Counter with Unused states:

We can construct a Sequential Circuit (counter) that uses fewer than the maximum possible number of states.

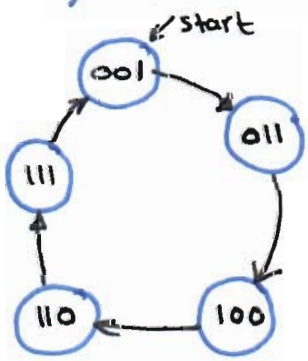
States that are not used in specifying the sequential Circuit are not listed in the state table.

In simplifying the input equations, the unused states may be treated as don't care conditions or may be assigned specific next states.

Example 1:

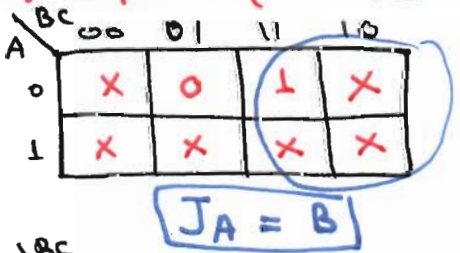
Design a synchronous counter which counts as the following: 1, 3, 4, 6, 7, 1, ... using JK flip-flop (consider missed (unused) states as don't care Condition)

a) State diagram: (b) - State table & excitation table:

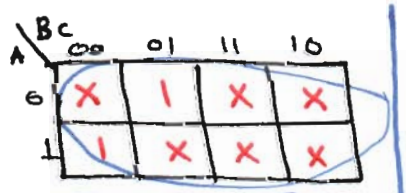


Present state			Next state			Input equations					
A	B	C	A	B	C	J_A	K_A	J_B	K_B	J_C	K_C
0	0	1	0	1	1	0	X	1	X	X	0
0	1	1	1	0	0	1	X	X	1	X	1
1	0	0	1	1	0	X	0	1	X	0	X
1	1	0	1	1	1	X	0	X	0	1	X
1	1	1	0	0	1	X	1	X	1	X	0

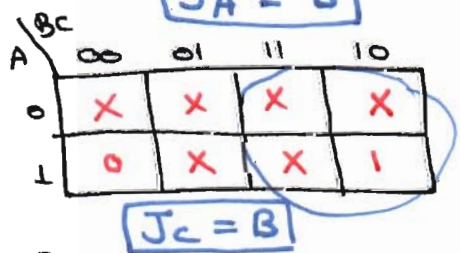
c) Input equations:



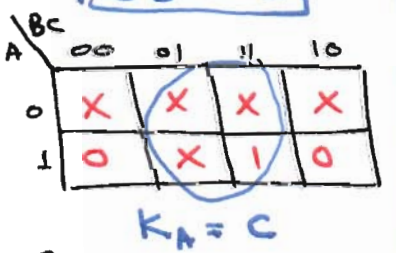
$J_A = B$



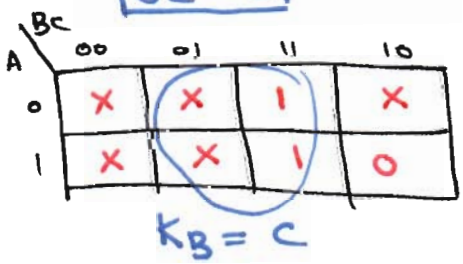
$J_B = 1$



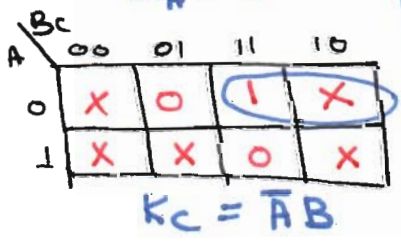
$J_C = B$



$K_A = C$

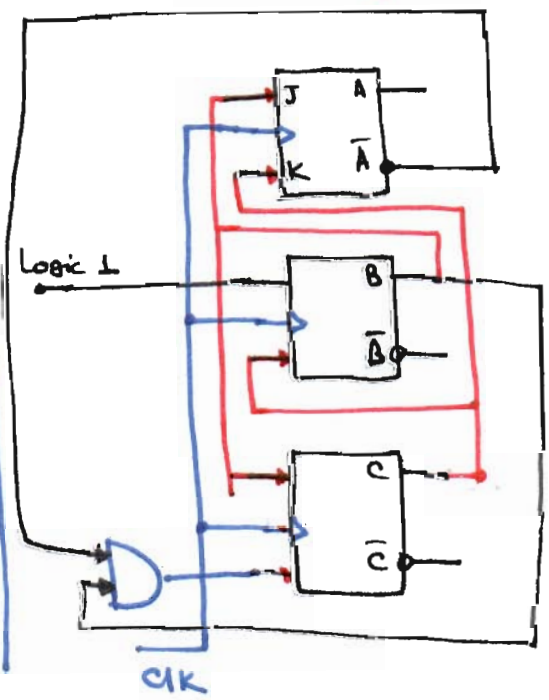


$K_B = C$



$K_C = \bar{A}B$

d) Logic diagram.



