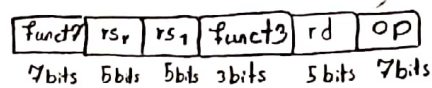


R type: register-type (3 register operand)

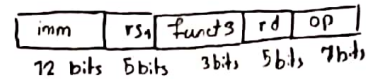


(1)

Sub شروع R-type هست!

$\Rightarrow \text{sub } t_3, t_2, t_1 \Rightarrow \text{funct } 7, 3, 2, \text{funct } 3:0, \text{op}:51$

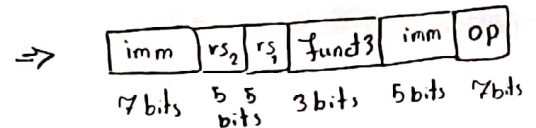
I type: Immediate type  $\Rightarrow$  3 operand  $\Rightarrow$  rs - rd - imm



$\rightarrow \begin{cases} \text{xori} \Rightarrow \text{funct } 3 \rightarrow 4, \text{op} \rightarrow 3 \\ \text{addi} \Rightarrow \text{funct } 3 \rightarrow 0, \text{op} \rightarrow 19 \\ \text{lw} \Rightarrow \text{funct } 3 \rightarrow 2, \text{op} \rightarrow 3 \end{cases}$

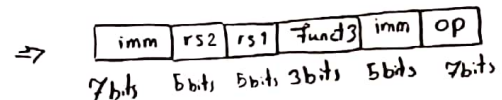
, jalr (jr ra)  $\Rightarrow \text{funct } 3:0, \text{op}:103$

S type  $\Rightarrow$  store type  $\Rightarrow (rs_1, rs_2, \text{imm}) \rightarrow 12 \text{ bits}$



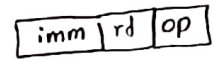
$\rightarrow \text{sw} \Rightarrow \text{funct } 3:2, \text{op}:35$

B type  $\Rightarrow$  Branches type  $\Rightarrow (rs_1, rs_2, \text{imm})$



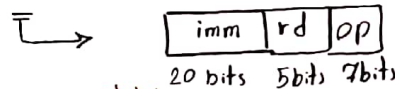
$\rightarrow \text{beq} \Rightarrow \text{funct } 3:0, \text{op}:99$

U type: upper immediate type  $\Rightarrow$  lui  $\Rightarrow$  load upper immediate  $\Rightarrow$



lui  $\Rightarrow \text{op}:55$

J type: used for jal: jump and link



+++

+++

+++

addi  $t_3, a_2, 12 \Rightarrow \dots\dots\dots 1100 \ 0110 \dots\dots 1110 \ 0010 \Rightarrow 0x00c60e13$   
 I type  $\in$  xori  $s_6, a_0, 14 \Rightarrow \dots\dots\dots 1110 \ 0101 \ 1001 \ 1011 \dots\dots 1001 \Rightarrow 0x00e54b13$   
 jal  $L_4 \rightarrow \text{opcode: } 1101111$   
 sw  $s_5, 4(a_0) \rightarrow \dots\dots 100 \ 0101 \ 1011 \ 0100 \ 1001 \Rightarrow 0x08aa223$   
 j  $L_1$

(ب)

beq  $t_3, a_f, L_3 \Rightarrow \text{funct } 3:000, \text{opcode: } 1100011, t_3 = x8 = 001\dots$

Lui  $t_4, 0xffff \Rightarrow \dots\dots 0101 \ 0111 \Rightarrow 0xffff002bf$

lw  $s_4, 8(t_1) \Rightarrow \text{imm: } 8 \quad \begin{matrix} rs_1: & rd: \\ t_1 & s_2 \\ x_6 & x_0 \end{matrix} \Rightarrow \dots\dots\dots 0110 \ 0100 \ 0000 \ 0011 \Rightarrow 0x00062403$

sub  $t_3, t_2, t_1 \Rightarrow \dots\dots\dots 0110 \ 0011 \dots\dots 1110 \ 0110 \Rightarrow 0x0038e33$

jr ra

addi  $\Rightarrow$  immediate addressing

sw  $\Rightarrow$  base addressing

xori  $\Rightarrow$  immediate addressing

jal  $\Rightarrow$  program counter (pc-relative addressing)

lw  $\Rightarrow$  immediate addressing

add, sub  $\Rightarrow$  R types  $\Rightarrow$  Register only Addressing

all b and j types  $\Rightarrow$  pc - relative addressing

- 0x00100293 → 1000 0000 0010 1001 0011  
 rs<sub>1</sub> funct<sub>3</sub> rd op  
 addi t<sub>5</sub>, zero, 1

⇒ I type: imm - rs1 - funct3 - rd - op

- 0x00a2c463 → op: 110011 ⇒ B type  
 funct3: 100, 1 ⇒ blt t<sub>0</sub>, a<sub>0</sub>, 0  
 imm rs<sub>2</sub> rs<sub>1</sub> funct<sub>3</sub> imm

- 0x00008067 → imm funct3 rd op  
 op: 1100111 ⇒ J type ⇒ jal x<sub>0</sub>, label  
 zero x<sub>0</sub>

