A Software-based Interactive System on BZK.SAU.FPGA10.1 Micro Computer Design for Teaching Computer Architecture and Organization

Halit Oztekin[1,2], Feyzullah Temurtas[2], and Ali Gulbag[1]

[1]Bozok University, Electrical and Electronics Eng. Dep., Yozgat, Turkey
oztekinhalit@gmail.com, temurtas@gmail.com
[2]Sakarya University, Institute of Science Technology, Computer Eng. Dep., Sakarya, Turkey
agulbag@sakarya.edu.tr

Abstract

An important problem in Computer Architecture and Organization course is the migration from theory to practice. Computer Architecture designs are developed to tackle this problem. This paper presents a learning activity designed to improve motivation towards learning computer architecture and organization subjects, mainly on BZK.SAU.FPGA10.1[1] micro computer architecture design. This software-based interactive system gives a visual presentation of all parts of such a micro computer architecture design; displays the states of registers; performs at the level of clock cycle, an instruction and complete program. So, when learning standard micro computer architecture subjects, students can understand executing a program instruction by instruction at the system level, where the microcomputer is considered as a black box.

1. Introduction

Computer architecture and organization course is one of the main courses in the computer engineering and computer science curricula [4-5]. This course covers the basic concepts like the processor, the memory, the input/output system, ALU (Arithmetic and Logic Unit) and the bus design. Also, it is given the knowledge on how to develop computer architecture with the basic hardware units and how to create coordination between these units in this course. Traditionally, teaching concepts in the computer architecture and organization course to computer and electrical science students can be inefficient if the teaching method is only on textbook materials. One common solution to this state is to provide an active learning environment for teaching computer architecture and organization course by taking advantage of software technology [6]. In other words, it is very important to understand and see the internal behavior and structure of a computer architecture design.

Visualization that is absolutely necessary keyword and idea improves learner's understanding level[7-8]. With tools concerning visualization, many students will understand any subject in computer architecture and organization course in a shorter period than the case without using visualization tool[6-8].

This paper presents learning activity developed for BZK.SAU.FPGA10.1 micro computer architecture design to improve motivation towards learning computer architecture subjects. The software-based interactive system supports animation of instruction execution and allows students to write their own assembly programs and examine values of memory locations and registers. So, the visual presentation of all parts of such micro computer architecture gives

The rest of the paper is organized as follows. The background and motivation section gives short information about the existing tools proposed and used in computer architecture education. Section 3 gives a short overview about BZK.SAU.FPGA10.1 micro computer design. Section 4 concentrates on the software-based interactive system and Section 5 concludes the paper.

2. The Background and Motivation

Computer architecture laboratories often use software simulators to teach the basic computer systems. A variety of educational simulators are designed and developed to support teaching courses in computer architecture and organization [9-18]. These simulators are developed by using software programming languages or emulator programs. In other words, none of them has their own specific hardware. The basic characteristics of selected simulators are shown in Table I.

Table 1. The Basic properties of selected computer architecture simulators[3]

Simulator Name	HW or SW	#Inst.	#Addr. Modes	Int. or Stack
BZK.SAU[3]	Fully HW	59	6	Int. & Stack
Marie Computer by Timothy D.Stanley[9]	SW	13	2	-
Marie Computer by Timothy D.Stanley[10]	Limited HW	13	2	-
p88110[11]	SW	NA	NA	NA
Easy CPU[12]	SW	NA	2	Stack
LittleMan[12]	SW	11	NA	NA
RTLSim[12]	SW	NA	NA	NA
Z80[13]	SW	NA	NA	NA
68000[14]	SW	NA	NA	NA
Simple CPU[15]	SW	8	3	-
Relatively Simple CPU[16]	SW	16	NA	-
Very Simple CPU[17]	SW	4	NA	-
SIMPLE[18]	SW	28	1	-

Int.:Interrupt NA: Not Available HW:Hardware SW: Software

Table 2. The basic properties of FPGA-based microprocessor architecture designs [1]

