horizontal line

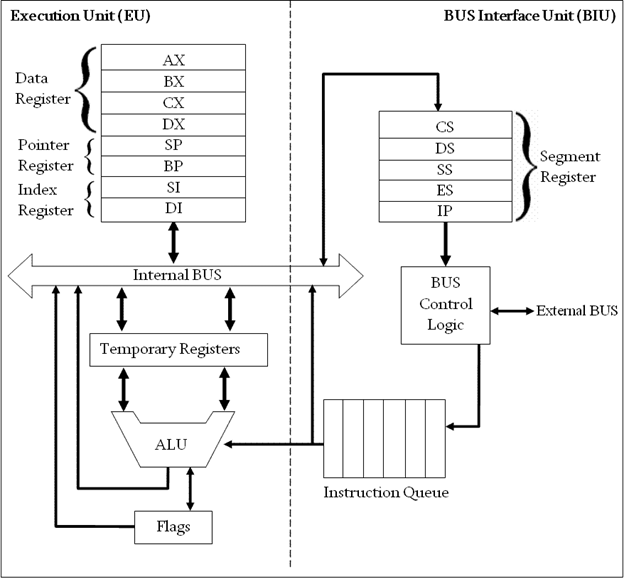
**Delhi Technological University**

Department of Applied Physics

IVth Semester

**MICROPROCESSORS & INTERFACING**

**MPI EP - 206**



# **Submitted by :**

**ADITYA SINGH, 2K19/EP/005**

**Submitted to:**

**Dr. Rishu Chaujar**

# **Experiment 8**

**Find the Summation:**

**[xj2/ n], where x is an 8-bit integer.**

**THEORY**

1. SI points to the first memory location(where the first number is stored).
2. Increment SI to go to the next term and Increment CL to store the number of terms.
3. Repeat steps 1-3 until SI encounters 00 in memory location (which indicates the end of the list of numbers).
4. Reset SI so that it points to the beginning of the list of numbers.
5. Decrement CL (because it counts one term extra).
6. square:
7. Store the number pointed to by the SI register in AL and DL registers.
8. Multiply the contents of AL and DL registers and store it back in the location pointed by the SI registers. This stores the square of the number in place of the number itself.
9. Repeat steps 7-9 until SI encounters 00 in memory location.
10. Reset SI so that it points to the beginning of the squared numbers and store 0 in the AL register.
11. sum:
12. Add the terms and increment SI in each iteration and store the sum in the AL register.
13. Repeat step 13 until SI encounters 00 in memory location.
14. Divide the sum by value in the CL register. Remainder goes in the AH register and quotient in the AL register.

**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**

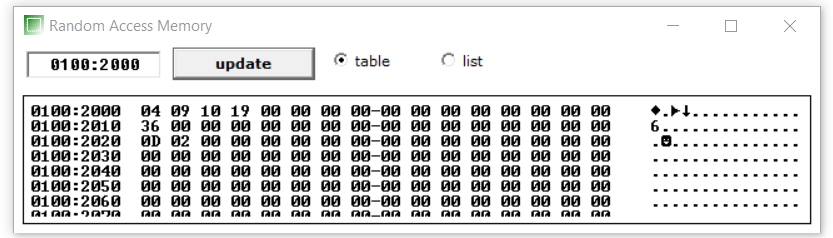
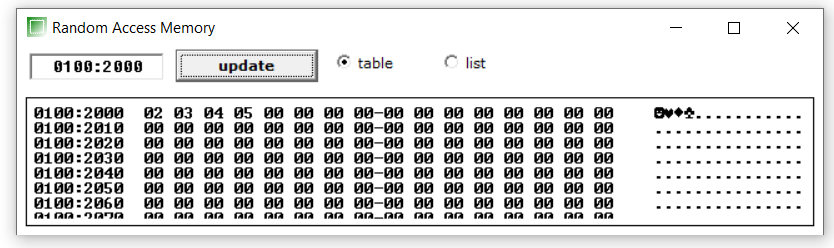
**div cl**

**mov si,2020h**

**mov [si],ax**

**hlt**

**OUTPUT**



**END**