International Journal of Applied Engineering Research
ISSN 0973-4562 Volume 10, Number 12 (2015) pp. 32415-32422

http://www.ripublication.com

A New Integrated Computer Architecture Simulation Framework Towards Next Generation Approach

[1]M. Narayana Moorthi and [2]R. Manjula

Assistant Professor, (SG), Associate Professor,
School of Computing Science and Engineering, VIT University, Vellore-14
Email: [1]mnarayanamoorthy@vit.ac.in, [2]rmanjula@vit.ac.in

ABSTRACT

The computer architecture is a set of disciplines which describes the organization, implementation and functionality of computer systems. It defines how the internal organization of various components and system is designed and implemented to meet the specified capabilities. Computer architecture involves many aspects which includesInstruction Set Architecture, Micro Architecture, Logic Design etc. A computer architecture simulator is a piece of software to model computer devices to predict outputs and its performance metrics on a given input. These simulators can be an instruction set simulator which models a target Microprocessor or a full system simulator whose model includes a processor, a memory system, I/O devices etc. As on today, the instruction set simulator available in the industry uses only a single type of processor (either 8085 or 8086 and so on). This single instruction set simulator has the limitation that at a given time only one type of microprocessor can be simulated between different log-in (any other limitation). To, overcome this, we propose a new integrated computer architecture simulation framework which integrates all microprocessors on a single IDE environment, for better understanding, development, and programming of architecture.

KEYWORDS: Computer Architecture, Microprocessor, Instruction set Simulator, Usability, Code Compatibility, Future challenges.

1. INTRODUCTION

The design of computer systems started with Single Core Processors [1, 2, 3].Today's Computer Architecture deals with Multicore / Many Core / GPGPU's and Multiprocessor techniques. To understand the working and measure the performance

growth of different advanced Microprocessors many simulators and emulators [4, 6, 8, 9] are used. These simulators and Emulators are software's that simulates the working of a physical CPU. These simulators are used by many Scientists, Academicians, Researchers, Engineers and Students for various purposes like experimenting, practicing, debugging, profiling and testing the real world environments. These simulators allow developers and users to create new virtual CPU which can be programmed by machine or assembly language.

1.1. Microprocessor Introduction

The Microprocessor is a semiconductor device in the IC [1, 2, 5, 6, 10] Form which consists of arithmetic and logical circuits to perform mathematical calculations. Each of the Microprocessors has the following characteristics. They are

(1) Number of Pins and IC Package
(2) Number of Data Lines (Data Capacity)
(3) Number of Address Lines
(4) Memory Support (If n is the number of address lines then the memory support is 2 to power n)
(5) Mode of Operations
(6) Support of cache memory and virtual memory
(7) Speed of Operation
(8) Support of Co-Processor
(9) Instruction set and Addressing Modes

Each Microprocessor has set of instructions to operate. The instruction has two parts like op code and operands. The program for any Microprocessor can be written by machine language or Low Level Languages (Mnemonics). There are software tools like compilers and assemblers and simulators and emulators are used for this purpose. Programming the Processors involves the understanding of processor architecture, programmer's model, instruction sets, addressing modes. The following simulators illustrate the various processor architectures and working models. They are 8085 simulator, 8086 emulator and CPU Simulator.

1.2. Microprocessor Simulator

The emu8086 [7] is an open source Simulator tool used for programming 8086 processor. Here virtual devices can be simulated for real time applications.In the Fig 1 the program for addition of two numbers is written and compiled and executed. The 8086 development environment is shown above which illustrates the text editor for developing the program, the programmer's model of 8086 architecture with various register values, memory locations, and flags conditions. The execution of the program [9] can be done step by step to monitor the various register values. Few other Computer design simulator tools are MM Logic, Spice, Electronic Workbench, Logic Works, and Circuit Maker [11]

1.3. Hypothetical Model of Computer System

Any computer system is designed with a type of Microprocessor. The following notations are used for any computer system. Pa-Program written in Assembly Language
Pb-Program written in High Level Language
Ri (i=1 to n)-General Purpose registers
PC-Program counter
MAR –Memory Address Register
M-Memory
MBR-Memory Buffer Register
IR-Instruction Register
ID-Instruction Decoder
CU-Control Unit
ALU –Arithmetic and Logical Unit

