Chapter 1

Introduction

In this Chapter

- Digital systems
- Number systems
- Digital circuits
- Computer organization
- Computer architecture
- Computer security
  - Security through hardware
- Other chapters

Digital Systems

- Computers, iPad, cell phones, digital cameras, etc. created digital revolution, changing ways we:
  - Communicate, work, are entertained, shop
- Digital systems are in everything we see and use
  - Cars, grocery checkout equipment, utility meters, set-top, boxes, emergency equipment, etc.
- Thus, more data is created, processed, stored, transmitted, and accessed
  - Results in demands for more powerful computers
    - Personal computers
    - Large computers used in
      - E-commerce, banking, search engines, research
  - Create more chances for unauthorized access to data and information

Impact of Technologies (1)

1. Impact of innovations in integrated chip (IC) technologies
   - Moore's law:
     - 35% increase in transistor density per year
     - 40% to 50% increase in transistor count per year
     - Has been used as a guide to design each next generation of microprocessors that revolutionized personal computers
     - Also resulted in increased power use and heat dissipation

Impact of Technologies (2)

2. Impact of innovations in application developments
   - Revolutionizing the way digital systems are designed
   - Digital circuits are described in HDL
   - CAD tools simulate (validate) HDL descriptions
   - CAD tools synthesize (translate) HDL descriptions to circuits

Number Systems

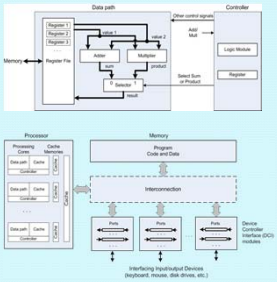
- Digital circuits make logical decisions as True or False logic values
- A voltage range defines each logic value.
  - E.g., 5-volt power source
    - 2.4 to 5V as True
    - 0 to 0.8V as False
  - Lower voltage sources help save power in battery powered devices
- True and False values as 1 and 0 form binary numbers to represent
  - Characters
    - 8-bit, or 256 ASCII codes
    - 16-bit numbers, or over 65,000 Unicodes
  - Pixels to create images
  - Audio and video data
  - Integer and real numbers used in computations

Digital Systems as Von Neumann machines

- Computer system that contains
  - One or processors
    - Each consisting of one or more *processing cores* (CPUs)
  - Memory
  - I/O devices
  - OS and application programs
- Embedded system that is
  - A complete system as circuit board or ASIC or FPGA , known as SoC
    - Contain CPU(s) and memory
    - Dedicated software known as firmware
  - May include transmitter/receiver modules
  - May include signal conversion modules (converters)
    - Analog-to-Digital (A/D)
    - Digital-to-Analog (D/A)
  - Applications:
    - Cell phones, digital camcorder, etc.
    - Host device controller interface
      - E.g., USB

Von Neumann Computer

- Processing core (CPU) consists of
  - Data path that includes
    - Digital circuit modules perform computations
    - Storage modules store computed data
  - Controller that orders data path operations
- As microcomputer that includes
  - Multicore processor (s)
  - Memory
  - Interconnection medium
  - I/O devices
  - I/O device controller and interface
  - Potential for bottleneck between faster processor and slower memory



Digital Logic Design (1)

1. Requires logic expressions
   - Example: $f = ((\text{NOT } a) \text{ AND } b) \text{ OR } c$
     - NOT, AND, and OR indicate Boolean logic operators
   - Evaluation
     - Suppose a = 0, b = 1, and c = 0

```
f  = (NOT 0) AND 1) OR 0
   = (1 AND 1) OR 0
   = 1 OR 0
   = 1
```

What function does f perform?

f is 1 when a, b, c forming a 3-bit number is prime

   - Requirement: Minimal expressions to reduce hardware

Digital Logic Design (2)

2. Requires logic gates
   - Gates perform logic operations
   - Modern gates are built as CMOS circuits
     - Complementing MOS transistors reduce power consumption and heat dissipation
   - CMOS chips can be fan-cooled when hot
     - Thus, enabled personal computers
     - Exclusively used today in all types of digital systems

Digital Logic Design (3)

3. Require logic circuits
   - Logic circuits implement logic expressions
   - All NAND or all NOR gates show implementation details
   - Two types of digital circuits:
     1. Combinational: Outputs are generated concurrently
        - Outputs depend on current inputs only
        - E.g., Adders and selectors
     2. Sequential: Outputs are generated in sequence (in steps)
        - Output depends on current inputs as well as previous inputs
        - Uses combinational circuits to generate outputs
        - E.g., registers, counters, and control units

