LECTURE 14: DIGITAL LOGIC CIRCUIT DESIGN

Synchronous Counters

Synchronous counters can be designed using the design procedure of the clocked sequential circuits. The count pulses are applied to the clock inputs of all flip-flops of the counter. Therefore the changes in the outputs occur at the same time. Several typical synchronous counters are presented in this section and their operation explained.

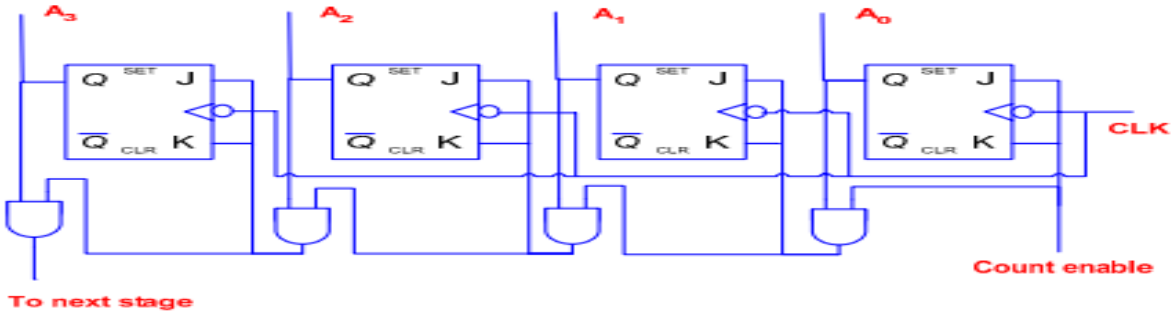
Synchronous binary counter

The operation of the synchronous binary counter is so simple and follows an easy complementing pattern that a simple design procedure can be followed. There is no need to follow the general procedure for the design of synchronous sequential circuits. If we inspect the count cycle or the timing diagram of the binary counter we find the following:

1. A_0 complements every time the count pulses go from 1 to 0.
2. A_1 complements only when A0 is 1 and goes to 0.
3. A_2 complements only when A1 and A0 are 1 and going to 0.
4. A_3 complements only when A_2, A_1 and A_0 are all 1 and going to 0.

From this pattern it becomes clear that the J and K inputs of flip-flop A0 should be kept at 1 (the count enable can be used in stead of 1). The count enable is Anded with A0 and applied to the J and k inputs of A1. This pattern is repeated by Anding the J and K input of each flip-flop with the flip-flop output and applied to the J and K inputs of the next flip-flop.

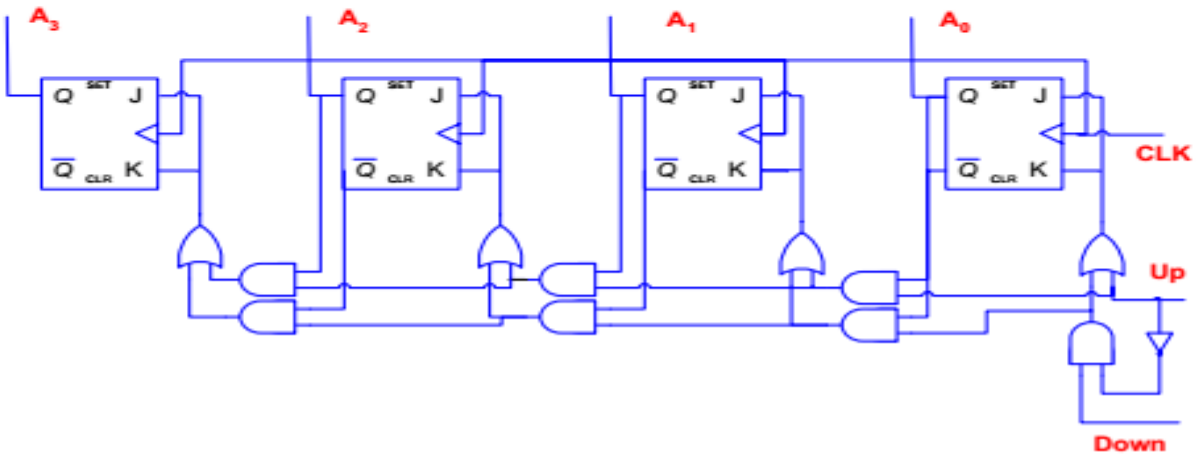
The logic circuit of a modulus-16 synchronous binary counter using JK flip-flops is given next.



