16 BIT ARITHMETIC OPERATIONS

## Exp No: 02 Name : Kshitij Sharma

Date: 31/08/2020 Reg No: 185001080

**Aim:-**

To write Programs for 16 bit arithmetic operations : addition , subtraction , Multiplication and Division.

# Algorithm:-

1. **16 bit add :**
   * Move the data segment to the AX register. Then move it to the DS register.
   * Move the first operator to AX register.
   * Move the second operator to the BX register.
   * Initialize the CH register with 00H
   * Then add the both register value with add ax,bx.
   * Move the carry bit value to the CH register.
   * If carry value is 1, increase by 1 by using “inc ch”.
   * If carry value is 0, jump to next statement.
   * The result will be stored in ax register.
   * The carry value will be stored in ch register.
   * We int 21h= ah the program will be terminated.

# 16 bit sub:

* + Move the data segment to the AX register. Then move it to the DS register.
  + Move the first operator to AX register.
  + Move the second operator to the BX register.
  + Initialize the CH register with 00H
  + Then subtract both register value with sub ax,bx.
  + If carry value is 1, increase CH by 1 by using “inc ch” and find 2’s comp of AX.
  + If carry value is o, jump to next statement.
  + The result will be stored in ax register.
  + The sign value will be stored in ch register.
  + We int 21h= ah the program will be terminated.

# 16 bit mul:

* + Move the data segment to the AX register. Then move it to the DS register.
  + Move the first operator to AX register.
  + Move the second operator to the BX register.
  + Then mul the both register value with mul bx.
  + The result will be stored in AX register.
  + The high order will be stored in DX register
  + We int 21h= ah the program will be terminated.

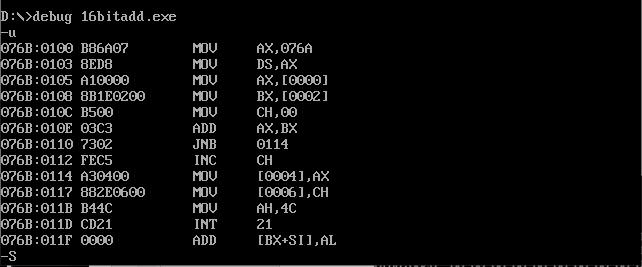
# 16 bit div:

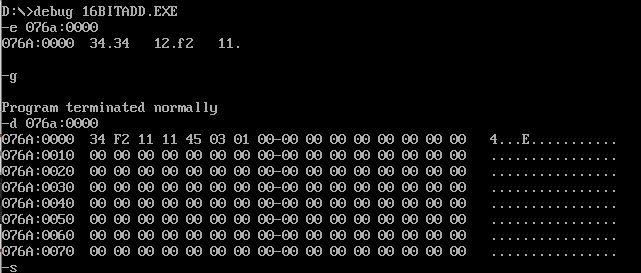
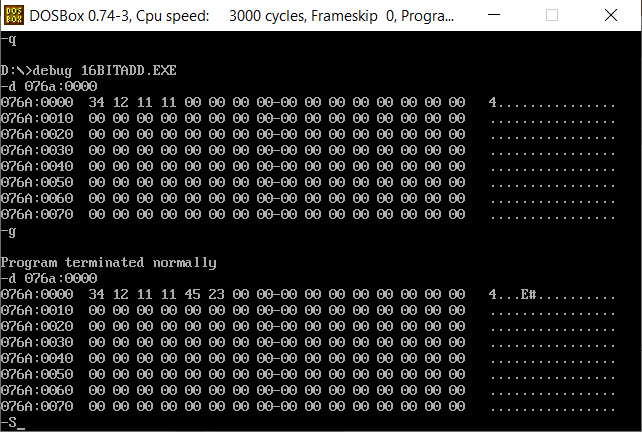
* + - Move the data segment to the AX register. Then move it to the DS register.
    - Load 0000H in the DX register.
    - Move the first operator to AX register.
    - Move the second operator to the BX register.
    - Then div the both register value with div bx
    - The quotient will be stored in AX register.
    - The remainder will be stored in DX register.
    - We int 21h= ah the program will be terminated.

