Chapter 1

Microprocessor, Microcontroller and Programming Basics

Course objectives

- to develop an in-depth understanding of
  - the operation of microprocessors and microcontrollers
  - machine language programming
  - microprocessor interfacing techniques
- to be able to use Motorola 68HC11 microcontroller competently
- to be able to design and implement microprocessor-based systems in both hardware and software
- to be able to apply this knowledge to more advanced structures

Common computer organization

ALU (Arithmetic Logic Unit) is a circuitry, which is capable of doing various operations (for example ADD, SHIFT, AND, OR, etc) on certain on-chip registers.

CPU (Central Processing Unit) is the combination of the control logic, associated registers and the arithmetic logic unit.

Computer buses

Address bus: carries the address of a unique memory or input/output (I/O) device
Data bus: carries data stored in memory (or an I/O device) to the CPU or from the CPU to the memory (or I/O device)

Control bus: is a collection of control signals that coordinate and synchronize the whole system

**Block diagram of a typical CPU**

![Block diagram of a typical CPU]

**CPU structure**

The main functions are
- data transfer
- arithmetic and logic operations
- decision making (instructional flow control)

The register array consists of at least one accumulator, program counter and stack pointer.

The control unit controls all the operations in a CPU and basically it puts the CPU in one of the *fetch* and *execution* phases.

**Memory**

The memory in a computer system stores the data and instructions of the programs.
Main memory types

- ROM (read-only memory)
  - programmed permanently at the factory, cannot be altered

- RAM (random-access memory)
  - read and write memory

- EPROM (erasable programmable ROM)
  - nonvolatile, written electrically but erased optically

- EEPROM (electrically ROM)
  - nonvolatile, both written and erased electrically

Operation of the CPU
• Fetch cycle (phase):
  - The CPU puts the address of the instruction to be executed on the address bus. The address information comes from the program counter (PC) maintained by the control unit.
  - The control bus holds the information for reading the memory location and the data bus holds the instruction from the memory which is stored into the instruction register (IR)
  - PC is updated to point to the next instruction.

• Execute cycle (phase):
  - Instruction in the IR is decoded
  - The required data transfer and the required logical and arithmetic operation are performed
  - The result is written back either to one of the registers or memory or I/O device

• Common operations performed in ALU are:
  - addition, subtraction,
  - logical AND, OR, XOR, NOT
  - increment, decrement, shift, clear, etc.

Microcomputer

  - CPU on a single IC ➔ microprocessor (µP).
  - The terms CPU, µP and MPU (microprocessor unit) are synonymous
  - CPU is the combination of the ALU and control unit of any computer.
  - When the CPU is a single IC; it is called a µP and is also referred to as the MPU.
  - When the MPU is connected to memory and I/O, the arrangement becomes a MICROCOMPUTER.
  - Many different MPUs are produced by many different IC vendors (e.g., Motorola, Intel, Texas Instruments, Zilog, National Semiconductors, etc.).
  - Major differences are in
    - word size,
    - the number and types of instructions that can be performed,
    - the types of external control signals available,
    - the amount of memory that can be addressed.
Microcontroller

- A microcontroller unit (MCU) contains an MPU, memory, and I/O circuitry on a single chip.
- Such a chip can perform control operations without the need for any external circuitry.
- E.g. Motorola MC68HC11 (also a microprocessor in the expanded multiplexed mode), Intel 8051

Microprocessors vs microcontrollers

- Microprocessors: high performance, general purpose “brains” for PCs and workstations
  - Instruction decode and control, arithmetic/logic operations, registers, timing, external control
  - Typical cost: $75 -- $500, Annual demand: 10s of millions
- Microcontrollers: devices with high levels of integration for embedded control
  - Microprocessor functions plus on-chip memory and peripheral functions (e.g. ports, timers)
  - "Swiss army knife" of microprocessor technology
  - Typical cost: $1-- $25, Annual demand: billions!

