IOSR Journal of Engineering (IOSRJEN) www.iosrjen.org
ISSN (e): 2250-3021, ISSN (p): 2278-8719
Vol. 04, Issue 12 (December 2014), ||V4|| PP 52-57

Microcontroller based kit design for Octal to Hexadecimal system Conversion without returning to another system

Noaman Abd Elnour Abd Elrahim[1], Abdelrasoul jabar Alzubaidi[2]

[1]*Applied Sciences Collage- Red sea University- Port Sudan- Sudan.*

[2] *Sudan University of science and technology- Engineering Collage-School of electronics-*

ABSTRACT: *The ordinary method of conversion from the octal numbering system to the hexadecimal numbering system goes through two steps . The first step is to convert from octal to binary and the second step is to convert from the binary to hexadecimal. The same procedure of conversion can be conducted by going through the decimal system .This paper deals with a direct method of conversion from octal into hexadecimal without returning to another system. Although it is little complicated, but it is useful to find such procedure of conversion and then prove it mathematically. Building such a direct conversion algorithm is useful in the computing work. The system design and the algorithm can be used as a model for teaching the undergraduate students in the field of computer and electronics.*

KEYWORDS: *Binary, Octal, Hexadecimal, Numbering systems, Computer, Algorithm.*

I. INTRODUCTION

Calculation systems are very important in the theory and practical in physics, mathematics and computers. It is possible and easy to calculate in the decimal, binary, hexadecimal and octal number systems. Also it is possible to convert between decimal and any other systems, and between binary and any other system without returning to a third system. But if it is needed to convert octal into hexadecimal and vise versa, we must first convert to binary or decimal then convert to the needed system.

We try to find out a direct method of conversion between octal and hexadecimal without returning to another system. Table (1) below shows a sample of look up table for conversion from octal to hexadecimal system and vice versa.

Table (1): A look up table for converting numbers octal to hexadecimal and vice versa

Octal	**Hexadecimal**
0	**0**
1	**1**
2	**2**
3	**3**
4	**4**
5	**5**
6	**6**
7	**7**
10	**8**
11	**9**
12	**A**
13	**B**
14	**C**
15	**D**
16	**E**
17	**F**

Several steps have to be conducted to attain this goal. A step by step procedure will be demonstrated in this paper. The example below assumes four digits octal number $(A_3A_2A_1A_0)_8$ and three digits equivalent hexadecimal number $(X_2X_1X_0)_{16}$.

Octal to hexadecimal:

$(A_3A_2A_1A_0)_8 \Rightarrow (X_2X_1X_0)_{16}$

Calculation steps for the digits X_0 , X_1 , X_2 are as follows :

Digit X0 :

$X_0 = A_0$ if $A_1 even$.

$X_0 = (A_0 + 8)_{16}$ if A_1 odd .

Digit X_1 :

$X_1 = \left(\frac{A_2A_1}{2}\right)_8$ then convert to hexadecimal if A_1 even and A_2 less than four.

$X_1 = \left(\frac{(A_2 - 4)A_1}{2}\right)_8$ then convert to hexadecimal and hold $(1)_{16}$ to X_2 if A_1 even and A_2 more than three.

$X_1 = \left(\frac{(A_2)(A_1 - 1)}{2}\right)_8$ then convert to hexadecimal and carry $(8)_{16}$ to X_0 if A_2 less than four and A_1 odd

$X_1 = \left(\frac{(A_2 - 4)(A_1 - 1)}{2}\right)_8$ then convert to hexadecimal and carry $(8)_{16}$ to X_0 and hold $(1)_{16}$ to X_2 if A_1 odd and A_2 more than three

Digit X_2 :

$X_2 = (2A_3)_{16}$ if there is no hold

$X_2 = (2A_3+1)_{16}$ if there is $(1)_{16}$ held from X_1

Hexadecimal to octal:

$(X_2X_1X_0)_{16} \Rightarrow (A_3A_2A_1A_0)_8$

Calculation steps for the digits A0 , A_1 , A_2 , A3 are as follows :

Digit A_0 :

$A_0 = X_0$ if X_0 less than $(10)_8$

$A_0 = (X_0 - 8)_{16}$ then convert to octal and hold $(1)_8$ to A_1 if X_0 more than $(7)_8$

Digits A_2A_1 :