LECTURE 14: DIGITAL LOGIC CIRCUIT DESIGN

Up-Down Counter

The down counter counts in reverse from 1111 to 0000 and then goes to 1111. If we inspect the count cycle, we find that each flip-flop will complement when the previous flip-flops are all 0 (this is the opposite of the up counter). The down counter can be implemented similar to the up counter, except that the AND gate input is taken from Q' instead of Q. This is shown in the following Figure of a 4-bit up-down counter using T flip-flops.



BCD Synchronous Counter

The BCD counter does not have a regular pattern. Therefore we have to follow the design procedure of synchronous sequential circuits. Using T flip-flops, the state table of the counter will be given as follows:

P.S.				N.S.					Flip-flop inputs			
Q_8	Q_4	Q_2	Q_1	Q_8	Q_4	Q_2	Q_1	y	T_8	T_4	T_2	T_1
0	0	0	0	0	0	0	1	0	0	0	0	1
0	0	0	1	0	0	1	0	0	0	0	1	1
0	0	1	0	0	0	1	1	0	0	0	0	1
0	0	1	1	0	1	0	0	0	0	1	1	1
0	1	0	0	0	1	0	1	0	0	0	0	1
0	1	0	1	0	1	1	0	0	0	0	1	1
0	1	1	0	0	1	1	1	0	0	0	0	1
0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	0	0	1	0	0	0	0	1
1	0	0	1	0	0	0	0	1	1	0	0	1

The flip-flop input equations can be simplified by means of Karnaugh maps:

$T_1 = 1$

$T_2 = Q_8' Q_1$

$T_4 = Q_2 Q_1$

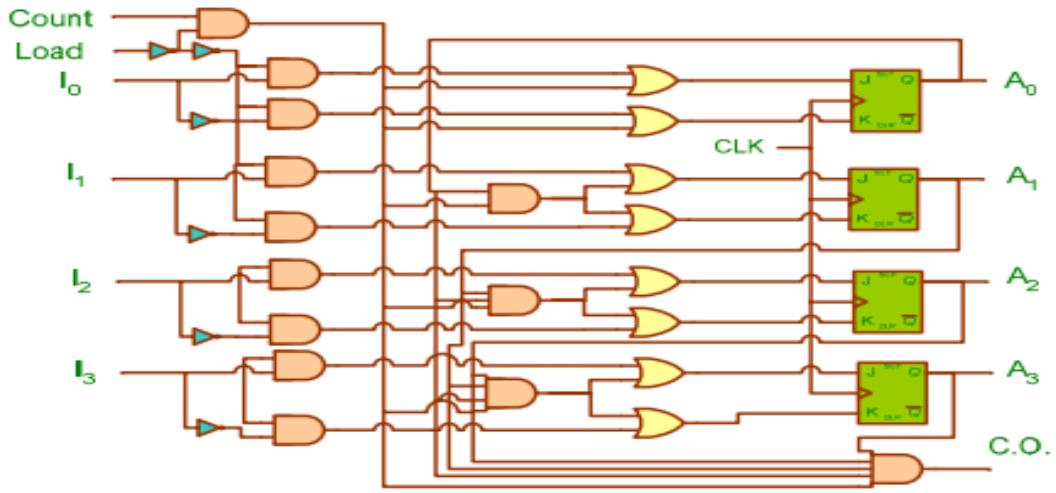
$T_8 = Q_8 Q_1 + Q_4 Q_2 Q_1$

And $y = Q_8 Q_1$

Binary counter with parallel load

Parallel load capability is used for transmission of an initial binary number into the counter prior to the count operation. The logic diagram of a 4-bit register with parallel load , which can be used as a counter. When the input load is one, it disables the count operation and allows the transfer of the data inputs into the flip-flops. If the count and load inputs are both zero, then the outputs will not change. Only when the count is one and the load is zero, then the register will act as a counter and starts counting from the last output. The function table of this counter with parallel load is shown, followed by the logic diagram.

LECTURE 14: DIGITAL LOGIC CIRCUIT DESIGN



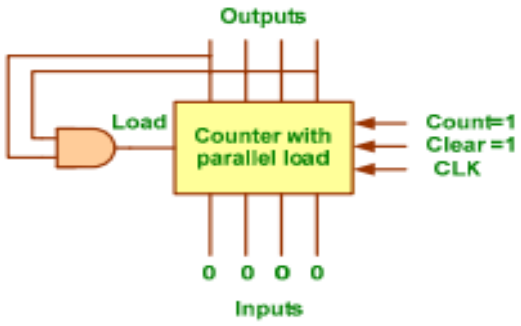
The function table for the counter shown above is as follows:

Clear	Clock	Load	Count	Function
0	X	X	X	Clear
1	↑	1	X	Load I's
1	↑	0	1	Count next
1	↑	0	0	No change

