

Intel 8085 Microprocessor Simulation Tool

“OneX Simulator”

Annirudh Shukla

Department of Computer Science and
Engineering
Neotia Institute of Technology,
Management and Science
Kolkata, India
e-mail: officialani2@gmail.com

Writwick Das

Department of Computer Science and
Engineering
Neotia Institute of Technology,
Management and Science
Kolkata, India
e-mail: writzx@gmail.com

Pronoy Mukherjee

Department of Computer Science and
Engineering
Neotia Institute of Technology,
Management and Science
Kolkata, India
e-mail: mukherjee_pronoy@yahoo.in

Deep Suman Dev

Department of Computer Science and Engineering
Neotia Institute of Technology, Management and Science
Kolkata, India
e-mail: deepsumandev@yahoo.co.in

Basudev Halder

Department of Computer Science and Engineering
Neotia Institute of Technology, Management and Science
Kolkata, India
e-mail: basudev_h@yahoo.com

Abstract— In spite of the advancement in computer architecture and availability of microprocessors (ex. Intel Core i7 etc.) with speed thousand times greater, microprocessor 8085 is still widely used in academia for education and research purposes. To make the microprocessor 8085 more accessible and portable, many simulators for the same have been introduced over the past years with various level of user friendliness. This paper proposes a simple yet powerful enhanced simulator, namely OneX which is capable of real time code parsing, error handling and label parsing for different addresses. Proposed software also addresses few flaws present in the earlier version of 8085 microprocessor software. The latest copy of the software is available at (<https://github.com/Pronoy999/Project-OneX>)

Keywords: *assembly language, education, machine language, microprocessor, simulator, intel 8085*

I. INTRODUCTION

The development in computer science and software development have also impacted the academic practices in engineering education. To visualize highly abstract mathematical problems and theories, the use of graphics and pictorial representation have become popular and essential over the past few years^[1]

In the year of 1976, Intel produced an 8-bit microprocessor. They used the new "5" in the name to signify that the new microprocessor uses only +5-volt of power using depletion mode of power supply^[4].

In educational and engineering fields, where students or professionals use the Intel 8085 microprocessor might face some problems with the hardware simulator boards which are not readily available. If a student wants to test an assembly language program, it is not always possible to have access to labs which house the hardware kits. If one wishes to test an assembly language program of 8085 microprocessor, the first thing he must do is to convert his assembly language program to machine language, which is not understandable for the user. They also must calculate the memory addresses manually which can be strenuous. Then they must enter that machine code to hardware board and double-check their entry. Even after executing the program correctly they would not have been able to save the program and would have to enter it again for every time they want to execute it.

Another problem is that students might not have access to

the hardware board while doing their homework. Additionally, the hardware boards are costly. However, a software approach which can be run in any computer, including personal laptops, would ease up the process of testing and debugging significantly^[1]. OneX tries to solve these problems. OneX is portable and it can be easily run on a windows computer with a very basic configuration.

The users don't need to convert the assembly language code to machine language code to run in OneX. OneX provides automatic conversion. The strenuous work of calculating the address space for code is no longer required. OneX also provides a way to edit any single line of the code at point of time. OneX also provides a way to debug the code with line by line execution facility which is lacking in most of the simulators out in the market.

Nowadays, every engineering student has a computer and hence they no longer need to search for the hardware kit to test an 8085 code. One can sit at his will and test out the code. OneX is also perfect for a beginner. It provides a data monitor where the user can see every step that is happening inside the processor. These helps in better understanding of the processor's architecture. It is also beneficial for experienced programmer. They can take the full advantage of the beautiful syntax highlighting feature which is expected from a high-level integrated development environment (IDE) used for a large project. OneX also helps in visualizing the entire code. These features are very much lacking in the simulators out on the market. OneX has taken great care for both the students, who

are going to love its simplicity and detailed description of the process and for the experienced programmers who are going to love the syntax highlighting and error correction which they expect from an IDE.

