EE 224: INTRODUCTION TO DIGITAL CIRCUITS & COMPUTER DESIGN

Lecture 6: Sequential Logic – 3 Registers & Counters

05/10/2010

Avinash Kodi, kodi@ohio.edu

Introduction

- A Flip-Flop stores one bit of information – when a set of n Flip Flops is used to store n bits of data, we refer to them as REGISTER
- Common usage includes
  - Holding an output value from an arithmetic circuit (data registers)
  - Holding a count value in a counter circuit (Program Counter)
  - Keeping addresses for memory locations (Index Registers)
  - Many more ….
- A common clock signal is typically used for each FF in a register

Data Transfer

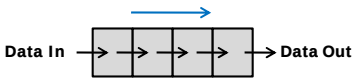
3

- Data transfer is a common function in computer systems
  - Parallel transfer: n-bits (full register contents) at a time
  - Serial transfer: 1-bit at a time (bit-by-bit)
- It is possible to convert transfer types in special registers:
  - Parallel-to-serial: load data into a register in one clock cycle (in parallel) but send out in series in n clock cycles
  - Serial-to-parallel: receive data in n clock cycles but can be accessed to all contents in parallel (n-bits)

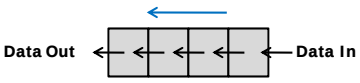
Data Transfer

4

Serial Registers

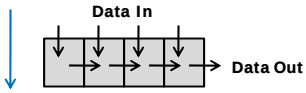


Serial In-Shift Right-Serial Out

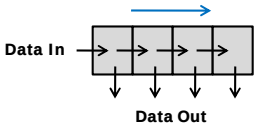


Serial In-Shift Left-Serial Out

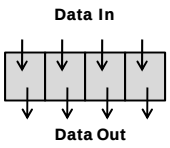
Parallel Registers



Parallel In-Serial Out

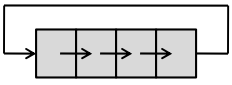


Serial In-Parallel Out

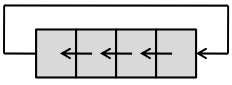


Parallel In-Parallel Out

Rotate Registers



Rotate Right

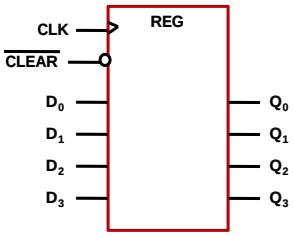


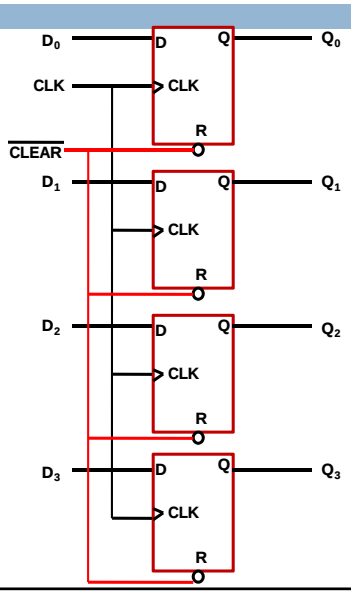
Rotate Left

Registers with Parallel Loading

5

- The simplest register consists of only FFs without external gates
  - The transfer of data is referred to as loading the data
  - Parallel loading if all the bits are loaded simultaneously with a common clock pulse

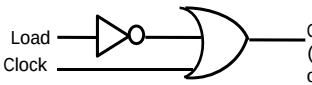


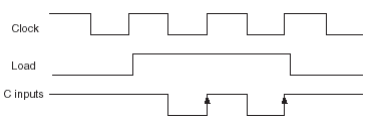


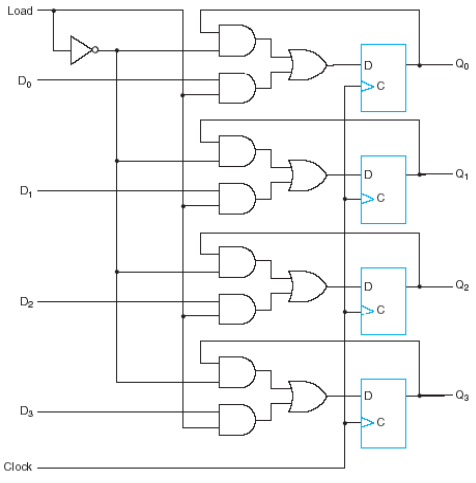
Registers with Parallel Loading

6

- The master clock generator supplies a continuous train of clock pulses. When the register contents are to remain same the clock must be prevented from reaching the clock input. Thus, a separate control signal is used to activate register loading.





