Experiment 1

Combinational Logic Circuits

OBJECTIVES

1. To become familiar with AND, OR, NOT, NAND, NOR, XOR operations and their implementation.
2. To construct NOT, AND, OR and XOR gates using NAND gates.
3. To become familiar with concept of Truth table.
4. To implement different Boolean function using NAND gate only.
5. To learn techniques of solution of logic design problems.
6. To become familiar with minimization techniques and with use of Karnaugh maps.
7. To construct AOI gate with basic gates.

EQUIPMENT REQUIRED

1. KL-22001 Basic Electricity Circuit Lab
2. KL-26001 Combinational Logic Circuit Experiment Module (1)

THEORY

Combinational logic circuits are constructed with basic logic gates. Its output will correspond only to the current input, previous inputs and outputs can't influence the current output. Therefore the output of any combinational logic circuits can be expressed by Boolean functions.

The major components of a combinational logic circuit include input variables, logic gates and output variables. The input variable could be either higher or lower than the output variable but both are binary signals, or "0" and "1".

Assuming there are "n" input variables, there will be 2 possible input combinations, each with one corresponding output combination. Before designing and constructing combinational logic circuits, the following information should be taken into consideration:

1. Truth tables of logic gates
2. Boolean function
3. Karnaugh map
4. De Morgan's theorem

The following combinational logic gates are used very often and they are discussed in this chapter, along with many other combinational logic gates.

1. Combinational logic circuits with NAND and NOR gates

2. AND-OR-INVERTER (A-O-I) gate
3. XOR gate
4. Open-collector gates
5. Tristate gate
6. Arithmetic circuits
7. Encoder and decoder circuits
8. Multiplexer and demultiplexer circuits
9. Comparator circuits

1-1 NOR Gate Circuit

The symbol of a NOR gate is shown in Fig. 1-1-1. The Boolean expression for the NOR gate is F= (A + B)'; in De Morgan's theorem, F= (A + B)' = A' B'
When A=B, F = (A + B)' = A' A' =A', When B=0, F= (A + B)' = (A+0)' =A'. Therefore, the NOR gate can be used to construct NOT; OR; AND; NAND; and XOR gates. However we will not attempt to construct various logic gates in this experiment by connecting NOR gates in different ways.

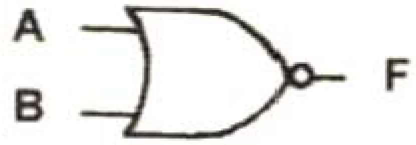
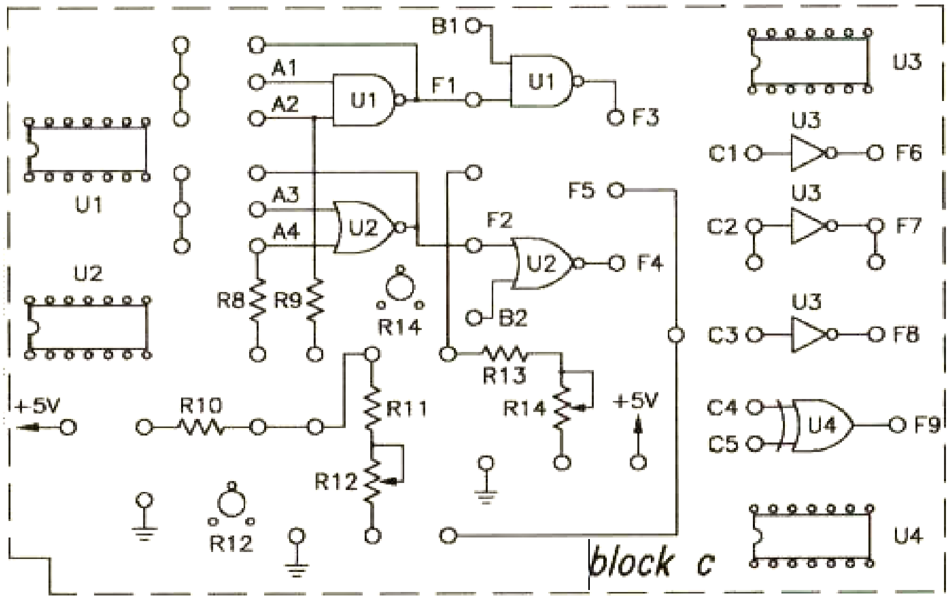


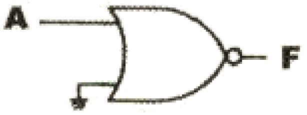
Fig. 1-1-1 Symbol of NOR gate

PROCEDURE

1. Set the KL-26001 Module on the KL-22001 Basic Electricity Circuit Lab, and locate block c. U2 of Fig. 1-1-2(a) will be used to construct a NOT gate as shown in Fig. 1-1-2(b). Apply +5VDC from the Fixed Power on KL-22001 Lab to KL-26001 Module.



(a) KL-26001 block c



(b) Equivalent to a NOT gate

Fig. 1-1-2 NOR gate used as NOT gate

Connect inputs A3, A4 to Data Switches SW0, SW1 and the output F2 to Logic Indicator L1. Set SW0 to "0" and observe states of F1 at SW1 ="0" and SW1 ="1".

