8 Counters

In this chapter, we will look at using flip flops and logic gates to design counters. There are two types of counters: asynchronous and synchronous. Asynchronous counters are also known as ripple counters as the clock pulse ripples from one flip-flop to the next. Incorrect counter output can result if the accumulative ripple delay is longer than the clock pulse. Synchronous counters, on the other hand, have clock pulse input to each flip-flop and hence do not suffer from this ripple effect. However, these counters often require additional circuitry.

8.1 Asynchronous up-counter

Figure 8.1 shows an example of a two bit asynchronous up-counter. J-K flip flops are used here although any flip-flop could be used. Two flip-flops are required in this instance as it is a two bit counter. Figure 8.2 shows the state diagram and state table, i.e. the sequence of the counter output. As there are two bits, the counter cycles through four states[12]: 00, 01, 10, 11 and it is known as up-counter since it counts in increasing order.

As can be seen from the figure, all *J* and *K* inputs are tied to logic level 1. This ensures that all the flip-flops operate in toggle mode only. The output from flip-flop 1, Q_1 is the LSB, while the output from flip-flop 2, Q_2 is MSB. The output Q_1 also acts as the input clock pulse for flip-flop 2.

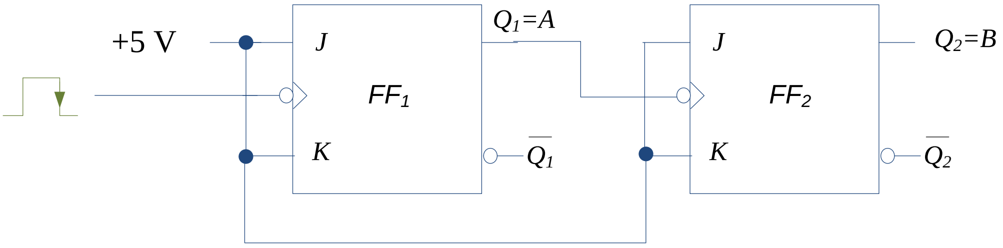
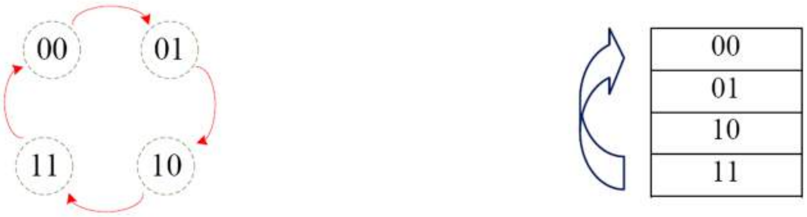


Figure 8.1: Two bit asynchronous up-counter with NGT clock pulse.



00
01
10
11

Figure 8.2: State diagram and table for two bit asynchronous up-counter.

[12] *N* flip flop give 2^N states, sometimes *N* number of flip flop counter is known as modulo *N* counter.

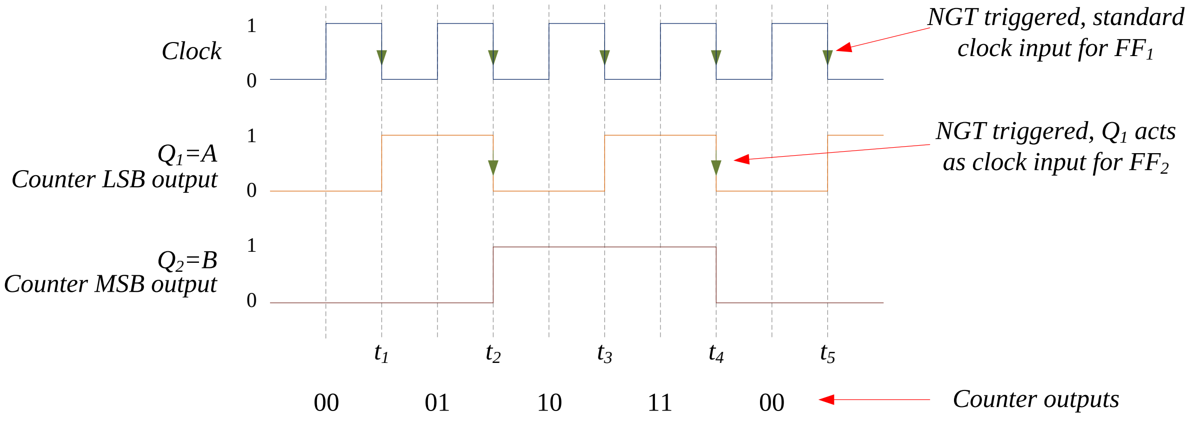


Figure 8.3: Timing diagram for two bit asynchronous up-counter with NGT clock pulse.

Analysing the timing diagram shown in Figure 8.3: at time t_1, NGT clock pulse triggers Q_1 to toggle from logic 0 to logic 1. At time t_2, NGT clock pulse causes Q_1 to change state to logic 0. As output from flip-flop 1 acts as clock input for flip-flop 2, at time t_2, Q_2 toggles to logic level 1. At time t_3, the NGT clock input toggles Q_1 to logic level 1 but there is no change in Q_2 since the clock input to flip-flop 2 at this time is PGT and not NGT. At time t_4, both Q_1 and Q_2 toggles to logic 0. It can be seen that the counter cycles through states 00→01→10→11 and the cycle is repeated.

