1

INTRODUCTION TO DIGITAL SYSTEMS

Digital systems are designed to store, process, and communicate information in digital form. They are found in a wide range of applications, including process control, communication systems, digital instruments, and consumer products. The digital computer, more commonly called the "computer," is an example of a typical digital system.

A computer manipulates information in digital, or more precisely, binary form. A binary number has only two discrete values — zero or one. Each of these discrete values is represented by the OFF and ON status of an electronic switch called a "transistor." All computers, therefore, only understand binary numbers. Any decimal number (base 10, with ten digits from 0 to 9) can be represented by a binary number (base 2, with digits 0 and 1).

The basic blocks of a computer are the central processing unit (CPU), the memory, and the input/output (I/O). The CPU of the computer is basically the same as the brain of a human being. Computer memory is conceptually similar to human memory. A question asked to a human being is analogous to entering a program into the computer using an input device such as the keyboard, and answering the question by the human is similar in concept to outputting the result required by the program to a computer output device such as the printer. The main difference is that human beings can think independently, whereas computers can only answer questions that they are programmed for. Computer hardware refers to components of a computer such as memory, CPU, transistors, nuts, bolts, and so on. Programs can perform a specific task such as addition if the computer has an electronic circuit capable of adding two numbers. Programmers cannot change these electronic circuits but can perform tasks on them using instructions.

Computer software, on the other hand, consists of a collection of programs. Programs contain instructions and data for performing a specific task. These programs, written using any programming language such as C, must be translated into binary prior to execution by the computer. This is because the computer only understands binary numbers.

Therefore, a translator for converting such a program into binary is necessary. Hence, a translator program called the *compiler* is used for translating programs written in a programming language such as C into binary. These programs in binary form are then stored in the computer memory for execution because computers only understand 0's and 1's. Furthermore, computers can only add. This means that all operations such as subtraction, multiplication, and division are performed by addition.

Due to advances in semiconductor technology, it is possible to fabricate the CPU in a single chip. The result is the *microprocessor*. Both Metal Oxide Semiconductor (MOS) and Bipolar technologies were used in the fabrication process. The CPU can be placed on a single chip when MOS technology is used. However, several chips are required with the bipolar technology. HCMOS (High Speed Complementary MOS) or BICMOS (Combination of Bipolar and HCMOS) technology (to be discussed later in this chapter) is normally used these days to fabricate the microprocessor in a single chip. Along with the microprocessor chip, appropriate memory and I/O chips can be used to design a *microcomputer*. The pins on each one of these chips can be connected to the proper lines on the system bus, which consists of address, data, and control lines. In the past, some manufacturers have designed a complete microcomputer on a single chip with limited capabilities. Single-chip microcomputers were used in a wide range of industrial and home applications.

Digital Logic with an Introduction to Verilog and FPGA-Based Design. First Edition. M. Rafiquzzaman and Steven A. McNinch.

"Microcontrollers" evolved from single-chip microcomputers. The microcontrollers are typically used for dedicated applications such as automotive systems, home appliances, and home entertainment systems. Typical microcontrollers, therefore, include a microcomputer, timers, and A/D (analog to digital) and D/A (digital to analog) converters — all in a single chip. Examples of typical microcontrollers are Intel 8751 (8-bit) / 8096 (16-bit) and Microchip PIC18F (8-bit) / PIC24F (16-bit).

In this chapter, we first define some basic terms associated with digital systems such as computers. We then describe briefly the basics of transistors, digital circuits, evolution of digital logic, microprocessors, and microcontrollers. Finally, a typical practical application is included.

1.1 Explanation of Terms

Before we go on, it is necessary to understand some basic terms (arranged in alphabetical order).

- *Address* is a pattern of 0's and 1's that represents a specific location in memory or a particular I/O device. An 8-bit microcontroller with 16 address bits can produce 2^{16} unique 16-bit patterns from 0000000000000000 to 1111111111111111, representing 65,536 different address combinations (addresses 0 to 65,535).
- *Arithmetic-logic unit* (ALU) is a digital circuit that performs arithmetic and logic operations on two n-bit digital words. The value of *n* for microcontrollers can be 8-bit or 16-bit. Typical operations performed by an ALU are addition, subtraction, ANDing, ORing, and comparison of two n-bit digital words. The size of the ALU defines the size of the microcontroller. For example, an 8-bit microcontroller contains an 8-bit ALU.
- *Bit* is an abbreviation for the term *binary digit*. A binary digit can have only two values, which are represented by the symbols 0 and 1, whereas a decimal digit can have 10 values, represented by the symbols 0 through 9. The bit values are easily implemented in electronic and magnetic media by two-state devices whose states portray either of the binary digits 0 and 1. Examples of such two-state devices are a transistor that is conducting or not conducting, a capacitor that is charged or discharged, and a magnetic material that is magnetized north to south or south to north.
- *Bit size* refers to the number of bits that can be processed simultaneously by the basic arithmetic circuits of a computer. A number of bits taken as a group in this manner is called a *word*. For example, an 8-bit microcontroller can process an 8-bit word. An 8-bit word is referred to as a *byte*, and a 4-bit word is known as a *nibble*.
- *Bus* consists of a number of conductors (wires) organized to provide a means of communication among different elements in a computer system. The conductors in a bus can be grouped in terms of their functions. A computer normally has an address bus, a data bus, and a control bus. Address bits are sent to memory or to an external device on the *address bus*. Instructions from memory, and data to/from memory or external devices, normally travel on the *data bus*. Control signals for the other buses and among system elements are transmitted on the control bus. Buses are sometimes *bidirectional*; that is, information can be transmitted in either direction on the bus, but normally in only one direction at a time.
- *Clock* is analogous to human heart beats. Computers require synchronization among its components, and this is provided by a *clock* or timing circuits.
- The *chip* is an integrated circuit (IC) package containing digital circuits.
- *CPU* (Central Processing Unit) contains several registers (memory elements), an ALU, and a control unit. Note that the control unit translates instructions and performs the desired task. The number of peripheral devices depends on the particular application involved and may even vary within an application.
- *EEPROM* or *E*^{2}*PROM* (Electrically Erasable Programmable ROM) is nonvolatile. EEPROMs can be programmed without removing the chip from the socket. EEPROMs are called Read Most Memories (RMMs), because they have much slower write times than read times. Therefore, these memories are usually suited for applications when mostly reading rather than writing is performed.

An example of EEPROM is the 28C64 (8K × 8).

- *EPROM* (Erasable Programmable ROM) is nonvolatile. EPROMs can be programmed and erased. The EPROM chip must be removed from the socket for programming. This memory is erased by exposing the chip to ultraviolet light via a lid or window on the chip. Typical erase times vary between 10 and 30 minutes. The EPROM is programmed by inserting the chip into a socket of the EPROM programmer, and providing proper addresses and voltage pulses at the appropriate pins of the chip. An example of EPROM is the 2764 (8K × 8).
- *Flash memory* is designed using a combination of EPROM and EEPROM technologies. Flash memory is nonvolatile and was invented by Toshiba in mid 1980s. Flash memory can be programmed electrically while embedded on the board. One can change multiple bytes at a time. An example of flash memory is the Intel 28F020 (256K × 8). Flash memory is typically used in cell phones and digital cameras.
- An *FPGA* (Field Programmable Gate Array) chip contains an array of digital logic blocks along with input and output blocks which can be connected together via programming using a Hardware Description Language (HDL) such as Verilog or VHDL. There are three types of components inside an FPGA. These are lookup table (stored in SRAM), flip-flops, and switch matrices. The concept of FPGA is based on the fact that a combinational circuit can be implemented using memory. In the past, digital logic circuits were built using all hardware (logic gates). It was a time-consuming task to debug the circuits. However, digital circuits implemented using FPGA's are faster to debug since they are programmable. Note that it is much faster to debug software than hardware. Hence, products can be developed using FPGA from conceptual design via prototype to production in a very short time. Note that FPGAs are primarily used for a product before mass production.
- The term *gate* refers to digital circuits which perform logic operations such as AND, OR, and NOT. In an AND operation, the output of the AND gate is one if all inputs are one; the output is zero if one or more inputs are zero. The OR gate, on the other hand, provides a zero output if all inputs are zero; the output is one if one or more inputs are one. Finally, a NOT gate (also called an inverter) has one input and one output. The NOT gate produces one if the input is zero; the output is zero if the input is one.
- *Microcomputer* typically consists of a microprocessor (CPU) chip, input and output chips, and memory chips in which programs (instructions and data) are stored.
- *Microcontroller* is implemented on a single chip typically containing a CPU, memory, Input/Output (I/O), timers, A/D (Analog-to-Digital) converter, and serial I/O.
- *Microprocessor* is the CPU of a microcomputer contained on a single chip, and must be interfaced with peripheral support chips in order to function.
- *Program* contains instructions and data.
- *Random-access memory* (RAM) is a storage medium for groups of bits or words whose contents cannot only be read but can also be altered at specific addresses. A RAM normally provides volatile storage, which means that its contents are lost in case power is turned off. There are two types of RAM: static RAM (SRAM), and dynamic RAM (DRAM). *Static RAM* stores data in flip-flops. Therefore, this memory does not need to be refreshed. An example of SRAM is 6116 (2K × 8). *Dynamic RAM*, on the other hand, stores data in capacitors. That is, it can hold data for a few milliseconds. Hence, DRAMs are refreshed typically by using external refresh circuitry. DRAMs are used in applications requiring large memory. DRAMs have higher densities than SRAMs. Typical examples of DRAMs are the 4464 (64K × 4), 44256 (256K × 4), and 41000 (1M × 1). DRAMs are inexpensive, occupy less space, and dissipate less power than SRAMs.
- *Read-only memory* (ROM) is a storage medium for the groups of bits called words, and its contents cannot normally be altered once programmed. A typical ROM is fabricated on a chip and can store, for example, 2048 eight-bit words, which can be accessed individually by presenting to it one of 2048 addresses. This ROM is referred to as a 2K by 8-bit ROM. 10110111 is an example of an 8-bit word that might be stored in one location in this memory. A ROM is a *nonvolatile storage device*,

which means that its contents are retained in case power is turned off. Because of this characteristic, ROMs are used to store permanent programs (instructions and data).

- *Register* can be considered as volatile storage for a number of bits. These bits may be entered into the register simultaneously (in parallel) or sequentially (serially) from right to left or from left to right, 1 bit at a time. An 8-bit register storing the bits 11110000 is represented as follows:

1	1	1	1	0	0	0	0

- The *speed power product* (SPP) is a measure of performance of a logic gate. It is expressed in picojoules (pJ). SPP is obtained by multiplying the speed (in ns) by the power dissipation (in mW) of a gate.
- *Transistors* are basically electronic switching devices. There are two types of transistors. These are *Bipolar Junction Transistors* (*BJTs*) and *Metal-Oxide Semiconductor* (*MOS*) transistors. The operation of the BJT depends on the flow of two types of carriers: electrons (*n*-channel) and holes (*p*-channel), whereas the MOS transistor is unipolar and its operation depends on the flow of only one type of carrier, either electrons (*n*-channel) or holes (*p*-channel).

1.2 Design Levels

Three design levels can be defined for digital systems: systems level, logic level, and device level.

- *Systems level* is the type of design in which CPU, memory, and I/O chips are interfaced to build a computer.
- *Logic level*, on the other hand, is the design technique in which chips containing logic gates such as AND, OR, and NOT are used to design a digital component such as an ALU. Emphasis is given on logic level design in this book.
- Finally, *device level* utilizes transistors to design logic gates.

1.3 Combinational vs. Sequential Systems

Digital systems at the logic level can be classified into two types. These are *combinational* and *sequential*.

Combinational systems contain no memory whereas sequential systems require memory to remember the present state in order to go to the next state. A binary adder capable of providing the sum upon application of the numbers to be added is an example of a combinational system. For example, consider a 4-bit adder. The inputs to this adder will be two 4-bit numbers; the output will be the 4-bit sum. In this case, the adder will generate the 4-bit sum output upon application of the two 4-bit inputs.

Sequential systems, on the other hand, require memory. The counter is an example of a sequential system. For instance, suppose that the counter is required to count in the sequence 0, 1, 2 and then repeat the sequence. In this case, the counter must have memory to remember the present count in order to go to the next. The counter must remember that it is at count 0 in order to go to the next count, 1. In order to count to 2, the counter must remember that it is counting 1 at the present state. In order to repeat the sequence, the counter must count back to 0 based on the present count, 2, and the process continues. A chip containing sequential circuit such as the counter will have a clock input pin.

In general, all computers contain both combinational and sequential circuits. However, most computers are regarded as clocked sequential systems. In these computers, almost all activities pertaining to instruction execution are synchronized with clocks.

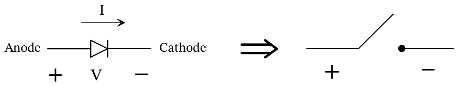
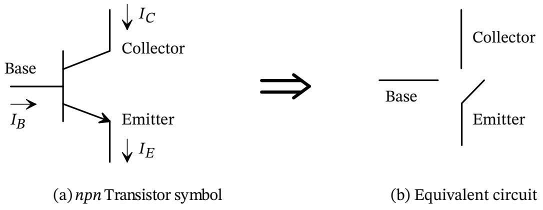


FIGURE 1.1 Symbolic representations of a diode



(a) *npn* Transistor symbol (b) Equivalent circuit

FIGURE 1.2 Symbolic representations of a *npn* transistor

1.4 Digital Circuits

Circuits which understand 0's and 1's (binary numbers) are called "Digital Circuits". Transistors are used to design digital circuits. The transistor can be considered as an electronic switch. The ON and OFF states of a transistor are used to represent binary digits. Transistors, therefore, play an important role in the design of digital systems. This section describes the basic characteristics of digital devices and logic families. These include diodes, transistors, and a summary of digital logic families. These topics are covered from a very basic point of view. This will allow the readers with some background in digital devices to see how they are utilized in designing digital systems.

1.4.1 Diodes

A diode is an electronic switch. It is a two-terminal device. Figure 1.1 shows the symbolic representation. The positive terminal (made with the *p*-type semiconductor material) is called the anode; the negative terminal (made with the *n*-type semiconductor material) is called a cathode. When a voltage, $V = 0.6$ volt is applied across the anode and the cathode, the switch closes and a current I flows from anode to the cathode.

1.4.2 Transistors

A bipolar junction transistor (BJT) or commonly called the transistor is also an electronic switch like the diode. Both electrons (*n*-channel) and holes (*p*-channel) are used for carrier flow; hence, the name "bipolar" is used. The BJT is used in transistor logic circuits that have several advantages over diode logic circuits. First of all, the transistor acts as a logic device called an inverter. Note that an inverter provides a LOW output for a HIGH input and a HIGH output for a LOW input. Secondly, the transistor is a current amplifier (buffer). Transistors can, therefore, be used to amplify these currents to control external devices such as a light emitting diode (LED) requiring high currents. Finally, transistor logic gates operate faster than diode gates.