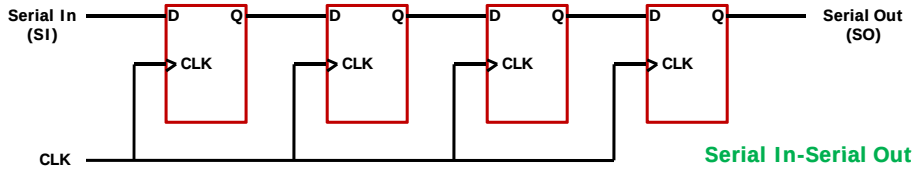


A 4-bit register with a control load input

Shift Registers

7

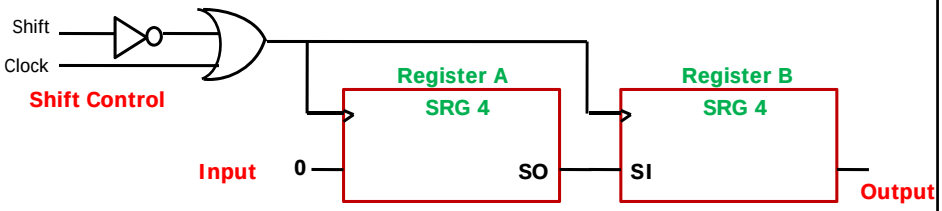
- Capable of shifting its storage bits laterally in one or both directions. They consist of a chain of flip-flops in cascade, with the output of one FF connected to input of the next FF. All FF receive a common clock pulse, which activates the shift from each stage to the next.

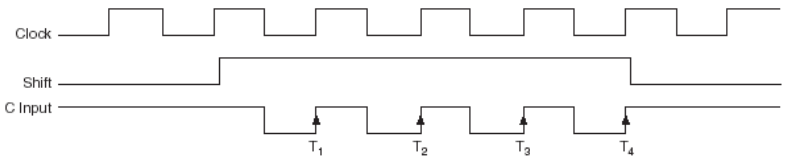


Serial Data Transfer

8

- A digital system is said to operate in a serial mode when information in the system in transferring one bit at a time

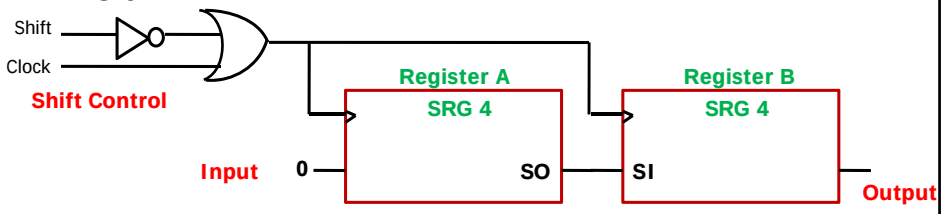




Example

9

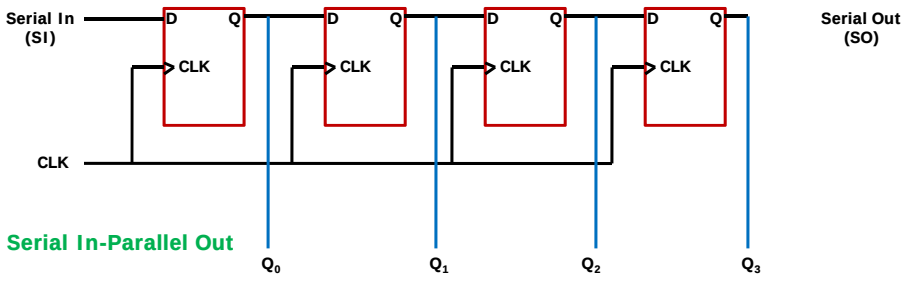
- Assume that binary content of registers A before the shift is 1011 and that of B is 0010 and SI of register A is 0



