INTRODUCTION TO MICROPROCESSORS & COMPUTERS (Lecture 1)

Adane Tadesse

Faculty of Electrical and Computer Engineering
Computer and control Stream
Jimma University
Jimma, Oromia, Ethiopia
email: addmomcomp@gmail.com

October 29, 2019

Outline

Numbering and coding System

- human beings use base 10(decimal)arithmetic,computers use the base 2(binary)system. the convenient representation of binary numbers in base 16(hexadecimal)and the binary format of the alphanumeric code called ASCII.
- **Converting from decimal to binary** one method of converting from decimal to binary is to divide the decimal number by 2 repeatedly,keeping track of the remainders.this process continues until the quotient becomes zero.

- The reminders are then written in reverse order to obtain the binary number.

  Example1:convert 25_{10} to binary

  convert 740683_{10} to binary

- **converting from binary to decimal** to convert from binary to decimal,it is important to understand the concept of weight associated with each digit position. First as an analogy recall the weight of numbers in the base 10 system.By the same token each digit position in a number in base 2 has a weight associated with it.

  Example2:convert 110101_2 to Decimal

- **Hexadecimal system Base 16**, or the hexadecimal system as it is called in computer literature, is used as a convenient representation of binary numbers. For example, it is much easier for a human being 10 represent a string of Os and Is such as 100010010110 a its hexadecimal equivalent of896H.
- The binary System has 2 digits, 0 and 1.The base 10 system has 10 digits, 0 through 9. The hexadecimal (base 16) system ha 16 digits. In base 16, the first 10 digits, 0 to 9, are the same as in decimal, and for the remaining six digits, the letters A, B, C, D, E, and F are used.

- **Converting between binary and hex** To represent a binary number as its equivalent hexadecimal number, start from the right and group 4 bits at a lime, replacing cache 4-bit binary number with its hex equivalent. *Represent binary **100111110101** in hex. First the number is grouped into sets of 4 bits: 1001 1111 0101. Then each group of 4 bits is replaced with its hex equivalent: 1001 1111 0101 9F5 Therefore, 1001111101012 = **9F**5 hexadecimal*
- **Addition and subtraction of hex numbers** In studying issues related to software and hardware of computers, it is often necessary to add or subtract hex numbers.

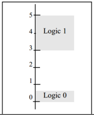
- Mastery of these techniques is essential.**Addition of hex numbers** the process of adding hex numbers. Starting with the least significant digits, the digits are added together. If the result is less than 16, write that digit as the sum for that position. If it is greater than 16, subtract 16 from it to get the digit and carry 1 to the next digit.
    - Example: 23D9 + 94BE.
- **Subtraction of hex numbers** In subtracting two hex numbers, if the second digit is greater than the first, borrow 16 from the preceding digit.
    - Example: B897-23D9

- **ASCII code:** In the 1960s a standard representation called ASCII (American Standard Code for Information Interchange) was established. The ASCII (pronounced "ask-E") code assigns binary patterns for numbers 0 to 9, all the letters of the English alphabet, both uppercase (capital) and lowercase, and many control codes and punctuation marks. The great advantage of this system is that it is used by most computers, so that information can be shared among computers. ASCII system uses a total of 7 bits to represent each code. For Example, **100 0001** is assigned to the uppercase letter "**A**" and **110 0001** is "**a**".

- Notice that the pattern of ASCII codes was designed to allow for easy manipulation of ASCII data. For example, digits 0 through 9 are represented by ASCII codes 30 through 39. This enables a program to easily convert ASCII to decimal by masking off the "3" in the upper nibble. Also notice that there is a relationship between the uppercase and lowercase letters. The uppercase letters are represented by ASCII codes 41 through 5A while lowercase letters are represented by codes 61 through 7A. Looking at the binary code, the only bit that is different between the uppercase "A" and lowercase "a" is bit 5.

Digital Logic Design

- **Binary logic** As mentioned earlier, computers use the binary number system because the two voltage levels can be represented as the two digits 0 and 1. Signals in digital electronics have two distinct voltage levels. For example, a system may define 0 V as logic 0 and +5 V as logic 1. in the below figure system with the built-in tolerances for variations in the voltage. A valid digital signal in this example should be within either of the two shaded areas.



