STUDY OF UNIVERSAL GATES

EXPT. NO :
DATE :

AIM: To study Universal gates.

APPARATUS REQUIRED:

Sr. No.	COMPONENT	SPECIFICATION	QTY
1	NAND GATE 2 I/P	IC 7400	1
2	NOR GATE	IC 7402	1
3	IC TRAINER KIT	-	1
4	PATCH CORD	-	14

THEORY:

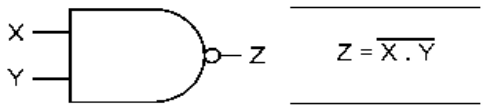
NAND Gate:

The NAND gate represents the complement of the AND operation. Its name is an abbreviation of NOT AND. The graphic symbol for the NAND gate consists of an AND symbol with a bubble on the output, denoting that a complement operation is performed on the output of the AND gate

The truth table and the graphic symbol of the NAND gate is shown in the figure.

The truth table clearly shows that the NAND operation is the complement of the AND.

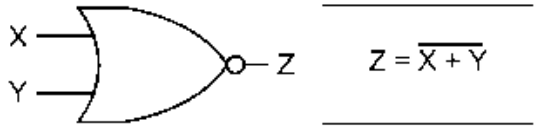
X	Y	NAND
0	0	1
0	1	1
1	0	1
1	1	0



NOR Gate:

The NOR gate represents the complement of the OR operation. Its name is an abbreviation of NOT OR. The graphic symbol for the NOR gate consists of an OR symbol with a bubble on the output, denoting that a complement operation is performed on the output of the OR gate. The truth table and the graphic symbol of NOR gate is shown in the figure.

X	Y	NOR
0	0	1
0	1	0
1	0	0
1	1	0



The truth table clearly shows that the NOR operation is the complement of the OR.

Universal Gates:

A universal gate is a gate which can implement any Boolean function without the need to use any other gate type. The NAND and NOR gates are universal gates.

In practice, this is advantageous since NAND and NOR gates are economical and easier to fabricate and are the basic gates used in all IC digital logic families. In fact, an AND gate is typically implemented as a NAND gate followed by an inverter, not the other way around. Likewise, an OR gate is typically implemented as a NOR gate followed by an inverter, not the other way around.

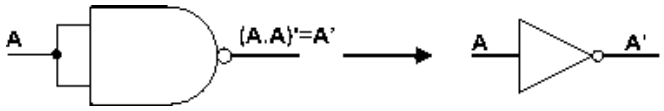
NAND Gate is a Universal Gate:

To prove that any Boolean function can be implemented using only NAND gates, we will show that the AND, OR, and NOT operations can be performed using only these gates.

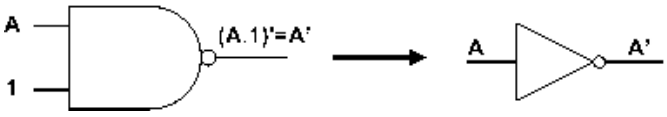
Implementing an Inverter Using only NAND Gate

The figure shows two ways in which a NAND gate can be used as an inverter (NOT gate).

1. All NAND input pins connect to the input signal A gives an output A'

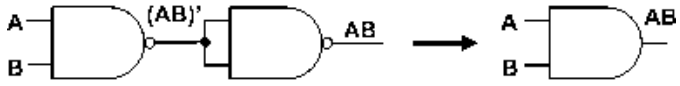


2. One NAND input pin is connected to the input signal A while all other input pins are connected to logic 1. The output will be A'.



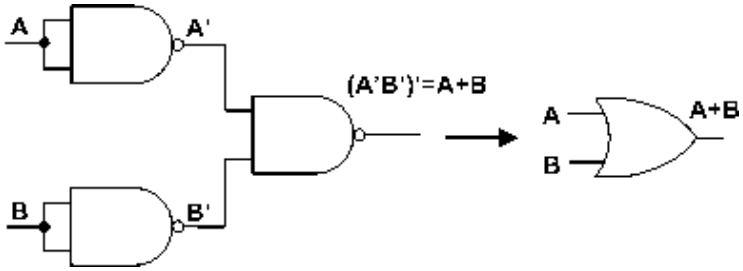
Implementing AND Using only NAND Gates

An AND gate can be replaced by NAND gates as shown in the figure (The AND is replaced by a NAND gate with its output complemented by a NAND gate inverter).



Implementing OR Using only NAND Gates

An OR gate can be replaced by NAND gates as shown in the figure (The OR gate is replaced by a NAND gate with all its inputs complemented by NAND gate inverters).



Thus, the NAND gate is a universal gate since it can implement the AND, OR and NOT functions.

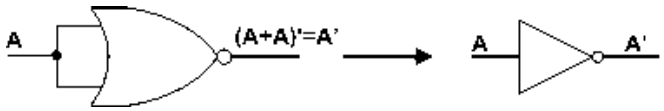
NOR Gate is a Universal Gate:

To prove that any Boolean function can be implemented using only NOR gates, we will show that the AND, OR, and NOT operations can be performed using only these gates.

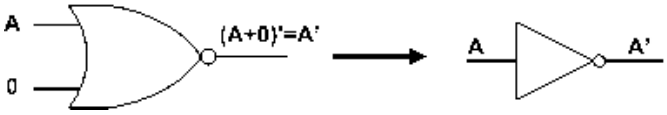
Implementing an Inverter Using only NOR Gate

The figure shows two ways in which a NOR gate can be used as an inverter (NOT gate).

1. All NOR input pins connected to the input signal A gives an output A'.

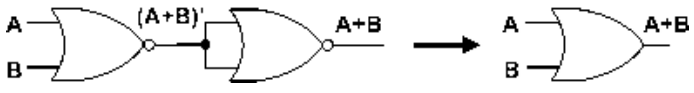


2. One NOR input pin is connected to the input signal A while all other input pins are Connected to logic 0. The output will be A'.



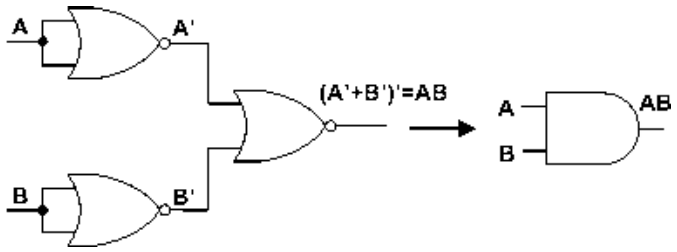
Implementing OR Using only NOR Gates

An OR gate can be replaced by NOR gates as shown in the figure (The OR is replaced by a NOR gate with its output complemented by a NOR gate inverter)



Implementing AND Using only NOR Gates

An AND gate can be replaced by NOR gates as shown in the figure (The AND gate is replaced by a NOR gate with all its inputs complemented by NOR gate inverters)



Thus, the NOR gate is a universal gate since it can implement the AND, OR and NOT functions.

PROCEDURE:

(i) Connections were given as per circuit diagram.

(ii) Logical inputs were given as per truth table

(iii) Observe the logical output and verify with the truth tables.

RESULT: