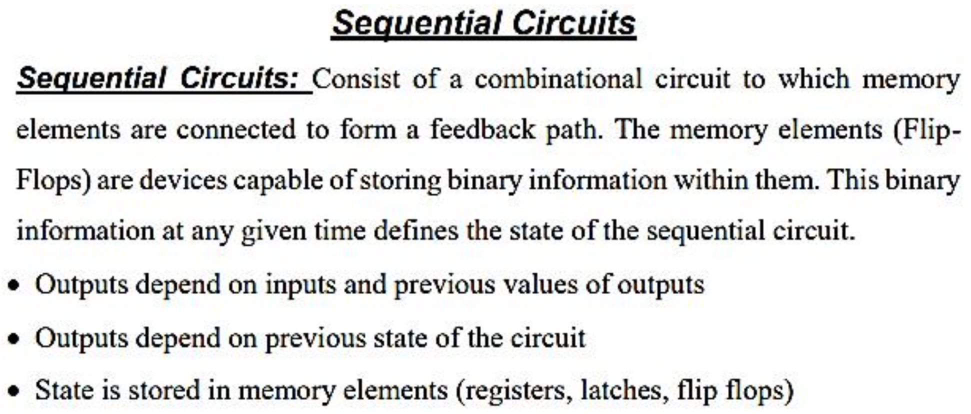
Digital Electronics **Assi.Pro. Yussra Malalah Abdulla**
Second class **Physics Science 2023 - 2024**

(Combinational Logic Circuits) الدوائر المنطقية الترابطية

Introduction to combinational circuits:

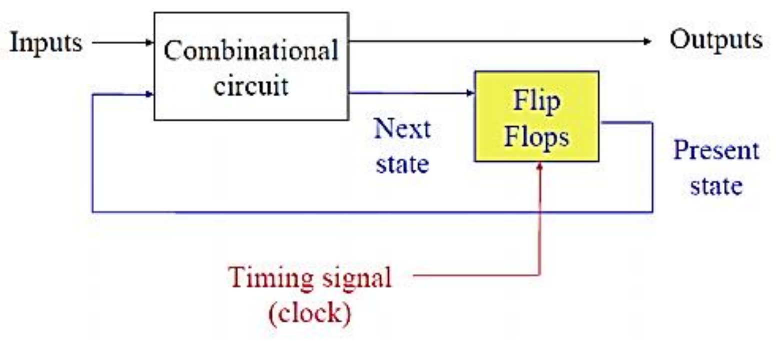
A combinational circuit is the digital logic circuit in which the output depends on the combination of inputs at that point of time with total disregard to the past state of the inputs. The digital logic gate is the building block of combinational circuits. The function implemented by combinational circuit is depend upon the Boolean expressions. On the other hand. **Sequential logic circuits,** consists of both logic gates and memory elements such as flip-flops. Figure below shows the combinational circuit having n inputs and m outputs. The n number of inputs shows that there are 2^n possible combinations of bits at the input. Therefore, the output is expressed in terms m Boolean expressions.



Sequential Circuits

Sequential Circuits: Consist of a combinational circuit to which memory elements are connected to form a feedback path. The memory elements (Flip-Flops) are devices capable of storing binary information within them. This binary information at any given time defines the state of the sequential circuit.

- Outputs depend on inputs and previous values of outputs
- Outputs depend on previous state of the circuit
- State is stored in memory elements (registers, latches, flip flops)



Adder

It is a combinational circuit that performs arithmetic addition according to the binary addition rules. There are two types of adder: **half adder and full adder.**

Half- Adder

It is a combination circuit that performs the arithmetic addition of two bits. This circuit needs two binary inputs and two binary outputs. The input variables consist of the two binary bits to be added; and the output consists of the sum of these two bits and a carry bit. When adding two binary numbers whose sum is greater than one bit in length, a carry must be generated (a one added to one results in sum of zero and a carry of one).

The truth table that identifies exactly the function of half adder is shown in table

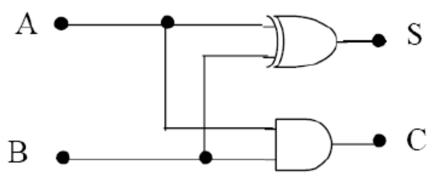
A	B	C_{out}	Σ
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Half adder truth table

The simplified Boolean function for the two outputs can be obtained from truth table. The simplification of the outputs of half adder are:

$S = \overline{A}B + A\overline{B}$
$S = A \oplus B$

The Carryout : C = AB , The half adder circuit is figure



Full Adder

A full adder is a combinational circuit that performs the arithmetic sum of three input bits. It consists of three inputs and two outputs. Two of the inputs variables, denoted by A and b, represent the two bits to be added. The third input, denoted by C, represents the carry from the previous lower are the position. The two outputs sum and carry.