$A_2A_1 = 2(X_1$ converted to octal) if X_0 less than $(10)_8$ and X_2 even

$A_2A_1 = 2(X_1$ converted to octal) then $[(A_2 + 4)(A_1)]_8$ if X_0 less than $(10)_8$ and X_2 odd

$A_2A_1 = 2(X_1$ converted to octal) then $[(A_2)(A_1+1)]_8$ if X_0 more than $(7)_8$ and X_2 even

$A_2A_1 = 2(X_1$ converted to octal) then $[(A_2+4)(A_1+1)]_8$ if X_0 more than $(7)_8$ and X_2 odd

Digit A_3 :

$A_3 = \left(\frac{X_2 \text{ converted to octal}}{2}\right)_8$ if X_2 even

$A_3 = \left(\frac{X_2 \text{ converted to octal} - 1}{2}\right)_8$ if X_2 odd

The above operations in octal and hexadecimal repeat themselves when adding any other digits for any of the two systems.

II. HARDWARE COMPONENTS

The main hardware components in the design are:

1. Breadboard:

Bread board (protoboard) is a construction base for a one-of-a-kind electronic circuit, a prototype. Because the solderless breadboard does not require soldering, it is reusable, and thus can be used for temporary prototypes and experimenting with circuit design more easily. A variety of electronic systems may be prototyped by using breadboards, from small analog and digital circuits to complete central processing units (CPUs).

2. Microcontroller (Atmega32):

Atmega 32 is a microcontroller from Atmel 8-bit family with 32KB flash memory is used.

Figure (1) shows the pins description of the microcontroller (Atmega-32).

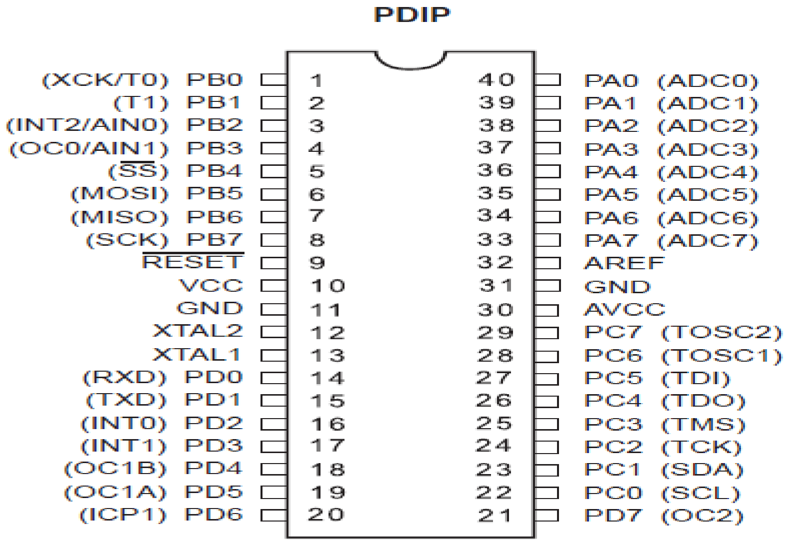

Figure (1) Atmega32 microcontroller pins

3. LCD 40x4

The LCD 40x4 is used in the system design. The function of the LCD is as a tool for displaying the data entry and the results of conversion between the two numbering systems.

4. (3×4) matrix keypad:

This device is used as a means for data entry.

III. THE SYSTEM DESIGN

The main block diagram is based on using a keypad for data entry, a microcontroller, a LCD. Figure (2) shows th3e block diagram for the hardware circuit.