- **Logic gates** Binary logic gates are simple circuits that take one or more input signals and send out one output signal.
- **AND gate** takes two or more inputs and performs a logic AND on them.

Logical AND Function

Inputs		**Output**
X	**Y**	**X AND Y**
0	0	0
0	1	0
1	0	0
1	1	1

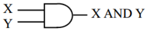


Figure : Logical AND Function

- In the case of AND, if all inputs are 1, the output is 1. If any input is 0, the output is 0.

- **OR logic** function will output a 1 if one or more inputs is 1. If all inputs are 0, then and only then will the output be 0.

Logical OR Function

Inputs		**Output**
X	**Y**	**X OR Y**
0	0	0
0	1	1
1	0	1
1	1	1

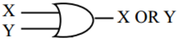
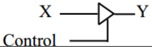


Figure : Logical OR Function

- **Tri-state** buffer gate does not change the logic level of the input. It is used to isolate or amplify the signal

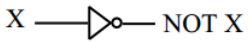
Buffer



- **Inverter**, also called NOT, outputs the value opposite to that input to the gate. That is, a 1 input will give a 0 output, while a 0 input will give a 1 output.

Logical Inverter

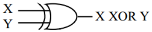
Input	Output
X	**NOT X**
0	1
1	0



- **XOR gate** performs an exclusive-OR operation on the inputs.

Logical XOR Function

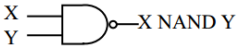
Inputs		Output
X	**Y**	**X XOR Y**
0	0	0
0	1	1
1	0	1
1	1	0



- **NAND gate** functions like an AND gate with an inverter on the output.It produces a 0 output when all inputs are 1; otherwise, it produces a 1 output.

Logical NAND Function

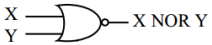
Inputs		Output
X	Y	X NAND Y
0	0	1
0	1	1
1	0	1
1	1	0



- **NOR gate** functions like an OR gate with an inverter on the output. It produces a 1 if all inputs are 0; otherwise, it produces a 0.

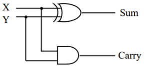
Logical NOR Function

Inputs		Output
X	Y	X NOR Y
0	0	1
0	1	0
1	0	0
1	1	0

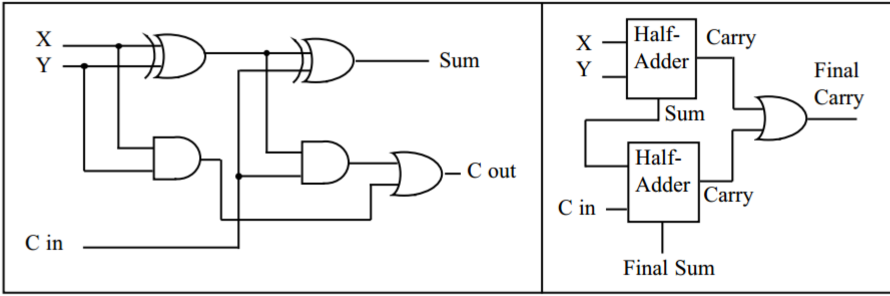


- NAND and NOR gates are used extensively in digital design because they are easy and **inexpensive to fabricate.**
- Any circuit that can be designed with AND, OR, XOR, and INVERTER gates can be implemented using **only NAND and NOR gates.**
- Logic design using gates *a simple logic design to add two binary digits.

	Carry	*Sum*
0 + 0 =	0	0
0 + 1 =	0	1
1 + 0 =	0	1
1 + 1 =	1	0

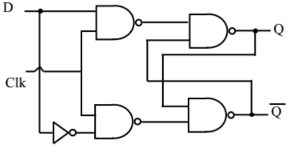


Full-Adder Built from a Half-Adder

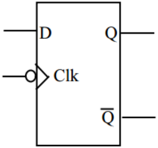


- in many computers and other kinds of processors adders used in the ALU part.

- **Flip-flops** A widely used component in digital systems is the flip-flop. Frequently, flip-flops are used to store data.
- The D flip-flop is widely used to latch data. Notice from the truth table that a D-FF grabs the data at the input as the clock is activated. A D-FF holds the data as long as the power is on.



