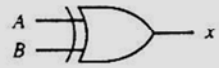
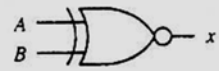
Lectured Three

Logic Gats: 1- Set of Gets

Name	Graphic symbol	Algebraic function	Truth table
AND		$x = A \cdot B$ or $x = AB$	A B | x 0 0 | 0 0 1 | 0 1 0 | 0 1 1 | 1
OR		$x = A + B$	A B | x 0 0 | 0 0 1 | 1 1 0 | 1 1 1 | 1
Inverter		$x = A'$	A | x 0 | 1 1 | 0
Buffer		$x = A$	A | x 0 | 0 1 | 1
NAND		$x = (AB)'$	A B | x 0 0 | 1 0 1 | 1 1 0 | 1 1 1 | 0
NOR		$x = (A + B)'$	A B | x 0 0 | 1 0 1 | 0 1 0 | 0 1 1 | 0
Exclusive-OR (XOR)		$x = A \oplus B$ or $x = A'B + AB'$	A B | x 0 0 | 0 0 1 | 1 1 0 | 1 1 1 | 0
Exclusive-NOR or equivalence		$x = (A \oplus B)'$ or $x = A'B' + AB$	A B | x 0 0 | 1 0 1 | 0 1 0 | 0 1 1 | 1

2- ***Half – Adder:*** The basic digital arithmetic circuit is the addition of two binary digits. Input variables of a half-adder call augends & addend bits. The output variables the sum & carry.

Figure (1-a) Logic diagram for half adder

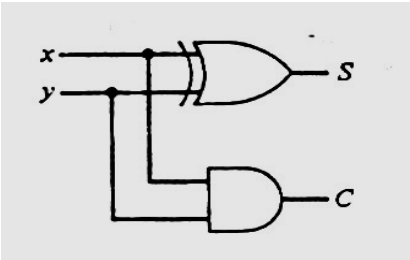


Figure (1-b) Truth table for half adder

X	Y	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Half- Adder questions:

S=X$\overline{Y}$+X$\overline{Y}$

S=X (+)Y

C=X*Y

3- ***Full-Adder:*** A full - adder is a combinational circuit that forms the arithmetic sum of three input bits. It consists of three inputs &two outputs.

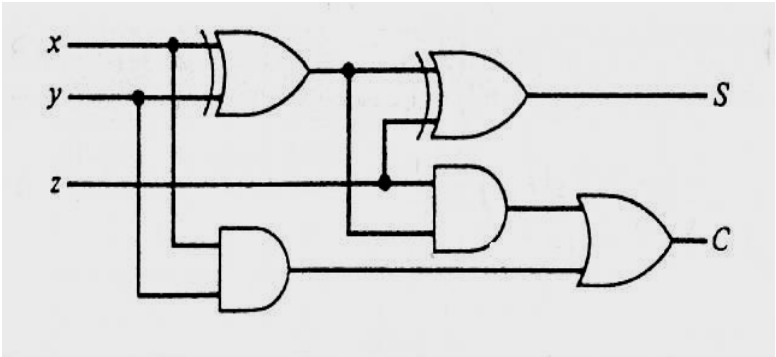


Figure (2-a) Logic diagram for full adder (Logic Diagram)

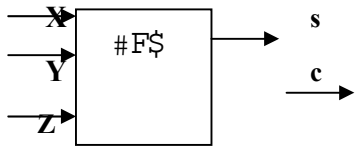


Figure (2-b) Block diagram for full adder

Inputs			Out puts	
X	Y	Z	C	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Figure (2-c) Truth table for full adder

Full - Adder questions: **S=x (+) y (+) z**

C=XY+ (XZ (+) YZ)

C=X*Y + (X (+) Y) Z

#

Lecture Four

Boolean Algebra &Logic Simplification:

1- Rules of Boolean algebra: 1-

A+0=A

2- A+1=1

3- A*0=0

4- A*1=A

5- A+A=A

6- A+A=1

7- A*A=A

8- A*A=0

9- A=A ====== Demoragan's theorem 10-

A+BA=A

11- A+AB=A+B

12- (A+B)(A+C)=A+BC

2- Examples:

Example 1:

F = X + ýz

Determine the truth table and logic diagram

X	Y	Z	F
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

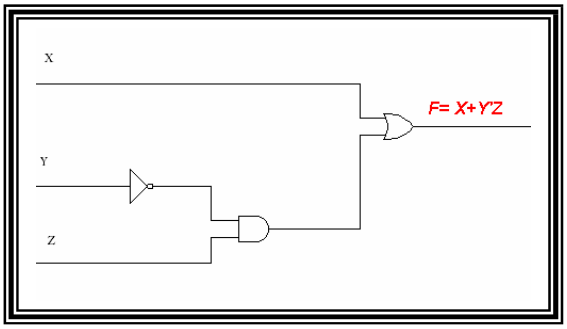


Figure (3-a) Truth table **figure (3-b) Logic diagram**

Example 2:

AB+ A (B+C)+ B(B+C)

1- **AB+AB+AC+BB+BC**
2- **AB+AB+AC+B+BC**
3- **AB+AC+B+BC**
4- **AB+AC+B**
5- **B+AC**

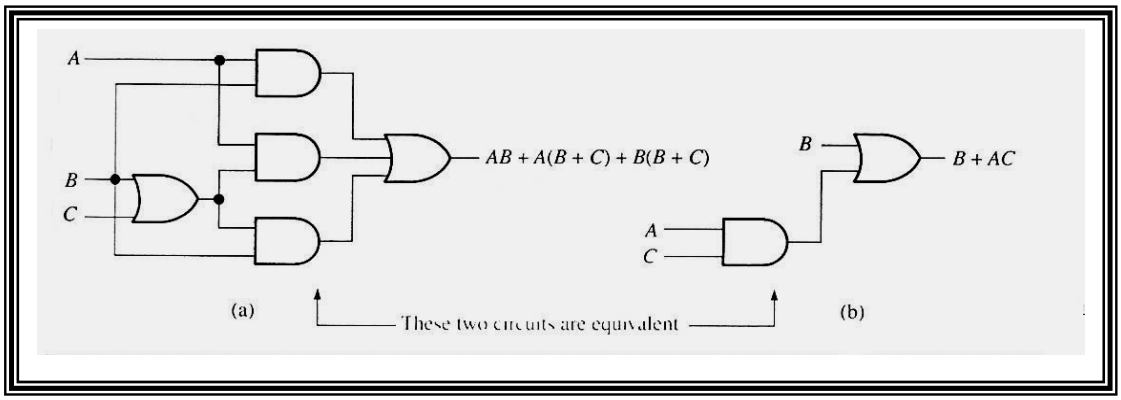


Figure (4)

Example 3:

F=ABC+ABĆ+ĂC F=

AB(C+Ć) +ĂC F=

AB+ĂC

Example 4:

Simplify the following Boolean expression:

$$\overline{AB + AC} + \overline{A}\,\overline{B}C$$

Solution **Step 1.** Apply DeMorgan's theorem to the first term.

$$(\overline{AB})(\overline{AC}) + \overline{A}\,\overline{B}C$$

Step 2. Apply DeMorgan's theorem to each term in parentheses.

$$(\overline{A} + \overline{B})(\overline{A} + \overline{C}) + \overline{A}\,\overline{B}C$$

Step 3. Apply the distributive law to the two terms in parentheses.

$$\overline{A}\,\overline{A} + \overline{A}\,\overline{C} + \overline{A}\,\overline{B} + \overline{B}\,\overline{C} + \overline{A}\,\overline{B}C$$

Step 4. Apply rule 7 ($\overline{A}\,\overline{A} = \overline{A}$) to the first term, and apply rule 10 [$\overline{A}\,\overline{B} + \overline{A}\,\overline{B}C = \overline{A}\,\overline{B}(1 + C) = \overline{A}\,\overline{B}$] to the third and last terms.

$$\overline{A} + \overline{A}\,\overline{C} + \overline{A}\,\overline{B} + \overline{B}\,\overline{C}$$

Step 5. Apply rule 10 [$\overline{A} + \overline{A}\,\overline{C} = \overline{A}(1 + \overline{C}) = \overline{A}$] to the first and second terms.

$$\overline{A} + \overline{A}\,\overline{B} + \overline{B}\,\overline{C}$$

Step 6. Apply rule 10 [$\overline{A} + \overline{A}\,\overline{B} = \overline{A}(1 + \overline{B}) = \overline{A}$] to the first and second terms.

$$\overline{A} + \overline{B}\,\overline{C}$$

3- Demorgan's theorems:

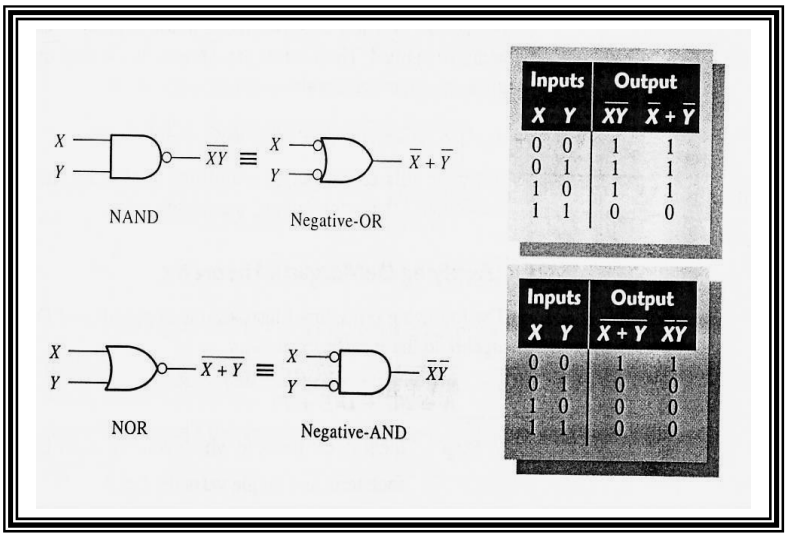


Figure (5) Demorgan's theorems

Example 1:

a- (A+B)+C = (A+B) C = (A+B)C

b- (A+B) +CD = (A+B) CD =(A B) (C+D) = A B (C+D)

c- (A+B) C D + E + F = ((A+B) C D) (E+ F)

= (A*B+C+D)*(E F)

= (A*B+C+D) E F

5- *Sum – Of – Products (SOP):*

X=AB+BCD+AC

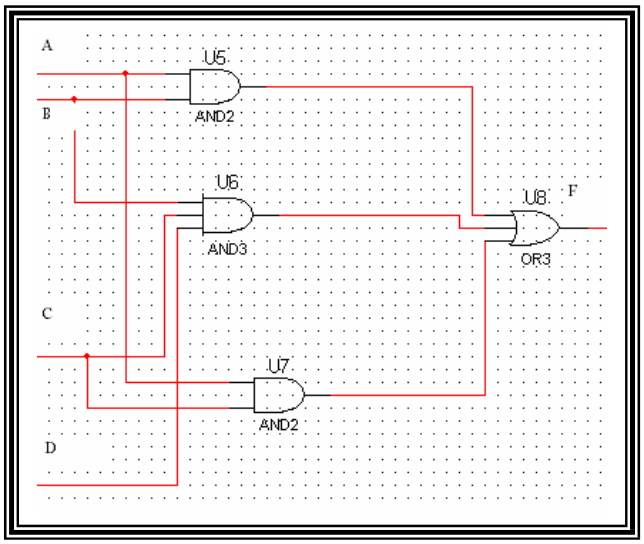


Figure (4) SOP

Examples:

a- AB+B(CD+EF)=AB+BCD+BEF

b- (A+B)(B+C+D)=AB+AC+AD+BB+BC+BD c-

(A+B)+C=(A+B)*C =(A+B)C=AC+BC

6- *Product – Of – Sum(POS):*

(A+B)(B+C+D)(A+C)