There are two types of transistors, namely *npn* and *pnp*. The classification depends on the fabrication process. npn transistors are widely used in digital circuits. Figure 1.2 shows the symbolic representation of an *npn* transistor. The transistor is a three-terminal device. These are base, emitter, and collector.

The transistor is a current-controlled switch. This means that an adequate current at the base will close the switch allowing a current to flow from the collector to the emitter.

This current direction is identified on the *npn* transistor symbol in Figure 1.2(a) by a downward arrow on the emitter. Note that a base resistance is required to generate the base current.

The transistor has three modes of operation: cutoff, saturation, and active. In digital circuits, a transistor is used as a switch, which is either ON (closed) or OFF (open). When no base current flows, the emitter~collector switch is open and the transistor operates in the cutoff (OFF) mode. On the other hand, when a base current flows such that the voltage across the base and the emitter is at least 0.6V, the switch closes. If the base current is further increased, there will be a situation in which V_{CE} (voltage across the collector and the emitter) attains a constant value of approximately 0.2V. This is called the saturation (ON) mode of the transistor. The “active” mode is between the cutoff and saturation modes. In this mode, the base current (I_B) is amplified so that the collector current, IC = β $I_{B,}$ where β is called the gain, and is in the range of 10 to 100 for typical transistors. Note that when the transistor reaches saturation, increasing I_B does not drop V_{CE} below V_{CE} (Sat.) of 0.2V. On the other hand, V_{CE} varies from 0.8V to 5V in the active mode. Therefore, the cutoff (OFF) and saturation (ON) modes of the transistor are used in designing digital circuits. The active mode of the transistor in which the transistor acts as a current amplifier (also called buffer) is used in digital output circuits.

Operation of the Transistor as an Inverter Figure 1.3 shows how to use the transistor as an inverter. When $V_{IN} = 0$, the transistor is in cutoff (OFF), and the collector-emitter switch is open. This means that no current flows from $+V_{CC}$ to ground. V_{OUT} is equal to $+V_{CC}$. Thus, V_{OUT} is high.

On the other hand, when V_{IN} is HIGH, the emitter-collector switch is closed. A current flows from $+V_{CC}$ to ground. The transistor operates in saturation, and $V_{OUT} = V_{CE}$ (Sat) = 0.2V ≈ 0. Thus, V_{OUT} is basically connected to ground.

Therefore, for V_{IN} = LOW, V_{OUT} = HIGH, and for V_{IN} = HIGH, V_{OUT} = LOW. Hence, the *npn* transistor in Figure 1.3 acts as an inverter.

Note that V_{CC} is typically +5V DC. The input voltage levels are normally in the range of 0 to 0.8 volts for LOW and 2 volts to 5 volts for HIGH. The output voltage levels, on the other hand, are normally 0.2 volts for LOW and 3.6 volts for HIGH.

Light Emitting Diodes (LEDs) and Seven Segment Displays LEDs are extensively used as outputs in digital systems as status indicators. An LED is typically driven by low voltage and low current. This makes the LED a very attractive device for use with digital systems. Table 1.1 provides the current and voltage requirements of red, yellow, and green LEDs.

Basically, an LED will be ON, generating light, when its cathode is sufficiently negative with respect to its anode. A digital system such as a microcomputer can therefore light an LED either by grounding the cathode (if the anode is tied to +5V) or by applying +5V to the anode (if the cathode is grounded) through an appropriate resistor value. A typical hardware interface between a microcomputer and an LED is depicted in Figure 1.4. A digital circuit using BJTs normally outputs 400 μA at a minimum voltage, V_M = 2.4 volts for a HIGH. The red LED requires 10 mA at 1.7 volts. A buffer (current amplifier) such as a transistor is required to turn the LED ON. Since the transistor is an inverter, a HIGH input to the transistor will turn the LED ON. We now design the interface; that is, the values of R_1, R_2, and the gain β for the transistor will be determined.

A HIGH at the output of a digital circuit will turn the transistor ON into active mode. This will allow a path of current to flow from the +5V source through R_2 and the LED to the ground. The appropriate value of R_2 needs to be calculated to satisfy the voltage and current requirements of the LED. Also, suppose that VBE = 0.6V when the transistor is in active mode. This means that R1 needs to be calculated with the specified values of V_M = 2.4V and I = 400 μA. The values of R_1, R_2, and β are calculated as follows:

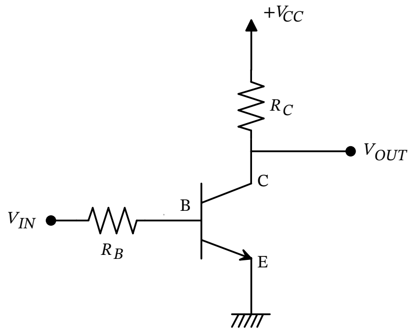


FIGURE 1.3 An inverter

TABLE 1.1 Current and Voltage Requirements of LEDs

LEDs	Red	Yellow	Green
Current	10 mA	10 mA	20 mA
Voltage	1.7V	2.2V	2.4V

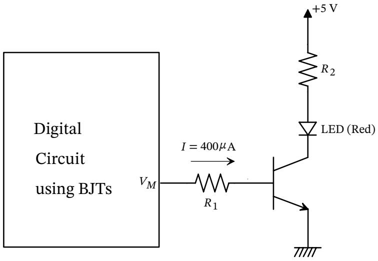


FIGURE 1.4 Digital circuit - LED interface

$$R_1 = \frac{V_M - V_{BE}}{400\,\mu A} = \frac{2.4 - 0.6}{400\,\mu A} = 4.5\ K\Omega$$

$$\text{Assuming} V_{CE} \cong 0,$$

$$R_2 = \frac{5 - 1.7 - V_{CE}}{10\ mA} = \frac{5 - 1.7}{10\ mA} = 330\ \Omega$$

$$\beta = \frac{I_C}{I_B} = \frac{10\ mA}{400\,\mu A} = \frac{10 \times 10^{-3}}{400 \times 10^{-6}} = 25$$

Therefore, the interface design is complete, and a transistor with a minimum β of 25, R_1 = 4.5 KΩ, and R_2 = 330 Ω are required.

An inverting buffer chip such as the 74LS04 can be used in place of a transistor in Figure 1.4. A typical interface of a LED to a BJT-based digital circuit via an inverter is shown in Figure 1.5. Note that the transistor base resistance is inside the inverter. Therefore, R_1 is not required to be connected to the output of the digital circuit. The symbol −▷o− is used to represent an inverter. Inverters will be discussed in more detail later. In Figure 1.5, when the digital circuit outputs a HIGH, the transistor switch inside the inverter closes. A current flows from the +5V source, through the 330-ohm resistor and the LED, into the ground inside the inverter. The LED is thus turned ON.

Note that if 5V is used to turn the LED ON and 0V to turn it OFF, the LED should be connected as shown in Figure 1.6. However, if 0 is used to turn the LED ON and 5V to turn it OFF, the LED should be connected as shown in Figure 1.7.

A seven-segment display can be used to display, for example, decimal numbers from 0 to 9. The name "seven segment" is based on the fact that there are seven LEDs — one in each segment of the display. Figure 1.8 shows a typical seven-segment display.

In Figure 1.8, each segment contains an LED. All decimal numbers from 0 to 9 can be displayed by turning the appropriate segment "ON" or "OFF". For example, a zero can be displayed by turning the LED in segment *g* "OFF" and turning the other six LEDs in segments *a* through *f* "ON." There are two types of seven segment displays. These are common cathode and common anode. Figure 1.9 shows these display configurations.