# Program:-

1. **16 bit addition:**

|  |  |
| --- | --- |
| **PROGRAM** | **COMMENTS** |
| MOV AX,DATA | Transfer the data to ax register |
| MOV DS,AX | Transfer the the value of ax register to data segment register |
| MOV AX,OPR1 | To transfer the value from opr1 to ax register |
| MOV BX,OPR2 | To transfer the value from opr2 to bx register |
| MOV CH,00H | Initialize 00 value in ch register |
| ADD AX,BX | ax=ax+bx |
| JNC HERE | Jump to Label Here if there is no carry |
| INC CH | Increment value of ch by one i.e, ch=ch+1 |
| HERE: MOV RESULT,AX | The value of ax will be transfer to result |
| MOV CARRY,CH | The value of the carry bit we be seen in carry |
| MOV AH,4CH | Loads 4ch value in ah register |
| INT 21H | Interrupts the Program |

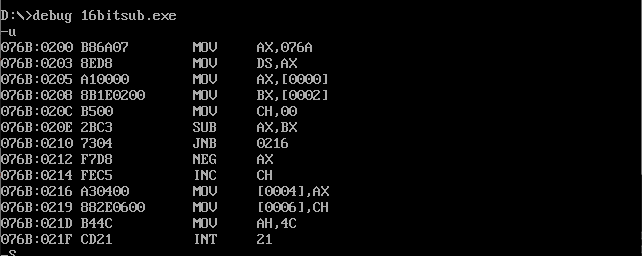
**Snapshot:-**

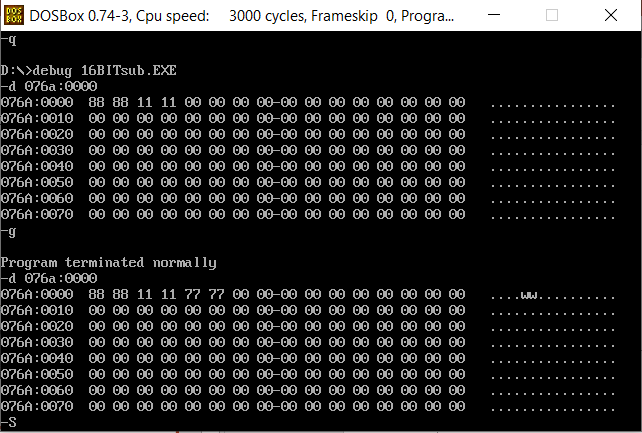


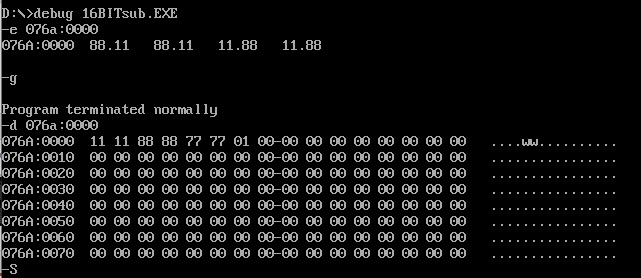
1. **16 bit Subtraction:**

|  |  |
| --- | --- |
| **PROGRAM** | **COMMENTS** |
| MOV AX,DATA | Transfer the data to ax register |
| MOV DS,AX | Transfer the the value of ax register to data segment register |
| MOV AX,OPR1 | To transfer the value from opr1 to ax register |
| MOV BX,OPR2 | To transfer the value from opr2 to bx register |
| MOV CH,00H | Initialize 00 value in ch register |
| SUB AX,BX | ax=ax-bx |
| JNC HERE | Jump to Label Here if there is no carry |
| NEG AX | Stores the 2’s compliment of ax in ax |
| INC CH | Increment value of ch by one i.e, ch=ch+1 |
| HERE: MOV RESULT,AX | The value of ax will be transfer to result |
| MOV CARRY,CH | The value of the carry bit we be seen in carry |
| MOV AH,4CH | Loads 4ch value in ah register |
| INT 21H | Interrupts the Program |

