ADDI, SLTI, ANDI, ORI, XORI, SLTIU, SLLI, SRAI, SRLI, JALR, OR

LUI (U), AUIPC (U), JAL (J), BEQ, BNE, BGEU, ECALL

BLT, BGE, LW, SW, FENCE, ~~EBREAK~~, AND, ADD

SLT, SLTU, XOR, SLL,

SRL, SUB, SRA, BLTU,

LH, LHU, LB, LBU, SH, SB, LD, SLLI

SRLI, SLTI

! Risc-V сохраняет положение регистров источников (rs1 и rs2) и назначения (rd) в одном и том же положении.
Бит знака в X31 положении

...1100011111111

1) a|b
2) (a|b) & X ← ~((01 ...111) << 7)
opcode + funck

1) xori rd, rs1, imm

imm[11:0]	rs1	100	rd	0010011
12	5	3	5	7

I

Вычисляет побитовое „или" и кладёт результат в rd.

101
& 100
100

111000001111111
...... 0110011

& 01100
01000
01000

3) srli rd, rs1, imm

000000	imm[4:0]	rs1	101	rd	0010011
		5	3	5	7

R

x[rd] = x[rs1] >>u imm (верхние биты нулями)

4) srai rd, rs1, imm

0100000, imm[4:0]	rs1	101	rd	0010011	R
12	5	3			

$x[rd] = x[rs1] >>_s imm$ (верхние биты старшими битами)

5) addi rd, rs1, imm

imm[11:0]	rs1	000	rd	0010011	I
12	5	3			

$x[rd] = x[rs1] + sext(imm)$
↑ расширенный по знаку

12) add rd, rs1, rs2

0000000	rs2	rs1	000	rd	0110011	R
7	5	5	3			

7) andi rd, rs1, imm

imm[11:0]	rs1	111	rd	0010011	I
12	5	3	5	7	

8) and 5

0000000	rs2	rs1	111	rd	0110011	R
12		5	3	5	7	

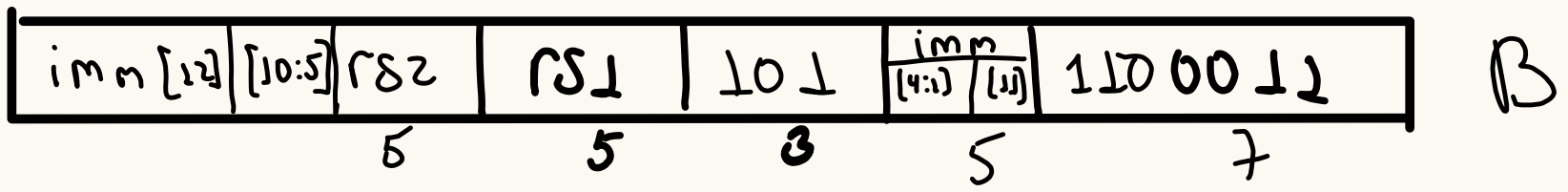
9) auipc rd, imm[31,12]

imm[31;12]	rd	0010111	U
20	5	7	

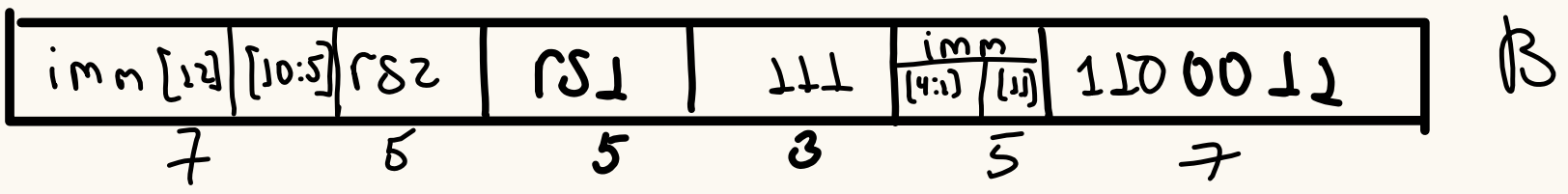
10) BEQ rs1, rs2, imm

imm[12] (31)	[10:5] (30)	rs2	rs1	000	imm [4:1]	[11]	1100011	B
		5	5	3	5 (8)	(7)	7	

