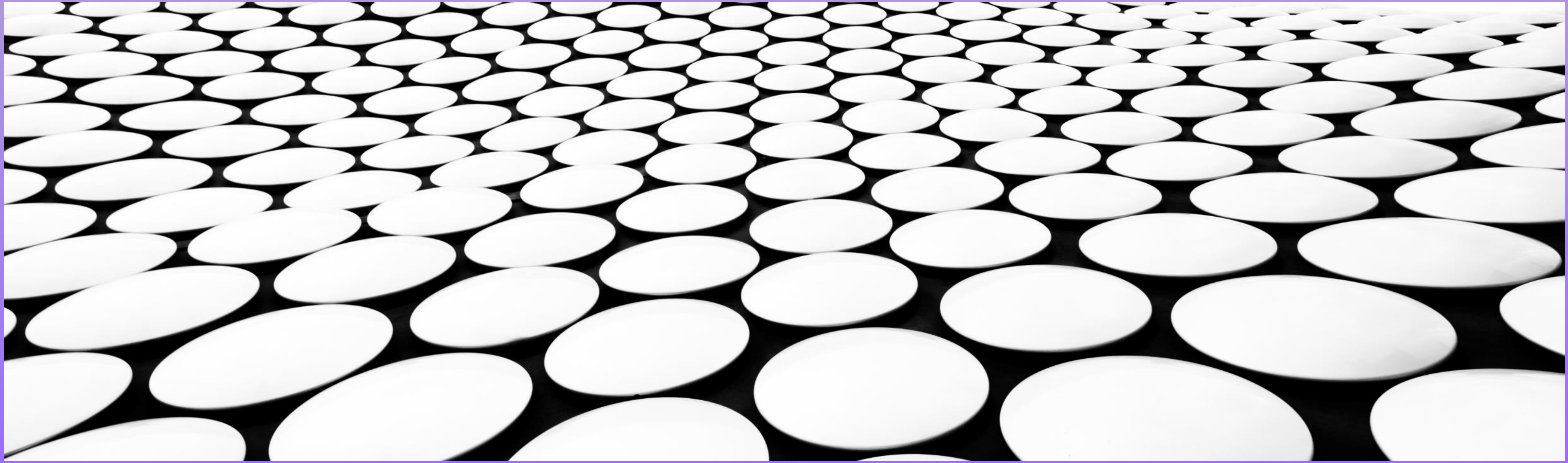
MICROPROCESSOR ARCHITECTURE



UNIT 1 SYLLABUS

- **Microprocessor, microcomputers, and Assembly Language:** Microprocessor, Microprocessor Instruction Set and Computer Languages, From Large Computers to Single-Chip Microcontrollers, Applications.
- **Microprocessor Architecture and Microcomputer System:** Microprocessor Architecture and its operation's, Memory, I/O Devices, Microcomputer System, Logic Devices and Interfacing, Microprocessor-Based System Application.
- **8085 Microprocessor Architecture and Memory Interface:** Introduction, 8085 Microprocessor unit, 8085-Based Microcomputer, 8085 Machine Cycles & Bus Timings, Memory Interfacing, Interfacing the 8155 Memory Segment,
- **Illustrative Example:** Designing Memory for the MCTS Project, Testing and Troubleshooting Memory Interfacing Circuit, 8085- Based Single-Board microcomputer.

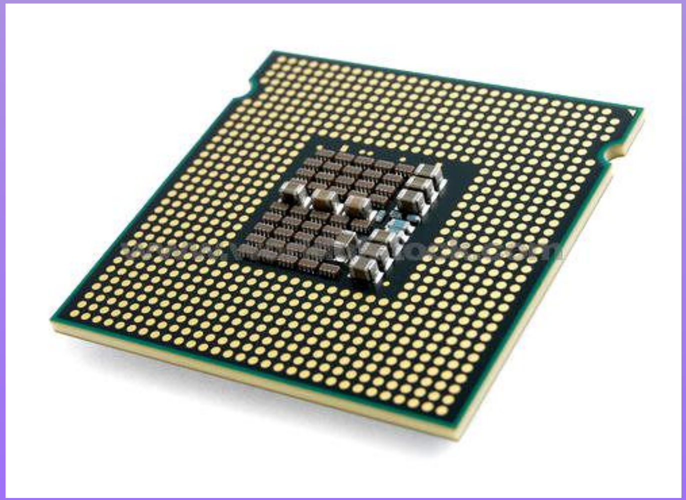
CHAPTER 1
MICROPROCESSOR, MICROCOMPUTERS, AND ASSEMBLY LANGUAGE

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

Microprocessor: silicon chip which includes ALU, register circuits & control circuits.

Microcontroller: silicon chip which includes microprocessor, memory & I/O in a single package.

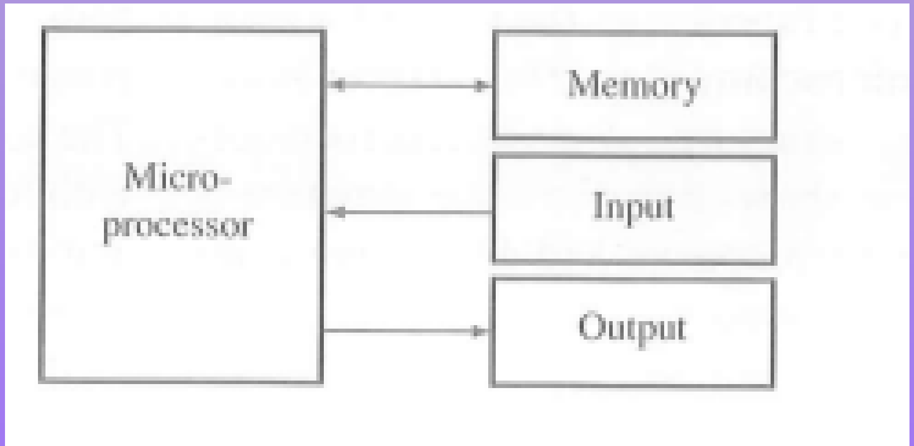
- Microprocessor communicates and operates in the binary numbers 0 and 1 called as bits
- Each microprocessors has a fixed set of instructions in the form of binary patterns called a machine language.
- Binary instructions are given abbreviated names called mnemonics which form assembly language for a given microprocessor.



MICROPROCESSOR

- The microprocessor is a programmable device that takes in numbers, performs on them arithmetic or logical operations according to the program stored in memory and then produces other numbers as a result
- Programmable device: The microprocessor can perform different sets of operations on the data it receives depending on the sequence of instructions supplied in the given program
- Instructions: Each microprocessor is designed to execute a specific group of operations. This group of operations is called an instruction set. This instruction set defines what the microprocessor can and cannot do
- Memory is the location where information is kept while not in current use.

- A typical programmable machine can be represented with four components: microprocessor, memory, input, and output.
- The physical components of this system are called hardware. A set of instructions written for the microprocessor to perform a task is called a program, and a group of programs is called software.
- The microprocessor applications are classified primarily in two categories: reprogrammable systems and embedded systems.
- In reprogrammable systems, such as microcomputers, the microprocessor is used for computing and data processing.
- In embedded systems, the microprocessor is a part of a final product and is not available for reprogramming to the end user.



BINARY DIGITS

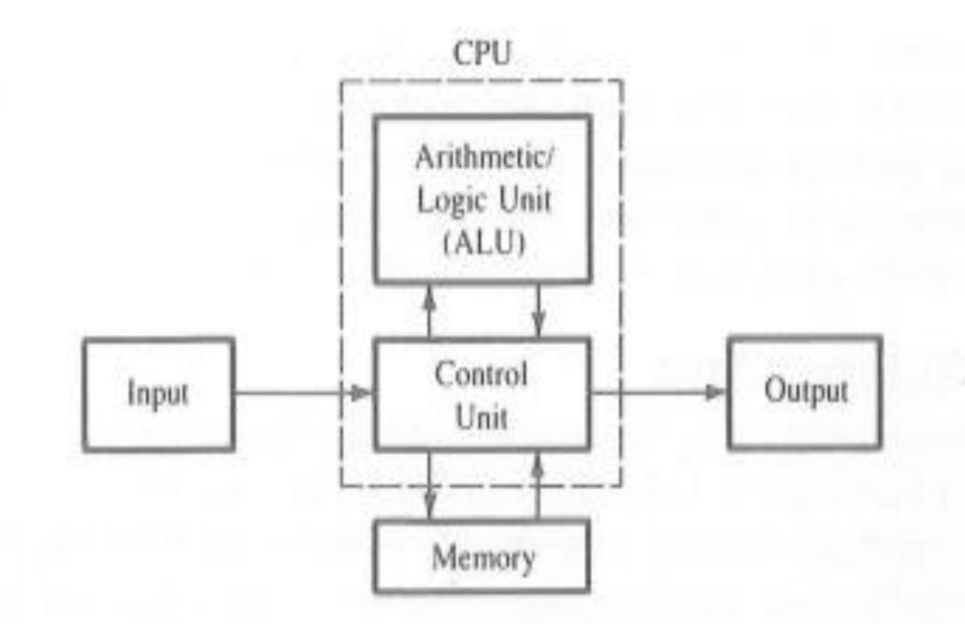
- The microprocessor operates in binary digits, 0 and 1, also known as bits.
- Bit is an abbreviation for the term binary digit.
- Each microprocessor recognizes and processes a group of bits called the word, and microprocessors are classified according to their word length.
- For example, a processor with an 8-bit word is known as an 8-bit microprocessor, and a processor with a 32-bit word is known as a 32-bit microprocessor.

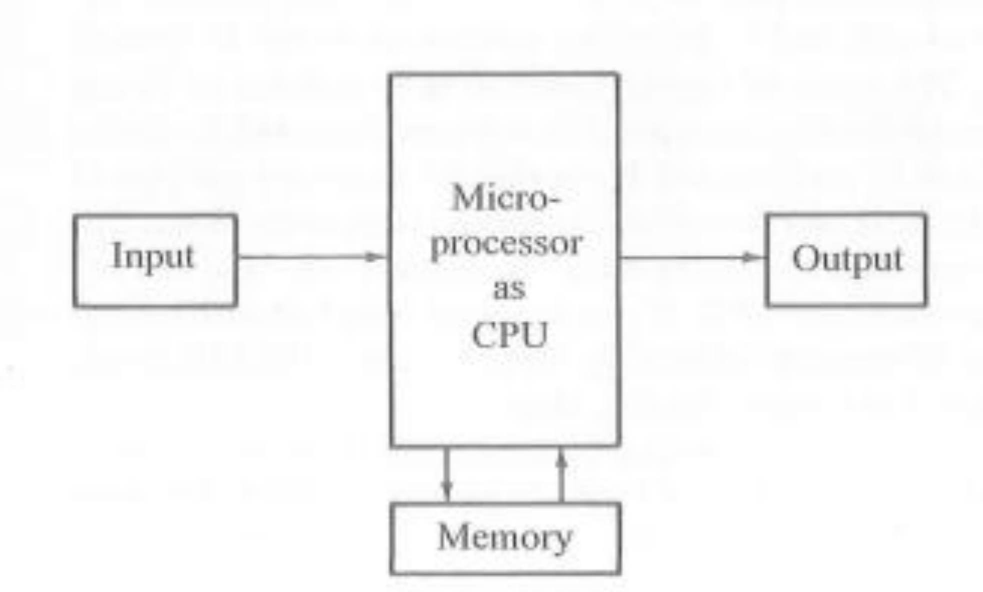
MICROPROCESSOR AS A PROGRAMMABLE DEVICE

- Microprocessor is programmable means it can be instructed to perform given tasks within its capability.
- Today's Microprocessors is designed to understand and execute many binary instructions.
- Memory
- Input/Output

MICROPROCESSOR AS AN CPU (MPU)

- A computer with microprocessor as its CPU is known as Microcomputer.





ORGANIZATION OF MICROPROCESSOR BASED SYSTEM

- Internally, the microprocessor can be divided into 3 main units.

- The Arithmetic/Logic Unit (ALU)
- The Control Unit.
- An array of registers for holding data while it is being manipulated.

