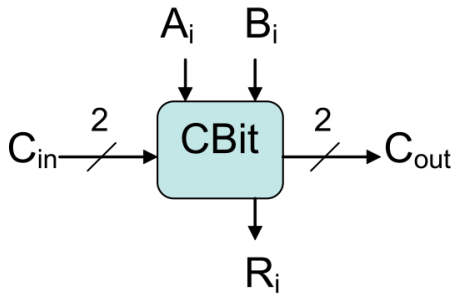
Sequential Logic Timing of FSM

4/4 points (ungraded)

MaxOut is a Cambridge startup whose products are binary comparators which determine the largest of several unsigned binary integers. A building block common to all MaxOut products is the combinational CBit module depicted below.



Each CBit module takes corresponding bits of two unsigned binary numbers, A and B, along with two C_{in} bits from higher-order CBit modules. Its output bit, R, is the appropriate bit of the larger among A and B, as determined from these inputs; it passes two C_{out} bits to lower-order CBit modules.

The propagation delay of each CBit module is 10ns. The two C_{out} bits indicate, respectively, if A>B or B>A in the bits considered thus far.

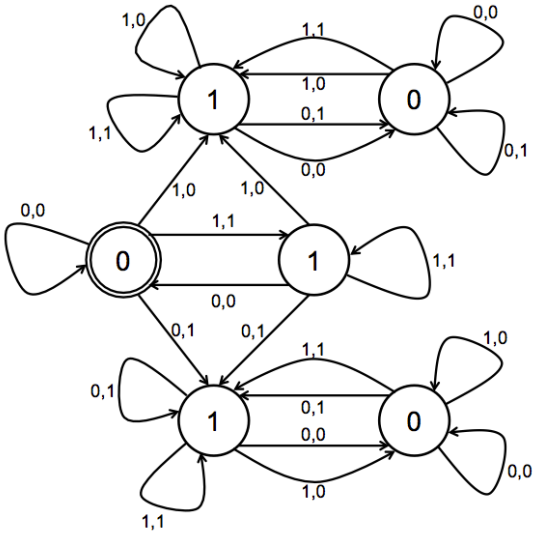
MaxOut's latest product is the MAXFSM. The MAXFSM is to be a clocked finite state machine that takes two N-bit binary numbers, $A_{N-1:0}$ and $B_{N-1:0}$ in bit-serial form, most significant bit first, and outputs the larger of these numbers as $R_{N-1:0}$ also in bit-serial form, but delayed by one clock cycle. The MAXFSM is a Moore machine; recall that this means its output is strictly a function of its current state (like those FSMs shown in lecture).

Before each active clock edge, the i^{th} bit of the A and B inputs are applied; during the **next** clock cycle, the i^{th} bit of the larger of the two input numbers appears at the R output. Note that the serial output bits are delayed with respect to the input bits by one clock cycle in order to allow each i^{th} output bit to be influenced by the i^{th} input bits.

(A) What is the minimum number of states necessary to implement a Moore machine obeying the above specifications? 6 ✔ Answer: 6

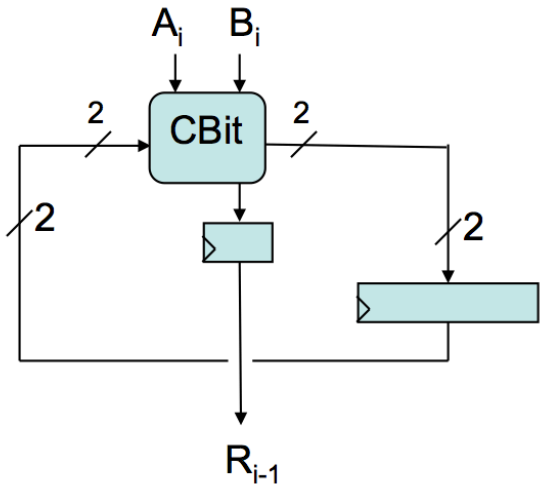
Explanation

The state machine below implements the specification in 6 states.



The middle two states are the output when A and B have been the same so far.
The upper two states are when A is greater than B, so the output follows A. The bottom two states are when B is greater than A, so the output follows B.

Ignoring your answer, MaxOut decides to build the MAXFSM using a CBit module and several flipflops, as shown in the diagram below:



The registers have the following specifications:

t_{pd}	5ns
t_{cd}	2ns
t_s	4ns
t_h	2ns

Recall that the CBit has a 10ns propagation delay; assume that its contamination delay is 0ns.

(B) What is the shortest clock period, in ns, for which this circuit will operate reliably?

Clock period (ns) $\geq$ 19 ✔ Answer: 19

Explanation

The clock period needs to be long enough for the next state to be stored in the register. There are three parts to that duration: the propagation delay of the register (5ns), the propagation delay of the CBit (10ns), and the setup time of the register (4ns). So the minimum clock period is $\mathbf{5 + 10 + 4 = 19}$.

