EXPERIMENT 11

Array In Assembly

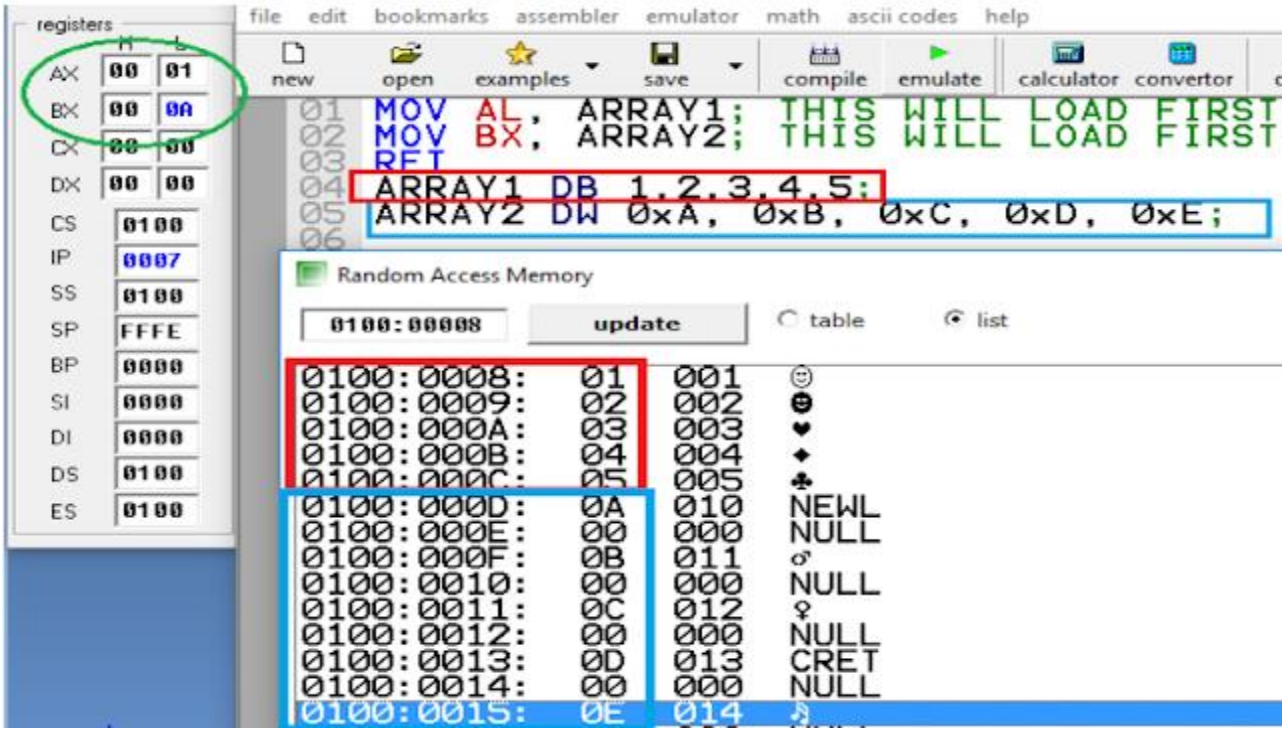
Objective

- Understand the Working of Array in Assemble language

Theory

What is array? We already know the answer. An array is a collective name given to a group of similar quantities. Like other programming languages, in assembly there are some methods to declare an array. Common things are there will be a name of the array, it's data type, it's length and it's initial value.
To define an array of 10 elements, each of 1-byte size, one can write
ArrayName db 1,2,3,4,5,6,7,8,9,10; This will reserve 10 bytes in consecutive memory locations.
Similarly, to define an array of 10 elements, each of two byte sized, one can write
ArrayName dw 1,2,3,4,5,6,7,8,9,10; Same goes for double type or quad-word type arrays with db replaced with dd or dq, respectively.

