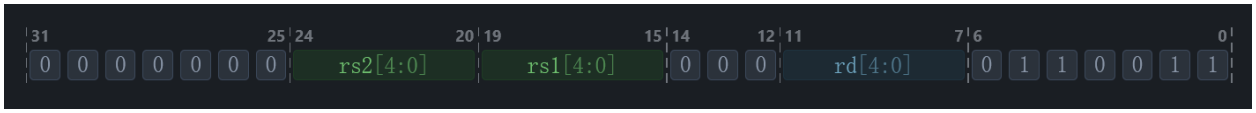
•**Mnemônico ->** ver instruções escolhidas

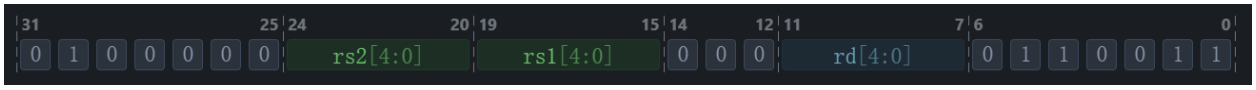
•**Tamanho da instrução** 32 bits

ADD: Quantidade e tipos de operandos: tem 2 operando, sendo do tipo endereço.
Tipo de operação: Aritmética
Tipo de endereçamento: Endereçamento por registrador



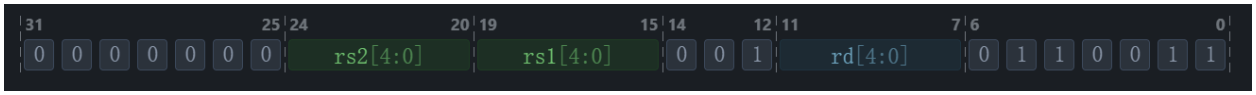
rd <- rs1 + rs2
ADD rs1, rs2 , rd

SUB: Quantidade e tipos de operandos: tem 2 operando, sendo do tipo endereço.
Tipo de operação: Aritmética
Tipo de endereçamento: Endereçamento por registrador



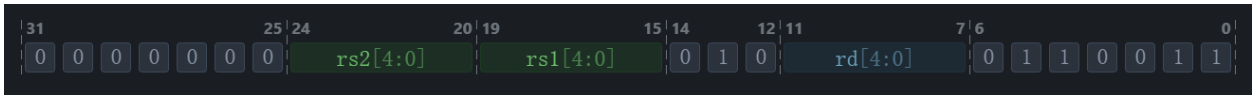
rd <- rs1 - rs2
SUB rs1, rs2 , rd

SLL: Quantidade e tipos de operandos: tem 2 operando, sendo do tipo endereço.
Tipo de operação: Lógico
Tipo de endereçamento: Manipulação de bits



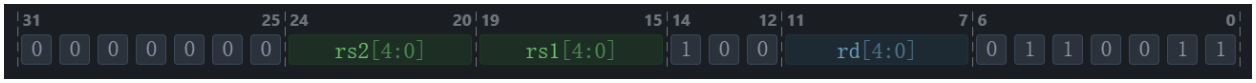
rd <- rs1 << (rs2 & 0b11111)
SLL rs1, rs2, rd

SLT: Quantidade e tipos de operandos: tem 2 operando, sendo do tipo endereço.
Tipo de operação: Lógico
Tipo de endereçamento: Endereçamento por registrador



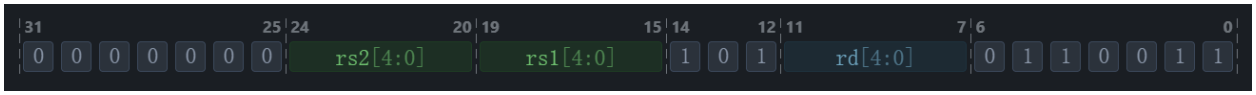
rd <- s'rs1 < s'rs2
SLT rs1, rs2, rd

XOR: Quantidade e tipos de operandos: tem 2 operando, sendo do tipo endereço.
Tipo de operação: Lógico
Tipo de endereçamento: Endereçamento por registrador



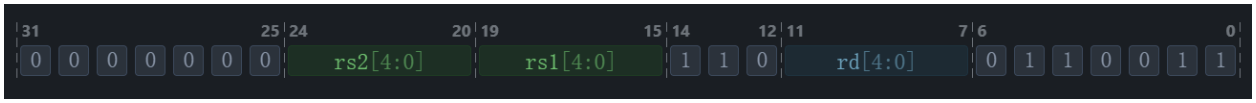
rd <- rs1 ^ rs2
XORrs1, rs2, rd

SRL: Quantidade e tipos de operandos: tem 2 operando, sendo do tipo endereço.
Tipo de operação: Lógico
Tipo de endereçamento: Endereçamento por registrador