Input			**Output**	
A	B	C	Sum	Carry
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Full adder truth table

The implementation of full adder can be done by using the following Boolean functions.

$$S = A \oplus B \oplus C_i$$

$$C_o = AB + (A+B)C_i$$

$$\text{Sum} = \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC$$

$$= \bar{A}(\bar{B}C + B\bar{C}) + A(\bar{B}\bar{C} + AB)$$

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

$$= A \oplus B \oplus C.$$

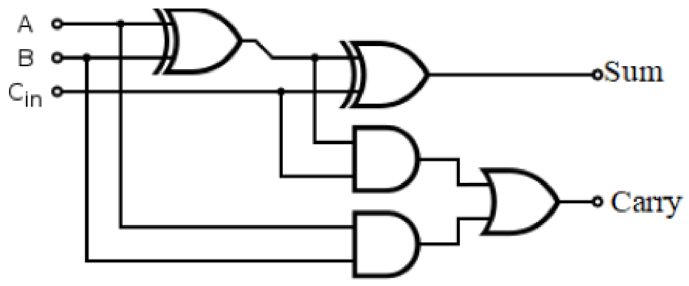
$$\text{Carry} = \bar{A}BC + A\bar{B}C + AB\bar{C} + ABC$$

$$= C\,(\bar{A}B + A\bar{B}) + AB(\bar{C} + C)$$

$$= C\,(A \oplus B) + AB(1)$$

$$= C\,(A \oplus B) + AB$$

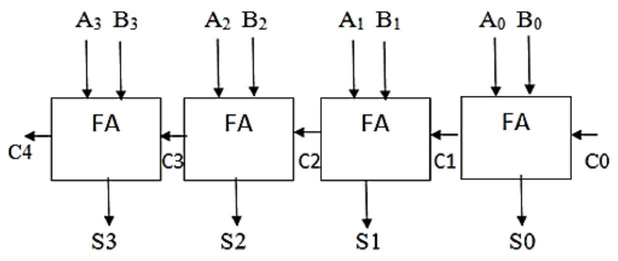
Full binary adder can combined from two half adder taking the OR of their carry with the sum of the second being as the output. The circuit of full adder is shown



Full adder circuit

<u>Binary parallel adder</u> 4-bit full adder

A binary parallel adder is a digital circuit that generates the arithmetic sum of two binary numbers in parallel. It consists of full adders connected in cascade, with the output carry from one full adder connected to the input carry of the next full adder. Fig show the 4- bit binary parallel adder.



Subtractor

The Subtractor is a combinational logic circuit that performs the arithmetic subtraction operation of number. There are two types of Subtractor: half subtractor and full subtractor.

Half Subtractor

A half subtractor is a combinational circuit that subtracts two bits and produces their difference. It has two outputs. One output generates the difference and will be designated by the symbol D. The second designate B for borrowed. The truth table for the half subtractor is shown in table The Boolean functions for the two outputs of the half subtractor are:

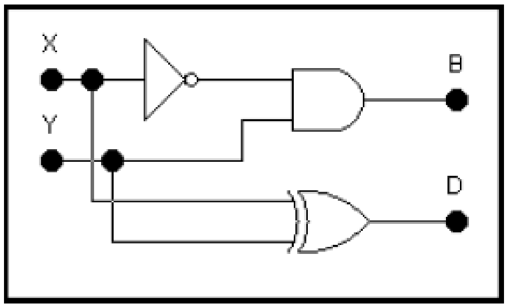
The question for differential

$D = A\bar{B} + \bar{A}B = A \oplus B$ XOR

$B = \bar{A} \cdot B$ مع عاكس AND

Input		Output	
A	B	Bo	Di.
0	0	0	0
0	1	1	1
1	0	0	1
1	1	0	0

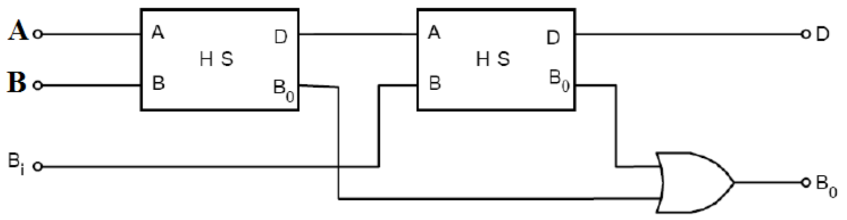
Half subtractor truth table

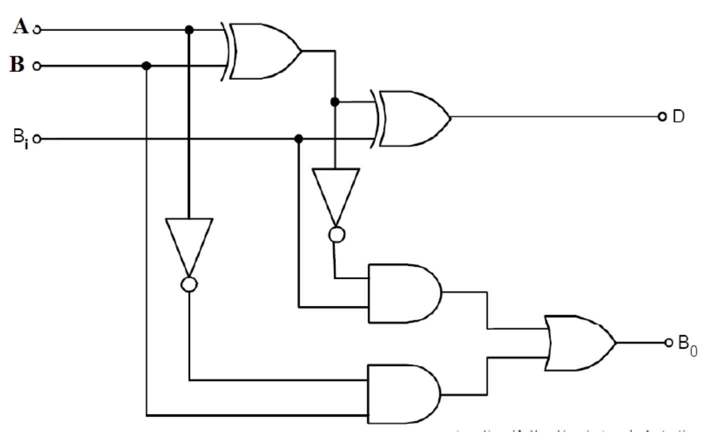


The logic circuit of half subtractor

Full Subtractor

A full subtractor is a combinational circuit that performs a subtraction between two bits, taking in account that a 1 may have been borrowed by a lower significant stage. This circuit has three inputs and two out puts. The two outputs, D and B, represent the difference and borrow, respectively. The truth table for the full
subtractor is shown in figure





The logic circuit of Full - subtractor

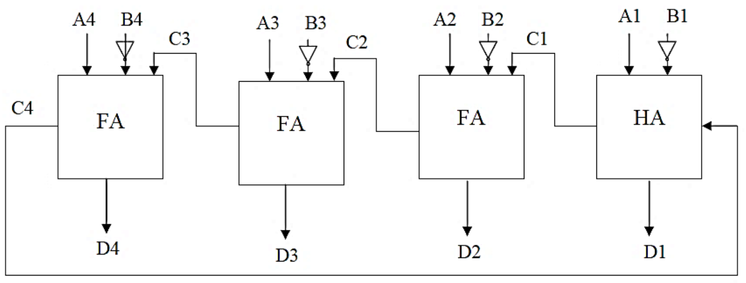
Input			Output	
A	B	Bi	D	Bo
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

Full subtractor truth table

The two outputs of the full subtractor are:

$$D = A \oplus B \oplus B_i$$

$$B_o = B_i + (A \oplus \overline{B})\,AB$$



4 bit Full - subtractor

Digital Comparator

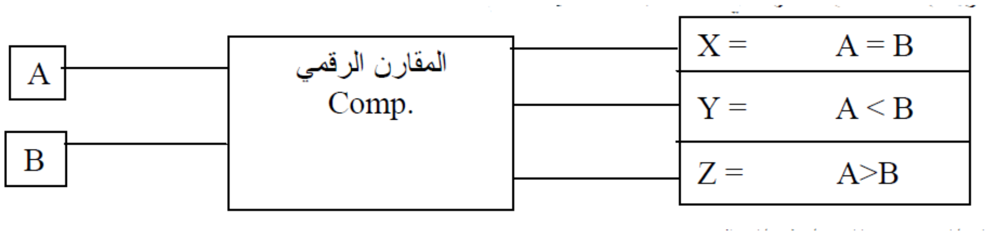
The comparator is a numerical circuit whose function is to compare two or more binary quantities for the purpose of finding a relationship between these two.

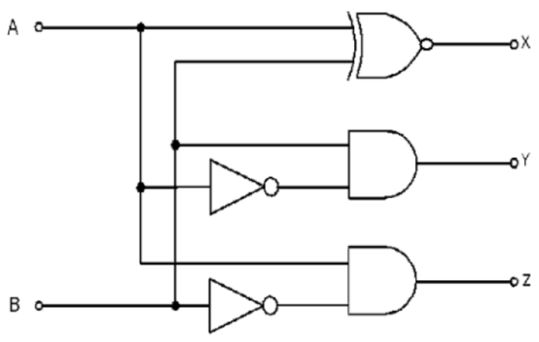
The two quantities. The comparison indicates whether these two quantities or binary numbers are equal or different. An **XNOR gate** is a basic comparator, because its output is "1" only if its two input bits are equal. For more on the comparison process, we can design a circuit containing two inputs to compare two numbers that make up each one They have 1 bit and 3 outputs, each of which indicates whether the first bit is greater than, equal to or less than a bit The second is what is shown in the figure:

$$X = \overline{A}\,\overline{B} + AB = \overline{A \oplus B}$$

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

$$Z = A\overline{B}$$

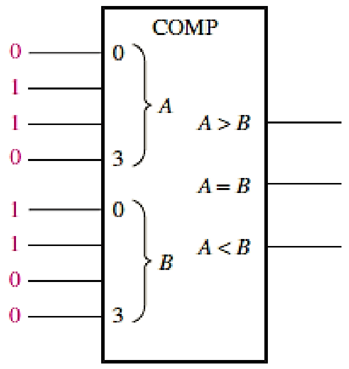


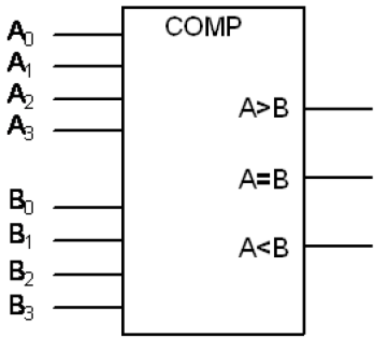


A	B	X A=B	Y A<B	Z A>B
0	0	1	0	0
0	1	0	1	0
1	0	0	0	1
1	1	1	0	0

Logic circuit for comparator Truth table 2 bit cpmparator

<u>4- bit comparator</u>





Determine the $A = B$, $A > B$, and $A < B$ outputs for the input numbers shown on the comparator in Figure .

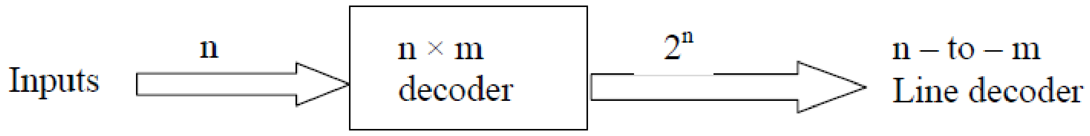
The number on the A inputs is 0110 and the number on the B inputs is 0011. The $A > B$ output is HIGH and the other outputs are LOW.

Decoder محلل الشفرة

وحدة فك التشفير هي دائرة تكتشف وجود عدد أو كلمة ثنائية محددة. المدخل إلى مفكك التشفير هو رقم ثنائي متوازي والمخرج عبارة عن إشارة ثنائية تشير إلى وجود أو عدم وجود هذا الرقم المحدد

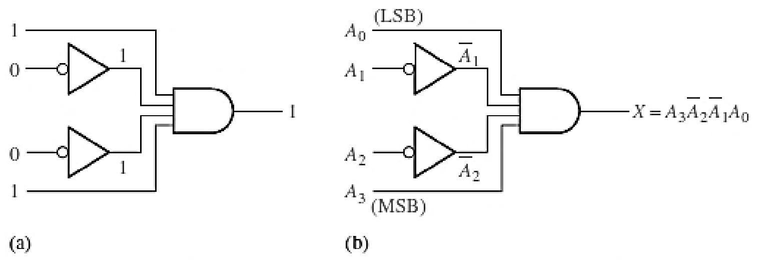
A decoder is a combinational circuit that converts binary information from n input lines to a maximum of 2^n unique output lines. If the n-bit have less than 2^n outputs.. The decoders represented here are called n –to –m line decoders.

The AND gate can be used as a basic decoder circuit, since the **AND gate's output** will be a binary one only when all inputs are binary one. Proper connections of AND gate's inputs to the data will ensure detection of any binary number.



The Basic Binary Decoder

Suppose you need to determine when a binary 1001 occurs on the inputs of a digital circuit. An AND gate can be used as the basic decoding element because it produces a HIGH output only when all of its inputs are HIGH. Therefore, you must make sure that all of the inputs to the AND gate are HIGH when the binary number 1001 occurs; this can be done by inverting the two middle bits (the 0s), as shown in Figure



Decoding logic for the binary code 1001 with an active-HIGH output.

The logic equation for the decoder of Figure (a) is developed as illustrated in Figure (b). You should verify that the output is 0 except when $A_0 = 1$, $A_1 = 0$, $A_2 = 0$, and $A_3 = 1$ are applied to the inputs. A_0 is the LSB and A_3 is the MSB. In the representation of a binary number or other weighted code in this book, the LSB is the right-most bit in a horizontal arrangement and the topmost bit in a vertical arrangement, unless specified otherwise. If a NAND gate is used in place of the AND gate in Figure, a LOW output will indicate the presence of the proper binary code, which is 1001 in this case.

Example:

Determine the logic required to decode the binary number 1011 by producing a HIGH level on the output.

