LTCE

Aim :- Use of programming tools to Perform basic arithmatic operations on 8 and 16 bit data

Theory :-

Assembling the program :

To assemble the program two assemblers are available for the IBM-PC. They are : Micro soft Macro Assembler and Borland Turbo Assembler. Besides doing the tedious task of producing binary codes for the instruction statements, an assembler also allows the user to refer to data items by name rather by numerical address. This makes the program much more readable. In addition to program instructions, the source program contains directives to the assembler.

MASM <file name.ASM>

The <file name.ASM> file that contains the assembly language program is assembled. The assembler generates error messages if there are any error.

Test and Debug :-

The executable program can be run under DOS or DEBUG. As a thumb rule a program under DOS only when there is no error or it produces some not visible or audible result. If the program result is stored in registers or in memory, the result is visible. Hence it should be run using DEBUG or TD or code-view only .EXE file can be loaded into memory using DEBUG.

Commands :-

1) a (Assembler)

Format : -a starting address <enter>

It is used to enter the assembly language program in the code segment from the given offset address.

2) e (Enter)

Format: -e memory-address <enter>

It is used to edit (modify) or enter data segment from the given offset address. To enter data in consecutive memory locations use space bar for next memory location.

3) u (Un assemble)

Format: -u starting-address ending-address <enter>

It is used to unassemble program or part of the program. It displays Program or part of the program in following format.

Address op code mnemonic operand
XXXX : YYYY

It displays program or part of the program along with its operation codes from the starting-address to ending address.

4) g (go)

Format: -g = starting-address ending address <enter>

It is used to execute program or part of the program which is present in code segment from start-address to end-address. It displays the result if it is present in the registers of the 8086 Microprocessor.

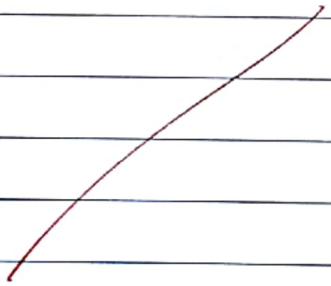
5) d (Dump or display)

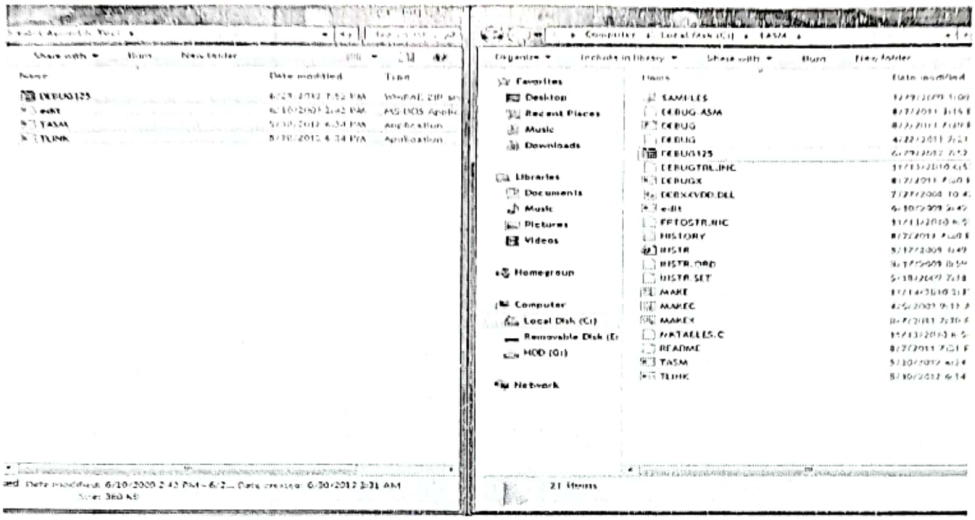
Format: -d start-address end address <enter>