Bottom-up view of microcontroller systems
Microcontroller applications

- Pocket pagers (low-power, interprets characters, user interface)
- Cameras (low-power, exposure and focus control, user interface)
- "Levelmeter" (measures angle, audible and visual user interface)
- Keyboard controllers (scanning, debounce, autorepeat, diagnostics)
- Modems (one for data transmission, one for command processing)
- Plotters (command interpretation, encoders, motor control)
- Color copiers (paper positioning, color exposure, sensors)
- Charge card pay phones (card reading, dialing, carrier access)
- Lawn sprinkler controller (timer, valve control, user interface)
- Instrumentation (user interface, GPIB interface, compute values)
- Closed-loop engine control (fuel/air mixture, ignition, pressure sensing, etc.)
- Anti-lock braking system control (monitors traction, controls brake)
- Dynamic ride control (adjusts suspension)

**Block Diagram of a Motorola M68HC11 Based Applications Example: An Emergency Control System**

**Comparison of some popular microcontrollers**

- Motorola HC11
  
The HC11 family includes a large number of variations, ranging from single-chips to larger micros with bank-switching support.
- Motorola HC12
The HC12 family is the next step up from the HC11, offering better performance, better memory management, and more features.

- **Motorola 6805**
  The 6805 family is the next step down from the HC11, with more single-chip variations, smaller packages, and lower cost.

- **Microchip PIC**
  The devices are readily available, reasonably priced, very fast, and come in lots of variations, however, with small memory maps and poor support for compilers.

- **Intel 8051**
  The 8051 was one of the first microcontroller families, and remains one of the most commonly used. The devices are available from multiple sources, are cheap, have decent tools, and offer a nice upgrade path to larger and more capable part.

**What is EE447 about?**

- MC68HC11 is the MPU choice in EE447
- You will be studying
  - how it is combined with memory and I/O devices to produce a 68HC11 based microcomputer, and
  - how it is programmed to perform various operations
- Most of what you learn about the 68HC11 can be applied to any other MPU or microcomputer you may encounter later.

**Major features of Motorola M68HC11**

- HCMOS Technology (low power / high speed)
- On-chip RAM, ROM, EEPROM
- Basic core functions of MC6801 – improved instruction set functionality
- 2 operating modes and 2 test modes
- On-chip counter / timer
- On-chip analog- to- digital conversion
- On-chip parallel and serial ports
- Improved interrupt capabilities than earlier products -- supports 21 interrupt vectors
- Some fault detection capability for major errors (power, illegal instruction, hung processor)
- Available in at least 25 different versions
  - Different pin counts and packaging
  - Different amounts/ types of memory
    - RAM size (192 to 1.25K bytes)
    - ROM size (4K to 32K bytes)
    - EEPROM (512 to 2K bytes)
    - ROM, EPROM, or EEPROM program memory
      
  *Memory maps vary from version to version!*
  - Different I/ O capabilities (number of timers, chip selects, DMA channels, A/ D types, etc.)

**Modes of operation**

2 operating modes and 2 test modes

**Operating modes**

- **Single chip**
  - No external address and data bus functions
- CPU can only access on-chip memory
  - Ports B and C are general purpose parallel I/O
  - All software needed to control MCU must be in internal memory
  - On reset, execution begins at address $E000
    - Located in ROM

- **Expanded multiplexed**
  - External memory and peripheral devices can be accessed by time-multiplexed address-data bus
  - Port B used for high byte of address (output)
  - Port C provides low byte of address (output) and 8-bit data (bi-directional)
  - External address latch is required
  - Execution begins at address $E000
Test modes

- **Special bootstrap**
  - On power up or reset, the program in the bootstrap ROM is executed
  - CPU waits for a 256-byte program segment to be downloaded through the serial link and stored starting at address $0000$
  - Execution then begins at address $0000$
  - Permits wide variety of programs to be downloaded

- **Special test**
  - Primarily a testing mode for the manufacturer
  - Overrides some automatic protection mechanisms -- risky!