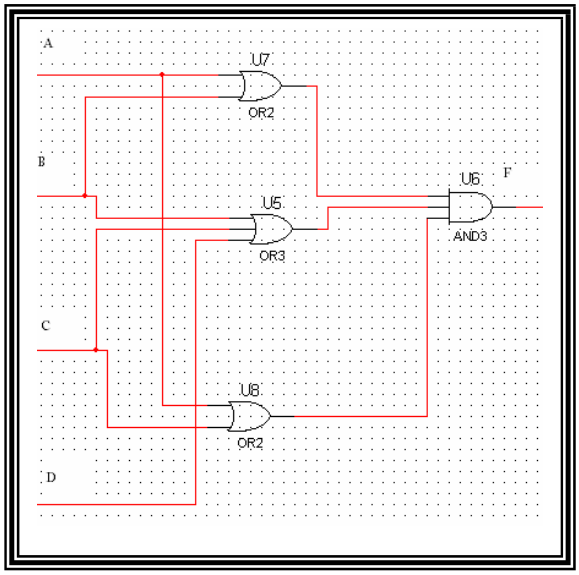


Figure (5) POS

Example: SOP

A	B	X	F
0	0	0	
0	1	1	$\bar{A}B$
1	0	1	$A\bar{B}$
1	1	0	

Example: POS

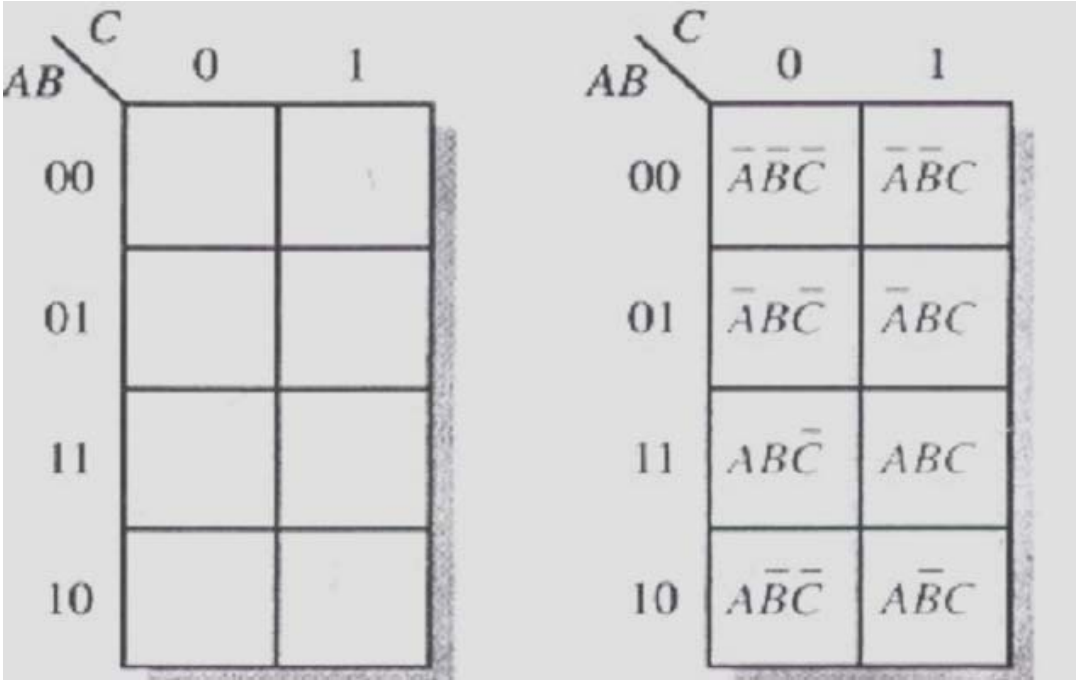
A	B	X	F
0	0	0	$\bar{A}+\bar{B}$
0	1	1	
1	0	1	
1	1	0	A+ B

#

Lectured Five

Karnaugh map:

1- Three – variable karnaugh map.



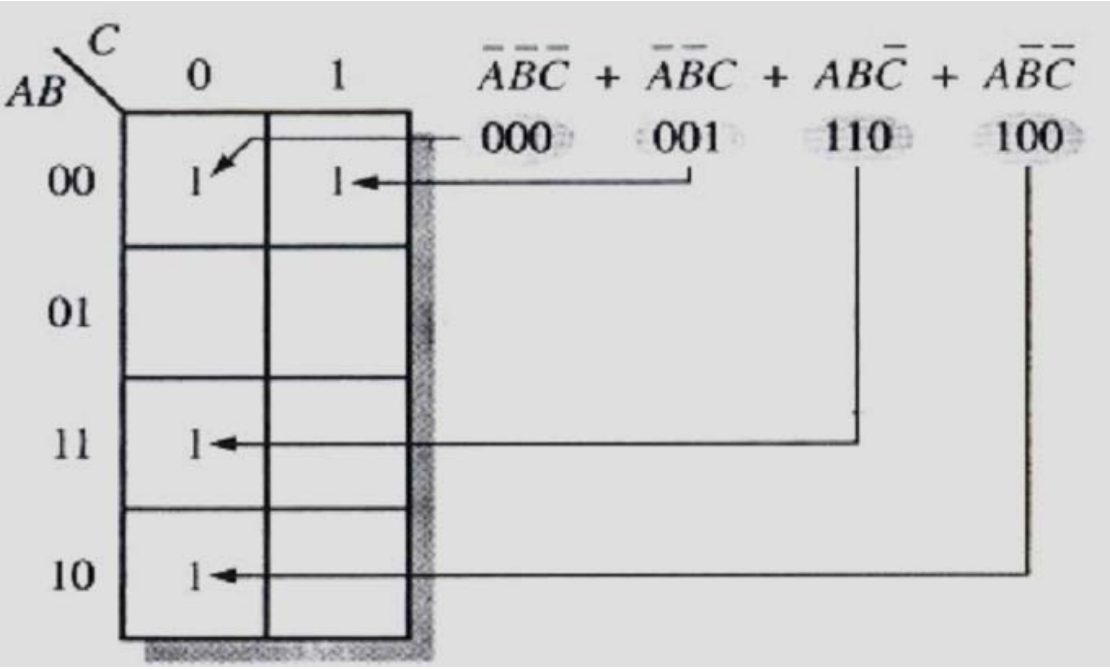
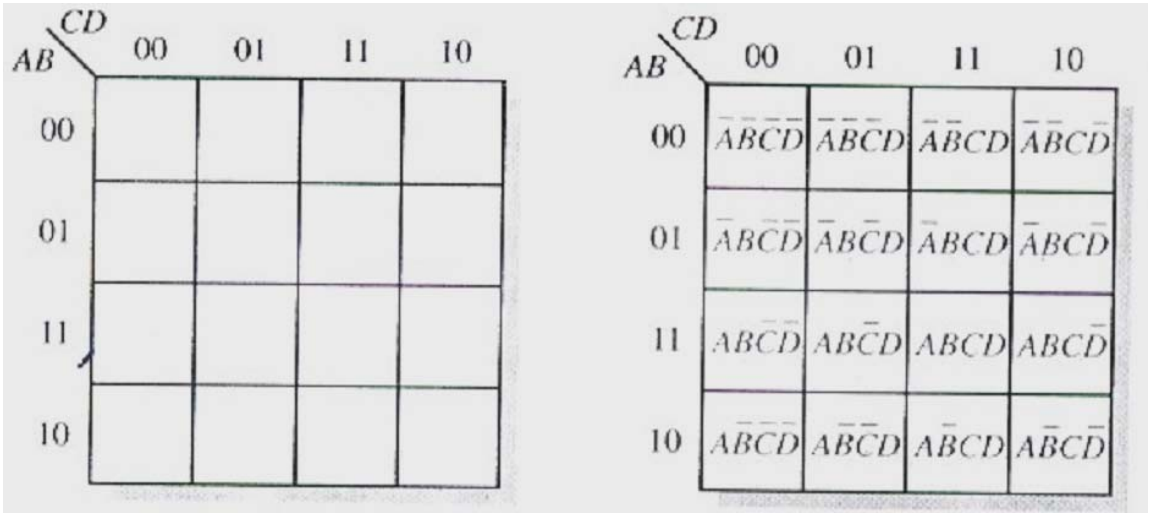


Figure (6)

2- Four – variable karnaugh map.



