Digital Logic Design 1
Tran Ngoc Thinh
HCMC University of Technology
http://www.cse.hcmut.edu.vn/~tnthinh

Administrative Issues

• Class
  – Time and venue: Tuesdays, 8:15 am - 11:35 am, 403A4
  – Web page:
    • http://www.cse.hcmut.edu.vn/~tnthinh/DS1
  – Textbook:

• Grades
  – 20% assignments / quizzes
  – 10% presentation
  – 20% midterm
  – 50% final exam

• Personnel
  – Instructor: Dr. Tran Ngoc Thinh
  • Email: tnthinh@cse.hcmut.edu.vn
  • Phone: 8647256 (5843)
  • Office: A5 building, CE Lab
  • Office hours: Thursdays, 08:30-11:00

What is This Course All About?

• What is covered?
  – This course provides fundamentals of logic design, such as: number presentation and codes, Boolean algebra and logic gates, analysis and design of combinational and sequential circuits.

• Learning outcomes
  – Knowledge: Number presentation and codes, Boolean algebra and logic gates.
  – Skill: Design and Analyze combinational circuits and sequential circuits.

• Next slides:
  – Outline of this course

Overview of the course

➢ Number presentation and codes
➢ Boolean algebra and logic gates
➢ Combinational circuits
➢ Sequential circuits
Course Outline – Part I

• Number system and codes
  – Decimal, Binary, Octal, Hexadecimal Number Systems
  – Conversions
  – Codes: Gray, Alphanumeric Codes
  – Parity Method for Error Detection

• Logic gates and Boolean Algebra
  – Boolean Constants and Variables
  – Truth Tables
  – Basic gates: OR AND NOT Operation with OR Gates
  – NOR Gates and NAND Gates
  – Boolean Theorems
  – DeMorgan’s, DeMorgan’s Theorems

Course Outline – Part II

• Combinational Logic Circuits
  – Sum-of-Product Form
  – Simplifying Logic Circuits
  – Algebraic Simplification
  – Designing Combinational Logic Circuits
  – Karnaugh Map Method
  – Parity Generator and Checker
  – Enable/Disable Circuits
  – Basic Characteristics of Digital ICs
  – Troubleshooting Digital Systems

Course Outline – Part II

• Flip-Flops and Related Devices
  – Latches, D Latch
  – Clock Signals and Clocked Flip-Flops
  – S-C, J-K, D Master/Slave Flip-Flops
  – Flip-Flop Application
    • Detecting an Input Sequence
    • Data Storage and Transfer
    • Serial Data Transfer: Shift Registers
    • Frequency Division and Counting
    • Microcomputer Application
  – Schmitt-Trigger, On-shot Devices
  – Analyzing Sequential & Clock Generator Circuits
  – Troubleshooting Flip-Flop Circuits

Introduction to Chapter 1

• Digital technology is widely used. Examples:
  – Computers
  – Manufacturing systems
  – Medical Science
  – Transportation
  – Entertainment
  – Telecommunications

• Basic digital concepts and terminology are introduced
Numerical Representations

- Analog Representation
  - A continuously variable, proportional indicator.
  - Examples of analog representation:
    - Sound through a microphone causes voltage changes.
    - Mercury thermometer varies over a range of values with temperature.

- Digital Representation
  - Varies in discrete (separate) steps.
  - Examples of digital representation:
    - Passing time is shown as a change in the display on a digital clock at one minute intervals.

Digital and Analog Systems

- Digital system
  - A combination of devices that manipulate values represented in digital form.

- Analog system
  - A combination of devices that manipulate values represented in analog form.

Digital and Analog Systems

- Advantages of digital
  - Ease of design
  - Well suited for storing information.
  - Accuracy and precision are easier to maintain
  - Programmable operation
  - Less affected by noise
  - Ease of fabrication on IC chips

Digital and Analog Systems

- There are limits to digital techniques:
  - The world is analog
  - The analog nature of the world requires a time consuming conversion process:
    1. Convert the physical variable to an electrical signal (analog).
    2. Convert the analog signal to digital form.
    3. Process (operate on) the digital information
    4. Convert the digital output back to real-world analog form.