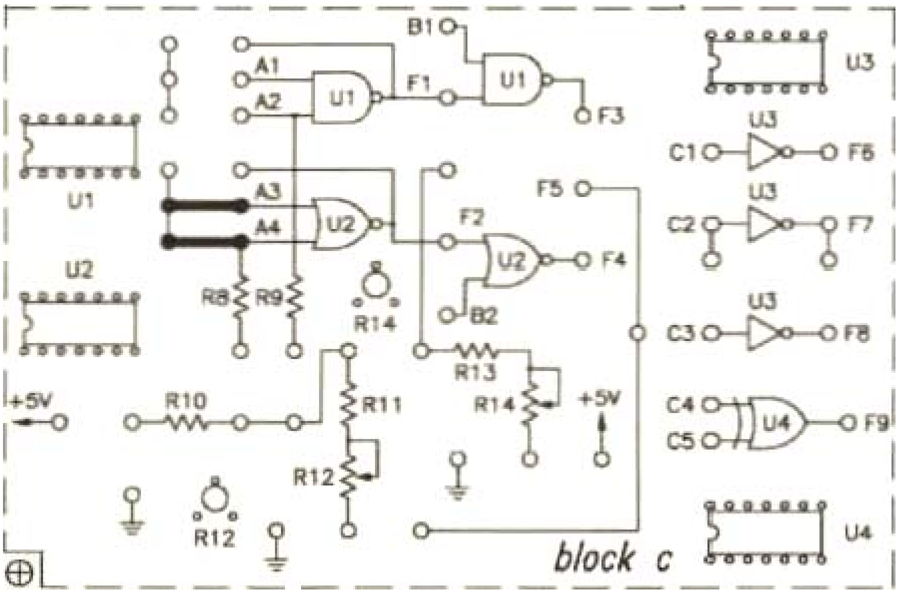
When SW1="0". F2= _______________
When SW1="1", F2= _______________

Does the circuit operate as a NOT gate?

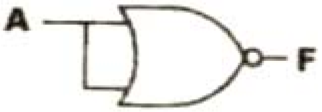
Complete the connections by referring to the wiring diagram in Fig.1-1-3(a) and the circuit in Fig. 1-1-3(b). This connects A3 and A4 together (A3=A4), Connect A3 to Data Switch SWO and the output F2 to Logic Indicator L1.

When SW0="0", F2=_____________
When SW0="1", F2=_____________

Does the circuit operate as a NOT gate?



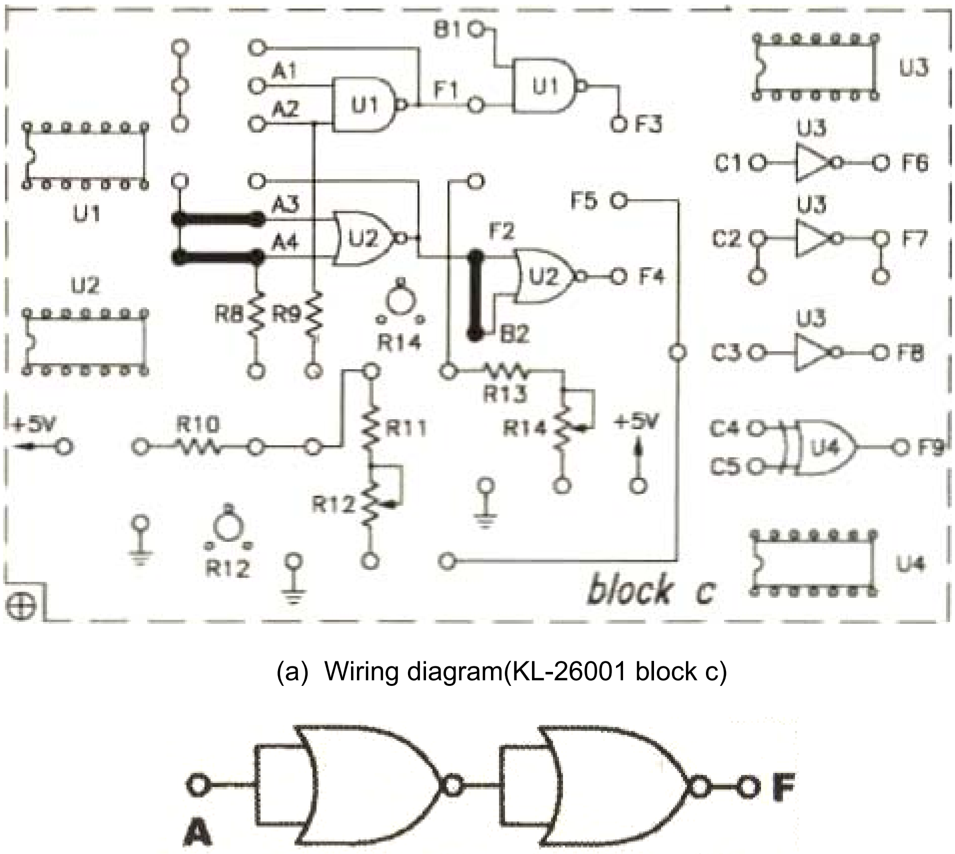
(a) Wiring diagram (KL-26001 block c)



(b) Equivalent to a NOT gate
Fig. 1-1-3 NOR gate used as NOT gate

Complete the connections by referring to the wiring diagram in Fig. 1-1-4(a) and the circuit in Fig. 1-1-4(b). Connect A3 to Data Switch SW0 and the output F4 to Logic Indicator L1.
When SW0= "0", F4 =____________
When SW0= "1", F4 =____________

Does the circuit operate as a buffer?

