Outline - Microprocessors

- General Concepts
  - Memory
  - Bus Structure
  - Central Processing Unit
    - Registers
    - Instruction Set
  - Clock
- Architecture
  - Von Neuman vs. Harvard
  - CISC vs. RISC

General Concepts - Computer

- Hardware
  - Microprocessor: Main information processor
    - Central Processing Unit (CPU)
  - Memory: Place to store programs and data
    - RAM and ROM
  - I/O Devices: Peripheral units
    - Graphics processor(s)
    - Audio/sound processor(s)
    - Keyboard / mouse interfaces
    - Serial communication interfaces (e.g. USB controller)
    - Bus drivers, timers, etc.
- Software: program
  - A set of instructions for the microprocessor to implement a certain task.
    - For instance, operating system program, application programs

General Organization

- Processor
  - Control Unit
  - Datapath
    - Arithmetic Logic Unit
    - Registers
  - Common Bus (address, data, & control)
  - Memory
    - Program Data Storage
  - Output Units
  - Input Units

Mini PC Mainboard (Mini-ITX)

- Microprocessor
- Slots for Memory Cards
- “Ports” for I/O Devices
Bus Structure

- Parallel electrical pathways connecting various computer components:
  - **Address bus**: specify address locations where data and instructions reside in memory
  - **Data bus**: carries instructions and data to and from memory
  - **Control bus**: sends and receives control commands among system components

Random Access Device

- **Address Bus**: m
- **Data Bus**: n
- **Control Lines**: p

Any data residing in such devices can be addressed and accessed at will.

- **Length**: number of separately addressable memory locations: $2^m$
- **Width**: n bits of data lines
- **Size (Capacity)**: Length × Width = $2^m$ of n-bit numbers

Linear Memory Model

- **Computer memory** is frequently visualized as a large collection of mailboxes:
  - Mail → Content
  - Box location → Address

- Memory is divided into 8-bit (1 byte) cells with unique addresses.

Memory

- **Two main categories**:
  - **Random Access**
    - Random Access Memory (RAM)
      - SRAM, DRAM
    - Read-Only Memory (ROM)
      - MROM, PROM, EPROM, EEPROM, Flash
  - **Sequential Access**
    - Disk drives, CDROMs, DVDs, etc.

- Devices can be classified also by their physical structures or operating principles:
  - Solid-state, Magnetic, Optical, etc.
Random Access Memory (RAM)

- Can read / write any location in the address space,
- **Volatile**: device loses memory contents when power-off,
- Faster access time,
- Applications: Storage of temporary data
  - Variables and temporary data generated

Variants of RAM

- **Static RAM (SRAM)**
  - More transistors to hold one “bit” of information.
  - Very fast and reliable.
  - Suitable for cache memory of computers.
- **Dynamic RAM (DRAM)**
  - Small “die” area if compared to SRAM,
  - Suitable for microfabrication of high capacity RAM.
  - Need to refresh memory periodically to avoid losing contents
    - every a few milliseconds to over a hundred ms.
  - Slower and less reliable if compared to SRAM.
  - Used as main memory of computer systems.

Read-Only Memory (ROM)

- Can only read the contents.
- **Non-volatile**
- Slower if compared to RAM devices
- Applications: storage of permanent data
  - Programs / firmwares
  - Constants
  - Look-up tables
  - Vector tables etc.

Variants of ROM

- **MROM**: Mask-programmed ROM
  - Programmed when being manufactured.
- **PROM**: Programmable ROM
  - One-time programmable ROM using PROM programmers.
- **EPROM**: Erasable PROM
  - Users can erase the ROM (in bulk) via strong ultra-violet light.
Variants of ROM (Cont’d)

- EEPROM: Electrically Erasable PROM
  - Also called E²PROM.
  - Erased and reprogrammed by electrical signals.
  - Unlike EPROM, individual locations of the device can be erased and reprogrammed.

- Flash Memory:
  - Takes advantages of EPROM and EEPROM technologies.
  - Efficient in terms of fabrication.
  - Unlike EEPROM, chunks of data can be manipulated one at a time.

