Indian Journal of Science and Technology, *Vol 9(48), DOI: 10.17485/ijst/2016/v9i48/97653, December 2016*
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

JAVA based Virtual Lab for Embedded Processor Design

M. K. Srilekha*, P. Ponnammal and R. Bhakkiyalakshmi

Department of Electronics and Communication, SRM University, Kattankulathur, Chennai - 603203, Tamil Nadu, India; srilekha.m@ktr.srmuniv.ac.in, ponnammal.p@ktr.srmuniv.ac.in, bhakkiyalakshmi.r@ktr.srmuniv.ac.in

Abstract

Objectives: In the emerging trends of engineering, experimentation plays a vital role in understanding the ideas. Learning engineering is continually tested as hypothetical study ought to be increased with trials, so the students get a feel of genuine uses of innovative ideas they concentrate on. **Methods**: This paper proposes a JAVA based virtual lab for embedded processor. In this paper, virtual models are introduced as educating/learning tools for the specific idea on ARM processor. **Findings:** This tool is outlined utilizing JAVA NetBeans. NetBeans alludes to both a stage system for Java desktop applications for creating with Java, JavaScript, C, C++. The Design method includes the specific movement of processor like tending to modes, Instruction execution, ALU operation, Memory Management Unit (MMU). **Application:** Students could design, develop and implement digital networks and control units using simulation environment given in the virtual lab platforms by themselves.

Keywords: ARM Processor, Embedded Processor, IDE, Translation Table, Virtual Lab

1. Introduction

The clarification of inner computer operation is a troublesome errand. Low-level computing construct programming finds out about the inner structure of computer. There are different devices accessible for the improvement of low-level computing construct programs. These sorts of devices were created. Representation based reenactment is generally utilized as a part of the sciences to help learners comprehend the system[1]. Moreover, as computer frameworks turn out to be more intricate, perception apparatuses are utilized to manage the many-sided quality and expansion the rate of the execution procedure[2] . This paper shows a JAVA based virtual environment for ARM processor which gives the configuration strategy required in guideline execution, ALU operation, memory administration and port subtle elements. Plan information way with clock strategy and information way and outline control flags and control unit.

2. Virtual Workbench

Virtual workbench a visualization tool developed using JAVA. It provides a virtual environment to study internal architecture of a processor as step by step procedure in the execution of instruction set[4,5]. Moreover, it provides the port details of ARM processor. This JAVA based virtual lab provides

(1) An assembly code
(2) Visualization of the load process
(3) Visualization of the memory references
(4) Different execution modes

3. Virtual Lab Design

This virtual lab outlined utilizing NET BEANS. NetBeans gives a stage system to Java desktop applications, and a Integrated Development Environment (IDE) created with Java, JavaScript, C, C++. The NetBeans IDE utilizes Java and runs all around on JVM including Windows, Mac OS, Linux, and Solaris. The NetBeans Platform use secluded

**Author for correspondence*

programming to create applications. The Netbeans stage has a reusable system to streamline the improvement of Java Swing desktop applications[6]. The stage offers administrations which are reused for desktop applications, permitting engineers to concentrate on a particular application. Figure 1 demonstrates JAVA Netbeans manager.

The components of this stage are client interface administration, client settings administration, stockpiling administration, Window administration, Wizard system, and Netbeans visual library.

3.1 Branch Instruction

Branch instructions are utilized to call a subroutine and change the flow of execution. This sort of instruction permits program to call subroutine and loop. The execution stream change constrains the Program Counter to indicate another location. The ARMv5E instruction set has four different branch instruction. Fig 4 indicates Branch instruction group

Syntax: B 2010

Figure 2 depicts the branch instructions of ARM processor. The branch instruction offset is calculated by the assembler. This guideline takes the distinction between the branch instruction and the objective location to take into account the pipeline which gives a 26-bit offset right shifted 2 bits. The last two bits are constantly zero as instruction are word adjusted and encoded. This gives a scope of 32 Mbytes. When branch instruction executed the L bit is equivalent to one which branch to the area 2010.

Figure 3 shows the instruction format for Branch with exchange. At the point when executing this instruction, the processor shifts the offset left by two bits; sign extended to 32 bits and added with the program counter. Once the pipeline is refilled, the execution proceeds from the new program counter[3]. The "Branch with link" instruction utilizes a subroutine by composing program counter-4 into the link register of the present bank register.

