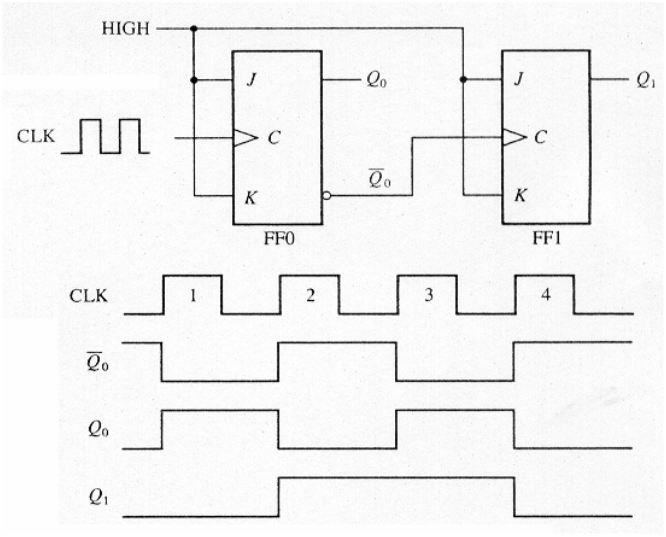
Asynchronous Counters

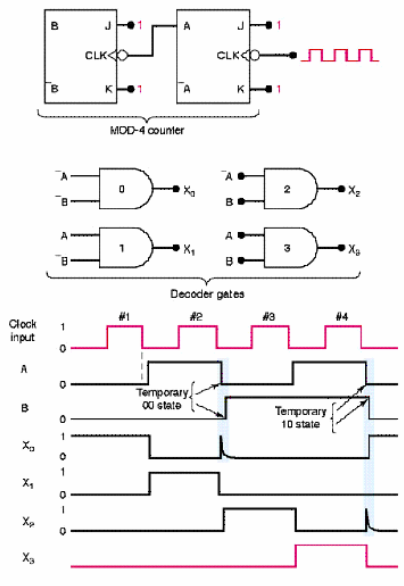
ENGI 251
ELEC 241

Asynchronous Counters

- The term **Asynchronous** refers to events that do not occur at the same time
- With respect to counter operation, asynchronous means that the Flip-Flops within the counter are not connected in a way to cause all Flip-Flops states at exactly the same time
  - they are wired in a way that links the clock of the next flip-flop to the Q of the current device
  - this causes the output count states to ripple through the counter