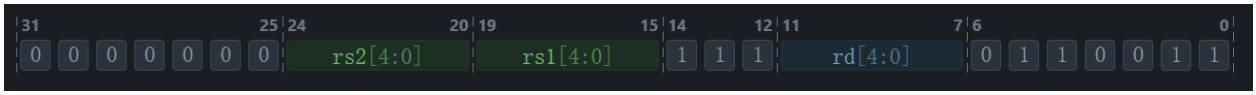
rd <- u'rs1 >> (rs2 & 0b11111)
SRL rs1, rs2, rd

OR: Quantidade e tipos de operandos: tem 2 operando, sendo do tipo endereço.
Tipo de operação: Lógico
Tipo de endereçamento: Endereçamento por registrador



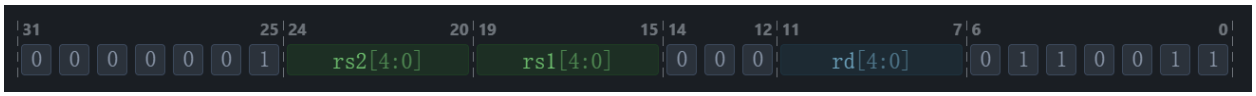
rd <- rs1 | rs2
OR rs1, rs2, rd

AND: Quantidade e tipos de operandos: tem 2 operando, sendo do tipo endereço.
Tipo de operação: Lógico
Tipo de endereçamento: Endereçamento por registrador



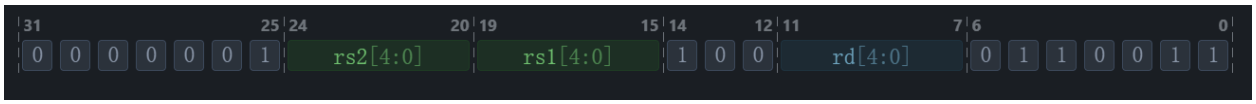
rd <- rs1 & rs2
AND rs1, rs2, rd

MUL: Quantidade e tipos de operandos: tem 2 operando, sendo do tipo endereço.
Tipo de operação: Aritmética
Tipo de endereçamento: Endereçamento por registrador



x[rd] <- x[rs1] × x[rs2]
MUL rs1, rs2, rd

DIV: Quantidade e tipos de operandos: tem 2 operando, sendo do tipo endereço.
Tipo de operação: Aritmética
Tipo de endereçamento: Endereçamento por registrador

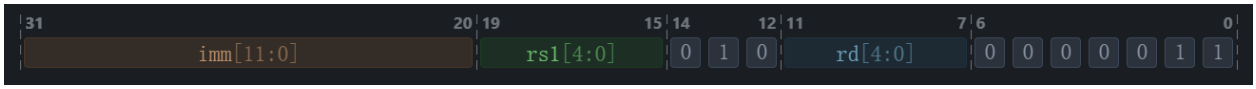


x[rd] <- x[rs1] /s x[rs2]
DIV rs1, rs2, rd

NOP: Quantidade e tipos de operandos: 0
Tipo de operação: Nenhuma
Tipo de endereçamento:

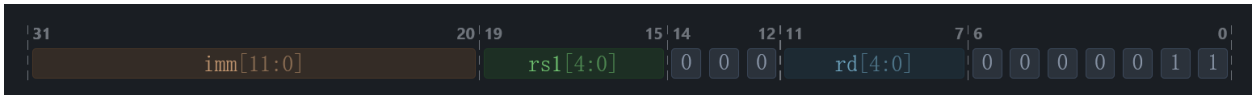
NOP = ADDI x0, x0, 0.

LW: Quantidade e tipos de operandos: tem 2 operando, sendo do tipo endereço e imediato.
Tipo de operação: Acesso à memória
Tipo de endereçamento: Endereçamento por registrador



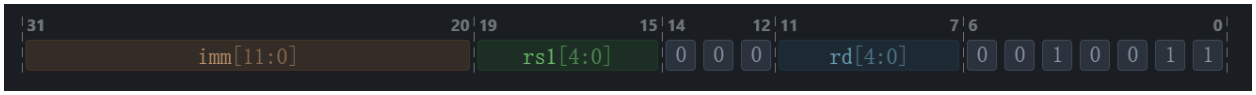
rd <- $(rs1 + imm) & 0xFFFF
LW rd , imm(rs1)

LB: Quantidade e tipos de operandos: tem 2 operando, sendo do tipo endereço e imediato..
Tipo de operação: Movimentação de dados
Tipo de endereçamento: Endereçamento por registrador