Counter with Control input :

Example 2 :

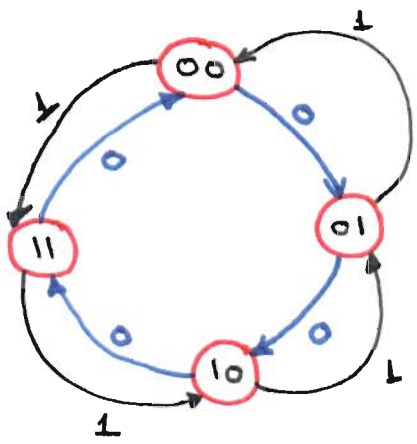
Design a synchronous counter with control input X which counts as the following :

if X = 0 → 0, 1, 2, 3, 0

if X = 1 → 0, 3, 2, 1, 0

Use D Flip-flops in your design.

a) State diagram :



b) State table :

Present state A	B	input X	Next state A	B
0	0	0	0	1
0	1	0	1	0
1	0	0	1	1
1	1	0	0	0
0	0	1	1	1
0	1	1	1	0
1	0	1	0	1
0	0	1	0	0

Because all states are exist, so we can rewrite the state table like the next :

Present state A	B	Input X	Next state A	B
0	0	0	0	1
0	0	1	1	1
0	1	0	1	0
0	1	1	0	0
1	0	0	1	1
1	0	1	0	1
1	1	0	0	0
1	1	1	1	0

⇒

c) Input equations :

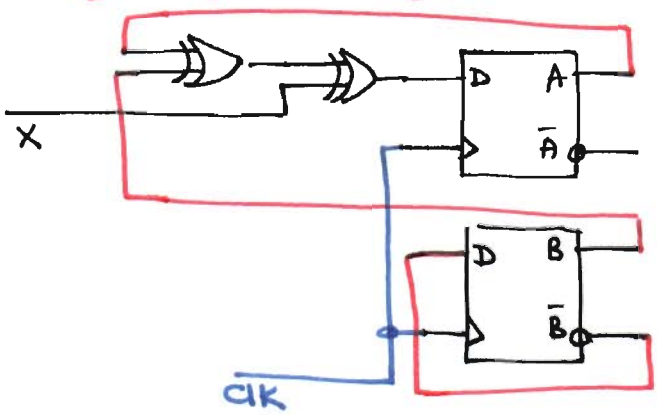
A \ BX	00	01	11	10
0	0	1	0	1
1	1	0	1	0

$A(t+1) = D_A = A \oplus B \oplus X$

A \ BX	00	01	11	10
0	1	1	0	0
1	1	1	0	0

$B(t+1) = D_B = \overline{B}$

d) Logic diagram :



Example 3:

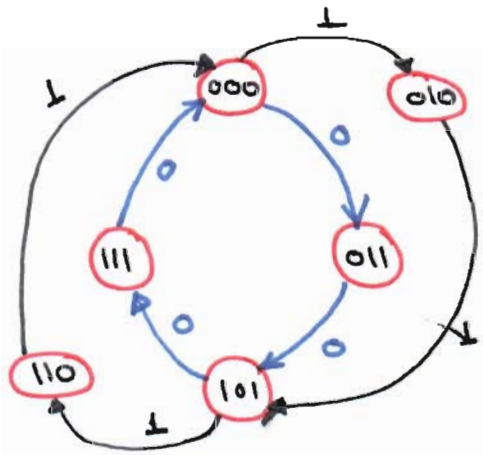
Design a synchronous counter with control input X which counts as the following:

if $X=0 \Rightarrow 0, 3, 5, 7, 0$

$X=1 \Rightarrow 0, 2, 5, 6, 0$

Use D Flip-flops in your design (consider the unused states as don't care condition).

a) State diagram:

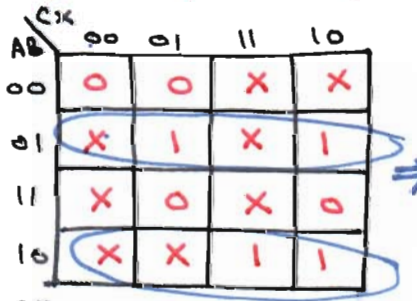


State 001 & 100 not exist.

