KHULNA UNIVERSITY OF ENGINEERING & TECHNOLOGY

Project Report

Department: Elecctronics and Communications Engineering

Course No.: ECE 2204

Course Title: Microprocessors and Microcomputers Laboratory

Topic: Series Analyzer and Generator Using 8086 Microprocessor.

Date:04/12/2024

Submitted By:

Muntasir Billah Nakeeb

Roll: 2109016

Objectives:

1. To implement an assembly code for analyzing and building a series.
2. To perform division, multiplication, addition, subtraction.
3. To find difference and sum for a nth element of a given series.
4. To find the series from a given 1st element of a series and the difference between two elements. Also, find the sum for nth elements.
5. To be able to use 8086 microprocessor.

Apparatus:

Table-1: List of required apparatus: -

Apparatus name	Quantity	Ratings
Microprocessor kit (8086-trainer-kit)	01	0-5V
Microprocessor software (emu8086)	01	_____

Theory:

In this project there are two sections. One is series analyzer and the other is series Generator. Let's discuss both sections:

Mathematically,

$$Tn=a+(n-1)\times d$$

Where,

Tn= Nth element of a series

a= First element of a series

n= No of element of a series

d= difference between two elements of a series

Sn=Sum of n element of a series

$$Sn=n\times(2a+(n-1)\times d)/2$$

Series analyzer: For a given series it will find the difference by which the series was built and store the result in a location. Then find the sum of the n element and store into the next element.

Series Generator: IF the data like difference between two respective elements and the first element is given then the series can be built. The Nth element of the series and sum of N element can be found.

Code with Description:

Mnemonics for the project with necessary description:

https://github.com/Nakeeb27/Series-Analyzer-and-Generator-Using-8086-Microprocessor./blob/main/M.pdf

User Manual:

For Series analyzer:

1. Give the input 0 at 9400.
2. Give the first term of the series at 9401.
3. Give the second term of the series at 9402-9403 (in terms of 16-bit data).
4. Give the third term of the series at 9404-9405 (in terms of 16-bit data).
5. Give the value of N (To find Nth term of the series) at 9406.
6. Give the value of N (To find Sum of Nth term of the series) at 9407.
7. Now run the program and see the output at-
    - 9500= Difference between two respective terms.
    - 9501-9502=Nth term of the series (in terms of 16-bit data).
    - 9503-9506=Sum of nth term of the series (in terms of 32-bit data).

For Series Generator:

1. Give the input 1 at 9400.
2. Give the first term of the series at 9401.
3. Give the difference between the respective two terms at 9402.
4. Give the value of N (To find Nth term of the series) at 9403.
5. Give the value of N (To find Sum of Nth term of the series) at 9404.
6. Now run the program and see the output at-
    - 9500-9509= First five terms of the series in terms of 16-bit data.
    - 950A-950B=Nth term of the series (in terms of 16-bit data).
    - 950C-950F=Sum of nth term of the series (in terms of 32-bit data).