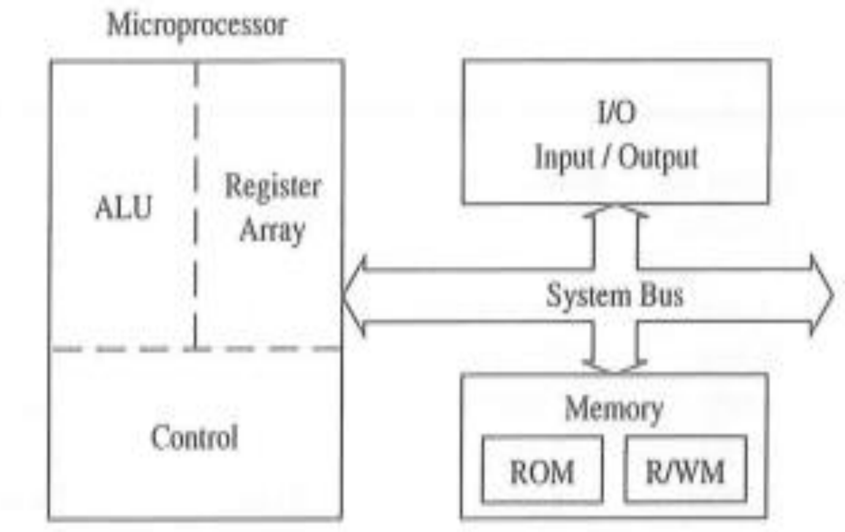


FIGURE 1.3
Microprocessor-Based System with Bus Architecture

Arithmetic Logic Unit

- This is the area of Microprocessor where various computing functions are performed on data.
- The ALU performs operations such as addition, subtraction and logic operations such as AND, OR and exclusive OR.

Control Unit

- The Control Unit Provides the necessary timing and control signals to all the operations in the Microcomputer
- It controls the flow of data between the Microprocessor and Memory and Peripherals.

The Control unit performs 2 basic tasks

- Sequencing
- Execution

Register Array

- These are storage devices to store data temporarily.
- There are different types of registers depending upon the Microprocessors.
- These registers are primarily used to store data temporarily during the execution of a program and are accessible to the user through the instructions.

MEMORY

- Memory stores such binary information as instructions and data, and provides that information to the microprocessor whenever necessary.
- To execute programs, the micro- processor reads instructions and data from memory and performs the computing operations in its ALU section.
- Results are either transferred to the output section for display or stored in memory for later use.
- The memory block has two sections: Read-Only memory (ROM) and Read/Write memory (R/WM), popularly known as Random-Access memory (RAM).

- The ROM is used to store programs that do not need alterations.
- Programs stored in the ROM can only be read; they cannot be altered.
- The Read/Write memory (R/WM) is also known as user memory. It is used to store user programs and data
- The information stored in this memory can be easily read and altered.

I/O (INPUT/OUTPUT)

- The third component of a microprocessor-based system is I/O (input/output); it communicates with the outside world.
- I/O includes two types of devices: input and output; these I/O devices are also known as peripherals.
- The input devices such as a keyboard, switches, transfer binary information (data and instructions) from the outside world to the microprocessor.
- The output devices transfer data from the microprocessor to the outside world. Eg:light emitting diodes (LEDs)

SYSTEM BUS

The system bus is a communication path between the microprocessor and peripherals; it is nothing but a group of wires to carry bits.All peripherals (and memory) share the same bus; however, the microprocessor communicates with only one peripheral at a time. The timing is provided by the control unit of the microprocessor.

WORKING OF MICROPROCESSOR

Fetch → Decode → Execute

MICROPROCESSOR INSTRUCTION SET AND COMPUTER LANGUAGES
MACHINE LANGUAGE

- The number of bits that form the “word” of a microprocessor is fixed for that particular processor.
- These bits define a maximum number of combinations.
- For example an 8-bit microprocessor can have at most $2^8 = 256$ different combinations.
- Instructions are made up of one word or several words.
- The set of instructions designed into the machine makes up its machine language a binary language

THE 8085 MACHINE LANGUAGE

- The 8085 is a microprocessor with 8-bit word length
- Its instruction set (or language) is designed by using various combinations of these eight bits.
- An instruction is a binary pattern entered through an input device in memory to command the microprocessor to perform that specific function.
- The 8085 microprocessor has 246 such bit patterns, amounting to 74 different instructions
- These 74 different instructions are called its instruction set.
- This binary language with a predetermined instruction set is called the 8085 machine language.

ASSEMBLY LANGUAGE

- Entering the instructions using hexadecimal is quite easier than entering the binary combinations.
- However, it still is difficult to understand what a program written in hexadecimal does. So, each company defines a symbolic code for the instructions. These codes are called “mnemonics”.
- The mnemonic for each instruction is usually a group of letters that suggest the operation performed.
- The complete set of 8085 mnemonics is called the 8085 assembly language, and a program written in these mnemonics is called an assembly language program
- Machine language and assembly language are microprocessor-specific and are both considered low-level languages.
- It is important to remember that a machine language and its associated assembly language are completely machine dependent.
- In other words, they are not transferable from one microprocessor to a different one.

- A computer is a binary machine
- The commonly used code is known as ASCII-American Standard Code for Information Interchange. It is a 7-bit code with 128 (2^7) combinations, and each combination from 00H to 7FH
- For example, hexadecimal 30H to 39H represent 0 to 9 decimal digits, 41H to 5AH represent capital letters A through Z,
- When the key "9" is pressed on an ASCII key- board, the computer receives 39H in binary, called an ASCII character, and the system program translates ASCII characters into appropriate binary numbers.
- Extended ASCII is an 8-bit code that provides 256 (2^8) combinations; the additional 128 combinations are assigned to various graphics characters.

ASSEMBLY LANGUAGE PROGRAM EXECUTION

1. Write the instructions in mnemonics obtained from the instruction set supplied by the manufacturer.

2. Find the hexadecimal machine code for each instruction by searching through the set of instructions.

3. Enter (load) the program in the user memory in a sequential order by using the Hex keyboard as the input device.

4. Execute the program by pressing the Execute key. The answer will be displayed by the LEDs.

This procedure is called either manual or hand assembly.

ASSEMBLER

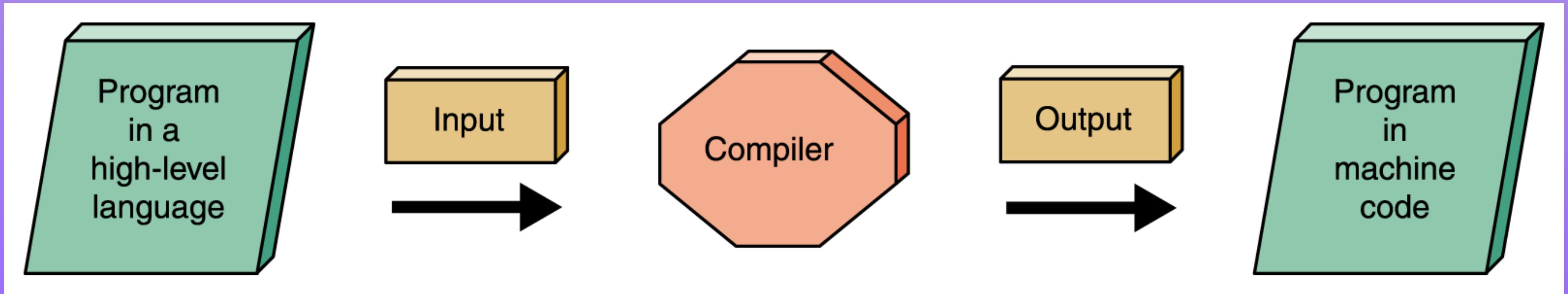
- The assembler is a program that translates the mnemonics entered by the ASCII keyboard into the corresponding binary machine codes of the microprocessor.
- Each microprocessor has its own assembler because the mnemonics and machine codes are specific to the microprocessor being used, and each assembler has rules that must be followed by the programmer.

HIGH LEVEL LANGUAGE

- Programming languages that are intended to be machine-independent are called high-level languages.
- eg: BASIC, PASCAL, C, C++, and Java,
- Instructions written in these languages are known as statements rather than mnemonics.
- How are words in English converted into the binary languages of different microprocessors? →Through another program called either a compiler or an interpreter.
- The programs accept English-like statements as their input, called the source code.
- The compiler or interpreter then translates the source code into the machine language compatible with the microprocessor being used in the system.
- This translation in the machine language is called the object
- Each microprocessor needs its own compiler or an interpreter for each high-level language

COMPILERS

- The primary difference between a compiler and an interpreter lies in the process of generating machine code
- The compiler reads the entire program first and translates it into the object code that is executed by the microprocessor.
- . Compilers are generally used in such languages as FORTRAN, PASCAL, C, and C++.

