i281e CPU

SR Design Team 14 - Implement the i281 CPU in Hardware

Daryl Damman, Logan Lee, Grant Nordling, Braxton Rokos, Gavin Tersteeg

Client/Advisor: Professor Alexander Stoytchev

Problem

The students in the digital logic course lack a physical design of an i281 CPU they can use to apply their learning.

Solution

This project provides a raw physical design of the i281 CPU on breadboards that students can interact and learn with and a slick PCB physical design containing all expanded features and design choices.

Tools: KiCad, Freerouting, GitLab, OneDrive, and Microsoft Office

Medium: Breadboard, PCB, and Software

Context

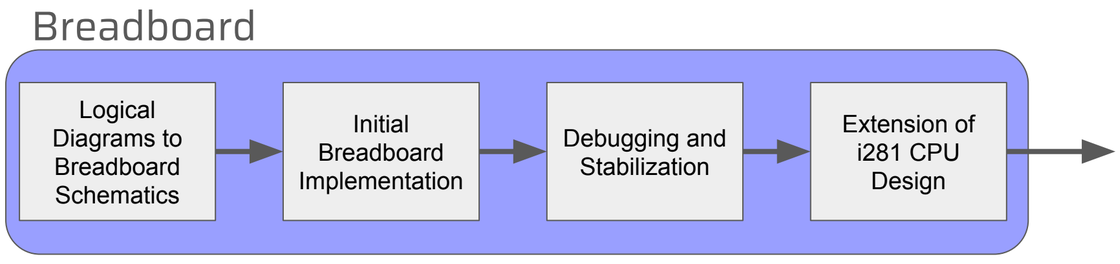
Intended Audience: CprE 2810 and 4810 Students