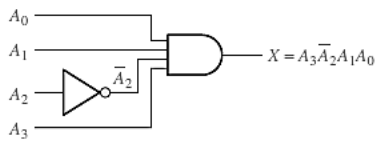
<u>Solution</u>

The decoding function can be formed by complementing only the variables that appear as 0 in the desired binary number, as follows:

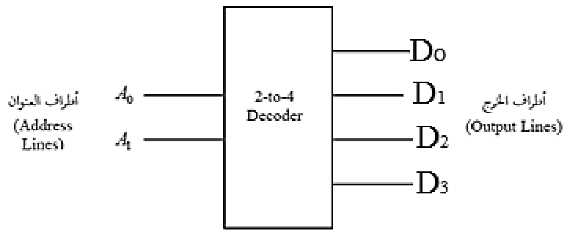
$$X = A_3\overline{A}_2A_1A_0 \qquad (1011)$$

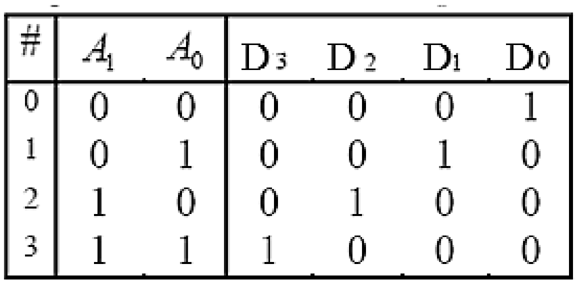
This function can be implemented by connecting the true (un complemented) variables

A_0, A_1, and A_3 directly to the inputs of an AND gate, and inverting the variable A_2 before applying it to the AND gate input. The decoding logic is shown in Figure



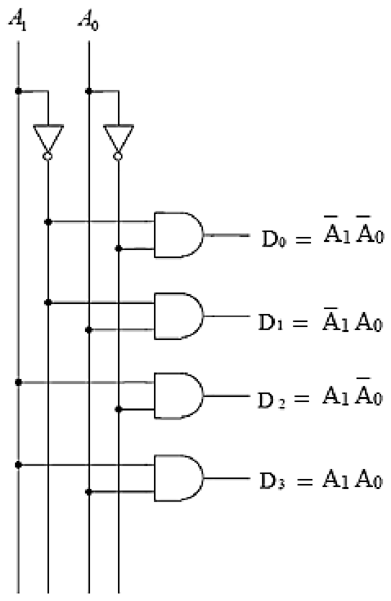
Decoding logic for producing a HIGH output when 1011 is on the



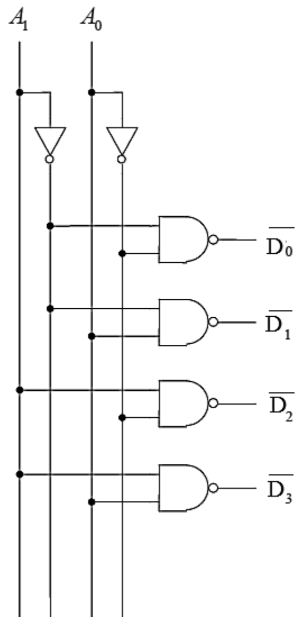


#	A_1	A_0	D_3	D_2	D_1	D_0
0	0	0	0	0	0	1
1	0	1	0	0	1	0
2	1	0	0	1	0	0
3	1	1	1	0	0	0

(2-to-4 Decoder) Active High



A_1
A_0
$D_0 = \overline{A}_1\overline{A}_0$
$D_1 = \overline{A}_1A_0$
$D_2 = A_1\overline{A}_0$
$D_3 = A_1A_0$



2 - 4 bit - decoder (Active HIGH) **2 - 4 bit - decoder (Active LOW)**

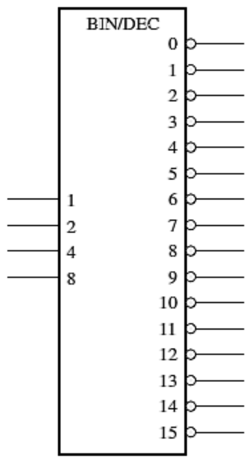
The 4-Bit Decoder

In order to decode all possible combinations of four bits, sixteen decoding gates are required (24 = 16). This type of decoder is commonly called either a *4-line-to-16-line* *decoder* because there are four inputs and sixteen outputs or a *1-of-16 decoder* because for any given code on the inputs, one of the sixteen outputs is activated. A list of the sixteen binary codes and their corresponding decoding functions is given in Table

