

Fundamentals of Digital System Design

ECE/CS 3700

Spring 2009, Homework # 5

Due Date: Wed, April 29 by 5pm in the HW locker.

1. (15 points) **Generalized Counter Design: Interpreting counters as finite state machines.** You are asked to design a “synchronous” 3-bit counter that advances through the sequence: 000, 010, 011, 101, 110, 000, and repeats. Not all possible combinations of the 3 bits represent valid (or legal, or care) states of the counter. The unused (illegal, or don’t care) states are 001, 100, 111. These states can be used as don’t care conditions to simplify the logic used to implement the counter.

- The state transition table of the above counter is shown below, where a, b, c represent present state values and a^+, b^+, c^+ , represent next state values in the latches. d denotes don’t cares in the next state columns.

TABLE I

STATE TRANSITION TABLE OF THE COUNTER

Present State	Next State
abc	$a^+b^+c^+$
0 0 0	0 1 0
0 0 1	d d d
0 1 0	0 1 1
0 1 1	1 0 1
1 0 0	d d d
1 0 1	1 1 0
1 1 0	0 0 0
1 1 1	d d d

- Express each next-state bit as a combinational logic function of the three present-state bits. Use the don’t cares to minimize the functions.
- Implement the counter using D -type flip-flops. Draw the circuit diagram and show that the counter advances through $000 \rightarrow 010 \rightarrow 011 \rightarrow 101 \rightarrow 110 \rightarrow 000 \rightarrow \dots$, and repeats.

2. (15 points) **State Assignment:** Solve Problem number 8.1 (page 573) in the text-book. Is the machine described in the state transition table a Mealy or a Moore machine? Why? Draw the circuit schematic.

3. (20 points) **Analyzing the effect of different state encodings on the quality of the circuit:** In this question, you will analyze the effect of selecting a different code assignment for states of the machine used in the previous question. Consider the machine described in the state table in Fig. P8.1 (page 575) in the textbook. The machine has four states. Let us call them A, B, C, and D. In the figure, Code 00 corresponds to state A; 01 to state B; 10 to

state C; and 11 to state D. In the previous question you have already derived the logic expressions for the next states and outputs, corresponding to the above state encodings. Let us now re-encode the machine using the following assignments: 11 for A; 10 for B; 00 for C; and 01 for D. Re-write the state table using the new assignments. *Note: Re-assignment of the codes takes place not just in the present states, but also in the next states.* Derive the new next state (Y_1, Y_2) and output (z) equations and draw the circuit diagram. Which code assignment is a better choice? The one shown in Fig. P8.1, or the one analyzed in this question?

4. **(20 points) FSM minimization:** For the FSM shown in Table II, do the following:

- Identify all the equivalent states of the machine.
- Minimize the machine and show the minimized state table.

TABLE II
STATE TRANSITION TABLE OF THE MACHINE

Present State	Next State, Output	
	$x = 0$	$x = 1$
A	B, 1	H, 1
B	F, 1	D, 1
C	D, 0	E, 1
D	C, 0	F, 1
E	D, 1	C, 1
F	C, 1	C, 1
G	C, 1	D, 1
H	C, 0	A, 1

5. (15 points) **FSM design for string matching circuits.** In this question, you are asked to design a FSM, and subsequently the corresponding sequential circuit, for Problem 8.9 in the text-book, page 575. Derive the FSM and the circuit. Is yours a Mealy or a Moore machine?

6. **(15 points)** You are asked to design a Finite State Machine (FSM) for a washing machine controller. The machine starts when a *laundry token* is deposited. It then sequences through the following stages: *soak*, *wash*, *rinse* and *spin*. There is a “double wash” switch, which, if turned on, causes a second *wash* and *rinse* to occur. In other words, the “double wash” switch causes the following sequence: *soak*, *wash*, *rinse*, *wash*, *rinse* and then finally the *spin* cycle. Each stage takes the same amount of time. There is a timer that begins ticking as soon as the laundry token is deposited. The timer generates a T signal at the end of the time period, and then resets itself and starts again. This timer signal causes the washing machine to sequence through its stages. If the lid of the machine is raised during (and only during) the spin cycle, the machine stops spinning until the lid is closed. To simplify your problem, you may assume that the timer stops ticking when the lid is raised. Do the following:

(a) Identify and list all the inputs and outputs of your system. You may assume that there are sensors associated with the lid, the “double wash” switch, token handler, and any other sensing device you may require.

(b) Draw the state transition diagram (or a state table, if you wish) of your finite state machine.

(c) Is your machine a Mealy or Moore type FSM?

Note: Just design the FSM. No need to minimize, encode or synthesize it.