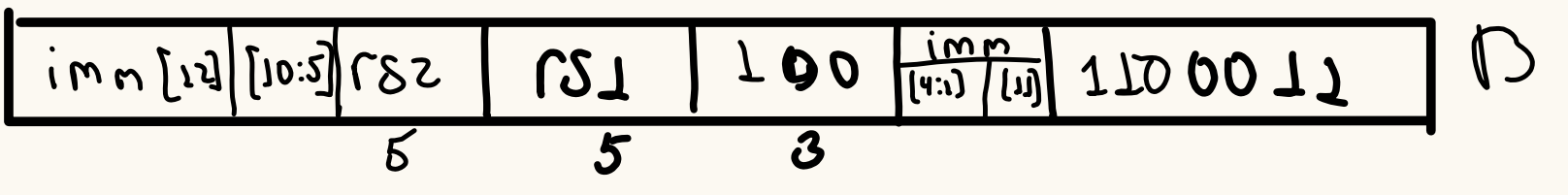
11) BGE rs1, rs2, imm



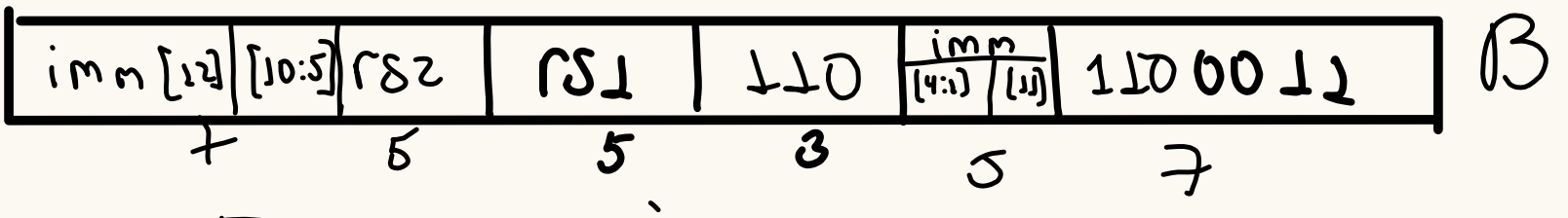
12) BGEU rs1, rs2, imm



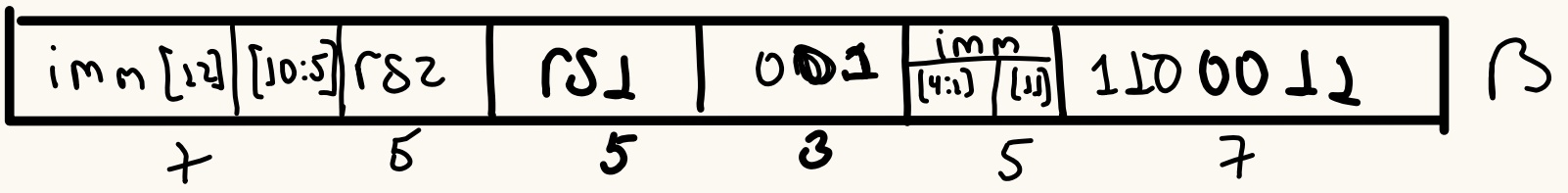
13) BLT rs1, rs2, imm



