The PHINIX+ System Architecture Documentation
Part 1: The CPU

Come discuss with us at the official PHINIX+ Discord server:
https://discord.gg/EFKDF3VE9C

Version: 0.2.0
Date: 1st of August 2024
by Martin Andronikos

Contents

1 Preface

This section discusses the nature of the documentation itself, the scope and aim of the PHINIX+ project, and about the author as an individual and their motives. As a result, the use of the first person in the following section is unavoidable. The formal specification begins at Section 2 if such details are irrelevant for the reader.

1.1 About Styling

This document was written using the "Typst" typesetting program. If the source code of the used template is not available or the reader is not aware of Typst's syntax, the decisions made regarding styling are hereby given:

- Pages are A4 sized with 25mm of vertical and 20mm of horizontal margins.
- For the bulk of the text the serif font "IBM Plex Serif" was used.
- For the headings and for the title the sans serif font "IBM Plex Sans" was used.
- For the code blocks the monospace font "Inconsolata" was used.
- Internal links (references) are in blue color with the exception of the contents page and footnotes.
- External links (hyperlinks) are underlined and in blue color (as shown above).

1.2 About the Author

Though I do know my way around the field of processor design and implementation, I have no formal experience with the subject. Everything I know about regarding the topic I have learned by myself and with help from other people online. However I am in the process of attending a computer engineering course at a polytechnic university.

Typesetting is also an activity which I have had to teach myself. My university did provide me a "Technical Document Writing" lesson though it was in reality of little help. As a result, if you would like to suggest something regarding the document don't refrain from reaching out on Discord.

1.3 About the Project

This documentation and the overall design and direction of the PHINIX+ project is my personal project which I have been working on during my free time. I never got to experience early computing or the home computer revolution. As a result, I made it my goal to come up with a completely independent computational system that would mimic the experience of using systems of the late 1980s to early 1990s.

PHINIX+ attempts to be a platform from which many of the concepts common in the modern computing environment could be understood (such as Operating Systems) through re-implementation, as well as a platform on which the retro community could build upon. PHINIX+ thus tries to cater to many use cases and it should be wholly up to the implementer which of those use cases is most important for what they want out of the system.

2 Introduction

This document is the official specification for the PHINIX+ Central Processing Unit. It is intended to explain in detail the capabilities and the layout of the processor in an abstract manner in order to remain agnostic of the possible implementations of it. While this document doesn't try to make any assertions of a "correct" sort of implementation, the architecture was built with the intention to exploit pipelining to gain in performance.

2.1 Ancestral History

PHINIX+ is a "constructed" acronym which stands for *Pipelined High-speed INteger Instruction eXecutor*. The "+" in the name is to signify and advancement from a previously designed processor, PHINIX, from whom most ideas where directly taken and improved upon. PHINIX used 16-bit word-addressing which turned out to be unwieldy and not deliver in terms of memory capacity. PHINIX+ expands to 32 bits while also adding byte-addressing to simplify integration with the existing computing paradigms, all based around 8-bit units.

2.2 Influence Sources

PHINIX+ mainly derives from the *Reduced Instruction Set Computing* (RISC) paradigm. However that does not mean it follows the established norm for a RISC processor, opting instead for a more expansive set of instructions, mainly concerning the improvement of flags management and bit math. The core principles of RISC like the load-store paradigm and the general usage nature of the provided registers do exist in PHINIX+ but not without being improved upon.

One of the most apparent features a programmer wishing to use PHINIX+ encounters is the dual register file. This is a feature influenced directly by the Motorola 68000 series of processors. Though that processor was in no way following RISC, the the adoption of the dual register file was due to similar reasons. As a result PHINIX+ has been lovingly nicknamed the *Actually-RISC™ M68k*[1].

[1]Disclaimer: Not actually a trademark.