In a common cathode arrangement, the computer can send a HIGH to light a segment and a LOW to turn it off. In a common anode configuration, on the other hand, the computer sends a LOW to light a segment and a HIGH to turn it off. In both configurations, R = 330 ohms can be used.

Transistor Transistor Logic (TTL) and its Variations The transistor transistor logic (TTL) family of chips evolved from diodes and transistors. This family used to be called DTL (diode transistor logic). The diodes were then replaced by transistors, and thus the name "TTL" evolved. The power supply voltage (V_{CC}) for TTL is +5V. The two logic levels are approximately 0 and 3.5V.

There are several variations of the TTL family. These are based on the saturation mode (saturated logic) and active mode (nonsaturated logic) operations of the transistor. In the saturation

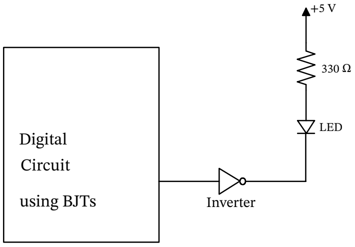


FIGURE 1.5 Digital circuit - LED interface via an inverter

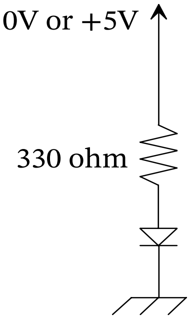


FIGURE 1.6 An LED connection to be turned ON by 5V and turned OFF by 0V

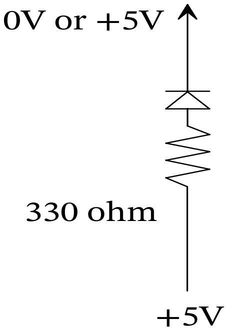


FIGURE 1.7 An LED connection to be turned ON by 0V and turned OFF by 5V

mode, the transistor takes some time to come out of the saturation to switch to the cutoff mode. On the other hand, some TTL families define the logic levels in the active mode operation of the transistor and are called nonsaturated logic. Since the transistors do not go into saturation, these families do not have any saturation delay time for the switching operation. Therefore, the nonsaturated logic family is faster than saturated logic.

The saturated TTL family includes standard TTL (TTL), high-speed TTL (H-TTL), and low-power TTL (L-TTL). The nonsaturated TTL family includes Schottky TTL (S-TTL), low-power Schottky TTL (LS-TTL), advanced Schottky TTL (AS-TTL), and advanced low-power Schottky TTL (ALS-TTL). The development of LS-TTL made TTL, H-TTL, and L-TTL obsolete. Another technology, called emitter-coupled logic (ECL), utilizes nonsaturated logic. The ECL family provides the highest speed. ECL is used in digital systems requiring ultrahigh speed, such as supercomputers.

The important parameters of the digital logic families are fan-out, power dissipation, propagation delay, and noise margin. Fan-out is defined as the maximum number of inputs that can be connected to the output of a gate. It is expressed as a number. The output of a gate is normally connected to the inputs of other similar gates. Typical fan-out for TTL is 10. On the other hand, fan-outs for S-TTL, LS-TTL, and ECL, are 10, 20, and 25, respectively.

Power dissipation is the power (milliwatts) required to operate the gate. This power must be supplied by the power supply and is consumed by the gate. Typical power consumed by TTL is 10 mW. On the other hand, S-TTL, LS-TTL, and ECL absorb 22 mW, 2 mW, and 25 mW respectively.

Propagation delay is the time required for a signal to travel from input to output when the binary output changes its value. Typical propagation delay for TTL is 10 nanoseconds (ns). On the other

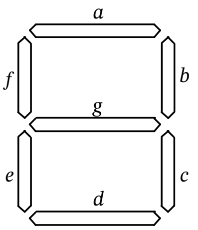


FIGURE 1.8 A seven-segment display

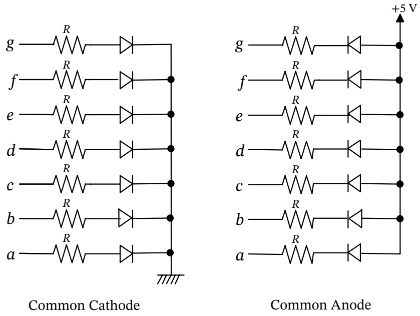


FIGURE 1.9 Seven-segment display configurations

hand, S-TTL, LS-TTL, and ECL have propagation delays of 3 ns, 10 ns, and 2 ns, respectively.

Noise margin is defined as the maximum voltage due to noise that can be added to the input of a digital circuit without causing any undesirable change in the circuit output. Typical noise margin for TTL is 0.4V. Noise margins for S-TTL, LS-TTL, and ECL are 0.4V, 0.4V, and 0.2V, respectively.

TTL Outputs There are three types of output configurations for TTL. These are open-collector output totem-pole output, and tristate (three-state) output.

The open-collector output means that the TTL output is a transistor with nothing connected to the collector. The collector voltage provides the output of the gate. For the open-collector output to work properly, a resistor (called the pullup resistor), with a value of typically 1K, should be connected between the open collector output and a +5V power supply.

If the outputs of several open-collector gates are tied together with an external resistor (typically 1K) to a +5V source, a logical AND function is performed at the connecting point. This is called wired-AND logic. Figure 1.10 shows two open-collector outputs (*A* and *B*) are connected together to a common output point *C* via a 1 KΩ resistor and a +5V source.

The common-output point *C* is HIGH only when both transistors are in cutoff (OFF) mode, providing A = HIGH and B = HIGH. If one or both of the two transistors is turned ON, making one (or both open-collector outputs) LOW, this will drive the common output *C* to LOW. Note that a LOW (Ground for example) signal when connected to a HIGH (+5V for example) signal generates a LOW. Thus, *C* is obtained by performing a logical AND operation of the open collector outputs *A* and *B*.

Let us briefly review the totem-pole output circuit shown in Figure 1.11. The circuit operates as follows:

When transistor Q_1 is ON, transistor Q_2 is OFF. When Q_1 is OFF, Q_2 is ON. This is how the totem-pole output is designed. The complete TTL gate connected to the bases of transistors Q_1 and Q_2 is not shown; only the output circuit is shown.

In the figure, Q_1 is turned ON when the logic gate circuit connected to its base sends a HIGH output. The switches in transistor Q_1 and diode D close while the switch in Q_2 is open. A current flows from the +5V source through R, Q_1, and D to the output. This current is called I_{source} or output high current, I_{OH}. This is typically represented by a negative sign in front of the current value in the TTL data

