# 5 CHAPTER

# MEMORIES AND MICROPROCESSORS

## LEARNING OBJECTIVES

After studying this chapter, the student should be able to:

- Outline various categories of memories and their role in data storage.
- Discuss semiconductor memories, memory addressing, and extend ROMs, PROMs, EPROMs, DRAMs, and PLAs.
- Describe the microprocessor and its various elements, applications, and types.
- Understand computers and their functionalities.

## 5.1 INTRODUCTION

Modern data-processing systems require permanent or semi-permanent storage of large amounts of data. Microprocessor-based systems rely on memories for their operation because memories are used to store programs and data for processing. A number of reliable memory circuits are available which have replaced older devices like magnetic cores. A typical semiconductor memory consists of a rectangular array of memory cells, which are nothing but transistor flip-flops or a circuit capable of storing charge and is used to store 1 bit of information. Memories can be divided into the following categories—Read only memories (ROM) and random access, read–write memories (RAM).

## 5.2 SEMICONDUCTOR MEMORIES

**FLASH CARD**

**What You Learn Here**
Discuss and differentiate various memory classes of semiconductors.

Random access memory is used for applications where in the data changes frequently. Logic circuitry is available to store a single bit of information in a memory cell (WRITE) or to detect a 0 or a 1 stored in the memory cell (READ). A bit of data can be stored in any cell or read from any cell and so this type of memory is called as **random access memory**. The data stored in this type of memory is lost once the power is switched off. So, this memory is said to provide volatile storage.

ROM is used in applications where the data do not change. Applications like monitor programs, look-up tables can be stored in a ROM. The content of a ROM is fixed during manufacturing. A ROM is random access and logic circuitry is available to read data from the selected cell, however, there is no write mode. Since data is permanently stored, loss of power never lead to loss of data.

### 5.2.1 Memory Addressing

Memory addressing is the process of selecting one of the cells in a memory to be written into or to be read from. Memories are generally arranged by placing cells in a rectangular arrangement of rows and columns. Figure 5.1 shows a memory array with $m$ rows and $n$ columns, with a total of $m \times n$ cells in the memory.

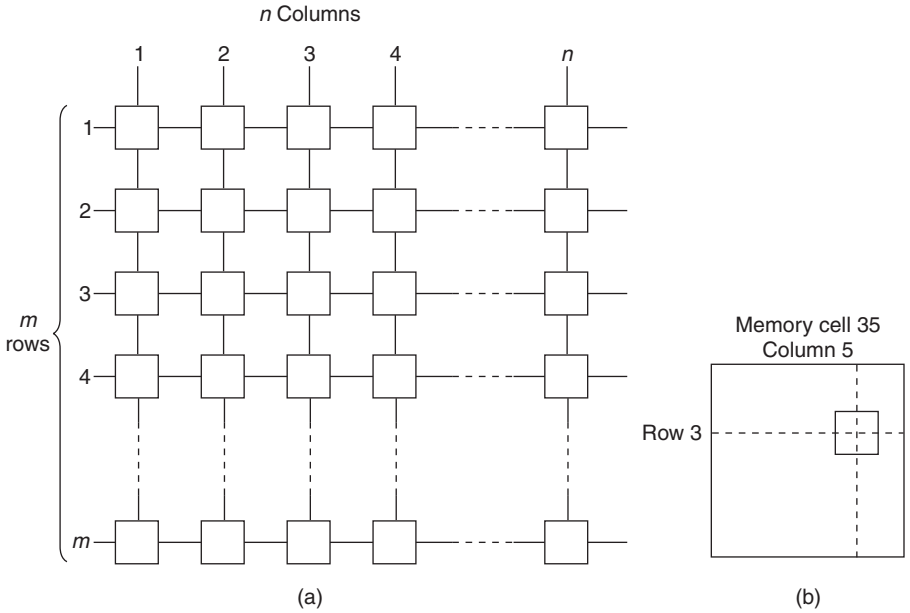



**Fig. 5.1** *(a) Array of m × n cells (b) Selecting the cell at memory address*

The control circuitry is designed such that if only one row line is activated and only one column line is activated, the memory cell at the intersection of these two lines is selected. For example, in Fig. 5.1, if row 3 and column 5 is activated, the cell at the intersection of this row and column is selected, and the address of this cell is designated as 35. Any row or column can be activated by placing a logic 1 on it.

**PROBE**

How is the control circuitry designed?

Memory can be arranged in various possible configurations of rectangular array of memory cells. If there are $M$ memory cells, these can be arranged in $M \times 1$ array, or $1 \times M$ array, or $m \times n = M$ array. A signal column with $n$ rows is frequently called a linear array and addressing a cell in this array is referred to as linear addressing. Linear addressing requires the maximum number of address lines. The arrangement that requires the minimum address lines is a square array of $m$ rows and $m$ columns with a total of $m^2$ memory cells. This arrangement is frequently known as matrix addressing.

**PROBE**

Define the different address lines used in addressing.

The number of address lines can be further reduced using decoders. To illustrate this, consider a 4 × 4 memory array as shown in Fig. 5.2. Normally, eight address lines are required to select a memory cell in the array. To select a single cell, we must activate only one row and only one column. This suggests the use of two 2 to 4 binary-to-decimal decoders.

**PROBE**

What is activated to select a single cell?

If the memory 34 (row 3 and column 4) is to be selected, $A_4A_3A_2A_1$ must be 1011. That is, if $A_4 = 1$ and $A_3 = 0$, the decoder will hold row 3 high, while all other row lines will be low. Similarly, if $A_2 = 1$ and $A_1 = 1$, the decoder will hold column 4 high and all other column lines low. Thus, the input $A_4A_3A_2A_1 = 1011$ will select the memory cell 34. We can consider $A_4A_3$ as row address of 2 bits and $A_2A_1$ as a column address of 2 bits. In general, an address of $n$ bits can be used to define a square memory of $2^n$ cells, where there are $n/2$ bits for the rows and $n/2$ bits for the columns as shown in Fig. 5.3.

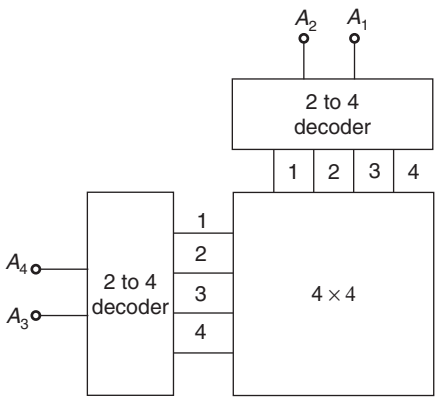



**Fig. 5.2** *Addressing a 4 × 4 memory unit*

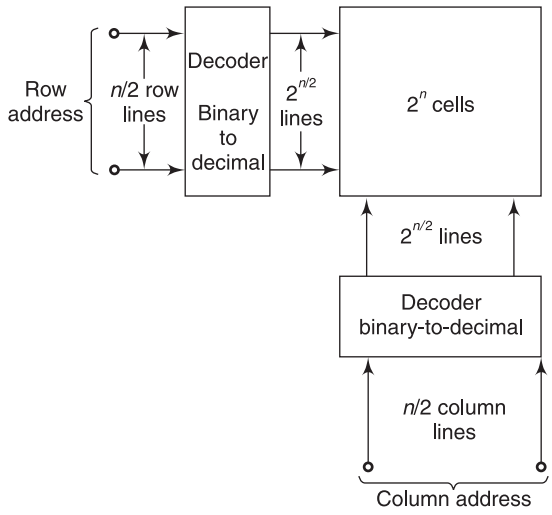



**Fig. 5.3** *Memory addressing of $2^n$ cells*

### 5.2.2 ROMs, PROMs, and EPROMs

▪ ***Read Only Memories (ROM)*** Semiconductor ROMs are manufactured with bipolar technology or with MOS technology. A ROM contains permanently or semi-permanently stored data which can be read from the memory, but which either cannot be changed at all or cannot be changed without specialized equipment.

A ROM is permanently programmed during the manufacturing process to provide widely used standard functions such as popular conversions or to provide user-specified functions. Once the memory is programmed, it cannot be changed. The logic 1 or 0 is represented by the presence or absence of a transistor connection at a row/column junction as shown in Fig. 5.4(a). Presence of a connection from a row line to the base of a transistor represents a 1 at that location because when the row line is taken HIGH, all transistors with a base connection to that row line turn on and connect the HIGH (1) to the associated column lines. At row/column junctions, where there are no base connections, the column lines remain LOW (0), when the row is addressed.

**PROBE**

How is the logic 1 or 0 represented?

Figure 5.4(b) illustrates MOS ROM cells. The presence or absence of a gate connection at the junction permanently stores a 1 or a 0. A simple ROM array is shown in Fig. 5.5. The blank squares represent that 0s are stored and the dark squares represent that 1s are stored. When a binary address is given to the address

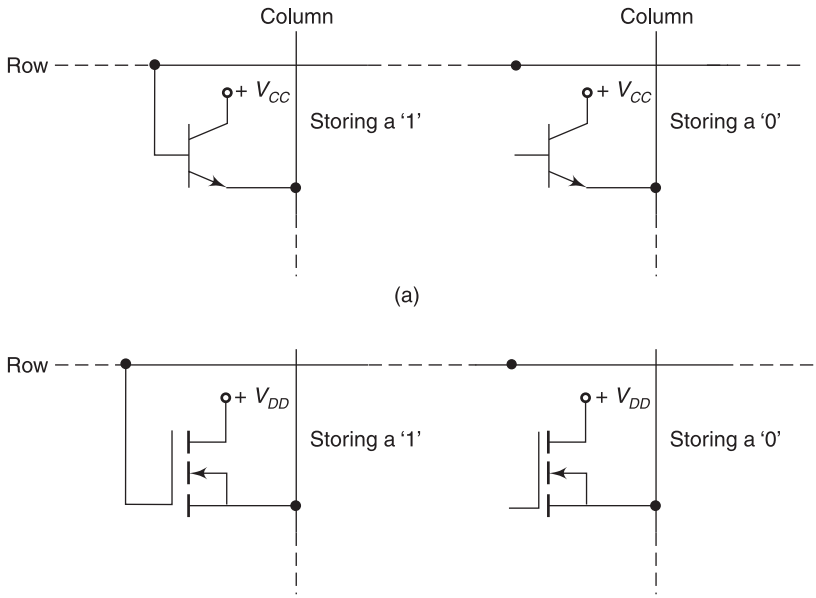



**Fig. 5.4** *ROM (a) Bipolar cells, (b) MOS cells*

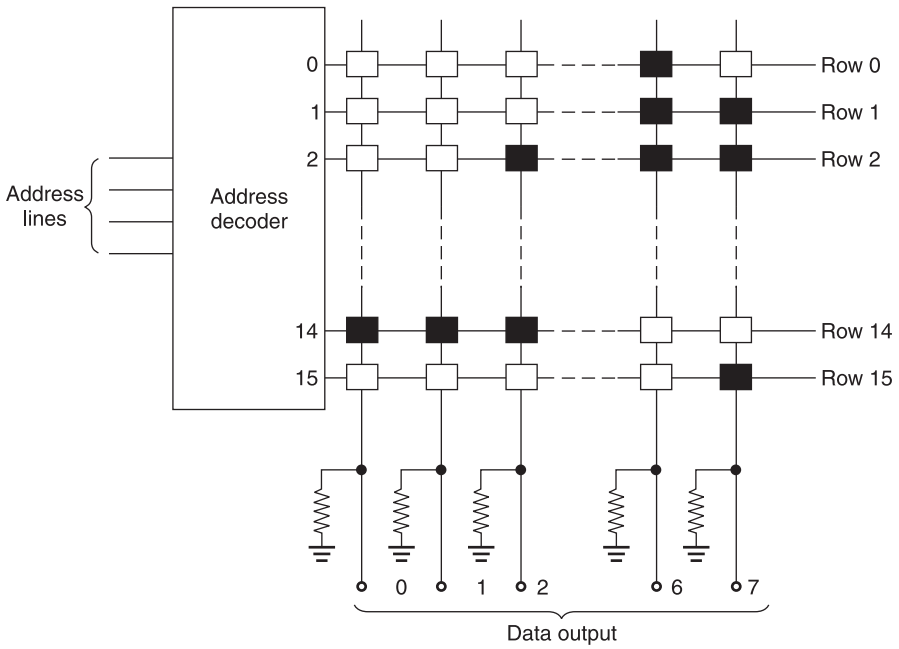



**Fig. 5.5** *A 16 × 8-bit ROM array*

decoder, the corresponding row line goes HIGH. This HIGH is connected to the column lines through the transistors at each junction where a 1 is stored. At each cell where a 0 is stored, the column line stays LOW. The eight data bits stored in the selected row appear on the output lines.

