CS2100 - Tutorial 4 - MIPS: Memory & Control Flow Instructions with Encoding

Week 6

1 (a).

Variable Mappings	Comments
address of array "A[]" ➔ $s0 i ➔ $s1 (**initialized to 5**) j ➔ $s2	
`addi $s2, $zero, 1`	`j = 1`

`loop:`
`slt $t8, [ $s1, $s2 ]`
`b[ne $t8, $zero ], end`
	`while (j <= i) {`

`t1 [ addi $t2, $s2, -1 ] # j-1`
`t2 [ sll $t3, $t2, 2 ] # (j-1)*4`
`t3 [ add $t4, $s0, $t3 ] # &A[j-1]`
`t4 lw $t0, 0($t4)`
	`t0 = A[j-1]`

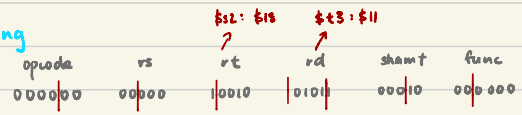
`t5 sll $t3, $s2, 2`
`t6 add $t5, $s0, $t3`
`t7 lw $t1, 0($t5)`
	`t1 = A[j]`

`t8 s[lt] $t8, [ $t1, $t0 ]` `t9 b[eq $t8, $zero ], skip`	`if (a[j-1] > a[j] ) {`
`t10 sw $t1, 0($t4)` `t11 sw $t0, 0($t5)`	`//swap a[j-1] with a[j]` `}//end of if`

`skip:`
`t12 addi $s2, $s2, 1`
`t13 j loop`
	`j++`
	`} //end of while`

`t14 end:`	

(b). Encoding



`sll $t3, $s2, 2` = 0x

0	0	1	2	5	8	8	0

(c). Branches

t13 = 0xD

(d). Jump

$$\begin{array}{r} \text{CF21AB00} \\ -\ \ \ \ \ \ \ \ \ 38 \\ \hline \text{CF21AAC8} \end{array}$$

0x C F 2 1 A B 0 0

1100 1111 0010 0001 1010 1011 0000 0000

$14 \times 4 = 56_{10} = 0x\ 38$

loop is 14 words away.

⇒ new address: 1100 1111 0010 0001 1010 1010 1100 1000

Instruction: (opcode) 000010 1111 0010 0001 1010 1010 1100 10

⇒ 0x 0BC86AB2

(e). MIPS

Purpose: First iteration of bubble sort ✗

Exchange adjacent items if they are not in ascending order

(f). MIPS

A[] = { 3, 4, 1, 2, 6, 5, 7 } (7, 5)

Only perform 5 times

(g). MIPS Execution

Initial addi: 1

Each loop: if swap: 15

if no swap: 13

No. of swaps: 5

No. of loops: 6

∴ Total no. of instructions: 5×15 + 13 + 1 = 89 ✗

Code "Section"	Total Instructions
1st addi = 1 instruction	**1 instruction**
"loop" until "skip" + after "skip" 11 + 2	executed 5 iterations **13 x 5 = 65 instructions**
if-body 2	executed 4 times (for out of order pairs) **2 x 4 = 8 instructions**
slt + beq for the last iteration 2	 **2 instructions**
Grand Total	**76 instructions**

2. Instruction Encoding

Instruction Encoding	MIPS Code
	# **$s1** is the result, **$t0** stores a non-negative number
0x20100000	addi $s1, $zero, 0 #Inst. address is 0x00400028
0x00084042	loop: srl $t0, $t0, 1
0x11000002	beq $zero, $t0, exit
0x22310001	addi $s1, $s1, 1
0x08100000C	j loop
	exit:

(i). op code | st | sd → $16_10 = 17_10 | Imm

001000 | 00000 | 10000 | 00...0 (16)

⇒ 0x 2010 0000

0x 2011 0000

(ii). 0x 1100 0002 : beq | st: $t0 | sd: $zero | Imm: +2

0001 0001 0000 0000 | 0000 0000 0000 0010

(iii). 0x 2231 0001 : addi | st: $s1 | sd: $s1 | Imm: +1

0010 0010 0011 0001 | 0000 0000 0000 0001

(iv). loop address: 0x00400028 + 4 = 0x0040002C

Destination : 0000 0000 0100 0000 0000 0000 0010 0000

0000 0000 0100 0000 0000 0000 0010 1100

∴ Instruction 0000 10 0000 0100 0000 0000 0000 0001 00

⇒ 0x08100000C

0000 10 0000 0100 0000 0000 0000 0010 11

⇒ 0x08100008

(b). ?

١٠