System Name	HW or SW	#Inst.	#Addr. Modes	Dev. Board
BZK.SAU.FPGA.10.0* [2]	Fully HW	51	6	Altera Cyclone DE2
MOVE[19]	SW	1	4	Xilinx Spartan II
HIP[20]	HW & SW	52	NA	Xilinx Spartan 3
No-Name[21]	SW	8	NA	Vantis MACH211
SIMPLE RISC COMPUTER[22]	HW	16	3	Altera UP2
ASAP-0/f0,f1,f2[23]	SW	10	NA	ASAver.1 (Xilinx XC52156 PQ208)
No-Name[24]	SW	47	7	Xilinx Spartan II
TINYCPU[25]	SW	28	NA	Xilinx Spartan 3A & 3E Starter Kit
Edulent[26]	HW & SW	40	4	Xilinx Spartan IIE
Micro-FIMEE-08[27]	SW	100	4	Xilinx Spartan3 Starter Kit

* available free.

Because of the virtual of designing with this simulator, projects in an introductory computer architecture laboratory are limited. Field-programmable gate arrays (FPGAs) offer a design platform that allows students to implement their projects on actual hardware. More educational applications in reconfigurable design were found at literature in recent years. Table 2 summarizes the basic properties of selected FPGA-based microprocessor architecture designs.

As a result of a critical analysis of both the existing simulators and FPGA-based designs we found that there is a gap between simulators and FPGA-based designs. So we developed the software-based interactive system with communication support. In this way, students can both write their assembly programs on actual hardware and interactively examine values of registers and memory locations.

3. BZK.SAU.FPGA10.1: FPGA-based Micro Computer Architecture Design

The BZK.SAU.FPGA10.1 is the new version of BZK.SAU.FPGA10.0[2]. We have adopted the modular approach to the second FPGA version of the BZK.SAU[3] named BZK.SAU.FPGA10.1[2]. [1] and [2] have the same architecture. The only difference between them the development environments. While the development environment of [2] is FPGA, other is an emulator program.

Modular design is an important factor for the educational training of micro computer architecture. We took the approach of modularization in order to avoid having students be overwhelmed by the complexity of a complete computer system design. Since it has modular nature, students do not have to build a microcomputer architecture from scratch. In this approach, it is quite simple to include their designs like adding unit, subtraction unit etc. to our system. So they can see the operation of their own designs on our system. In other words, our modular approach is ***plug&see***. Therefore, teachers can teach more productive course by applying this approach to their teaching methods since our approach aims to learn piece by piece rather than complete system. All units in BZK.SAU.FPGA10.1 architecture are own specific designs and we built these units using only schematic design at logic level. A detailed block diagram that shows the components of the BZK.SAU.FPGA10.1 design is shown in Fig. 1.

The common features of both versions are listed as the following:

a) Support six addressing modes: immediate, direct, indirect, index, relative and inherent mode.

b) It has eight general and special registers: Address Register(AR), Data Register(DR), Accumulator(AC), Program Counter(PC), Stack Pointer(SP), Index Register(IX) and Temporary Register(TR) are 16-bit; Instruction Register(IR), Output Register(OUTR) and Input Register(INPR) are 8-bit.

c) 16-bit data bus and address bus.

d) The instructions which have relative and index addressing modes need an effective address. So, it has a unit that calculates this address.

e) The instruction set consists of 51 instructions: 21 instructions for memory and accumulator, 8 instructions on index and stack registers, 22 instructions to change the flow direct of program execution.

Instructions that use direct, immediate, indirect addressing modes occupy 3 bytes to execute in the memory. Instructions that use index and relative addressing modes occupy 2 bytes to execute while inherent addressing mode instructions occupy 1 byte to execute. In order to provide readers with the detailed features of instructions and instruction structures, more data are given [1,2,3].