8.1.1 Asynchronous up-counter – PGT clocked flip-flops

Figure 8.4 shows a two bit asynchronous up-counter but with the clock triggering edge to be positive going. The figure is nearly the same as Figure 8.1 except that the clock input for flip-flop 2 comes from $\overline{Q_1}$ rather than Q_1. The state diagram and state table will be the same as shown in Figure 8.2. The timing diagram is shown in Figure 8.5 where it can be seen that the flip-flop changes at PGT clock edges.

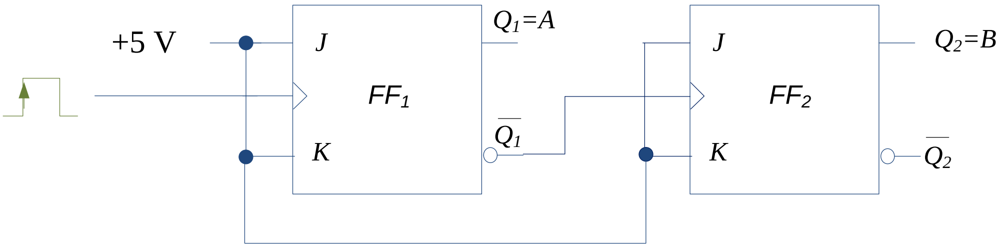


Figure 8.4: Two bit asynchronous up-counter with PGT clock pulse.

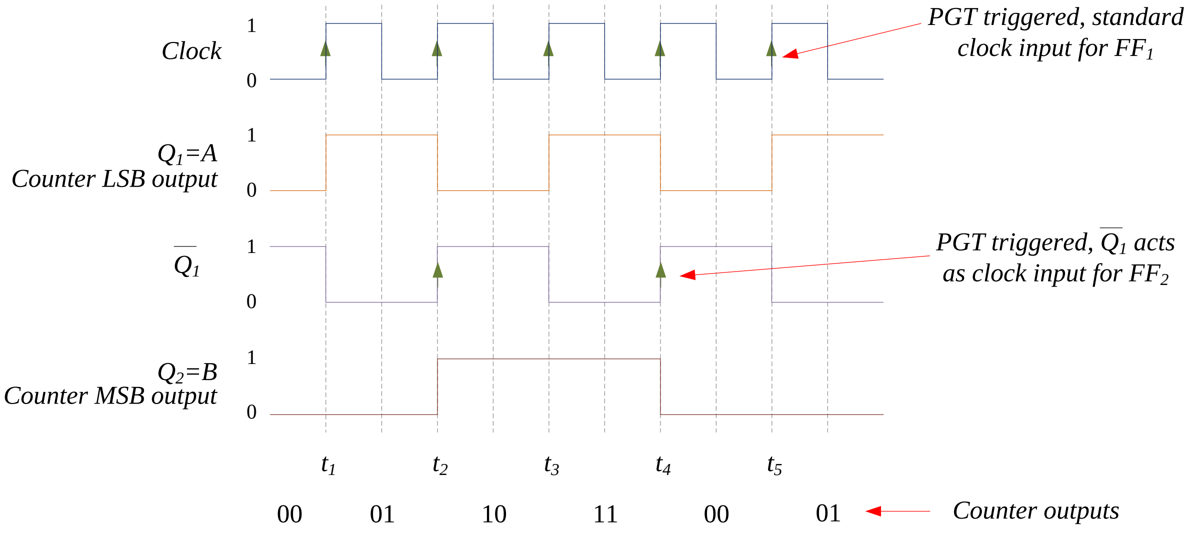


Figure 8.5: Timing diagram for two bit asynchronous up-counter with PGT clock pulse.

The timing diagram in Figure 8.5 can be analysed at follows: at time t_1, PGT clock pulse triggers Q_1 to toggle from logic 0 to logic 1. There is no change in Q_2 and the counter output is 01. At time t_2, PGT clock pulse causes Q_1 to change state to logic 0. As output from $\overline{Q}$ of flip-flop 1 acts as clock input for flip-flop 2, at time t_2, Q_2 toggles to logic level 1 and the counter output is now 10. At time t_3, the PGT clock input toggles Q_1 to logic level 1 but there is no change in Q_2 since the clock input to flip-flop 2 at this time is NGT and not PGT giving counter output of 11. At time t_4, both Q_1 and Q_2 toggles to logic 0 giving counter output of 00. At t_5, Q_1 toggles to logic 1 but there is no change in Q_2. It can be seen that the counter cycles through states 00→01→10→11 and the cycle is repeated.

8.2 Asynchronous down-counter

Figure 8.6 shows an example of a two bit asynchronous down-counter using T flip flops triggered with NGT clock pulse. The clock for the second flip-flop comes from Q_1 (similar to up-counter using PGT as shown in Figure 8.4). Figure 8.7 shows the state diagram and state table, i.e. the sequence of the counter output. The counter cycles through four states: 11à10à01à00 i.e. in decreasing order as it is a down-counter (as shown in Figure 8.8).

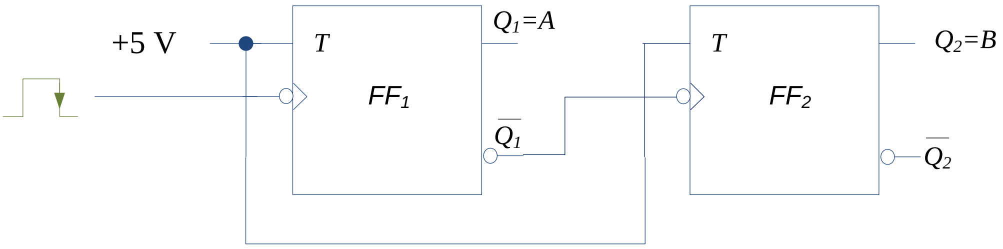


Figure 8.6: Two bit asynchronous down-counter with NGT clock pulse using T flip-flop.

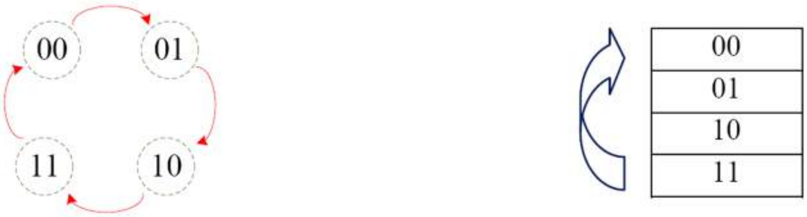


Figure 8.7: State diagram and table for two bit asynchronous down-counter.

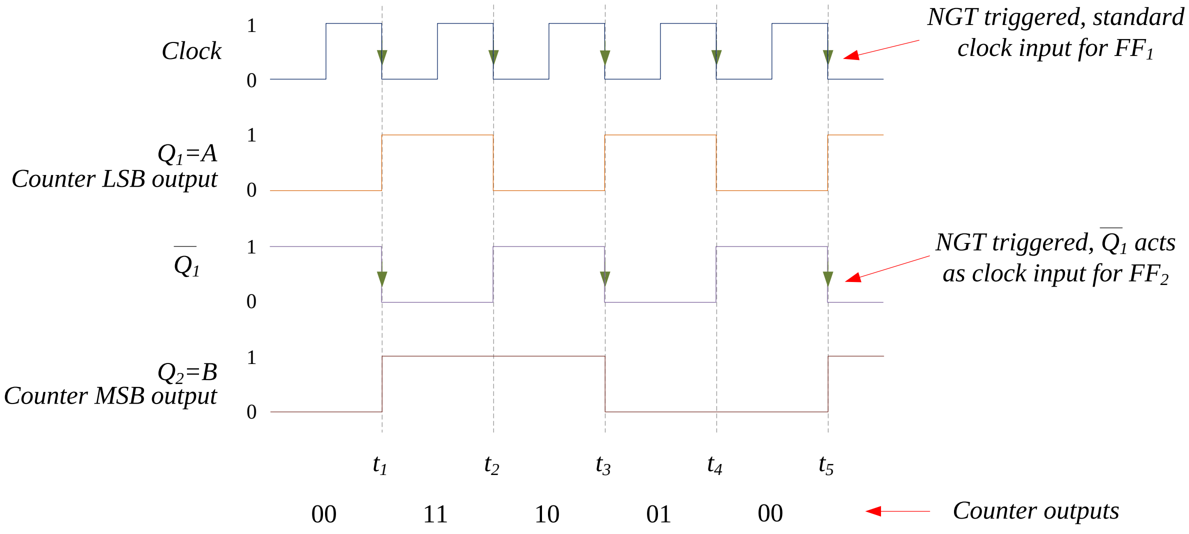


Figure 8.8: Timing diagram for two bit asynchronous down-counter with NGT clock pulse.

At t_1, the NGT clock pulse toggles the flip-flop to logic level 1. As $\overline{Q}_1$ is now the clock input for the second flip-flop, at time t_1, flip-flop 2 output toggles to logic level 1. The counter output is now 11. At time t_2, flip-flop 1 toggles to logic level 0 while there is no change in flip-flop 2 as the clock input for the second flip-flop at this time is PGT. The output is now 10. At time t_3, both flip-flop receive NGT clock inputs and toggles to opposing states as previously giving output as 01. At time t_4, flip-flop 1 toggles to logic level 0 giving counter output as 00. Hence, the counter cycles through 11→10→01→00.

Similarly, a counter with higher number of bits can be constructed. For example, a four bit asynchronous down-counter with PGT clock pulse using J-K flip flops is shown in Figure 8.9. The clock inputs (except for the first flip-flop) come from Q output of the previous flip-flop. The counter will cycle through 1111→1110→1101→1100→1011→1010→1001 →1000→0111→0110→ 0101→0100→ 0011→0010→0001→0000.

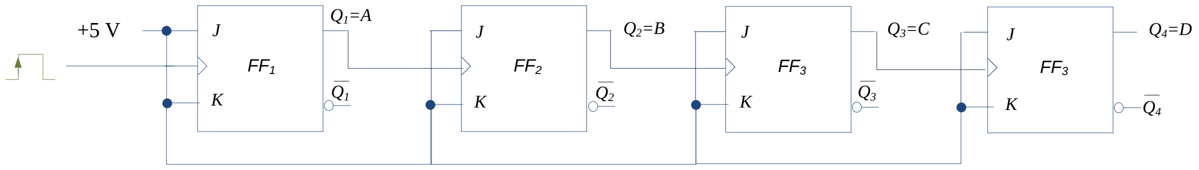


Figure 8.9: Four bit asynchronous down-counter with PGT clock pulse.