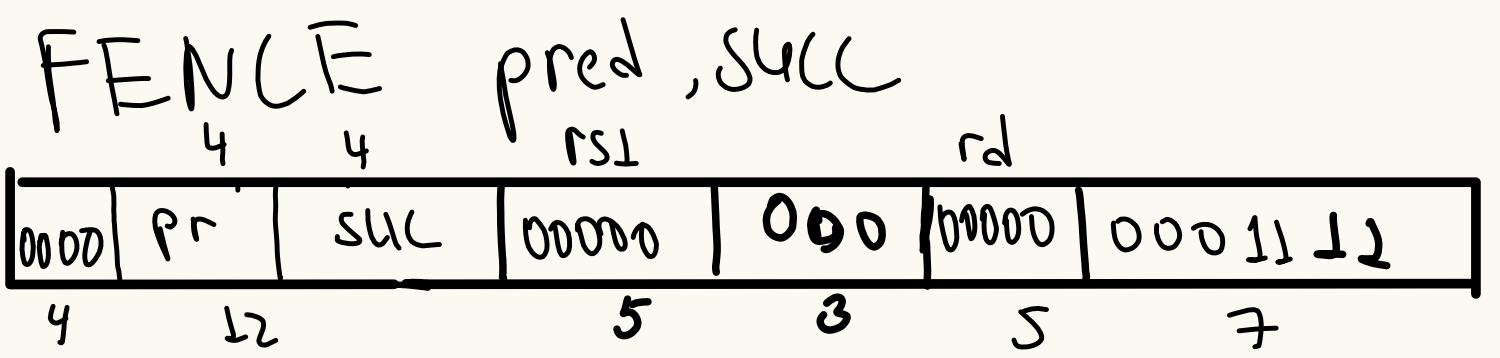
14) BLTU rs1, rs2, imm



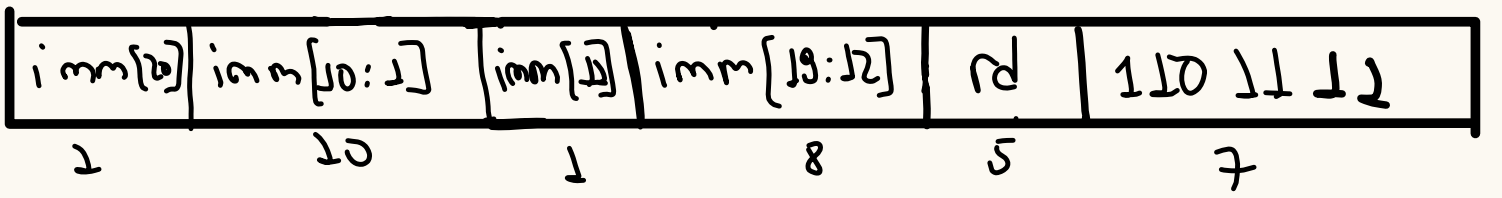
15) BNE rs1, rs2, imm



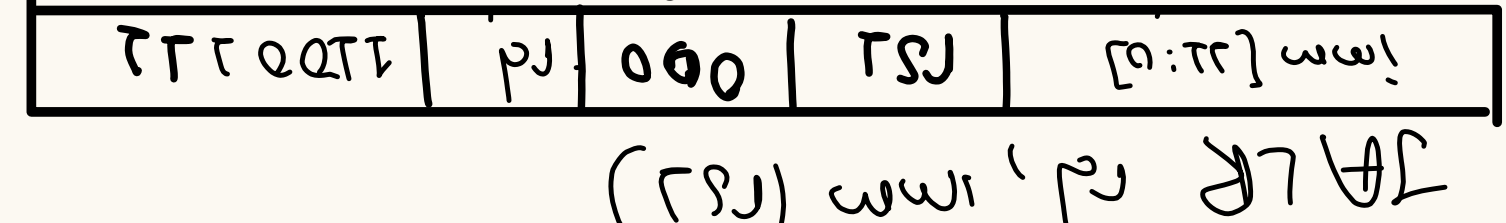
16) FENCE pred, succ



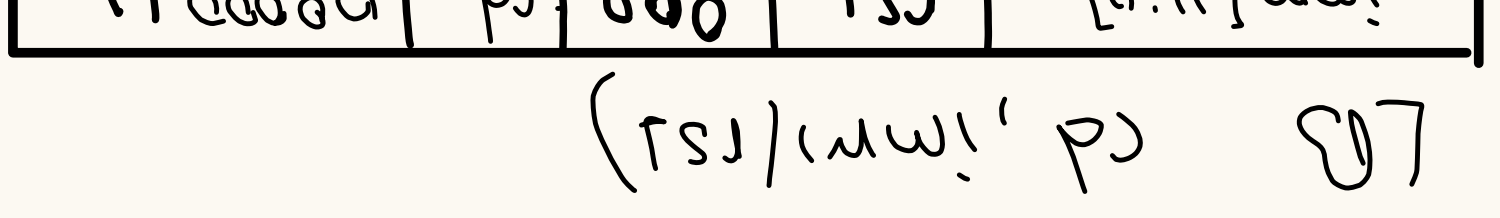