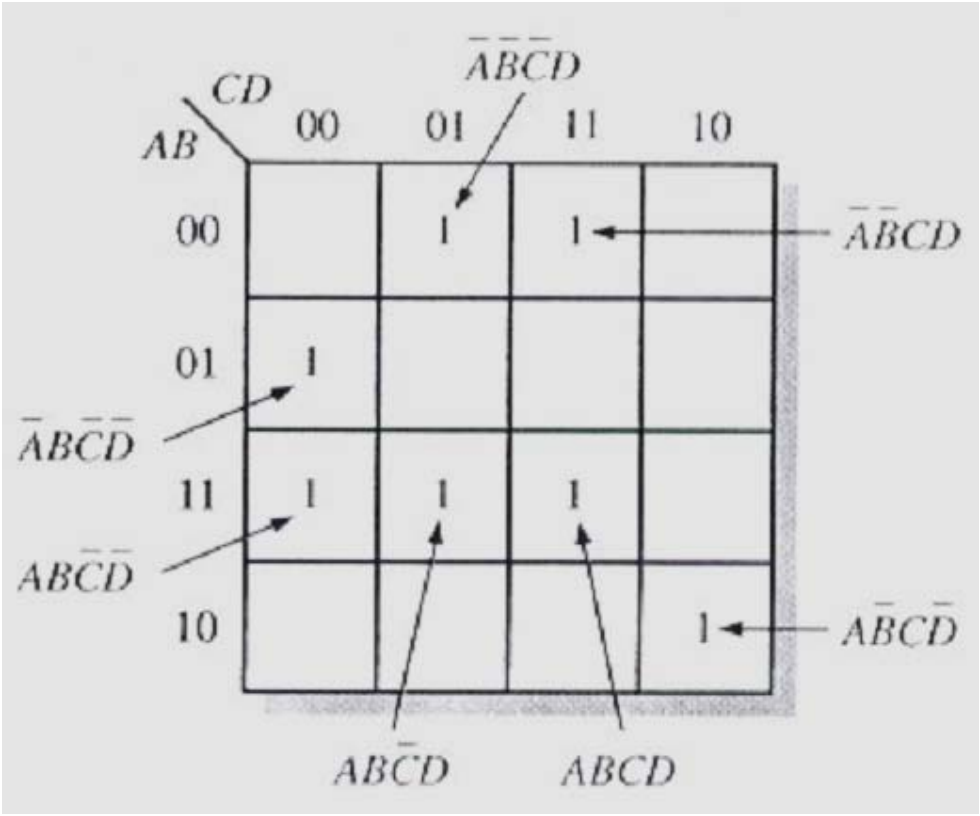


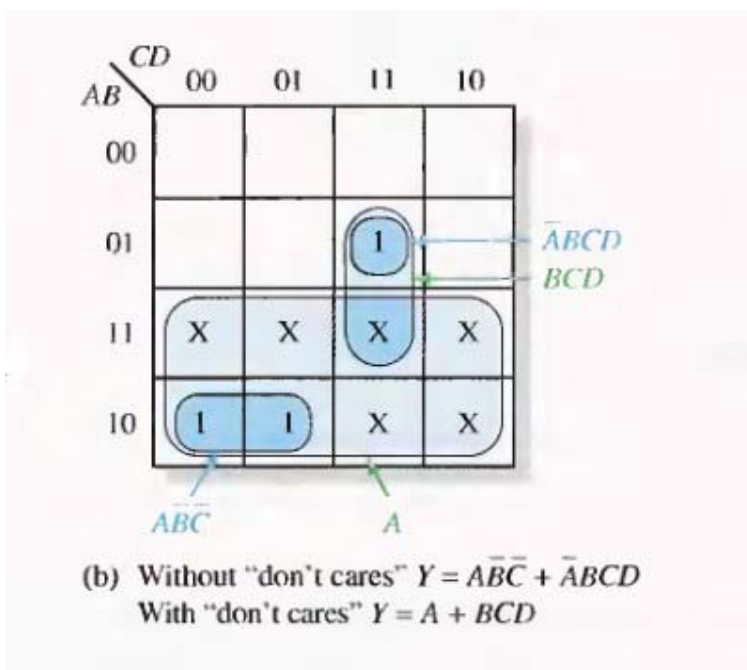
Figure (7)

3- Don't care karnaugh map:-

The squares of a K-map marked with 1's for the function. The other squares are assumed to be 0's. This is not always true, because there may be situations

AB \ CD	00	01	11	10
00	1	X	0	1
01	X	1	X	X
11	1	0	X	0
10	1	0	0	1

Example:



(b) Without "don't cares" $Y = A\bar{B}\bar{C} + \bar{A}BCD$
With "don't cares" $Y = A + BCD$

Figure (8)

Second course

#

Lectured Six

1- The NAND Gate as a Universal Logic Element:

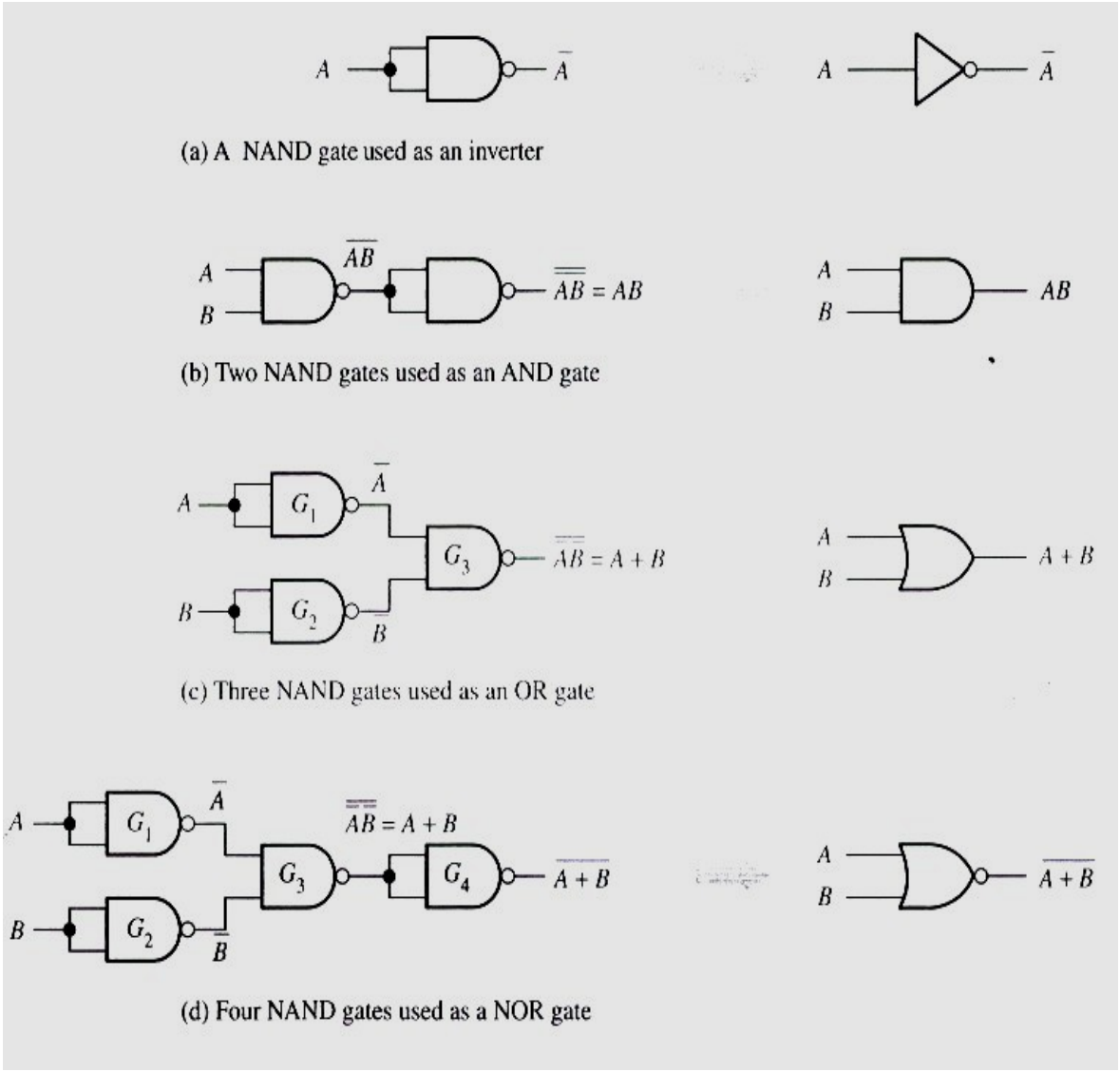


Figure (9) NAND Gates

2- The NOR Gate as a Universal Logic Element:

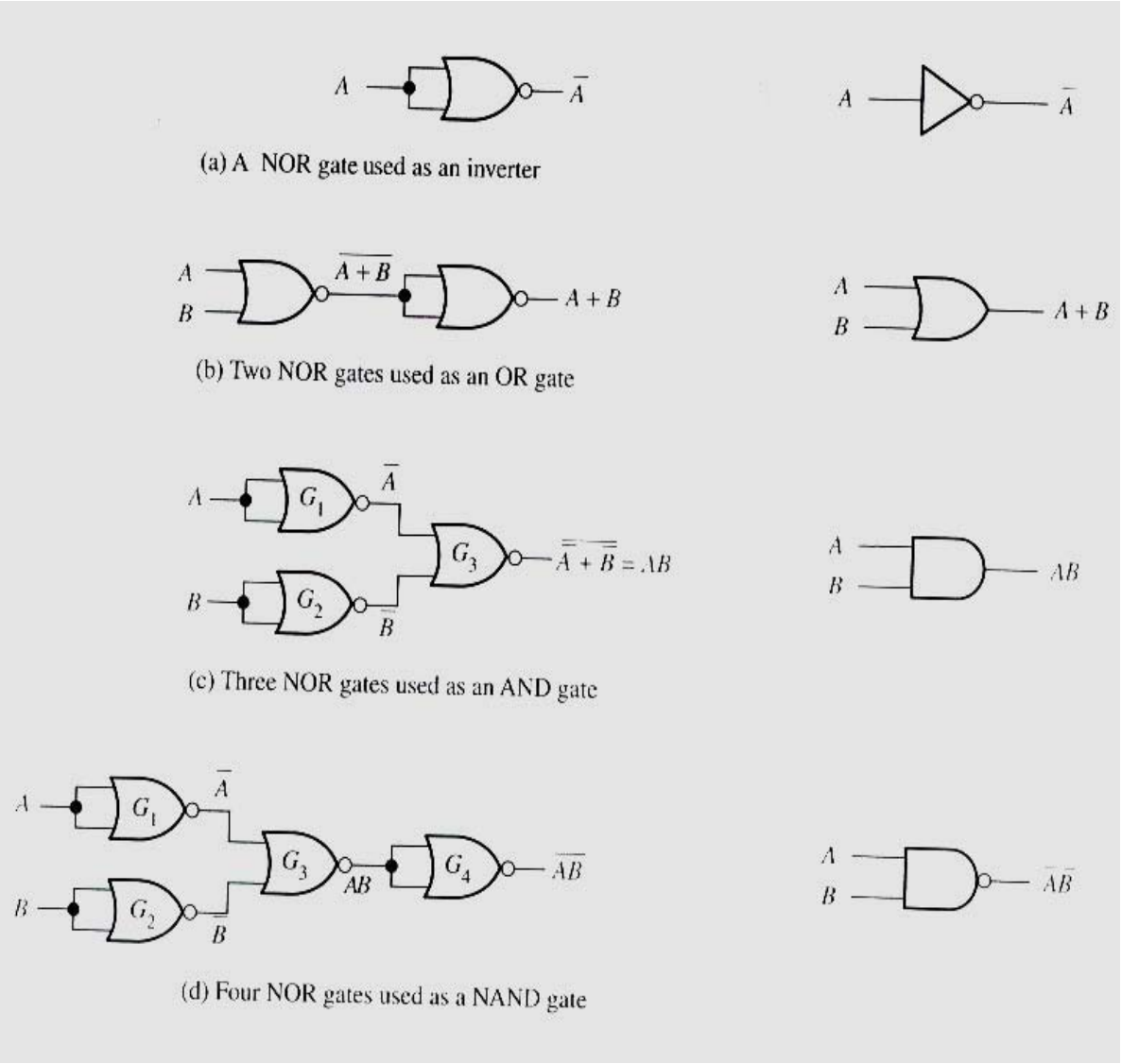


Figure (10) NOR Gates

3- 4- Bit Parallel Adder:

A group of four bits is a nibble. A basic 4-bit parallel adder is implementation with four full adder stages.