▪ ***Programmable Read Only Memories (PROMs)*** This type of memory is not programmed during the manufacturing process, but are custom programmed in the field. PROMs are available in bipolar as well as in MOS technologies. Some type of fusing process is used to store bits. The fuse is open to represent a 0 or left intact to represent a 1. The contents of a PROM, once programmed, cannot be changed. Figure 5.6 shows a PROM with fuses between the emitter of the transistor of each cell and its column line.

**PROBE**

Which kind of memory is custom programmed in the field?

The fuse can be blown by passing sufficient current through it, and a blown fuse represent a 0 and the unblown fuse represent a 1. A PROM is normally programmed using an instrument called a PROM programmer. An address is selected by the switch settings on the address lines and a pulse is applied to those output lines corresponding to cell locations where 0s are to be stored. These pulses blow the fuse links, thus creating the desired bit pattern. The next address is now selected and the process is repeated. This sequence is done automatically by the PROM programmer.

**PROBE**

How are blown and unblown fuses represented?

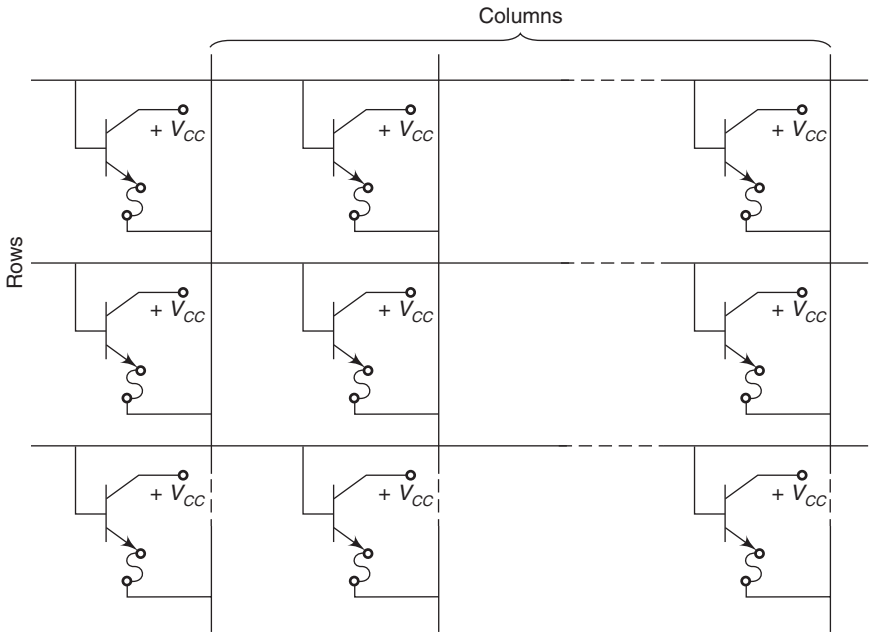



**Fig. 5.6** *Bipolar PROM array with fusible links*

▪ ***Erasable Programmable Read Only Memory (EPROM)*** The contents of an EPROM can be erased and reprogrammed. The memory arrays in these devices use an NMOS array with an isolated gate structure. The isolated gate has no electrical connections and store electrical charge for indefinite period of time. The presence of stored charge represents logic 1 and absence of stored charge represents a logic 0. Erasure of a data bit is a process that removes the gate charge.

**PROBE**

Which memory arrays are used in NMOs?

Two types of erasable PROMs are available, namely, the Ultraviolet Light Erasable PROM (UVEPROM) and the Electrically Erasable PROM (EEPROM). In UVEPROMs, there is a quartz transparent lid on the package. When UV light is passed through this lid, the positive charge stored on every gate is neutralized. In EEPROM, electrical pulses are used for both programming and erasure of data bits. This type of device is also known as Electrically Alterable ROM (EAROM). The application of a voltage pulse on the control gate in the floating gate structure either stores or removes the charge from the floating gate.

**PROBE**

Differentiate between different types of erasable PROMs.

### 5.2.3 Random Access Memory (RAM)

The random access memory is an array of memory-storage cells. In this memory, information can be randomly written into or read out of each storage element as required. The basic monolithic storage cell is the latch, or flip-flop. The operation of a RAM cell can be understood easily by considering the simple, 1-bit D flip-flop circuit shown in Fig. 5.7.

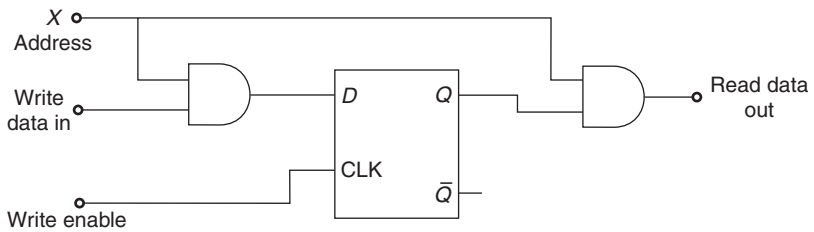



**Fig. 5.7** *A 1-bit read/written memory*

To read a data or to write it, the address line must be excited, i.e. $X$ must be equal to logic 1. To perform write operation, the write enable line must also be excited. If the write input is a logic 1, then $D = 1$. Hence, $Q = 1$, and the data read out is 1. When the write input is logic 0, then $D = 0$, $Q = 0$ and the data read out is also 0. A system that can store 16 words of 8 bits each requires $16 \times 8 = 128$ storage cells, eight data input lines, eight data output lines and sixteen address lines.

Figure 5.8 shows a block diagram of a MOS static memory. The 4096 bits of memory are arranged in an array of 64 rows and 64 columns. The memory cells are accessed by simultaneously decoding the $X$ address $A_0 - A_5$ for the rows and the $Y$ address $A_6$–$A_{11}$ for the columns. The Chip Select (CS) input controls the operation of the memory. When CS is low, the input address buffers, decoders, sensing circuits, and output stages are held in the off state, and power is supplied only to the memory elements. When CS goes high, the memory is enabled. The CS pulse clocks the TTL level addresses, READ–WRITE, and DATA input into D flip-flops and enables the output stage. When a cell is read from, one of the two outputs will be a 1. Figure 5.9 shows

a random access memory of 16 words by 1-bit. For M-bit words, there will be *M* planes, like the one shown.

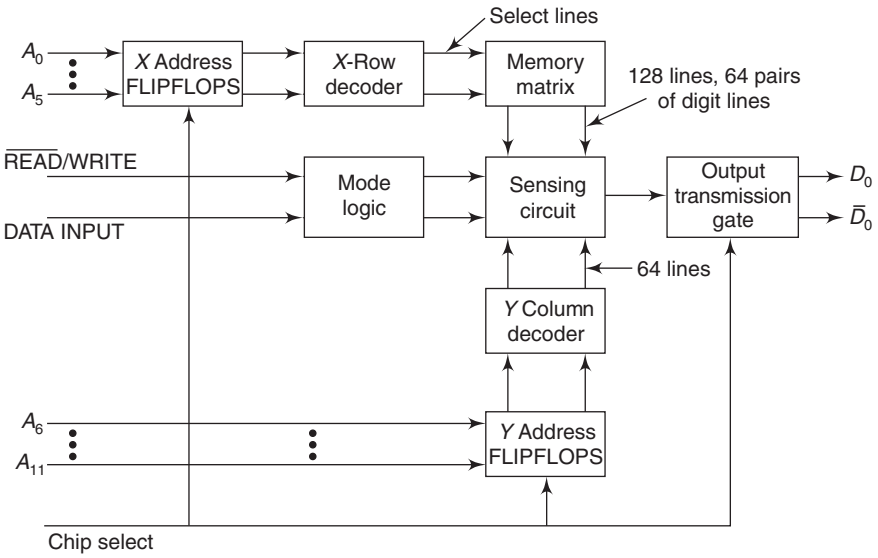



**Fig. 5.8** *Block diagram of a MOS static memory*

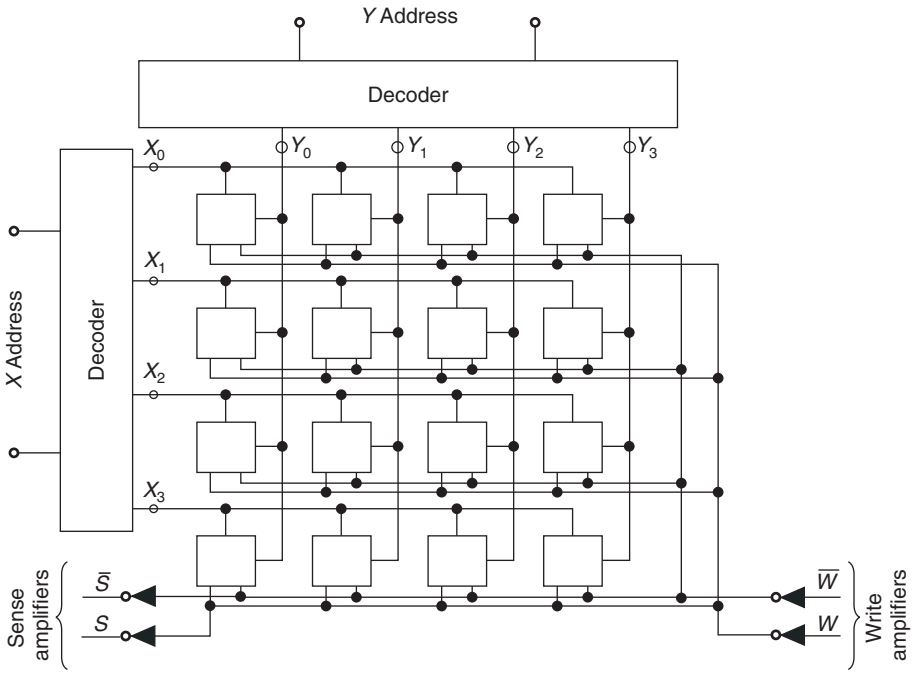



**Fig. 5.9** *A 16 word × 1-bit RAM*

### 5.2.4 Dynamic Random Access Memories (DRAMs)

The DRAMs have individual memory cells composed of from one to three MOS transistors and a capacitor. The state of the cell is determined by placing or not placing a positive charge on the capacitor. A logic 0 is represented by no charge on the capacitor and logic 1 by placing a positive charge on the capacitor. This kind of memory requires individual memory cells that are simpler than flip-flops, less area on the chips and has lower power consumption. The disadvantages of DRAMs are that they are slower and the charge slowly leaks from the capacitor. Thus, the contents of each memory cells must be rewritten periodically. This is called *refreshing the memory*. In spite of these disadvantages, the memory costs are lower than the static memories and hence DRAMs are widely used in present day systems.