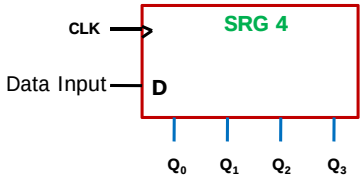
Timing pulse	Shift Register A				Shift Register B			
Initial value	1	0	1	1	0	0	1	0
After T_1	0	1	0	1	1	0	0	1
After T_2	0	0	1	0	1	1	0	0
After T_3	0	0	0	1	0	1	1	0
After T_4	0	0	0	0	1	0	1	1

Registers with Serial Loading

10

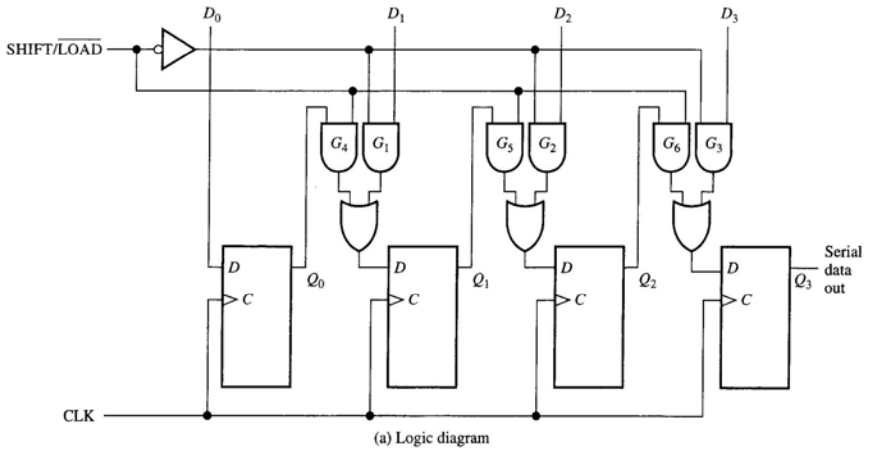


A 4-bit register



Registers with Parallel Loading

11



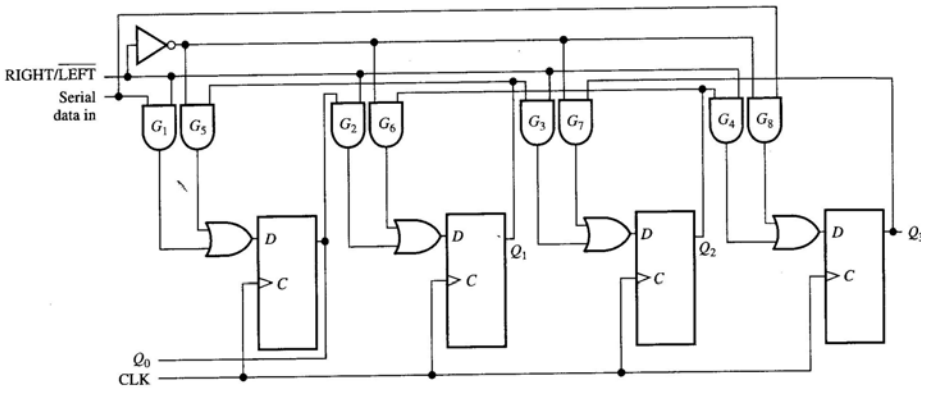
(a) Logic diagram

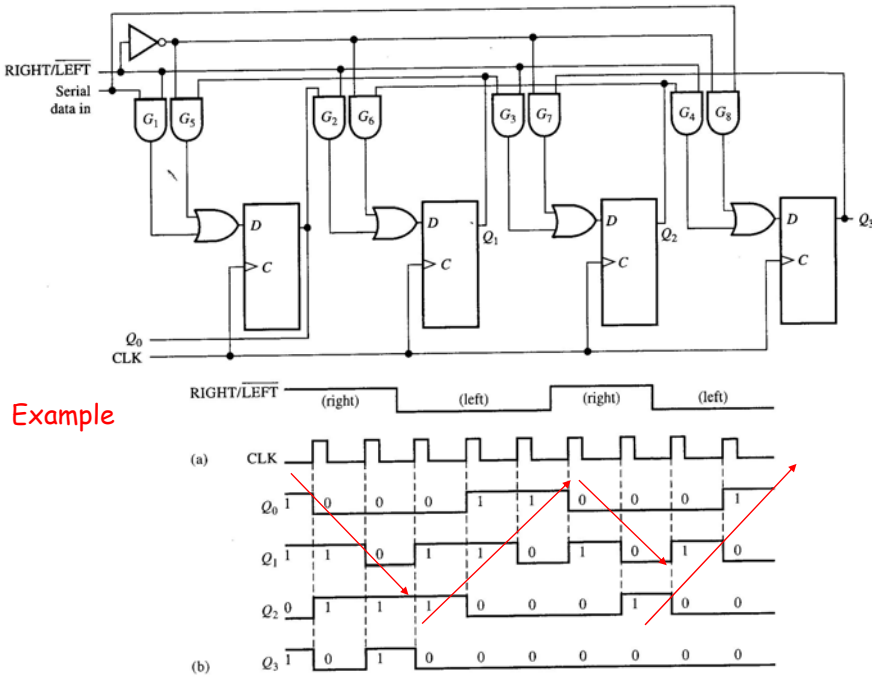
Parallel In – Serial Out
Shift Register with Shift/Load Control

Bidirectional Shift Registers

12

The data can be shifted either left or right. It can be implemented by using gating logic that enables the transfer direction of a data bit from one stage to the next stage (to the right or to the left), depending on the level of control signal (R/L').





Flip Flop Summary

14

Flip-Flop Characteristic Table

JK Flip-Flop				SR Flip-Flop			
J	K	Q^+	Operation	S	R	Q^+	Operation
0	0	Q	No Change	0	0	Q	No Change
0	1	0	Reset	0	1	0	Reset
1	0	1	Set	1	0	1	Set
1	1	Q'	Complement	1	1	?	Undefined

D Flip-Flop			T Flip-Flop		
D	Q^+	Operation	T	Q^+	Operation
0	0	Reset	0	Q	No Change
1	1	Set	1	Q'	Complement

FF Transition Input Codes

15

- Using FF state tables we can derive the input conditions that will cause specific transitions in each FF output