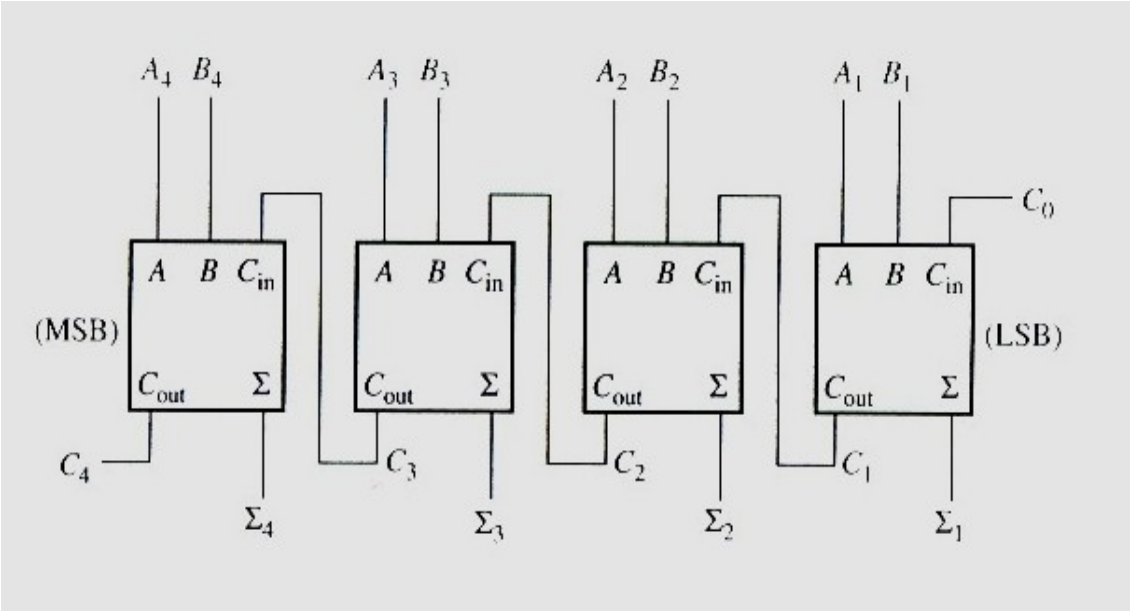


Figure (11) 4-bit parallel adder

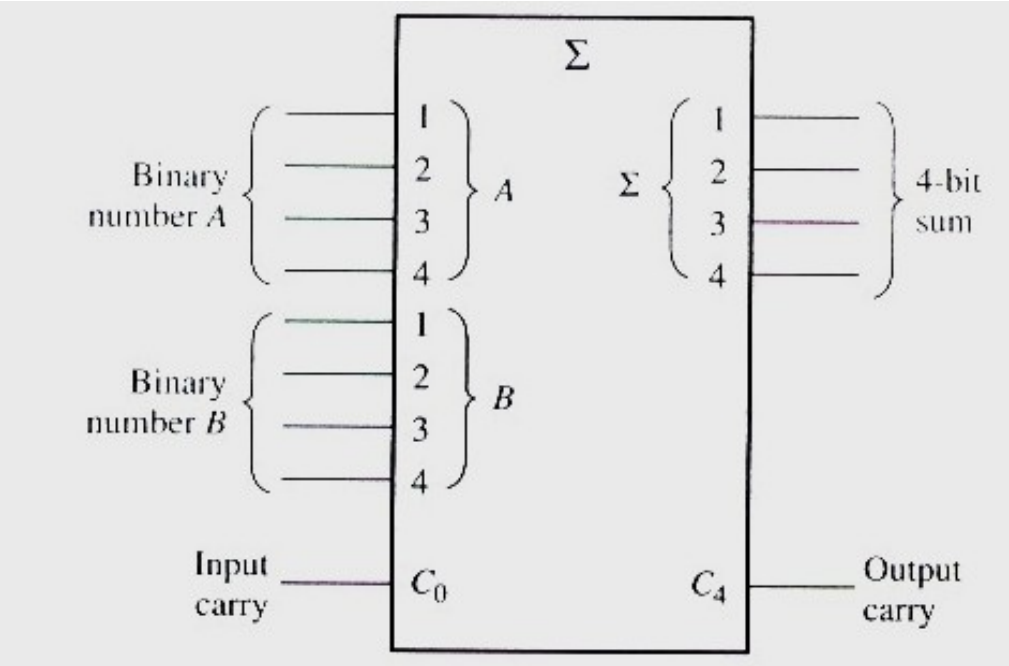


Figure (12) Symbol Logic

4- Example:

Draw the 4-bit parallel adder, find the sum and output carry for the addition of the following two 4-bit numbers if the input carry (C_{n-1}) is 0:

A4A3A2A1=1010 and B4B3B2B1=1011

<u>Solution:</u>

For n=1

A1=0, B1=1, C_{n-1}=0

$\sum$ =1, and C1=0 For

n=2

A2=1, B2=1, C_{n-1}=0

$\sum$=0, and C2=1 For

n=3

A3=0, B3=0, C_{n-1}=1

$\sum$=1, and C3=0 For

n=4

A4=1, B4=1, C_{n-1}=0

$\sum$=0, and C4=1

4- 4-Bit subtracted Adder:

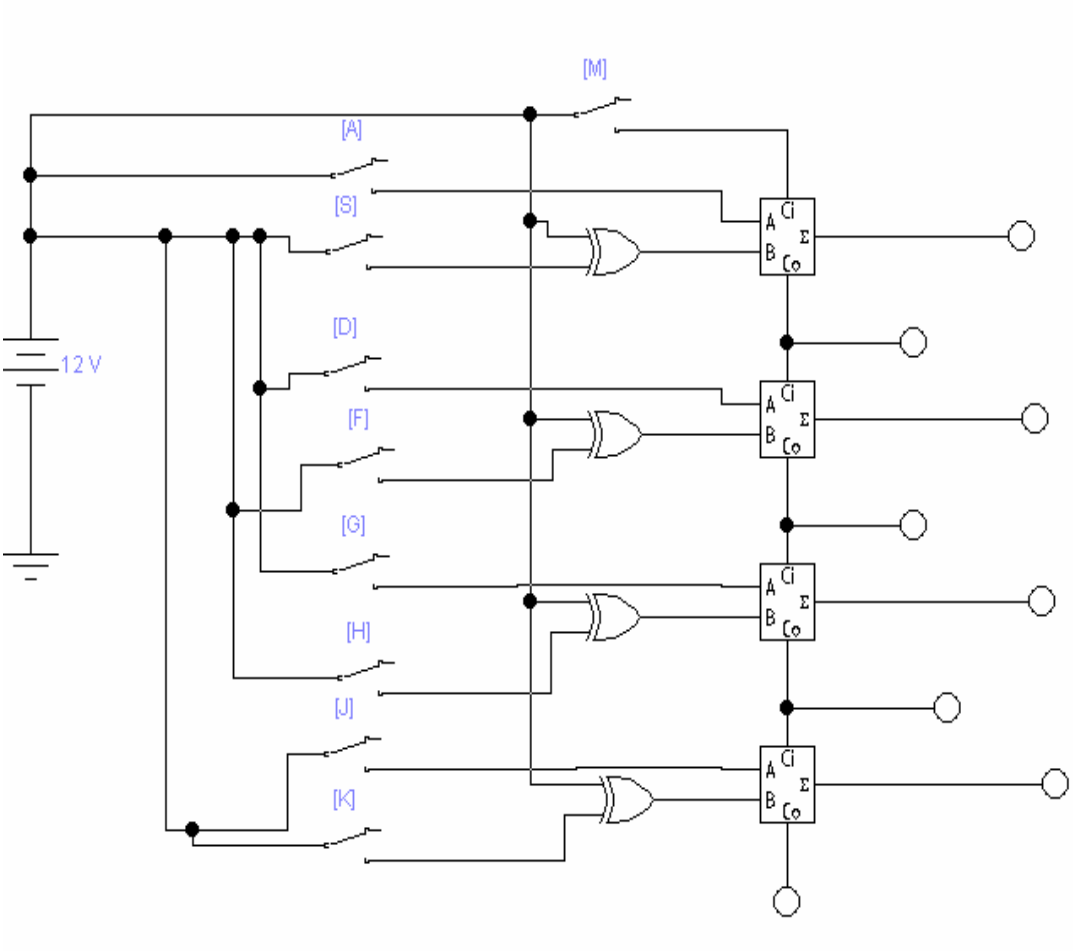


Figure (13) 4-Bit subtracted Adder (Logic Diagram)

#

Lectured Seven

<u>Decoders & encoders:</u> 1-

<u>Decoder</u>:

A decoders is combinational circuit that converts binary information form the n coded inputs to a maximum of 2^n unique outputs.

That decoders are called n-to-m line decoders where m <=2^n.

The logic diagram of a 3-to-8 line decoder is three data inputs, A0, A1, and A2 are decoded into eight out puts, each out puts representing one of the combinations of the three binary input variables.

This decoder is a binary – to – octal conversion.

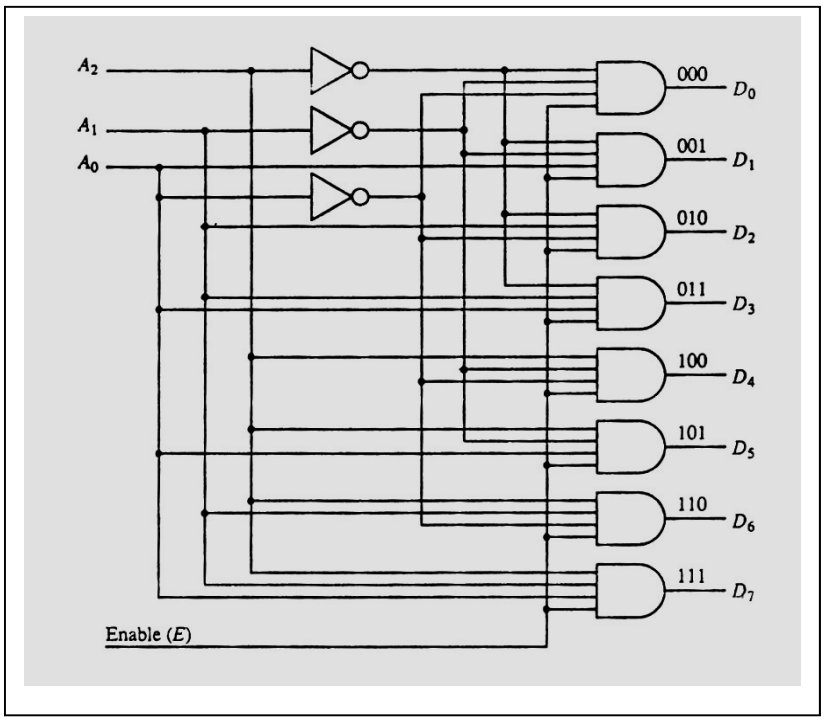


Figure (14-a) 3-to-8 line decoder (Logic Diagram)

Enable	Inputs			Outputs							
E	A2	A1	A0	D7	D6	D5	D4	D3	D2	D1	D0
0	X	X	X	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	1
1	0	0	1	0	0	0	0	0	0	1	0
1	0	1	0	0	0	0	0	0	1	0	0
1	0	1	1	0	0	0	0	1	0	0	0
1	1	0	0	0	0	0	1	0	0	0	0
1	1	0	1	0	0		0	0	0	0	0
1	1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	1	0	0	0	0	0	0	0

Figure (14-b)Truth table for 3-to-8 line decoder

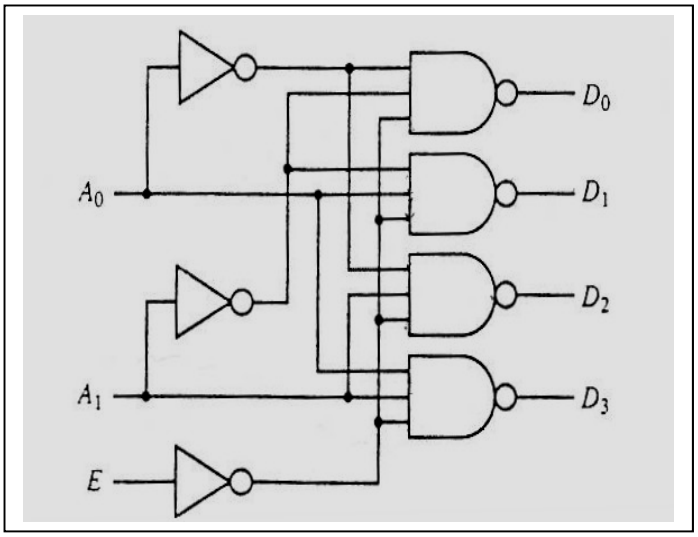


Figure (15-a) 2-to-4 line decoder (Logic Diagram)

Enable	Inputs		Outputs			
E	A1	A0	D0	D1	D2	D3
0	0	0	0	1	1	1
0	0	1	1	0	1	1
0	1	0	1	1	0	1
0	1	1	1	1	1	0
0	X	X	1	1	1	1

Figure (15-b)Truth table for 2-to-4 line decoder

2- Encoder:

An encoder is a digit circuit that performs the inverse operation of a decoder. An encoder has 2^n (or less) input lines and n output lines. An encoder is the octal – to – binary encoder.