Table 8.1 gives a summary of the clock inputs for the second flip-flop onwards against the up/down counter and trigger edge types.

Table 8.1: Clock inputs for the second flip-flop onwards against the up/down counter and trigger edge types

	Clock input
Up-counter NGT	Q
Up-counter PGT	$\overline{Q}$
Down-counter NGT	$\overline{Q}$
Down-counter PGT	Q

8.3 Asynchronous counters with incomplete cycles

So far, we have seen counters that complete the cycle for the specific number of bits, for example a two bit up-counter would have four states: 00→01→10→11 and a three bit down counter would have eight states: 111→110→101→100→011→010→001→000. Consider a case where we need a counter only to count from 00→01→10. Two flip-flops will be required but the counter has to reset to 00 after 10 and not after 11. Therefore additional circuitry will be required to reset the counter after 10. The state diagram is shown in Figure 8.10 where the temporary state of 11 will only occur for a very short period of time and hence will not appear in the counter cycle. For this purpose, the clear asynchronous input of the flip-flops together with a NAND gate could be used to reset both flip-flops instantaneously[13] . This situation is shown in Figure 8.11. As soon as the state Q_2=1 (i.e. B=1) and Q_1=1 (i.e. A=1) occur, the clear inputs reset all the flip-flips to 0 and the counter then resumes its cycle. Figure 8.12 shows the timing diagram for this counter.

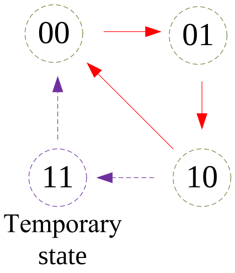


Figure 8.10: Three state up-counter showing a temporary state.

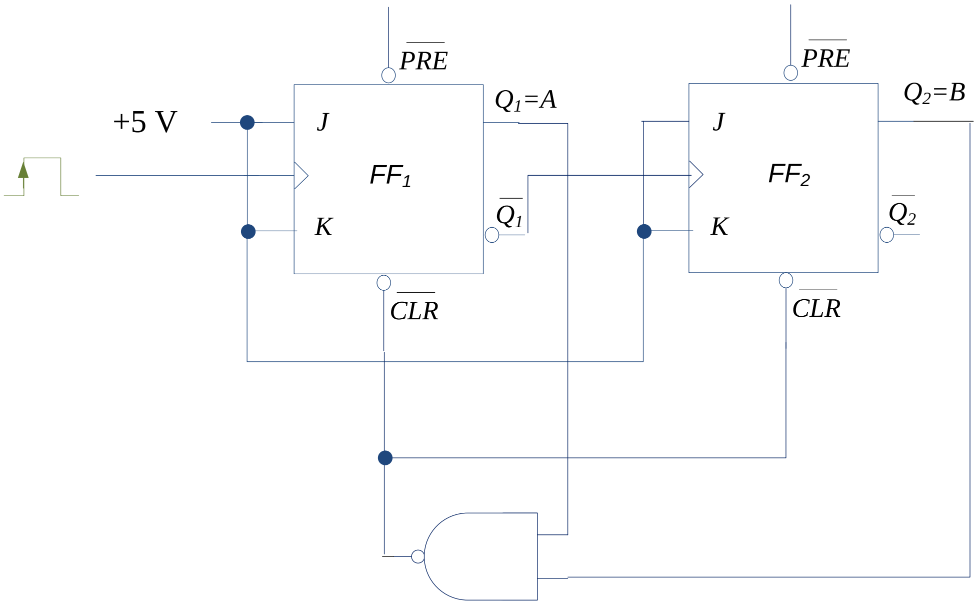


Figure 8.11: Three state asynchronous up-counter with PGT clock pulse (with $\overline{CLR}$ input).

[13] Clear input does not depend on clock edge and hence the change is immediate.

At time t_1, the PGT edge of the clock toggles the first flip-flop output to 1 (i.e. Q_1=1). There is no change in Q_2=0. At time t_2, Q_1 toggles to 0 and Q_2 toggles to 1. At time t_3, Q_2 remains at 1 and Q_1 toggles to 1. However, at this time point, the output of the NAND gate is logic level 0 and hence activates the active low $\overline{CLR}$ inputs, which reset both flip-flops to logic level 0. The counter then resumes its count. The effect of $\overline{CLR}$ is instantaneous and though the timing diagram shows a glitch during time t_3, it occurs only for a very short period of time and does not appear as part of the counter output.

