Faculty of Engineering & Technology
Electrical & Computer Engineering Department

Digital Electronics and Computer Organization Lab
ENCS 2110

Report #2

combinational circuit

Prepared by:

Maha Maher Mali	1200746

Partners:

Lana Thabit	1200071
Jana Herzallah	1201139

Instructor: Dr.Anjad Badran
Assistant: Eng. Mohammed Bassam

Section: 6

Date: 10/4/2022

Abstract

The aim of this experiment is to understand the construction and operating principle of digital comparators, construct comparators with basic gates and, implement half- and full adders using basic logic gates and construct half- and full- subtractor circuits. Also, understand the operating principles of Encoders, Decoders, Multiplexers and Demultiplexers. And to construct them using basic logic gates and ICs. And finally, using Multiplexer to implement Logic Functions.

Table of contents

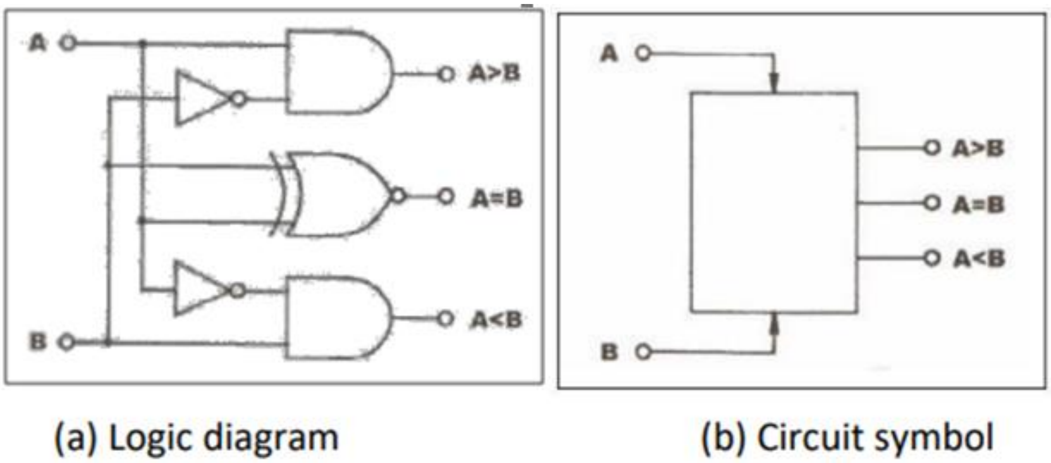
List of figures

List of tables

2.Theory

2.1Comparator circuit

Comparator is a combinational circuit used to compare at least two numbers or more than two numbers. The simplest form of the comparator has two inputs, if the two inputs are called A and B then there are three possible outputs: A>B, A=B and A<B. Also, comparator constructed from basic gates AND, Invertor, X-NOR.

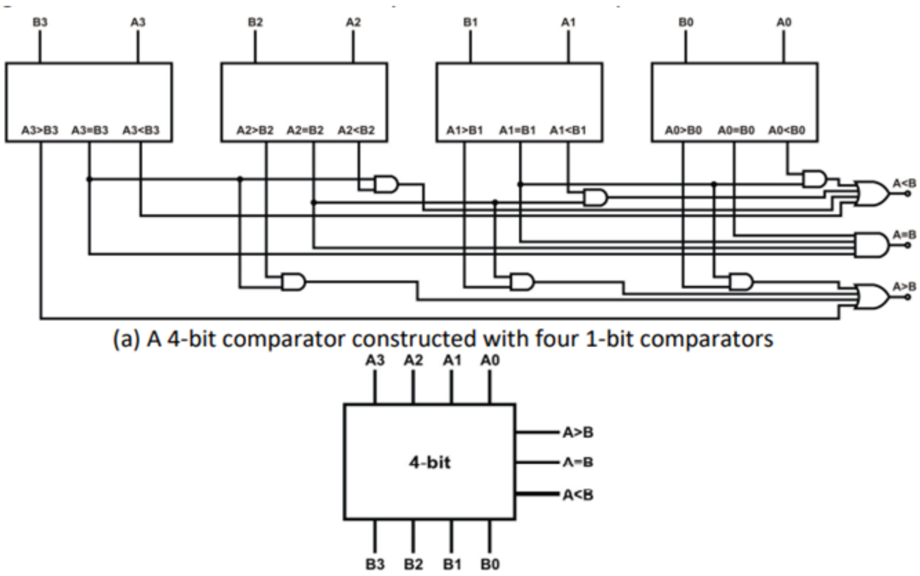


(a) Logic diagram

(b) Circuit symbol

Figure 1 comparator circuit

In a 4-bit comparator, each bit represents 2^0, 2^1, 2^2, and 2^3 . Comparison will start from the most significant bit (2^3), if input A is greater than input B at the 2^3 bit, the “A>B” output will be in the high state. If A and B are equal at the 2^3 bit, the comparison will be carried out at the next highest bit (2^2). If there is still no result at this bit the process is repeated again at the next bit. At the lowest bit (2^0), if the inputs are still equal then the “A=B” output will be in the high state.



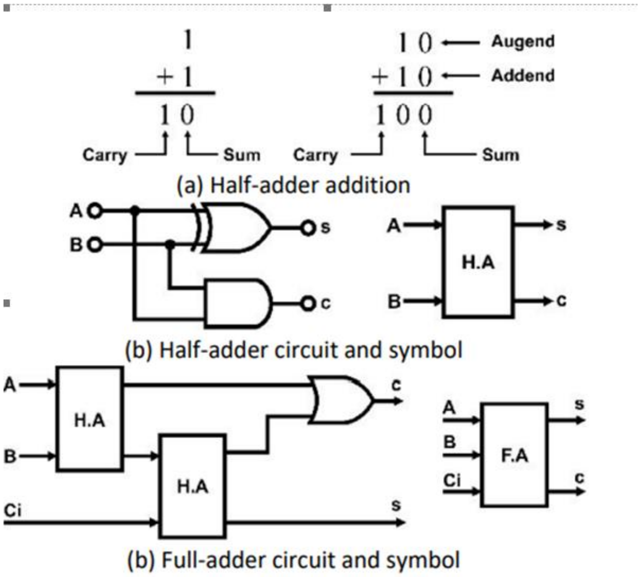
(a) A 4-bit comparator constructed with four 1-bit comparators

(b) Symbol of a 4-bit comparator

Figure 2 4-bit comparator

2.2 Half adder, full adder, and 4-bit Binary Adder Circuits

The combinational circuit that performs the addition of two bits is called half adder (HA). One that performs the addition of three bits (two significant bits and previous carry) is a full adder (FA). The result of addition is a "carry" and a" sum". In binary additions, a "carry" is generated when the sum of two numbers is greater than 1.