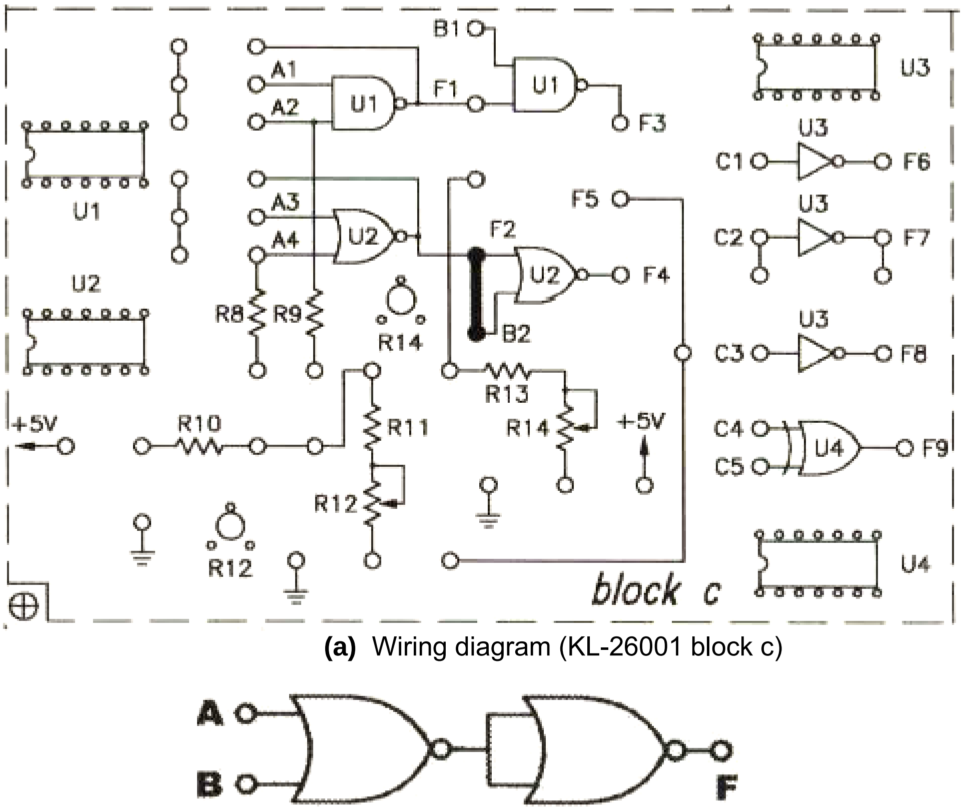


(a) Wiring diagram(KL-26001 block c)

(b) Equivalent to a buffer

Fig. 1-1-4 NOR gate used as a buffer

5. Complete the connections by referring to the wiring diagram in Fig. 1-1-5(a) and the circuit in Fig. 1-1-5(b). Connect inputs A3 to SW0, A4 to SW1; and output F4 to Logic Indicator L1.



(a) Wiring diagram (KL-26001 block c)

(b) Equivalent to an OR gate

Fig. 1-1-5 NOR gate used as an OR gate

6. Follow the input sequences and record the output states in Table 1-1-1.

SWO(A3)	SW1 (A4)	F4
0	**0**	
0	**1**	
1	**0**	
1	**1**	

Table 1-1-1

1-2 NAND Gate Circuit

The symbol of a NAND gate is shown in Fig. 1-4. The Boolean expression for a NAND gate is F= (AB)'; in De Morgan's theorem, (AB)' = A' + B'.
When A=B, F= (AB)'=A'. When B=1, F= (AB)'= (A·1)'=A'. Like the NOR, gates, NAND gates can be used to construct just about any basic logic gates. We will attempt to construct various basic gates in this experiment by connecting NAND gates in different ways.