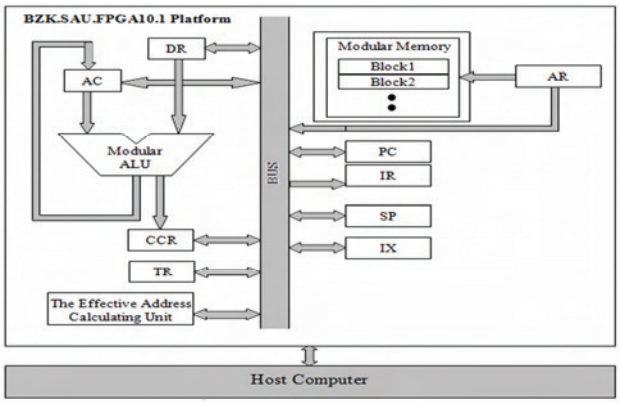

Fig. 1. The Block diagram of BZK.SAU.FPGA10.1

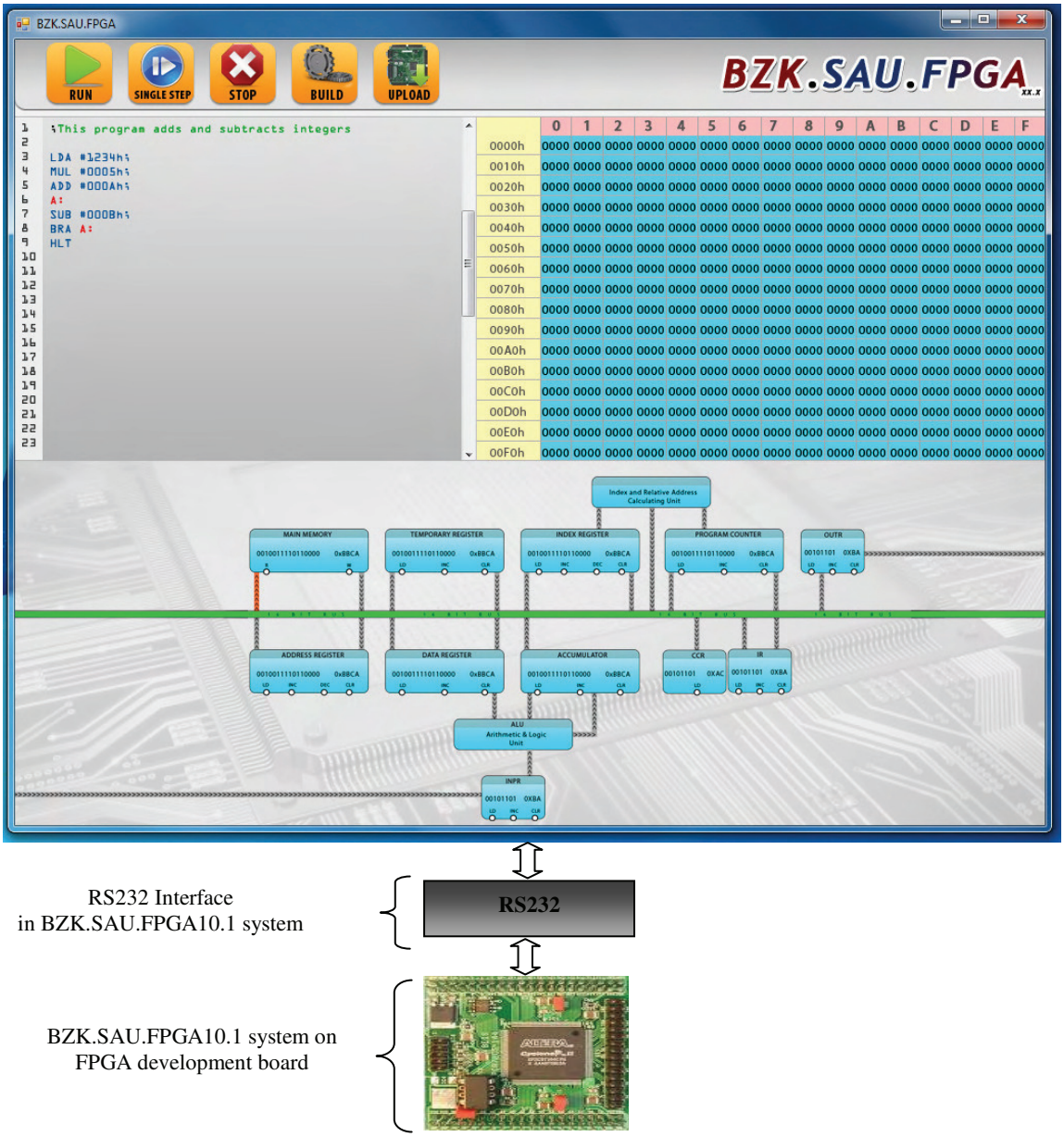

Fig. 2. Overview of Software-based Interactive System with Communication Support to FPGA board