Decimal Digit	Binary Inputs				Decoding Function	Outputs															
	A_3	A_2	A_1	A_0		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	0	0	$\overline{A}_3\overline{A}_2\overline{A}_1\overline{A}_0$	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	0	0	0	1	$\overline{A}_3\overline{A}_2\overline{A}_1A_0$	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	0	0	1	0	$\overline{A}_3\overline{A}_2A_1\overline{A}_0$	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
3	0	0	1	1	$\overline{A}_3\overline{A}_2A_1A_0$	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1
4	0	1	0	0	$\overline{A}_3A_2\overline{A}_1\overline{A}_0$	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1
5	0	1	0	1	$\overline{A}_3A_2\overline{A}_1A_0$	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1
6	0	1	1	0	$\overline{A}_3A_2A_1\overline{A}_0$	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1
7	0	1	1	1	$\overline{A}_3A_2A_1A_0$	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1
8	1	0	0	0	$A_3\overline{A}_2\overline{A}_1\overline{A}_0$	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1
9	1	0	0	1	$A_3\overline{A}_2\overline{A}_1A_0$	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1
10	1	0	1	0	$A_3\overline{A}_2A_1\overline{A}_0$	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1
11	1	0	1	1	$A_3\overline{A}_2A_1A_0$	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1
12	1	1	0	0	$A_3A_2\overline{A}_1\overline{A}_0$	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1
13	1	1	0	1	$A_3A_2\overline{A}_1A_0$	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1
14	1	1	1	0	$A_3A_2A_1\overline{A}_0$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1
15	1	1	1	1	$A_3A_2A_1A_0$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0

Decoding functions and truth table for a 4-line-to-16-line (1-of-16) decoder with **active-LOW** outputs.

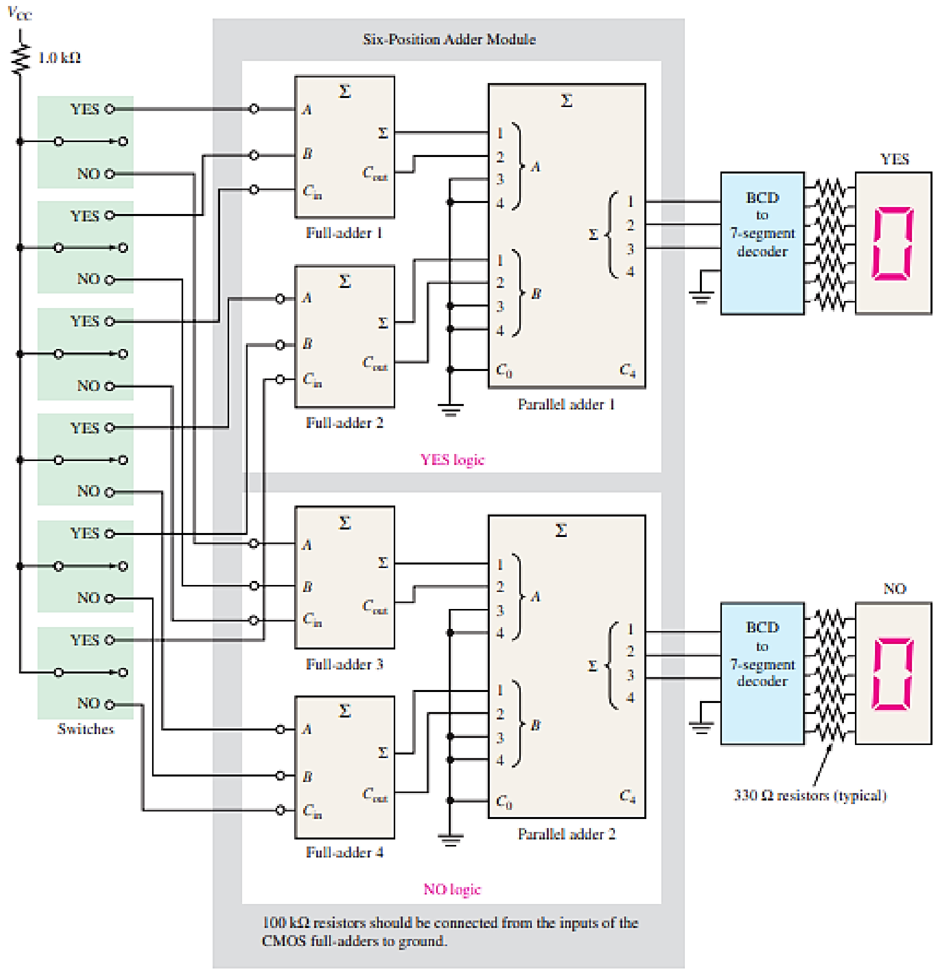
If an active-LOW output is required for each decoded number, the entire decoder can be implemented with NAND gates and inverters. In order to decode each of the sixteen binary codes, **sixteen NAND gates are required** (**AND gates can be used to produce active-HIGH** outputs).
A logic symbol for a 4-line-to-16-line (1-of-16) decoder with active-LOW outputs is shown in Figure. The BIN/DEC label indicates that a binary input makes the corresponding decimal output active. The input labels 8, 4, 2, and 1 represent the binary weights of the input bits ($2^3 2^2 2^1 2^0$)