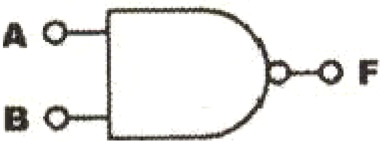
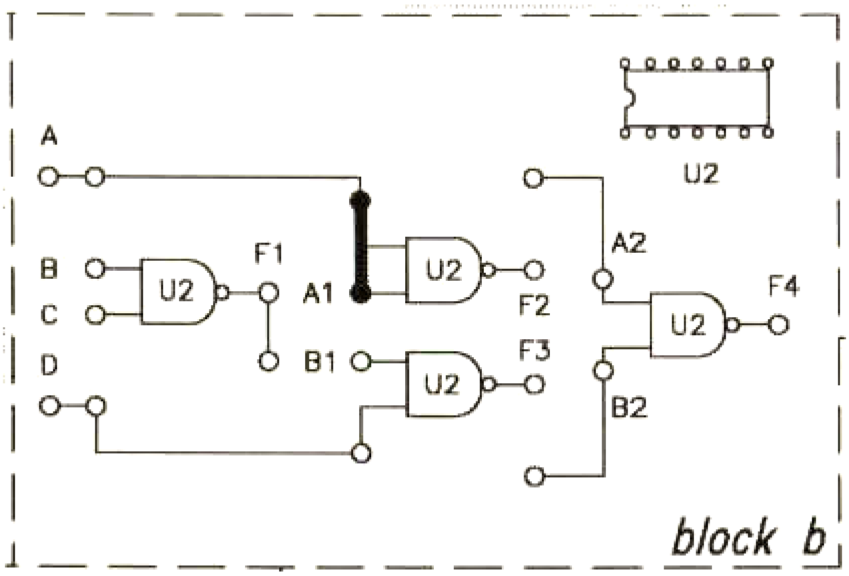


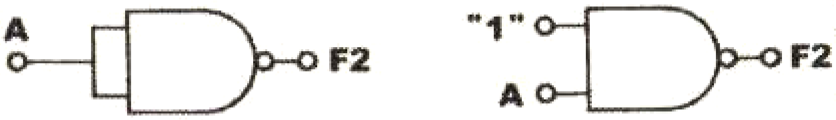
Fig. 1-2-1 Symbol of NAND gate

PROCEDURE

1. Set the KL-26001 Module on the KL-22001 Basic Electricity Circuit Lab, and locate block b. Insert the bridging plug shown in Fig. 1-2-2(a), using U2 to construct the NOT gate shown in the left-hand side of Fig. 1-2-2(b). Apply +5VDC from the Fixed Power on KL-22001 Lab to KL-26001 Module.



(a) Wiring diagram (KL-26001 block b)



(b) NOT gate constructed with NAND gate

Fig. 1-2-2 NOT gate constructed with NAND gate

2. Connect inputA to Data Switch SW1 and output F2 to Logic Indicator

   L1. Observe and record the output states.

   When SW1 ="0", F2 = ______________
   When SW1 ="1", F2 = ______________

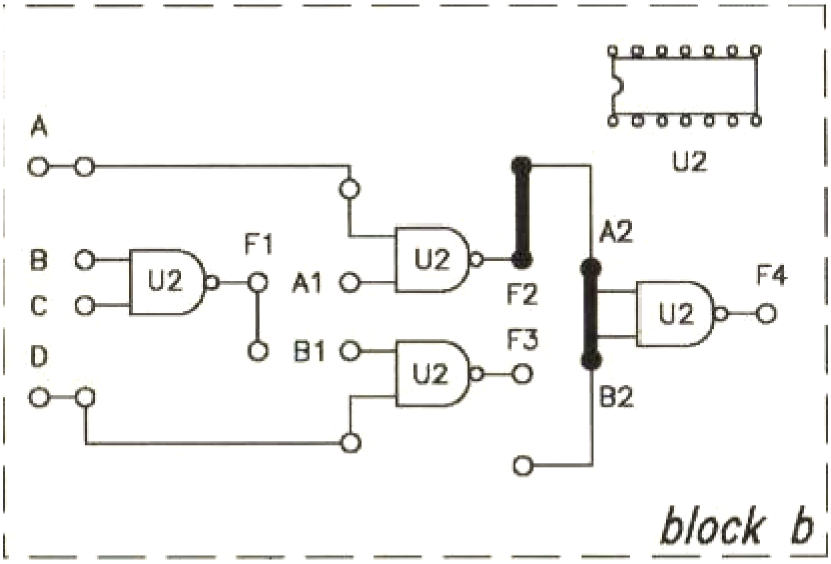
   Does the circuit act as a NOT gate?

3. Remove the bridging plug between A and A1. Connect input A1 to +5V ("1") to create the NOT gate shown in the right-hand side of Fig. 1-2-2(b). Remain other connections unchanged. Observe the output states.

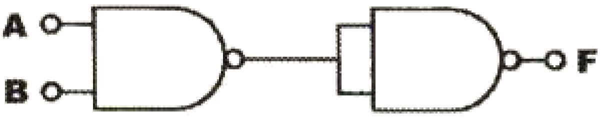
   When SW1="0", F2 = _____________
   When SW1="1", F2 = _____________

   Does the circuit act as a NOT gate?

4. Complete the connections by referring to the wiring diagram in Fig. 1-2-3(a) and the circuit in Fig. 1-2-3(b). Connect A to SW1, A1 to SW2 and F4 to L1.