Uses: an educational teaching tool

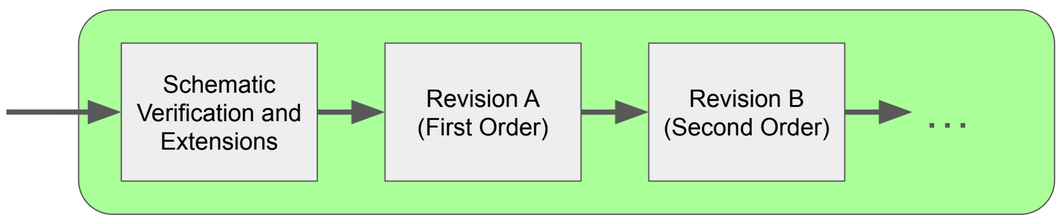
Operating Environment: lab and lecture

Design Approach

Breadboard



PCB



Testing

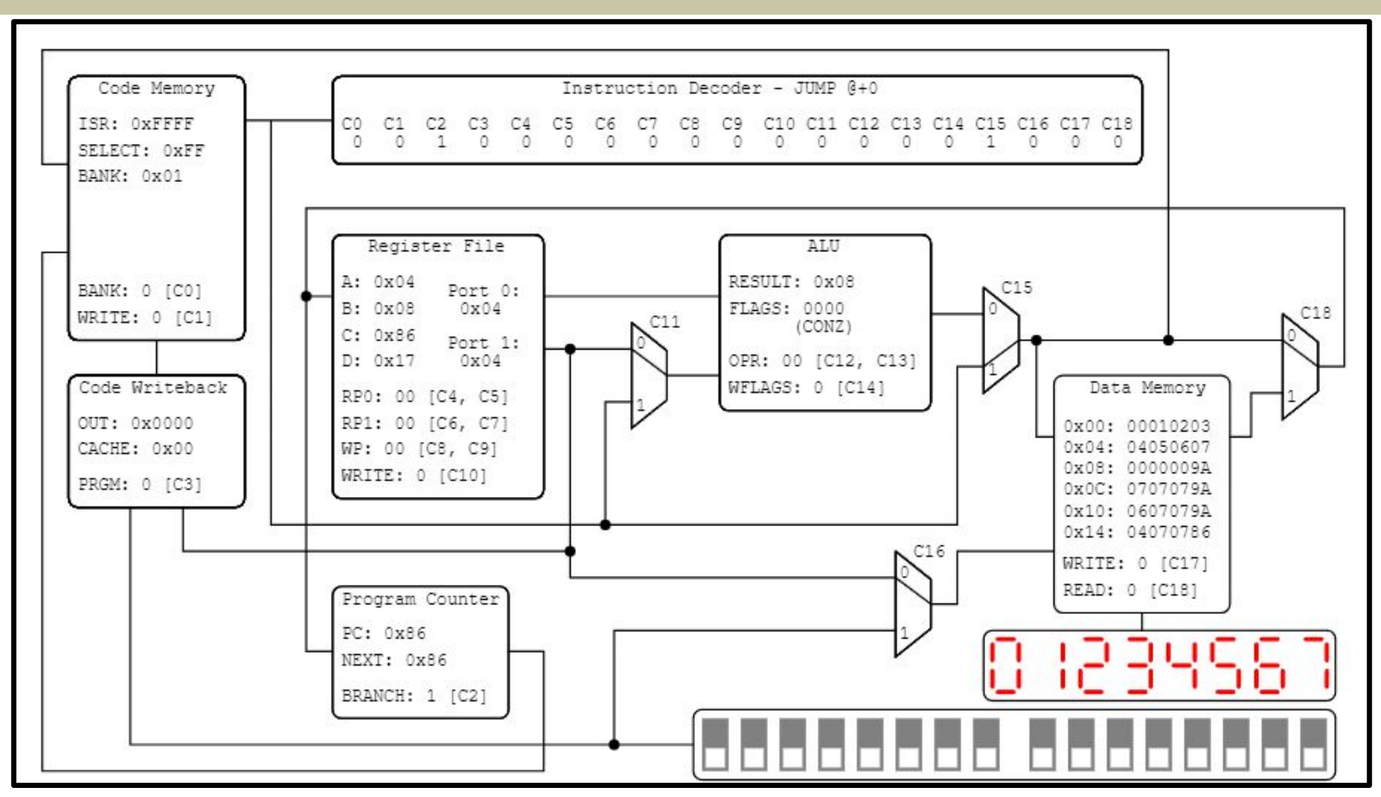
- Individual Module Testing
- Self-Check Program for Several Opcodes
- Reassurances Testing
- Power Analysis
- Clock Frequency Limit Tests
- Verification Testing
- Full System Integration Testing
- Comparing Simulator Results to Hardware Results

Specifications

- Clock Speed: 1Hz up to 2MHz (up to 2.5MHz overclock)
- Power Requirements: 0.8A @ 5VDC (Input 5-12VDC)
- Memory: uses EEPROM for ROM and Control Table
- BIOS: 128 Words
- Code Memory: 32 Kw
- Data Memory: 32 KB
- Compact Flash Memory “Hard Disk”: 32 MB