- 4 possible transitions:
  - D & T FFs have well-defined input conditions in each case
  - J-K FF has 'Don't Care' (X) conditions

- Key to follow sequential logic circuit design examples (i.e. counters)

FF Outputs		FF Inputs		
Q(t)	**Q(t+1)**	**T**	**D**	**J-K**
0 →	0	0	0	0-X
0 →	1	1	1	1-X
1 →	0	1	0	X-1
1 →	1	0	1	X-0

FF Excitation Table

16

Flip-Flop Excitation Table

JK Flip-Flop

Q	Q^+	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

SR Flip-Flop

Q	Q^+	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

D Flip-Flop

Q	Q^+	D
0	0	0
0	1	1
1	0	0
1	1	1

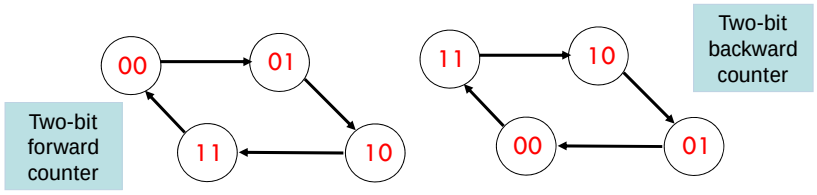
T Flip-Flop

Q	Q^+	T
0	0	0
0	1	1
1	0	1
1	1	0

Counters

17

- A register that goes through a prescribed sequence of states upon the application of input pulses.
  - The input pulses may be clock pulses or other sources, occurring at fixed or random intervals.
- An n-bit binary counter consists of n FF and can count in binary from 0 through 2^n-1. Counters can be of two types:
  - Ripple counters: the FF output transition serves as a source for triggering other FF.
  - Synchronous counter: the clock triggers all of the FFs
- Binary counters are most efficiently constructed with complementing T or JK FFs. They also can be designed with D FFs.