4. The Architecture of Software-based Interactive System developed for BZK.SAU.FPGA10.1

In this section, the overview of our software-based interactive system developed for BZK.SAU.FPGA10.1 is illustrated. This system is designed for an educational tool to demonstrate how the micro computer architecture design called BZK.SAU.FPGA10.1 works visually. This system has four sections: Assembler, the internal view of the 64 KB main memory in BK.SAU.FPGA10.1, control buttons to load, run etc. written program in the Assembler section and the states of register cycle by cycle. Fig. 2 shows an overview of this system.

BZK.SAU.FPGA10.1 works visually. This system has four sections: Assembler, the internal view of the 64KB main memory in BZK.SAU.FPGA10.1, control buttons to load, run etc. written program in the Assembler section and the states of registers cycle by cycle. Fig. 2 shows an overview of this system.

4.1. Assembler Section

(1) of Fig. 2 shows Assembler section of our interactive system. Our assembler and compiler program developed for BZK.SAU.FPGA10.1 microcomputer architecture is written using C# programming language in the Microsoft Visual Studio .NET platform. In this section, students can write their program using our assembly language rules. The following procedure summarizes how to write the program:

Table 3. Used symbols according to addressing mode type

The type of Addressing mode	Symbol
Immediate	#
Direct	$
Indirect	@
Relative	*
Index	%
Inherent	none

Table 4. The Assembly examples according to addressing modes

Immediate addressing mode	ADD #1234
Direct addressing mode	ADD $1234
Indirect addressing mode	ADD @1234
Relative addressing mode	BRA *23
Index addressing mode	ADD %12
Inherent addressing mode	CLR
In use the label	BZR *"LabelName"*

Step-1: If you want to add a description or title to any line, start with the ";" punctuation mark.

Step-2: Obtain the mnemonic name of instruction from Instruction Table in [2] and add addressing mode symbol from Table 3. Then add data and address information in hex format according to addressing mode types. If the label is used, it is enough to write the name of label after the mnemonic name of instruction. The various examples are given in Table 4.

Step-3: Add the ";" punctuation mark to the end of every line except description, title and label lines.

4.2. The Visualization of the Components in BZK.SAU.FPGA10.1 Microcomputer Architecture

(2) of Fig. 2 shows the internal view of main memory(64KB) in BZK.SAU.FPGA10.1. This section is used to show the machine code of assembly program written in section 1 of Fig.2 and to observe the content of memory during the program execution. The assembly program is converted to machine code and loaded to main memory using button named BUILD in section (3) of Fig. 2. We have improved our BZK.SAU.FPGA10.1 microcomputer design by including RS-232 interface to receive and transmit data between our microcomputer design and this software-based interactive system. Using this interface, the content of memory can be uploaded to FPGA development board using button named UPLOAD and the states of registers can be transmitted to this interactive system step by step. The op-code of instruction that is fetched and decoded from memory is in red color to be able to distinguish from others during the execution of program. Users can choose to execute their program in the step-by-step mode or in the continuous one, by clicking buttons named RUN and SINGLE STEP respectively.

(4) of Fig. 2 shows the visualization of registers and bus of BZK.SAU.FPGA10.1 microcomputer system. During program execution, the content of related registers, memory and bus change both in hexadecimal and binary format in every step. In section 4 of Fig. 2, registers and other units have some control inputs, (Increment (INC), Decrement (DEC), Load (LDA) etc. and data input and output symbol. In every instruction cycle, the related control inputs are in blinking red color unlike other units. Also, data input and output symbol between bus and the related unit is in blinking blue color.

