HW4.5. RISC-V U-types

Refer to the Reference Card to help you out.

What do the following instructions do?

`lui`

- [✓] (a) Lower 12 bits always set to 0 ✓
- [ ] (b) Lower 12 bits set to instruction-type immediate
- [ ] (c) Stands for "Load Unsigned Immediate"
- [✓] (d) Can be the first sub-instruction used in `li` ✓
- [✓] (e) Upper 20 bits set to instruction-type immediate ✓
- [✓] (f) Stands for "Load Upper Immediate" ✓
- [ ] (g) Can be the second sub-instruction used in `la`
- [ ] (h) Upper 20 bits always set to 0
- [ ] (i) Is considered a PC-relative addressing instruction
- [✓] (j) Is considered a 0-relative (absolute) addressing instruction ✓

Select all possible options that apply. ?

✓ 100%

`auipc`

- [ ] (a) Can be the second sub-instruction used in `la`
- [✓] (b) Stands for "Add Upper Immediate to Program Counter" ✓
- [ ] (c) Often used as the second instruction with `jalr` in order to call a function
- [ ] (d) Sets `rd` to the upper 20 bits of `PC + imm` and lower 12 bits to 0
- [✓] (e) Often used as the first instruction with `jalr` in order to call a function ✓
- [✓] (f) Can be the first sub-instruction used in `la` ✓
- [ ] (g) Sets `rs1` to the upper 20 bits bits of (`PC + imm`)
- [✓] (h) Sets `rd` to `PC` + (`imm<<12`) ✓
- [✓] (i) Is considered a PC-relative addressing instruction ✓
- [ ] (j) Stands for "Add Unsigned Immediate to Personal Computer"
- [ ] (k) Is considered a 0-relative (absolute) addressing instruction

Select all possible options that apply. ?

✓ 100%

Try a new variant

Homework 4

Assessment overview

Total points:	50/100
Score:	50%

Question

Value:	8
History:	8
Awarded points:	8/8

Report an error in this question

Previous question

Next question

Attached files

No attached files

Attach a file

Attach text

Correct answer

Refer to the Reference Card to help you out.

What do the following instructions do?

`lui`

(a) Lower 12 bits always set to 0
(d) Can be the first sub-instruction used in `li`
(e) Upper 20 bits set to instruction-type immediate
(f) Stands for "Load Upper Immediate"

(j) Is considered a 0-relative (absolute) addressing instruction

`auipc`

(b) Stands for "Add Upper Immediate to Program Counter"
(e) Often used as the first instruction with `jalr` in order to call a function
(f) Can be the first sub-instruction used in `la`
(h) Sets `rd` to `PC` + (`imm<<12`)
(i) Is considered a PC-relative addressing instruction

1: `lui` is an U-type instruction (upper immediate); it like all other instructions with a destination register `rd` will set all 32-bits of the register. However, it is unique in the fact that it'll set the upper bits to something **other** than 0. This is because it takes the 20-bit immediate and puts it in the upper 20 bits (and the remaining lower 12 bits are set to 0). This is often useful when you want to load a 32-bit value (without considering sign/0-extended bits) into register. It's impossible to do with the RISC-V 32-bit architecture as if we need 32 bits for the immediate to load into a register, there is no other available bits to identify `rd` or instruction type. These two fields fill up the remaining 12 bits (7 bits for opcode, 5 bits for `rd`).

`lui` is also considered a 0-relative, or absolute, addressing instruction. This means that the U-type immediate it uses is added to 0 instead of the PC (program counter) and thus will address things starting from the "start" of memory as opposed to being an offset to where the current instruction is in memory.

One of the common ways you might see this instruction used is as the first instruction composing the pseudo-instruction `li` (load immediate) along with `addi` being the second instruction. This is because `li` takes in any arbitrary 32-bit number; however because we know we can't load 32 bits at a time, we have to break that up so `lui` takes the upper 20 bits of the address and puts it in `rd` and the subsequent `addi` instruction adds the lower 12 bits to make the 32-bit address. This general procedure is also how we load addresses and other large values.

2: `auipc` stands for "Add Upper Immediate to Program Counter"; the presence of "program counter" indicates that this is a PC-relative instruction, unlike `lui`. This means that the U-type immediate it uses is added to the `PC` at that instruction. However, the nuance behind how the immediate is added is that the immediate added to the PC is shifted to the upper 20 bits and the lower 12 bits are zeroed out again. This does not mean that the immediate itself is 32 bits and gets added to `PC` and **then** the lower 12 bits get zeroed out as the immediate, as with `lui`, cannot be a full 32 bits due to bit-restrictions.

There are two common ways you might see `auipc` used. The first is similarly to `lui`, it loads the label in the `la` pseudo-instruction; the difference between using `auipc` and `lui` that one address will be taken as a PC-relative address whilst the latter will be taken as an absolute address (you can also determine this at compile-time with various flags). Since most functions are relative-addressed, auipc is often used for `la` instead of `lui`.

The second common usage is returning to a specific memory address, e.g. using it in conjunction with `jalr` in order to call a function that's too far to jump normally (J-type instructions only have 20 bits of immediate, so we're limited in how far we can jump when calling a function). The way in which this works is such that `auipc` will load a PC-relative address into a register, and then `jalr` can address to an `imm` number of bytes offset from that address in order to unconditionally jump to a memory location.

Submitted answer 3 **correct: 100%**
Submitted at 2022-09-21 00:07:13 (PDT)

hide

Refer to the Reference Card to help you out.

What do the following instructions do?

`lui`

(a) Lower 12 bits always set to 0 ✓

(d) Can be the first sub-instruction used in `li` ✓
(e) Upper 20 bits set to instruction-type immediate ✓
(f) Stands for "Load Upper Immediate" ✓
(j) Is considered a 0-relative (absolute) addressing instruction ✓
✓ 100%

`auipc`

(b) Stands for "Add Upper Immediate to Program Counter" ✓
(e) Often used as the first instruction with `jalr` in order to call a function ✓
(f) Can be the first sub-instruction used in `la` ✓
(h) Sets `rd` to `PC` + (`imm<<12`) ✓
(i) Is considered a PC-relative addressing instruction ✓
✓ 100%

Submitted answer 2 **partially correct: 50%**
Submitted at 2022-09-21 00:06:28 (PDT)
show

Submitted answer 1 **incorrect: 0%**
Submitted at 2022-09-21 00:05:57 (PDT)
show