**TITLE:** Hex to BCD & BCD to Hex Conversion.

**OBJECTIVES:**

To Learnthe implementation of ALP for conversion of Hex to BCD & vice a versa.

**PROBLEM STATEMENT:**

To write 64 bit ALP to convert 4-digit Hex number into its equivalent BCD number and 5-digit BCD number into its equivalent HEX number. Make your program user friendly to accept the choice from user for: (a) HEX to BCD b) BCD to HEX (c) EXIT.

Display proper strings to prompt the user while accepting the input and displaying the result.

**SOFTWARE REQUIRED:**

1. CPU: Intel I5 Processor
2. OS:- Windows XP (16 bit Execution ), Fedora 18 (32 & 64 bit Execution)
3. Editor: gedit, GNU Editor
4. Assembler: NASM (Netwide Assembler)
5. Linker:-LD, GNU Linker

**INPUT:**

1. Hexadecimal number

2. BCD Number

**OUTPUT:**

**1.** Conversion of hex to BCD number

2. Coversion of BCD to hex number

**THEORY:**

1. Hexadecimal to BCD conversion:

Conversion of a hexadecimal number can be carried out in different ways e.g. dividing number by 000Ah and displaying quotient in reverse way.

2. BCD to Hexadecimal number:

Conversion of BCD number to Hexadecimal number can be carried out by multiplying the BCD digit by its position value and the adding it in the final result.

**Special instructions used:**

**DIV**: Unsigned Divide. Result 🡪 Quotient in AL and Remainder in AH for 8-bit division and for 16-bit division Quotient in AX and Remainder in DX

**MUL**: Unsigned Multiply. For 8-bit operand multiplication result will be stored in AX and for 16-bit multiplication result is stored in DX:AX

Commands

* To assemble

**nasm –f elf 64 hello.nasm -o hello.o**

* To link

**ld –o hello hello.o**

* To execute -

**./hello**

**ALGORITHM:**

1. Start

2. Initialize data section

3 Display the Menu Message.

4 Accept the choice from the user.

5 If choice=1 then call Hex to bcd procedure. If choice=2 then call Bcd to Hex Procedure If choice=3 then call exit procedure

6. Hex to BCD Procedure:

a) Accept 4 Digit Hexadecimal Number.

b) Number=Number/10 & Number=Quotient

c) Push the remainder on the stack

d) If Number=0, Go to next step otherwise go to step b

e) Pop remainder , Convert into ascii & display untill all remainder are popped out

8. Bcd to hex Procedure:

a) RESULT=0

b) Accept BCD Digit

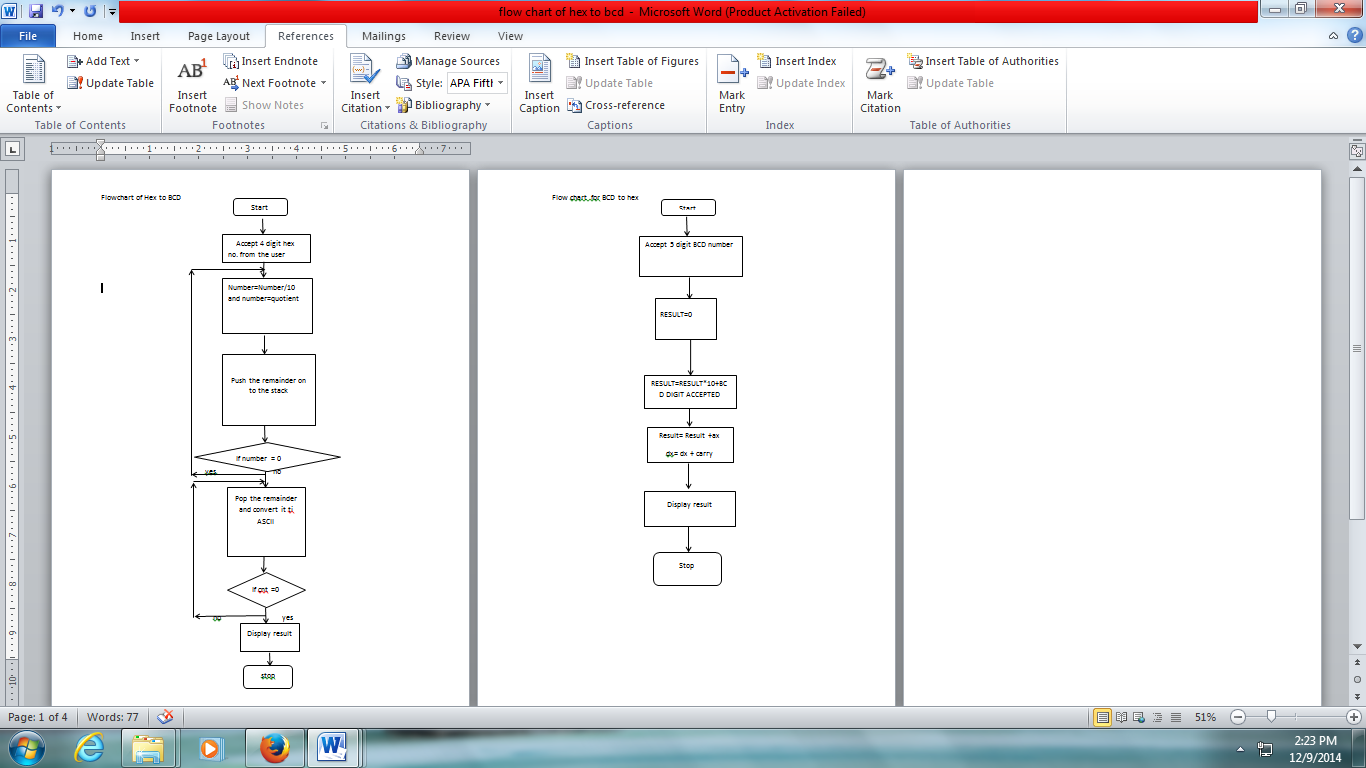
c) Check whether all BCD Digits are accepted. If YES then go to step e