II. RELATED WORK AND A COMPARATIVE STUDY

Various simulators for 8085 microprocessor have come into picture over the past few decades. The simulators evolved from supporting very basic operations to visualizing the code interpretation as well as memory mapping. Some of the earlier simulators, developed between 1980-2000, are ‘‘uPsim version 1.12’’ [3], ‘‘up8085 Simulator’’ [4], ‘‘8085 Simulator Bubble SORT 2.85’’ [5], and ‘‘8085 under DOS’’ [6]. Most of these earlier simulators did not support assembly code editor or machine code editor. To build a program, the user had to click the memory location from the list of memory locations and then enter the code or the user could load only an assembly language program and not a machine language program. ‘‘up8085 Simulator’’ by Pinkesh Creation’s and ‘‘8085’’ (under DOS) by V. Kumar did not support assembly language. One of the first simulator which supported assembly code was ‘‘8085 Simulator BubbleSORT 2.85’’. It also introduced the assembly code editor. But it did not support interrupt testing or I/O ports. Those two features were introduced in up8085 Simulator although it was Available with range of only seven port addresses. In the year 2003, an enhanced simulator, ‘‘8085 SimuKit’’, was proposed by ALI CHEHAB, SAMER HANNA, KARIM Y. KABALAN, ALI EL-HAJJ. [7] This simulator came with an assembly code editor, machine code editor and supported the execution of codes in both the formats. It also allowed the user to verify interrupt requests during program execution and offered the user with the complete range of 256 input/output ports. Moreover, in ‘‘8085SimuKit, the user can view the port setting in both bit format as well as hexadecimal format. Needless to say, it was a remarkable step towards the modern-day simulators.

But, the simulators which have been popular so far, have had some major limitations. In one of the most popular simulators among Indian students, named Sim3 8085 simulator, the user must rewrite the whole code if they make any single mistake in any line subsequently. It also doesn't have any support for typing the code through a keyboard. The user must enter the entire code by clicking various buttons on the screen. They also didn't have any option to save the program and to run that later. In some there were options to save the code but not memory. Saving memory along with the code is essential as some may depend on the memory values to function. Sim3 didn't show any error while typing and it didn't let the user even edit. So, a user would be notified of the error only he has completed the code and was ready to run and they wouldn't have any option to edit after that, the only solution left was to rewrite the entire code. Hence, a solution was needed for students or enthusiast professionals who would want a convenient yet economical and portable solution for the problem.

OneX provides a convenient way to simulate the 8085 microprocessor under Microsoft Windows using the .NET framework. It lets user to verify their programs in both the

assembly language and in machine language. It allows users to save their program and edit and run them later in time. It is also capable of saving the editable memory used to run the program of the 8085 along with an individual program. Users have been benefited from it as they have been benefited from an integrated development environment (IDE) for a high-level language such as Java or C++.

It also has the functionality of parsing the code in real time. It can detect an error in syntax instantly and report it back to user by highlighting erroneous code. It also supports label parsing in real time.

The user can store the code in a file and run the code later. Thus, adding in the reusability and portability. The file extension used for saving the code is ‘‘.onex’’ according to the name of the project. This file stores the whole code written by the user and the snapshot of the memory at the time of saving. It is possible to associate the application with the ‘‘.onex’’ extension for future convenience.

III. SOFTWARE DESCRIPTION

At present, the 8085 microprocessor trainer kit is made by Minimax Electronics in India which is used as practice devices for the microprocessor-related subjects. The 8085 microprocessor has 246 instruction sets. The software has been written in the C# using the .NET framework. The developed software is user friendly. The key feature that marks OneX simulator much convenient than already available ones, is the use of keyboard to write the assembly language program (ALP), instead of typing the hex code in the traditional trainer kit. It gives the user freedom to correct the errors, save their work and run the code at some suitable time after.

A. Editor

The editor is the first screen the user sees and the place where they enter the code using the computer qwerty keyboard instead of the hex keyboard used in the trainer previously. We have seen that the previous simulators such as the Sim 3 uses the button click approach to enter the instructions to the editor. The user can't enter the instruction with the keyboard, we have removed that shortcoming of Sim 3.

B. Dynamic Error Checking

OneX has also provided the user with the dynamic error checking capability and syntax highlighting. The user can check the error while typing the instructions. He wouldn't have to wait for the code to compile to check the typing mistakes.

C. Assembler

OneX has provided the user with the assembler to ease the understanding of every instructions and the memory that they occupy for every instruction. The assembler also provides a way for the traditional users who want to enter the instructions using the Hex code. The assembler shows the memory address, hex code, mnemonics, byte size, machine cycles and T states of each instructions of the written ALP. The user has an option to set the starting address from where they would want to begin writing the ALP.

