



THE UNIVERSITY OF ZAMBIA
SCHOOL OF ENGINEERING
DEPARTMENT OF ELECTRICAL AND ELECTRONICS
ENGINEERING

LAB 4 REPORT

NAME: NTALASHA MWANSA

COMPUTER NUMBER: 2019109646

COURSE: EEE 3131

LECTURER: MR GEORGE ZIBA

1.0 TITLE:

EQUIVALENCE, NON-EQUIVALENCE AND OTHER CIRCUITS

2.0 OBJECTIVE

- To introduce a group of important circuits based upon two inputs to perform equivalence, non-equivalence, equality, inequality and half-adder functions.

3.0 EQUIPMENT REQUIRED

- Multisim software
- Hp ENVY laptop

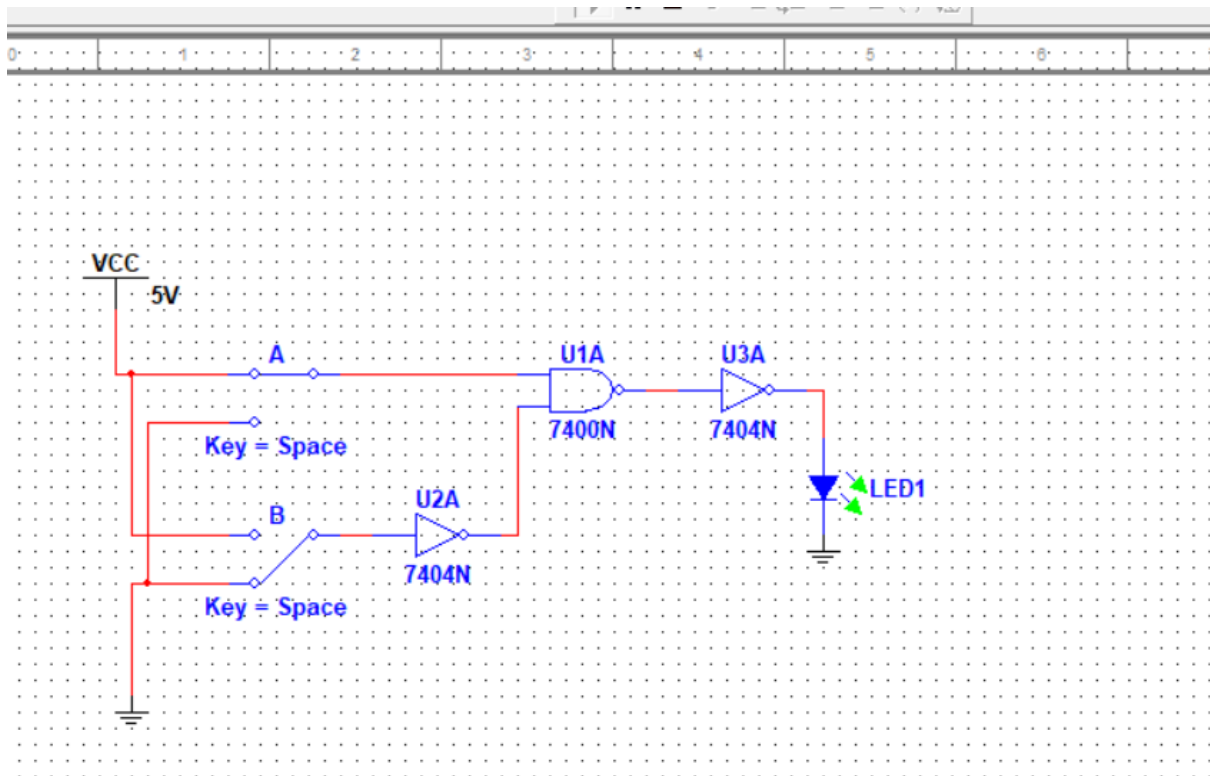
4.0 PROCEDURE

For Practicals 7.1 to 7.6:

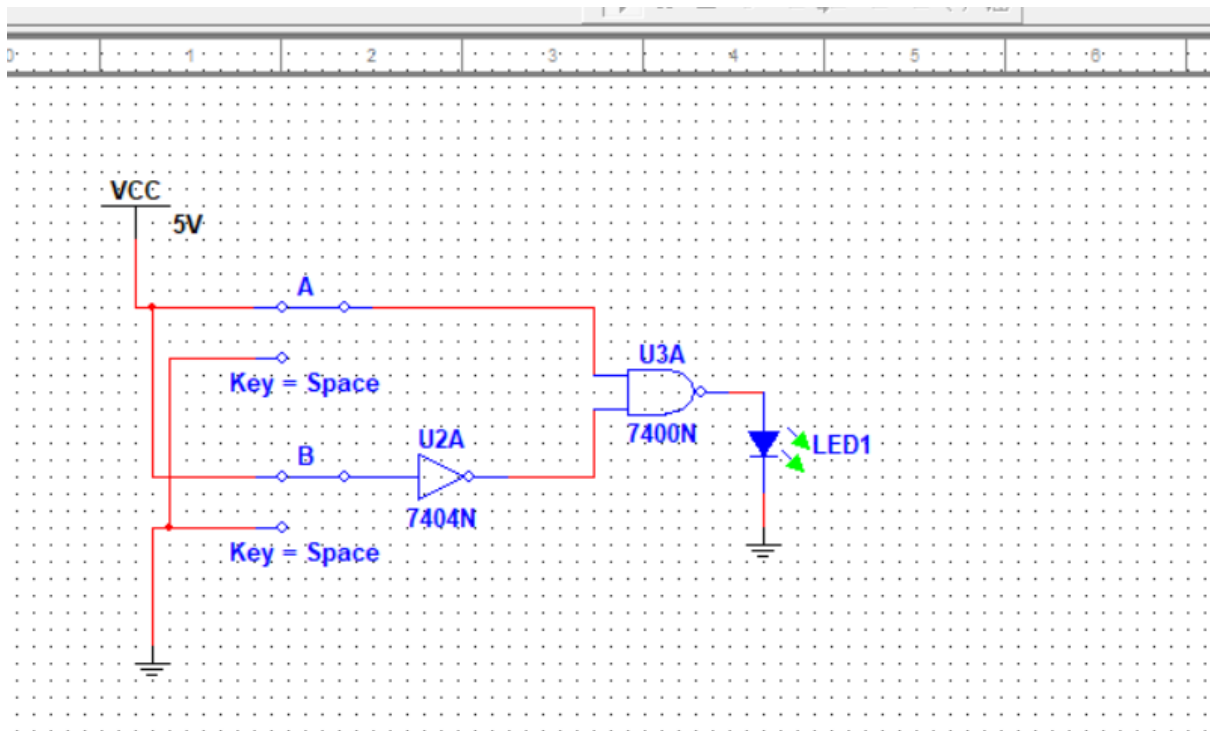
Set up the circuit as instructed in the lab manual and simulate on multism software