(a) Half-adder addition

(b) Half-adder circuit and symbol

(b) Full-adder circuit and symbol

Figure 3 Half-Full Adder

To perform the addition of numbers greater than 2-bitsin length, we use 4-bit binary adder the connection is shown in Figure 4, or "Parallel Input" should be used to generate sums simultaneously. However, the sum of the next adder will be stable only after the previous adder's carry has stabilized. For example, in Figure 4, the sum of FA2 will not be stable unless the carry of FA1 is stable.

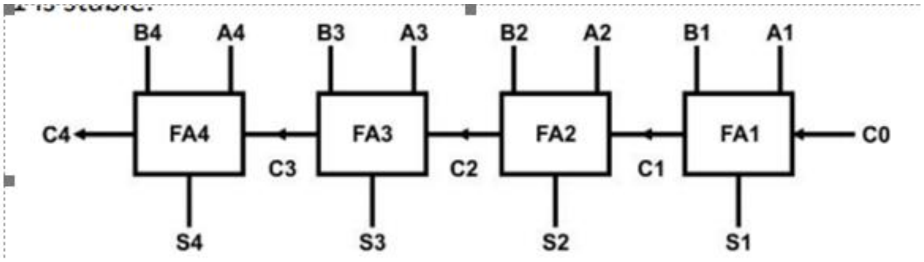


Figure 4 4-bit binary Adder

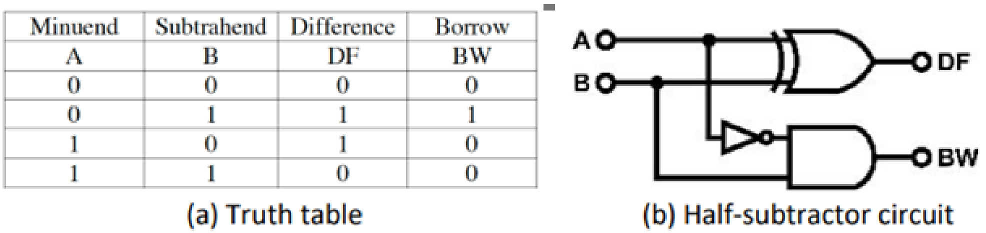
When FA1 adds A1 and B1, a sum S1 and a carry C1 is generated. C1 will be added to A2 and B2 by FA2, generating another sum S2 and another carry C2. In the case of

Figure 2.4, the sum of the four adders does not stabilize at the same time, delaying the adding process. This delay can be eliminated by using the “Look-Ahead” adder. Look-ahead adders do not have to wait for the previous adder to stabilize before performing the next addition, saving valuable time.

2.3 Half and Full Subtractor Circuits

Half-subtractor and full-subtractor circuits can be built by referring to the truth tables and the Boolean expressions. Binary subtraction is usually performed by using 2’s complement. Two steps are required to obtain 2’s complement. First, the subtrahend is inverted to 1’s complement, i.e., a “1” to a “0” and a “0” to a “1”. Secondly, a “1” is added to the least significant bit of the subtrahend in 1’s complement.

A half-subtractor performs the task if subtraction 1-bit at a time regardless of whether the minuend is greater or less than the subtrahend.



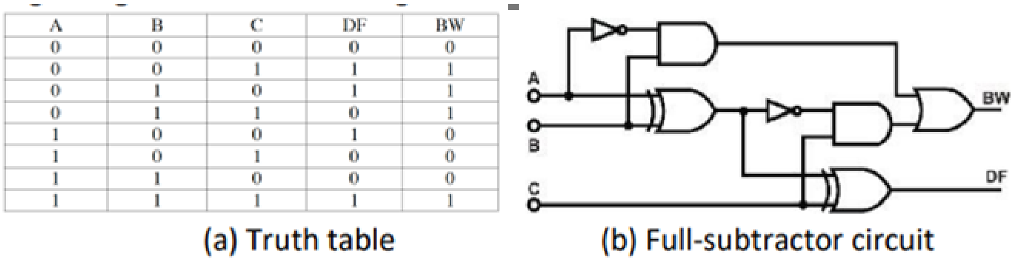
Minuend	Subtrahend	Difference	Borrow
A	B	DF	BW
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

(a) Truth table

(b) Half-subtractor circuit

Figure 5 Half subtractor

The full-subtractor has to consider borrow(s) from previous stages.