Synchronous Counters: Example 1

18

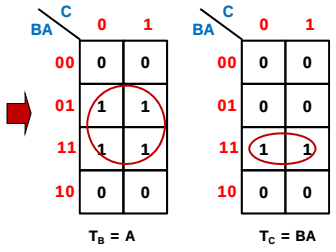
- The binary counters are constructed from T FFs
- All FFs are initially set to 0
- The sequence of states are:

000, 001, 010, 011, 100, 101, 110, 111

Present State C B A	Next State C^+ B^+ A^+	FF Inputs T_C T_B T_A
0 0 0	0 0 1	0 0 1
0 0 1	0 1 0	0 1 1
0 1 0	0 1 1	0 0 1
0 1 1	1 0 0	1 1 1
1 0 0	1 0 1	0 0 1
1 0 1	1 1 0	0 1 1
1 1 0	1 1 1	0 0 1
1 1 1	0 0 0	1 1 1

T Flip-Flop

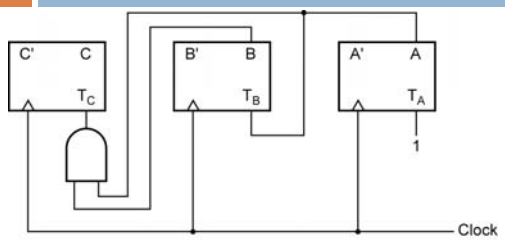
Q	Q^+	T
0	0	0
0	1	1
1	0	1
1	1	0



T_A= 1
T_B= A
T_C= BA

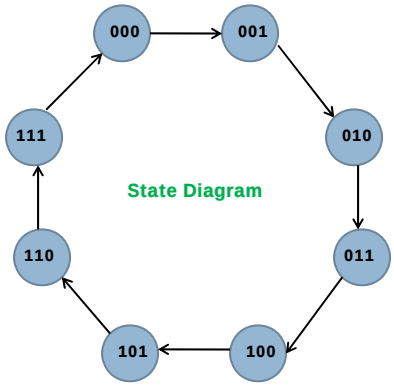
Synchronous Counters: Example 1

19



C B A	C^+ B^+ A^+	T_C T_B T_A
0 0 0	0 0 1	0 0 1
0 0 1	0 1 0	0 1 1
0 1 0	0 1 1	0 0 1
0 1 1	1 0 0	1 1 1
1 0 0	1 0 1	0 0 1
1 0 1	1 1 0	0 1 1
1 1 0	1 1 1	0 0 1
1 1 1	0 0 0	1 1 1

State Table



Note that there is NO external input variable in this circuit (besides Clk signal) and FF outputs determine counter output

Synchronous Counter: Example 2

20

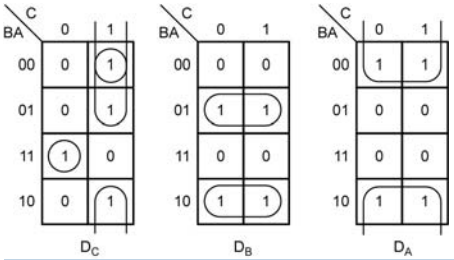
- Use D FFs instead of T FF

000, 001, 010, 011, 100, 101, 110, 111

Present State C B A	Next State C^+ B^+ A^+	FF Inputs D_C D_B D_A
0 0 0	0 0 1	0 0 1
0 0 1	0 1 0	0 1 0
0 1 0	0 1 1	0 1 1
0 1 1	1 0 0	1 0 0
1 0 0	1 0 1	1 0 1
1 0 1	1 1 0	1 1 0
1 1 0	1 1 1	1 1 1
1 1 1	0 0 0	0 0 0

