EE480 Assignment 2: IoQ Don Encoding And Assembler

Implementor's Notes

Matt Ruffner
matthew.ruffner@uky.edu

ABSTRACT

This assignment focuses on designing and laying out the encoding of the instruction set who's decoding will be implemented in future projects. An instruction set with 20 operations was given to be encoded, with 16 bit instruction words, and a maximum of three parameters per operation. This leaves, in a worst case scenario, 4 bits for the opcode field in the instruction. Extra opcode identification bits were employed in order to identify instructions with fewer than three parameters.

1. GENERAL APPROACH

When laying out bitfields for registers and immediate values, one must keep consistency in mind. Rearranging fields for similar instruction parameters would cause a hassle in having to treat each instruction as a special case.

By grouping operation parameters as much as possible, factoring fields to create minimal vertical overlap, decoding logic necessary in a hardware implementation will be less than that of an encoding which has arbitrary argument-field arrangements.

The high 4 bits of the 16 bit instruction word were used as the opcode field. For the four instructions that were not able to have a unique value relative to this field, the lower 4 bits were used for secondary identification.

The value of `0` in this field is reserved for all two-parameter instructions that do not require this secondary decoding process. The remaining instructions, excluding `nop`, were given secondary identification starting at `1` in this field. `nop` is encoded as all ones, for simplicity's sake.

2. ISSUES

No issues were encountered when progressing through this project. The AIK assembler definition was input into the class website interface, along with the provided test case. The results of the simulation were identical to the non-hidden parts of the solution on the assignment website.

ACM ISBN N.A..
DOI: N.A.