لدائرة حلال الشفرة استخدامات عديدة، إلا أن أهم تلك الاستخدامات هو استخدامه في دوائر الذاكرة (Memory)، بأنواعها المختلفة، للوصول إلى موقع معين من مواقع الذاكرة عن طريق عنوانه. فلكل موقع من مواقع الذاكرة عنوان (Address) خاص به، و للوصول إلى ذلك الموقع يتم وضع عنوانه على أطراف العنوان لدائرة فك الشفرة، فينشط طرف الخرج في الجهاز المتصل بذلك الموقع و يقوم بفتح الموقع لعمليات القراءة (Read) أو الكتابة (Write). أي أن مهمة هذه الدائرة هي الربط ما بين مواقع الذاكرة و عناوينها.

Logic symbol for a 4-line-to-16-line (1-of-16) decoder

Application of Full-Adder and Decoder



A voting system using full-adders and parallel binary adders.

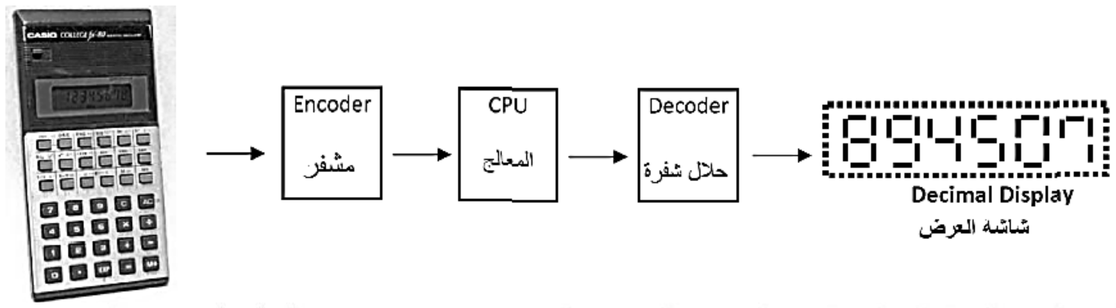
In Figure each full-adder can produce the sum of up to three votes. The sum and output carry of each full-adder then goes to the two lower-order inputs of a parallel binary adder. The two higher-order inputs of the parallel adder are connected to ground (0) because there is never a case where the binary input exceeds 0011 (decimal 3). For this basic 6-position system, the outputs of the parallel adder go to a BCD-to-7-segment decoder that drives the 7-segment display. As mentioned, additional circuits must be included when the system is expanded. The resistors from the inputs of each full-adder to ground assure that each input is LOW when the switch is in the neutral position (CMOS logic is used). When a switch is moved to the “yes” or to the “no” position, a HIGH level (VCC) is applied to the associated full adder input

Encoders المشفر

An encoder is a combinational logic circuit that essentially performs a **“reverse”** decoder function. An encoder accepts an active level on one of its inputs representing a digit, such as a decimal or octal digit, and converts it to a coded output,

إن احد استخدامات البوابات المنطقية في المنظومات الرقمية هو استخدامها كمحولات الشفرة (Code Converters) .
الشفرات الشائعة الاستخدام هي : الثنائي والثماني والسداسي عشر والعشري المشفر ثنائيا .

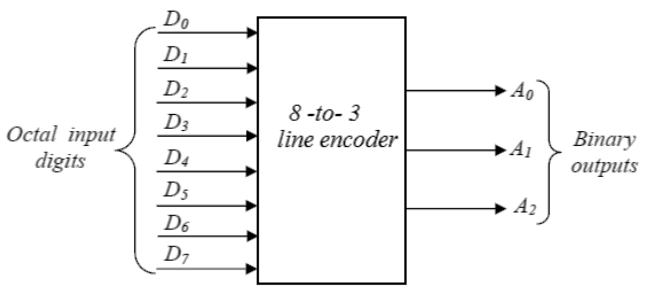
إن معظم الغموض الذي يكتنف الحاسبات الالكترونية وغيرها من النظم الرقمية يأتي من اللغة غير المألوفة للدوائر الرقمية ، فالاجهزة الرقمية يمكنها إن تتعامل فقط مع الارقام الثنائية (1و0) حيث انه من الصعب على الانسان إن يفهم سلسلة طويلة من الواحدات والاصفار ولهذا السبب فان محولات الشفرة تصبح ضرورية للتحويل من لغة البشر الى لغة الالة وبالعكس .