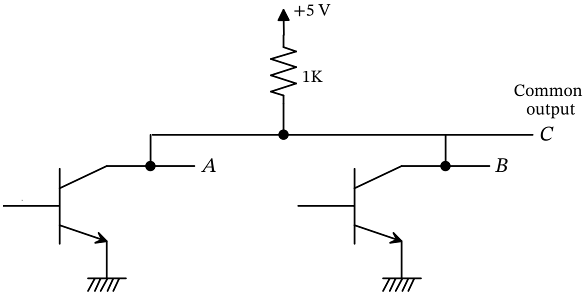


FIGURE 1.10 Two open-collector outputs A and B tied together

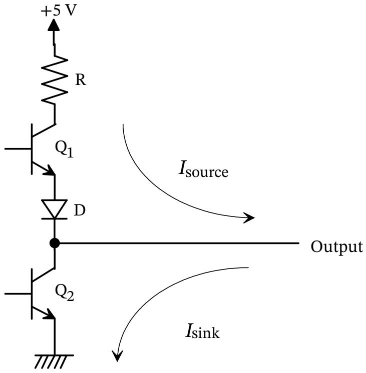


FIGURE 1.11 TTL Totem-pole output

sheet, a notation indicating that the chip is losing current. For a low output value of the logic gate, the switches in Q_1 and D are open and the switch in Q_2 closes. A current flows from the output through Q_2 to ground. This current is called I_{sink} or Output Low current, I_{OL}. This is represented by a positive sign in front of the current value in the TTL data sheet, indicating that current is being added to the chip. Either I_{source} or I_{sink} can be used to drive a typical output device such as an LED. Isource (I_{OH}) is normally much smaller than I_{sink} (IOL). I_{source} (I_{OH}) is typically –0.4 mA (or –400 μA) at a minimum voltage of 2.7V at the output. I_{source} is normally used to drive devices that require high currents. A current amplifier (buffer) such as a transistor or an inverting buffer chip such as the 74LS368 needs to be connected at the output if I_{source} is used to drive a device such as an LED requiring high current (10 mA to 20 mA). I_{sink} is normally 8 mA.

The totem-pole outputs must not be tied together. When two totem-pole outputs are connected together with the output of one gate HIGH and the output of the second gate LOW, the excessive amount of current drawn can produce enough heat to damage the transistors in the circuit.

Tristate is a special totem-pole output that allows connecting the outputs together like the open-collector outputs. When a totem-pole output TTL gate has this property, it is called a tristate (three state) output. A tristate has three output states:

1. A LOW level state when the lower transistor in the totem-pole is ON and the upper transistor is OFF.

2. A HIGH level when the upper transistor in the totem-pole is ON and the lower transistor is OFF.

3. A third state when both output transistors in the totem-pole are OFF. This third state provides an open circuit or high-impedance state which allows a direct wire connection of many outputs to a common line called the bus.

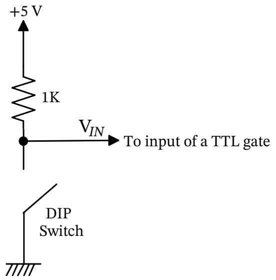


FIGURE 1.12 A typical circuit for connecting an input to a TTL gate

A Typical Switch Input Circuit for TTL Figure 1.12 shows a switch circuit that can be used as a single bit into the input of a TTL gate. When the DIP switch is open, V_{IN} is HIGH. On the other hand, when the switch is closed, V_{IN} is low. V_{IN} can be used as an input bit to a TTL logic gate for performing laboratory experiments.

1.4.3 MOS Transistors

Metal-Oxide Semiconductor (MOS) transistors occupy less space in the circuit and consume much less power than bipolar junction transistors. Therefore, MOS transistors are used in highly integrated circuits. The MOS transistor is unipolar. This means that one type of carrier flow, either electrons (n-type) or holes (p-type) are used. The MOS transistor works as a voltage-controlled resistance. In digital circuits, a MOS transistor operates as a switch such that its resistance is either very high (OFF) or very low (ON). The MOS transistor is a three-terminal device: gate, source, and drain. There are two types of MOS transistors, namely, nMOS and pMOS.

Figure 1.13 shows the symbolic representation of an nMOS transistor. When $V_{GS} = 0$, the resistance between drain and source (RDS) is in the order of megaohms (Transistor OFF state). On the other hand, as V_{GS} is increased, RDS decreases to a few tens of ohms (Transistor ON state). Note that in a MOS transistor, there is no connection between the gate and the other two terminals (source and drain). The nMOS gate voltage (V_{GS}) increases or decreases the current flow from drain to source by changing RDS. Popular 8-bit microprocessors such as the Intel 8085 and the Motorola 6809 were designed using nMOS.

Figure 1.14 depicts the symbol for a pMOS transistor. The operation of the pMOS transistor is very similar to the nMOS transistor except that V_{GS} is typically zero or negative. The resistance from drain to source (RDS) becomes very high (OFF) for $V_{GS} = 0$. On the other hand, RDS decreases to a very low value (ON) if V_{GS} is decreased. pMOS was used in fabricating the first 4-bit microprocessors (Intel 4004/4040) and 8-bit microprocessor (Intel 8008). Basically, in a MOS transistor (nMOS or pMOS), V_{GS}creates an electric field that increases or decreases the current flow between source and drain. From the symbols of the MOS transistors, it can be seen that there is no connection between the gate and the other two terminals (source and drain). This symbolic representation is used in order to indicate that no current flows from the gate to the source, irrespective of the gate voltage.

Operation of the nMOS Transistor as an Inverter Figure 1.15 shows an nMOS inverter. Furthermore, when V_{in} = LOW, the resistance between the drain and the source (R_{DS}) is very high, and no current flows from V_{CC} to the ground. V_{out} is, therefore, high. On the other hand, when V_{in} = high, R_{DS} is very low, a current flows from V_{CC} to the source, and V_{out} is LOW. Therefore, the circuit acts as an inverter.

Complementary MOS (CMOS) CMOS dissipates low power and offers high circuit density compared to TTL. CMOS is fabricated by combining nMOS and pMOS transistors together. The nMOS transistor transfers logic 0 well and logic 1 inefficiently. The pMOS transistor, on the other hand, outputs logic 1 efficiently and logic 0 poorly. Therefore, connecting one pMOS and one nMOS transistor in parallel provides a single switch called *transmission gate* that offers efficient output drive capability for CMOS

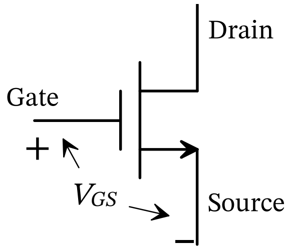


FIGURE 1.13 nMOS transistor symbol

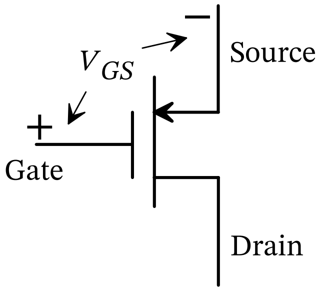


FIGURE 1.14 pMOS transistor symbol

logic gates. The transmission gate is controlled by an input logic level.

Although MOS devices are voltage-controlled devices, they can be visualized as switches. The nMOS transistor is a switch which turns ON when the gate voltage, V_{gs} = +5V; otherwise, the nMOS transistor is OFF (Logic 0). The PMOS transistor is also a switch which turns ON (Logic 1) when the gate voltage, V_{gs} = 0V; otherwise, the pMOS transistor is OFF (Logic 0). Figures 1.16 (a) and (b) show the new symbols for nMOS and pMOS illustrating the concept just described. The nMOS and pMOS transistors can be combined to obtain CMOS.

Figure 1.17 shows a typical CMOS inverter with the new symbols for nMOS and pMOS. The CMOS inverter is very similar to the TTL totem-pole output circuit. That is, when Q_1 is ON (low resistance), Q_2 is OFF (high resistance), and vice versa. When Vinput = LOW, Q_1 is ON and Q_2 is OFF. This makes V_{out} HIGH. On the other hand, when V_{in} = HIGH, Q_1 is OFF (high resistance) and Q_2 is ON (low resistance). This provides a low V_{out}. Thus, the circuit works as an inverter.

