**SSN College of Engineering**

**Department of Computer Science and Engineering**

**UCS1512 – Microprocessors Lab**

EX:06 – Matrix Operations

**Exp No: 05 Name : Kshitij Sharma**

**Date: 01/09/2020 Reg No: 185001080**

**Aim:-**

To write and execute 8086 ASL programs for performing matrix addition and subtraction.

# Algorithm:-

1. **Matrix Addition:**
   * Move the data segment to the AX register. Then move it to the DS register.
   * Check if row1 and row2 are equal if not exit else continue
   * Check if col1 and col2 are equal if not exit else continue.
   * Load offset of mat1,mat2,result in SI,DI,BX resp.
   * Add corresponding elements from both matrices and store it in result matrix.

# Matrix Subtraction:

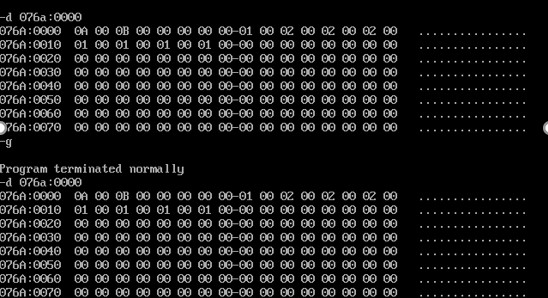
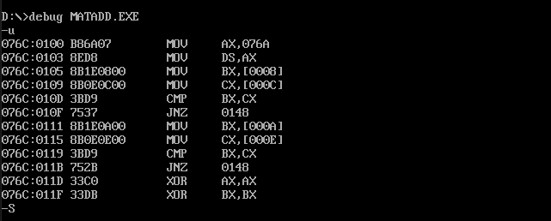
* + Move the data segment to the AX register. Then move it to the DS register.
  + Check if row1 and row2 are equal if not exit else continue
  + Check if col1 and col2 are equal if not exit else continue.
  + Load offset of mat1,mat2,result in SI,DI,BX resp.
  + Subtract corresponding elements from both matrices and store it in result matrix.

# Program:-

1. **Matrix 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 BX,ROW1 |  |
| MOV CX,ROW2 |  |
| CMP BX,CX | Compares BX and CX register |
| JNZ LAST | Jumps to label last if BX not equal to CX |
|  |  |
| MOV BX,COL1 |  |
| MOV CX,COL2 |  |
| CMP BX,CX | Compares BX and CX register |
| JNZ LAST | Jumps to label last if BX not equal to CX |
|  |  |
| XOR AX,AX | AX = AX xor AX , Clears AX register |
| XOR BX,BX | BX = BX xor BX , Clears BX register |
| XOR DX,DX | DX = DX xor DX , Clears DX register |
|  |  |
| MOV AX,ROW1 |  |
| MOV BX,COL1 |  |
| MUL BX |  |
| MOV CX,AX |  |
| INC CX |  |
|  |  |
| MOV SI,OFFSET MAT1 |  |
| MOV DI,OFFSET MAT2 |  |
| MOV BX,OFFSET RESULT |  |
| HERE: |  |
| MOV AX,[SI] |  |
| MOV DX.[DI] |  |
| ADD AX,DX | AX = AX + DX |
| MOV [BX],AX |  |
| INC SI | SI is incremented twice since each increment corresponds to a |
| INC SI | shift to the next 8 bit value |
| INC DI | DI is incremented twice since each increment corresponds to a |
| INC DI | shift to the next 8 bit value |
| INC BX | BX is incremented twice since each increment corresponds to a |
| INC BX | shift to the next 8 bit value |
| LOOP HERE |  |
| LAST: |  |
| MOV AH,4CH | Loads 4ch value in ah register |
| INT 21H | Interrupts the Program |

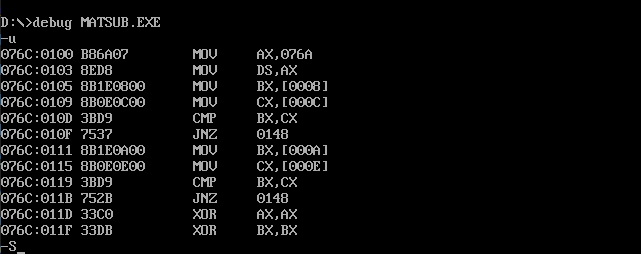
**Snapshot:-**



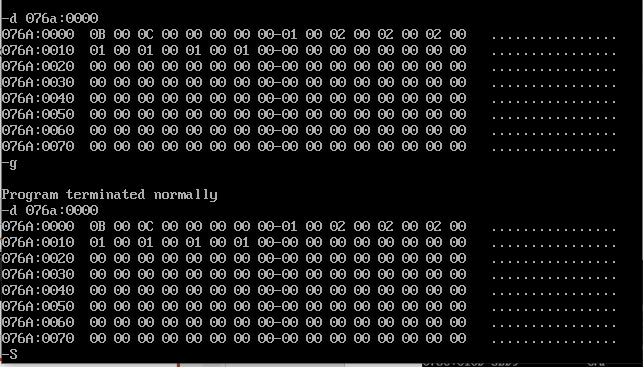
1. **Matrix Subtraction :**

|  |  |
| --- | --- |
| **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 BX,ROW1 |  |
| MOV CX,ROW2 |  |
| CMP BX,CX | Compares BX and CX register |
| JNZ LAST | Jumps to label last if BX not equal to CX |
|  |  |
| MOV BX,COL1 |  |
| MOV CX,COL2 |  |
| CMP BX,CX | Compares BX and CX register |
| JNZ LAST | Jumps to label last if BX not equal to CX |
|  |  |
| XOR AX,AX | AX = AX xor AX , Clears AX register |
| XOR BX,BX | BX = BX xor BX , Clears BX register |
| XOR DX,DX | DX = DX xor DX , Clears DX register |
|  |  |
| MOV AX,ROW1 |  |
| MOV BX,COL1 |  |
| MUL BX |  |
| MOV CX,AX |  |
| INC CX |  |
|  |  |
| MOV SI,OFFSET MAT1 |  |
| MOV DI,OFFSET MAT2 |  |
| MOV BX,OFFSET RESULT |  |
| HERE: |  |
| MOV AX,[SI] |  |
| MOV DX.[DI] |  |
| SUB AX,DX | AX = AX - DX |
| MOV [BX],AX |  |
| INC SI | SI is incremented twice since each increment corresponds to a |
| INC SI | shift to the next 8 bit value |
| INC DI | DI is incremented twice since each increment corresponds to a |
| INC DI | shift to the next 8 bit value |
| INC BX | BX is incremented twice since each increment corresponds to a |
| INC BX | shift to the next 8 bit value |
| LOOP HERE |  |
| LAST: |  |
| MOV AH,4CH | Loads 4ch value in ah register |
| INT 21H | Interrupts the Program |

**Snapshot:-**







**Result:-**

## Thus the above programs for performing matrix addition and subtraction are executed successfully using MS – DOSBox.