ARM MICROPROCESSOR SOFTWARE BASED EMULATOR

SUNIL SHASHIKANT GATHANI

A project report submitted in partial fulfilment of the
requirements for the award of the degree of
Master of Engineering (Electrical - Computer and Microelectronic System)

Faculty of Electrical Engineering
Universiti Teknologi Malaysia

JANUARY 2013

Specially dedicated to my family, lecturers, fellow friends and those who have guided and inspired me through my journey of education

ACKNOWLEDGEMENT

In preparing this thesis, I was in contact with many people, researchers, academicians, and practitioners. They have contributed towards my understanding and thoughts. In particular, I wish to express my sincere appreciation to my thesis supervisor and mentor, Dr. Usman Ullah Sheikh, for his willing encouragement, unstinting guidance, and friendship.

I am also indebted to Intel Microelectronics for funding my MSc study and my fellow postgraduate students who have selflessly supported and enabled the pursuit of this dream. My sincere appreciation also extends to all my colleagues and others who have provided assistance at various occasions. Finally, the bedrock of my success, I am grateful to all my family members.

ABSTRACT

With the recent explosion of devices driving “smart technologies” such as tablets, phones, in-vehicle infotainment systems, and many such devices, ARM has taken center stage in being the core of choice for many such device vendors. Thus the appreciation and workings of the ARM core has become more relevant than ever. In light of that fact, over the years, many emulators have been designed with the intent of emulating the ARM core on a software paradigm. Software based emulation lends itself to many uses, from early application validation to an educational tool for the masses. Hence, this work has emulated the ARM instruction set based on the ARM 7 core. With the objective to enable an extensible and modular design, the framework was developed by designing classes for certain core components which can be replicated as objects and encapsulating execution based entities into functions. The final result of this project is the development of a mechanism for updating the CPSR for each instruction, alongside 16 Data Processing instructions with rotational and register shifting support, all aspects of single data transfer load and store, positive and negative branching with and without link alongside 16 conditional code evaluation, and all User Mode visible registers. The ARM emulator also supports both normal assembler instruction and conditional code instructions in both 2 and 3 operand format. The emulator was verified using single instructions and the GCD conditional code instruction as a program.

ABSTRAK

Dengan perkembangan peranti-peranti baru yang digelar “teknologi pintar” seperti tablet, telefon, sistem hiburan dalam kenderaan, dan banyak peranti sedemikian, ARM telah menjadi pilihan pertama untuk kebanyakan pembekal peranti. Oleh demikian, perhargaan “ARM” telah menjadi lebih relevan berbanding sebelum ini. Memandangkan kepentingan ARM kini, banyak perisian emulator telah dibentuk dengan tujuan mencontohi teras ARM. Emulasi berasaskan perisian mempunyai banyak kegunaan, dari pengesahan awal applikasi-applikasi sistem kepada alat pendidikan untuk orang ramai. Oleh itu, tujuan penyelidikan ini adalah untuk membentuk perisian emulator berasaskan teras ARM7. Dengan objektif untuk membolehkan reka bentuk yang mudah diubah-suai dan modular, rangka kerja ini dibangunkan dengan kelas-kelas and fungsi-fungsi untuk komponen teras tertentu yang boleh digunakan sebagai objek atau sebagai entiti modular. Hasil akhir projek ini ialah pembangunan mekanisme bagi mengemaskini CPSR bagi setiap arahan, di samping 16 arahan pemprosesan data dengan sokongan putaran dan peralihan, semua aspek pemindahan data seperti capai dan simpan, percabangan positif dan negatif dengan dan tanpa penghubung, bersama-sama 16 penilaian kod bersyarat, dan semua register Mod Pengguna. Emulator ARM juga menyokong kedua-dua arahan ARM normal dan arahan ARM bersyarat dalam format dua dan tiga kendalian. Emulator telah disahkan menggunakan arahan tunggal dan arahan program GCD dengan kod bersyarat.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ARM	-	Acorn RISC Machines
CISC	-	Complex Instruction Set Architecture
CMN	-	Compare Negative
CMP	-	Compare
CPSR	-	Current Program Status Register
FIQ	-	Fast Interrupt Request
GCD	-	Greatest Common Divisor
IRQ	-	Interrupt Request
I/O	-	Input / Output
LD	-	Load
OOP	-	Object Oriented Programming
OS	-	Operating System
PC	-	Program Counter
RISC	-	Reduced Instruction Set Architecture
Rs	-	Source Register
Rd	-	Destination Register
Rn	-	n-Operand Register
SPSR	-	Saved Program Status Register
STR	-	Store
TEQ	-	Test if Equal
TST	-	Test
X86	-	x8086 Intel Instruction Set
.NIX	-	Unix based OS

CHAPTER 1

INTRODUCTION

This project is about the software emulation of the ARM 7 Core processor. This chapter gives an overview of the whole project, starting with a brief introduction and the background, followed by the problem statement, project objectives, scope of work and report outline.

1.1 Background

The development of microprocessors with the predictions of Moore Law, has followed a path with consistency such that no one person predicted the lasting of this law. Almost a quarter of a century later, the battles at the heart of the microprocessor, is very much what this project about.