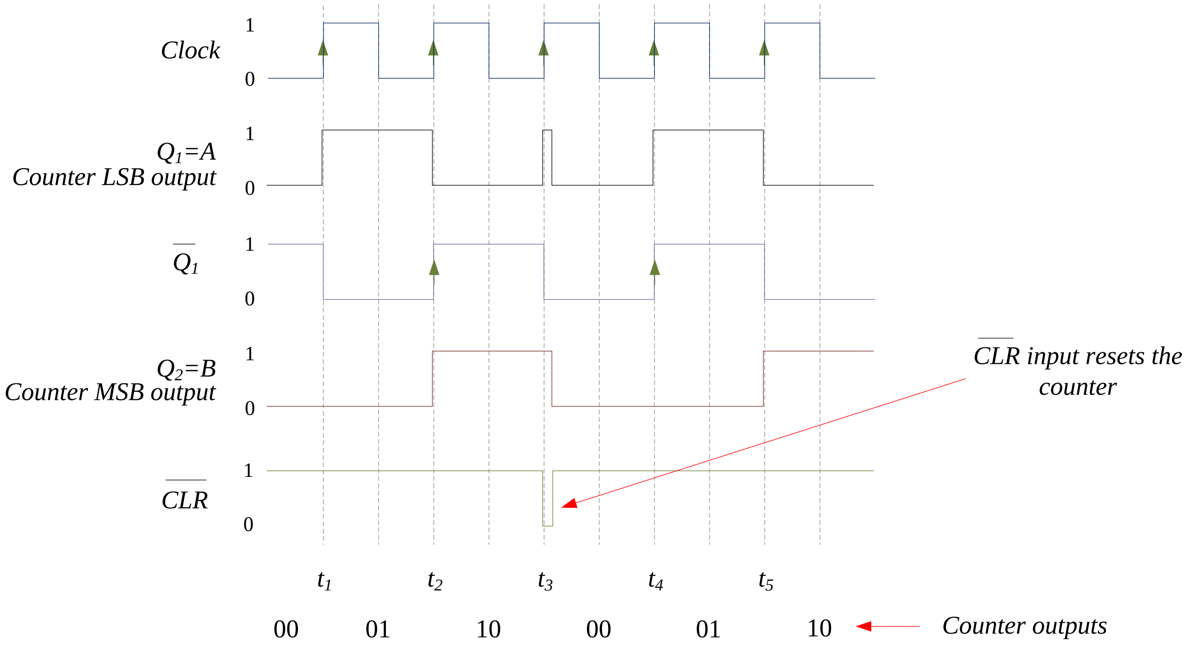


Figure 8.12: Timing diagram for three state asynchronous up-counter with PGT clock pulse.

Let us consider another example: a counter to count 000→001→010→011→100 only. In this situation, we will need three flip-flops and the counter has to stop the cycle at 100 (and skip 101, 110 and 111) and return to 000. In other words, the counter has to reset after 100. The state diagram is shown in Figure 8.13. As mentioned earlier, the temporary state of 101 occurs only for a very short period of time and hence will not appear in the counter cycle. The additional circuitry using NAND gate and $\overline{CLR}$ inputs reset the counter to 000 when the state 101 occurs. The logic circuit is shown in Figure 8.14. As soon as the state Q_3=1 (i.e. C=1) and Q_1=1 (i.e. A=1) occur, the clear inputs reset all the flip-flips to 0 and the counter then resumes its cycle.

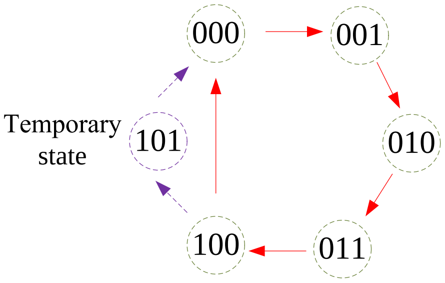


Figure 8.13: Five state up-counter showing a temporary state.

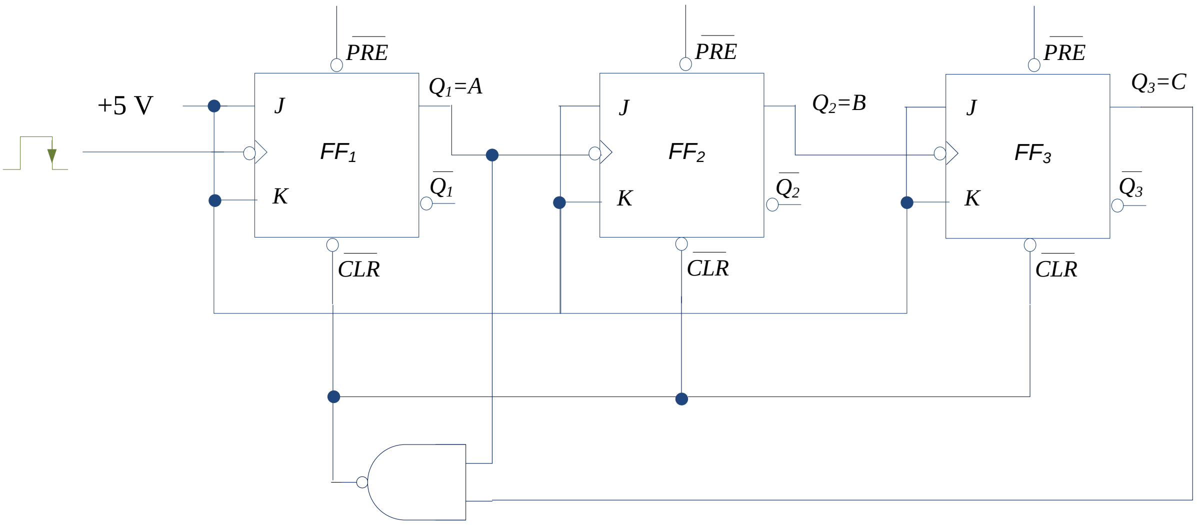


Figure 8.14: Five state asynchronous up-counter with NGT clock pulse (with $\overline{CLR}$ input).

