BINARY TO GRAY CODE CONVERTER

DEPARTMENT OF ELECTRICAL POWER
TECHNOLOGICAL UNIVERSITY
PAKOKKU
JULY, 2007

BINARY TO GRAY CODE CONVERTER

DEPARTMENT OF ELECTRICAL POWER
TECHNOLOGICAL UNIVERSITY
PAKOKKU
JULY, 2007

BINARY TO GRAY CODE CONVERTER

BY

U AUNG THAT TUN

DEMONSTRATOR (E.P)

A.G.T.I (E.P)

B.Sc (Physics)

TECHNOLOGICAL UNIVERSITY

CONTENTS

ACKNOLEDGEMENT

The authors wish to express their deep gratitude to H.E U Thaung, Minister, Ministry of Science and Technology for the opening for Technological University, Pakokku.

The authors would like to express the heart felt gratitude to Dr Shwe Sin Htay, Principle, Technological University, Pakokku for her kindly permission.

The author would be greatly indebted to Daw Mi Mi Aung Lecturer and Head of Department of Electrical Power, Technological University, Pakokku for accepting and giving guidance.

ABSTRACT

The Gray Code is useful code used in digital systems. It is used primarily for indicating the angular position of a shaft on rotating machinery such as automated lathes and drill presses. This code is like binary in that it can have as many bits as and the more bits, the more possible combinations of output codes.

The difference between the Gray Code and the regular binary code is that the Gray Code varies only 1 bit from entry to the next entry. The conversion between Gray Code and binary code are done by using Karnaugh Map. By using this method, the conversion can be done simply with exclusive-OR gates.

LIST OF FIGURES

LIST OF TABLES

CHAPTER I

INTRODUCTION

Electronic circuits can be divided into two broad categories, digital and analog. Digital electronic involves quantities with discrete values and analog electronics involves quantities with continuous values.

Digital electronic involves circuits and systems in which there are only two possible states representing two different voltage levels (High / Low), opened and closed switches. In digital system combinations of two states called codes are used to represent numbers, symbols, alphabetical characters and other types of information. The two state numbers system is called binary and its two digits are '0' and '1'.

There are many specialized codes used in digital systemsw such as Binary, BCD, Gray, ASCII, etc. The Binary is commonly used in digital system and the conversion of ~~BCD~~ Binary to Gray is done simply with exclusive-OR gates by using Karnaugh maps.

Table 2.1 Four Bits Gray Code

Decimal	Binary	Gray Code
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101
10	1010	1111
11	1011	1110
12	1100	1010
13	1101	1011
14	1110	1001
15	1111	1000

