Facultad de Informática
Universidad Complutense de Madrid

INTRODUCTION TO COMPUTERS II
MODULE 4

Basic problems:

1. Given the following RISC-V assembly instructions, provide the corresponding 32-bit machine code.

a) **`lw t0, 0(t2)`** b) **`bge t1, t0, 0x2C`** c) **`sw t1, 0(t4)`**

2. Given the following RISC-V 32-bit machine code, provide the corresponding assembly instructions.

a) `0x03528b33` b) `0x00190913` c) `0x0000006F`

Additional problems:

3. Suppose that a new instruction is added to the RISC-V ISA. This instruction reads a word from memory using a base register and a variable offset stored in an index register, **`lwi rd, rs1, rs2`**:

$$\{ \text{rd} \leftarrow \text{Mem}[\ \text{rs1} + \text{rs2}\ ] \}$$

Provide:

a) The most appropriate format, among the existing ones, to encode this instruction.
b) The smallest RISC-V instruction sequence needed to implement it as a pseudo-instruction.

4. Suppose that a new instruction is added to the RISC-V ISA. This instruction loads a word from memory, pre-incrementing the base register, **`lwpreinc rd, imm(rs1)`**:

$$\{ \text{rd} \leftarrow \text{Mem}[\ \text{rs1} + \text{sExt(imm)}\ ],\ \text{rs1} \leftarrow \text{rs1} + \text{sExt(imm)} \}$$

Provide:

a) The most appropriate format, among the existing ones, to encode this instruction.
b) The smallest RISC-V instruction sequence needed to implement it as a pseudo-instruction.

5. Suppose that a new instruction is added to the RISC-V ISA. This instruction loads a word from memory, post-incrementing the base register, **`lwpostinc rd, imm(rs1)`**:

$$\{ \text{rd} \leftarrow \text{Mem}[\ \text{rs1}\ ],\ \text{rs1} \leftarrow \text{rs1} + \text{sExt(imm)} \}$$

Provide:

a) The most appropriate format, among the existing ones, to encode this instruction.
b) The smallest RISC-V instruction sequence needed to implement it as a pseudo-instruction.

6. Suppose that a new instruction is added to the RISC-V ISA. This instruction swaps the value of 2 registers, **`swap rs1, rs2`**:

$$\{ \text{rs1} \leftarrow \text{rs2},\ \text{rs2} \leftarrow \text{rs1} \}$$

Provide:

a) The most appropriate format, among the existing ones, to encode this instruction.
b) The smallest RISC-V instruction sequence needed to implement it as a pseudo-instruction.

7. Suppose that a new instruction is added to the RISC-V ISA. This instruction loads a data read from a certain memory position into a register, **`lwa`** ***`rd, imm`***:

{ rd ← Mem[zExt(imm)] }

Provide:

a) The most appropriate format, among the existing ones, to encode this instruction.
b) The smallest RISC-V instruction sequence needed to implement it as a pseudo-instruction.

8. Provide the RISC-V machine code for each of the following assembly instructions:

a) `sub x5, x6, x7`
b) `addi s0, s1, 12`
c) `lw t2, -6(s3)`
d) `srai t1, t2, 29`
e) `sb t5, 45(zero)`
f) `beq s0, t5, 0x10`
g) `lui s5, 0x8cdef`
h) `jal ra, 0xa67f8`

9. Provide the RISC-V assembly instructions for each of the following machine codes:

a) `0x41fe83b3`
b) `0xfda48293`
c) `0x05ce2a03`

Solutions

1. a) `0x0003a283` b) `0x02535663` c) `0x006ea023`

2. a) `mul s6, t0, s5` b) `addi s2, s2, 1` c) `jal zero, 0`

3. R-type

```
add  rd, rs1, rs2
lw   rd, 0(rd)
```

4. I-type

```
addi rs1, rs1, imm
lw   rd, 0(rs1)
```

5. I-type

```
lw   rd, 0(rs1)
addi rs1, rs1, imm
```

6. R-type

```
add  rs1, rs1, rs2
sub  rs2, rs1, rs2
sub  rs1, rs1, rs2
```

7. U-type

```
lui  rd, imm
srli rd, rd, 12
lw   rd, 0(rd)
```

8. a) `0x407302b3` d) `0x41d3d313` g) `0x8cdefab7`
b) `0x00c48413` e) `0x03e006a3` h) `0x7f8a60ef`
c) `0xffa99383` f) `0x01e40563`

9. a) **`sub`** `x7, x29, x31` b) **`addi`** `x6, x9, -38` c) **`lw`** `x20, 92(x28)`