2017 Architecture

Project 1 – Instruction simulator

Outline

- Workstation setting : MobaXterm
- Project

Outline

- Workstation setting : MobaXterm
  - Please check “ project_setup.pdf ”
- Project

Outline

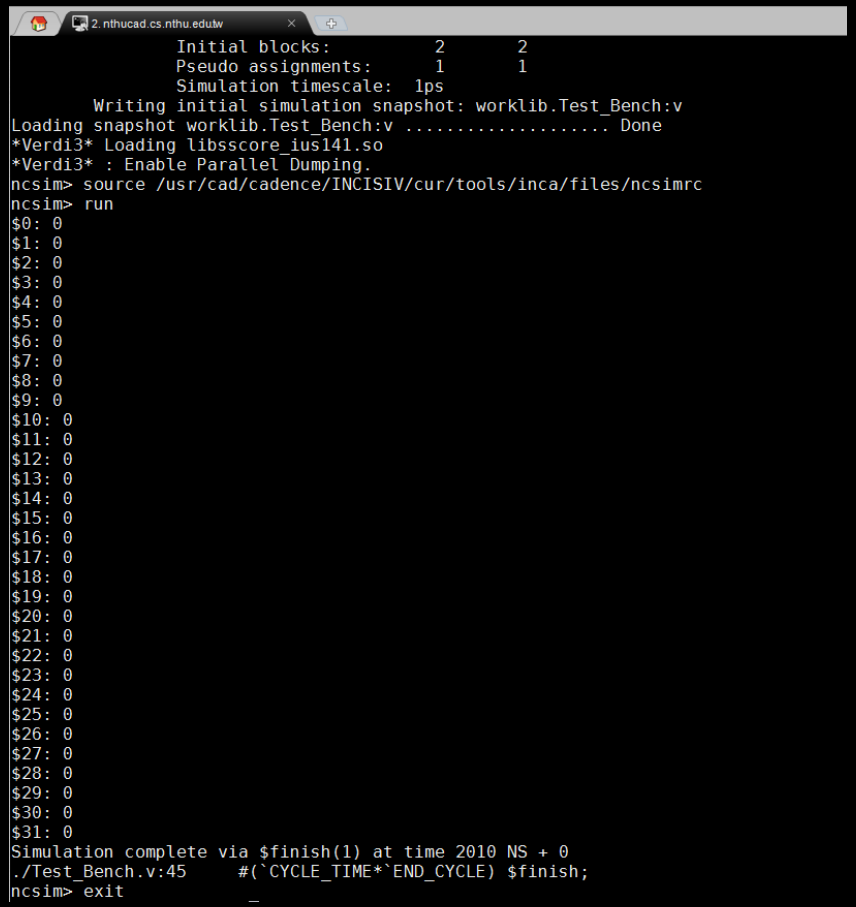
- Workstation setting : MobaXterm
- Project

Project Goal

- Implement chosen instructions
- We provide you:
  - Simulator.v
    - Already a prototype
  - Test_Bench.v
    - Read instruction from testcase
    - Generate clk/rst signal and simulator instance
    - Print out simulator's register value
  - 3 example testcase files

How to run simulation

- $ ncverilog *all .v files you need*
  - i.e. $ncverilog Simulator.v Test_Bench.v



R-type instruction

Instruction	Example	Meaning	Op field	Function field
ADD(Addition)	add r1, r2, r3	r1 = r2 + r3	0x00	32(0x20)
SUB(Subtraction)	sub r1, r2, r3	r1 = r2 - r3	0x00	34(0x22)
AND(Logic And)	and r1, r2, r3	r1 = r2 & r3	0x00	36(0x24)
OR(Logic Or)	or r1, r2, r3	r1 = r2 | r3	0x00	37(0x25)
SLT(Set on Less Than) signed comparison	slt r1, r2, r3	if (r2 < r3) r1 = 1 else r1 = 0	0x00	42(0x2a)

I-type instruction

Instruction	Example	Meaning	Op field
ADDI(Add Immediate)	addi r1, r2, 100	r1 = r2 + 100	0x08
LW(Load word)	lw r1, 0(s1)	R1 = 4 byte from Mem[s1+0]	0x23
SW(Store word)	sw r1, 0(s1)	4 byte from Mem[s1+0] = r1	0x2B
SLTI(Set on Less Than Immediate)	slti r1, r2, 10	if(r2 < 10) r1 =1 else r1 = 0	0x0A
BEQ(Branch On Equal)	beq r1, r2, 25	if (r1 == r2) go to PC+4+100	0x04

Immediate is signed for these instructions

Initial status

- Data memory and register should be set to 0 at the beginning
- PC starts with 0

- Be able to reset program by rst_i

Hand in your project

- Please upload all your .v file to iLMS
  - No file I/O
  - No clk and rst setting
  - You don’t have to upload Test_Bench.v
  - DO NOT compress your file

- There shouldn’t be $stop in your code

Violation: -30 points

Notice

- Don’t modify name, usage, size
  - Top module Simulator.v Name: Simulator
  - Register file : [32-1:0] Reg_File [0:32-1]
    - Store value of register
    - Simulator.Reg_File
  - Instruction memory: [32-1:0] Instr_Mem [0:256-1]
    - Store value of instruction memory
    - Simulator.Instr_Mem

Violation: -30 points

Notice

- Copy or you let other copy **Get 0 point**
- Discussion is encouraged
  - You have to do your own work
- Take good care of your code
  - Some online storage space might not be safe

Test case

- 3 types of test cases

Type	It will cover
1	ADD, SUB, AND, OR, SLT, ADDI, SLTI
2	type 1 + BEQ
3	type2 + LW,SW

Example testcase

- Testcase 0
  - Format : Each line = one assembly code

testcase_0.txt

```
1  00100000000000010000000000000101
2  00100000000000010000000000000110
3  00000000001000100001100000101010
4  00000000001000100010000000100000
5  00000000100000110010100000100010
6  00000000101001000011000000100100
7  00101000101001110000000000010100
8  00000000110001000100000000100101
```

Testcase_0.txt

addi $1, $0, 5
addi $2, $0, 6
slt $3, $1, $2
add $4, $1, $2
sub $5, $4, $3
and $6, $5, $4
slti $7, $5, 20
or $8, $6, $4

Result
$1=5,$2=6,$3=1, $4=11,$5=10,$6=10 $7=1,$8=11

Example testcase

Testcase_1.txt

```
addi $1, $0, 19
addi $2, $0, 8
beq $0, $0, 2
slt $3, $2, $1
add $4, $1, $2
sub $5, $4, $3
and $6, $5, $4
slti $7, $5, 20
or $8, $1, $2
```

Result

```
$1=19,$2=8,$3=0,
$4=0,$5=0,$6=0
$7=1,$8=27
```

Testcase_2.txt

```
addi $1, $0, 19
addi $2, $0, 8
sw $1, 0($10)
beq $0, $0, 1
slt $3, $2, $1
add $4, $1, $2
sub $5, $4, $3
beq $2, $0, 3
and $6, $5, $4
slti $7, $5, 20
lw $8, 0($10)
or $9, $1, $3
addi $1, $1, -4
```

Result

```
$1=15,$2=8,$3=0,
$4=27,$5=27,$6=27
$7=0,$8=19,$9=19
```

Deadline

- Deadline : 4/3(Mon), 23:59
- No late submission is allowed