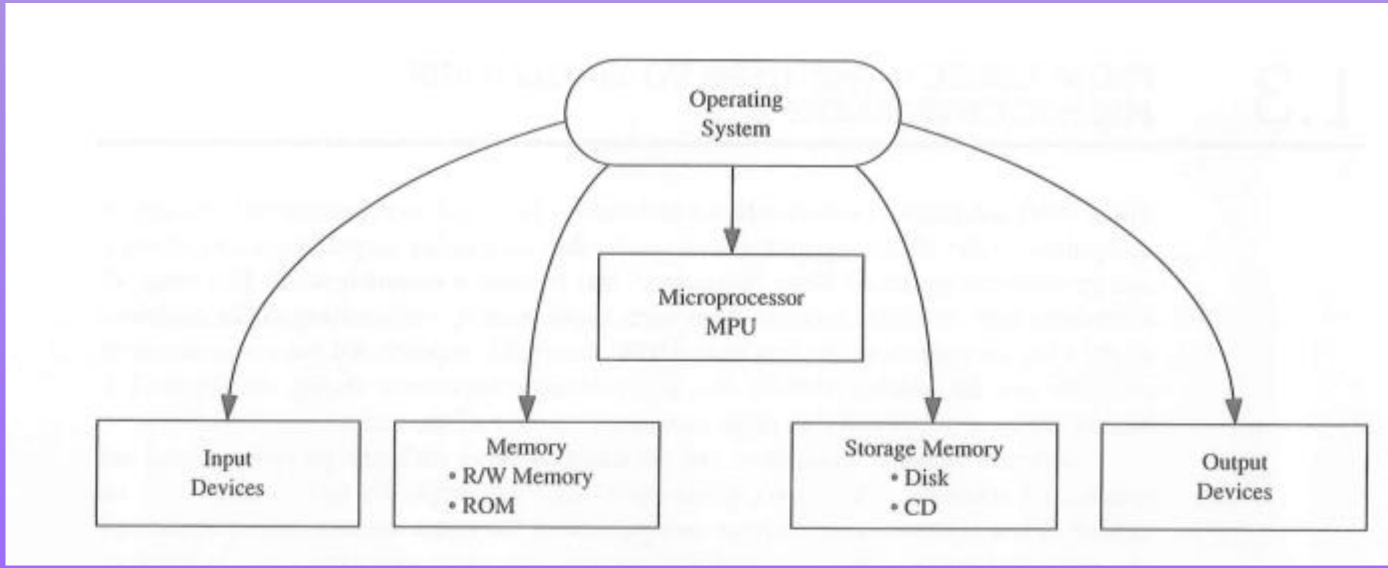


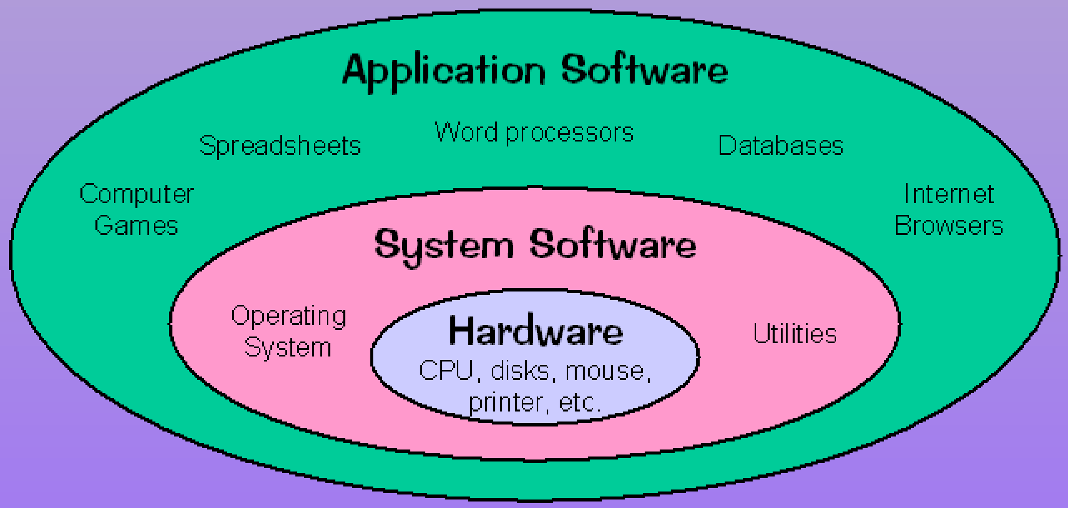
INTERPRETERS

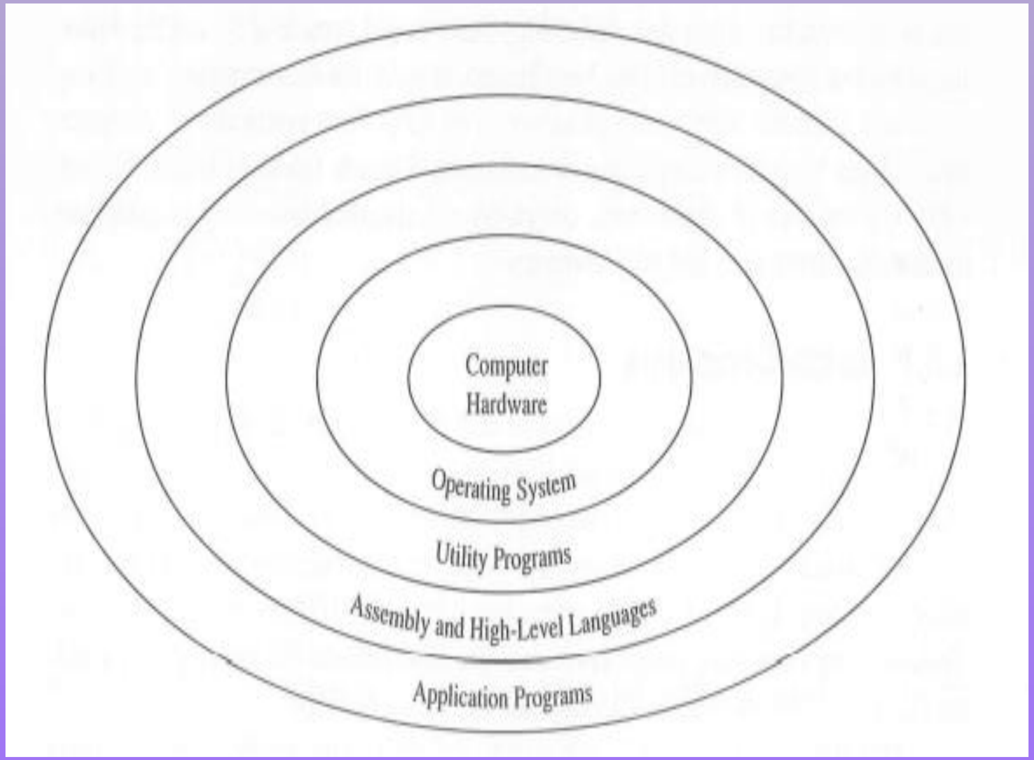
- On the other hand, the interpreter reads one instruction at a time, produces its object code (a sequence of machine actions), and executes the instruction before reading the next instruction.
- An interpreter translates a statement and then immediately executes the statement
- M-Basic is a common example of an interpreter for BASIC language
- Compilers and interpreters require large memory space because an instruction in English requires several machine codes to translate it into binary.

OPERATING SYSYEMS

- The interaction between the hardware and the software is managed by a set of programs called an operating system of a computer.
- The computer transfers information constantly between memory and various peripherals
- It also stores programs on disk.




Application Software
Spreadsheets
Word processors
Databases
Computer Games
Internet Browsers
System Software
Operating System
Hardware
CPU, disks, mouse, printer, etc.
Utilities


Computer Hardware
Operating System
Utility Programs
Assembly and High-Level Languages
Application Programs

MICROPROCESSOR ARCHITECTURE AND MICROCOMPUTER SYSTEM

- The microprocessor is a programmable digital device, designed with registers, flip-flops, and timing elements. The process of data manipulation and communication is determined by the logic design of the microprocessor, called the architecture.
- The microprocessor can be programmed to perform functions on given data by selecting necessary instructions from its set. These instructions are given to the microprocessor by writing them into its memory. Writing (or entering) instructions and data is done through an input device such as a keyboard.
- The microprocessor reads or transfers one instruction at a time, matches it with its instruction set, and performs the data manipulation indicated by the instruction. The result can be stored in memory or sent to such output devices as LEDS
- All the various functions performed by the microprocessor can be classified in three general categories:
  - ✓ Microprocessor-initiated operations
  - ✓ Internal operations
  - ✓ Peripheral (or externally initiated) operations
- To perform these functions, the microprocessor requires a group of logic circuits and a set of signals called control signals.

MICROPROCESSOR-INITIATED OPERATIONS AND 8085 BUS ORGANIZATION

- The MPU performs primarily four operations:

  1. Memory Read: Reads data (or instructions) from memory.
  2. Memory Write: Writes data (or instructions) into memory.
  3. I/O Read: Accepts data from input devices.
  4. I/O Write: Sends data to output devices.

- To communicate with a peripheral (or a memory location), the MPU needs to perform the following steps:
  - Step 1: Identify the peripheral or the memory location (with its address).
  - Step 2: Transfer binary information (data and instructions).
  - Step 3: Provide timing or synchronization signals.

- The 8085 MPU performs these functions using three sets of communication lines called buses: the address bus, the data bus, and the control bus.

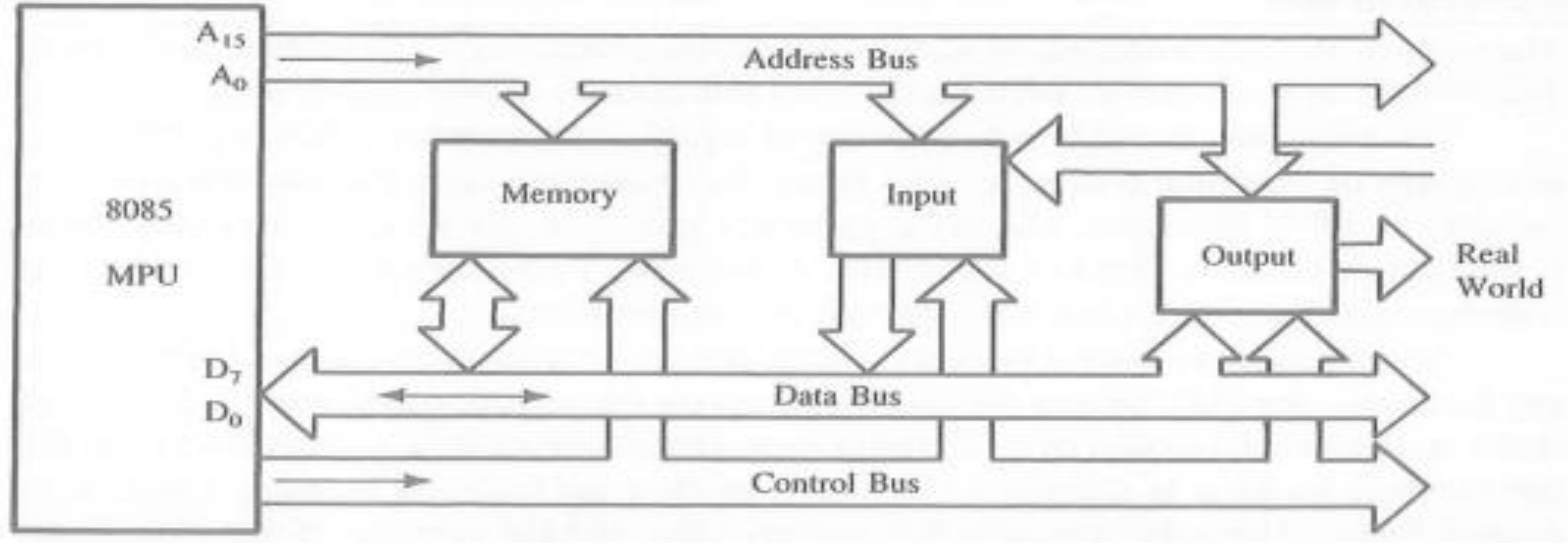


FIGURE 3.1
The 8085 Bus Structure

- ADDRESS BUS
- The address bus is a group of 16 lines generally identified as A_0 to A_{15}.
- The address bus is unidirectional: bits flow in one direction from the MPU to peripheral devices.
- The MPU uses the address bus to perform the first function: identifying a peripheral or a memory location (Step 1).
- In a computer system, each peripheral or memory location is identified by a binary number, called an address, and the address bus is used to carry a 16-bit address.
- The number of address lines is arbitrary; it is determined by the designer of a microprocessor based on such considerations as availability of pins and intended applications of the processor. For example, the Intel 8088 processor has 20 and the Pentium processor has 32 address lines.

- DATA BUS
- The data bus is a group of eight lines used for data flow
- These lines are bidirectional data flow in both directions between the MPU and memory and peripheral devices. The MPU uses the data bus to perform the second function: transferring binary information (Step 2).
- The eight data lines enable the MPU to manipulate 8-bit data ranging from 00 to FF. The largest number that can appear on the data bus is 1111111. The 8085 is known as an 8-bit microprocessor.
- Microprocessors such as the Intel 8086, Zilog Z8000, and Motorola 68000 have 16 data lines; thus they are known as 16- bit microprocessors. The Intel 80386/486 have 32 data lines; thus they are classified as 32-bit microprocessors.

- CONTROL BUS
- The control bus is comprised of various single lines that carry synchronization signals. The MPU uses such lines to perform the third function: providing timing signals (Step 3).
- The term bus, in relation to the control signals, is somewhat confusing. These are not groups of lines like address or data buses, but individual lines that provide a pulse to indicate an MPU operation.
- The MPU generates specific control signals for every operation (such as Memory Read or I/O Write) it performs.
- To communicate with a memory-for example, to read an instruction from a memory location the MPU places the 16-bit address on the address bus. The address on the bus is decoded by an external logic circuit, and the memory location is identified.
- The MPU sends a pulse called Memory Read as the control signal. The pulse activates the memory chip, and the contents of the memory lo- cation (8-bit data) are placed on the data bus and brought inside the microprocessor.