**PROBE**

How does refreshing of memory play an important role in the functioning of DRAMs?

Dynamic random access memories are arranged in the same way as static memories are. The two-dimensional selection with decoders is standard. Memory controller chips are used to control several dynamic memory chips assembled into a memory. These controller chips handle the sequencing of the row and column addresses into the individual chips during a normal memory access and also control necessary refreshing of the memory. The refreshing generally involves a counter which sequences through all possible states during a 2 ms period. The memory controller chip attempts to perform refresh operations between memory accesses, whenever possible, thus reducing the time lost to memory refreshes. Use of dynamic memory controller chips simplifies dynamic memory design and optimizes memory operation.

### 5.2.5 Programmable Logic Arrays (PLAs)

A programmable logic array is similar to a ROM in many ways but is very different in terms of their internal structure. The PLA can be either programmed while manufacturing or field programmed by the user. The user programmable PLAs are called as Field Programmable Logic Arrays (FPLAs). A PLA consists of two arrays, one for AND logic and one for OR logic, and can be programmed to produce desired logic functions on the outputs. Figure 5.10 shows a layout of a small PLA. This particular array has three AND gates and two OR gates. In actual PLAs, an array would have several hundred or more gates.

**PROBE**

What is the alternative term for user-programmable PLAs?

In the figure, note that the connection from inputs $A$, $B$, and $C$ to the AND gates and the AND gate outputs to the OR gates are all connected through fusible links. These fuses are blown and connections established as desired by the gate network designer.

To realize the Boolean algebra expressions, $Y_1 = ABC + \overline{A}B$ and $Y_2 = \overline{A}B\overline{C} + \overline{A}B$, the fuses at the respective crossover points are kept intact and the fuses at the other crossover points are blown out, as shown in Fig. 5.11. The unblown fuses are represented by black dots at the crossover points.

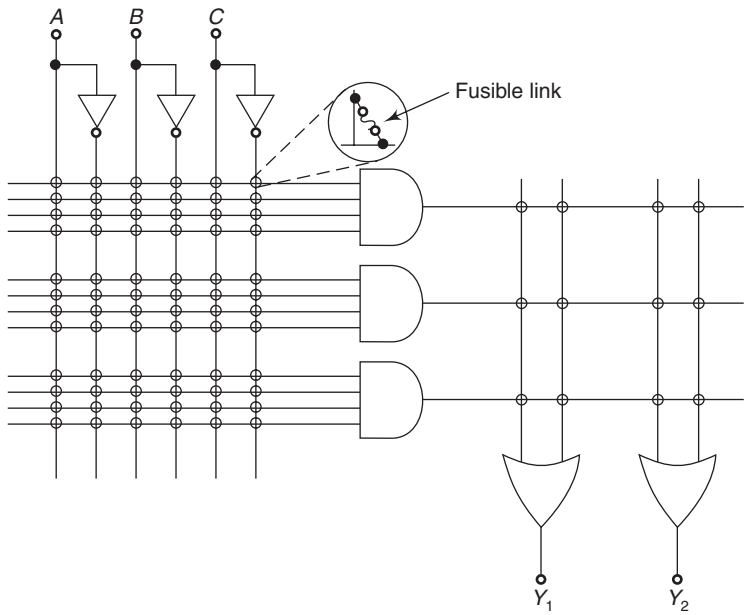



**Fig. 5.10** *Layout of three-input, two-output PLA*

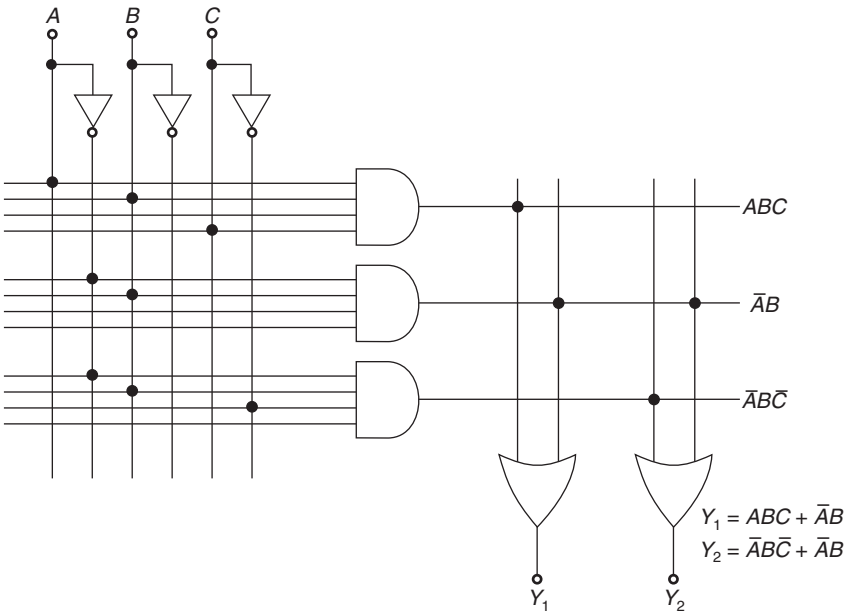



**Fig. 5.11** *Connection design for three-input, two-output PLA*

## 5.3 INTRODUCTION TO MICROPROCESSORS

**FLASH CARD**

**What You Learn Here**
Become familiar with different classes of microprocessors and discuss their applications.

A microprocessor is an electronic device which can be extensively used for various control applications when it is interfaced with memories and several input and output devices. It is a device that has a limited set of on-chip memory locations, known as registers, to hold information. It can understand a fixed set of basic commands and can generate signals to control external devices.

Inside the chip, there is an Arithmetic Logic Unit (ALU). The ALU executes all arithmetic and logic instructions. For example, the arithmetic addition and logical AND operations will be performed by the ALU. The registers inside the microprocessor hold data on which the operations are performed. The control unit generates the external control signals and also controls the operation of the internal on-chip circuitry.

**PROBE**
What are different instructions executed by ALU?

The on-chip memory, in the form of registers, is generally very limited. Thus, almost every microprocessor-based system has an off-chip memory also. The set of basic commands that a microprocessor can understand, is known as the instruction set of the microprocessor.

**PROBE**
What is the ALU?

Figure 5.12 shows the signals available in a typical microprocessor chip. The chip itself has several pins, like any other chip. The microprocessor sends or receives information over these pins. Each pin transmits or receives a Boolean signal which is either at logical 0 or at logical 1. Some pins may be in neither of these two states at certain points of time. Such pins, or lines, are said to be tri-state outputs.

**PROBE**
What is tri-state output? State its role in a microprocessor chip.

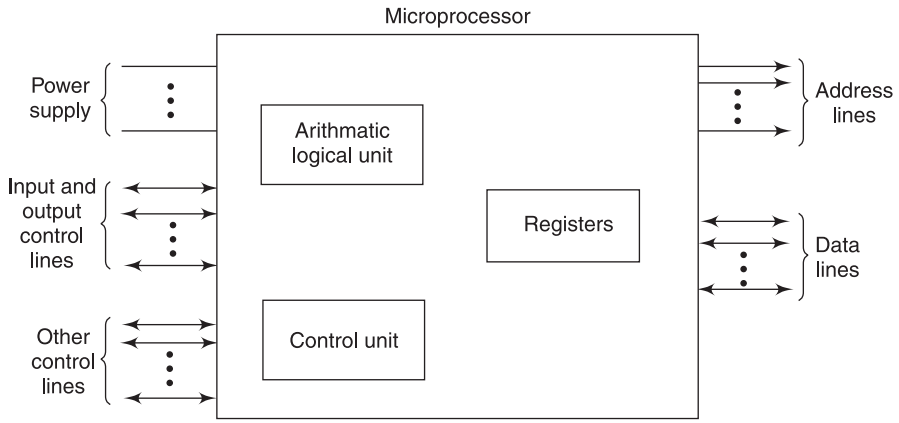



**Fig. 5.12** *Simple block diagram of a microprocessor*

As shown in Fig. 5.12, a typical microprocessor has several lines over which it transmits an address to the off-chip memory or to the I/O devices. These are referred to as address lines. More often, the *address lines* are known as the *address bus*. A typical 8-bit microprocessor has an address bus consisting of 16 lines for transmitting the address. Thus, such a microprocessor can transmit an address that is of 16-bit wide.

Every microprocessor has a set of lines for transmitting and receiving data. These lines are referred to as the *data bus*. A 8-bit microprocessor will generally have eight lines in its data bus. A 16-bit microprocessor may

have either 16 lines in its data bus or only 8 lines. For example, the 8085 microprocessor from Intel is an 8-bit microprocessor and has 8 lines in its data bus. The 8086 microprocessor, has 16 lines in its data bus and is a 16-bit microprocessor. The 8088 is a 16-bit microprocessor but has only 8 lines in its data bus. A lesser number of data buses is provided to reduce the space required on a printed circuit board for the data bus lines.

A microprocessor also has lines for controlling the input and output devices. These devices could be an electric motor, or a display lamp, or one of a variety of other devices. In addition to the control signals for input/output devices, a microprocessor also has some other control signals for (i) controlling off chip memory, (ii) providing information about its own status, and (iii) performing other miscellaneous tasks. A brief description of the function of each element in a microprocessor is given below.

**PROBE**

Summarize various elements of a microprocessor and their functions.

- ***Accumulator*** One function of the accumulator is to store an operand prior to an operation by the arithmetic logic unit (ALU), and the other function is to temporarily store the result of the operation.

- ***Program Counter*** This 16-bit counter produces the sequence of memory addresses in which the program instructions are stored. The content of the program counter is always the memory address from which the next instruction is to be taken.

- ***Instruction Register*** When the contents of the program counter go out onto the address bus to the memory, an eight-bit instruction is read from that memory location and is transferred on the data bus, to be temporarily stored in the instruction register until it is decoded and executed by the instruction decode and control element.

- ***Instruction Decode and Control*** An *instruction* is a binary code that tells the microprocessor what it is to do. A sequence of many different instructions designed to accomplish a specific result is a *program*. In other words, a program is a step-by-step procedure used by the microprocessor to carry out a specified task.

The instruction decode and control element decodes an instruction code that has been transferred on the data bus from the memory. The instruction code is often called an *op code*. When the op code is decoded, the instruction decoder provides the control unit with this information so that it can produce the proper signals and timing sequence to execute or carry out the instruction.

- ***Condition Code Register*** The basic purpose of this element is to indicate the status of the contents of the accumulator or certain other conditions within the microprocessor. For example, it can indicate a zero result, a negative result, the occurrence of a carry, and the occurrence of an overflow from the accumulator.

- ***Stack Pointer*** This register is used in conjunction with a portion of RAM that is set aside for use as a memory *stack*. The stack is used mainly to temporarily store the contents of all the registers when a subroutine is called or an interrupt requires the microprocessor to temporarily cease its current operation to respond to a request from an external device that needs to send data or requires service from the microprocessor. When an interrupt request has been serviced, the contents of the stack are transferred back into the appropriate registers and the microprocessor continues where it left off. The stack pointer is used to address the stack portion of RAM for storing and retrieving data in such a situation.