A	B	C	DF	BW
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

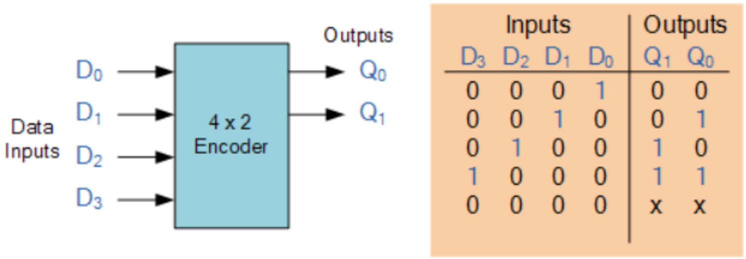
(a) Truth table

(b) Full-subtractor circuit

Figure 6 Full subtractor

2.4 Encoder

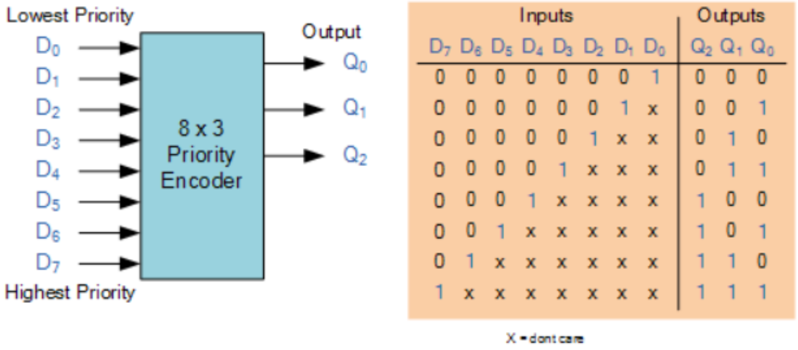
An Encoder is a combinational circuit that performs the reverse operation of Decoder. It has maximum of 2^n input lines and ‘n’ output lines, hence it encodes the information from 2^n inputs into an n-bit code. It will produce a binary code equivalent to the input, which is active High. Therefore, the encoder encodes 2^n input lines with ‘n’ bits. It is optional to represent the enable signal in encoders. The main disadvantage of the encoder is that it will cause a wrong output when more than input lines are at HIGH state.



Inputs				Outputs	
D_3	D_2	D_1	D_0	Q_1	Q_0
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1
0	0	0	0	x	x

Figure 7 Encoder

The problem mentioned above can be solved using priority encoder. A priority encoder provide n bits of binary coded output representing the position of the highest order active input of 2^n inputs. If two or more inputs are high at the same time, the input having the highest priority will take precedence.



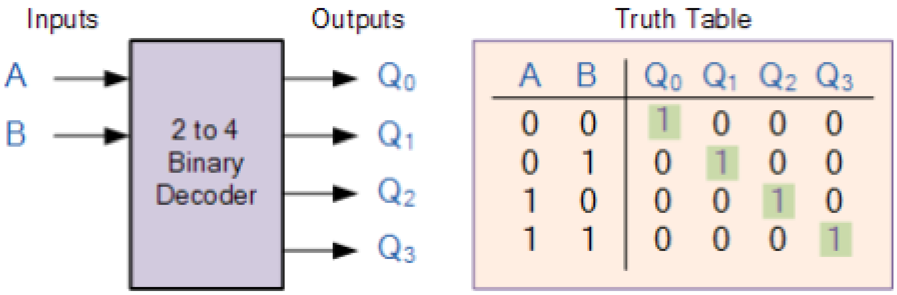
Inputs								Outputs		
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Q_2	Q_1	Q_0
0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	x	0	0	1
0	0	0	0	0	1	x	x	0	1	0
0	0	0	0	1	x	x	x	0	1	1
0	0	0	1	x	x	x	x	1	0	0
0	0	1	x	x	x	x	x	1	0	1
0	1	x	x	x	x	x	x	1	1	0
1	x	x	x	x	x	x	x	1	1	1

X = dont care

Figure 8 priority encoder

2.5 Decoder

A Decoder is a combinational circuit that converts binary information from n input lines to 2^n unique output lines. Apart from the Input lines, a decoder may also have an Enable input line. Also, one output of decoder is on and other output is off. The decoder is the dual of the encoder which is the decoder performs the reverse operation of encoder.

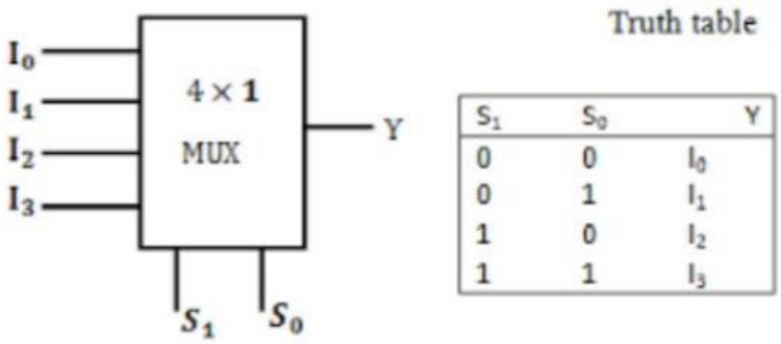


A	B	Q_0	Q_1	Q_2	Q_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

Figure 9 Decoder

2.6Multiplexer

It is a combinational circuit which have many data inputs and single output depending on control or selection lines, or we can say that for 2^n input lines, n selection lines are required. Multiplexers are also called as MUX.

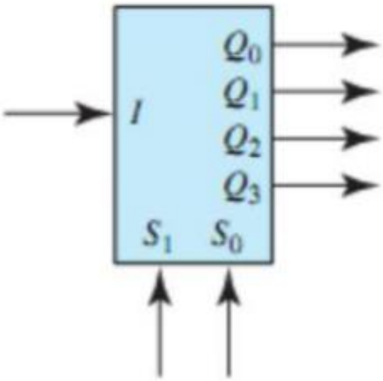


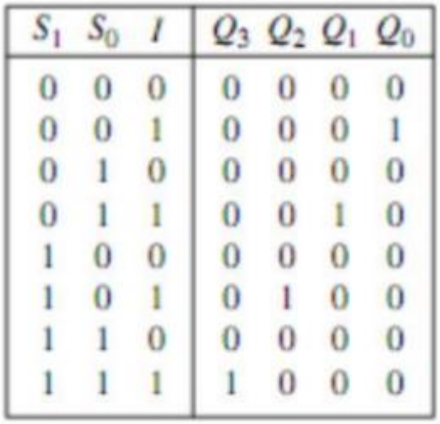
S_1	S_0	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

Figure 10 Multiplexers

2.2Demultiplexer

De-Multiplexer is a combinational circuit that performs the reverse operation of Multiplexer. It has single input, ‘n’ selection lines and maximum of 2^n outputs. The input will be connected to one of these outputs based on the values of selection lines. Since there are ‘n’ selection lines, there will be 2^n possible combinations of zeros and ones. So, each combination can select only one output. De-Multiplexer is also called as De-Mux.





