الجامعة التقنية الجنوبية
المعهد التقني القرنـــــة
قسم تقنيات انظمة الحاسوب

التصميم المنطقي
Logic Design

محاضرة رقم (1)

د. حيدر احمد عبدالمحسن

Number Systems

- In computers, Number System is defined as a writing system to represent the numbers in different ways i.e., we are using different symbols and notations to represent numbers.
- There are four popular types of Number System - Binary, Octal, Decimal and Hexadecimal.
- The decimal number system is employed in everyday arithmetic to represent numbers by strings of digits. Depending on its position, each digit has an associated value of an integer raised to the power of 10.

$$953.78_{10} = 9 \times 10^2 + 5 \times 10^1 + 3 \times 10^0 + 7 \times 10^{-1} + 8 \times 10^{-2}$$

- Similarly, for binary (base 2) numbers, each binary digit is multiplied by the appropriate power of 2.

$$1011.11_2 = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2}$$
$$= 8 + 0 + 2 + 1 + \tfrac{1}{2} + \tfrac{1}{4} = 11\tfrac{3}{4} = 11.75_{10}$$

Positional Systems

Number Systems

- Any positive integer R (R > 1) can be chosen as the radix or base of a number system.
- If the base is R, then R digits (0,1, . . . , R − 1) are used. For example, if R = 8, then the required digits are 0,1,2,3,4,5,6, and 7.
- A number written in positional notation can be expanded in a power series in R.

$$\begin{aligned} N &= (a_4a_3a_2a_1a_0.a_{-1}a_{-2}a_{-3})_R \\ &= a_4 \times R^4 + a_3 \times R^3 + a_2 \times R^2 + a_1 \times R^1 + a_0 \times R^0 \\ &\quad + a_{-1} \times R^{-1} + a_{-2} \times R^{-2} + a_{-3} \times R^{-3} \end{aligned}$$

where a_i is the coefficient of R^i and $0 \le a_i \le R - 1$.

- If the arithmetic indicated in the power series expansion is done in base 10, then the result is the decimal equivalent of N. For example,

$$\begin{aligned} 147.3_8 &= 1 \times 8^2 + 4 \times 8^1 + 7 \times 8^0 + 3 \times 8^{-1} = 64 + 32 + 7 + \tfrac{3}{8} \\ &= 103.375_{10} \end{aligned}$$

- (243.01) is base 4 number.
  - True
  - False
- Write the following numbers in power expansion form:
  - $(312.4)_5$
  - $(110101.11)_2$
  - $(127.4)_8$
  - $(B65F)_{16}$
  - $(544.1)_9$
  - $(A52.A4)_{11}$
- Count form $(2)_{10}$ to $(15)_{10.}$
- Count form $(2)_3$ to $(102)_{3.}$
- Count form $(2)_7$ to $(25)_{7.}$

Questions

Number Systems

- The power series expansion can be used to convert to any base. For example, converting 147_{10} to base 3 would be written as.

$$147_{10} = 1 \times (10)^2 + 4 \times (10)^1 + 7 \times (10)^0$$

$$147_{10} = 1 \times (101)^2 + (11) \times (101)^1 + (21) \times (101)^0$$

where all the numbers on the right-hand side are base 3 numbers (Note: In base 3, 10 is 101, 7 is 21, etc.). To complete the conversion, base 3 arithmetic would be used.

- Similarly, if 147_{10} is being converted to binary, the calculation would be.

$$147_{10} = 1 \times (1010)^2 + (100) \times (1010)^1 + (111) \times (1010)^0$$

- For bases greater than 10, more than 10 symbols are needed to represent the digits. For example, in hexadecimal (base 16), A represents 10, B represents 11, C represents 12, D represents 13, E represents 14, and F represents 15. Thus,

$$A2F_{16} = 10 \times 16^2 + 2 \times 16^1 + 15 \times 16^0 = 2560 + 32 + 15 = 2607_{10}$$

Conversion form Other Bases to Decimal

- The power series expansion is used to convert form any base to decimal
- $(312.4)_5$ → (? $)_{10.}$

$$(312.4)_5 = 3 \times 5^2 + 1 \times 5^1 + 2 \times 5^0 + 4 \times 5^{-1}$$
$$= 75 + 5 + 2 + 0.8 = (82.8)_{10}$$

- $(11010)_2$ → (? $)_{10.}$

$$(11010)_2 = 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 = (26)_{10}$$

- $(127.4)_8$ → (? $)_{10.}$

$$(127.4)_8 = 1 \times 8^2 + 2 \times 8^1 + 7 \times 8^0 + 4 \times 8^{-1} = (87.5)_{10}$$

- $(B65F)_{16}$ → (? $)_{10.}$

$$(B65F)_{16} = 11 \times 16^3 + 6 \times 16^2 + 5 \times 16^1 + 15 \times 16^0 = (46687)_{10}$$

Numbers with Different Bases

Decimal (base 10)	Binary (base 2)	Octal (base 8)	Hexadecimal (base 16)
00	0000	00	0
01	0001	01	1
02	0010	02	2
03	0011	03	3
04	0100	04	4
05	0101	05	5
06	0110	06	6
07	0111	07	7
08	1000	10	8
09	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

- List the binary, octal, and hexadecimal numbers from $(16)_{10}$ to $(31)_{10}$.
- Convert the following to decimal
  - $(13.3)_4$
  - $(15.7)_8$
  - $(10110.011)_2$
  - $(1FE3.A1)_{16}$
  - $(56.53)_7$
  - $(23.4)_5$
- Determine the radix r for each of the following:
  - $(365)_r = (194)_{10}$
  - $(1726)_r = (982)_{10}$
  - $(551)_r = (361)_{10}$

Questions

الجامعة التقنية الجنوبية
المعهد التقني القرنـــــة
قسم تقنيات انظمة الحاسوب

التصميم المنطقي
Logic Design

محاضرة رقم (2)

د. حيدر احمد عبدالمحسن

Conversion form Decimal to Other Bases

- The conversion of a decimal integer to a number in base r is done by dividing the number and all successive quotients by r and accumulating the remainders (i.e., the division method).

Convert decimal 153 to octal	Convert decimal 41 to binary
$153/8 = 19 + 1/8$ Remainder $= 1$	$41/2 = 20 + 1/2$ Remainder $= 1$
$19/8 = 2 + 3/8$ $= 3$	$20/2 = 10$ $= 0$
$2/8 = 0 + 2/8$ $= 2$	$10/2 = 5$ $= 0$
$(153)_{10} = (231)_8$	$5/2 = 2 + 1/2$ $= 1$
	$2/2 = 1$ $= 0$
	$1/2 = 0 + 1/2$ $= 1$
	$(41)_{10} = (101001)_2$

Conversion form Decimal to Other Bases

- The conversion of a decimal fraction to base r is accomplished by a method similar to that used for integers, except that multiplication by r is used instead of division, and integers are accumulated instead of remainders.

Convert decimal 0.6875 to binary	Convert decimal 0.513 to octal
$0.6875 \times 2 = 1.3750$ Integer $= 1$	$0.513 \times 8 = 4.104$ Integer $= 4$
$0.3750 \times 2 = 0.7500$ $= 0$	$0.104 \times 8 = 0.832$ $= 0$
$0.7500 \times 2 = 1.5000$ $= 1$	$0.832 \times 8 = 6.656$ $= 6$
$0.5000 \times 2 = 1.0000$ $= 1$	$0.565 \times 8 = 5.248$ $= 5$
$(0.6875)_{10} = (0.1011)_2$	

Questions

- Convert 0.625_{10} to binary.
- Convert 0.7_{10} to binary.
- Convert 356.89_{10} to hexadecimal.
- Convert 231.3_4 to base 7.
- Convert $3BA.25_{14}$ to base 6.
- Convert the following to hexadecimal
  - $(757.25)_{10}$
  - $(123.17)_{10}$
  - $(1063.5)_{10}$

Conversion form Binary to Octal and Hexadecimal

- The conversion from binary to octal is easily accomplished by partitioning the binary number into groups of three digits each, starting from the binary point and proceeding to the left and to the right.

$$\left(\underbrace{10}_{2}\ \underbrace{110}_{6}\ \underbrace{001}_{1}\ \underbrace{101}_{5}\ \underbrace{011}_{3}\bullet\underbrace{111}_{7}\ \underbrace{100}_{4}\ \underbrace{000}_{0}\ \underbrace{110}_{6}\right)_2=(26153.7406)_8$$

- Conversion from binary to hexadecimal is similar, except that the binary number is divided into groups of four digits.

$$\left(\underbrace{10}_{2}\ \underbrace{1100}_{C}\ \underbrace{0110}_{6}\ \underbrace{1011}_{B}\bullet\underbrace{1111}_{F}\ \underbrace{0010}_{2}\right)_2=(2C6B.F2)_{16}$$

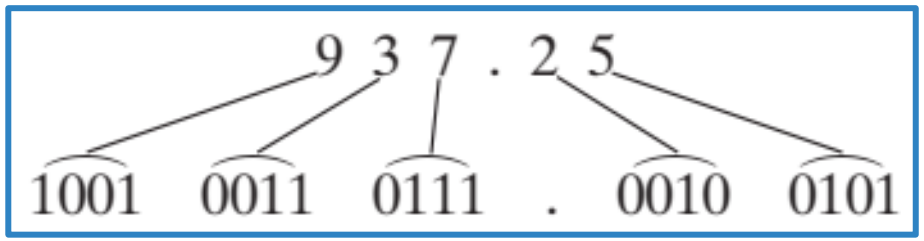
- Conversion from octal or hexadecimal to binary is done by a procedure reverse to the above.

$$(673.124)_8=\left(\underbrace{110}_{6}\ \underbrace{111}_{7}\ \underbrace{011}_{3}\bullet\underbrace{001}_{1}\ \underbrace{010}_{2}\ \underbrace{100}_{4}\right)_2$$

$$(306.\ D)_{16}=\left(\underbrace{0011}_{3}\ \underbrace{0000}_{0}\ \underbrace{0110}_{6}\bullet\underbrace{1101}_{D}\right)_2$$

Binary-Coded Decimal (BCD)

- Although most large computers work internally with binary numbers, the input-output equipment generally uses decimal numbers.
- Because most logic circuits only accept two-valued signals, the decimal numbers must be coded in terms of binary signals.
- In the simplest form of binary code, each decimal digit is replaced by its binary equivalent. For example, 937.25 is represented by



- This representation is referred to as binary-coded-decimal (BCD) or more explicitly as 8-4-2-1 BCD.

Decimal Symbol	BCD Digit
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

- Convert to octal
  - $(111010110001.011)_2$
  - $(10110011101.11)_2$
  - $(110111011.00101)_2$
  - $(10111101.101)_2$
  - $(10101110.1001)_2$
- Convert to hexadecimal
  - $(111010110001.011)_2$
  - $(10110011101.11)_2$
  - $(110111011.00101)_2$
  - $(10111101.101)_2$
  - $(10101110.1001)_2$
- Find the binary for each of the following BCD numbers:
  - $(0100\ 1000\ 0110\ 0111)_{BCD}$
  - $(0011\ 0111\ 1000.0111\ 0101)_{BCD}$

Questions

الجامعة التقنية الجنوبية
المعهد التقني القرنـــــة
قسم تقنيات انظمة الحاسوب

التصميم المنطقي
Logic Design

محاضرة رقم (3)

د. حيدر احمد عبدالمحسن

Binary Arithmetic

- **Arithmetic operations in digital systems are usually done in binary because design of logic circuits to perform binary arithmetic is much easier than for decimal.**
- **The addition table for binary numbers is**

$0 + 0 = 0$
$0 + 1 = 1$
$1 + 0 = 1$
$1 + 1 = 0$ and carry 1 to the next column

$$\begin{array}{rl} & 1111 \longleftarrow \text{carries} \\ 13_{10} = & 1101 \\ 11_{10} = & 1011 \\ \hline & 11000 = 24_{10} \end{array}$$

- The subtraction table for binary numbers is

$0 - 0 = 0$
$0 - 1 = 1$ and borrow 1 from the next column
$1 - 0 = 1$
$1 - 1 = 0$

(a)
$$\begin{array}{rl} & 1 \longleftarrow \text{(indicates} \\ 11101 & \text{a borrrow} \\ -\,10011 & \text{from the} \\ \hline 1010 & \text{3rd column)} \end{array}$$

(b)
$$\begin{array}{rl} & 1111 \longleftarrow \text{borrows} \\ 10000 & \\ -\quad 11 & \\ \hline 1101 & \end{array}$$

- Add and subtract the following in binary:
  - 1111 and 1010
  - 110110 and 11101
  - 100100 and 10110
- Perform the addition $159F2_{16}$ + $1E462_{16}$
- Perform the subtraction $(762)_8$ - $(45)_8$

Questions

Binary Arithmetic

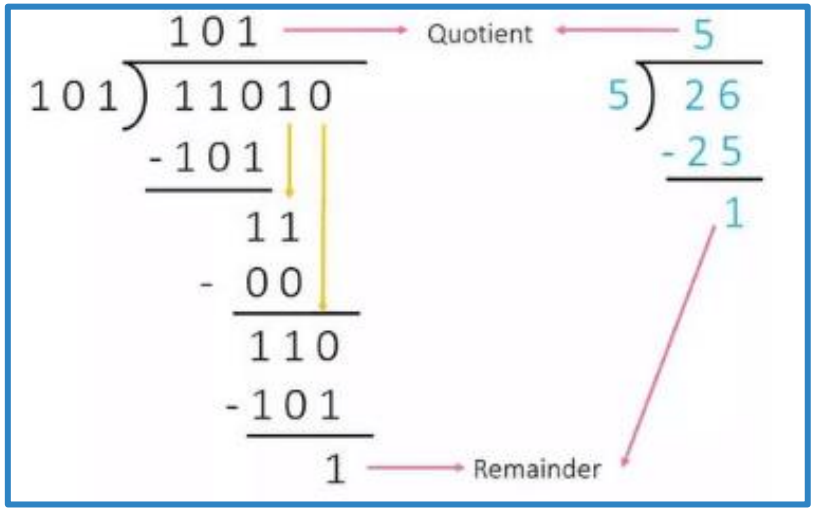
- Arithmetic operations with binary numbers follow the same rules as for decimal numbers.
- The multiplication table for binary numbers is

$$0 \times 0 = 0$$
$$0 \times 1 = 0$$
$$1 \times 0 = 0$$
$$1 \times 1 = 1$$

$$\begin{array}{r} 1101 \\ 1011 \\ \hline 1101 \\ 1101 \\ 0000 \\ 1101 \\ \hline 10001111 \end{array} = 143_{10}$$

$$\begin{array}{r} 1011 \\ \times\ 101 \\ \hline 1011 \\ 0000 \\ 1011 \\ \hline 110111 \end{array}$$

- The following example illustrates division of 26_{10} by 5_{10} in binary:



- Perform the following binary multiplications:
  - 1010 * 1100
  - 0110 * 1001
  - 1111001 * 011101
- Perform the following binary division:
  - 11101001 ÷ 101
  - 110000001 ÷ 1110
  - 1110010 ÷ 1001
  - 10001101 ÷ 110
  - 110000011 ÷ 1011
  - 1110100 ÷ 1010
  - 1010110 ÷ 101
- Perform the multiplication $(62)_8 * (45)_8$

Questions

Representation of Negative Numbers

- Up to this point we have been working with unsigned positive numbers.
- The most common methods for representing both positive and negative numbers are sign and magnitude, 2's complement, and 1's complement.
- In each of the above methods, the leftmost bit of a number is 0 for positive and 1 for negative numbers.
- In sign and magnitude, an n-bit number is represented by a sign bit followed by n − 1 bits that represent the magnitude of the number. For example, 0011 represents +3 and 1011 represents −3.

sign	magnitude
0	011
+	3

sign	magnitude
1	011
-	3

- Designing logic circuits to perform arithmetic on sign and magnitude binary numbers is awkward. One method is to convert the numbers into 2's (or 1's) complement and, after performing the arithmetic operation, convert the result back to sign and magnitude.

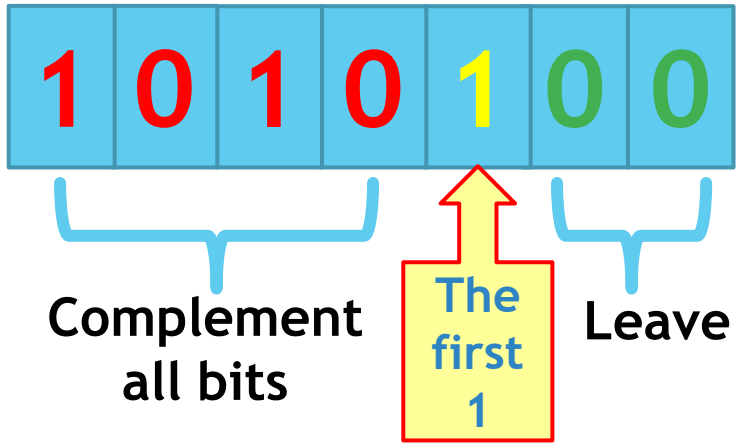
2's Complement Numbers

- In the 2's complement number system, a negative number, −N, is represented by its 2's complement, N*. If the word length is n bits, the 2's complement of a positive integer N is defined as:

$$N^* = 2^n - N \qquad N^* = (2^n - 1 - N) + 1$$

- An alternative way to form the 2's complement of N is to start at the right and leave any 0's on the right end and the first 1 unchanged, then complement all bits to the left of the first 1.
- For example, if n = 7 and N = 0101100,

$$\begin{aligned} 2^n - 1 &= 1111111 \\ &\quad - 0101100 \\ &\quad \overline{1010011} \\ &\quad + 0000001 \\ N^* &= 1010100 \end{aligned}$$



How to find 10's complement of decimal numbers?

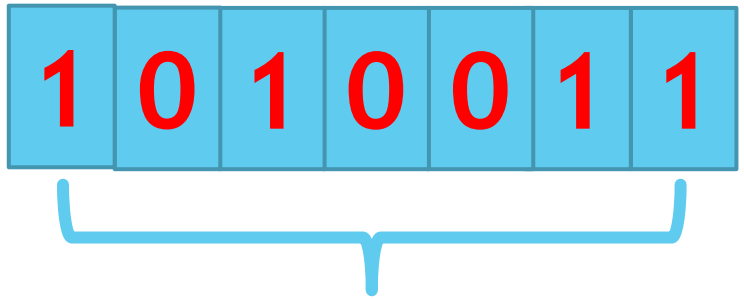
1's Complement Numbers

- In the 1's complement number system, a negative number, −N, is represented by the 1's complement of N, $\overline{N}$, defined as:

$$\overline{N} = (2^n - 1) - N$$

- An alternative way to form the 1's complement of N is to complement all bits bit-by-bit.
- For example, if n = 7 and N = 0101100,

$$\begin{array}{r} 2^n - 1 = 1111111 \\ -\ 0101100 \\ \hline 1010011 \end{array}$$



How to find 9's complement of decimal numbers?

Signed Binary Integers (word length: n = 4)

	Positive Integers		Negative Integers		
$+N$	(all systems)	$-N$	Sign and Magnitude	2's Complement N^*	1's Complement $\overline{N}$
+0	0000	−0	1000	——	1111
+1	0001	−1	1001	1111	1110
+2	0010	−2	1010	1110	1101
+3	0011	−3	1011	1101	1100
+4	0100	−4	1100	1100	1011
+5	0101	−5	1101	1011	1010
+6	0110	−6	1110	1010	1001
+7	0111	−7	1111	1001	1000
		−8	——	1000	——

- Obtain the l’s and 2’s complement of the following binary numbers:
  - 1010101
  - 0111000
  - 0000001
  - 10000
  - 00000
- Obtain the 9’s and 10’s complement of the following decimal numbers:
  - 13579
  - 09900
  - 90090
  - 10000
  - 00000

Questions

الجامعة التقنية الجنوبية
المعهد التقني القرنـــــة
قسم تقنيات انظمة الحاسوب

التصميم المنطقي
Logic Design

محاضرة رقم (4)

د. حيدر احمد عبدالمحسن

Subtraction Using Complement

- Complements are used in digital computers for simplifying the subtraction operation.
- There are two types of complements for each base-r system:
  - the r's complement (i.e., 2's complement and 10's complement).
  - the (r - 1)'s complement (i.e., 1's complement and 9's complement).
- The subtraction of two positive numbers (M — N), both of base r, may be done as follows:
  - Add the minuend M to the r's complement of the subtrahend N.
  - Inspect the result obtained in step 1 for an end carry:
    1) If an end carry occurs, discard it.
    2) If an end carry does not occur, take the r's complement of the number obtained in step 1 and place a negative sign in front.

Subtraction with r's Complements

EXAMPLE 1 : Using 10's complement, subtract 72532 – 3250.

$M = 72532$ 72532
$N = 03250$
+
10's complement of $N = 96750$ 96750
end carry → 1 / 69282

answer: 69282

EXAMPLE 2 : Subtract: $(3250 - 72532)_{10}$.

$M = 03250$ 03250
$N = 72532$
+
10's complement of $N = 27468$ 27468
no carry / 30718

answer: – 69282 = – (10's complement of 30718)

EXAMPLE 3 : Use 2's complement to perform $M - N$

(a) $M = 1010100$ 1010100
$N = 1000100$
+
2's complement of $N = 0111100$ 0111100
end carry → 1 / 0010000

answer: 10000

(b) $M = 1000100$ 1000100
$N = 1010100$
+
2's complement of $N = 0101100$ 0101100
no carry / 1110000

answer: – 10000 = – (2's complement of 1110000)

- Perform the subtraction with the following decimal numbers using 10’s complement:
  - 5250 – 321
  - 753 – 864
  - 3570 – 2100
  - 20 – 1000
- Perform the subtraction with the following binary numbers using 2’s complement:
  - 11010- 1101
  - 10010- 10011
  - 11010-10000
  - 100 – 110000

Questions

Subtraction with (r – 1)’s Complement

- **The subtraction of M – N, both positive numbers in base r, may be calculated in the following manner:**
  - **Add the minuend M to the (r – l)’s complement of the subtrahend N.**
  - **Inspect the result obtained in step 1 for an end carry.**
    1) **If an end carry occurs, add 1 to the least significant digit (end-around carry).**
    2) **If an end carry does not occur, take the (r – l)’s complement of the number obtained in step 1 and place a negative sign in front.**

Subtraction with (r - 1)'s Complements

EXAMPLE 4 : Repeat Examples 1 and 2 using 9's complements.

(a) $M = 72532$, $N = 03250$, 9's complement of $N = 96749$

```
      72532
    + 96749
  1 / 69281
  end-around carry  + 1
      69282
```

answer: 69282

(b) $M = 03250$, $N = 72532$, 9's complement of $N = 27467$

```
      03250
    + 27467
no carry  30717
```

answer: – 69282 = – (9's complement of 30717)

EXAMPLE 5 : Repeat Example 3 using l's complement.