CHAPTER II

BACKGROUND THEORY

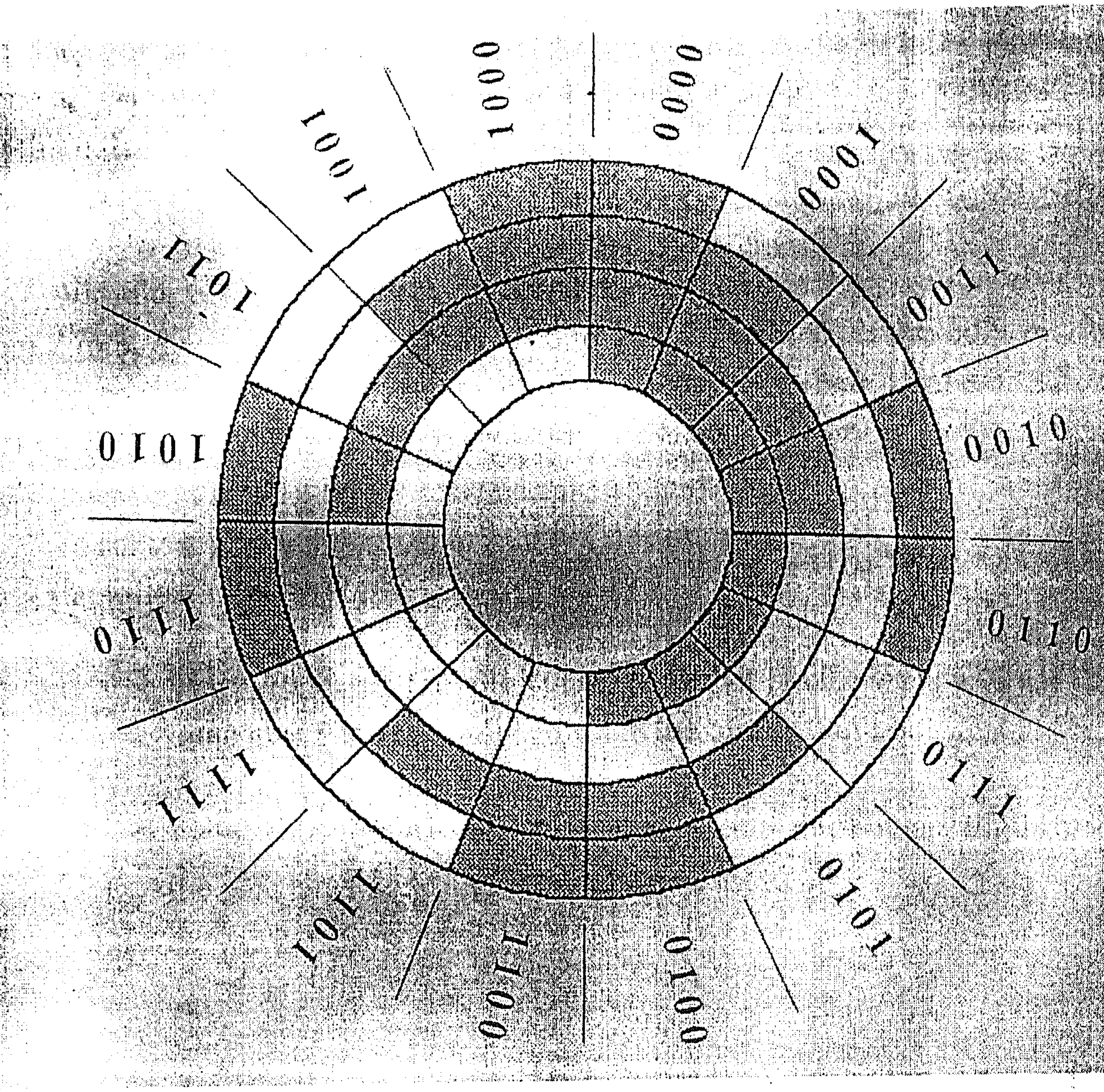
2.1 The Gray Code

The Gray Code is unweighted and is not an arithmetic code: that is, there are no specific weights asssigned to the bit positions. The important feature of the Gray Code is that it exhibits only a single bit change from one code word to the next in sequence. This property is important in many applications, such as shaft position encoders, where error susceptibility increases with the number of bit changes between adjacent numbers in a sequence.

Table-2.1 is a listing of the 4-bits Gray Code for decimal numbers 0 through 15. Binary numbers are shown in the table for reference. Like binary numbers, the Gray code can have any number of bits. Notice the single bit change between successive Gray Code words. For instance, in going from decimal 3 to 4, the Gray Code changes from 0010 to 0110, while the binary code changes from 0011 to 0100, a change of three bits. The only bit change is in the third bit from the right in the Gray Code ; the others remain the same.

2.2 Gray Code Application

The Gray code is a useful code used in digital systems. It is used primarily for indicating the angular position of a shaft on rotaitng machinery such as automated lathes and drill presses. This code is like binary in that it can have as many bits as necessary and the more bits, the more possible combinations of output codes (numbers of combinations = 2^N) are also available. A 4-bits Gray Code, for example, will have $2^4 = 16$ different representations, giving a resolution of one out of 16 possible angular positions at 22.5 degrees each (360/16=22.5). The difference between the Gray Code and the regular binary code is illustrated in table 2.1. Notice in the table that the Gray Code varies by only 1-bit from one entry to the next and from the last entry (15) back to the beginning (0). Now, if each Gray Code represents a different position on a rotating wheel, as the wheel turns, the code reads from one position to the next would vary by only 1-bit.



OUTPUT '1'

OUTPUT '0'

Figure 2.1 Gray Code Wheel

If the same wheel labeled in binary, as the wheel turned from 7 to 8, the code would change from 0111 to 1000. If the digital machine happened to be reading the shaft position just as the code was changing, it might see 0111 or 1000, but since all 4-bits are changing (0 to 1 or 1 to 0) the code that it reads may be anything from 0000 to 1111. Therefore, the potential for an error using the regular binary system is great.

With the Gray code wheel, on the other hand, when the position changes from 7 to 8, the code changes from 0100 to 1100. The MSB is the only bit that changes, so if a read is taken right on the border between the two numbers, either a 0100 is read or a 1100 is read (no problem).

To convert a binary number to a Gray Code number,
the following rules apply.

1. The most significant digit (Left Most Bit) in the Gray Code is the same as the corresponding digit in the binary number.

2. Going from left to right, add each adjacent pair of binary digits to get the next Gray code digit, regradless carries.

For instance - Let us convert the binary number 1010 to Gray Code.

