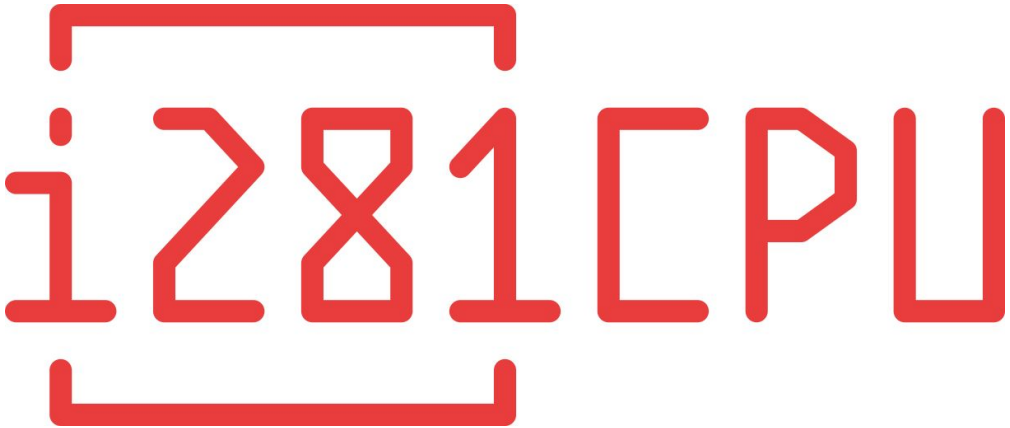



# Testing

Team Members:

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

# Unit Testing

**Parameters Tested:**

- Voltage
- Ameres
- Proper Function

**Proper Function Testing:**

- Custom Testing Rigs
- Equipped with LEDs and Switches to Simulate External Inputs

