

13

Microprocessors

The microprocessor is the heart of a microcomputer system. In fact, it forms the central processing unit of any microcomputer and has been rightly referred to as the *computer on a chip*. This chapter gives an introduction to microprocessor fundamentals, followed by application-relevant information, such as salient features, pin configuration, internal architecture, instruction set, etc., of popular brands of eight-bit, 16-bit, 32-bit and 64-bit microprocessors from international giants like INTEL, MOTOROLA and ZILOG.

13.1 Introduction to Microprocessors

A microprocessor is a programmable device that accepts binary data from an input device, processes the data according to the instructions stored in the memory and provides results as output. In other words, the microprocessor executes the program stored in the memory and transfers data to and from the outside world through I/O ports. Any microprocessor-based system essentially comprises three parts, namely the microprocessor, the memory and peripheral I/O devices. The microprocessor is generally referred to as the heart of the system as it performs all the operations and also controls the rest of the system. The three parts are interconnected by the data bus, the address bus and the control bus (Fig. 13.1).

The *memory* stores the binary instructions and data for the microprocessor. The memory can be classified as the primary or main memory and secondary memory. Read/write memory (R/WM) and read only memory (ROM) are examples of primary memory and are used for executing and storing programs. Magnetic disks and tapes are examples of secondary memory. They are used to store programs and results after the completion of program execution. Microprocessors do not execute programs stored in the secondary memory directly. Instead, they are first copied on to the R/W primary memory.

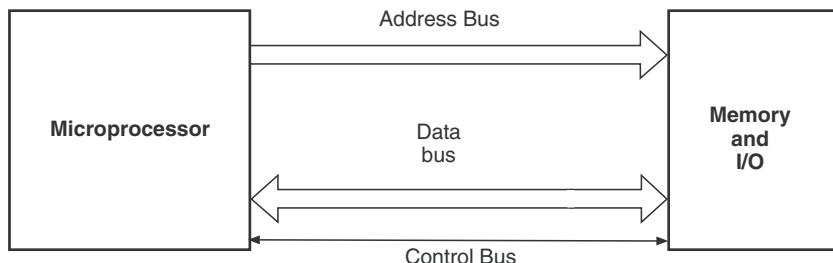


Figure 13.1 Microprocessor-based system.

Input/output devices are means through which the microprocessor interacts with the outside world. The commonly used input devices include keyboards, A/D converters, switches, cameras, scanners, microphones and so on. LEDs, seven-segment displays, LCD displays, printers and monitors are some of the commonly used output devices.

A *bus* is basically a communication link between the processing unit and the peripheral devices. It is a group of wires that carry information in the form of bits. The *address bus* is unidirectional and is used by the CPU to send out the address of the memory location to be accessed. It is also used by the CPU to select a particular input or output port. It may consist of 8, 16, 20 or an even greater number of parallel lines. The number of bits in the address bus determines the maximum number of data locations in the memory that can be accessed. A 16-bit address bus, for instance, can access 2^{16} data locations. It is labelled as A_0, \dots, A_{n-1} , where n is the width (in bits) of the address bus.

The *data bus* is bidirectional, that is, data flow occurs both to and from the microprocessor and peripherals. Data bus size has a considerable influence on the computer architecture, as parameters such as the word length and the quantum of data that can be manipulated at a time are determined by the size of the data bus. There is an internal data bus, which may not be of the same width as the external data bus that connects the microprocessor to I/O and memory. The size of the internal data bus determines the largest number that can be processed by the microprocessor in a single operation. The largest number that can be processed, for instance, by a microprocessor having a 16-bit internal data bus is 65535. The data bus is labelled as D_0, \dots, D_{n-1} , where n is the data bus width (in bits).

The *control bus* contains a number of individual lines carrying synchronizing signals. The term ‘bus’ would normally imply a group of lines working in unison. The control bus (if we call it a bus) sends out control signals to memory, I/O ports and other peripheral devices to ensure proper operation. It carries control signals such as memory read, memory write, read input port, write output port, hold, interrupt, etc. For instance, if it is desired to read the contents of a particular memory location, the CPU first sends out the address of that location on the address bus and a ‘memory read’ control signal on the control bus. The memory responds by outputting data stored in the addressed memory location onto the data bus. ‘Interrupt’ tells the CPU that an external device needs to be read or serviced. ‘Hold’ allows a device such as the direct memory access (DMA) controller to take over the address and data buses.

Figure 13.2 shows the bus interface between the microprocessor and its peripheral devices. The microprocessor considered in the diagram is an eight-bit microprocessor such as Intel’s 8085.

Microprocessor-based systems can be categorized as general-purpose reprogrammable systems and embedded systems. Reprogrammable systems include microcomputers and mainframe and miniframe computers where microprocessors are used for computing and data processing. In embedded systems, they perform a specific task and are not available for reprogramming to the end-user. Examples of these systems include mobile phones, washing machines, microwave ovens, dish washers and so on.

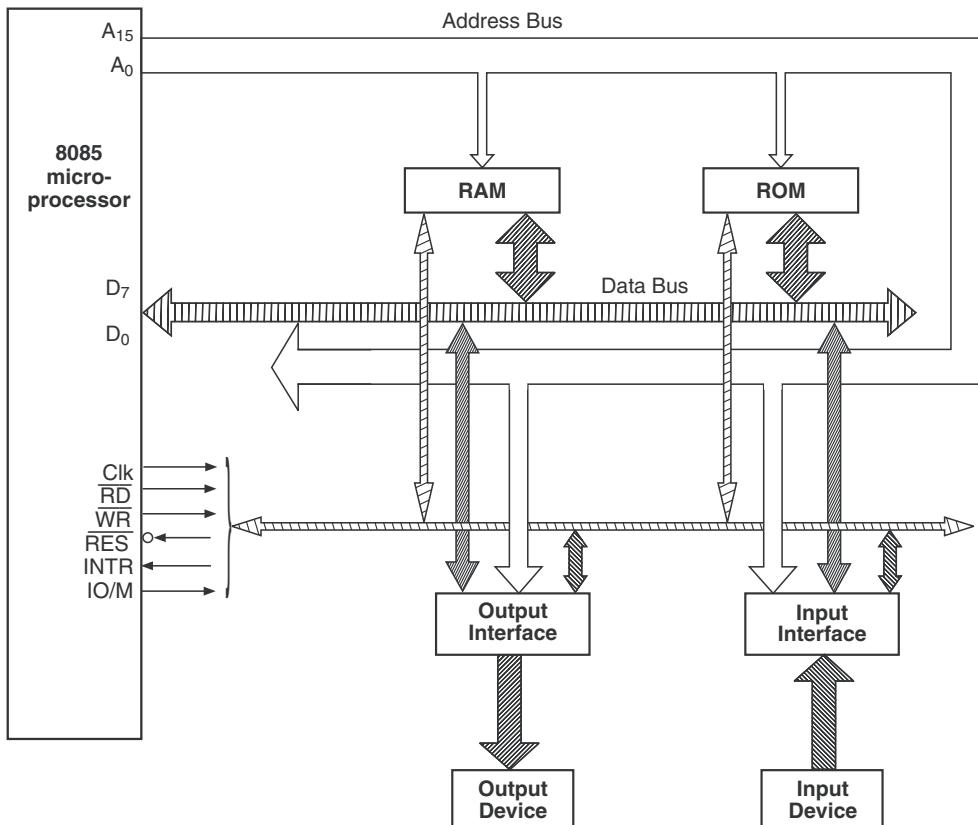


Figure 13.2 Bus interface between the microprocessor and its peripheral devices.

In most of these systems, the microprocessor, memory and I/O ports are combined onto one chip, known as the *microcontroller*. Microcontrollers are discussed in detail in Chapter 14.

13.2 Evolution of Microprocessors

The evolution of microprocessors has been known to follow Moore's law, which suggests that the complexity of an integrated circuit, with respect to the minimum component cost, doubles every 24 months. This rule has been generally followed, since the humble beginning of microprocessors as the drivers for calculators to the present-day scenario where every system, from the largest mainframes to the smallest handheld computers, uses a microprocessor at its core.

The first microprocessor was introduced in 1971 by the Intel Corporation. It was a four-bit microprocessor, Intel 4004. Other four-bit microprocessors developed were Intel 4040 by Intel, PPS-4 by Rockwell International, T3472 by Toshiba and so on. The first eight-bit microprocessor, named Intel 8008, was also developed by Intel in the year 1972. All these microprocessors were made using PMOS technology. The first microprocessor using NMOS technology was Intel 8080, developed by Intel in the

year 1973. Intel 8080 was followed by Intel 8085 in the year 1975, which became very popular. Other popular eight-bit microprocessors were Zilog's Z80 (1976) and Z800, Motorola's MC6800 (1974) and MC6809 (1978), National Semiconductor's NSC 800, RCA's 1802 (1976) and so on.

The first multichip 16-bit microprocessor was National Semiconductor's IMP-16, introduced in 1973. The first 16-bit single-chip microprocessor was Texas Instrument's TMS 9900. Intel's first 16-bit microprocessor was Intel 8086 introduced in the year 1978. Other 16-bit microprocessors developed by Intel were Intel 80186 (1982), Intel 8088, Intel 80188 and Intel 80286 (1982). Other popular 16-bit microprocessors include Motorola's 68000 (1979), 68010 and 68012, Zilog's Z8000, Texas Instruments TMS 9900 series and so on.

32-bit microprocessors came into existence in the 1980s. The world's first single-chip 32-bit microprocessor was introduced by AT&T Bell Labs in the year 1980. It was named BELLMAC-32A. The first 32-bit processor introduced by Intel was iapx 432, introduced in 1981. The more popular 32-bit microprocessor was Intel 80386, introduced by Intel in 1985. It was widely used for desktop computers. The 32-bit microprocessor family of Intel includes Intel 486, Pentium, Pentium Pro, Pentium II, Pentium III and Pentium IV. AMD's K5, K6 and K7, Motorola's 68020 (1985), 68030 and 68040, National Semiconductor's 32032 and 32332 and Zilog's Z80000 are other popular 32-bit microprocessors. All these microprocessors are based on CISC (Complex Instruction Set Computers) architecture. The first commercial RISC (Reduced Instruction Set Computers) design was released by MIPS Technologies, the 32-bit R2000. Some of the popular RISC processors include Intel's 80860 and 80960, Motorola's 88100 and Motorola's, IBM and Apple's PowerPC series of microprocessors.

While 64-bit microprocessor designs have been in use in several markets since the early 1990s, the early 2000s have seen the introduction of 64-bit microchips targeted at the PC market. Some of the popular 64-bit microprocessors are AMD's AMD64 (2003) and Intel's x86-64 chips. Popular 64-bit RISC processors include SUN's ULTRASPARC, PowerPC 620, Intel's Itanium, MIPS R4000, R5000, R10000 and R12000 and so on.

13.3 Inside a Microprocessor

Figure 13.3 shows a simplified typical schematic arrangement of a microprocessor. The figure shown is a generalized one and is not the actual structure of any of the commercially available microprocessors. The important functional blocks include the arithmetic logic unit (ALU), the register file and the control unit.

These functional blocks are briefly described in the following paragraphs.

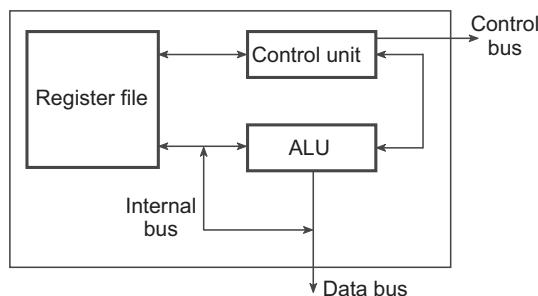


Figure 13.3 Typical schematic arrangement of a microprocessor.

13.3.1 Arithmetic Logic Unit (ALU)

The arithmetic logic unit (ALU) is the core component of all microprocessors. It performs the entire integer arithmetic and bit-wise logical operations of the microprocessor. ALU is a combinational logic circuit and has two data input lines, a data output line and a status line. It gets data from the registers of the microprocessor, processes the data according to the instructions from the control unit and stores the results in its output registers. All modern ALUs use binary data in 2's complement format.

The integer arithmetic operations performed by the ALU include addition and subtraction. It performs AND, OR, NOT and EXCLUSIVE-OR logical operations. Some 16-bit, 32-bit and 64-bit microprocessors also perform multiplication and division operations. In other microprocessors, the multiplication and division operations are performed by writing algorithms using addition and subtraction operations. Some such algorithms were outlined in Chapter 3 on digital arithmetic. ALU also performs the bit-shifting operations and the comparison of data operations.

13.3.2 Register File

The register file comprises various registers used primarily to store data, addresses and status information during the execution of a program. Registers are sequential logic devices built using flip-flops. Some of the commonly found registers in most of the microprocessors include the *program counter*, *instruction registers*, *buffer registers*, *the status register*, *the stack pointer*, *general-purpose registers* and *temporary registers*.

13.3.2.1 Program Counter

The *program counter* is a register that stores the address of the next instruction to be executed and hence plays a central role in controlling the sequence of machine instructions that the processor executes. After the instruction is read into the memory, the program counter is automatically incremented by '1'. This is of course on the assumption that the instructions are executed sequentially. Its contents are affected by jump and call instructions. In the case of a jump instruction, the program counter is first loaded with the new address and then incremented thereafter until another jump instruction is encountered. When the microprocessor receives an instruction to begin a subroutine, the contents of the program counter are incremented by '1' and are saved in the stack. The program counter is loaded with the address of the first instruction of the subroutine. Its contents are incremented by '1' until a return instruction is encountered. The saved stack contents are then loaded into the program counter and the program continues, executing each instruction sequentially until another jump instruction or a subroutine call is encountered. The interrupt process also alters the contents of the program counter.

13.3.2.2 Instruction Register

The *instruction register* stores the code of the instruction currently being executed. The control unit extracts the operation code from the instruction register, which determines the sequence of signals necessary to perform the processing required by the instruction.

13.3.2.3 Buffer Register

Buffer registers interface the microprocessor with its memory system. The two standard buffer registers are the *memory address register* (MAR) and the *memory buffer register* (MBR). The MAR is connected

to the address pins of the microprocessor and holds the absolute memory address of the data or instruction to be accessed. The MBR, also known as the memory data register, is connected to the data pins of the microprocessor. It stores all data written to and read from memory.

13.3.2.4 Status Register

The *status register* stores the status outputs of the result of an operation and gives additional information about the result of an ALU operation. The status of bits stored in the status register tells about the occurrence or nonoccurrence of different conditions, and one or more bits may be updated at the end of an operation. Each bit is a Boolean flag representing a particular condition. The most common conditions are the carry, overflow, zero and negative. For instance, a '1' in the carry status bit position shows that the result of the operation generates a carry. The significance of the status register lies in the fact that the condition code set by the status of different bits in the status register forms the basis of decision-making by the microprocessor during the execution of a program.

13.3.2.5 Stack Pointer

The *stack pointer* is a register used to store the address of a memory location belonging to the most recent entry in the stack. In fact, a stack is a block of memory locations designated for temporary storage of data. It is used to save data of another general-purpose register during execution of a subroutine or when an interrupt is serviced. The data are moved from a general register to the stack by a PUSH instruction at the beginning of a subroutine call, and back to the general register by a POP instruction at the end of the subroutine call. Microprocessors use a stack because it is faster to move data using PUSH and POP instructions than to move data to/from memory using a MOVE instruction.

13.3.2.6 General-purpose Registers

There is a set of registers for general-purpose use, designated as *general-purpose registers*. They are used explicitly to store data and address information. Data registers are used for arithmetic operations, while the address registers are used for indexing and indirect addressing. These enhance the processing speed of the microprocessor by avoiding a large number of external memory read/write operations while an ALU operation is being performed, as it is much easier and faster to read from or write into an internal register than to read from or write into an external memory location. Earlier microprocessors had only one register called the *accumulator* for ALU operations. It needed at least four assembly language instructions to perform a simple addition, including carrying data from an external memory location to the accumulator, adding the contents of the accumulator to those of another memory location, storing the result in the accumulator and transferring the contents of the accumulator back to the external memory location. With the availability of a greater number of general-purpose registers, it would be possible to perform many ALU operations without even a need to store data in external memory.

13.3.2.7 Temporary Registers

These are used when data have to be stored during the execution of a machine instruction. They are completely hidden from the user of the microprocessor.

13.3.3 Control Unit

The *control unit* governs and coordinates the activities of different sections of the processor and I/O devices. It is responsible for controlling the cycle of fetching machine instructions from memory and executing them. It also coordinates the activities of input and output devices. It is undoubtedly the most complex of all functional blocks of the microprocessor and occupies most of the chip area. The control unit is a sequential logic circuit, which steps the processor through a sequence of synchronized operations. It sends a stream of control signals and timed pulses to the components and external pins of the microprocessor. As an illustration, to execute an instruction from the memory, the control unit sends out a 'read' command to the memory and reads the instruction (or data) that comes back on the data bus. The control unit then decodes the instruction and sends appropriate signals to the ALU, the general-purpose registers, the multiplexers, the demultiplexers, the program counter and so on. If the instruction was to store data in the memory, the control unit sends out the address of the memory location on the address bus, the data to be stored on the data bus and a 'write' command on a control line.

Control units are categorized into two types depending upon the way they are built. These include *hard-wired* and *microcoded* control units. Hard-wired controllers are sequential logic circuits, the states of which correspond to the phases of the instruction execution cycle. In the case of hard-wired controllers, there is an electronic circuitry in the control unit to generate control signals for each instruction. They are very compact and fast, but are difficult to design. This design is also known as RISC (Reduced Instruction Set Computer) design. Microcoded control units are easy to design, and execution of an instruction in this case involves executing a microprogram consisting of a sequence of microinstructions. This design is also known as CISC (Complex Instruction Set Computer) design. Microcoded control units offer more flexibility than do hard-wired control units but they are comparatively slower than the latter.

Figure 13.4 shows a more descriptive block diagram of a microprocessor. Multiplexers and demultiplexers do not represent primary functions and are there to facilitate the flow of data between different blocks and also between different blocks and the outside world.

13.4 Basic Microprocessor Instructions

Microprocessors perform various basic operations including data transfer instructions, arithmetic instructions, logic instructions, control transfer instructions and machine control instructions.

13.4.1 Data Transfer Instructions

Data transfer instructions transfer data from one location designated as the source location to another location designated as the destination. The data transfer could take place from one register to another, from one memory location to another memory location, from a memory location to a register or from a register to a memory location, and so on. In fact, they are more correctly referred to as data movement operations as the contents of the source are not transferred but are copied into the destination register without modifying the contents of the source. It may be mentioned here that these operations do not affect the flags. Data transfer operations of the 8085 microprocessor are of three types, namely MOVE, LOAD and STORE:

MOV destination, source	Copy data from the source to the destination location
LDA address	Copy the data byte at the memory location specified by the 16-bit address into the accumulator
STA address	Copy the data from the accumulator to the memory location specified by the 16-bit address

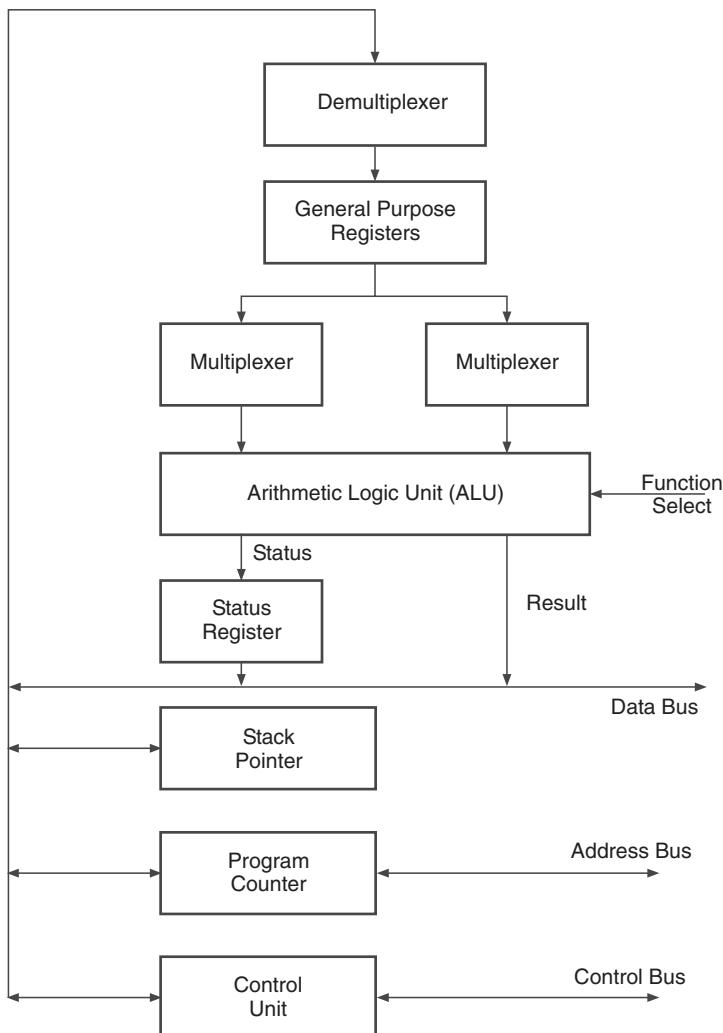


Figure 13.4 Descriptive block diagram of a microprocessor.

13.4.2 Arithmetic Instructions

Arithmetic instructions performed by microprocessors include addition, subtraction, multiplication, division, comparison, negation, increment and decrement. It may be mentioned here that most of the eight-bit microprocessors do not support multiplication and division operations. These operations are supported by the 16-bit and 32-bit microprocessors. The arithmetic operations supported by the 8085 microprocessor are addition, subtraction, increment and decrement operations. Examples are as follows:

- ADD R** Adds the contents of the register to the accumulator
- ADI eight-bit** Adds the eight-bit data to the accumulator

SUB R	Subtracts the contents of the register from the accumulator
SUI eight-bit	Subtracts eight-bit data from the contents of the accumulator
INR R	Increments the contents of the register
DCR R	Decrements the contents of the register

13.4.3 Logic Instructions

Microprocessors can perform all the logic functions of hard-wired logic. The basic logic operations performed by all microprocessors are AND, OR, NOT and EXCLUSIVE-OR. Other logic operations include ‘shift’ and ‘rotate’ operations. All these operations are performed on a bit-for-bit basis on bytes or words. For instance, 11111111 AND 10111010 equals 10111010, and 11111111 OR 10111010 equals 11111111. Some microprocessors also perform bit-level instructions such as ‘set bit’, ‘clear bit’ and ‘complement bit’ operations. It may be mentioned that logic operations always clear the carry and overflow flags, while the other flags change to reflect the condition of the result.

The basic shift operations are the ‘shift left’ and ‘shift right’ operations. In the *shift left* operation, also known as the arithmetic shift left, all bits are shifted one position to the left, with the rightmost bit set to ‘0’ and the leftmost bit transferred to the carry position in the status register. In the *shift right* operation, also known as logic shift right, all bits are shifted one bit position to the right, with the leftmost bit set to ‘0’ and the rightmost bit transferred to the carry position in the status register. If in the shift right operation the leftmost bit is left unchanged, it is called arithmetic shift right. In a ‘rotate’ operation, the bits are circulated back into the register. Carry may or may not be included. As an illustration, in a ‘rotate left’ operation without carry, the leftmost bit goes to the rightmost bit position, and, in a ‘rotate right’ with carry included, the rightmost bit goes to the carry position and the carry bit takes the position of the leftmost bit.

Examples of logic instructions performed by the 8085 microprocessor include the following:

ANA R/M	Logically AND the contents of the register/memory with the contents of the accumulator
ANI eight-bit	Logically AND the eight-bit data with the contents of the accumulator
ORA R/M	Logically OR the contents of the register/memory with the contents of the accumulator
ORI eight-bit	Logically OR the eight-bit data with the contents of the accumulator
XRA R/M	Logically EXCLUSIVE-OR the contents of the register memory with the contents of the accumulator
XRI eight-bit	Logically EXCLUSIVE-OR the eight-bit data with the contents of the accumulator
CMA	Complement the contents of the accumulator
RLC	Rotate each bit in the accumulator to the left position
RRC	Rotate each bit in the accumulator to the right position

13.4.4 Control Transfer or Branch or Program Control Instructions

Microprocessors execute machine codes from one memory location to the next, that is, they execute instructions in a sequential manner. Branch instructions change the flow of the program either unconditionally or under certain test conditions. Branch instructions include ‘jump’, ‘call’, ‘return’ and ‘interrupt’.

‘Jump’ instructions are of two types, namely ‘unconditional jump’ instructions and ‘conditional jump’ instructions. If the microprocessor is so instructed as to load a new address in the program

counter and start executing instructions at that address, it is termed an unconditional jump. In the case of a conditional jump, the program counter is loaded with a new instruction address only if and when certain conditions are established by the microprocessor after reading the appropriate status register bits. ‘Call’ instructions transfer the flow of the program to a subroutine. The ‘call’ instruction differs from the ‘jump’ instruction as ‘call’ saves a return address (the address of the program counter plus one) on the stack. The ‘return’ instruction returns control to the instruction whose address was stored in the stack when the ‘call’ instruction was encountered. ‘Interrupt’ is a hardware-generated call (externally driven from a hardware signal) or a software-generated call (internally derived from the execution of an instruction or by some internal event). Examples of transfer control instructions of the 8085 microprocessor are as follows:

JMP 16-bit address	Change the program sequence to the location specified by the 16-bit address
JZ 16-bit address	Change the program sequence to the location specified by the 16-bit address if a zero flag is set
JC 16-bit address	Change the program sequence to the location specified by the 16-bit address if a carry flag is set
CALL 16-bit address	Change the program sequence to the location of the subroutine specified by the 16-bit address
RET	Return to the calling program

13.4.5 Machine Control Instructions

Machine control instructions include HALT and NOP instructions. Machine control instructions performed by the 8085 microprocessor include the following:

HLT	Stop processing and wait
NOP	No operation

13.5 Addressing Modes

Microprocessors perform operations on data stored in the register or memory. These data are specified in the operand field of the instruction. The data can be specified in various ways as a direct data value or stored in some register or memory location, and so on. These are referred to as the *addressing modes* of the microprocessor. In other words, the addressing mode as expressed in the instruction tells us how and from where the microprocessor can get the data to act upon. Addressing modes are of direct relevance to compiler writers and to programmers writing the code in assembly language.

Different microprocessor architectures provide a variety of addressing modes. RISC microprocessors have far fewer addressing modes than CISC microprocessors. The most commonly used addressing modes are absolute, immediate, register direct, register indirect, indexed, program counter relative, implicit and relative addressing modes. They account for more than 90 % of the total addressing modes.

13.5.1 Absolute or Memory Direct Addressing Mode

In *absolute addressing mode*, the data are accessed by specifying their address in the memory [Fig. 13.5(a)]. This mode is useful for accessing fixed memory locations, such as memory mapped I/O devices. For example, the instruction MOV A, 30H in the 8085 microprocessor moves the contents of memory location 30H into the accumulator [Fig. 13.5(b)]. In this case the accumulator has the value 07H.

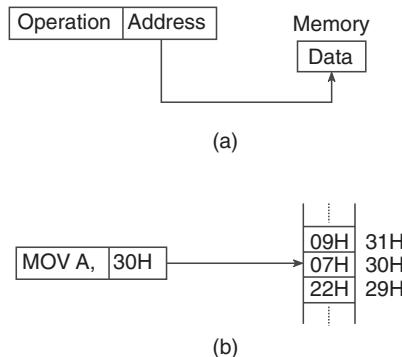


Figure 13.5 Absolute addressing mode.

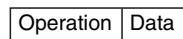


Figure 13.6 Immediate addressing mode.

13.5.2 Immediate Addressing Mode

In *immediate addressing mode* the value of the operand is held within the instruction itself (Fig. 13.6). This mode is useful for accessing constant values in a program. It is faster than the absolute addressing mode and requires less memory space. For example, the instruction MVI A, #30H moves the data value 30H into the accumulator. The sign # in the instruction tells the assembler that the addressing mode used is immediate.

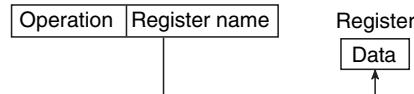
13.5.3 Register Direct Addressing Mode

In *register direct addressing mode*, data are accessed by specifying the register name in which they are stored [Fig. 13.7(a)]. Operations on registers are very fast, and hence instructions in this mode require less time than absolute addressing mode instructions. As an example, the instruction MOV A, R1 in the 8051 microprocessor moves the contents of register R1 into the accumulator [Fig. 13.7(b)]. The contents of the accumulator after the instruction are 06H.

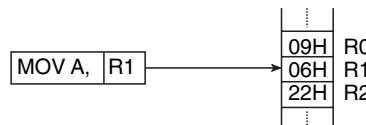
13.5.4 Register Indirect Addressing Mode

In all the modes discussed so far, either the value of the data or their location is directly specified. The *indirect addressing mode* uses a register to hold the actual address where the data are stored. That is, in this case the memory location of the data is stored in a register [Fig. 13.8(a)]. In other words, in indirect addressing mode, the address is specified indirectly and has to be looked up. This addressing mode is useful when implementing the pointer data type of high-level language.

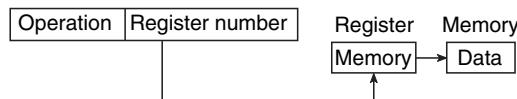
In the 8085 microprocessor, the R0 and R1 registers are used as an eight-bit index and the D PTR as a 16-bit index. The mnemonic symbol used for indirect addressing is @. As an example, the instruction



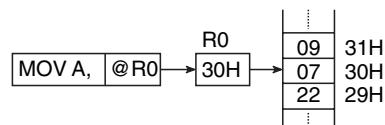
(a)



(b)

Figure 13.7 Register direct addressing mode.

(a)



(b)

Figure 13.8 Register indirect addressing mode.

`MOV A, @R0` moves the contents of the memory location whose address is stored in `R0` into the accumulator. The value of the accumulator in this example is `07H` [Fig. 13.8(b)]. This addressing mode can also be enhanced with an offset for accessing data structures in data space memory. This is referred to as *register indirect* with displacement. As an example, the instruction `MOVC A, @A+DPTR` copies the code byte at the memory address formed by adding the contents of `A` and `DPTR` to `A`.

13.5.5 Indexed Addressing Mode

In the indexed addressing mode, the address is obtained by adding the contents of a register to a constant (Fig. 13.9). The instruction ‘move the contents of accumulator `A` to the memory location whose address is given by the contents of register `1` plus `5`’ is an example of indexed addressing. The indexed addressing mode is useful whenever the absolute location of the data is not known until the program is running. This addressing mode is used to access a continuous table or array of data items stored in memory. The content of the constant gives the starting address, while the contents of the

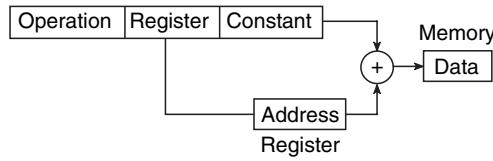


Figure 13.9 Indexed addressing mode.

register determine the element of the array or table to be accessed. If the program counter is used in the indexed addressing mode, it is known as the *program counter relative addressing mode*.

13.5.6 Implicit Addressing Mode and Relative Addressing Mode

In *implicit addressing mode*, no operand is used in the instruction and the location of the operand is obvious from the instruction itself. Examples include ‘clear carry flag’, ‘return from subroutine’ and so on.

The relative addressing mode is used for ‘jump’ and ‘branch’ instructions only. In this, a displacement is added to the address in the program counter and the next instruction is fetched from the new address in the program counter. This mode is particularly useful in connection with conditional jumps.

13.6 Microprocessor Selection

There are thousands of microprocessors available on the market. Selection of the right microprocessor for a given application is not an easy task and cannot take place in a vacuum; it must be done with the application in mind. Not only this, the quantity to be produced and the experience and capabilities of the designers must also be considered. The selection process begins with the definition of the application to be followed by matching a given processor with the well-defined application.

13.6.1 Selection Criteria

Sometimes it becomes difficult to extract microprocessor requirements from the application at the early stage of the project. This may be due to several factors, which include the following:

1. Speed compatibility of the microprocessor with peripherals.
2. The time-critical behaviour of the application.
3. The size of the program required to implement certain functions is not known in advance.

These ambiguities serve as a warning that perhaps the project is not adequately defined for the microprocessor selection to be made. Factors to be considered while selecting the microprocessor are price, power consumption, performance, availability, software support and code density.

13.6.1.1 Price

Price is one of the important factors that is considered by designers to evaluate a processor. It assumes more importance for those embedded systems that have price constraint.

13.6.1.2 Power consumption

Power consumption is an important factor for battery-operated systems. The power consumption of a microprocessor varies with the supply voltage (square of supply voltage), speed (linearly) and with the software the chip is running. The bus structure of the processor and its interconnection with the memory ICs should also be looked into.

13.6.1.3 Performance

Processors that are good for one task may not be suitable for another. It is therefore very important to define the processor requirements for the given application. These include the estimated size and complexity of the program, speed requirements (time-critical functions), the language to be used, the arithmetic functions needed, memory requirements (ROM, RAM and mass storage), I/O requirements and interrupt source and response time required.

After defining the application requirements, they should be matched with those that a processor can offer. Table 13.1 enumerates the main parameters of the processor to be considered while selecting it for a particular application.

13.6.1.4 Availability

Before zeroing onto a particular microprocessor, it is important to ensure that it is easily available.

Table 13.1 Microprocessor characteristics checklist.

Instruction set	Data types: bit operations, long words Arithmetic functions: multiply and divide Encoding efficiency: RISC or CISC
Register set	Number of registers Width of registers Number of special-purpose registers
Addressing	Number of modes: direct, indirect, etc. Segmented or linear addressing Memory and I/O address ranges (memory mapped/I/O mapped) Memory management
Bus and control signals	Bus timings Interrupts DMA/bus arbitration control signals Data and address bus width Clock speed and bus cycle time
Miscellaneous	Prefetch (instruction queue length), cache memory Coprocessor support: floating point, I/O processors Power requirements
Nontechnical considerations	Documentation quality and availability Development tools: emulators, debuggers and logic analysers Software support: OS, compiler, assembler, utilities

13.6.1.5 Software support

The associated software with the microprocessor, such as the debugger, compiler and operating system, constitutes one of the factors that needs to be considered.

13.6.1.6 Code density

The code density is the ratio between the size of the source code and the size of the object code. The smaller the object code, the better is the code density. Processors having high code densities require less memory to execute the code. RISC processors have poor code density compared with CISC processors.

Moreover, there is seldom one right microprocessor for a given task. There are several chips that can be used for a given task. Factors such as past experience, the market reputation of the processor and availability are considered before making the final decision.

13.6.2 Microprocessor Selection Table for Common Applications

Single-chip microcomputers are commonly used in control applications. In more complex control applications requiring large amounts of I/O, memory or high-speed processing, eight-bit or 16-bit microprocessors are used. Data processing applications, which require more memory and I/O, use a PC. The 32-bit and 64-bit microprocessors are used in systems that require high performance such as engineering workstations and in multi-user systems. Table 13.2 gives typical microprocessor types for various application classes.

Table 13.2 Microprocessor types for various application classes.

Application classes			Typical device types					
Type	Speed and complexity	Type example	Single-chip microcomputer			Microprocessor		
			four-bit	eight-bit	16-bit	eight-bit	16-bit	32/64-bit
Control Data processing	Low	Automatic thermostat	✓	✓				
	Medium	Digital multimeter		✓		✓		
	High	Engine control		✓	✓	✓	✓	
	Low	Home computer				✓		
	Medium	Mid-range PC				✓	✓	
	High	Engineering workstation, multiuser computer					✓	✓

13.7 Programming Microprocessors

Microprocessors execute programs stored in the memory in the form of a sequence of binary digits. Programmers do not write the program in binary form but write it either in the form of a text file containing an assembly-language source code or using a high-level language. Programs such as editor, assembler, linker and debugger enable the user to write the program in assembly language, convert it into binary code and debug the binary code. *Editor* is a program that allows the user to enter, modify and store a group of instructions or text under a file name. The assembly language source code is translated into an object code by a program called *assembler*. *Linker* converts the output of the assembler into a format that can be executed by the microprocessor. The *debugger* is a program that allows the user to test and debug the object file.

Programming in assembly language produces a code that is fast and takes up little memory. However, it is difficult to write large programs using assembly language. Another disadvantage of assembly language programming is that it is specific to a particular microprocessor. High-level language programming overcomes these problems. Some of the popular high-level languages used include C, C++, Pascal and so on. Compiler programs are primarily used to translate the source code from a high-level language to a lower-level language (e.g. assembly language or machine language). Figures 13.10(a) and (b) show the various steps involved in executing assembly language programs and programs written in high-level languages respectively.

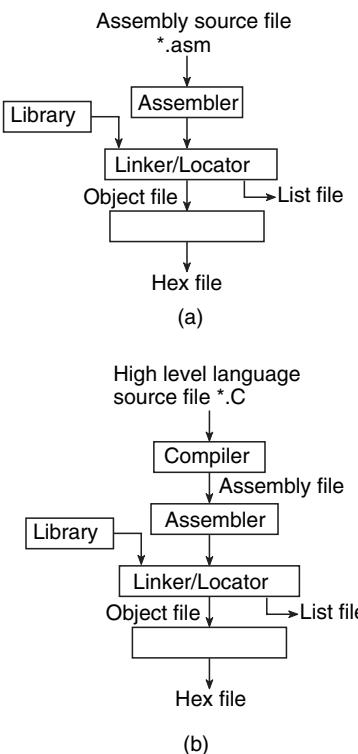


Figure 13.10 (a) Various steps involved in executing assembly language programs and (b) various steps involved in executing programs written in high-level languages.

13.8 RISC Versus CISC Processors

CISC is an acronym for Complex Instruction Set Computer. The primary goal of CISC architecture is to complete a task in as few lines of assembly as possible. This is achieved by building processor hardware that is capable of understanding and executing a series of complex operations. In this case, each instruction can execute several low-level instructions. One of the primary advantages of this system is that the compiler has to do very little work to translate a high-level language statement into assembly. Because the length of the code is relatively short, very little RAM is required to store instructions. In a nutshell, the emphasis is to build complex instructions directly into the hardware. Examples of CISC processors are the CDC 6600, System/360, VAX, PDP-11, the Motorola 68000 family, and Intel and AMD x86 CPUs.

RISC is an acronym for Reduced Instruction Set Computer. This type of microprocessor emphasizes simplicity and efficiency. RISC designs start with a necessary and sufficient instruction set. The objective of any RISC architecture is to maximize speed by reducing clock cycles per instruction. Almost all computations can be done from a few simple operations. The goal of RISC architecture is to maximize the effective speed of a design by performing infrequent operations in software and frequent functions in hardware, thus obtaining a net performance gain.

To understand this phenomenon, consider any assembly-level language program. It has been observed that it uses the MOV instruction much more frequently than the MUL instruction. Therefore, if the architectural design implements MOV in hardware and MUL in software, there will be a considerable gain in speed, which is the basic feature of RISC technology. Examples of RISC processors include Sun's SPARC, IBM and Motorola's PowerPCs, and ARM-based processors.

The salient features of a RISC processor are as follows:

1. The microprocessor is designed using hard-wired control. For example, one bit can be dedicated for one instruction. Generally, variable-length instruction formats require microcode design. All RISC instructions have fixed formats, so no microcode is required.
2. The RISC microprocessor executes most of the instructions in a single clock cycle. This is due to the fact that they are implemented in hardware.
3. The instruction set typically includes only register-to-register load and store.
4. The instructions have a simple format with few addressing modes.
5. The RISC microprocessor has several general-purpose registers and large cache memories, which support the very fast access of data.
6. The RISC microprocessor processes several instructions simultaneously and so includes pipelining.
7. The software can take advantage of more concurrency.

13.9 Eight-Bit Microprocessors

This section describes the block diagram, pin-out diagram, salient features and instruction set of the most popular eight-bit microprocessors, namely 8085 of Intel, Z80 of Zilog and 6800 of Motorola.

13.9.1 8085 Microprocessor

Figure 13.11 gives the pin-out configuration and Fig. 13.12 shows a block diagram of the 8085 microprocessor. Table 13.3 lists the pin details.

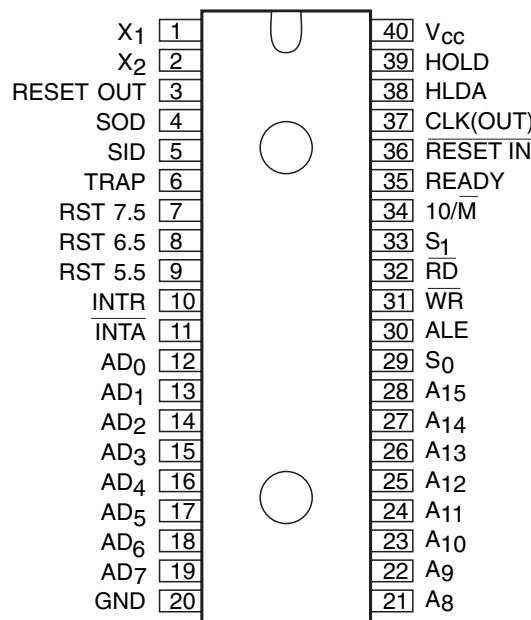


Figure 13.11 Pin-out configuration of 8085.

13.9.1.1 8085 Registers

The 8085 microprocessor registers include an eight-bit accumulator, an eight-bit flag register (five one-bit flags, namely sign, zero, auxiliary carry, parity and carry), eight-bit B and C registers (which can be used as one 16-bit BC register pair), eight-bit D and E registers (which can be used as one 16-bit DE register pair), eight-bit H and L registers (which can be used as one 16-bit HL register pair), a 16-bit stack pointer and a 16-bit program counter.

13.9.1.2 Addressing Modes

8085 has four addressing modes. These include register addressing, register indirect addressing, direct addressing mode and immediate addressing mode.

13.9.1.3 8085 Instructions

An instruction is a binary pattern designed inside a microprocessor to perform a specific function. The entire group of instructions a microprocessor can perform is referred to as its *instruction set*. An *instruction cycle* is defined as the time required to complete the execution of an instruction. An 8085 instruction cycle consists of 1–6 machine cycles. A *machine cycle* is defined as the time required to complete one operation of accessing memory, I/O and so on. This will comprise 3–6 *T*-states, which is defined as one subdivision of the operation performed in one clock period.

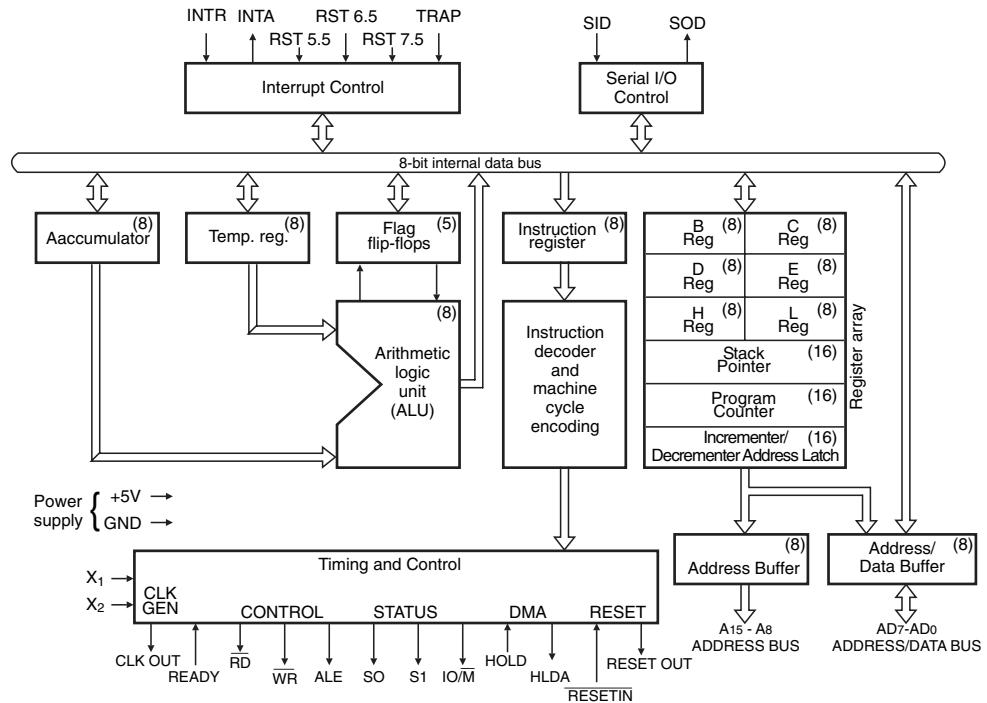


Figure 13.12 Block diagram of 8085.

Table 13.3 Pin details of 8085.

Signals	Description
Address bus (12–19, 21–29)	A 16-bit address bus. The lower eight bits are multiplexed with the data bus. The most significant eight bits of the memory address (or I/O address) are denoted by A ₈ -A ₁₅ . The lower eight bits of the memory address (or I/O address) appear on the multiplexed address/data bus (AD ₀ -AD ₇) for the first clock cycle of the machine cycle. It then becomes the data bus during the second and third clock cycles
Data bus (12–19)	Eight-bit data bus is multiplexed with lower eight bits of the address bus (AD ₀ -AD ₇)
Control and status signals	
ALE (Address Latch Enable) (30)	It is a positive-going pulse during the first clock state of the machine cycle that indicates that the bits on AD ₇ -AD ₀ are address bits. It is used to latch the low-order address on the on-chip latch from the multiplexed bus
READ (\overline{RD}) (32)	A LOW on \overline{RD} indicates that the selected memory or I/O device is ready to be read and the data bus is available for data transfer
WRITE (\overline{WR}) (31)	A LOW on \overline{WR} indicates that data on the data bus are to be written into a selected memory or I/O location. Data are set up at the trailing edge of the \overline{WR} signal

(continued overleaf)

Table 13.3 (continued).

Signals	Description
IO/M(34)	This is a status signal that is used to differentiate between I/O and memory operations
S_1 and $S_{0(29,33)}$	These are status signals and can identify various operations
Power supply and clock frequency	
V_{CC} (40)	+ 5 V
V_{SS}	Ground
X_1, X_2 (20)	A crystal, LC or RC network is connected at these two pins to drive the internal clock generator. X_1 can also be an external clock input from a logic gate. The frequency is internally divided by 2 to give the internal operating frequency of the processor. The crystal frequency must be at least 1 MHz and must be twice the desired internal clock frequency
CLK OUT – clock output (37)	This output signal can be used as a system clock for devices on the board. The period of CLK is twice the X_1, X_2 input period
Interrupts and other operations: 8085 has five interrupt signals	
INTR: INTerrupt Request (10)	This is a general-purpose interrupt signal. The microprocessor issues an interrupt acknowledge signal (INTA) when the interrupt is requested
RST 7.5 (7)	These are restart interrupts. These are vectored interrupts and transfer the program control to specific memory locations
RST 6.5 (8)	
RST 5.5 (9)	
TRAP (6)	It is a nonmaskable interrupt and has the highest priority
In addition to these interrupts RESET, HOLD and READY pins accept externally initiated signals as inputs	
HOLD (39)	
HLDA (38)	A HOLD signal indicates that another master device is requesting the use of data and address buses. The microprocessor, upon receiving the HOLD request, will relinquish the use of the bus after completion of the current bus transfer. It sends the HOLD ACKNOWLEDGE (HLDA) signal, indicating that it will relinquish the bus in the next clock cycle
READY (35)	A READY signal is used to delay the microprocessor READ or WRITE cycles until a slow-responding peripheral is ready to send or accept data. If READY is HIGH during the READ or WRITE cycle, it indicates that the memory or peripheral is ready to send or receive data. If READY is LOW, the processor will wait for an integral number of clock cycles for READY to go to HIGH
RSEST IN (36)	
RESET OUT (3)	A LOW on the RESET IN pin causes the program counter to be set to zero, the buses are tristated and the microprocessor is reset. RESET OUT indicates that the microprocessor is being reset
Serial I/O parts	
SID (5)	Serial Input Data
SOD (4)	Serial Output Data

13.9.2 Motorola 6800 Microprocessor

This is an eight-bit microprocessor housed in a 40-pin dual in-line package (DIP) and released at the same time as Intel 8080. An important feature of 6800 is that it does not have I/O instructions, and therefore 6800-based systems had to use memory-mapped I/O for input/output capabilities. Motorola 6800 started a family of 680X microcontrollers and microprocessors, many of which are in use today. 6800 microprocessors can operate at a maximum frequency of 2 MHz. Figure 13.13 shows a block schematic representation of the internal architecture of the Motorola 6800 microprocessor.

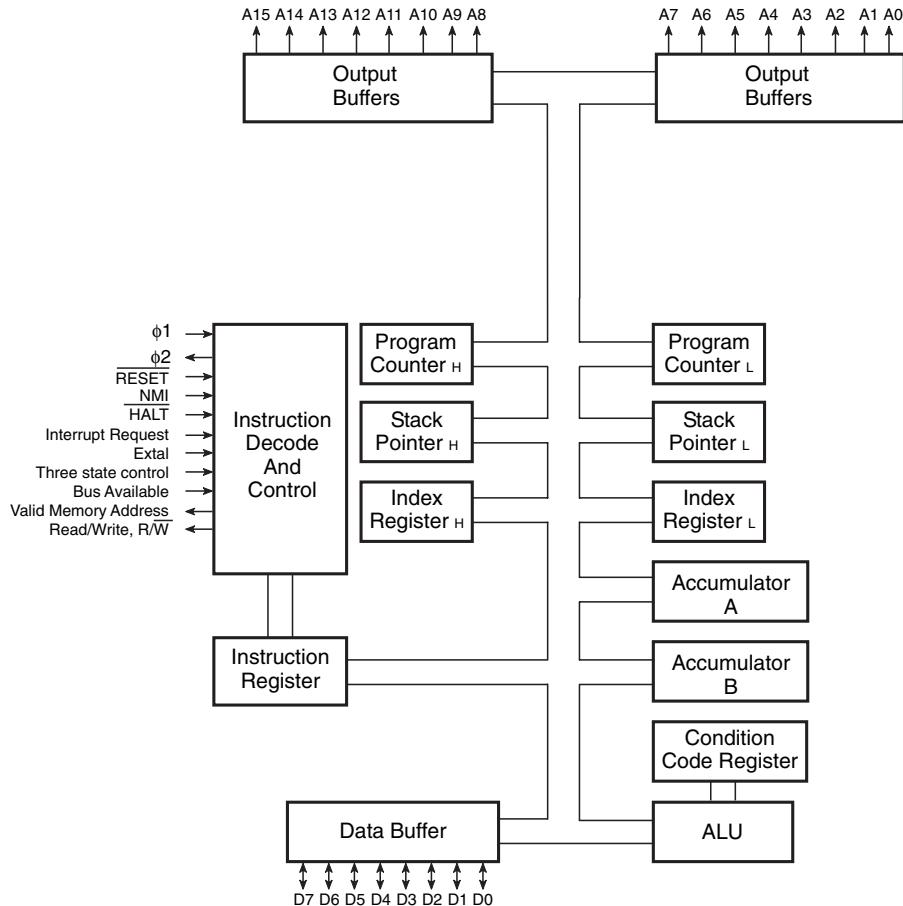


Figure 13.13 Block diagram of 6800.

13.9.2.1 6800 Registers

The 6800 microprocessors have six internal registers, namely accumulator A (ACCA), accumulator B (ACCB), an index (IX), a program counter, a stack pointer (SP) and a condition code register.

13.9.2.2 Addressing Modes

It has the implied addressing mode, accumulator addressing mode, immediate addressing mode, direct addressing mode, extended addressing mode, relative addressing mode and indexed addressing mode.

13.9.2.3 Instruction Set

The 6800 instruction set consists of 72 instructions. It supports data moving instructions, arithmetic instructions (add, subtract, negate, increment, decrement and compare), logic instructions (AND,

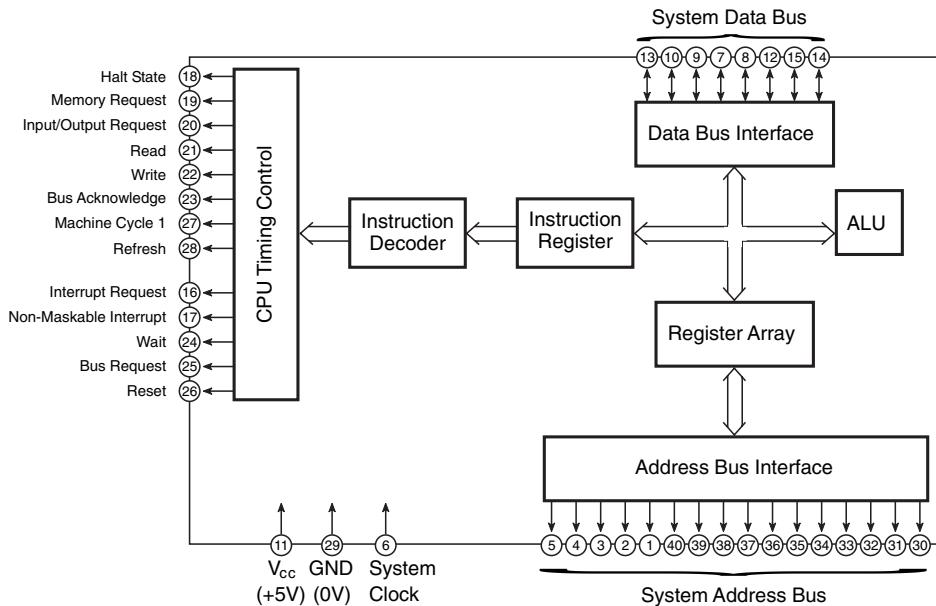


Figure 13.14 Block diagram of Z80.

OR, EXCLUSIVE-OR, complement and shift/rotate), control transfer (conditional, unconditional, call subroutine and return from subroutine) and others – clear/set condition flags, bit test, stack operations, software interrupt, etc.

13.9.3 Zilog Z80 Microprocessor

The Zilog Z80 microprocessor is an eight-bit processor that is object-code compatible with Intel 8080. It is available in industry-standard 40-pin dual in-line and 44-pin chip carrier packages. The maximum operating frequency is 2.5 MHz. Figure 13.14 shows a block diagram of Z80.

13.9.3.1 Z80 registers

The Z80 microprocessor has registers compatible with the 8080 microprocessor as well as some other registers. The 8080-compatible registers include the accumulator, flag register (F), general-purpose registers (six programmable general-purpose registers designated B, C, D, E, H and L), stack pointer (SP) and program counter. The registers introduced with Z80 are the alternate accumulator register (A'), the alternate flag register (F'), the alternate B, C, D, E, H and L registers (represented as A', B', C', D', E', H' and L'), the index registers (IX and IY), the interrupt vector register (I) and the memory refresh register (R).

13.9.3.2 Instruction set

The Z80 microprocessor has 158 instructions. They perform data copy (transfer) or load operations, arithmetic, logic operations, bit manipulation, branch operations and machine control operations.

13.10 16-Bit Microprocessors

Eight-bit microprocessors are limited in their speed (the number of instructions that can be executed in 1 s), directly addressable memory, data handling capability, etc. Advances in semiconductor technology have made it possible for the manufacturers to develop 16-bit, 32-bit, 64-bit and even-larger-bit microprocessors. This section describes the block diagram, pin-out configuration and salient features of some of the most popular 16-bit microprocessors including 8086 of Intel and Motorola's MC68000.

13.10.1 8086 Microprocessor

This is a 16-bit microprocessor introduced by Intel. It was designed using HMOS technology and contains approximately 29 000 transistors. It has a maximum operating frequency of 10 MHz. The 8086, 8088, 80186 and 80286 microprocessors have the same basic set of registers and addressing modes. The 8086 microprocessor is available in DIP, CeraDIP and PLCC packages. Figure 13.15 shows a block diagram of 8086.

13.10.1.1 8086 registers

8086 has four segment registers and other general-purpose registers. The segment registers include code segment (CS), stack segment (SS), data segment (DS) and extra segment (ES). The general-purpose registers of 8086 include the accumulator register, base register, count register, data register, stack pointer (SP), base pointer (BP), source index (SI) and destination index (DI). The stack pointer, base pointer, source index and destination index registers are both general and index registers. Other registers include the instruction pointer (IP) and the flag register containing nine one-bit flags.

13.10.1.2 Addressing modes

The addressing modes of 8086 are implied addressing, register addressing, immediate addressing, direct addressing, register indirect addressing, base addressing, indexed addressing, base indexed addressing and base indexed with displacement addressing.

13.10.1.3 Internal Architecture and Pin-out Configuration

The internal functions of the 8086 processor are portioned logically into two processing units. The first is the bus interface unit (BIU) and the second is the execution unit (EU), as shown in Fig. 13.15. The BIU provides the functions related to instruction fetching and queuing, operand fetch and store and address relocation. It also provides the basic bus control. The EU receives prefetched instructions from the BIU queue and provides unrellocated operand addresses to the BIU.

13.10.1.4 Instruction set

The instruction set includes the following: data transfer operations, arithmetic operations, logical instructions, string manipulation instructions, control transfer instructions, processor control instructions and input/output operations.

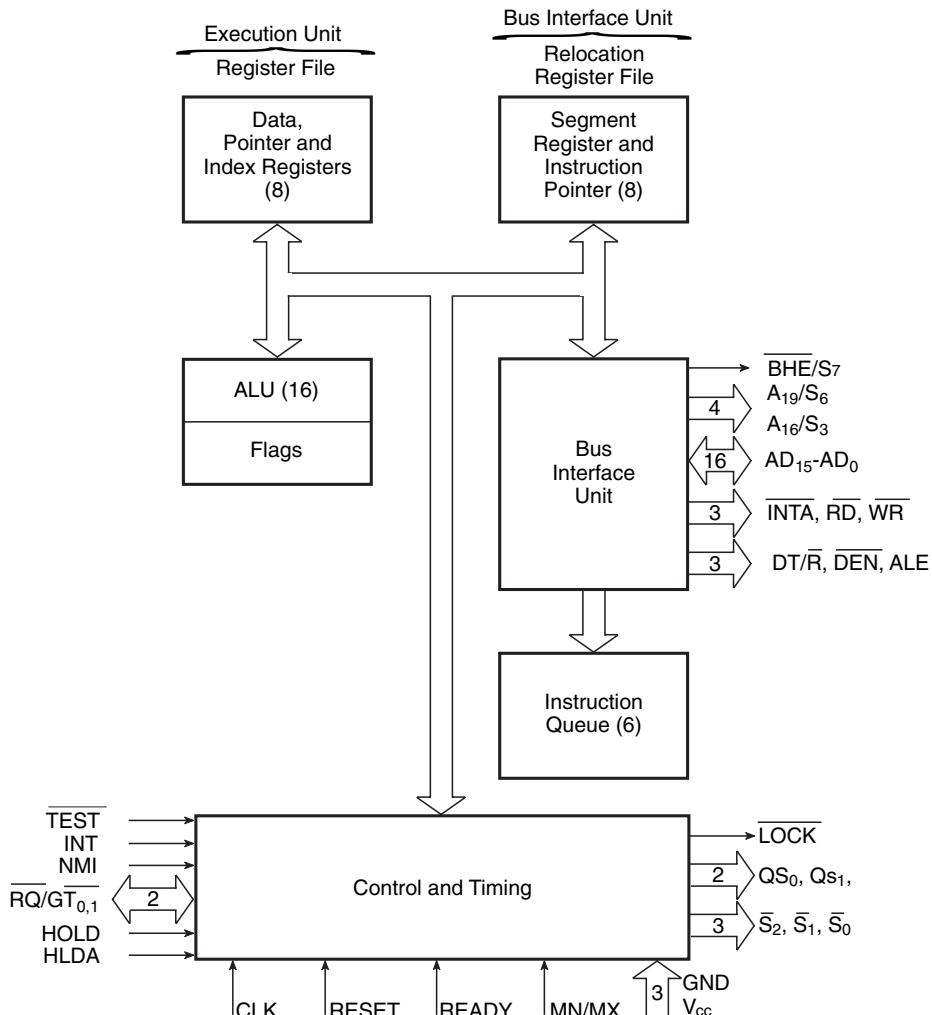


Figure 13.15 Block diagram of 8086.

13.10.2 80186 Microprocessor

The Intel 80186 is packaged in a 68-pin leadless package. It includes the Intel 8086 and several additional functional units on a single chip. The major on-chip circuits included are a clock generator, two independent DMA channels, a programmable interrupt controller, three programmable 16-bit timers and a chip select unit. It operates at a maximum frequency of 10 MHz.

13.10.3 80286 Microprocessor

The 80286 microprocessor is an advanced version of the 8086 microprocessor that was designed for multi-user and multitasking environments. It addresses 16 MB of physical memory and 1 GB of virtual

memory by using its memory management system. The 80286 is packaged in a 68-pin ceramic flat package and PGA, CLCC and PLCC packages. The 80286 microprocessor can work at a maximum frequency of 12.5 MHz.

13.10.4 MC68000 Microprocessor

68000 is the first member of Motorola's family of 16-bit and 32-bit processors. It is a successor to the 6809 and was followed by the 68010. The 68000 has 32-bit registers but only a 16-bit ALU and external data bus. It has 24-bit addressing and a linear address space. Addresses are computed as 32-bit, but the top eight bits are cut to fit the address bus into a 64-pin package (address and data share a bus in the 40-pin packages of the 8086). It is available in several clock frequencies. These include 6, 8, 10, 12.5, 16.67 and 25 MHz. The 68000 microprocessor is available in two packages, namely the 64-pin ceramic DIP and the 68-pin ceramic LLCC package. Figure 13.16 shows a simplified block diagram of the 68000 microprocessor.

13.10.4.1 68000 registers

The 68000 microprocessor has 16 32-bit registers and a 32-bit program counter. There are eight data registers for byte (eight-bit), word (16-bit) and long-word (32-bit) operations. There are seven address registers. These seven registers and the user stack pointer (USP) may be used as software stack pointers and base address registers. They are also used for word and long-word operations. Data, address and USP registers may also be used as index registers. In supervisor mode, the upper byte of the status register and the supervisor stack pointer (SSP) are also available to the programmer. The status register contains the interrupt mask as well as the condition codes [extend (X), negative (N), zero (Z), overflow (V) and carry (C)]. It also has status bits to indicate whether the processor is in trace (T) mode or in supervisor (S) mode.

13.10.4.2 Instruction Set

68000 has the following instruction types: data movement operations, integer arithmetic operations, logical operations, shift and rotate operations, bit manipulation operations, program control operations and system control operations.

13.10.4.3 Addressing Modes

The 68000 microprocessor supports the following addressing modes:

1. Register direct addressing (data register direct and address register direct).
2. Absolute data addressing (absolute short and absolute long).
3. Program counter relative addressing (relative with offset, relative with index and offset).
4. Register indirect addressing (register indirect, post-increment register indirect, predecrement register indirect, register indirect with offset, indexed register indirect with offset).

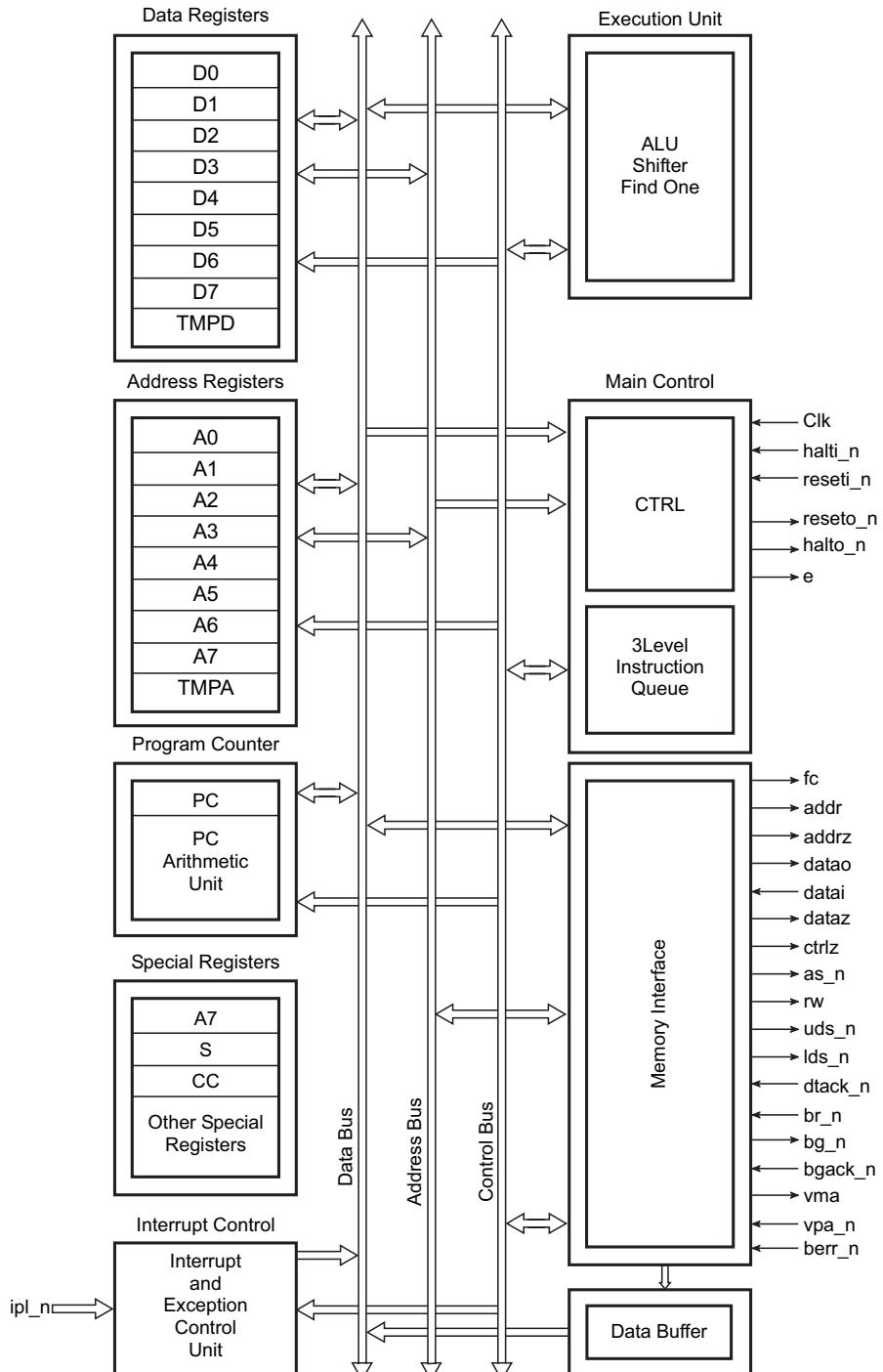


Figure 13.16 Block diagram of the 68000 microprocessor.

5. Immediate addressing (immediate and quick immediate).
6. Implied addressing (implied register).

13.11 32-Bit Microprocessors

This section describes the block diagram, internal architecture, salient features and instruction set of some of the most popular 32-bit microprocessors, namely 80386 of Intel and 68020 and 68030 of Motorola. It also gives an introduction to Intel's 80486 and Pentium series of processors.

13.11.1 80386 Microprocessor

80386 is a 32-bit microprocessor and is the logical extension of 80286. It provides multitasking support, memory management, pipeline architecture, address translation caches and a high-speed bus interface in a single chip. 80386 can be operated from a 12.5, 16, 20, 25 or 33 MHz clock. The 80386 has three processing modes, namely the protected mode, the real address mode and the virtual 8086 mode. The protected mode is the natural 32-bit environment of the 80386 processor. In this mode, all instructions and features are available. The real address mode is the mode of the processor immediately after RESET. In real mode, 80386 appears to programmers as a fast 8086 with some new instructions. Most applications of the 80386 will use the real mode for initialization only. The virtual 8086 mode (also called the V86 mode) is a dynamic mode in the sense that the processor can switch repeatedly and rapidly between V86 mode and protected mode.

Two versions of 80386, namely the 80386DX and the 80386SX, are commonly available. 80386SX is a reduced bus version of the 80386. The 80386DX addresses 4 GB of memory through its 32-bit data bus and 32-bit address bus. The 80386SX addresses 16 MB of memory with its 24-bit address bus. It was developed after the 80386DX for applications that did not require the full 32-bit bus version. A new version of 80386, named the 80386EX, incorporates the AT bus system, dynamic RAM controller, programmable chip selection guide, 26 address pins, 16 data pins and 24 I/O pins. Figure 13.17 shows the block diagram of the 80386 processor.

13.11.1.1 80386 DX registers

80386 DX contains a total of 32 registers. These registers may be grouped into general registers, segment registers, status and instruction registers, control registers, system address registers and debug and test registers.

13.11.1.2 Instruction Set

80386 DX executes the following instruction types:

1. Data movement instructions (general-purpose data movement instructions, stack manipulation instructions and type conversion instructions).
2. Binary arithmetic instructions (addition and subtraction instructions, comparison and size change instructions, multiplication instructions and division instructions).
3. Decimal arithmetic instructions (packed BCD adjustment and unpacked BCD adjustment instructions).

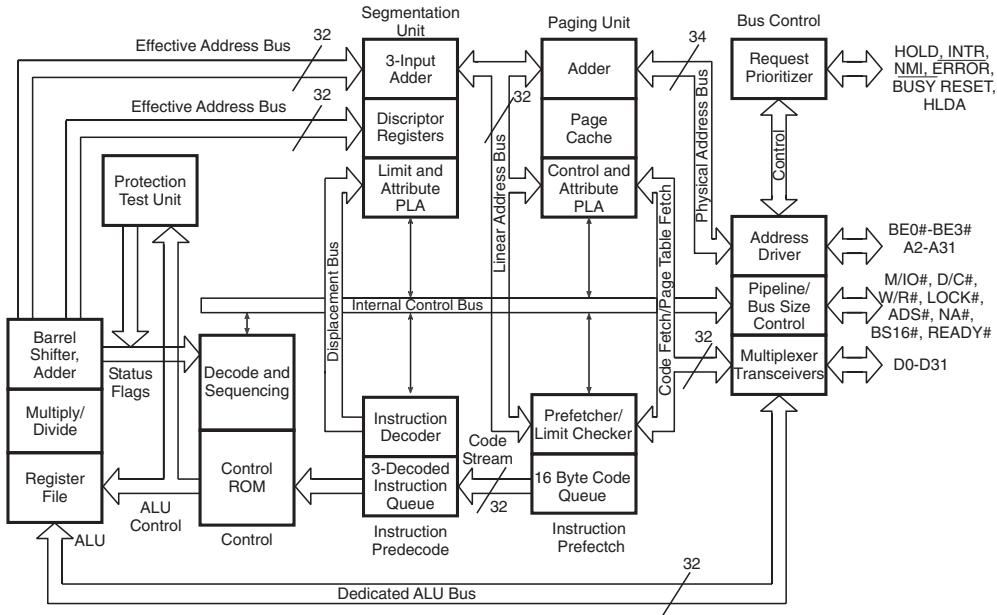


Figure 13.17 Block diagram of the 80386 microprocessor.

4. Logical instructions (Boolean operation, bit test and modify, bit scan, rotate and shift, byte set ON condition).
5. Control transfer instructions (unconditional transfer, conditional transfer, software-generated interrupts).
6. String and character translation instructions (repeat prefixes, indexing and direction flag control, string instructions).
7. Instructions for block-structured languages.
8. Flag control instructions (carry and direction flag control instructions and flag transfer instructions).
9. Coprocessor interface instructions.
10. Segment register instructions (segment register transfer, far-control transfer and data pointer instructions).
11. Miscellaneous instructions (address calculation, no-operation instruction and translate instruction).

13.11.1.3 Addressing Modes

80386 DX supports a total of 11 addressing modes as follows:

1. *Register and immediate modes.* These two modes provide for instructions that operate on register or immediate operands. These include register addressing mode and immediate addressing mode.
2. *32-bit memory addressing modes.* The remaining nine modes provide a mechanism for specifying the effective address of an operand. Here, the effective address is calculated by using combinations of displacement, base, index and scale address elements. The combination of these four elements

makes up the additional nine addressing modes. These include the direct mode, register indirect mode, based mode, index mode, scaled index mode, based index mode, based scaled index mode, based index mode with displacement and based scaled mode with displacement.

13.11.2 MC68020 Microprocessor

This is a 32-bit microprocessor introduced by Motorola. It can execute an object code written for MC68000, and therefore upward compatibility is maintained. It can operate at 12.5, 16.67, 20, 25 or 33 MHz. The MC68020 is supported by an array of peripheral devices and can directly be interfaced to coprocessor chips such as the MC68881/MC68882 floating-point and MC68851 memory management unit (MMU) coprocessor. It can directly address 4 GB of memory. The 68020 microprocessor also has an on-chip cache of size 128 words (16-bit). It is available in a PGA 114 ceramic-pin grid-array package and in CQFP 132 (Ceramic Quad Flat Package). Figure 13.18 shows the block diagram of Motorola's MC68020.

13.11.2.1 68020 Registers

68020 is a true 32-bit processor and it is object-code compatible with 68000. It has many more registers than 68000. Besides the eight data registers, seven address registers, one program counter and one status register (SR), there are three stack pointer (SP) registers instead of two. There is also one 16-bit vector-based register (VBR), two three-bit function code registers, one 32-bit cache address register (CAAR) and one 32-bit cache control register (CACR).

13.11.2.2 Instruction set

More than 20 new instructions have been added over MC68000. The new instructions include some minor improvements and extensions to the supervisor state, several instructions for software management of a multi-processing system, some support for high-level languages, bigger multiply (32×32) and divide (64/32) instructions and bit field manipulations.

13.11.2.3 Addressing modes

The 68020 microprocessor supports a total of 18 addressing modes with nine basic types:

1. Register direct (data register direct and address register direct).
2. Register indirect (address register indirect, address register indirect with post-increment, address register indirect with predecrement and address register indirect with displacement).
3. Register address indirect with index (register address indirect with index and register address indirect with index).
4. Memory indirect (memory indirect post-indexed and memory indirect pre-indexed).
5. Program counter indirect with displacement.
6. Program counter indirect with index (eight-bit displacement and base displacement).
7. Program counter memory indirect (post-indexed, pre-indexed).
8. Absolute data addressing (short and long).
9. Immediate addressing.

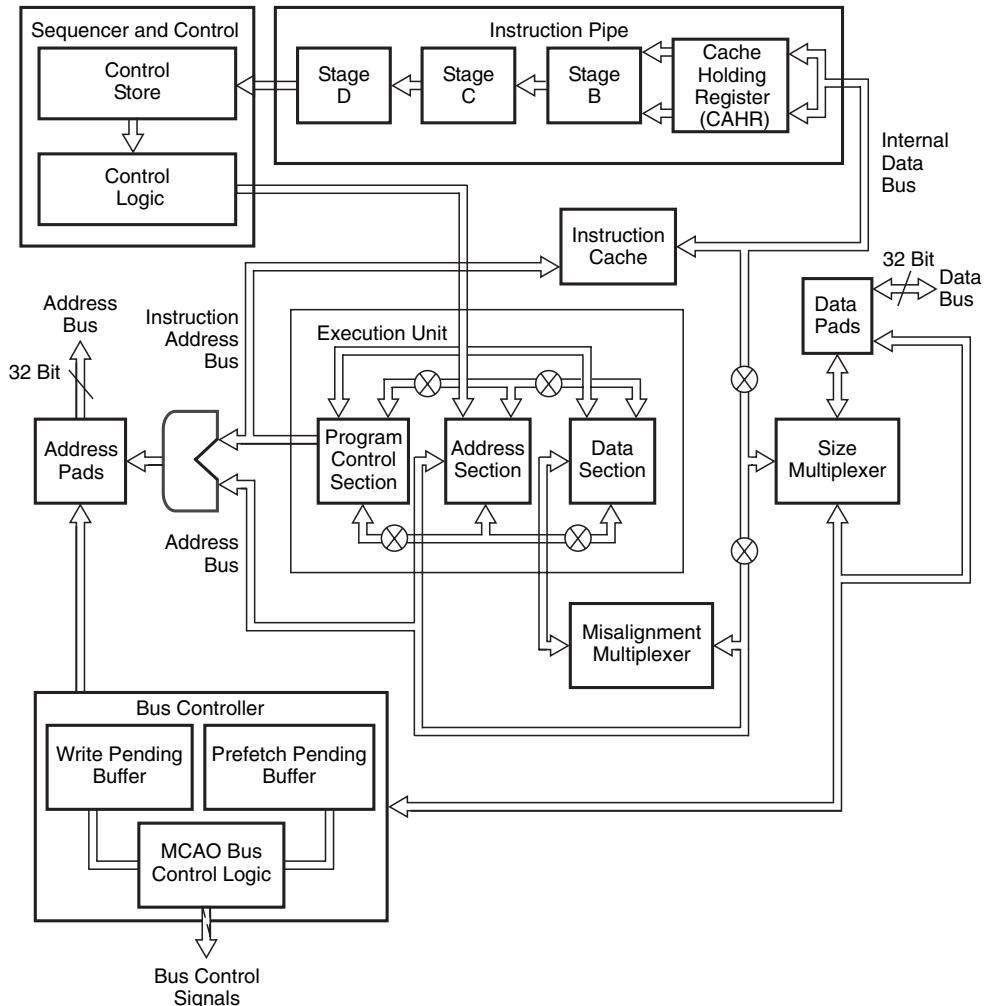


Figure 13.18 Block diagram of 68020.

13.11.3 MC68030 Microprocessor

The MC68030 is a second-generation full 32-bit virtual-memory microprocessor designed using HCMOS technology (Combining HMOS and CMOS on the same device) from Motorola. It is based on an MC68020 core with additional features. The MC68030 is a member of the M68000 family of devices that combines a central processing unit (CPU) core, a data cache, an instruction cache, an enhanced bus controller and a memory management unit (MMU) in a single VLSI device. It can be operated at 16.67, 20 and 33 MHz clocks. The MC68030 is upward-object-code compatible with the earlier members of the M68000 family and has the added features of an on-chip MMU, a data cache

and an improved bus interface. The MC68030 fully supports the nonmultiplexed bus structure of the MC68020, with 32 bits of address and 32 bits of data. The MC68030 bus has an enhanced controller that supports both asynchronous and synchronous bus cycles and burst data transfers.

13.11.4 80486 Microprocessor

The 80486 (i486 is the trade name) offers high performance for DOS, OS/2, Windows and UNIX System V applications. It is 100 % compatible with 80386 DX and SX microprocessors. One million transistors integrate cache memory, floating-point hardware and a memory management unit on-chip while retaining binary compatibility with previous members of the x86 architectural family. Frequently used instructions execute in one cycle, resulting in RISC performance levels. An eight-byte unified code and data cache combined with an 80/106 MB/s burst bus at 25/33 MHz ensure high system throughput even with inexpensive DRAMs.

The 80486 microprocessor is currently available in versions operating at 25, 33, 50, 66 and 100 MHz frequency. It is available as 80486DX and 80486SX. The only difference between these two devices is that 80486SX does not contain the numeric coprocessor. The 80487SX numeric coprocessor is available as a separate component for the 80486SX microprocessor.

Salient features of the 80486 processor include:

1. Full binary compatibility with 386 DX CPU, 386 SX CPU, 376 embedded processor and 80286, 8086 and 8088 processors.
2. Execution unit designed to execute frequently used instructions in one clock cycle.
3. 32-bit integer processor for performing arithmetic and logical operations.
4. Internal floating-point arithmetic unit for supporting the 32-, 64- and 80-bit formats specified in IEEE standard 754 (object-code compatible with 80387 DX and 387 SX math coprocessors).
5. Internal 8 kB cache memory, which provides fast access to recently used instructions and data.
6. Bus control signals for maintaining cache consistency in multiprocessor systems.
7. Segmentation, a form of memory management for creating independent, protected address space.
8. Paging, a form of memory management that provides access to data structures larger than the available memory space by keeping them partly in memory and partly on disk.
9. Restartable instructions that allow a program to be restarted following an exception (necessary for supporting demand-paged virtual memory).
10. Pipelined instruction execution overlaps the interpretation of different instructions.
11. Debugging registers for hardware support of instruction and data breakpoints.

The 80486 is object-code compatible with three other 386 processors, namely the 386 DX processor, the 386 SX processor (16-bit data bus) and the 376 embedded processor (16-bit data bus). 80486SX is also available in the same package with a few differences, as mentioned below:

1. Pin B15 is NMI on the 80486DX and pin A15 is NMI on 80486SX.
2. Pin A15 is \overline{IGNNE} on 80486DX. It is not present on 80486SX.
3. C14 is \overline{FERR} on 80486DX and pins B15 and C14 on 80486SX are not connected.

The architecture of 80486DX is almost the same as that of 80386 except that it contains a math coprocessor and an 8K byte level 1 cache memory. 80486SX does have the math coprocessor. Figure 13.19 shows the internal architecture of 80486DX. The major difference between 80386 and 80486 is that almost half of the instructions of 80486 execute in one clocking period instead of the two clocking periods for the 80386 microprocessor for the same instructions.

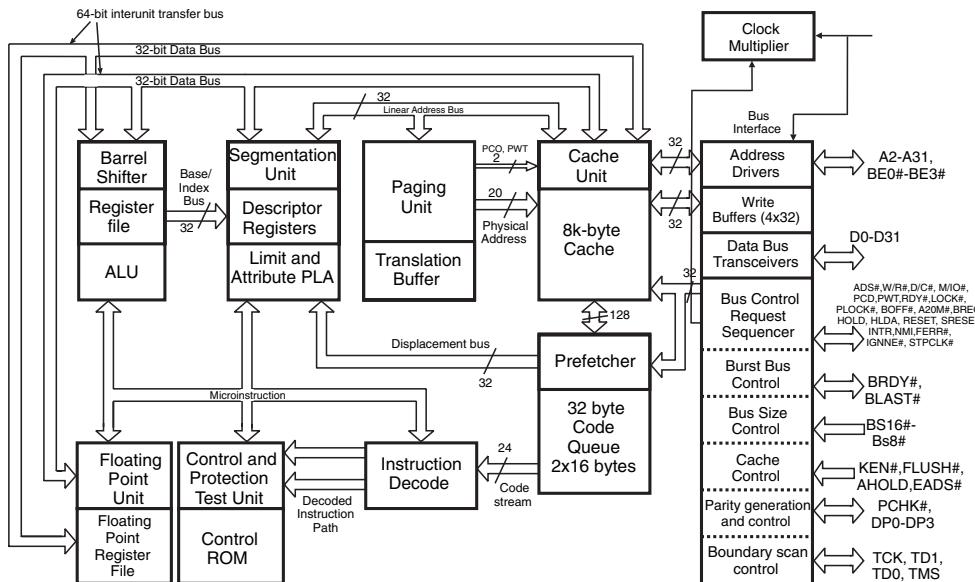


Figure 13.19 Internal architecture of 80486DX.

13.11.4.1 80486DX instruction set

The instruction set can be divided into 11 categories including data transfer operations, arithmetic operations, shift/rotate instructions, bit manipulation instructions, string manipulation instructions, control transfer instructions, high-level language support instructions, operating system support instructions, floating point processor control instructions and floating-point control instructions

13.11.4.2 80486DX registers

It contains all the registers of the 80386 microprocessor and 80386 math coprocessor. The register set is divided into the following categories: base architecture registers, general-purpose registers, instruction pointers, flag registers, segment registers, systems-level registers, control registers, system address registers, floating-point registers, data registers, tag word, status word, instruction and data pointers and control word and debug and test registers.

13.11.4.3 80486DX operating modes

The operating mode of the 80486 processor determines which instructions and architectural features are accessible. The 80486 has three modes for running programs. These are as follows:

1. The protected mode uses the native 32-bit instruction set of the processor. In this mode, all instructions and architectural features are available.

2. The real address mode (also called the ‘real mode’) emulates the programming environment of the 8086 processor, with a few extensions (such as the ability to break out of this mode). Reset initialization places the processor into real mode.
3. The virtual 8086 mode (also called the ‘V86 mode’) is another form of 8086 emulation mode. Unlike the real address mode, virtual 8086 mode is compatible with protection and memory management. The processor can enter virtual 8086 mode from protected mode to run a program written for the 8086 processor, then leave virtual 8086 mode and re-enter protected mode to run programs that use the 32-bit instruction set.

13.11.5 PowerPC RISC Microprocessors

The PowerPC family of microprocessors are high-performance superscalar RISC microprocessors developed jointly by MOTOROLA, IBM and Apple. They are used in personal computers, workstations and servers as well as for industrial and commercial embedded applications. Different versions of PowerPC microprocessors include PowerPC 601, 602, 603, ec603e, 604, 604e, 620, 740, 750, 745, 755, 750CX, 750CXE, 750FX and 750X microprocessors. PowerPC 601, 602, 603, ec603e and 604 are 32-bit microprocessors with a 32-bit address bus and a 64-bit data bus. They have 32 32-bit general-purpose registers and 32 64-bit floating-point registers. PowerPC 601 was introduced in the year 1993 for desktop PCs and low-end workstation applications. It uses 0.5 μm process technology and is available in 100 and 120 MHz clock frequency versions. PowerPC 602 was designed for graphical and multimedia applications. PowerPC 603 was introduced in the year 1993 and was used for applications where low power consumption was a critical requirement. It operates at 100 MHz. Its version 603e is an embedded microprocessor operating at 300 MHz. PowerPC 604 introduced in the year 1994 is available in different versions operating at 80, 100, 120, 133 and 250 MHz.

PowerPC 740/750 are 32-bit RISC microprocessors with special added features including a faster memory bus (66 MHz), larger L1 caches, enhanced integer and floating-point units and higher core frequency. PowerPC 750CX was developed by IBM using an 0.18 μm copper process. PowerPC 750FX was introduced in the year 2002 and had an operational frequency of up to 900 MHz. PowerPC 750GX introduced in the year 2004 is the latest and most powerful G3 processor from IBM.

13.12 Pentium Series of Microprocessors

The Pentium family of processors has its roots in the Intel 486 processor and has the same instruction set with a few additional instructions. Pentium processors have a 64-bit data bus and represent a major step forward in personal computer CPU design. The first Pentium processors (the P5 variety) were introduced in 1993. They were fabricated in 0.8 μm bipolar complementary metal oxide semiconductor (BiCMOS) technology. The P5 processor runs at a clock frequency of either 60 or 66 MHz and has 3.1 million transistors. The next version of the Pentium processor family was the P54 processor. The P54 processors were fabricated in 0.6 μm BiCMOS technology. The P54 was followed by P54C, introduced in 1994, which used a 0.35 μm CMOS process, as opposed to the bipolar CMOS process used for the earlier Pentiums. The P5 operated on 5 V supply and the P54 and P54C series operated on a 3.5 V supply voltage. All these processors had a problem in the floating-point unit. They were followed by the P55C processor, also referred to as the Pentium MMX. It was based on the P5 core and fabricated using the 0.35 μm process. The performance of the P55C was improved over the previous versions by doubling the level 1 CPU cache from 16 to 32 kB. Intel has retained the Pentium trademark for naming later generations of processor architectures, which are internally quite different

from the Pentium itself. These include Pentium Pro, Pentium II, Pentium M, Pentium D and Pentium Extreme Edition.

The Pentium processor has two primary operating modes and a system management mode. The operating mode determines which instructions and architectural features are accessible. These modes are as follows:

1. *Protected mode*. This is the native state of the microprocessor. In this mode, all instructions and architectural features are available, providing the highest performance and capability.
2. *Real address mode*. This mode provides the programming environment of the Intel 8086 processor, with a few extensions. Reset initialization places the processor in real mode where, with a single instruction, it can switch to protected mode.
3. *System management mode*. It provides an operating system and application independent transparent mechanism to implement system power management and OEM differentiation features. SMM is entered through activation of an external interrupt pin (SMI#), which switches the CPU to a separate address space while saving the entire content of the CPU.

13.12.1 Salient Features

The Pentium series (P5, P54 and P54C) of microprocessors has the following advanced features:

1. *Superscalar execution*. The Intel 486 processor can execute only one instruction at a time. With superscalar execution, the Pentium processor can sometimes execute two instructions simultaneously.
2. *Pipeline architecture*. Like the Intel 486 processor, the Pentium processor executes instructions in five stages. This staging, or pipelining, allows the processor to overlap multiple instructions so that it takes less time to execute two instructions in a row. Because of its superscalar architecture, the Pentium processor has two independent processor pipelines.
3. *Branch target buffer*. The Pentium processor fetches the branch target instruction before it executes the branch instruction.
4. *Dual 8 kB on-chip caches*. The Pentium processor has two separate 8 kB caches on chip, one for instructions and the other for data. This allows the Pentium processor to fetch data and instructions from the cache simultaneously.
5. *Write-back cache*. When data are modified, only the data in the cache are changed. Memory data are changed only when the Pentium processor replaces the modified data in the cache with a different set of data.
6. *64-bit bus*. With its 64-bit wide external data bus (in contrast to the Intel 486 processor's 32-bit wide external bus), the Pentium processor can handle up to twice the data load of the Intel 486 processor at the same clock frequency.
7. *Instruction optimization*. The Pentium processor has been optimized to run critical instructions in fewer clock cycles than the Intel 486 processor.
8. *Floating-point optimization*. The Pentium processor executes individual instructions faster through execution pipelining, which allows multiple floating-point instructions to be executed at the same time.
9. *Pentium extensions*. The Pentium processor has fewer instruction set extensions than the Intel 486 processors. The Pentium processor also has a set of extensions for multiprocessor (MP) operation. This makes a computer with multiple Pentium processors possible.