S_1	S_0	I	Q_3	Q_2	Q_1	Q_0
0	0	0	0	0	0	0
0	0	1	0	0	0	1
0	1	0	0	0	0	0
0	1	1	0	0	1	0
1	0	0	0	0	0	0
1	0	1	0	1	0	0
1	1	0	0	0	0	0
1	1	1	1	0	0	0

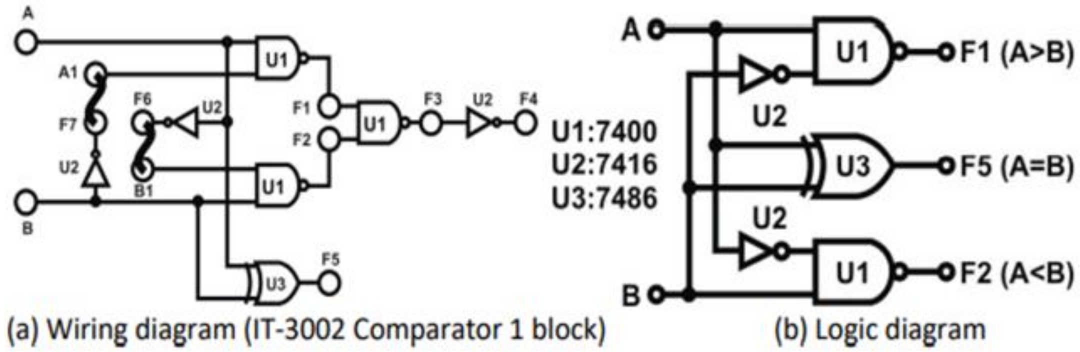
Figure 11 Demultiplexer

3.Procedure

3.1 Comparator circuits

A. Constructing Comparator with basic logic gates

The circuit in Figure 12 shown below was connected.



(a) Wiring diagram (IT-3002 Comparator 1 block)

(b) Logic diagram

Figure 12 comparator circuits

The inputs A and B were connected to Switches SW1 and SW2 respectively and the Outputs F1, F2, F5 were connected to logic indicators L1, L2, L3 respectively.

The input sequences in Table 1 for B and A were followed and the output

states were recorded and measured.

Table 1 comparator

Input			output		
SW2(B)	SW1(A)		F1	F2	F3
0	0	A=B	1	1	0
0	1	A>B	0	1	1
1	0	A<B	1	0	1
1	1	A=B	1	1	0

B. Constructing Comparator with TTL IC

The 74LS85(U5) 4-bit Comparator IC of module IT-3002 was used in this section.

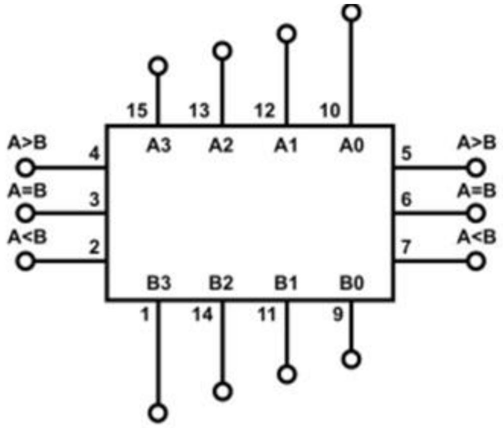


Figure 13 Comparator IC

The inputs A<B, A=B and A>B were connected to Switches SW1, SW2and SW3 respectively. Connected the inputs A0-A3 and B0-B3 of the 74LS85 to the rotary switch. The outputs F1, F2, F5 were connected to the logic indicators respectively L1,L2.

The input sequences in Table 2 for A<B, A=B and A>B were followed and the output states were recorded.

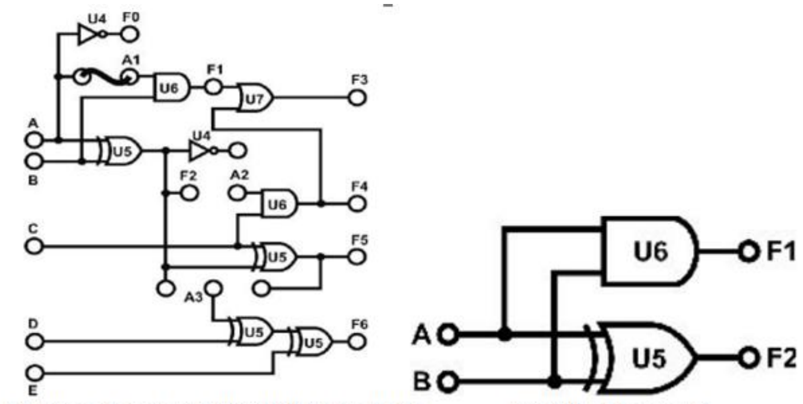
Table 2 Comparator IC

input			output		
A>B	A=B	A<B	A>B	A=B	A<B
0	0	1	0	0	0
0	1	0	0	1	0
0	1	1	0	1	0
1	0	0	1	0	0
1	0	1	0	0	0
1	1	1	0	1	0

3.2Half and full adder circuits

A. Constructing Half and full adders with basic logic gates

The circuit in Figure 14 shown below was connected



(a) Wiring diagram (IT-3003 Half-Adder block) (b) Half-Adder circuit

Figure 14 Half Adder

The inputs A and B were connected to Switches SW0 and SW1 respectively. Connected The Outputs F1, F2 to logic indicators L1, L2 respectively.

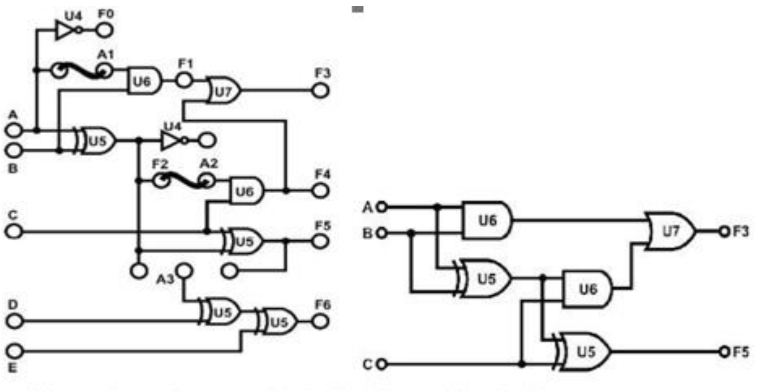
The input sequences in Table 3 for B and A were followed and the output

states were recorded and measured.

