S_0$ $S_1 \equiv S_2$	$S_4 \equiv S_0$ $S_1 \equiv S_2$	$S_6 \equiv S_0$ $S_5 \equiv S_2$	always $S_0 \equiv S_0$ $S_2 \equiv S_2$ true

original impossibilities due to differences in outputs

always ✓ true

inferred impossibilities

Therefore: $S_0 \equiv S_2$ $S_5 \equiv S_6$

which turns out to be only the obvious states based upon the simple application of the equivalent state theorem

Reduced State Table

Present State	Next State		Output	
	X=0	X=1	X=0	X=1
S_0	S_4	S_1	1	0
S_1	S_5	S_5	1	0
S_3	S_4	S_1	0	1
S_4	S_5	S_5	1	1
S_5	S_0	S_0	0	1

12. [10 Points] A synchronous sequential circuit has one input and one output. If the input sequence is 1001, an output of 1 will occur. This 1 should occur coincident with the last input of the 1001 sequence. Otherwise the output is 0. For example,

```
input sequence:   X = 1 1 1 0 0 1 1 0 1 0 0 1 0 0 1 1 0 ...
output sequence:  Z = 0 0 0 0 0 1 0 0 0 0 0 1 0 0 1 0 0 ...
```

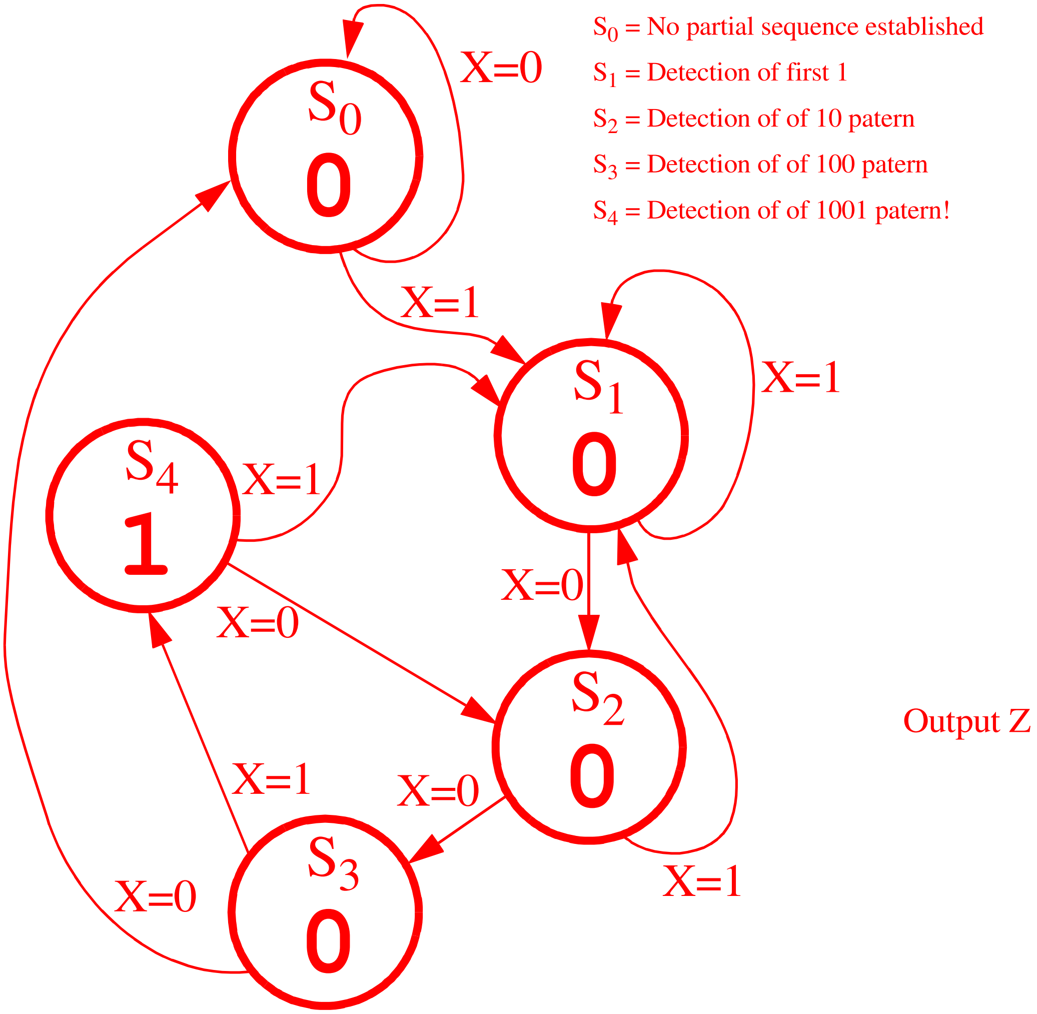
Derive a **Moore** state graph with a minimum number of 5 states.

Moore State Graph

-- output is only a function of the current state not the current inputs,

-- output becomes valid after the transition to the next state,

-- in the diagram output is shown inside the node of the state that it takes affect.



S_0 = No partial sequence established

S_1 = Detection of first 1

S_2 = Detection of of 10 patern

S_3 = Detection of of 100 patern

S_4 = Detection of of 1001 patern!