**Snapshot:-**



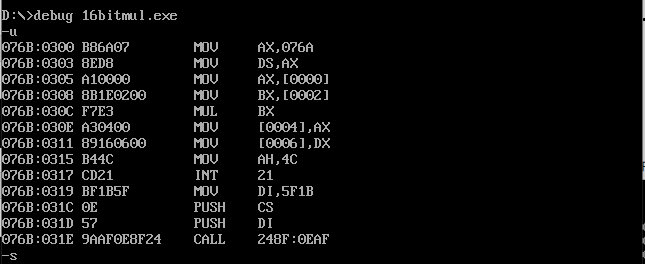


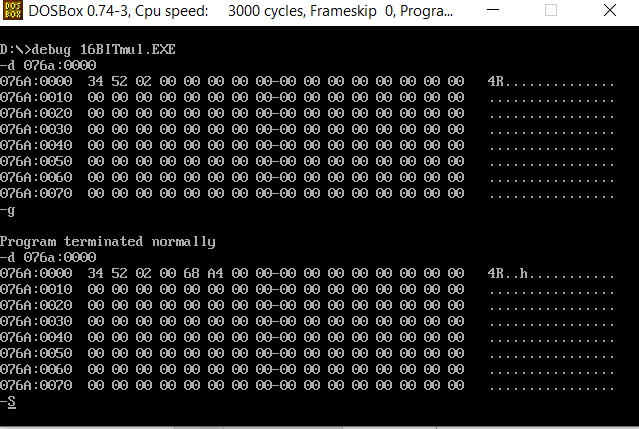


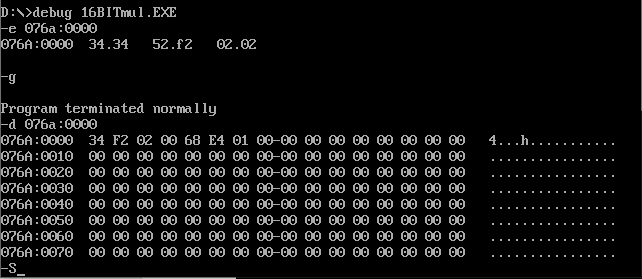
1. **16 bit Multipaction:**

|  |  |
| --- | --- |
| **PROGRAM** | **COMMENTS** |
| MOV AX,DATA | Transfer the data to ax register |
| MOV DS,AX | Transfer the the value of ax register to data segment register |
| MOV AX,OPR1 | To transfer the value from opr1 to ax register |
| MOV BX,OPR2 | To transfer the value from opr2 to bx register |
| MUL BX | ax = ax \* bx |
| MOV RESULT1,AX | The value of ax will be transfer to result |
| MOV RESULT2,DX | The value of dx will be transfer to result |
| MOV AH,4CH | Loads 4ch value in ah register |
| INT 21H | Interrupts the Program |

**Snapshot:-**



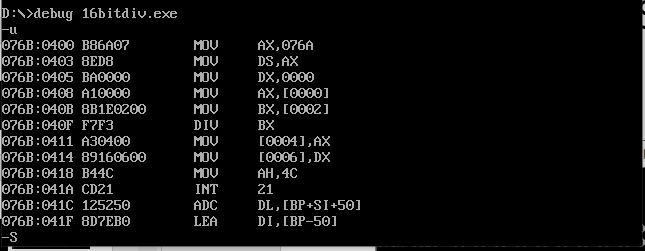


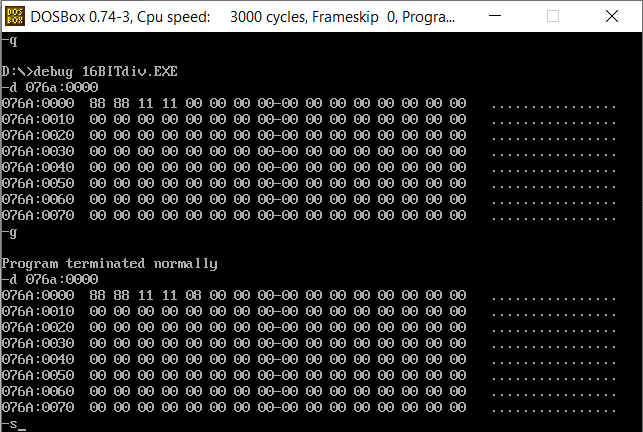
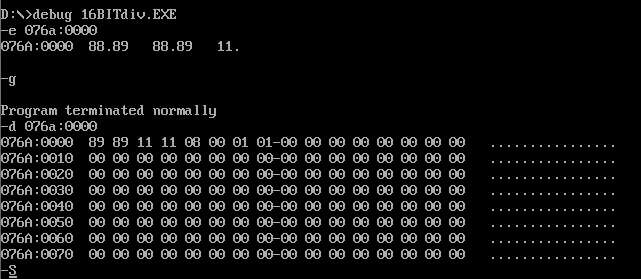
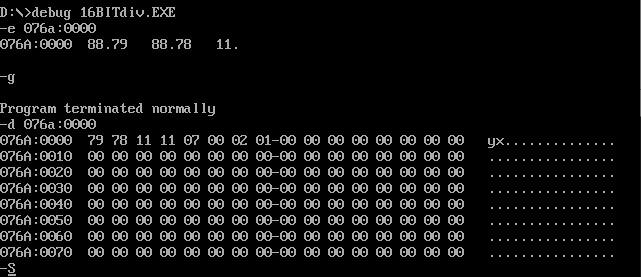


1. **16 bit Division:**

|  |  |
| --- | --- |
| **PROGRAM** | **COMMENTS** |
| MOV AX,DATA | Transfer the data to ax register |
| MOV DS,AX | Transfer the the value of ax register to data segment register |
| MOV DX,00H | Loads value 0000 in DX register. |
| MOV AX,OPR1 | To transfer the value from opr1 to ax register |
| MOV BX,OPR2 | To transfer the value from opr2 to bx register |
| DIV BX | ax=ax/bx & dx = ax%bx |
| MOV RESULT,AX | The value of ax will be transferred to result |
| MOV REM,DX | The value of dx will be transferred to Rem |
| MOV AH,4CH | Loads 4ch value in ah register |
| INT 21H | Interrupts the Program |

**Snapshot:-**





**Result:-**

## Thus the above program executed successfully in the assembler by the dosbox environment.