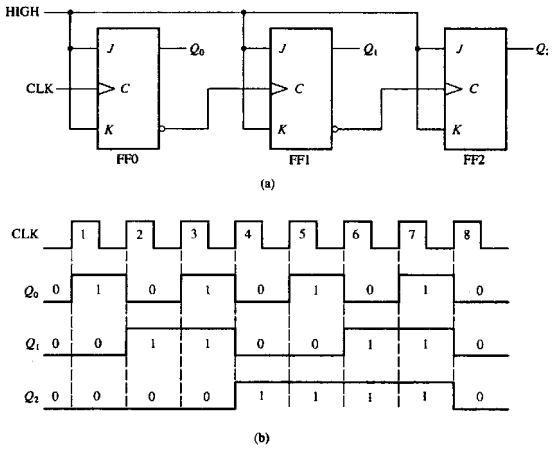
2-Bit (MOD 4) Asynchronous Counter



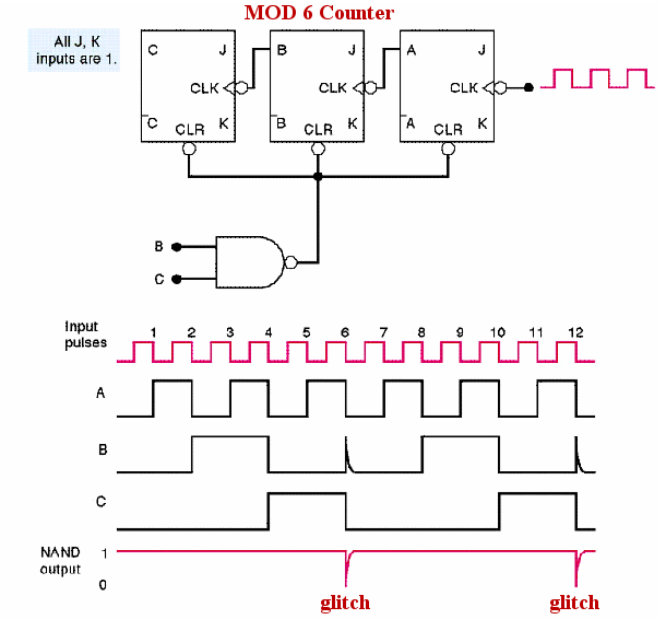


Counter with Glitches

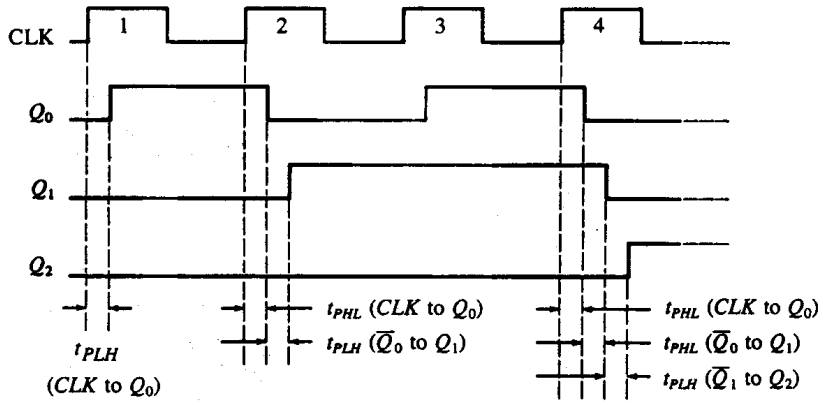
3-Bit (MOD 8) Asynchronous Counter



MOD 6 Counter



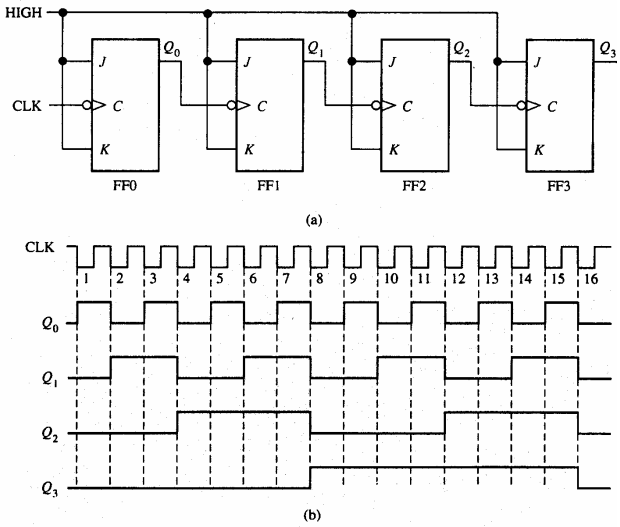
Propagation Delay in Ripple Clocked Binary Counters



4-Bit Counter State Table

CLK	QD	QC	QB	QA
↓	0	0	0	0
↓	0	0	0	1
↓	0	0	1	0
↓	0	0	1	1
↓	0	1	0	0
↓	0	1	0	1
↓	0	1	1	0
↓	0	1	1	1
↓	1	0	0	0
↓	1	0	0	1
↓	1	0	1	0
↓	1	0	1	1
↓	1	1	0	0
↓	1	1	0	1
↓	1	1	1	0
↺	1	1	1	1

