Alt_ISA Preface

Alt_ISA is an OpenCores processor project collecting a variety of “alternative” Instruction Set Architectures. These are variations on existing ISAs. The variations are intended to improve some aspects thought to be limiting or incomplete. This project is large: eight distinct ISAs each with a full complement of op-codes. Current status is somewhere between planning and fully defined or specified. In each case there is a justification document which may or may not contain a full description. There are spreadsheets showing the instruction formats and the associated op-codes.

Commonalities exist in that each ISA has a small number of instruction formats and that immediate fields are variable length with length encoded as part of the field.

The ISAs are designed to be compatible with FPGA implementation: Asynchronous multi-port LUT RAM, synchronous block RAM, D flip-flops, LUT based logic and a single clock. Initial implementations will be minimal and use “single cycle” design. Multiplication is supported using associated FPGA primitives. Code space is reserved for floating-point. Usually there is ~12% unoccupied op-code space for future expansion. There is an escape mechanism defined for augmenting the number of op-codes and the register field sizes using prefix instructions.

The designs support three or four data sizes, typically byte addressable, and register files whose word size matches the largest data size. Simple block RAM memory interfaces are provided. Interrupts, virtual memory, caches and pipelining are future topics.

At places there are implementation choices. The set of data sizes built-in is the most basic. Another is the number of registers in the register file, which affects the instruction formats. Code generation is either manual or via a list of subroutine calls in a Python or Julia application program. The author is not handy with compiler development.