How data in arrays is represented in memory?
Consider few examples below of defining arrays; array1 is byte type array whereas array2 is word type array.
array1 db 1,2,3,4,5;
array2 dw 0xa, 0xb, 0xc, 0xd, 0xe;
array1 is shown via highlighted area in the figure attached as contiguous locations in memory. Array2 is stored in memory locations immediately after array1 elements. Note that array2 is word type array, requires two bytes for one elements, and those two bytes are represented in little endian notation as lower byte at lower address in memory. 0xA when written in two bytes is represented as 000A, here, 0A being the lower byte is stored at lower address 010D and 00 being the higher byte is stored at next location address 010E. Next word from array2 is stored at next location in same format.



To define an array of 100 elements ''dup'' operator can be used such as;
Array1 db 100 dup(0); defines an array of 100 elements with zero as initial values.

Accessing data in arrays?
Consider examples below.

i. Using array name in instruction refers to the first element of that array interpreting array name as the address of first element.

```
MOV AL, ARRAY1; THIS WILL LOAD FIRST ELEMENT FROM ARRAY1
MOV BX, ARRAY2; THIS WILL LOAD FIRST ELEMENT FROM ARRAY2
RET
ARRAY1 DB 1,2,3,4,5;
ARRAY2 DW 0XA, 0XB, 0XC, 0XD, 0XE;
```

To access next elements within an array, add offsets to array name depending upon array type.

```
MOV AL, ARRAY1+1; THIS WILL LOAD SECOND ELEMENT FROM ARRAY1
MOV BX, ARRAY2+2; THIS WILL LOAD SECOND ELEMENT FROM ARRAY2
RET
```

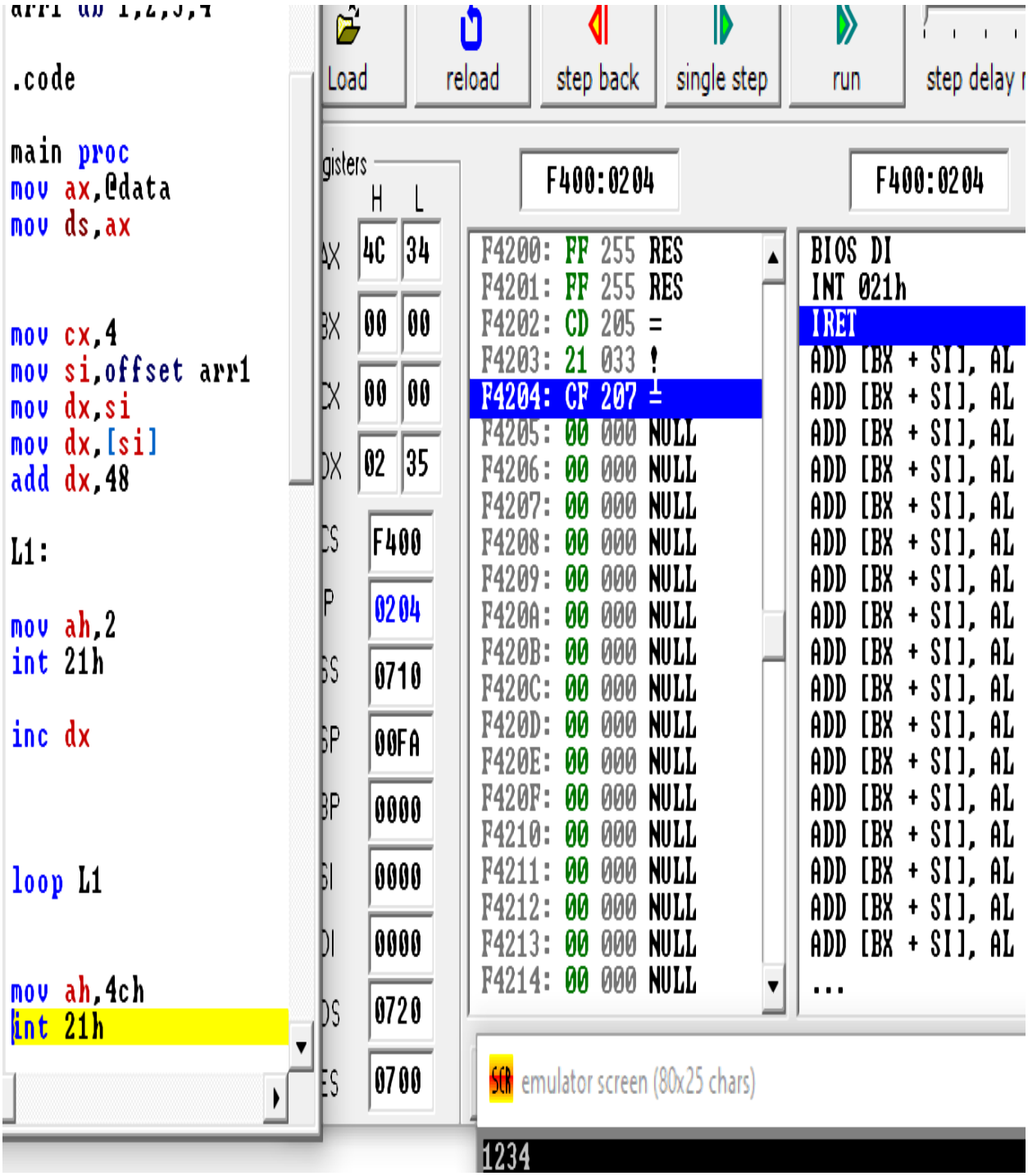
ARRAY1 DB 1,2,3,4,5;
ARRAY2 DW 0XA, 0XB, 0XC, 0XD, 0XE;