The heart of any processor is dependent upon the computer architecture which defines it. Hence drawing further from that, the computer architecture drives innovation in computing from software and the layers that it comprises off to hardware and the advancements that it has made from the crude old days of Intel x8086. Today the most populous instruction set is the x86 Instruction Set which drives the x86 platform, or the Intel platform as it is known. In most mainstream computing devices (laptops, desktops and servers), Intel clearly dominates the market significantly.

But a diminutive company, ARM Holdings, is still the clear leader in the number of devices that carries its microprocessor. So much so that, the combined sales of microprocessors of all other companies, still does not match the almost ubiquitous presence of ARM in portable / mobile and embedded systems.

While plainly at first sight, the battle of dominance seems very much an architecture battle between ARM (RISC) and Intel(CISC), the determining battle of architectures is of the power consumption that each microprocessor sips. ARM Cores are known to be highly thrifty on their power consumption which is the reason why it has an overtly dominating presence in embedded markets and portable consumer device.

Intel on the other hand comes from a position of computing prowess. Till recently in its history, very little was made in the way of reducing power consumption as their markets were power agnostic.

Moving to an age of portable/mobile devices where battery life is a key selling matrix, Intel needs to drive innovations to their architecture as to drive power consumption down. On the other hand, ARM is trying to muscle its way to desktop, laptop and server products. And in doing so, power consumption for it architecture clearly would rise. Naturally both grapple with the same physics of power consumptions.

Thus with the advent of ARM based devices, understanding the nuances of the architecture or utilizing the emulator for application development or many such uses, is tremendously beneficial for product development point of view or as a learning tool. Unfortunately over the very many years, the most matured platform of development of ARM based emulators has been the UNIX platform. Moreover, the Intel Platform based ARM emulators, mostly tends to be non-open source, and thus limited by the exorbitant licensing fee. In light of this, the cost benefit value of designing a home grown ARM Emulator, will allow a more cost prudent option for

users of this Emulator while providing all the benefits without executing application on an ARM Core.

1.2 Motivation

Having laid the groundwork on the perspective and importance of ARM in the micro-architectural battle, a tool is needed to provide easier access and understanding of the architecture and a simpler way to build a software stack which readily runs on ARM cores. That is the development which defines this project.

The heart of this project is to develop an emulator which would readily allow ARM based applications and OS'es to run above it. The emulator simply, functions to mimic the behavioral aspect of an ARM core.

While, naturally one expects a significant number of ARM emulators already present for educational and development purposes; there are significant impediments for each of currently available emulators. All ARM emulators currently available are not natively developed in the Microsoft family of OS'es. This manifests itself in a situation whereby the applications or OS'es running on emulators non-natively developed in Microsoft OS tend to have significant drawbacks. Hence, the motivation behind ARM Emulator is that not only does it have to be developed, it has to be developed natively on Microsoft OS.

This ensures that the ARM development vehicle for application and OS'es can be natively developed on Windows.

1.3 Objectives

The objective of the ARM emulator project is to develop an emulator that would be able to functionally/behaviorally emulate the function of an ARM core that would be designed to work natively on Microsoft Windows OS.

Hence the emulator should:

i. Accept assembly set instructions

ii. Decode those instructions

iii. Execute each instruction with the end goal of producing result exactly per an actual ARM core

The secondary goal would be to enhance the emulator to run in the most adept way on Windows OS.

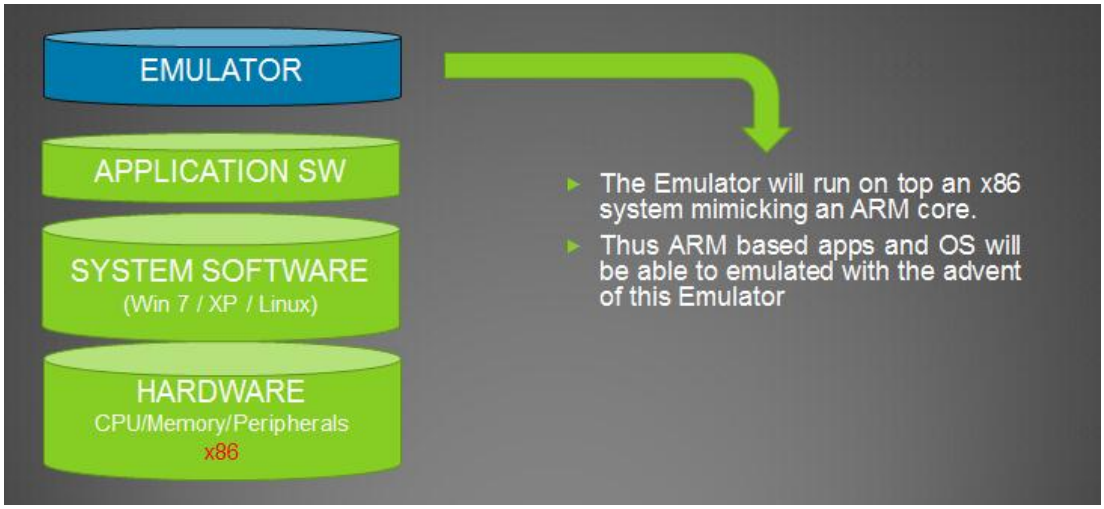

Figure 1.1: The ARM emulator would sit on top of system application software and functionally replicate behavior of the ARM core.