### 5.3.1 8085A Microprocessor

A simple block diagram of the Intel 8085A microprocessor is shown in Fig. 5.13. The 8085A has an 8-bit data bus which is shared with a 16-bit address bus. The 8085A has a 16-bit program counter, a 16-bit stack

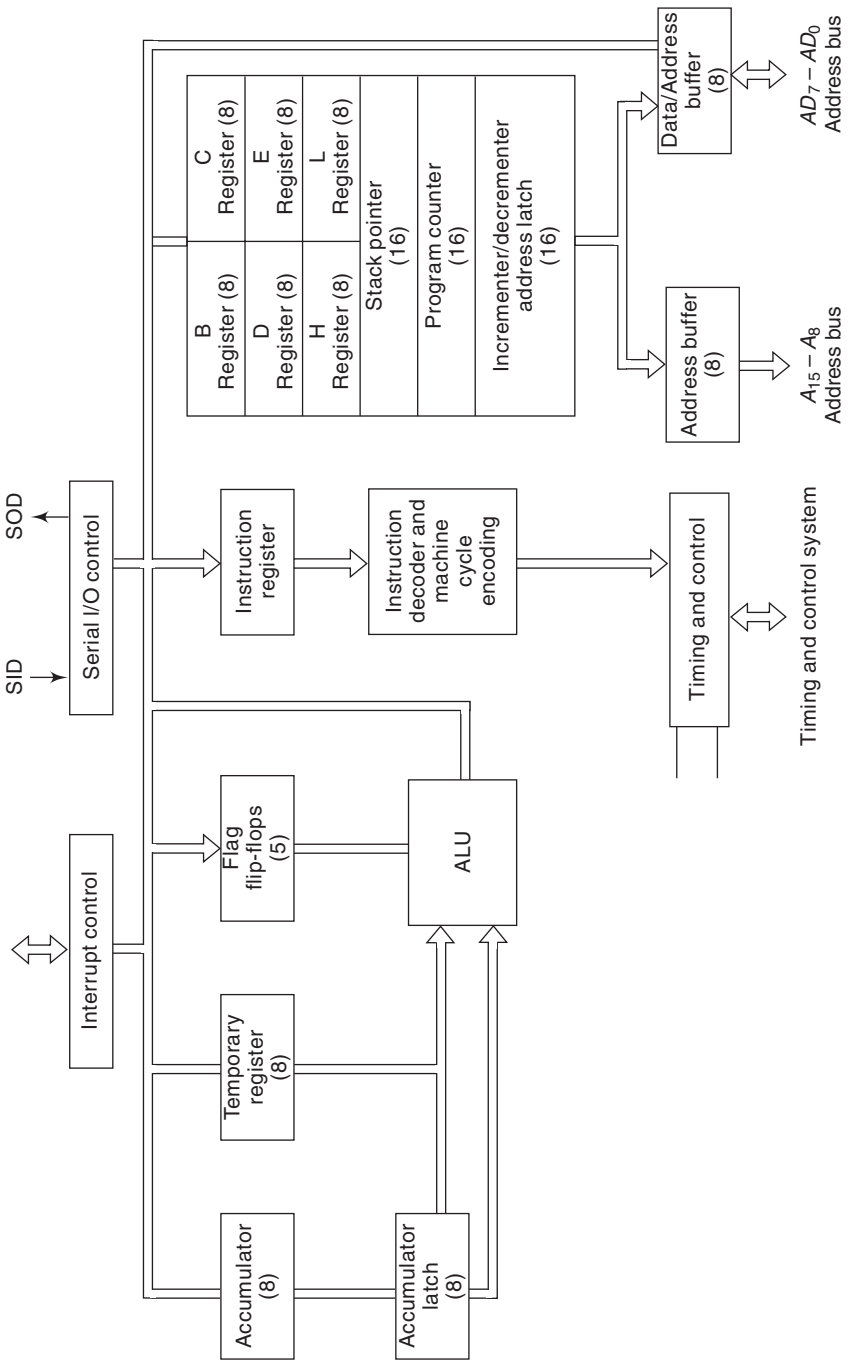



**Fig. 5.13** *Block diagram of the 8085A microprocessor*

pointer, and an 8-bit instruction register and flag registers. The 8085A has a single 8-bit accumulator and six 8-bit general-purpose registers (*B*, *C*, *D*, *E*, *H*, and *L*). Two general-purpose registers in the register array can be combined to form 16-bit registers.

The *incrementer/decrementer* is associated with the register array and allows each register to be incremented or decremented independent of the ALU and accumulator. The single *temporary* register associated with the accumulator holds one of the operands for an ALU operation. The 8085A operates on a single 5 V supply. The required clock signals are created internally by connecting an external crystal between $X_1$ and $X_2$. Also, the 8085A has Serial Input Data (SID) and Serial Output Data (SOD) capability for use in simple forms of serial communications, such as teletype. A logic symbol showing inputs and outputs with pin numbers appears in Fig. 5.14.

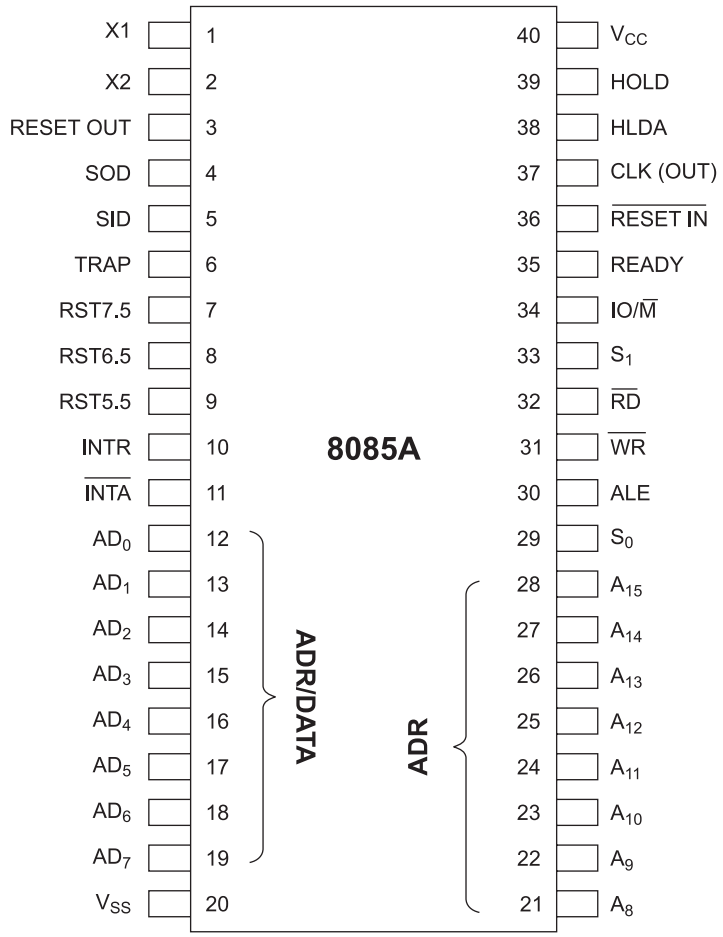



**Fig. 5.14** *Pin Configuration of 8085A microprocessor*

### 5.3.2 6800 Microprocessor

A generalized block diagram of the 6800 microprocessor is shown in Fig. 5.15. All the elements are interconnected by an 8-bit common internal bus structure. This microprocessor can handle data in groups

of eight bits from the data bus, and it has a 16-bit address bus. Thus, up to 65,536 memory locations can be addressed. All memory (other than registers) is external to this particular microprocessor, thus requiring additional chips to form a complete microcomputer system.

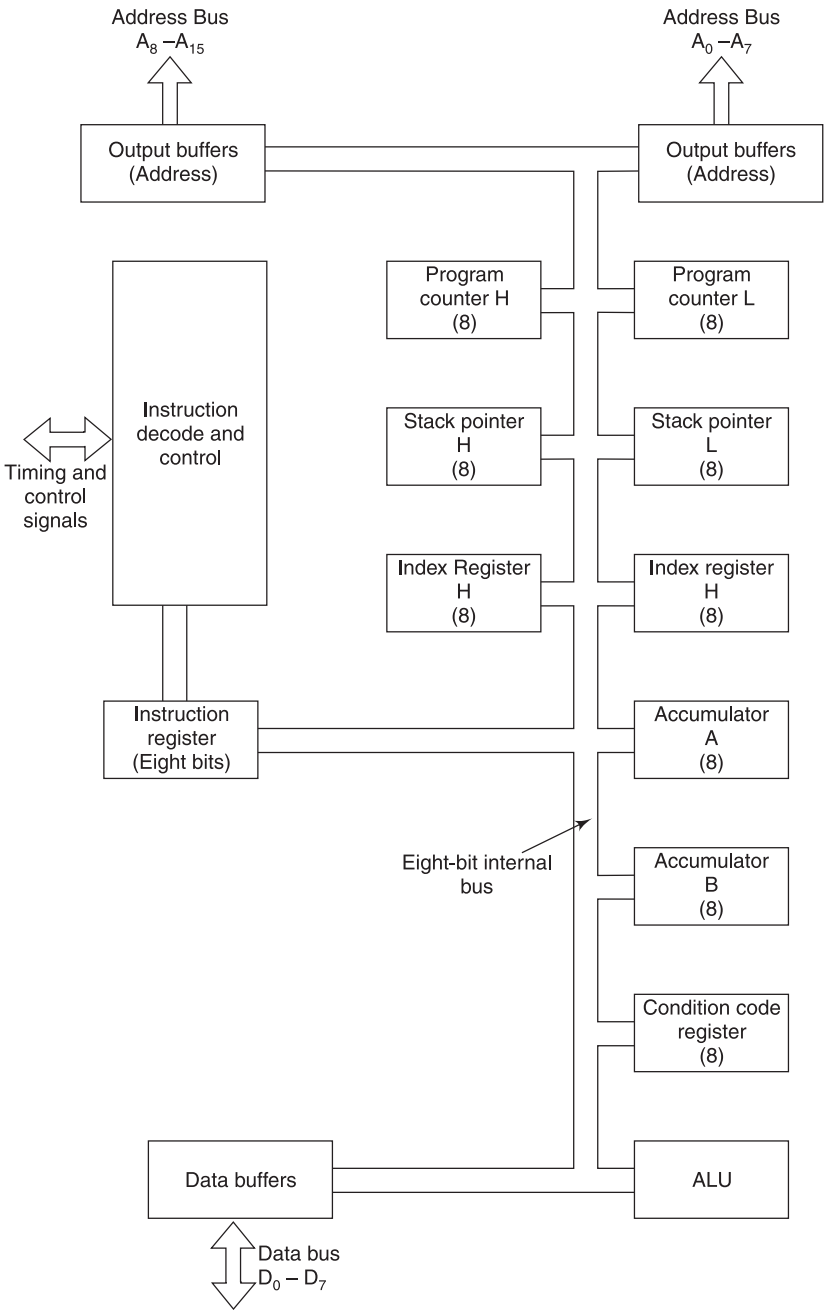



**Fig. 5.15** *Block diagram of the 6800 microprocessor*

The registers in the 6800 are as follows: two 8-bit accumulators, one 16-bit index register, one 16-bit program counter, one 16-bit stack pointer, one 8-bit condition code register, and one 8-bit instruction register. The 16-bit registers are actually treated as a combination of two 8-bit registers. The H label refers to the high-order bits, the L label to the low-order 8-bits.

