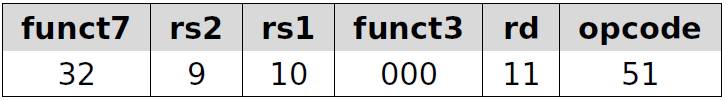
1)



SUB x11, x10, x9

2)

a)

**la**: sirve para cargar una dirección de memoria en un registro x[rd] = &offset

Se convierte en **auipc** y **addi**

b)

Da igual, por convensión a1 se utiliza como argumento para las funciones.

c)

Porque el direccionamiento es de a byte, por lo tanto se indexa de a 1.

d)

**sltu** rd, rs1, rs2

x[rd] = x[rs1] < x[rs2]

Compara x[rs1] con x[rs2] como números sin signo, y escribe 1 en x[rd] si x[rs1] es menor, ó 0 si no.

Sirve como valor bandera.

**bltu** rs1, rs2, offset

if (rs1 < rs2) => PC += sext(offset)

Si el registro x[rs1] es menor que x[rs2], tratando los valores como números sin signo, asignar al PC su valor actual más el offset sign-extended.

Sirve para comparar y saltar si cumple la condición.

e)

**jal** rd, offset [Instrucción tipo J]

x[rd] = PC + 4; PC += sext(offset)

Escribe la dirección de la siguiente instrucción (PC+4) en x[rd], luego asigna al PC su valor actual más el offset extendido en signo. Si rd es omitido, se asume x1.

**j** offset [Pseudoinstrucción]

PC += sext(offset)

Escribe al PC su valor actual más el offset extendido en signo. Se extiende a jal x0, offset.

**jr** rs1 [Pseudoinstrucción]

PC = x[rs1]

Escribe x[rs1] al PC.