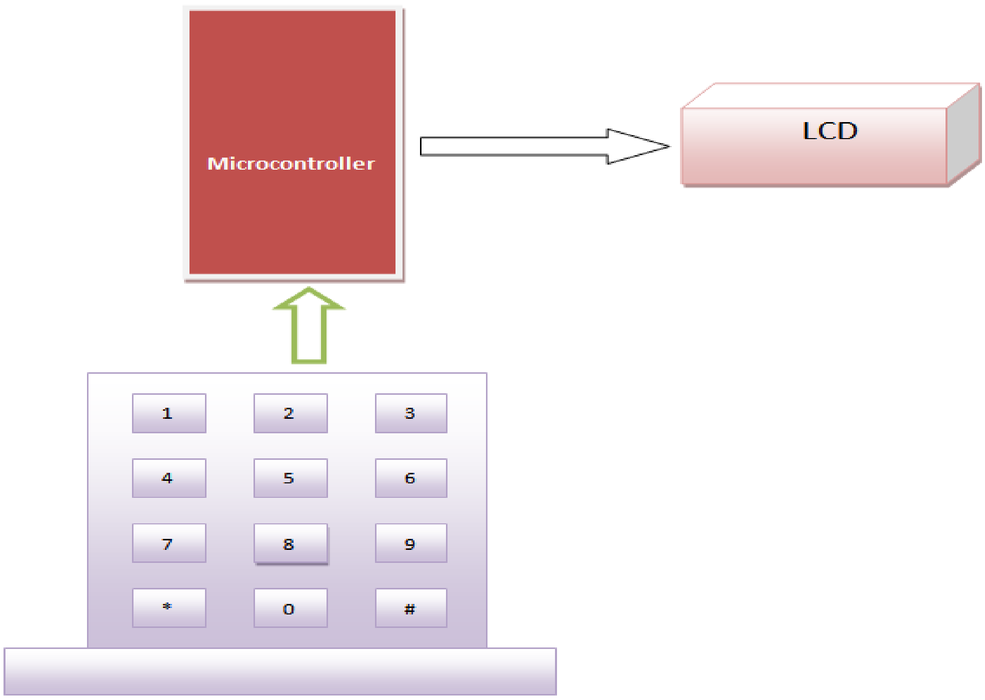

Figure (2) block diagram of the hardware circuit.

IV. SOFTAWE PROGRAM & ALGORITHM

To achieve the operation of the design, we need to go through five steps as follows:

1. Step one: Developing a program in Bascom language in the computer and producing (.hex) file format of the program.
2. Step two: Downloading the (.hex) into the microcontroller by using (Pony Prog) program as shown in figure (3) below.

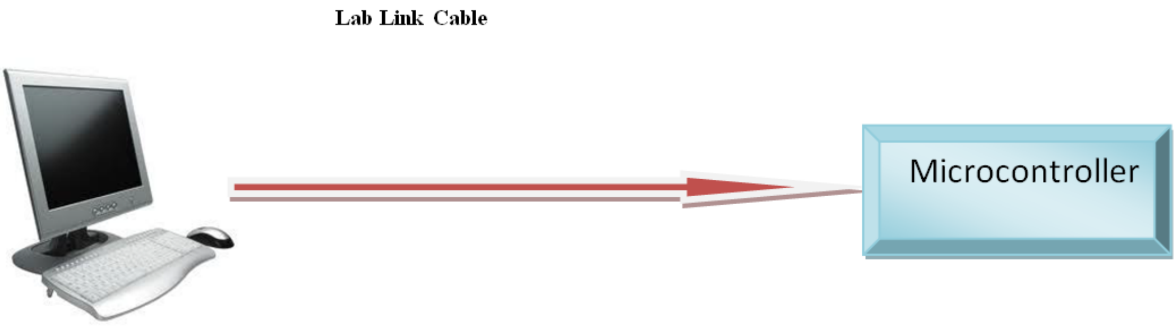

Figure (3) Connection for programming the microcontroller.

3. Step three: Operating the system by entering data from the matrix keypad and watching the results on the LCD.

The algorithm for performing a demonstration for the conversion between the octal and the hexadecimal systems contains an initialization step and two subroutines. The condition for calling the subroutine depends on the commands entered from the keypad. The command (#) is for calling octal to hexadecimal conversion subroutine. The command (*) is for calling hexadecimal to octal conversion subroutine the algorithm is:

Start

Initialization:

--- Clear all outputs from the microcontroller to the LCD.

--- Display on the LCD "system is ready".

--- Delay few seconds.

--- Display on the LCD "enter (#) for octal to hexadecimal conversion and (*) for hexadecimal to octal conversion".

Check keypad entry:

--- If the (key pressed is (#)), then call the octal to hexadecimal conversion subroutine.

--- If the (key pressed is (*)), then call the hexadecimal to octal conversion subroutine.

--- Go to check incoming data.

End.

Octal to hexadecimal conversion subroutine:

Start:

Digit X0 :

--- If the (A1 is even), then X0 is equal to A0.

--- If the (A1 is odd), then X0 is equal to (A0 + 8)16.

Digit X1 :

--- If the (A1 is even and A_2 less than four), then $X_1 = \left(\frac{A_2A_1}{2}\right)_8$.

--- If the (A1 is even and A_2 more than three), then hold $(1)_{16}$ to X_2 and X1 = $\left(\frac{(A_2-4)A_1}{2}\right)_8$.

