Constructive Computer Architecture

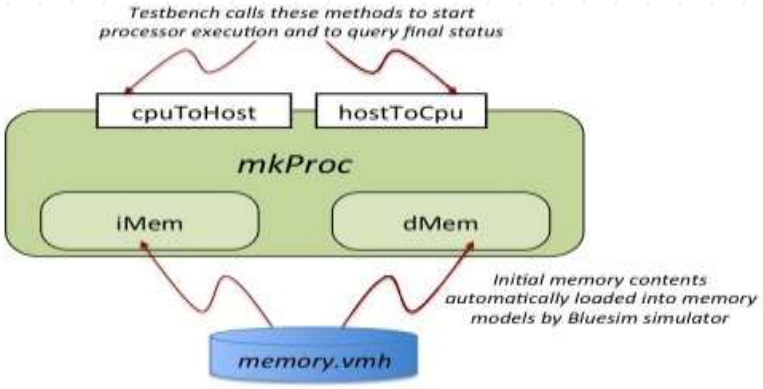
Tutorial 3: Debugging SMIPS

Andy Wright
6.S195 TA

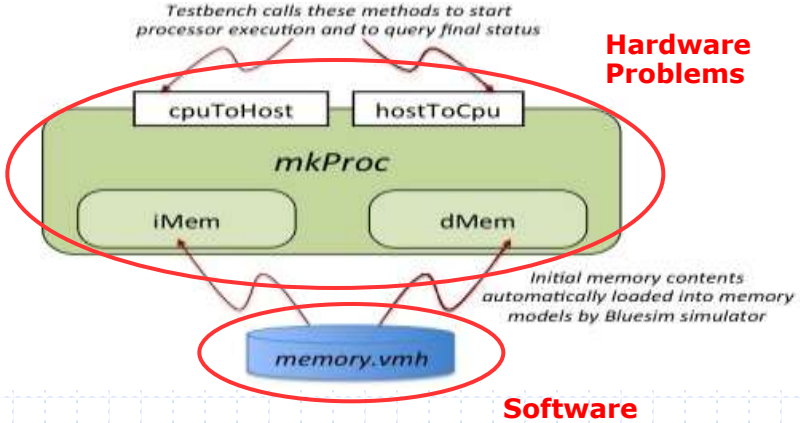
Introduction

- In lab 4, you will be making modifications to an existing, functional, SMIPS processor
- How do you know if your processor is working?
  - You will run an existing suite of C and assembly software test benches on your processor
- What could go wrong?
  - Software and Hardware
- How will you debug this?
  - Print statements!

SMIPS Cpu Interface



SMIPS Cpu Interface



Running SMIPS Test Benches - Success

```
$ ./smips -x -b 1cyc.bsv -c all -r
…
Compiling programs/src/assembly/smipsv2_addu.S
Running smipsv2_addu.S on 1cyc.bsv
PASSED
…
Compiling programs/src/vvadd
Running vvadd on 1cyc.bsv
Cycles =     3018
Insts  =     3018
PASSED
…
```


SMIPS's Coprocessor

- The coprocessor is what enables these software tests
  - Reg 10 (read): Number of clock cycles passed
  - Reg 11 (read): Number of instructions executed
  - Reg 18 (write): Printing an integer
  - Reg 19 (write): Printing a character
  - Reg 21 (write): Finished executing (data is return code)

Test Benches

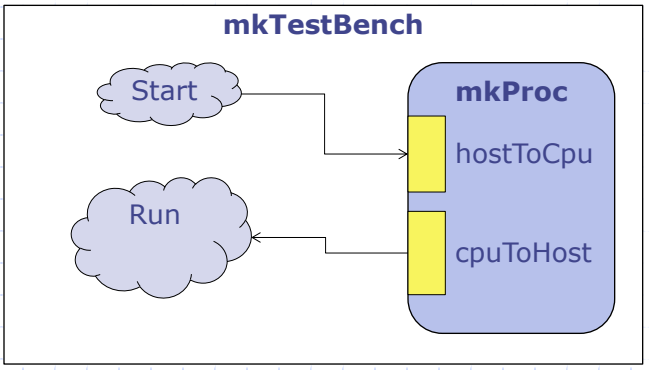
- Test Benches can be written in assembly or in C
  - Assembly examples:
    - smipsv1_simple.s
    - smipsv2_addu.s
  - C example:
    - vvadd