8.4 Synchronous counters

Synchronous counters are advantageous over asynchronous counters as they do not suffer from clock ripple effect due to all flip-flops being clocked at the same time. Also, they allow counter design in any arbitrary sequence. However, synchronous counters often require additional circuitry. In this section, several examples will be used to illustrate the synchronous counter design.

The basic steps in the design are:

1) Obtain the state diagram/table
2) Decide the number of flip-flops and type of flip-flop
3) Derive the state excitation table
4) Obtain the simplified expressions for the flip-flop inputs (for example using K-maps)
5) Draw the logic circuit diagram

8.4.1 Synchronous counter – example 1

Assume that we wish to design a counter that counts 000→010→011→111 and then recycles back to 000. In this counter, there are several unused states: 001, 100, 101 and 110. Though these states should not occur in our design, it is good practice to set the counter to go to 000 if any of these undesired states do occur.

Step 1: State diagram is shown in Figure 8.15.

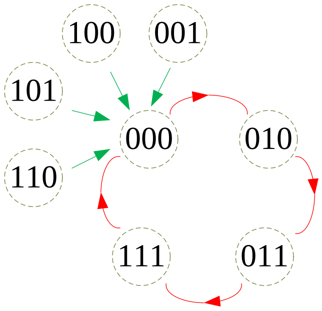


Figure 8.15: State diagram for synchronous counter in example 1.

Step 2: The number of flip-flops is three and let us assume that J-K flip flops are used.

Step 3: The excitation table is basically a truth table that gives the necessary J and K inputs to enable a change in the current output Q to next state Q^+. Table 8.2 shows the general excitation table for a J-K flip-flop (with don't care conditions, X).

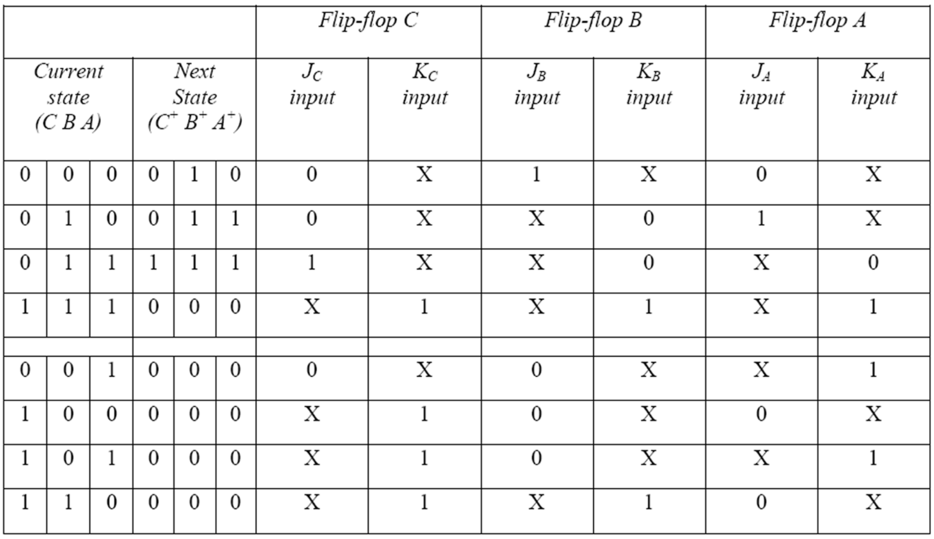
Table 8.2: Excitation table for general J-K flip-flop

J input	*K input*	*Current output, Q*	*Next output, Q^+*
0	X	0	0
1	X	0	1
X	1	1	0
X	0	1	1

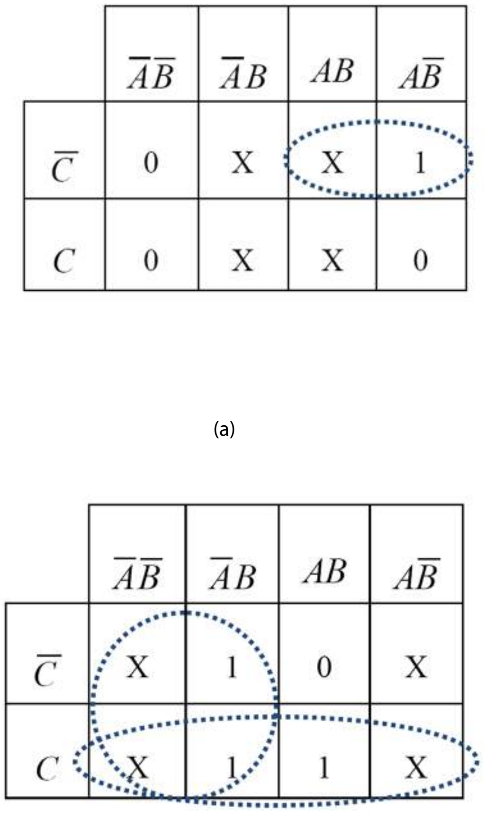
The excitation table for the counter to be designed is shown in Table 8.3.

Step 4: Using the excitation Table 8.3, we can obtain the K-maps for each input as shown in Figures 8.16-8.18 where present states should be used to draw the K-maps.

