1 Description

The behavioral synthesizer that you will build in this project converts a C program file into a hardware module. For the sake of simplicity, we make the following assumptions for the input C program:

1. There is a behMain function, which servers as the starting point of hardware behavior.
2. The global variables are assumed to be the ports of the hardware module.
3. There are no functional calls, array usage, struct/union usage, and pointer usage.

Apparently, A C frontend is needed to convert the C program into an appropriate intermediate form for further processing. Both the frontend and a library for manipulating the intermediate form is provided in a simple compiler infrastructure called SimpMeta. The goal of this lab is to get familiar with it and hence get prepared for the subsequent labs.

The concept involved in SimpMeta is covered in the class, but you are required to study the detail by reading its documentation, installed on EECG computer systems at /jzhu/b/b0/jzhu/opt/simpmeta. You should use the tool mcc to compile the benchmarks. You should use the tool mdump to dump a compiled C program into human readable form. In the end, you should write your own version of mdump that uses the SimpIR API to traverse the intermediate form and print useful information on the screen.

2 Deliverables

- The source code of your behdump program;
- A makefile that includes
  - a build target that builds an executable from your source programs;
  - a test target that run your program over all benchmarks.
3 Due Date

8am, February 5, Wednesday, 2003.