Effect of Increased Power Consumption and Heat Dissipation

- Examples,
  - 2 watts for Intel 80386 processor
  - 130 watts for 3.3 GHz (Giga Hertz) Intel Core i7 processor
    - 65 times more watts
- Problem: Harder to make processors any faster
  - Affects computer organization, programming model, and OS
  - Current Solutions:
    - Divide tasks into subtasks using multithreaded programming
    - OS assigns processing cores to perform subtasks
      - Creates thread-level parallelism
    - Use multiprocessor systems to perform many independent and dependent tasks faster
      - Modern Supercomputers for scientific computations
      - Warehouse-scale computers for
        - Interactive applications (Facebook, Google, etc.)
        - Large-scale storage and computing (e.g., cloud computing)

Computer Organization

Specifies implementation details:

- Circuit and their physical relationship that makeup
  - Processing core
    - data path organization
    - Example: 32-bit Intel vs. AMD processors
      - Two different data paths but same instruction set
  - processor,
  - memory,
  - I/O device controller and interface
  - Interconnection of a computer components
- Memory organization
  - Cache, SDRAM, multi-channel, etc.

Computer Architecture (1)

Specifies design concepts:

- Pipelining
  - Concept of an assembly line

Stage 3: Install wheels			Car1	Car2	Car3	. . .
Stage 2: Install doors		Car1	Car2	Car3	. . .	. . .
Stage 1: Install Engine	Car1	Car2	Car3	. . .	. . .	. . .
Time slots (e.g., 10-minutes slots)	1	2	3	4	5	. . .

If each stage is 2 minutes, how many cars built in a year (assume perfect scenario)?

260,000+

  - E.G., pipelining CPU data path

Execute			Load r1, B	Load r2, C	Add r3, r1, r2	Store A, r3
Decode		Load r1, B	Load r2, C	Add r3, r1, r2	Store A, r3	. . .
Fetch	Load r1, B	Load r2, C	Add r3, r1, r2	Store A, r3	. . .	. . .
Time (T)	1	2	3	4	5	6

What can go wrong?

Computer Architecture (2)

- Parallelism
  - Single Instruction Multiple Data (SIMD)
    - E.g., Intel's Streaming SIMD Extension (SSE) instruction set
    - E.g., AMD's 3DNow instruction set
    - Also in GPUs
  - Instruction level parallelism (ILP)
    - Also called Superscalar processor (i.e., processing core)
    - E.g., processing cores in Intel Core i7
  - Multiple Instructions Multiple Data (MIMD)
    - Multicore Processors
      - E.g., Intel Core i7
    - Multiprocessor Systems
      1. *Shared memory*: Processors communicate using memory
      2. *Message passing*: Processors communicate by sending/receiving messages

Computer Security

- Protecting digital assets (programs and data) from malware attacks
- Protecting digital assets from unauthorized access by employees
- Affects all
  - individuals, government, business organizations
- Potable devices also subject to *physical attacks*
- Application examples
  - Secure data storage
  - Secure communication
  - Secure e-commerce
  - Etc.
- How hardware can help?
  - Hardware more secure than software and disk storage
  - Computer security through hardware
    - Secure co-processor
      - E.g., Crypto-processor
    - Secure processor
      - Also, supports secure execution
      - Also, guards against physical attacks

Remaining Chapters (1)

- Combinational circuits
  - Design methodology for small circuits (Ch2)
    - Circuits with fewer (e.g., ≤ 4) inputs
  - Design methodology for large circuits (Ch3)
    - Circuits with many inputs (e.g., 32-bit Adder)
- Sequential circuits
  - Basic core modules (Ch4)
    - Basic storage elements
  - Design methodology for small circuits (Ch5)
    - Registers, counters, etc.
  - Design methodology for large circuits (Ch6)
    - Data paths and control units

Remaining Chapters (2)

- Memory (Ch7)
  - Memory organization
  - Memory timing
- Processing core (CPU) design, a very complex sequential circuit (Ch8)
  - CPU data path and control
- Microcomputer organization, history and modern designs (Ch9)
  - CPU, memory, I/O device interconnections
  - Device communication
- Memory system (Ch10)
  - Cache memory organization
  - Main memory as physical memory
  - Disk space as virtual memory
- Computer security for computer architects (Ch11), an introduction
  - Threat models
  - HW and SW security models, policies, and mechanisms
  - Trusted computing base as secure co-processor or secure processor