Lecture 3

Assembly language
Say something once, why say it again?

CS 241: Foundations of Sequential Programs
Spring 2017

Troy Vasiga et al. (where Dan Holtby ∈ al)
University of Waterloo
Review

- MIPS has 18 instructions
- CPU contains 32 32-bit registers
- MIPS reference sheet for encoding of instructions
- cs241.wordasm to create binary files
- mips.twoints to start MIPS machine
Special Registers

- $0
- $30
- $31
Assembly language

- encoding details can be automated
- 1-1 correspondence with machine language (almost)
- easier on the human eye
- uses different version of the assembler
Example 0 revisited

▶ A1 level (machine language):

00000000 10100111 00011000 00100000

▶ A2 level (assembly language):

add $3, $5, $7

▶ Which one would you rather read and debug?
Example - adding two immediate values

lis $1 ; $1 <- 13
.word 13
lis $2 ; $2 <- 42
.word 42
add $3, $1, $2 ; $3 <- $1 + $2
jr $31
Example - Finding absolute value

C

if ($1 < 0) $1 = -$1

MIPS

slt $2, $1, $0 ; $2 is set iff $1 < 0
beq $2, $0, 1 ; skip 1 instruction if $2 not set
sub $1, $0, $1 ; $1 <- 0 - $1
jr $31
Please do (comment character is a semi-colon).

Its helpful (for you) if you document what each register is being used for.

(This also helps us help you in office hours)
Loop example

lis $4          ; $4 <- 14
.word 14
add $3, $0, $0 ; sum <- 0
lis $1          ; $1 <- 1
.word 1
add $2, $0, $1 ; i <- 1

;;===loop body===
beq $4 $2, 3
add $3, $3, $2 ; sum += i
sub $2, $2, $1 ; --i
beq $0, $0, -4 ; goto loop test at top

;;==return to OS===
jr $31