3.2 Memory Management Unit

The ARM MMU makes an interpretation of virtual locations into physical locations, it controls memory access consent, and it determines the individual behavior of the cache and writes buffer for each page in memory. At the point when MMU is debilitated, every virtual location outlines same physical location. In the event that the MMU can't decipher a location, it produces a special case. The MMU will prematurely end on interpretation, consent, and area issues. The fundamental control parts in the MMU are

- Page table
- Translation Lookaside Buffer (TLB)
- Cache and write buffer
- The CP15:c1 control register
- The Fast Context Switch Extension

Figure 4 demonstrates the First level fetch translation table for Memory administration unit of ARM processor.

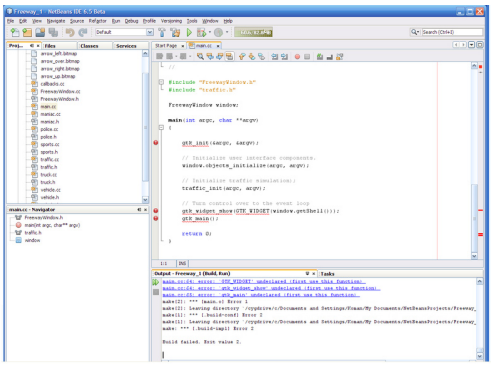

Figure 1. Netbeans editor.

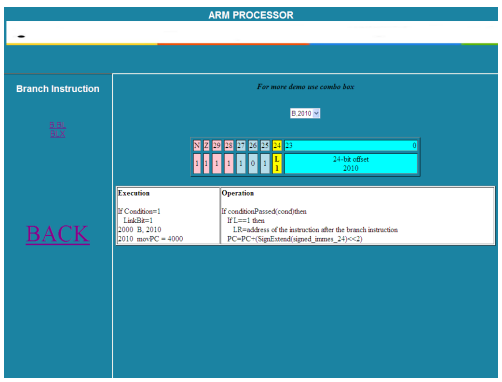

Figure 2. Branch instruction format.

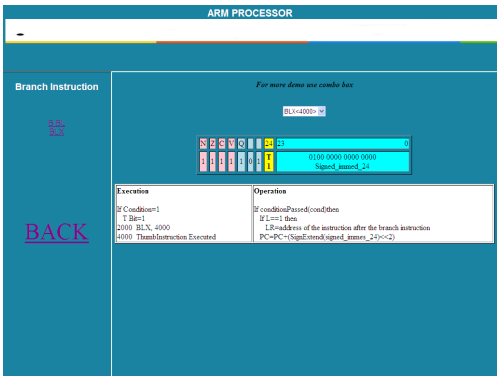

Figure 3. Branch with exchange instruction.

The MMU of ARM processor executes the equipment page table strolling system for ARMv5 reserved processors[7,8].

The processor actualizes the page table walk impair the component. Two bits PD0 and PD1, are utilized as a part of the Translation Table Base Control register. At the point when a TLB miss happens, the TLB processes the parameters for programmed equipment page table walk. The location of the page table walk is processed from TTB0 or TTB1. The ARM contains two Translation Table Base Registers, TTBR0, and TTBR1 and a Translation Table Base Control Register (TTBCR).

At the point when a TLB miss, the bits of the changed virtual location figure out if the first or second Translation Table Base utilized. Figure10 indicates first-level descriptor address.

Figure 5 shows the format of a second level descriptor. In the event that bits [1:0] of the principal level descriptor bits are b0, then a page table walk is required. The MMU asks for the second-level page table descriptor from outer memory. When the page table location is produced, a solicitation is made to outer memory for the second-level descriptor

4. Port Details

In Figure 6 bit 6, 9, 18, 25, 31 is enabled whose hex code is 0x82040240 whose p0.0, p0.6, p0.13, p0.22 and p0.25 are enabled which shows port details used in ARM processor

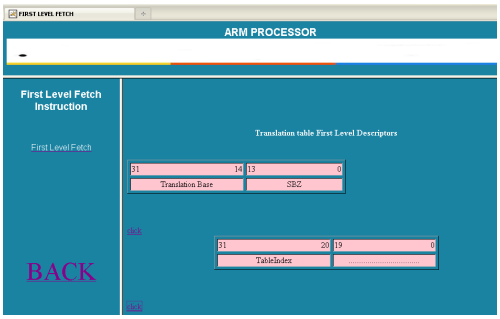

Figure 4. Translation table first level fetch.

