

Laboratorio 3

INF245

1.2 Conversión a Lenguaje de Máquina RISC-V

A continuación se muestran varios ejemplos de la codificación a máquina de las instrucciones más significativas de nuestro código, para cada instrucción se indica su tipo, el desglose de sus campos de bits y la representación en hexadecimal.

Es importante mencionar que nuestro programa no emplea S-Type (almacenamiento) ni U-Type (LUI/AUIPC). Fuera de eso, hay para todos los demás tipos (I-type, R-type, B-type, J-type).

imm[11:0]	rs1	funct3	rd	opcode
000000001010	00000	000	01010	0010011

Table 1: I-Type: `addi x10, x0, 10`

Hexadecimal: 0x00A00513

imm[11:0]	rs1	funct3	rd	opcode
000000000011	01100	111	00111	0010011

Table 2: I-Type: `andi x7, x12, 3`

Hexadecimal: 0x00367393

imm[11:0]	rs1	funct3	rd	opcode
000000000001	01100	001	01100	0010011

Table 3: I-Type: `slli x12, x12, 1`

Hexadecimal: 0x00161613

funct7	rs2	rs1	funct3	rd	opcode
0000000	01010	01100	000	01100	0110011

Table 4: R-Type: add x12, x12, x10

Hexadecimal: 0x00A60633

funct7	rs2	rs1	funct3	rd	opcode
0100000	01011	01100	000	01100	0110011

Table 5: R-Type: sub x12, x12, x11

Hexadecimal: 0x40B60633

funct7	rs2	rs1	funct3	rd	opcode
0000001	11101	01100	110	11110	0110011

Table 6: R-Type: rem x30, x12, x29

Hexadecimal: 0x03D66F33

imm[12]	imm[10:5]	rs2	rs1	funct3	imm[4:1]	imm[11]	opcode
0	000001	00110	00101	101	0100	0	1100011

Table 7: B-Type: bge x5, x6, end_for

Hexadecimal: 0x0262D463

imm[12]	imm[10:5]	rs2	rs1	funct3	imm[4:1]	imm[11]	opcode
0	000000	00000	00111	001	0110	0	1100011

Table 8: B-Type: bne x7, x0, else_1

Hexadecimal: 0x00039663

imm[12]	imm[10:5]	rs2	rs1	funct3	imm[4:1]	imm[11]	opcode
0	000000	00000	11110	000	1010	0	1100011

Table 9: B-Type: beq x30, x0, if_2

Hexadecimal: 0x000F0A63

imm[20]	imm[10:1]	imm[11]	imm[19:12]	rd	opcode
0	0000000100	0	00000000	00000	1101111

Table 10: J-Type: jal x0, increment

Hexadecimal: 0x0080006F

imm[20]	imm[10:1]	imm[11]	imm[19:12]	rd	opcode
1	1111101110	1	11111111	00000	1101111

Table 11: J-Type: jal x0, loop_for

Hexadecimal: 0xFDDFF06F

imm[20]	imm[10:1]	imm[11]	imm[19:12]	rd	opcode
0	0000001000	0	00000000	00000	1101111

Table 12: J-Type: jal x0, after_if

Hexadecimal: 0x0100006F

imm[20]	imm[10:1]	imm[11]	imm[19:12]	rd	opcode
1	1111100110	1	11111111	00000	1101111

Table 13: J-Type: jal x0, loop_while

Hexadecimal: 0xFCDDFF06F

Nota sobre pseudoinstrucciones: En el código fuente utilizamos instrucciones más cómodas como `li`, `mv`, `j`, `neg` y `bnez`, que el ensamblador expande internamente a instrucciones base de RISC-V:

- `li rd, imm` \rightarrow `addi rd, x0, imm` (I-Type)
- `mv rd, rs` \rightarrow `addi rd, rs, 0` (I-Type)
- `j etiqueta` \rightarrow `jal x0, etiqueta` (J-Type)
- `neg rd, rs` \rightarrow `sub rd, x0, rs` (R-Type)
- `bnez rs, etiqueta` \rightarrow `bne rs, x0, etiqueta` (B-Type)

En las tablas expuestas anteriormente siempre mostramos la forma base resultante de esta expansión.