**SSN College of Engineering**

**Department of Computer Science and Engineering**

**UCS1512 – Microprocessors Lab**

**EX:07 – BCD Addition and Subtraction**

**Exp No: 07 Name : Kshitij Sharma**

**Date: 08/10/2020 Reg No: 185001080**

# Aim:

To write and execute 8086 programs for BCD addition and subtraction.

# Procedure:

* Mount masm folder to a drive on DOSBOX.
* Navigate to mounted drive using ‘dir’ .
* Save 8086 program with the extension **‘.asm’** in the same folder using the command **‘edit’**.
* Assemble the **.asm** file using the command **‘masm filename.asm’**.
* Link the assmebled **.obj** file using the command **‘link filename.obj’**.
* Debug the executable file **.exe** with the **‘debug filename.exe’** command.
  1. **U:** To view the un-assembled code.
  2. **D:** Used as ‘D segment:offset’ to see the content of memory locations starting from segment:offset address.
  3. **E:** To change the values in memory.
  4. **G:** Execute the program using command.
  5. **Q** exits from the debug session.

# Algorithm:

## BCD Addition

* + First operand is stored in opr1 & second operand is stored in opr2.

Move the data segment address to the AX register and then move it to the DS register.

∗

* + Move opr1 into AL using MOV.
  + Move opr2 into BL using MOV.
  + Add AL, BL using ADD.
  + Decimal adjust after addition using DAA.
  + IF there is a carry, move 01 into carry variable.
  + ELSE move 00 into carry variable.
  + Move value in AL into result using MOV.

## BCD Subtraction

* + First operand is stored in opr1 & second operand is stored in opr2.

Move the data segment address to the AX register and then move it to the DS register.

∗

* + Move opr1 into AL using MOV.
  + Move opr2 into BL using MOV.
  + Subtract BL from AL using SUB.
  + Decimal adjust after subtraction using DAS.

IF borrow(carry) flag is NOT set, move 00 into sign variable, and AL into difference using MOV.

∗

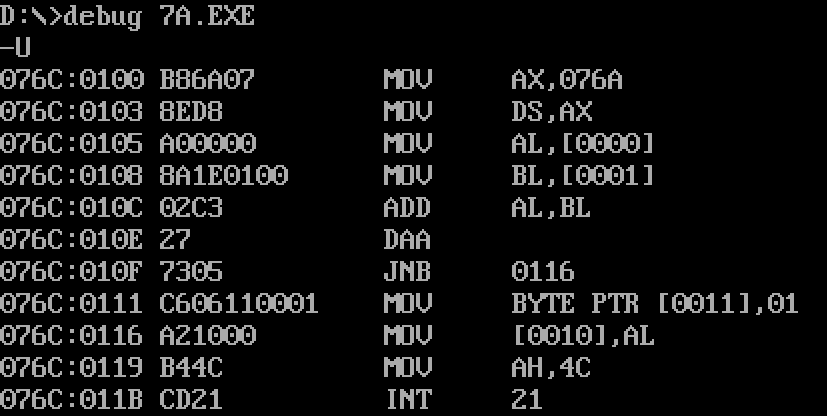
* + ELSE,
    - Move 01h into sign variable.
    - Mov 99 into BL register.
    - Subtract AL from BL using SUB.
    - Increment BL by 1.
    - BL now contains 10s complement of AL.
    - Move BL into difference.

