Digital System

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


- Instructor: Assoc. Prof. Dr. Tran Ngoc Thinh
  - Email: tnthinh@cse.hcmut.edu.vn
  - Phone: 38647256 (5843)
  - Office: A3 building, CE Department
  - Office hours: Mondays, 09:00-11:00

Administrative Issues

- Class
  - Time and venue: Fridays, 15:05 - 17:30, 407A4
  - Web page:
    - http://www.cse.hcmut.edu.vn/~tnthinh/DS1
  - Textbook:
- **[1] "*Digital Systems* - 8th *Edition*" - Ronald J. Tocci, Prentice-Hall 2001**
- [2] "*Digital Logic Design Principles*"– N. Balabanian, B. Carlson, John Wiley & Sons, Inc , 2004
- [3] "*Digital Design* -3rd *Edition*"–John F. Wakerly, Prentice-Hall 2001
- [4] "*Fundamentals of Digital Logic – 2nd edition*" – Stephen Brown, Zvonko Vranesic, McGraw Hill 2008

Administrative Issues (cont.)

- Grades
  - 20% Lab
  - 20% assignments/quizzes + presentation
  - 30% midterm
  - 30% final exam

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.

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 III

- 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

9

Course Outline – Part IV

- Operation and Circuits
  - Representing Signed Numbers
  - Addition, Subtraction in the 2's-Complement System
  - Multiplication, Division of Binary Numbers
  - BCD Addition
  - Hexadecimal Arithmetic
  - Arithmetic Circuits
    - Parallel Binary Adder
    - Design of a Full Adder
    - Carry Propagation
    - Integrated Circuit Parallel Adder
  - 2's Complement System
  - BCD Adder
  - ALU Integrated Circuits

10

Course Outline – Part V

- Counters and Registers
  - Asynchronous & Synchronous Counters
  - Up/Down Counters
  - Cascading BCD Counters
  - Synchronous Counter Design
  - Shift-Register Counters
  - Counter Application: Frequency Counter, Digital Clock
  - Integrated-Circuit Registers
  - Some ICs:
    - Parallel In/Parallel Out – The 74ALS174/HC174
    - Serial In/Serial Out – The 4731B
    - Parallel In/Serial Out – The 74ALS185/HC165
    - Serial In/Parallel Out – The 74ALS164/HC164

11

Course Outline – Part VI

- MSI Logic Circuits
  - Decoders
  - Encoders
  - Multiplexers
  - Demultiplexers

12

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

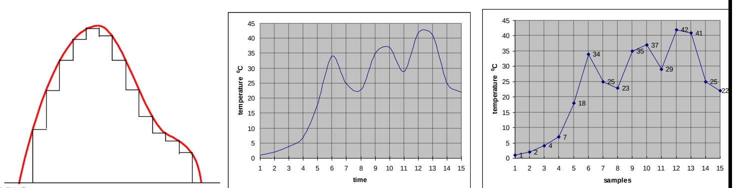
Skip

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

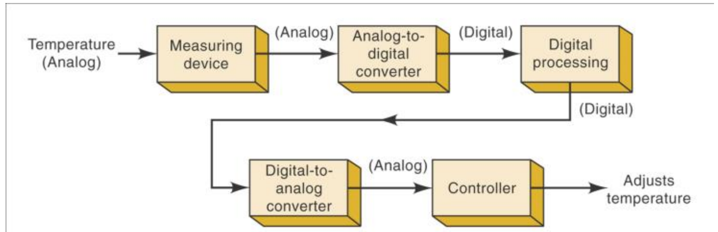
- Advantages of digital
  - Ease of design
  - Well suited for storing information.
  - Accuracy and precision are easier to maintain
  - Programmable operation
  - Less affected by noise
  - Ease of fabrication on IC chips

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 number of symbols they use
  - Decimal – 10 symbols (base 10)
  - Hexadecimal – 16 symbols (base 16)
  - Octal – 8 symbols (base 8)
  - Binary – 2 symbols (base 2)
- Generalized form of number system base *b*

$$a_{n-1}a_{n-2}\ldots a_2a_1a_0.a_{-1}a_{-2}a_{-3}\ldots a_{-m+1}a_{-m}$$

$$N_b = a_{n-1} \times b^{n-1} + a_{n-2} \times b^{n-2} + \ldots + a_0 \times b^0 + a_{-1} \times b^{-1} + \ldots + a_{-m} \times b^{-m}$$

