Computer Architecture and Programming Languages
Jacobs University Bremen
Arsenij Percov

Course: CO20-320241
October 14th, 2019

Homework 5

Problem 5.1

Solution:

(a) $14_{10} + 37_{10} = 110_2 + 100101_2 = 101011_2$ Carry 1 bit on 3rd position from end.

(b) $12_{10} - 27_{10}$
-27_{10} Sign is 1, convert 27 to binary, invert, and add 1
$27_{10} = 11011_2$
Make it 8 bit
00011011, invert
11100100, add 1
$-27_{10} = 11100101_2$
$12_{10} - 27_{10} = 1100_2 + 11100101_2 = 11110001_2$ Invert again, and add 1
$00001110_2 + 1$=
$1111_2 = -15$is the answer.

(c) 69 = 0110 1001
58 = 0101 1000

		0110	1001	
+		0101	1000	
=		1011	10001	Add 6 to right sum
+		0001	0110	Add carry and 6
=		1100	0111	Add 6 to left sum(bigger than 9)
+		0110	0000	
=		10010	0111	
=	0001	0010	0111	

(d) 275 = 0010 0111 0101
642 = 0110 0100 0010

	0010	0111	0101	
+	0110	0100	0010	
=	1000	1011	0111	Add 6 to the middle sum
+	0000	0110	0000	
=	1000	10001	0111	Carry the bit
+	0001	0000	0000	
=	1001	0001	0111	

(e) 6AF + 23C =

	6	A	F	
+	2	3	C	
=	8	D	1B	Carry last bit
+	0	1	0	
=	8	E	B	

(f) 594 - 3A8 =
First, find 15 compliment of the second number.

$3A8 \xrightarrow{\text{invert}} C57$

	5	9	4	
+	C	5	7	
=	1	E	B	Carry bit to least significant digit
=	1	E	C	

Problem 5.2

Solution:

(a) $a = b + c$
$\$t0 = \$s0 + \$s1$
MIPS:

add $t0, $s0, $s1 Add b and c, and store it in a

(b) sub $t0, $s0, $s2 Subtract b -d and store it in a
add $t0, $t0, $s1 Add a to c and store the result in a

(c) add $t0, $s0, $s0 Add b and b and store it in a
add $t0, $t0, $s0 Add a and b and store it in a

(d) add $t0, 1, $s0 Add 1 to b, and store in a
add $t0 $t0 $t0 Add a to a, and store in a (multiply by 2)

Problem 5.3

Solution:

Let's first transform register # of variables a, b, c to binary numbers.
a = $t0 = 8 = 01000_2
b = $s0 = 16 = 10000_2
c = $s1 = 17 = 10001_2
d = $s2 = 18 = 10010_2

(a)

MIPS instruction	op	rs	rt	rd	shamt	funct
add $t0 $s0 $s1	000000	10000	10001	01000	00000	100000

(b)

MIPS instruction	op	rs	rt	rd	shamt	funct
sub $t0, $s0, $s2	000000	10000	10010	01000	00000	100010
add $t0, $t0, $s1	000000	01000	10001	01000	00000	100000

Problem 5.4

Solution:
B[5] = A[4] + A[2]
lw $t0, 16($s0) Load A[4] into temp. reg. $t0
lw $t1, 8($s0) Load A[2] into temp. reg $t1
lw $t2, 20($s1) Load B[5] into temp. reg $t2
add $t2, $t0, $t1 Add A[4] and A[2] and store it to $t2
sw $t2, 20($s1) Save the result of $t2 to B[5]

Problem 5.5

Solution:
add $t1, $t0 , 7 Add 7 to x, and store it at $t1
add $t1, $t1, $t1 Multiply $t1 by 2.
add $t1, $t1, $t1 Multiply $t1 by 2 (Now we got offset for A[x+7].
add $t1, $t1, $s0 Get adress of A[x+7]
add $t2, $t0 , 2 Add 2 to x, and store it at $t2
add $t2, $t2, $t2 Multiply $t2 by 2.
add $t2, $t2, $t2 Multiply $t2 by 2 (Now we got offset for A[x+2].
add $t2, $t2, $s0 Get adress of A[x+2]
add $t0, $t0, $t0 Multiply $t0 by 2
add $t0, $t0, $t0 Multiply $t0 by 2
add $t0, $t0, $s1 Add offset for B[x]
lw $t3, 0($t1) Load value of A[x+7]
lw $t4, 0($t2) Load value for A [x+2]
add $t5, $t3, $t4. Add value of A[x+7] to A[x+2] and store it at $t5
sw $t5, 0($t0) Store value of sum to B[x] ($t5)

Problem 5.6

Solution:
It will change number of bits used for representing the id of registers. Before it was 5, but now since we have only 16 numbers, we can use 4, since it is enough to represent all of them. Since every MIPS instruction have 32 bits, we will have 2 free bits(1 from first register, one from second), that can be used for representing bigger constant (18 bits instead of 16)