(a) Wiring diagram (KL-26001 block b)



(b) Equivalent to an AND gate

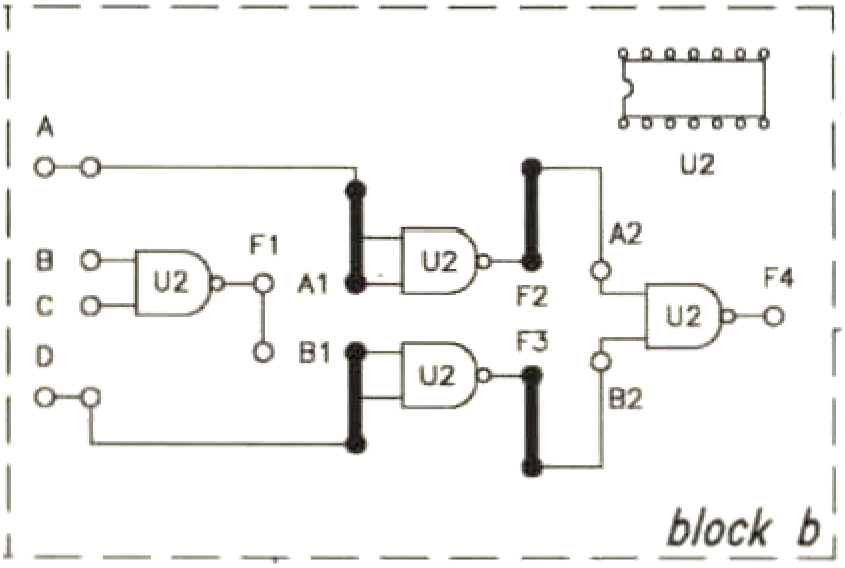
Fig. 1-2-3 AND gate constructed with NAND gates

5. Follow the input sequences and record the outputs in Table 1-2-1. Does the circuit act as an AND gate?

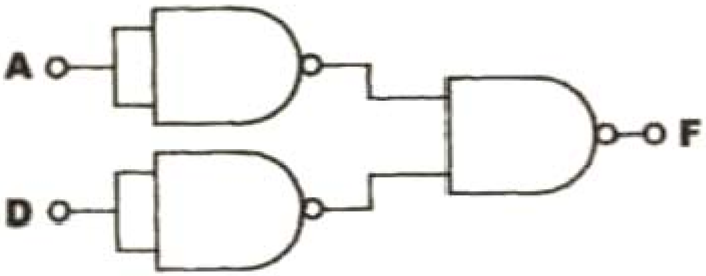
SW2(A1)	SW1(A)	F4
0	**0**	
0	**1**	
1	**0**	
1	**1**	

Table 1-2-1

6. Complete the connections by referring to the wiring diagram in Fig. 1-2-4(a) and the circuit in Fig. 1-2-4(b). Connect A to SW1; D to SW2, and F4 to L1.



(a) Wring diagram (KL-26001 block b)



(b) Equivalent to an OR gate

Fig 1-2-4 OR gate constructed with NAND gates

7. Follow the input sequences in Table 1-2-2 and record the outputs.

Does the circuit act as an OR gate (F=A+B)?

SW2(D)	SW1(A)	F4
0	**0**	
0	**1**	
1	**0**	
1	**1**	

Table 1-2-2

1-3 XOR Gate Circuit

The symbol of an XOR gate is shown in Fig. 1-3-1. The output F is equal to $A \oplus B = A'B + AB'$. XOR gates can be constructed using NOT, OR, AND, NOR or NAND gates or by using four NAND gates, as shown in Fig. 1-3-2 (a) and (b).

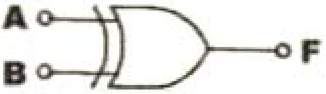
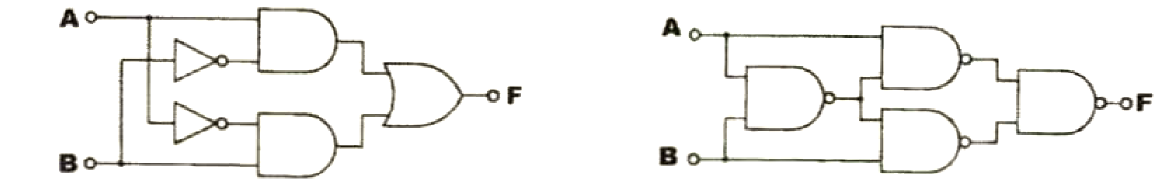