# BCD Addition

## Program:

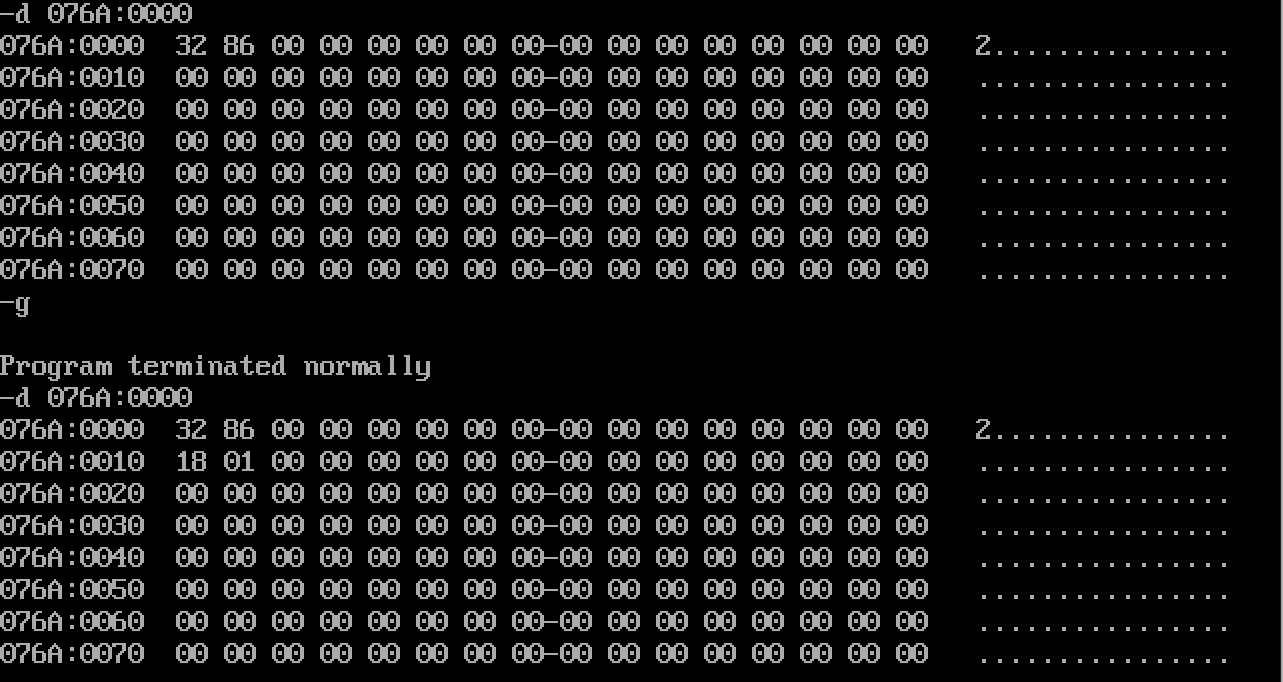
|  |  |
| --- | --- |
| **Program** | **Comments** |
| **start**: MOV AX,data | Move data segment address contents to AX register |
| MOV ds,AX | Move data in AX register to DS register |
| MOV AL, opr1 | Move first operand into AL |
| mov BL ,opr2 | Move second operand into BL |
| ADD AL, BL | Add AL and BL |
| DAA | Decimal Adjust after addition |
| JNC **skip** | IF there is no carry skip |
| MOV carry, 01h | Load 01 into carry |
| **skip:** MOV sum,AL | Store sum obtained from AL register |
| MOV ah,4ch |  |
| INT 21h | Request interrupt routine |