I/O Ports

- **Port A (8 bits)**
  - Also used for timer

- **Port D (6 bits)**
  - Also used for asynchronous (SCI) and synchronous serial (SPI) I/O

- **Port E (8 bits)**
  - Also used for A/D converter

- **Port B (8 bits)**
  - Also used as address in expanded mode

- **Port C (8 bits)**
  - Also used as data/address in expanded mode

MC68HC11 detailed block diagram

*Yield applies only to devices with EPROM/PROM*
Programming languages

- **Application programs**
  - User “runs” an existing software package, Limited flexibility outside intended application
  - Examples: Excel, Netscape, Word, . . .

- **High-level language (HLL)**
  - User writes programs to perform task(s)
  - Very flexible, easy to use (once language is learned!)
  - Examples: C, C++, Java, Fortran

- **Assembly language**
  - Harder to use than HLLs
  - Machine and configuration dependent
  - Requires detailed knowledge of the μP itself and its instruction set
  - Still used where extremely high performance, short programs are required

- **Machine code**
  - Native language of the processor itself
  - Programs are the actual bytes as stored in memory
  - Not intended for human consumption

Programming procedure

![Diagram showing file names and their relationships]

What features of the processor are most important to the assembly-language programmer?

- Register set
- Memory organization
- Instruction set
- Addressing modes
Register set

```
15 0
14
13
12
11
10
9
8
7
6
5
4
3
2
1
0
```

Memory Organization

- On chip memory
  - ROM (12K bytes)
    - Factory programmed, Special bootstrap ROM
  - RAM (512 bytes)
    - Data, stack
    - Can be used for downloaded code
    - Low-power standby mode
  - EEPROM (512 bytes)
    - Programmed and erased on-chip
    - Calibration storage, diagnostic log, critical data logging, security data
    - Can also be used for downloaded code

- Off-chip memory
  - EPROM
    - For prototype development
    - Windowed and one-time programmable versions
Instruction set

- The instruction set specifies the kinds of data transfers and transformations that can occur in the machine.
- Instructions can be grouped into 5 broad categories:
  - **Data transfers**: instructions that move data to and between registers.
  - **Logical**: instructions that perform logic operations on data -- AND, OR, etc.
  - **Arithmetic**: addition, subtraction, increment, etc.
  - **Flow control**: instructions that change the sequence of execution of a program -- conditional and unconditional branches, stack operations, etc.
  - **Input / Output** operations.
- An instruction generally consists of an opcode and some operand(s).
- HC11 instructions are of different lengths (1-5 bytes).

Opcodes

1-byte opcodes
- Most opcodes use just 1 byte.
- Downward compatible with 6800.

2-byte opcodes
- Most 2-byte instructions deal with register Y, which was not present in the 6800 processor.
- It takes longer to fetch and execute 2-byte opcodes.
- New instructions use a "pre-byte", which signals that another opcode byte follows:
  - $18, $1A, or $CD.
- **Ex:**
  - INX $08
  - INY $18 $08

Instruction format

- An instruction is made up of an opcode and a set of operands.
  - Opcode may be one or two bytes.
  - Instructions may use 0, 1, 2, or 3 operands.
- Operands may be 1 or 2 bytes.
- Instruction lengths range from 1 to 5 bytes.
- Example (assume this is stored at address $E000):
  - LDAA #$ FF; load ACCA with the value $FF.
Machine code:
$E000 $86
$E001 $FF

Fetch/execute operation

Addressing modes

An instruction specifies the location of the data that it is going to process by using one of the existing addressing modes. M68HC11 has the following six different addressing modes, which are discussed in detail in Chapter 2:

- Inherent addressing
- Immediate addressing
- Extended addressing
- Direct addressing
- Indexed addressing
- Relative addressing