4/7/2020

MIPS Programming : Example

- GCD v1

  Euclid's algorithm for greatest common divisor

- MIPS reference 1, 2

MIPS Instruction Encoding & Decoding

- R-format : Arithmetic & Logic

GCD v1 line 19 : move $s0, $v0 (pseudo instruction)

addu $16, $0, $2 ≡ addu $rd, $rs, $rt

opcode = 000000 ⇒ R-type, funct 100001

rs = 0, rt = 2, rd = 16

	opcode	rs	rt	rd	shift	funct
encoding :	000000	00000	00010	10000	00000	100001
	0	0	0	2	8	0	2	1

- I-format : Arithmetic & Logic

GCD v1 line 31 : la $a0, TXT1 ≡ lui $1, upper(TXT1)
ori $4, $1, lower(TXT1)

lui $1, upper(TXT1) ≡ lui $rt, imm

opcode = 001111, rt = 1, imm = (0x10010000) >> 16

encoding : 001111 . 00000 . 00001 . 0001 0000 0000 0001

3 C 0 1 1 0 0 1

ori $4, $1, lower(TXT1) ≡ ori $rt, $rs, imm

opcode = 001101, rs = 1, rt = 4

imm = (0x10010000) & 0xff

encoding : 001101 . 00001 . 00100 . 0000 . —— . 0000

3 4 2 4 0 0 0 0

I-format : Branch

line 31 : beq $s1, $0, L1 ≡ beq $rs, $rt, offset

opcode = 000100, rs = 17, rt = 0

offset = (addr(L1) − PC) >> 2
next instr

= (0x00400050 − 0x0040003c) >> 2

= (0x14 = 20) >> 2

= 5

encoding : 000100 . 10001 . 00000 . 0000 0000 0000 0101

1 2 2 0 0 0 0 5

decoding : addr = (offset << 2) + PC

= (0x14 + 0x0040003c)

= 0x00400050

GCD V2-V3