Evaluation of the NESizer2 method as a means of wrapping embedded legacy systems

Contents

1 Introduction

When integrating legacy information system components into a modern system, one of several usual approaches is to create an interface for the modern system to control or communicate with the legacy component. This approach is known as wrapping.[1], [2] Wrapping as a concept could be adapted for legacy hardware in embedded systems, to enable the original and proven functionality of the outdated system, by providing an interface to control the legacy components. There are few documented examples of migration of legacy systems that include the incorporation of the system including the hardware platform, and the few examples that exist are often designed with a specific functionality in mind. This thesis will evaluate how one of these methods could be expanded upon to allow for a general use case of the legacy component, and to evaluate how well the method performs as a means of wrapping.

1.1 Background

Computer based information systems are an invaluable asset for modern enterprises and corporations. The use of information systems can range from data storage, analysis and organization, to communication systems such as mail servers. With continued updating and upgrading of these systems they become increasingly complex, and as technology evolves the existing technology of the systems are quickly rendered obsolete, and "eventually, the existing information systems become too fragile to modify and too important to discard" [3], and as such it can be considered a legacy system. [4] At this point the legacy systems must be modernized, or migrated into a more modern system to remain viable.

When incorporating legacy information systems into modern systems, there are usually three popular approaches: redevelopment, wrapping and migration [1, pp. 2–4]. While redeveloping an entire system is usually the best option in the long run, it is also the most expensive and risky. Therefore, migration is usually a more popular method, as it provides an interface to control the legacy components, while retaining its data and functionality. However, migrating systems to a modern platform can lead to unexpected behaviour, with a notable example being NASA's Ariane 5 flight 501 [5], [6].

When redevelopment and migration is too risky or expensive, wrapping offers a cost-effective option with less risk. It surrounds existing data, systems and interfaces with new interfaces, giving the legacy components a "new and improved" look [1, p. 3], and lets organizations reuse trusted components in a new, more up-to-date manner. While these methodologies and frameworks usually focus on legacy software systems, they are very much applicable to hardware systems and components as well.

1.2 Problem

To address the issues of unexpected behaviour in an otherwise proven, well functioning system, the solution could be to keep only the crucial legacy components including their hardware platforms, and provide an interface for a modern system to control them - creating a wrapper for both hardware and software. While there exists implementations of similar approaches, they are often designed with a specific functionality of the legacy component in mind, and as such does not provide a method of controlling the component for a general use case. This poses the question;

Could these specific implementations be generalized into methods of controlling a legacy component, without any specific use case in mind? If so, how well do they perform as a means of modernization?

1.3 Purpose

The purpose of this report is to investigate how the NESizer2 method performs as a means of modernization, by repurposing the method to handle the entire instruction set of the RP2A03/6502 microprocessor, and measuring its performance in speed of execution aswell as investigating the complexity of implementing the method.

The purpose of the work is to provide some insight to how well a relatively simple method of wrapping an outdated microprocessor can be expanded upon to function as method of modernizing a legacy embedded system. Although our work is very basic and does not cover the entirety of how to wrap a whole legacy system, we hope our findings can be used as a future reference for others interested in modernizing embedded hardware.

1.4 Goal

The goal with the work is to provide insight into how an existing method of controlling legacy hardware can be extended to allow for general usage, and to give a performance evaluation of the method. This can hopefully give an indication of their usefulness as a method of modernizing an information system that uses legacy components.

1.4.1 Social benefits, ethics and sustainability

If it is possible to wrap entire embedded legacy systems with relatively easy means, it could provide an alternative for businesses that are dependent on legacy embedded components to upgrade their systems without having to invest in, what most likely would be, expensive migrations, and with minimized risk - as wrapping would keep the legacy components intact. We also hope that our work can contribute to other research that aims towards a more sustainable solution than discarding still functioning computer systems, which is becoming an increasing threat to our environment.[7]

We acknowledge that our research could contribute to the continued use of legacy hardware. While the process of discarding obsolete hardware etc. for new parts can have a negative impact on the environment, it is also important to note that upgrading hardware could prove to be a better solution, as much research and development is aimed towards lower power consumption and with a more up-to-date view on sustainable engineering.

1.5 Methodology

In order to expand our knowledge and theoretical background in the field of research and define the research objectives, literature studies were conducted on several occasions during the research. A literature study is the systematic and methodical analysis of literature for a scientific purpose [8].

A literature study was also conducted to decide on a suitable scientific method under which to conduct the research. The scientific method acts as a framework or guidance for the researcher to conduct their research in a well defined and systematic way, based on the works and experiences of researchers before them, and it is crucial to a research in order to ensure quality and correctness of gathered results and analysis. We found that experimental research was most suited to the nature of our research. The experimental approach allows for observing how a system's behavior changes as one variable is manipulated while other variables are kept stable [9], and as such it is suitable for analyzing performance of a system [10].

1.6 Stakeholder

No stakeholder.

1.7 Delimitations

The scope of this report is limited to the design and performance analysis of the NESizer2 method when it has been expanded to handle the entire 6502 instruction set, on a RP2A03 microprocessor. Performance evaluation has been limited to speed of execution per cycle and the complexity of implementation, measured in components required and lines of source code required to write firmware. The details of the implementation can be found in subsequent chapters.

For a better indication of how well the communication method studied in our research can be adopted for other microchips/hardware and for a better picture of the behaviour of these communication methods on other systems, it would have been beneficial to implement them for two or more devices with different architectures. We have compared the performance in execution time against a theoretical execution time of one instruction per second, however measuring the execution speed of the RP2A03 used “as intended” with a ROM could possibly have been more insightful.