Microprocessor

- A processor implemented on a single IC.
  - Classified by number of bits that a microprocessor can manipulate in one operation.
    - 4-bit (i4004), 8-bit, 16-bit, 32-bit, 64-bit (Pentium 4).
  - Peripheral chips are needed to construct a “computer.”
  - A microcomputer is a computer using a microprocessor as its CPU (such as today’s desktop).

- Limitations:
  - Requires external memory to execute programs;
  - Peripheral chips are needed to interface with I/O devices;
  - Glue Logic (decoders, buffers) is needed to interconnect external memory and peripheral interface chips with the microprocessor.

Central Processing Unit

- CPU is the key component of a microprocessor, containing the digital circuitry necessary to interpret and execute program instructions.
- Includes a number of units
  - Arithmetic Logic Unit
    - Used to perform arithmetic and logic operations
  - Control Unit
    - Instruction decoding and controlling relevant processor units.
  - Register File

Registers

- A register is a special storage location inside the CPU:
  - holds data or a memory address during the execution of an instruction.
  - number of registers varies from one microprocessor to another.
  - operations on the registers are extremely fast.
- CPU Registers
  - solely perform general-purpose operations
    - arithmetic, logic, flow control
  - do not occupy (outside) memory space.
- Control Registers (mostly used by microcontrollers)
  - Configures the peripheral functions
Instruction Set

• Instruction set of a generic microprocessor can be divided into 6 basic categories:
  – Data transfer and manipulation: Load, store, transfer, exchange, move.
    • Memory ↔ Memory
    • Register ↔ Memory
    • Register ↔ Register
  – Arithmetic operations: +, -, *, and /.
  – Logic and bit operations: AND, OR, XOR, bit shifting, etc.
  – Data test and compare
  – Branch (“Go to” or “Jump”) instructions
  – Subprogram call instructions

Programs and Execution Cycles

• A program consists of a number of instructions telling the processor how to carry out a specific task.
• Program resides in memory:
  – Instructions must be represented as binary numbers (called “machine code”) to be decoded by the microprocessor.
• A generic processor goes through three cycles:
  – fetches instructions (one-by-one) from the memory
  – decodes (“interprets”) a particular instruction
  – executes it. If applicable, processor does
    • Read the relevant data from memory,
    • Carry out the operation on the data,
    • Store the result back in the memory.

Clock

• Every operation inside the computer system must be perfectly synchronized.
• A precision oscillator (called “clock”) generating a square waveform (“clock ticks”) is utilized to time the operations of the microprocessor along with its peripheral units.
• The execution speed of processor is loosely related to the clock frequency:
  – Depending on the computer architecture, it may take one to several clock periods to execute a particular instruction.

Von Neuman Architecture

• Von Neuman architecture is used in most microprocessors like Intel Pentium processors.
• Known also as Princeton architecture.
• There is a single bus system (address-, data-, and control bus) between CPU and memory:
  – RAM and program memory share the same bus
  – Bottleneck: Getting instructions interferes with accessing RAM
Harvard Architecture

- Harvard architecture is employed in some CPUs such as PowerPC (PPC) processors and PIC microcontrollers:
  - Separate program- and memory bus systems are utilized.
  - Attributes of these bus systems might be entirely different.
  - Architecture enables very fast execution of a program.

- Bottleneck: Hardware design is quite complicated.

CISC versus RISC

- Traditionally, most CPUs are Complex Instruction Set Computer (CISC):
  - Used in: i80x86, 8051, 68HC12, etc.
  - Many instructions (usually > 100)
  - Many addressing modes
  - Usually takes more than 1 internal clock cycle to execute.

CISC versus RISC (Cont’d)

- Most microprocessor with Harvard architecture are Reduced Instruction Set Computer (RISC):
  - Used in: SPARC, ALPHA, Atmel AVR processors.
  - Few instructions (usually < 50)
  - Limited addressing modes
  - Executes 1 instruction within 1 internal clock cycle.

Embedded Systems

- A product that uses one or more microcontrollers as controller(s). Also called embedded products.
- Users (consumers) are interested in the functionality of the product, not the microcontroller itself.
- Cellular phones, security systems, home appliances, and modern automobiles are examples of embedded products.