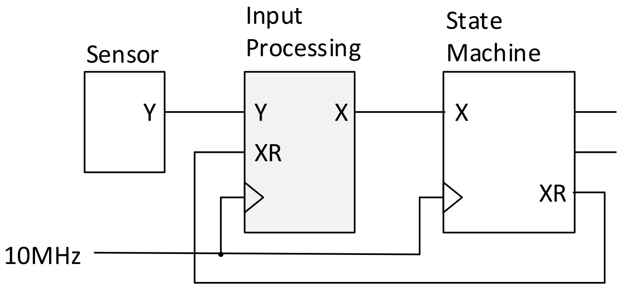
# Digital Systems Problem Set #2

Revision: 2/3/19

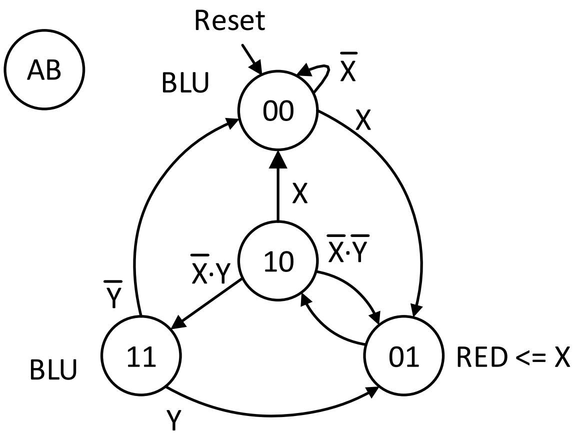
RealDigital

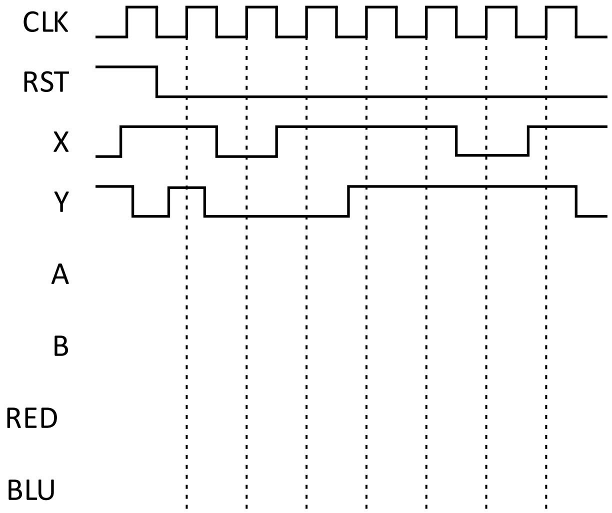
**1.** (6 points) A signal "Y" arising from a sensor can be asserted for an amount of time between 1us and 1ms. Design a state diagram for an input processing circuit that asserts an output signal "X" on the next rising edge of clock after Y has been received. X must remain asserted until after the input signal Y is no longer asserted, and until the feedback signal "XR" has been received to indicate the signal X has been processed by the state machine.



**2.** (10 points) After creating the previous state machine, you learned the sensor signal Y can bounce for up to 10us. Sketch a new state diagram for a circuit that can debounce the signal in addition to meeting the previous requirements.

**3.** (12 points) In the timing diagram below, show the time courses of the flip-flops (labeled A and B) and output signals defined by the state diagram. Then complete the following sentences.





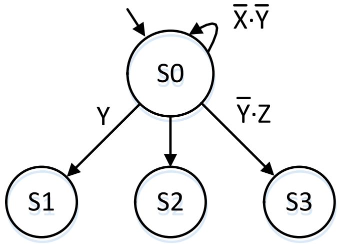
Can RED suffer from a negative output race glitch? [ Yes / No ]
If you circled yes, on what transition? Leaving state ____ and going to state ____.

Can RED suffer from a positive output race glitch? [ Yes / No ]
If you circled yes, on what transition? Leaving state ____ and going to state ____.

Can BLUE suffer from a negative output race glitch? [ Yes / No ]
If you circled yes, on what transition? Leaving state ____ and going to state ____.