Figures 1.18 and 1.19 show the equivalent circuits with switches for nMOS and pMOS with V_{in} = 0V and V_{in} = +5V respectively. In Figure 1.18, note that for V_{in} = +5V, the pMOS transistor is OFF while the nMOS transistor is ON. Hence, V_{out} is 0. In Figure 1.19, on the other hand, for V_{in} = 0V, the nMOS transistor is OFF while the pMOS transistor is ON. Hence, V_{out} = +5V.

Digital circuits using CMOS consume less power than MOS and bipolar transistor circuits. In addition, CMOS provides high circuit density. That is, more circuits can be placed in a chip using CMOS. Finally, CMOS offers high noise immunity. In CMOS, unused inputs should not be left open. Because of the very high input resistance, a floating input may change back and forth between a LOW and a HIGH, creating system problems. All unused CMOS inputs should be tied to V_{CC}, ground, or another high or low signal source appropriate to the device's function. CMOS can operate over a large range of power supply voltages (3V to 15V). Two CMOS families, namely CD4000 and 54C/74C, were first introduced. CD 4000A is in the declining stage.

There are four members in the CMOS family which are very popular these days: the high-speed CMOS (HC), high-speed CMOS/TTL-input compatible (HCT), advanced CMOS (AC), and advanced CMOS/TTL-input compatible (ACT). The HCT chips have a specifically designed input circuit that is compatible with LS-TTL logic levels (2V for HIGH input and 0.8V for LOW input). LS-TTL outputs can directly drive HCT inputs while HCT outputs can directly drive HC inputs. Therefore, HCT buffers can be placed between LS-TTL and HC chips to make the LS-TTL outputs compatible with the HC inputs.

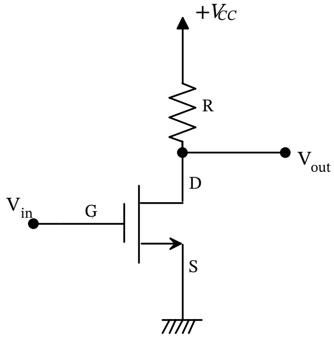
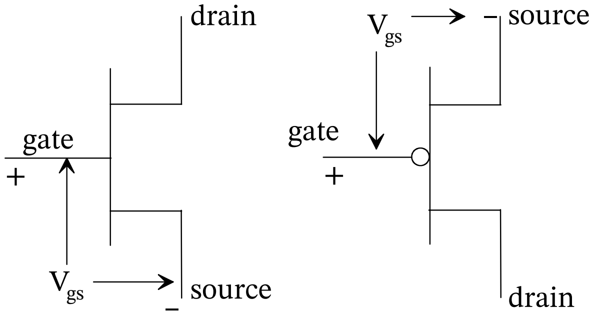


FIGURE 1.15 A typical nMOS inverter



(a) nMOS Transistor (b) pMOS Transistor

FIGURE 1.16 NMOS and PMOS transistors

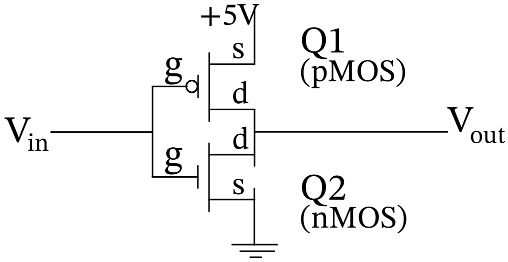


FIGURE 1.17 A CMOS inverter

Several characteristics of 74HC and 74HCT are compared with 74LS-TTL and nMOS technologies in Table 1.2. Note that in the table, HC and HCT have the same source (I_{OH}) and sink (I_{OL}) currents. This is because in a typical CMOS gate, the ON resistances of the pMOS and nMOS transistors are approximately the same.

The input characteristics of HC and HCT are compared in Table 1.3.

Table 1.3 shows that LS-TTL is not guaranteed to drive an HC input. The LS-TTL output HIGH is greater than or equal to 2.7V while an HC input needs at least 3.15V. Therefore, the HCT input requiring VIH of 2.0V can be driven by the LS-TTL output, providing at least 2.7V; 74HCT244 (unidirectional) and 74HCT245 (bidirectional) buffers can be used.

HCMOS Outputs Like TTL, the MOS logic offers three types of outputs. These are push-pull (totem-pole in TTL), open drain (open collector in TTL), and tristate outputs. For example, the 74HC00 contains four independent 2-input NAND gates and includes push-pull output. The 74HC03 also contains four independent 2-input NAND gates, but has open drain outputs. The 74HC03 requires a pull-up resistor for each gate. The 74HC125 contains four independent tri-state buffers in a single chip.

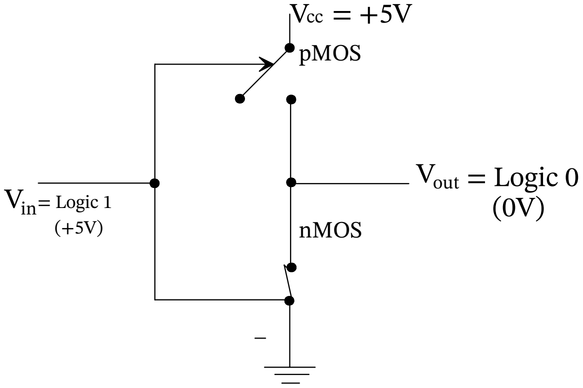


FIGURE 1.18 Representation of the CMOS inverter using switches with V_{in} = 5V and V_{out} = 0V

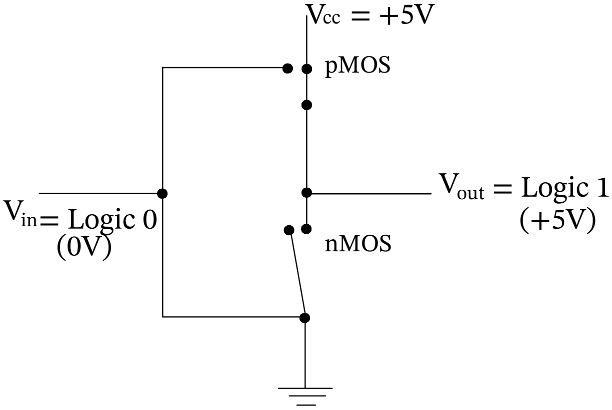


FIGURE 1.19 Representation of the CMOS inverter using switches with V_{in} = 0V and V_{out} = +5V

A Typical Switch Input Circuit for MOS Chips Figure 1.20 shows a switch circuit that can be used as a single bit into the input of a MOS gate. When the DIP switch is open, V_{IN} is HIGH. On the other hand, when the switch is closed, V_{IN} is LOW. V_{IN} can be used as an input bit for performing laboratory experiments. Note that unlike TTL, a 1K resistor is connected between the switch and the input of the MOS gate. This provides for protection against static discharge. This 1-Kohm resistor is not required if the MOS chip contains internal circuitry providing protection against damage to inputs due to static discharge.

1.5 Integrated Circuits (ICs)

Device level design utilizes transistors to design circuits called gates, such as AND gates and OR gates. One or more gates are fabricated on a single silicon chip by an integrated circuit (IC) manufacturer in an IC package.

An IC chip is packaged typically in a ceramic or plastic package. The commercially available ICs can be classified as small-scale integration (SSI), medium-scale integration (MSI), large-scale integration (LSI), and very large-scale integration (VLSI).

- A single SSI IC contains a maximum of approximately 10 gates. Typical logic functions such as AND, OR, and NOT are implemented in SSI IC chips. The MSI IC, on the other hand, includes from 11 to up to 100 gates in a single chip. The MSI chips normally perform specific functions such as add.
- The LSI IC contains more than 100 to approximately 1000 gates. Digital systems such as 8-bit microprocessors and memory chips are typical examples of LSI ICs.

