Apex Instruction Set Architecture Simulator (`apex-sim`) Phase 1 Documentation

Matthew Cole
mcole8@binghamton.edu

Brian Gracin
bgracin1@binghamton.edu

19 November 2016

Contents

List of Figures

1 Design

`apex-sim` is a simulator for the *Architecture Pipeline EXample* (APEX) Instruction Set Architecture (ISA). `apex-sim` consists of the following components:

- `main.cpp` contains the driver program. The driver program provides file input for instructions, user interface operations, and maintaining persistent simulator state. This component is discussed in section 1.1.
- `code.cpp`, `data.cpp`, `registers.cpp`, `cpu.cpp`, and `stage.cpp` provide the objects modeling components of the pipeline. These components are discussed in section 1.2.
- `simulate.cpp` provides the functions that allow the CPU to simulate working on each of its stages, inter-stage communication through advancement, stalls for basic inter-stage interlocks, and forwarding. These implementation details are described in section 2.

Figure 1 shows class interactions and data flow between each of the stages and support classes. Finally, we opensource our work under the MIT license through a GitHub repository located at `https://github.com/colematt/apex-sim`. We discuss our team's work log in section 3.

1.1 Driver Program

The `apex-sim` entry point file is `main.cpp`. This program shepherds execution through the lifecycle of the program and provides a user interface for interacting with the simulator. The functionality of the driver program is as follows:

1. Verify sanity of command line inputs (lines 107-116).
2. Instantiate class instances for the simulator (lines 119-122).
3. Perform the initialization of each pipeline stage (line 125).
4. Prepare and begin the simulator user interface's operations (lines 128-134).
5. Parse user interface inputs and delegate actions to interface helper functions (lines 134-161).

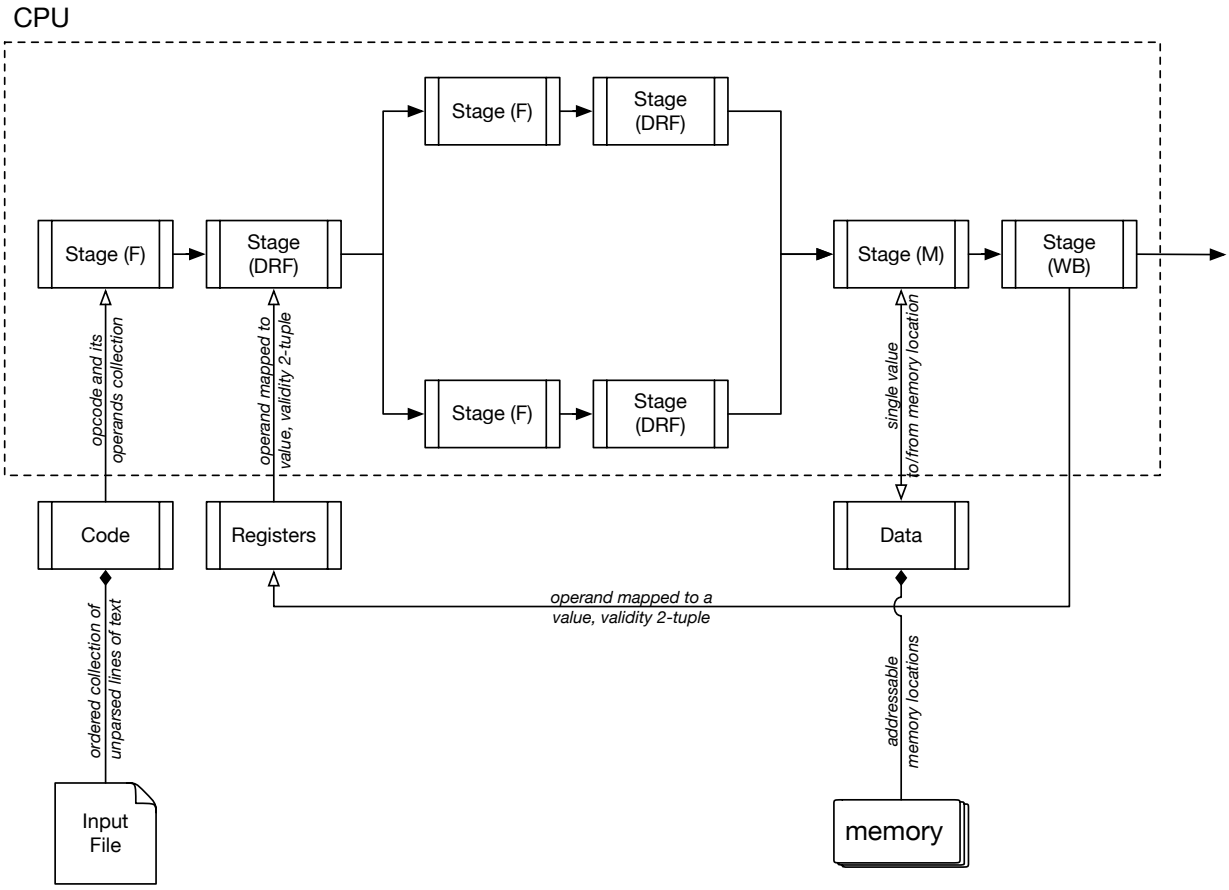

Figure 1: The APEX pipeline and class interactions.

1.2 Classes

1.2.1 Code

1.2.2 Data

1.2.3 Registers

1.2.4 CPU

1.2.5 Stages

2 Implementation

2.1 Work Phase

2.2 Advance Phase

2.3 Stalls

2.4 Forwarding

3 Production