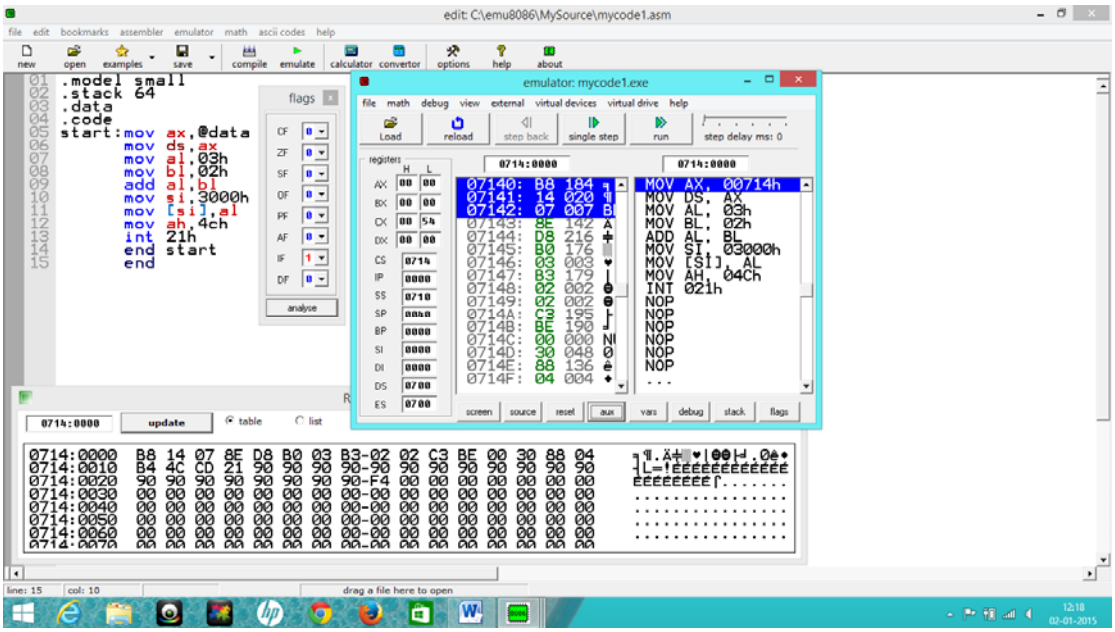

Fig-1:8086 Simulator

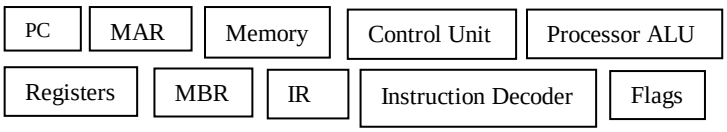

Fig-2: Hypothetical Model of Computer System

2. PROPOSED MODEL

Till 2004 the computer performance is improved by improving the number of transistors on the given chip and increase in clock frequency. But this leads to other

problems such as heat dissipation and power consumption. So the computer designers have looked into the innovations and come out with advanced processors which has multiprocessor or multicore and many core architecture features for improved performance. We need tools, simulators and emulators to predict the performance of these processors. We have many open source tools and simulators to work with earlier processors; the need to develop and propose new tools and simulators is increasing.The future of computer architecture is so complex. It has a mixture of various components, technologies and communication tools. The 21st computer architecture refers not only with desktop systems but also various handheld and wearable devices collecting data through sensors and controlling and activating devices. These devices are more focused on energy minimization models.

Name: Advanced Microprocessor Simulator

Input: Type of Microprocessor, number of pins, data bits, address bits, clock speed, modes of operation, virtual memory, cache memory size, Instruction set Type and classification, types of addressing modes

Output: Program Execution Time, Cache hit, cache miss, Graph illustration of program execution time,

Functions: New program editing, modification of existing programs, error checking and debugging, compile and build options, emulation of hardware and virtual devices. The flow of working with this proposed tool is as follows. The following table Fig-3 illustrates the integrated representation of the processors like 8085,8086,80286 or other processors under the proposed tool framework.

3. IMPLEMENTATION

This tool is implemented and tested using c language. The snap shot of the part of the code snippet is given below.

```
#include<stdio.h>
#include<conio.h>
void main(void)
{       int a;
printf("---------------------------------\n");
printf("  \n");
printf("Integrated tool for Microprocessor Simulation\n");
printf("  \n");
printf("-------------------------------\n");
printf(" List of Microprocessors\n ");
printf("(1) 8085 Microprocessor\n");
printf("(2) 8086 Microprocessor\n");
printf("(3) 80286 Microprocessor\n");
printf("(4) 80386 Microprocessor\n");
printf("(4) 80486 Microprocessor\n");
printf("(5) pentium Processor\n");
printf("(6) Pentium II Processor\n");
```

```
printf("(7) Pentium III Processsor\n");
printf("(8) Pentium IV Processor\n");
printf("(9) Dual core processor\n");
printf("select a processor to simulate\n");
scanf("%d",&a);
switch(a)
{       case 1:
printf("8085 Microprocessor simulation\n");
printf("features of 8085 processor are\n");
printf("General Purpose processors of 8085 are \n");
printf("A --, BC --,DE--,HL--,PC--,SP--,\n");
printf("Memory\n");
printf("00 ---01---02---03---04------\n:);
printf("05---06---07---08---09---10---\n");
printf(" (1) step by step (2) Run \n");break;
case 2:
printf("8086 Microprocessor Simulation\n");
printf("features of 8086 processor are\n");
printf("General purpose processors of 8086 are \n");
printf("AX-- ,BX--,CX---,DX--,SP-,IP----\n");
printf("SI--,DI--,BP--,FLAGS-----\n");
printf("memory\n");
printf("00--01--02--03--04---\n");
printf("05---06--07--08-09-10---\n");
printf("(1) Step (2) Run\n");break;
default:
printf("thanks for choosing this tool");
break; }printf("hallo\n"); }
```