Table 8.3: Excitation table for the counter in example 1



						Flip-flop C		*Flip-flop B*		*Flip-flop A*	
Current state (C B A)			*Next State (C^+ B^+ A^+)*			*J_C input*	*K_C input*	*J_B input*	*K_B input*	*J_A input*	*K_A input*
0	0	0	0	1	0	0	X	1	X	0	X
0	1	0	0	1	1	0	X	X	0	1	X
0	1	1	1	1	1	1	X	X	0	X	0
1	1	1	0	0	0	X	1	X	1	X	1
0	0	1	0	0	0	0	X	0	X	X	1
1	0	0	0	0	0	X	1	0	X	0	X
1	0	1	0	0	0	X	1	0	X	X	1
1	1	0	0	0	0	X	1	X	1	0	X



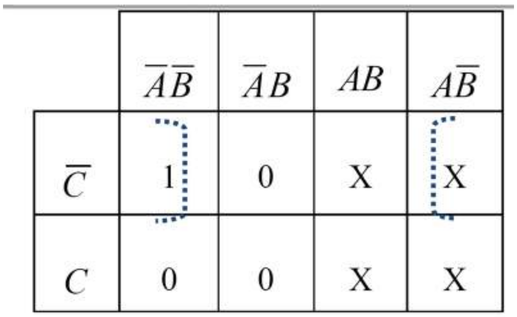
	$\overline{A}\overline{B}$	$\overline{A}B$	AB	$A\overline{B}$
$\overline{C}$	0	X	X	1
C	0	X	X	0

(a)

	$\overline{A}\overline{B}$	$\overline{A}B$	AB	$A\overline{B}$
$\overline{C}$	X	1	0	X
C	X	1	1	X

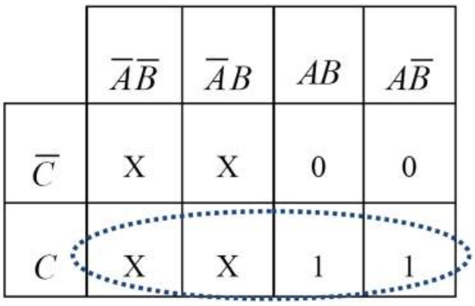
(b)

Figure 8.16: K-maps for inputs (a) J_A and (b) K_A.



	$\overline{A}\overline{B}$	$\overline{A}B$	AB	$A\overline{B}$
$\overline{C}$	1	0	X	X
C	0	0	X	X

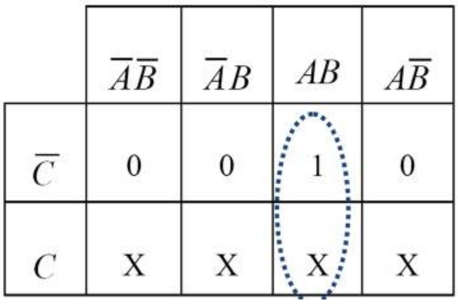
(a)



	$\overline{A}\overline{B}$	$\overline{A}B$	AB	$A\overline{B}$
$\overline{C}$	X	X	0	0
C	X	X	1	1

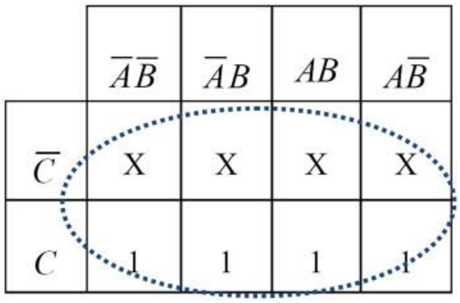
(b)

Figure 8.17: K-maps for inputs (a) J_B and (b) K_B.



	$\overline{A}\overline{B}$	$\overline{A}B$	AB	$A\overline{B}$
$\overline{C}$	0	0	1	0
C	X	X	X	X

(a)



	$\overline{A}\overline{B}$	$\overline{A}B$	AB	$A\overline{B}$
$\overline{C}$	X	X	X	X
C	1	1	1	1

(b)

Figure 8.18: K-maps for inputs (a) J_C and (b) K_C.

From the K-maps, the simplified expressions for the inputs are:

$$J_A = A\overline{C} \qquad K_A = C + \overline{A}$$

$$J_B = \overline{B}\overline{C} \qquad K_B = C$$

$$J_C = AB \qquad K_C = 1$$

Step 5: The logic circuit diagram is given in Figure 8.19. Notice that all the clock inputs are tied together and hence the flip-flops are clocked simultaneously.