Digital and Analog Systems

- Analog-to-digital conversion (ADC) and digital-to-analog conversion (DAC) complicate circuitry.

Digital and Analog Systems

- The audio CD is a typical hybrid (combination) system.
  - Analog sound is converted into analog voltage.
  - Analog voltage is changed into digital through an ADC in the recorder.
  - Digital information is stored on the CD.
  - At playback the digital information is changed into analog by a DAC in the CD player.
  - The analog voltage is amplified and used to drive a speaker that produces the original analog sound.
Digital Number Systems

Number systems differ in the amount of symbols they use
- Decimal – 10 symbols (base 10)
- Hexadecimal – 16 symbols (base 16)
- Octal – 8 symbols (base 8)
- Binary – 2 symbols (base 2)

The Decimal (base 10) System
- 10 symbols: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
- Each number is a digit (from Latin for finger)
- Most significant digit (MSD) and least significant digit (LSD)
- Positional value may be stated as a digit multiplied by a power of 10

The Binary (base 2) System
- 2 symbols: 0, 1
- Lends itself to electronic circuit design since only two different voltage levels are required.
- Other number systems are used to represent binary quantities.
- Positional value may be stated as a digit multiplied by a power of 2.

Binary Counting

Other two state devices:
- Light bulb (off or on)
- Diode (conducting or not conducting)
- Relay (energized or not energized)
- Transistor (cutoff or saturation)
- Photocell (illuminated or dark)
Representing Binary Quantities

- Exact voltage level is not important in digital systems.
- A voltage of 3.6 V will mean the same (binary 1) as a voltage of 4.3 V.

Digital Circuits/Logic Circuits

- Digital circuits - produce and respond to predefined voltage ranges.
- Logic circuits – used interchangeably with the term, digital circuits.
- Digital integrated circuits (ICs) – provide logic operations in a small reliable package.

Parallel and Serial Transmission

- Parallel transmission – all bits in a binary number are transmitted simultaneously. A separate line is required for each bit.
- Serial transmission – each bit in a binary number is transmitted per some time interval.

Parallel and Serial Transmission

- Parallel transmission is faster but requires more paths.
- Serial is slower but requires a single path.
- Both methods have useful applications which will be seen in later chapters.

Memory

- A circuit which retains a response to a momentary input is displaying memory.
- Memory is important because it provides a way to store binary numbers temporarily or permanently.
- Memory elements include:
  - Magnetic
  - Optical
  - Electronic latching circuits

Digital Computers

- Computer – a system of hardware that performs arithmetic operations, manipulates data (usually in binary form), and makes decisions.
- Computers perform operations based on instructions in the form of a program at high speed and with a high degree of accuracy.
Digital Computers

- Major parts of a computer
  - Input unit – processes instructions and data into the memory.
  - Memory unit – stores data and instructions.
  - Control unit – interprets instructions and sends appropriate signals to other units as instructed.
  - Arithmetic/logic unit – arithmetic calculations and logical decisions are performed.
  - Output unit – presents information from the memory to the operator or process.
  - The control and arithmetic/logic units are often treated as one and called the central processing unit (CPU)

- Types of computers
  - Microcomputer
    - Most common (desktop PCs)
    - Has become very powerful
  - Minicomputer (workstation)
  - Mainframe
  - Microcontroller
    - Designed for a specific application
    - Dedicated or embedded controllers
    - Used in appliances, manufacturing processes, auto ignition systems, ABS systems, and many other applications.

Conversion

- The hexadecimal number system is introduced.
- Since different number systems may be used in a system, it is important for a technician to understand how to convert between them.
- Binary codes that are used to represent different information are also described.