Results and Discussions:

The following step illustrates how to simulate a microprocessor operation. Start →new → Select Microprocessor type and Characteristics→Editor to type program using instruction set→Assemble or Compile → Check for Errors → Run the program → Observe the Output → Modify the processor characteristics → Run the program → Observe the Output

Conclusion and Future Work

In this paper the new integrated tool is proposed. This tool is more helpful for students to understand the working of CPU. The scientist and Design Engineers may improve this tool for comparative analysis of various advanced Microprocessors. The future work involves the implementation of Tool using C / Visual Basic Language. The future work can include the following parameters to be analyzed for the given Microprocessor.

(1) Operating Frequency and Time
(2) Exectuion Speed and time of the given program
(3) Main memory and cache memory size
(4) Power consumption analysis with change in frequency
(5) Cache hit and cache miss rates
(6) FLOPS measurements
(7) Coprocessor simulation
(8) Memory simulation
(9) Cache memory simulation
(10) Virtual device programming simulation environment

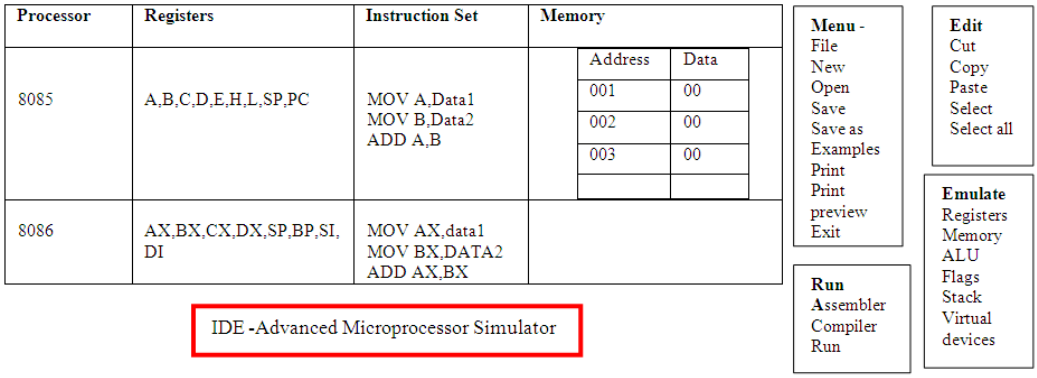

Fig-3: Framework of the proposed Tool

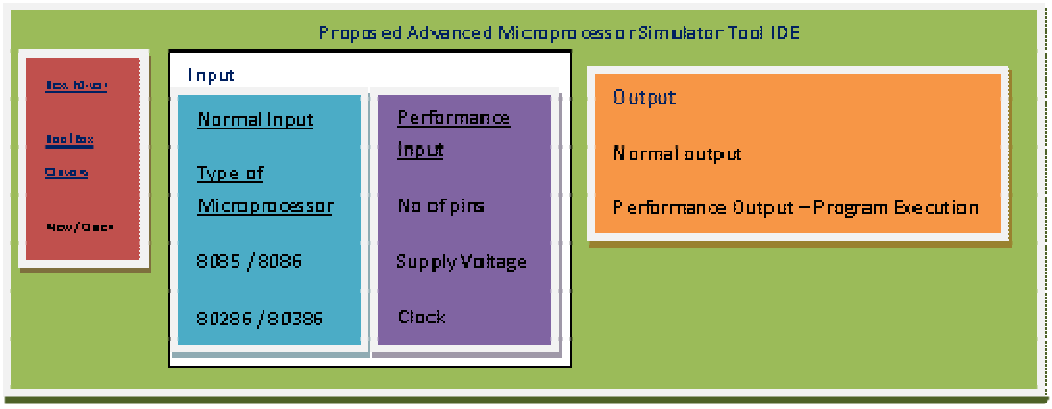

Fig -4: Advanced Microprocessor Simulator IDE

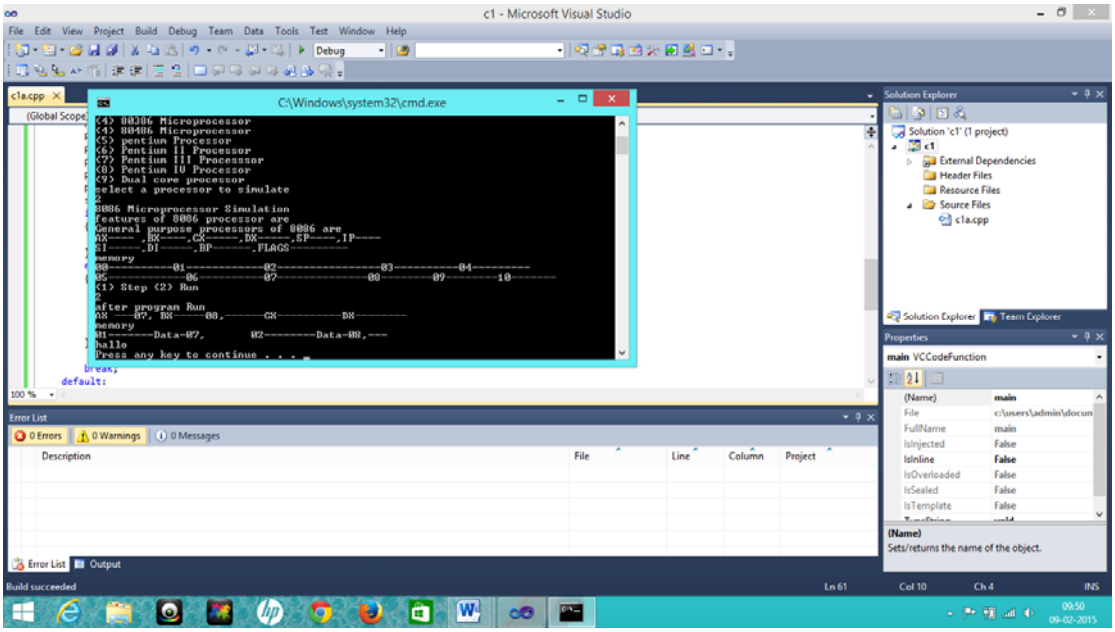

Fig-5: Snap shot of the simulator Tool

6. REFERENCES

[1] J. L. Hennessy, D. A. Patterson, "Computer Architecture: A Quantitative Approach", 2nd Edition, Morgan Kaufmann Publishing Co. 1996

[2] Frank Schirrmeister, "Multi-core Processors: Fundamentals, Trends, and Challenges', Embedded Systems Conference 2007 ESC351, Imp eras, Inc.

