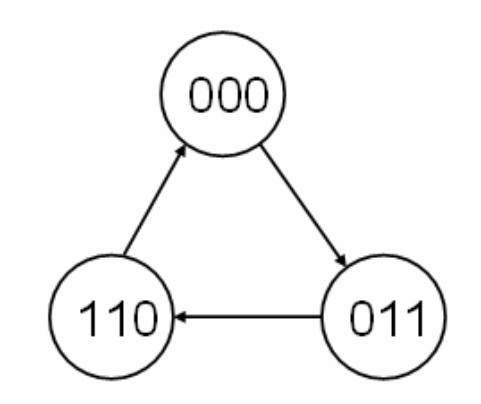
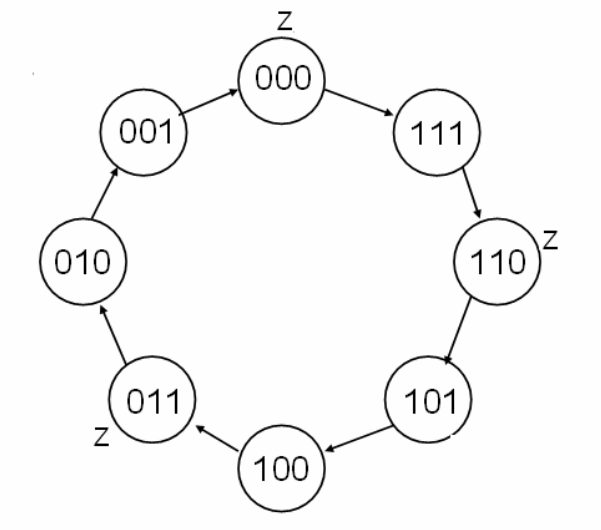
ECEn/CS 224
Chapter 15 Homework Solutions

15.1 Draw the state graph and corresponding transition table for a 3-bit counter with no control inputs and which counts in multiples of 3. That is, the count sequence is: 000 - 011 - 110 - 000. Use don't cares in the transition table as appropriate.



Q2	Q1	Q0	N2	N1	N0
0	0	0	0	1	1
0	1	1	1	1	0
1	1	0	0	0	0
0	0	1	x	x	x
0	1	0	x	x	x
1	0	0	x	x	x
1	0	1	x	x	x
1	1	1	x	x	x

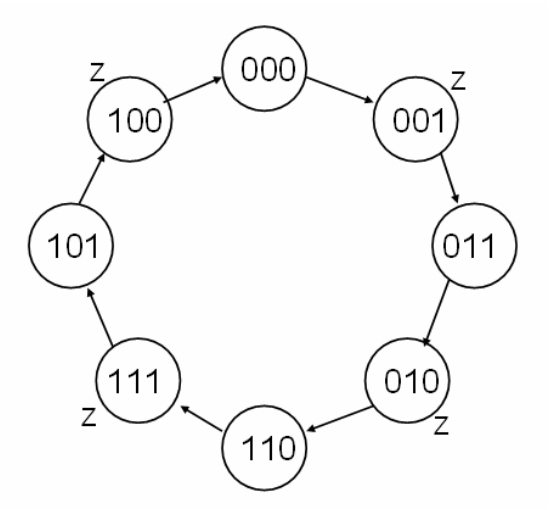
15.2 Draw the state graph and corresponding transition table for a 3-bit binary counter with no control inputs and which counts down rather than up. Include a Z output which signifies when the “count value modulo 3 is equal to 0”.



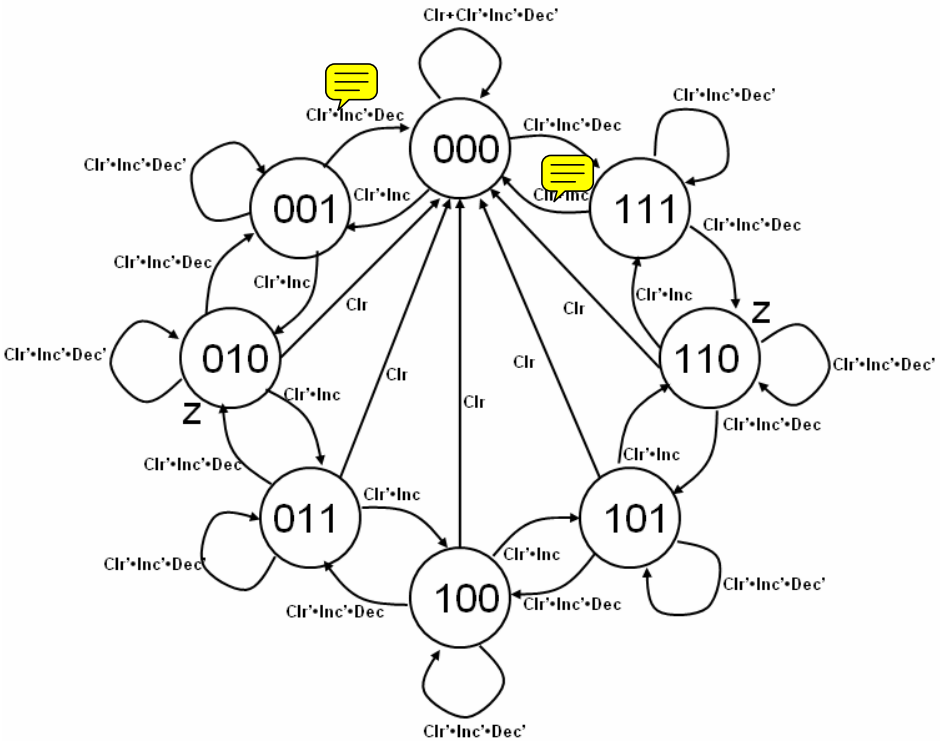
Q2	Q1	Q0	N2	N1	N0	Z
0	0	0	1	1	1	1
1	1	1	1	1	0	0
1	1	0	1	0	1	1
1	0	1	1	0	0	0
1	0	0	0	1	1	0
0	1	1	0	1	0	1
0	1	0	0	0	1	0
0	0	1	0	0	0	0