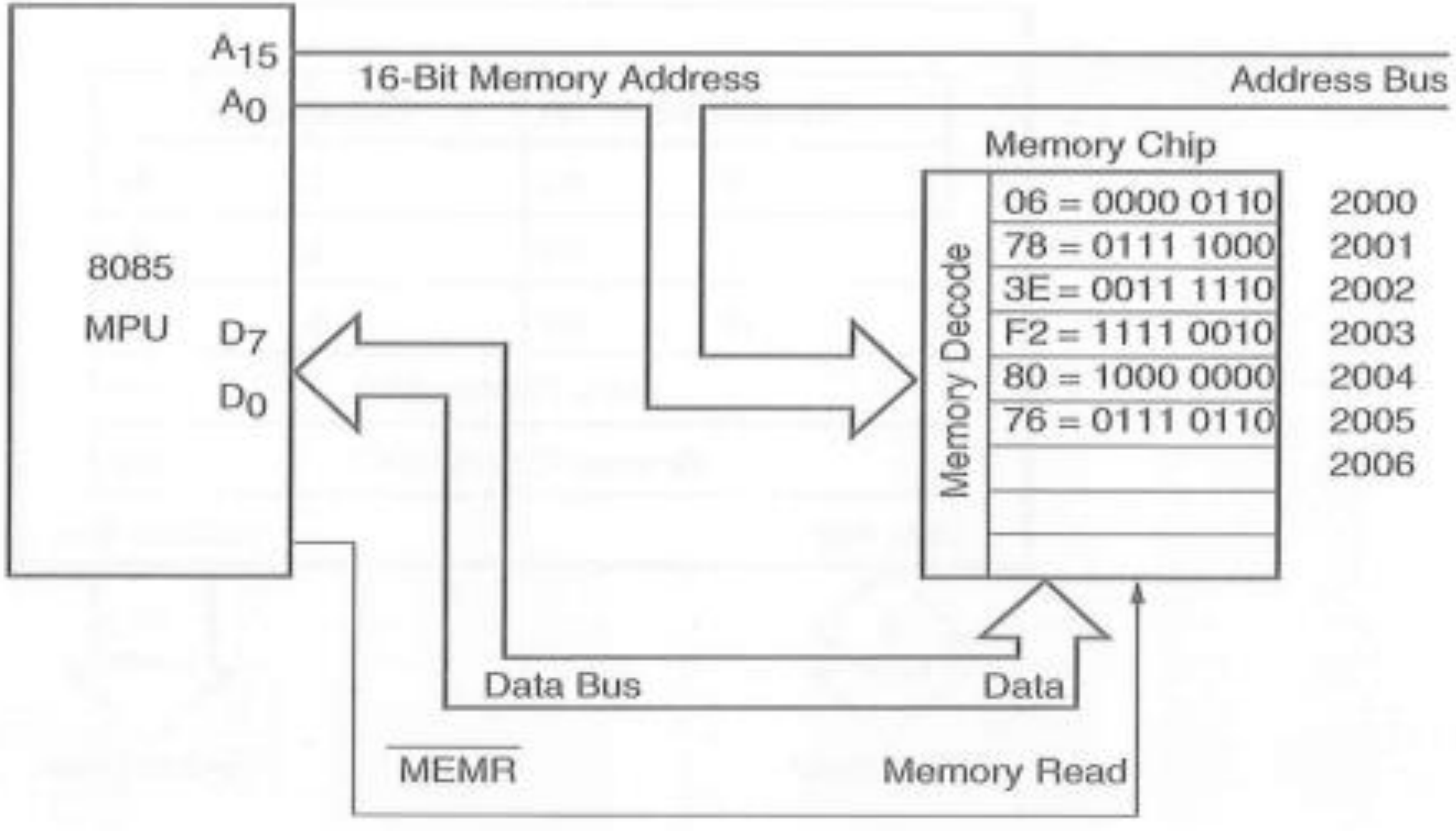


FIGURE 3.2
Memory Read Operation

INTERNAL DATA OPERATION AND 8085 REGISTERS

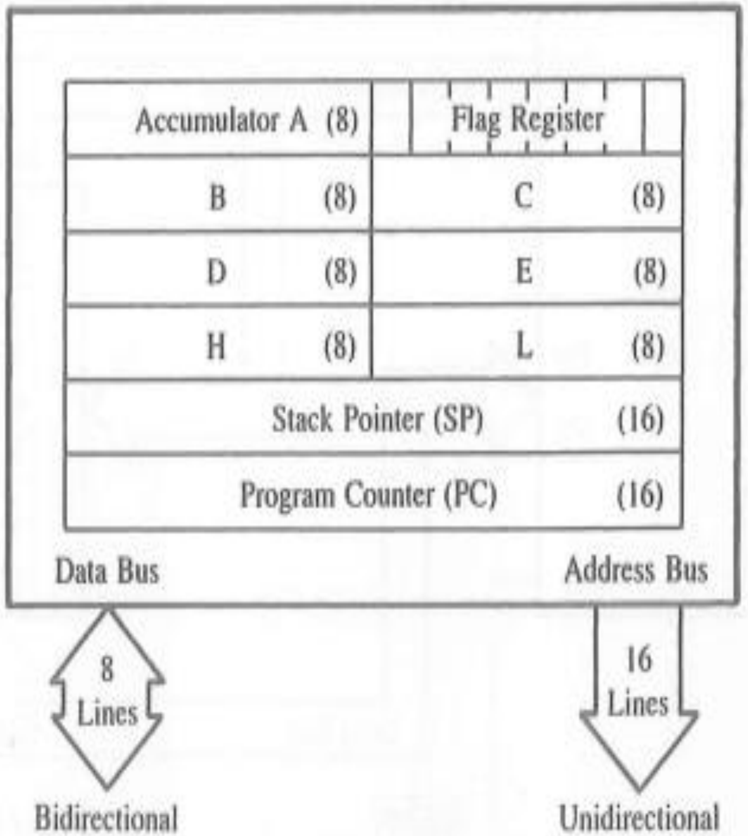
- The internal architecture of the 8085 microprocessor determines how and what operations can be performed with the data. These operations are:

  1. Store 8-bit data.
  2. Perform arithmetic and logical operations.
  3. Test for conditions.
  4. Sequence the execution of instructions.
  5. Store data temporarily during execution in the defined R/W memory locations called the stack.

- To perform these operations, the microprocessor requires registers, an arithmetic/ logic unit (ALU) and control logic, and internal buses (paths for information flow).

2000	06	MVI B, 76H
2001	78	
2002	3E	MVI A, F2H
2003	F2	
2004	80	ADD B
2005	76	HLT

FIGURE 3.3
The 8085 Programmable Registers



- When the user enters the memory address 2000H and pushes the execute key of the trainer, the processor places the address 2000H in the program counter (PC).

1. The program counter is a 16-bit register that performs the fourth operation in the list: sequencing the execution of the instructions. When the processor begins execution, it places the address 2000H on the address bus and increments the address in the PC to 2001 for the next operation. It brings the code 06, interprets the code, places the ad- dress 2001H on the address bus, and then gets byte 78H and increments the address in PC to 2002H. The processor repeats the same process for the next instruction, MVIA, F2H.

2. When the processor executes the first two instructions, it uses register B to store 01H and A to store 02H in binary (Operation 1).

3. When the processor executes the instruction ADD B in the ALU (Operation 2), it adds 01H to 02H, resulting in the sum 03H It replaces 02H by 03H in A

5. The fifth operation deals with the concept of the stack. The stack pointer is a 16-bit register used as a memory pointer to identify the stack, part of the R/W memory de- fined and used by the processor for temporary storage of data during the execution

- When the user enters the memory address 2000H and pushes the execute key of the trainer, the processor places the address 2000H in the program counter (PC).

1. The program counter is a 16-bit register that performs the fourth operation in the list: sequencing the execution of the instructions. When the processor begins execution, it places the address 2000H on the address bus and increments the address in the PC to 2001 for the next operation. It brings the code 06, interprets the code, places the ad- dress 2001H on the address bus, and then gets byte 78H and increments the address in PC to 2002H. The processor repeats the same process for the next instruction, MVIA, F2H.

2. When the processor executes the first two instructions, it uses register B to store 78H and A to store F2H in binary (Operation 1).

3. When the processor executes the instruction ADD B in the ALU (Operation 2), it adds 78H to F2H, resulting in the sum 16AH (78H+ F2H = 16AH). It replaces F2H by 6AH in A and sets the Carry flag as described next.

4. In our example, the addition operation generates a carry because the sum is larger than the size of the accumulator (8 bits). To indicate the carry, the processor sets the flip- flop called Carry (CY flag) to 1 and places logic 1 in the flag register at the designated bit position for the carry.

5. The fifth operation deals with the concept of the stack. The stack pointer is a 16-bit register used as a memory pointer to identify the stack, part of the R/W memory de- fined and used by the processor for temporary storage of data during the execution

PERIPHERAL OR EXTERNALLY INITIATED OPERATION

External devices (or signals) can initiate the following operations, for which individual pins on the microprocessor chip are assigned: Reset, Interrupt, Ready, Hold.

- Reset: When the reset pin is activated by an external key (also called a reset key), all internal operations are suspended and the program counter is cleared (it holds 0000H). Now the program execution can again begin at the zero memory address.
- Interrupt: The microprocessor can be interrupted from the normal execution of instructions and asked to execute some other instructions called a service routine (for example, emergency procedures). The microprocessor resumes its operation after completing the service routine
- Ready: The 8085 has a pin called READY. If the signal at this READY pin is low, the microprocessor enters into a Wait state. This signal is used primarily to synchronize slower peripherals with the microprocessor.
- Hold: When the HOLD pin is activated by an external signal, the microprocessor relinquishes control of buses and allows the external peripheral to use them. For example, the HOLD signal is used in Direct Memory Access (DMA) data transfer

MEMORY

- Memory is an essential component of a microcomputer system; it stores binary instructions and data for the microprocessor. There are various types of memory, which can be classified in two groups: prime (or main) memory and storage memory.
- The R/W memory is made of registers, and each register has a group of flip-flops or field-effect transistors that store bits of information; these flip-flops are called memory cells. The number of bits stored in a register is called a memory word; memory devices (chips) are available in various word sizes. The user can use this memory to hold programs and store data.
- On the other hand, the ROM stores information permanently in the form of diodes; the group of diodes can be viewed as a register. In a memory chip, all registers are arranged in a sequence and identified by binary numbers called memory addresses.

- To communicate with memory, the MPU should be able to
  - ✓ select the chip,
  - ✓ identify the register, and
  - ✓ read from or write into the register.

- The MPU uses its address bus to send the address of a memory register and uses the data bus and control lines to read from (as shown in Figure 3.2) or write into that register.

FLIP-FLOP OR LATCH

- What is memory? It is a circuit that can store bits-high or low, generally voltage levels or capacitive charges representing 1 and 0.
- A flip-flop or a latch is a basic element of memory. To write or store a bit in the latch, we need an input data bit (D_{IN}) and an enable signal (EN), as shown in Figure 3.4(a).
- In this latch, the stored bit is always available on the output line D_{OUT}
- To avoid unintentional change in the input and control the availability of the output, we can use two tri-state buffers on the latch, as shown in Figure 3.4(b).
- Now we can write into the latch by enabling the input buffer and read from it by enabling the output buffer. Figure 3.4(b) shows the Write signal as WR and the Read signal as RD; these are active low signals indicated by the bar.
- This latch, which can store one binary bit, is called a memory cell. Figure 3.5(a) shows four such cells or latches grouped together; this is a register, which has four input lines and four output lines and can store four bits; thus the size of the memory word is four bits. The size of this register is specified either as 4-bit or 1 x 4-bit, which indicates one register with four cells or four I/O lines. Figures 3.5(b) and (c) show simplified block diagrams of the 4-bit register.

- In Figure 3.6, four registers with eight cells (or an 8-bit memory word) are arranged in a sequence. To write into or read from any one of the registers, a specific register should be identified or enabled. This is a simple decoding function; a 2-to-4 decoder can perform that function. However, two more input lines A_0 and A_1, called address lines, are required to the decoder.
- These two input lines can have four different bit combinations (00, 01, 10, 11), and each combination can identify or enable one of the registers named as Register 0 through Register 3. Thus the Enable signal of the flip-flops in Figure 3.5 is replaced by two address lines in Figure 3.6. Figure 3.6(a) has 8-bit registers and Figure 3.6(b) has two chips with 4-bit registers.
- This is an illustration of how smaller word size chips can be connected to make up an 8-bit word memory size. Now we can expand the number of registers.
- If we have eight registers on one chip, we need three address lines, and if we have 16 registers, we need four address lines.