1-3 Digital Number Systems

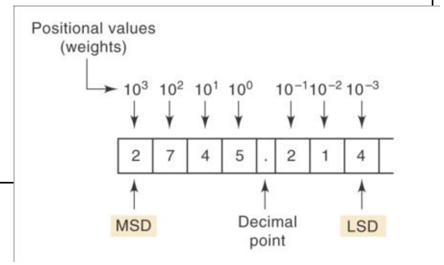
- Example
  - $(7,239)_{10} = 7 \times 10^3 + 2 \times 10^2 + 3 \times 10^1 + 9 \times 10^0$
  - $(4103.2)_5 = 4 \times 5^3 + 1 \times 5^2 + 0 \times 5^1 + 3 \times 5^0 + 2 \times 5^{-1}$
    $= 4 \times 125 + 1 \times 25 + 0 \times 5 + 3 \times 1 + 2 \times 5^{-1}$
    $= (528.4)10$
  - $(11011)_2 = 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$
    $= (27)10$
  - $(B65F)_{16} = 11 \times 16^3 + 6 \times 16^2 + 5 \times 16^1 + 15 \times 16^0$

$24.6_{(8)} = 2 \times 8^1 + 4 \times 8^0 + 6 \times 8^{-1} = 20.75_{(10)}$

Digital Number Systems

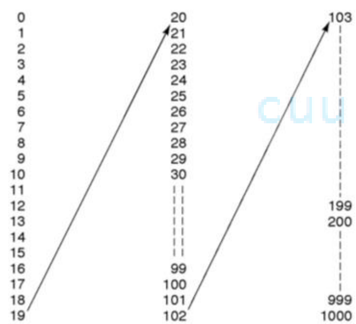
- The Decimal (base 10) System
  - 10 symbols: 0, 1, 2, 3, 4, 5, 6 , 7, 8, 9
  - Each number is a digit (from Latin for finger)
  - Most significant digit (MSD) and least significant digit (LSD)
  - Positional value may be stated as a digit multiplied by a power of 10

This image cannot currently be displayed.



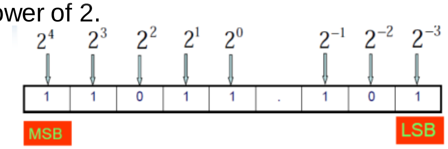
Digital Number Systems

- Decimal Counting



Digital Number Systems

- 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.



$(11011101)_2 = 1*2^0 + 1*2^1 + 0*2^2 + 1*2^3 + 1*2^4 + 1*2^{-1} + 0*2^{-2} + 1*2^{-3}$

Digital Number Systems

- Binary Counting

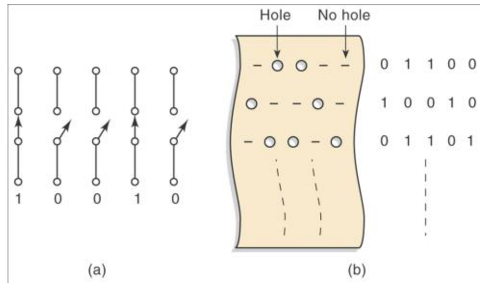
Weights → $2^3 = 8$	$2^2 = 4$	$2^1 = 2$	$2^0 = 1$	Decimal equivalent
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	10
1	0	1	1	11
1	1	0	0	12
1	1	0	1	13
1	1	1	0	14
1	1	1	1	15

LSB

25

Representing Binary Quantities

- Open and closed switches
- Paper Tape



26

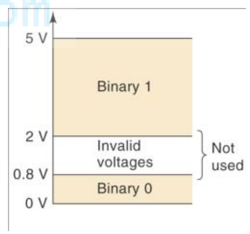
Representing Binary Quantities

- 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)

27

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.



28

Representing Binary Quantities

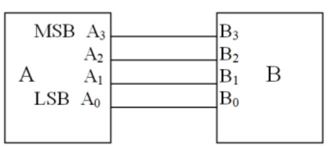
- Digital Signals and Timing Diagrams
  - Timing diagrams show voltage versus time.
  - Horizontal scale represents regular intervals of time beginning at time zero.
  - Timing diagrams are used to show how digital signals change with time.
  - Timing diagrams are used to compare two or more digital signals.
  - The oscilloscope and logic analyzer are used to produce timing diagrams.

