

Question 1 (12 points):

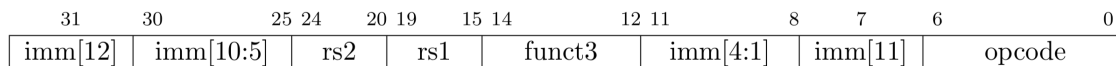


Figure 1: SB-Type format. Used for branch instructions in RISC-V.

Figure 1 shows the SB-Type format that is used for branch instructions in RISC-V.

- a. (3 points) Assume that the value of the Program Counter is 0x8000FFFC when a branch instruction is fetched from the memory. The binary representation of this branch instruction is 0x04B00263.

- What is the address of the next instruction that will be executed if the value in register a1 is equal -1?
- What is the address of the next instruction that will be executed if the value in register a1 is equal 0?

0x04B00263

0000 0100 1011 0000 0000 0010 0110 0011

0 000010 01011 00000 000 0010 0 110011

Registers being compared are x0 and a1, the branch is beq because funct3 is 000

Thus, if a1=-1 then the next instruction to be executed is at the address 0x80010000.

Immediate in instruction: = 0 0 000010 0010

Actual Immediate = 0 0 000010 00100 = 0x00000044

Target address = 0x8000FFFC + 0x00000044 = 0x80010040

Thus, if a1=0 then the next instruction to be executed is at the address 0x80010040.

- b. (3 points) Assume that a branch instruction is at address 0x80000000. Which is the lowest memory address that can be the target for this branch instruction?

The most negative immediate value in the instruction is 1000 0000 0000.

But there is the bit zero that is not represented in the instruction. Therefore the value of the immediate is 1 0000 0000 0000.

After sign extension, the value becomes: 1111 1111 1111 1111 1111 0000 0000 0000 = 0xFFFFF000 The lowest memory address that can be target of this branch is:

0x80000000

+ 0xFFFFF00

0x7FFFF000

- c. (3 points) Assume that a branch instruction is at address 0xCBA9EDC0. Which is the highest memory address that can be the target for this branch instruction?

The most positive immediate value in the instruction is 0111 1111 1111.

But there is the bit zero that is not represented in the instruction. Therefore the value of the immediate is 0 1111 1111 1110.

After sign extension, the value becomes: 0000 0000 0000 0000 0000 1111 1111 1110 = 0x00000FFE. The lowest memory address that can be target of this branch is:

```
0xCBA9EDC0
+ 0x00000FFE
-----
0xCBA9FDBE
```

- d. (**3 points**) RISC-V was designed to allow the implementation of 16-bit architectures and thus it needed to allow for every memory address that is a multiple of two to be a target of a branch instruction. The consequence of this design decision in the ISA is that the 12 bits available to represent the immediate value in the branch instruction contain the bits `imm[12]-imm[1]`. Consider an alternative ISA design called RISC-V-64 that only allow addresses that are multiple of 8 to be target of a branch instruction. In RISC-V-64 the 12 bits for the immediate value store `imm[14]-imm[3]` — bits 0, 1, and 2 are always zero and do not need to be represented. We define the range of a branch instruction as the maximum difference between the address of the branch instruction and the address of the target instruction. How does the range of a branch in RISC-V-64 compares with the range of a branch in RISC-V?

RISC-V-64 has two more bits for the immediate value, therefore the range is $2^2 = 4$ times larger than the range of RISC-V.