Chapter 1

Microprocessor-Based Systems

Objectives

- Block diagram of a microprocessor-based system and function of each section
- Processing cycle of a microprocessor
- Way software is used to initialize hardware and peripherals
- History of the microprocessor and of the different generations of computers
- How 68000 code is created and used in a b8000-based system
- Calculator project
- Some typical errors encountered during program development

Introduction

- The purpose of this chapter is to show how a microprocessor is used in a small system and to introduce you to the operation of the personal computer.
- We will see what types of hardware may be connected to the microprocessor, and why each type is needed.
- We will also see how software control hardware and how software can be developed.

Evolution of Microprocessors

- First generation: large, vacuum tubes, slow, prone to breakdown, and performed a limited number of instructions.
- Second generation: transistor, shrank in size, and increased in power.

Evolution of Microprocessors

- Third generation: integrated circuit, decrease in size and increase in computing power. Intel 4004 and 8008. Limited in power and addressing capability.
- Forth generation: LSI and VLSI, even more powerful.

Evolution of Microprocessors

- Modern computers can do more complex operations and address billions of bytes of information.
- CISC(complex instruction set computer) and RISC(reduced instruction set computer).
- Fifth generation: whole artificial intelligence movement is pushing toward that goal, a machine that can think!
System Block Diagram

- Standard elements: memory, timing, and input/output (I/O)
- Special sections: A/D converters, D/A converters, interval timers, math coprocessors, complex interrupt circuitry, speech synthesizers, and video display controllers.

System Block Diagram

- All components communicate with **system bus**, which is composed of address, data, and control signals.
- CPU, Memory, Timing, parallel I/O, serial I/O, and interrupt circuitry.

System Block Diagram

- CPU: central part.
- Memory: ROM and RAM. RAM: static RAM, dynamic RAM, and nonvolatile memory. Floppy or hard disk.
- Interrupt circuitry.
- Serial and parallel communications.

Serial Communication

Microprocessor Operation

- Same pattern: endless fetch, decode, and execute cycles.
- Decode: decide what type of instruction has been fetched.
- Execute cycle: may need to read more data from memory or write results to memory.
Microprocessor Operation

- Most microprocessors will also include a set of control signals that allow external circuitry to take over the system bus.
- Parallel processing – multiple-processor systems.
- Microcontrollers – simple control systems with built-in RAM, ROM, etc.

Hardware/Software Requirements

- ROM—to take care of peripheral initialization.
- Start-up software.
- Watchdog monitor.
- Power-down program.
- For some systems, system bus must be made available to the outside world.

Historic Perspective: The Macintosh Personal Computer

- The 68000 came out in the late 1970s, the Apple chose it for use in the Macintosh personal computer.
- Now, use PowerPC technology.

Historic Perspective: The Macintosh Personal Computer

- The initial Macintosh contained a keyboard, a monochrome video display, one or two floppy disk drives, and a memory.
- Equipped with a software program – operating system.
- Most of the electronics were on a single printed circuit board – motherboard.
Historic Perspective: The Macintosh Personal Computer

Developing software for the 68000
- Machine language vs. assembly language.
- Hello: our first machine language program.
- Creating machine code with ASM68K
- Running machine language programs with EMU68K
- COM68K

The Calculator Project
- To show how an entire 68000-based system is developed.

Troubleshooting Techniques
- Typographical errors
- Invoke EMU68K incorrectly or not use the correct options
- Run-time errors
- Keep these common errors in mind
- Pay attention to details

Summary
- Examined operation of microprocessor-based systems.
- Through the use of many different types of peripherals, such as parallel and serial devices and A/D converters, a system can be tailored to perform almost any job.