FPGA-based RISC Microprocessor and Compiler (Rev. 1.00)

PRCO304 - Final Stage Computing Project

Ben Lancaster 10424877

March 11, 2018

Revision History

Table 1: Document revisions.

Date	Version	Changes
11/03/2018	1	Initial section outline.

Abstract

ben

Table of Contents

1 Introduction

2 Background

2.1 Current Implementations

3 Objectives

4 Legal and Ethical Considerations

The BEN-1816 processor will be able to read and write to all data passing through it and in connecting peripherals.

The processor does not track or store usage behaviour, instructions and their frequency, memory contents, or timing statistics.

The BEN-1816 processor is not designed to run general purpose operating systems, such as Linux or embedded RTOS systems. The processor lacks common components required to run modern operating systems, such as a memory management unit (MMU),

It is not designed to run in high-reliability or safety-critical environments that require established safety standards, such as the UK Defence Standard 00-56 (Bowen and Stavridou, 1993) and IEC 61508 (Bell, 2006).

This project uses only 1 external library for the processor core's universal asynchronous receiver-transmitter (UART) module that does not depend on any other libraries. The UART module does feature a large first-in-first-out (FIFO) buffer for temporary storage of in- and out- going messages. This FIFO is internal to the FPGA design and so is protected from external viewing or modification.

The compiler sub-project does not use any external library dependencies, does not record telemetry or usage statistics, and does not require an internet connection to run.

5 Project Management

6 Requirements

7 High Level Design

8 Testing and Verification

9 Conclusion

10 Appendices

10.1 Appendix A. PRCO Core Reference Guide

10.2 Appendix B. PRCO Compiler Reference Guide

10.3 Appendix C. Project Initiation Document

11 References

Bell, R. (2006). Introduction to iec 61508, pp. 3–12.

Bowen, J. and Stavridou, V. (1993). Safety-critical systems, formal methods and standards, Software Engineering Journal **8**(4): 189–209.