17) JAL rd, imm



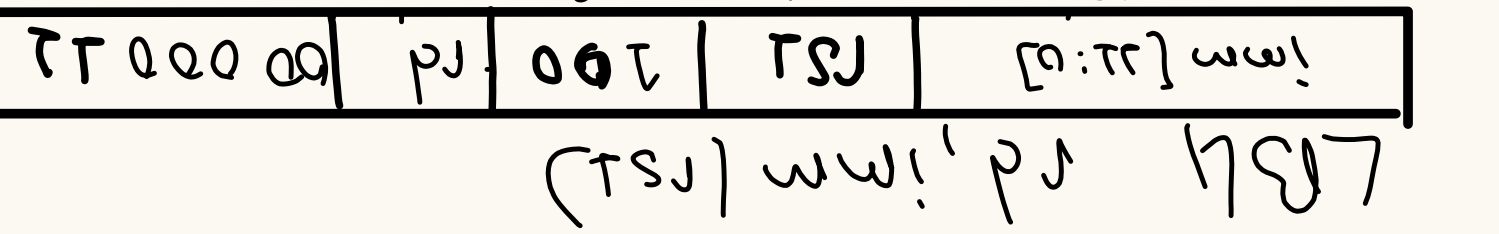
22) LH rd, imm(rs1)



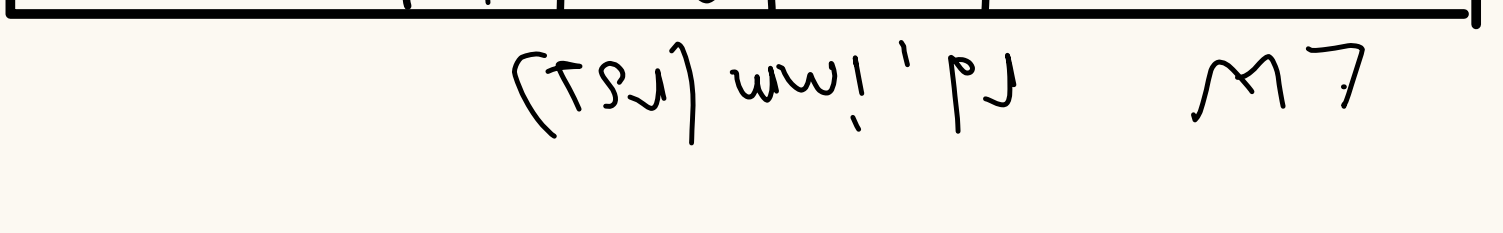
23) LHU rd, imm(rs1)



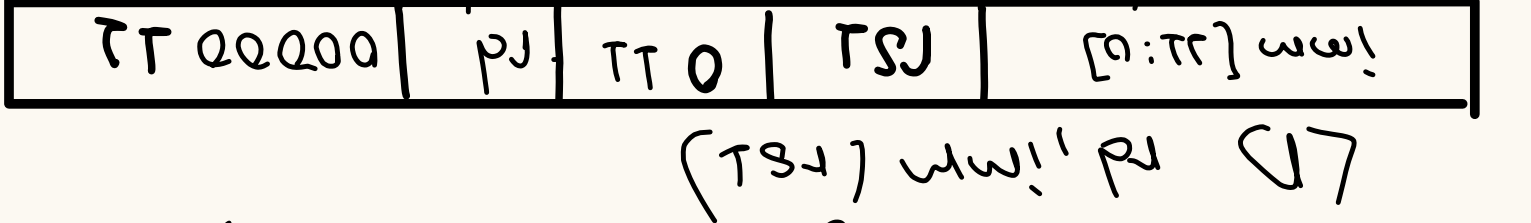
24) LUI rd, imm[31:12]