Fig. 1-3-1 Symbol of XOR gate



(a) Constructed with basic gates (b) Constructed with NAND gates

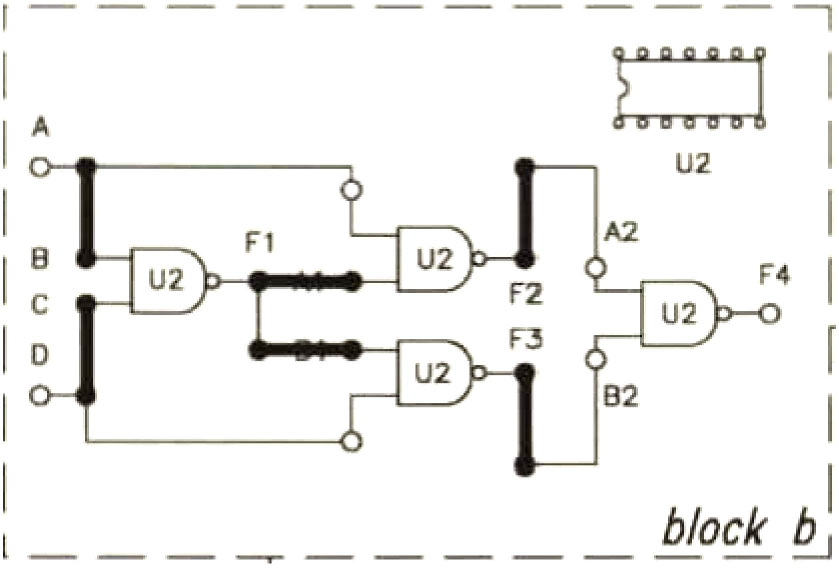
Fig. 1-3-2 XOR gate circuits

Since $F = A'B+AB'$, when B=0, $F = A' \cdot 0+A \cdot 0' = A \cdot 1 = 1$ and the circuit act as buffer. When B=1, $F = A' \cdot 1 + A \cdot 1' = A' \cdot 1 = A'$, the circuit act as an inverter. In other words, the input state of an XOR gate determines whether it will act as a buffer or an inverter. In this experiment, we will use basic logic gates to construct XOR gates and study the relationship between the inputs and outputs.

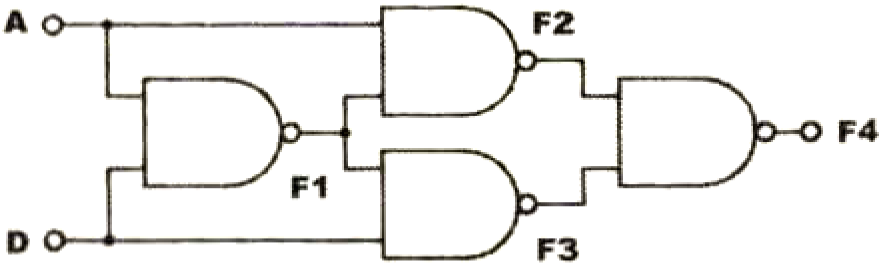
PROCEDURE

A. Constructing XOR Gate with NAND Gates

1. Set the KL-26001 Module on the KL-22001 Basic Electricity Circuit Lab, and locate block b. Complete the connections by referring to the wiring diagram in Fig. 1-3-3(a) and the circuit in Fig. 1-3-3(b). Connect inputs A to SW1, D to SW2; outputs F1 to L1, F2 to L2; F3 to L3 and F4 to L4. Apply +5VDC from the Fixed Power to KL-26001 Module.



(a) Wiring diagram (KL-26001 block b)



(b) Equivalent circuit

Fig.1-3-3 XOR gate constructed with NAND gates

2. Follow the input sequences for A and 0 in Table 1-3-1 and record the outputs.

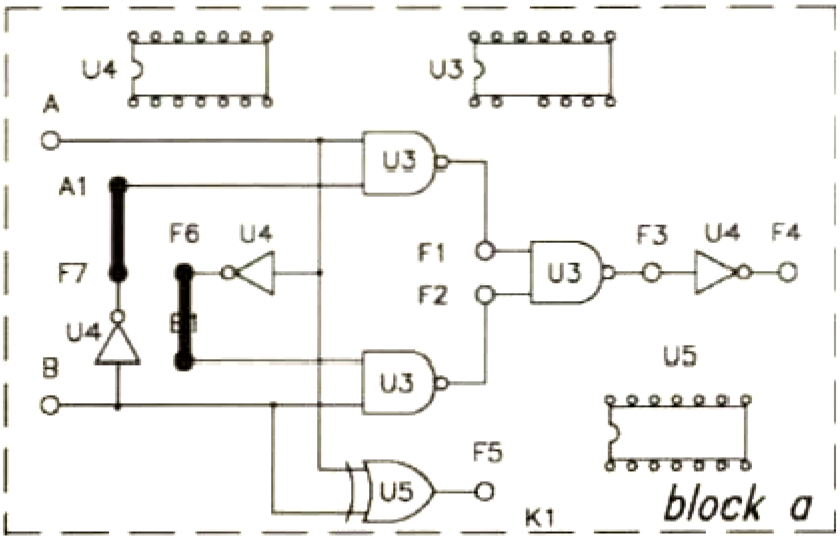
SW2(D)	SW1(A)	F1 F2 F3 F4
0	**0**	
0	**1**	
1	**0**	
1	**1**	

