Assignment 3
Due: 11:59 pm, October 5th, 2015

Purpose: Understand the HARP ISA and SIMT execution model

Target Machines: Your program must run on linux system using gcc 4.8 and above or clang 3.0 and above.

Assignment: Write an emulator for a subset of the HARP ISA according to the provided HARP ISA documentation. Example program binaries and golden outputs are provided to verify correctness. Assembly versions of example programs are also provided for reference.

1. Write the emulator to conform to the HARP ISA specification and support the architecture ID 4w8/8/8/8, consult section 1 of the ISA documentation for details. The emulator should be parameterizable, from 4w8/8/1/1 to 8w/64/64/64/64
2. Implement all instructions ***except*** for privilege instructions, SIMD control, warp control, and trap for this assignment (skip 5.2, 5.10, 5.11, 5.12 in the HARP ISA manual).
3. The launch arguments of your emulator should be in the form of:
   `./harp_emulator program output_generated`
   - `program` - name of the program to be executed
   - `output_generated` - dump the program output to this file
   - **DO NOT** change the name of the emulator from `harp_emulator` - this simplifies grading greatly.
4. The example programs provided are: bubble sort, quick sort, psum, and shuffle. Inputs to example programs are included in the program binary. Be sure to unzip them before using (~130MB after unzip).
5. The example programs contain the necessary bootstrap instructions to setup memory map and stack registers (not exposed in assembly versions). All your emulator has to do is load the program as a series of bytes and start executing the first instruction at PC=0.
6. Program outputs are written out to the IO device one character per instruction execution. Take the outputs from thread 0 and print them to a file whose name is defined by the `output_generated` argument.
7. Grading will use a different binary (one of the example programs with different inputs) to verify correctness.

Grading Guidelines

For your information here are the grading guidelines

- Program compiles without errors (and appears to be correct): 25 points
- Program executes correctly: (additional) 50 points
- Documentation (comments): 25 points
  - Thoroughly **comment** your functions and variables

- Have a commented program header with your name, class, and assignment

Submission Guidelines:

All submissions should be electronic (both program and problems). Submissions must time stamped by midnight on the due date. Submissions will be via Tsquare.

Note: No late assignments will be graded. Remember, you are expected to make a passing grade on the assignments to pass the course!