

lui in risc-v with small immediate

Asked 2 years ago Modified 2 years ago Viewed 847 times



1

For a risc-v instruction like `lui x28 123`, venus shows that its machine code is `0x0007BE37`. However `lui` should load only upper 20 bits of immediate 123. Since in hex 123 is `0x0000007B`, I think the corresponding machine code should be `0x00000E37`.



I am puzzled why venus produces different results, is there a problem with my understanding of `lui`?



[assembly](#) [32-bit](#) [riscv](#)

[Share](#) [Improve this question](#) [Follow](#)

asked Mar 20, 2021 at 22:21



[MissSirius](#)

140 7

Sorted by:

1 Answer

Highest score (default)



8



When you provide a numeric value as a literal to an instruction like `LUI`, then some assemblers will take that as the *exact* immediate value you want encoded in the instruction. This is important and useful for compilers generating assembly or assembly programmers who want to use `LUI` and specify the exact bit pattern for the 20 bits in that instruction, without any interference from the assembler.

If you want it to take only the high 20 bits of your literal, there's compile time functions¹ like `%hi` and `%lo` that some assemblers will accept, e.g. `lui %hi(123)` — the `%hi` is for use in `lui` and the `%lo` for use in an instruction that wants the 12 low bits, like `addi` and `sw / lw`.

¹ These may be evaluated as late as link time.

These functions, (`%hi` & `%lo`) are complicated due to the traditional choice of pairing with instructions like `lw` and `addi`, wherein their immediate makes up the 12-bit rest, *but itself is signed*. So, the `%hi` function offsets the value by 1 in the specific cases where the `lw`'s offset will appear negative and will sign extend with 1's at runtime. So, when these are paired² the environment makes things work, but if you were to use `%hi()` in `LUI` on a random constant without such pairing, you might not get the exact 20 bits you're expecting.

² More than one `%lo` can ~pair with the same `%hi` as long as the function arguments match. So, for example, a read/modify/write sequence to a global variable might have one `lui %hi()` for both `lw %lo()` and `sw %lo()`.

Share Improve this answer Follow

edited Mar 20, 2021 at 22:46

answered Mar 20, 2021 at 22:28



Erik Eidt

21.7k

2

28

53