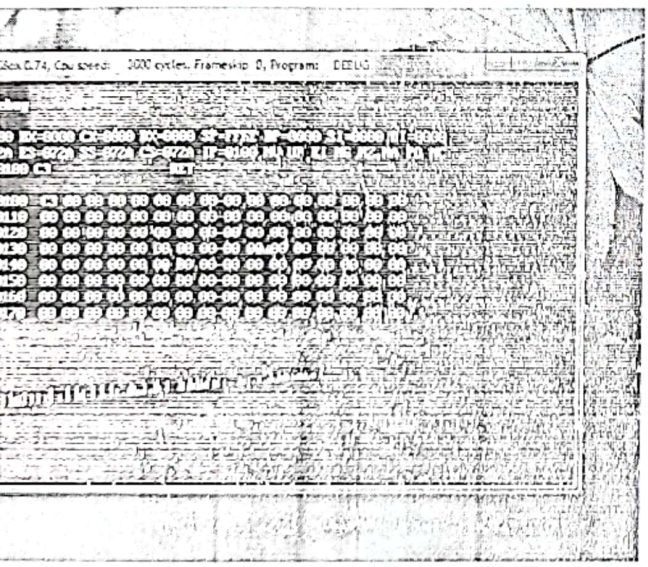
It is used to display the contents of the memory locations from the given address range. Hence it can be used to display result of the program if it is stored in the memory.

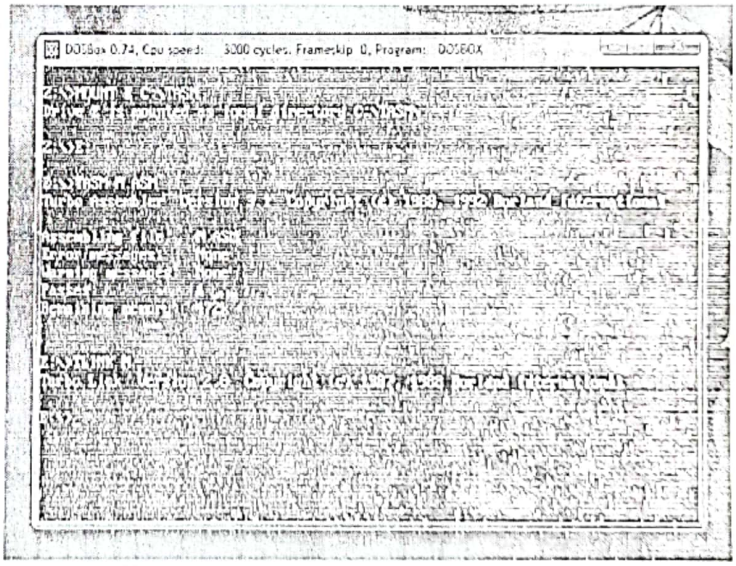
Conclusion :

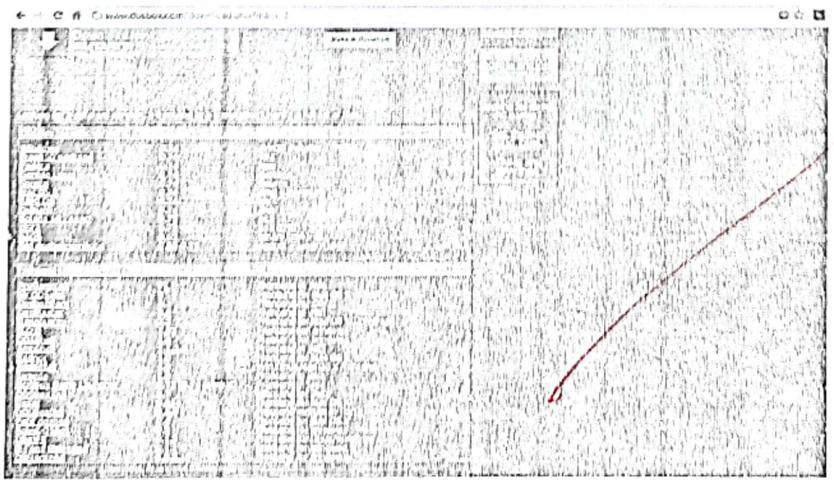
All above ~~debug~~ commands of debug are studied by entering test program, data, executing program and displaying the contents of 8086 register as well as contents of memory locations