D.Data Monitor

OneX provides users with data monitor to see the details of each and every step while executing the code. This helps the user in a great way to debug the code. Users can also choose to execute one step at a time, while debugging and hence they can pin point the origin of the error. OneX provides means to code for 8085 microprocessor like an IDE would provide for high level language like JAVA, C or C++

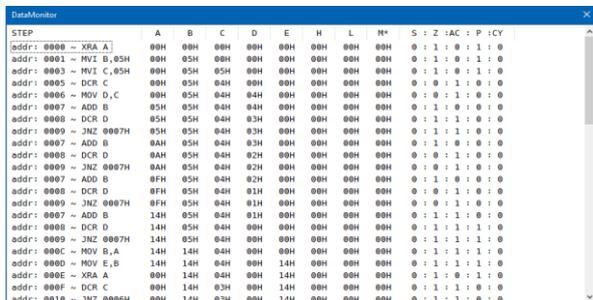


Fig 1. Data monitor showing the changes after executing instructions.

E.Memory Dump

OneX provides the user with a memory dump. It provides a bird’s eye view of the entire memory of the Intel 8085 microprocessor. The user can edit or insert new data or instruction directly in the memory in hexadecimal. In the Memory dump the user can jump to any offset.

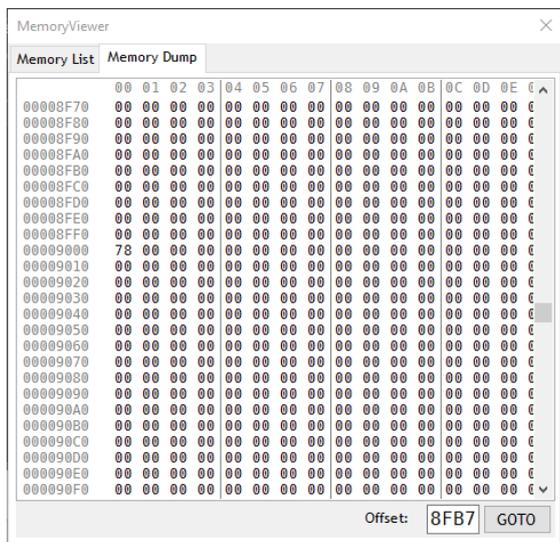


Fig 2. Memory dump of the Intel 8085 microprocessor.

F. Executer

OneX has a feature like the “dashboard” to control the execution of the Intel 8085 microprocessor. Here the user can also get the view of each register with the current value. The user can check the value of the program counter and the stack pointer. This information also helps in debugging. The user can choose to execute the ALP all at once or they can choose to run the ALP step by step to debug the code and check the changes that place either in Data Monitor or in the Executer. The user gets to view the flags after each execution, or they get to see the flags after completion of the execution.

IV. EXAMPLE CODE

A. Factorial of a number

```
MVI B,03H
MOV C,B
DCR C
LOOP1: MOV E,C
SUB A
LOOP2: ADD B
DCR E
JNZ LOOP2
MOV B,A
DCR C
JNZ LOOP1
STA 8000H
HLT
```



Fig 3: Writing the factorial code in OneX Simulator.

B.Fibonacci Series

```
MVI A,00H
STA 8000H
MVI A,01H
STA 8001H
MVI B,08H
LXI H,8000H
BACK: MOV A,M
INX H
ADD M
INX H
MOV M,A
DCR B
DCX H
JNZ BACK
HLT
```

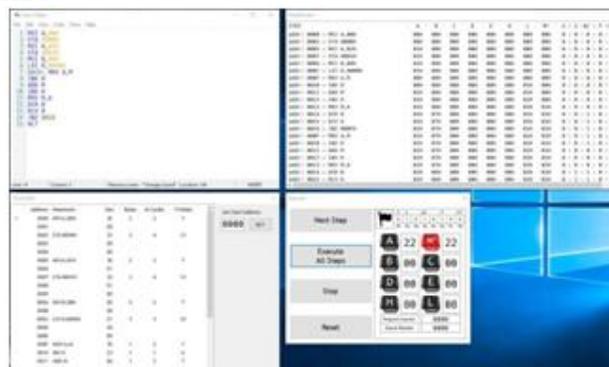


Fig 4: Writing the Fibonacci series code using the OneX Simulator.