15.3 Draw the state graph for a 3-bit counter which counts in the gray code shown below. Include a Z output which signifies when the even/odd characteristic of the current state and of the next state values (when interpreted as unsigned integers) are the same. That is, if the transition is '000' → '010' (0 → 2) the output should be asserted. However, if the transition is '101' → '100' (5 → 4) the output should not be asserted.

Gray Code
000
001
011
010
110
111
101
100



15.4 Draw the state graph for a 3-bit up/down counter which has the following inputs: CLR, INC, DEC. Make it so the inputs have priority in that order. Add a Z output which signifies when the current state of the counter is either '010' or '110'.



15.5 Perform the completeness and conflict tests on the state graph created for the previous problem. For each state, show the OR of all the conditions on the arcs leaving that state and verify that it is indeed '1'. For each state, list all the pairs of arc conditions that must be ANDed for conflict testing. Then do all the ANDs and verify that they indeed result in 0's.

Because of the way the way we drew the state diagram in the previous problem, states 000, 001, 010, 011, 100, 101, and 110 all have the same transitions. Thus, we do not need to perform the checks for every state, since many of them will be the same. (Note that there are other valid ways to draw the state diagram in which more or less checks may be needed). The checks for the first seven states are as follows:

Complete:

(Clr'•Inc'•Dec') + (Clr'•Inc'•Dec) + (Clr'•Inc) + Clr
= Clr'•Inc'(Dec'+Dec) + (Clr'•Inc) + Clr
= (Clr'•Inc') + (Clr'•Inc) + Clr
= Clr'(Inc'+Inc) + Clr
= Clr' + Clr
= 1

Conflict free:

(Clr'•Inc'•Dec') (Clr'•Inc'•Dec) = (Clr'•Inc')(Dec'•Dec) = 0
(Clr'•Inc'•Dec') (Clr'•Inc) = (Clr'•Dec') (Inc'•Inc) = 0
(Clr'•Inc'•Dec') Clr = (Inc'•Dec') (Clr'•Clr) = 0
(Clr'•Inc'•Dec) (Clr'•Inc) = (Clr'•Dec) (Inc'•Inc) = 0
(Clr'•Inc'•Dec) Clr = (Inc'•Dec') (Clr'•Clr) = 0
(Clr'•Inc) Clr = Inc(Clr'•Clr) = 0

For state 111, the checks are as follows:

Complete:

(Clr + Inc) + (Clr'•Inc'•Dec') + (Clr'•Inc'•Dec)
= (Clr + Inc) + Clr'•Inc'(Dec'+Dec)
= (Clr + Inc) + Clr'•Inc'
= (Clr'•Inc')'+ (Clr'•Inc')
= 1

Conflict free:

(Clr+Inc)(Clr'•Inc'•Dec') = Clr(Clr'•Inc'•Dec') + Inc(Clr'•Inc'•Dec') = 0
(Clr + Inc)(Clr'•Inc'•Dec) = Clr(Clr'•Inc'•Dec) + Inc(Clr'•Inc'•Dec) = 0
(Clr'•Inc'•Dec')(Clr'•Inc'•Dec) = (Clr'•Inc')(Dec'•Dec) = 0

15.6 Draw the state graph and transition table for a 2-bit counter which has the following inputs: CLR, ONE/TWO#, UP/DOWN#, where CLR has the highest priority. When ONE/TWO#=’1’, have it count by 1. When it is ’0’, have the counter count by 2. When UP/DOWN# = '1' have the counter count up. When this signal is '0', have the counter count "down".

The solution shown below also adds a Y output, which is not required for this problem.