Step 1 - The left most Gray digit is the same as the left most binary digit.

$$\begin{array}{ccccl} 1 & 0 & 1 & 0 & \text{Binary} \\ \downarrow & & & & \\ 1 & & & & \text{Gray} \end{array}$$

Step 2 - Add the left most binary digit to the adjacent one.

$$\begin{array}{ccccccl} 1 & + & 0 & & 1 & 0 & \text{Binary} \\ & & \downarrow & & & & \\ 1 & & 1 & & & & \text{Gray} \end{array}$$

Step 3 - Add the next adjacent pair

$$\begin{array}{ccccccl} 1 & 0 & + & 1 & & 0 & \text{Binary} \\ & & & \downarrow & & & \\ 1 & 1 & & 1 & & & \text{Gray} \end{array}$$

Step 4 -Add the last adjacent pair

$$\begin{array}{cccccl} 1 & 0 & 1 & + & 0 & \text{Binary} \\ & & & & \downarrow & \\ 1 & 1 & 1 & & 1 & \text{Gray} \end{array}$$

The conversion is now complete and the Gray Code is 1111.

2.3 Karnaugh Map

Karnaugh map named for its originator, is another method of simplifying logic circuits. It still requires to reduce the equation to a sum-of-products (SOP) form, but from that a sytem approach which will always produce the simplest configuration possible for the logic circuit.

A Karnaugh map (K-map) is similar to a truth table in that table in that it graphically shows the output level of a Boolean equation for each of the possible input variable combinations. Each output level is placed in a separate cell of the K-map. K-maps can be used to simplify equations having two, three, four, five or six different input variables. Solving five and six variable K-maps is extremely cumbersome and can be more practically solved using advanced computer techniques.

Determining the number of cells in a K-map is the same as finding the number of combinations or entries in a truthe table. A two variable map will require $2^2 = 4$ cells. A three variable map will require $2^3 = 8$ cells. A four variable map will require $2^4 = 16$ cells. Teh three different K-maps are shown in figure 2.2.

Each cell within the K-map corresponds to a particular combination of the input variables. For example, in the two variable K-map, the upper left cell corresponds to , $\overline{A}\overline{B}$ the lower left cell is $A\overline{B}$. the upper right cell is $\overline{A}B$, and the lower right cell is AB.

Also notice that when moving from one cell to an adjacent cell, only one variable changes. For example, look at the three variable K-map. The upper left cell is $\overline{A}\overline{B}\overline{C}$, the adjacent cell just below it is $\overline{A}B\overline{C}$. In that case the $\overline{A}\,\overline{C}$ remained the same and only the $\overline{B}$ change to B. The same holds true for each adjacent cell.

	$\overline{B}$	B
$\overline{A}$		
A		

Figure 2.2 (a) Two variable

	$\overline{C}$	C
$\overline{A}\,\overline{B}$		
$\overline{A}B$		
AB		
$A\overline{B}$		

Figure 2.2 (b) Three variable

	$\overline{C}\,\overline{D}$	$\overline{C}D$	CD	$C\overline{D}$
$\overline{A}\,\overline{B}$				
$\overline{A}B$				
AB				
$A\overline{B}$				

Figure 2.2 (c) Four variable

2.4 The Exclusive - OR Gate

The OR gate provides a HIGH output if one input or the other input is HIGH, or if both input are HIGH. The exclusive-OR; on the other hand, provides a HIGH output if one input or the other input is HIGH but not both. This point is made more clear by comparing the truth tables for an OR gate versus an exclusive-OR gate, as shown in table 2.2.

The Boolean equation for the exclusive-OR function is written $X = \overline{A}B + A\overline{B}$ and can be construsted using the combinationl logic shown in figure 2.3. By experimenting and using Boolean reduction, the other combinations of the basic gates that provide the exclusive-OR function. For example, the combination of AND,OR and NAND gates shown in figure 2.4 will reduce to the "one-or-the-other-but-not-both" (X-OR) function.

The exclusive-OR gate is common enough to deserve its own logic symbol and equation,shown in figure 2.5.

Note - The shorthand method of writing the Boolean equation is to use a plus sign with a circle around it.