Table 3 Half Adder

input		output	
SW1(B)	SW0(A)	F1(CARRY)	F2(SUM)
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

The circuit in Figure 15 shown below was connected



(a) Wiring diagram (IT-3003 Full-Adder block) (b) Full-Adder circuit

Figure 15 Full Adder

The inputs A, B and C were connected to Switches SW1, SW2 and SW3 respectively.

Were A and B augends while C was the previous carry.

Connected The Outputs F3 to L1, F5 to L2. Followed the input sequences in Table 4 and recorded output states.

Table 4 Full Adder

input			output	
SW3(C)	SW2(B)	SW1(B)	F3(CARRY)	F5(SUM)
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

B. Constructing 4-Bit Full-Adder with IC

U9 on block Full-Adder of module IT-3003 was used as a 4-bit adder in this section.

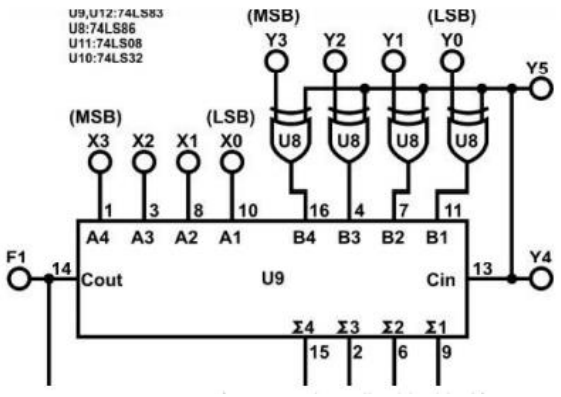


Figure 16 4-bit adder

The input Y5 was connected to SW0, so the XOR gates U8, which was connected to Y0~Y3, will act as buffers. The input X0~X3 (addends)was connected, connected Y0~Y3 (augends) to DIP switches DIP2.0~2.3 and DIP1.0~1.3 respectively as shown in Figure 16 Connected F1, Σ1, Σ2, Σ3, Σ4 to L1~L5 respectively.

Followed the input sequences in Table 5 and set SW0 to “0”; recorded F1 and Σ in binary numbers.

Table 5 4-bit adder

inputs								outputs				
Y3	Y2	Y1	Y0	X3	X2	X1	X0	∑4	∑3	∑2	∑1	F1(CARRY)
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	0	0	0	1	0
0	0	0	1	1	0	0	0	1	0	0	1	0
0	0	1	1	0	1	1	0	1	0	0	1	0
1	0	0	1	1	0	0	1	0	0	1	0	1
1	0	1	0	1	0	1	1	0	1	0	1	1

C. High-Speed Adder Carry Generator Circuit

The 74LS85(U3) on block High-Speed Adder of module IT-3002 was used in this section to construct a carry carry generator circuit shown in Figure17.

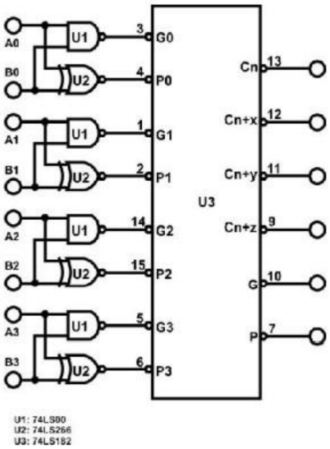


Figure 17 High-Speed Adder Carry Generator Circuit

The inputs A0~A3 (addends) was Connected to DIP Switches 1.0~1.3; B0~B3 (augends) to DIP2.0~2.3, Cn was connected to SW0, and set SW0 to “0”. Followed the input sequences in Table 6 and record output states.

Table 6 High-Speed Adder Carry Generator

Inputs								outputs				
B3	B2	B1	B0	A3	A2	A1	A0	Cn+x	Cn+y	Cn+z	G′	P′
0	0	0	1	0	0	0	1	1	0	0	1	1
0	0	1	0	0	0	1	0	0	1	0	1	1
0	0	1	0	0	0	1	1	1	1	0	1	1
1	0	1	0	1	0	0	0	0	0	0	0	1
0	1	1	1	0	1	1	0	0	1	1	1	1
1	0	0	1	0	1	0	1	1	0	0	1	1

D. Constructing BCD Adder

The circuit in Figure 18 shown below act as a BCD adder

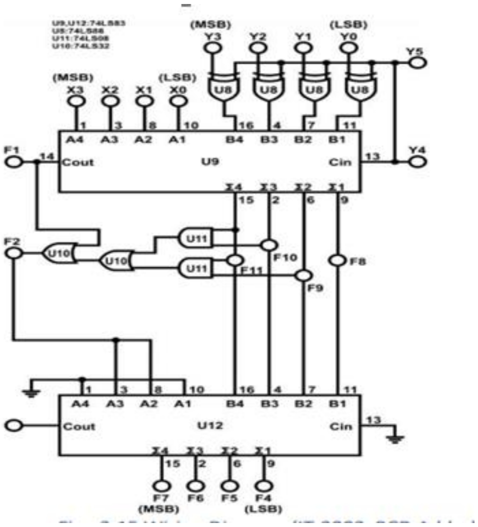


Figure 18 BCD Adder

The inputs X0~X3 was connected to DIP 1.0~1.3, Y0~Y3 to DIP 2.0~2.3; Y5 to “0”.

U9 and U12 are 74LS83 look-ahead 4-bit BCD adders.

