Chapter 3

RV32E Base Integer Instruction Set

This chapter describes the RV32E base integer instruction set, which is a reduced version of RV32I designed for embedded systems. The main change is to reduce the number of integer registers to 16, and to remove the counters that are mandatory in RV32I. This chapter only outlines the differences between RV32E and RV32I, and so should be read after Chapter 2.

RV32E was designed to provide an even smaller base core for embedded microcontrollers. Although we had mentioned this possibility in version 2.0 of this document, we initially resisted defining this subset. However, given the demand for the smallest possible 32-bit microcontroller, and in the interests of pre-empting fragmentation in this space, we have now defined RV32E as a fourth standard base ISA in addition to RV32I, RV64I, and RV128I. The E variant is only standardized for the 32-bit address space width.

3.1 RV32E Programmers' Model

RV32E reduces the integer register count to 16 general-purpose registers, (`x0`–`x15`), where `x0` is a dedicated zero register.

We have found that in the smallest RV32I core designs, the upper 16 registers consume around one quarter of the total area of the core excluding memories, thus their removal saves around 25% core area with a corresponding core power reduction.

This change requires a different calling convention and ABI. In particular, RV32E is only used with a soft-float calling convention. Systems with hardware floating-point must use an I base.

3.2 RV32E Instruction Set

RV32E uses the same instruction set encoding as RV32I, except that use of register specifiers `x16`–`x31` in an instruction will result in an illegal instruction exception being raised.

Any future standard extensions will not make use of the instruction bits freed up by the reduced register-specifier fields and so these are available for non-standard extensions.

A further simplification is that the counter instructions (`rdcycle[h]`,`rdtime[h]`, `rdinstret[h]`) are no longer mandatory.

The mandatory counters require additional registers and logic, and can be replaced with more application-specific facilities.

3.3 RV32E Extensions

RV32E can be extended with the M and C user-level standard extensions.

We do not intend to support hardware floating-point with the RV32E subset. The savings from reduced register count become negligible in the context of a hardware floating-point unit, and we wish to reduce ISA fragmentation.

The privileged architecture of an RV32E system can include user mode as well as machine mode, and the Mbare, Mbb, and Mbbid memory management schemes.

We do not intend to support full Unix-style operating systems with the RV32E subset. The savings from reduced register count become negligible in the context of an OS-capable core, and we wish to avoid OS fragmentation.