Table 2.2. Truth Tables for an OR Gate Versus an Exclusive-OR Gate

A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

(OR)

A	B	X
0	0	0
0	1	1
1	0	1
1	1	0

(Exclusive-OR)

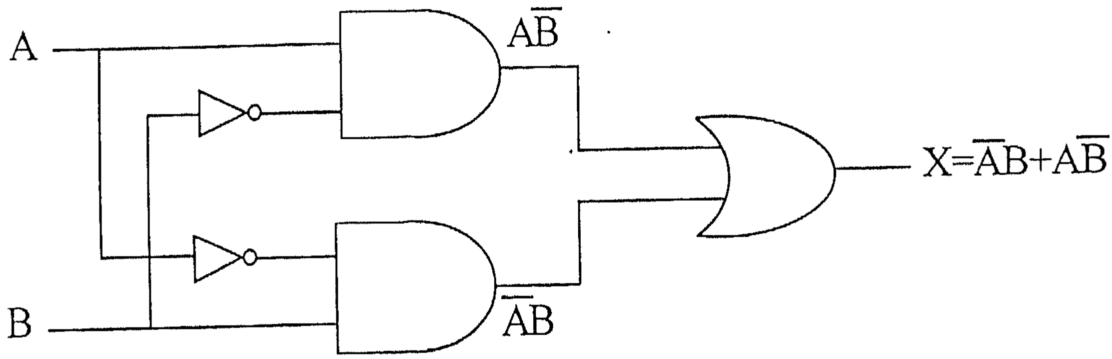
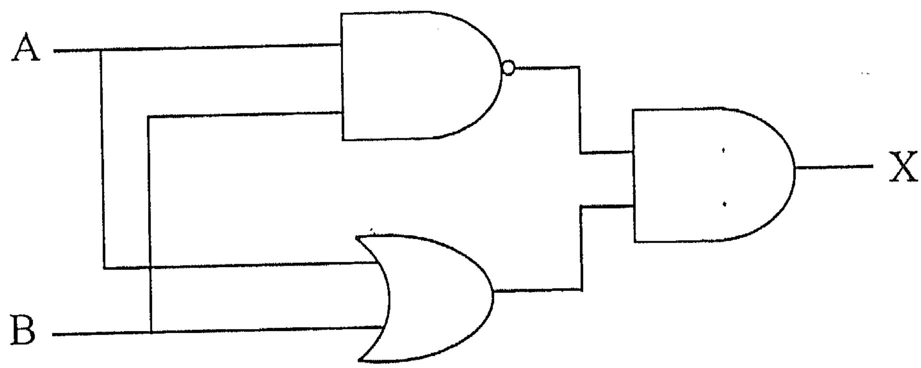
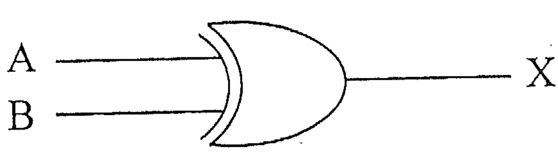


Figure 2.3 Logic Circuit for providing for exclusive-OR funciton.



$$X=\overline{AB}\,(A+B)$$
$$X=(\overline{A}+\overline{B})\,(A+B)$$
$$X=\overline{A}A+\overline{A}B+\overline{B}A+\overline{B}B$$
$$X=\overline{A}B+A\overline{B}$$

Figure 2.4 Exclusive-OR builts with and AND-OR-NAND combination



$$X=A\oplus B=\overline{A}B+A\overline{B}$$

Figure 2.5 Logic symbol and equation for the exclusive-OR

2.5 LED

LED is a light emitting diode. A LED emits radiation when forward-biased. Because free electrons recombine with holes near the junction. As the free electrons fall from a higher energy level to a lower one, they gives up energy in the from of heat and light. LEDs emit red, green,yellow, blue, orange and infrared (invisible) light. By using elements like Gallium , Arsenic and Phosphorus, a manufacture can produce LEDs that produce visible radiation are useful in test instrument, pocket calculator, etc.

LEDs may be connected the common anode type and common cathode type. The common anode type connects a current limiting resistor between each LED and ground. The size of this resistor determines how much current flows through the LED. (Typical LED current is between 1 and 50 mA). The common cathode type uses a current-limiting resistor between each LED and +Vcc.

2.6 Exclusive-OR Gate (74LS86 IC)

The pinout diagram of a 74LS86 IC, a TTL quad 2-input exclusive-OR gate is shown in figure 2.6. This digital integreated circuit (IC) contains four 2-input exclusive-OR gate inside a 14-pin dual-in-line package (DIP).