TABLE 1.2 Comparison of output characteristics of LS-TTL, nMOS, HC, and HCT

	V_{OH}	I_{OH}	V_{OL}	I_{OL}
LS-TTL	2.7V	–400 μA	0.5V	8 mA
nMOS	2.4V	–400 μA	0.4V	2 mA
HC	3.7V	–4 mA	0.4V	4 mA
HCT	3.7V	–4 mA	0.4V	4 mA

TABLE 1.3 Comparison of input characteristics of HC and HCT

	V_{IH}	I_{IH}	V_{IL}	I_{IL}	Fanout
HC	3.15V	1μA	0.9V	1μA	10
HCT	2.0V	1μA	0.8V	1μA	10

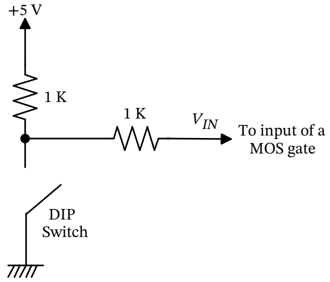


FIGURE 1.20 A typical switch for MOS input

- The VLSI IC includes more than 1000 gates. More commonly, the VLSI ICs are identified by the number of transistors (containing over 500,000 transistors) rather than the gate count in a single chip. Typical examples of VLSI IC chips include 32-bit microprocessors and one megabit memories. For example, the Intel Pentium is a VLSI IC containing 3.1 million transistors in a single chip.

An IC chip is usually inserted in a printed-circuit board (PCB) that is connected to other IC chips on the board via pins or electrical terminals. In laboratory experiments or prototype systems, the IC chips are typically placed on breadboards or wire-wrap boards and connected by wires. The breadboards normally have noise problems for frequencies over 4 MHz. Wire-wrap boards are used above 4 MHz. The number of pins in an IC chip varies from ten to several hundred, depending on the package type. Each IC chip must be powered and grounded via its power and ground pins. The VLSI chips such as the Pentium have several power and ground pins. This is done in order to reduce noise by distributing power in the circuitry inside the chip.

The SSI and MSI chips normally use an IC package called *dual in-line package* (DIP). The LSI and VLSI chips, on the other hand, are typically fabricated in surface-mount or pin grid array (PGA) packages. The DIP is widely used because of its low price and ease of installation into the circuit board.

SSI chips are identified as 5400-series (these are for military applications with stringent requirements on voltage and temperature and are expensive) or 7400 series (for commercial applications). Both series have identical pin assignments on chips with the same part numbers, although the first two numeric digits of the part name are different. Typical commercial SSI ICs can be identified as follows:

74S	Schottky TTL
74LS	Low-power Schottky TTL
74AS	Advanced Schottky TTL
74F	Fast TTL (Similar to 74AS; manufactured by Fairchild)
74ALS	Advanced low-power Schottky TTL

Note that two digits appended at the end of each of these IC identifications define the type of logic operation performed, the number of pins, and the total number of gates on the chip. For example, 74S00, 74LS00, 74AS00, 74F00, and 74ALS00 perform NAND operation. All of them have 14 pins and contain four independent NAND gates in a single chip.

The gates in the ECL family are identified by the part numbers 10XXX and 100XXX, where XXX indicates three digits. The 100XXX family is faster, requires low power supply, but it consumes more power than the 10XXX. Note that 10XXX and 100XXX are also known as 10K and 100K families.

The commercially available CMOS family is identified in the same manner as the TTL SSI ICs. For example, 74LS00 and 74HC00 (High-speed CMOS) are identical, with 14 pins and containing four independent NAND gates in a single chip. Note that 74HCXX gates have operating speeds similar to 74LS-TTL gates. For example, the 74HC00 contains four independent two-input NAND gates. Each NAND gate has a typical propagation delay of 10 ns and a fanout of 10 LS-TTL.

Unlike TTL inputs, CMOS inputs should never be held floating. The unused input pins must be connected to V_{CC}, ground, or an output. The TTL input contains an internal resistor that makes it HIGH when unused or floating. The CMOS input does not have any such resistor and therefore possesses high resistance. The unused CMOS inputs must be tied to V_{CC}, ground, or other gate outputs. In some CMOS chips, inputs have internal pull-up or pull-down resistors. These inputs, when unused, should be connected to V_{CC} or ground to make the inputs high or low.

The CMOS family has become popular compared to TTL due to better performance. Some major IC manufacturers such as National Semiconductor do not make 7400 series TTL anymore. Although some others, including Fairchild and Texas Instruments still offer the 7400 TTL series, the use of the SSI TTL family (74S, 74LS, 74AS, 74F, and 74ALS) is in the declining stage, and will be obsolete in the future. On the other hand, the use of CMOS-based chips such as 74HC has increased significantly because of their high performance. These chips will dominate the future market.

1.6 CAD (Computer-Aided Design)

Digital logic circuits were used in building the first computers. With the advent of VLSI technology, millions of transistors are contained in the same chip. Hence, it has become a difficult task to design these circuits without using computer-aided design tools.

CAD tools include programs that assist in developing the digital hardware. The CAD tools perform the design process automatically, and comes up with an optimized circuit which will satisfy design specifications. The designer is required to provide the precise description of the design in order to obtain the best possible circuit. In order to accomplish this, the designer must have a clear understanding of the theory of digital logic.

CAD tools along with Hardware Description Language (HDL) can be used to design digital logic circuits. FPGAs have become popular in recent years. These logic circuits can be implemented in FPGAs using CAD tools and HDL. These topics are covered in this book in a very simplified manner.

1.7 Evolution of Digital Logic, Microprocessors, and Microcontrollers

George Boole, an English mathematician, introduced the theory of digital logic called “Boolean Algebra” in 1847. The term “Boolean variable” is used to mean the two-valued binary digit 0 or 1. Devices such as transistors are used to represent the binary digits (bits). Many years after creation of Boolean logic, Claude Shannon of MIT, successfully implemented the theory of Boolean logic in electric circuits with relays in 1938. This provided the foundation for emergence of today’s computer.

Dr. Maurice Karnaugh of Bell Laboratories, developed Karnaugh maps in 1953 to reduce cost of designing a digital circuit by minimizing the logic gates . Due to complexity of digital logic circuits since the 70s, designers needed software to design digital circuits. Hence, HDLs evolved.

Design using Programmable Logic Devices (PLDs) became popular during the late 1970s. Note that PLD chips are capable of being programmed by the user after they are manufactured. CPLD (Complex PLD) containing several PLDs in a single chip and Field Programmable Gate Array (FPGA) were introduced. Later, Data I/O Corporation introduced the first HDL called ABEL. With the popularity of VLSI design, the first modern HDL called Verilog (no acronym), based on C-language, was introduced by Gateway Design Automation in 1985. Cadence Design Systems later acquired the rights to Verilog - XL, the HDL simulator de facto standard of Verilog simulators. Verilog was then accepted as an IEEE number 1364 in 1995.

In 1987, U.S. Department of Defense developed the HDL called VHDL (Very High Speed Integrated Circuits). VHDL is based on Ada programming language. Note that VHDL was introduced in 1983, and after going several versions, was formally accepted as an IEEE standard number 1076 in 1987.