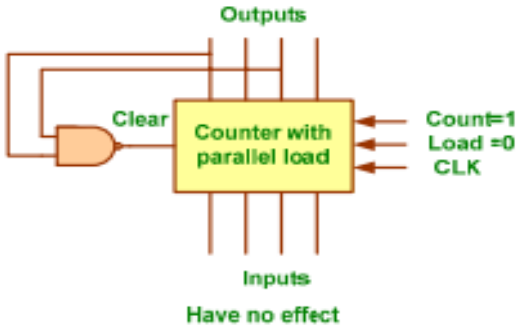
A counter with parallel load can be used to generate any desired count sequence. In the first example, the counter is used to generate the BCD count sequence by using the load input to load four zeros after reaching 1001. In the second example, the clear input is used to clear the counter after detecting that the count passed 1001 and attempts to go to 1010.

LECTURE 14: DIGITAL LOGIC CIRCUIT DESIGN

BCD counter , using the counter with parallel load and the load input:



The second example is fro the BCD counter using the counter with parallel load. This time the clear input is used instead of the load input.



Design of Counters with Unused States

When we design a counter with modulus m such that $m < 2^N$, where N is the number of flip-flops, then some states will be

unused. The unused states can be treated as don't care conditions. After completing the design, we must test to see that the counter is self correcting and goes back to one of the used states after being in an unused state.

Design of a modulus-6 counter using JK flip-flops

The count cycle of this counter is 0, 1, 2, 4, 5, 6. The state table for this counter together with the required flip-flop inputs is given in the following table:

P.S.			N.S.			Flip-Flop Inputs					
A	B	C	A	B	C	J_A	K_A	J_B	K_B	J_C	K_C
0	0	0	0	0	1	0	X	0	X	1	X
0	0	1	0	1	0	0	X	1	X	X	1
0	1	0	1	0	0	1	X	X	1	0	X
1	0	0	1	0	1	X	0	0	X	1	X
1	0	1	1	1	0	X	0	1	X	X	1
1	1	0	0	0	0	X	1	X	1	0	X

The two states 011 and 111 are not used and can be considered as don't care when simplifying the flip-flop inputs. This will result in the following simplified input functions:

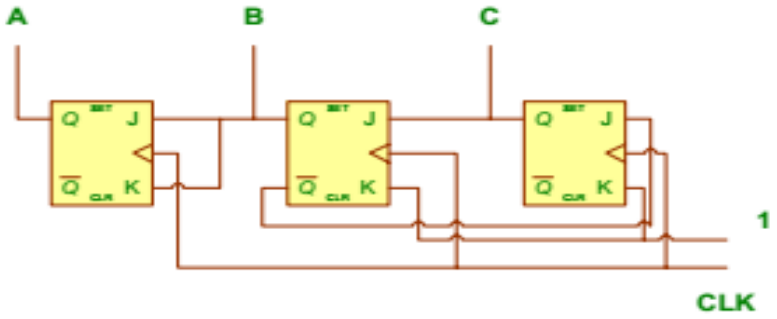
$J_A = B$ $\quad K_A = B$

$J_B = C$ $\quad K_B = 1$

$J_C = B'$ $\quad K_C = 1$

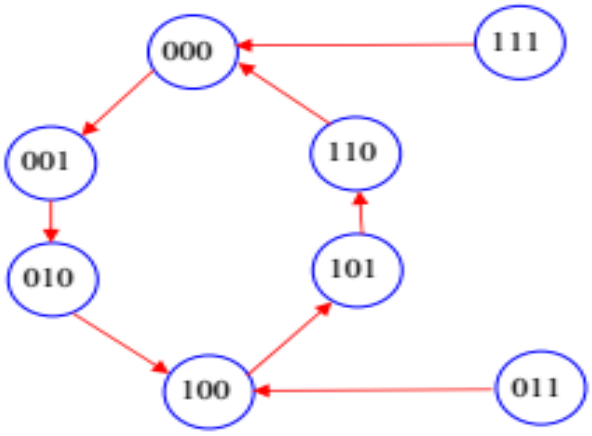
The logic diagram of the counter is shown next. We also have to determine the next state to any of the unused states. This will determine whether the design is successful or not. A successful design means that the next state to unused one is a used state after a limited number of clock pulses.

LECTURE 14: DIGITAL LOGIC CIRCUIT DESIGN



To check if the design is self correcting or not, we have to start from the unused state e.g. ABC → 011 and find the next state when a clock pulse is applied. B=1 will make A toggle (because JA=KA=B). Similarly, C=1 will make B toggle (because JB=C and KB=1). Finally B'=0 will cause no change in C (because JC=B'=0 and KC=1). This results in the next state of ABC → 100.