1.8 Disposition

2 Background theory/Technical background

This chapter provides an introduction of what legacy hardware means, and also discusses how legacy hardware is used in modern systems and particularly how old hardware is still used to create and produce retro-sounding music. It also introduces a hobby project that sparked the idea for this research. The second part of this chapter discusses some earlier work related to the research problem, and work that was used as a basis for the communication methods designed for this research.

2.1 Legacy

Legacy is a term used in computing to mean “of, relating to, or being a previous or outdated computer system” [11]. This could, for example, be computer systems or components that might have had a widespread usage or been considered a standard in the past, but are no longer manufactured or maintained by the producer.

2.2 Trackers and the Chiptunes scene

In the mid 1980’s, a type of music sequencer commonly referred to as a “tracker” appeared on the personal computing markets. Today, a community colloquially named the “chiptune scene” consisting of musicians and retro enthusiasts fascinated with the characteristic sounds of the old video game consoles, create and perform their own music with these trackers. While many artists use software that can emulate the sounds of these machines on modern systems, it is often considered high status to create the music directly on the old hardware. An often recurring example representative of the scene is the tracker software LSDj [ref], written for the Nintendo Game Boy. Its portable nature makes it an ideal option for artists, being able to carry their “instrument” anywhere with ease.

We wanted to, as a hobby project, develop a prototype for a portable music tracker, similar to the Game Boy and LSDj, using the characteristic sound from the popular Nintendo Entertainment Systems (NES) processor Ricoh RP2A03. In our research, we realized that it would be beneficial if we could write the tracker software for a modern microcontroller that would in turn control the Ricoh chip as a slave unit. This would give us all the expansive capabilities of a modern microcontroller, while also providing us with the actual audio output of the NES.

We realized that our need to control the Ricoh chip in this fashion could also be applicable to other legacy systems that are in need of upgrades, and where emulation is not a viable option.

2.3 MOS Technology 6502 architecture and the Ricoh RP2A03

The MOS Technology 6502 microprocessor and architecture was introduced on the market in 1975. It gained almost instant popularity due to its competative performance for a cheaper price than its competitors.[12]

The microprocessor that was used in the Nintendo Entertainment System was a Ricoh RP2A03 chip. [får man reffa till Nintendos patent? isf reffa det här] The RP2A03 is a proprietary chip based on the MOS Technology 6502 microprocessor architecture, with the difference that it has an added Audio Processing Unit (APU), and it does not support *decimal mode*[1] that would normally be available on a 6502 architecture.[12]

2.4 Atmega328

2.5 Related work

- NESizer2
- Shared memory, injection-grejen
- Andra artiklar vi hittade tidigare

3 Development and methodology

This chapter gives an introduction to experimental research and how it can be used in system performance comparison and analysis, followed by a theoretical background to agile development.

3.1 Experimental research

Introducera experimentell research lite generellt här

[1] Decimal mode allows the processor to compute memory addresses written in decimal values, whereas it otherwise would use hexadecimal. REVIEW THIS FOOTNOTE AND ADD REFERENCES.

3.1.1 Experimental research in performance comparison

Utgå från ref och skriv om de metoderna vi använt oss av.

3.2 Design and software development

4 Project work/Development

References

[1] J. Bisbal, D. Lawless, B. Wu, and J. Grimson, “Legacy information systems: Issues and directions,” *IEEE Software*, vol. 16, no. 5, pp. 103–111, 1999.

[2] H. M. Sneed, *Annals of Software Engineering*, vol. 9, no. 1/4, pp. 293–313, 2000.

[3] Comella-Dorda, Wallnau, Seacord, and Robert, “A survey of black-box modernization approaches for information systems,” in *Proceedings international conference on software maintenance ICSM-94*, 2000.

[4] R. C. seacord, D. Plakosh, and G. A. Lewis, *Modernizing legacy systems: Software technologies, engineering process and business practices.* Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2003.

[5] M. Dowson, “The ariane 5 software failure,” *SIGSOFT Softw. Eng. Notes*, vol. 22, no. 2, pp. 84, Mar. 1997.

[6] “N° 33–1996: Ariane 501 - presentation of inquiry board report.” https://www.esa.int/For_Media/Press_Releases/Ariane_501_-_Presentation_of_Inquiry_Board_report.

[7] “Poison pcs and toxic tvs, californias biggest environmental crisis that you’ve never heard of.” http://svtc.org/wp-content/uploads/ppc-ttv1.pdf.

[8] “Examensarbete t5 läkarprogrammet: Göra litteraturstudie.” http://libguides.lub.lu.se/c.php?g=297461&p=4112444.

[9] D. Ary, L. C. Jacobs, A. Razavieh, and C. K. Sorensen, *Introduction to research in education.* Wadsworth Publishing, 2009.

[10] A. Håkansson, “Portal of research methods and methodologies for research projects and degree projects,” in *Proceedings of the international conference on frontiers in education : Computer science and computer engineering fecs’13*, 2013, pp. 67–73.

[11] Merriam-Webster.com, “Legacy.” https://www.merriam-webster.com/dictionary/legacy; Merriam Webster.

[12] "IEEE chip hall of fame: MOS technology 6502 microprocessor." https://spectrum.ieee.org/tech-history/silicon-revolution/chip-hall-of-fame-mos-technology-6502-microprocessor.