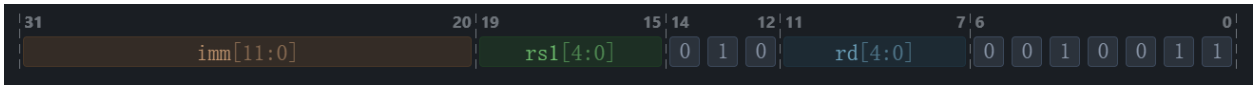
rd <- rs1 + imm
LB rd , imm(rs1)

ADDI: Quantidade e tipos de operandos: tem 2 operando, sendo do tipo endereço e imediato.
Tipo de operação: Aritmética
Tipo de endereçamento: Endereçamento por registrador



rd <- rs1 + imm
ADDI rd, rs1, imm

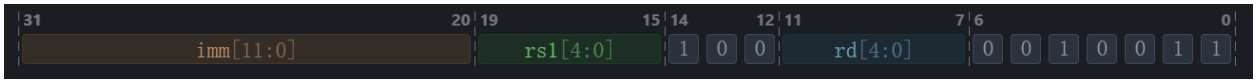
SLTI: Quantidade e tipos de operandos: tem 2 operando, sendo do tipo endereço e imediato.
Tipo de operação: Lógico
Tipo de endereçamento: Endereçamento por registrador



rd <- s'rs1 < imm
SLTI rd, rs1, imm

XORI: Quantidade e tipos de operandos: tem 2 operando, sendo do tipo endereço e imediato.

Tipo de operação: Lógico
Tipo de endereçamento: Endereçamento por registrador

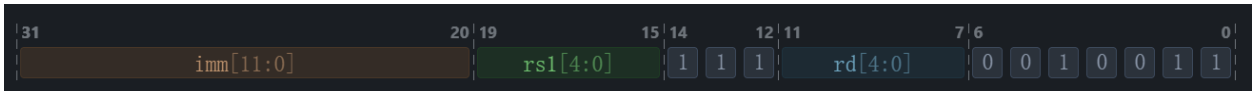


rd <- rs1 ^ imm
XORI rs1, rd , imm

ORI: Quantidade e tipos de operandos: tem 2 operando, sendo do tipo endereço e imediato.
Tipo de operação: Lógico
Tipo de endereçamento: Endereçamento por registrador

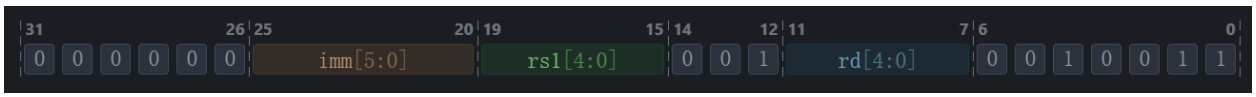
rd <- rs1 | imm
ORI rs1, rs2 , rd

ANDI: Quantidade e tipos de operandos: tem 2 operando, sendo 1 do tipo endereço e 1 imediato.
Tipo de operação: Lógico
Tipo de endereçamento: Endereçamento por registrador



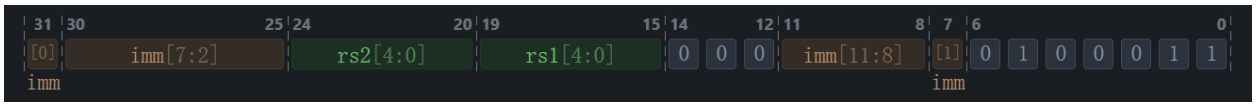
rd <- rs1 & imm
ANDI rd, rs1, imm

SLLI: Quantidade e tipos de operandos: tem 2 operando, sendo 1 do tipo endereço e outro do tipo imediato.
Tipo de operação: Manipulação de bits
Tipo de endereçamento: Endereçamento por registrador



rd <- rs1 << B64(imm)
SLLI rd, rs1

SB: Quantidade e tipos de operandos: tem 3 operando, sendo 2 do tipo endereço e 1 do tipo imediato.
Tipo de operação: Movimentação de dados
Tipo de endereçamento: Endereçamento a base mais índice



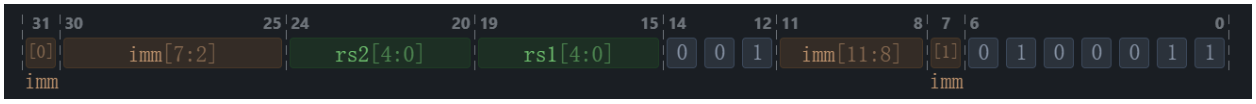
$mask <- 0xF

$(rs1 + imm) <- 8'rs2 | ($(rs1 + imm) & ~$MASK)
SB imm12hi, rs1, rs2, imm12lo

SH: Quantidade e tipos de operandos: tem 3 operando, sendo 2 do tipo endereço e 1 do tipo imediato.