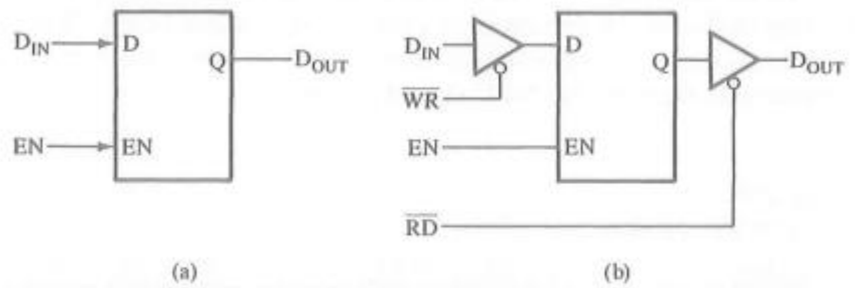


FIGURE 3.4
Latches as Storage Element: Basic Latch (a) and Latch with Two Tri-State Buffers (b)

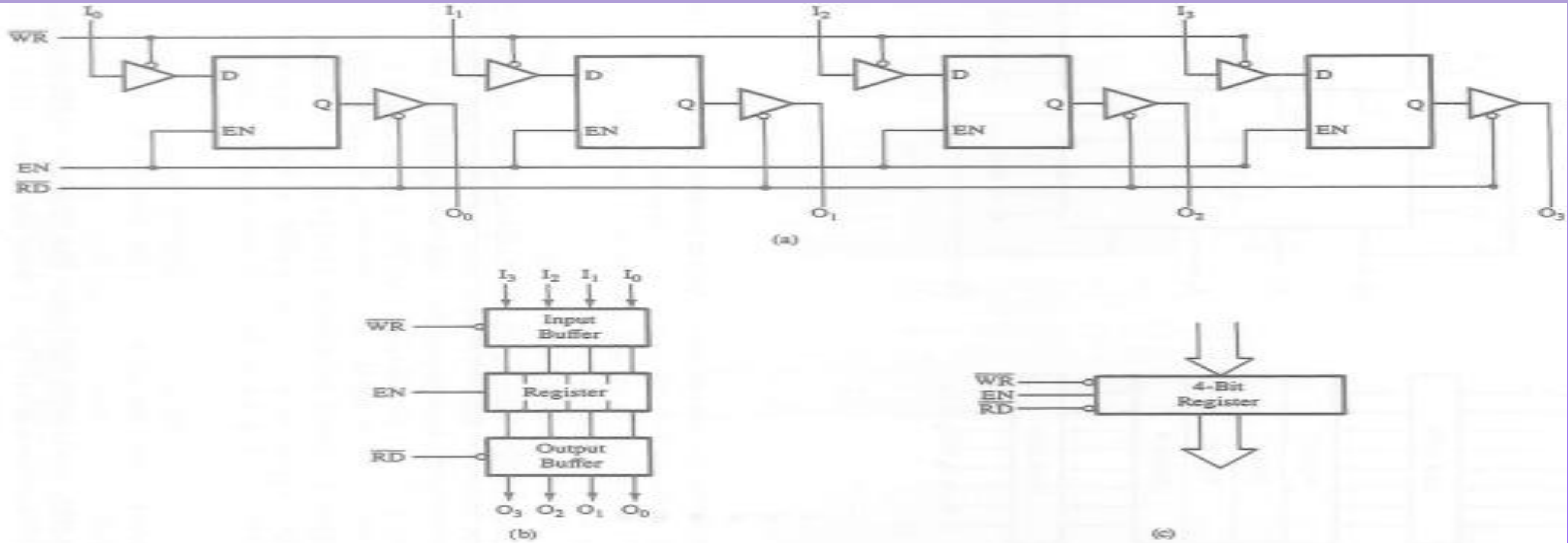


FIGURE 3.5
Four Latches as a 4-Bit Register (a) and Block Diagrams of a 4-Bit Register (b and c)

- for example, two chips with four registers each. We have a total of eight registers; therefore, we need three address lines, but one line should be used to select between the two chips. Figure 3.7(b) shows two memory chips, with an additional signal called Chip Select (CS), and A_2 (with an inverter) is used to select between the chips. When A_2 is 0 (low), chip M_1, is selected, and when A_2 is 1 (high), chip M_2 is selected.
- The addresses on A_0 and A_1 will determine the registers to be selected; thus, by combining the logic on A_2, A_1 and A_0, the memory addresses range from 000 to 111. The concept of the Chip Select signal gives us more flexibility in designing chips and allows us to expand memory size by using multiple chips.
- Assume that we have available four address lines and two memory chips with four registers each as before. Four address lines are capable of identifying 16 (2^4) registers; however, we need only three address lines to identify eight registers. What should we do with the fourth line? One of the solutions is shown in Figure 3.8. Memory chip M_1 is selected when , A_3 and , A_2 are both 0; therefore, registers in this chip are identified with the addresses ranging from 0000 to 0011 (0 to 3). Similarly, the addresses of memory chip , M_2 range from 1000 to 1011 (8 to B); this chip is selected only when , A_3 is 1 and , A_2 is 0. In this example, we need three lines to identify eight registers: two for registers and one for Chip Select. However, we used the fourth line for Chip Select also. This is called complete or absolute decoding.

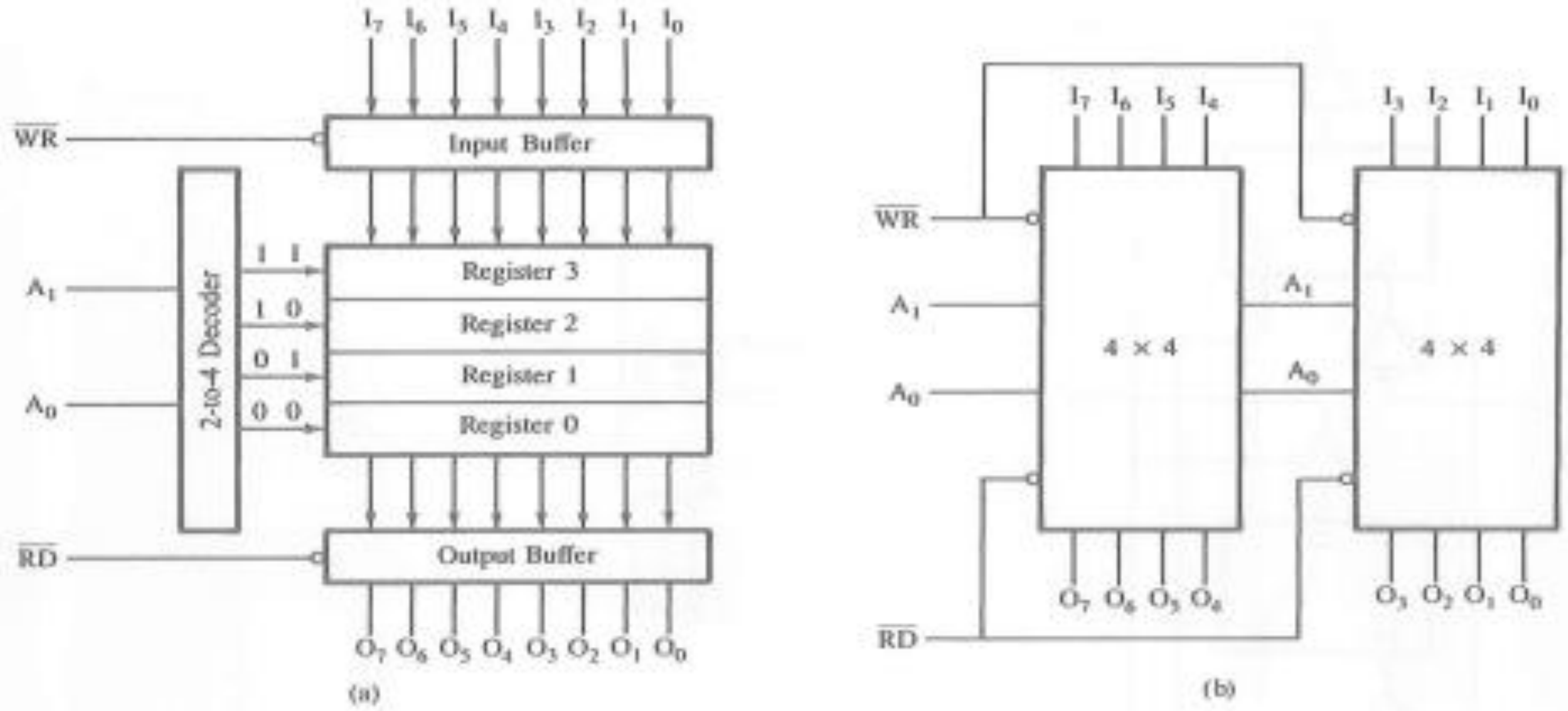


FIGURE 3.6
4 × 8-Bit Register

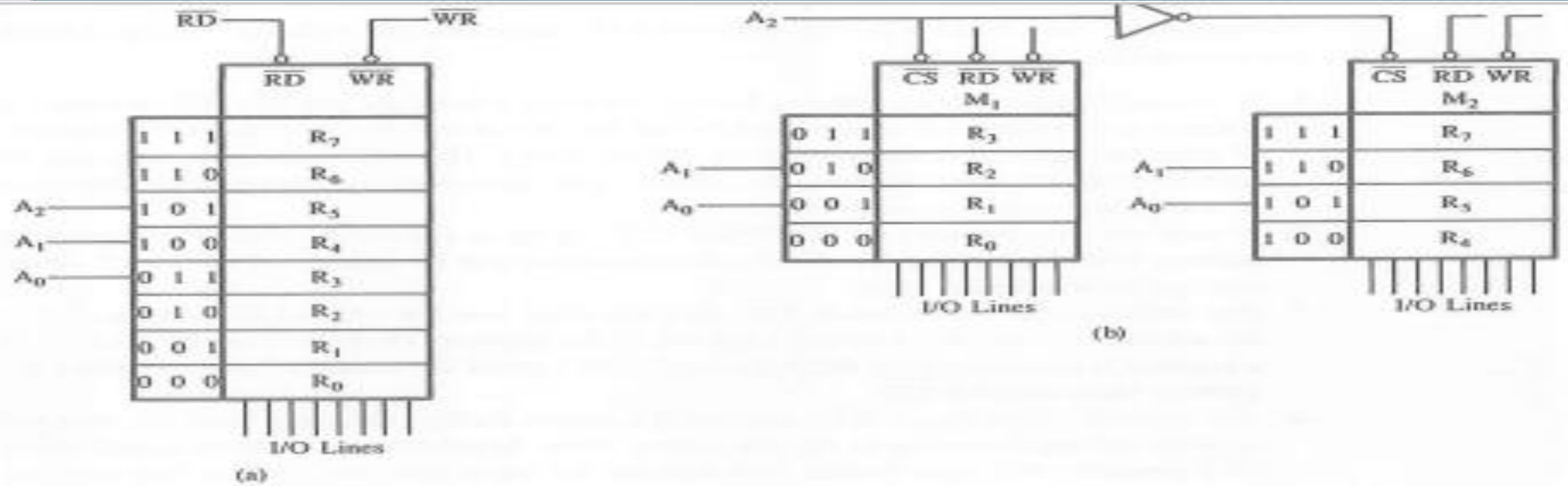


FIGURE 3.7
Two Memory Chips with Four Registers Each and Chip Select

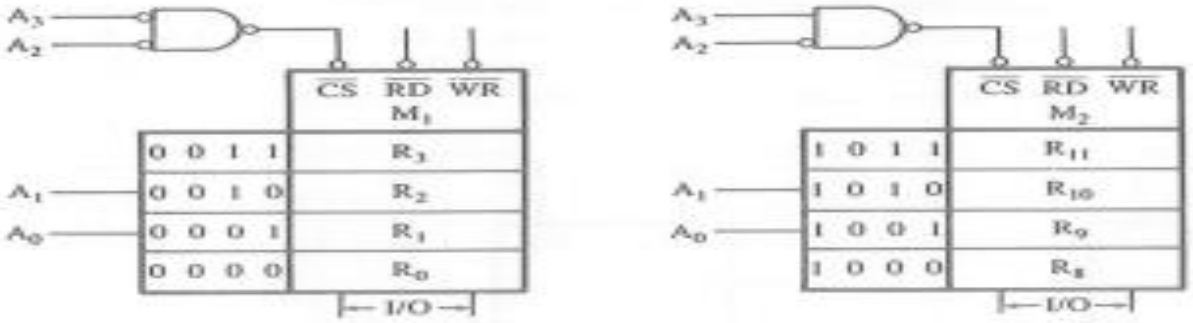


FIGURE 3.8
Addressing Eight Registers with Four Address Lines

1. A memory chip requires address lines to identify a memory register. The number of address lines required is determined by the number of registers in a chip (2^n = Number of registers where n is the number of address lines). The 8085 microprocessor has 16 address lines. Of these 16 lines, the address lines necessary for the memory chip must be connected to the memory chip.

2. A memory chip requires a Chip Select (CS) signal to enable the chip. The remaining address lines (from Step 1) of the microprocessor can be connected to the CS signal through an interfacing logic.

3. The address lines connected to CS select the chip, and the address lines connected to the address lines of the memory chip select the register. Thus the memory address of a register is determined by the logic levels (0/1) of all the address lines (including the address lines used for CS).

