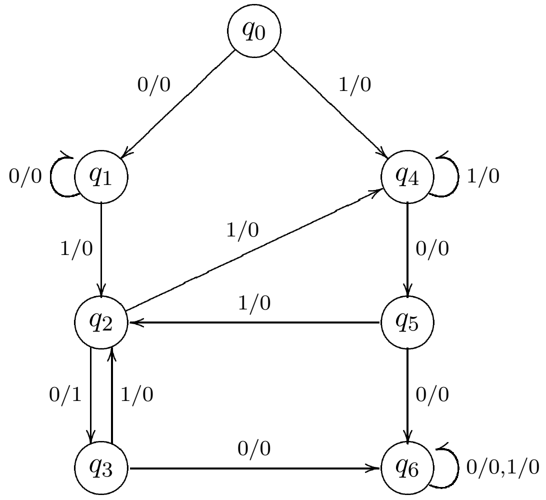
DigSim Assignment 1: A Finite State Machine

Due: November 25, 2003

Objective: The objective of this assignment is to implement a finite state machine using DigSim.

Assignment: Consider the finite state machine represented below as a transition diagram[1]:



This finite state machine starts in state q_0 and has one input bit and one output bit. The output bit is 1 if the input sequence up to the current point ends with 010 as long as the sequence 100 has never been seen. For example, the machine outputs 1 after reading 00011010, but outputs 0 after reading 110110010. (Verify for yourself that the transition diagram fits the description.)

Your assignment is to implement this finite state machine in DigSim. You must:

1. Use three D flip-flops to store the 7 states of the machine. State q_0 will be represented as 000, $q_1 = 001$, $q_2 = 010$, ..., $q_6 = 110$. The bit pattern 111 is not used.
2. Let s_2, s_1, s_0 be the state bits stored in the D flip-flops, x be the input bit and z be the output bit. Fill in the attached truth table for the next state bits s'_2, s'_1, s'_0 and the output bit z.
3. For s'_2, s'_1, s'_0 and z, use Karnaugh maps to obtain simplified SOP or POS Boolean formulas.
4. Implement the finite state machine in DigSim. You should study the "Sequence Detector" example in DigSim (use "Open example" under the File menu) for suggestions on the layout of your finite state machine.

Implementation notes:

- Label the switches and flip-flops in your circuit appropriately.
- If you need a 4-input OR gate, you need to use two layers of 2-input or 3-input OR gates to accomplish the same function. Ditto for 4-input AND gates.

[1] Adapted from *Contemporary Logic Design,* Randy H. Katz, Benjamin/Cummings Publishing, 1994.

- Make sure to leave time to debug your circuit. Note that to restart the finite state machine in the 000 state, you need to save the circuit and load it back into DigSim.
- The D flip-flops in DigSim are positive-edge triggered. To change the state of the flip-flop, change the input when the clock is low, then bring the clock from low to high. The input to the D flip-flop when the clock changes from low to high will be stored in the flip-flop.

What to submit:

1. Make a copy of your truth-table and Karnaugh maps and submit the hard copy in class on Tuesday November 25.
2. Save your circuit as you did in the DigSim part of Homework 4. Submit the circuit file using the Unix `submit` command as in previous assignments. The submission name for this assignment is: `digsim1`. The UNIX command to do this should look something like:

```
submit cs313_0101 digsim1 fsm.sim
```

Name: ______________________________

Truth table:

m	s_2	s_1	s_0	x	s_2'	s_1'	s_0'	z
0	0	0	0	0				
1	0	0	0	1				
2	0	0	1	0				
3	0	0	1	1				
4	0	1	0	0				
5	0	1	0	1				
6	0	1	1	0				
7	0	1	1	1				
8	1	0	0	0				
9	1	0	0	1				
10	1	0	1	0				
11	1	0	1	1				
12	1	1	0	0				
13	1	1	0	1				
14	1	1	1	0	d	d	d	d
15	1	1	1	1	d	d	d	d

				s2		
	s0 x \ s2 s1	00	01	11	10	
	00	0	4	12	8	
	01	1	5	13	9	x
s0	11	3	7	15	11	
	10	2	6	14	10	
			s1			

s2’ =

				s2		
	s0 x \ s2 s1	00	01	11	10	
	00	0	4	12	8	
	01	1	5	13	9	x
s0	11	3	7	15	11	
	10	2	6	14	10	
			s1			

s1’ =

				s2		
	s0 x \ s2 s1	00	01	11	10	
	00	0	4	12	8	
	01	1	5	13	9	x
s0	11	3	7	15	11	
	10	2	6	14	10	
			s1			

s0’ =

				s2		
	s0 x \ s2 s1	00	01	11	10	
	00	0	4	12	8	
	01	1	5	13	9	x
s0	11	3	7	15	11	
	10	2	6	14	10	
			s1			

z =