25) LW rd, imm(rs1)



18) JALR rd, imm(rs1)



19) LB rd, imm(rs1)

20) LBU rd, imm(rs1)

21) LB rd, imm(rs1)

29) `sh rs2, imm(rs1)`

imm[11:5]	rs2	rs1	001	imm[4:0]	0100011
7	5	5	3	5	7

S

30) `sll rd, rs1, rs2`

0000000	rs2	rs1	001	rd	0110011
7	5	5	3	5	7

R

31) `sltiu rd, rs1, imm`

000000000000	rs1	011	rd	0010011
12	5	3	5	7

I

32) `sb rs2, imm(rs1)`

imm[11:5]	rs2	rs1	000	imm[4:0]	0100011
7	5	5	3	5	7

S

33) `ori rd, rs1, imm`

imm[11:0]	rs1	110	rd	0010011
12	5	3	5	7

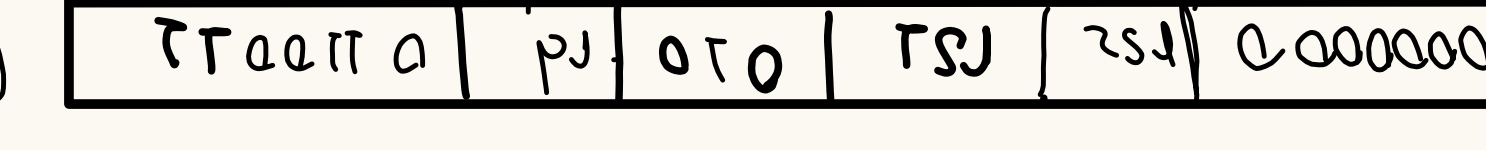
I

34) `or rd, rs1, rs2`

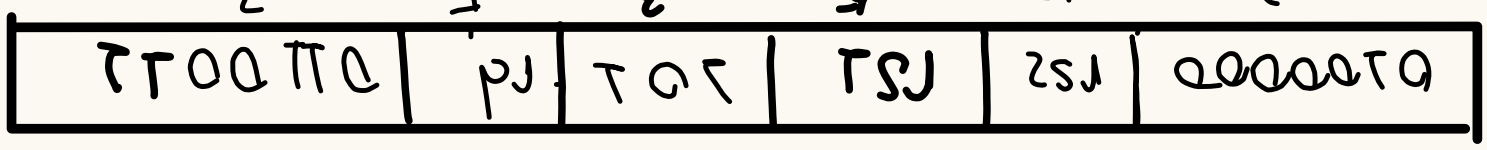
0000000	rs2	rs1	110	rd	0110011
7	5	5	3	5	7