Table 1-3-1

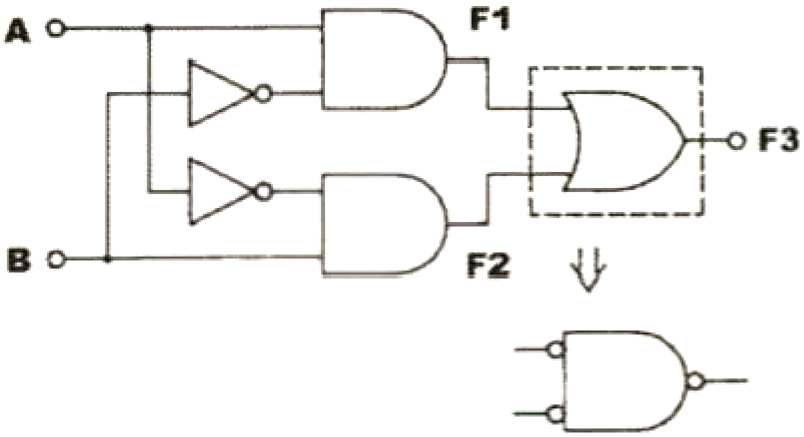
B. Constructing XOR Gate with Basic Gates

1. Set the KL-26001 Module on the KL-22001 Basic Electricity Circuit Lab, and locate block a. Complete the connections by referring to the wiring diagram in Fig. 1-3-4(a) and the circuit in Fig. 1-3-4(b). Apply +5VDC from the Fixed Power on KL-22001 Lab to KL-26001 Module.

2. Connect inputs A, B to SW1, SW2; outputs F1. F2. F3 to L1. L2. L3.



(a) Wiring diagram (KL-26001 block a)



(b) Equivalent circuit

Fig.1-3-4 XOR gale constructed with basic gates

3. Follow the input sequences for A and B in Table 1-3~2 and record the outputs.

SW2(B)	SW1 (A)	F1 F2 F3
0	**0**	
0	**1**	
1	**0**	
1	**1**	

Table 1-3-2

1-4 AOI Gate Circuits

AND-OR-INVERTER (AOI) gates consist of two AND gates, one OR gate and one INVERTER (NOT) gate. The symbol of an AOI gate is shown in Fig. 1-4-1. The Boolean expression for the output F is:

F = (AB+CD)' (1)

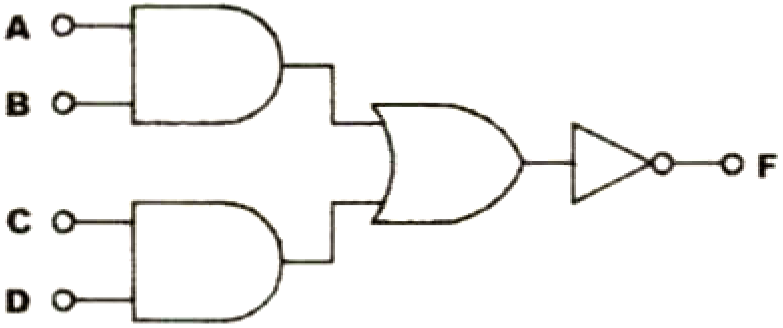


Fig. 1-4-1 AOI gate

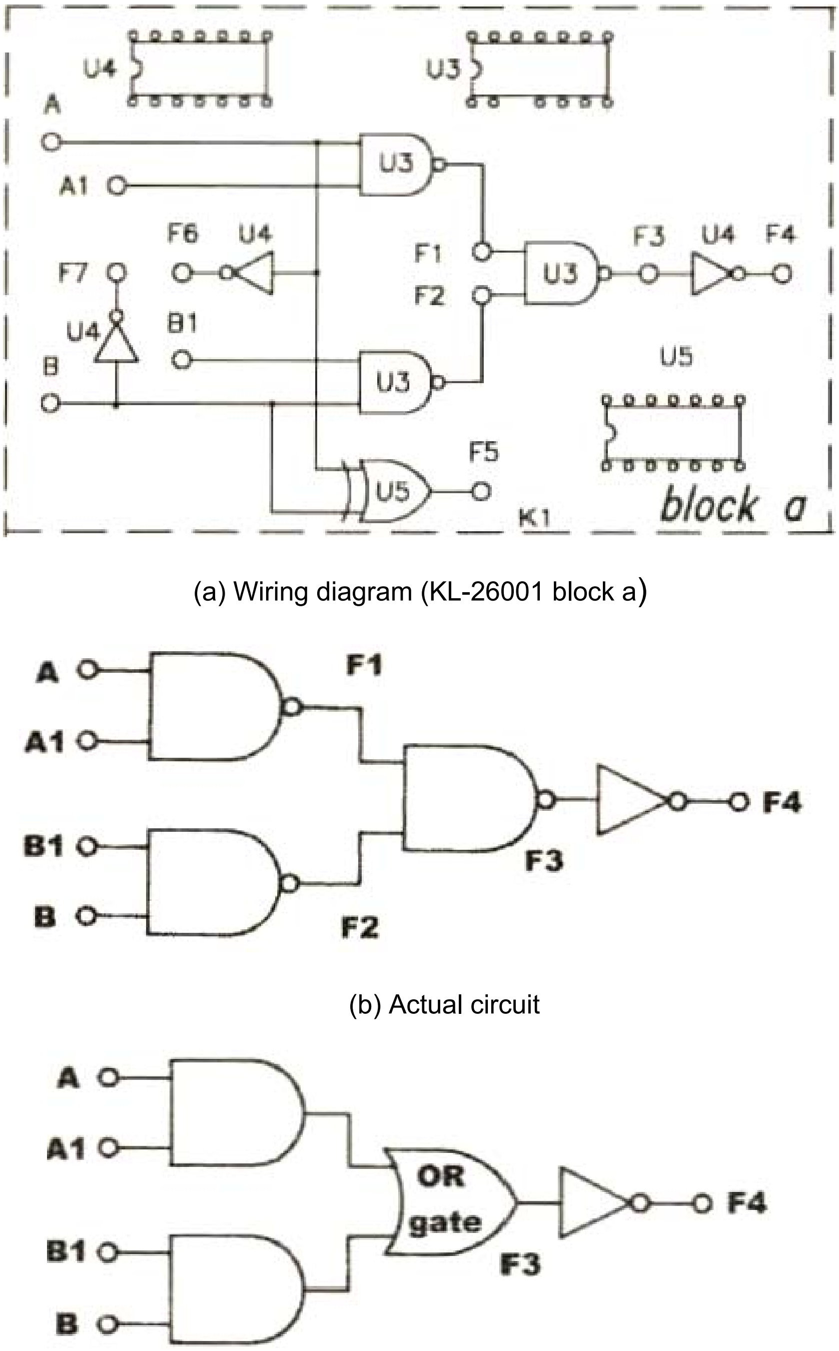
By De Morgan's theorem, Eq. (1) can be converted to:
F= (A'+B') (C'+D') (2)