Note that to refer to next element within array, you will have to add offset according to array element size. As in above example, to access second element in array1 you add '1' whereas to access second element in array2 you add '2' to primary address.

Task:

Q1: Print an array using loop condition.

```
02
03 .model small
04 .stack 100h
05 .data
06
07 arr1 db 1,2,3,4
08
09 .code
10
11 main proc
12     mov ax,@data
13     mov ds,ax
14
15
16     mov cx,4
17     mov si,offset arr1
18     mov dx,si
19     mov dx,[si]
20     add dx,48
21
22     L1:
23
24     mov ah,2
25     int 21h
26
27     inc dx
28
29
30
31     loop L1
32
33
34   mov ah,4ch
35   int 21h
36
37 ret
38
39
40
```


arr1 dw 1,2,3,4
.code
main proc
mov ax,@data
mov ds,ax
mov cx,4
mov si,offset arr1
mov dx,si
mov dx,[si]
add dx,48
L1:
mov ah,2
int 21h
inc dx
loop L1
mov ah,4ch
int 21h
Load
reload
step back
single step
run
step delay
F400:0204
F400:0204
H L
AX 4C 34
BX 00 00
CX 00 00
DX 02 35
CS F400
IP 0204
SS 0710
SP 00FA
BP 0000
SI 0000
DI 0000
DS 0720
ES 0700
F4200: FF 255 RES
F4201: FF 255 RES
F4202: CD 205 =
F4203: 21 033 !
F4204: CF 207 ╧
F4205: 00 000 NULL
F4206: 00 000 NULL
F4207: 00 000 NULL
F4208: 00 000 NULL
F4209: 00 000 NULL
F420A: 00 000 NULL
F420B: 00 000 NULL
F420C: 00 000 NULL
F420D: 00 000 NULL
F420E: 00 000 NULL
F420F: 00 000 NULL
F4210: 00 000 NULL
F4211: 00 000 NULL
F4212: 00 000 NULL
F4213: 00 000 NULL
F4214: 00 000 NULL
BIOS DI
INT 021h
IRET
ADD [BX + SI], AL
...
SCR emulator screen (80x25 chars)
1234