Course : Computer Organization –

ENCM 369

Lab # : Lab 5

Student Names : Nimna Wijedasa, Magdy Hafez

Lab Section : B04

t5 0x 6a00_9005
t6 0x 0fff_a000

```
lui   t0 , 0xb670
srli  t1 , t0 , 5
slli  t2 , t0 , 3
or    t3 , t5 , t6
andi  t4 , t5 , 0x3ff
xor   s0 , t5 , t6
xori  s1 , t5 , -16
```

final values

t0 = 0x 0b67_0000
t1 = 0x 00 5b_3800
t2 = 0x 5b38_0000
t3 = 0x 6fff_b005
t4 = 0x 0000_0005
t5 = 0x 6a00_9005
t6 = 0x 0fff_a000
s0 = 0x 65ff_3005
s1 = 0x 95ff_6ff5

srli (5)

t0 0000 1011 0110 0111 0000 0000 0000 0000
0000 0000 0101 1011 0011 1000 0000 0000
0 0 5 B 3 8 0 0

slli (3)

t0 0000 1011 0110 0111 0000 0000 0000 0000
0101 1011 1100 1000 0000 0000 0000 0000
5 B 3 8 0 0 0 0

or

t5 0110 1010 0000 0000 1001 0000 0000 1010
t6 0000 1111 1111 1111 0101 0000 0000 0000
0110 1111 1111 1111 1101 0000 0000 1010
6 F F F D 0 0 A

andi

t5 0110 1010 0000 0000 1001 0000 0000 1010
0000 0000 0000 0000 0000 0011 1111 1111
0000 0000 0000 0000 0000 0000 0000 1010
0 0 0 0 0 0 0 A

xor

t5 0110 1010 0000 0000 1001 0000 0000 1010
t6 0000 1111 1111 1111 0101 0000 0000 0000
0110 0101 1111 1111 1100 0000 0000 1010
6 5 F F C 0 0 A

xori t5

0110	1010	0000	0000	1001	0000	0000	0101
1111	1111	1111	1111	1111	1111	1111	0000
1001	0101	1111	1111	0110	1111	1111	0101
9	5	f	f	6	f	f	5

16 = 0001 0000

-16 = 1110 1111 + 1

2's compliment = 1111 0000 //

---

(1) value of auipc address of auipc shifted 12 bits = 0x0fc1_0000
= 0x0040_0000

0x1001_0000 //

(2) '16' is in decimal but when converted to hex it is '10'

adding it to t0 gives us the addrss of alpha 0x1001_0000
0x0000_0010
0x1001_0010 //

(3) value of auipc address of auipc shifted 12 bits = 0x0fc1_0000
= 0x0040_000c
0x1001_000c //

(4) '8' is in decimal also '8' but when converted to hex it is

adding it to t3 gives us the addrss of Beta 0x1001_000c
0x0000_0008
0x1001_0014 //

---

(1)

```
L1 :   lbu    t6, (s1)          0x0040_1034   3C
       beq    t6, zero, L2                    38   40
                                              24   a0
       addi   s1, s1, 1                       a4
       j      L1                              a8
L2 :   or     s3, s1, zero      0x0040_10ac
```

machine code for beq

t6, zero, L2

$$\text{offset} = \frac{0x0040\_10ac - 0x0040\_1038}{4}$$

$$= 29 \text{ jumps} = 116 \text{ bytes}$$

7	5	5	3	5	7
0000 011	00000	11111	000	10100	1100011
	zero	t6	Beq	Jump	Branch

hex = 0x60F8A63

(2)

$$\text{offset} = \frac{0x0040\_1034 - 0x0040\_10a8}{4}$$

$$= -2a \quad x4 = -116$$

116 = 111 0100

000 1011

1111	1111	1111	1000	1100
imm			rd	op
1	111000	1	1111	00000	110 1111
20	10:1	11	19:12		

Hex = F8DFF06F

Exercise D

run 5:
pointers: 178258
indexes: 196672

run 10:
pointers: 180497
indexes: 196784.00

run 6:
pointers: 180647
indexes: 196751.00

PART I

Avg ptrs: 179800.67

Avg index: 19673.5.67

Speedup = $\frac{196735.67}{179800.67} = 1.094$

Part II

run 9:
pointers: 49415.00
index: 49398.00

run 4
pointers: 49421.00
index: 49392.00

run 2:
pointers: 49454
index: 49431

AVG pointers: 49430

AVG indexes: 49407

Speedup pointers: $\frac{179800.67}{49430} = 3.637$

Speed up indexes: $\frac{196735.67}{49407} = 3.982$

- Asking for compiler optimization is more important to get the speed. Having compiler optimized has more impact on speedup rate than changing arithmetic to pointers.

run 9:
pointer: 45888.00
index : 45891.00

run 10:
pointer: 45821.00
index: 45895.00

run 4:
pointer: 46880.00
index: 46908.00

Part III

pointer: 46196.33
index : 46231.33

Speedup pointers $= \frac{49430}{46196.33} = 1.07$

Speedup_index $= \frac{49407}{46231.33} = 1.069$

Exercise E:

2)

```
24      jmp .L2
25  .L4:
26      movq    -24(%rbp), %rax
27      movl    (%rax), %eax
28      cmpl    %eax, -12(%rbp)
29      jge .L3
30      movq    -24(%rbp), %rax
31      movl    (%rax), %eax
32      movl    %eax, -12(%rbp)
33  .L3:
34      addq    $4, -24(%rbp)
35  .L2:
36      movq    -24(%rbp), %rax
37      cmpq    -8(%rbp), %rax
38      jne .L4
```

3) . machine code for addl $0x1, -0x8(%rbp) is 83 45 F8 01

4) . for(int i=0; i<((int) 1e5); i++)

. 81 7d c4 9f 86 01 00