5.0 RESULTS

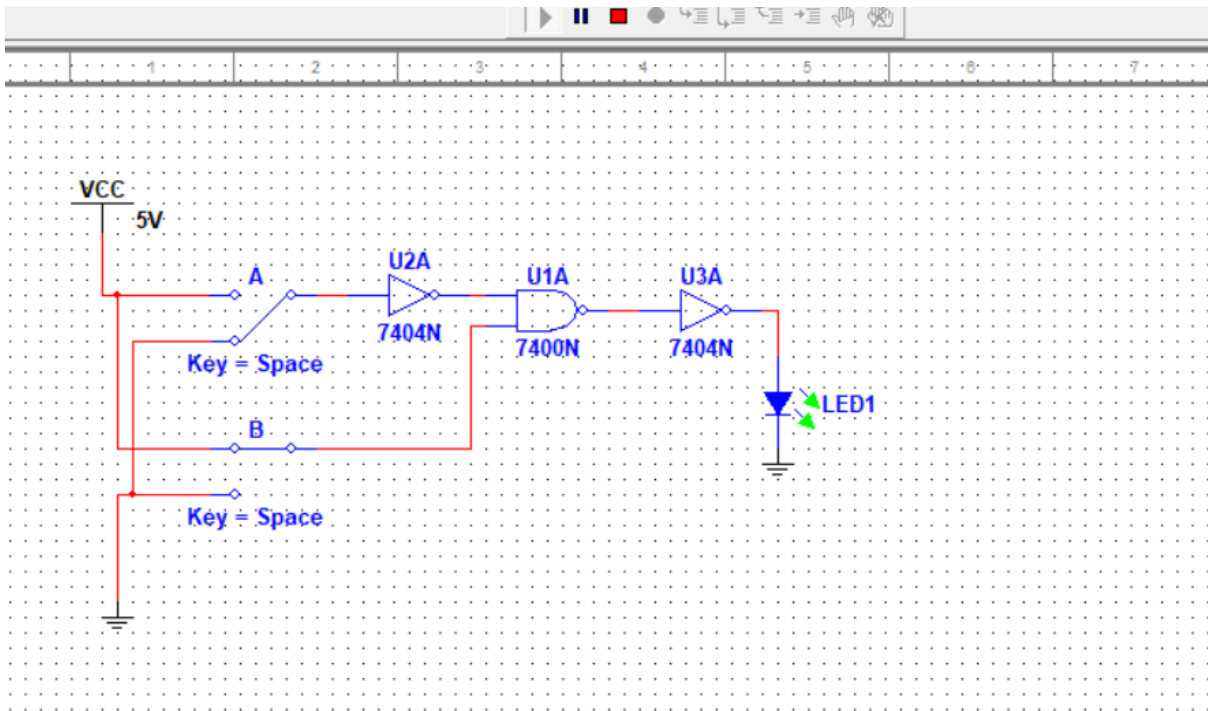
Practical 7.11 A greater than B



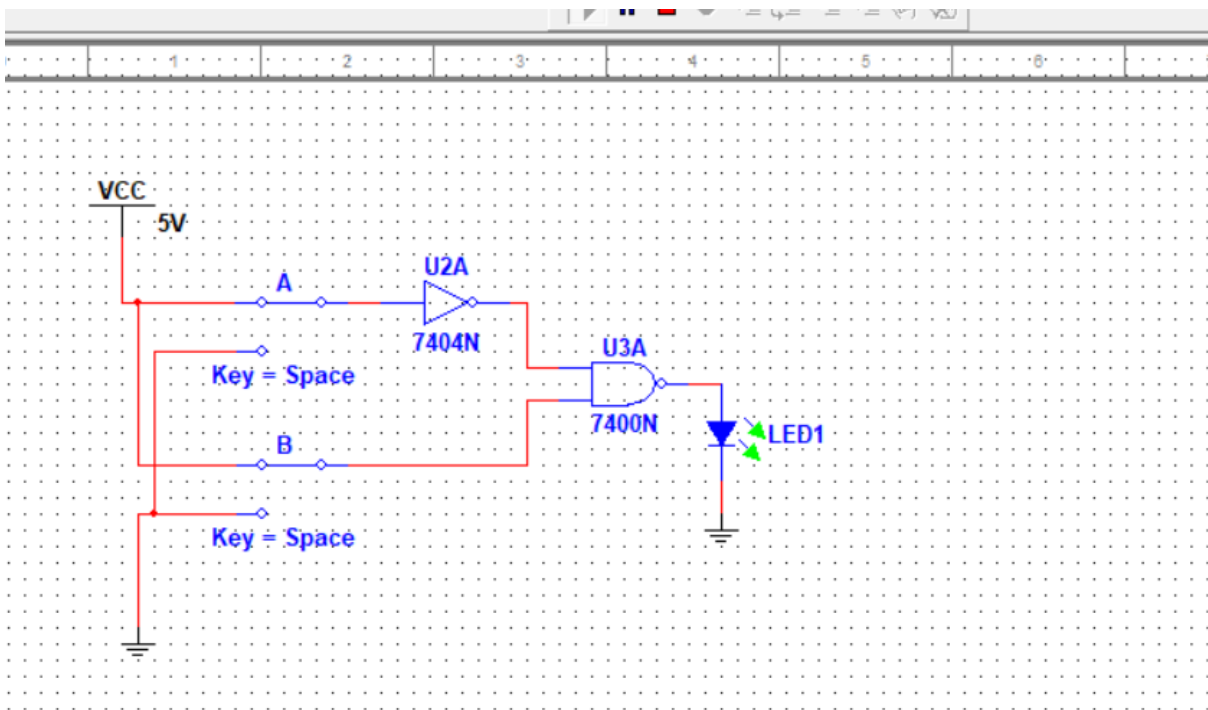
7.12 A less than or equal to B



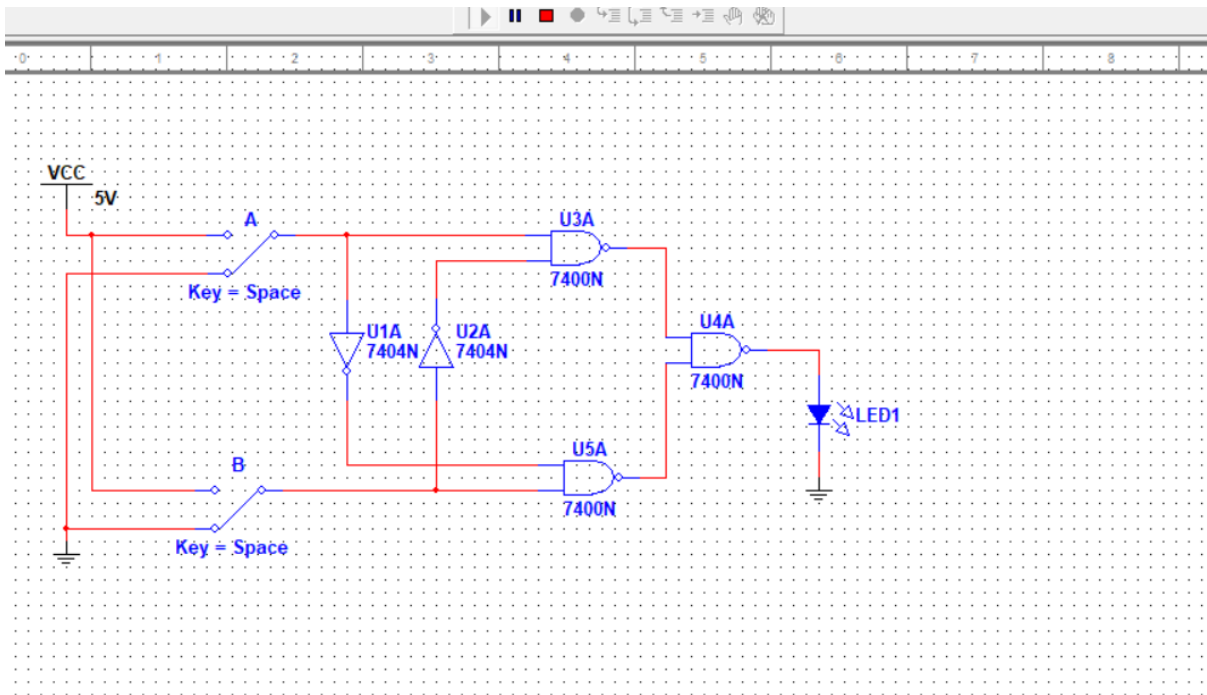
