

Software CPU Design Project

CMPE220 – Systems Software – Final Project

Team Members

- Neel Asheshbhai Shah
- Vedant Tushar Shah
- Aarav Pranav Shah
- Harshavardhan Kuruvella

Course: CMPE220 – Computer Architecture

Semester: Fall 2025

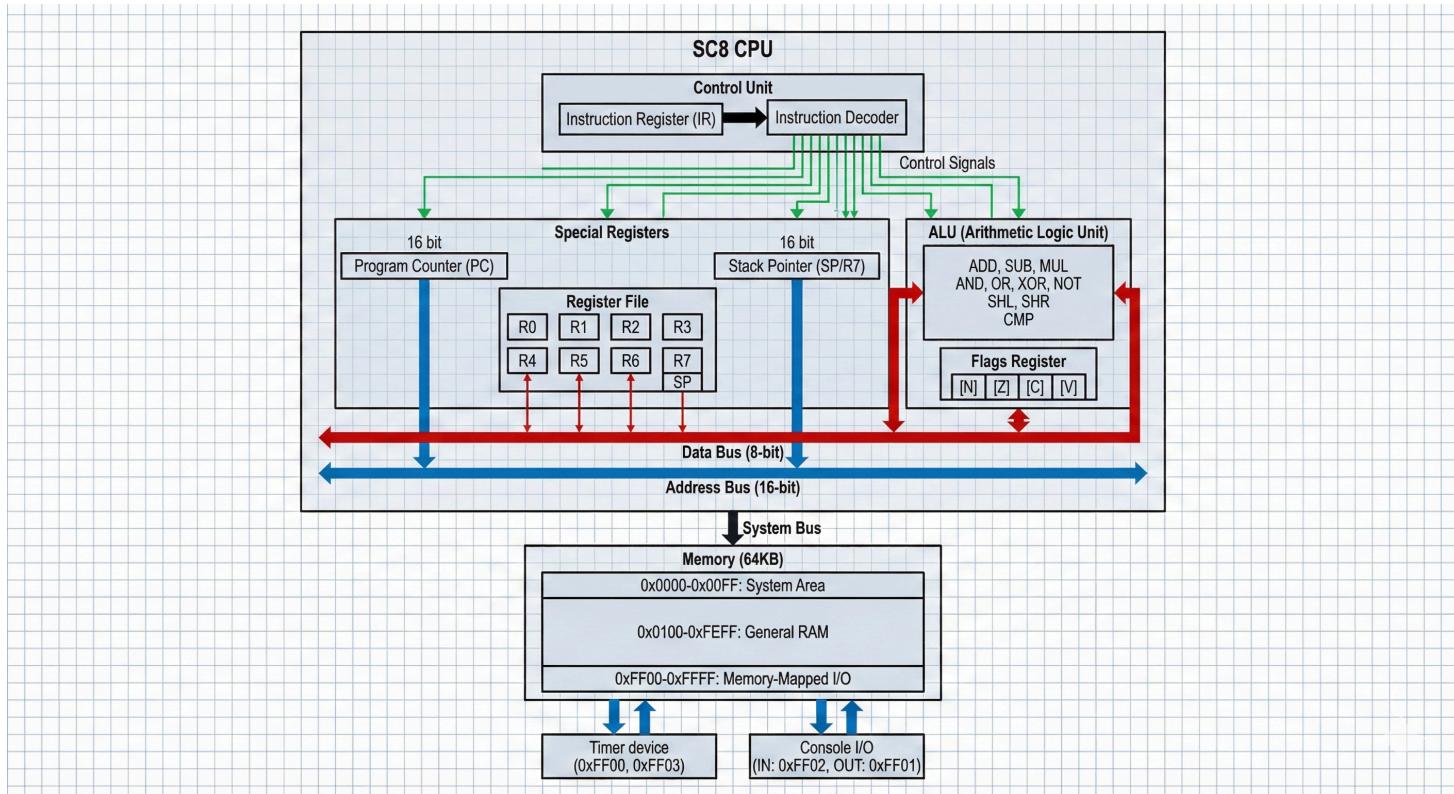
Submission Date: November 26, 2025

Demo & Architecture

Video Demonstration

[Watch Demo Video](#)

CPU Architecture Diagram



GitHub Repository

The complete project source code, documentation, and demo video are available on GitHub.

