pr5 Simulator - Part 2

Sandeep Chandran

6-Oct 2024

Lab Assignment 4 (Graded)

In this lab, we will simulate a single cycle RISC-V processor. Given below are a few mandatory points:

- Implement a single cycle processor (as discussed earlier).
- The address of the first instruction to execute (or simulate) should be taken as a command-line argument with the option identifier `--start`. The default start address (if no `--start` argument was passed) should be `0x80000000`. Example commands to simulate the execution of the binary `1-even.r5o` are given below. The first instruction executed in the case of the first command is at `0x80000020`. In the case of the second command, the first instruction executed is at `0x80000000`.

  ```
  $ cd pr5
  $ python3 src/main.py --start 0x80000020 programs/bins/1-even.r5o
  $ python3 src/main.py programs/bins/1-even.r5o
  ```

  (*Hint: Use the package argparse to handle command line options*)

- The number of the instructions to simulate should be taken as a command-line argument with the option identifier `--num_insts`. The default value (if no `--num_insts` argument is passed) should be `1000000`.
- After each instruction completes its execution, you should print a message in the following format.

  ```
  <cycle>: <PC>: <Disassembly> : <reg/mem>-<value>[, <reg/mem>-<value>]
  ```

  Here,

  - < *cycle* > - Cycle number when the instruction completed its execution.
  - < *PC* > - The PC of the instruction that completed its execution in this cycle.
  - < *Disassembly* > - The disassembly of the instruction that completed its execution in this cycle.
  - < *reg*/*mem* > - The register id which is updated and the address of the memory location accessed by the instruction.
  - < *value* > - The new value which is written to the register or memory location.

  An example is shown below:

  ```
  1: 0x80000000: li x1, 0: x1-0x00000000
  2: 0x80000004: lui x5, 0x1e: x5-0x0001e000
  3: 0x80000008: blt x5, x0, 10:
  4: 0x8000000c: lw x14, 0(x11): x14-0x00000002, mem-0x80002048
  ```