Spike RISC-V ISA Simulator Usage Guide

CS340400 Compiler Design

NTHU PL LAB

Outline

- Generate Executable
  - codegen.S
  - Compile Executable
- Spike RISC-V ISA Simulator
  - Spike Introduction
  - Spike Usage

PL NTHU LAB

2. Generate Executable

PL NTHU LAB

codegen.S

- The rules are the same as those for Andes Corvette-F1/T1, including but not limited to the following items:
  - Same set of Testcases
  - Implement `delay , digitalWrite`
  - `.global codegen`
  - …

NTHU PL LAB

Compile Executable

- TAs provide a tweaked version of the assembly sample project, which includes:
  - `main.c` : The main program
  - `codegen.S` : The same one as in the assembly project

PL NTHU LAB

Compile Executable (cont.)

- To compile your codegen.S into an executable, use `riscv32-unknown-elf-gcc`
  - e.g. `riscv32-unknown-elf-gcc -o sample_prog main.c codegen.S`
  - The above command does the following:
    - Compile `main.c`
    - Assemble `codegen.S`
    - Link them together to produce `sample_prog`
  - `sample_prog` is the executable we want

PL NTHU LAB

2. Spike RISC-V ISA Simulator

https://github.com/riscv-software-src/riscv-isa-sim

PL NTHU LAB

Spike Introduction

- Spike is a functional-level simulator for the RISC-V ISA
- It operates in a bare-metal manner, i.e. it behaves like a hardware without OS
  - In HW 3 Spike, we need OS support for `printf` in `assembly/main.c` , so we make use of the `pk` utility provided by the RISC-V community (`pk` stands for "proxy kernel").

PL NTHU LAB

Spike Usage

- Suppose we have our compiled `sample_prog` , to execute it, run:
  - `spike pk sample_prog` in the assembly folder
  - You should have a correct invocation log of delay and `digitalWrite` as output
    - This is the correct output for the `assembly` sample project:

```
bbl loader
Arduino digitalWrite(27, 1);
Arduino delay(1000);
Arduino digitalWrite(27, 0);
Arduino delay(1000);
```

```
ta@2023spring:~/hmlai/hw3/spike_demo$ spike pk sample_prog
bbl loader
Arduino digitalWrite(27, 1);
Arduino delay(1000);
Arduino digitalWrite(27, 0);
Arduino delay(1000);
```

PL NTHU LAB

Thanks

PL NTHU LAB