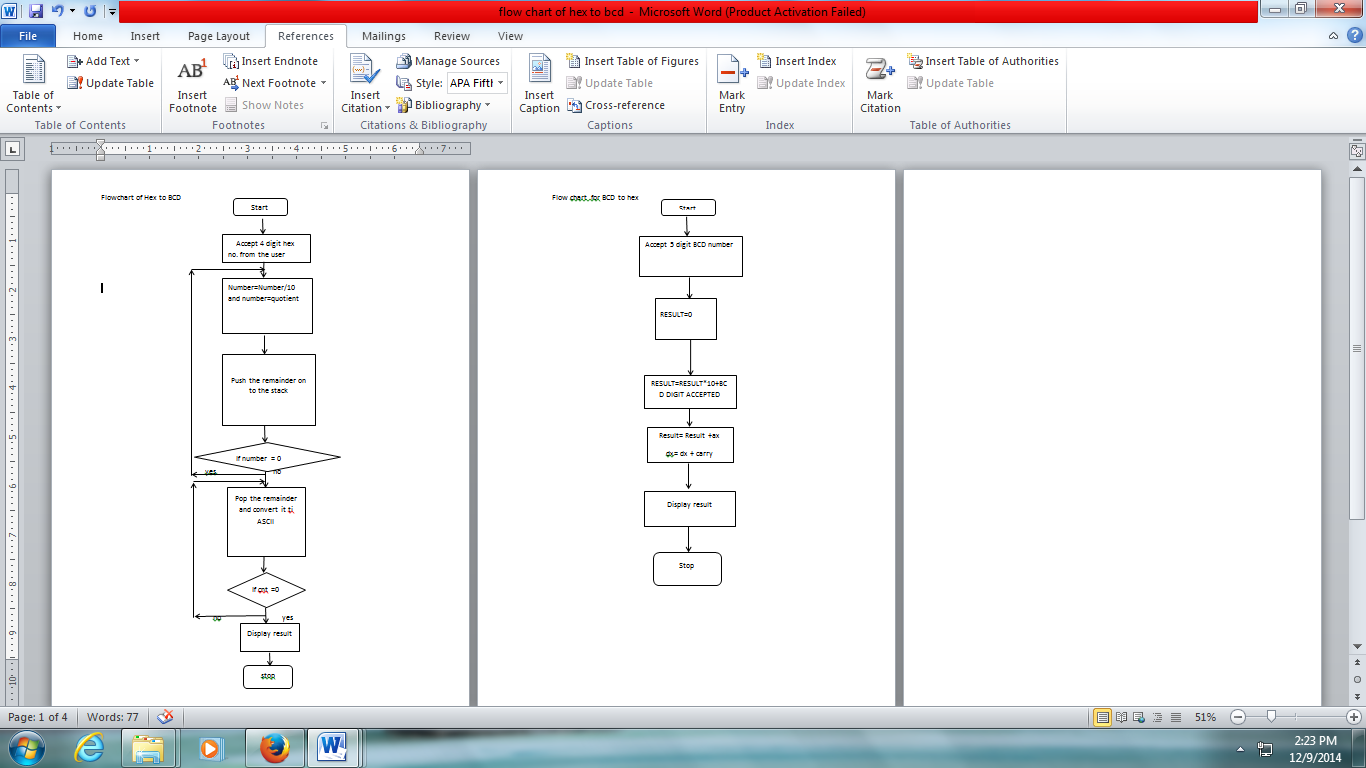
otherwise go to next step

d) RESULT= RESULT\*10 + BCD Digit accepted

e) Display RESULT

**FLOW CHART:**





**CONCLUSION:**

Hence we conclude that we can perform the Hex to BCD conversion & BCD to hex Conversion.

Output:

;[admin@localhost ~]$ vi conv.nasm

;[admin@localhost ~]$ nasm -f elf64 conv.nasm -o conv.o

;[admin@localhost ~]$ ld -o conv conv.o

;[admin@localhost ~]$ ./conv

;###### Menu for Code Conversion ######

;1: Hex to BCD

;2: BCD to Hex

;3: Exit

;Enter Choice:1

;Enter 4 digit hex number::FFFF

;BCD Equivalent::65535

;###### Menu for Code Conversion ######

;1: Hex to BCD

;2: BCD to Hex

;3: Exit

;Enter Choice:1

;Enter 4 digit hex number::00FF

;BCD Equivalent::255

;###### Menu for Code Conversion ######

;1: Hex to BCD

;2: BCD to Hex

;3: Exit

;Enter Choice:2

;Enter 5 digit BCD number::65535

;Hex Equivalent::0FFFF