DIGITAL CIRCUIT LAB ONE.

Group 7 Members.

1. COMMODORE DERRICK – 10904196
2. ANTWI BOASIAKO MAAME YAA HENEWAA – 10889628
3. NYAME KOFI – 10895681
4. BAAH NANA KWADWO ASARE - 10907896

INTRODUCTION

The logic gate is the fundamental component of a digital circuit. The vast majority of system gates have two inputs and one output. At any given moment, the input and output are in one of two states: down (0) or high (1), which are commonly referred to as “off” or “on” and are represented by different voltage levels. Most logic gates have a down input of roughly zero volts (0 V) and a high input of about five volts (+5 V).

AIM OF EXPERIMENT: The core aim of this lab experiment is to manipulate a NAND gate to form the following gates: NOT gate, AND gate, NOR gate, OR gate, and EXOR gate.

METHODOLOGY AND RESULTS

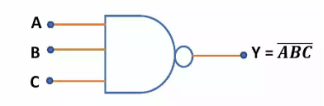
Using a couple of wires and a NAND gate IC, we connected three or two wires from the input A and B or A, B, and C of the NAND to the logic ‘0’ and logic ‘1’ using different combinations of inputs A, B, and C. We then connected the output to an LED. If the LED stays on, we record it as

a high and when it goes off, as a low. After the connection, we began to test for each of the gates mentioned above.

A. NAND

The NAND gate has its output as the negative product of its inputs (A.B.C). Using a couple of wires, we were able to connect the three inputs of the NAND gate to a logical '0' and logical '1' in turn and connected the output to an LED.

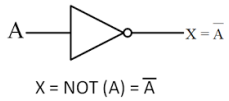
After the test, we were able to provide the truth table below.



A	B	C	$A = \overline{A.B.C}$
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

B. NOT

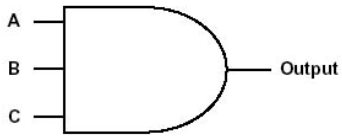
The NOT gate has one input and one output. To derive a NOT gate from a NAND gate, we connected all the three inputs of the NAND gate and connected the common point to logical '0' and logical '1' in turn. We then connected the output to an LED and observed. After the test, we were able to provide the truth table below.



A	$F = \overline{A}$
1	0
0	1

C. AND

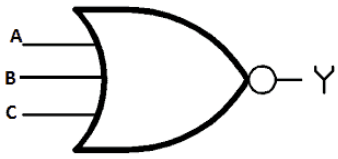
The AND gate has its output as the product of its inputs (A.B.C). The AND gate is the negation or opposite of the NAND gate in a simpler term. To produce the AND gate, we connected the output of one of the NAND gates to the NOT gate we had already created. We then tabulated the output against the different combinations of the input A, B, and C. We were able to produce the truth table below.



A	B	C	$F = A.B.C$
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

D. NOR

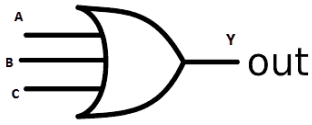
The NOR gate has its outputs as the negative sum of its inputs (A+B+C). To produce a NOR gate from a NAND gate, we connected three NOT gates to a NAND gate and then connected it to another NOT gate. We then tabulated the output against the different combinations of the input A, B, and C. We were able to produce the truth table below.



A	B	C	$F = \overline{A + B + C}$
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

E. OR

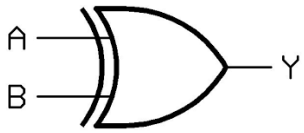
The OR gate has its outputs as the sum of its inputs (A+B+C). To produce an OR gate from a NAND gate, we connected three NOT gates to a NAND gate. We then tabulated the output against the different combinations of the input A, B, and C. We were able to produce the truth table below.



A	B	C	$F = A + B + C$
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

F. EXOR.

The EXOR gate accepts two inputs only. The truth table is provided below.



A	B	$F = \overline{A}B + A\overline{B}$
0	0	0
0	1	1
1	0	1
1	1	0

PROOF THAT GIVEN FUNCTION IS AN EXOR FUNCTION

From diagram 6B,

$$\Longrightarrow F = \overline{\overline{A}B + A\overline{B}}$$

By De-Morgan's law (Applying it to two times)

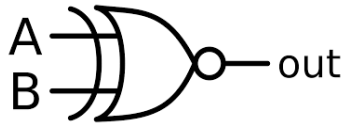
$$\Longrightarrow F = \overline{\overline{\overline{A}B}.\overline{A\overline{B}}}$$

$$\Rightarrow F = \overline{A}B + A\overline{B}$$

Hence, this system is an EXOR gate since the derived equation is the general EXOR function.

G. EXNOR.

When the EXOR gate is connected to an inverter, an EXNOR gate is formed. The truth table of an ENXOR gate is proved below.



A	B	$F = \overline{\overline{A}B + A\overline{B}}$
0	0	1
0	1	0
1	0	0
1	1	1

DISCUSSION

The results of the experiment duly support the expected findings stated in theory or hypothesis. In Lab A, the various combinations of inputs A, B, and C were fed into a NAND gate. This is equivalent to feeding various combinations of A, B, and C into an AND gate and feeding the output of that AND gate into a NOT gate. The results of the experiment match the theoretical expectations of a NAND gate.

Also, in Lab B, the three inputs of the NAND gate were connected, and then the common point connected to logic 0 and 1 in turn. This makes the original NAND gate behave as an OR gate. The reason is that when all the inputs of an AND gate are the same (i.e. 0 or 1), the same output is produced. And a NAND gate produces the opposite result of the AND gate. Hence the result for Lab A above (that of a NOT gate).

In Lab C, the output of one of the NAND gates in Lab A was connected to the NOT gate produced in Lab B, the produced the result of an AND gate as shown above in the methodology and results section of the report.

Similarly, in Lab E, different combinations of inputs A, B, and C are fed into three of the NOT gates produced in Lab B and turn, fed into a NAND gate. This produced an OR gate and hence the results of lab E above.

Moreover, in Lab D, the outputs of the OR gate produced in Lab E were fed in turns into one of the NOT gates produced in Lab B, this produced a NOR gate. The table for Lab D shows the outputs of a NOR gate for various combinations of inputs A, B, and C. These results match the theoretical outputs of a NOR gate as expected.

Furthermore, in Lab F, the inputs A and B were each fed into a NOT gate. The original input A and the derived NOT output of A were both fed into a NAND gate system. The same was done for B and it's derived NOT output. Each of the newly created systems for A and B and their derived NOT outputs was in turn connected to a two-input NAND gate. This produced an EXOR gate and hence the tabulated results in F.

Last but not least, for Lab G, the circuit system for Lab F was connected to an inverter. This produced an EXNOR gate. The tabulated results of the outputs for Lab G shows that the created circuit system is that of an EXOR gate. This is a true representation of the theoretical true values obtained from feeding various combinations of inputs A and B into an EXNOR system.

CONCLUSION

In conclusion, the NAND logic gate is a universal logic gate which means it can implement any Boolean function without the use of any other logic gate type.