(a) $M = 1010100$, $N = 1000100$, l's complement of $N = 0111011$

```
      1010100
    + 0111011
  1 / 0001111
  end-around carry  + 1
      0010000
```

answer: 10000

(b) $M = 1000100$, $N = 1010100$, l's complement of $N = 0101011$

```
      1000100
    + 0101011
no carry  1101111
```

answer: – 10000 = – (l's complement of 1101111)

- Perform the subtraction with the following decimal numbers using 9’s complement:
  - 5250 – 321
  - 753 – 864
  - 3570 – 2100
  - 20 – 1000
- Perform the subtraction with the following binary numbers using 1’s complement:
  - 11010- 1101
  - 10010- 10011
  - 11010-10000
  - 100 – 110000

Questions

الجامعة التقنية الجنوبية
المعهد التقني القرنـــــة
قسم تقنيات انظمة الحاسوب

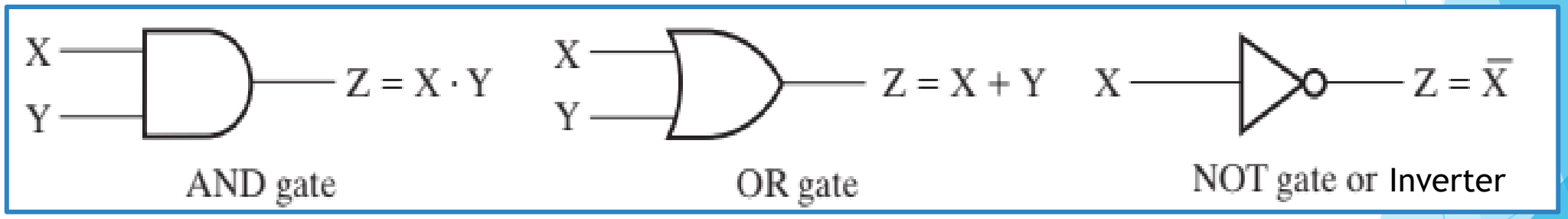
التصميم المنطقي
Logic Design

محاضرة رقم (5)

د. حيدر احمد عبدالمحسن

Logic Gates

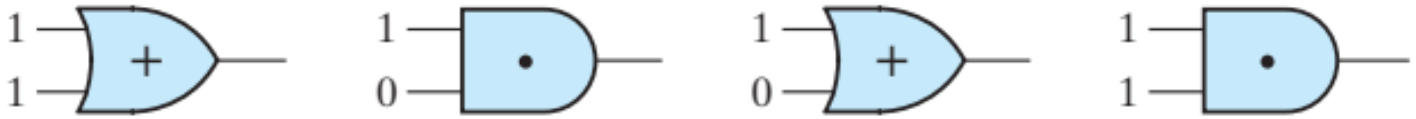
- Logic gates are electronic circuits that operate on one or more input signals to produce an output signal.
- The input terminals of logic gates accept binary signals (logic 1 or logic 0) within the allowable range and respond at the output terminals with binary signals that fall within a specified range.
- The graphics symbols used to designate the basic types of gates—AND, OR, and NOT—are shown bellow:



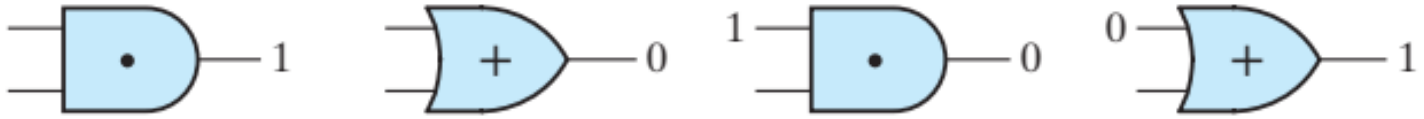
- The AND gate responds with a logic-1 output signal when both input signals are logic 1.
- The OR gate responds with a logic-1 output signal if either input signal is logic 1.
- The NOT gate is more commonly referred to as an inverter.

Questions

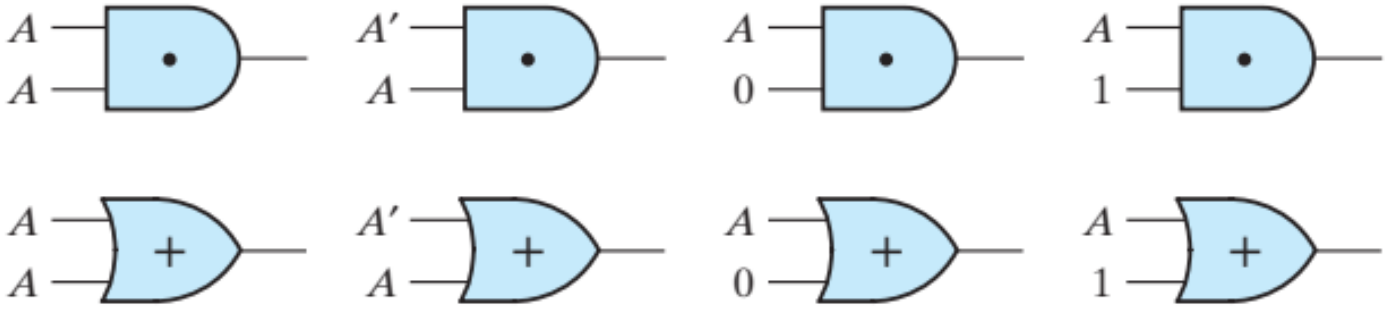
- Determine the output of each of the following gates:



- Determine the unspecified inputs to each of the following gates if the outputs are as shown:

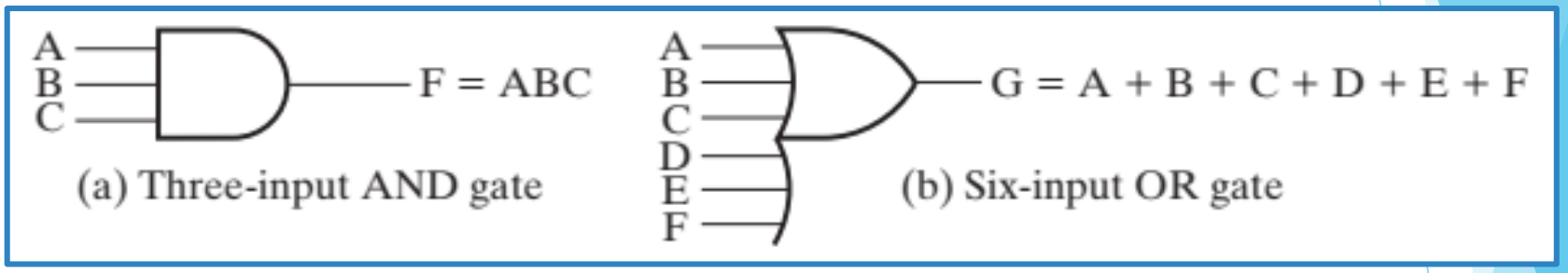


- Determine the output of each of these gates:



Logic Gates

- AND and OR gates may have more than two inputs. An AND gate with three inputs and an OR gate with six inputs are shown:

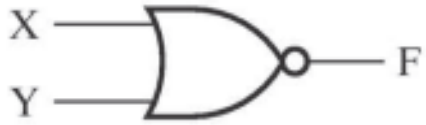
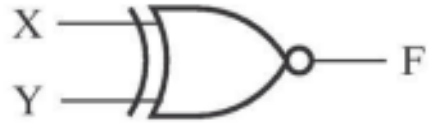


- The three-input AND gate responds with a logic 1 output if all three inputs are logic 1.
- The six-input OR gate responds with a logic 1 if any input is logic 1.
- The NAND gate represents the complement of the AND operation, and the NOR gate represents the complement of the OR operation. Their respective names are abbreviations of NOT-AND and NOT-OR.
- Two other gates that are commonly used are the exclusive-OR (XOR) and exclusive-NOR (XNOR) gates.
- The XOR gate is similar to the OR gate, but excludes (has the value 0 for) the combination with both X and Y equal to 1.

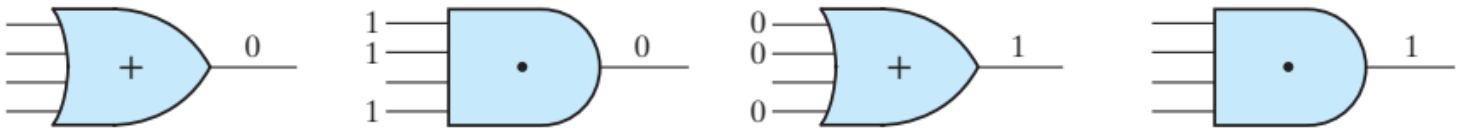
Commonly Used Logic Gates

Name	Distinctive-Shape Graphics Symbol	Algebraic Equation	Truth Table
AND		$F = XY$	X Y | F 0 0 | 0 0 1 | 0 1 0 | 0 1 1 | 1
OR		$F = X + Y$	X Y | F 0 0 | 0 0 1 | 1 1 0 | 1 1 1 | 1
NOT (inverter)		$F = \overline{X}$	X | F 0 | 1 1 | 0
NAND		$F = \overline{X \cdot Y}$	X Y | F 0 0 | 1 0 1 | 1 1 0 | 1 1 1 | 0

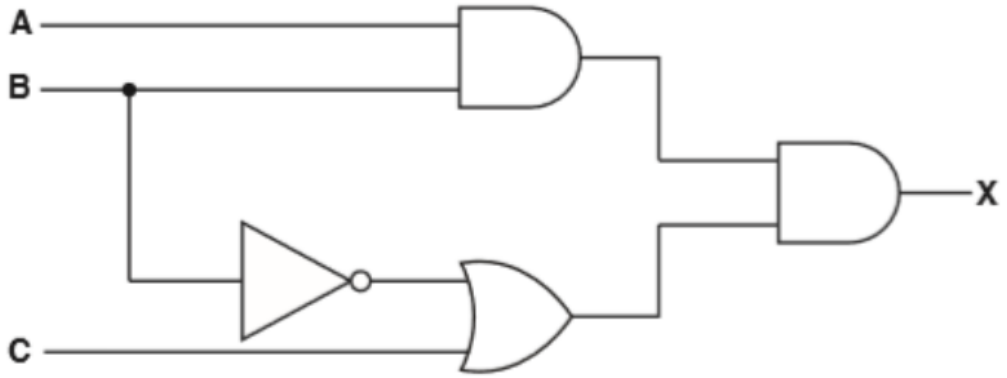
Commonly Used Logic Gates

Name	Distinctive-Shape Graphics Symbol	Algebraic Equation	Truth Table
NOR		$F = \overline{X + Y}$	X Y | F 0 0 | 1 0 1 | 0 1 0 | 0 1 1 | 0
Exclusive-OR (XOR)		$F = X\overline{Y} + \overline{X}Y$ $= X \oplus Y$	X Y | F 0 0 | 0 0 1 | 1 1 0 | 1 1 1 | 0
Exclusive-NOR (XNOR)		$F = XY + \overline{X}\overline{Y}$ $= X \oplus Y$	X Y | F 0 0 | 1 0 1 | 0 1 0 | 0 1 1 | 1

- For each gate determine the value of the unspecified input(s):



- Write a truth table for:
  - three-input OR gate.
  - three-input AND gate.
  - three-input exclusive-OR gate.
- Construct AND Gate using NOR Gate only.
- Write a logic equation and a truth table for the following logic circuit:



Questions

الجامعة التقنية الجنوبية
المعهد التقني القرنـــــة
قسم تقنيات انظمة الحاسوب

التصميم المنطقي
Logic Design

محاضرة رقم (6)

د. حيدر احمد عبدالمحسن

Boolean Algebra

- The basic mathematics needed for the study of logic design of digital systems is Boolean algebra.
- George Boole developed Boolean algebra in 1847 and used it to solve problems in mathematical logic.
- We will use a Boolean variable, such as X or Y, to represent the input or output of a logic circuit where each of these variables can take on only two different values, “0” and “1”.
- A Boolean expression is an algebraic expression formed by using Boolean (binary) variables, the constants 0 and 1, and the logic operations (AND, OR, and NOT). Examples of expressions are:

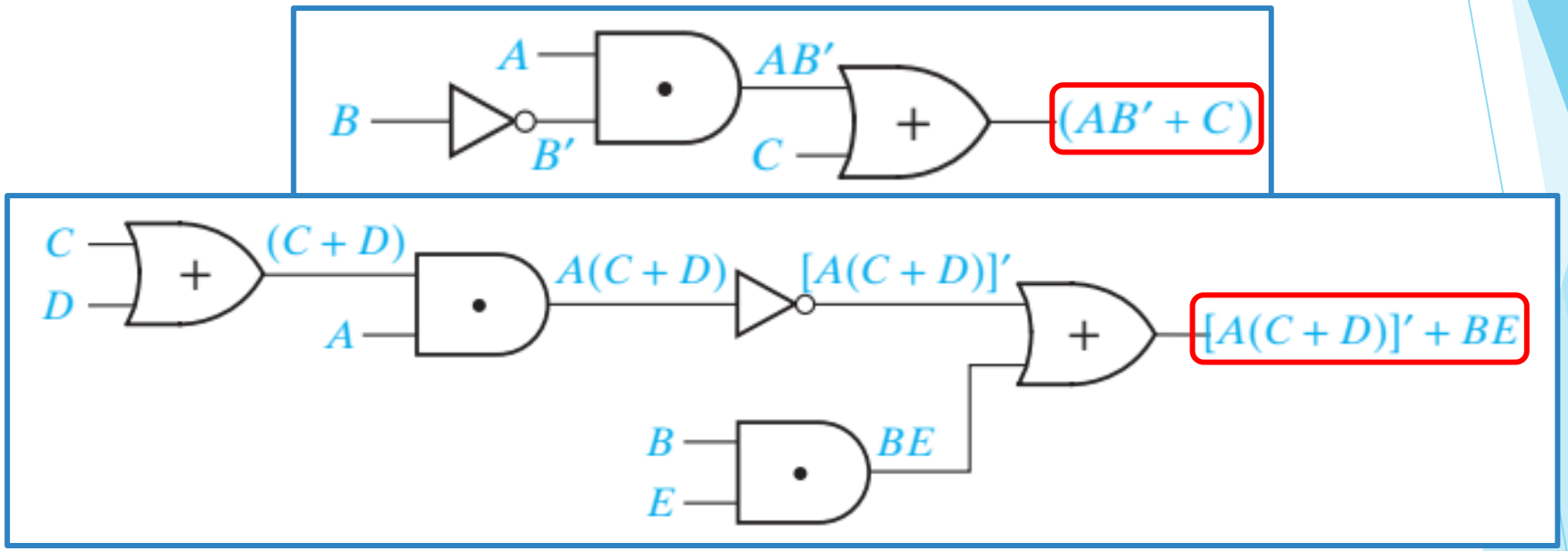
$$AB' + C \qquad (6\text{-}1)$$

$$[A(C + D)]' + BE \qquad (6\text{-}2)$$

- A Boolean expression (equation) expresses the logical relationship between binary variables.
- Each expression corresponds directly to a circuit of logic gates.

Boolean Algebra

- Figures below give the logic circuits for Boolean expressions (equations) (6-1) and (6-2):



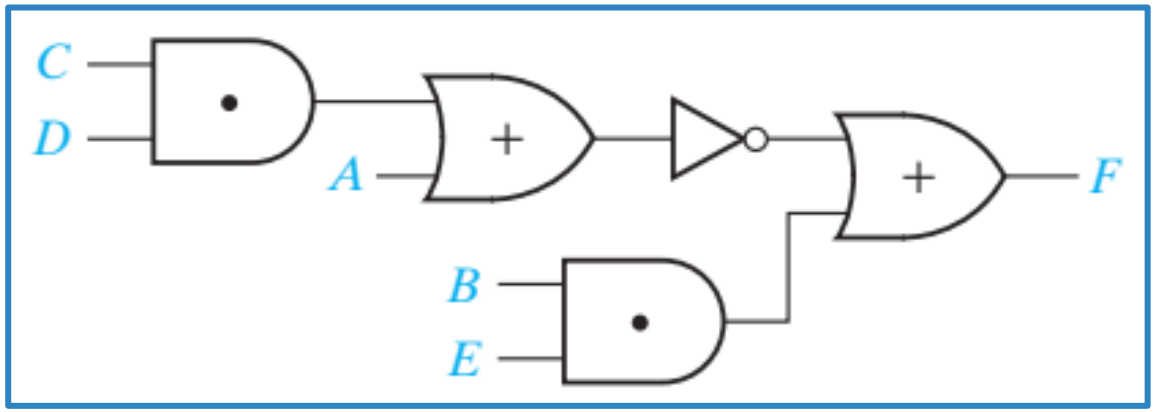
- Expression is evaluated by substituting a value of 0 or 1 for each variable. If A = B = C = 1 and D = E = 0, the value of Expression (6-2) is:

$$[A(C + D)]' + BE = [1(1 + 0)]' + 1 \cdot 0 = [1(1)]' + 0 = 0 + 0 = 0$$

- How many variables does the following Boolean expression contain?

$$A'BC'D + AB + B'CD + D'$$

- For the following circuit, if A = B = 0 and C = D = E = 1, indicate the output of each gate on the circuit diagram:

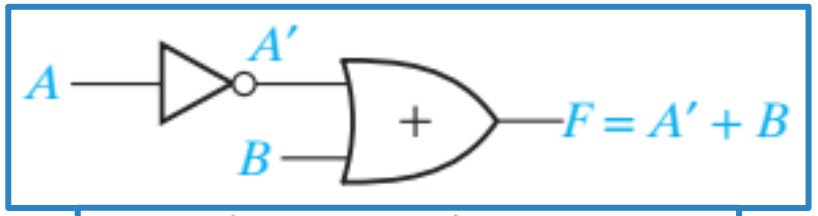


- Derive a Boolean expression for the above circuit output. Then substitute A = B = 0 and C = D = E = 1 into your expression and verify that the value of F obtained on the above circuit diagram.

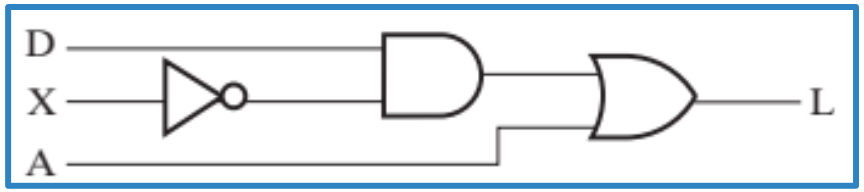
Questions

Truth Table

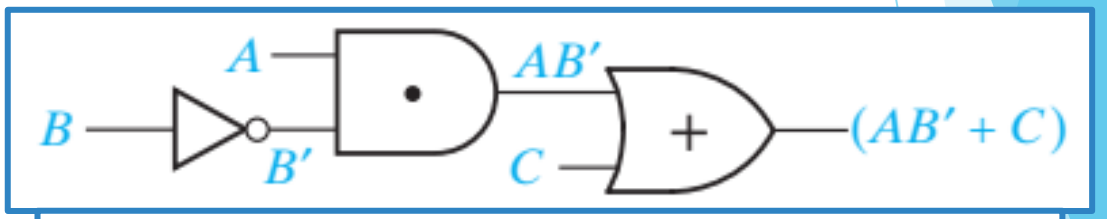
- A truth table for a function is a list of all combinations of 1s and 0s that can be assigned to the binary variables and a list that shows the value of the function for each binary combination.
- The number of rows in a truth table is 2^n, where n is the number of variables in the function.



A	B	A′	F = A′ + B
0	0	1	1
0	1	1	1
1	0	0	0
1	1	0	1



?

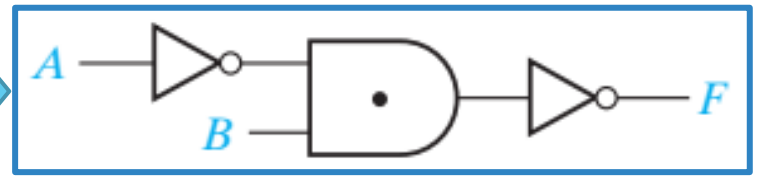


A	B	C	B′	AB′	AB′ + C
0	0	0	1	0	0
0	0	1	1	0	1
0	1	0	0	0	0
0	1	1	0	0	1
1	0	0	1	1	1
1	0	1	1	1	1
1	1	0	0	0	0
1	1	1	0	0	1

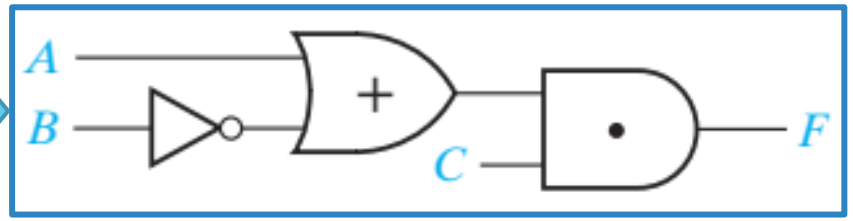
Questions

- Write an expression for the output of the following circuits and write the truth table:

a



b



- Draw a gate circuit which has an output:

$$Z = [BC' + F(E + AD')]'$$

- Write an expression and design a logic circuit for the following truth table:

A	B	C	Output
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

الجامعة التقنية الجنوبية
المعهد التقني القرنـــــة
قسم تقنيات انظمة الحاسوب

التصميم المنطقي
Logic Design

محاضرة رقم (7)

د. حيدر احمد عبدالمحسن

Basic Identities of Boolean Algebra

- Table below lists the most basic identities of Boolean algebra.
- The basic rules listed in the table have been arranged into two columns that demonstrate the property of duality of Boolean algebra (i.e. interchanging OR and AND operations and replacing 1s by 0s and 0s by 1s).

Basic Identities of Boolean Algebra

1.	$X + 0 = X$	2. $X \cdot 1 = X$	
3.	$X + 1 = 1$	4. $X \cdot 0 = 0$	
5.	$X + X = X$	6. $X \cdot X = X$	
7.	$X + \overline{X} = 1$	8. $X \cdot \overline{X} = 0$	
9.	$\overline{\overline{X}} = X$		
10.	$X + Y = Y + X$	11. $XY = YX$	Commutative
12.	$X + (Y + Z) = (X + Y) + Z$	13. $X(YZ) = (XY)Z$	Associative
14.	$X(Y + Z) = XY + XZ$	15. $X + YZ = (X + Y)(X + Z)$	Distributive
16.	$\overline{X + Y} = \overline{X} \cdot \overline{Y}$	17. $\overline{X \cdot Y} = \overline{X} + \overline{Y}$	DeMorgan's

DeMorgan's Theorem

- The last two identities, 16 - 17, are referred to as DeMorgan's theorem.

$$\overline{X + Y} = \overline{X} \cdot \overline{Y} \text{ and } \overline{X \cdot Y} = \overline{X} + \overline{Y}$$

- This is a very important theorem and is used to obtain the complement of an expression and of the corresponding function.
- Table below shows two truth tables that verify the first part of DeMorgan's theorem.

Truth Tables to Verify DeMorgan's Theorem

(a) X	Y	X + Y	$\overline{X + Y}$
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

(b) X	Y	$\overline{X}$	$\overline{Y}$	$\overline{X} \cdot \overline{Y}$
0	0	1	1	1
0	1	1	0	0
1	0	0	1	0
1	1	0	0	0

Verify the second part of Demorgan's theorem using truth table?

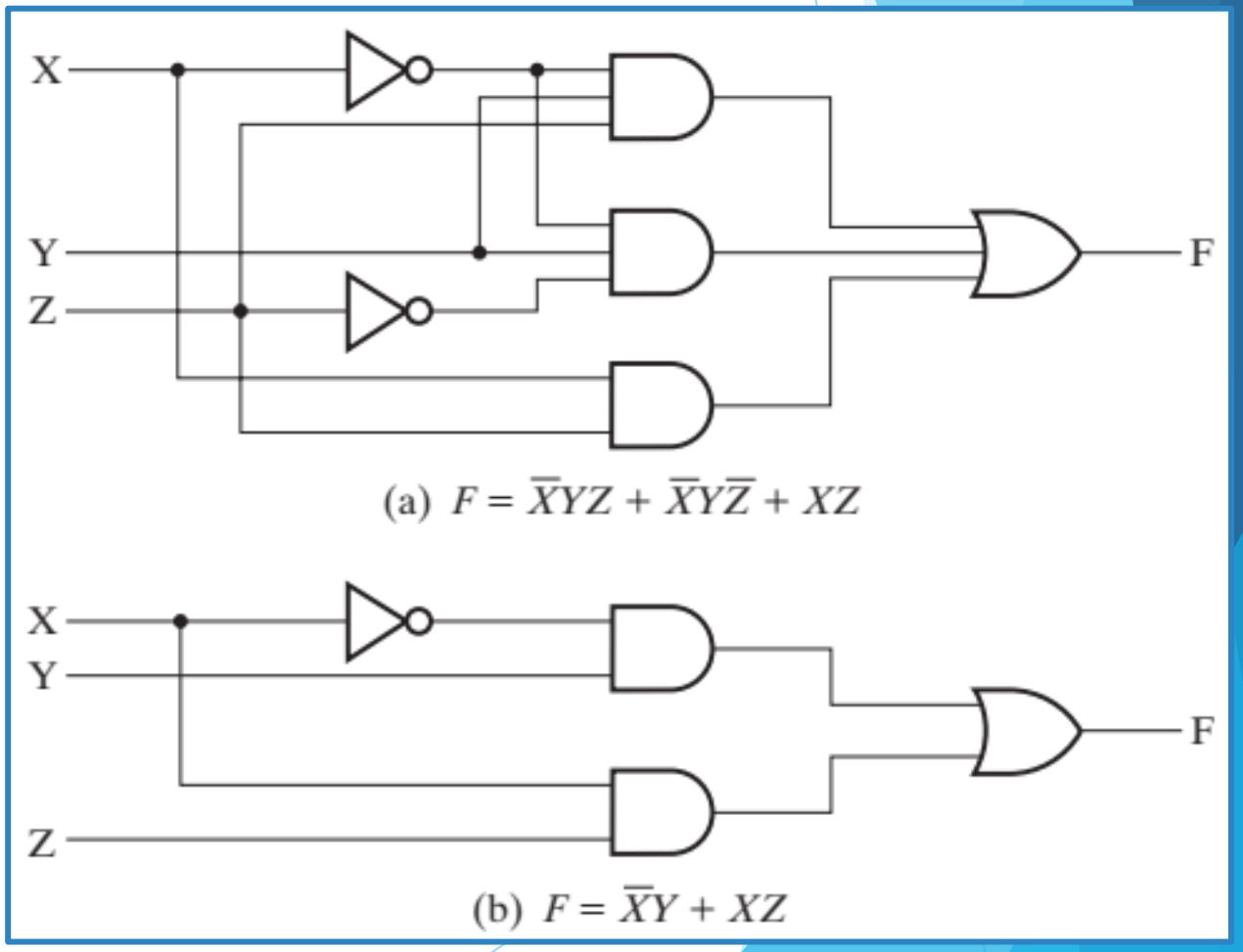
Algebraic Manipulation

- Boolean algebra is a useful tool for simplifying digital circuits. Consider, for example,

$$
\begin{aligned}
F &= \overline{X}YZ + \overline{X}Y\overline{Z} + XZ \\
&= \overline{X}Y(Z + \overline{Z}) + XZ && \text{by identity 14} \\
&= \overline{X}Y \cdot 1 + XZ && \text{by identity 7} \\
&= \overline{X}Y + XZ && \text{by identity 2}
\end{aligned}
$$

- The expression is reduced to only two terms.
- It is obvious that the circuit in Figure (b) is simpler than the one in Figure (a).

Proof that the two circuits are equivalent using truth table?



- Simplify each of the following expressions:

(a) $X'Y'Z + (X'Y'Z)'$

(b) $(AB' + CD)(B'E + CD)$

(c) $ACF + AC'F$

(d) $A(C + D'B) + A'$

(e) $(A'B + C + D)(A'B + D)$

(f) $(A + BC) + (DE + F)(A + BC)'$

- Simplify the following Boolean expressions

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

(b) $\overline{(A + B + C)} \cdot \overline{ABC}$

(c) $AB\overline{C} + AC$

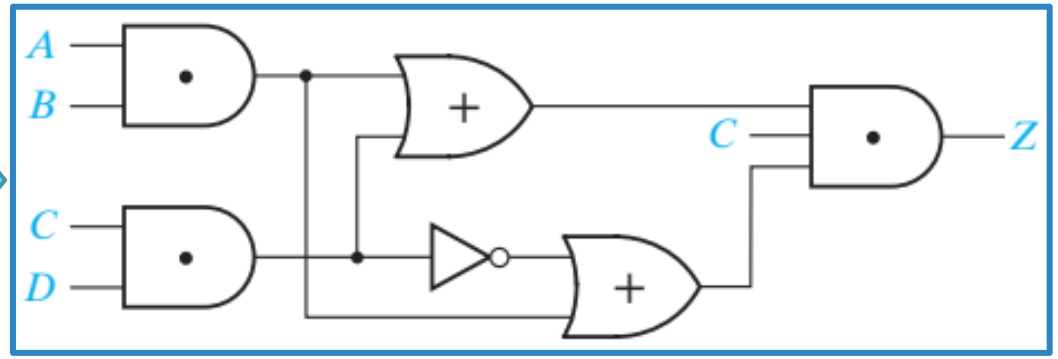
(d) $\overline{A}\,\overline{B}D + \overline{A}\,\overline{C}D + BD$

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

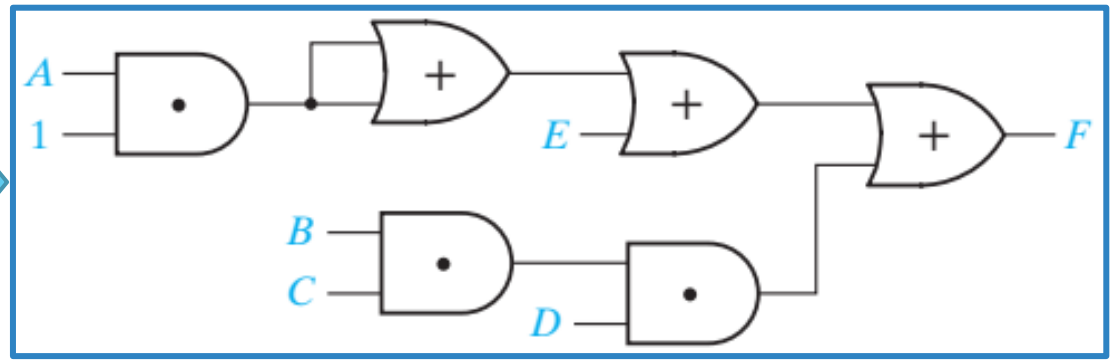
Questions

- Simplify the following circuits:

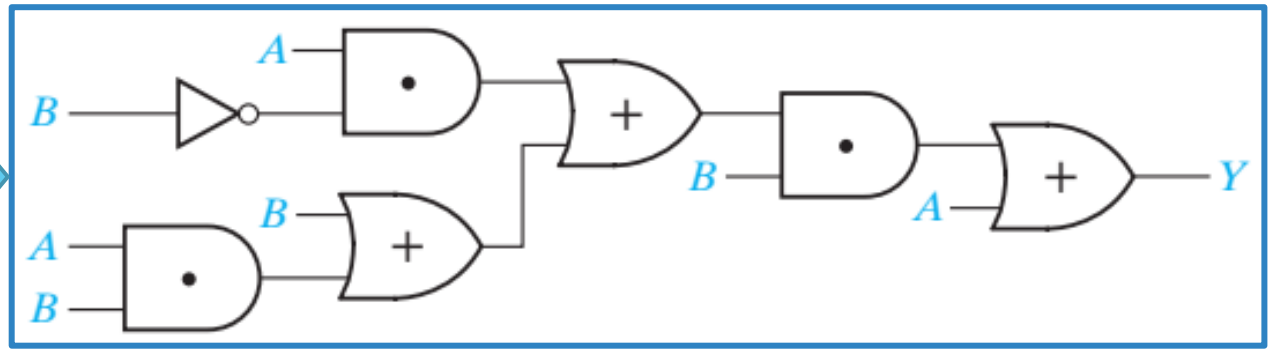
a



b



c



Questions

Complement of a Function

- The complement representation for a function F, $\bar{F}$, is obtained from an interchange of 1s to 0s and 0s to 1s for the values of F in the truth table.
- The complement of a function can be derived algebraically by applying DeMorgan's theorem.
- A simpler method for deriving the complement of a function is to take the dual of the function equation and complement each literal.

$$\bar{F}_1 = \overline{\bar{X}YZ + X\bar{Y}Z} = (\overline{\bar{X}YZ}) \cdot (\overline{X\bar{Y}Z})$$
$$= (X + \bar{Y} + Z)(X + Y + \bar{Z})$$
$$\bar{F}_2 = \overline{X(\bar{Y}Z + Y\bar{Z})} = \bar{X} + \overline{(\bar{Y}Z + Y\bar{Z})}$$
$$= \bar{X} + \overline{\bar{Y}Z} \cdot \overline{Y\bar{Z}}$$
$$= \bar{X} + (Y + Z)(\bar{Y} + \bar{Z})$$

$$F_1 = \bar{X}YZ + \bar{X}\bar{Y}Z = (\bar{X}YZ) + (\bar{X}\bar{Y}Z)$$

The dual of F_1 is

$$(\bar{X} + Y + \bar{Z})(\bar{X} + \bar{Y} + Z)$$

Complementing each literal, we have

$$(X + \bar{Y} + Z)(X + Y + \bar{Z}) = \bar{F}_1$$

- Find the complements of the following functions:

(a) $(ab'c')' =$

(b) $(a' + b + c + d')'$

(c) $(a' + bc)' =$

(d) $(a'b' + cd)' =$

(e) $[a(b' + c'd)]' =$

- Find the complements of the following functions:

(a) $f(A, \bar{B}, C, D) = [A + (BCD)'][(AD)' + B(C' + A)]$

(b) $f(A, B, C, D) = AB'C + (A' + B + D)(ABD' + B')$

Questions

الجامعة التقنية الجنوبية
المعهد التقني القرنـــــة
قسم تقنيات انظمة الحاسوب

التصميم المنطقي
Logic Design

محاضرة رقم (8)

د. حيدر احمد عبدالمحسن

Standard Forms

- A Boolean function expressed algebraically can be written in a variety of ways. There are, however, specific ways of writing algebraic equations that are considered to be standard forms.
- The standard forms contain product terms and sum terms. An example of a product term is XYZ. An example of a sum term is X + Y + Z.
- In Boolean algebra, the words “product” and “sum” do not imply arithmetic operations—instead, they specify the logical operations AND and OR, respectively.
- An algebraic expression for the function can be derived from the table by finding a logical sum of product terms for which the function assumes the binary value 1.
- A product term in which all the variables appear exactly once, either complemented or uncomplemented, is called a minterm. There are 2^n minterms for n variables. The four minterms for the two variables X and Y are $\overline{X}\,\overline{Y}$, $\overline{X}\,Y$, $X\,\overline{Y}$, and XY.

Standard Forms

- The eight minterms for the three variables X, Y, and Z are listed in Table below.

Minterms for Three Variables

X	Y	Z	Product Term	Symbol	m_0	m_1	m_2	m_3	m_4	m_5	m_6	m_7
0	0	0	$\bar{X}\bar{Y}\bar{Z}$	m_0	1	0	0	0	0	0	0	0
0	0	1	$\bar{X}\bar{Y}Z$	m_1	0	1	0	0	0	0	0	0
0	1	0	$\bar{X}Y\bar{Z}$	m_2	0	0	1	0	0	0	0	0
0	1	1	$\bar{X}YZ$	m_3	0	0	0	1	0	0	0	0
1	0	0	$X\bar{Y}\bar{Z}$	m_4	0	0	0	0	1	0	0	0
1	0	1	$X\bar{Y}Z$	m_5	0	0	0	0	0	1	0	0
1	1	0	$XY\bar{Z}$	m_6	0	0	0	0	0	0	1	0
1	1	1	XYZ	m_7	0	0	0	0	0	0	0	1

Standard Forms

- A sum term that contains all the variables in complemented or uncomplemented is called a maxterm.
- a minterm and maxterm with the same subscript are the complements of each other; that is, $M_j = \overline{m_j}$ and $m_j = \overline{M_j}$. For example, for j = 3, we have $M_3 = X + \overline{Y} + \overline{Z} + = \overline{\overline{X}YZ} = \overline{m}_3$

Maxterms for Three Variables

X	Y	Z	Sum Term	Symbol	M_0	M_1	M_2	M_3	M_4	M_5	M_6	M_7
0	0	0	$X + Y + Z$	M_0	0	1	1	1	1	1	1	1
0	0	1	$X + Y + \overline{Z}$	M_1	1	0	1	1	1	1	1	1
0	1	0	$X + \overline{Y} + Z$	M_2	1	1	0	1	1	1	1	1
0	1	1	$X + \overline{Y} + \overline{Z}$	M_3	1	1	1	0	1	1	1	1
1	0	0	$\overline{X} + Y + Z$	M_4	1	1	1	1	0	1	1	1
1	0	1	$\overline{X} + Y + \overline{Z}$	M_5	1	1	1	1	1	0	1	1
1	1	0	$\overline{X} + \overline{Y} + Z$	M_6	1	1	1	1	1	1	0	1
1	1	1	$\overline{X} + \overline{Y} + \overline{Z}$	M_7	1	1	1	1	1	1	1	0

Standard Forms

- A Boolean function can be represented algebraically from a given truth table by forming the logical sum of all the minterms that produce a 1 in the function. This expression is called a sum of minterms.
- Consider the Boolean function F in Table below. The function is equal to 1 for each of the following binary combinations of the variables X, Y, and Z : 000, 010, 101 and 111.
- These combinations correspond to minterms 0, 2, 5, and 7.
- the function F can be expressed algebraically as the logical sum of the stated minterms:

$$F = \overline{X}\,\overline{Y}\,\overline{Z} + \overline{X}Y\overline{Z} + X\overline{Y}Z + XYZ = m_0 + m_2 + m_5 + m_7$$

$$F(X, Y, Z) = \Sigma m(0, 2, 5, 7)$$

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

- Given F_1 and F_2 as bellow, find the minterm expression for $F_1 + F_2$.

$$F_1 = \Sigma\, m(0, 4, 5, 6) \text{ and } F_2 = \Sigma\, m(0, 3, 6, 7)$$

- Obtain the truth table of the following functions, and express each function in sum-of-minterms form:

(a) $(XY + Z)(Y + XZ)$

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

(c) $WX\overline{Y} + WX\overline{Z} + WXZ + Y\overline{Z}$

Questions

Standard Forms

- Now consider the complement of a Boolean function. The binary values of $\overline{F}$ in Table below are obtained by changing 1s to 0s and 0s to 1s in the values of F. Taking the logical sum of minterms of F, we obtain:

$$\overline{F}(X,Y,Z) = \overline{X}\,\overline{Y}Z + \overline{X}YZ + X\overline{Y}\,\overline{Z} + XY\overline{Z} = m_1 + m_3 + m_4 + m_6$$

$$\overline{F}(X, Y, Z) = \Sigma m(1, 3, 4, 6)$$

- We now take the complement of $\overline{F}$ to obtain F:

$$\begin{aligned} F &= \overline{m_1 + m_3 + m_4 + m_6} = \overline{m_1} \cdot \overline{m_3} \cdot \overline{m_4} \cdot \overline{m_6} \\ &= M_1 \cdot M_3 \cdot M_4 \cdot M_6 \,(\text{since } \overline{m_j} = M_j) \\ &= (X + Y + \overline{Z})(X + \overline{Y} + \overline{Z})(\overline{X} + Y + Z)(\overline{X} + \overline{Y} + Z) \end{aligned}$$

X	Y	Z	F	$\overline{F}$
0	0	0	1	0
0	0	1	0	1
0	1	0	1	0
0	1	1	0	1
1	0	0	0	1
1	0	1	1	0
1	1	0	0	1
1	1	1	1	0

- This shows the procedure for expressing a Boolean function as a product of maxterms. The abbreviated form for this product is

$$F(X, Y, Z) = \prod M(1, 3, 4, 6)$$

- For the Boolean functions E and F, as given in the following truth table:

(a) List the minterms and maxterms of each function.

(b) List the minterms of $\overline{E}$ and $\overline{F}$

(c) List the minterms of $E + F$ and $E \cdot F$.

(d) Express E and F in sum-of-minterms algebraic form.

(e) Simplify E and F to expressions with a minimum of literals.

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

Questions

الجامعة التقنية الجنوبية
المعهد التقني القرنـــــة
قسم تقنيات انظمة الحاسوب

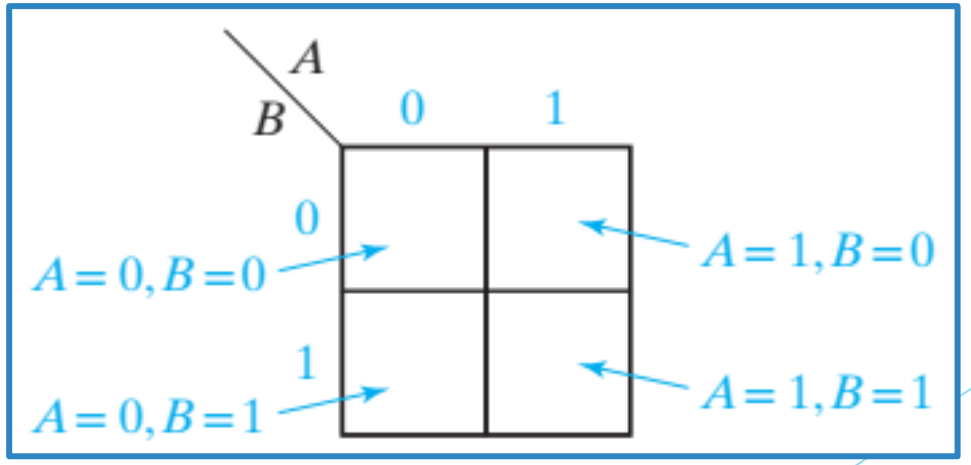
التصميم المنطقي
Logic Design

محاضرة رقم (9)

د. حيدر احمد عبدالمحسن

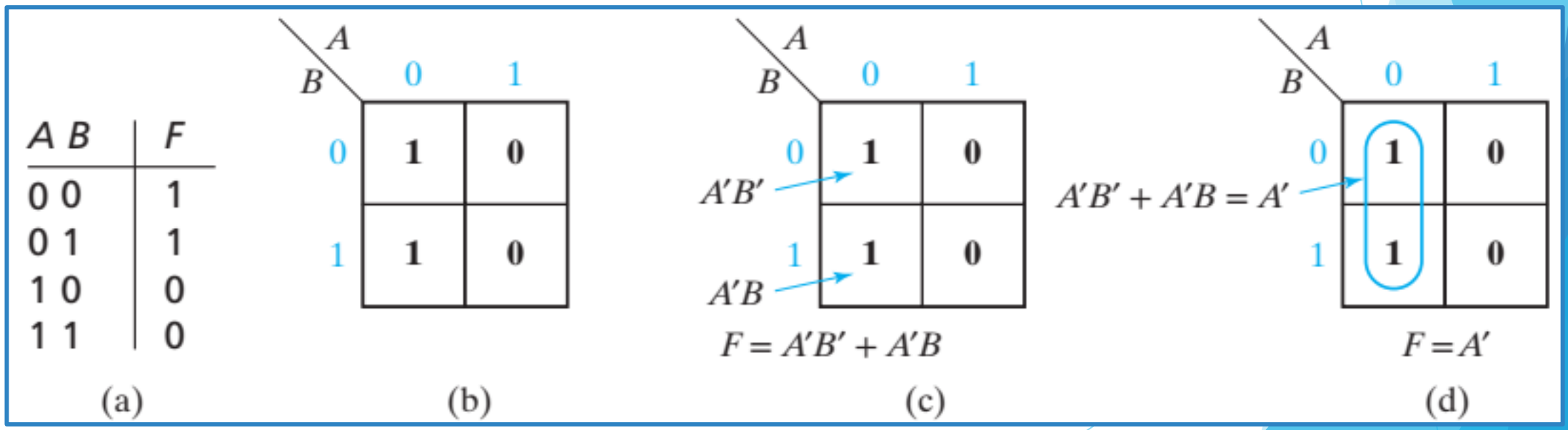
Karnaugh Maps

- The Karnaugh map is an especially useful tool for simplifying and manipulating switching functions of three or four variables, but it can be extended to functions of five or more variables.
- Generally, the Karnaugh map method is faster and easier to apply than other simplification methods.
- Just like a truth table, the Karnaugh map of a function specifies the value of the function for every combination of values of the independent variables.
- A two-variable Karnaugh map is shown below. Each square of the map corresponds to a pair of values for A and B as indicated.



Two-Variable Karnaugh Maps

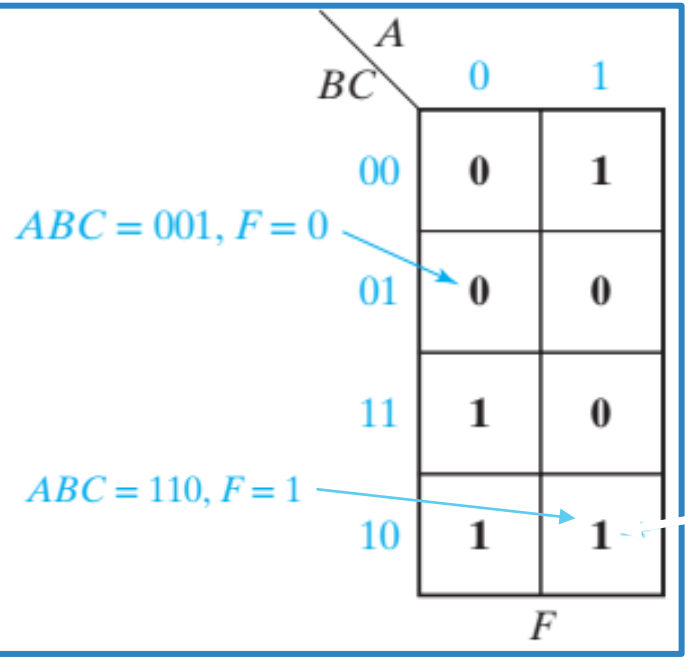
- Figure below shows the truth table for a function F and the corresponding Karnaugh map.
- Each 1 on the map corresponds to a minterm of F.
- Minterms in adjacent squares of the map can be combined since they differ in only one variable.
- Thus, A'B' and A'B combine to form A', and this is indicated by looping the corresponding 1's on the map.