D Flip-Flop

Q	Q^+	D
0	0	0
0	1	1
1	0	0
1	1	1



$D_A = A^+ = A'$
$D_B = B^+ = BA' + B'A$
$D_C = C^+ = C'BA + CB' + CA'$
$= C'BA + C(B' + A') = C'BA + C(BA)'$

Synchronous Counters: Example 2

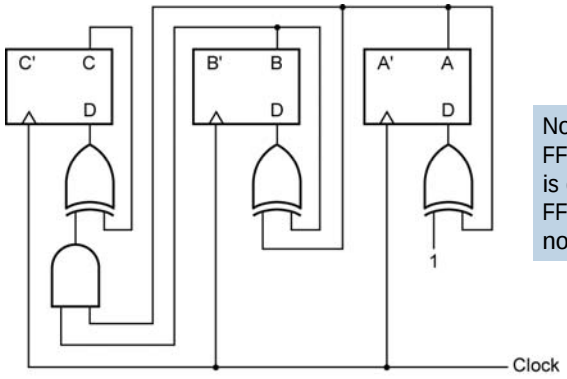
21

$$D_A = A^+ = A'$$

$$D_B = B^+ = BA' + B'A = B \oplus A$$

$$D_C = C^+ = C'BA + CB' + CA' = C'BA + C(BA)' = C \oplus BA$$

Synchronous up-counter



Note that XOR of D FF output and input is equivalent to T FF (see previous notes on FF)

Example 3: Up and Down Counters

22

Up-down binary counter with D FFs, with U & D select bits for direction. Based on earlier examples, consider the modification:

$$D_A = A^+ = A \oplus (U + D)$$

$$D_B = B^+ = B \oplus (UA + DA')$$

$$D_C = C^+ = C \oplus (UBA + DB'A')$$

When U=0 and D=1 down-counter,
(instead of A & BA we get A' & B'A')

$$D_A = A^+ = A \oplus 1 = A'$$

$$D_B = B^+ = B \oplus A'$$

$$D_C = C^+ = C \oplus B'A'$$

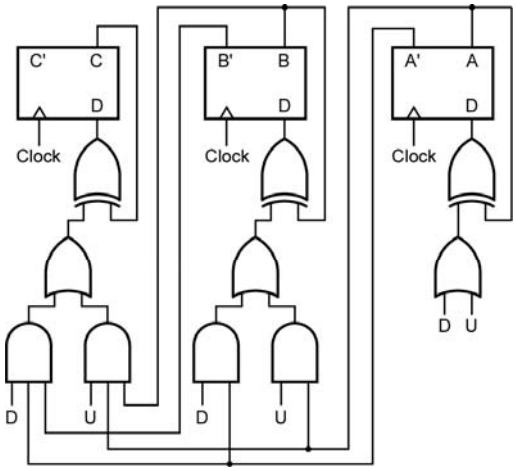
CBA	$C^+B^+A^+$	
	U=1	D=1
000	001	111
001	010	000
010	011	001
011	100	010
100	101	011
101	110	100
110	111	101
111	000	110

When U=1 and D=0 up-counter. See earlier
When U=0 and D=0 no change. ABC preserved
When U=1 and D=1 is not to be used

Example 3: Up and Down Counters

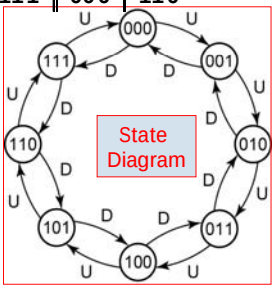
23

Logic Circuit



CBA	$C^+B^+A^+$	
	U=1	D=1
000	001	111
001	010	000
010	011	001
011	100	010
100	101	011
101	110	100
110	111	101
111	000	110

State Table



Example 4: Synchronous Counters

24

- Design a 4-bit synchronous counter using JK FF

JK Flip-Flop

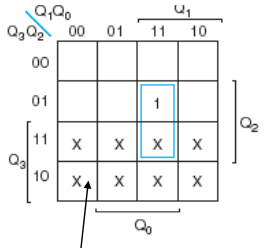
Q	Q^+	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