In a similar way we can find the next state to the second unused state 111 which will be 000. The complete state diagram, showing the transitions from the unused states, is shown next.

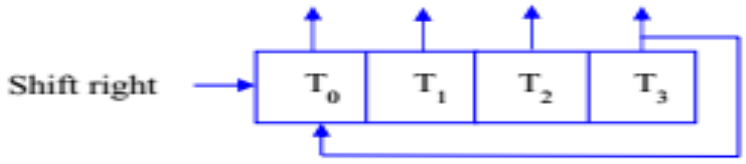


State Diagram of the modulus-6 counter

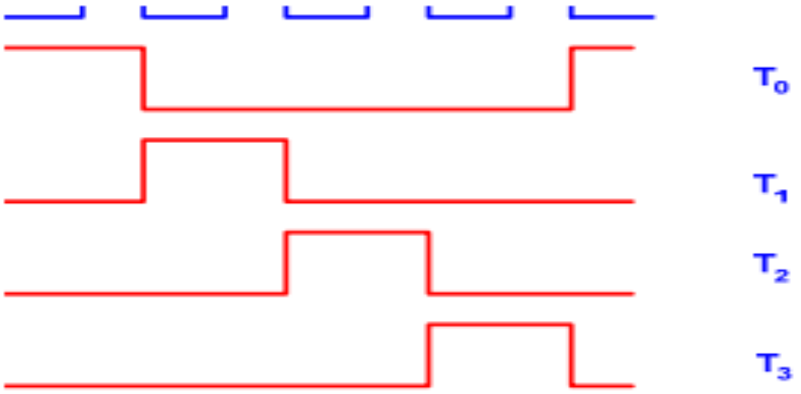
LECTURE 14: DIGITAL LOGIC CIRCUIT DESIGN

Ring Counter

Ring counter is useful in generating the timing signals that control the sequence of operations in a digital system. A ring counter is a circular shift register with only one flip-flop set at any particular time while all the others are cleared. The single bit is shifted from one flip-flop to the next to produce the sequence of timing signals. This is demonstrated by the 4-bit shift register connected as a ring counter. The generated timing signals are also shown.

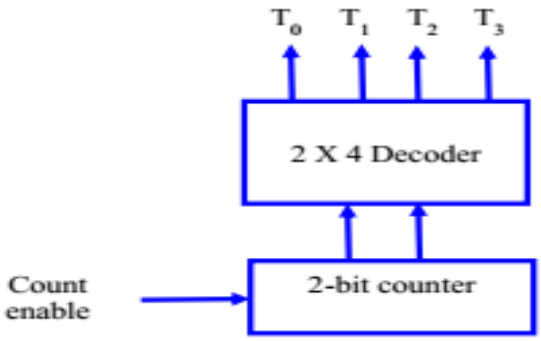


Sequence of the four timing signals



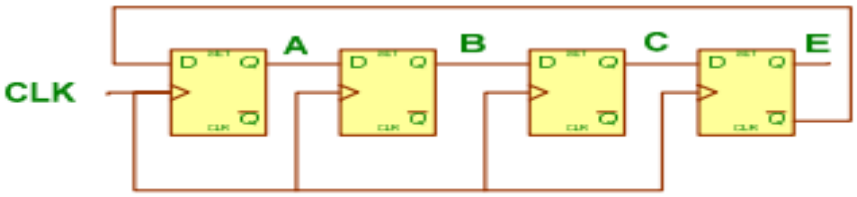
The ring counter can also be implemented with a counter and a decoder. The 4-bit ring counter needs a 2-bit counter and a 2X4 decoder. This is illustrated in the following block diagram.

LECTURE 14: DIGITAL LOGIC CIRCUIT DESIGN



Johnson Counter

The number of timing signals can be doubled if the shift register is connected as a switch-tail ring counter. This is a circular shift register with the complement output of the last flip-flop connected to the input of the first flip-flop. The register shifts its contents once to the right with every clock pulse and at the same time, the complement value of the last flip-flop is transferred into the first flip-flop. Starting from a cleared state, this switch-tail counter goes through a sequence of eight states (2k) as listed in the accompanying table.



The count sequence and required decoding are as follows:

Sequence No.	Flip-Flop Outputs				AND gate required
	A	B	C	E	
1	0	0	0	0	A'E'
2	1	0	0	0	AB'
3	1	1	0	0	BC'
4	1	1	1	0	CE'
5	1	1	1	1	AE
6	0	1	1	1	A'B
7	0	0	1	1	B'C
8	0	0	0	1	C'E