4. The control signal Read (RD) enables the output buffer, and data from the selected register are made available on the output lines. Similarly, the control signal Write (WR) enables the input buffer, and data on the input lines are written into memory cells. The microprocessor can use its Memory Read and Memory Write control signals to enable the buffers and the data bus to transport the contents of the selected register- between the microprocessor and memory.

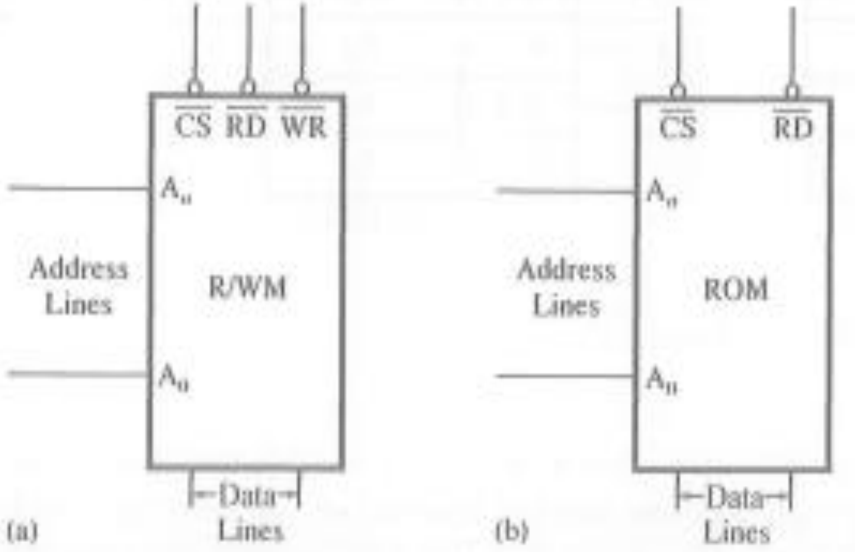


FIGURE 3.9
R/W Memory Model (a) and ROM Model (b)

MEMORY MAP AND ADDRESSES

- Typically, in an 8-bit microprocessor system, 16 address lines are available for memory.
- The entire memory addresses can range from 0000 to FFFF in Hex. A memory map is a pictorial representation in which memory devices are located in the entire range of addresses. Memory addresses provide the locations of various memory devices in the system, and the interfacing logic defines the range of memory addresses for each memory device.
- In the hexadecimal number system, 16 binary bits are equivalent to four Hex digits that can be used to assign addresses to 65,536 (0000H to FFFFH) memory registers in various memory chips.

MEMORY WORD SIZE

- Memory devices (chips) are available in various word sizes (1, 4, and 8) and the size of a memory chip is generally specified in terms of the total number of bits it can store. On the other hand, the memory size in a given system is generally specified in terms of bytes.
- Therefore, it is necessary to design a byte-size memory word. For example, a memory chip of size 1024 x 4 has 1024 registers and each register can store four bits; thus it can store a total of 4096 (1024 x 4 = 4096) bits. To design 1K-byte (1024 x 8) memory, we will need two chips; each chip will provide four data lines.

MEMORY AND INSTRUCTION FETCH

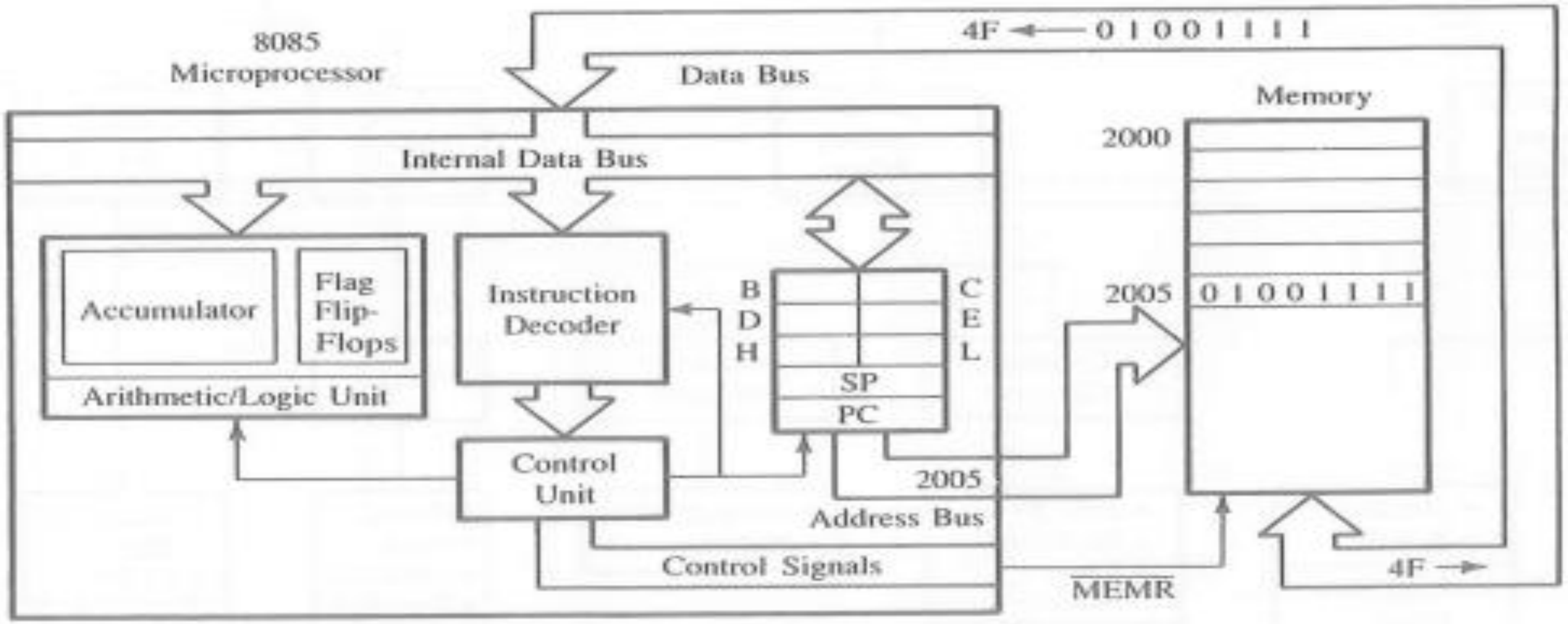
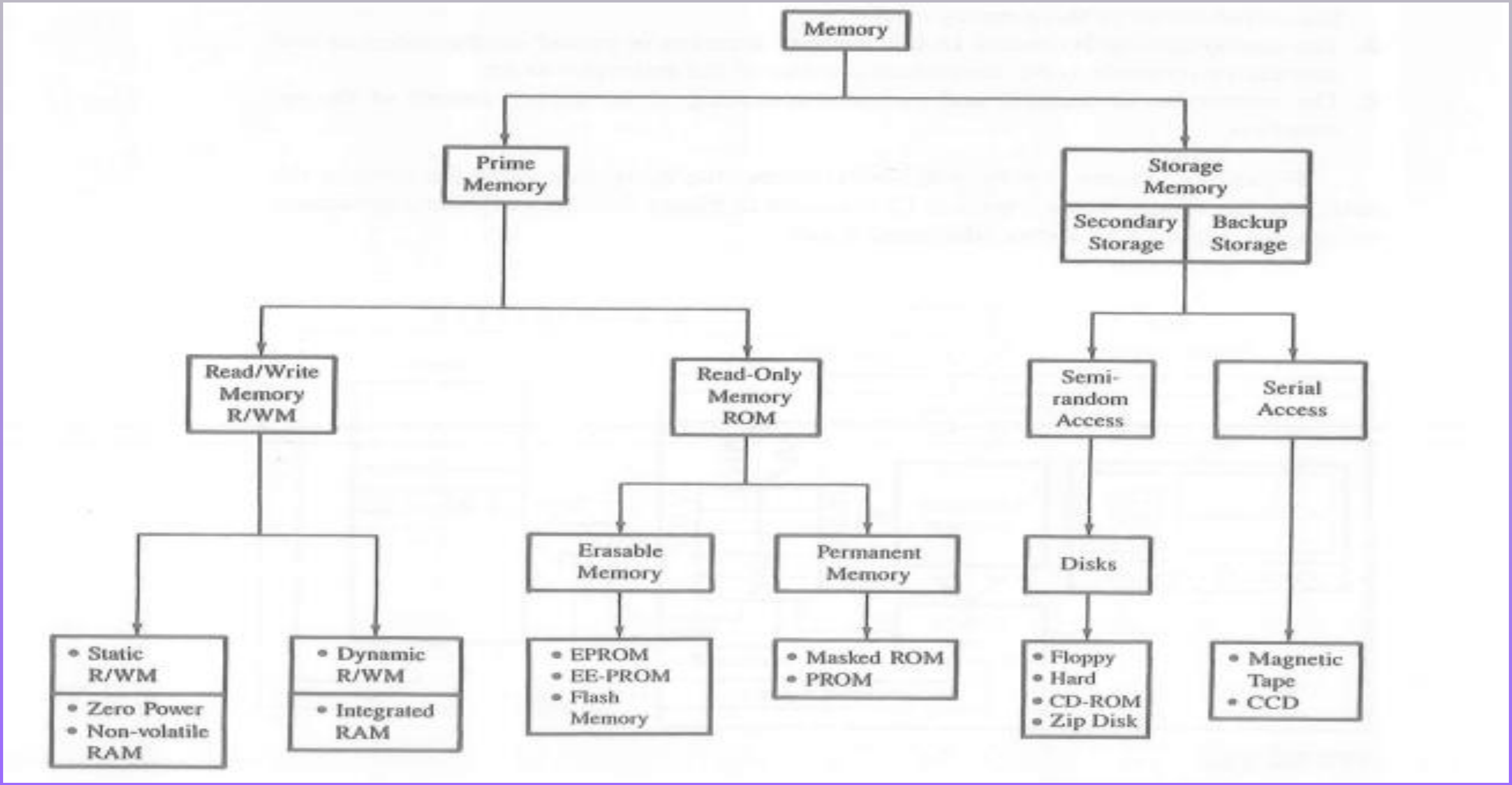


FIGURE 3.12
Instruction Fetch Operation

MEMORY CLASSIFICATION

- Memory can be classified into two groups: prime (system or main) memory and storage memory.
- The R/W M and ROM are examples of prime memory; this is the memory the microprocessor uses in executing and storing programs. This memory should be able to respond fast enough to keep up with the execution speed of the microprocessor. Therefore, it should be random access memory, meaning that the microprocessor should be able to access information from any register with the same speed (independent of its place in the chip).
- For example, a 1K memory chip means it can store 1K (1024) bits (not bytes). On the other hand, memory in a system such as a PC is specified in bytes. For example, 4M memory in a PC means it has 4 megabytes of memory.
- The other group is the storage memory, such as magnetic disks and tapes. This memory is used to store programs and results after the completion of program execution. Information stored in these memories is nonvolatile, meaning infor- mation remains intact even if the system is turned off.

- The microprocessor cannot directly execute or process programs stored in these devices; programs need to be copied into the R/W prime memory first. Therefore, the size of the prime memory, such as 512K or 8M (megabytes), determines how large a program the system can process. The size of the storage memory is unlimited; when one disk or tape is full, the next one can be used.
- two groups in storage memory: secondary storage and backup storage.
- The secondary storage and the backup storage include devices such as disks, magnetic tapes, magnetic bubble memory, and charged-coupled devices.
- The primary features of these devices are high capacity, low cost, and slow access. A disk is similar to a record; the access to the stored information in the disk is semirandom.