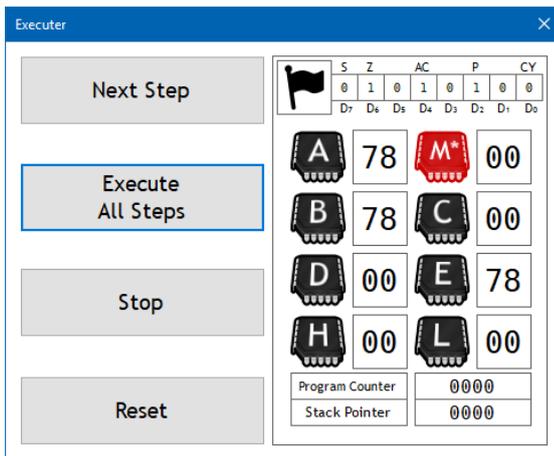


Fig 5. Executer of the OneX Simulator.

V. ALGORITHM FOR ONEX

In OneX simulator the main algorithm used is to parse the user's code for real time syntax highlighting and then interpreting it to execute.

The steps followed are:

1. The entire code of user is checked for labels.
2. Once the labels are inserted, the entire code is parsed.
3. Each line is read and checked for labels, the labels are extracted only after a white space.
4. Its then checked for instructions (left literal), which can be of either one byte, two byte or three bytes. Using Regular expression, each instruction and its type are extracted.
5. After the instruction, the right literal is extracted, that is the memory address or any other register name depending on the instruction type extracted earlier.
6. If there is a mismatch between the left and the right literal, the error is populated.
7. If any label is found in the right literal, it is replaced with the actual memory address, if its already defined previously else its left blank and later when the label is defined the memory is mapped accordingly.

Once the entire code is extracted and parsed successfully, the user chooses to execute it either one step at a time for debugging purposes or all at one step to get the result at desired register or memory address.

Once the code is executed user has an option to save the program along with the memory. OneX accumulates the program and the memory file and then stores it in a (.dot)onex format. This format can be directly opened by the OneX simulator with ease.

VI. FUTURE SCOPE AND CONCLUSION

The main objective of designing the simulator have been to provide a user-friendly model which is also open source. The implementation has been kept as much lucid as possible. The software can be easily used by anyone who have a basic

knowledge of 8085 microprocessor. An attempt has been made to implement all the functions of the instruction set which can be performed by the microprocessor. Thus, the user can get an inside knowledge of the Intel 8085 without using a microprocessor trainer kit. The working of Intel 8085 can be explained and studied in a very cost-effective yet simpler manner with the help of OneX.

The only drawback of OneX is that by design, it supports only Windows operating system because of its dependency on .NET framework. Use of .NET enables some very powerful feature to be added to OneX. With further evolution in software industries, the OS compatibility of OneX may hopefully be increased in future. That is, OneX can be designed with modified software tools as a cross-platform simulator without compromising its existing feature anyway.

In the near future, enhancements to this project can be made by including support for the external peripherals and the micro controller to integrate with the Intel 8085. OneX can be improved to behave as a virtual Intel 8085 microprocessor.

VII. REFERENCES

- [1] "8085 Microprocessor Simulation Tool", Ali Chehab, Samer Hanna, Karim Y Kabalan American university of Beirut.
- [2] "A Review Paper on 8085 Microprocessor" by Deepak Kumar, Divanshu Kaushik. Dronacharya College of Engineering, India.
- [3] K. Mladen, Advanced education and training using new digital simulator designs Proceedings of the American Power Conference, Vol. 59-1, 1997, pp 481486.
- [4] J. David Jeff, Hierarchical digital systems modelling utilizing hardware description languages for computer engineer education, Comp ElectrEng 21 (1995), 311-320.
- [5] H. Lawrence and R. Charles, "Teaching digital system design with a multilevel digital systems simulator" Proceedings—Frontiers in Education Conference, 1987, pp 3036.
- [6] I. Longair, Digital filter—an interactive computer program for the design and simulation of a finite impulse response (F.I.R.) digital filter, Int J Elect Eng Educ 23 (1986), 339-348.
- [7] A. Chehab, S. Hanna, K. Y. Kabalan, A. El-Hajji, 8085 microprocessor simulation tool "8085 Simukit", Comput Appl Eng Educ 12 (2004), 249-256.
- [8] K. George, Digital filter simulation in the classroom, Modeling and Simulation, Proceedings of the Annual Pittsburgh Conference, Vol. 21, Computers, Computer Architecture, and Microprocessors in Education, 1990, pp 1269-1273.
- [9] "A Simple Simulator for a basic Microprocessor" by Marek Repak.
- [10] "Microprocessor Quick Reference Guide" by Intel
- [11] "The MCS-80/85 Family User Guide" by Intel