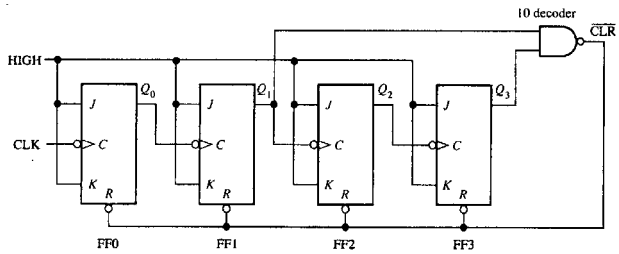
4-Bit Asynchronous Counter

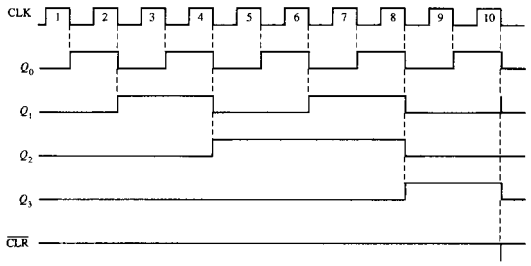


MOD 10 Counter State Table

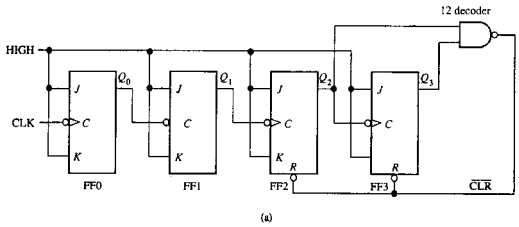
CLK	QD	QC	QB	QA
↓	0	0	0	0
↓	0	0	0	1
↓	0	0	1	0
↓	0	0	1	1
↓	0	1	0	0
↓	0	1	0	1
↓	0	1	1	0
↓	0	1	1	1
↓	1	0	0	0
↓	1	0	0	1
↺	1	0	1	0
	1	0	1	1
	1	1	0	0
	1	1	0	1
	1	1	1	0
	1	1	1	1