7.13 A less than B



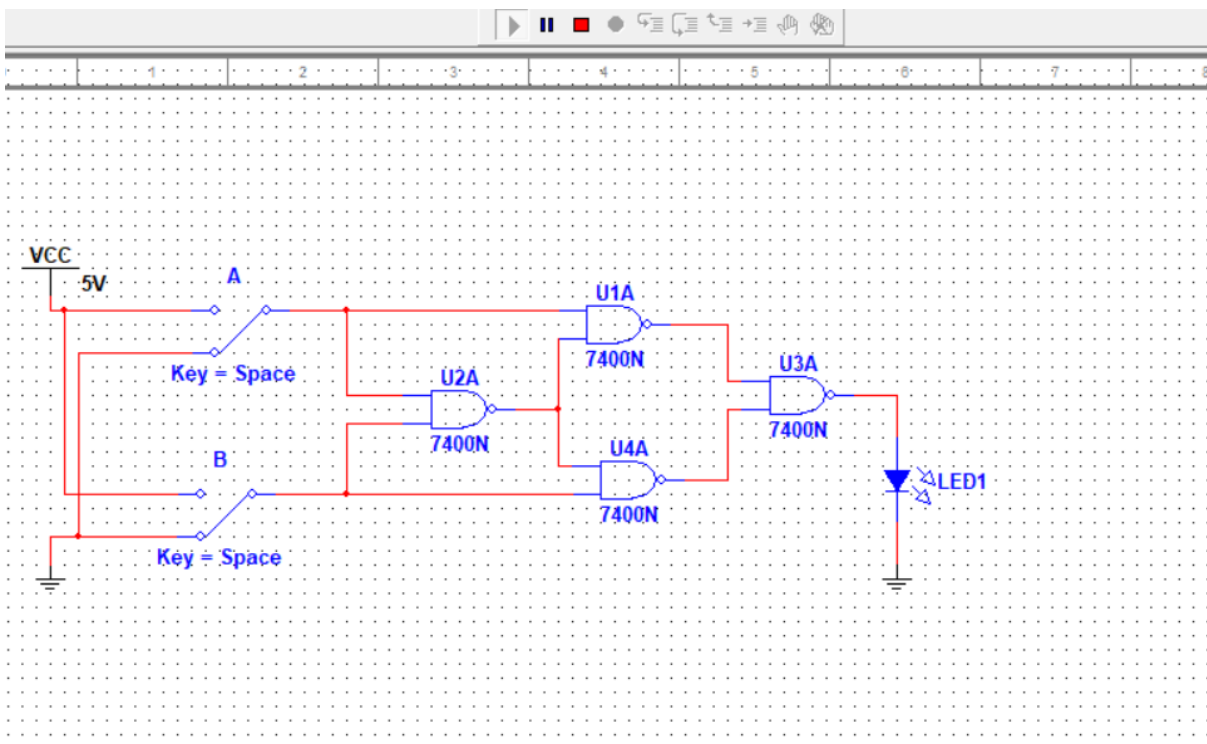
7.14 A greater or equal to B



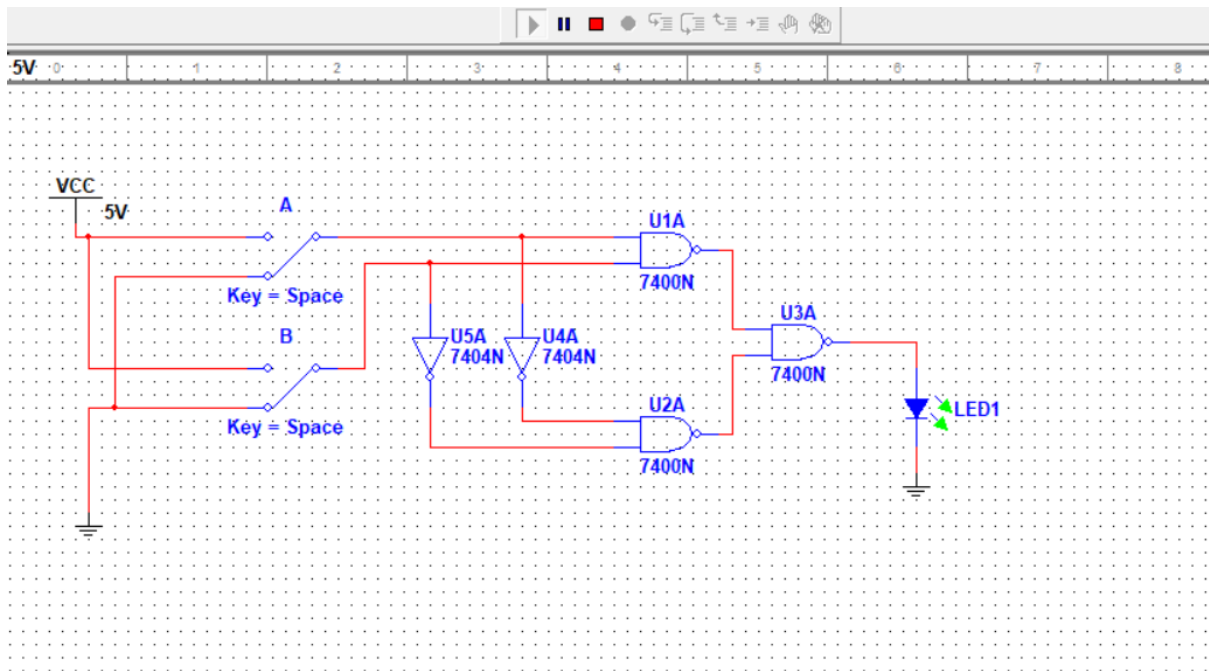
Practical 7.2



Practical 7.3