It has eight inputs, one for each of the octal digits, and three outputs that generate the corresponding binary number.

A0 = D1+D3+D5+D7 A1 =

D2+D3+D6+D7 A2 =

D4+D5+D6+D7

(Implementation in three OR gates)

Inputs								Outputs		
D7	D6	D5	D4	D3	D2	D1	D0	A2	A1	A0
0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	1	0	0	0	1	0
0	0	0	0	1	0	0	0	0	1	1
0	0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	0	1	0	1
0	1	0	0	0	0	0	0	1	1	0
1	0	0	0	0	0	0	0	1	1	1

Figure (16-a) Truth table for octal – to – binary encoder

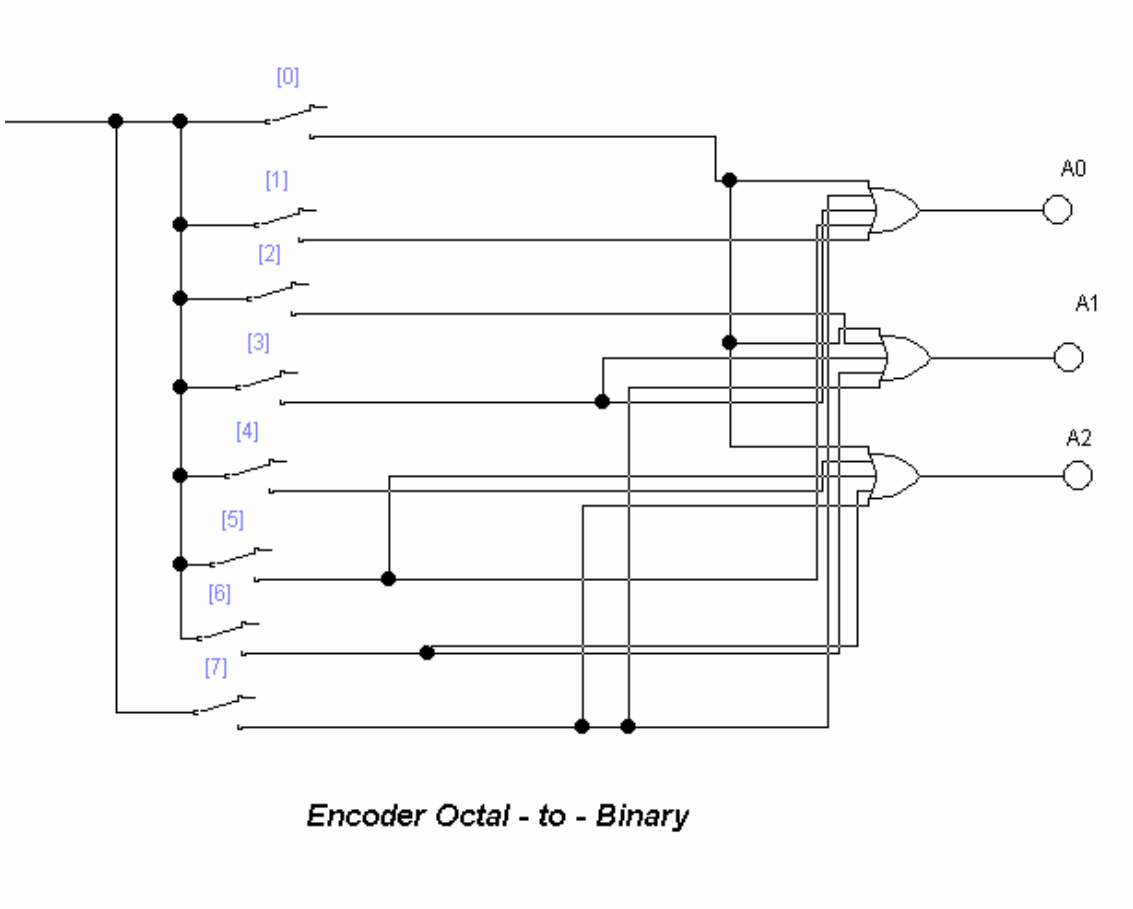


Figure (16-b) 8 – to – 3 lines Encoder (Logic Diagram)

3- Multiplexers:

A multiplexer is a combinational circuit that receiver binary information form one of 2^n input data lines and directs it to a single out put line.

The selection of a particular input data line for the output is determined by a set of selection inputs. A 2^n- to- 1 , A 4-to-1. Multiplexer is called Data Selector.

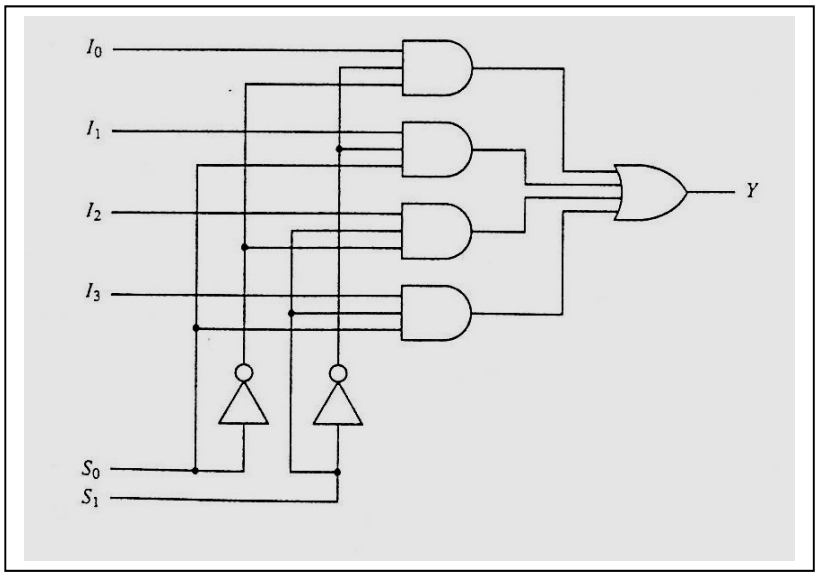


Figure (17-a) 4-to-1 line multiplexer (Logic Diagram)

Inputs		Outputs
S0	**S1**	**Y**
0	**0**	**Y1**
0	**1**	**Y2**
1	**0**	**Y3**
1	**1**	**Y4**

Figure (17-b) Truth table for 4-to-1 multiplexer

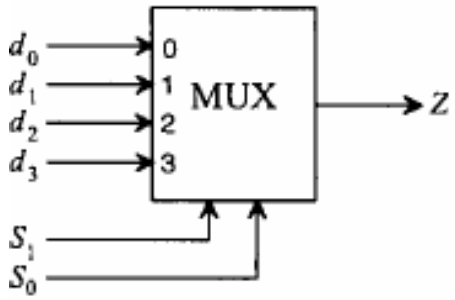


Figure (17-c) Implementation 4-to-1 MUX (Block Diagram)

4-Demultiplexers:

A demultiplexer (DEMUX) basically reverses the multiplexing function. It takes digital information from one line and distributes it to a given number of output lines. For this reason, the demultiplexer is also known as a data distributor. As you will learn, decoders can also be used as demultiplexers.

A 1 to 4 lines demultiplexer (DEMUX) circuit. The data input line goes to all of the AND gates. The two data select lines enable only one gate at a time, and the data appearing on the data input line will pass through the selected gate to the associated data output line.

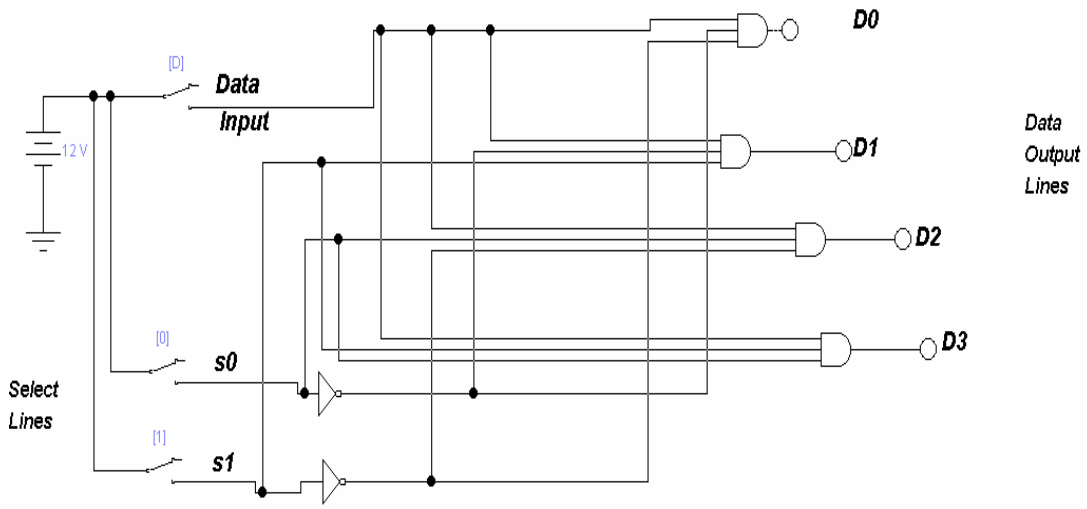


Figure (18-a) Demultiplexer 1 to 4 lines (Logic Diagram)

Inputs			Outputs			
Data	S0	S1	D4	D3	D2	D1
0	0	0	0	0	0	0
1	0	0	0	0	0	1
1	0	1	0	0	1	0
1	1	0	0	1	0	0
1	1	1	1	0	0	0

Figure (18-b) Truth table for 4-to-1 Demultiplexer

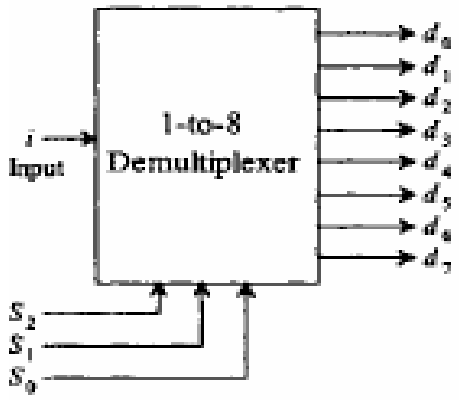


Figure (18-c) Implementation 1-to-8 DEMUX (Block Diagram)

#

Lectured Eight

Flip-Flop:

The storage elements employed in clocked sequential circuits are called flip-flops. A flip -flops is a binary cell capable of storing one bit of information. It has two outputs, one for the normal value and one for the complement value of the bit stored in it.

Type of flip-flops:

1- **SR flip-flops.**
2- **D flip-flops.**
3- **JK flip-flops.**

Latches :

The latch is a type of temporary storage device that has two enable states (bistable) and is normally placed in a category separate from that of flip-flops. Latches are similar to flip-flops because they are bistable devices that can reside in either of two states using a feedback arrangement, in which the outputs are connected back to the opposite inputs. The main difference between latches and flip-flop is the method used for changing their state.

The S-R (SET-RESET) Latch:

A latch is a type of bistable logic device or multivibrator. An active – HIGH input S-R (SET-RESET) latch is formed with two cross-couple NOR gates, as shown in figure (19-a); an active-LOW input $\overline{S}$-$\overline{R}$ latch is formed with two-couple NAND gates, as shown in figure (19-b). Notice that the output of each gate is connected to an input of the opposite gate; this produces the regenerative feedback that is characteristic of all latches and flip-flops.

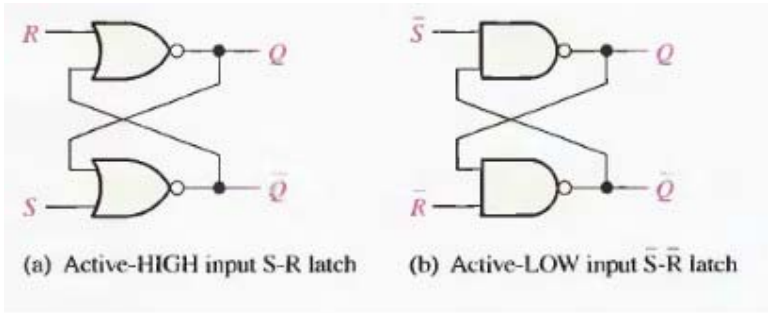
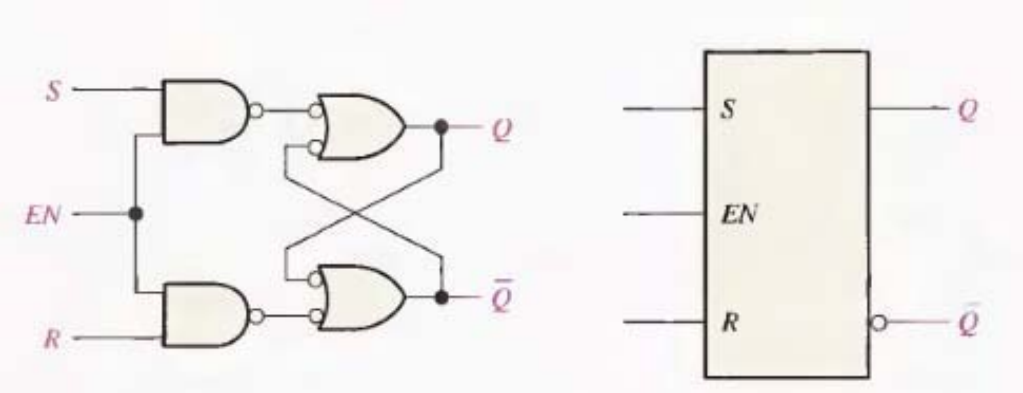


Figure (19-a, b) The S-R (SET-RESET) Latch

1- <u>The Gated S-R Latch :</u>

A gated latch requires an enable input, EN (G is also used to designate an enable input). The logic diagram and logic symbol for a gated S-R latch are shown in figure (19- c, d). the S and R input control the state to which the latch will go when a HIGH level is applied to the EN input. The latch will not change until EN is HIGH; but as long as it remains HIGH, the output is controlled by the state of the S and R inputs. In this circuit, the invalid state occurs when both S and R are simultaneously HIGH.



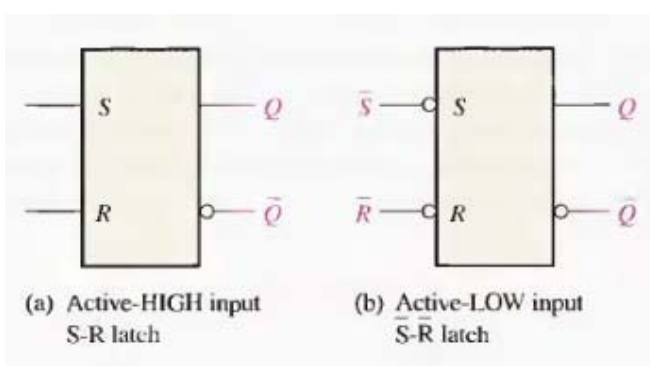
c- Logic diagram d-Logic symbol Figure (19-c, d) SR Latch

Inputs		Outputs		Comments
S	R	Q	$\bar{Q}$	
1	1	1	1	Invalid condition
0	1	1	0	Latch set
1	0	0	1	Latch reset
0	0	N.C	N.C	No change

Figure (19-e) Truth table for SR Latch

INPUTS $\bar{S}$	$\bar{R}$	OUTPUTS Q	$\bar{Q}$	COMMENTS
1	1	NC	NC	No change. Latch remains in present state.
0	1	1	0	Latch SET.
1	0	0	1	Latch RESET.
0	0	1	1	Invalid condition

Figure (19-f) Truth table for $\bar{S}$-$\bar{R}$ Latch



(a) Active-HIGH input S-R latch

(b) Active-LOW input $\bar{S}$-$\bar{R}$ latch

Figure (19-g) logic symbol for the $S\text{-}R$ **and** $\overline{S}\text{-}\overline{R}$ **latch**

2- The Gated D Latch:

Another type of gated latch is called the D latch. It differs from the S-R latch because it has only one input in addition to EN. This input is called the D (data) input. Figure (20–a) contains a logic diagram and logic symbol of a D latch. When the D input is HIGH an the EN input is HIGH, the latch will set. When the D input is LOW and EN is HIGH, the latch will reset. Stated another way, the output Q follows the input D when EN is HIGH.

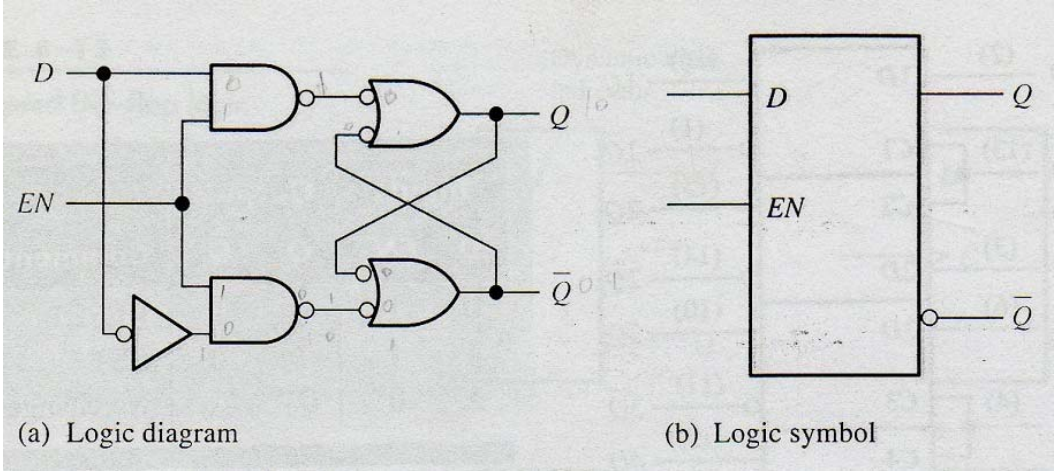


Figure (20-a) D Latch

Inputs		Outputs		
D	**CLK**	**Q**	□ *Q*	**Comments**
1	↑	**1**	**0**	**Set(stor1)**
0	↑	**0**	**1**	**Reset(stor0)**

Figure (20-b) Truth table for D Latch

3- J K FLIP-FLOPS:

The J-K flip-flop is versatile is a widely used type of flip-flop. The functioning of the J-K flip-flop is identical to that of the S-R flip-flop in the SET. RESET and no-change conditions of operation. The deference is that the J-K flip-flop has no invalid state as does the S-R flip-flop.

Figure (21-a) shows the basic internal logic for a positive edge- triggered J-K flip-flop. It differs from the S-R edge-triggered flip- flop in that Q output is connected back to the input of gate G2, and

the $\bar{Q}$ **output is connected back to the input of gate G1. The two**

control inputs are labeled J and K in honor of jack kilby, who invented the integrated circuit. A J-K flip-flop can also be of the negative edge-triggered type, in which case the clock input is inverted.

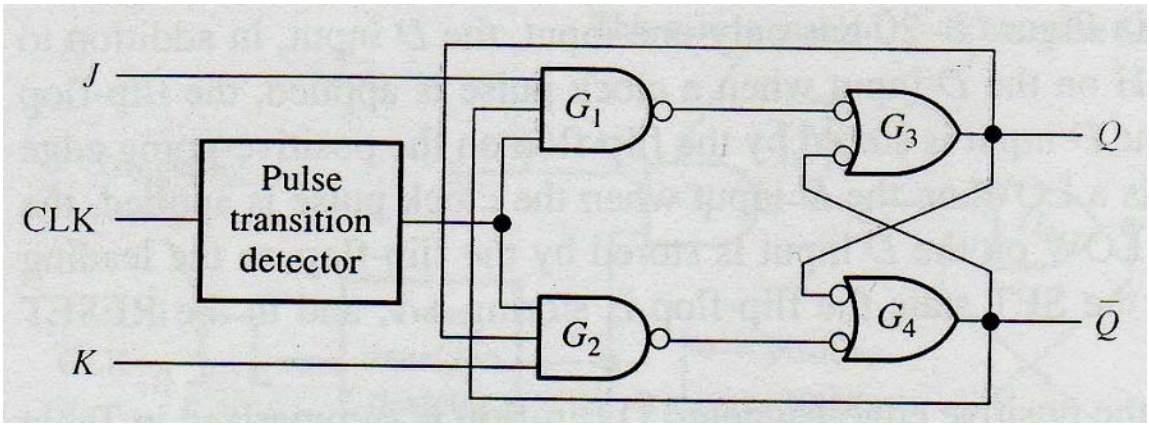


Figure (21-a) JK Flip-flop

Inputs			Outputs		
J	**K**	**CLK**	**Q**	$\overline{Q}$	**Comments**
0	**0**	↑	**Q0**	$\overline{Q}$ **0**	**No change**
0	**1**	↑	**0**	**1**	**Reset**
1	**0**	↑	**1**	**0**	**Set**
1	**1**	↑	$\overline{Q}$ **0**	**Q0**	**Toggle**

Figure (21-b) Truth table for JK flip-flop

#

Lectured Nine

<u>Shift Register:</u> A register is a digital circuit with two basic functions: 1- data storage, 2- data movement.

The storage capability of a register makes it an important type of memory device. The concept of storing a 1 or 0 in a D flip flop. A 1 is applied to the data input, and clock puls is applied that stores the 1 by setting the flip-flop when the 1 on the input is removed, the flip-flop remains in the set state, there by storing the 1. A similar procedure applies to the storage of a 0 by resetting the flip-flop.

