CS61C: Assembly, Labels & Calling Conventions

CS61C Fall2007 - Discussion #5

Greg Gibeling

Comics

- My Favorites:
  `http://gdgib.gotdns.com/~gdgib/Comics.aspx`
- Relevant
  - http://xkcd.com/292/
  - http://xkcd.com/293/

Assembly Labels

- Example:
  `N: .word 0`
  - N: A constant number, the address of the word
  - .word: Allocate a word's worth of storage
  - 0: fill the allocated storage with 0
- Basic Use
  ```
  lw $t0, 25($t1) # $t0 = *($t1 + 25)
  sw $t0, 25($t1) # *($t1 + 25) = $t0
  ```
- Advanced Uses
  `sw $s0, 10+N($a1)`
  - 10+N is a constant computed by the assembler

  `sw $s0, N # sw $s0, N($0)`
  - Simple shorthand (useable only because of $0!)
- PsuedoInstructions
  ```
  la $t0, N # lui + ori as needed
  li $t0, N # should be the same as la…
  ```


Width & Meaning

- C has variables
  - Can be stored in memory or a register
  - Variable is just a name for a value
- Assembly has locations
  - Locations can be named (registers & labels)
  - The value in a location can change, the location cannot

		Meaning	
		Known	Unknown
Width	Known	Java	Assembly
	Unknown	C	

Branches

- Instructions
  - MIPS: beq, bne, j, jar, jal
  - IA32 (x86): many more branches
- Control Flow
  - Normally instructions are executed sequentially
  - Branches change this in some way
- Costs
  - Very slow in modern processors
    - Superscalar processors
    - Delayed branches
  - Major source of design complexity

HW4Q2 Optimized

- C Code
  ```
  int compare (int a, int b) {
    if (sub (a, b) >= 0) return 1;
    else return 0;
  }

  int sub (int a, int b) {
    return a-b;
  }
  ```
- MIPS Code
  ```
  slt $v0, $a1, $a0
  jr $ra
  ```

Quiz10 Optimized

```
gcd: bne $a1,$0,recursive_case
                # if b == 0 …
     add $v0,$a0,$0 # gcd = a
     jr $ra
recursive_case:
     div $a0,$a1     # hi=a%b, lo=a/b
     mfhi $t0        # $t0 = hi
     add $a0,$a1,$0 # gcd = gcd (b, a%b)
     add $a1,$t0,$0
     j gcd
```


Stack Management

- See the project 2 background first!
- Always draw diagrams when confused
  - Label absolute addresses (not relative ones!)
  - Sketch with pencil and paper

Open Question

- Problem: Design A Processor
  - How many and what kind of registers?
    - Native data width? MIPS is 32…
  - How many and what kind of instructions?
    - Will you have branches?
    - Encoding?
  - Superscalar? Vector? Dataflow? OOO?
- Assignment: Explain Your Thinking
  - Groups of 4 or 5 (no more)
  - I'll answer ANY questions