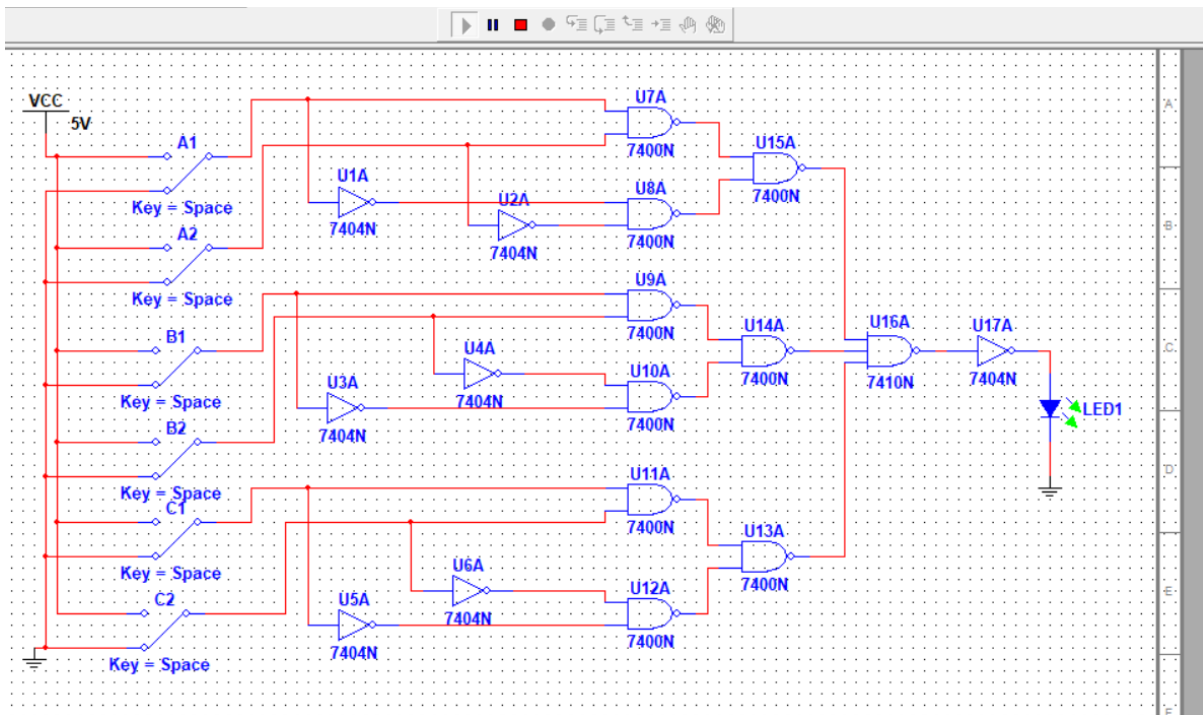


Practical 7.4

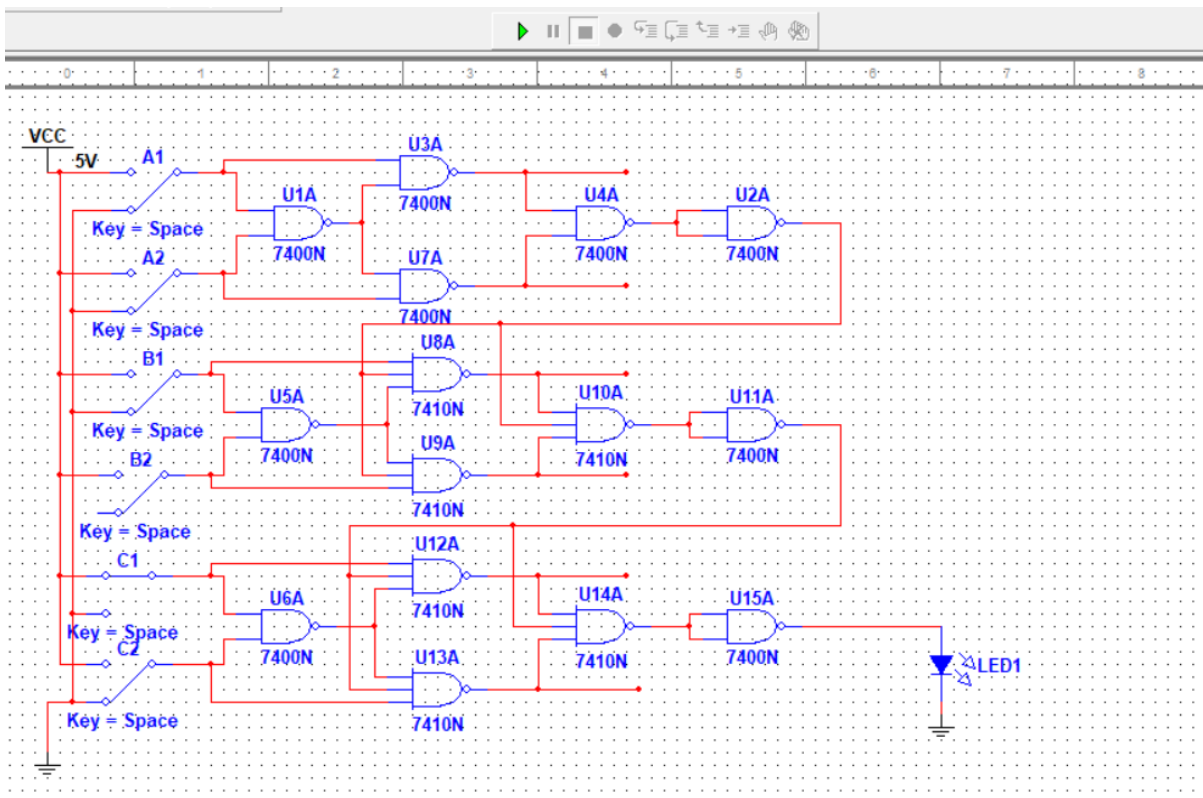


A	B	Z
0	0	1
0	1	0
1	0	0
1	1	1

PRACTICAL 7.5



PRACTICAL 7.6



ANSWERS TO EXERCISES

EXERCISE 7.1

A	B	0	AB	\overline{AB}	A	\overline{AB}	B	$\overline{AB} + \overline{BA}$	$\overline{A+B}$	$\overline{A+B}$	$\overline{AB} + \overline{AB}$	\overline{A}	$A + \overline{B}$	\overline{B}	$\overline{A+B}$
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	1	0	0	1
1	0	0	0	1	1	0	0	1	1	0	0	0	1	1	0
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