[3] Intel@ Multi-core technology, http://www.intel.com/multicore/

[4] Rana W. Alaskar, and Imtiaz Ahmad, "Data Center Architectures: Challenges and Opportunities",Computer Engineering Department College of Computing Sciences and Engineering Kuwait University, P. O. Box 5969, Safat 13060, Kuwait rana.alaskar@hotmail.com, imtiaz.ahmad@ku.edu.kw International Journal of New Computer Architectures and their Applications (IJNCAA) 4(2): 108-116 The Society of Digital Information and Wireless Communications, 2014 (ISSN: 2220-9085)

[5] 21st Century Computer Architecture A community white paper May 25, 2012 computing community consortium (CCC)

[6] David H. Albonesi, "Future Directions in Computer Architecture Research",albonesi@csl.cornell.edu 0272-1732/10/$26.00 _c 2010 IEEE Published by the IEEE Computer Society

[7] www.emu8086.com

[8] AnastasMiˇsev, MarjanGuˇsev, "Simulators for Courses in Advance Computer Architecture",FactaUniversitatis (NI ˇS) SER.: ELEC. ENERG. vol. 18, No. 2, April 2005, 237-252

[9] Martignoni, L., Paleari, R., Reina, A., Roglia, G. F., and Bruschi, D. 2013. “A methodology for testing CPU emulators”.2013 ACM Trans. Softw.Eng. Methodol.22, 4, Article 29 (October 2013), 26 pages. DOI: http://dx.doi.org/10.1145/2522920.25229220]

[10] SameeraA'Amer Abdul-Kader, **“**Emulation of the Microprocessor Intel 80386**”,** Computer Department, College of Education, Diyala University,Diyala Journal of Engineering Sciences**,** Vol. 02 , pp. 13-34 , June 2009

[11] Kenneth R. Vollmar, “Hardware Simulation Tools for Computer Design”,Southwest Missouri State University Springfield, MO 65804KenVollmar@smsu.edu