A logic symbol of the 6800 showing input and outputs with pin numbers appears in Fig. 5.16. The 6800 was the first of the Motorola microprocessor family and is still a widely used device. It requires three other chips to form a complete microcomputer system: an external clock generator, a RAM, and a ROM.

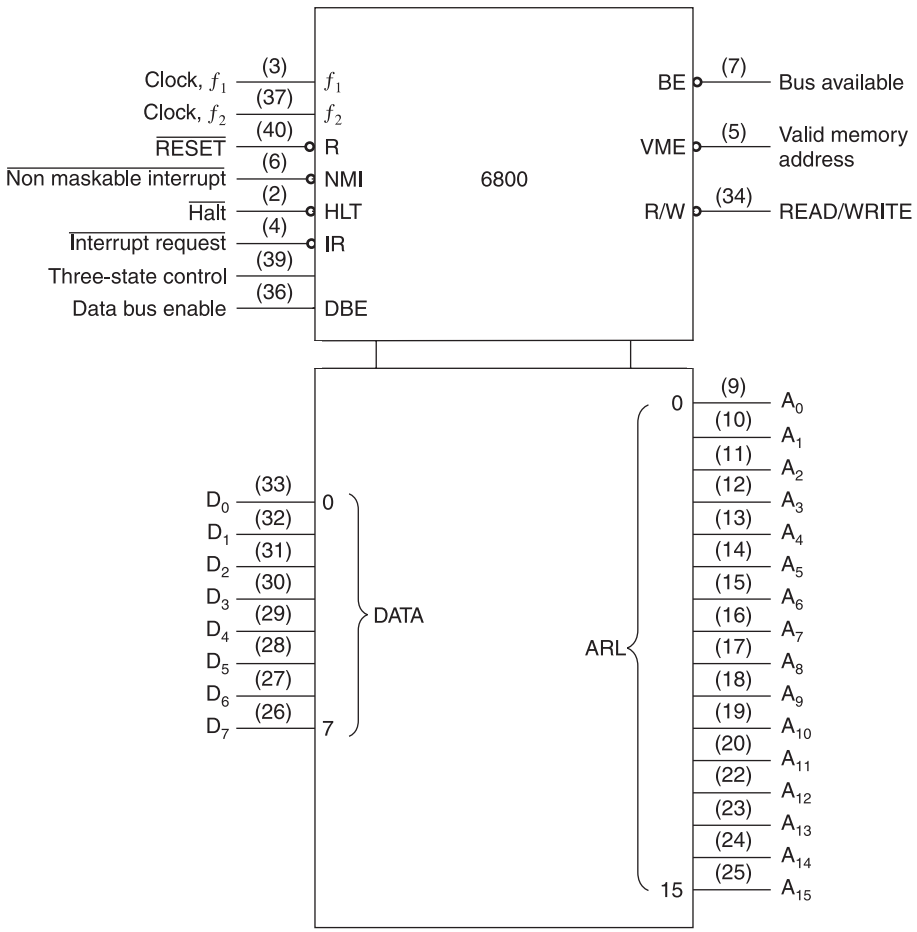



**Fig. 5.16** *Logic symbol of 6800 microprocessor*

### 5.3.3 8086 and 8088 Microprocessors

The 8086 and 8088 microprocessors are extensions of Intel's 8080 microprocessor series. There are a number of changes in the 8086/8088, the most obvious being the fact that computations can be performed using 16-bit data. There are a number of other advantages including multiplication and division instructions, instruction queuing to improve

**PROBE**

What are the advantages of including multiplication and division instructions?

operation speed, the ability to address a million bytes of memory, more general registers, and more instructions and addressing modes.

The pin-outs for the 8086 and 8088 are shown in Fig. 5.17. The address and data lines are shared by using time-division multiplexing. The principal difference between the 8086 and 8088 lies in the number of data-lines output to the bus. The 8086 has 16 data lines on its bus, and the 8088 has only 8. The 8086 use 16 of the address lines for data also, so that AD0 to AD15 are used for address and data while the 8088 has only AD0 to AD7 for data and uses A8 to A19 for addresses. The internal paths on the chips are the same and each can add, subtract, multiply, or divide 16-bit binary numbers.

**PROBE**

What is the principal difference between 8086 and 8088?

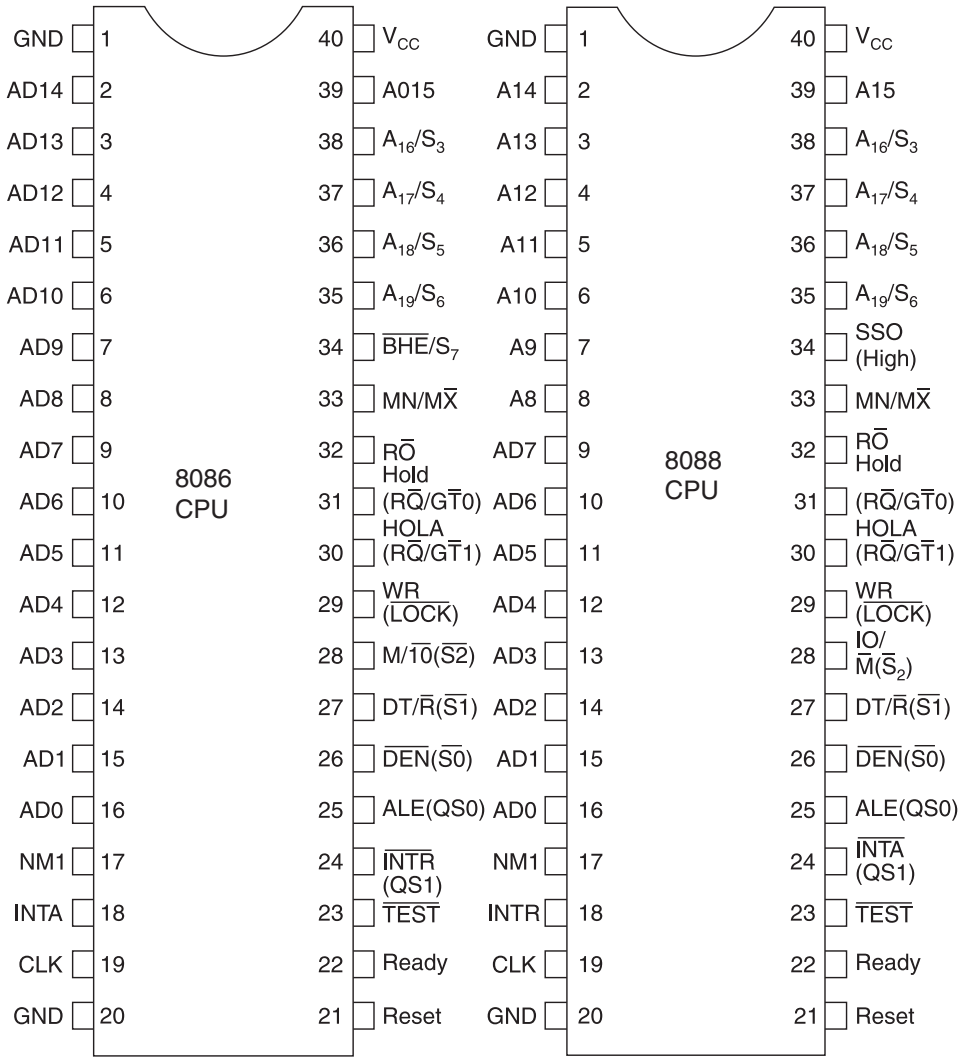



**Fig. 5.17** *Pin-outs for 8086 and 8088 processors*

An important feature of the 8086 and 8088 microprocessors is the instruction queue used in each. The 8086/8088 chips read instructions in order from the memory in advance of their operation, and the instructions are placed in a queue consisting of a set of flip-flop registers. This speeds up operation because the processor can continue executing a time-consuming instruction and at the same time, read instructions from the memory of the processor. Then the processor can execute fast instructions from the queue at a speed faster than memory cycle times. Logic is supplied so that if the computer branches, the instructions in the queue are discarded if necessary.

The 8086/8088 pair each have a special output pin, the $MN/\overline{MX}$ pin. When this pin is connected to logic '1', the processor is placed in a minimum mode; when it is connected to logic '0', the processor is placed in a maximum mode. When in the minimum mode, the processor is used in single processor systems. In the maximum mode, several processors can be used with an 8288 bus controller which provides a special multibus architecture for multiprocessor systems. The maximum mode is for large arrays of memory, processors, and I/O devices.

A block diagram of the registers of the 8086 and 8088 is shown in Fig. 5.18. The 8086/8088 processors have a number of addressing modes. Addresses are 20 bits in length. Each address is formed in two sections which are then added: a segment address and an offset. The segment address is a full 20 bits, and the offset address is 16 bits.

There are four segment registers: CS, DS, SS, and ES, each containing 16 bits. These registers must be loaded by the program to starting values because the contents of one of these registers are automatically added to each address as it is generated. The contents of the 16-bit segment registers are first shifted left four binary places. (This is equivalent to multiplying the contents of the registers by 16.) Loading the segment registers with 0s would simply place the program and stacks in the first $2^{16}$ words in memory and would effectively remove this feature for simple programs.

When a program is operated, the content of the program counter is automatically added to CS to form each instruction address. Data offsets are automatically added to DS (or ES in special cases), and stack offsets are automatically added to SS. Setting the CS, DS, and SS registers to addresses in different parts of a large memory would cause the instructions, data and stacks to be in different parts of the memory. Setting the CS, SS, and DS registers to the same number would place everything in the same part of memory. Once the segment registers are set, the processor simply generates 16-bit offset addresses in a conventional manner from the instruction words while adding the segment register to each address to form the final 20-bit address. If a program really needed $2^{20}$ addresses, it would be necessary to change the segment registers from time to time to utilize the entire memory.

In effect, the 8086 and 8088 generate conventional 16-bit (offset) addresses by using instruction words and then add the contents of a 20-bit number to each of these offsets to form a 20-bit final address.

Quite a number of addressing modes are used to form the offsets in the 8086/8088 chips. Operands can be in general registers, memory, or I/O ports, and immediate addressing is provided. When 20-bit addresses are generated, the second byte in an instruction word contains the information as to how the 16-bit offset or effective address part of the address is to be calculated. (The first 3 and last 2 bits in this byte provide that information). In general, this section of the address is formed by summing the contents of a displacement (part of the instruction word), an index register, and a base register. Any combination of these three can be used. And this implements, for example, direct addressing, register indirect addressing, and indexed addressing.

The segment registers make it possible to address a $2^{20}$ word memory while only generating 16-bit offset addresses in the instruction words. Another advantage is that use of the segment registers makes a program relocatable in memory. This means that an operating system can place a program in memory where it desires. It can even place several programs in different parts of the memory while the programmer simply writes a program without concern about where it will be run.

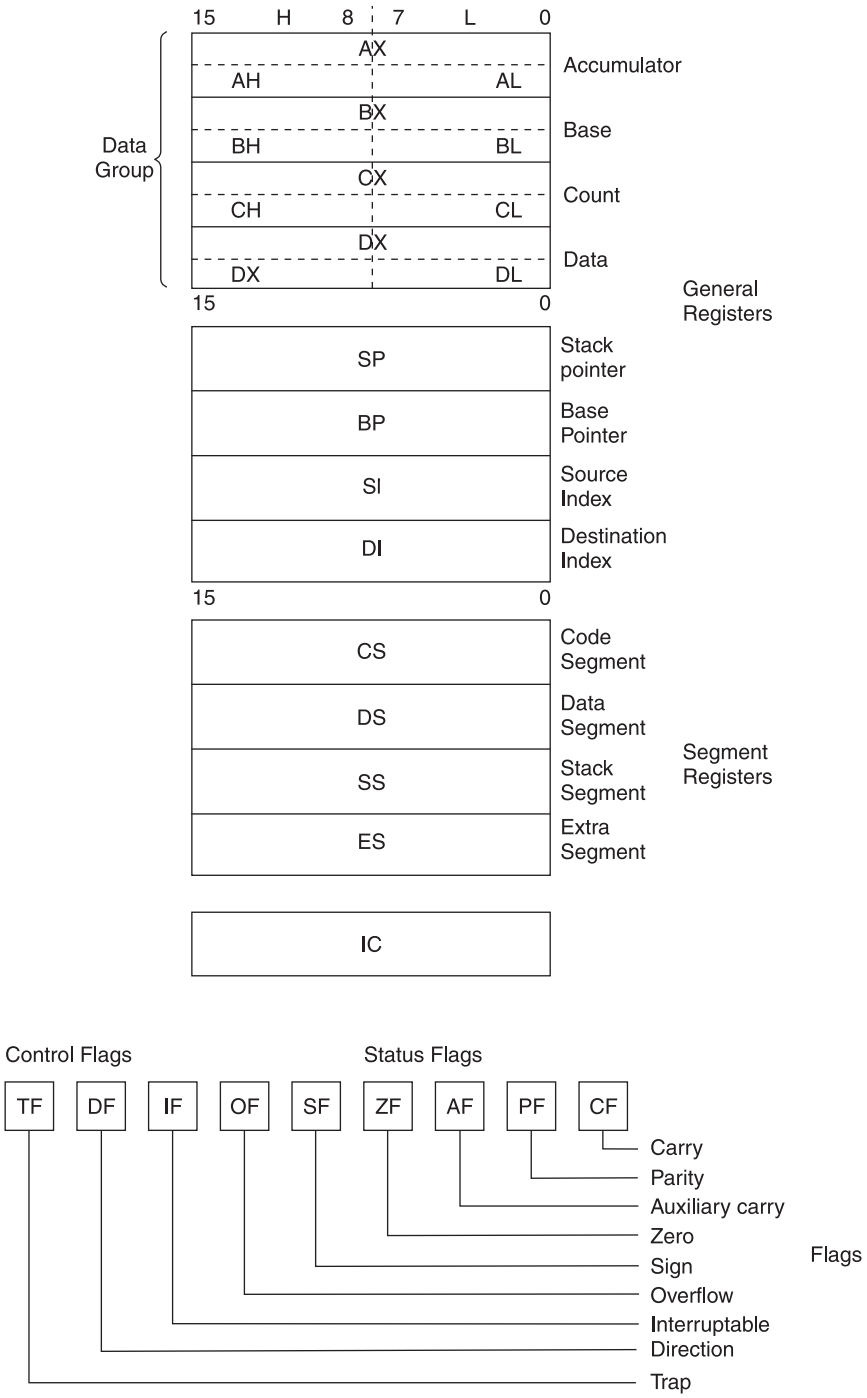



**Fig. 5.18** *General-purpose registers in the 8086 and 8088*

### 5.3.4 68000 Microprocessor

The 68000 microprocessor is a semiconductor chip with a number of support chips such as I/O processors, a floating-point arithmetic chip, and bus-handler chips. The 68000 has 16-bit data paths on its system bus and performs 32-bit arithmetic and logic operations internally. The 68000 microprocessor can directly address 16 Mbytes of memory, having a 24-bit address bus. There are 14 addressing modes and 56 types of instructions. The I/O is memory mapped.

The basic registers in the 68000 are shown in Fig 5.19. The registers are 32 bits, and there are eight data registers along with seven address registers and a program counter. There are actually two stack pointers. A status bit determines whether the 68000 is in the *supervisor* (operating system) *mode or user mode*; this bit also determines which of the two stack pointers are in use. The status register contains 5 bits for conditions codes.

The 68000 *supervisor* and *user* modes are an important feature. There are privileged instructions which can be executed in supervisor mode, but not in user mode. When the supervisor-user mode select bit is a 1, the 68000 uses the supervisor stack pointer and the privileged instructions are available. When the select bit is a 0, the user stack pointer is employed, and certain instructions will not execute.

**PROBE**

What are the important features of the 68000 microprocessor?

Instruction words can be from one word (16 bits) to four words in length. Stacks in the 68000 go from high memory to low memory. So the stack pointer is decremented when data are pushed into a stack and incremented when data are popped from a stack.

### 5.3.5 Microcontrollers

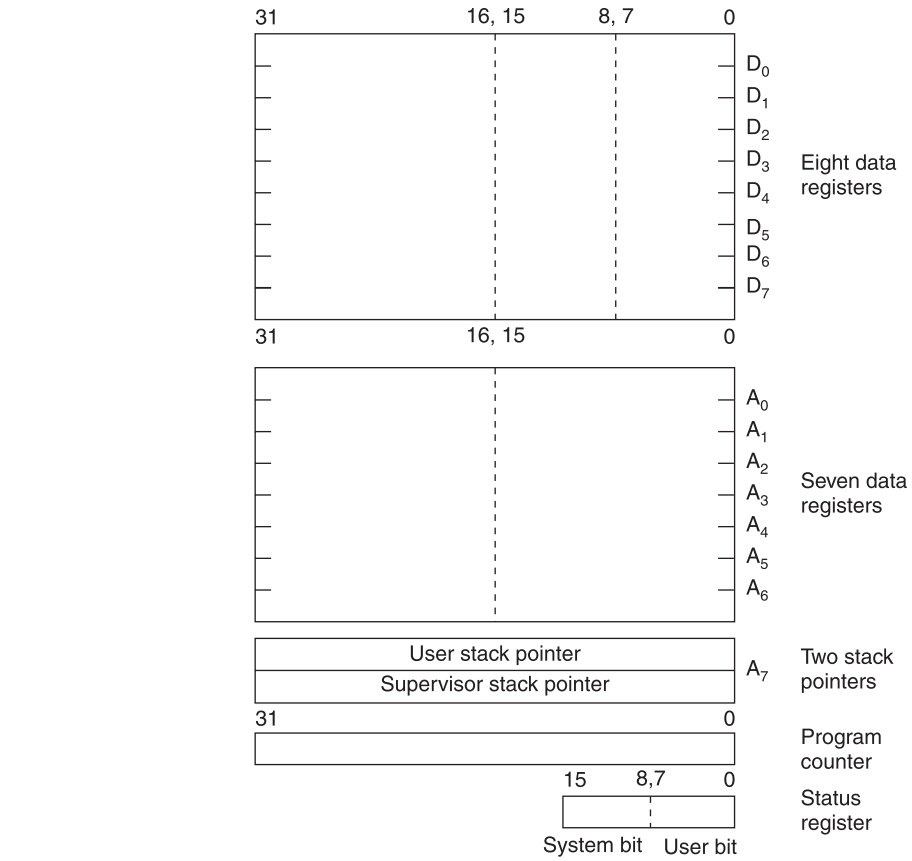
A microprocessor does not have enough memory for program and data storage, neither does it have any input and output devices. Thus, when a microprocessor is used to design a system for specific applications, additional chips are also used to make up a complete system. These extra chips imply additional cost and increased size of the product. For example, when used inside a toy, a designer would like to minimize the size and cost of the electronics equipment inside the toy. Therefore, in such applications, a microcontroller is often used than a microprocessor.

**PROBE**

How are microcontrollers different from microprocessors?

A microcontroller is a chip consisting of a microprocessor, memory, and input/output devices. Microcontrollers vary in their data-handling capacities. Thus there are 4-bit as well as 16-bit microcontrollers.

▪ ***The 8051 Microcontroller*** The MCS-51 is a family of microcontroller ICs developed by Intel Corporation. A dedicated controller readily available for control applications in MCS-51 family is 8051 microcontroller. This is an 8-bit microcontroller. Other 8-bit microcontrollers in this family are 8031, 8751 and 8052 microcontrollers. The 8031 microcontroller does not contain internal ROM. It only uses an external ROM for making a complete system. The 8051 has mask-programmed ROM whereas the 8751 contains the UV erasable on-chip ROM. The 8052 microcontroller offers two enhancements to the 8051 microcontroller. First, there is an additional 128 bytes of on-chip RAM. The second enhancement is an additional 16-bit Timer.



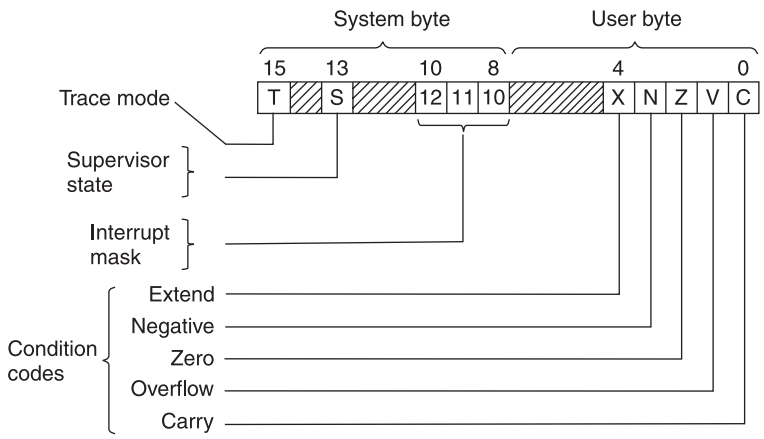




**Fig. 5.19** *68000 programmable registers*

The main hardware features of the 8051 series are the on-chip incorporation of

1. Program ROM (4 K bytes)
2. Data RAM (128 bytes)
3. Special function registers
4. 32 bit programmable I/O Port
5. UART (Universal Asynchronous Receiver Transmitter) for series data input and output
6. Two programmable timers and counters
7. Two external interrupts with priority and masking.

The software features available in 8051 that are not found in the microprocessors are as follows:

1. Bit manipulation
2. Single instruction multiplication
3. Individual program and data memory
4. Four banks of eight temporary registers each and
5. Direct, indirect, paged and relative addressing modes

The main advantage of the 8031 is its low cost and the possibility of using it with popular EPROM's externally. Though the primary 8051 has built-in ROM, this will become an advantage only in volume production, since mask programming of the ROM has to be done at the time of manufacturing itself. The 8751 is ideally suited for single chip systems, since it has a self-contained EPROM. Though it is costlier, the number of ports available will be a total of 32 bits. This is in contrast to the 8031 system wherein the number of port bits is reduced to 16 as the rest of the port lines are used for accessing the external ROM. Addition of external programmable I/O ports such as 8255 to the 8031 can, however, enhance the capacity of the system to more than 40 lines of programmable I/O ports.

▪ ***Hardware Details of 8051*** A schematic block diagram of the 8051 microcontroller is shown in Fig. 5.20. The microcontroller comprises a 4 KB ROM, 128-byte RAM, two timers/counters TIMER0 and TIMER1, four 8-bit I/O ports, serial port, and two interrupt controls. The program counter is 16-bit wide and provides addressing capability up to 64 KB of memory. Two types of memories can be separately addressed. These are program memory (64 K) and data memory (64 K). In 8051 and 8751, the lower 4 K of the program memory is filled by internal ROM and EPROM respectively. By tying the $\overline{EA}$ pin high, the processor can be forced to fetch data from the internal ROM/EPROM for program memory addresses 0 through 4 K. Bus expansion for accessing program memory beyond 4 K is automatic since external instruction fetch occurs automatically when the program counter increases above 4095. If the $\overline{EA}$ point is tied low, all program memory fetches are from external memory.