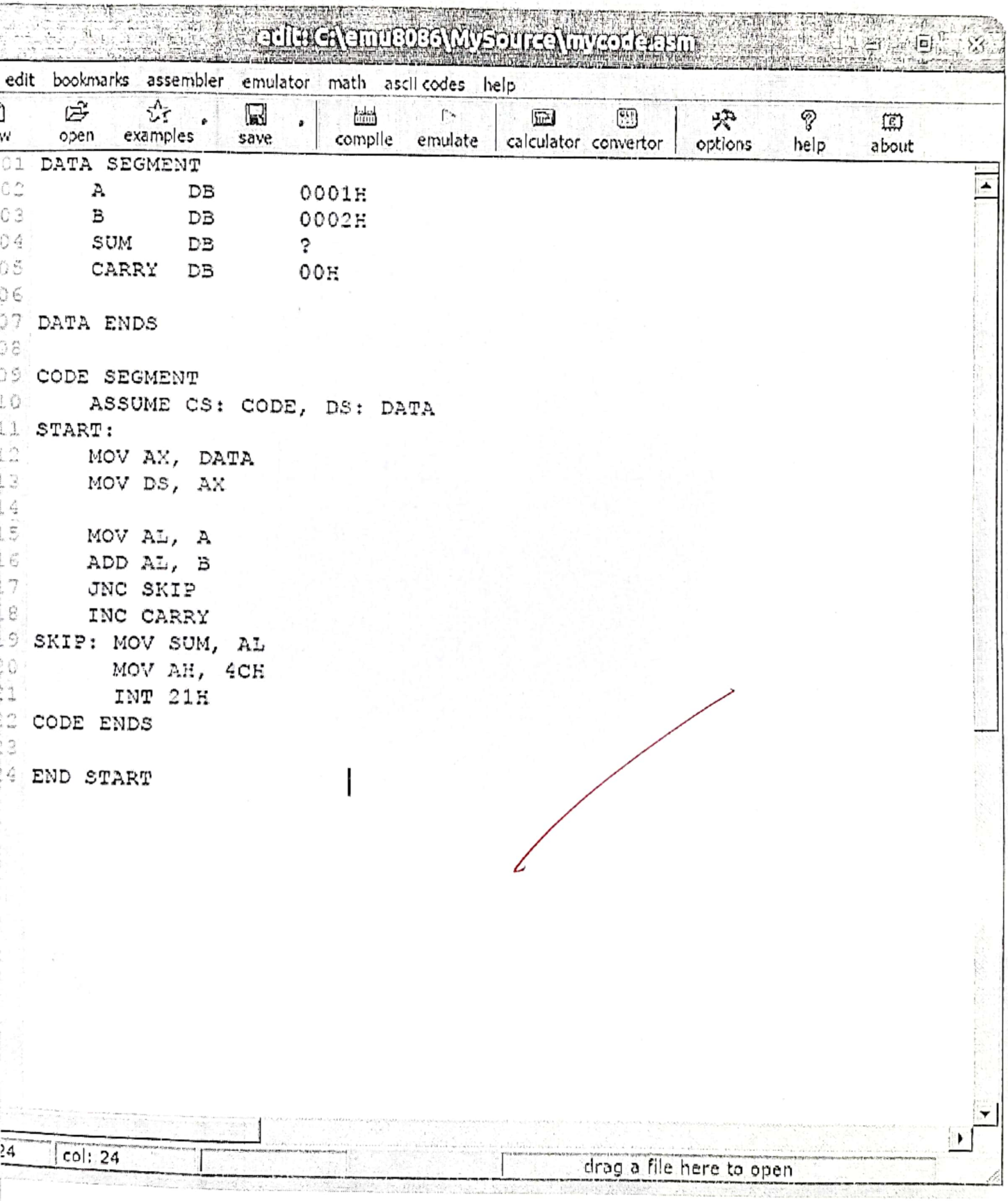


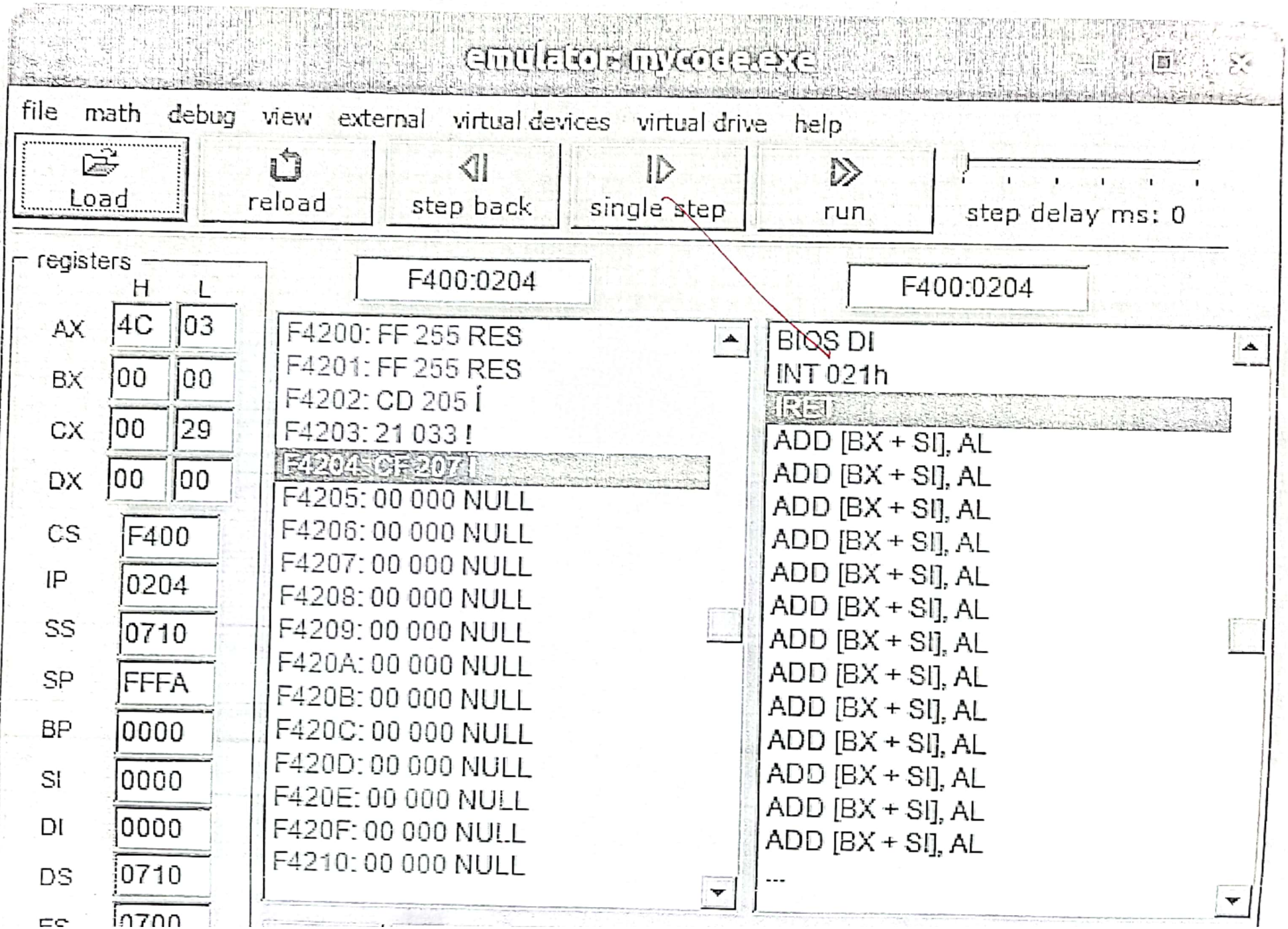












LTCE

Name : Vivek Manik Chaudhari

Roll No : 150

Batch : A2

DOP :