نلاحظ في الرسم التخطيطي المبسط مراحل احدى الحاسبات الصغيرة حيث إن اداة الادخال الى اليسار هي لوحة المفاتيح الشائعة (Keyboard)، وبين لوحة المفاتيح ووحدة المعالجة المركزية (CPU) الخاصة بالحاسبة نجد المشفر حيث يقوم هذا المشفر بترجمة العدد العشري الذي يتم ادخاله بواسطة الضغط على احد ازرار لوحة المفاتيح الى شفرة ثنائية ثم تقوم وحدة المعالجة المركزية بتأدية وظائفها على الشفرة الثنائية الداخلة وبعد معالجتها تخرج على شكل نظام ثنائي يقوم بعد ذلك حلال الشفرة (Decoder) بترجمة هذه الشفرة الثنائية الخارجة من وحدة المعالجة المركزية الى شفرة عشرية وفي هذه المنظومة يكون المشفر وحلال الشفرة عبارة عن مترجمات شفرة الكترونية ويمكننا إن نتصور المشفر على انه اداة الترجمة من لغة البشر الى لغة الالة ، اماحلال الشفرات فيقوم بعملية معكوسة حيث يقوم بالترجمة من لغة الالة الى لغة البشر .

المشفر هو عبارة عن دائرة منطقية توافقية بالأساس تؤدي عكس عمل دائرة محلل الشفرة. دخل المشفر عبارة عن عدد من الخطوط كل خط يمثل رقم (*digit*) مثل رقم عشري أو رقم ثماني، ويحول هذه الأرقام إلى خرج مشفرمثل الثنائي. والمشفرات تستطيع أيضاً أن تشفر الرموز المختلفة وحروف الهجاء. عملية التحويل من الرموز والأعداد المعتادة إلى الشكل المشفر يطلق عليها عملية التشفير.

الشكل يوضح المخطط الصندوقي لدائرة مشفر يحول الأرقام الثمانية إلى مكافئها الثنائي. المشفر له ثماني مداخل وثلاثة مخارج ممثل المكافئ الثنائي للدخل.



Logic symbol for a Octal -to-Binary encoder

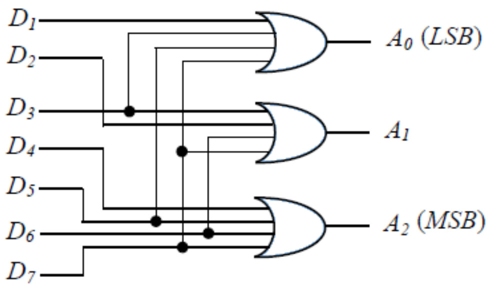
المدخلات	الخرج		
الأرقام الثمانية	A_2	A_1	A_0
D_0	0	0	0
D_1	0	0	1
D_2	0	1	0
D_3	0	1	1
D_4	1	0	0
D_5	1	0	1
D_6	1	1	0
D_7	1	1	1

Truth table for Octal -to-Binary encoder

From the truth table we get :

$$A_2 = D_4 + D_5 + D_6 + D_7$$

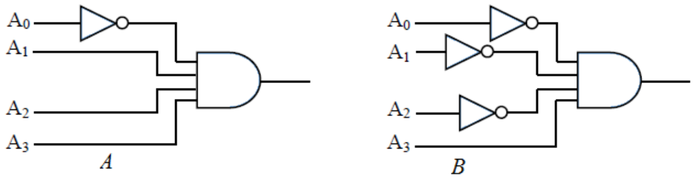
$$A_1 = D_2 + D_3 + D_6 + D_7$$
$$A_0 = D_1 + D_3 + D_5 + D_7$$



Logic circuit for Encoder

Home Work:

1) If the outputs is 1 ,what is the Binarty code appear in the input for each gate ?



. If a 1-of-16 decoder with active-LOW outputs exhibits a LOW on the decimal 12 output, what are the inputs?

(a) $A_3A_2A_1A_0 = 1010$ (b) $A_3A_2A_1A_0 = 1110$
(c) $A_3A_2A_1A_0 = 1100$ (d) $A_3A_2A_1A_0 = 0100$

Show the decoding logic for each of the following codes if an active-HIGH (1) output is required:

(a) 1101 (b) 1000 (c) 11011 (d) 11100