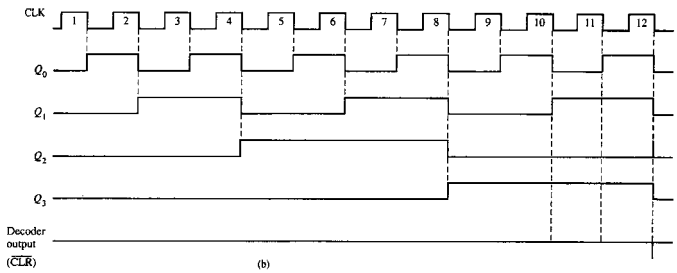
Asynchronous Decade (MOD 10) Counter

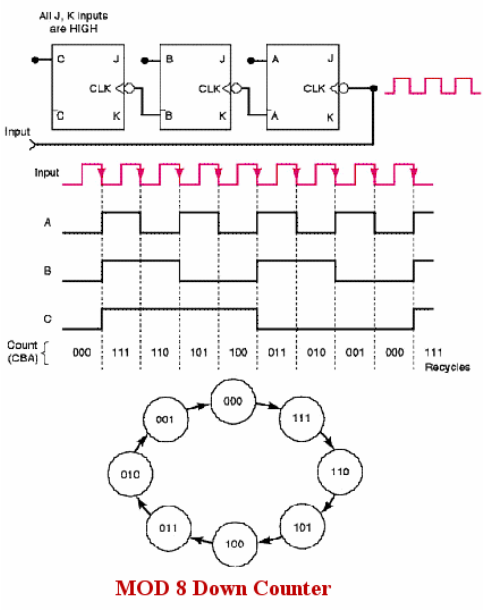




MOD 12 Asynchronous Counter

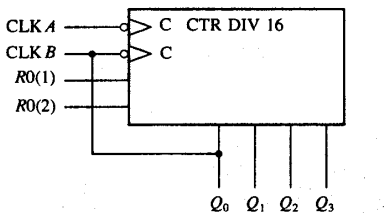




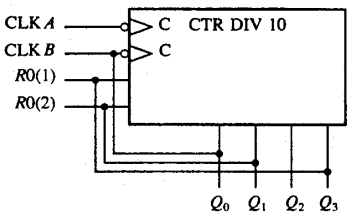


MOD 8 Down Counter

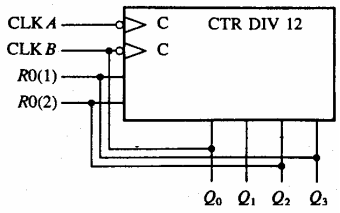
7493A MOD 16 Counter



(a) 7493A connected as a modulus-16 counter



(b) 7493A connected as a decade counter



7493A connected as a modulus-12 counter

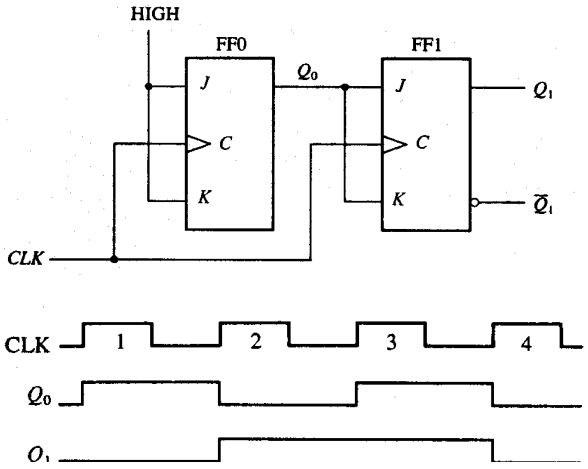
Synchronous Counters

ENGI 251
ELEC 241

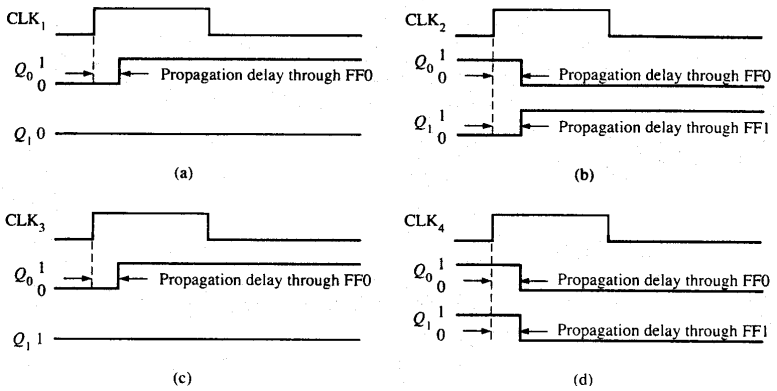
Synchronous Counters

- The term **synchronous** refers to events that do occur simultaneously
  - In communications, both ends must be connected – telephone call
- with respect to counter operation, synchronous means that the counter is connected such that all the Flip-Flops change at the same time
  - they are wired in a way that links all the flip-flop clock inputs together
  - this causes the output count states to change at the same time
  - There is a propagation delay, but they are typically very close in similar devices

