Imperial College London

ELEC50010 - Instruction Architectures and Compilers

CPU Coursework by Team 12

Instructors:

Dr. David Thomas

Submission date: December 20, 2020

Contents

1 CPU **1**

2 Test bench **1**

2.1 Test bench Overview 1

2.2 Parser 2

1 CPU

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Quisque nisl eros, pulvinar facilisis justo mollis, auctor consequat urna. Morbi a bibendum metus. Donec scelerisque sollicitudin enim eu venenatis. Duis tincidunt laoreet ex, in pretium orci vestibulum eget. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Duis pharetra luctus lacus ut vestibulum. Maecenas ipsum lacus, lacinia quis posuere ut, pulvinar vitae dolor. Integer eu nibh at nisi ullamcorper sagittis id vel leo. Integer feugiat faucibus libero, at maximus nisl suscipit posuere. Morbi nec enim nunc. Phasellus bibendum turpis ut ipsum egestas, sed sollicitudin elit convallis. Cras pharetra mi tristique sapien vestibulum lobortis. Nam eget bibendum metus, non dictum mauris. Nulla at tellus sagittis, viverra est a, bibendum metus.

2 Test bench

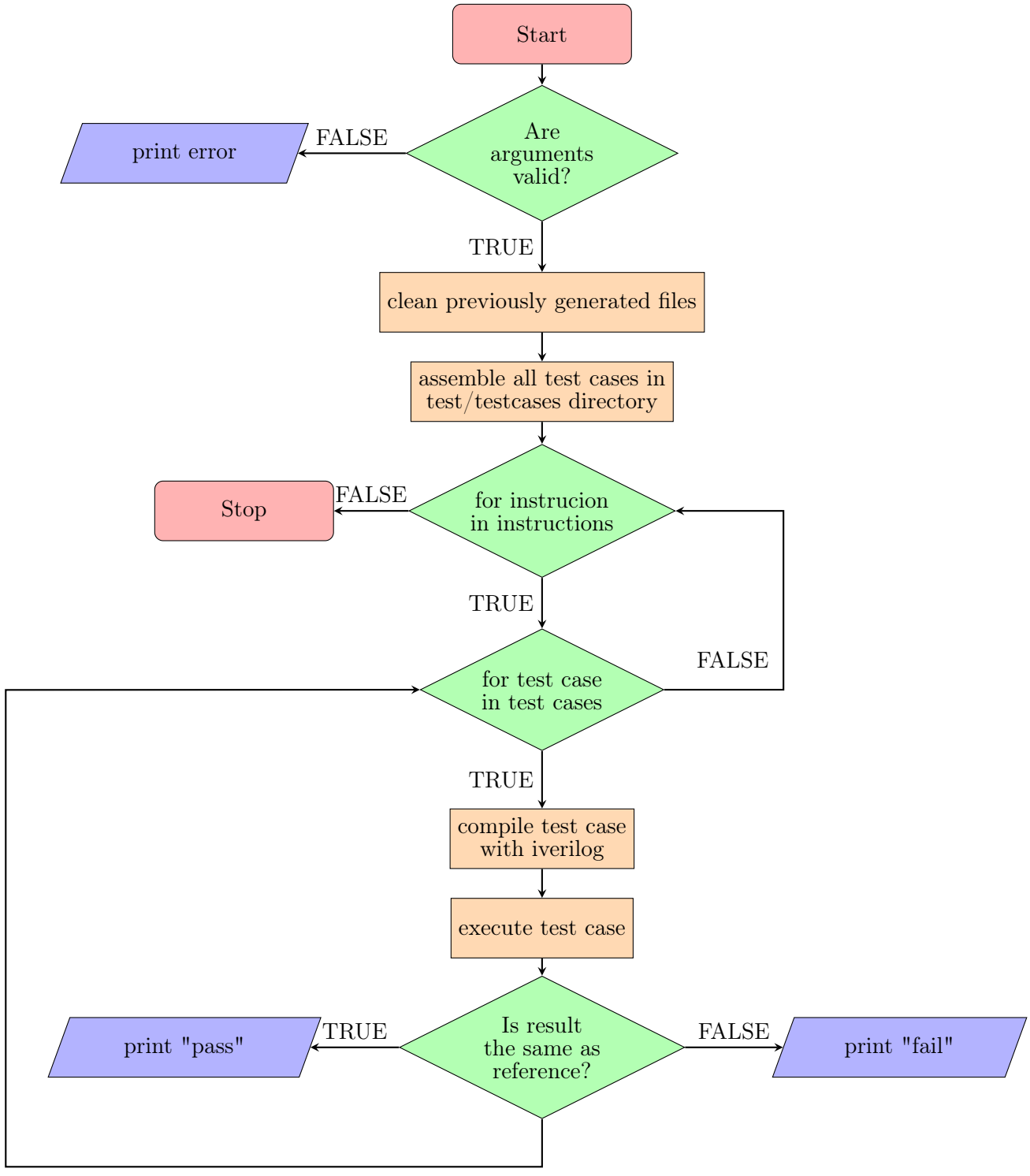
2.1 Test bench Overview

This test bench is designed to be able to carry out test on any cpu that satisfies MIPS1 specification and avalon memory-mapped interface.

The main shell script contains three parts: argument validation, preprocessing on test cases, compilation/execution/comparation of test case on the target cpu.

During argument validation, the main shell script assumes one compulsory argument for source directory of cpu and one optional argument for a specific instruction to test. The shell script would abort either when an invalid source directory or an unsupported instruction is supplied.

Preprocessing of the test cases removes previously generated files if they exist, and generates test cases from ‘readme.md‘ file in each instrucion folder. Each test case contains four files: assembly code, v0 reference, initial content for data section, reference for data section.



2.2 Parser

```
test/
├── RAM_32x64k_avalon.v
├── assembler.sh
├── clear_all.sh
├── compare_result.sh
├── generate_testcases.py
└── generate_testcases.sh
```

```
├── mips_cpu_bus_tb.v
├── test_mips_cpu_bus.sh
├── test_mips_cpu_bus_one_instruction.sh
├── test_mips_cpu_bus_one_testcase.sh
├── test_mips_cpu_bus_real.sh
└── testcases/
    ├── instruction_1
    │   ├── testcase_1
    │   │   ├── testcase.S
    │   │   ├── testcase_data.init
    │   │   ├── testcase_data.ref
    │   │   └── testcase_v0.ref
    │   ├── testcase_2
    │   └── ...
    ├── instruction_2
    └── ...
```

Listing 1: Parser part 1