Eq. (1) is also referred to as "Sum of Products".

Eq. (2) is also referred to as "Product of Sums".
Basically, the A-Q-I gate is a "Sum of Products" logic combination.

PROCEDURE

1. Set the KL-26001 Module on the KL-22001 Basic Electricity Circuit Lab, and locate block a. The circuits in Fig. 1-4-2 include the actual AOI circuit and the equivalent circuit.



(a) Wiring diagram (KL-26001 block a)

(b) Actual circuit

(c) Equivalent circuit

Fig. 1-4-2 AOI circuit

2. Connect inputs A, A1, B, B1 to Data Switches SW0, SW1, SW2, SW3 respectively. Connect outputs F3, F4 to Logic Indicators L1, and L2. Apply +5VDC from the Fixed Power on KL-22001 Lab to KL-26001 Module.

3. Set B.B1 to "0", follow the input sequences for A and A1 in Table 1-4-1 and record the outputs.

B.B1=0

A1	A	F3 F4
0	**0**	
0	**1**	
1	**0**	
1	**1**	

Table 1-4-1

Does F3 act as an AND function (F3=AoA1)?

4. Does F3 act as an AND function (F3=A.A1) when B.B1 ≠0?

5. When A-A1=0, follow the input sequences for Band B1 in Table 1-4-2 and record the outputs.

A1-A=0

B1	B	F3 F4
0	**0**	
0	**1**	
1	**0**	
1	**1**	

Table 1-4-2

Does F3 act as an AND function (F3=B.B1)?

6. Does F3 act as an AND function (F3=B.B1) when A.A1≠0?

7. Does F3 implement the function F3 = A.A1+B.B1?

1·5 Logic problem

1. A given logic system is shown in Fig. 1-5-1; output of the system F is in”1”state in each of the following condition
   C and D are in the “1” state.
   A, B and D are in the “1” state, C is in "0" state.
   B and D are in the "1" state A and C are in "0" state.
   C and B are in the “1” state A and D are in “0” state.
   C is in the “1” state A, B and D are in the “0” state. Complete the truth table of the above logic system.

Fig. 1-5-1 Block diagram

2. Write the Boolean equation as canonical sum.
3. Minimize the expression that you obtained using laws of Boolean algebra. Write dawn reduced equation.
4. Draw the karnaugh map of logic problem and indicate columns and rows circle the sub-cubes and write down the equation that you obtained is the equation identical to the one that you obtained in paragraph C.
5. Implements the reduced expression that you obtained in the previous paragraph using NAND gates only.

In your report solve these problems:

1. Draw the logic diagram showing the implementation of the following Boolean equation using "AND" gates F = AB (CA).
2. Draw the logic diagram of the following Boolean equations
   a. F2= (A+B) (CD+A)
   b. F3= (ABC+D) C
3. Implement the OR operation using AND, NOT gate.
4. Implement the AND gate using OR, NOT gate. Draw the logic diagram used in both cases and write Boolean equation.
5. Prove that the equality operation Fl =AB+A'B' is the inverse of exclusive OR operation F2=AB'+A'B (use Demorgarn's theorem).
6. Suggest a logic diagram which will give a NAND gate with four inputs using two input NAND gates, Implement suggested network.
7. Show how is it possible to reduce Boolean expressions by means of karnaugh map
   F1 = A'BCD + ABCD' + A'BCD' + ABCD'
   F2 = A'B'C'D' + AB'CD' + A'B'CD' + A'BC'D'

Implement the minimal expressions using NAND gates.