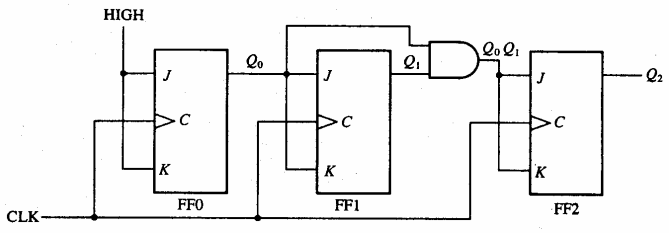
MOD 4 Synchronous Counter



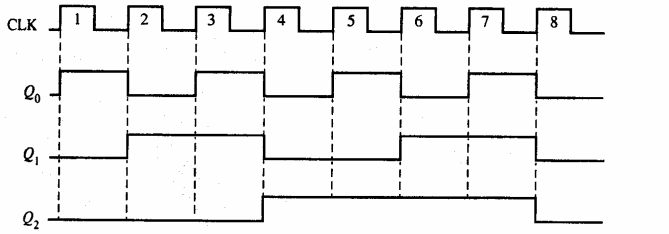
MOD 4 Counter Timing Diagram



MOD 8 Synchronous Counter

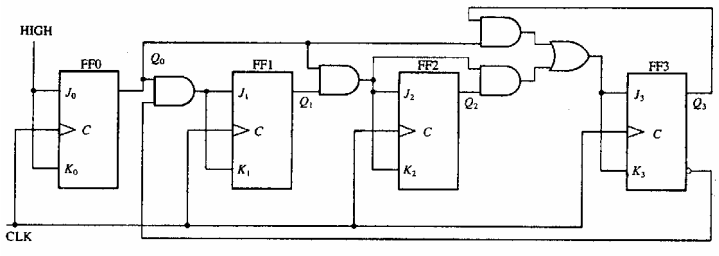


A three-bit synchronous binary counter



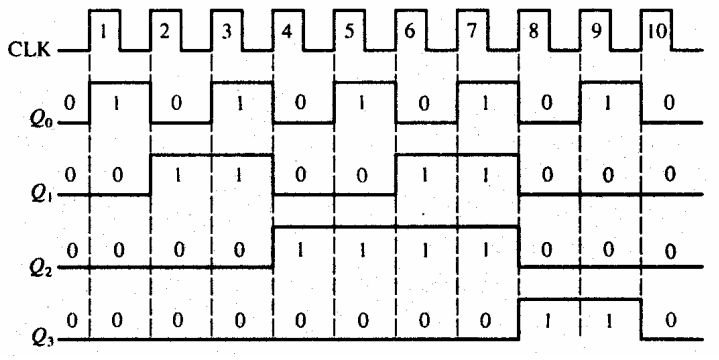
Timing diagram for the counter

MOD 10 Synchronous Counter



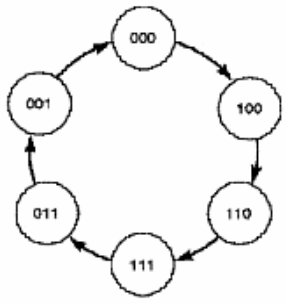
A synchronous BCD decade counter

MOD 10 Timing Diagram

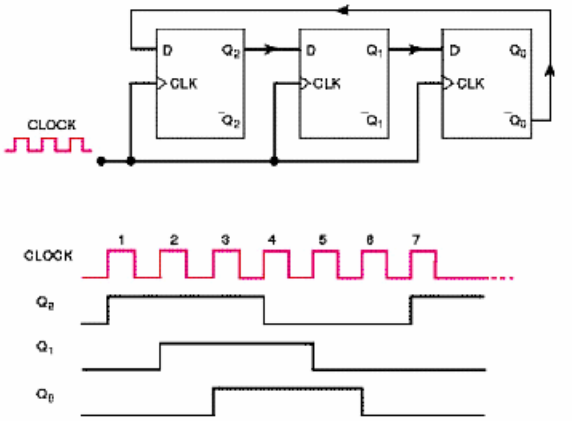


Johnson Counter State Diagram

Q_2	Q_1	Q_0	CLOCK pulse
0	0	0	0
1	0	0	1
1	1	0	2
1	1	1	3
0	1	1	4
0	0	1	5
0	0	0	6
1	0	0	7
1	1	0	8
.	.	.	.
.	.	.	.
.	.	.	.