**Unassembled Code:**



**Input and Output:**

Figure 1: **Input:** opr 1 = 32 & opr 2 = 86 **Output:** carry: 01, sum: 18

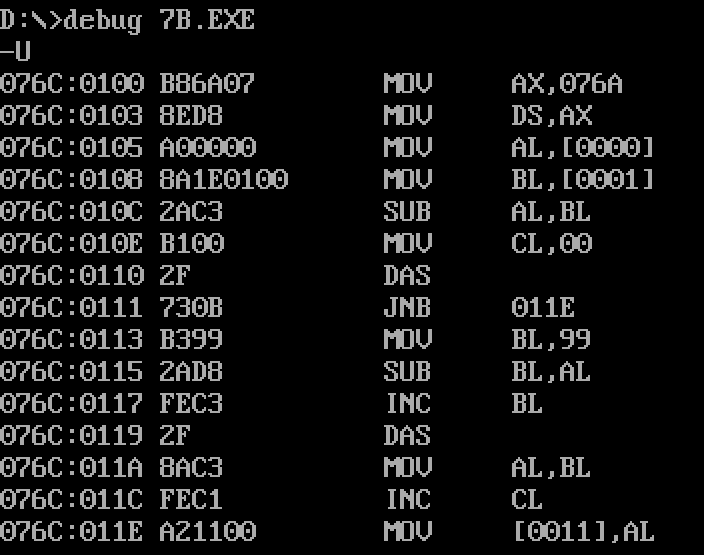


# BCD Subtraction

## Program:

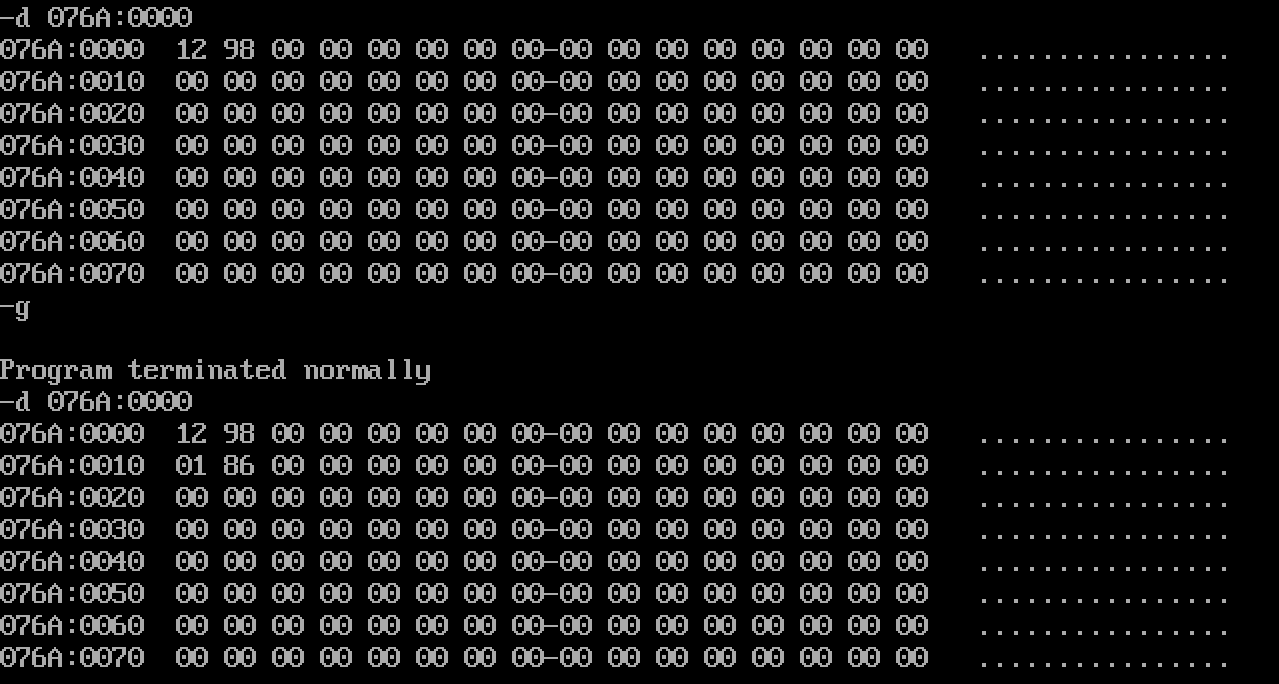
|  |  |
| --- | --- |
| **Program** | **Comments** |
| **start**: MOV AX,data | Move data segment address contents to AX register |
| MOV ds,AX | Move data in AX register to DS register |
| MOV AL, opr1 | Move first operand into AL |
| MOV BL ,opr2 | Move second operand into BL |
| SUB AL, BL | Subtract BL from AL |
| MOV CL, 00h | Initialise CL to 0 |
| DAS | Decimal adjust after subtraction |
| JNC **stop** | If there is no borrow(carry) |
| MOV BL, 99h | Load 99 into BL register |
| SUB BL, AL | Subtract AL from 99 |
| INC BL | BL now contains 10s complement of AL |
| DAS | Decimal Adjust after subtraction |
| mov AL, BL | Move 10s complement into AL |
| MOV CL, 01h | Set sign flag |
| **stop:** MOV difference, AL | Store difference from AL |
| MOV sign, CL | Store sign flag from CL |
| MOV ah,4ch |  |
| INT 21h | Request interrupt routine |

**Unassembled Code:**



**Input and Output:**

Figure 2: **Input:** opr 1 = 12 & opr 2 = 98 **Output:** sign = 01, difference = 86



# Result:

8086 ASL programs for BCD addition and subtraction have been executed successfully using MS – DOSBox.