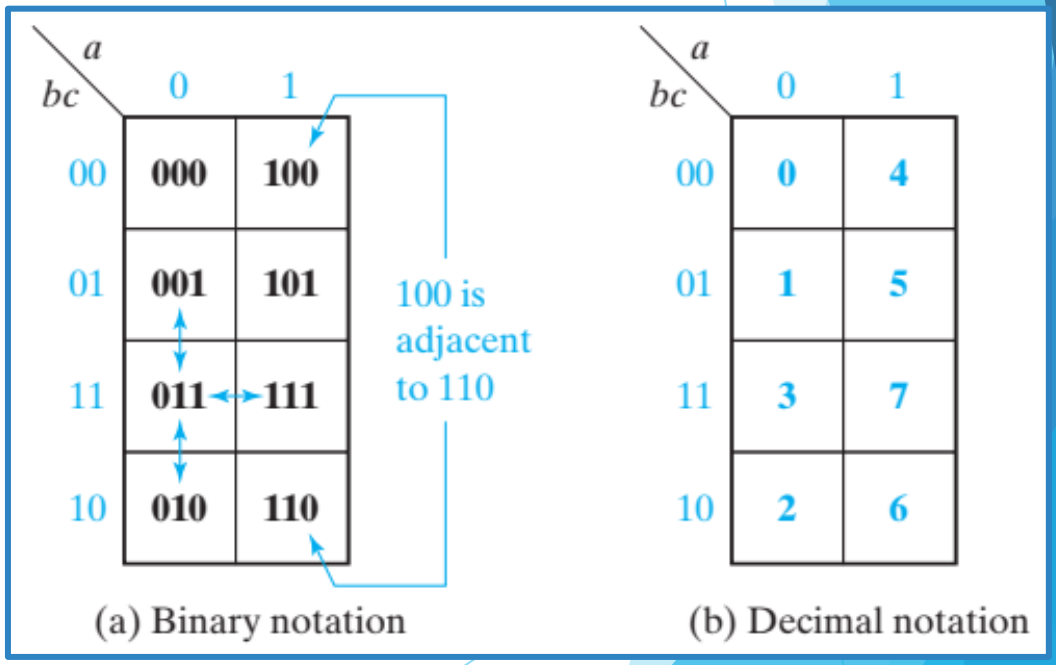


Three-Variable Karnaugh Maps

- Figure on the left shows a three-variable truth table and the corresponding Karnaugh map.
- Figure on the right shows the location of the minterms on a three-variable map. Minterms in adjacent squares of the map differ in only one variable and therefore can be combined using XY' + XY = X.

A B C	F
0 0 0	0
0 0 1	0
0 1 0	1
0 1 1	1
1 0 0	1
1 0 1	0
1 1 0	1
1 1 1	0

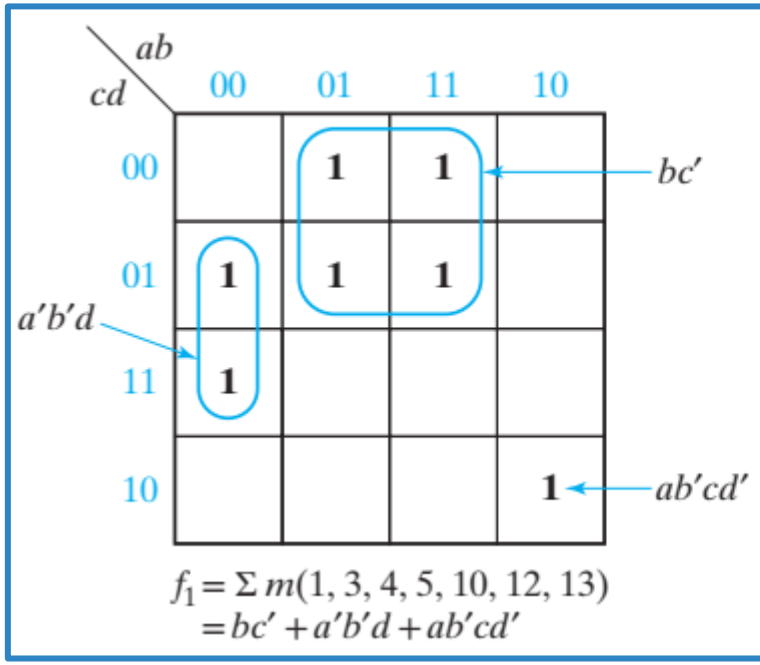




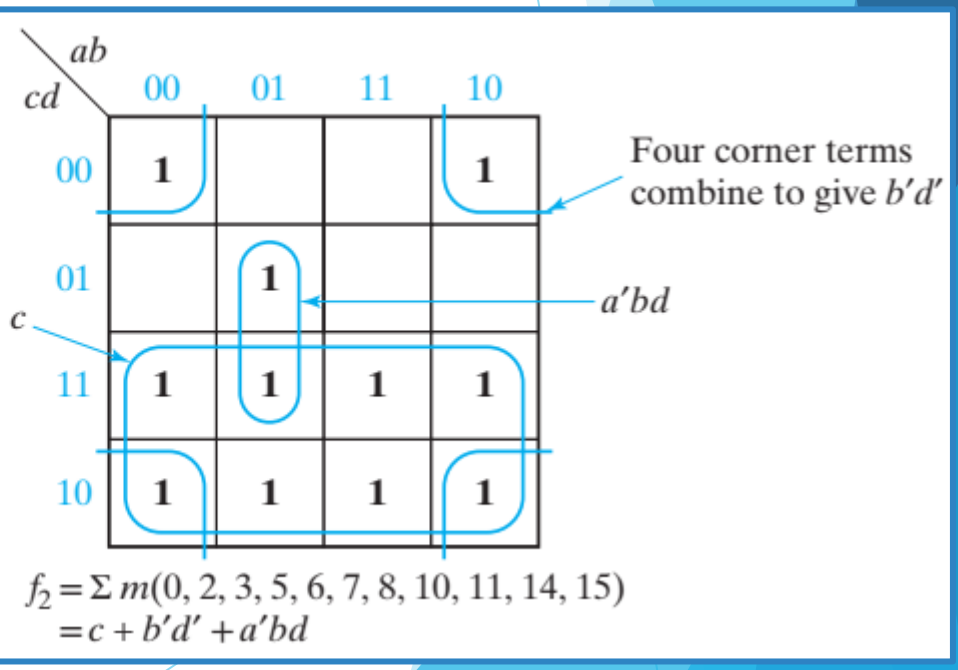
Four-Variable Karnaugh Maps

- Figure below shows the location of minterms on a four-variable map. Each minterm is located adjacent to the four terms with which it can combine.
- The definition of adjacent squares must be extended so that not only are top and bottom rows adjacent as in the three-variable map, but the first and last columns are also adjacent.

CD \ AB	00	01	11	10
00	0	4	12	8
01	1	5	13	9
11	3	7	15	11
10	2	6	14	10



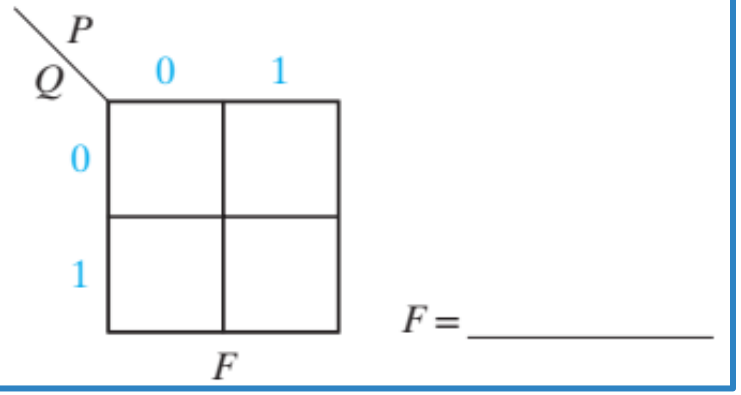
$f_1 = \Sigma\, m(1, 3, 4, 5, 10, 12, 13)$
$= bc' + a'b'd + ab'cd'$



$f_2 = \Sigma\, m(0, 2, 3, 5, 6, 7, 8, 10, 11, 14, 15)$
$= c + b'd' + a'bd$

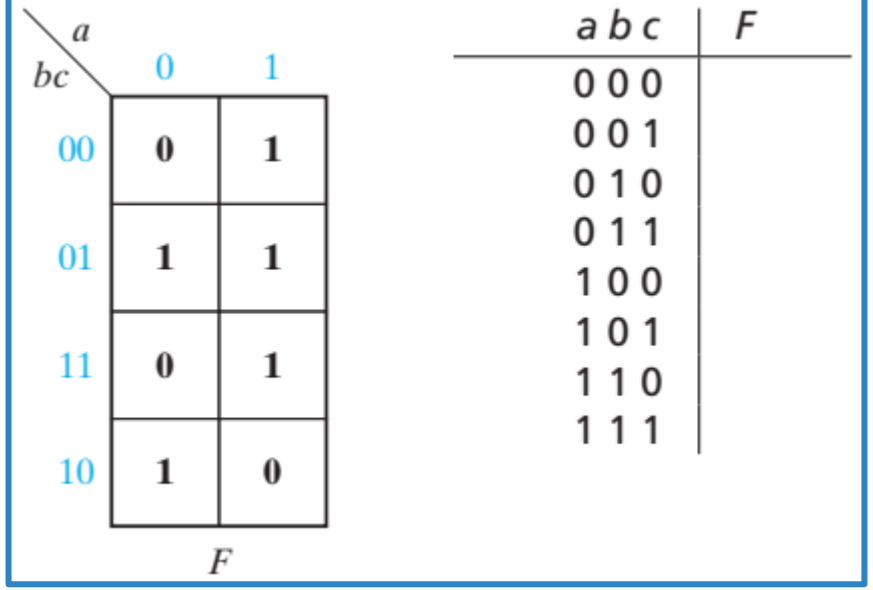
- Plot the truth table on the map. Then, loop two pairs of 1's on the map and write the simplified form of F:

PQ	F
0 0	1
0 1	1
1 0	0
1 1	1



$F =$ _____

- F(a, b, c) is plotted below. Find the truth table for F:



a b c	F
0 0 0	
0 0 1	
0 1 0	
0 1 1	
1 0 0	
1 0 1	
1 1 0	
1 1 1	

Questions

- Plot the following functions on Karnaugh maps:

$$F_1(R, S, T) = \Sigma\, m(0, 1, 5, 6) \quad F_2(R, S, T) = \Pi\, M(2, 3, 4, 7)$$

- Plot the following function on Karnaugh map:

$$f(x, y, z) = z' + x'z + yz$$

- Plot the following function on Karnaugh map:

(1) $f(w, x, y, z) = \Sigma\, m(0, 1, 2, 5, 7, 8, 9, 10, 13, 14)$

(2) $f(w, x, y, z) = x'z' + y'z + w'xz + wyz'$

- Plot the following function on a Karnaugh map. Then, Find the minimum sum of products.

$$F(A, B, C, D) = BD' + B'CD + ABC + ABC'D + B'D'$$

Questions

- Simplify the following Boolean functions using K-maps:

  (a) $F(X, Y, Z) = \Sigma m(2, 3, 4, 7)$

  (b) $F(X, Y, Z) = \Sigma m(0, 4, 5, 6)$

  (c) $F(A, B, C) = \Sigma m(0, 2, 4, 6, 7)$

  (d) $F(A, B, C) = \Sigma m(0, 1, 3, 4, 6, 7)$

- Simplify the following Boolean functions using K-maps:

  (a) $\overline{X}\,\overline{Z} + Y\overline{Z} + XYZ$

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

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

- Simplify the following Boolean functions using K-maps:

  (a) $F(A, B, C, D) = \Sigma m(0, 2, 4, 5, 8, 10, 11, 15)$

  (b) $F(A, B, C, D) = \Sigma m(0, 1, 2, 4, 5, 6, 10, 11)$

  (c) $F(W, X, Y, Z) = \Sigma m(0, 2, 4, 7, 8, 10, 12, 13)$

Questions

- Simplify the following Boolean functions using K-maps:

(a) $F(W, X, Y, Z) = \Sigma m(0, 1, 2, 4, 7, 8, 10, 12)$

(b) $F(A, B, C, D) = \Sigma m(1, 4, 5, 6, 10, 11, 12, 13, 15)$

- Simplify the following Boolean functions using K-maps:

(a) $XY + XZ + \overline{X}YZ$

(b) $XZ + \overline{W}X\overline{Y} + WXY + \overline{W}YZ + W\overline{Y}Z$

(c) $\overline{B}\,\overline{D} + ABD + \overline{A}BC$

- Find the minimum sum-of-products expression for each function:

(a) $f(a, b, c, d) = \Sigma\, m(0, 2, 3, 4, 7, 8, 14)$

(b) $f(a, b, c, d) = \Sigma\, m(1, 2, 4, 15) + \Sigma\, d(0, 3, 14)$

(c) $f(a, b, c, d) = \Pi\, M(1, 2, 3, 4, 9, 15)$

(d) $f(a, b, c, d) = \Pi\, M(0, 2, 4, 6, 8) \cdot \Pi\, D(1, 12, 9, 15)$

Questions

الجامعة التقنية الجنوبية
المعهد التقني القرنـــــة
قسم تقنيات انظمة الحاسوب

التصميم المنطقي
Logic Design

محاضرة رقم (10)

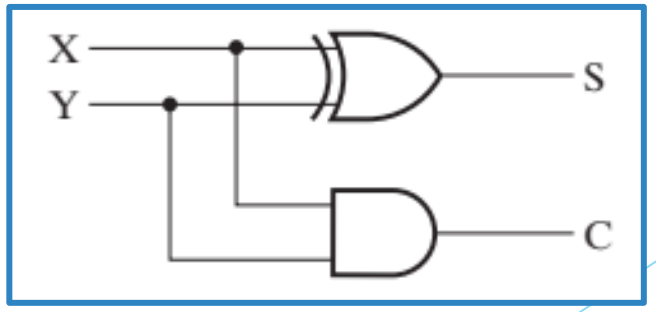
د. حيدر احمد عبدالمحسن

Binary Adders – Half Adder

- An arithmetic circuit is a combinational circuit that performs arithmetic operations such as addition, subtraction, multiplication, and division with binary numbers or with decimal numbers in a binary code.
- A combinational circuit that performs the addition of two bits is called a half adder.
- A half adder generates the sum of two binary digits where the circuit has two inputs and two outputs.
- The input variables are the augend and addend bits to be added (X and Y), and the output variables produce the sum and carry (S and C).
- The truth table for the half adder is listed below.
- The Boolean functions for the outputs are:

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

$$C = XY$$



Truth Table of Half Adder

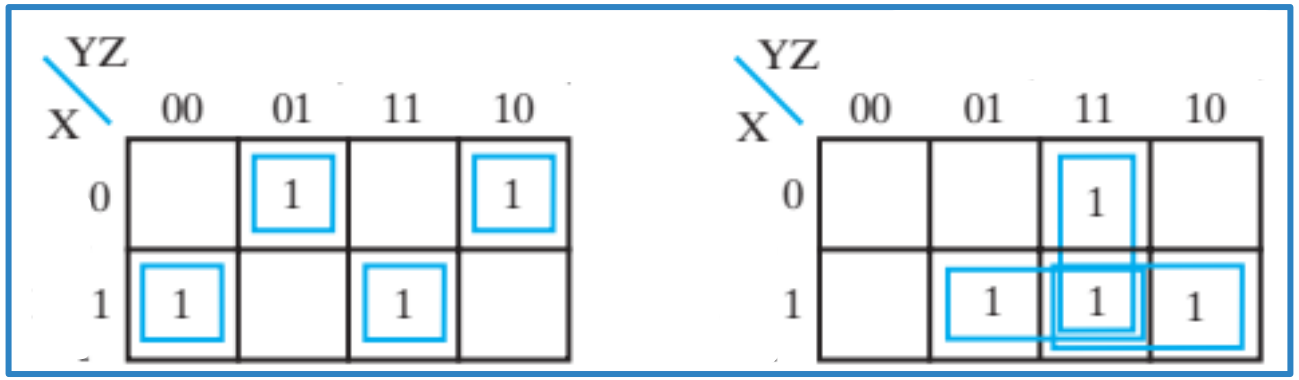
Inputs		Outputs	
X	Y	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

- Total number of inputs in a half adder is _____.
- If A and B are inputs of a half adder, the sum is given by
  - A AND B
  - A OR B
  - A XOR B
  - A XNOR B
- If A and B are inputs of a half adder, the carry is given by
  - A AND B
  - A OR B
  - A XOR B
  - A XNOR B
- Write the truth table for half adder.
- Derive the Boolean expression for half adder outputs, S and C.
- Draw the logic circuit of half adder.

Questions

Binary Adders – Full Adder

- A full adder is a combinational circuit that forms the arithmetic sum of three input bits. Besides the three inputs, it has two outputs (S and C).
- Two of the input variables, denoted by X and Y, represent the two significant bits to be added. The third input, Z, represents the carry from the previous lower significant position.
- The truth table of the full adder is listed below.



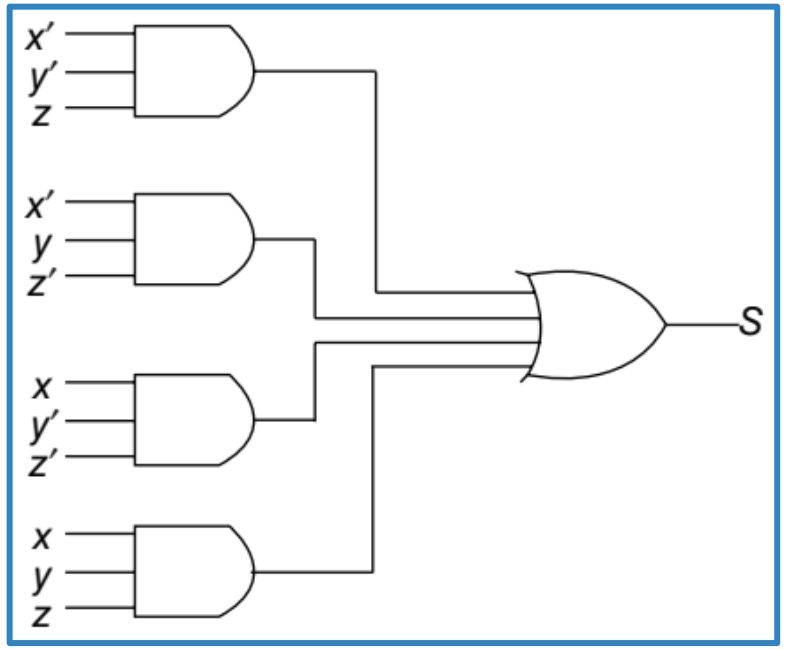
$$S = \overline{X}\,\overline{Y}Z + \overline{X}Y\overline{Z} + X\overline{Y}\,\overline{Z} + XYZ$$

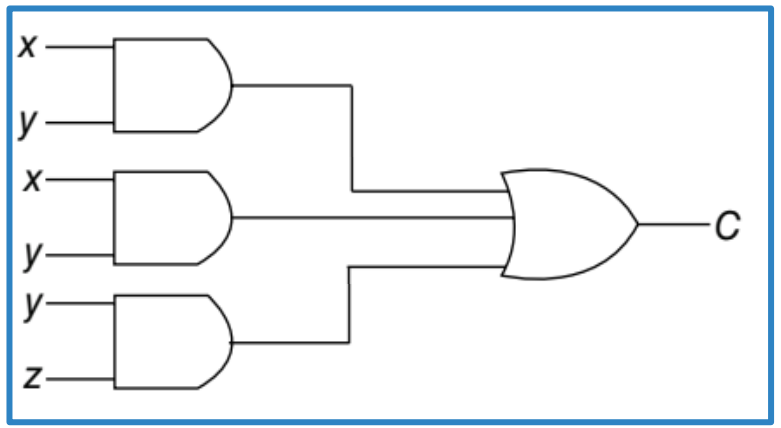
$$C = XY + XZ + YZ$$

Truth Table of Full Adder

Inputs			Outputs	
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

Full Adder Logic Diagram



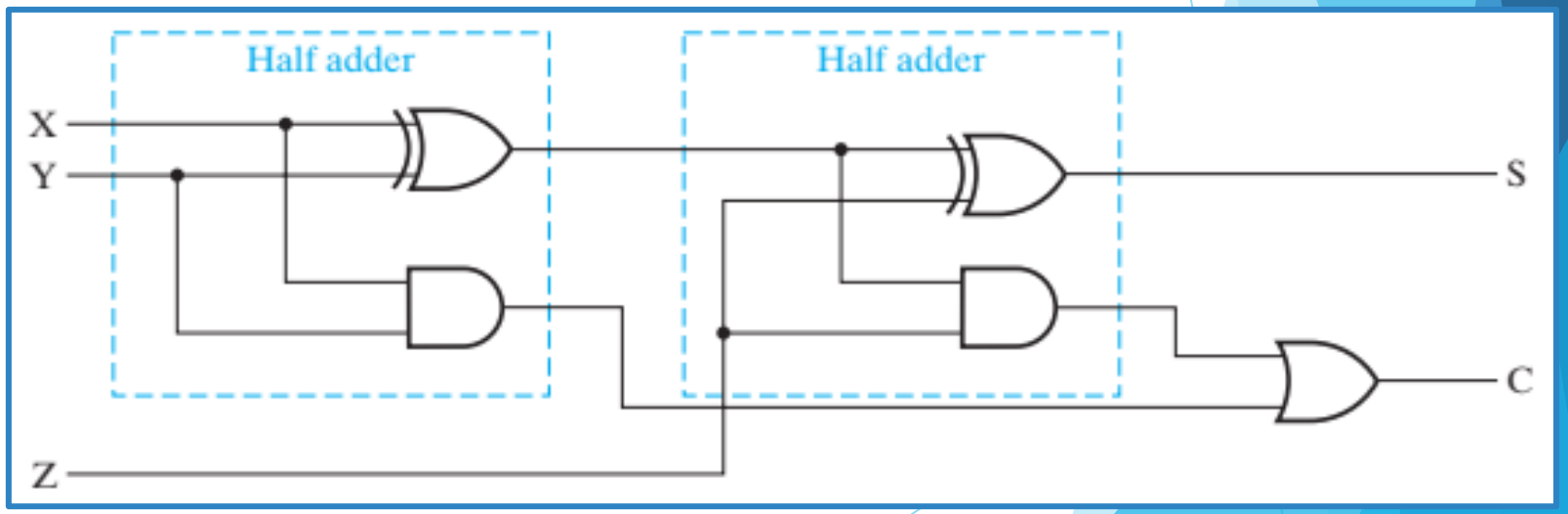


$$S = \overline{X}\,\overline{Y}Z + \overline{X}Y\overline{Z} + X\overline{Y}\,\overline{Z} + XYZ$$

$$C = XY + XZ + YZ$$

$$S = (X \oplus Y) \oplus Z$$

$$C = XY + Z(X \oplus Y)$$



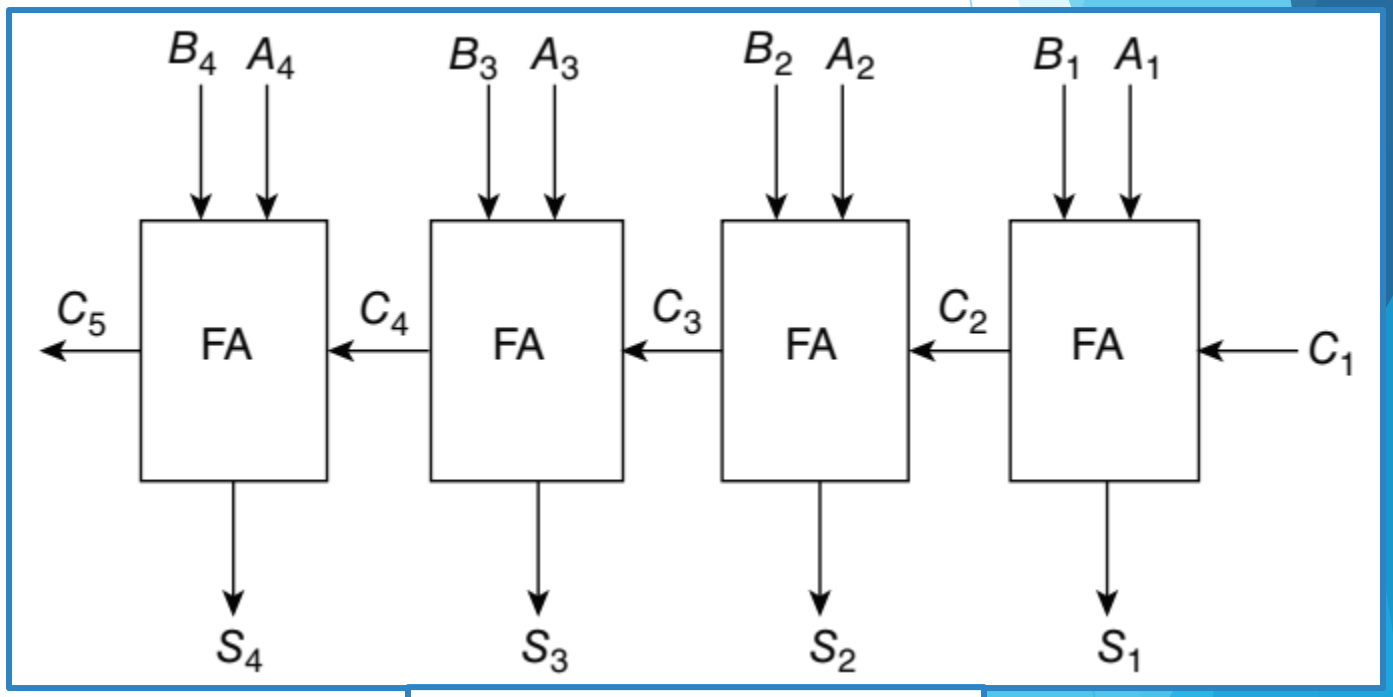
Questions

- Total number of inputs in a full adder is _____.
- If A, B and C are inputs of a full adder, the sum is
  - A AND B AND C
  - A XOR B XOR C
  - d) A OR B OR C
- If A, B and C are inputs of a full adder, the carry is
  - A AND B OR (A OR B) AND C
  - A OR B OR (A AND B) C
  - (A AND B) OR (A AND B) C
- Write the truth table for full adder.
- Derive Boolean expressions for full adder outputs, D and B.
- Draw the logic circuit of full adder.
- What is the difference between half adder and full adder.