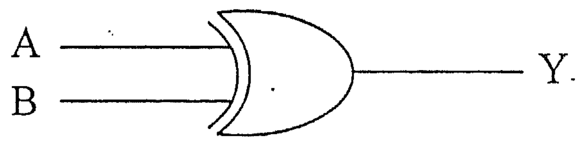


Figure 2.6 Logic symbol for exclusive-OR gate

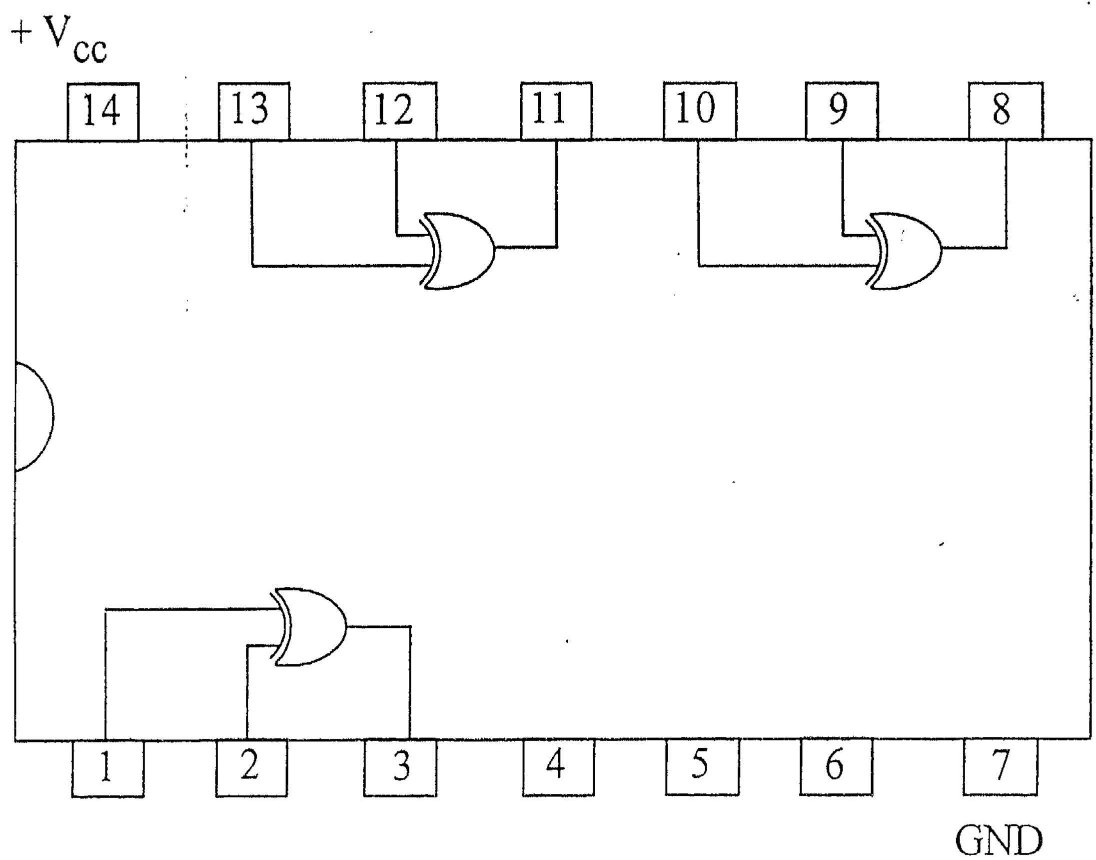


Figure 2.7 Pinout diagram of a 74LS86 IC

CHAPTER III
OPERATION OF THE SYSTEM

3.1 Gray Code

Table 3.1

Decimal	Binary				Gray			
	A	B	C	D	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1
2	0	0	1	0	0	0	1	1
3	0	0	1	1	0	0	1	0
4	0	1	0	0	0	1	1	0
5	0	1	0	1	0	1	1	1
6	0	1	1	0	0	1	0	1
7	0	1	1	1	0	1	0	0
8	1	0	0	0	1	1	0	0
9	1	0	0	1	1	1	0	1
10	1	0	1	0	1	1	1	1
11	1	0	1	1	1	1	1	0
12	1	1	0	0	1	0	1	0
13	1	1	0	1	1	0	1	1
14	1	1	1	0	1	0	0	1
15	1	1	1	1	1	0	0	0

3.2 Karnaugh Maps

For practical consideration code conversions are made for each bit.

(a) For Y_3, the Karnaugh map can draw as follow.

	$\overline{C}\,\overline{D}$	$\overline{C}D$	CD	$C\overline{D}$
$\overline{A}\,\overline{B}$	0	0	0	0
$\overline{A}B$	0	0	0	0
AB	1	1	1	1
$A\overline{B}$	1	1	1	1

(a)

By minimization, $Y_3 = A$

(b) For Y_2, the Karnaugh map can draw as follow.

	$\overline{C}\,\overline{D}$	$\overline{C}D$	CD	$C\overline{D}$
$\overline{A}\,\overline{B}$	0	0	0	0
$\overline{A}B$	1	1	1	1
AB	0	0	0	0
$A\overline{B}$	1	1	1	1

(b)

By minimization, $Y_2 = A \oplus B$

(c) For Y_1, the Karnaugh map can draw as follow.