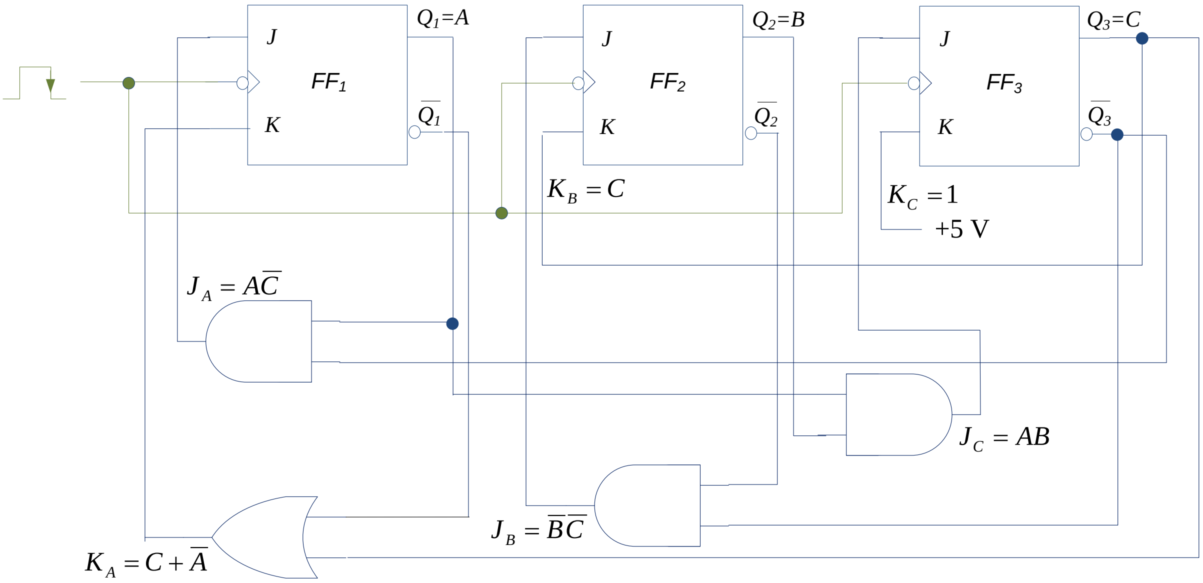


Figure 8.19: Logic circuit diagram for counter in example 1.

8.4.2 Synchronous counter – example 2

Now consider another example using T flip-flops and the counter to be designed cycles through 000→010→100→110 and then resets to 000.

Step 1: Since the LSB of the counter does not change, we need not be concerned about the design for this bit and can set Q_A=0. So, the simplified state diagram is as shown in Figure 8.20.

Step 2: The number of flip-flops is two only and T flip-flops will be used.

Step 3: The general excitation table for T flip-flop is given in Table 8.4, while the excitation table for the counter is given in Table 8.5.

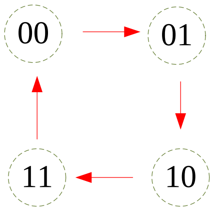


Figure 8.20: Simplified state diagram for example 2.

Table 8.4: General excitation table for T flip-flop

T input	*Current* *output, Q*	*Next* *output, Q^+*
0	0	0
1	0	1
1	1	0
0	1	1

Table 8.5: Excitation table for the counter in example 2

Current *state* *(C B)*		*Next* *State* *(C^+ B^+)*		*Flip-flop C* *T_C*	*Flip-flop B* *T_B*
0	0	0	1	0	1
0	1	1	0	1	1
1	0	1	1	0	1
1	1	0	0	1	1

Step 4: The K-maps are shown in Figure 8.21. The simplified expressions are

$$T_B = 1$$

$$T_C = B$$

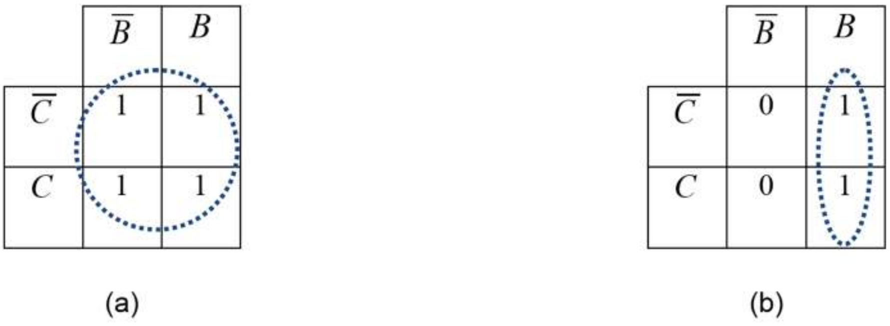


Figure 8.21: K-maps for example 2: (a) T_B (b) T_C.

Step 5: The logic circuit diagram is shown in Figure 8.22.

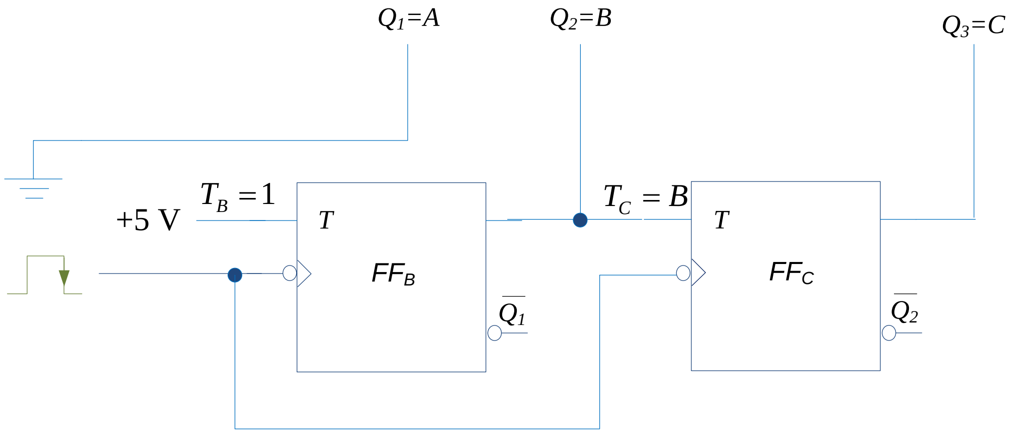


Figure 8.22: Logic circuit diagram for example 2.