

CS 2410 Graduate Computer Architecture

Fall 2017

Project 3: Transparent, Reproducible and Reusable Simulation

Assigned: December 1, 2017.

Due December 15, 2017 by 1:00 PM.

1 Introduction

In this project, you will learn about best practices for modeling and simulation in computer architecture. Using OCCAM, you will incorporate your project 1 and project 2 simulators into an ecosystem of modeling and simulation content that follows the principles of FAIR (findable, accessible, interoperable and reusable) to achieve transparent, reproducible and reusable experimentation.

2 Overview

Recall that in project 2, you developed a simulation of Tomasulo's algorithm. Your simulator took as an input an instruction trace. This instruction trace was generated by a modified version of project 1. In project 3, you will add both project 2 and the modified project 1 to an instance of OCCAM. Using the two projects, you'll create a *workflow* that describes the steps of an experiment and the initial conditions of the experiment. The steps in this case will be relatively simple: an input program will be connected to the modified project 1. Project 1 will be connected to project 2, which will generate output. In essence, this is an experimental workflow! Project 2 will be configurable to set initial experimental conditions.

Below is an example of the workflow. You will demonstrate project 3 by creating this workflow and running a sample program. Your completed project should appear similar to the example, be modifiable and runnable from OCCAM.

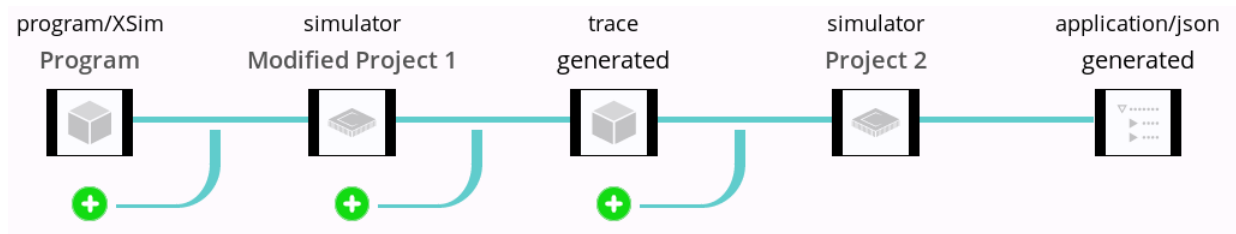


Figure 1: Example workflow – your project should be similar

3 Using OCCAM

To integrate projects 1 and 2 into OCCAM will involve writing “wrappers” for configuration input, building the simulator, running the simulator, and parsing the statistics output. A separate example, with the necessary wrappers, will be provided to help you get started. You can change these wrappers to add your simulators. You will need to do this for both simulators!

The OCCAM integration should allow configuring project 2, selecting an input program, running the program, and viewing the output statistics.

4 Project Requirements

Both project 1 and project 2 should be added to your OCCAM instance. You can use the virtual machine (VM) given to you at the start of the semester. This VM has a runnable, ready to use instance of OCCAM.

Project 1 needs to be modified to output an instruction trace, as described in the project 2 handout. Project 1 will output a file type that can be read as an input to project 2. You will need to specify this “type” information in the OCCAM wrappers.

For project 1, you do not need to specify any configuration inputs. You may assume unit latency (really, make any latency assumptions you want – it doesn’t matter since project 1 is used only to generate a trace).

For project 2, it should be possible to configure all the parameters within OCCAM. Using the wrappers for the configuration inputs, OCCAM will generate a web page that will let the user configure the simulator. OCCAM generates a JSON file from these web pages, which is the input configuration to the Tomasulo simulator.

The output from project 2 should be viewable as JSON from within OCCAM. You do not have to plot or otherwise manipulate the data output.

5 Turning in the Project

For grading, you will turn in the project by email (as explained below) and give a demo. Your project will be graded primarily during the demo.

5.1 What to Submit

Prior to scheduling the demo, you must submit all files for the project in an archive. Please submit only the source files—don’t submit SST or the OCCAM source.

Be sure to include a README with any problems that you might have encountered, i.e., bugs in your implementation. In the README, put your name and e-mail address at the top of the file. The README file should list what works and any known problems. If you have known problems/issues (bugs!), then you should clearly specify the problems in this file. The explanation and bug list are critical to grading. So, if you’re uncertain whether to include something, then err on the side of writing too much and just include it.

The project is due: December 15, 2017 by 1:00 PM. Submit your files by email.

5.2 Where to Submit

Submit your files to Wenchen Wang, email wew50@pitt.edu.

5.3 Demonstration

You will demo your simulator for grading. We will give you a test program and ask you to configure specific latencies for your project 2 simulator. The test programs will be available from Bitbucket, and they will already be “wrapped” for use in OCCAM. You should be able to load the test files into your OCCAM instance (running your simulator), configure the simulator from the OCCAM interface, and run the simulation. You should be comfortable with demonstrating the project in OCCAM, such as configuring the simulator and viewing the JSON results.

For the demo, you will configure a “workflow” of project 1 that executes a X program and outputs a trace, which is input to project 2.

All projects should be demonstrated by December 15, 2017. A sign-up sheet with appointment times will be made available prior to the demo date. If you have a class conflict with the demo day/times, we can make accommodations. If we have not received your file submission by the due day/time, we may not grade the project and you may receive a 0.

Because the demos are intended to be short, please be prepared to quickly and smoothly show what you have done! The time allotted for each demo is strict.

6 Collaboration

In accordance with the policy on individual work, this project is strictly an individual effort. You must not collaborate with a partner. It is your responsibility to secure and back up your files.

7 Hints

1. The output of your modified Project 1 is not a structured JSON file. Therefore, unlike the example created as part of the OCCAM tutorial presentations, it will have no schema.
2. Don't forget to add the intermediate "generated" block! As explained in the OCCAM tutorial this block will hold the intermediate results of the modified Project 1 simulator (the trace) before being used by the Project 2 simulator.
3. Make use of the auxiliary materials provided for the OCCAM tutorial, and modify them according to your needs.