CSEE W3827

Fundamentals of Computer Systems

Homework Assignment 3

Prof. Martha A. Kim

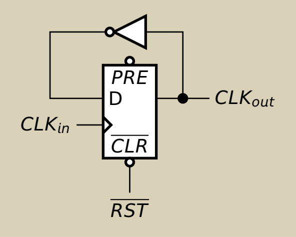
Columbia University

Due October 10, 2013 at 10:10 AM

Write your name **and UNI** on your solutions

Show your work for each problem; we are more interested in how you get the answer than whether you get the right answer.

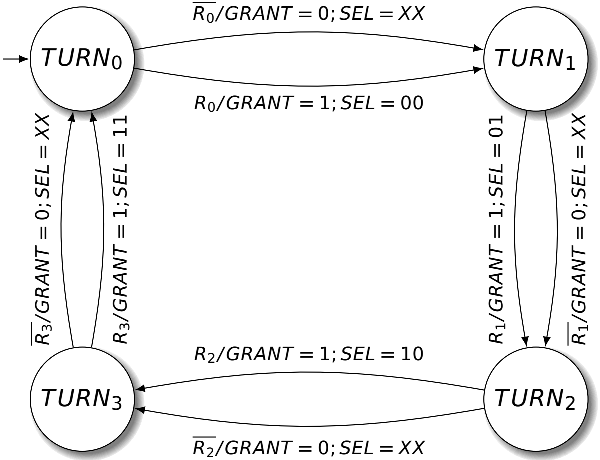
1. (10 pts.) Draw the bubble and arc diagram for the clock divider shown below.



2. (10 pts.) Draw a schematic for an 32-bit up/down counter, using any combinational blocks you wish (except for a counter). Given two inputs INC and DEC, and the current value $Q_{31:0}$, your counter should behave as specified in the table below. The superscript $^+$ denotes the *next* value. For this problem, you may ignore any reset signal.

INC	DEC	$Q^+_{31:0}$
0	0	$Q_{31:0}$
1	0	$Q_{31:0}+1$
0	1	$Q_{31:0}-1$
1	1	$Q_{31:0}$

3. (20 pts.) Examine the state transition diagram below. It specifies the behavior of a four-input round robin arbiter. Each input i gets a turn, and if there is a request on input i ($R_i = 1$) during i's turn, the arbiter grants access to that input ($GRANT = 1; SEL = i$).


$\overline{R_0}/GRANT = 0; SEL = XX$
$TURN_0$
$TURN_1$
$R_0/GRANT = 1; SEL = 00$
$\overline{R_3}/GRANT = 0; SEL = XX$
$R_3/GRANT = 1; SEL = 11$
$R_1/GRANT = 1; SEL = 01$
$\overline{R_1}/GRANT = 0; SEL = XX$
$R_2/GRANT = 1; SEL = 10$
$TURN_3$
$TURN_2$
$\overline{R_2}/GRANT = 0; SEL = XX$

(a) (10 pts.) Give expressions for the next state logic (D_1 and D_0) and output logic ($GRANT$ and $SEL_{1:0}$) using the state encoding below.

State	Q_1	Q_0
$TURN_0$	0	0
$TURN_1$	0	1
$TURN_2$	1	0
$TURN_3$	1	1

(b) (10 pts.) Give expressions for the next state logic (D_3, D_2, D_1 and D_0) and output logic ($GRANT$ and $SEL_{1:0}$) using the state encoding below.

State	Q_3	Q_2	Q_1	Q_0
$TURN_0$	0	0	0	1
$TURN_1$	0	0	1	0
$TURN_2$	0	1	0	0
$TURN_3$	1	0	0	0

4. (30 pts.) In this problem you will implement a two-entry queue controller. The queue can hold up to two entries, and supports both enqueue and dequeue operations. The controller FSM, which you will implement, has the following interface and behavior:

   - Input E is 1 bit indicating an enqueue operation.
   - Input D is 1 bit indicating a dequeue operation.
   - Input CLK, has the usual behavior.
   - Input $\overline{RST}$ resets the controller to the initial state where the queue is empty.
   - Output ERR, is 1 bit indicating whether or not there has been an error in the operation of the queue. Any attempt to enqueue to a full queue, dequeue from an empty queue, or enqueue and dequeue simultaneously cause $ERR = 1$. After an error the only way to resume normal operation is via a reset.
   - Output CE, for "can enqueue", is a 1 bit signal indicating that it is safe to enqueue.
   - Output CD, for "can dequeue", is a 1 bit signal indicating that it is safe to dequeue. When $ERR = 1$, both CD and CE are don't cares.

(a) (10 pts.) Draw a bubble and arc diagram for a Moore machine implementation of the queue controller, using the following four states

- HAS_0,HAS_1,HAS_2 indicating the number of entries in the queue, i.e., in HAS_0 state the queue is empty.
- ERR is the error state

(b) (10 pts.) Using the state encoding shown below, give *minimal* expressions for the output logic, CE, CD, ERR

State	Q_1	Q_0
HAS_0	0	0
HAS_1	0	1
HAS_2	1	0
ERR	1	1

(c) (10 pts.) Using the same state encoding, give *minimal* expressions for the next state logic.

5. (30 pts.) Consider a subway turnstile that is either locked or unlocked. When locked, nobody can pass through the turnstile, except by inserting a coin, to unlock it. When unlocked, one person can push on the crossbar and pass through, at which point the turnstile becomes locked again. Three attempts to push on a locked turnstile will trigger an alarm. Inserting a coin when the turnstile is already unlocked will cause the turnstile to stay unlocked until one person has passed. In other words, inserting two coins will allow only one person to pass through, not two. Once an alarm has sounded, only a reset will stop it.

   The controller for this turnstile accepts two inputs push (P) and coin (C) as well as clock (CLK) and reset ($\overline{RST}$). The outputs indicate whether the turnstile is locked (L) and whether or not the alarm should be sounding (A).

(a) (10 pts.) Draw the state transition diagram for a Moore implementation of this turnstile.

(b) (0 pts.) Pick a state encoding for the turnstile FSM.

(c) (10 pts.) Give expressions for the next state logic of the turnstile FSM.

(d) (10 pts.) Give expressions for the output logic of the turnstile FSM.