Design of the *Nebula* Instruction Decoder

Elijah Creed Fedele

May 28, 2023

Contents

1 Introduction

In the traditional five-stage RISC pipeline model, the purpose of the instruction decode (ID) stage is to translate the 32-bit instruction into smaller chunks of signals which can be passed to the control and execution units to direct and sequence the instruction's operations. Because the instruction decoder is directly responsible for translating the instruction encoding into "actionable" commands, it is here where the complexity of an instruction set is most felt. As such, it can range from a few basic constructions of look-up tables (LUTs) and multiplexers in RISC architectures to large, state machine-driven, microcoded instruction translators in CISC processors.

The RISC-V architecture is designed with the intent to be relatively simple to decode, and this feature is primarily expressed in the modularity of the instruction opcode layout. While the instruction set is specifically designed for ease of decoding, much of the instruction set is unnecessary for base RV32G implementations, and can be cut down much further to reduce resource utilization on FPGAs.

1.1 Two approaches to instruction decoding

There are two distinct approaches to instruction decoding which immediately suggest themselves. The first approach decodes instructions nearly completely into a bundle of control signals which are passed either to the control & sequencing unit or to the execution units directly. This "bundle-of-signals" method is ideal for small, tightly-encoded instruction sets whose opcodes and functional descriptors can be very cheaply converted to discrete command signals. However, as the available instruction space and number of encoding formats increases, the gate overhead required to fully decompose opcodes greatly increases and another method must be sought.

The second method consists of "subcoding", or partial reduction of the opcode/function descriptors into a much smaller "subcode“ which is passed in parallel with the register data buses to the functional units. Each independent functional unit then completes the final stage of reducing the subcode into required control signals on-the-fly. This localized secondary decoding step presents net benefits for pipelining, where an overly-complex ID stage would otherwise bog the maximum frequency unnecessarily. It also provides an aspect of modularity, as the smaller internal opcode can be made to accommodate instruction additions more easily than the "bundle-of-signals" decoder.