2318

9/6/16

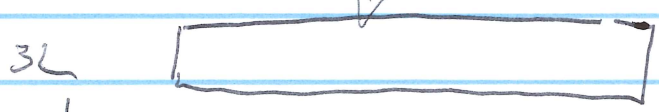
R-Type instructions

Mainly used for Arithmetical Logical instructions

(op R1, R2, R3)

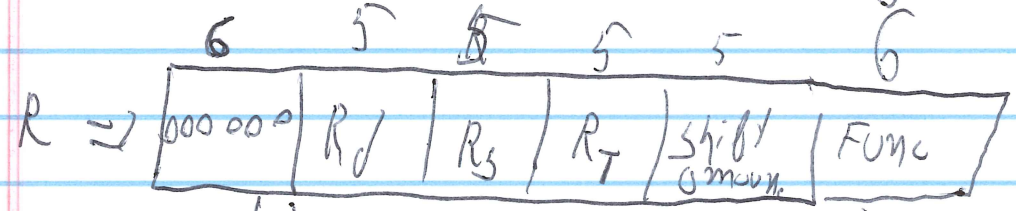
ALU R1 ← R2 op R3

only with GPR



Shift
ADD
OR XOR

Machine code 32 registers



3 operand instruction (compiler "likes")

2 n objects represent each object

$\lceil \log_2 n \rceil$ $32 \rightarrow 5$

IR => 4 bits

4 bits how many unique objects can be encoded ?

2^4

Consider a 2 operand arith ALU

ADD R1, R2 Not in MIPS

R1 ← R1 + R2

op R1 R2 R1 ← R1 op R2

dst source

implied

3 $f = (g+h) - (i+j)$

s_0 s_1 s_2 s_3 s_4 ← Register allocation

add $Rt, $Rs

sub $Rt, Rs

$Rt ← $Rt ± $Rs

Add $5, $6] any Register

sub $11, $21

Assume $Zero ≡ 0

sub $t0, $t0 $t0 ← 0

t0 — add $t0, $s1 $t0 ← s1 (g)

add $t0, $s2 $t0 ← g+h

t1 [some var i+j in $t1

4 add R1, R2

R1 ⇐ R1 + R2

↑ ss

0 f = (g+h) - (i+j)

t0 gets g+h

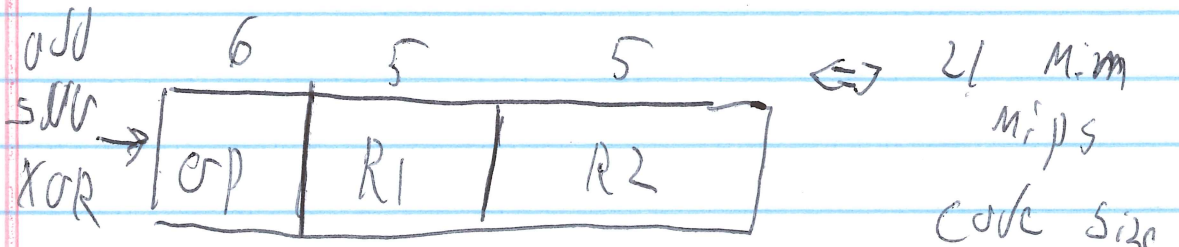
t1 get i+j

, inst to get it all in t0

inst zero the reg for f

~~add t1, t0~~ place to in f

Why is 2 operand better than 3?



⇔ 21 min MIPS code size

16 bit per instruction

5

2 operand

Better code size

3 operand

Better compiler support

ARM has 2 sets of ISA

1) good support for compiler

on the account of code size

2) Minimize code size

Real instructions

↓

Machine code

Pseudo instruction

↓

Real instruction[s]

↓

Machine code

6 MIPS mov pseudo inst.

or $Rd, $Rs, $Rt Real

mov $Rd, $Rs

mov $t1, $zero

↓ can only use $1 for temp

or $t1, $t2, $0

$t1 ← $t2

neg a ← -(a)

neg $t1, $t2

sub $t1, $0, $t2

$t1 ← -($t2)

7 ADD SUB, XOR, OR, ~~NOT~~ AND

NOR, ~~XOR~~, NAND

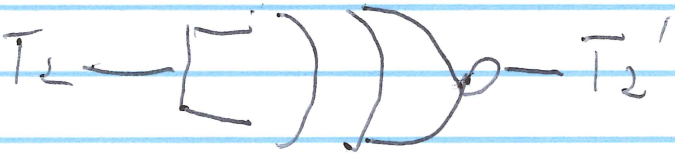
↖ bitwise

.NOT $t1, $t2

Universal { NOR $t1, $t2, $t2

NAND $t1, $t2, $t2

Universal



Assignment take pseudo inst

convert into real

8

Using Memory

(// ong data structure)

// arrays

Mem is an array id

Consider Arrays

→ int A[20]; - static

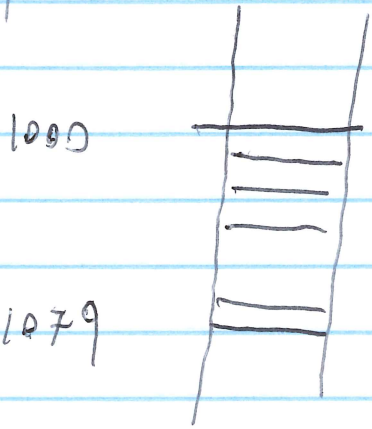
g = h + A[8] ⇐

↑

Memory access

for loader

Allocate 8 bytes



A ← 1000

compiler

Table A is an array

base address is 1000

9 int A[20]; Allocate memory Assign A as pointer

g = h + A[8] (lose address)