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	0	0	1	1
$\bar{A}B$	1	1	0	0
AB	1	1	0	0
$A\bar{B}$	0	0	1	1

(a)

By minimization, $Y_1 = B \oplus C$

(d) For Y_0, the Karnaugh map can draw as follow.

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	0	1	0	1
$\bar{A}B$	0	1	0	1
AB	0	1	0	1
$A\bar{B}$	0	1	0	1

(b)

By minimization, $Y_0 = C \oplus D$

3.3 Gray Code Conversions

By Karanugh map minimization the conversion circuit can be (ne as followed by using 3 exclusive-OR gates.

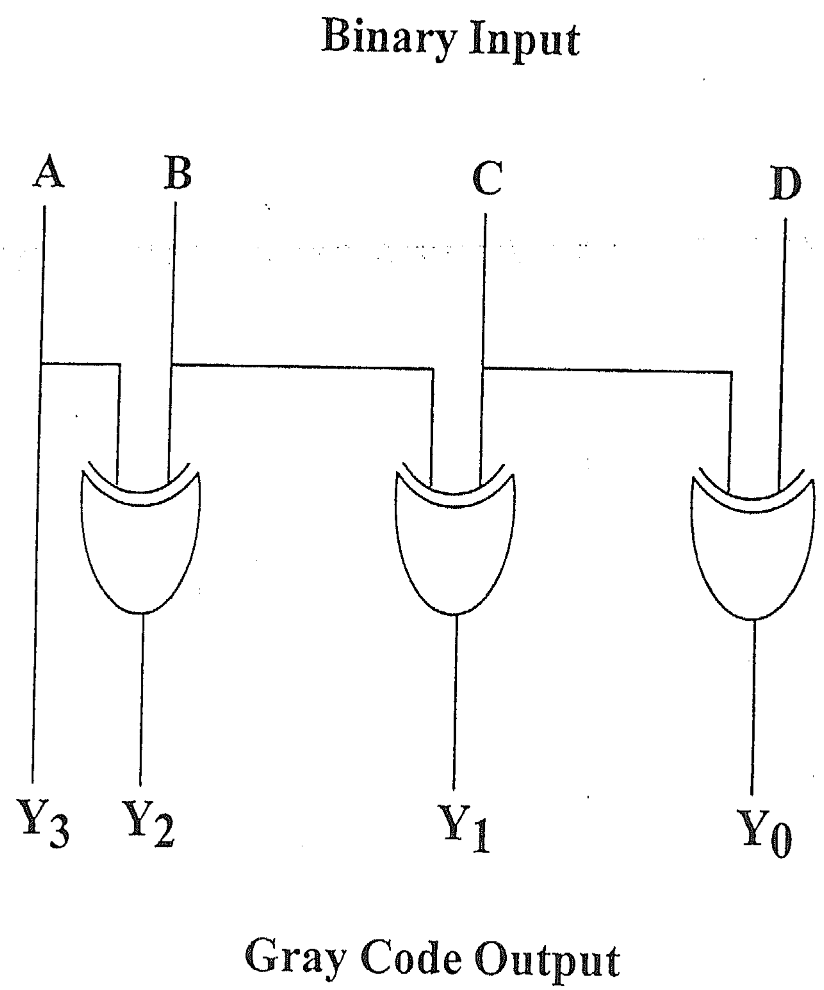


Figure 3.1 Binary-to-Gray Code

ABCD are used as **Binary** inputs and **D** stands for LSB and **A** stands for MSB corresponding output Gray Code take from Y_3 Y_2 Y_1 Y_0.

For exclusive-OR gate 7486(quad-X-OR) IC is used and for power supply, 3AA dry cells are used for simplicity. The pins connections and the whole circuit block diagram is shown in figure 3.2.

For Binary input, four slide switches are used and up position corresponds to 1 state and down position corresponds to 0 state.

3.4 Required Equipments

1. 74LS86 IC	1 No
2. Pin Base	1 No
3. Resister 330 ohm	8 Nos
4. LED	8 Nos
5. 1.5 Volt Battery	3 Nos
6. Universal Chip	1 No