--- If the (A1 is odd and A_2 less than four), then carry $(8)_{16}$ to X0 and X1 = $\left(\frac{(A_2)(A_1-1)}{2}\right)_8$.

--- If the (A1 is even and A_2 more than three), then carry $(8)_{16}$ to X0 , hold $(1)_{16}$ to X_2 and X1 =

$$\left(\frac{(A_2-4)(A_1-1)}{2}\right)_8 .$$

Digit X2 :

--- If there is no hold , then $X_2 = (2A_3)_{16}$.

--- If there is $(1)_{16}$ held from X_1 , then $X_2 = (2A_3+1)_{16}$.

Return.

Hexadecimal to octal conversion subroutine:
Start:
Digit A0 :
--- If the (X_0 less than $(10)_8$), then A0 is equal to X0.
--- If the (X_0 more than $(7)_8$), then hold $(1)_8$ to A_1 and $A_0 = (X_0 - 8)_{16}$.
Digit A1, A2 :

--- If the (X_0 less than $(10)_8$ and X_2 is even) , then $A_2A_1 = 2(X_1$ converted to octal).

--- If the (X_0 less than $(10)_8$ and X_2 is odd) , then $[(A_2 + 4)(A_1)]_8$ and $A_2A_1 = 2(X_1$ converted to octal).
--- If the (X_0 is more than $(7)_8$ and X_2 is even) , then $A_2A_1 = 2(X_1$ converted to octal).
--- If the (X_0 is more than $(7)_8$ and X_2 is odd) , then $[(A_2+4)(A_1+1)]_8$ and $A_2A_1 = 2(X_1$ converted to octal).

Digit A3 :

--- If (X_2 is even) , then $A_3 = \left(\frac{X_2 \text{ converted to octal}}{2}\right)_8$.

--- If (X_2 is odd) , then $A_3 = \left(\frac{X_2 \text{ converted to octal} - 1}{2}\right)_8$.

Return

V. RESULTS

Table (2) shows the results obtained from entering the three different octal numbers of and converting them into hexadecimal . Table (3) shows the results obtained from entering two hexadecimal numbers and converting it into octal .The program is made simple and user friendly.

Table (2) Results from entering the octal numbers and converting it into hexadecimal

Experiment number	**Octal number**	**Hexadecimal number**
Experiment one	**347**	**E7**
Experiment two	**6236**	**C9E**
Experiment three	**3556**	**76E**

Table (3) Results from entering the hexadecimal numbers and converting it into octal

Experiment number	**Hexadecimal number**	**Octal number**
Experiment one	**4C5**	**2305**
Experiment two	**579**	**2571**

VI. CONCLUSION

The functionality of the design is simple . The user has to follow the messages displayed on the LCD and enter the digits for the related numbering system. The system in its turn performs the conversion asked from the keypad. Based on the entered digits the program performs the related conversion subroutine . The results of conversion are displayed on the screen of the LCD. The hardware and the software of the system can be used as a means for demonstration for the undergraduate students . The program is written in Bascom language and donloaded into the microcontroller

REFERANCES:

[1] Sanchez, Julio; Canton, Maria P. (2007). Microcontroller programming: the microchip PIC. Boca Raton, FL: CRC Press. p. 37. ISBN 0-8493-7189-9.

[2] 2-Knuth, D. E. (1997), "Volume 2: Seminumerical Algorithms", The Art of Computer Programming (3rd ed.), Addison-Wesley, p. 192, ISBN 0-201-89684-2 .

[3] Arndt, Jörg (March 5, 2009). Algorithms for Programmers: Ideas and source code (draft).

[4] Morris Mano , Digital design, McGraww Hill Publishing company, 2001.

[5] Thomas L. Floyd, Digital fundamentals, Merill Publishing company, 2002

[6] -Mazidi Muhammed Ali, The 8051 microcontroller and Embedded system, Prentic Hall, 2007.

[7] -Sedra and Smith, Microelectronic circuits, Fourth edition, Oxford university Press, 2006

[8] R. S. Sedha, A text Book of Applied Electronics, Chad and company Ltd, New Delhi, 2002

[9] © MSC Electronic, BASCOM – AVR, Help Reference V1.11.8.3, 2006

[10] Jan Axelson, Microcontroller idea book, Copy right 1999

[11] Thomas L. Floyd, Electronic Devices 2nd Edition, Merill Publishing company, 2003