Tipo de operação: Movimentação de dados

Tipo de endereçamento: Endereçamento a base mais índice

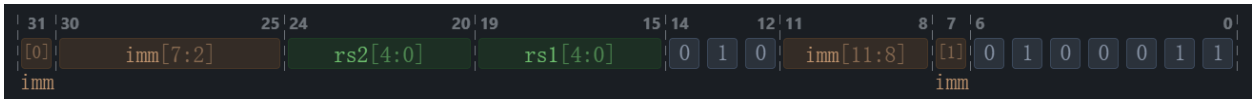


$mask <- 0xFF
$(rsl +imm) <- 16'rs2 | ($(rs1 + imm) & ~$mask)
SH rs2,offset(rs1)

SW: Quantidade e tipos de operandos: tem 3 operando, sendo 2 do tipo endereço e 1 do tipo imediato.

Tipo de operação: Movimentação de dados

Tipo de endereçamento: Endereçamento a base mais índice

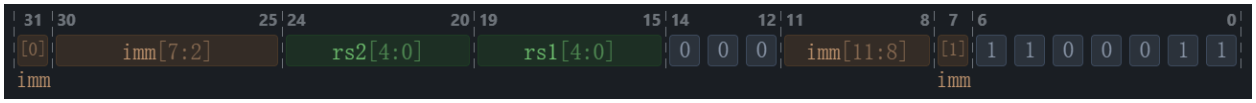


$mask <- 0xFFFF,
$(rs1 + imm) <- 32'rs2 | ($(rs1 + imm) & ~$MASK)
SW mm12hi, rs1, rs2, imm12lo

BEQ: Quantidade e tipos de operandos: tem 3 operando, sendo 2 do tipo endereço e 1 do tipo imediato.

Tipo de operação: Controle de fluxo

Tipo de endereçamento:Endereçamento a base mais índice ,sendo a base o PC

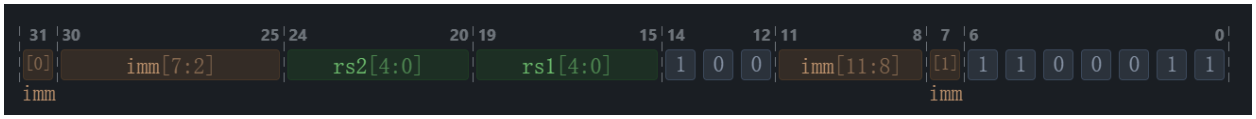


pc <- rs1 == rs2 ? pc + imm, pc + 4
BEQ rs1, rs2, bimm12

BLT: Quantidade e tipos de operandos: tem 3 operando, sendo 2 do tipo endereço e 1 do tipo imediato.

Tipo de operação: Controle de fluxo

Tipo de endereçamento: Endereçamento a base mais índice ,sendo a base o PC

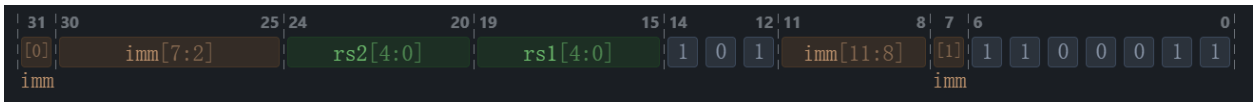


rs1 < rs2 ? pc + imm, pc + 4
BLT rs1, rs2, bimm12

BGE: Quantidade e tipos de operandos: tem 3 operando, sendo 2 do tipo endereço e 1 do tipo imediato.

Tipo de operação: Controle de fluxo

Tipo de endereçamento: Endereçamento a base mais índice ,sendo a base o PC



Pc <- rs1 >= rs2 ? pc + imm, pc + 4
BGE rs1, rs2, bimm12

LUI: Quantidade e tipos de operandos: tem 1 operando, sendo do tipo imediato =.

Tipo de operação: Movimentação de dados

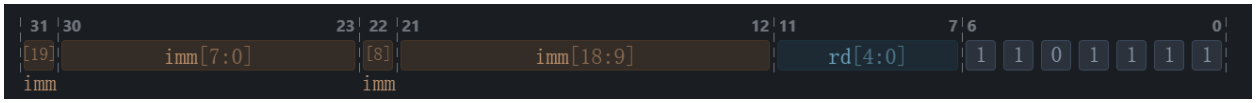
Tipo de endereçamento: Endereçamento por registrador

Rd <- imediato << 12
LUI rd, imm20

JAL: Quantidade e tipos de operandos: tem 2 operando, sendo do tipo imediato.

Tipo de operação: Controle de fluxo

Tipo de endereçamento: Endereçamento por registrador



Rd <- pc + 4
Pc <- pc + imediato
JAL rd, jimm20

ECALL: Quantidade e tipos de operandos: Instruções de sistema , terá um registrador exclusivo

Tipo de operação: —-

Tipo de endereçamento: —-

RaiseException(ENVIRONMENTCALL)
ECALL
