Branching Instructions

SB type instruction — beq, bne, blt, bge

31 … 0

1	6	5	5	3	4	1	7

represents branch addresses in multiples of 2.

each field has unique name and size.

imm [12]	imm [10:5]	rs2	rs1	funct3	imm [4:1]	imm [11]	opcode

forward & backward moving.

the unusual encoding of imm. simplifies datapath design.

```
        Loop:
#80000    slli  x10, x22, 3       — line 1
#80004    add   x10, x10, x25     — line 2
#80008    ld    x9, 0(x10)        — line 3
#80012    bne   x9, x24, Exit     — line 4
                rs1  rs2
#80016    addi  x22, x22, 1       — line 5
#80020    beq   x0, x0, loop      — line 6
        Exit:
#80024                            — line 7
```

$3 \times 4 = 12$

0000 0000 1100

$\underset{12}{0}\ \underset{11}{0}\underbrace{000\ 0000}_{10:5}\ \underbrace{110}_{4:1}\cancel{0}$ Discard it.

0	000 000	11000	01001	XXX	0110	0	XXXXXXX
imm [12]	imm [10:5]	rs2	rs1	funct3	imm [4:1]	imm [11]	op code

1	111 111	00000	00000	XXX	0110	1	XXXXXXX
imm [12]	imm [10:5]	rs2	rs1	funct3	imm [4:1]	imm [11]	op code

$5 \times 4 = 20$ but its going upward so -20.

= 0000 0001 0100

= 0 0000 0001 0100

= 1 1111 1110 1011

+ 1

$\underset{12}{1}\ \underset{11}{1}\underbrace{111\ 111}_{10:5}\ \underbrace{0\ 110}_{4:1}\underset{0}{0}$ ⇒ -20 in 2s com.

PC relative addressing

= PC + immediate × 2

1	111 111	11000	01001	XXX	0110	1	XXXXXXX

0

(i) Form the 12 bit number

11 111 111 0110

(ii) Detect if positive or negative number.

if neg perform 2s comp. again,

1111 1111 0110

0000 0000 1001

+1

0000 0000 1010 ⇒ 10

(iii) PC $\pm$ imm ×2 → offset

Based on the sign bit

UJ type instruction – Jal

31					0
1	10	1	8	5	7

each field has unique name and size.

imm [20]	imm [10:1]	imm [11]	imm [19:12]	rd	opcode

uses 20 bit immediates for larger jumps.

For more large jump, (32 bit)
lui: load address [31:12] to a temp reg.
jalr: add address [11:0] and jump to target.

Jal X0, 2000

0	1111 1010 00	0	0000 0000	00000	opcode

2000 = 0000 0000 0111 1101 0000

$= \underbrace{0}_{20}\ \underbrace{0000\ 0000}_{19:12}\ \underbrace{0}_{11}\underbrace{111\ 1101\ 000}_{10:1}0$

Given a branch on register `x10` being equal to zero,

```
beq x10, x0, L1
```

replace it by a pair of instructions that offers a much greater branching distance.

Answer

These instructions replace the short-address conditional branch:

```
   bne x10, x0, L2
   jal x0, L1
L2:
```