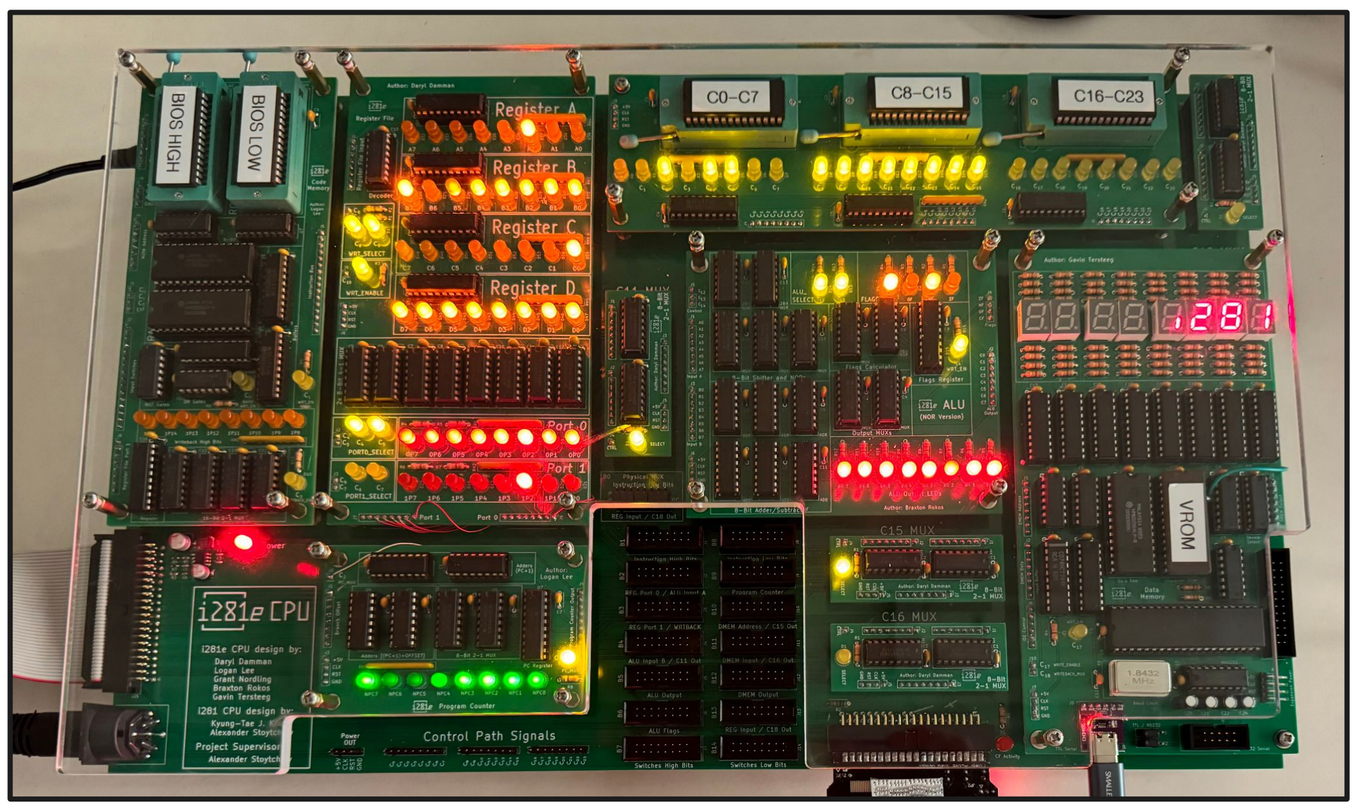
External Functionally

- External Terminal Support
  - DOS 281
  - Selection and Uploading Programs
- Expansion Bus
  - Printer, Sound, PS2 Keyboard, PS2 Mouse, etc.



Design Requirements

- Must be explainable to sophomores in CprE 2810.
- CPU data path and control path must be visualized through LEDs.
- Adjustable CPU clock to enable stepping through individual instructions and operating at different clock frequencies.
- CPU must be capable of playing the original i281 PONG video game and other example programs from FPGA and Web Simulator.
- CPU design must be as close as possible to the original design.
- It must look *cool*.



Technical Details

ALU
Performs the basic arithmetic between two operands via Register File and/or immediate values.

Register File
Handles intermediate volatile memory to store results from the ALU, Data Memory, or instruction immediate values.

Program Counter
Indicates the location of instruction line in Code Memory.

Data Memory
Manages runtime and long-term memory storage between registers and compact flash. Handles peripherals

Code Memory
Contains of the instruction set for the i281e CPU during operation. The BIOS is kept in ROM chips, well the RAM is used to load in the user program.

Control Table
Converts the instruction set into control signals and distributions them to the rest of the i281e CPU.

Mainboard
Connects all of the modules together. Allows for modules to be swapped and replaced with external versions.

MUXs
Outputs one of two input busses depending on the corresponding input control signal.

Front Panel
How the user interacts with the i281e CPU when running. Home to the inputs switches, clock controls, reset line, and debugging switches.

Acknowledgements: Matthew Post, Lee Harker, ETG Student Staff, Megan Pezley