Question 1 (12 points):

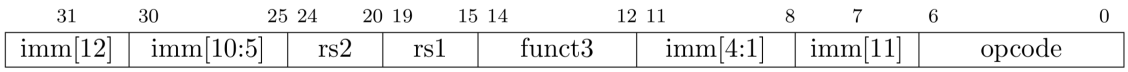

Figure 1: SB-Type format. Used for branch instructions in RISC-V.

Figure **??** 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?

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?

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?

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?