- 0x7f010113 → I type  
 ⇒ addi sp, sp, -16  
 imm rs<sub>1</sub> funct3 rd op ⇒ I type  
 x<sub>r</sub> x<sub>r</sub>

- 0x00a12623 → stype ⇒ sw x<sub>10</sub>, 0(x<sub>2</sub>)  
 a<sub>0</sub> sp  
 imm rs<sub>2</sub> rs<sub>1</sub> funct3 imm op ⇒ stype

- 0x00112423 → stype ⇒ sw x<sub>1</sub>, 8(x<sub>2</sub>)  
 ra sp  
 imm rs<sub>2</sub> rs<sub>1</sub> funct3 imm op

- 0x00812223 → S type ⇒ sw x<sub>8</sub>, 4(x<sub>2</sub>)  
 s<sub>0</sub> sp  
 rs<sub>2</sub> rs<sub>1</sub> funct3 imm op

- 0x00912023 → stype ⇒ sw x<sub>9</sub>, 0(x<sub>2</sub>)  
 s<sub>1</sub> sp  
 rs<sub>2</sub> rs<sub>1</sub> imm op

- 0x fff 50 513  $\Rightarrow$    
I type  $\Rightarrow$  addi  $a_2, a_2, -1$

- 0x fdd fff 0ef  $\Rightarrow$    
J type  $\Rightarrow$  jal ra, 0x fddff (NaN)  $\Rightarrow$  J type : jal

- 0x 00 a 00 433  $\Rightarrow$    
rd: 01000 : 8    rs1: 0, rs2: 01010 = 10  
R type  $\Rightarrow$  add s0, zero, a0  
funct3: 000

- 0x 00 c 12 503  $\Rightarrow$    
I type  $\Rightarrow$  lw a0, sp, 0  $\Rightarrow$  I type

- 0x fff e 50 513  $\Rightarrow$    
I type  $\Rightarrow$  addi a0, a0, -2  $\Rightarrow$  I type  $\Rightarrow$  addi

- 0x fcd fff 0ef  $\Rightarrow$    
J type  $\Rightarrow$  jal ra, 0x fcdff (table)  $\Rightarrow$  J type

- 0x 00 a 00 4 b 3  $\Rightarrow$    
opcode: 0110011, funct: 000  $\Rightarrow$  R type  $\Rightarrow$  add s1, zero, a0

- 0x 00 84 84 33  $\Rightarrow$  opcode: 0110011, funct3: 000  $\Rightarrow$  R type  $\Rightarrow$  add s0, s1, s0

- 0x 000 40 533  $\Rightarrow$  R type  $\Rightarrow$  add a0, s0, zero

- 0x00812083  $\Rightarrow$  I type  $\Rightarrow$  Lw  $\Rightarrow$  lw ra, sp, 8

imm<sub>16</sub> funct<sub>3</sub> op<sub>6</sub>

- 0x00412403  $\Rightarrow$   $\underbrace{\text{op}}_{\text{opcode}} \text{funct3: 010} \Rightarrow$  lw s<sub>0</sub>, sp, 4

- 0x00412483  $\Rightarrow$  I type  $\Rightarrow$  lw s<sub>1</sub>, sp, 4 ;

- 0x01010113  $\Rightarrow$   $\underbrace{\text{imm}}_{\text{imm16}} \underbrace{\text{rs1}}_{\text{rs1}} \underbrace{\text{funct3}}_{\text{funct3}} \underbrace{\text{rd}}_{\text{rd}} \underbrace{\text{op}}_{\text{op}} \Rightarrow$  I type: addi sp, sp, 10

- 0x00008067  $\Rightarrow$   $\underbrace{\text{op}}_{\text{opcode}}$

\*\*\*

\*\*\*

\*\*\*

addi t<sub>0</sub>, zero, 1  $\Rightarrow$  temporary = 1

blt t<sub>0</sub>, a<sub>0</sub>, t<sub>0</sub>  $\Rightarrow$  if n > 1: go to

Jal x<sub>0</sub>, lable  $\Rightarrow$  صبر کردن

addi sp, sp, -16  $\Rightarrow$  Save register

sw a<sub>0</sub>, 0(sp)  $\Rightarrow$  store word

sw ra, 8(sp)

sw s<sub>0</sub>, 4(sp)

sw s<sub>1</sub>, 0(sp)

addi a<sub>2</sub>, a<sub>2</sub>, -1  $\Rightarrow$  x = x - 1

Jal ra, 0xffff  $\Rightarrow$

Lable: add s<sub>0</sub>, zero, a<sub>0</sub>

lw a<sub>0</sub>, sp, 0

addi a<sub>0</sub>, a<sub>0</sub>, -2  $\Rightarrow$  n = n - 1  $\Rightarrow$  recursive functional

jal ra, 0xffff

add s<sub>1</sub>, zero, a<sub>0</sub>

add s<sub>0</sub>, s<sub>1</sub>, s<sub>0</sub>

add a<sub>0</sub>, s<sub>0</sub>, zero

lw ra, sp, 8

lw s<sub>0</sub>, sp, 4

lw s<sub>1</sub>, sp, 4

addi sp, sp, 10  $\Rightarrow$  restore sp

تابع فیوق، تابعی برای واسطه

ra همیشه رتورن

فیوناچی است.