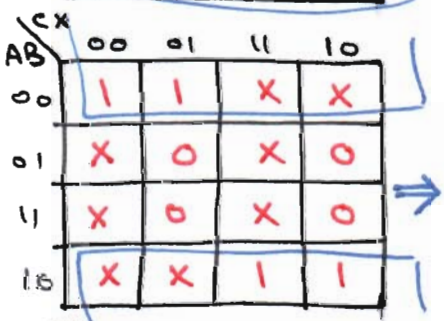
b) State & excitation table.

Present state A	B	C	Input X	Next State A	B	C
0	0	0	0	0	1	1
0	1	1	0	1	0	1
1	0	1	0	1	1	1
1	1	1	0	0	0	0
0	0	0	1	0	1	0
0	1	0	1	1	0	1
1	0	1	1	1	1	0
1	1	0	1	0	0	0

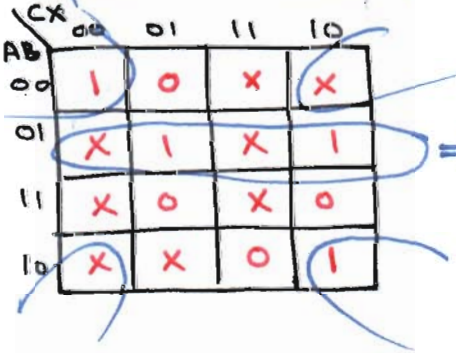
c) input equations:



$A(t+1) = \bar{A}B + A\bar{B} = A \oplus B$ (= D_A)

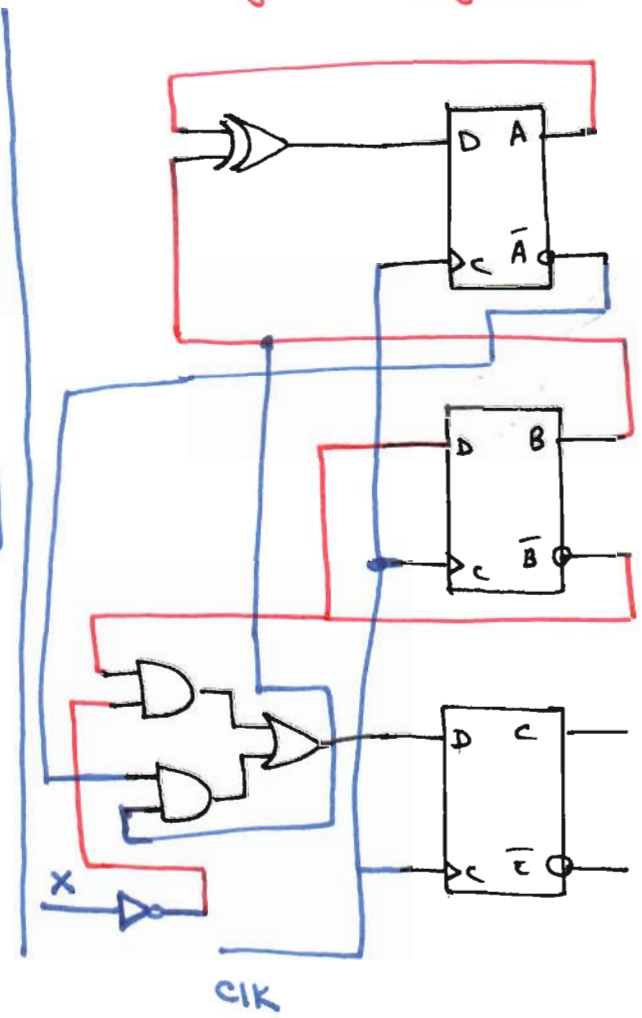


$B(t+1) = D_B = \bar{B}$



$C(t+1) = D_C = \bar{B}\bar{X} + \bar{A}B$

d- Logic diagram:

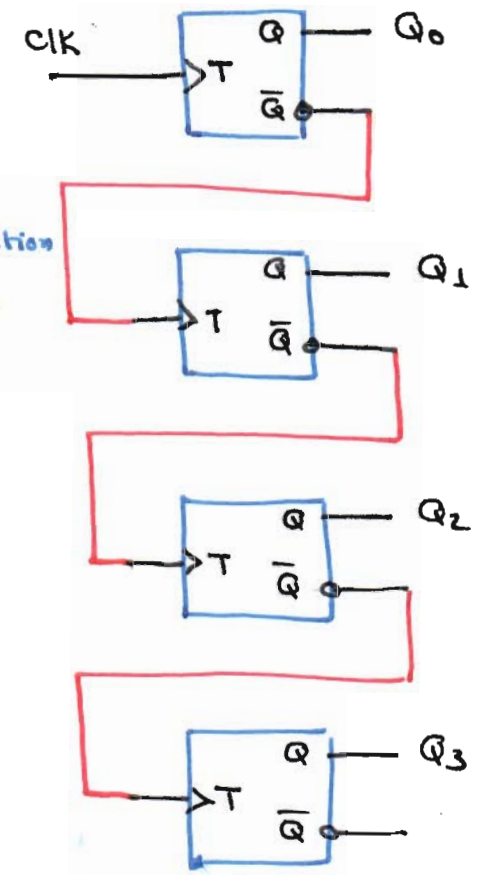


Asynchronous Counters (Ripple Counters)

