← Project 2: mcu91 CPU

Due Saturday, 12/8 by 11:59 PM (or late on Sunday)

For this project, you will be **building a small single-cycle CPU.** Your CPU will be capable of running real programs (with conditionals, function calls, etc) and outputting numbers.

Project overview

The mcu91 architecture is a **16-bit architecture** similar in scope and complexity to real CPUs from the 1970s and 1980s. It's mostly a RISC architecture but has some CISC bits in there too, to make it interesting 😉

Breaking it up into pieces will make it much easier to build, test, and fix.

You also get partial credit for broken instructions, or for pieces of circuitry for an instruction that isn't implemented.

Click here for the mcu91 ISA.

Keep it open along side these details:

Project Details

Click the link above for all the juicy implementation details.

Extra Credit

Done? Bored? Try this.

Logisim Hints

Ways to use Logisim more effectively, letting you work quicker.

Test Programs

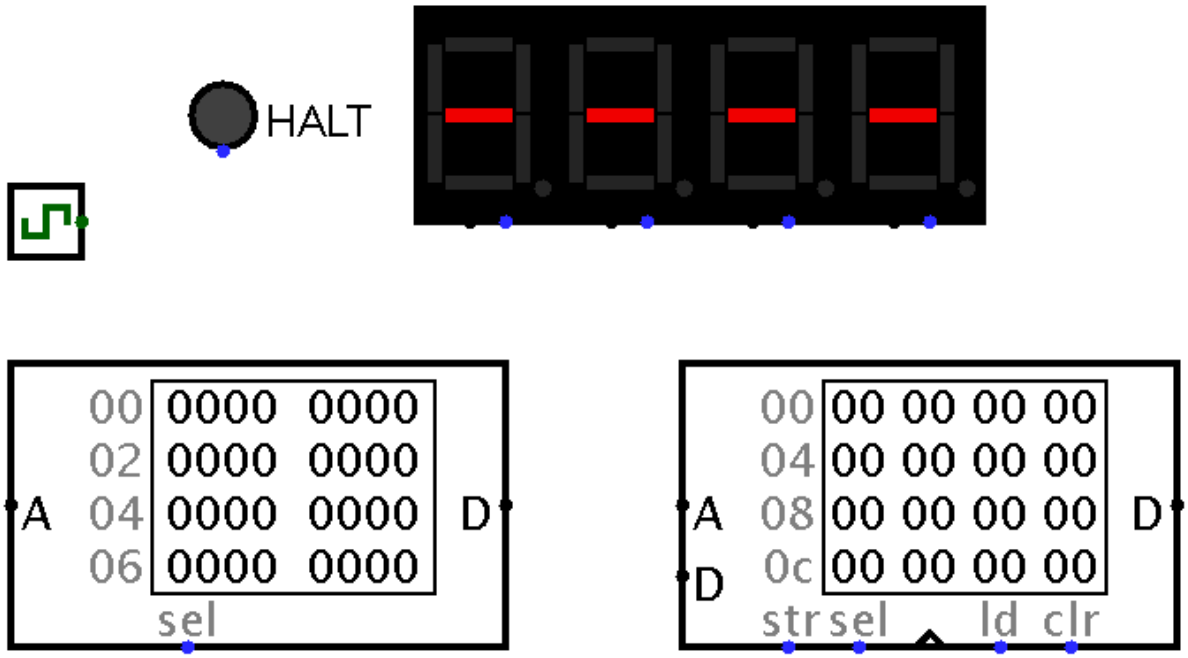
Assembled programs for you to load into your machine's ROM to test it.

Submission details

> Your CPU can look any way you like, **BUT IT MUST HAVE THE FOLLOWING THINGS ON ONE SCREEN IN THE MAIN CIRCUIT:**

- The Program ROM component
- The Data RAM component
- The output display digits
- The halt LED
- The ONE clock component that will control the entire CPU

Kinda like this:



But with, like, wires and tunnels and a whole CPU hooked up to them.

You will submit a ZIP file named `username_proj2.zip` where `username` is your Pitt username. For example, I would name mine `jfb42_proj2.zip`.

Do not put a folder in the zip file, just the files you are submitting. When we open the zip file, we should see the following:

- Your circuit file, named `username.circ`, like `jfb42.circ`.
  - if you got fancy and used multiple circuit files, name the main one like that.
- A **readme.txt** file. **DO NOT SUBMIT A README.DOCX. DO NOT SUBMIT A README.PDF. SUBMIT A PLAIN TEXT FILE. *PLEASE.*** It should contain:
  - Your name
  - Your Pitt username
  - Anything that does not work

    - Any extra credit you did
    - Anything else you think might help us grade your project more easily

Submit here as usual.

Grading Rubric

The points break down as follows:

- **[10 points]:** Submission and style
    - **[5]:** You submitted your project properly (follow the directions above).
    - **[5]:** Your circuit is split into components, and you've **made it easy to grade.**
        - This is the "grader is mad at you because your circuit is a mess" category.
- **[90 points]:** The CPU
    - **[40 points]** for building the CPU components other than the control
        - **[5]** Display/Halt LED
        - **[10]** Register File
        - **[5]** ALU
        - **[10]** Call stack
        - **[10]** PC Controller
    - **[50 points]** for implementing the control for each instruction
        - **[4]** Halting (`halt`)
        - **[4]** Display (`put`)
        - **[6]** Loading immediates (`li, lui`)
            - *Up to here is a 64*

        - **[6]** Load/store (`ld, st`)
        - **[8]** Basic ALU (`add, sub, and, or, not, shl, shr`)
            - *Up to here is a 78*

        - **[4]** Immediate ALU (`adi, sbi, ani, ori, sli, sri`)
        - **[4]** Unconditional branches/jumps (`j` , `jr`)
            - *Up to here is an 86*

        - **[8]** Conditional branches (`blt, bge, beq, bne, bez, bnz`)
        - **[6]** Call instructions (`call, ret`)
- **[+20 points]:** Extra Credit (up to 20 points extra!)