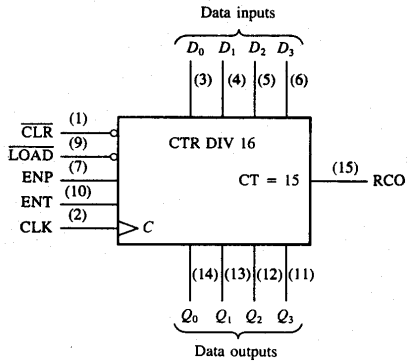


Johnson Counter



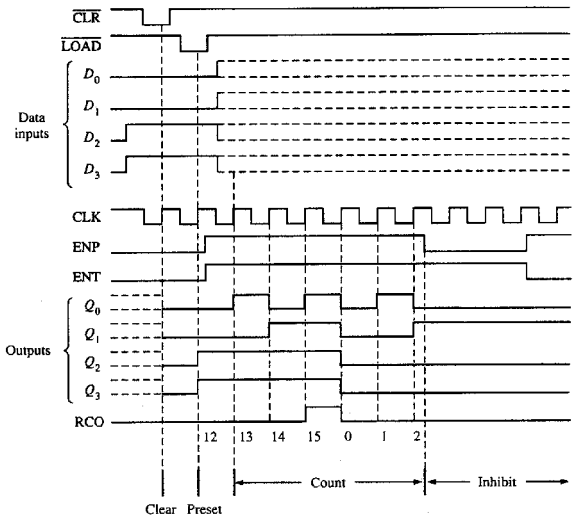
A Johnson counter is a special counter where the output of the last stage is inverted and fed back as input to the first stage. A pattern of bits equal in length circulates indefinitely. These counters are sometimes called "walking ring" counters, and find special applications.

74LS163A 4-bit Binary Counter

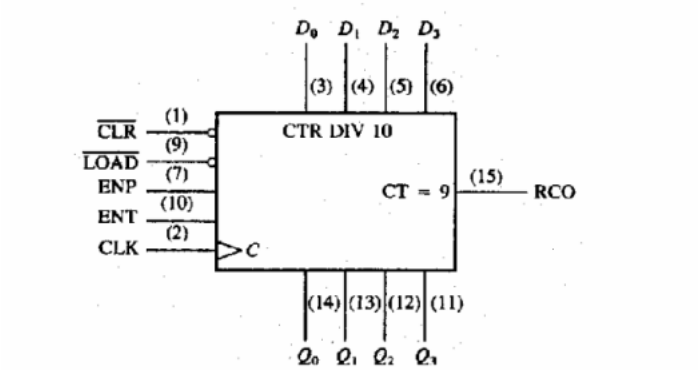


- The counter can be synchronously preset to any four-bit binary number by
  - When a LOW is applied to the LOAD input, the counter will assume the state of the data inputs on the next clock pulse
- The active-LOW CLR input synchronously RESETS all four flip- flops in the counter

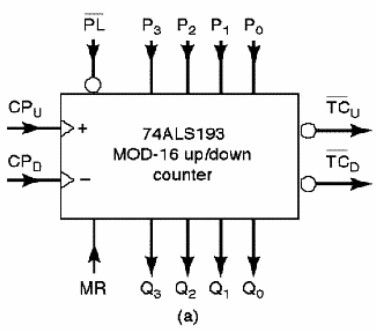
74LS163A Timing Diagram



74LS160A



The 74LS160A synchronous decade (BCD) counter



(a)

Mode Select

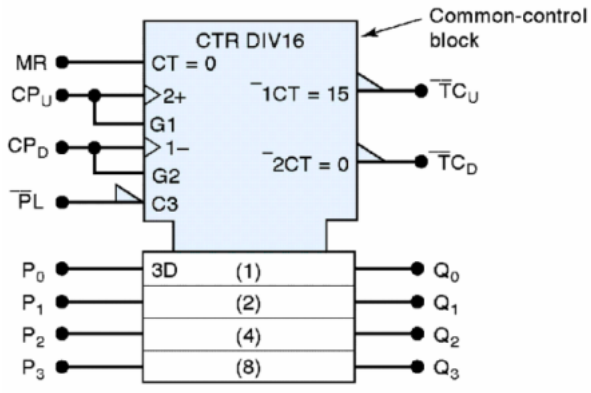
MR	$\overline{PL}$	CP_U	CP_D	Mode
H	X	X	X	Asynch. reset
L	L	X	X	Asynch. preset
L	H	H	H	No change
L	H	↑	H	Count up
L	H	H	↑	Count down

