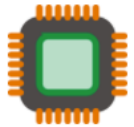

CS305: Computer Architecture

World of Instructions-III (The MIPS language)

https://www.cse.iitb.ac.in/~biswa/courses/CS305/main.html

https://www.cse.iitb.ac.in/~biswa/

Logistics

Get your hands dirty with lab assignments.

TAs (Rwitaban and Shashank) have done all the hard work to make it happen. You will learn a lot by doing things. Assigment-1 will be MIPS instructions in action.

A digression/clarification on the next slide: As per assignment zero.

Instructor Biswa and THE Biswa

The only thing that is in common:
we, both are from the state of Odisha ☺

Do not expect a stand-up from the instructor ☺

A quick recap

Von Neumann (stored program) concept

As registers are limited, data can be there in the registers or in the memory

Register accesses are through register names/numbers

Memory accesses are through addresses stored in registers

Let’s move on: Decision Making Instructions

- Decisions: if, else ….

Two instructions: beq (branch equals to) and
bne (branch not equals to)

```
beq $t0, $t1, L1
bne $t0, $t1, L1
```

Branch Instructions: Conditional branches

beq $t0, $t1, L1

goto L1 (statements labeled as L1) if $t0 equals $t1

bne $t0, $t1, L1

goto L1 (statements labeled as L1) if $t0 does not equal to $t1

The slt instruction (Set on less than)

```
if (a < b)  // beq and bne won't work here
  c=1
else
  c=0
```

```
slt $t3, $t1, $t2  // t1 and t2 contain a and b
```

We can slti too; one of the operand will be a constant

Loops: How to deal with it?

```
while(CS305[i] == k)
    i+=1;
```

say i and k are in $s3 and $s5, and the base of CS305 in $s6

Loops continued

```
while(CS305[i] == k)
    i+=1;
```

1. LOAD CS305[i], base address of CS305 is in $s6
2. We need to go to CS305[i]
3. Assuming CS305 is an integer array, each index is of 4 bytes. We need to go to CS305 [i*4 bytes]

Loops contd. ($s3=i, $s5=k, $s6=base address)

```
sll $t1, $s3, 2          // i*4
add $t1, $t1, $s6   // address of CS305[i]
lw $t0, 0($t1)        // t0 = CS305[i]
bne $t0, $s5, Exit // go to Exit if CS305[i] not equals to k
addi $s3, $s3,1     // i=i+1

Exit:                        // do nothing
```

```
while(CS305[i] == k)
  i+=1;
```

Where is the Loop?

Loops continued

```
Loop: sll $t1, $s3, 2        // i*4
      add $t1, $t1, $s6   // address of CS305[i]
      lw $t0, 0($t1)       // t0 = CS305[i]
      bne $t0, $s5, Exit // go to Exit if CS305[i] not equals to k
      addi $s3, $s3,1     // i=i+1

Exit:                      // do nothing
```

```
while(CS305[i] == k)
  i+=1;
```

How to jump to the Loop?

Loops continued

```
Loop: sll $t1, $s3, 2          // i*4                    while(CS305[i] == k)
      add $t1, $t1, $s6   // address of CS305[i]    i+=1;
      lw $t0, 0($t1)        // t0 = CS305[i]
      bne $t0, $s5, Exit // go to Exit if CS305[i] not equals to k
      addi $s3, $s3,1     // i=i+1
      j      Loop             // go to loop. j here is jump


Exit:                    // do nothing
```

Sequential execution and jumps

PC, PC+4, PC+8, ……

PC, PC+4, {if condition here, TRUE} PC+32, ……

j instruction loads an immediate into the PC. It can be either specified as an offset or the label (assembler will convert this label into an offset). Next lecture: jr, jal, ..

Dhanyavad