Binary Parallel Adder

- The full-adder forms the sum of two bits and a previous carry. Two binary numbers of n bits each can be added using binary parallel adder.
- For example, consider two binary numbers, A = 1011 and B = 0011, whose sum is S = 1110.

Input carry	0 1 1 0	C_i
Augend	1 0 1 1	A_i
Addend	0 0 1 1	B_i
Sum	1 1 1 0	S_i
Output carry	0 0 1 1	C_{i+1}



4-bit full-adders

- In a parallel adder, all the bits of multi-digit numbers are added
  - one bit at a time
  - 4 bit at a time
  - simultaneously
  - none of above.
- Construct a binary parallel adder to add two 6-bit binary numbers using half adder and full adder.
- What is the basic difference between parallel adder and serial adder

Questions

الجامعة التقنية الجنوبية
المعهد التقني القرنـــــة
قسم تقنيات انظمة الحاسوب

التصميم المنطقي
Logic Design

محاضرة رقم (11)

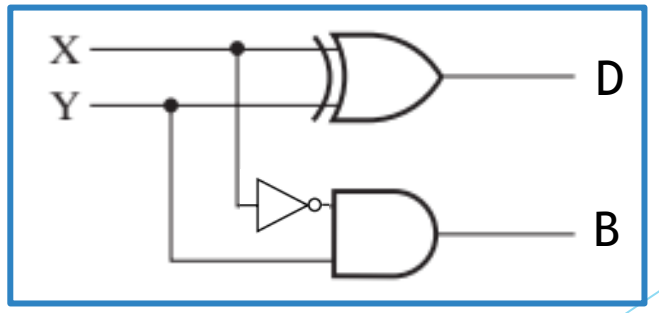
د. حيدر احمد عبدالمحسن

Binary Subtractors – Half Subtractor

- A half-subtractor is a combinational circuit that subtracts two bits and produces their difference.
- A half subtractor generates the difference of two binary digits where the circuit has two inputs and two outputs.
- The input variables are the minuend and subtrahend bits to be subtracted (X and Y), and the output variables produce the difference and borrow (D and B).
- The truth table for the half subtractor is listed below.
- The Boolean functions for the outputs are:

$$D = \overline{X}Y + X\overline{Y} = X \oplus Y$$

$$B = \overline{X}Y$$



Truth table of half subtractor

X	Y	B	D
0	0	0	0
0	1	1	1
1	0	0	1
1	1	0	0

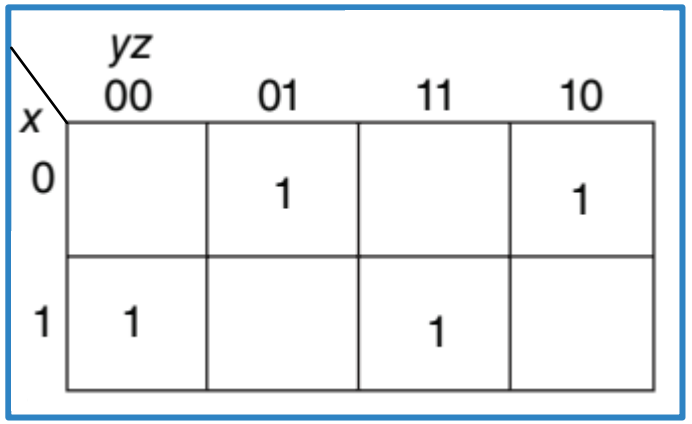
- Total number of inputs in a half subtractor is _____.
- If A and B are inputs of a half subtractor, the difference is
  - A AND B
  - A OR B
  - A XOR B
  - A XNOR B
- If A and B are inputs of a half subtractor, the borrow is
  - NOT(A) AND B
  - A AND NOT(B)
  - NOT(A) XOR B
  - A XOR NOT(B)
- Write the truth table for a half subtractor.
- Derive the Boolean expression for half subtractor outputs, D and B.
- Draw the logic circuit of half subtractor.

Questions

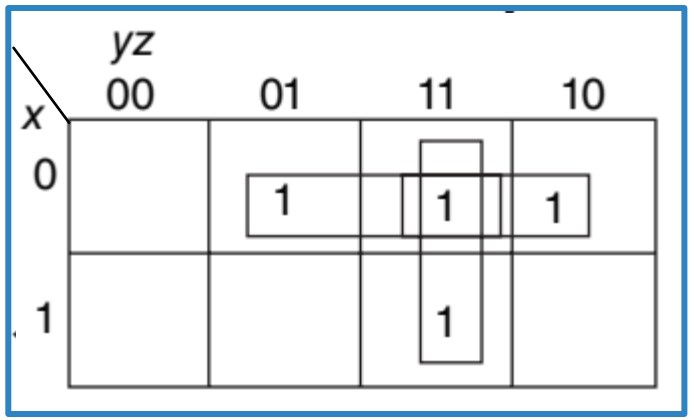
Binary Subtractors – Full Subtractor

- A full-subtractor is a combinational circuit that performs a subtraction between two bits, considering that a 1 may have been borrowed by a lower significant stage. The circuit has three inputs and two outputs.
- The three inputs, x, y, and z, denote the minuend, subtrahend, and previous borrow, respectively. The two outputs, D and B, represent the difference and output borrow, respectively.
- The truth table for the circuit is as follows:

x	y	z	B	D
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	1	0
1	0	0	0	1
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1



$$D = x'y'z + x'yz + xy'z' + xyz$$



$$B = x'y + x'z + yz$$

- Total number of inputs in a full subtractor is _____.
- If A, B and C are inputs of a full subtractor, the difference is:
  - A AND B AND C
  - A XOR B XOR C
  - d) A OR B OR C
- If A, B and C are inputs of a full subtractor, the borrow is
  - A AND B OR (A OR B) AND C
  - A OR B OR (A AND B) C
  - (A AND B) OR (A AND B) C
- Write the truth table for full subtractor.
- Derive Boolean expressions for full subtractor outputs, D and B.
- Draw the logic circuit of full subtractor.
- What is the difference between half subtractor and full subtractor.

Questions

Binary Subtractors – Full Subtractor

- We note that the logic function for output D in the full-subtractor is the same as output S in the full adder. Moreover, the output B resembles the function for C in the full-adder, except that x is complemented.

Full Adder	$S = \overline{X}\overline{Y}Z + \overline{X}Y\overline{Z} + X\overline{Y}\overline{Z} + XYZ$ $C = XY + XZ + YZ$
Full Subtractor	$D = x'y'z + x'yz' + x\,y'z' + x\,yz$ $B = x'y + x'z + yz$

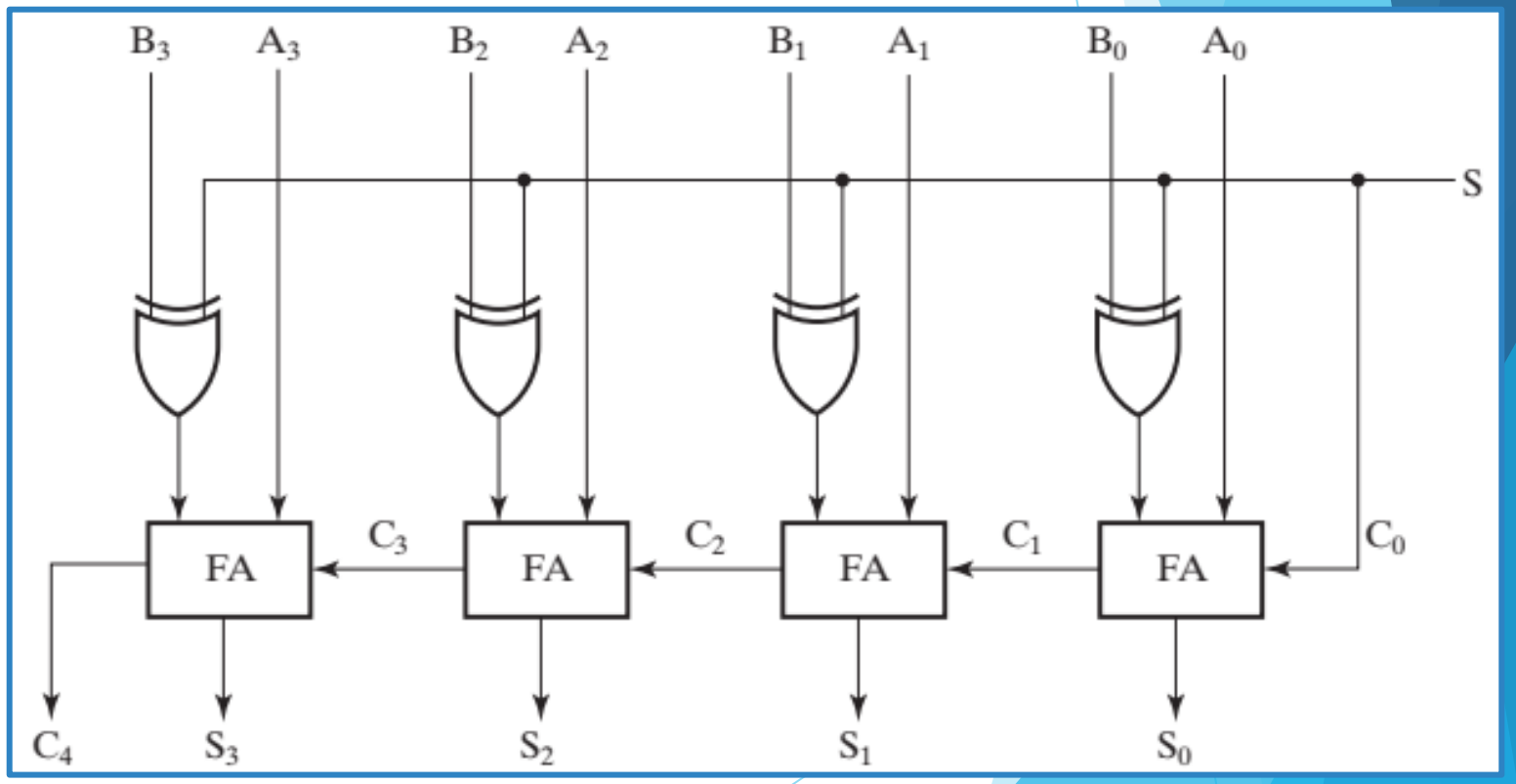
- As we explained in Week 4, the subtraction can be performed using complement as follows:

M – N = M + 2's complement of (N)

- Using the 2s complement, we have eliminated the subtraction and need only the complementor and an adder.
- The 2's complement can be obtained by taking the 1's complement and adding 1 to the least significant bit. The 1's complement can be implemented easily with inverter circuits, and we can add 1 to the sum by making the input carry of the parallel adder equal to 1.

Binary Adder-Subtractor

- The addition and subtraction operations can be combined into one circuit with one common binary adder.
- This is done by including an exclusive-OR gate with each full adder. A 4-bit adder-subtractor circuit is shown in Figure below:
- Input S controls the operation.
- When S = 0 the circuit is an adder, and when S = 1 the circuit becomes a subtractor.



- Construct a 6-bit parallel adder/subtractor using full adder and XOR gate.

Questions

الجامعة التقنية الجنوبية
المعهد التقني القرنـــــة
قسم تقنيات انظمة الحاسوب

التصميم المنطقي
Logic Design

محاضرة رقم (12)

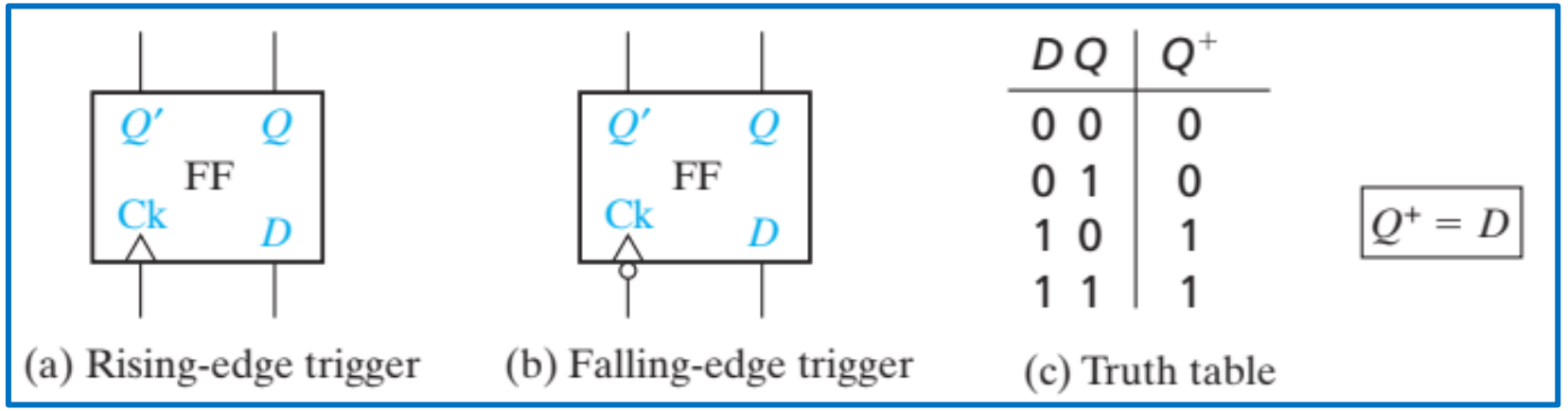
د. حيدر احمد عبدالمحسن

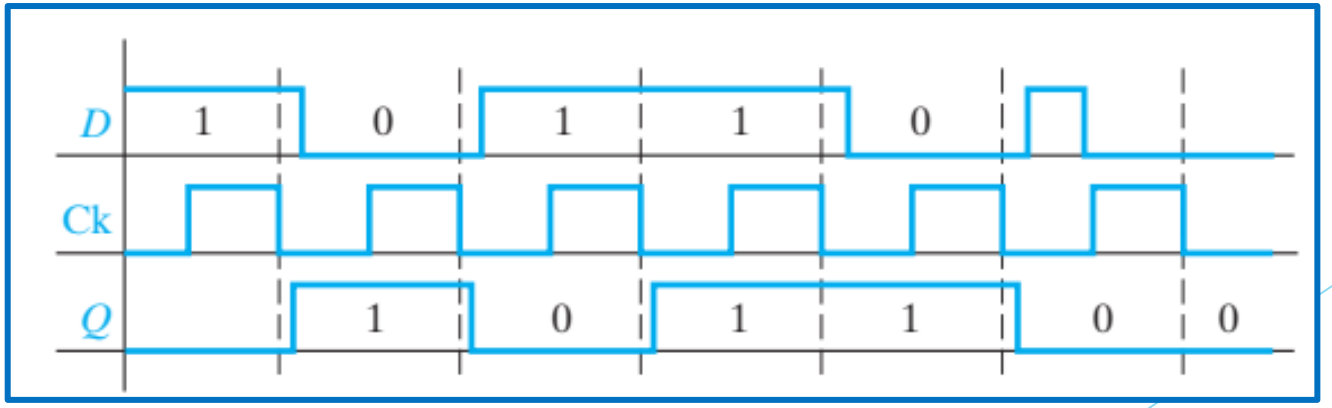
Flip-Flops

- latches and flip-flops are memory devices which can assume one of two stable output states and which have one or more inputs that can cause the output state to change.
- In synchronous digital systems, it is common practice to synchronize the operation of all flip-flops by a common clock or pulse generator. Each of the flip-flops has a clock input, and the flip-flops can only change state in response to a clock pulse.
- If the output can change in response to a 0 to 1 transition on the clock input, we say that the flip-flop is triggered on the rising edge (or positive edge) of the clock. If the output can change in response to a 1 to 0 transition on the clock input, we say that the flip-flop is triggered on the falling edge (or negative edge) of the clock.
- A memory element that has no clock input is often called a latch.
- Flip flops are the main components of sequential circuits. Some of the most common applications of flip-flops are counters, registers and data transfer.

D Flip-Flop

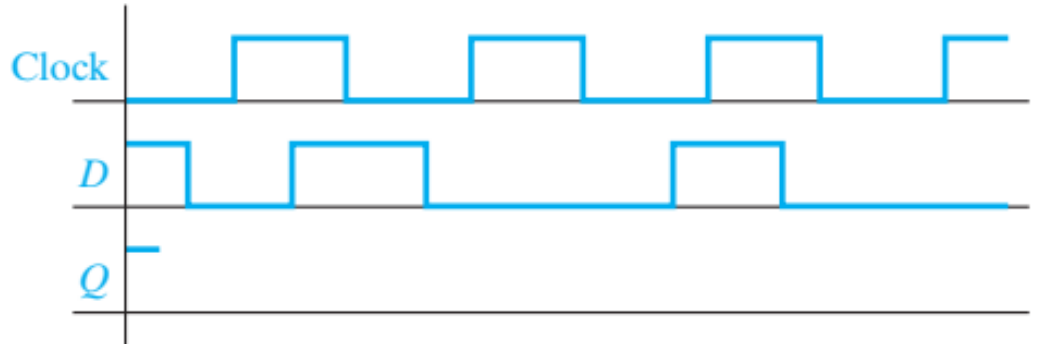
- A D flip-flop has two inputs, D (Data) and Ck (clock). The state of a D flip-flop after the active clock edge (Q^+) is equal to the input (D) before the active edge.



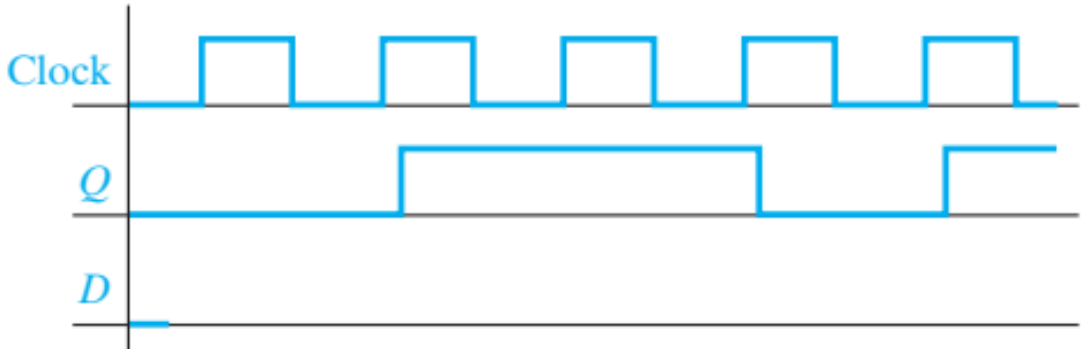


Questions

- Complete the following diagrams for the rising-edge-triggered D flip-flop. Assume Q begins at 1.



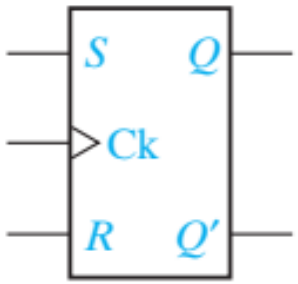
- Find the input for a rising-edge-triggered D flip-flop that would produce the output Q as shown.



- Derive the characteristic equations for D flip-flop in product-of-sums form.

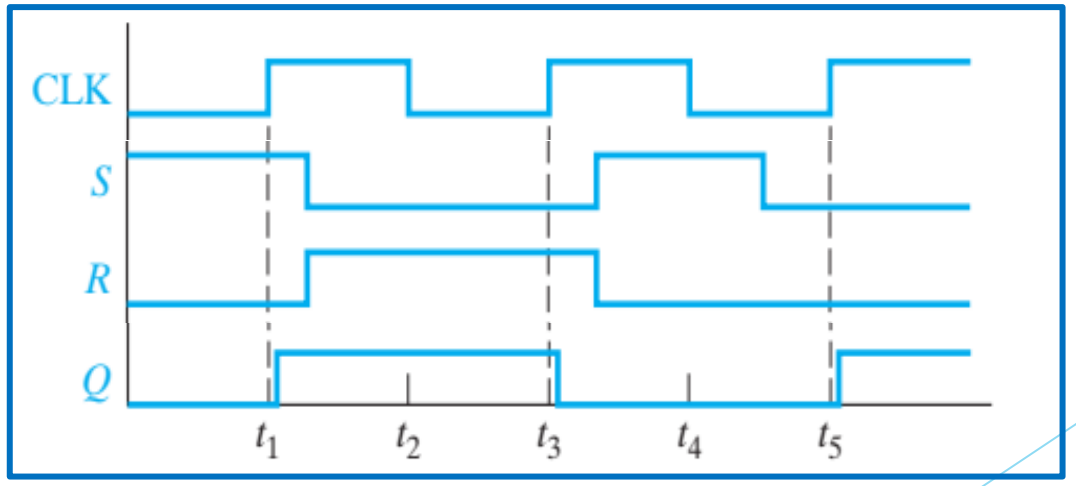
S-R Flip-Flop

- A S-R flip-flop has three inputs, S (Set), R (Reset) and Ck (clock). S = 1 sets the Q output to 1, and R = 1 resets the Q output to 0.