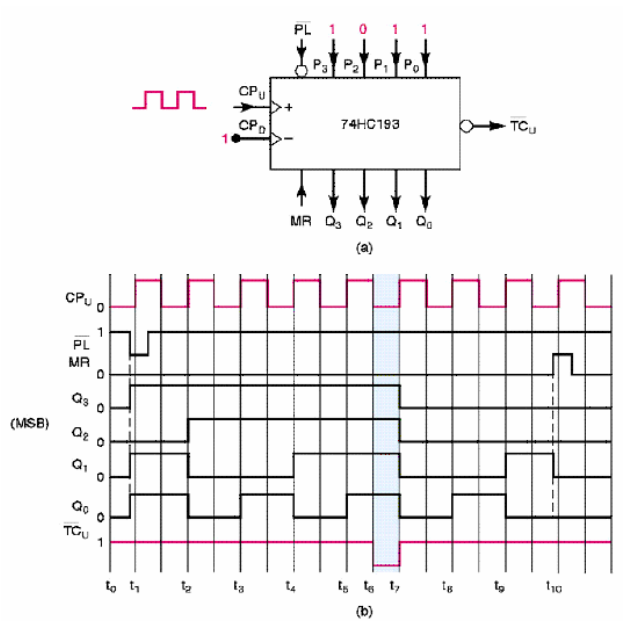
H = HIGH; L = LOW
X = Don't care; ↑ = PGT

(c)

Pin	Description
CP_U	Count-up clock input (active rising edge)
CP_D	Count-down clock input (active rising edge)
MR	Asynchronous master reset input (active HIGH)
$\overline{PL}$	Asynchronous parallel load input (active LOW)
P_0-P_3	Parallel data inputs
Q_0-Q_3	Flip-flop outputs
$\overline{TC}_D$	Terminal count-down (borrow) output (active LOW)
$\overline{TC}_U$	Terminal count-up (carry) output (active) LOW

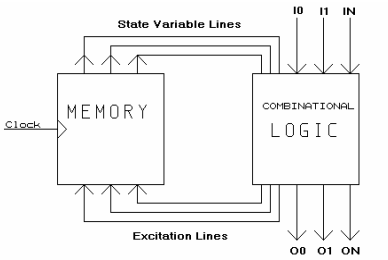
(b)





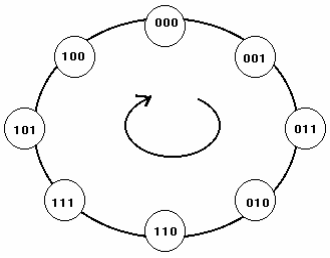
State Machine Design

State Machine Design



- The Figure above is the general diagram of a State Machine
- There are two basic components in a state machine, Memory which are usually JK Flip-Flops, and Combinational Logic
- To design a state machine, JK flip-flops are usually connected as a counter
- The following will demonstrate the design the logic necessary to allow the counter to sequence any desired binary pattern

Gray Code State Diagram



- The figure on the left shows the progression of states and the input and outputs for a Gray Code Counter
- The arrow in the center shows the direction of the counter
- If we start at 000, the next state is 001
- From 001 the next state is 011
- We progress through all possible states until the sequence repeats
- We use the State Diagram to create the Next State Table

Gray Code Next State Table

PRESENT STATE			NEXT STATE		
Q2	Q1	Q0	Q2	Q1	Q0
0	0	0	0	0	1
0	0	1	0	1	1
0	1	1	0	1	0
0	1	0	1	1	0
1	1	0	1	1	1
1	1	1	1	0	1
1	0	1	1	0	0
1	0	0	0	0	0

Gray Code Transition Table

STATE TRANSISTIONS			OUTPUT CONDITION			FLIP-FLOP INPUTS		
QC	QB	QA	QC	QB	QA	JC KC	JB KB	JA KA
$0 \rightarrow 0$	$0 \rightarrow 0$	$0 \rightarrow 1$	R/H	R/H	S/T	0 X	0 X	1 X
$0 \rightarrow 0$	$0 \rightarrow 1$	$1 \rightarrow 1$	R/H	S/T	S/H	0 X	1 X	X 0
$0 \rightarrow 0$	$1 \rightarrow 1$	$1 \rightarrow 0$	R/H	S/H	R/T	0 X	X 0	X 1
$0 \rightarrow 1$	$1 \rightarrow 1$	$0 \rightarrow 0$	S/T	S/H	R/H	1 X	X 0	0 X
$1 \rightarrow 1$	$1 \rightarrow 1$	$0 \rightarrow 1$	S/H	S/H	S/T	X 0	X 0	1 X
$1 \rightarrow 1$	$1 \rightarrow 0$	$1 \rightarrow 1$	S/H	R/T	S/H	X 0	X 1	X 0
$1 \rightarrow 1$	$0 \rightarrow 0$	$1 \rightarrow 0$	S/H	R/H	R/T	X 0	0 X	X 1
$1 \rightarrow 0$	$0 \rightarrow 0$	$0 \rightarrow 0$	R/T	R/H	R/H	X 1	0 X	0 X

