Erik Ayavaca-Tirado

9/27/17

Computer organization

exercises from the text: 2.1, 2.2, 2.7, 2.12.

2.1

addi f, h, -5

add f, g, f

2.2

f = g +h

f = I + f

2.7

Little-endian

|  |  |
| --- | --- |
| Address | Data |
| 12 | ab |
| 8 | cd |
| 4 | ef |
| 0 | 12 |

Big-endian

|  |  |
| --- | --- |
| Address | Data |
| 12 | 12 |
| 8 | ef |
| 4 | cd |
| 0 | ab |

2.12

assume:

$s0=0x80000000 = 1000 0000 0000 0000 0000 0000 0000 0000

$s1=0xD0000000 = 1101 0000 0000 0000 0000 0000 0000 0000

2.12.1

add $t0 , $ts0 , $s1

so, $t0 =

1000 0000 0000 0000 0000 0000 0000 0000

+ 1101 0000 0000 0000 0000 0000 0000 0000

10101 0000 0000 0000 0000 0000 0000 0000

So, $t0 = 10101 0000 0000 0000 0000 0000 0000 0000

So $t0= 0x150000000 because it had an overflow since it was more than 32 bit’s.

2.12.2

There is an overflow as the answer is more than 32 bit’s so it is not the desired result,

2.12.3

$s0 = 8 0x80000000

- $s1 = 13 0xD0000000

$t0 = -5

-5 = 0xFFFFFFFB

2.12.4

The result is the desired value, with no overflow

2.12.5

add $t0, $s0, $s1

$s0 = 1000 0000 0000 0000 0000 0000 0000 0000

$s1 = 1101 0000 0000 0000 0000 0000 0000 0000

$t0 =10101 0000 0000 0000 0000 0000 0000 0000

0x50000000 = $t0

add $t0, $t0, $s0

$t0 = 0001 0101 0000 0000 0000 0000 0000 0000 0000

+$s0 = 0000 1000 0000 0000 0000 0000 0000 0000 0000

$t0 = 0001 1101 0000 0000 0000 00000 0000 0000 0000

or 0x1D0000000

2.12.6

The Desired result is not the correct answer because of an overflow.