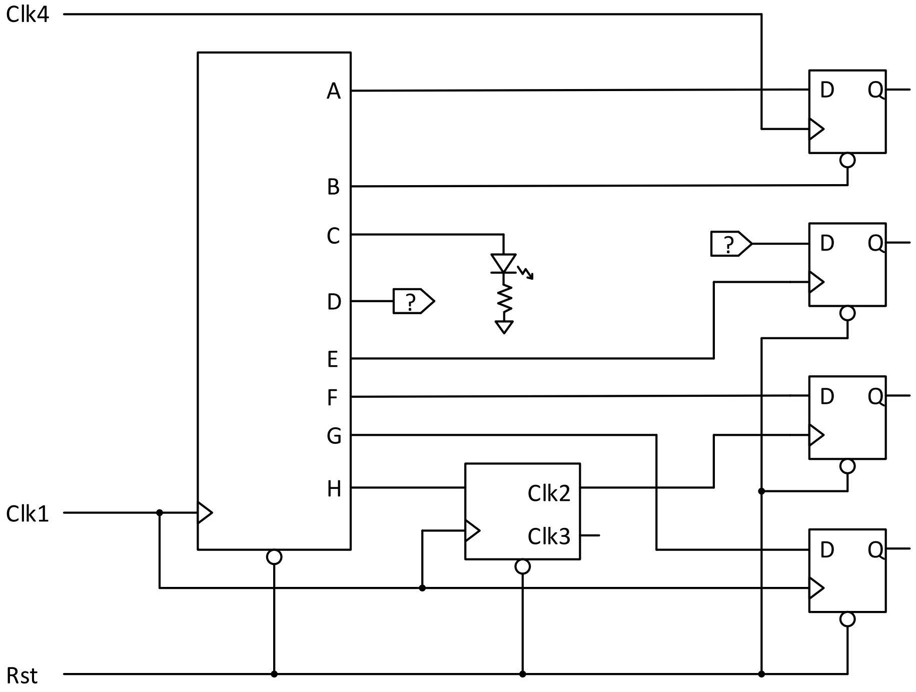
Can BLUE suffer from a positive output race glitch? [ Yes / No ]
If you circled yes, on what transition? Leaving state ____ and going to state ____.

**4.** (4 points) No branch condition is shown for the S2 branch. If some combination of branch conditions exists for which no next state is specified, then make that combination the S2 branch condition. Then, modify the hold condition so that only one next state is specified for all combinations of branch conditions. Enter the branch conditions below.



To S2:___________ Holding condition:___________

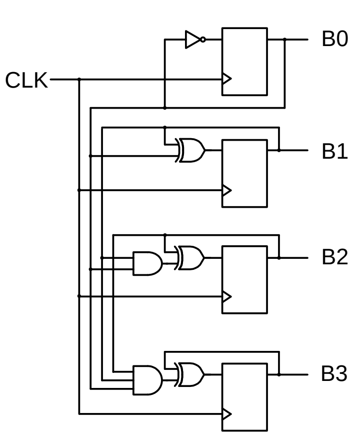
**5.** (7 points) The figure shows a state machine with outputs going to various destinations. Circle the letter of any output that should have logic noise (glitches) removed.



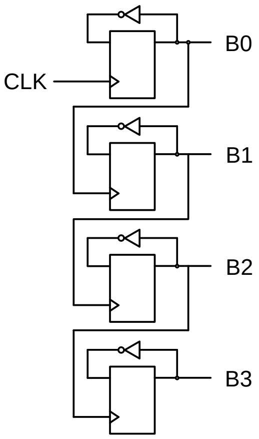
**6.** (10 points) Sketch a block diagram, state diagram, and circuit for the Verilog code shown below. Your circuit should show a D flip-flop together with any other needed logic circuits.

```
module FSM (
   input clk, rst, J, K,
   output Q);

reg Q;

always @ (posedge(clk))
begin
    if (rst) Q <= 0;
    else Q = (J&~Q) + (~K&Q);
end

endmodule
```

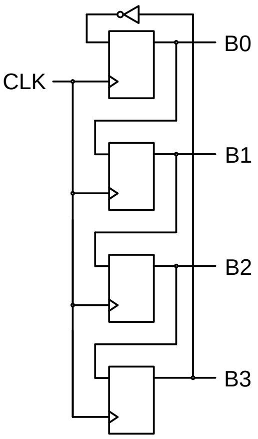
**7.** (10 points) The figures below show three different types of 4-bit counters. Circle the letters to indicate which counters (if any) the statements describe - you may circle 0, 1, 2, or all 3 letters for any row.



A



B



C

A B C Creates $2^N$ binary numbers from N flip-flops

A B C Suffers from output-bit skew problems in higher-order bits

A B C Is not limited in operating frequency by next-state logic delays

A B C Is commonly called an Asynchronous Counter

A B C Is commonly called a Ring or Johnson Counter

A B C Is commonly called a Binary Counter

A B C Creates (2 x N) binary numbers from N flip-flops

A B C Outputs (bit signals B0 – B3) can have glitches

A B C Outputs (bit signals B0 – B3) can be used as clocks for other circuit blocks

A B C Generates numbers in a natural counting sequence (0,1, 2, 3, etc)

A B C Can be used as a memory register in a state machine

A B C Can be limited in operating frequency by next-state logic delays