

Homework 10 – due 6/17

6-11) A binary ripple counter uses FFs that trigger on the positive-edge of the clock. What will be the count if (a) the normal outputs of the FFs are connected to the clock and (b) the complement outputs are connected to the clock?

Assume a 4-bit counter for demonstration purposes. A is input FF

a) If Q is connected to next stage clock:

State			
A	B	C	D
0	0	0	0
1	1	1	1
0	1	1	1
1	0	1	1
0	0	1	1
1	1	0	1
0	1	0	1
1	0	0	1
0	0	0	1
1	1	1	0
0	1	1	0
1	0	1	0
0	0	1	0
1	1	0	0
0	1	0	0
1	0	0	0
0	0	0	0

This counter is counting backwards (15, 14, 13, ... 2, 1, 0, 15)

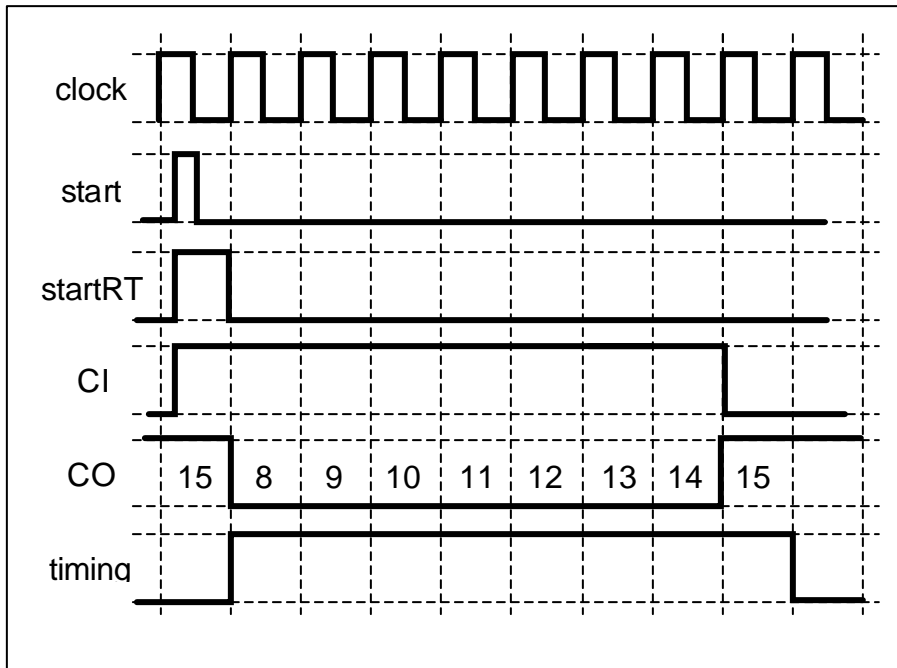
b) If Q' is connected to next stage clock:

State			
A	B	C	D
0	0	0	0
1	0	0	0
0	1	0	0
1	1	0	0
0	0	1	0
1	0	1	0
0	1	1	0
1	1	1	0
0	0	0	1
1	0	0	1
0	1	0	1
1	1	0	1
0	0	1	1
1	0	1	1
0	1	1	1
1	1	1	1
0	0	0	0

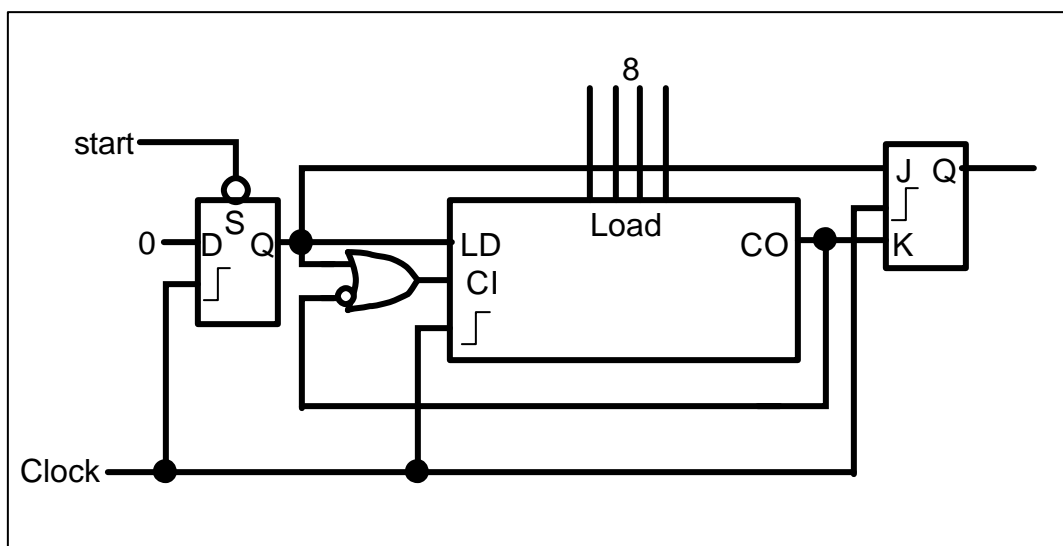
This counter is counting forwards (0, 1, 2, 3, ... , 14, 15, 0)

6-23) Design a timing circuit that provides an output signal that stays on for exactly eight clock cycles. A start signal sends the output to the 1 state and after eight clock cycles the signal returns to the 0 state.

Here is the timing diagram for the timing circuit. The first signal is the clock. The second signal is an asynchronous start signal. The third signal is the retimed start signal. The fourth signal is the counter carry in. The fifth signal is the carry out (with state shown), while the last signal is the 8 clock cycle timing signal.



Here is the circuit to implement this timing circuit:



6-29) List the eight unused states in the switch-tail ring counter of Fig 6-18(a).

Determine the next state for each of these states and show that, if the counter finds itself in an invalid state, it does not return to a valid state. Modify the circuit as recommended in the text and show that the counter produces the same sequence of states and that the circuit reaches a valid state from any one of the unused states.

The used states are 0000, 1000, 1100, 1110, 1111, 0111, 0011, and 0001, so the unused states are 0010, 0100, 0101, 0110, 1001, 1010, 1011, and 1101. The next state is the right shifted version of the present state with the leftmost of the new state bit set as the complement of the rightmost bit of the previous state. The present state and next state are written to emphasize the derivation of the values.

State	Next state
001 0	1 001
010 0	1 010
010 1	0 010
011 0	1 011
100 1	0 100
101 0	1 101
101 1	0 101
110 1	0 110

All of these states have an invalid state as the next state, so the counter never recovers.

Instead of connecting Q_B to D_C , the text recommends setting $D_C = (Q_A + Q_C)Q_B$. Here is the state table that results:

State	$D_A=Q_E'$	$D_B=Q_A$	$D_C = (A+C)B$	$D_E=Q_C$	Next state
0000	1	0	0	0	1000
1000	1	1	0	0	1100
1100	1	1	1	0	1110
1110	1	1	1	1	1111
1111	0	1	1	1	0111
0111	0	0	1	1	0011
0011	0	0	0	1	0001
0001	0	0	0	0	0000
0100	1	0	0	0	1000
0101	0	0	0	0	0000
0110	1	0	1	1	0011
1001	0	1	0	0	0100
1010	1	1	0	1	1101
1011	0	1	0	1	0101
1101	0	1	1	0	0110

With this change, the previous valid states perform as they should. Invalid states 0100, 0101, and 0110 transition directly into a valid state. Invalid states 1001, 1010, 1011, and 1101 transition into valid states after 2 or 3 clock cycles.