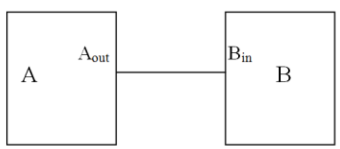
Digital Circuits/Logic Circuits

- 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

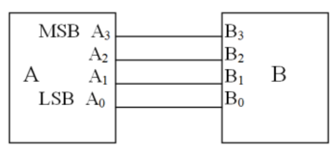
- Parallel transmission – all bits in a binary number are transmitted simultaneously. A separate line is required for each bit.
- Serial transmission – each bit in a binary number is transmitted per some time interval.

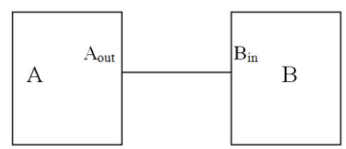




Parallel and Serial Transmission

- 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.





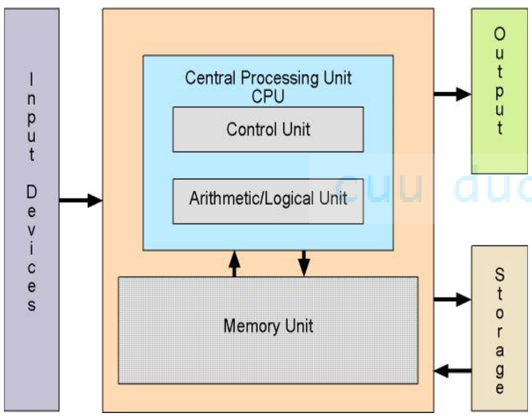
Memory

- 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.

Block diagram of digital computer



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)

Digital Computers

- 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.

$$N_{10} = (a_n a_{n-1} a_{n-2} \dots a_2 a_1 a_0)_b$$
$$= a_n \times b^n + a_{n-1} \times b^{n-1} + a_{n-2} \times b^{n-2} + \dots + a_0 \times b^0$$

$$\frac{N}{b} = a_n * b^{n-1} + a_{n-1} * b^{n-2} + a_{n-2} * b^{n-3} + \dots + a_1 = Q_1 \quad a_0$$

$$\frac{Q_1}{b} = a_n * b^{n-2} + a_{n-1} * b^{n-3} + a_{n-2} * b^{n-4} + \dots + a_2 = Q_2 \quad a_1$$

$$\frac{Q_2}{b} = a_n * b^{n-3} + a_{n-1} * b^{n-4} + a_{n-2} * b^{n-5} + \dots + a_3 = Q_3 \quad a_2$$

Binary to Decimal Conversion

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

$$1 \quad 0 \quad 0 \quad 1 \quad 0 \quad 1_2$$
$$2^5 + 2^4 + 2^3 + 2^2 + 2^1 + 2^0 =$$
$$32 + 0 + 0 + 4 + 0 + 1 = 37_{10}$$

$1011.101_2 = ?$

Decimal to Binary Conversion

- Two methods to convert decimal to binary:
  - Reverse process described above
  - Use repeated division

Decimal to Binary Conversion

- Reverse process described above
  - Note that all positions must be accounted for

$$37_{10} = 2^5 + 0 \;\; + 0 \;\; + 2^2 + 0 \;\; + 2^0$$

$$1 \quad 0 \quad 0 \quad 1 \quad 0 \quad 1_2$$

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

$$\frac{41}{2} = 20 \quad a_0 = 1 \qquad \frac{5}{2} = 2 \quad a_3 = 1$$

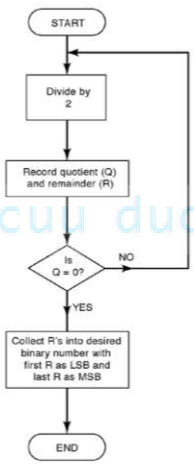
$$\frac{20}{2} = 10 \quad a_1 = 0 \qquad \frac{2}{2} = 1 \quad a_4 = 0$$

$$\frac{10}{2} = 5 \quad a_2 = 0 \qquad \frac{1}{2} = 0 \quad a_5 = 1$$

$$(41)_{10} = (a_5a_4a_3a_2a_1a_0)_2 = (101001)_2$$

Decimal to Binary Conversion

- Repeated division – describes the process and can be used to convert from decimal to any other number system.



