

## Experiment 8

**Aim: Find the summation as mentioned in the list of experiments already circulated.**

Find the following summation:  $\sum x_j^2 / n$ , where x is an 8-bit number.

**Algorithms:**

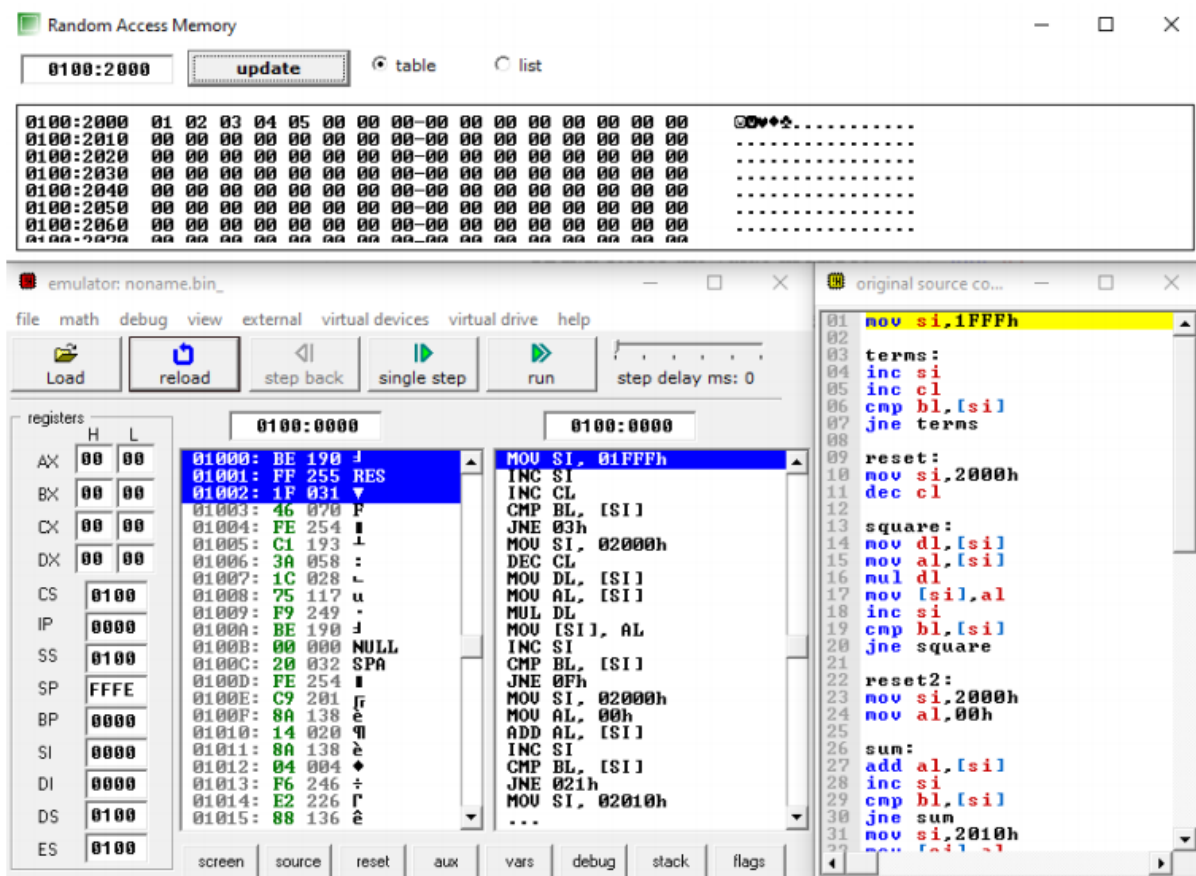
- 1) Assigns source index a value to point at an address '0100:1FFFh' in RAM.
- 2) Initializes 'terms' loop, increments the source index value by 1, increments the value in CL register by 1.
- 3) Compares the value in BL register and value in RAM address pointed by source index, jumps back and repeats 'terms' loop as long as value RAM address value is not 0.
- 4) Initializes 'reset' section of code, assigns source index a value to point at an address '0100:2000h' in RAM, decrements the value in CL register by 1.
- 5) Initialized 'square' loop, assigns the value in RAM pointed by source index to DL register, assigns the value in RAM pointed by source index to AL register.
- 6) Multiplies the values at AL and DL register and stores the result in the AL register, stores the value in AL register to RAM address pointed by source index, increments the source index value by 1.
- 7) Compares the value in BL register and value in RAM address pointed by source index, jumps back and repeats 'square' loop as long as value RAM address value is not 0.
- 8) Initializes 'reset2' section of code, assigns source index a value to point at an address '0100:2000h' in RAM, assigns the value '00h' to the AL register.
- 9) Initializes the 'sum' loop, adds the value at RAM address pointed by source index to the AL register, increments the source index value by 1.