0 1 2 3 4 5 6 7 8 9 10 11 12 13

\overline{AB}	1
1	1
1	1
1	1
0	1

14 15

EXERCISE 7.2

$$= \overline{AB} + \overline{BA}$$

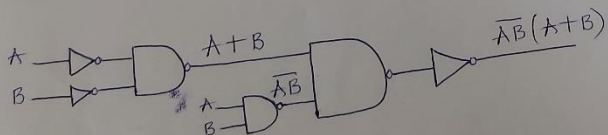
$$= \overline{AB} + \overline{BA}$$

$$= (A + \overline{B})(B + \overline{A})$$

$$= AB + A\overline{A} + B\overline{B} + \overline{B}A$$

$$= \underline{\underline{AB + \overline{AB}}}$$

EXERCISE 7.3



EXERCISE 7.5

X	Y	OUTPUT
000	001	0
010	010	0
111	000	1
100	101	0
101	110	1
100	011	1
110	000	1

EXERCISE 7.6 : In relation to diagram 7.6 above

EXERCISE 7.6

a) $\overline{B_1 B_1 B_2} P \Rightarrow B_1 (\overline{B_1 + B_2}) A_1 \equiv A_2$
 $= \overline{(B_1 \overline{B_1} + B_1 \overline{B_2})} A_1 \equiv A_2$
 $= \overline{(0 + B_1 \overline{B_2})} A_1 \equiv A_2$
 $= \overline{(B_1 \overline{B_2})} (A_1 \equiv A_2)$
 $= \overline{(B_1 > B_2)} (A_1 \equiv A_2)$

b) $\overline{B_2 \overline{B_1} \overline{B_2}} (A_1 \equiv A_2)$
 $\Rightarrow \overline{B_2 (\overline{B_1 + B_2})} (A_1 \equiv A_2)$
 $= \overline{(B_2 \overline{B_1} + B_2 \overline{B_2})} (A_1 \equiv A_2)$
 $= \overline{(B_2 \overline{B_1})} (A_1 \equiv A_2)$
 $= \overline{(B_1 < B_2)} (A_1 \equiv A_2)$

c) $\overline{(B_1 \overline{B_2}) P \cdot (\overline{B_1} B_2) P} \cdot P$
 $= \overline{\{(B_1 \overline{B_2}) P \cdot (\overline{B_1} B_2) P\}} P$ take $P = \{A_1 \equiv A_2\}$
 $= \overline{\{(B_1 + B_2 + \overline{P}) (B_1 + \overline{B_2} + \overline{P})\}} P$
 $= \overline{(B_1 + B_2) (B_1 + \overline{B_2})} P$
 $= \overline{(\overline{B_1} B_1 + \overline{B_1} \overline{B_2} + B_1 B_2 + B_2 \overline{B_2})} P$
 $= \overline{(0 + \overline{B_1} \overline{B_2} + B_1 B_2 + 0)} P$
 $= \overline{(\overline{B_1} \overline{B_2} + B_1 B_2)} P$
 $\Rightarrow \overline{(B_1 \equiv B_2)} (A_1 \equiv A_2)$

7.0 CONCLUSION

All circuits were successfully connected.

The experiment was successfully done as the objectives were met. All connected circuits responded in accordance with their respective truth tables

A better understanding of logic circuits was attained simply by observing the outputs produced by several different logic circuit connections. The output was observed on an LED which lit up upon a "HIGH" output and was off when output was "LOW" once the software was simulated.

8.0 APPLICATIONS

The NAND gate is widely used in integrated circuit logic since just about all types of operation can be obtained, although usually at greater cost. However, in designing logic systems, as we shall see later, the design tools and techniques are mostly better adapted to the use of the three fundamental operations of AND, OR, NOT so methods of converting the resulting expressions to the exclusive use of NAND have to be used.

Some formal methods do exist but in most practical cases but these are heavy-handed and cumbersome and have little advantage over simple experience. The fact that multi-input gates can be used for operations requiring fewer inputs than are available by leaving inputs disconnected means that when designing a large logic system it is easier to ensure that the number of modules (or packages) can be kept to a minimum. This is usually the most important economic factor nowadays.

9.0 REFERENCES

- [1] William Kleitz, 2006, Digital Electronics with VHDL, Prentice Hall ISBN-100131714902 [Practical 1]
- [2] Maini Anil K., Digital Electronics: Principles, Devices and Applications, 2007, John Wiley and Sons Ltd, ISBN 978-0-470-03214-5. [Theory 1]
- [3] Thomas L. Floyd, 2006, Digital Fundamentals with PLD Programming, Prentice Hall ISBN-10: 0131701886 [Practical 2]
- [4] Sedha R.S, A textbook of Digital Electronics, S. Chand, 2010 [Theory 2]
- [5] Alan C. Diixon, JamesL. Antonakos, 2000, A Practical Approach To Digital Electronics, Prentice Hall ISBN-10: 0137275 951.