DOS :

Sign : 13/2/23

Marks : 15/15

LTCE

Aim : WAP to convert BCD to Hexadecimal, hexadecimal to BCD.

Theory :

8086 program to convert an 8 bit BCD number into hexadecimal number.

Assumption :-

Initial value of segment register is 00000. Calculation of physical memory address :

Memory Address = Segment Register * 10H + Offset where Segment Register is 00000 and Offset is given in the program.

Algorithm :-

1: Assign value 500 in SI and 600 in DI
2. Move the contents of [SI] in BL.
3. Use AND instruction to calculate AND between 0F and contents of [BL]
4. Move the contents of [SI] in AL
5. USE AND instruction to calculate AND between F0 and contents of AL
6. Move 04 in CL
7. Use ROR instruction on AL
8. Move 0A in DL
9. Use MUL instruction to multiply AL with DL

10. Use ADD instruction to ADD AL with BL
11. Move the contents of AL in [DI]
12. Halt the program.

8086 program to convert 8 bit BCD number into ASCII Code.

Assumption :-

Starting address of program : 400
Input memory location : 2000
Output memory location : 3000

Input :

DATA : 98 H in memory location 2000

Output :

DATA : 38 H in memory location 3000 and 39 H in memory location 3001

Algorithm :

1. Load contents of memory location 2000 in register AL
2. Copy contents of register AL in register AH.
3. Perform AND operation register AL with 0F
4. Assign 04 to CL Register.
5. Shift the contents of AH by executing SHR instruction using CL.

6. Perform OR operation on register AX with 3030
7. Store the content of AX in memory location 3000.

Problem: Write a program to convert ASCII to BCD 8-bit number where starting address is 2000 and number is stored at 2050 memory address and store result into 3050 memory address.

Input: Location - 2050
Data - 37

Output: Location - 3050
Data - 07

Algorithm:

1. Move value at [2050] into AL
2. Perform AND operation on AL with 0F
3. Move contents of accumulator AL into 3050
4. Stop

Explanation:

Register AL is used for general purposes.

LTCE

8086 program to convert 8 bit ASCII to BCD number.

Problem : Write a program to convert ASCII to BCD 8-bit number where starting address is 2000 and number is stored at 2050 memory address and store result into 3050 memory address.

Input : Location - 2050
Data - 37

Output : Location - 3050
Data - 07

Algorithm :

