CS2318

09/20/2016

SLL / SRL

SLL, $5, $6, $7

shift left $6 by the number

stored in the least significant

5 bits of $7 place result in $5

SLL $5, $6, 2

$5 ← $6 << 2

multiplying by 4 / divide

Using MUL

SHL by 1 is multiplying by

SHR by 1 is int divide

0000 11 0000...1

3 mult $6, $7 (→ $8, $9)

mfhi → $8

mflo → $9

mul $6, $7, $8

mult $6, $7
mflo $8

DIV $6, $7

[Hi:Lo] = $6 divided by $7

mfhi $8 $8 ← $6 % $7

mflo $9 $8 ← $6 / $7 int division

DIV $6, $7, $8

DIV $7 / $8
MFLO $6

2

XOR (XORI)

XOR is useful for swap

copy, encryption

- Detect difference

```
A    01101001
B    11011000
     --------
     10110001
```

MULT (Real)

MULT $R_s, $R_t

[Hi iLo] ← $R_s * $R_t

mfhi, mflo

MUL $1

4

ORI $5, $0, 0xABCD

$5 ← 0xABCD

li $5, 0xABCD 16 bit

JR $5 PC ← $5

pseudo [li $6, 0xABCD
li $6, 0xABCD1234

LUI Load upper immediate

LUI $Rd, constant

$Rd[31:16] ← constant

$Rd[15:0] ← 0

5

LUI $7, 0XABCD

| ABCD | 0000 | $7 |

→ 1) LI $9, 11000

ori $9, $0, 11000

LI $9, 0XABCD

| 0000 | ABCD |

ORI $9, $0, ~~0~~ ~~X~~ ABCD

→ 2) li $9, 0XABCDEF12

LUI $9, 0xABCD

ORI $9, 0xEF12

ADDI $5, $6, 200 — 16

ADDI $5, $6, -500 — 16

6 ADDI is doing sign extension

ADDI $5, $6, -2

to

-2 of 16 -2 of 32

ORI $5, $7, 0xABC

32 16

Zero padding

16 -> 32 Logical operations

.7

BLE R_s, R_t, LABEL

$t2, $t3

LE inverse of GT

SLT $at, $T3, $T2

$at $T2 $T3

beq $at, $0, out

0	3	4	4
0	4	4	0
1	4	3	0

convex
vs temp

$6

ABS $Rd, $Rs

default $R_s \geq 0$

OR $6, $7, $0 Mov

SLT $at, $7, $0

beq $at, $0, out1

sub $6, $0, $7 ← neg

ADD Addition Subtraction

Can Result in overflow

OR cannot overflow

Mov ADD ORi
SUB
OR ← does not overflow

ADDU - ADD with no ovf

SUBU with no overflow

(Unsigned)

to t1

9 SLEI $R_d, $R_s, constant

~~SLTI $R_s, cons1~~

→ ORI $t0, $0, 1

ORI $at, $Zero, constant

SLT $at, $at, $t1

beq $at, $zero, Done

ORI $t0, $zero, 0

→ BGE $5, $6, out

$t1 constant

ORI $at, $0, constant

SLT $t0, $at, $t0