## Code:

```
mov si,1FFFh
terms:
inc si
inc cl
cmp bl,[si]
jne terms
reset:
mov si,2000h
dec cl
square:
mov dl,[si]
mov al,[si]mul dl
mov [si],al
inc si
cmp bl,[si]
jne square
reset2:
mov si,2000h
mov al,00h
sum:
add al,[si]
inc si
cmp bl,[si]
jne sum
mov si,2010h
mov [si],al
div cl
mov si,2020h
mov [si],ax
hlt
```

## Output:

## Before execution



Random Access Memory

0100:2000 update table list

Address	Hex	ASCII
0100:2000	01 04 09 10 19 00 00 00-00 00 00 00 00 00 00 00	.....
0100:2010	37 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00	7.....
0100:2020	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00	.....
0100:2030	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00	.....
0100:2040	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00	.....
0100:2050	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00	.....
0100:2060	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00	.....
0100:2070	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00	.....

emulator: noname.bin

file math debug view external virtual devices virtual drive help

Load reload step back single step run step delay ms: 0

registers

Register	H	L
AX	00	00
BX	00	00
CX	00	05
DX	00	05
CS	0100	
IP	0034	
SS	0100	
SP	FFFE	
BP	0000	
SI	2020	
DI	0000	
DS	0100	
ES	0100	

0100:0034 0100:0034

Address	Hex	Disassembly
01021: 02 002	MOV [SI], AL	
01022: 04 004	DIU CL	
01023: 46 070 F	MOV SI, 02020h	
01024: 3A 058	MOV [SI], AX	
01025: 1C 028	HLT	
01026: 75 117	NOP	
01027: F9 249	NOP	
01028: BE 190	NOP	
01029: 10 016	NOP	
0102A: 20 032 SPA	NOP	
0102B: 88 136	NOP	
0102C: 04 004	NOP	
0102D: F6 246	NOP	
0102E: F1 241	NOP	
0102F: BE 190	NOP	
01030: 20 032 SPA	NOP	
01031: 20 032 SPA	NOP	
01032: 89 137	NOP	
01033: 04 004	NOP	
01034: F4 244	NOP	
01035: 90 144	NOP	
01036: 90 144	NOP	

original source code

```

07 jne terms
08
09 reset:
10 mov si,2000h
11 dec cl
12
13 square:
14 mov dl,[si]
15 mov al,[si]
16 mul dl
17 mov [si],al
18 inc si
19 cmp bl,[si]
20 jne square
21
22 reset2:
23 mov si,2000h
24 mov al,00h
25
26 sun:
27 add al,[si]
28 inc si
29 cmp bl,[si]
30 jne sun
31 mov si,2010h
32 mov [si],al
33 div cl
34 mov si,2020h
35 mov [si],ax
36
37 hlt

```

screen source reset aux vars debug stack flags

**(ARKAJYOTI CHAKRABORTY 2K19/EP/022)**