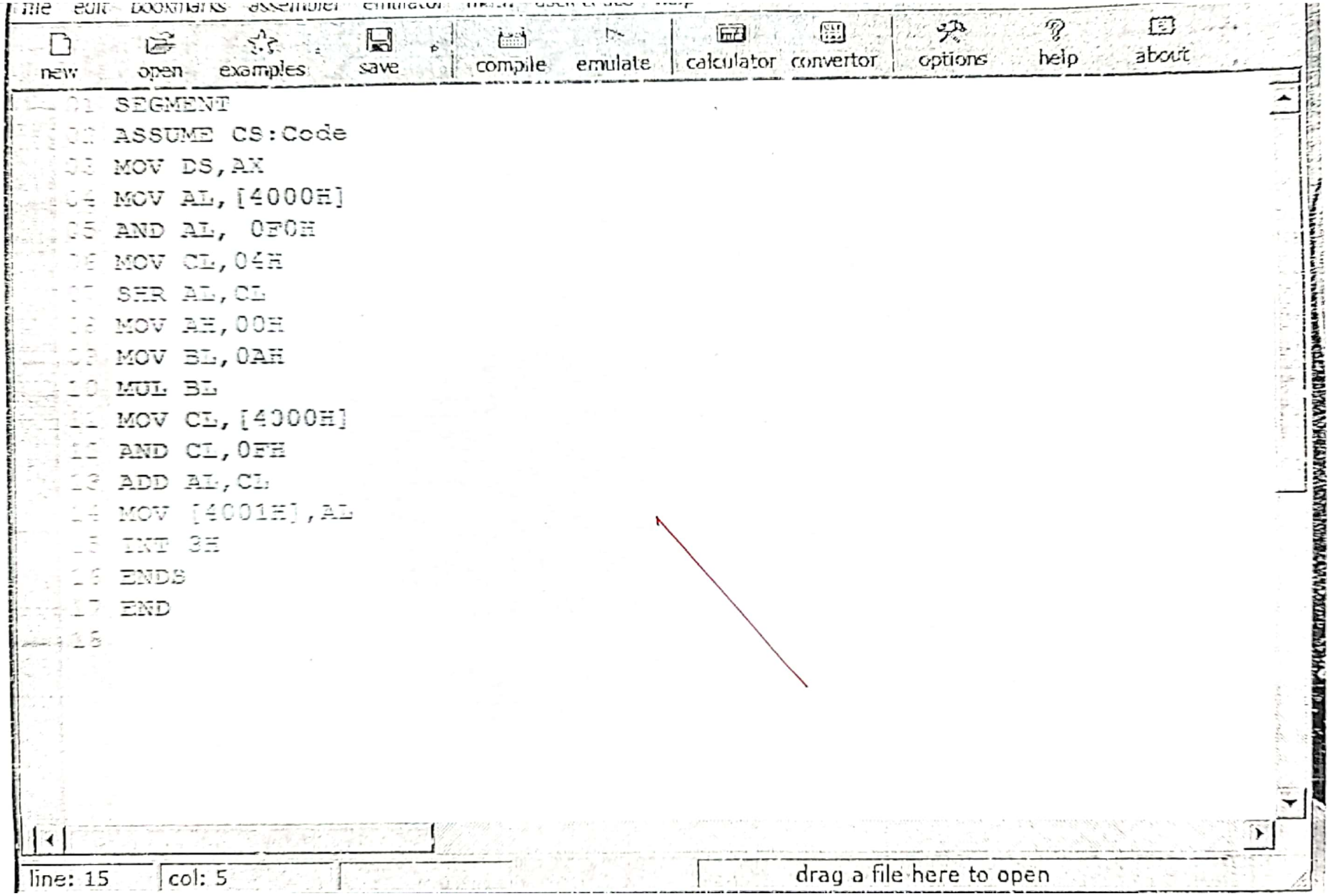
1. Move value at [2050] into AL
2. Perform AND operation on AL with 0F
3. Move content of accumulator AL into 3050
4. Stop

