

Gabriel Lucas França do Nascimento

190107111

OAC – T03

1. Qual a razão do embaralhamento dos bits do imediato no RISC-V?

Como as instruções precisam armazenar informações de múltiplos campos (como opcode, funct3, funct7, registradores e imediato), o imediato é construído de forma embaralhada, mas sempre respeitando o alinhamento lógico para que os valores possam ser reconstituídos corretamente pelo hardware.

O embaralhamento dos bits do imediato nos diferentes formatos do RISC-V ocorre para maximizar a utilização do espaço disponível na codificação de uma instrução de 32 bits.

Por exemplo, no formato SB-type, usado para instruções de desvio condicional, os bits do imediato são distribuídos em diferentes partes da instrução para acomodar os registradores e outros campos. O embaralhamento permite incluir mais informações úteis no mesmo espaço, mantendo a eficiência do conjunto de instruções.

2. Por que alguns imediatos não incluem o bit 0?

Em alguns formatos de instrução, como o SB-type (branch) e o UJ-type (jump), o bit menos significativo do imediato (bit 0) é sempre zero pois esses valores representam deslocamentos de endereço múltiplos de 2 ou 4 bytes alinhados, dependendo do tamanho das instruções. É uma otimização comum em arquiteturas baseadas em instruções de tamanho fixo, como é o caso do RISC-V, onde todos os endereços são naturalmente alinhados.

Ao eliminar o bit 0, economiza-se espaço na codificação do imediato sem perder funcionalidade, já que o alinhamento é garantido pelo hardware.

3. Os imediatos de operações lógicas estendem o sinal?

Não, os imediatos usados em operações lógicas (como ANDI, ORI e XORI) não estendem o sinal. Nesses casos, o imediato é tratado como um valor binário sem sinal (zero-extend), independentemente de seu valor. Essa decisão é coerente com a semântica das operações lógicas, que não fazem distinção entre números negativos e positivos.

Por outro lado, operações aritméticas, como ADDI, utilizam extensão de sinal porque trabalham com valores inteiros com sinal.

4. Como é implementada a instrução NOT no RISC-V?

A operação pode ser implementada utilizando a instrução XORI com o imediato -1. A operação NOT é equivalente a “XORI x, x, -1”, já que o operador XOR com -1 inverte todos os bits do operando.