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}

$$163_{16} = 1\times(16^2)+6\times(16^1)+3\times(16^0)$$
$$= 1\times 256+6\times 16+3\times 1$$
$$= 355_{10}$$

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.

Hexadecimal Number System

- Example of hex to binary conversion:

$9F2_{16}$ = 9 F 2

1001 1111 0010 =

100111110010_2

Các hệ đếm thông dụng			
Decimal	Binary	Octal	Hexa
0	0000	00	0
1	0001	01	1
2	0010	02	2
3	0011	03	3
4	0100	04	4
5	0101	05	5
6	0110	06	6
7	0111	07	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

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 = **00**11 1010 0110

= 3 A 6

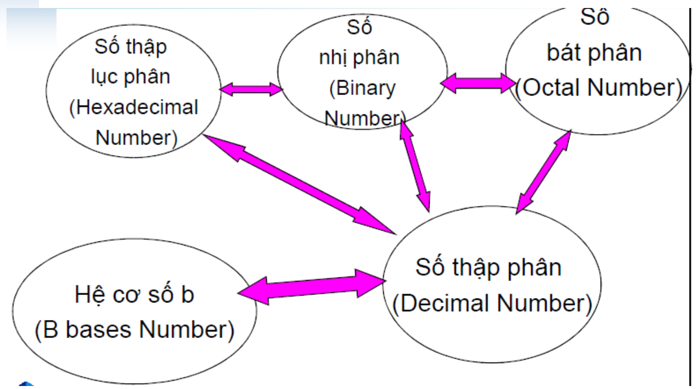
= $3A6_{16}$

Các hệ đếm thông dụng			
Decimal	Binary	Octal	Hexa
0	0000	00	0
1	0001	01	1
2	0010	02	2
3	0011	03	3
4	0100	04	4
5	0101	05	5
6	0110	06	6
7	0111	07	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

Hexadecimal Number System

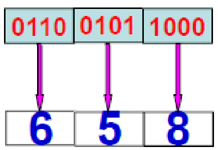
- Hexadecimal is useful for representing long strings of bits.
- Understanding the conversion process and memorizing the 4 bit patterns for each hexadecimal digit will prove valuable later.

Number Systems Conversion



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.



Decimal	BCD
0	0 0 0 0
1	0 0 0 1
2	0 0 1 0
3	0 0 1 1
4	0 1 0 0
5	0 1 0 1
6	0 1 1 0
7	0 1 1 1
8	1 0 0 0
9	1 0 0 1

BCD

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

8 7 4

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.

Binary	Gray Code
000	000
001	001
010	011
011	010
100	110
101	111
110	101
111	100

Gray Code

1 bit	2 bit	3 bit	4 bit
0	00	000	0000
1	01	001	0001
	11	011	0011
	10	010	0010
		110	0110
		111	0111
		101	0101
		100	0100
			1100
			1101
			1111
			1110
			1010
			1011
			1001
			1000

Putting It All Together

Decimal	Binary	Hexadecimal	BCD	Gray
0	0	0	0	0
1	1	1	0001	0001
2	10	2	0010	0011
3	11	3	0011	0010
4	100	4	0100	0110
5	101	5	0101	0111
6	110	6	0110	0101
7	111	7	0111	0100
8	1000	8	1000	1100
9	1001	9	1001	1101
10	1010	A	0001 0000	1111
11	1011	B	0001 0001	1110
12	1100	C	0001 0010	1010
13	1101	D	0001 0011	1011
14	1110	E	0001 0100	1001
15	1111	F	0001 0101	1000

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.
- ASCII – American Standard Code for Information Interchange.
  - 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).

Parity Method for Error Detection

- The parity method of error detection requires the addition of an extra bit to a code group.
- 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

- 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

- 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 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.

Odd Parity Error Detection

- Original data 10011010
- With Odd Parity 110011010
- 1-bit error 110111010
- Number of 1s even indicates 1-bit error
- 2-bit error 110110010
- Number of 1s odd no error indicated
- 3-bit error 100110010
- Number of 1s even indicates error