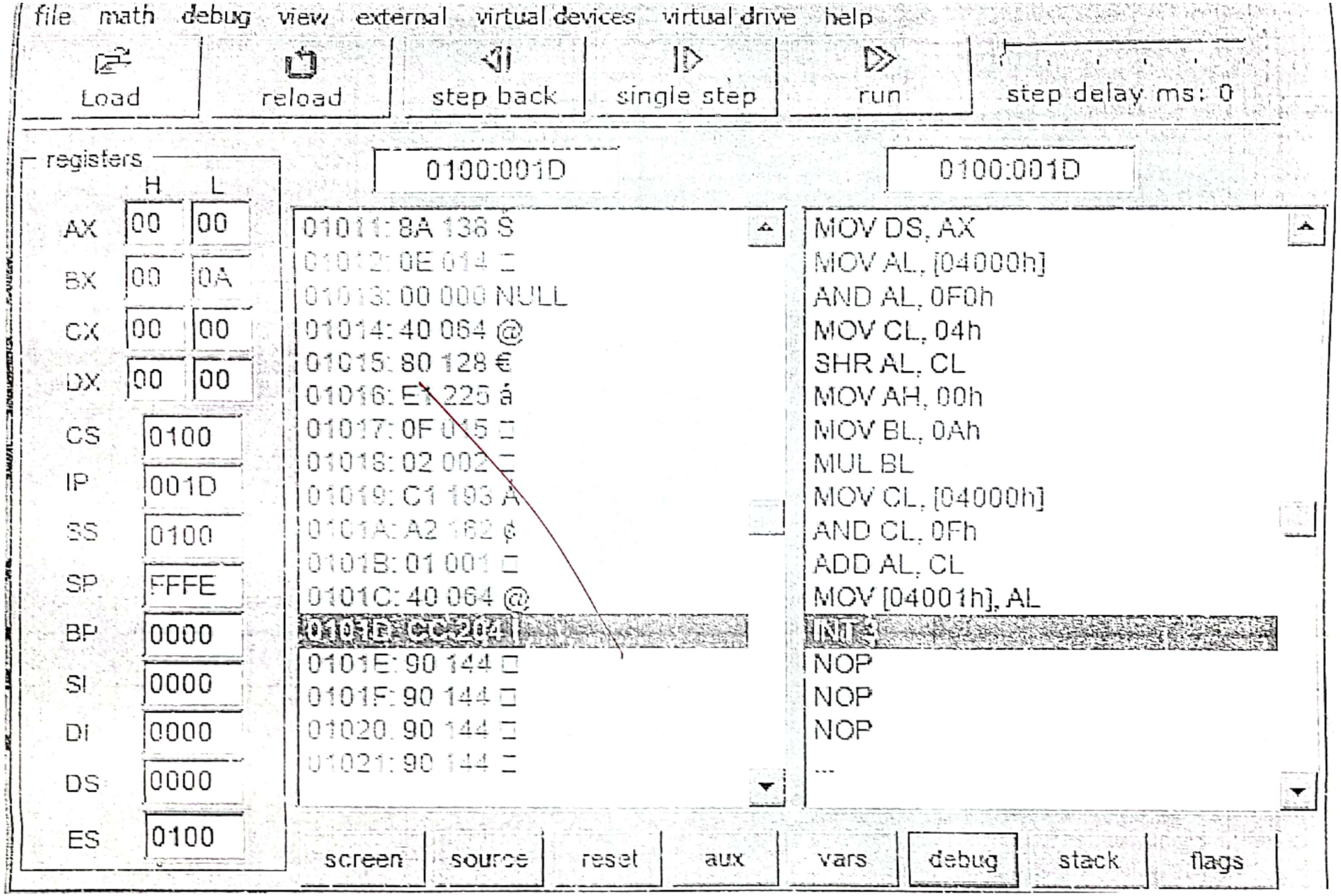
Explanation :

Register AL is used for general purpose
1. MOV is used to transfer Data
2. AND is used for Multiplication (logically)
3. HLT is used to Halt program.

Conclusion :

We Successfuly implemented program to convert Dec BCD to Hexadecimal and Hexadecimal to BCD





new open examples save compile emulate calculator convertor options help about

```
01 CODE SEGMENT
02      ASSUME CS:Code
03      MOV AX,2000H
04      MOV DS,AX
05      MOV AL,[4000H]
06      MOV AH,00H
07      MOV BL,0AH
08      DIV BL;AX÷BL
09      MOV CH,AH
10      MOV AH,00H
11      DIV BL
12      MOV [4002H],AL
13      MOV AL,AH
14      MOV CL,04H
15      SHL AL,CL
16      ADD AL,CH
17      MOV [4001H],AL
18      INT 3H
19 CODE ENDS
```