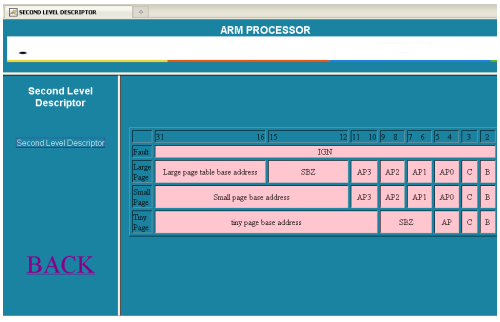

Figure 5. Second level descriptor format.

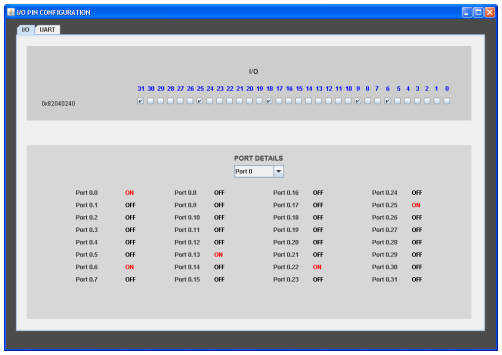

Figure 6. I/O ports in ARM processor.

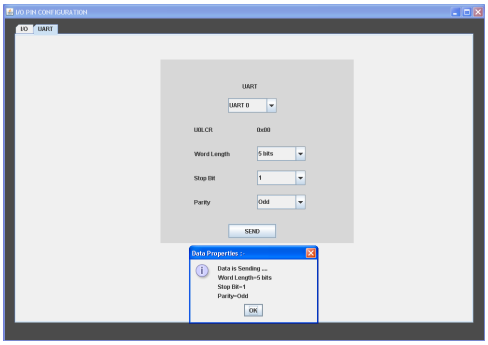

Figure 7. UART of ARM processor.

Figure 7 shows UART information like word length, stop bit and parity of ARM processor. There are two UART used in ARM processor with stop bits and parity bits set.

5. Conclusion

In this paper JAVA based virtual lab is setup for ARM processor. JAVA is a powerful language for modeling formal specifications of hardware systems. Visualizations with graphical interfaces provide a view on internal operation of a computer. The visualization techniques are used to learn concepts associated with assembly language programming and CPU operation. In this paper color changes and animations are widely used to illustrate data flow during execution of a single instruction. This paper can be enhanced by improving ARM port, dynamically changing Thumb code and ARM code.

6. Acknowledgement

This research was supported by Dr. P. Vanaja Ranjan Prof. Department of Electrical and Electronics College of Engineering Guindy Anna University, Chennai who provided expertise that greatly assisted the research.

7. References

1. Theja PR, Babu SKK. An adaptive genetic algorithm based robust QOS oriented green computing scheme for VM consolidation in large scale cloud infrastructures. Indian Journal of Science and Technology. 2015 Oct; 8(27).
2. Mandekar AV, Chandramouli K. Centralization of network using openflow protocol. Indian Journal of Science and Technology. 2015 Jan; 8(S2):165–70.
3. Zhou Q, Zhang Y, Sun H, Wu T, Yang M, Zhou R, Chou W, Li K-C. The design and implementation of embedded online laboratory. 44th International Conference on Parallel Processing Workshops (ICPPW); 2015. p. 86–9.
4. Larraza-Mendiluze E, Garay-Vitoria N. Approaches and tools used to teach the computer input/output subsystem: A survey. IEEE Transactions on Education. 2015; 58(1).
5. Rodriguez-Sanchez MC, Torrado-Carvajal A, Vaquero J, Borromeo S, Hernandez-Tamames JA. An embedded systems course for engineering students using open-source platforms in wireless scenarios. IEEE Transactions on Education. 2016; PP(99).
6. Han AH, Hwang Y-S, An Y-H, Lee S-J, Chung K-S. Virtual ARM platform for embedded system developers. International Conference Conference on Audio, Language and Image Processing ICALIP; 2008. p. 586–92.
7. Torroja Y, López A, Portilla J, Riesgo T. A serial port based debugging tool to improve learning with Arduino. Conference on Design of Circuits and Integrated Systems (DCIS); 2015. p. 1–4.
8. Nikolic B, Radivojevic Z, Milutinovic JDV. A survey and evaluation of simulators suitable for teaching courses in computer architecture and organization. IEEE Transactions on Education. 2009; 52(4):449–58.