Facultad de Informática
Universidad Complutense de Madrid

INTRODUCTION TO COMPUTERS II
MODULE 2

Basic problems:

1. Indicate the result of executing the following instructions in RISC-V, providing the final content of the registers and the memory positions.

a) `add  x1, x1, x2`
b) `addi x3, x2, 2`
c) `sub  x4, x3, x0`
d) `andi x2, x3, 0xf0`
e) `sll  x4, x2, x5`
f) `or   x1, x1, x2`
g) `add x2, x0, x4`
h) `lw  x1, 0(x4)`
i) `lw  x2, 4(x5)`
j) `and x5, x1, x3`
k) `sw  x3, 0(x5)`
l) `sw  x4, 4(x4)`

Assume that, for each instruction, the initial content of the registers and the memory positions is the following:

Registers	
x1	0x00000016
x2	0x00000054
x3	0xffffffff
x4	0x00000000
x5	0x00000004

Memory	
0x00	0x03393826
0x04	0xea0063af
0x08	0x17fa8912
0x0C	0xbc983304
0x10	0x7845f34a
0x14	0x534b4aaa

Additional problems:

2. Explain why the following instructions are not valid:

```
addi x3, 3, x2
add  x3, x2, 0(x1)
beq  x3, 0, 8
beq  x3, x2, 3
slli x3, x3, 40
muli x3, x2, 28
lw   x8, -4000(x1)
```

3. The following constants are placed in memory, starting at position 0x1000:

- 0x10203040 word
- 0x50 byte
- 0x6070 half word
- 0x80 byte
- 0x90a0b0e0 word

Assuming that they are placed in the given order, taking the minimum space possible and following the RISC-V alignment and organization, provide:

a) The initial address of each one.
b) The value of the byte contained in these addresses: 0x1001, 0x1006 y 0x100c.
c) The percentage of wasted memory.

d) An alternate order to reduce the required memory as much as possible.

4. Assuming that register x7 contains address 0x10000000 and that the word data 0x1020d040 is located in such address, indicate the word that is stored in address 0x10000004 after executing the following pairs of instructions:

a)
```
lb  x6, 0(x7)
sw  x6, 4(x7)
```

b)
```
lh  x6, 0(x7)
sw  x6, 4(x7)
```

c)
```
lhu x6, 0(x7)
sw  x6, 4(x7)
```

5. Write the instruction/s needed to load the following constants into register x10:

- 0xabc
- 0x1abc
- 0x12345678
- 0x56789abc

Solutions

1. See slides.

2. The immediate operand must be the right-hand one.
Arithmetic-logic instructions do not support relative addressing.
Branch instructions do not operate with immediate data.
Branch addresses must be multiples of 4.
Shifts can be up to a maximum of 31 bits.
Multiplication instructions do not operate with immediate operands.
The offset exceeds the range representable with 12 bits.

3. a) 0x1000, 0x1004, 0x1006, 0x1008, 0x100c
b) 0x30, 0x70, 0xe0
c) 25%
d) 0x10203040, 0x50, 0x80, 0x6070, 0x90a0b0e0

4. a) 0x00000040
b) 0xffffd040
c) 0x0000d040

5. a)
```
lui  x10, 0x1
addi x10, x10, 0xabc
```
b)
```
lui  x10, 0x2
addi x10, x10, 0xabc
```
c)
```
lui  x10, 0x12345
addi x10, x10, 0x678
```
d)
```
lui  x10, 0x5678a
addi x10, x10, 0xabc
```