5. Conclusions

This paper introduced software-based interactive, educational environment designed to improve student motivation for topics studied in computer architecture and organization, a core course in electrical and computer engineering, and computer science programs. The software-based interactive system provides visualization of BZK.SAU.FPGA10.1 microcomputer architecture system that we developed for the educational purpose. The main advantage of the presented environment that is the FPGA implementation of BZK.SAU.FPGA10.1 microcomputer system that enables by allowing students to send their programs instead of manually, execute them in actual specific hardware and displaying the content of every register and other units step by step. The main disadvantage is that this system does not handle or check system errors for now. In the next version, this lack will be overcome. Since it is available online [28], the students can implement the complete BZK.SAU.FPGA10.1 on an FPGA chip. Next, they write their programs using our developed software-based interactive system, run on it, and observe the step by step program execution in section 4 of developed software-based interactive system. So students can observe step-by-step changes in the registers and other units by adjusting clock speed of BZK.SAU.FPGA10.1 system.

6. Acknowledgement

This work was fully supported under TUBITAK (The Scientific and Technological Research Council of Turkey) project no. 110E069.

7. References

[1] H.Oztekin, F. Temurtas, A. Gulbag, "BZK.SAU.FPGA10.1: A Modular Approach to FPGA-based Micro Computer Architecture Design for Educational Purpose", *Computer Applications in Engineering Education*, Online available: http://wileyonlinelibrary.com/journal/cae; DOI:10.1002/cae.20553.

[2] H. Oztekin, F. Temurtas, A. Gulbag, "BZK.SAU.FPGA10.0: Microprocessor Architecture Design on Reconfigurable Hardware as an Educational Tool", *IEEE Symposium on Computers and Electronics (ISCI 2011)*, Kuala Lumpur, 2011, pp. *385-389.*

[3] H. Oztekin, F. Temurtas, A. Gulbag, "BZK.SAU: Implementing a Hardware and Software-based computer Architecture simulator for educational purpose", *International Conference on Computer Science and Applications (ICCDA 2010)*, Quinhuangdao, 2010, vol. 4, pp. *90-97.*

[4] "Joint Task Force on Computer Engineering Curricula, Computer Engineering Curriculum Guidelines for Undergraduate Degree Programs in Computer

Engineering 2004", Dec.12, 2004. Online available: http://www.computer.org/education/cc2001/. [Accessed: Nov. 20, 2008]

[5] "Joint Task Force on Computing Curricula, Computing Curricula Final Report, 2001", Dec.15, 2001. Online available: http://www.computer.org/education/cc2001/. [Accessed: Nov. 20, 2008]

[6] W. Yurcik, G. S. Wolffe, M. A. Holliday, "A Survey of Simulators Uses in Computer Organization/Architecture Courses", *Proceedings of the 2001 Summer Computer Simulation Conference(SCSC)*, Orlando, 2001, pp. *176-180.*

[7] M. Llamas, L. Anido, M.J. Fernandez, "Simulators over the Network", *IEEE Transaction on Education*, vol. *44,* no. *2,* pp. *212,* May., 2001.

[8] L. Null, J. Lobur, "MarieSim: The MARIE Computer Simulator", *ACM Journal of Educational Resources in Computing (JERIC)*, vol. *3,* no. *2,* pp. *1-29,* Jun., 2003.

[9] D. Colton, G. Embrey, L. Fife, S. Mikolyski, D. Prigmore, T. D. Stanley, "Pedagogic value in understanding computer architecture of implementing the Marie Computer from Null and Lobur in the logic emulation software, multimedia logic", *Workshop on Computer Architecture Education(WCAE 07),* pp. *66-71,* Jun., 2007.

[10] D. Colton, G. Embrey, L. Fife, S. Mikolyski, D. Prigmore, T. D. Stanley, "From Archi Torture to Architecture: Undergraduate Students Design and Implement Computers Using the Multimedia Logic Emulator", *Computer Science Education*, vol. *17,* no. *2,* pp. *141-152,* Jun., 2007.