Running SMIPS Test Benches - Failure

```
$ ./smips -x -b 1cyc.bsv -c all -r
…
Compiling programs/src/assembly/smipsv1_lw.S
Running smipsv2_addu.S on 1cyc.bsv
FAILED       1
…
Compiling programs/src/vvadd
Running vvadd on 1cyc.bsv
Executing unsupported instruction at pc: 00000004.
Exiting
…
```

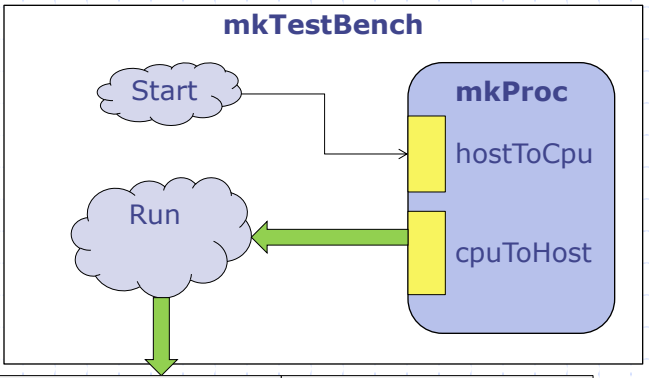
We need more feedback from the processor!

Test Bench Setup



Where does feedback from the processor come from in this system?

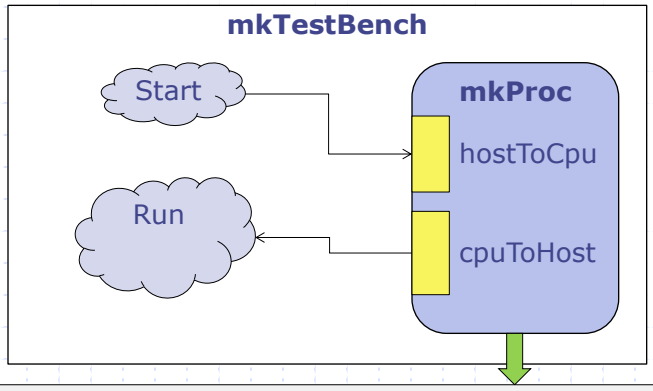
Printing from the COP




```
Cycles =   2052    PASSED
Insts  =   1026    FAILED    5
```

These messages are written to stderr.

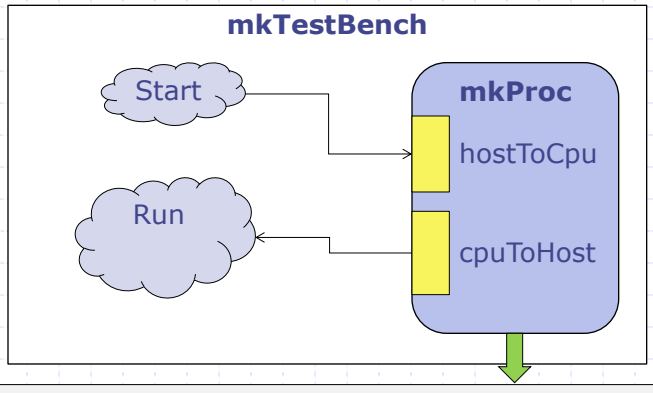
Printing from Bluespec: Error




```
$fwrite(stderr, "Executing unsupported
instruction at pc: %x. Exiting\n", pc );
```

This message is written to stderr.

Printing from Bluespec: Message




```
$display("pc: %h inst: (%h) expanded: ", pc,
inst, showInst(inst));
```

This message is written to stdout *every clock cycle*.

Output streams

- Stderr
  - Piped to stdout in the smips test script
  - Contains statements printed by software and error messages
- Sdtout
  - Piped to build/<program_name>/simOut
  - Contains debug messages about the processor status

Debugging

- To debug problems, look at simOut

```
…
Cycle    3057 ---------------------------------
pc: 00001098 inst: (03e00008) expanded: jr 'h1f

Cycle    3058 ---------------------------------
pc: 00000004 inst: (aaaaaaaa) expanded: nop
```

- To debug further, you can add more $display(…) statements