<u>Type of shift register:</u>

1- Serial in\ Serial out shift right. 2- Serial in\ Serial out shift left. 3- Parallel in\Serial out.

4- Serial in\Parallel out.

5- Parallel in\ Parallel out. 6 Rotate right.

7- Rotate left.

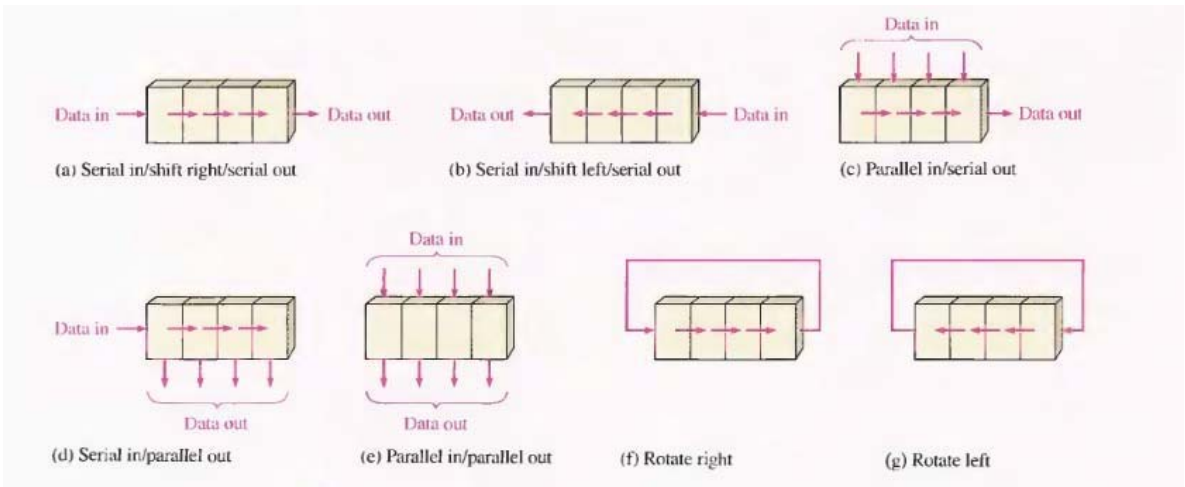


Figure (22) Type of shift register

1- **Serial in \ Serial out shift Register:**

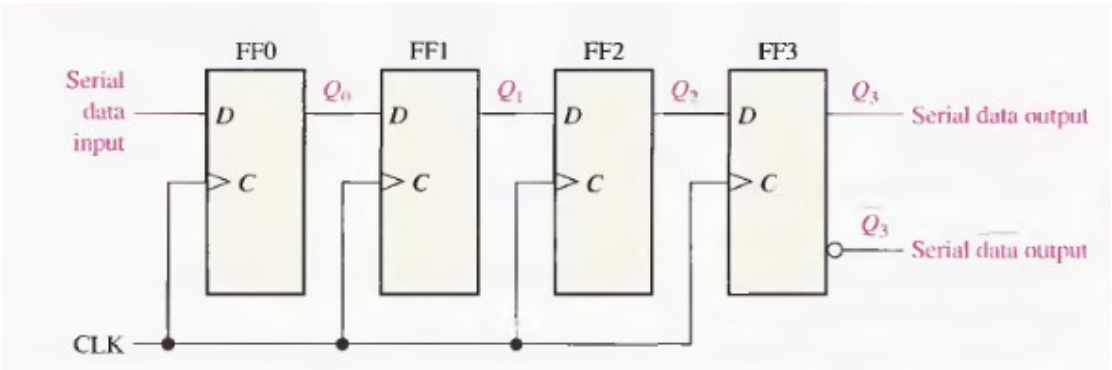


Figure (23) shift register 4-bit

Example: 1

Shift Register 4-bit

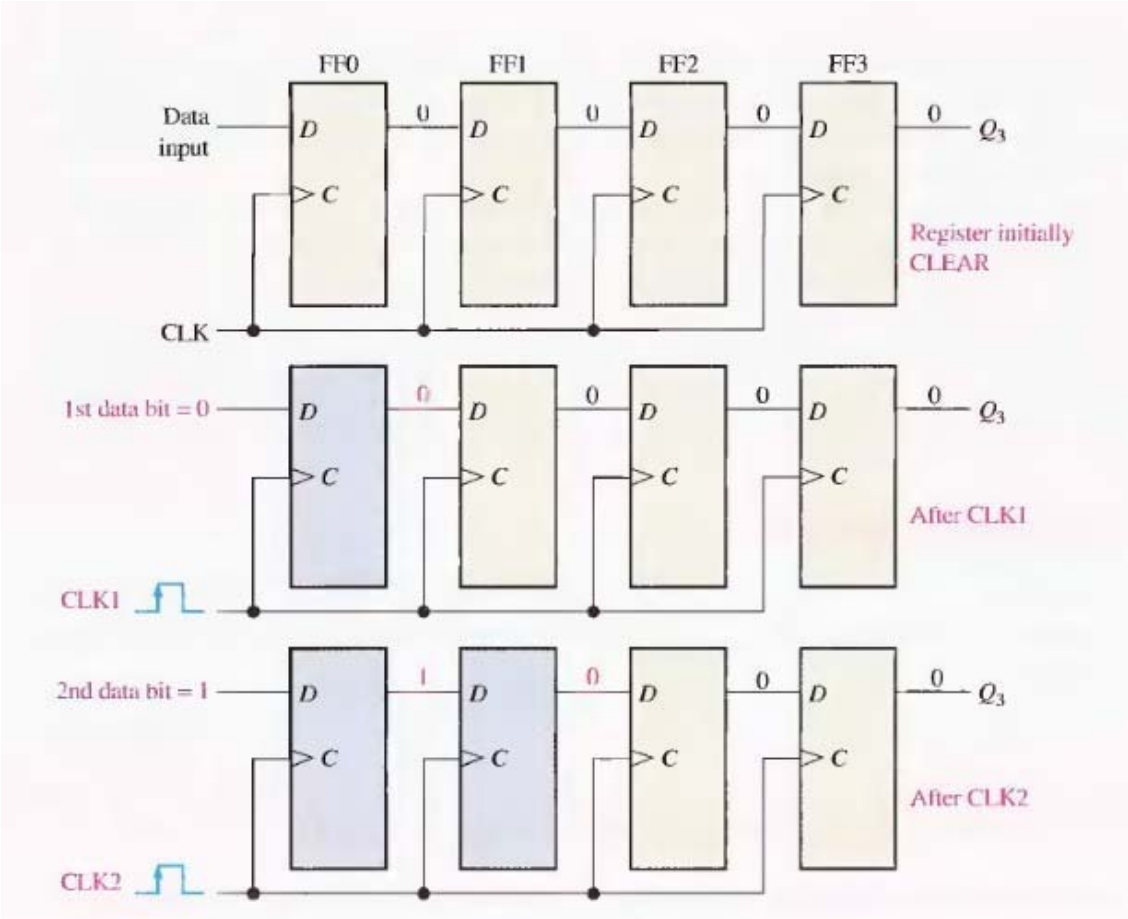


Figure (24-a) 4-bit shift register

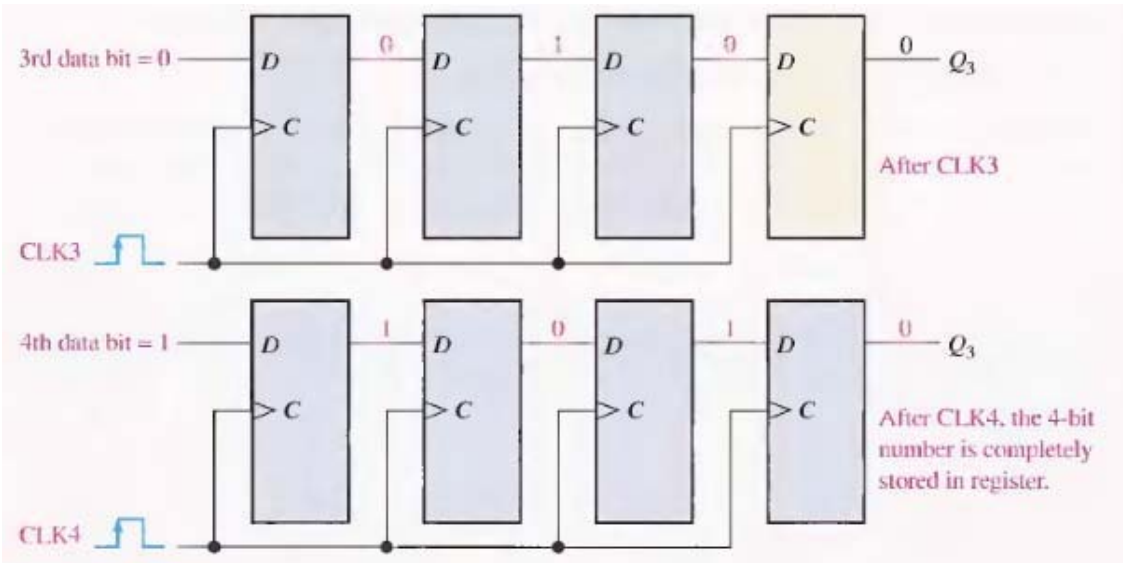


Figure (24-b) 4-bit shift register

<u>Example: 2</u>

Draw 5-bit shift register and write wave form?