Binary to Decimal Conversion

- Convert binary to decimal by summing the positions that contain a 1.

\[ 1 \ 0 \ 0 \ 1 \ 0 \ 2^2 \]

\[ 2^5 + 2^4 + 2^3 + 2^2 + 2^1 + 2^0 = \]

\[ 32 + 0 + 0 + 4 + 0 + 1 = 37_{10} \]

Decimal to Binary Conversion

- Two methods to convert decimal to binary:
  - Reverse process described in 2-1
  - Use repeated division

\[ 37_{10} = 2^4 + 0 + 0 + 2^2 + 0 + 2^0 \]

\[ 1 \ 0 \ 0 \ 1 \ 0 \ 1 \ 2 \]
Decimal to Binary Conversion

- Repeated division steps:
  - Divide the decimal number by 2
  - Write the remainder after each division until a quotient of zero is obtained.
  - The first remainder is the LSB and the last is the MSB

Hexadecimal Number System

- Most digital systems deal with groups of bits in even powers of 2 such as 8, 16, 32, and 64 bits.
- Hexadecimal uses groups of 4 bits.
- Base 16
  - 16 possible symbols
  - 0-9 and A-F
- Allows for convenient handling of long binary strings.

Hexadecimal Number System

- Convert from hex to decimal by multiplying each hex digit by its positional weight.
  Example: $163_{16}$

Hexadecimal Number System

- Convert from decimal to hex by using the repeated division method used for decimal to binary and decimal to octal conversion.
  - Divide the decimal number by 16
  - The first remainder is the LSB and the last is the MSB.
  - Note, when done on a calculator a decimal remainder can be multiplied by 16 to get the result. If the remainder is greater than 9, the letters A through F are used.

Example of hex to binary conversion:

$9F_{16} = 9 \ F \ 2$

$1001 \ 1111 \ 0010 = 10011110010_{2}$

Electronic Design Automation
**Binary to Hex Conversion**
- Convert from binary to hex by grouping bits in four starting with the LSB.
- Each group is then converted to the hex equivalent.
- Leading zeros can be added to the left of the MSB to fill out the last group.
- Example:
  (Note the addition of leading zeroes)

\[
1110100110_2 = 0011 \ 1010\ 0110
\]

**Hexadecimal Number System**
- Hexadecimal is useful for representing long strings of bits.
- Understanding the conversion process and memorizing the 4 bit patterns for each hexadecimal digit will prove valuable later.

**BCD**
- Binary Coded Decimal (BCD) is another way to present decimal numbers in binary form.
- BCD is widely used and combines features of both decimal and binary systems.
- Each digit is converted to a binary equivalent.

- To convert the number \(874_{10}\) to BCD:

\[
\begin{align*}
8 & = 0100 \\
7 & = 0111 \\
4 & = 0100
\end{align*}
\]

\[0100\ 0111\ 0100 = 010001110100_{BCD}\]

- Each decimal digit is represented using 4 bits.
- Each 4-bit group can never be greater than 9.
- Reverse the process to convert BCD to decimal.

**BCD**
- BCD is not a number system.
- BCD is a decimal number with each digit encoded to its binary equivalent.
- A BCD number is not the same as a straight binary number.
- The primary advantage of BCD is the relative ease of converting to and from decimal.

**Gray Code**
- The gray code is used in applications where numbers change rapidly.
- In the gray code, only one bit changes from each value to the next.