In the past, industries developed products for specific applications. It was difficult for the users to redesign these products according to their requirements. This led to the introduction of a flexible IC chip called the FPGA. An FPGA-based circuit board would mean that it can be programmed and re-configured according to the user's specification. Altera and Xilinx are two leading manufacturers of FPGA chips. Altera, founded in 1981, introduced the first EPROM-based reprogrammable logic device in 1984. Altera became a leading manufacturer of FPGA later on. Note that Intel acquired Altera in 2015.

CPLDs are less flexible. FPGAs, on the other hand, have the far more flexibility of interconnecting various digital blocks. CPLDs are typically used in smaller applications while FPGAs are used in larger applications. Both CPLDs and FPGAs may be used in a single application. In such a system, CPLDs usually perform "glue logic" functions including booting the FPGA and controlling the reset and boot sequence of the complete circuit board. Note that glue logic is a special form of digital circuitry that allows different types of logic circuits to work together by acting as an interface between them.

With the rapid growth of the FPGA applications primarily in telecommunications and networking, FPGAs were eventually used in consumer, automative, and industrial applications during the early 2000's. FPGAs are widely used these for simulating a design. This means that designers can develop prototype hardware on FPGAs first, and then manufacture the final product on the ICs in a very short period of time.

Next, we discuss the evolution of microprocessors and microcontrollers.

Intel Corporation is generally acknowledged as the company that introduced the first microprocessor successfully into the marketplace. Its first microprocessor, the 4004, was introduced in 1971 and evolved from a development effort while making a calculator chip set. The 4004 microprocessor was the central component in the chip set, which was called the MCS-4. The other components in the set were a 4001 ROM, a 4002 RAM, and a 4003 shift register.

Shortly after the 4004 appeared in the commercial marketplace, three other general-purpose microprocessors were introduced: the Rockwell International 4-bit PPS-4, the Intel 8-bit 8008, and the National Semiconductor 16-bit IMP-16. Other companies, such as General Electric, RCA, and Viatron, also made contributions to the development of the microprocessor prior to 1971.

The microprocessors introduced between 1971 and 1972 were the first-generation systems designed using PMOS technology. In 1973, second-generation microprocessors such as the Motorola 6800 and the Intel 8080 (8-bit microprocessors) were introduced. The second-generation microprocessors were designed using NMOS technology. This technology resulted in a significant increase in instruction execution speed over PMOS and higher chip densities. Since then, microprocessors have been fabricated using a variety of technologies and designs. NMOS microprocessors such as the Intel 8085, the Zilog Z80, and the Motorola 6800/6809 were introduced based on second-generation microprocessors. A third generation HMOS microprocessor, introduced in 1978 is typically represented by the Intel 8086 and the Motorola 68000, which are 16-bit microprocessors.

During the 1980's, fourth-generation HCMOS and BICMOS (a combination of bipolar and HCMOS) 32-bit microprocessors evolved. Intel introduced the first commercial 32-bit microprocessor, the problematic Intel 432, which was eventually discontinued. Since 1985, more 32-bit microprocessors have been introduced. These include Motorola's 68020, 68030, 68040, 68060, PowerPC, Intel's 80386, 80486, the Intel Pentium family, Core Duo, and Core2 Duo microprocessors.

The performance offered by the 32-bit microprocessor is more comparable to that of superminicomputers such as Digital Equipment Corporation's VAX11/750 and VAX11/780. Intel and Motorola also introduced RISC microprocessors: the Intel 80960 and Motorola 88100/PowerPC, which had simplified instruction sets. Note that the purpose of RISC microprocessors is to maximize speed by reducing clock cycles per instruction. Almost all computations can be obtained from a simple instruction set. Note that, in order to enhance performance significantly, Intel Pentium Pro and other succeeding members of the Pentium family and Motorola 68060 are designed using a combination of RISC and CISC.

Single-chip microcomputers such as the Intel 8048 evolved during the 80's. Soon afterwards, based on the concept of single-chip microcomputers, Intel introduced the first 8-bit microcontroller---the Intel 8051. The 8051 contains a CPU, memory, I/O, A/D (Analog-to-Digital) and D/A (Digital-to-Analog) converters, timer, serial communication interface----- all in a single chip. The microcontrollers became popular during the 1980's.

Microcontrollers gained popularity over the last several years. These microcontrollers are small enough for many embedded applications, but also powerful enough to allow a lot of complexity and flexibility in the design process of an embedded system. Several billion 8-bit microcontrollers were sold during the last decade. Microchip Technology Inc., with its PIC18F microcontroller family is one of the leading manufacturer of popular microcontrollers.

Note that embedded microcontroller systems, also called embedded controllers, are designed to manage specific tasks. Once programmed, the embedded controllers can manage the functions of a wide variety of electronic products. In embedded applications, the microcontrollers are embedded in the host system, their presence and operation are basically hidden from the host system.

Typical embedded control applications include office automation products such as copiers, laser products, fax machines, and consumer electronics such as microwave ovens. Applications such as printers typically utilize a microcontroller. The microcontrollers are ideal for these types of applications. Note that the Personal Computer interfaced to the printer is the host, and the microcontroller is embedded (hidden) in the printer controller.

1.8 A Typical Application of a Digital System such as a Microcontroller

To put basic digital concepts such as binary numbers into perspective, it is important to explore a simple application.

As an example, consider a simple application using a digital system such as a microcontroller as shown in Figure 1.21. Suppose that it is necessary to maintain the temperature of a furnace to a desired level to maintain the quality of a product. Assume that the designer has decided to control this temperature by adjusting the fuel. This can be accomplished using a typical microcontroller such as the PIC18F along with the interfacing components as follows. Temperature is an analog (continuous) signal. It can be measured by a temperature-sensing (measuring) device such as a thermocouple. The thermocouple provides the measurement in millivolts (mV) equivalent to the temperature.

Since microcontrollers only understand binary numbers (0's and 1's), each analog mV signal must be converted to a binary number using the microcontroller's on-chip analog-to-digital (A/D) converter. Note that the PIC18F contains an on-chip A/D converter. The PIC18F does not include an on-chip digital-to-analog (D/A) converter. However, the D/A converter chip can be interfaced to the PIC18F externally.

First, the millivolt signal is amplified by a mV/V amplifier to make the signal compatible for A/D conversion. A microcontroller such as the PIC18F can be programmed to solve an equation with the furnace temperature as an input. This equation compares the temperature measured with the temperature desired which can be entered into the microcontroller using the keyboard. The output of this equation will provide the appropriate opening and closing of the fuel valve to maintain the appropriate temperature. Since this output is computed by the microcontroller, it is a binary number. This binary output must be converted into an analog current or voltage signal.

The D/A (digital-to-analog) converter chip inputs this binary number and converts it into an analog current (*I*). This signal is then input into the current/pneumatic (*I/P*) transducer for opening or closing the fuel input valve by air pressure to adjust the fuel to the furnace. The furnace temperature desired can thus be achieved. Note that a transducer converts one form of energy (electrical current in this case) to another form (air pressure in this example).

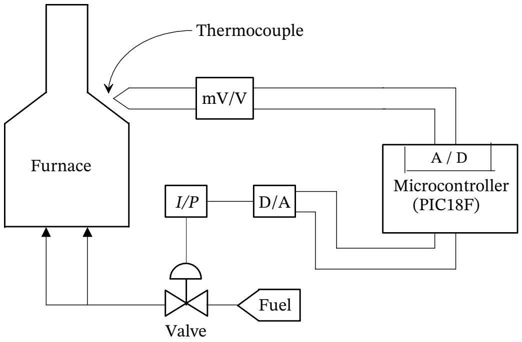


FIGURE 1.21 Furnace temperature control