Repository URL:

Example: https://github.com/SpartaNeel1010/CMPE220_CPU_DESIGN

Repository Contents

- Complete source code for CPU emulator (C++)
- Complete source code for assembler (C++)
- Sample assembly programs (Timer, Hello World, Fibonacci)
- Comprehensive documentation (ISA specification, architecture details)
- Build system (Makefile)
- Project report (this document)

Download, Compile, and Run Instructions

Step 1: Download the Project

Option A: Clone from GitHub

```
git clone https://github.com/SpartaNeel1010/CMPE220_CPU_DESIGN  
cd CMPE220_CPU_DESIGN
```

Step 2: Verify Prerequisites

Check Installed Tools

```
g++ --version  
make --version  
git --version
```

If Tools Are Missing

Ubuntu/Debian:

```
sudo apt update  
sudo apt install build-essential git
```

macOS:

```
xcode-select --install
```

Windows (WSL):

1. Install WSL
2. Follow Ubuntu instructions

Step 3: Build the Project

Navigate to the project directory and run:

```
make all
```

This builds:

1. CPU emulator
2. Assembler
3. All .asm programs → .bin format

Step 4: Run the Programs

Run Hello World

```
make run-hello
```

Run Fibonacci Sequence

```
make run-fib
```

Expected Output:

```
Running Fibonacci program...
=====
...
Fib: 0 1 1 2 3 5 8 ## ## ## Done!
```

represents numbers ≥ 10 (single-digit display limitation)

Run Timer Example

```
make run-timer
```

Expected Output:

```
Running Timer program...
=====
1
2
3
4
5
Done
```

Step 5: Run All Tests

```
make test
```

Runs all sample programs sequentially.

Custom Programs

1. Create your program:

```
programs/my_program.asm
```

2. Assemble it:

```
./bin/assembler programs/my_program.asm programs/my_program.bin
```

3. Run it:

```
./bin/emulator programs/my_program.bin
```

Step 6: Clean Build Artifacts

```
make clean
```

Removes:

- Compiled binaries (bin/)

- Assembled .bin files

Getting Help

- **ISA Reference:** docs/ISA_SPECIFICATION.md
- **Architecture Details:** docs/CPU_ARCHITECTURE.md
- **Main README:** README.md

Team Member Contributions

Neel Asheshbhai Shah — CPU Emulator & Memory System

- Implemented CPU core: register file, PC, fetch-decode-execute
- Added arithmetic, logical, and branch operations
- Built 64KB memory subsystem with memory-mapped I/O
- Tested and validated emulator
- **Key Files:**

src/emulator/cpu.{h,cpp} , memory.{h,cpp} , main.cpp

Vedant Tushar Shah — Assembler & Build System

- Designed and built two-pass assembler
- Implemented lexer, parser, and code generator
- Added symbol table for label resolution
- Created Makefile with dependency management
- **Key Files:**

src/assembler/*.h,*.cpp , Makefile

Aarav Pranav Shah — ISA Design & ALU Implementation

- Designed full ISA with 5 instruction formats

- Implemented ALU operations with flags: N, Z, C, V
- Developed system bus and control unit
- Authored ISA and architecture documentation
- **Key Files:**
`src/emulator/alu.{h,cpp}` , `bus.{h,cpp}` , `docs/*`

Harshavardhan Kuruvella – Sample Programs & Documentation

- Wrote sample programs: Hello World, Fibonacci, Timer
- Created documentation including README and demo script
- Authored project report
- Prepared demo video and presentation
- **Key Files:**
`programs/*.asm` , `README.md` , `docs/DEMO_SCRIPT.md` , `report/PROJECT_REPORT.md`