8-Bit Emulated Computer

By Venkat Rao, Alexander Chin, Betty Nguyen, and Vibhav Adivi

Introduction

Statement of Purpose

Our goal with this project was to emulate an 8-bit system with physical peripherals on a breadboard in order to learn about how CPUs work, how they interface with other devices, and to learn more about emulation. This project has allowed us to apply the assembly language and electronics skills that we learned in ECE 110/120. We planned to write an emulator for the Chip-8, a system designed for emulation on programmable calculators and to be relatively easy to emulate. We also planned to build a keypad and buzzer circuit, and attach them and a display to our microcontroller.

Features and Benefits

Because the Chip-8 was designed to be easy to emulate and code for, there is plenty of software already available for it. When testing, we were able to find many programs that ran perfectly, meaning that we didn't have to worry about learning assembly language. In addition, due to the Arduino Nano Every's relatively low cost, we ended up using only about ~$20 of off the shelf components, which is very cheap considering that the computer is fully functional, complete with simple keyboard, sound, and display systems. We believe that this computer would be useful for others to build as a learning experience, as it is inexpensive, and can help teach simple analog and digital electronics. Once completed, one can also easily program the system in assembly, and because there is a lot of documentation for the Chip-8, it is one of the easiest systems to write assembly code for.

Design

System Overview

This project consists of a simple computer, with a microcontroller connected to a display, keypad, and buzzer. For input, we planned on building a hexadecimal 4x4 keypad made with switches and a printed circuit board, which is polled by our microcontroller 60 times per second. We also created a buzzer circuit using a 555 timer, a TTL AND gate chip, and an off-the-shelf piezo buzzer. Finally, we used a 128x64 pixel display, connected via I2C to the Arduino, which we then wrote a driver to control. The driver upscales the 64x32 screen output of the CHIP-8, and sends packets of hardware commands and image data to the screen at a rate of 30 frames per second. The microcontroller itself, an Arduino Nano Every, emulates the Chip-8, an 8-bit computer designed to be easy to emulate and code for. It also controls the physical keypad, display, and buzzer, connecting them to the emulated computer.

Results

Our final system is completely functional, with working graphics, sound, and input. A demo video was recorded using a sample game, available at https://youtu.be/63IcjMNUtRg. We did not end up finishing the keypad, due to COVID-19's disruption of our schedule. However, we were still able to write a driver for it, using a functionally equivalent keypad purchased online as a substitute. The source code is available at https://github.com/venkatrao1/emulated-chip8.

Problems and Challenges

Because half of our group was in-person and the other half is not on campus, the project was a little slow to start and work through because we could not meet up and work on the project in person. In addition, our group didn't have as much experience on the hardware side of the project, so there was some difficulty in finding the right parts and making the keyboard, which required a lot of soldering. Finally, due to geographical limitations, we were not able to connect the keypad to the rest of the system physically, so we had to substitute with an off-the-shelf keypad.

Future Plans

In the future, we hope to finish the keypad, and perhaps add support for extensions of the instruction set such as SCHIP-8 and MegaChip8, allowing for more complex games to be run. However, the Arduino's 6 kilobytes of RAM mean that in order to make the system more complicated, a different microcontroller might have to be used.

References

B. Eater, "Build an 8-bit computer from scratch: Clock module", *Eater.net*. [Online]. Available: https://eater.net/8bit/clock. [Accessed: 10- Dec- 2020].

L. Muller, "How to write an emulator (CHIP-8 interpreter) — Multigesture.net", *Multigesture.net*, 2018. [Online]. Available: http://www.multigesture.net/articles/how-to-write-an-emulator-chip-8-interpreter/. [Accessed: 04- Dec- 2020].

T. Greene, "Cowgod's Chip-8 Technical Reference", *Devernay.free.fr*, 1997. [Online]. Available: http://devernay.free.fr/hacks/chip8/C8TECH10.HTM. [Accessed: 10- Dec- 2020].