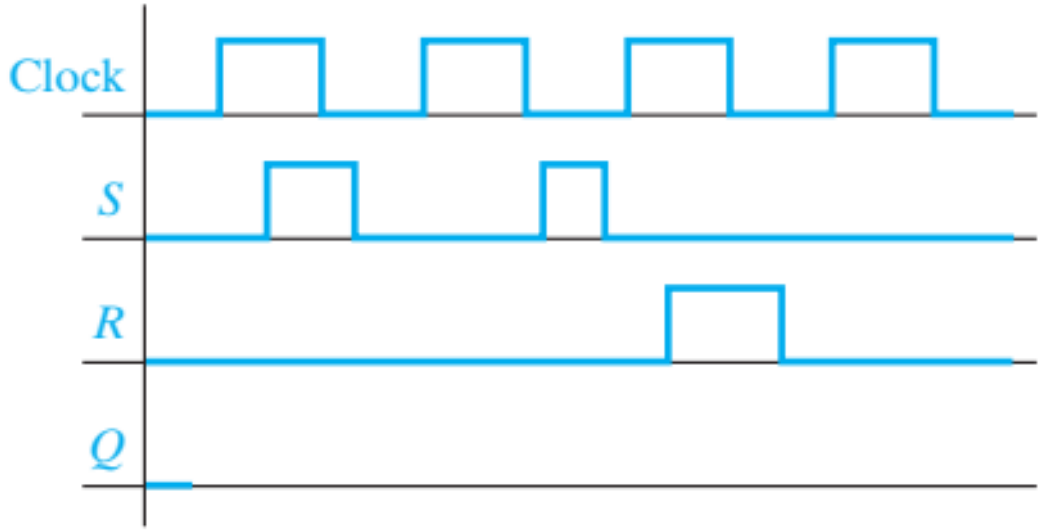


Operation summary:

$S = R = 0$	No state change
$S = 1, R = 0$	Set Q to 1 (after active Ck edge)
$S = 0, R = 1$	Reset Q to 0 (after active Ck edge)
$S = R = 1$	Not allowed



- Fill in the timing diagram for a falling-edge-triggered S-R flip-flop. Assume Q begins at 0.

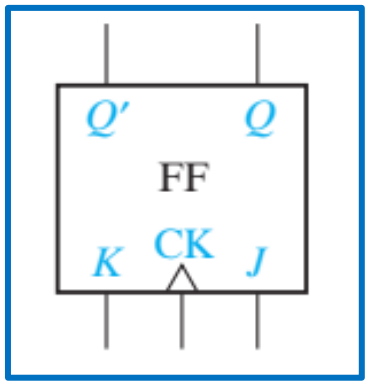


- Derive the characteristic equations for S-R flip-flop in product-of-sums form.

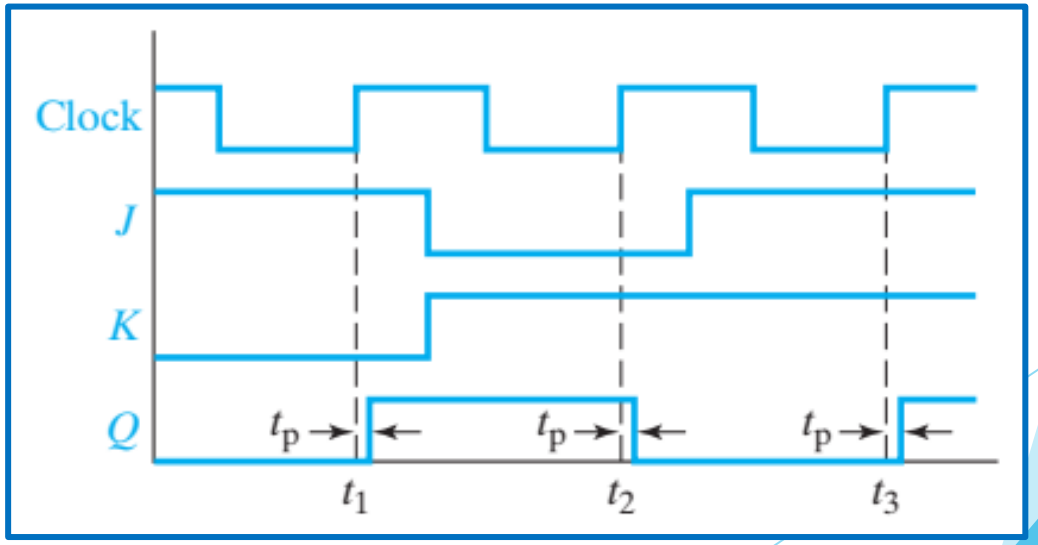
Questions

J-K Flip-Flop

- The J-K flip-flop is an extended version of the S-R flip-flop. The J-K flip-flop has three inputs—J, K, and the clock (CK). The J input corresponds to S, and K corresponds to R.
- Unlike the S-R flip-flop, a 1 input may be applied simultaneously to J and K, in which case the flip-flop changes state after the active clock edge.
- When J = K = 1, the active edge will cause Q to change from 0 to 1, or from 1 to 0.



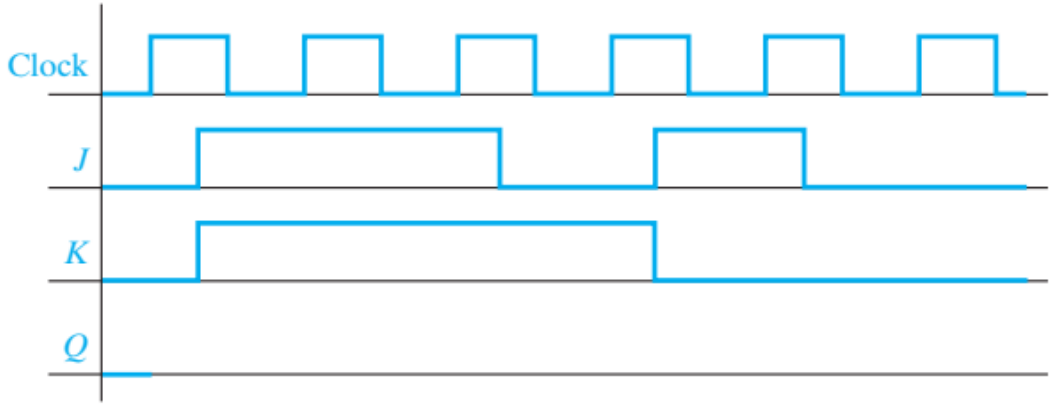
$$Q^+ = JQ' + K'Q$$



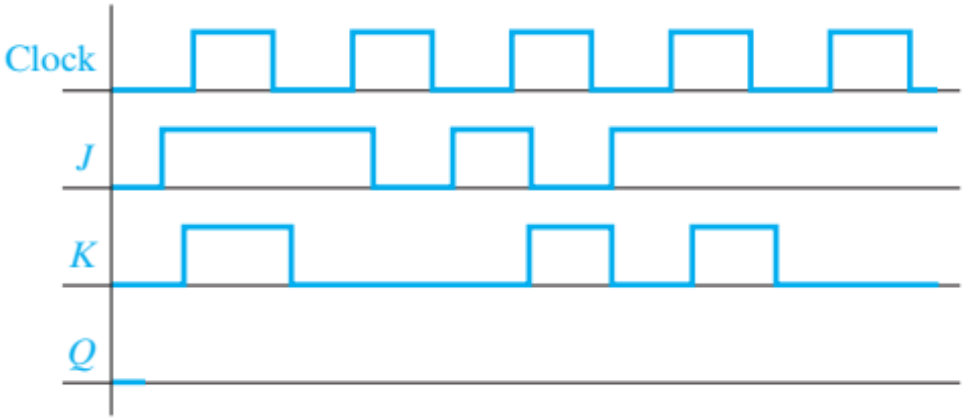
J	K	Q	Q^+
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

Questions

- Complete the following timing diagram for J-K flip-flop.



- Fill in the timing diagram for a falling-edge-triggered J-K flip-flop. Assume Q begins at 1.

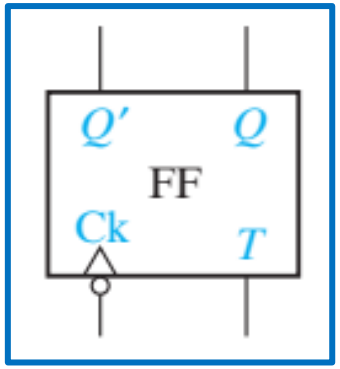


- Derive the characteristic equations for J-K flip-flop in product-of-sums form.

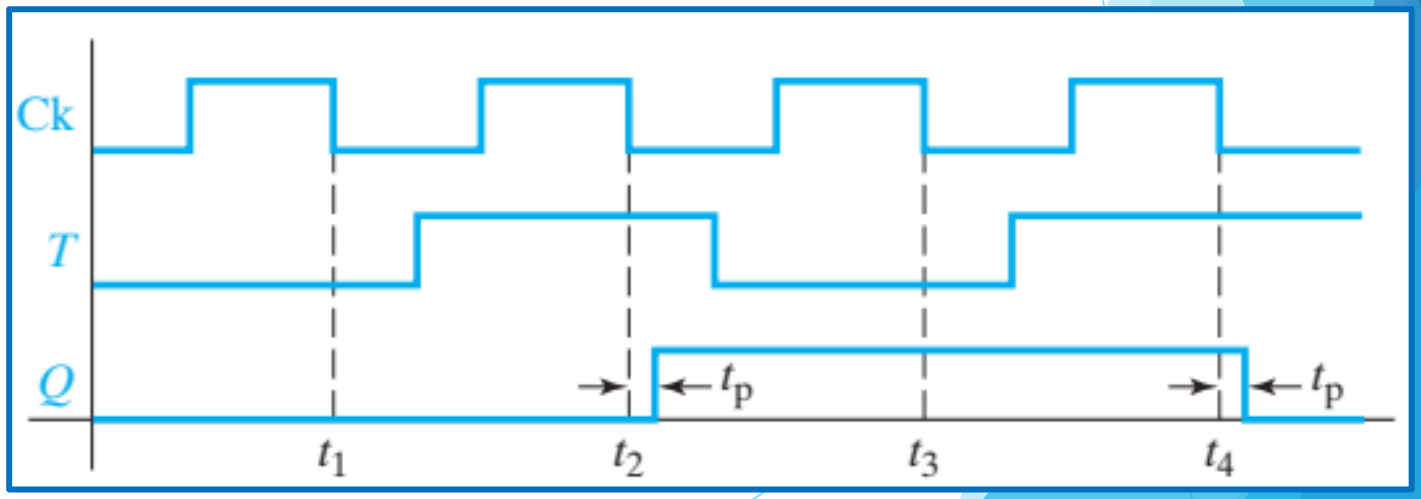
T Flip-Flop

- The T flip-flop, also called the toggle flip-flop. The T flip-flop has a T input and a clock input. When T = 1 the flip-flop changes state after the active edge of the clock. When T = 0, no state change occurs.
- In the timing diagram below, at times t_2 and t_4, the T input is 1 and the flip-flop state (Q) changes a short time (tp) after the falling edge of the clock pulse. At times t_1 and t_3, the T input is 0, and the clock edge does not cause a change of state.

$$Q^+ = T'Q + TQ' = T \oplus Q$$

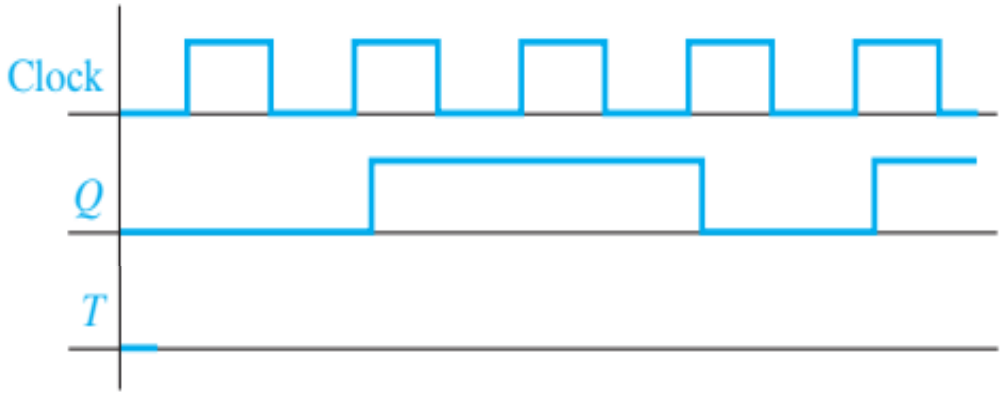


T Q	Q^+
0 0	0
0 1	1
1 0	1
1 1	0



Questions

- Find the input for a rising-edge-triggered T flip-flop that would produce the output Q as shown.



- Derive the characteristic equations for T flip-flop in product-of-sums form.
- Construct T flip-flop using J-K flip-flop.
- Construct T flip-flop using D flip-flop and XOR gate.

الجامعة التقنية الجنوبية
المعهد التقني القرنـــــة
قسم تقنيات انظمة الحاسوب

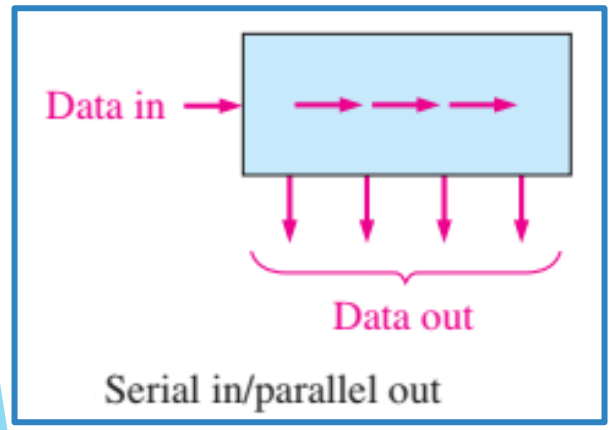
التصميم المنطقي
Logic Design

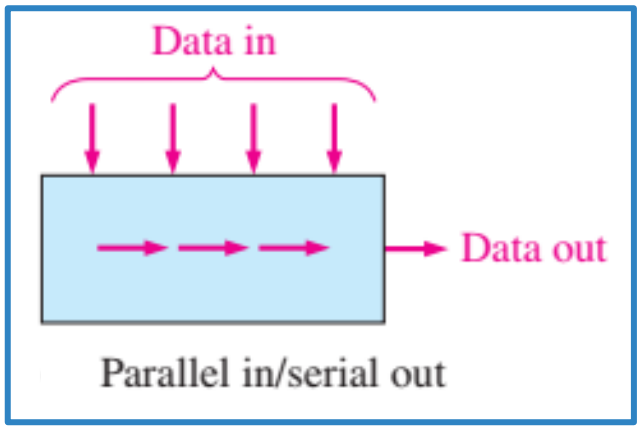
محاضرة رقم (13)

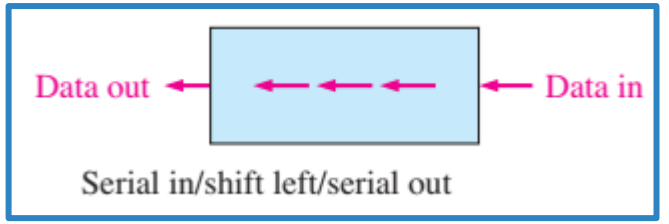
د. حيدر احمد عبدالمحسن

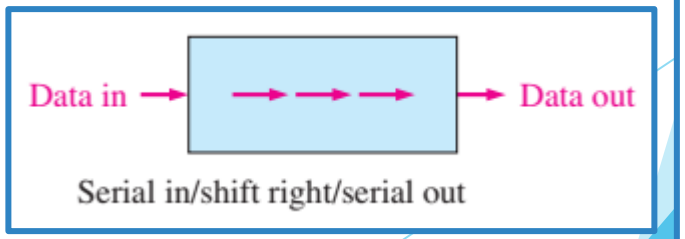
Shift Registers

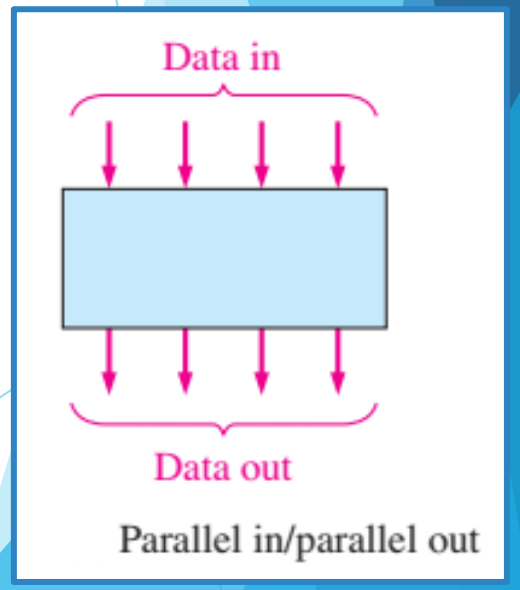
- A shift register is a digital circuit (consists of arrangements of flip-flops) with two functions: data storage and data movement. The storage capability of a register makes it an important type of memory device.
- The storage capacity of a register is the total number of bits (1s and 0s) of digital data it can retain. Each stage (flip-flop) in a shift register represents one bit of storage capacity.
- The shift capability of a register permits the movement of data from stage to stage within the register or into or out of the register upon application of clock pulses.





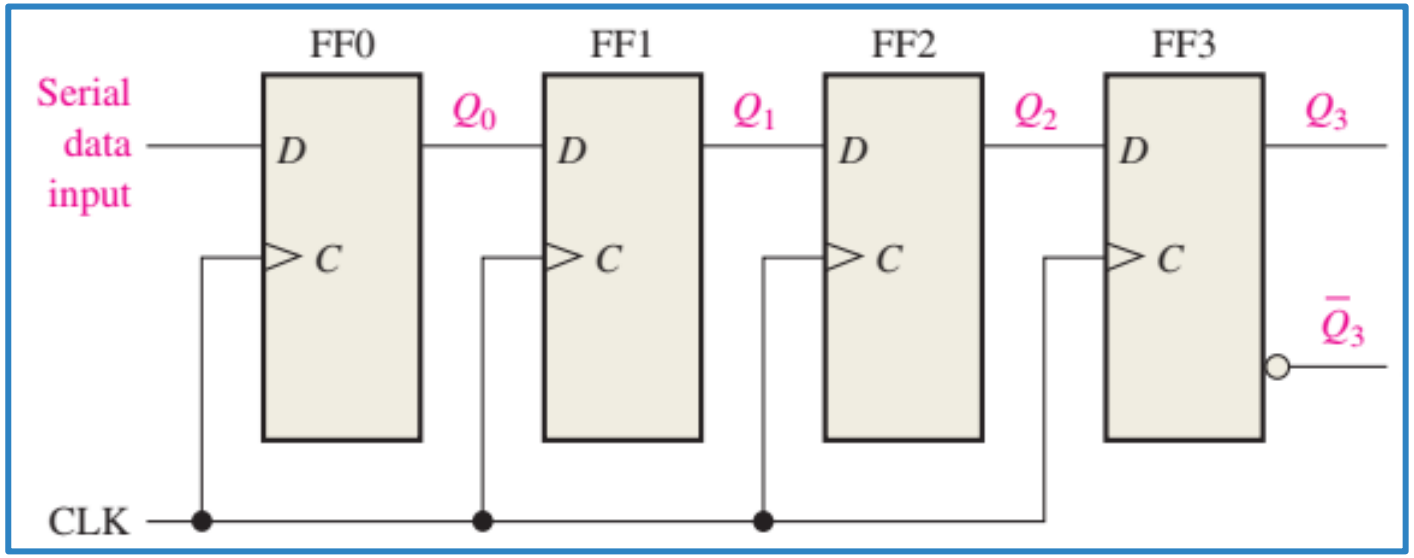






Serial In/Serial Out Shift Registers

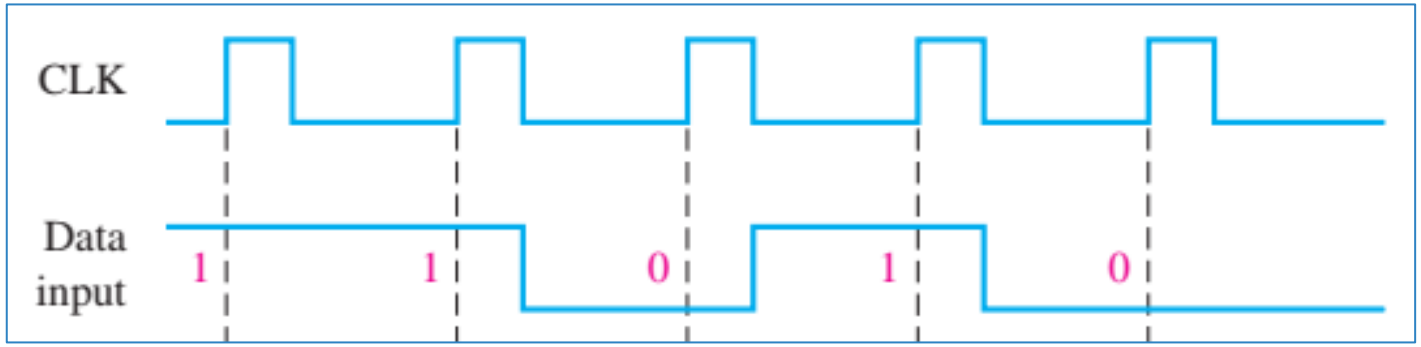
- The serial in/serial out (SISO) shift register accepts data serially—that is, one bit at a time on a single line. It produces the stored information on its output also in serial form.
- Figure below shows a 4-bit device implemented with D flip-flops. With four stages, this register can store up to four bits of data. Table below shows the entry of the four bits 1010 into the register, beginning with the least significant bit. The register is initially clear.



CLK	FF0 (Q_0)	FF1 (Q_1)	FF2 (Q_2)	FF3 (Q_3)
Initial	0	0	0	0
1	**0**	0	0	0
2	**1**	**0**	0	0
3	**0**	**1**	**0**	0
4	**1**	**0**	**1**	**0**

Questions

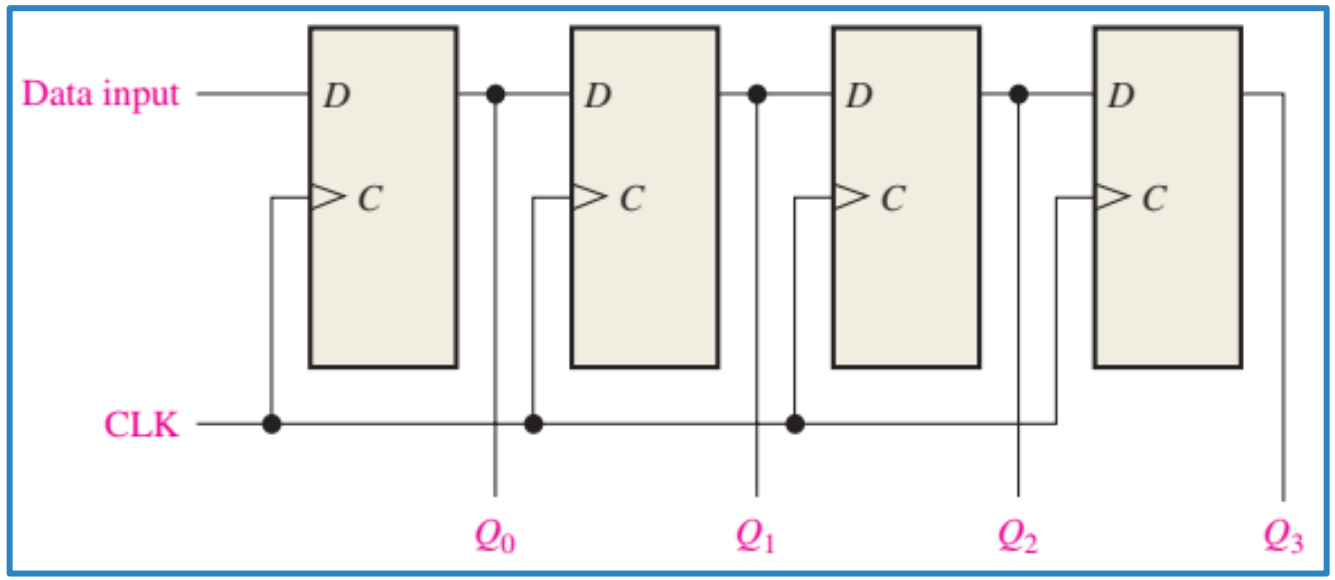
- What determines the storage capacity of a shift register?
- What two functions are performed by a shift register?
- Design a 5-bit serial in/serial out shift register?
- Show the states of the 5-bit register designed in above for the specified data input and clock waveforms. Assume that the register is initially cleared (all 0s).