The data memory is external and can be as wide as 64 KB. When a "MOVX" instruction is executed, external data memory is automatically accessed. One of the important features of the 8051 is the incorporation of the on-chip data RAM and special function registers. The data RAM is 128 bytes wide and a cell can be directly addressed by a single byte address. The internal structure of the RAM and special function registers is shown in Fig 5.21. The RAM cells with addresses in the range 00 H to 7F H can be addressed by direct address. However, the cells with address 00 H to 1F H also function as a set of four register banks with eight registers in each bank. Thus, Bank-0 registers R0 to R7 have addresses 00 H to 07 H; Bank-1 registers R0 to R7 have addresses 08 H to 0F H, and so on. The memory cells 20 H to 2F H have the special feature by which provision is made for bit addressing in this region. One can set or reset a bit by direct addressing of the bit cell, whose address is from 00 H to 7F H. The bit cells 00 H to 07 H forms the RAM Byte cell with address 20H.

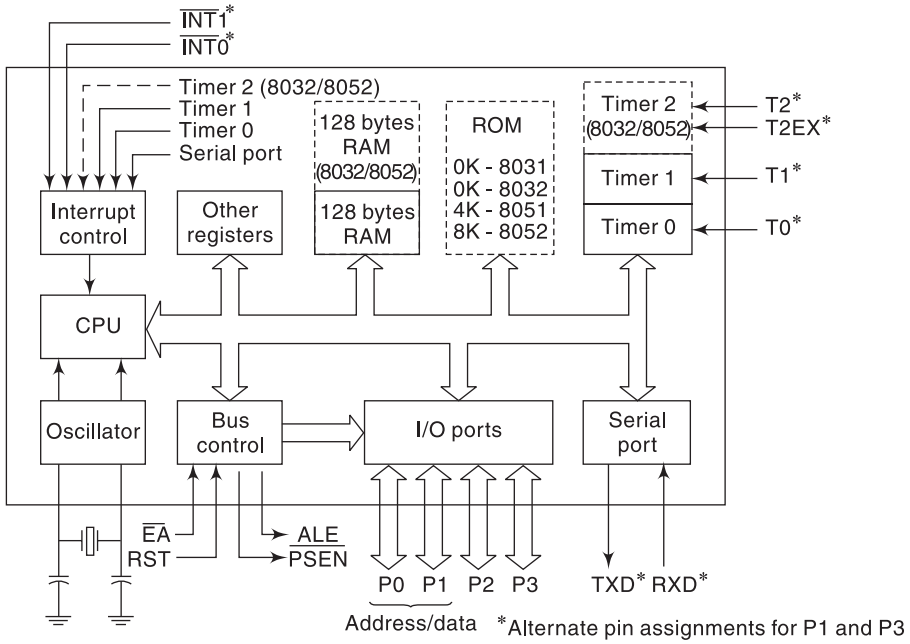



**Fig. 5.20** *Block diagram of 8051 microcontroller*

The special function registers are also part of the internal RAM but with specific addresses as shown in Fig. 5.21. All of these have addresses in the range 80 H to FF H. All of the 128 bytes from 80 H to FF H are not used, and the user has no access for the unused locations. Some of the locations in this region of the RAM also have the facility for "Bit Addressing". The functions of these registers are detailed below:

| | |
|---|---|
| ACC | Accumulator. This is the most active register in the microcontroller. |
| B | B register. This register is mainly used in multiplication and division. |
| PSW | Program Status Word. The bits of this register serve as flags and for switching various banks of the registers R0 to R7. |
| DPTR | Data Pointer. This is a 16-bit address, used for indirect addressing of external memory. DPTR consists of DPH and DPL. |
| P0-P3 | The microcontroller supports four 8-bit ports, viz., P0, P1, P2, and P3.<br>Each bit's contents pass through a buffer to the outgoing line. Because of the priority given to logic "0", all input ports should be initialized to logic "1" before reading a data through the port. When external memories are used, the lines of ports P0 and P2 serve as address and data buses. |
| IP/IE | Interrupt priority control, Interrupt enable control. |
| TMOD/TCON | These registers are used in the control of timers and the mode in which they are used. The 8051 has two timers that can also be used as counters. Each timer consists of two bytes, TH and TL. |
| TH0/TL0 | These registers hold the data in the timers, and consist of higher and lower bytes. |
| TH1/TL1 | |

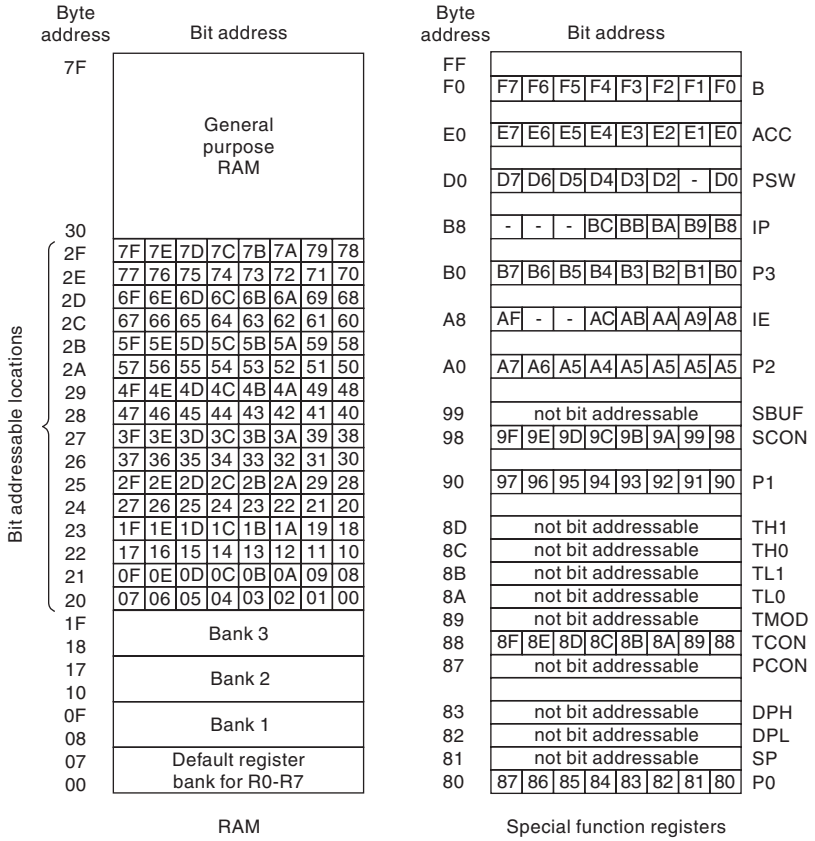



**Fig. 5.21** *Internal RAM and special function registers*

SCON — Control of serial communications. The 8051 has two lines which allow transmission and reception. SCON serves as the communication control.

SBUF — Serial data buffer. This buffer is used to store data that is transmitted serially.

PCON — Power control. Some of the bits of this register enable switching of the CPU and disconnection of it from the clock system, even to the point of stopping its functioning. This switching is done by software.

▪ ***The 8096 Microcontroller*** The MCS-96 family of microcontrollers has been optimized for modern control applications which require high-speed calculations and fast I/O operations. It is a 16-bit microcontroller with a number of dedicated I/O systems and a complete set of 16-bit arithmetic operations. Because of specific requirements of high speed and performance its architecture is different from MCS-51.

**PROBE**

Compare the feature of 8051 and 8096.

The basic features of MCS-96 are

1. 16-bit register based CPU
2. 8K bytes of ROM

3. 256 bytes of RAM and 232 bytes of Special Function Registers
4. Hardware multiply and divide instructions
5. Six addressing modes
6. High speed I/O with 4 dedicated and 4 programmable I/O lines
7. 10-bit A/D converter
8. Full duplex serial port
9. 40 I/O pins available
10. Two, 16-bit timers/counters
11. Watchdog timer to recover from errors
12. Programmable 8-source, 2-priority level interrupt system
13. Available in 48-pin DIP or 68-pin flat pack

The 8096 is used for sophisticated real time control and used in high-end instruments and instrumentation.

▪ ***CPU*** The CPU has a 16-bit ALU. Since the ALU does not have an accumulator, the location in the register file are used as source and destination. Hence, the ALU unit of 8096 microcontroller is called RALU (Register Arithmetic and Logic Unit).

Out of 64K memory, the first 256 bytes can be used as data/program memory which is a feature of Princeton architecture. When accessed as data memory (internal memory), the first 24 bytes correspond to SFRs (Special Function Registers) and the remaining 232 bytes can be used as general-purpose RAM. The upper 16 bytes of the register file has a power-down mode, so that data in it can be saved on power down. When accessed as program memory (external memory), it is reserved for use by Intel development systems.

The internal program memory of 8 Kbytes is from 2080 H to 3FFF H. The external memory can be expanded by using the ports P3 and P4 for the multiplexed address and data bus. For the reconstruction of ports P3 and P4, the locations 1FFF H to 2000 H has been kept. The vectored interrupt lie between 2000 H and 2011 H. Therefore, the memory 00 H to FF H and 1FFE H to 2011 H has a special purpose. On reset, the PC points to the location 2080H.

▪ ***Interrupts*** Eight interrupt sources are available on 8096, apart from TRAP. These are the external interrupt, serial port interrupt, software time interrupt, the high-speed input interrupt, the high speed output interrupt, the high-speed data available interrupt, the A/D conversion complete interrupt and the time over-flow interrupt. These interrupts can be masked, and can be assigned a priority. These interrupts are vectored.

▪ ***Timers*** There are two 16-bit timers, T1 and T2. The timer T1 is used to synchronize events to real time. It is clocked once every 8 state times. T2 can be clocked externally and synchronize events to external occurrences. Both the timers can cause the timer interrupt and also set I/O flags.

▪ ***High-speed Inputs*** These are four lines (HSI 0 to HSI 3) can be used to record events with respect to timer Tl. Up to 8 events can be recorded in a FIFO RAM, which holds the time value from timer as well as the state of HSI inputs at that instant. The HSI interrupt indicates that either the FIFO is full or the holding register is loaded.

▪ ***High-speed Outputs*** These are used to trigger events at specific times. These reduce the CPU overheads. These events could be to start A/D conversion, reset the timer T2, set software flags, generate interrupts or switch the 6 output lines (HSO 0 to HSO 5). The HSO 4 and HSO 5 share the same pins with HSI 2 and HSI 3. Up to 8 events can be kept pending at any time. This unit is controlled by a Content Addressable Memory (CAM) which has 8 location and stores the time for taking action, the type of action

and whether timer T1 or T2 is used as reference. Four software timers can be programmed through HSO to provide interrupts at predetermined times.

- ***Analog Inputs*** Any one of the analog inputs (ACH 0 to ACH 7) can be sampled and converted to a 10-bit digital value. The analog inputs share the pins with port P0. The results are put in the 2-byte A/D result register.

- ***Pulse Width Modulation Output*** The digital-to-analog conversion is available in the form of a PWM output. The output waveform is a pulse with a frequency of 256 state times. The width of the pulse is the analog equivalent of an 8-bit digital value. The PWM output shares the same pin as port P2 bit 5.

