**Exp No: 4** **Date:** 22/09/2020

Code Conversion **Name:** Swetha Saseendran

**Reg No:** 185001183

## Aim:

To write assembly language programs to perform the following code conversions:

1. BCD to Hexadecimal Code Conversion

2. Hexadecimal to BCD Code Conversion

# Programs:

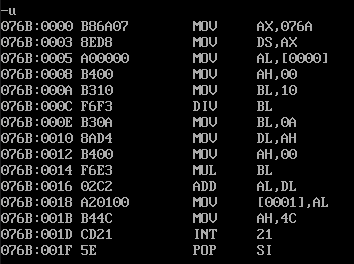
### (i) BCD to Hexadecimal Code Conversion

## Algorithm:

* Move the data segment to DS register through AX register.
* Move the extra segment to ES register through AX register.
* Clear AH register.
* Load the BCD value to AL.
* Load 10H to BL.
* Divide the value at AL by BL.
* Load the LSB at AH to DL.
* Multiple AL by 10 and add it to value at DL.
* Move the result at AL to HEX. 16.

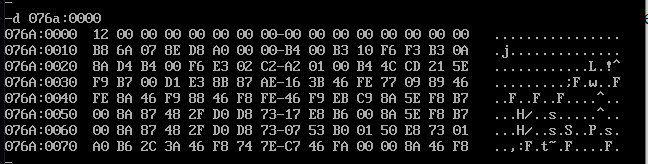
|  |  |
| --- | --- |
| **Program** | **Comments** |
| **ASSUME**  CS:CODE,DS:DATA | Using assume directive to declare data,extra and code segment |
| **DATA SEGMENT**  BCD DB 12H  HEX DB ?  **DATA ENDS** | Declaring and initialising variables in data segment |
| **CODE SEGMENT**  ORG 0100h | Set location for code segment at 0100h |
| **STARTt:**  MOV AX,data | Move the content of Data segment to AX register |
| MOV DS,AX | Move the content of AX register to DS register |
| MOV AL,BCD | Move the content of BCD to AL register |
| MOV AH,0H | Move the 0H to AH register.. |
| MOV BL,10H | Assign the offset of source to SI register. |
| DIV BL | Divide AX by BL. (Quotient in AL, Remainder in AH) |
| MOV BL,0AH | Transfer 10 to BL. |
| MOV DL,AH | Copy the contents of AH to DL. |
| MOV AH,0H | Clear AH register. |
| MUL BL | AX = AL \* BL (Multiply MSB by 10) |
| ADD AL,DL | AL = AL + DL (Add LSB to the hex result) |
| MOV AH,4CH | Moves the hexadecimal value 4c to ah. |
| INT 21H | When Software interrupt 21 is called with AH=4C, then current process terminates |
| **CODE ENDS** | Ending the code segment |
| **END START** | Ending start segment |

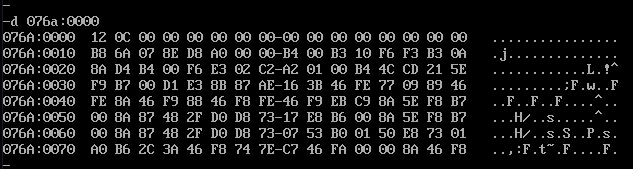
## Unassembled Code:



## Snapshot of sample input and output:

**INPUT:** Packed BCD No- 12



**OUTPUT:** Hexadecimal equivalent- 0C  


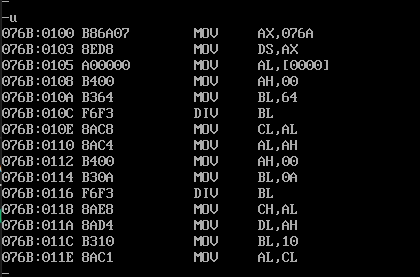
### (ii) Hexadecimal to BCD Code Conversion

## Algorithm:

* Move the data segment to DS register through AX register.
* Move the extra segment to ES register through AX register.
* Load 100(64H) to BL.
* Divide the value at AX by BL.
* Move the MSB at AL to CL.
* Move the LSBs at AH to AL.
* Clear AH register.
* Load the 10(0AH) to BL.
* Dive the value at AX by BL.
* Move the second bit of BCD to CH.
* Move the LSB of BCD to DL.
* Apply [CL]\*100 + [CH]\*10 + [DL] and store the result at AX.
* Move the result at AX to BCD.

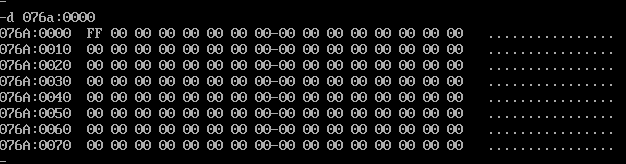
|  |  |
| --- | --- |
| **Program** | **Comments** |
| **ASSUME**  CS:CODE,DS:DATA, | Using assume directive to declare data, and code segment |
| **DATA SEGMENT**  HEX DB 0FFH  BCD DW ?  **DATA ENDS** | Declaring and initialising variables in data segment |
| **CODE SEGMENT**  ORG 0100H | Set location for code segment at 0100h |
| **START:**  MOV AX,DATA | Move the content of Data segment to AX register |
| MOV DS,AX | Move the content of AX register to DS register |
| MOV AL, HEX | Transfer the given BCD byte to AL. |
| MOV AH, 0H | Clear AH register. mov bl, 64h Transfer 100 to BL. |
| MOV BL, 64H | Transfer 100 to BL. |
| DIV BL | Divide AX by BL. (Quotient in AL, Remainder in AH) |
| MOV   CL, AL | Transfer the quotient to CL register. (MSB of BCD) |
| MOV AL, AH | Transfer the remainder to AL register. |
| MOV AH, 0H | Clear AH register. |
| MOV BL, 0AH | Transfer 10 to BL. |
| DIV     BL | Divide AX by BL. |
| MOV CH, AL | Transfer the quotient to CH register. (2nd MSB of BCD) |
| MOV DL, AH | Transfer the remainder to DL register. (LSB of BCD) |
| MOV   BL, 10H | Transfer 16 to BL. |
| MOV AL, CL | Transfer the MSB of BCD to AL register. |
| MUL BL | AX = AL \* BL (Multiply MSB by 10) |
| ADD AL, CH | AL = AL + CH (Add 2 nd MSB to the BCD result) |
| MUL BL | AX = AL \* BL (MSB \* 100 + 2nd MSB \* 10) |
| ADD AL, DL | AL = AL + DL (MSB \* 100 + 2nd MSB \* 10 + LSB) |
| MOV BCD, AX | Store the value in AX as the final BCD converted code |
| MOV AH,4CH | Moves the hexadecimal value 4c to ah. |
| INT 21H | When Software interrupt 21 is called with AH=4C, then current process terminates |
| **CODE ENDS** | Ending the code segment |
| **END START** | Ending start segment |
|  |  |

## Unassembled Code:

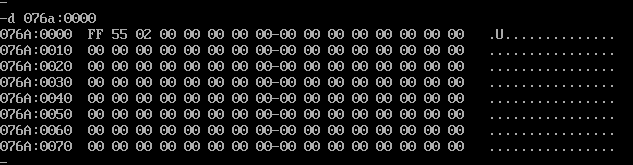


## Snapshot of sample input and output:

**INPUT:** Hexadecimal value-FF



**OUTPUT:** BCD output in packed form- location1 = 02, location2 = 55



## Result:

The assembly level programs were written to perform the above specified code conversions and the output was verified.