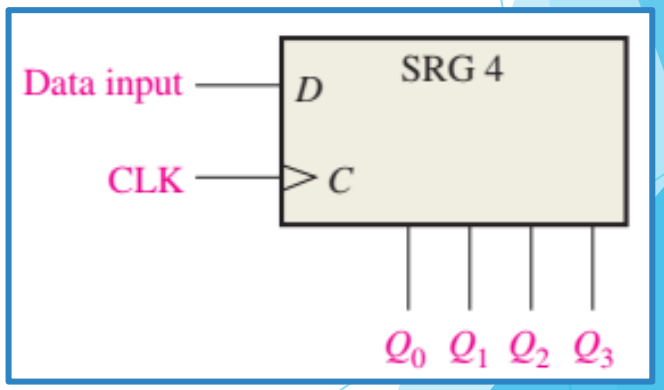


- The sequence 1011 is applied to the input of a 4-bit serial shift register that is initially cleared. What is the state of the shift register after three clock pulses?

Serial In/Parallel Out Shift Registers

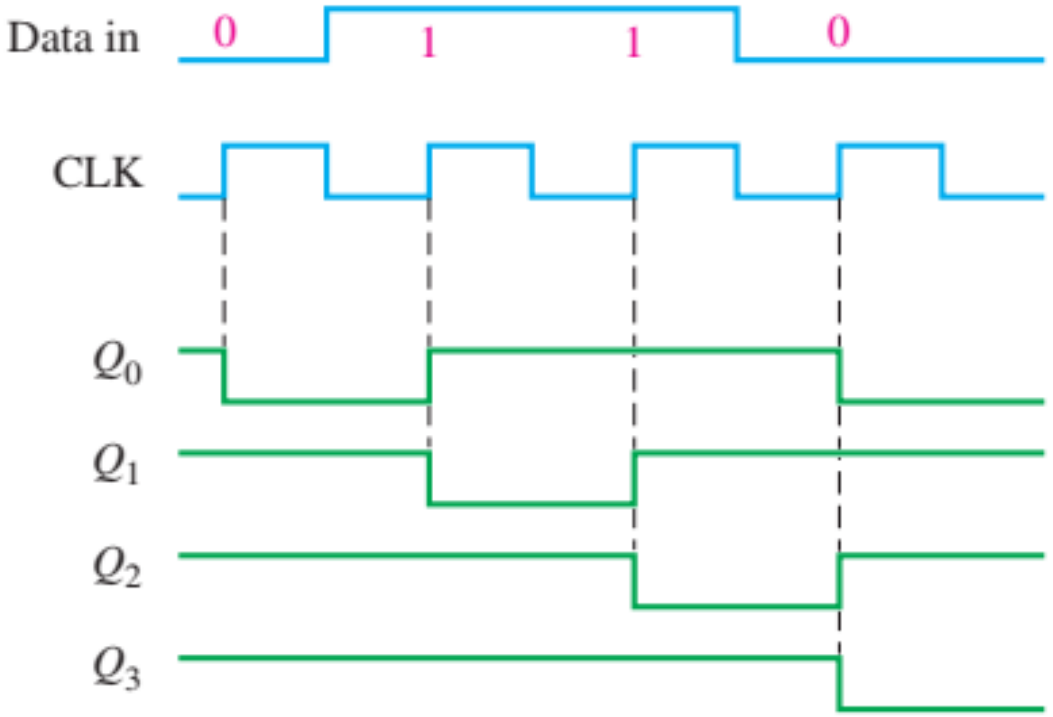
- Data bits are entered serially (least-significant bit first) into a serial in/parallel out shift (SIPO) register in the same manner as in SISO registers. The difference is the way in which the data bits are taken out of the register; in the parallel output register.
- Figure below shows a 4-bit serial in/parallel out shift register and its logic block symbol.

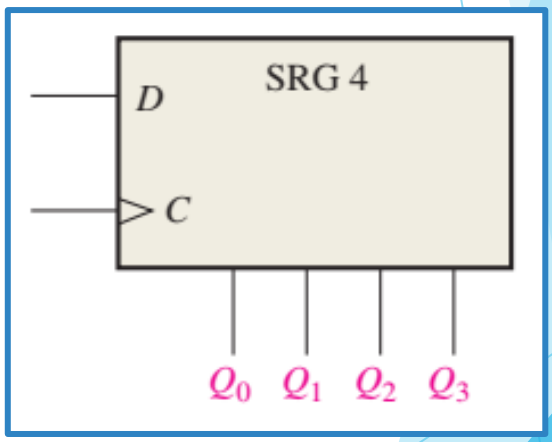




Serial In/Parallel Out Shift Registers

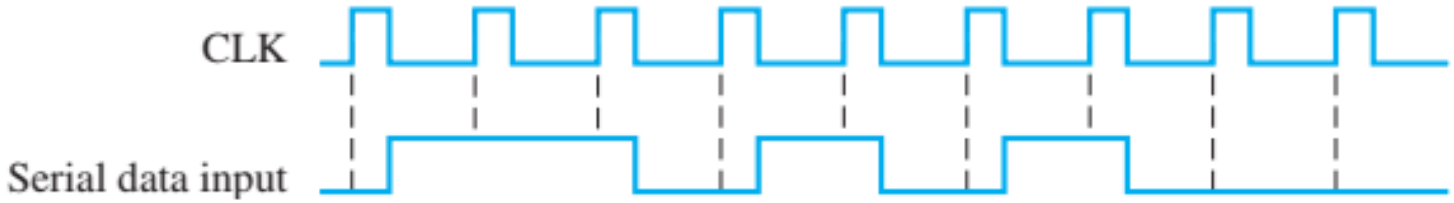
- **Example:** Show the states of the 4-bit register (SRG 4) for the data input and clock waveforms in Figure below. The register initially contains all 1s.





Questions

- Design a 6-bit serial in/parallel out shift register?
- Show the states of the 6-bit register designed in above for the specified data input and clock waveforms. Assume that the register is initially cleared (all 0s).

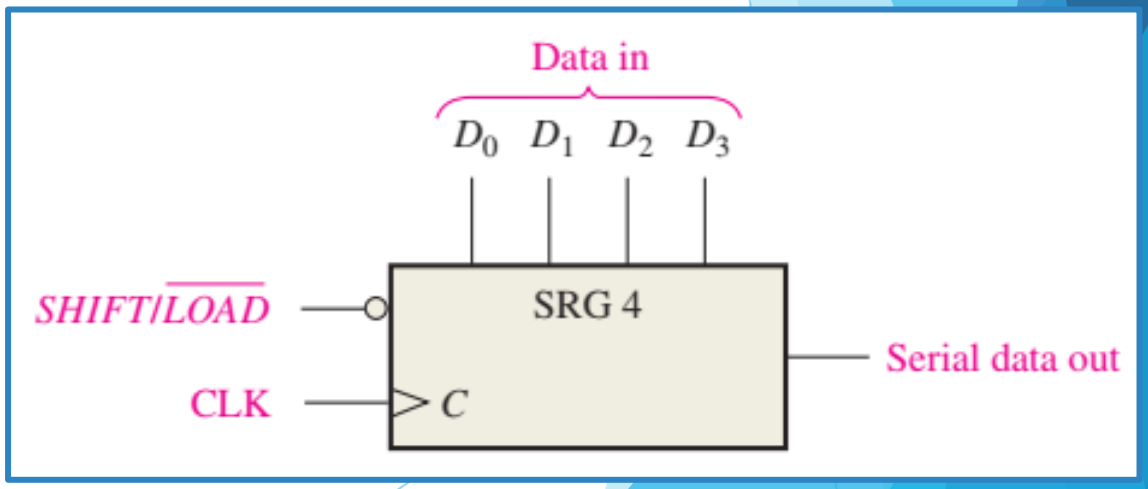


- The sequence 10010 is applied to the input of a 5-bit SIPO shift register that initially contains 1's. What is the state of the shift register after four clock pulses?
- The group of bits 10110101 is serially shifted (right-most bit first) into an 8-bit parallel output shift register with an initial state of 11100100 ($Q_0Q_1Q_2Q_3Q_4Q_5Q_6Q_7$). After two clock pulses, the register contains:

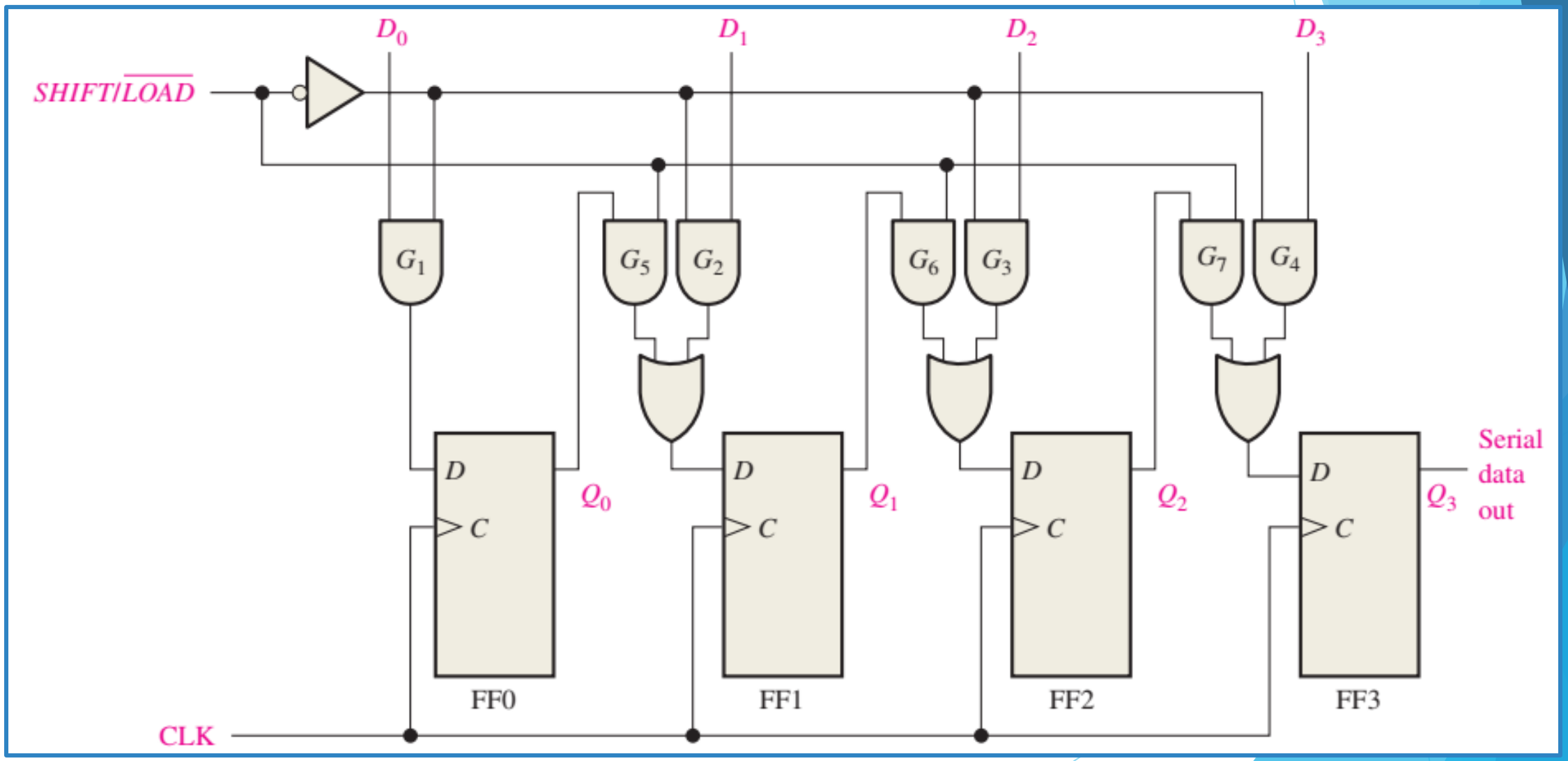
a) 01011110 **b)** 10110101 **c)** 01111001 **d)** 00101101

Parallel In/Serial Out Shift Registers

- For a register with parallel data inputs, the bits are entered simultaneously into their respective stages on parallel lines rather than on a bit-by-bit basis on one line as with serial data inputs. The serial output is the same as in serial in/serial out shift registers, once the data are completely stored in the register.
- Figure below illustrates a typical 4-bit parallel in/serial out shift register logic symbol.
- There are four data-input lines, D_0, D_1, D_2, and D_3, and a SHIFT/$\overline{\text{LOAD}}$ input, which allows four bits of data to load in parallel into the register.
- When SHIFT/$\overline{\text{LOAD}}$ is LOW allowing each data bit to be applied to the D input of its respective flip-flop.
- When SHIFT/$\overline{\text{LOAD}}$ is HIGH, allowing the data bits to shift right from one stage to the next.

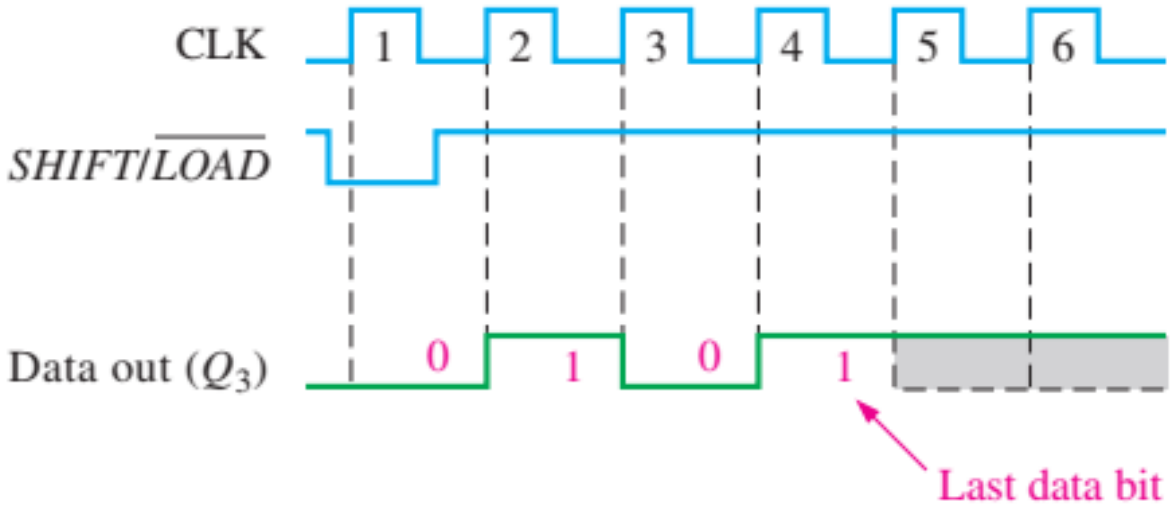


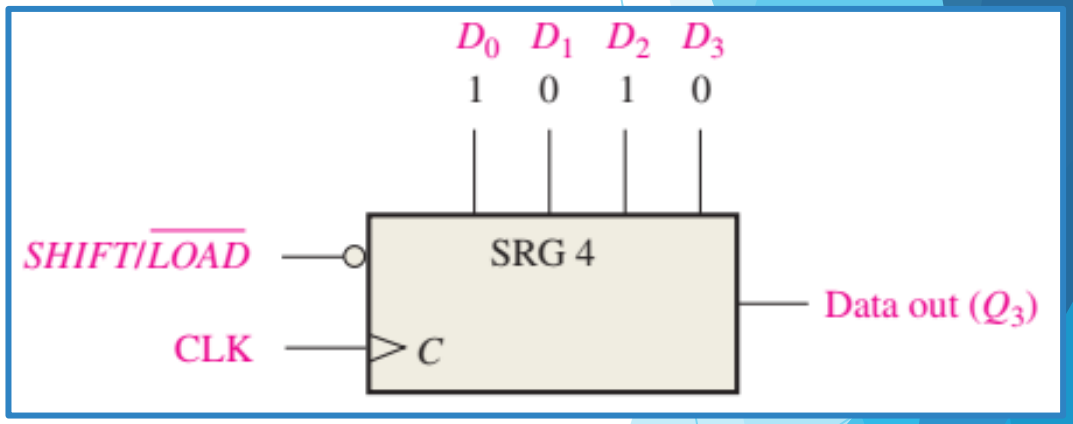
4-bit Parallel In/Serial Out Shift Register



Parallel In/Serial Out Shift Registers

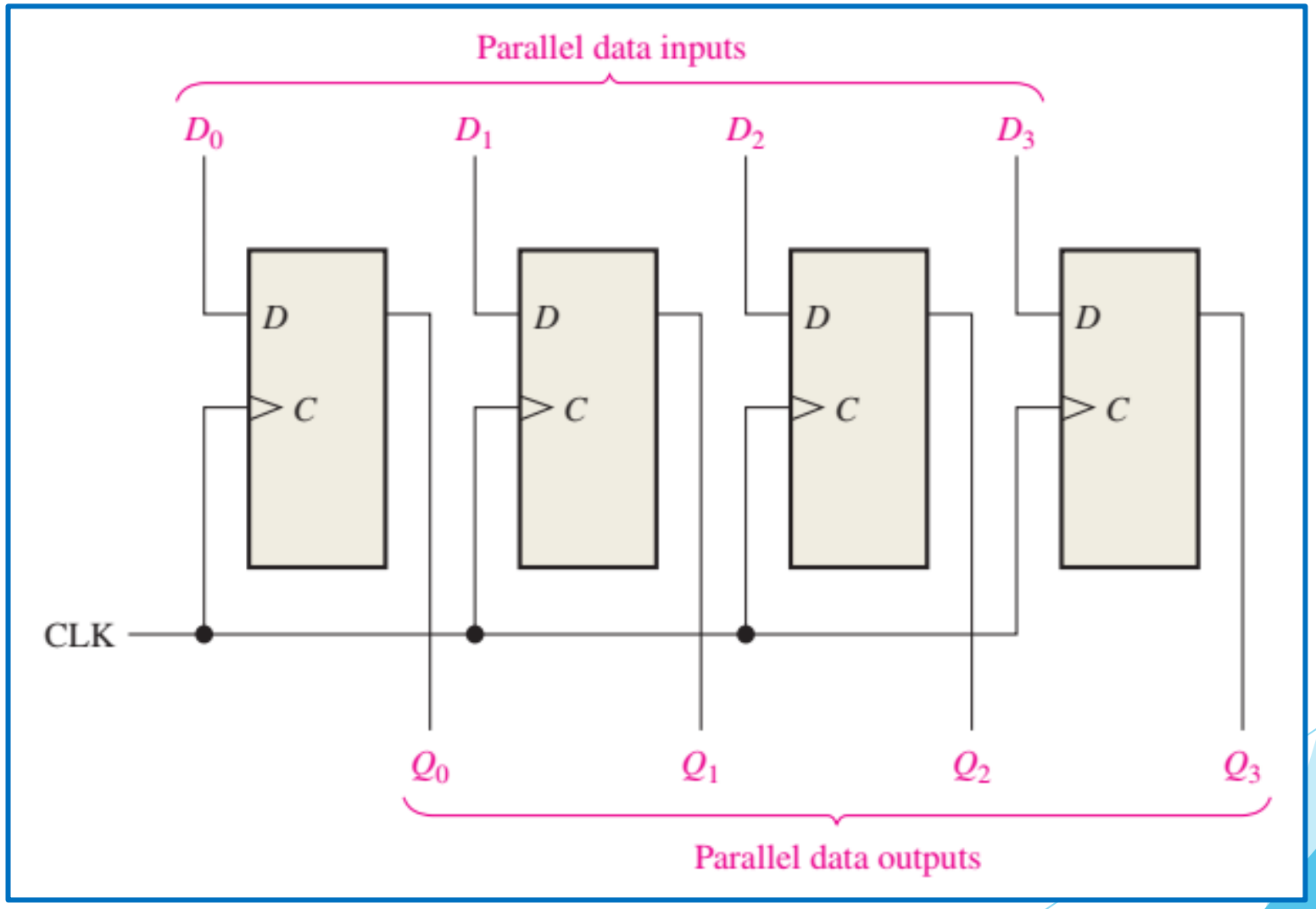
- **Example:** Show the data-output waveform for a 4-bit register with the parallel input data and the clock and SHIFT/$\overline{\text{LOAD}}$ waveforms given in Figure below.





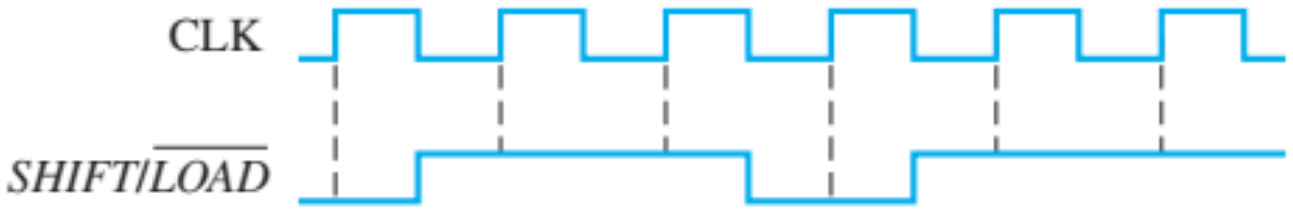
- On clock pulse 1, the parallel data (D_0 D_1 D_2 D_3 = 1010) are loaded into the register, making Q3 a 0. On clock pulse 2 the 1 from Q2 is shifted onto Q3; on clock pulse 3 the 0 is shifted onto Q3; on clock pulse 4 the last data bit (1) is shifted onto Q3; and on clock pulse 5, all data bits have been shifted out.

Parallel In/Parallel Out Shift Registers



Questions

- Design a 5-bit parallel in/serial out shift register?
- Show the data-output waveform of the 5-bit register designed in above for the specified SHIFT/$\overline{\text{LOAD}}$ and CLK inputs as shown in below. The parallel data inputs are D0 = 1, D1 = 0, D2 = 1, and D3 = 0, D4 = 0.



- Design a 6-bit parallel in/parallel out shift register?