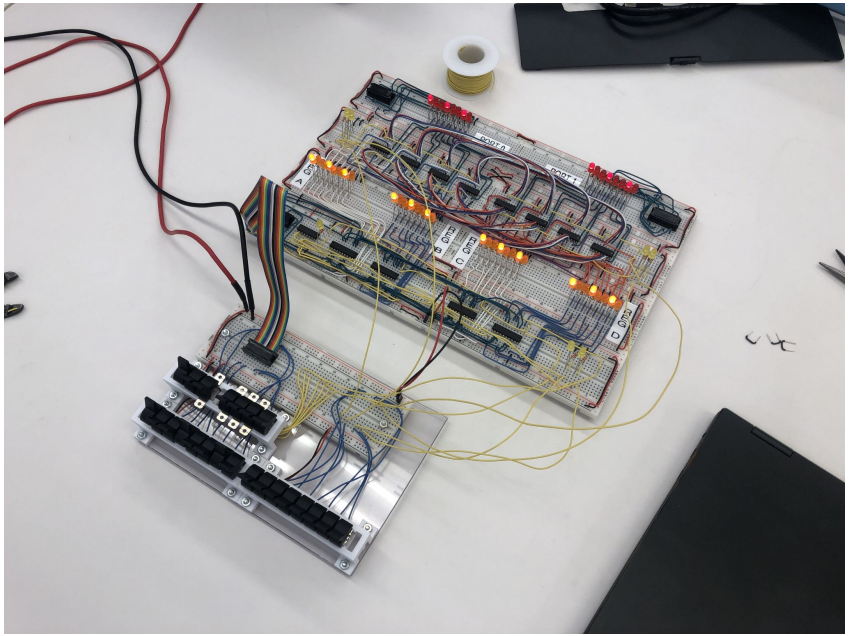
**Testing Tools:**

- Multimeter
- Oscilloscope
- Clock Pulse Generator
- Bench Power Supply

**Switch Testing Methodology:**

- Observing Circuit Reactions to Input Changes

# Ensuring Robust Functionality



**Comprehensive Testing:**

- Covers Most, If Not All, Circuit Scenarios

**Unit Definition:**

- Lowest Component of Our System
- Focus on Modules, Not Chips

**Alternative Definition:**

- Testing Units without State
- No Worry about Managing Multiple States

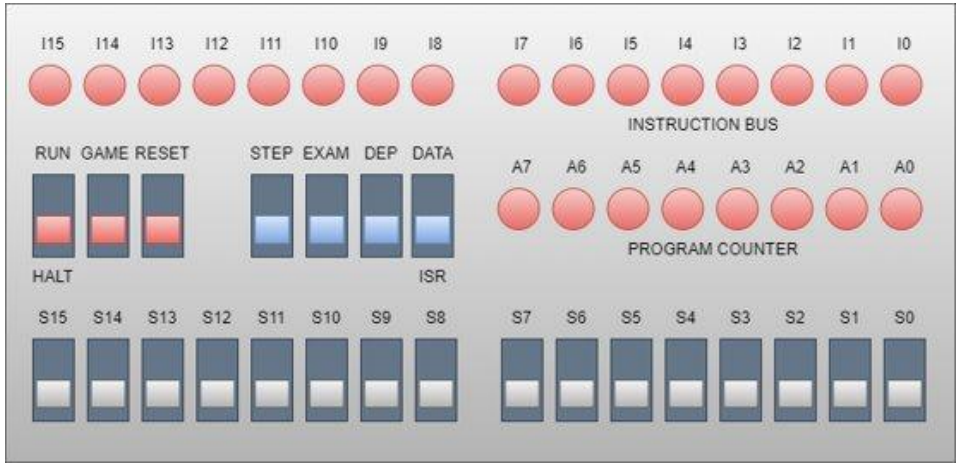
# User Panel - Primary Interface

**Primary Interface:**

- User Panel

**Input Devices:**

- 16 Switches with LED Indicators
  - Code and Data Entry into Memory
- Rotary Encoder
  - Control Clock Speed
- Manual Step Switch



# Data Bus Lines and Boot Hard Disk (BHD)

**Data Bus Lines:** High and Low Connections

- Direct Link to Processor
  - Both Connect to Code Memory (RAM) in INPUTC/CF Instruction
  - Low Bus Line Connects to Data Memory (DMEM) in INPUTD/DF Instruction

**Boot Hard Disk (BHD):**

- Swappable for Different User/Example Programs
  - Not Considered a Traditional Interface
- Testing Boot Procedure:
  - Swap Programs on BHD
- Used to Confirm:
  - Instructions Load into RAM
  - Critical Control Lines are Stable
  - Post-Boot Operation is as Expected

# Integration Testing

**Critical Integration Paths:**

- Data Bus Cables
- Individual Control Lines
- Justification: Previous project lacked modularization

**Testing Approach:**

- Integration of All Modules
- Final Stage Post System Testing

**Justification for Criticality:**

- Derived from Requirements

# Integration Testing

**Minimum Viable Processor:**

- CMEM, ALU, Register File, Control Table, Program Counter
- Demonstrates Core Processor Capability

**Full Processor:**

- Includes All Processor Modules
- Comprehensive Testing with All Components

**Testing Tools:**

- Front Panel Debugging Utilities
- Simple Test Programs to Verify Instructions

# System Testing

**Individual functionality tests for each island**

**Merging the islands will require full CPU testing to ensure proper function**

**Testing Tools:**

- Two testing rigs capable of two 8-bit inputs and one 8-bit output each
- Arduino Nano to simulate the clock
- LEDs to visualize the output of the module

# Regression Testing

**Ensuring Compatibility**

- Verify Compatibility Between New and Old Components
- Pre-connectivity Testing to Prevent Breaks
- All Existing i281 Assembly Programs Must Continue to Work

**Continuous Module Testing:**

- Throughout Both Semesters
- Module-by-Module Basis

| | Assembly Code: | | | Machine Code: |
|---|---|---|---|---|
| | .data | | | View Data Memory |
| 0 | x | BYTE | 3 | |
| 1 | z | BYTE | ? | |
| | .code | | | Instruction Memory: |
| 0 | LOAD | A, | [x] | 1000_00_00_00000000 |
| 1 | MOVE | C, | A | 0010_10_00_00000000 |
| 2 | MOVE | B, | A | 0010_01_00_00000000 |
| 3 | SHIFTL | B | | 1100_01_00_00000000 |
| 4 | SHIFTL | B | | 1100_01_00_00000000 |
| 5 | ADD | C, | B | 0100_10_01_00000000 |
| 6 | STORE | [z], | C | 1010_10_00_00000001 |

# Acceptance Testing

**Function Testing:**

- Post-Connection Testing
- Validate Functionality via 7-Segment Displays and LEDs

**Critical Features:**

- Driven by Requirements
- Must be Compatible with Existing i281 Software

**Requirement Verification:**

- Checked during Design
- Affirmed through Implementations

# Thank you

# Questions?