**SSN College of Engineering Department of Computer Science and Engineering**

**III year - UCS1512 – Microprocessors Lab**

**BCD Addition and Subtraction**

**Exp No:** 07

**Name:** Rahul Ram M

**Register Number:** 185001121

**Date:** 06/10/2020

**a) BCD Addition:**

**Aim:**

Design 8086 program for BCD addition(B bit).

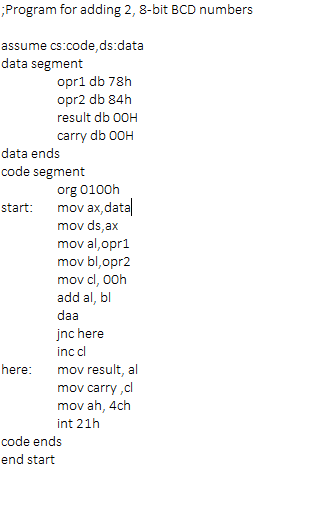
**Procedure for executing MASM:**

1. Run Dosbox and mount your masm folder to a drive in dosbox.
2. Goto the mounted drive.
3. Save the 8086 program with extension .asm in the same folder using command “edit”
4. After creating the file, assemble it using the command “masm filename.asm”
5. Link the file using the command “link filename.obj;”
6. Use debug command with filename.exe to execute and analyse the memory contents, “debug filename.exe”.
7. In debug, command “u” will display the unassembled code.
8. Use command “d segment:offset” to see the content of memory locations starting from segment:offset address.
9. To change the value in memory, use the command “e segment:offset”
10. Verify the memory contents to ensure the updates (using command “d”).
11. . Execute using the command “g” and check the outputs.
12. “q” to exit from debug and “exit” to exit from command prompt and to close the Dosbox.

**Algorithm:**

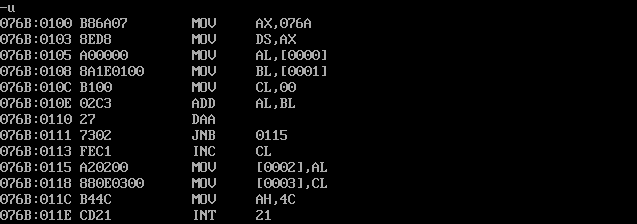
1. START: Move the starting address of data segment to AX register and move the data from AX register to DS register.
2. Move the data of first operator to AL register.
3. Move the data of second operator to BL register.
4. Load CL register with 00H.
5. Then add AL and BL using ADD AL, BL and the value will be stored in AL register.
6. Using DAA (Decimal Adjust after Addition) to represent the value in 8-bit packed BCD code.
7. If the carry flag is reset, jump to HERE.
8. Increment the CL register value.
9. HERE: Now store the data of AL register in result and CL register in carry MOV instruction.
10. Move the hexadecimal value 4C into AH register. INT 21H means invoke the interrupt identified by the hexadecimal number 21. In MS-DOS, invoking interrupt 21h while AH = 4Ch causes the current process to terminate and uses the value of register AL as the exit code of the process.

**Program:**



|  |  |  |
| --- | --- | --- |
|  | **Program** | **Comments** |
| START: | ORG 0100H | Memory instruction starts from 0100H. |
| MOV AX, DATA  MOV DS, AX | Transferring the data from DATA to AX register and  from AX register to DS register. |
| MOV AL, OPR1 | Transfer the data from opr1 to AL register. |
| MOV BL, OPR2 | Transfer the data from opr2 to BL register. |
| MOV CL, OOH | CL <- 00h. |
| ADD AL, BL | AL <- AL + BL |
| DAA | Decimal Adjust AL register. |
| JNC HERE | Jump if no carry to HERE |
| INC CL | Increment the value in the CL register. |
| HERE: | MOV RESULT, AL | RESULT <- AL |
| MOV CARRY, CL | CARRY <- CL |
| MOV AH, 4CH  INT 21H | Terminates the program. |

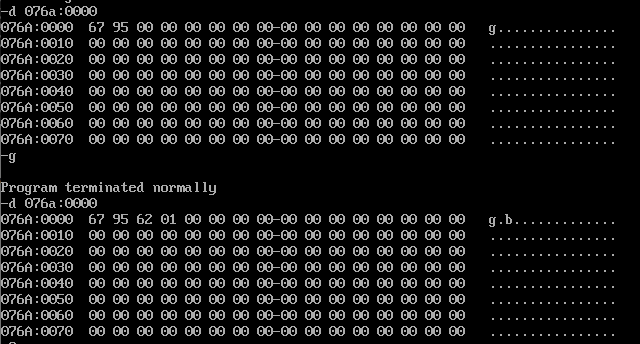
**Snapshot of sample input and output:**



**Without carry (op1 – 11, op2 – 23):**



**With carry (op1 – 67, op2 – 95):**



**Result:**

Thus the 8086 program for BCD addition (8-bit) is executed successfully in DOS-BOX.