1.4 Problem Statement

The current issue at helm is that, ARM emulators which successfully execute on an x86 Microsoft Windows platforms are very limited. More so, the ones that are available, are generally not free, and thus require licensing. Thirdly, the freely available ARM emulators, suffer from limited support on debug and more so in documentation.

Generally the most mature development platforms for ARM emulators run on the .NIX OS platforms.

1.5 Scope of Work

There are 5 key considerations which would define the scope of work:

i. Firstly, would be the implementation language of choice. The language will determine the ease of integration of different modules and the speed at which the emulation takes place.

ii. Secondly, there are multiple ARM cores in the market today. Choosing the right ARM core would determine the scope of development without trivializing or unnecessarily bloating the development phase.

iii. Thirdly would be choosing which instruction set would need emulation. ARM7 onwards, the ARM cores support the regular instruction set, along with a condensed instruction set called THUMB which would also be present in all future revision of ARM cores. The current proposal is to emulate the regular instruction set.

iv. Fourthly, the emulation which would take place can be generically categorized as either Instruction Accurate [Functional Accurate], Cycle Accurate or Timing Accurate emulation. The ARM emulator would be designed as the Functional emulation of an ARM Core.

v. Finally, the question of which parts of the ARM core would be emulated in the short time frame of development and which can be developed later and yet be integrated into the ARM emulator is to be determined.

REFERENCES

1. Akihiro Suzuki and Shuichi Oikawa. Implementation of Virtual Machine Monitor for ARM Architecture. *10th IEEE International Conf. On Computer and Information Technology*. June 29, 2010. Bradford : IEEE. 2010. 2244-2249.
2. Harold W. Cain, Kevin M. Lepak, and Mikko H. Lipasti . A Dynamic Binary Translation Approach to Architectural Simulation. *International Conference on Parallel Architechtures and Compilation Techniques*. Oct. 15, 2000. Philadelphia. 2000
3. Mingsong Lv, Qingxu Deng, Nan Guan and Yaming Xie, Ge Yu. ARMISS: An Instruction Set Simulator for the ARM Architecture. *The 2008 International Conf. on Embedded Software and Systems (ICESS2008).* August 12, 2008. Sichuan : IEEE. 2008. 548-555.
4. Peter Knaggs and Stephen Welsh. *ARM Assembly Language Programming*. Bournemouth University, United Kingdom : 2004
5. F. Bellard, QEMU, A fast and portable dynamic translator. *Proceedings of the annual conference on USENIX Annual Technical Conference*, FREENIX Track, pages 41-46, 2005.
6. "Instruction Set Simulator" Internet : http://en.wikipedia.org/wiki/Instruction_set_simulator, [Dec. 20th, 2012].
7. "C++ Language Tutorial". Internet : http://www.cplusplus.com/doc/ , [Jan. 8th, 2013].
8. Joel Pobar. "Create a Language Compiler for the .NET Framework". Internet: http://msdn.microsoft.com/en-us/magazine/cc136756.aspx, February 2008 [Dec. 1st, 2012]
9. "Console C++ Video Tutorials". Internet : http://xoax.net/cpp/crs/console/index.php [Aug. 10th, 2012]

10. Stever Furber. *System On Chip Architecture*. 2nd ed. United Kingdom.: Addison-Wesley Longman. 2000.
11. Advanced RISC Machines Ltd. (ARM). *Realview ARMulator ISS User Guide*. ARM DUI 0207A Version 1.3. United Kingdom. 2002
12. Maray Fayzullin. "How to write a computer emulator?". Internet : http://fms.komkon.org/EMUL8/HOWTO.html [Aug. 10th, 2011]
13. Richard Danter, "Emulation, Simulation and Native Development". Internet : http://www.electronicsweekly.com/blogs/open-source-linux/2009/03/emulation-simulation-and-native-development.html. March 2009 [Nov. 1st, 2011]
14. "Free ARM Emulators". Internet : http://www.thefreecountry.com/emulators/arm.shtml [Aug. 5th, 2011]
15. Advanced RISC Machines Ltd (ARM). *ARM 7TDMI Technical Reference Manual.* ARM DDI 0210C. Revision r4p1. United Kingdom. 2004.
16. Advanced RISC Machines Ltd (ARM). *ARM 7100 Preliminary Data Sheet.* ARM DDI 0035A. United Kingdom. (1996),.
17. Bruce Eckel. *Thinking in C++*. 2nd ed. Vol. 1, Upper Saddle River, New Jersey. : Prentice Hall Inc. 2000.
18. Walter Savitch. *Absolute C++*. 5th ed. University of California, California. : Pearson. 2012.
19. Dr. Usman Ullah Sheikh. *Advanced Microprocessor Systems Lecture Notes*. Universiti Teknologi Malaysia (UTM), Skudai, Johor. 2011-12