

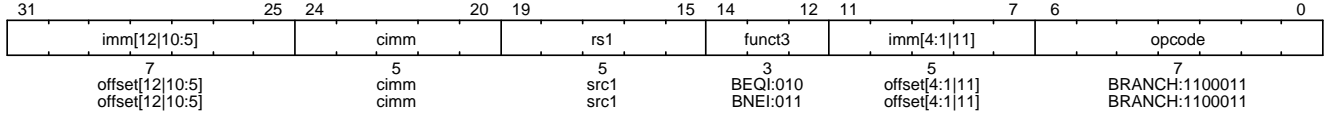
# "Zibi" Extension for Branch with Immediate, Version 0.6

This chapter defines the "Zibi" extension, which appends conditional branch instructions with an immediate operand. In real world applications, conditional branches comparing with constants are of great universality. With only conditional branch in current RISC-V ISA, we need to load immediate values into registers before branching, which will increase register pressure and potentially cause register spills/reloads.

## NOTE

Existing conditional branch instructions (e.g., beq, bne) in RISC-V mandate two-register comparisons. Loading immediate values introduces unnecessary load immediate operation (e.g., li a4,5; bne a5,a4,offset), and spills/reloads may occur under register scarcity. Based on the assessment with LLVM compiler, Zibi can reduce register spill & reload, and dynamic instruction count when compiling with branch with immediate, exposing inefficiencies in current designs.

The Zibi extension defines 2 instructions named BEQI and BNEI. BEQI and BNEI compare a register with an immediate, and it takes the branch if registers *rs1* and *cimm* are equal or unequal respectively. BEQI and BNEI use the B-type instruction format as conditional branches. The 12-bit B-immediate encodes signed offsets in multiples of 2 bytes. The offset is sign-extended and added to the address of the branch instruction to give the target address. The branch range is  $\pm 4$  KiB. The field *cimm* is the constant immediate used for comparing with the source register.



$$constant = \begin{cases} \text{zext}(cimm) & \text{if } cimm \neq 0 \\ -1 & \text{if } cimm = 0 \end{cases}$$

## NOTE

Profiling general benchmarks reveals that positive immediate values occur much more frequently than negative ones. Zibi extension encodes all 5-bit immediate as positive number, with the exception of immediate zero, which is represented as negative one (Comparison with immediate zero can be achieved using beq/bne rs1,x0,offset).