**b) BCD Subtraction:**

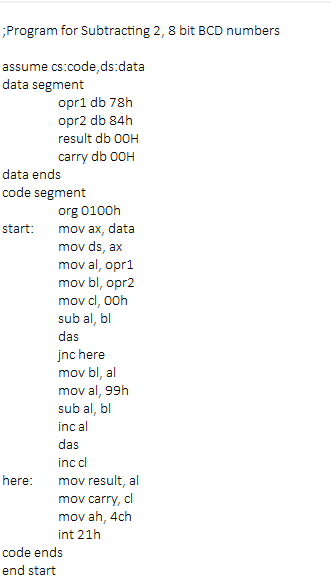
**Aim:**

Design 8086 program for BCD subtraction (8-bit).

**Algorithm:**

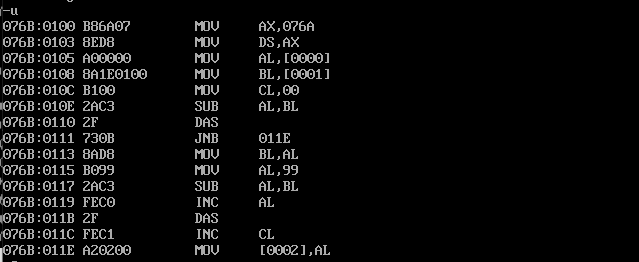
1. START: Move the starting address of data segment to AX register and move the data from AX register to DS register.
2. Move the data of first operator to AL register.
3. Move the data of second operator to BL register.
4. Load CL register with 00H.
5. Then sub AL and BL using SUB AL, BL and the result will be stored in AL register.
6. Apply DAS instruction (Decimal Adjust after Subtraction to convert hexadecimal to BCD).
7. If the carry flag is reset, jump to HERE.
8. Move the data in AL register to BL register.
9. Load AL register with 99H.
10. Subtract AL and BL register and result is stored in AL register
11. Increment AL register.
12. Apply DAS instruction (Decimal Adjust after Subtraction to convert hexadecimal to BCD).
13. The above 5 steps (8, 9, 10, 11 and 12) is used to obtain 10’s compliment of the result.
14. Increment the CL register.
15. HERE: Now store the data of AL register in result and CL register in carry using MOV instruction.
16. Move the hexadecimal value 4C into AH register. INT 21H means invoke the interrupt identified by the hexadecimal number 21. In MS-DOS, invoking interrupt 21h while AH = 4Ch causes the current process to terminate and uses the value of register AL as the exit code of the process.

**Program:**

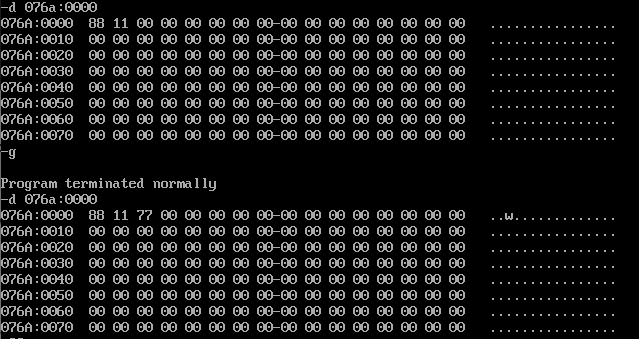


|  |  |  |
| --- | --- | --- |
|  | **Program** | **Comments** |
| START: | ORG 0200H | Memory instruction starts from 0200H. |
| MOV AX, DATA  MOV DS, AX | Transferring the data from DATA to AX register and  from AX register to DS register. |
| MOV AL, OPR1 | Transfer the data from opr1 to AL register. |
| MOV BL, OPR2 | Transfer the data from opr2 to BL register. |
| MOV CL, OOH | CL <- 00H. |
| SUB AL, BL | AL <- AL - BL |
| DAS | Decimal Adjust on AL register. |
| JNC HERE | Jump if no carry to HERE |
| MOV BL, AL | BL <- AL |
| MOV AL, 99H | AL <- 99H |
| SUB AL, BL | AL <- AL - BL |
| INC AL | Increment AL register. |
| DAS | Decimal Adjust on AL register. |
| INC CH | Increment the value in the CL register. |
| HERE: | MOV RESULT, AH | Move the data from AH register to result. |
| MOV CARRY, CH | Move the data from CH register to carry. |
| MOV AH, 4CH  INT 21H | Terminates the program. |

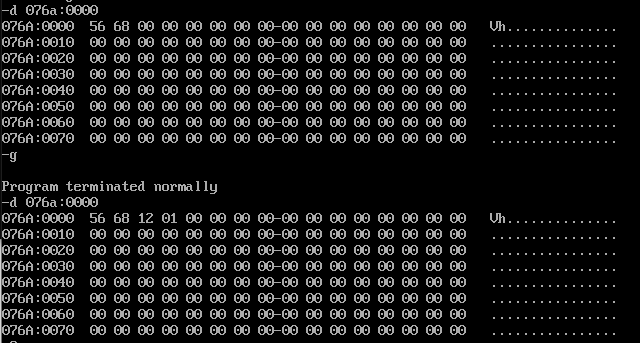
**Snapshot of sample input and output:**



**Without carry (op1 – 88, op2 – 11):**



**With carry (op1 – 56, op2 – 68):**



**Result:**

Thus the 8086 program for BCD subtraction (8-bit) is executed successfully in

DOS-BOX.