The outputs F8~F11 was connected to the inputs of the 7-Segment display SEG-1. F1, F2 was connected to Logic Indicators L4 and L5. outputs F4~F7 of U12 was connected to another 7-Segment display SEG-3 and F3 to L10. F8~F11 was the sum of X0~X3 added to Y0~Y3 while F1 was the carry. Followed the input sequences for X0~X3 and Y0~Y3 in Table 7 and recorded the output states.

Table 7 BCD Adder

inputs								Outputs(U9)					Outputs(U12)					
X3	X2	X1	X0	Y3	Y2	Y1	Y0	F1	F11	F10	F9	F8	F2	F3	F7	F6	F5	F4
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	1	1	0	0	1	0	0	0	0	0	1	0	0

3.3 Half- and Full Subtractor Circuits.

A. Constructing Half-/Full Subtractors with basic logic Gates

The circuit in Figure 19 shown below was connected.

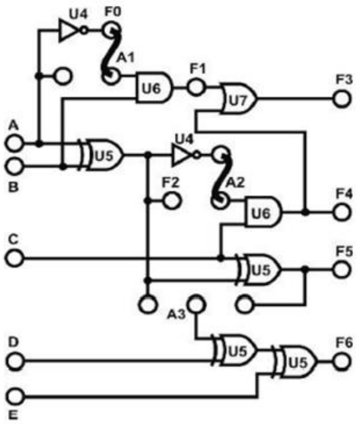


Figure 19 Full-Half subtractor

The inputs A~C was connected to Data Switches SW0~SW2.The Outputs F2 was connected to Logic Indicator L1, F1 to L2, F3 to L3, and F5 to L4. When C=0 the circuit was a half-subtractor. the borrow output was F1 , the difference was F2 and F5=F2, F4=0, F3=F1. The circuit was a full-subtractor when C=1. The borrow output was F3 and the difference output wasF5.

Followed the input sequences in Table 8 and record output states.

Table 8 Full-Half subtractor

	inputs						
	C	A	B	F1	F2	F3	F5
Half- Subtractors	0	0	1	1	1	1	1
Half-adders	0	0	0	0	0	0	0
Full- Subtractors	1	1	0	0	1	0	0
Full-adders	1	1	1	0	0	1	1

B. Constructing 4-Bit Full-Subtractor with IC

The circuit in Figure 20 shown below was connected.

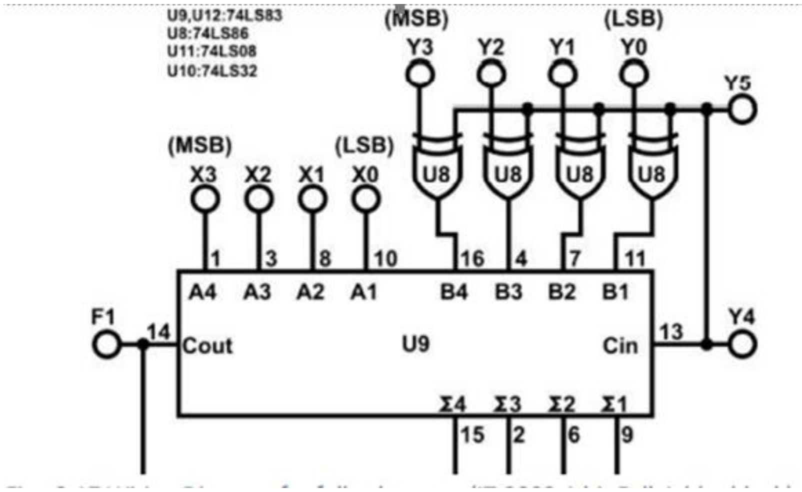


Figure 20 4-Bit Full Subtractor

The inputs X3~X0 (minuend) was connected to DIP Switch 1.3~1.0, Y3~Y0 (subtrahend) to DIP 2.3~2.0, Y5 to SW0. The outputs F1 was connected to L4, F11~F8 to L3~L0. To perform the subtraction operation, SW0 was set to "1" (or Cin of U9 = 1).

Followed the input sequences below and recorded the output states in Table 9.

Table 9 Full-Subtractor with IC

inputs								Outputs(U9)				
X3	X2	X1	X0	Y3	Y2	Y1	Y0	F1	F11	F10	F9	F8
0	1	0	0	0	1	0	0	1	0	0	0	0
0	1	0	0	0	0	1	1	1	1	0	1	0

3.4 Encoder, Decoders, Multiplexers, and Demultiplexers

A. Constructing a 4-to-2 Encoder with Basic Gates

The circuit in Figure 21 shown below was connected

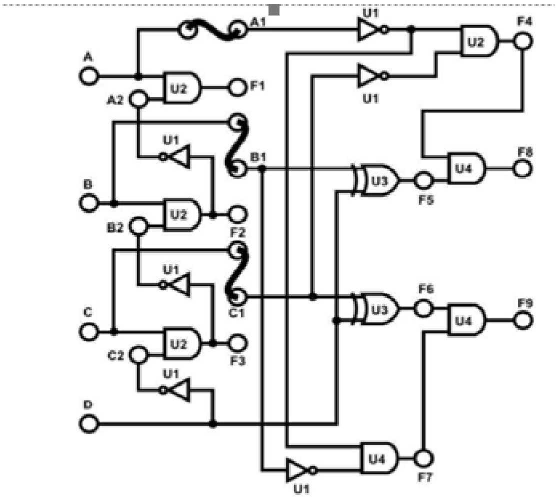


Figure 21 Encoder

The +5V of module IT-3004 was connected to the power supply section of IT3000.

The inputs A-D were connected to Switches SW0-SW3 respectively and the Outputs F8 and F9 were connected to Logic Indicators L0 and L1.

The input sequences in Table 10 for D, C, B and A were followed and the output states were recorded and listed.

Table 10 Encoder with Basic Gates

D	C	B	A	F9	F8
0	0	0	0	0	0
0	0	0	1	0	0
0	1	1	1	0	0
1	0	0	0	1	1
1	1	1	0	0	0
1	1	1	1	0	0

The circuit was active low inputs – active low outputs.

