**Exp No: 7** **Date:** 08/10/2020

BCD ADDITION AND SUBTRACTION **Name:** Swetha Saseendran

**Reg No:** 185001183

## Aim:

To program and execute 8 bit BCD Addition and Subtraction using DOS-BOX.

### (i) BCD Addition

# Programs:

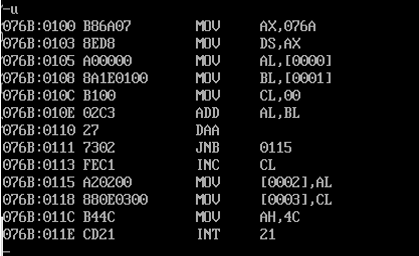
### 

## Algorithm:

* Program is set to run from any specified memory position.
* Load data from opr1 to register AL (first operand)
* Load data from opr2 to register BL (second operand)
* Initialize carry to 0.
* Add these two numbers (contents of register AL and register BL)
* Decimal adjust after addition
* Jump to here label if there is no carry.
* Increment carry. Store additional values to result.
* Store additional values to result.
* Terminate the program.

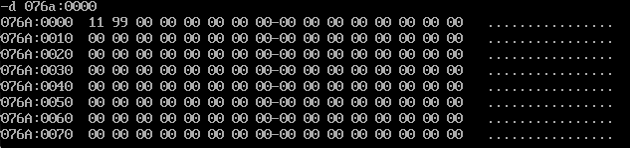
|  |  |
| --- | --- |
| **Program** | **Comments** |
| **assume** cs:code, ds:data | Using assume directive to declare data, extra and code segment |
| **data segment** | Declaring and initialising variables in data segment |
| opr1 db 11h |  |
| opr2 db 99h |  |
| result db 00h |  |
| carry db 00h |  |
| **data ends** | Data segment ends |
| **code segment** | Start of code segment |
| org 0100h | Originating address ←0100 |
| **start:** |  |
| mov ax, data | AX ←data |
| mov ds, ax | DS←ax |
| mov al, opr1 | Move opr1 to AL register |
| mov bl, opr2 | Move opr2 to BL register |
| mov cl,00h | CL←00h |
| add al, bl | AL=AL+BL |
| daa | Decimal adjust after addition |
| jnc here | Jump if no carry to here |
| inc cl | Increment CL |
| **here:** |  |
| mov result, al | Result ←AL |
| mov carry, cl | Move opr1 to AL register |
| Mov ah, 4ch | AH←4Ch |
| int 21h | When Software interrupt 21 is called with AH=4C, then current process terminates. (i.e., These two instructions are used for the termination of the process). |
| **code ends** | Code segment ends |
| **end start** | End of start label |

## Unassembled Code:



## Snapshot of sample input and output:

**INPUT:**



**OUTPUT:**



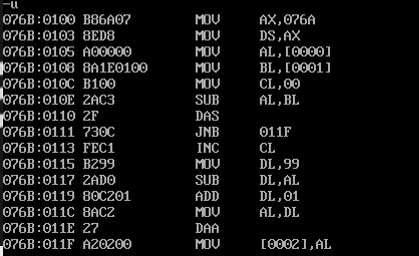
### (ii) BCD Subtraction

## Algorithm:

* Program is set to run from any specified memory position.
* Load data from opr1 to register AL (first operand)
* Load data from opr2 to register BL (second operand)
* Initialize carry to 0.
* Subtract these two numbers (contents of register AL and register BL)
* Decimal adjust after subtraction
* Jump to here label if there is no carry.
* Increment carry. Store additional values to result.
* Store additional values to result.
* Terminate the program

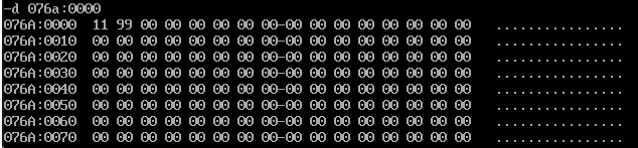
|  |  |
| --- | --- |
| **Program** | **Comments** |
| **assume** cs:code,ds:data | Using assume directive to declare data, extra and code segment |
| **data segment** | Declaring and initialising variables in data segment |
| opr1 db 11h |  |
| opr2 db 99h |  |
| result db 00h |  |
| carry db 00h |  |
| **data ends** | Data segment ends |
| **code segment** | Start of code segment |
| org 0100h | Originating address ←0100 |
| **start:** |  |
| mov ax,data | AX ←data |
| mov ds,ax | DS←ax |
| mov al, opr1 | Move opr1 to AL register |
| mov bl, opr2 | Move opr2 to BL register |
| mov cl, 00h | CL←00h |
| sub al, bl | AL=AL-BL |
| das | Decimal adjust after subtraction |
| jnc here | Jump if no carry to here |
| inc cl | Increment CL |
| mov dl, 99h | DL←99h |
| sub dl, al | DL=DL-AL |
| add dl, 01h | DL=DL+01h |
| mov al, dl | Move value in DL to AL |
| daa |  |
| **here:** |  |
| mov result, al | Result ←AL |
| mov carry, cl | carry←CL |
| mov ah, 4ch | AH←4Ch |
| int 21h | When Software interrupt 21 is called with AH=4C, then current process terminates. (i.e., These two instructions are used for the termination of the process). |
| **code ends** | Code segment ends |
| **end start** | End of start label |

## Unassembled Code:

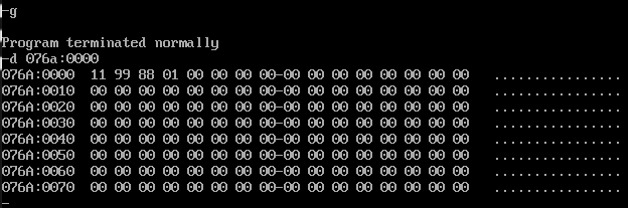


## Snapshot of sample input and output:

**INPUT:**



**OUTPUT:**



## Result:

8-bit BCD Addition and Subtraction have been programmed and executed in 8086 microprocessor using DOS-BOX.