4300 HW2

Tuesday, February 23, 2021 1:18 PM

1-7, 9, 10, 12, 18, 21, 22, 26, 29, 38, 47

---

① $f = g + (h - 5)$

test

- $i = h - 5$
- $f = g + i$

```
addi  i, h, -5
add   f, g, i
```

②
```
add  f, g, h
add  f, i, f
```

$f = g + h + i$

③ $B[8] = A[i - j]$

f	g	h	i	j	A	B
s0	s1	s2	s3	s4	s6	s7

```
sub  s5, s2, s3     compute i-j
sll  s5, s5, 2      *4 for memory
add  s5, s6, s5     do A[i-j]
lw   s6, 0(s5)      load into reg
sw   s6, 32(s7)     store into reg
```

(4)

```
sll  t0, s0, 2      t0 = f*4
add  t0, s6, t0     t0 = &A[f]
sll  t1, s1, 2      t1 = g*4
add  t1, s7, t1     t1 = &B[g]
lw   s0, 0(t0)      f = A[f]
```

```
addi t2, t0, 4      t2 = &A[f+1]
lw   t0, 0(t2)      A[f] = A[f+1]
add  t0, t0, s0     t0 = &A[f+A[f]/4]
sw   t0, 0(t1)      B[g] = A[f+A[f]/4]
```

(5) You can include the "lw" line 5 in the

"lw" line 7.

6

24	2
28	4
32	3
36	6
40	1

6.1

```
for (int i=0; i<Array.size()/5; i++)
{
    for (int K=i+1; K<Array.size()/5; K++)
    {
        if (Array[i] > Array[K])
            swap (Array[i], Array[K])
    }
}
```

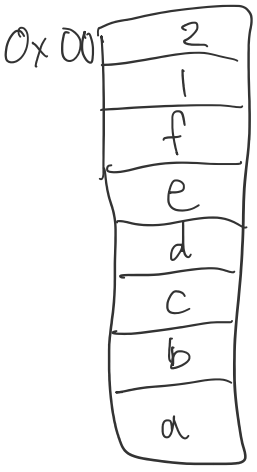
6.2 Array base = $56

```
add $t1, $0, $0
la  $a0, Array($s6)

lw  t2, 0(a0)
lw  t3, 4(a0)

slt t5, t2, t3

beq t5, $0, do_nothing
add t1, $0, 1
sw  t2, 4(a0)
sw  t3, 0(a0)
```

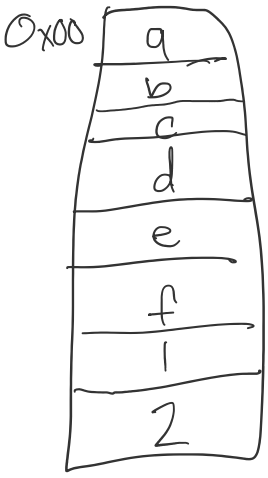
(7)

0xabcdef12

Little Endian



Big Endian



(9)

```
sll $t0, $s3, 2
```

```
add  t0, t0, s6
lw   t0, 0(t0)
sll  t1, s4, 2
add  t1, t1, s6
lw   t1, 0(t1)
add  t2, t0, t1
sw   t2, 32(s7
```

(10)

A[1] = A

f = 2 * A

12, 18, 21, 22, 26, 29, 38, 47

(12)

$s0 = 0x80000000
$s1 = 0xD0000000

(1)

t0 = 5000 0000

(2)

There was overflow

(2) There was overflow

(3)

$8 + 3 = B$

$= 0x\,B000\,0000$

(4) There was overflow (8-D = negative)

(5)

$5 + 8 = 13 = D$

$= 0x\,D000\,0000$

(6) Overflow, again

(18)

~128 Registers

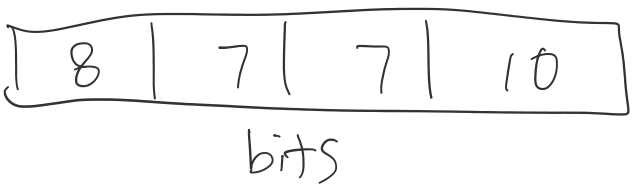
~X4 Instruction count

(1)

8	7	7	2	1

bits

(2)



(3) The shifting operations are lessened, but more instructions = more fail-safes ☺

(21)

```
nor $t1, $t2, $0
```

(22)

```
lw  $t1, 0($s1)    : store C[0] in A
sll $t1, $t1, 4    : shif it 4
```

☺

26, 29, 38, 47

(26)

(1) S2 = loops x2

= 20

(2)

```
for(i=10; i=0; i--)
{
```

```
{
    B += 2
}
```

(28) **5N-3**

5N for every loop, but the last loop doesn't execute the 3 last instructions

(29)

```
for (int i=0; i<100; i++)
{
    result += memarray[i];
}
```

(38)

0x10001 = 0x11
2 = 22
3 = 33
4 = 44

0x00000011

(47)

(1) $CPI = 0.7 \times 2 + 0.1 \times 6 + 0.2 \times 3$

$= \boxed{2.6}$

(2)

$CPI = 0.75 \times 2.6 = \boxed{1.95}$

(nothing else changed)

(3)

$CPI = 1.3 = 0.7 \times 0.5 \times 2.6 + 1.2$

$\boxed{CPI = 0.14}$