(a) Circuit diagram



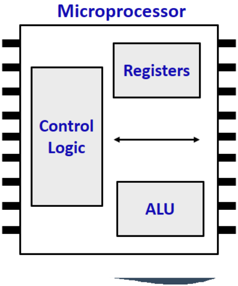
(b) Block diagram

Clk	D	Q
No	x	no change
$\downarrow$	0	0
$\downarrow$	1	1

x = don't care

(c) Truth table

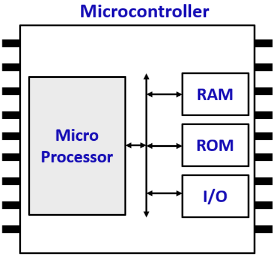
Basic concepts of Microprocessors



Basic concepts of Microprocessors

Microcontroller

Silicon chip which includes microprocessor, memory & I/O in a single package.

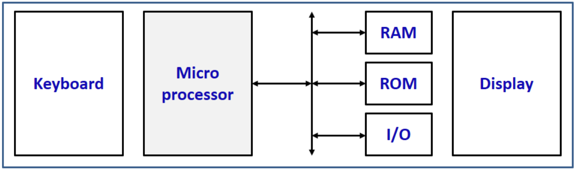


Basic concepts of Microprocessors

Microcomputer

A computer with a microprocessor as its CPU. Includes memory, I/O etc.

Microcomputer



- Microprocessor is a **computer Central Processing Unit (CPU**) on a single chip that contains millions of **transistors** connected by wires.
- A microprocessor is designed to perform arithmetic and logical operations that make use of small number-holding areas called registers.
- A microprocessor is communicates and operates in the binary number 0 & 1, called bits.
- Typical microprocessor operations include adding, subtracting, comparing two numbers and fetching numbers from one area to an another etc

Generally :

"The Microprocessor is a
Multipurpose,
Programmable,
Clock Driven,
Register Based
Digital-Integrated Circuit
which accepts binary data as input,
processes it according to instructions stored in its memory,
and provides results as output."

Can perform multiple tasks

Can be instructed to perform specific task

Provides Synchronization

Store Intermediate Processing data

Form of an Integrated circuit (IC)

History of Microprocessors

- Fairchild Semiconductors (founded in 1957) invented the first IC in 1959.
- In 1968, Robert Noyce, Gordan Moore, Andrew Grove resigned from Fairchild Semiconductors.
- They founded their own company Intel (Integrated Electronics).
- Intel grown from 3 man start-up in 1968

4-BIT MICROPROCESSORS

- INTEL 4004
  - Introduced in 1971.
  - It was the first microprocessor by Intel.
  - It was a 4-bit microprocessor.
  - Its clock speed was 740KHz.
  - It had 2,300 transistors.
  - It could execute around 60,000 instructions per second.
- INTEL 4040
  - Introduced in 1974.
  - It was also 4-bit microprocessor

8-BIT MICROPROCESSORS

- INTEL 8008
  - Introduced in 1972.
  - It was first 8-bit microprocessor.
  - Its clock speed was 500 KHz.
  - Could execute 50,000 instructions per second.

8-BIT MICROPROCESSORS

- INTEL 8080
    - Introduced in 1974.
    - It was also 8-bit Microprocessor.
    - Its clock speed was 2 MHz.
    - It had 6,000 transistors.
    - Was 10 times faster than 8008.
    - Could execute 5,00,000 instructions per second.

8-BIT MICROPROCESSORS

- INTEL 8085
  - Introduced in 1976.
  - It was also 8-bit microprocessor.
  - Its clock speed was 3 MHz.
  - Its data bus is 8-bit and address bus is 16-bit.
  - It had 6,500 transistors.
  - Could execute 7,69,230 instructions per second.
  - It could access 64 KB of memory.
  - It had 246 instructions.

16-BIT MICROPROCESSORS

- INTEL 8086
  - Introduced in 1978.
  - It was first 16-bit microprocessor.
  - Its clock speed is 4.77 MHz, 8 MHz and 10 MHz, depending on the version.
  - Its data bus is 16-bit and address bus is 20-bit.
  - It had 29,000 transistors.
  - Could execute 2.5 million instructions per second.
  - It could access 1 MB of memory.
  - It had 22,000 instructions memory.
  - It had Multiply and Divide instructions.