Central(Execution Engine)

Public Interface Specification

Table of Contents

Preface

This document presents the public interface of an *Execution Engine* of the **Central(**Architecture**)**. This means that the amount of information pertaining to the internals of an *Execution Engine* hardware implementation are kept at a minimum except when a choice in the public interface is specifically made to simplify said implementation.

Glossary

Central Processing Unit

An hardware unit containing one or more *Execution Engines*, a memory controller, an interrupt controller, an operator facility controller and various other minor components.

Execution Engine

The hardware responsible for the execution of user-written code, be it kernel code or user program code. An *Execution Engine* contains a fetch unit, a decode unit, a micro-instruction sequencer, an Arithmetic & Logic Unit, various registers and subsystems. An *Execution Engine* is analogous to an *hart* in the RISC-V nomenclature or to a *core* in other CPU specifications.

Data Manipulation

Data width

All the registers of the **Central(**Architecture**)** are 32bit wide.

Address width

Memory addresses are 24bit wide thus an *Execution Engine* can address up to 16MB of memory.

Memory alignment

The **Central(**Architecture**)** uses byte-addressable memory. Under the hood, memory accesses are done on a memory-word boundary. A memory-word is 32bit wide. To maximize performance, memory accesses should be done on a 32bit alignment.

The term “memory accesses” encompasses both read and write operations.

Endianness

Data is encoded in memory with the little endian scheme. For a given value, the least significant byte (LSB) is stored in the lowest address and the most significant byte (MSB) in the highest.