X = Don't Care (may be 0 or 1)

K-MAP for Gray Code Transistion

JC	QA 0	QA 1
QC QB 0 0	0	0
QC QB 0 1	0	1
QC QB 1 1	X	X
QC QB 1 0	X	X

$JC = QB\,QA$

JB	QA 0	QA 1
QC QB 0 0	0	1
QC QB 0 1	X	X
QC QB 1 1	0	0
QC QB 1 0	X	X

$JB = \overline{QC}\,QA$

JA	QA 0	QA 1
QC QB 0 0	1	X
QC QB 0 1	X	0
QC QB 1 1	X	0
QC QB 1 0	1	X

$JA = \overline{QA}$ or $JA = \overline{QB}$

KC	QA 0	QA 1
QC QB 0 0	X	X
QC QB 0 1	X	X
QC QB 1 1	0	1
QC QB 1 0	0	0

$KC = QB\,QA$

KB	QA 0	QA 1
QC QB 0 0	X	X
QC QB 0 1	0	0
QC QB 1 1	X	X
QC QB 1 0	0	1

$KB = QC\,\overline{QA}$ or $KB = \overline{QB}\,QA$

KA	QA 0	QA 1
QC QB 0 0	X	0
QC QB 0 1	1	X
QC QB 1 1	1	X
QC QB 1 0	X	0

$KA = \overline{QA}$ or $KA = QB$

Note: In the above K-Map, the Boolean equation for KB, JA, and KA have two possible solution.

BCD Counter Design

Current State				Count	Next State				MSB						LSB	
D	C	B	A		D	C	B	A	JD	KD	JC	KC	JB	KB	JA	KA
0	0	0	0	1	0	0	0	1								
0	0	0	1	2	0	0	1	0								
0	0	1	0	3	0	0	1	1								
0	0	1	1	4	0	1	0	0								
0	1	0	0	5	0	1	0	1								
0	1	0	1	6	0	1	1	0								
0	1	1	0	7	0	1	1	1								
0	1	1	1	8	1	0	0	0								
1	0	0	0	9	1	0	0	1								
1	0	0	1	0	0	0	0	0								

H = Hold, R = Reset, S = Set, T = Toggle

Current State				Count	Next State				MSB						LSB	
D	C	B	A		D	C	B	A	JD	KD	JC	KC	JB	KB	JA	KA
0	0	0	0	1	0	0	0	1								
0	0	0	1	2	0	0	1	0								
0	0	1	0	3	0	0	1	1								
0	0	1	1	4	0	1	0	0								
0	1	0	0	5	0	1	0	1								
0	1	0	1	6	0	1	1	0								
0	1	1	0	7	0	1	1	1								
0	1	1	1	8	1	0	0	0								
1	0	0	0	9	1	0	0	1								
1	0	0	1	0	0	0	0	0								
1	0	1	0	10	X	X	X	X	X	X	X	X	X	X	X	X
1	0	1	1	11	X	X	X	X	X	X	X	X	X	X	X	X
1	1	0	0	12	X	X	X	X	X	X	X	X	X	X	X	X
1	1	0	1	13	X	X	X	X	X	X	X	X	X	X	X	X
1	1	1	0	14	X	X	X	X	X	X	X	X	X	X	X	X
1	1	1	1	15	X	X	X	X	X	X	X	X	X	X	X	X

BCD Counter Design