R

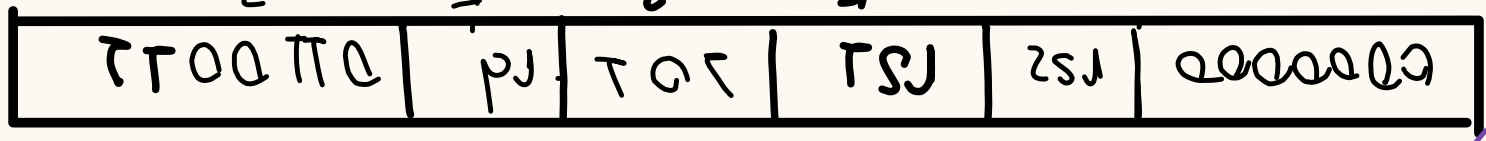
32) SLL rd, rs1, rs2



33) SRA rd, rs1, rs2

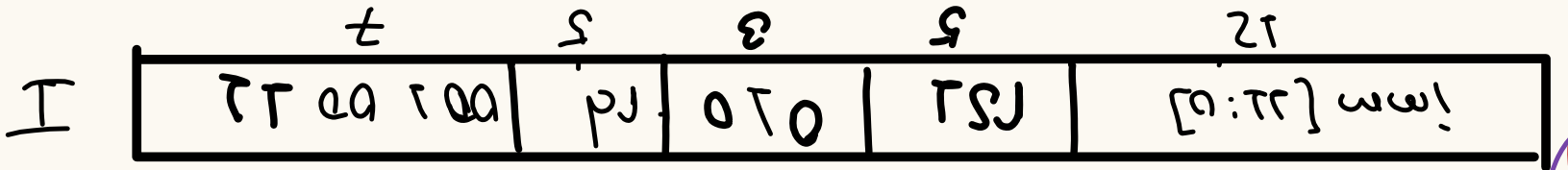


34) SRL rd, rs1, rs2

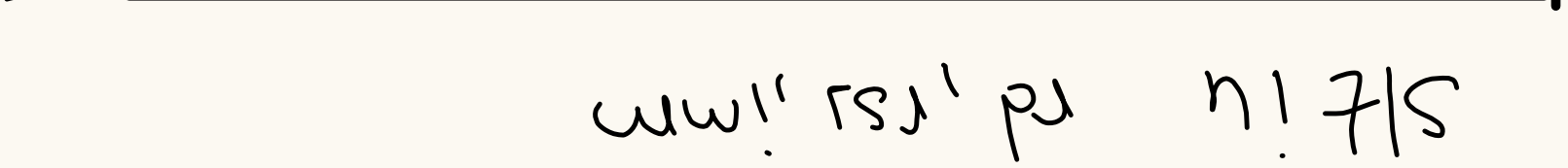


35) SLLi rd, rs1, imm

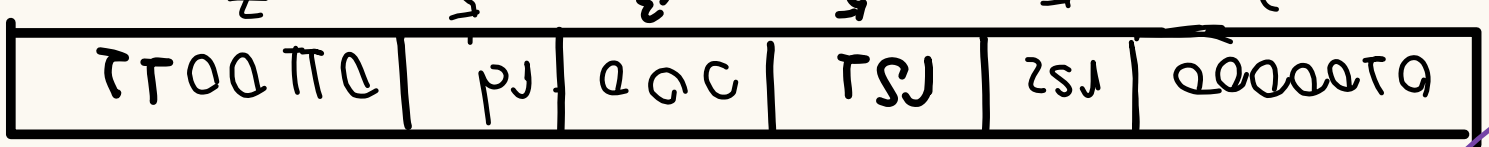
36) SRAi rd, rs1, imm



37) SRLi rd, rs1, imm

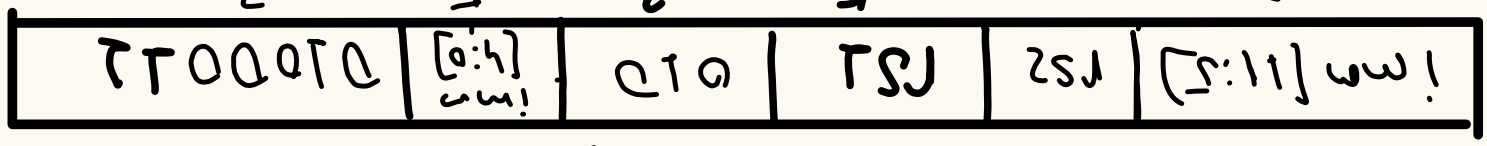


38) SUB rd, rs1, rs2



R

39) SW rs2, imm(rs1)



S

40) XOR rd, rs1, rs2

R