For better performance we have to consider these,

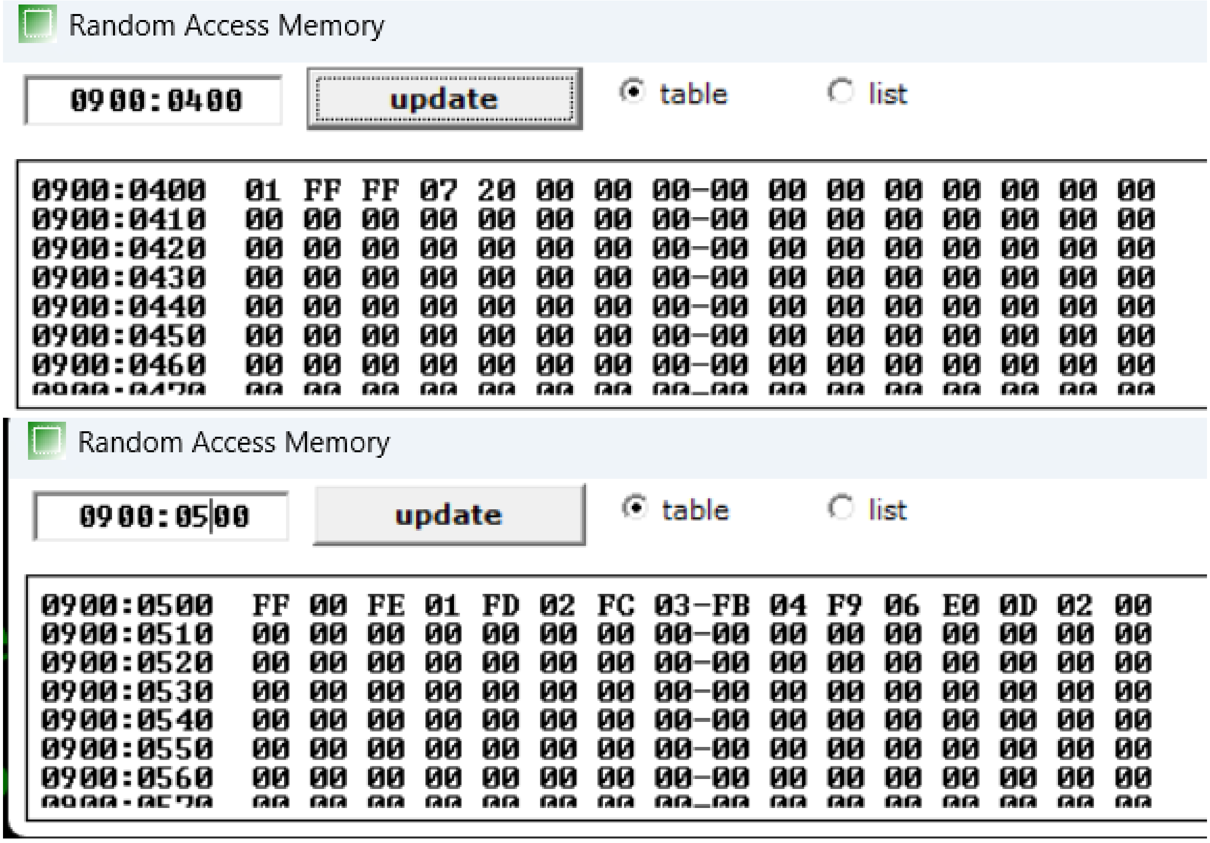
Limitation:

1. The first term of the series has to be 8-bit data.
2. Difference between respective two terms must be in between 8-bit
3. The value of N must be less than or equal to 256(decimal) or 100H.

If a user doesn’t follow the manual, the developer will not be responsible for this.

Result analysis & Discussion:

We presented our data in a memory figure given below.



Random Access Memory

0900:0400 update ● table ○ list

```
0900:0400   01 FF FF 07 20 00 00 00-00 00 00 00 00 00 00 00
0900:0410   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
0900:0420   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
0900:0430   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
0900:0440   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
0900:0450   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
0900:0460   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
0900:0470   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
```

Random Access Memory

0900:0500 update ● table ○ list

```
0900:0500   FF 00 FE 01 FD 02 FC 03-FB 04 F9 06 E0 0D 02 00
0900:0510   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
0900:0520   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
0900:0530   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
0900:0540   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
0900:0550   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
0900:0560   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
0900:0570   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
```

Fig 01: Status of memory after execution.

This figure showed the data for series generator. Where we take FFH as first term and also as difference between two respective terms, found 07Hth term and found first 20Hth term. Finally output satisfied us and we were successful to complete our project. We cross-checked our output in both cases. We analyzed several cases. But by sacrificing the limitations of the project it worked perfectly. There was no error. The basic planning and the execution were perfect. We tried to do our best. We have successfully completed our project with some tolerable limitations. This limitation was not our lack of planning, this is the limitations of 8086 microprocessor. As a teamwork, we all contributed our best efforts, and the output was presented previously. We have completed our all objectives, we have analyzed deeply the register and flag status and other everything.

Conclusion:

This project will provide to operate with series. Both the analyzer and the generator have practical applications in various fields, from academic mathematics to practical real-world problems in finance, engineering, and physics. They provide a structured way to work with sequences and can be implemented in many programming environments, including assembly language or higher-level languages, depending on the need for performance and simplicity.

Reference:

1. *https://www.tutorialspoint.com/microprocessor/microprocessor_8086_overview.htm*
2. *https://www.javatpoint.com/8086-microprocessor*
3. *https://www.tutorialspoint.com/microprocessor/microprocessor_8086_instruction_sets.htm*
4. *https://www.geeksforgeeks.org/8086-instruction-set/*