Computer Architecture and Programming Languages
Jacobs University Bremen
Jasmine Dayinta

Course: CO20-320241
October 16, 2018

Homework 5 - Addition, Subtraction and MIPS Instruction Set Architecture

Problem 5.1

Solution:

a. 14 + 36 using binary representation

$= 1110 + 100100$

$= 0001110$

$\quad 0100100 +$

$\quad \_\_\_\_\_$

$\quad 0110010$

The answer is: 0110010_2

b. 12 - 25 using binary and 2's complement representation

$12 = 1100_2$ and $25 = 11001_2$

Now, we use 2's complement to get -25

$25 = 11001_2$

$= 00110_2$ (invert)

$= 00111_2$ (add 1)

$\quad 00110$

$\quad 00111 +$

$\quad \_\_\_\_\_$

$\quad 10011$

The answer is: 10011_2

c. 69 + 58 using BCD representation

$69 = 0110\ 1001$ and $58 = 0101\ 1000$

$\quad 0110\ 1001$

$\quad 0101\ 1000 +$

$\quad \_\_\_\_\_$

$\quad 1011\ 10001$ Must add 0110 to both binaries because it exceeds 9

$\quad 1011\ 10001$

$\quad 0110\ 0110 +$

$\quad \_\_\_\_\_$

$\quad 0010\ 0111$ with a carry of 1

The answer is: $0001\ 0010\ 0111_{BCD}$

d. 275 + 642 using BCD representation

$275 = 0010\ 0111\ 0101$ and $642 = 0110\ 0100\ 0010$

$\quad 0010\ 0111\ 0101$

$\quad 0110\ 0100\ 0010 +$

$\quad \_\_\_\_\_$

$\quad 1000\ 1011\ 0111$ Must add 0110 because 1011 exceeds 9

$\quad 1000\ 1011\ 0111$

$\qquad 0110 \qquad +$

$\quad \_\_\_\_\_$

$\quad 1001\ 0001\ 0111$

The answer is: $1011\ 0001\ 0111_{BCD}$

e.6AF + 23C using hexadecimal representation

```
  1          Carry from adding F and C. F + C is 27 and we subtract 16, to get 11 or B
 6AF         A + 1 + 3 adds up to 14 or E
 23C +
 ____
 8EB
```

The answer is: $8EB_{16}$

d. 594 - 3A7 using hexadecimal representation

```
 48          Since 4 is smaller than 7, we take 16 from 9. 16 + 4 is 20 and subtract 7 to get 13
 594         Again, 8 is smaller than A, so we take 16 from 5
 3A7 −
 ____
 1ED
```

The answer is: IED_{16}

Problem 5.2

Solution:

a. Operation: a = b + c
MIPS code: add a, b, c OR add $t0, $s0, $s1

b. Operation: a = b − d + c

```
MIPS code: sub a, b, d    OR    sub $t0, $s0, $s2
           add a, a, c          add $t0, $t0, $s1
```

c. Operation: a = 3 ∗ b

```
MIPS code: add a, a, b    OR    add $t0, $t0, $s0
           add a, a, a          add $t0, $t0, $t0
           add a, a, a          add $t0, $t0, $t0
```

d. Operation: (1 + b) ∗ 2

```
MIPS code: add a, 1, b      OR      add $t0, 1, $s0
           add a, a, a              add $t0, $t0, $t0
```

Problem 5.3

Solution:

a. 000000 10001 10010 010010 0000 10000

b. 000000 10000 10010 010000 0000 100010
000000 01000 10001 01000 00000 100000

Problem 5.4

Solution:

```
lw $t0, 16($s0)        #temp. reg. $t0 gets A[4]
lw $t1, 8($s0)         #temp. reg. $t1 gets A[2]
add $t2, $t0, $t1      #add $t0 and $t1 together
sw $t2, 20($s1)        #stores result
```

Problem 5.5

Solution:

```
add $t1, $t0, $0
add $t1, $t1, $t1
add $t1, $s0, $t1     #adding base address A to $t1
lw $t2, 28($s0)       #loading A[x + 7]
lw $t3, 8($s0)        #loading A[x + 2]
add $t2, $t3, $t2
add $t1, $s1, $t1     #add the base address B to $t1
sw $t2, $t1           #stores result
```

Problem 5.6

Solution:

You could change the instruction format by reducing the number of bits from 5 to 4 bits.
This will result in 2 bits being free for allocation. We can then increase the bits from 16 bits to 18 bits.