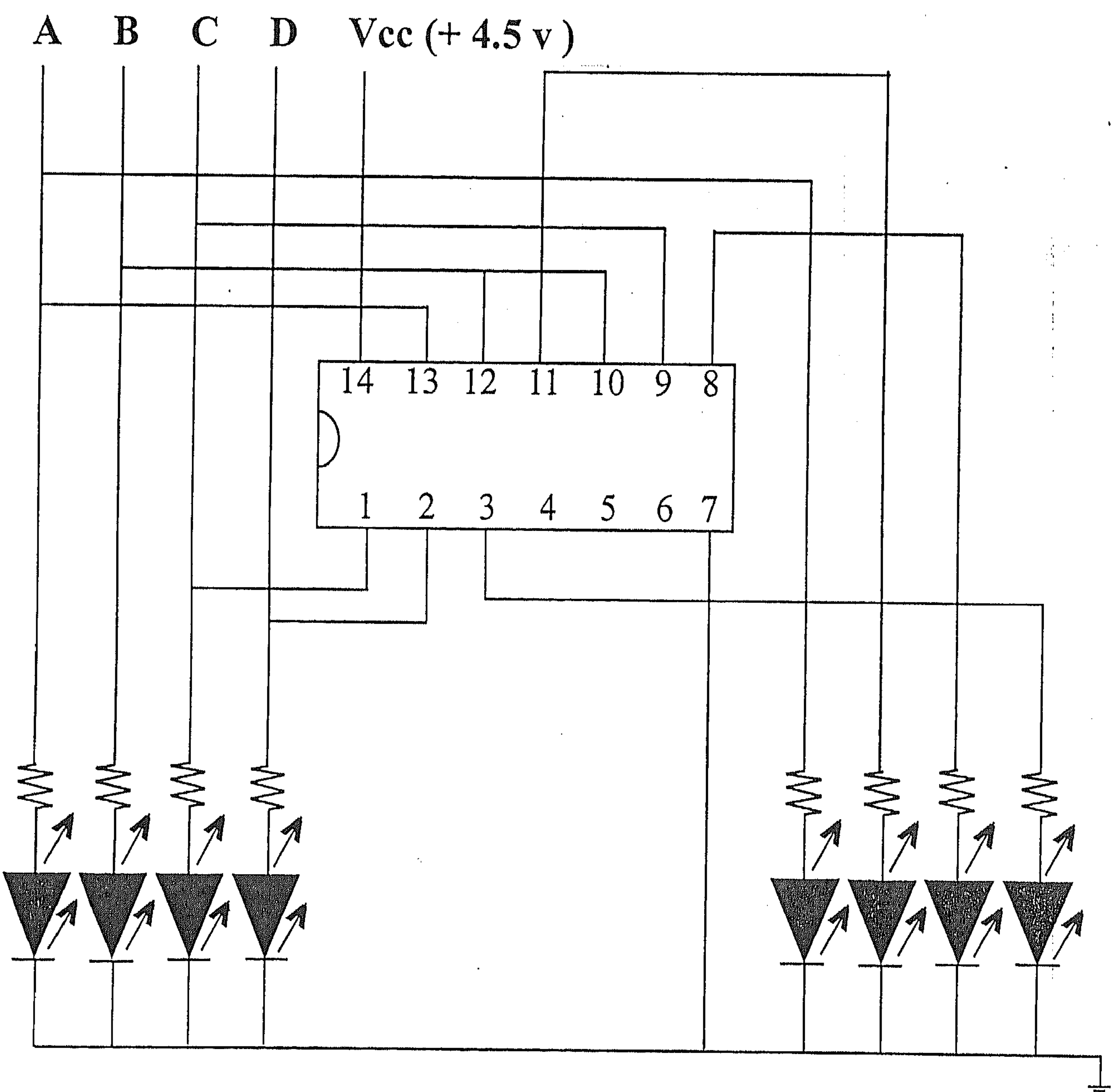


Figure 3.2 Circuit diagram for Binary to Gray Code Converter.