Memory
Prime Memory
Storage Memory
Secondary Storage
Backup Storage
Read/Write Memory R/WM
Read-Only Memory ROM
Semi-random Access
Serial Access
Erasable Memory
Permanent Memory
Disks
• Static R/WM
• Zero Power
• Non-volatile RAM
• Dynamic R/WM
• Integrated RAM
• EPROM
• EE-PROM
• Flash Memory
• Masked ROM
• PROM
• Floppy
• Hard
• CD-ROM
• Zip Disk
• Magnetic Tape
• CCD

R/WM (READ/WRITE MEMORY)

- it is popularly known as Random Access memory (RAM). It is used primarily for information that is likely to be altered, such as writing programs or receiving data. This memory is volatile. Two types of R/W:-
- Static Memory (SRAM) This memory is made up of flip-flops, and it stores the bit as a voltage. Each memory cell requires six transistors; therefore, the memory chip has low density but high speed. This memory is more expensive and consumes more power than the dynamic memory.
- Dynamic Memory (DRAM) This memory is made up of MOS transistor gates, and it stores the bit as a charge. The advantages of dynamic memory are that it has high density and low power consumption and is cheaper than static memory. The disadvantage is that the charge (bit information) leaks; therefore, stored information needs to be read and written again every few milliseconds. This is called refreshing the memory, and it requires extra circuitry, adding to the cost of the system. In comparison to the processor speed, the DRAM is too slow. To increase the speed of DRAM various techniques are being used. These techniques have resulted in high-speed memory chips such as EDO (Extended Data Out), SDRAM (Synchronous DRAM), and RDRAM (Rambus DRAM). ROM (READ-ONLY MEMORY)

ROM

- The ROM is a nonvolatile memory; it retains stored information even if the power is turned off. This memory is used for programs and data that need not be altered. The information can be read only, which means once a bit pattern is stored, it is permanent or at least semipermanent.
- The permanent group includes two types of memory: masked ROM and PROM.
- The semipermanent group also includes two types of memory: EPROM and EE-PROM
- When memory register 111 is selected data byte 0111 1000(78H) can be read D7-D0

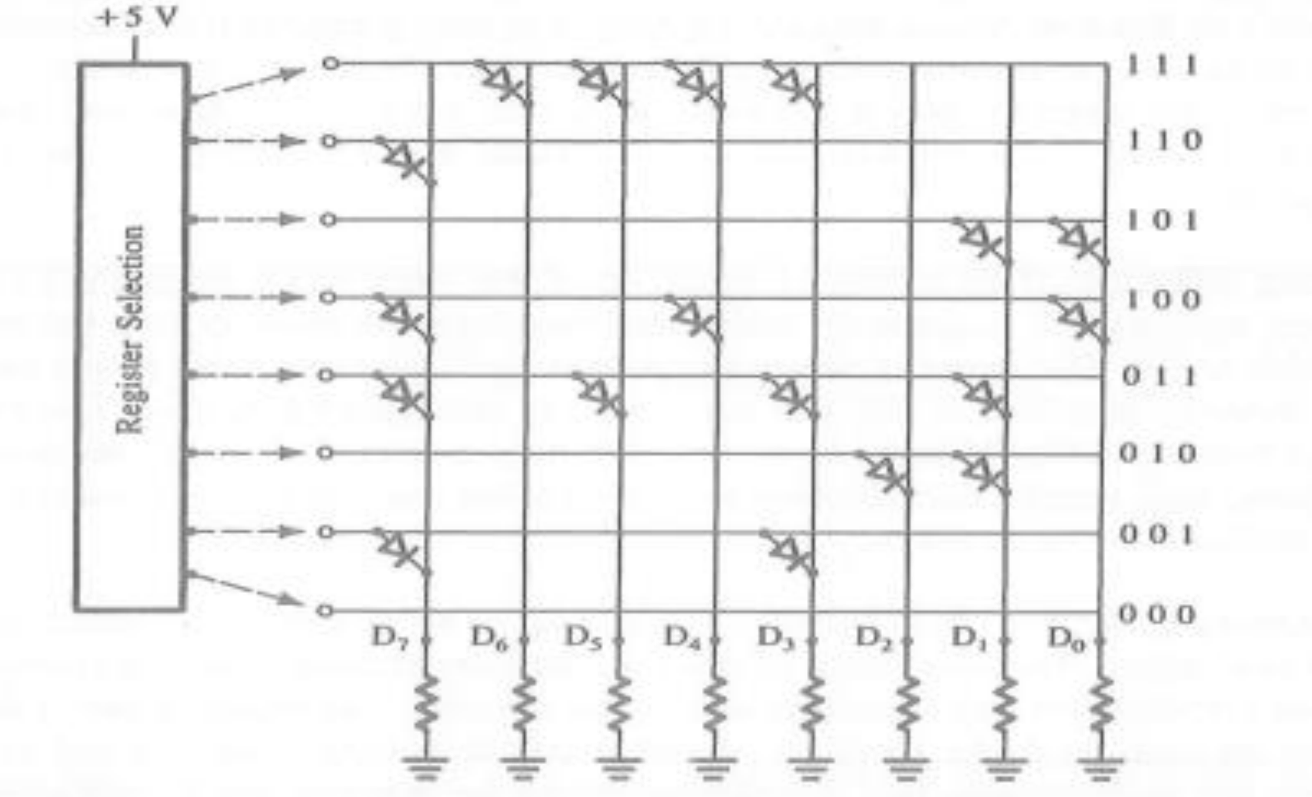


FIGURE 3.14
Functional Representation of ROM Memory Cell

- **Masked ROM** In this ROM, a bit pattern is permanently recorded by the masking and metalization process. Memory manufacturers are generally equipped to do this process. It is an expensive and specialized process, but economical for large production quantities.
- **PROM (Programmable Read-Only Memory)** This memory has nichrome or poly- silicon wires arranged in a matrix; these wires can be functionally viewed as diodes or fuses. This memory can be programmed by the user with a special PROM programmer that selectively burns the fuses according to the bit pattern to be stored. The process is known as "burning the PROM," and the information stored is permanent.
- **EPROM (Erasable Programmable Read-Only Memory)** This memory stores a bit by charging the floating gate of an FET. Information is stored by using an EPROM programmer, which applies high voltages to charge the gate. All the information can be erased by exposing the chip to ultraviolet light through its quartz window, and the chip can be reprogrammed.
- The disadvantages of EPROM are (1) it must be taken out of the circuit to erase it, (2) the entire chip must be erased, and (3) the erasing process takes 15 to 20 minutes.

- **EE-PROM (Electrically Erasable PROM)** This memory is functionally similar to EPROM, except that information can be altered by using electrical signals at the register level rather than erasing all the information. In microprocessor systems, software update is a common occurrence. However, this memory is expensive compared to EPROM or flash memory.
- **Flash Memory** This is a variation of EE-PROM that is becoming popular. The major difference between the flash memory and EE-PROM is in the erasure procedure: The EE- PROM can be erased at a register level, but the flash memory must be erased either in its entirety or at the sector (block) level. These memory chips can be erased and programmed at least a million times.

ADVANCES IN MEMORY TECHNOLOGY

- The Zero Power RAM is a CMOS Read/Write memory with battery backup built internally. This memory provides the advantages of R/W and Read-Only memory.
- The Nonvolatile RAM is a high-speed static R/W memory array backed up, bit for bit, by EE-PROM array for nonvolatile storage. When the power is about to go off, the contents of R/W memory are quickly stored in the EE-PROM by activating the Store signal on the memory chip, and the stored data can be read into the R/W memory segment when the power is again turned on. This memory chip combines the flexibility of static R/W memory with the nonvolatility of EE-PROM.
- The Integrated RAM (IRAM) is a dynamic memory with the refreshed circuitry built on the chip. For the user, it is similar to the static R/W memory. The user can derive the advantages of the dynamic memory without having to build the external refresh circuitry.

INPUT OUTPUT DEVICES(I/O) DEVICES

- Input/output devices are the means through which the MPU communicates with "the outside world."
- The MPU accepts binary data as input from devices such as keyboards and A/D converters and sends data to output devices such as LEDs or printers.
- There are two different methods by which I/O devices can be identified: one uses an 8-bit address and the other uses a 16-bit address.
- ## I/O s with 8-Bit Addresses (Peripheral-Mapped I/O)
- In this type of I/O, the MPU uses eight address lines to identify an input or an output device; this is known as peripheral-mapped I/O (also known as I/O-mapped I/O). This is an 8-bit numbering system for I/Os used in conjunction with Input and Output instructions. This is also known as I/O space, separate from memory space, which is a 16-bit numbering system.

- The eight address lines can have 256 (2^8) addresses; thus, the MPU can identify 256 input devices and 256 output devices with addresses ranging from 00H to FFH.
- The input and output devices are differentiated by the control signals; the MPU uses the I/O Read control signal for input devices and the I/O Write control signal for output devices. The entire range of I/O addresses from 00 to FF is known as an I/O map, and individual addresses are referred to as I/O device addresses or I/O port numbers.
- The steps in communicating with an I/O device are similar to those in communicating with memory and can be summarized as follows:
- 1. The MPU places an 8-bit address on the address bus, which is decoded by external de- code logic
- 2. The MPU sends a control signal (I/O Read or I/O Write) and enables the I/O device. 3. Data are transferred using the data bus.

- **I/Os with 16-Bit Addresses (Memory-Mapped I/O)**
- In this type of I/O, the MPU uses 16 address lines to identify an I/O device; an I/O is connected as if it is a memory register. This is known as memory-mapped I/O.
- The MPU uses the same control signal (Memory Read or Memory Write) and instructions as those of memory. In some microprocessors, such as the Motorola 6800, all I/Os have 16-bit ad- dresses; I/Os and memory share the same memory map (64K).
- In memory-mapped I/O, the MPU follows the same steps as if it is accessing a memory register.

