

Lab 2b

CSE141L

5/2

Announcements

- Teams
 - Everybody signed up?
- Lab 1c amnesty policy
 - 85% credit if submitted by next Monday
- Lab 2a
 - 5/2(W)
 - email draft of your manual and list of your team members
 - 5/4(F)
 - email revised draft of your manual and one benchmark
 - 5/11(F)
 - lab 2a + lab2b together (hardcopy + email)

Lab 2b

- Assembler
- Simulator

Assembler 1/2

- Manually writing machine code
 - Time consuming
 - Error prone
 - Monotonous
 - Hard to maintain
 -(bad bad bad)

Why not Automate?

Assembler 2/2

- Inputs
 - A program source written in your own assembly language
 - Instruction start address (entry point)
 - Data start address
- Outputs
 - *\$name_i.coe*
 - *\$name_d.coe*

*.coe

- Why coe?
 - Initialize vector format in Xilinx CoreGen
 - Your RAM modules will be initialized as specified in a *.coe file

◊ *Sname_i.coe* - 17 bit word size

```
MEMORY_INITIALIZATION_RADIX=16;  
MEMORY_INITIALIZATION_VECTOR=  
00000,  
00001,  
00002,  
00003,  
00004,  
00005,  
.....,  
1DEAD  
EOF
```

◊ *Sname_d.coe* - 34 bit word size

```
MEMORY_INITIALIZATION_RADIX=16;  
MEMORY_INITIALIZATION_VECTOR=  
000000000,  
000000001,  
000000002,  
000000003,  
000000004,  
000000005,  
.....,  
3DEADBEEF  
EOF
```

Simulator

- Why?
 - verify your ISA
 - debug your applications
 - improve your ISA by spotting performance bottlenecks in benchmark programs
- Inputs
 - *\$name_i.coe*
 - *\$name_d.coe*

Simulator Commands

- ◆ *load \$i_file.coe \$d_file.coe*
load *.coe files into instruction memory and data memory
- ◆ *go \$number*
simulate next *\$number* instructions
- ◆ *dump_reg*
print values in all the registers
- ◆ *set_reg \$reg_num \$value*
set the register *\$reg_num* with the value *\$value*
- ◆ *dump_imem \$addr \$size*
print instruction memory values from *\$addr* to *\$addr + \$size*
- ◆ *set_imem \$addr \$value*
set the value at *\$addr* of the instruction memory with the value *\$value*
- ◆ *dump_dmem \$addr \$size*
print data memory values from *\$addr* to *\$addr + \$size*
- ◆ *set_dmem \$addr \$value*
set the value at *\$addr* of the data memory with the value *\$value*
- ◆ *instr_count*
show the number of executed instructions

Lab 2b

- A detailed instruction will be posted soon

Questions?

- Lab 2b
- Lab 2a