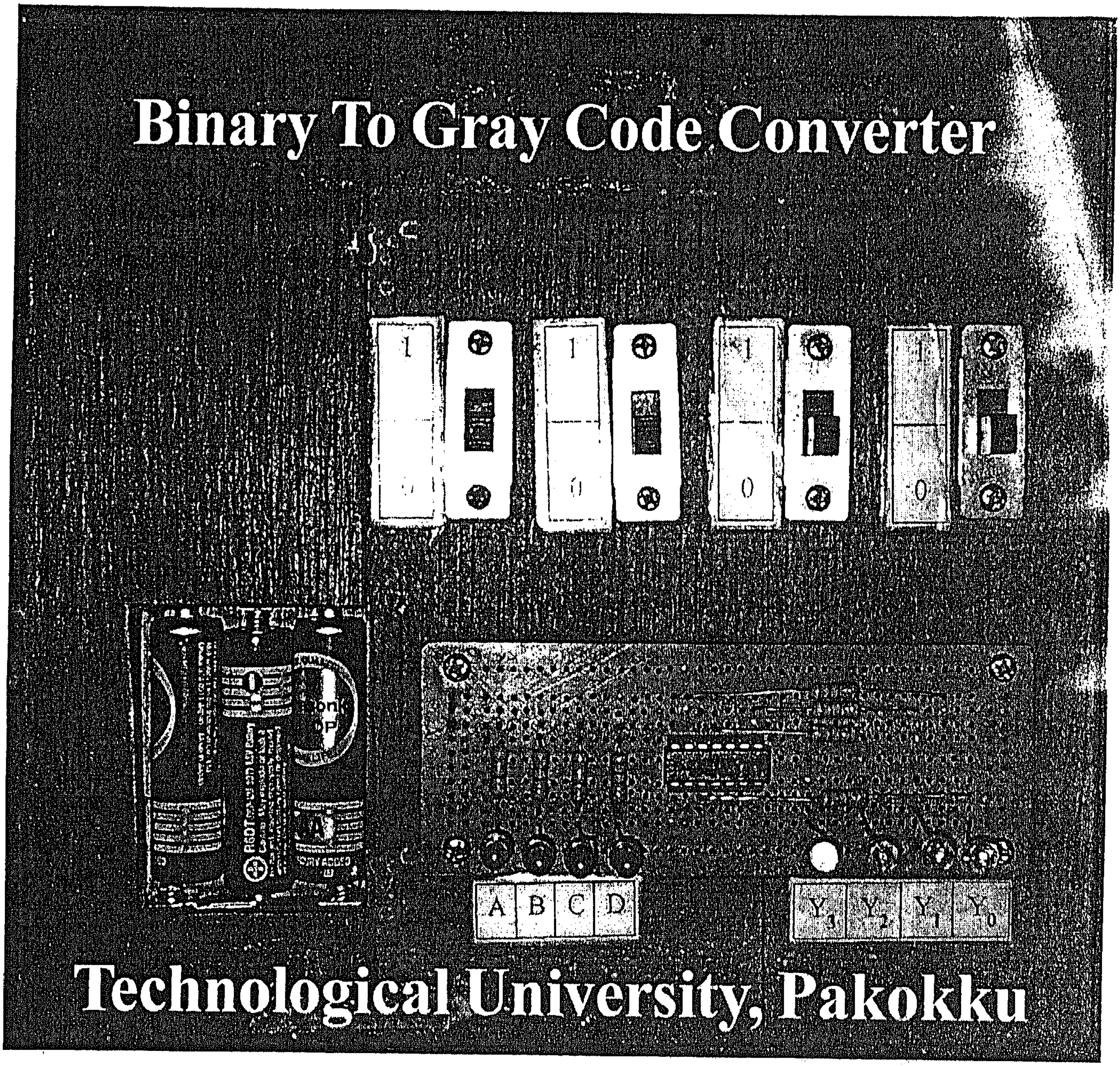


Figure 3.3 Photo of Binary To Gray Code Converter

CHAPTER 4
RESULT AND DISCUSSION

Decimal to Gray code converter converts correctly decimal 0 0 0 0 to 1 1 1 1 into Gray code. The circuit diagram is very simple and only use an 74LS86 IC . Unless the Karnaugh map is used, many gates may be used. But the result of Karnaugh map minimization, it can work only using the exclusive-OR gates. Normally IC supply voltage must be 5V by using power supply circuit. But for digital circuit, digital (1)state can get by using dc 4.5 V. For dc 4.5 V, serial combination of 3 AA dry cells are used to supply the IC.

Code conversion takes part an important role in digital system. Not only conversion to Gray code, other code conversion can be done by using this way. For example, conversion to ASCII code is also possible. By using 4-digits Binary number, we can control the shift's angular positions in 16 positions and the degree between two adjacent positions is 22.5°. If 5-digits Binary numbers are used, the 32 shaft's angular position can be controlled and the degree between two adjacent positions may be 11.25°. Code conversion for 5-digits Binary numbers can be done like this way. This project is the first step for code conversion and another code conversion can do in future projects based on this way.

REFERENCES

1. Bignell, J and Donovan, R. (2001):"Digital Electronics" (Singapore, Thomson Asia Pte.Ltd)

2. Duncan, T.(1999):"Success in Electronics" (London: Murray)

3. Floyd, Thomas.L (2003): "Digital Fundamentals" (Prentice Hall)

4. Maini, A.K. (2002):"Digital Electronics Principles and Applications" (Delhi, Khanna, R.C.)

5. Malvino, A.P. and Leanch, D.P. (1988):"Digital principls and Applications" (Singapore: Mc Graw-Hill)

6. Sinclair, I. (2000): "Practical Electronics Handbook"