[11] M. I. Garcia, S. Rodriguez, A. Perez, A. Garcia, " p88110: A Graphical Simulator for Computer Architecture and Organziation Courses", *IEEE Transaction on Education*, vol. *52*, no. *2*, pp. *248-256*, May. 2009.

[12] C. Yehezkel, W. Yurcik, M. Pearson, D. Armstrong, "Three Simulator Tools For Teaching Computer Architecture: Easy CPU, Little Man Computer and RTLSim", *Journal of Educational Resources in Computing(JERIC)*, vol. *1,* no. *4,* pp. *60-80,* Dec., 2001.

[13] H. B. Diab, I. Demashkieh, "A computer-aided teaching package for microprocessor systems education", *IEEE Transactions on Education*, vol. *34,* no. *2,* pp. *179-183,* May., 1991.

[14] W. D. Henderson, "Animated models for teaching aspects of computer system organization", *IEEE Transactions on Education*, vol. *37,* no. *3,* pp. *247-256,* Aug., 1994.

[15] J. R. Arias, D. F. Garcia, "Introducing Computer Architecture Education in the first course of computer science career", *IEEE Computer Society,* pp. *37-40,* Jul., 1999.

[16] J. Carpinelli, T. Zaman, "Instructional tools for designing and analyzing a very simple CPU", *International Journal of Electrical and Engineering Education*, vol. *43,* no. *3,* pp. *261-270,* Jul., 2006.

[17] J. D. Carpinelli, "The Very Simple CPU Simulator", *32nd Annual Frontiers in Education(FIE 02),* vol. *1,* pp. *T2F11-14,* 2002.

[18] T. Kawamura, Y. Kawaguchi, S. Nakanishi, K. Sugahara, "Machine Cycle CPU Simulator for Educational Use based on Squeok Environment", *First Conference on Creating Vonnecting and Collaborating through Computing,* pp. *120-121,* 2003.

[19] V. Guštin, P. Bulić, "Learning computer architecture concepts with the FPGA-based "Move" microprocessor", *Computer Applications in Engineering Education*, vol. 14, no. 2, pp. *135-141*, Jul., 2006.

[20] P. Bulić, V. Guštin, D. Šonc, A. Štrancar, "An FPGA-based integrated environment for computer architecture", *Computer Applications in Engineering Education*, Online available: http://wileyonlinelibrary.com/journal/cae; DOI:10.1002/cae.20448.

[21] V. Guštin, "Designing the microprocessor with Abel-HDL", *Computer Applications in Engineering Education*, vol. 9, no. 2, pp. *87-92*, Jan., 2001.

[22] B. Hatfield, M. Rieker, L. Jin, "Incorporating Simulation and Implementation into Teaching Computer Organization and Architecture", *35th. ASEE/IEEE Frontiers in Education Conference*, pp. *F1G-18*, Oct., 2005.

[23] H. Ochi, "ASAver.1: An FPGA-Based Education Board for Computer Architecture/System Design", *Design Automation Conference 1997*, pp. *157-165*, Jan., 1997.

[24] X. Tiejun, L. Fang, "16-bit Teaching Microprocessor Design and Application", *Proceedings of 2008 IEEE International Symposium on IT in Medicine and Education*, pp. 160-163, 2008.

[25] K. Nakano, Y. Ito, " Processor, Assembler and Compiler Design Education using an FPGA", *14th IEEE International Conference on Parallel and Distributed Systems*, pp. *723-728*, 2008.

[26] I. Mazei, V. Malbasa, "Using VHDL to Improve an FPGA Based Educational Microcomputer", *EUROCON 2005*, pp. *799-802*, 2005.

[27] R. de J. Romero-Troncoso, A. Ordaz-Moreno, J. A. Vite-Frias, A. Garcia-Perez, "8-bit CISC Microprocessor Core for Teaching Applications in the Digital Systems Laboratory", *IEEE International Conference on Reconfigurable Computing FPGA's*, pp. *1-5*, 2006.

[28] "BZK.SAU.FPGA10.1", Available: http://eem.bozok.edu.tr/database/1/BZK.SAU.FPGA10.1.rar