As we can see the circuit gave result only when one input was active If the inputs contain more than one the result was undefined 0.

B. Constructing 9-to-4-Line Encoder with TTL IC

The 74147 (U5) on block Encoder 2 of module IT-3004 was used in this section.

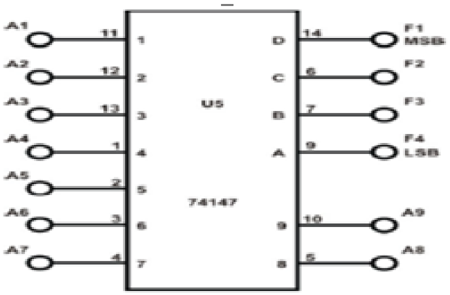


Figure 22 Encoder Block

The +5V of module IT-3004 was connected to the power supply section of IT3000. The inputs A1-A8 were connected to DIP Switches 1.0-1.7 and A9 was connected to Dip Switch 2.0. And the outputs F1-F4 were connected to Logic indicators L1-L4. The input sequences in Table 11 were followed and the output states were recorded and listed**.**

Table 11 Encoder

A9	A8	A7	A6	A5	A4	A3	A2	A1	F4	F3	F2	F1
0	1	1	1	1	1	1	1	1	0	1	1	0
0	0	1	1	1	1	1	1	1	0	1	1	0
1	1	1	1	1	1	0	1	1	0	0	1	1
1	1	1	1	1	0	0	0	0	1	1	0	1
1	1	0	0	0	1	1	1	1	0	0	0	1
1	0	0	0	0	1	1	1	1	1	1	1	0

According to the results shown in the previous table the circuit was worked as BCD Priority encoder with active low input and active low output .The priority is as follows A8,A7,A6,A5,A4,A3,A2,A1 . and A9 represent enable , when enable equal 0 the output will be don’t care condition so encoder doesn’t on . when an input with higher priority equal to 0 it will appear in the output and the other inputs with lower priority will be ignore.

C. Constructing 2-to-4 Line Decoder with Basic Gates

Block Decoder 1 of module IT-3004 was used in this section.

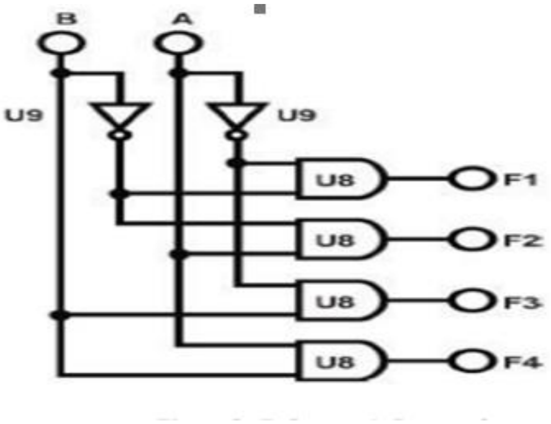


Figure 23 Decoder Block

The +5V of module IT-3004 was connected to the power supply section of IT3000.

The inputs A, B were connected to Switches SW0 and SW1 and the outputs F1-F4 were connected to Logic Indicators L0-L3.

The input sequences for A and B in Table 12 were followed and the output states were recorded and listed.

Table 12 2-to-4 Line Decoder with Basic Gates

B	A	F1	F2	F3	F4
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

- The circuit is implemented with AND gates
- The circuit was active high inputs and outputs.

- In this type of decoders, decoders have two inputs namely A, B, and four outputs denoted by F1-F4. As you can see from the above truth table – for every input combination, one line is turned on. F0 =B A, (minterm m0) which corresponds to input 00 , F1=B A, (minterm m1) which corresponds to input 01, F2=B A, (minterm m2) which corresponds to input 10 and F3 =B A, (minterm m3) which corresponds to input 11

D. Constructing 4-to-10 Line Decoder with TTL IC

U6 (7442) on block Decoder 2 of module IT-3004 was used

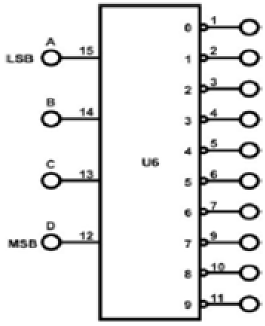


Figure 24 4 to 10 line Decoder

The Inputs A, B, C and D were connected to the Data Switches SW0-SW3 respectively and The Outputs were connected to indicator L0-L9 respectively.

The input sequences for D, C, B and A in Table 13 were followed and the output states were recorded and listed.

Table 13 4-to-10 Line Decoder with TTL IC

	Input				Output									
	D	C	B	A	0	1	2	3	4	5	6	7	8	9
0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
1	0	0	0	1	1	0	1	1	1	1	1	1	1	1
4	0	1	0	0	1	1	1	1	0	1	1	1	1	1
5	0	1	0	1	1	1	1	1	1	0	1	1	1	1
8	1	0	0	0	1	1	1	1	1	1	1	1	0	1
9	1	0	0	1	1	1	1	1	1	1	1	1	1	0

- This decoder is BCD active high input active low output that means the 0 is the main output, it has 4 inputs so there are 16 different input but there only 10 of them are valid and the output of them is 0. And If the 6 other cases were tried all of the outputs will be 1

