

ESP32 8086 Hardware Emulator using FabGL and PCF8574

This document provides a detailed guide on building a hardware emulator for the Intel 8086 microprocessor using an ESP32, the FabGL library, and PCF8574 I/O expanders. This project will allow you to interact with and understand the inner workings of the 8086 at a hardware level.

Project Overview

The core of this project is the FabGL library, which includes a powerful x86 emulator. We will use an ESP32 to run this emulator. The emulated 8086's address, data, and control lines will be brought out to physical pins using PCF8574 I/O expanders. This will allow you to connect external hardware, such as memory and peripherals, to your emulated 8086 and watch it interact with them in real-time.

Hardware Requirements

- ESP32 Development Board
- 5 x PCF8574 I/O Expander ICs
- LEDs for status indicators (optional, but recommended)
- Resistors for LEDs and pull-ups
- Breadboard and jumper wires

Hardware Design and Schematic

The 8086 has a 20-bit address bus and a 16-bit data bus. To manage this number of I/O lines, we will use five PCF8574 I/O expanders. Each PCF8574 provides 8 bits of I/O, so five will give us 40 bits, which is enough for the address and data buses, plus the essential control signals.

Here is the mapping of the 8086 signals to the PCF8574 expanders:

- **PCF8574 #1 (Address 0x20):** 8086 Address Bus A0-A7
- **PCF8574 #2 (Address 0x21):** 8086 Address Bus A8-A15
- **PCF8574 #3 (Address 0x22):** 8086 Address Bus A16-A19 and Control Signals (RD, WR, M/IO, ALE)
- **PCF8574 #4 (Address 0x23):** 8086 Data Bus D0-D7
- **PCF8574 #5 (Address 0x24):** 8086 Data Bus D8-D15

The I2C addresses of the PCF8574s are set by pulling their A0, A1, and A2 pins high or low.

Schematic

[Detailed schematic diagram will be placed here showing the connections between the ESP32, PCF8574s, and the 8086 bus breakout]

Note: A detailed schematic is a complex visual element. For now, please follow the connections described in the code and the pin mapping table. A full graphical schematic can be generated with appropriate tools.

Software Implementation

The software is written in C++ using the Arduino IDE and the FabGL library. You will need to install the FabGL library in your Arduino IDE before compiling the code.

The code is structured as follows:

1. **Initialization:**
 - Initialize the serial port for debugging.
 - Initialize the I2C bus for communication with the PCF8574s.
 - Initialize the FabGL library and the x86 emulator.
2. **Bus Control Functions:**
 - Functions to write to the address and data buses via the PCF8574s.
 - Functions to read from the data bus.
 - Functions to control the RD, WR, M/IO, and ALE signals.
3. **FabGL Hooks:**
 - We will use FabGL's memory and I/O hooks to intercept the 8086's bus cycles.
 - When the emulated 8086 performs a memory or I/O read or write, our custom functions will be called. These functions will then drive the physical bus using the PCF8574s.
4. **Main Loop:**
 - The main loop will run the FabGL emulator.
 - The emulator will execute 8086 code, and any interaction with memory or I/O will be reflected on the physical hardware bus.

Here is the complete Arduino sketch for the project.