Example 4: Synchronous Counters

25

State Table and Flip-Flop Inputs for Binary Counter

Present state				Next state				Flip-flop inputs							
Q_3	Q_2	Q_1	Q_0	Q_3	Q_2	Q_1	Q_0	J_{Q3}	K_{Q3}	J_{Q2}	K_{Q2}	J_{Q1}	K_{Q1}	J_{Q0}	K_{Q0}
0	0	0	0	0	0	0	1	0	×	0	×	0	×	1	×
0	0	0	1	0	0	1	0	0	×	0	×	1	×	×	1
0	0	1	0	0	0	1	1	0	×	0	×	×	0	1	×
0	0	1	1	0	1	0	0	0	×	1	×	×	1	×	1
0	1	0	0	0	1	0	1	0	×	×	0	0	×	1	×
0	1	0	1	0	1	1	0	0	×	×	0	1	×	×	1
0	1	1	0	0	1	1	1	0	×	×	0	×	0	1	×
0	1	1	1	1	0	0	0	1	×	×	1	×	1	×	1
1	0	0	0	1	0	0	1	×	0	0	×	0	×	1	×
1	0	0	1	1	0	1	0	×	0	0	×	1	×	×	1
1	0	1	0	1	0	1	1	×	0	0	×	×	0	1	×
1	0	1	1	1	1	0	0	×	0	1	×	×	1	×	1
1	1	0	0	1	1	0	1	×	0	×	0	0	×	1	×
1	1	0	1	1	1	1	0	×	0	×	0	1	×	×	1
1	1	1	0	1	1	1	1	×	0	×	0	×	0	1	×
1	1	1	1	0	0	0	0	×	1	×	1	×	1	×	1



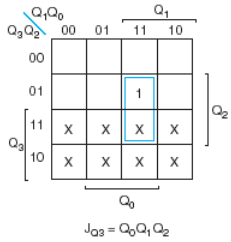
Don't care states

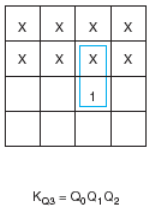
It may be necessary to control the operation of the counter with a count-enable input.

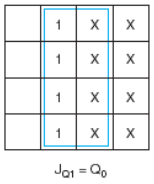
- When EN=0, all J and K inputs are equal to 0 and FFs remain in the same state.

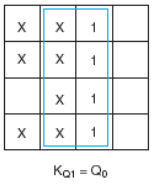
Example 4: Synchronous Counters

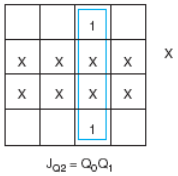
26

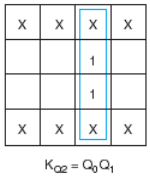












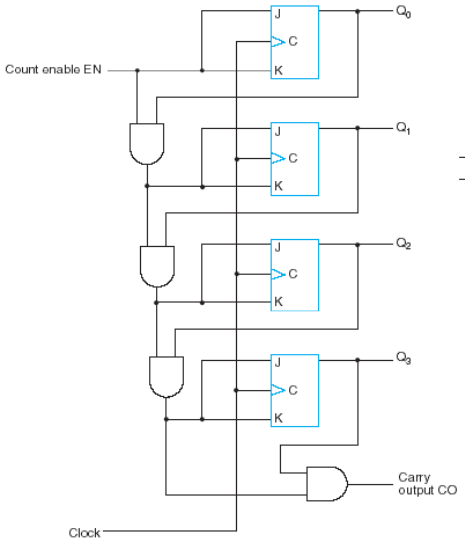
$$J_{Q0} = K_{Q0} = EN$$
$$J_{Q1} = K_{Q1} = Q_0EN$$
$$J_{Q2} = K_{Q2} = Q_0Q_1EN$$
$$J_{Q3} = K_{Q3} = Q_0Q_1Q_2EN$$

The FF in the least significant position of a synchronous binary counter is complemented with a clock pulse transition.

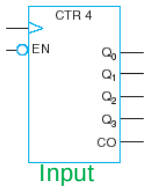
Example 4: Synchronous Counters

27

Logic Circuit



Symbol



Input Equations

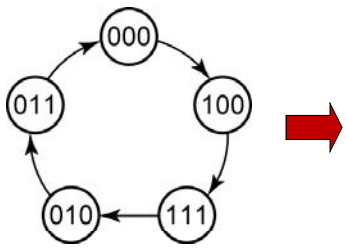
$$J_{Q0} = K_{Q0} = EN$$
$$J_{Q1} = K_{Q1} = Q_0 EN$$
$$J_{Q2} = K_{Q2} = Q_0 Q_1 EN$$
$$J_{Q3} = K_{Q3} = Q_0 Q_1 Q_2 EN$$

Example 5: Arbitrary Counters with T FF

28

Counters for other sequences with T FFs.