E. Constructing 2-to-1-Line Multiplexer with basic Gates

Block Multiplexer 1 of module IT-3005 was used as 2-1 mux as shown in Figure 25

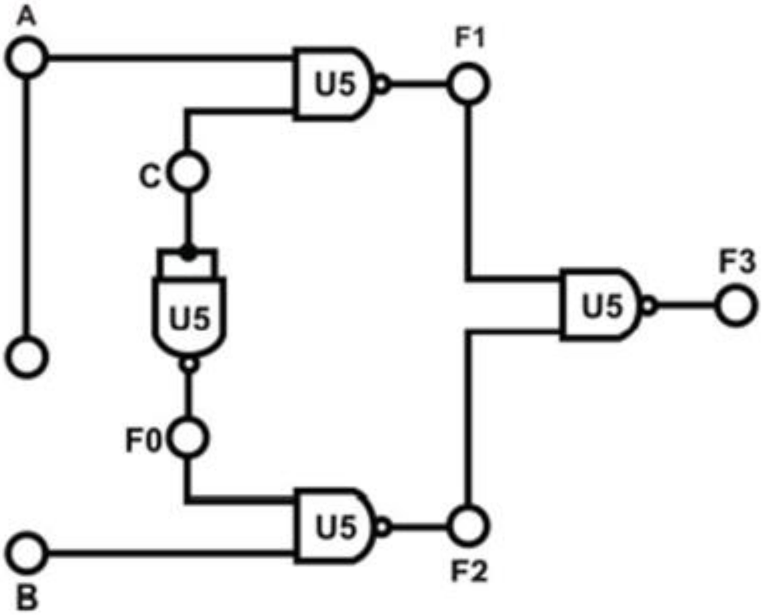


Figure 25 Multiplexer

The inputs A, B were Connected to Data Switches SW0, SW1 and selector C to SW2. The output F3 was Connected to Logic Indicator L0.

The input sequences in Table 14 were Followed and the states of F3 were recorded and listed.

Table 14 2-to-1-Line Multiplexer with basic Gates

C	A	B	F3
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

- when c equals 0 the output will be B, and when c equals 1 the output will be which is the definition of how a 2-1 multiplexer works.

F. Constructing 8-to-1 Line Multiplexer with IC

U3 (74LS151) on block Multiplexer 2 of module IT-3005 was used in this section.

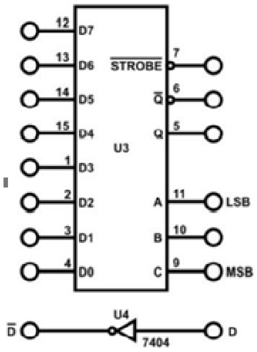


Figure 26 8-to-1 MUX

The Inputs from D0-D7 were connected to DIP Switch 1.0-1.7, the Inputs C, B, A were connected to Switches SW2, SW1 and SW0 and the Output Q was connected to indicator L0.

The input sequences in Table 15 were Followed and the states of F3 were recorded and listed.

Table 15 8-to-1 Line Multiplexer with IC

C	A	B	F3
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

- The IC will function properly only when STROBE = 0. When STROBE = 1 IC do not change output according to inputs, Q will remain 1. When CBA = 000, data at D0 will be send to output Q. When CBA = 010, data at D2 will be send to output Q. When CBA = 111, data at D7 will be send to output Q.

G. Using Multiplexer to implement a Logic Function

In this part, the 8-to-1 MUX was used to create the function:

F (D, C, B, A) = Σ(0,2,4,5,7,8,10,11,15)

Table 16 Multiplexer to implement a Logic Function

A	B	C	D	Y	implementation
0	0	0	0	1	D′
0	0	0	1	0	
0	0	1	0	0	D′
0	0	1	1	1	
0	1	0	0	1	1
0	1	0	1	1	
0	1	1	0	0	D
0	1	1	1	1	
1	0	0	0	1	D′
1	0	0	1	0	
1	0	1	0	1	1
1	0	1	1	1	
1	1	0	0	0	0
1	1	0	1	0	
1	1	1	0	0	D
1	1	1	1	1	

After testing various inputs, it was noticed that they matched the outputs in Table 16

SO , we implement the function from (16*1) MUX using (8*1) MUX

H. Constructing 1-to-2 Line Demultiplexer with Basic Logic Gates

Block Multiplexer 1 of module IT-3005 was used as 1-to-2 Demultiplexer.

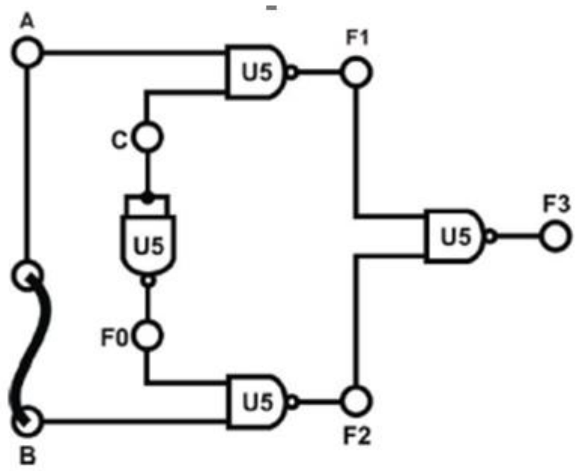


Figure 27 Demultiplexer

The input A was connected to data switch SW0, C to SW3, F1 and F2 to L0 and L1. The input sequences for C and A in Table 17 were followed and the output states were recorded and listed.

Table 17 1-to-2 Line Demultiplexer

C	A	F1	F2
0	0	1	1
0	1	1	0
1	0	1	1
1	1	0	1

- The circuit was active high inputs and outputs.
- According to the results the circuit represents a Demultiplexer function which is the inverse of multiplexer.

Conclusion

In conclusion, we understood the combinational circuit which is the digital logic circuit in which the output depends only in the input. Also, we knew the way of how Comparators, Adders, Subtractors, Encoders, Decoders, Multiplexers, and Demultiplexers work, and the importance of them in implementing Boolean functions and Karnaugh’s map. Moreover, we studied that the output of combinational circuit depends only on input. And combinational circuit can be constructed using basic logic gates such as OR, AND, X-OR and IC’s.

References

[1] https://www.geeksforgeeks.org/multiplexers-in-digital-logic/?ref=lbp. Accessed on 11-04-2022 at 1:23PM.

[2] https://www.electronics-tutorials.ws/combination/comb_5.html. Accessed on 11-04-2022 at 2:14PM.

[3] https://www.tutorialspoint.com/digital_circuits/digital_circuits_encoders.htm . Accessed on 11-04-2022 at 3:34PM.

[4] https://www.quora.com/How-do-I-design-an-8-t0-4-BCD-priority-encoder. Accessed on 11-04-2022 at 5:12PM.