(C) What setup and hold time requirements should be specified for this FSM?

FSM Setup Time (ns): 14 ✔ Answer: 14

FSM Hold Time (ns): 2 ✔ Answer: 2

Explanation
The setup time for the FSM is how long the input needs to be stable and valid before the rising clock edge. An input needs to propagate through the CBit and arrive at the register with enough time for the register to setup. So the setup time of the FSM is the propagation delay of the CBit (10ns) + the setup time of the register (4ns), which equals 14ns.

The hold time for the FSM is how long after a rising clock edge the input needs to be stable and valid in order for the register's hold-time specification to be met. The contamination delay of the CBit is zero, so the input to the register becomes invalid as soon as the input to the FSM does. In order to satisfy the register's hold time, the hold time for the FSM must be the same as the hold time of the register, which is 2ns.

Submit

ℹ Answers are displayed within the problem

FSM

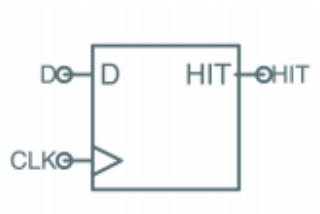
1 point possible (ungraded)
A "Froboz" is a clocked device built out of 3 interconnected components, each of which is known to be a 4-state FSM. What numeric upper bound, if any, can you put on the number of states of a Froboz?

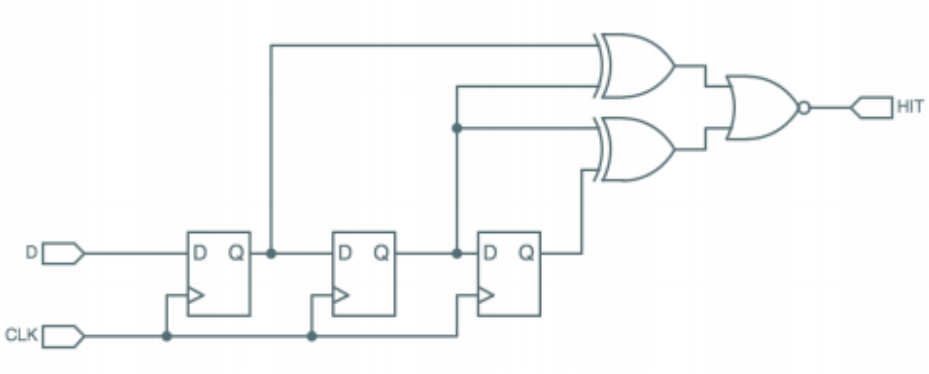
Submit

FSM

10 points possible (ungraded)

The NSA has an urgent need for a simple clocked FSM that monitors a sequence of binary data bits, appearing one bit per clock cycle, and signals a HIT whenever the last three data bits have been the same: in other words, whenever the sequence 000 or 111 appears in the input data. The circuit symbol for NSA's desired FSM is shown to the right: the D input senses one bit per clock cycle, and the HIT output must be 1 whenever three consecutive bits have had the same value. Of course, the timing of D input changes must obey the module's setup and hold time specifications. NSA's Chief Engineer, Philip Philop (a name you might recognize from his college celebrity as Captain of Harvard's Curling team) has proposed the following circuit:





A) How many states does Philip's FSM have?

Philip arranged to pre-determine the states of the three flip flops in the circuit on power-up, and had them all start in the 0 state. Unfortunately, this made the output logic (which correctly determines if all three flip flops are in the same state) report a HIT when no data had been entered. Philip, defending his circuit, said "Oh, I just picked the wrong starting state".

B) Is there ANY initial state for which Philip's design will work properly? If so, give the 3-bit initial state; otherwise, write NONE.

While emptying the trash, a sharp-eyed NSA janitor notices a partially-completed table showing state transitions for a 7-state FSM, apparently discarded by a recent MIT intern who was hired after acing 6.004 and worked on this project. The table is shown below.

On seeing the table, Philip shouts "EUREKA" and immediately orders his FSM to be replaced by the newly-discovered version.

Unfortunately, he needs your help...

C) Fill in the missing entries to the state transition table.

Current State	Input (D)	Next State	HIT
Sx	0	S0	0
Sx	1	S1	0
S0	0	S00	0
S0	1		0
S00	0	S000	0
S00	1		0
S000	0		
S000	1	S1	
S1	0	S0	0
S1	1	S11	0
S11	0	S0	
S11	1	S111	
S111	0	S0	1
S111	1		1

Submit

Discussion

Hide Discussion

Topic: 6. Finite State Machines / Tutorial : FSM

Add a Post

Show all posts by recent activity

? Sequential Logic Timing of FSM part A 3

The answer given for minimum number of states is 6 but I have a possible method to implement it i...

[[STAFF]] 2nd Question here & LE6.2.1 !! 3

[[STAFF]] Please I want to know the difference between the second question here and question (C...