EXAMPLE OF MICROCOMPUTER SYSTEM

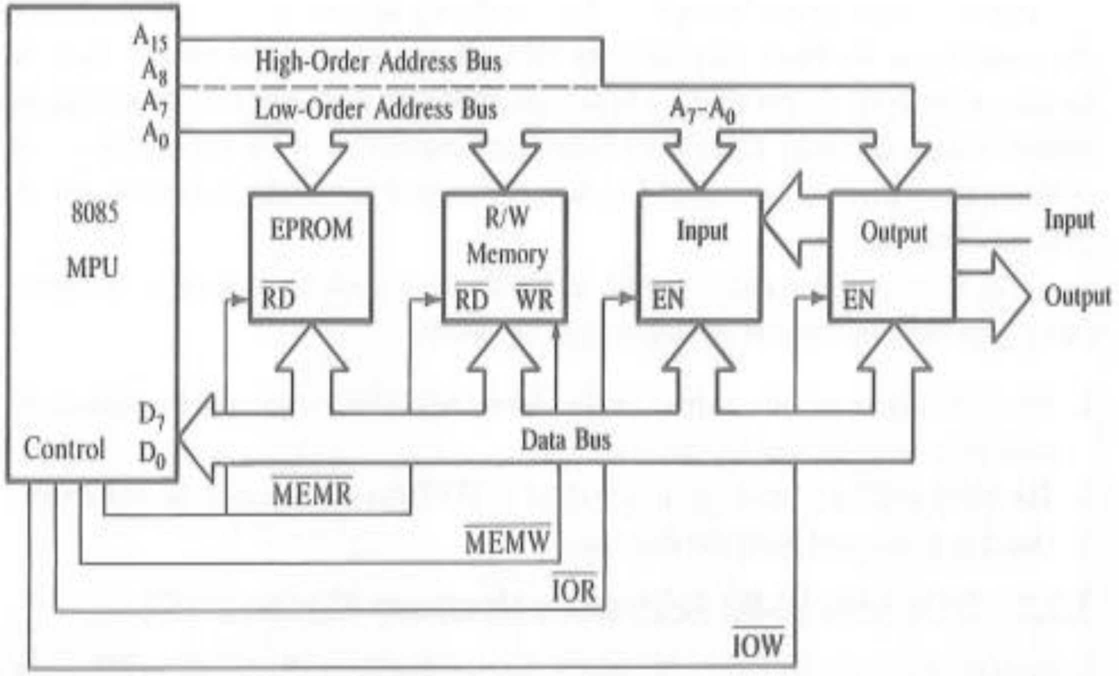


FIGURE 3.15
Example of a Microcomputer System

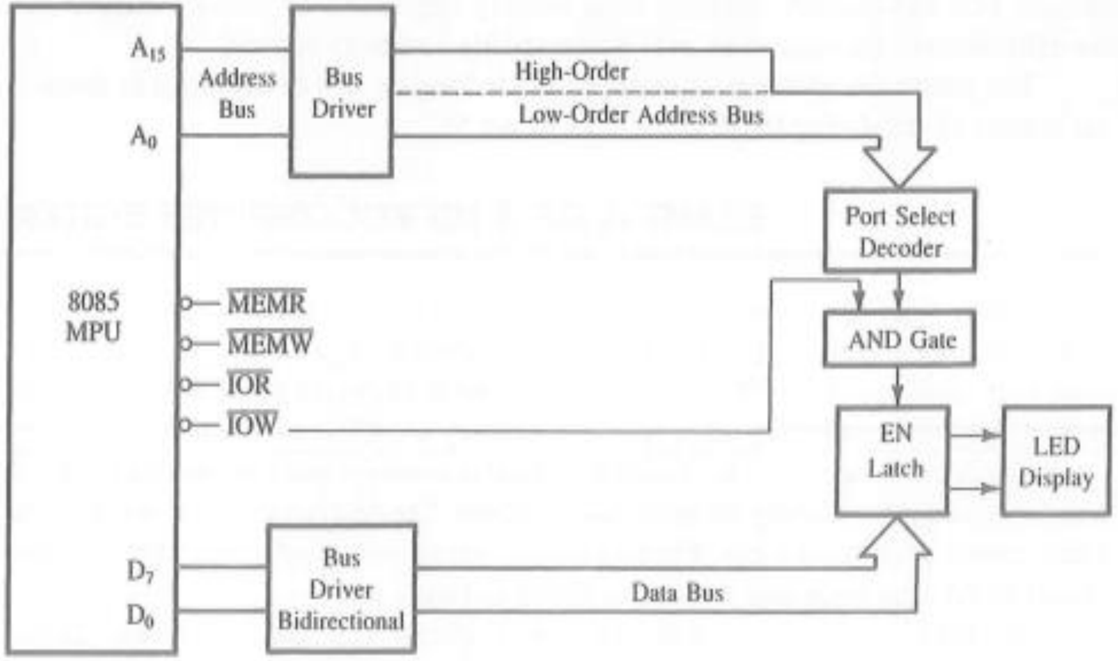
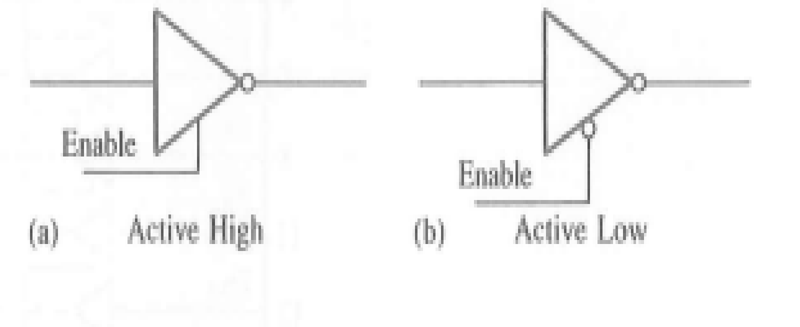


FIGURE 3.16
The Output Section of the Microcomputer System Illustrated in Figure 3.15

LOGIC DEVICES FOR INTERFACING

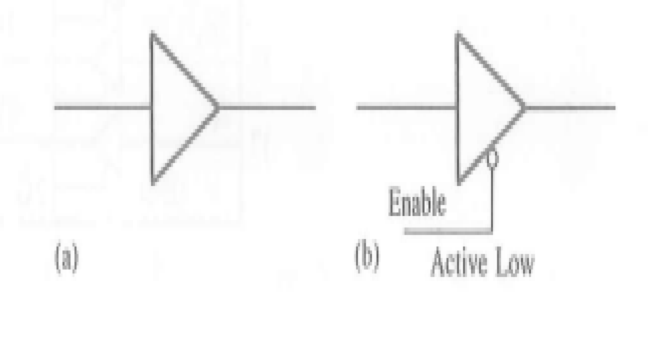


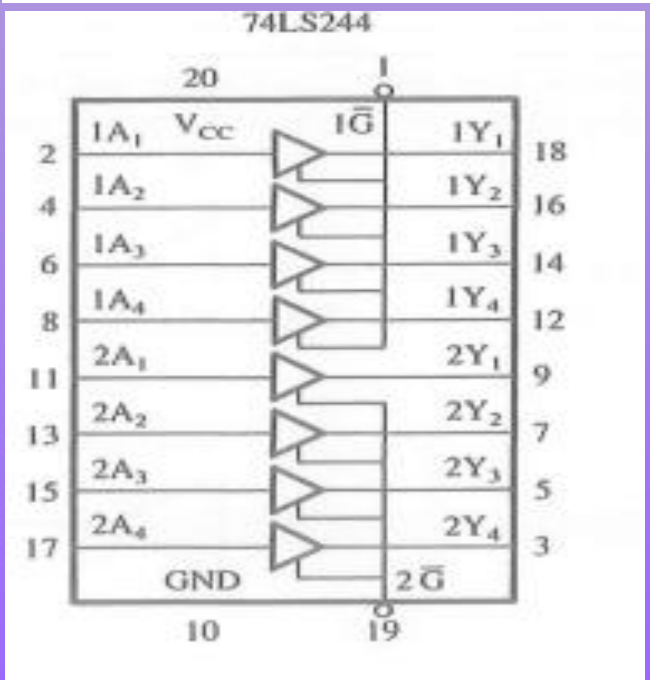
Tri-State Devices

- Tri-state logic devices have three states: logic 1, logic 0, and high impedance.
- A tri-state logic device has a third line called Enable, as shown in Figure 3.17. When this line is activated, the tri-state device functions the same way as ordinary logic devices. When the third line is disabled, the logic device goes into the high impedance state-as if it were disconnected from the system
- When the Enable is high, the circuit functions as an ordinary inverter; when the Enable line is low, the inverter stays in the high impedance state.
- The microprocessor communicates with one device at a time by enabling the tri-state line of the interfacing device.

Buffer

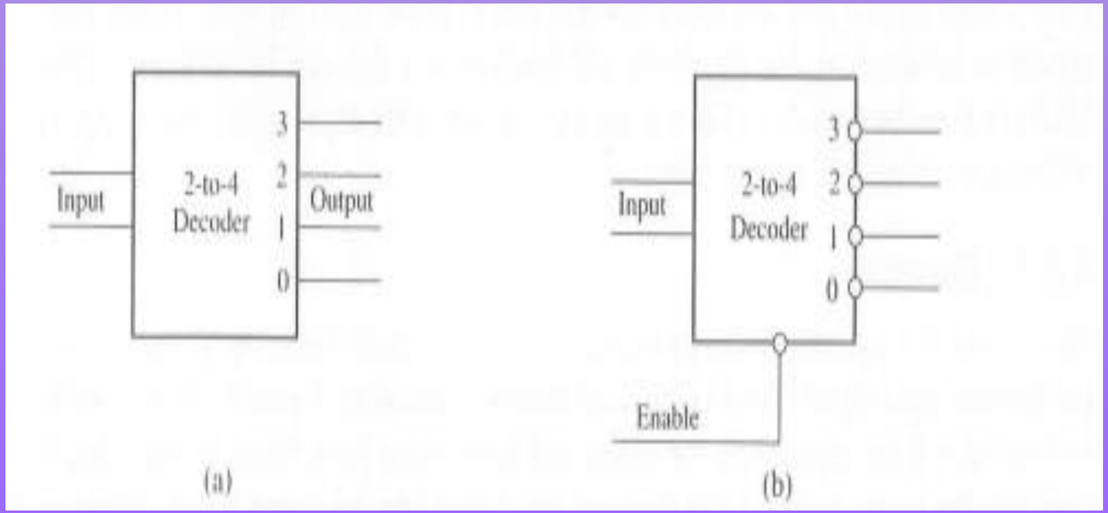
- The buffer is a logic circuit that amplifies the current or power. It has one input line and one output
- The logic level of the output is the same as that of the input; logic 1 input provides logic 1 output (the opposite of an inverter). The buffer is used primarily to increase the driving capability of a logic circuit. It is also known as a driver.
- When the Enable line is low, the circuit functions as a buffer; otherwise it stays in the high impedance state. The buffer is commonly used to increase the driving capability of the data bus and the address bus.
- EXAMPLES OF TRI-STATE BUFFERS
- The octal buffer 74LS244 shown in Figure 3.19 is a typical example of a tri-state buffer. It is also known as a line driver or line receiver. This device is commonly used as a driver for the address bus in a bus-oriented system.

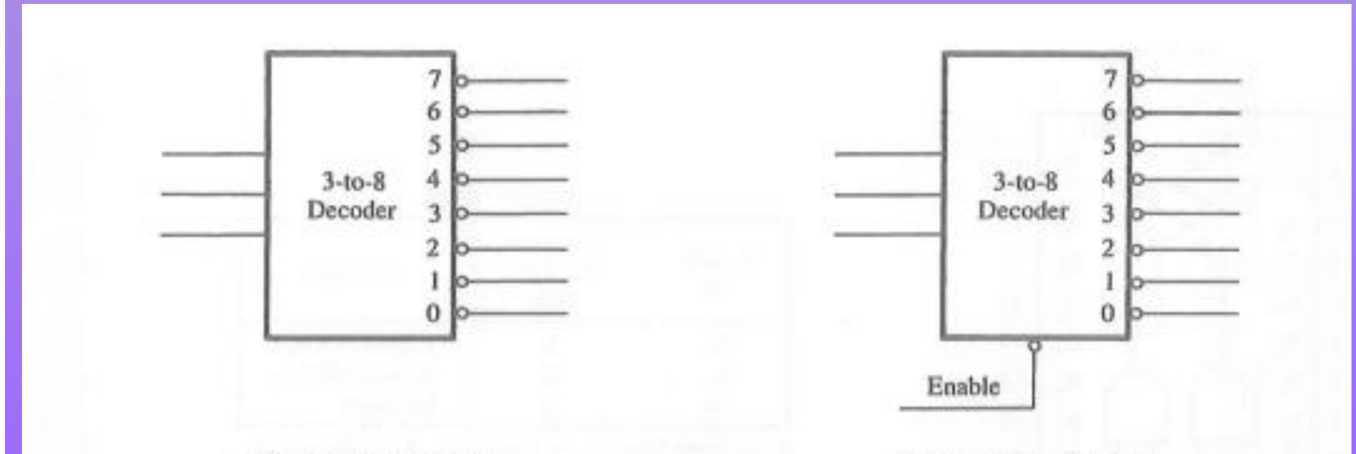




DECODER

- The decoder is a logic circuit that identifies each combination of the signals present at its input.
- For example, if the input to a decoder has two binary lines, the decoder will have four output lines.
- The two lines can assume four combinations of in- put signals-00, 01, 10, 11-with each combination identified by the output lines 0 to 3. If the input is 11, the output line 3 will be at logic 1, and the others will remain at logic 0. This is called decoding.





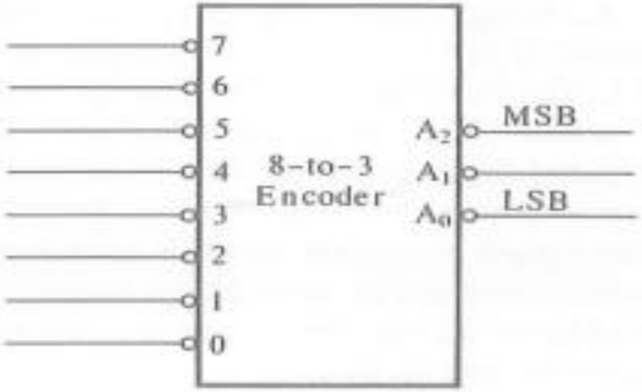
(a) Active Low Output

(b) Decoder with Enable

ENCODER

- The encoder is a logic circuit that provides the appropriate code (binary, BCD, etc.) as output for each input signal. The process is the reverse of decoding. an 8- to-3 encoder; it has eight active low inputs and three output lines. When the input line 0 goes low, the output is 000; when the input line 5 goes low, the output is 101. However, this encoder is unable to provide an appropriate output code if two or more input lines are activated simultaneously. Encoders called priority encoders can resolve the problem of simultaneous inputs.

FIGURE 3.24
Logic Symbols: 8-to-3 Encoder



MICROPROCESSOR-BASED SYSTEM APPLICATION-MCTS