<table>
<thead>
<tr>
<th>Binary</th>
<th>Gray Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>000</td>
</tr>
<tr>
<td>001</td>
<td>001</td>
</tr>
<tr>
<td>010</td>
<td>011</td>
</tr>
<tr>
<td>011</td>
<td>010</td>
</tr>
<tr>
<td>100</td>
<td>110</td>
</tr>
<tr>
<td>101</td>
<td>111</td>
</tr>
<tr>
<td>110</td>
<td>101</td>
</tr>
<tr>
<td>111</td>
<td>100</td>
</tr>
</tbody>
</table>
### Putting It All Together

<table>
<thead>
<tr>
<th>Decimal</th>
<th>Binary</th>
<th>Hexadecimal</th>
<th>BCD</th>
<th>Gray</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0000</td>
<td>0</td>
<td>0</td>
<td>0011</td>
</tr>
<tr>
<td>1</td>
<td>0001</td>
<td>1</td>
<td>0001</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0010</td>
<td>2</td>
<td>0003</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0011</td>
<td>3</td>
<td>0006</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0100</td>
<td>4</td>
<td>0100</td>
<td>0110</td>
</tr>
<tr>
<td>5</td>
<td>0101</td>
<td>5</td>
<td>0101</td>
<td>0111</td>
</tr>
<tr>
<td>6</td>
<td>0110</td>
<td>6</td>
<td>0110</td>
<td>0101</td>
</tr>
<tr>
<td>7</td>
<td>0111</td>
<td>7</td>
<td>0111</td>
<td>0100</td>
</tr>
<tr>
<td>8</td>
<td>1000</td>
<td>8</td>
<td>1000</td>
<td>0100</td>
</tr>
<tr>
<td>9</td>
<td>1001</td>
<td>9</td>
<td>1001</td>
<td>0101</td>
</tr>
<tr>
<td>10</td>
<td>1010</td>
<td>A</td>
<td>0001</td>
<td>0000</td>
</tr>
<tr>
<td>11</td>
<td>1011</td>
<td>B</td>
<td>0001</td>
<td>0001</td>
</tr>
<tr>
<td>12</td>
<td>1100</td>
<td>C</td>
<td>0001</td>
<td>0010</td>
</tr>
<tr>
<td>13</td>
<td>1101</td>
<td>D</td>
<td>0001</td>
<td>0011</td>
</tr>
<tr>
<td>14</td>
<td>1110</td>
<td>E</td>
<td>0001</td>
<td>0100</td>
</tr>
<tr>
<td>15</td>
<td>1111</td>
<td>F</td>
<td>0001</td>
<td>0101</td>
</tr>
</tbody>
</table>

### The Byte, Nibble, and Word

- 1 byte = 8 bits
- 1 nibble = 4 bits
- 1 word = size depends on data pathway size.
   - Word size in a simple system may be one byte (8 bits)
   - Word size in a PC is eight bytes (64 bits)

### Alphanumeric Codes

- Represents characters and functions found on a computer keyboard.
  - Seven bit code: \(2^7 = 128\) possible code groups
  - Examples of use are: to transfer information between computers, between computers and printers, and for internal storage.

### Parity Method for Error Detection

- Binary data and codes are frequently moved between locations. For example:
  - Digitized voice over a microwave link.
  - Storage and retrieval of data from magnetic and optical disks.
  - Communication between computer systems over telephone lines using a modem.
- Electrical noise can cause errors during transmission.
- Many digital systems employ methods for error detection (and sometimes correction).

- The parity method of error detection requires the addition of an extra bit to a code group.
  - **Even parity method** – the total number of bits in a group including the parity bit must add up to an even number.
    - The binary group 1 0 1 1 would require the addition of a parity bit 1 1 0 1 1

### Parity Method for Error Detection

- This extra bit is called the **parity bit**.
- The bit can be either a 0 or 1, depending on the number of 1s in the code group.
- There are two methods, even and odd.
Parity Method for Error Detection

- **Odd parity method** – the total number of bits in a group including the parity bit must add up to an odd number.
  - The binary group 1 1 1 1 would require the addition of a parity bit 1 1 1 1

Parity Method for Error Detection

- The transmitter and receiver must “agree” on the type of parity checking used.
- Two bit errors would not indicate a parity error.
- Both odd and even parity methods are used, but even seems to be used more often.