Arbitrary Count Sequence.



If these numbers represent a given pattern (say on a screen), then such arbitrary counting sequences may be useful for 'display' purposes

Present State			Next State			FF Inputs		
C	B	A	C^+	B^+	A^+	T_C	T_B	T_A
0	0	0	1	0	0	1	0	0
0	0	1	-	-	-	X	X	X
0	1	0	0	1	1	0	0	1
0	1	1	0	0	0	0	1	1
1	0	0	1	1	1	0	1	1
1	0	1	-	-	-	X	X	X
1	1	0	-	-	-	X	X	X
1	1	1	0	1	0	1	0	1

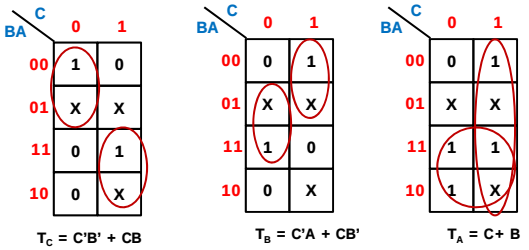
Don't care states
Not completely specified

Example 5: Arbitrary Counters with T FF

29

T Flip-Flop

Q	Q^+	T
0	0	0
0	1	1
1	0	1
1	1	0

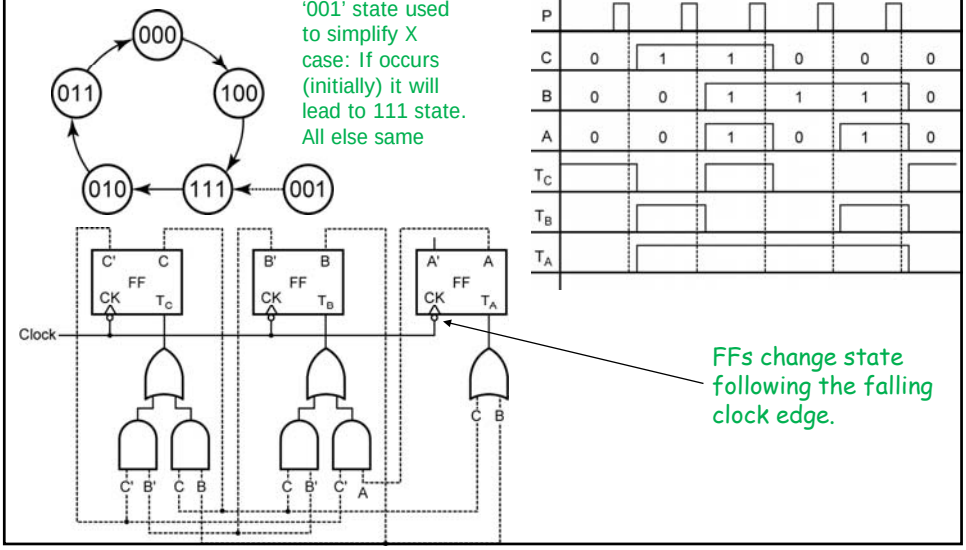


State Table

C	B	A	C^+	B^+	A^+	T_C	T_B	T_A
0	0	0	1	0	0	1	0	0
0	0	1	-	-	-	X	X	X
0	1	0	0	1	1	0	0	1
0	1	1	0	0	0	0	1	1
1	0	0	1	1	1	0	1	1
1	0	1	-	-	-	X	X	X
1	1	0	-	-	-	X	X	X
1	1	1	0	1	0	1	0	1

Example 5: Arbitrary Counters with T FF

30



Example 6: Arbitrary Counters with JK FF

31

State Table and Flip-Flop Inputs for Counter

Present State			Next State			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	×	0	×	1	×
0	0	1	0	1	0	0	×	1	×	×	1
0	1	0	1	0	0	1	×	×	1	0	×
1	0	0	1	0	1	×	0	0	×	1	×
1	0	1	1	1	0	×	0	1	×	×	1
1	1	0	0	0	0	×	1	×	1	0	×

Input Equations

$$J_A = B \qquad K_A = B$$
$$J_B = C \qquad K_B = 1$$
$$J_C = \overline{B} \qquad K_C = 1$$

State Diagram