Mcu (Microprocessor) Design

Project RISCII (PR0002)

Description of internal and external design of the RISCII microprocessor”

Rev A
23-JUN-2024

Purpose

This document is meant to describe how the RISCII microprocessor is designed. It primarily focuses on the external interfaces, though also goes into some detail about the internal, detailed design of the device.

The contents of this document should effectively describe the “high-level” interfaces and behavior of the microprocessor. It is expected this document is referenced in developing boards and low-level compiling software systems for the microprocessor.

Abbreviations

Abbreviation	Description

MCU	Microcontroller unit- though for this document, it is equivalent to microprocessor
word	2 byte value, unless context dictates it is referring to a sequence of letters (i.e. natural language)

References

Name	Document Number
Instruction Set Architecture	PR0001

NOTE: For the sake of getting the ball rolling on the assembler/linker, this document is submitted as is despite missing the following sections inferred by the section below:

1) Runtime chip
2) Storage chip
3) Reset logic
4) Boot logic
5) MCU state logic

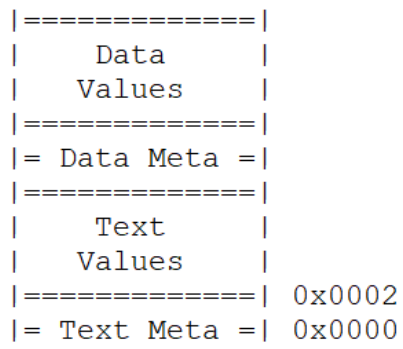
These sections will be added/incorporated as progress on the assembler/linker and microprocessor continue.

Binary Image Usage

The MCU, when properly connected to the storage and runtime memory chips, can be programmed with a binary image. Upon a hardware reset, the MCU reads the binary image and initializes the hardware before running the image (i.e. the read and initializing is done by hardware).

In general, the binary image is made up of two sections, each with their own values, metadata, and destination. The subsection below and its image describe the binary image's partitions in more detail.

Partitions



- Image 1: Binary Image, byte addressed

Text Section Metadata

This section consists of a one word value used to describe the length of the text section stored in the binary image. The value is equivalent to one less than the sum of 0x0000 and the number of words present in the text section.
It is important that this value is accurately computed as the MCU relies on it for extracting the text section. Improper values may lead to unexpected results.

Text Section Values

This section consists of the values that make up the text section of the program (i.e. instructions, as found in document PR0001 Instruction Set Architecture). Upon a hardware reset, these values are copied from the storage chip and initialized in the program address space located in the runtime chip (aligned to lowest address).

Data Section Metadata

This section consists of a one word value used to describe the length of the data section stored in the binary image. The value is equivalent to one less than the sum of 0x8000 and the number of words present in the data section.

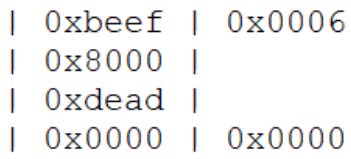
It is important that this value is accurately computed as the MCU relies on it for extracting the data section. Improper values may lead to unexpected results.

Data Section Values

This section consists of the values that make up the (initialized) data section of the program. Upon a hardware reset, these values are copied from the storage chip and initialized in “free memory” half of the data address space located in the runtime chip (aligned to lowest address- see PR0001 Instruction Set Architecture).

Minimum Viable Image

Due to how the MCU interprets the metadata, it is required that every partition consists of at least one word- even if the partition is not effectively being used. This results in a “minimum viable image”, an example of which is shown below.



- Image 2: Minimum Viable Image, 8 bytes total

Revision History

Rev	Date	Description of Changes	Initials
A	23-JUN-2024	Initial draft, focus on binary (for assembler work)	J.E.