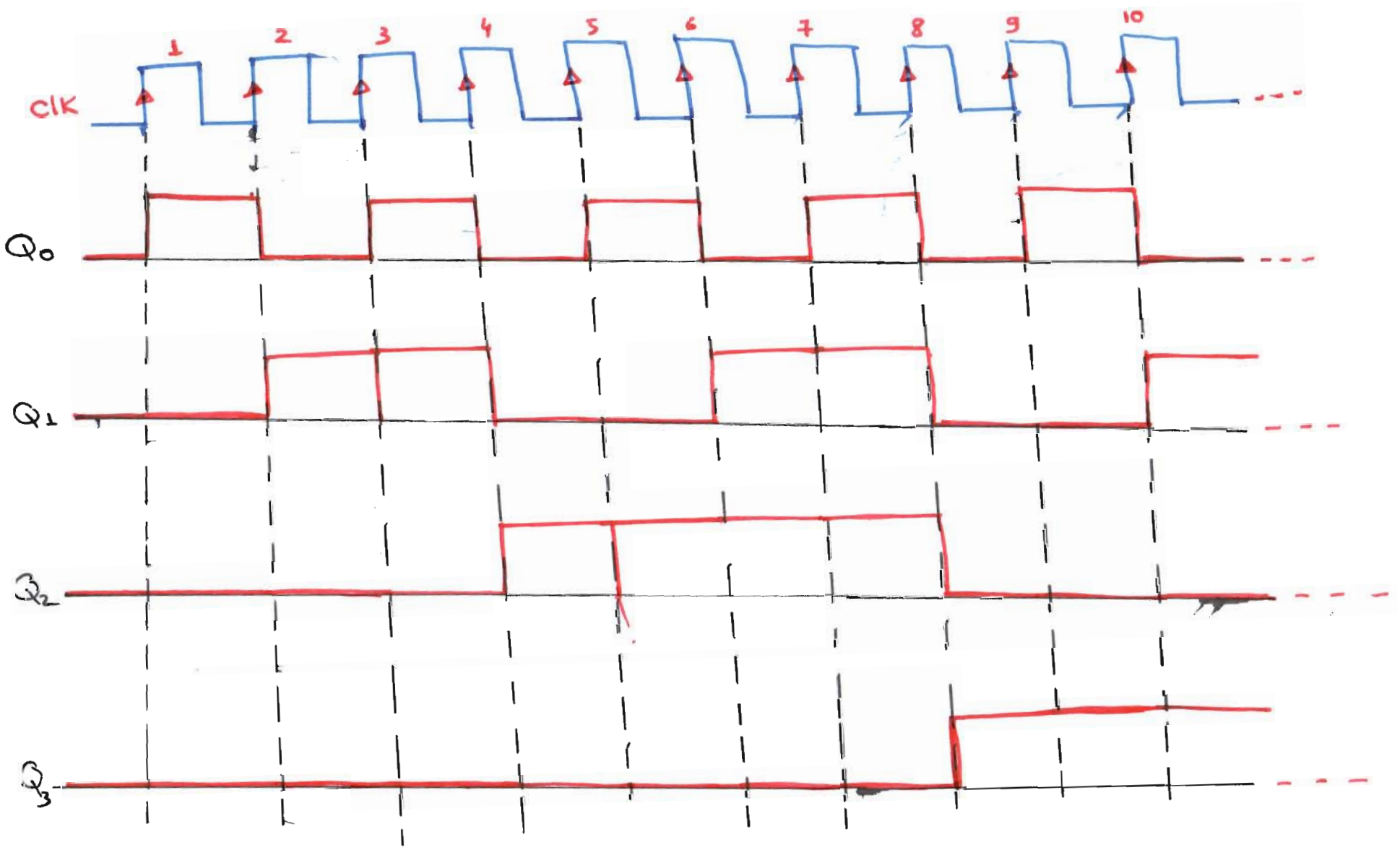
An n-bit binary counter can be constructed with just n flip-flops and no other components.

T flip-flop changes state on every rising edge of its clock input.

The counter is called a ripple counter because the carry information ripples from the less significant bits to more significant bits, one bit at a time.



4-bit binary ripple Counter



Timing diagram for 4-bit binary counter.