الجامعة التقنية الجنوبية
المعهد التقني القرنـــــة
قسم تقنيات انظمة الحاسوب

التصميم المنطقي
Logic Design

محاضرة رقم (14)

د. حيدر احمد عبدالمحسن

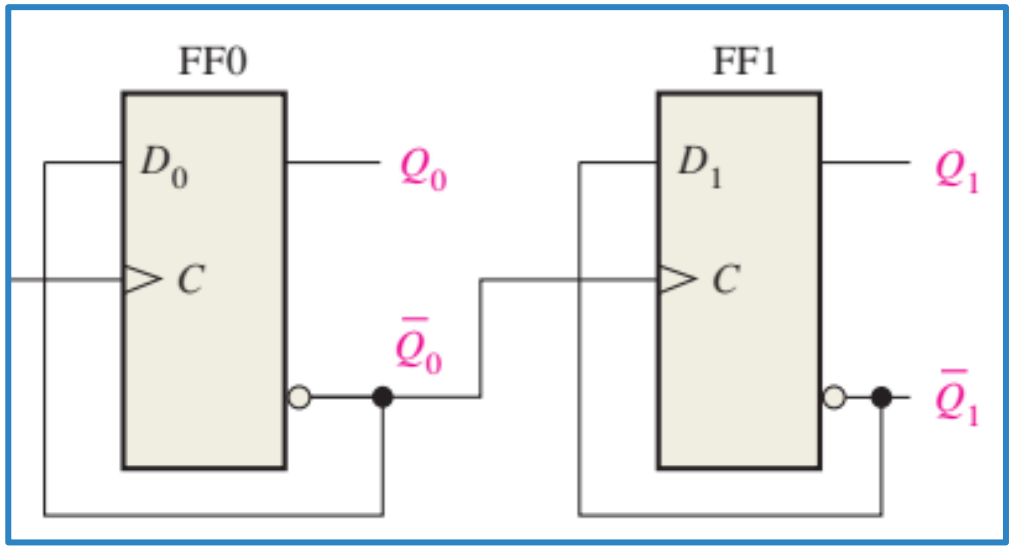
Counters

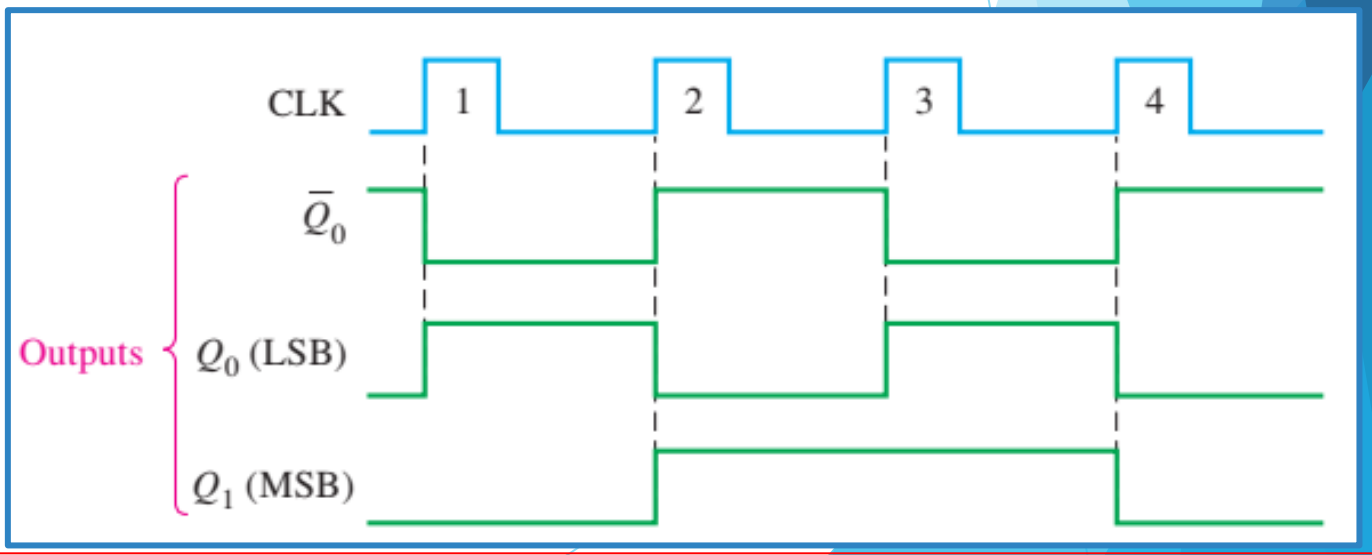
- Counters are used in a variety of digital applications as a way of counting events.
- Flip-flops can be connected to perform counting operations. Such a group of flip-flops is a counter, which is a type of finite state machine.
- Counters are classified into two broad categories according to the way they are clocked: asynchronous and synchronous.
- In asynchronous counters, the first flip-flop is clocked by the external clock pulse and then each successive flip-flop is clocked by the output of the preceding flip-flop. In synchronous counters, the clock input is connected to all the flip-flops so that they are clocked simultaneously.
- Synchronous counters are categorized in various ways. For example: Modulus counter, Decade counter, Ring counter, Johnson counter and Gray code counter.

Asynchronous Counters

- An asynchronous counter is one in which the flip-flops (FF) within the counter do not change states at exactly the same time because they do not have a common clock pulse.
- Figure below shows a 2-bit counter connected for asynchronous operation.
- Notice that CLK is applied to the clock input (C) of only the first flip-flop, FF0. The second flip-flop, FF1, is triggered by the $\overline{Q}_0$ output of FF0.

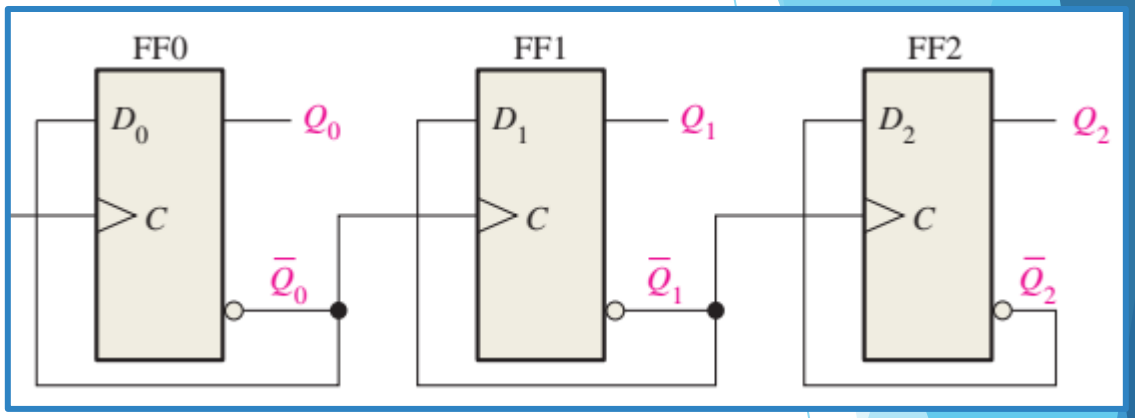
Clock Pulse	Q_1	Q_0
Initially	0	0
1	0	1
2	1	0
3	1	1
4 (recycles)	0	0



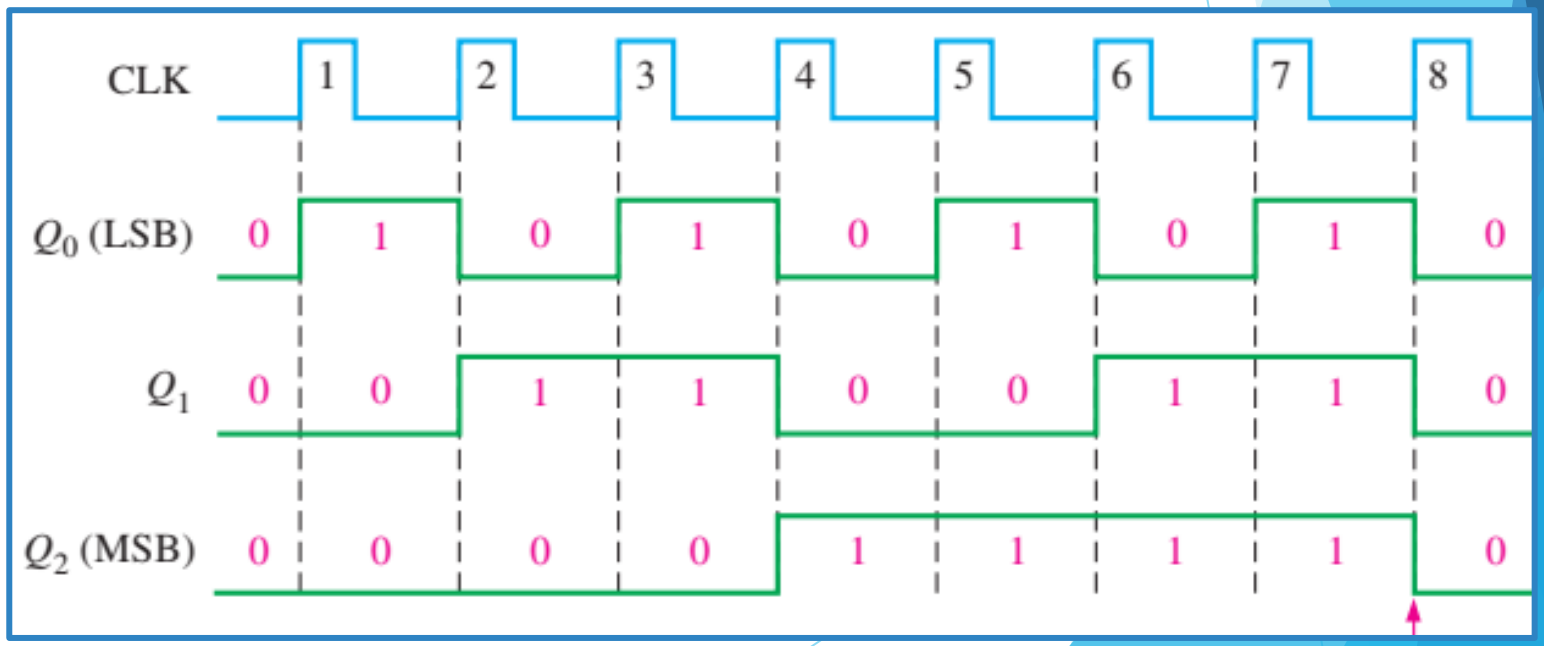


A 3-Bit Asynchronous Binary Counter

- The basic operation is the same as that of the 2-bit counter except that the 3-bit counter has eight states, due to its three flip-flops.

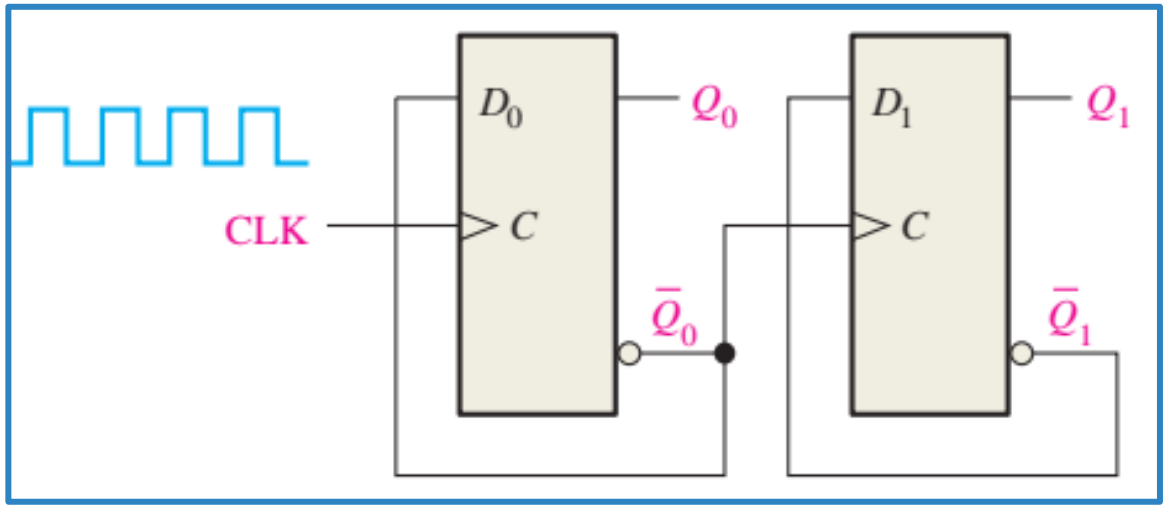


Clock Pulse	Q_2	Q_1	Q_0
Initially	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1
8 (recycles)	0	0	0



Questions

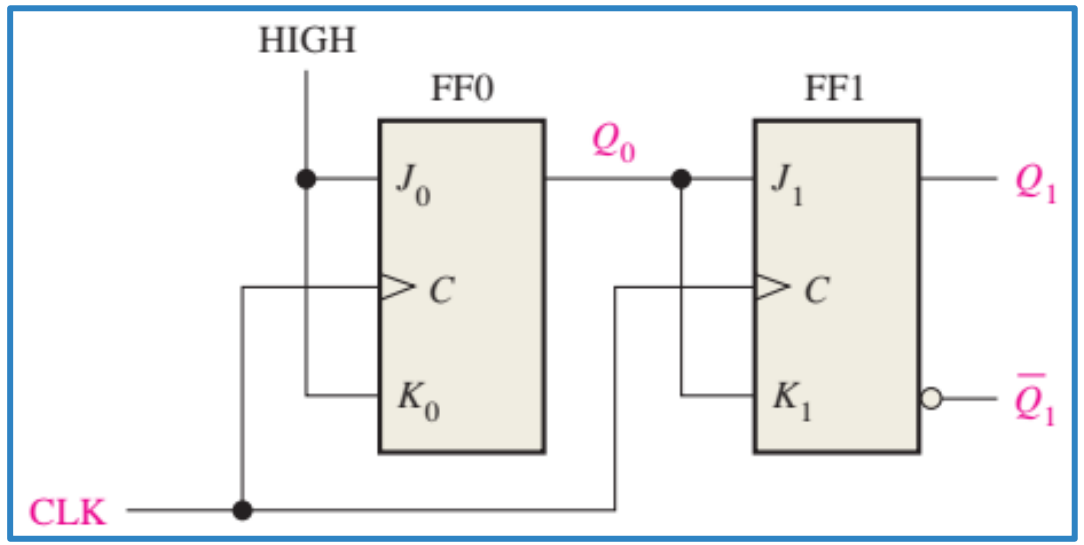
- For the ripple (asynchronous) counter shown in Figure below, show the complete timing diagram for eight clock pulses, showing the clock, Q0, and Q1 waveforms.

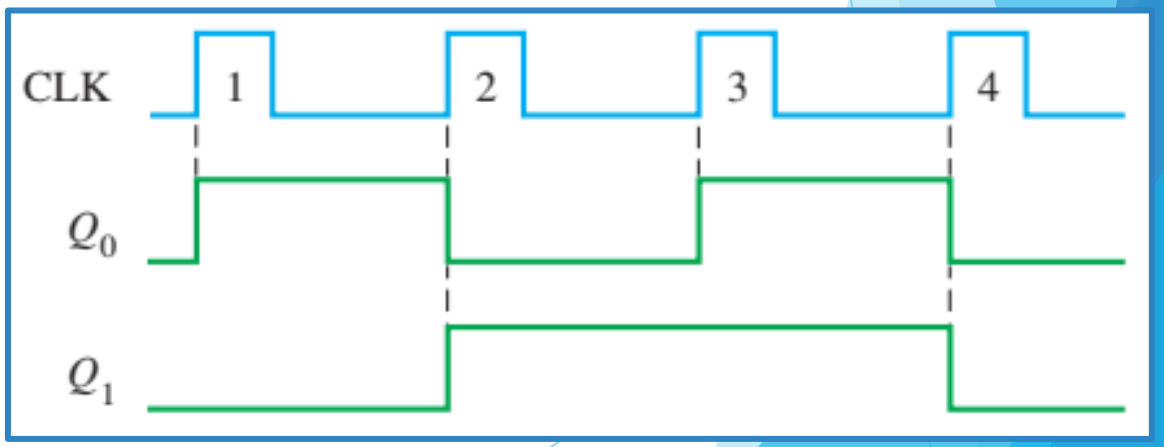


- Design a 3-bit ripple counter, show the complete timing diagram for sixteen clock pulses. Show the clock, Q0, Q1, and Q2 waveforms.
- Design a 4-bit asynchronous binary counter and show its timing diagram for 16 clock pluses (one cycle)?

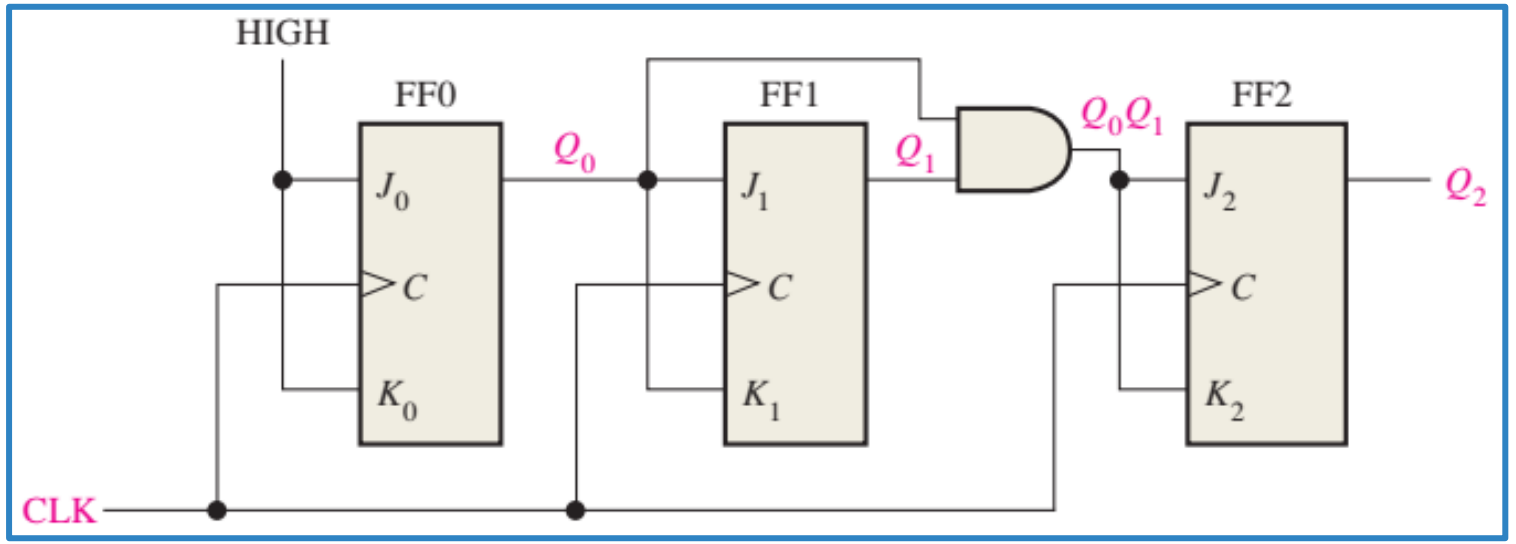
Synchronous Counters

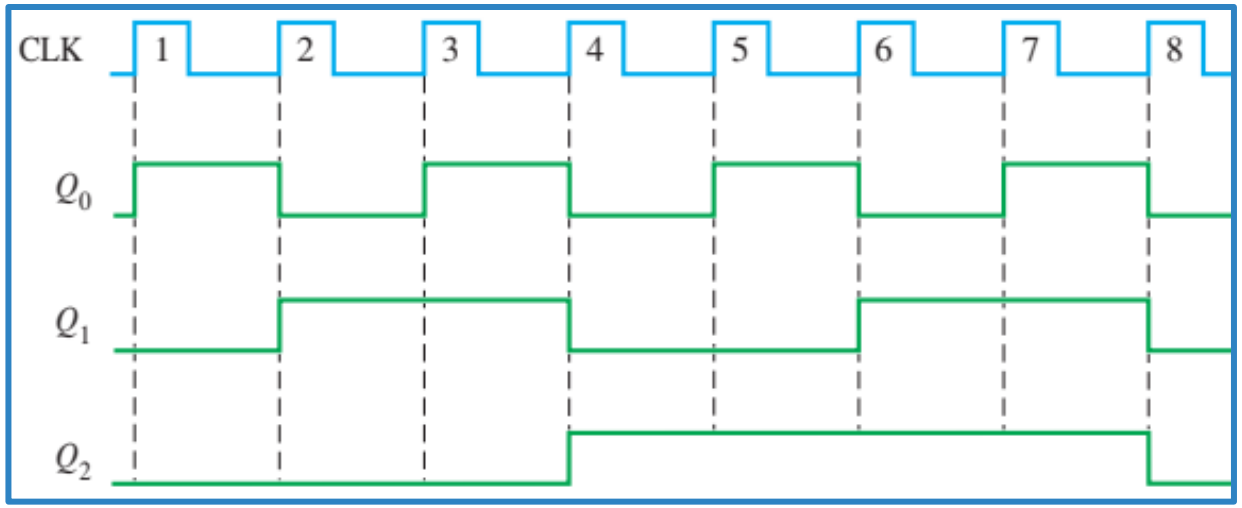
- A synchronous counter is one in which all the flip-flops in the counter are clocked at the same time by a common clock pulse. J-K flip-flops are used to illustrate most synchronous counters. D flip-flops can also be used but generally require more logic.
- Figure below shows a 2-bit synchronous binary counter. Notice that an arrangement different from that for the asynchronous counter must be used for the J1 and K1 inputs of FF1 to achieve a binary sequence.





A 3-Bit Synchronous Binary Counter

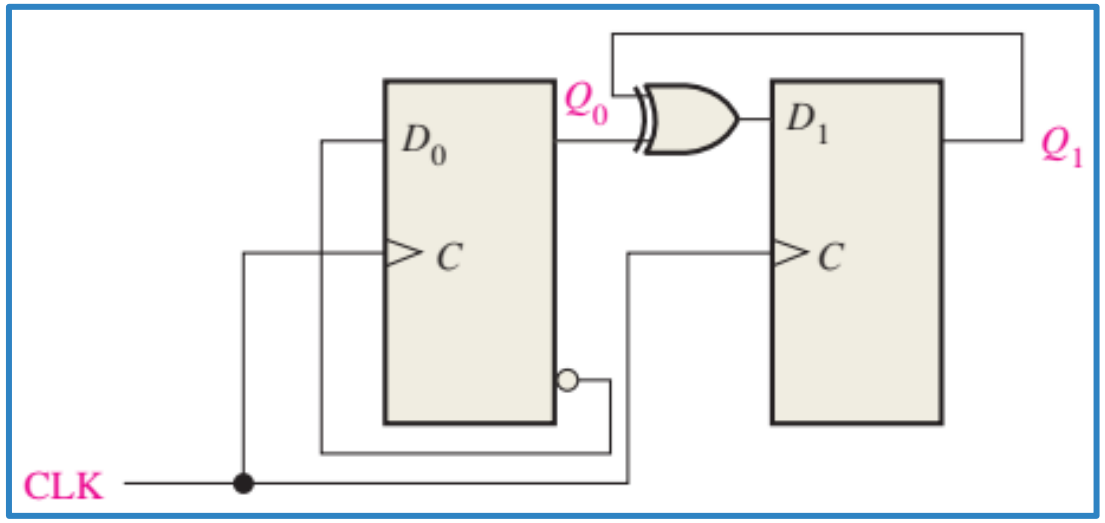




State sequence for a 3-bit binary counter.

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

- For the synchronous counter shown in Figure below, show the complete timing diagram for eight clock pulses.



- Design a 3-bit synchronous counter, show the complete timing diagram for sixteen clock pulses.
- Design a 4-bit synchronous binary counter and show its timing diagram for 16 clock pluses (one cycle)?
- Design a 5-stage synchronous binary counter and show the complete timing diagram. Verify that the waveforms of the Q outputs represent the proper binary number after each clock pulse.

Questions