

HW4.4. RISC-V Calling Convention

Q1.1: Which of the following statements are true about calling convention (abbreviated as CC)?

☐

(a) CC rules apply when jumping within the same function (ex. through loops or branches).

☐

(b) CC rules apply when jumping to a different function.

☐

(c) The purpose of CC is to build abstractions within function interactions and regulate assembly code structure.

☐

(d) In order to follow CC, we need to store all registers on the stack at the start of a function, even if we don't use a particular register during our function.

☐

(e) The RISC-V language mandates that CC be followed, and as such will refuse to assemble or run code that doesn't follow CC.

☐

(f) Apart from x0 and the starting values of certain registers such as **sp**, all 32 registers are fundamentally identical in behavior.

☐

(g) The register **sp** is used to store a pointer to the top of the stack

☐

(h) CC dictates that anything above the stack pointer at the start of a function call remains unchanged, while anything below the stack pointer is unallocated.

In order to allocate space on the stack, the stack pointer is moved down. This creates

☐

(i) a space which is guaranteed unchangeable by other functions, but modifiable by the current function, and can thus be used for temporary storage.

☐

(j) The caller of a function can assume that any "a" register is unchanged after a function call.

☐

(k) The caller of a function can assume that the "ra" register is unchanged after a function call.

☐

(l) The caller of a function can assume that any "s" register is unchanged after a function call.

☐

(m) The caller of a function can assume that the "sp" register is unchanged after a function call.

☐

(n) The caller of a function can assume that any "t" register is unchanged after a function call.

☐

(o) The callee does not need to restore the original value of any "a" register.

☐

(p) The callee does not need to restore the original value of the "ra" register.

☐

(q) The callee does not need to restore the original value of any "s" register.

☐

(r) The callee does not need to restore the original value of the "sp" register.

☐

(s) The callee does not need to restore the original value of any "t" register.

It doesn't matter how you ensure that register values get restored (ex. storing on the

☐

(t) stack, not using the register, adding and subtracting the same amount), as long as the registers that need to be restored are guaranteed to return to its original value.

☐

(u) Efficient code tends to maximize the amount of data stored on the stack, because accessing the stack is faster than accessing registers.

☐

(v) You should never use x0 as a destination register.

Select all possible options that apply. ?

Save & Grade 20 attempts left

Save only

Additional attempts available with new variants ?

Homework 4

Assessment overview

Total points: 32/100

Score:

32%

Question

Value: 10

History:

Awarded points: 0/10

Report an error in this question

Previous question

Next question

Attached files

No attached files

Attach a file

Attach text