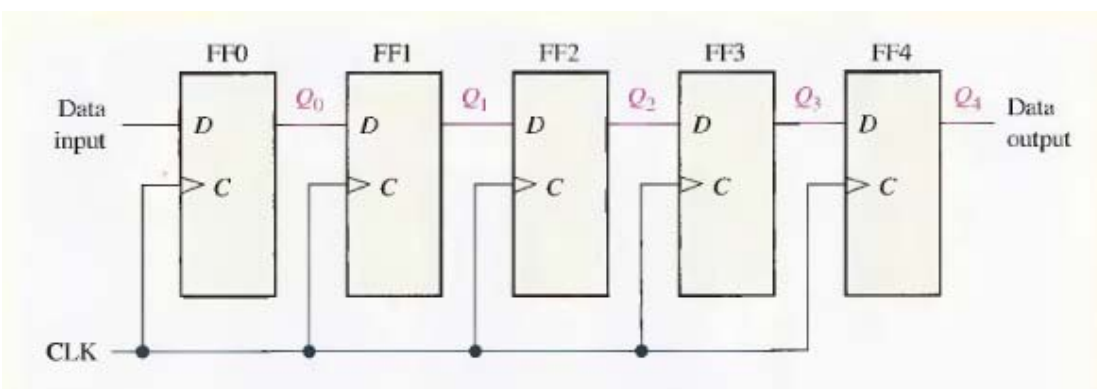


Figure (24-c) 5-bit shift register

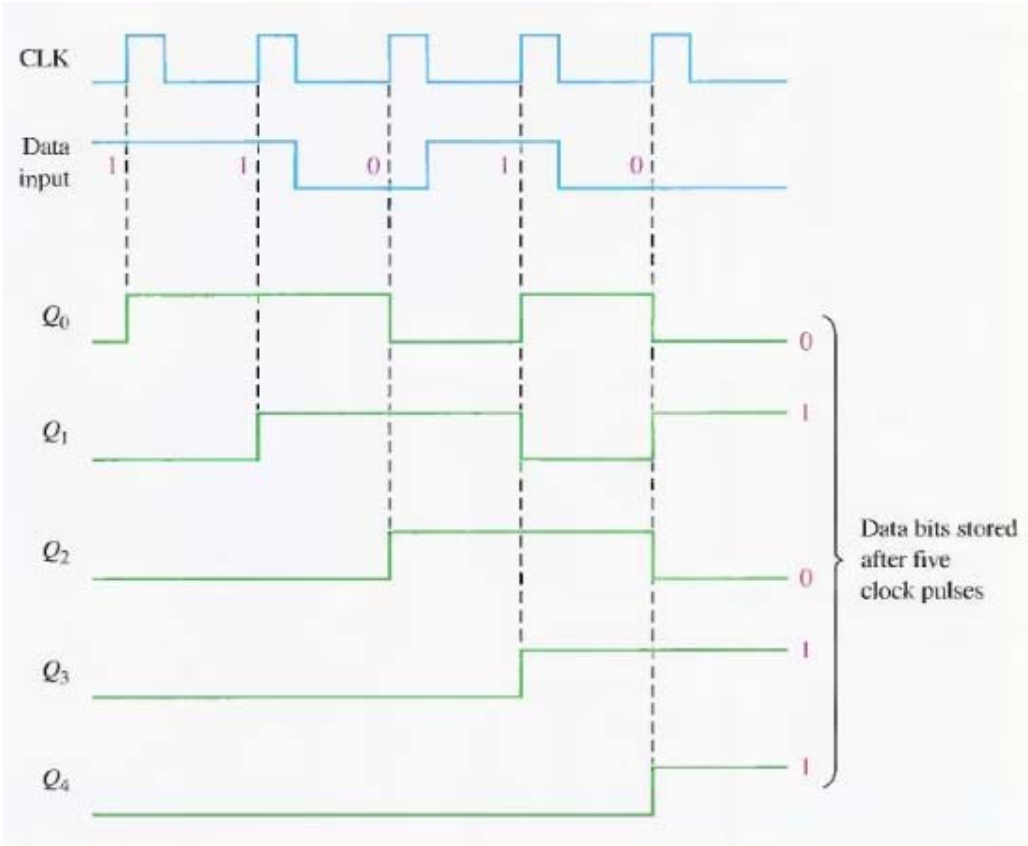


Figure (24-d) 5-bit shift register

#

Lectured Ten

Binary Counter: **The binary counter is consist two types.**

1- Asynchronous counter operation. 2- Synchronous counter operation.

1- Asynchronous counter operation:

In figure (25-a, b, c) shows a 2-bit counter connected for asynchronous operation. Notice that the clock (CLK) is applied to the clock input (C) of only the first flip-flop, FF0, which is always the least significant bit (LSB). The second flip-flop, FF1, is triggered by the Q0 output of FF0. FF0 changes state at the positive-going edge of each clock pulse, but FF1 changes only when triggered by a positive-going transition of the Q0 output of FF0. Because of the inherent propagation delay time through a flip-flop, a transition of the input clock pulse (CLK) and transition of the $\overline{Q0}$ output of FF0 can never occur at exactly the same time. Therefore, the two flip-flops are never simultaneously triggered, so the counter operation is asynchronous.

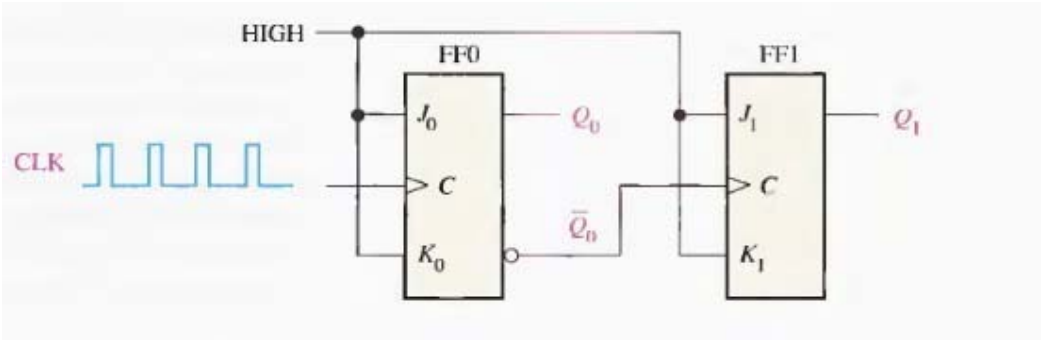


Figure (25-a) 2-Bit Asynchronous Binary Counter

The time diagram

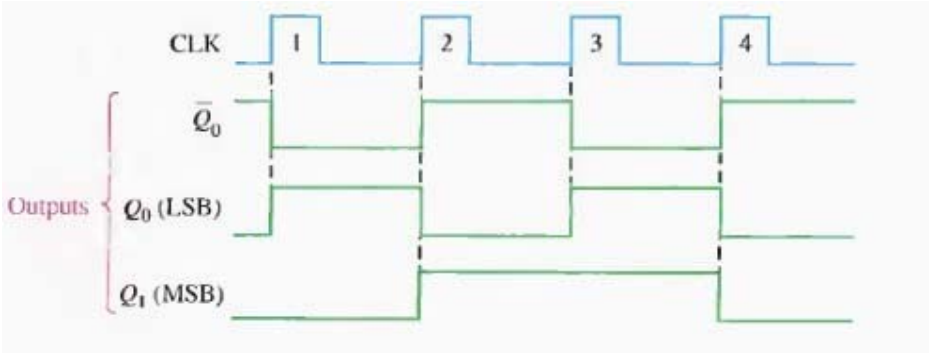


Figure (25-b) Time diagram 2-Bit Asynchronous Binary Counter

CLOCK PULSE	Q_1	Q_0
Initially	0	0
1	0	1
2	1	0
3	1	1
4 (recycles)	0	0

Figure (25-c) Truth table for 2 -Bit Asynchronous Binary Counter

2- <u>Synchronous counter operation:</u>

The term synchronous refers to events that have a fixed time relationship with each other. A synchronous counter is one in which all the flip-flops in the counter are clocked at the same time by a common clock pulse.

A 3-bit synchronous binary counter is shown in figure (26-a) and timing diagram is shown (26-b) you can understand this counter operation by examining its sequence of states as shown in truth table (26-c).

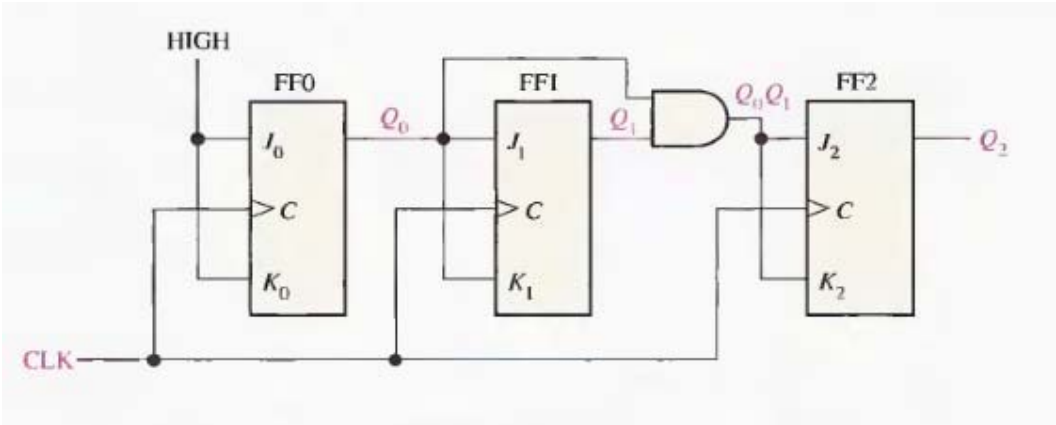


Figure (26-a) 3-Bit Synchronous Binary Counter

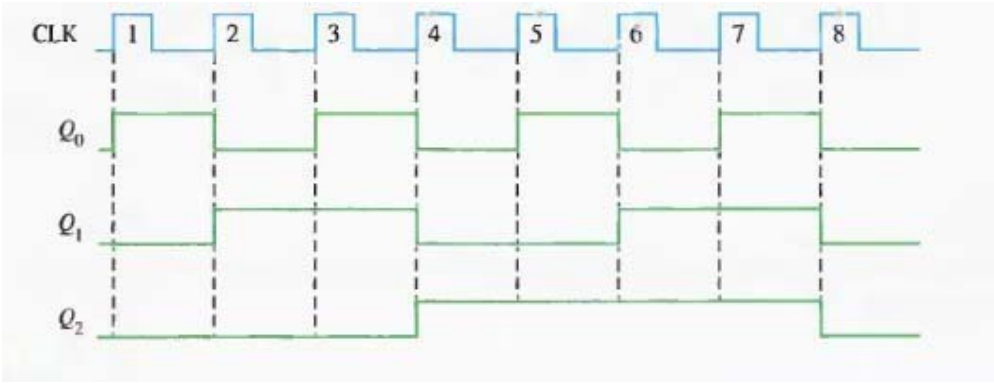


Figure (26-b) Time diagram 3-Bit Synchronous Binary Counter

CLOCK PULSE	Q_2	Q_1	Q_0
Initially	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1
8 (recycles)	0	0	0

Figure (26-c) truth table for 3-Bit Synchronous Binary Counter

<u>Answer these questions:</u>

Q1- Convert the following: 1- $(CF8E)_{16}$ to $(\ )_{10}$

2- $(1725)_{10}$ to $(\qquad)_{16}$

3- $(148.625)_{10}$ to $(\qquad)_{2}$

4- $(7526)_{8}$ to $(\qquad)_{10}$

5-$(2591)_{10)}$ to $(\qquad)_{16}$

6- $(B2F8)_{16}$ to $(\qquad)_{10}$

Q2- Perform the following: 1- $(2AB)_{16} - (317)_{16}$

2- $(101101)_{2} - (1110)_{2}$

3- $(6410)_{8} - (324)_{8}$

4- $(2CF)_{16} - (FDB)_{16}$

5- $(4732)_{8} + (4611)_{8}$

Q3- Express the decimal number -98, -68 as 8-bit number in the sign-magnitude, 1'S and 2'S Complement.

Q4- Design Full-Adder circuit. Q5- Design Half-Adder circuit.

Q6- Draw the 4-bit parallel adder, find the sum and output carry for the addition of the following two 4-bit numbers if the input carry (C_{n-1}) is 0:

A4A3A2A1=1011 and B4B3B2B1=0111.

Q7- use K- map to minimize the following SOP expression and convert to POS in K- map.

$F(A,B,C,D)= \sum 2,3,4,5,6,7,9,12,13,14,15$

Q8- Design 3-to-8 lines decoders. OR Design Binary-to-Octal line decoder.

Q9- Design 2- to – 4 lines decoders

Q10- Design block diagram of quadruple 2-to-1 line multiplexer. Q11- Design SR flip-flop and explain function.

Q12- Design D flip-flop and explain function.

Q13- Design JK flip-flop and explain function.

Q14- Design Octal-to- Binary line encoder.

Q15- Difference between SR and JK flip-flop.

#