- ***Serial Port*** The serial port is similar to that of MCS-51. It is a full-duplex port with four modes of operation. The serial port shares pins with port P2, bits 0 and 1. In any one of the modes of operation, it can be used for serial I/O using shift registers and latches.

- ***I/O Ports*** There are five ports. Some ports are inputs only, some ports are outputs only, while some are bidirectional and some share alternate functions.

Port 0 is an input port and also be used for A/D converter inputs. Port 1 is bidirectional. Port 2 has input, output as well as bidirectional pins. Ports 3 and 4 are bidirectional ports and can be used for expanding the memory.

- ***Watchdog Timer*** If the software fails to reset the watchdog every 64 K states, the 8096 is reset. This is a graceful method of recovering from software failure.

## 5.4 INTRODUCTION TO COMPUTERS

**FLASH CARD**

**What You Learn Here**
Illustrate and explain the different parts of digital computers.

Any computer system has essentially three important parts, namely, input device, central processing device, and output device. The central processing unit (CPU) itself has three parts, namely, memory unit, control unit, and arithmetic logic unit (ALU). These three units along with the input and output devices form the five important components of any computer system. In addition to the above, computers also employ secondary storage devices (or auxiliary storage or backing storage), which are used for holding data or instructions on a long term basis. Figure 5.22 shows a computer system with peripheral devices.

**PROBE**

Name three parts of the CPU.

The input devices are used to transfer the information into the memory unit of the computer. Information from the memory can be transferred to the ALU where comparisons or calculations are done and the results are sent back to the memory unit. Thus, the memory unit is used to store the set of instructions that determines the operations to be carried out on a set of data as well as the data on which these operations are performed. The memory unit is also called the *main memory* or the *immediate access store*. It accepts instructions held in store, interprets these instructions and process them for execution by appropriate parts of the system in the correct sequence. The control unit ensures that, according to the stored instructions, the right operation is done on the right data at the right time. The results that are stored in the memory can be transformed into a form that can be understood by us by means of an output device.

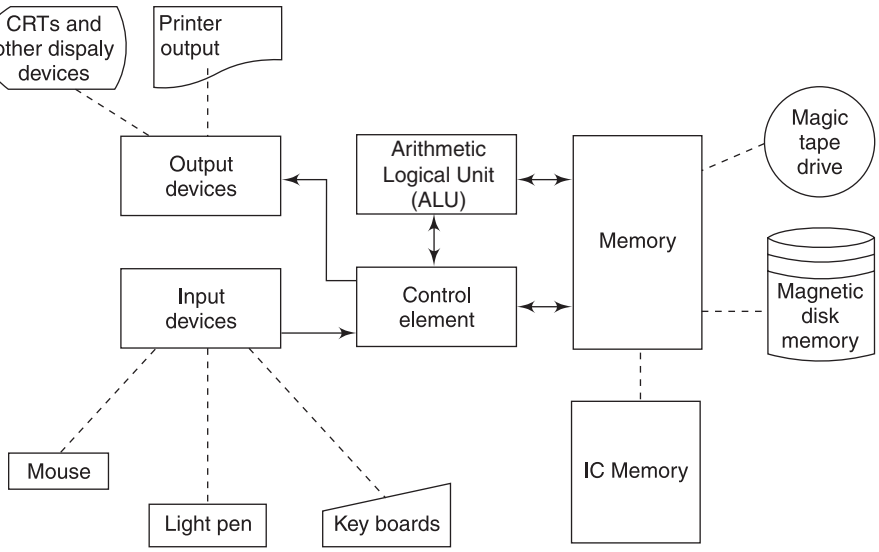



**Fig. 5.22** *Block diagram of a digital computer system*

## Key Terms

6800 Microprocessor
68000 Microprocessor
8085 Microprocessor
8085A Microprocessor
8086 Microprocessors
8088 Microprocessors
8051 Microcontroller
8096 Microcontroller
Accumulator
Address bus
Address lines
Analog inputs
Arithmetic Logic Unit (ALU)
Condition code register
Content Addressable Memory (CAM)
Control element
Data bus
Decrementer
Dynamic Random Access Memories (DRAMs)
Electrically Alterable PROM (EAPROM)
Electrically Erasable PROM (EEPROM)
Erasable Programmable Read Only Memory (EPROM)
Field Programmable Logic Arrays (FPLAs)
High-speed inputs
High-speed outputs
I/O Ports
Incrementer
Input devices
Instruction
Instruction decode and control
Instruction register
Interrupts
Linear addressing
Linear array
Matrix addressing
Memory
Memory addressing
Microcontrollers
Microprocessors
Op code
Output devices
Program
Program counter
Programmable Logic Arrays (PLAs)
Programmable Read Only Memories (PROMs)
Pulse width modulation output
Random Access Memory (RAM)
RALU (Register Arithmetic and Logic Unit)
READ
Read Only Memories (ROMs)
Semiconductor memories
Serial Input Data (SID)
Serial Output Data (SOD)
Serial Port
SFRs (Special Function Registers)
Stack pointer
Timers
Ultraviolet Light Erasable PROM (UVEPROM)
Watchdog timer
WRITE

## Review Questions

1. What are the applications of memory devices? ○●●
2. What are the two major types of semiconductor memories? ○●●
3. What is meant by memory addressing? Why is it needed? ○●●
4. How many address lines are needed to locate a cell in a 64-element square array? ○○●
5. What is referred to as linear addressing? ○●●
6. What are the different classifications of ROMs? ○●●
7. For what applications are permanent memories like ROMs used? Why? ○●●
8. Explain how data can be written into a PROM? ○●●
9. What are the two types of EPROMs? ○●●
10. What is the basic difference between a ROM and a RAM? ○●●
11. What is meant by refreshing the memory? Where is it employed and why? ○●●
12. Draw the block diagram of a typical MOS static memory and explain the organisation. ○●●
13. What are the advantages of a dynamic RAM over static RAM? ○●●
14. In what way a PLA is different from a ROM? ○●●
15. What are the two types of PLAs? ○○●
16. Explain how a PLA can be used to realize a logical expression.
17. What is an arithmetic logic unit (ALU)? ○○●
18. What is a microprocessor? ○○●
19. Explain with neat diagram the general architecture of a microprocessor. ○●●
20. Draw the block diagram of the 8085A microprocessor and explain the various elements available. ○●●
21. Explain the architecture of the 6800 microprocessor using a neat diagram. ○●●
22. Give the salient features of 8086 and 8088 microprocessors. ○●●
23. Give the salient features of the 68000 microprocessor. ○●●
24. What is a microcontroller? What is its principal advantage? ○●●
25. Explain the features of 8051. ○●●
26. How many times/counters are there in 8051? What is the bit length of the counters? ○●●
27. State the function of SCON (Serial Port Control register)? ○○●
28. Compare the features of 8051 and 8096. ○●●
29. What is the use of watchdog timer in 8096? ○●●
30. Give the different parts of a digital computer and explain. ○○●

**Note:** ○○● Level 1 & Level 2 category
○●● Level 3 & Level 4 category
●●● Level 5 & Level 6 category

## Objective-Type Questions

1. The number of hardware interrupts (which require an external signal to interrupt) present in an 8085 microprocessor is
   (a) 1 (b) 4 (c) 5 (d) 13
2. In a microprocessor, the service routine for a certain interrupt starts from a fixed location of memory which cannot be externally set, but the interrupt can be delayed or rejected. Such an interrupt is
   (a) non-maskable and non-vectored (b) maskable and non-vectored
   (c) non-maskable and vectored (d) maskable and vectored

**3.** In the 8085 microprocessor, the RST6 instruction transfers the program execution to the following location:
(a) 30H (b) 24H (c) 48H (d) 60H

**4.** The contents of register (B) and accumulator (A) of the 8085 microprocessor are 49H and 3AH respectively. The contents of A and the status of the carry flag (CY) and sign flag (S) after executing SUB B instructions are
(a) A = F1, CY = 1, S = 1 (b) A = 0F, CY = 1, S = 1
(c) A = F0, CY = 0, S = 0 (d) A = 1F, CY = 1, S = 1

**5.** In an 8085 microprocessor, the instruction CMO B has been executed while the content of the accumulator is less than that of the register B. As a result,
(a) the carry flag will be set but the zero flag will be reset
(b) the carry flag will be reset but the zero flag will be set
(c) both the carry flag and zero flag will be reset
(d) both the carry flag and zero flag will be set

**6.** The number of memory cycles required to execute the following 8085 instructions
(i) LDA 3000H, (ii) LXID, F0F1 H would be
(a) 2 for (i) and 2 for (ii) (b) 4 for (i) and 3 for (ii)
(c) 3 for(i) and 3 for (ii) (d) 3 for (i) and 4 for (ii)

**7.** The contents of the accumulator just after execution of the ADD instruction in line 4 will be
(a) C3H (b) EAH (c) DCH (d) 69H

**8.** The frequency of 8085 microprocessor is 2 MHz. If the instruction LXIH, 2562H is executed, then time taken by instruction cycle is
(a) 2.5 μ sec (b) 4 μ sec (c) 5 μ sec (d) 10 μ sec

**9.** How many and which types of machine cycles are needed to execute PUSH PSW by the 8085 microprocessor?
(a) 2, opcode fetch and 2 memory write (b) 3, opcode fetch and 2 memory write
(c) 3, opcode fetch and 2 memory read (d) 3, opcode fetch, memory read and memory write

**10.** Which of the following statements is true?
(a) ROM is a Read/Write memory.
(b) PC points to the last instruction that was executed.
(c) Stack works on the principle of LIFO.
(d) All instructions affect the flags.

**11.** In an 8085 microprocessor system with memory mapped I/O,
(a) I/O devices have 8-bit addresses
(b) I/O devices are accessed using IN and OUT instructions
(c) there can be a maximum of 256 input devices and 256 output devices
(d) arithmetic and logic operations can be directly preformed with the I/O date

**12.** In a microprocessor, the register which holds the address of the next instruction to be fetched is the
(a) accumulator (b) program counter (c) stack pointer (d) instructor register

**13.** In a microcomputer, wait states are used to
(a) make the processor wait during a DMA operation
(b) make the processor wait during a power interrupt processing
(c) make the processor wait during a power shutdown
(d) interface slow peripherals to the processor

**14.** The decimal equivalent of the hexadecimal number E5 is
(a) 279 (b) 229 (c) 427 (d) 3000

**15.** In an 8085 microprocessor, the instruction CMP B has been executed while the contents of accumulator is less than that of the register B. As a result, the carry flag and zero flag will be respectively
(a) set, reset (b) reset, set (c) reset, reset (d) set, set

**16.** A PLA can be
(a) as a microprocessor (b) as a dynamic memory
(c) to realize a sequential logic (d) to realize a combinational logic. **[GATE 1994]**

**17.** A dynamic RAM consists of
(a) 6 transistors (b) 2 transistors and 2 capacitors
(c) 1 transistor and 1 capacitor (d) 2 capacitors only **[GATE 1994]**

**18.** Each cell in a static Random Access Memory contains
(a) 6 MOS transistors (b) 4 MOS transistors and 2 capacitors
(c) 2 MOS transistors and 2 capacitors (d) 1 MOS transistors and 1 capacitors **[GATE 1996]**

**19.** When a CPU is interrupted, it
(a) stops execution of instructions
(b) acknowledges interrupt and branches to a subroutine
(c) acknowledge interrupt and continues
(d) acknowledges interrupt and waits for the next instruction from the interrupting device
**[GATE 1995]**