13.12.2 Pentium Pro Microprocessor

Pentium Pro is a sixth-generation x86 architecture microprocessor (P6 core) from Intel. It was originally intended to replace the earlier Pentium series of microprocessors in a full range of applications, but was later reduced to a narrow role as a server and high-end desktop chip. The Pentium Pro was capable of both dual- and quad-processor configurations. The Pentium Pro achieves a performance approximately 50 % higher than that of a Pentium of the same clock speed. In addition to its new way of processing instructions, the Pentium Pro incorporates several other technical features including superpipelining, an integrated level 2 cache, 32-bit optimization, a wider address bus, greater multiprocessing, out-of-order completion of instructions, a superior branch prediction unit and speculative execution.

13.12.3 Pentium II Series

Pentium II is an x86 architecture microprocessor introduced by Intel in the year 1997. It was based on a modified version of the P6 core improved 16-bit performance and the addition of the MMX SIMD instruction set. The Pentium II series of processors are available in speeds of 233, 266, 300, 330, 350, 400 and 450 MHz. Some of the product highlights include the use of Intel's 0.25 µm manufacturing process for increased processor core frequencies and reduced power consumption, the use of MMX bus (DIB) architecture to increase bandwidth and performance over single-bus processors, a 32 kB nonblocking level 1 cache, a 512 kB unified, nonblocking level 2 cache and data integrity and reliability features.

13.12.4 Pentium III and Pentium IV Microprocessors

Pentium III is an x86 architecture microprocessor from Intel, introduced in the year 1999. Initial versions were very similar to the earlier Pentium II. The most notable difference is the addition of SSE instructions and the introduction of a serial number which was embedded in the chip during the manufacturing process. Pentium III processors are available in speeds of 650, 667, 700, 733, 750, 800, 850 and 866 MHz and 1 GHz. The Pentium III processor integrates PC dynamic execution microarchitecture, DIB architecture, a multitransaction system bus and Intel's MMX media enhancement technology. In addition to these features, it offers Internet streaming and single-instruction multiple-data (SIMD) extension. It has 70 new instructions to enable advanced imaging, 3D, streaming audio and video and speech recognition. Pentium III processors were superseded by Pentium IV.

Pentium IV is a seventh-generation x86 architecture microprocessor from Intel. It uses a new CPU design, called the netburst architecture. The netburst microarchitecture featured a very deep instruction pipeline, with the intention of scaling to very high frequencies. It also introduced the SSE2 instruction set for faster SIMD integer and 64-bit floating-point computation. It operates at frequencies of over 1 GHz.

13.12.5 Pentium M, D and Extreme Edition Processors

Pentium M is an x86 architecture microprocessor from Intel, introduced in the year 2003. It forms part of the Intel Centrino platform. The processor was originally designed for use in laptop personal computers (thus the 'M' for mobile).

Pentium D is a series of microprocessors from Intel introduced in the year 2005. Pentium D was the first multicore CPU along with the Pentium Extreme Edition. It is the final processor to carry the

Pentium brand name. The Pentium Extreme Edition series of microprocessors was introduced by Intel in the year 2005. It is based on the dual-core Pentium D processor.

13.12.6 Celeron and Xeon Processors

Celeron processors were introduced by Intel as a low-cost CPU alternative for the Pentium processors. They were basically Pentium II processors without any L2 cache at all. However, this reduced the performance of Celeron processors as compared with AMD and Cyrix chips. Hence, subsequent Celeron versions (300A and up) were provided with 128 kB of L2 cache. It was about one-fourth the size of the Pentium cache but operated at the full speed of the respective CPU, rather than at half-speed as in the Pentium processors. Later Celeron versions were based on the Pentium III, Pentium IV and Pentium M processors. These processors are suitable for most applications, but their performance is somewhat limited when it comes to running intense applications. Xeon are high-end processors having a full-speed L2 cache of the same size as the Pentium cache. These processors are used for high-performance servers and workstations.

13.13 Microprocessors for Embedded Applications

Embedded microprocessors are microprocessors designed for embedded applications and not for use in personal computers. They are mostly used for embedded data control applications such as data processing, data formatting, I/O control, DMA data transfer, etc. In other words, they are designed for specific applications rather than for general-purpose applications. Intel has developed a number of embedded microprocessors, namely Intel 80960, Intel 80376 and embedded versions of 80486, 80386 and 80186 microprocessors. Other embedded microprocessors include Motorola's Coldfire, Sun's Sparc, Hitachi's SuperH, Advanced RISC Machines' ARM, and MIPS Computer Systems Inc.'s MIPS processors.

The Intel 80960 and 80376 microprocessors are 32-bit microprocessors designed for sophisticated industrial control applications. Embedded versions of 80486 include 486GX, 486SX, 496DX2 and 486DX4 microprocessors. The embedded versions of 80386 include 386CXSA, 386CXS, 386EX and 386SXSA microprocessors. Scalable processor architecture (SPARC) microprocessors are 32-bit and 64-bit CISC processors from Sun Microsystems. ARM microprocessors are 32-bit RISC microprocessors and are mostly used in the mobile electronics market, where low power is the most critical design requirement. MIPS (Microprocessor without Interlocked Pipeline Stages) is a RISC microprocessor from MIPS Computer Systems Inc. They are available in 32-bit and 64-bit versions.

13.14 Peripheral Devices

Microprocessors and peripheral devices provide a complete solution in increasingly complex application environments. A peripheral device typically belongs to the category of MSI logic devices. This section gives an introduction to the popular peripheral devices that are used along with the microprocessor in a microcomputer system. The different peripheral devices used in a microcomputer system include a programmable counter/timer, a programmable peripheral interface (PPI), EPROM, RAM, a programmable interrupt controller (PIC), a direct memory access (DMA) controller, a programmable communication interface – a universal synchronous/asynchronous receiver/transmitter (USART), a math coprocessor, a programmable keyboard/display interface, a CRT controller, a floppy disk controller and clock generators and transceivers.

13.14.1 Programmable Timer/Counter

The programmable timer/counter is used for the generation of an accurate time delay for event counting, rate generation, complex waveform generation applications and so on. Examples of programmable timer/counter devices include Intel's 8254 and 8253 family of devices. Intel 8254 contains three 16-bit counters that can be programmed to operate in several different modes. Some of the functions common to microcomputers and implementable with 8254 are a real-time clock, an event counter, a digital one-shot, programmable rate generator, a square-wave generator, a binary rate multiplier, a complex waveform generator and a complex motor controller. It is available in 24-pin CERDIP and plastic DIP packages.

13.14.2 Programmable Peripheral Interface

Programmable peripheral interface (PPI) devices are used to interface the peripheral devices with the microprocessors. 8255 PPI is a widely used programmable parallel I/O device. It is available in PDIP, CerDIP, PLCC and MQPF packages. 8255 can be programmed to transfer data under various conditions, from simple I/O to interrupt I/O. It can function in bit reset (BSR) mode or I/O mode. In I/O mode it has three ports, namely port A, port B and port C. The I/O mode is further divided into three different modes, namely mode 0, mode 1 and mode 2. In mode 0, all ports function as simple I/O ports. Mode 1 is a handshake mode whereby port A and/or B use bits from port C as handshake signals. In mode 2, port A can be set up for bidirectional data transfer using handshake signals from port C, and port B can be set up either in mode 0 or in mode 1. In BSR mode, individual bits in port C can be set or reset.

13.14.3 Programmable Interrupt Controller

A programmable interrupt controller (PIC) is a device that allows priority levels to be assigned to its interrupt outputs. It functions as an overall manager in an interrupt-driven system environment. When the device has multiple interrupt outputs, it will assert them in the order of their relative priority. Common modes of a PIC include hard priorities, rotating priorities and cascading priorities. Intel 8259 is a family of programmable interrupt controllers (PICs) designed and developed for use with the Intel 8085 and Intel 8086 microprocessors. The family originally consisted of the 8259, 8259A, and 8259B PICs, although a number of manufacturers make a wide range of compatible chips today.

It handles up to eight vectored priority interrupts for the CPU. It is designed to minimize the software and real-time overhead in handling multi-level priority interrupts. It accepts requests from peripheral equipment, determines which of the incoming requests is of the highest priority, ascertains whether an incoming request has a higher priority value than the level currently being serviced and issues an interrupt to the CPU on the basis of this determination.

13.14.4 DMA Controller

In a direct memory access (DMA) data transfer scheme, data are transferred directly from an I/O device to memory, or vice versa, without going through the CPU. The DMA controller is used to control the process of data transfer. Its primary function is to generate, upon a peripheral request, a sequential memory address that will allow the peripheral to read or write data directly to or from memory. One of the popular known programmable DMA controllers is Intel's 8257. It is a four-channel direct memory access (DMA) controller. It is specifically designed to simplify the transfer of data at high speeds

for microcomputer systems. It has a priority logic that resolves the peripheral requests and issues a composite hold request to the CPU. It maintains the DMA cycle count for each channel and outputs a control signal to notify the peripheral that the programmed number of DMA cycles is completed.

13.14.5 Programmable Communication Interface

Programmable communication interfaces (PCIs) are interface devices that are used for data communication applications with microprocessors. They basically convert the data from the microprocessor into a format acceptable for communication and also convert the incoming data into a format understood by the microprocessor.

8251 is a PCI device designed for Intel's 8085, 8086 and 8088 microprocessors and is used in serial communication applications. It is a 28-pin chip available in DIP and PLCC packages. It is basically a universal synchronous/asynchronous receiver/transmitter (USART) that accepts data characters from the CPU in parallel format and then converts them into a continuous serial data stream for transmission. Simultaneously, it can receive a serial data stream and convert it into parallel data characters for the CPU. The USART will signal the CPU whenever it can accept a new character for transmission or whenever it has received a character for the CPU.

13.14.6 Math Coprocessor

Math coprocessors are special-purpose processing units that assist the microprocessor in performing certain mathematical operations. The arithmetic operations performed by the coprocessor are floating-point operations, trigonometric, logarithmic and exponential functions and so on. Examples include Intel's 8087, 80287, etc. The 8087 numeric coprocessor provides the instructions and data types needed for high-performance numeric application, providing up to 100 times the performance of a CPU alone. Another widely used math coprocessor is 80287. The 80287 numeric processor extension (NPX) provides arithmetic instructions for a variety of numeric data types in 80286 systems. It also executes numerous built-in transcendental functions (e.g. tangent and log functions).

13.14.7 Programmable Keyboard/Display Interface

Programmable keyboard/display interfaces are devices used for interfacing the keyboard and the display to the microprocessor. The keyboard section of the device debounces the keyboard entries and provides data to the microprocessor in the desired format. The display section converts the data output of the microprocessor into the form desired by the display device in use.

8279 is a general-purpose programmable keyboard and display I/O interface device designed for use with Intel microprocessors. The keyboard portion can provide a scanned interface to a 64-contact key matrix. Keyboard entries are debounced and strobed in eight-character FIFO. If more than eight characters are entered, overrun status is set. Key entries set the interrupt output line to the CPU. The display portion provides a scanned display interface for LED, incandescent and other popular display technologies. Both numeric and alphanumeric segment displays may be used. The 8279 has a 16×8 display RAM.

13.14.8 Programmable CRT Controller

The programmable CRT controller is a device to interface CRT raster scan displays with the microprocessor system. Its primary function is to refresh the display by buffering the information

from the main memory and keeping track of the display position of the screen. One of the commonly used programmable CRT controllers is Intel's 8275H. It allows a simple interface to almost any raster scan CRT display with minimum external hardware and software overheads. The number of display characters per row and the number of character rows per frame are software programmable.

13.14.9 Floppy Disk Controller

The floppy disk controller is used for disk drive selection, head loading, the issue of read/write commands, data separation and serial-to-parallel and parallel-to-serial conversion of data. Examples of floppy disk controllers include Intel's 82078, 82077 and 8272.

13.14.10 Clock Generator

The clock generator is a circuit that produces a timing signal for synchronization of the circuit's operation. Examples of clock generators used in microprocessor systems include 8284 and 82284. 8284 generates the system clock for the 8086 and 8088 processors. It requires a crystal or a TTL signal source for producing clock waveforms. It provides local READY and MULTIBUS READY synchronization.

82284 is a clock generator/driver that provides clock signals for the 80286 processor and support components. It also contains logic to supply READY to the CPU from either asynchronous or synchronous sources and synchronous RESET from an asynchronous input with hysteresis. The 82284 is packaged in 18-pin DIP and contains a crystal-controlled oscillator, an MOS clock generator, a peripheral clock generator, multibus ready synchronization logic and system reset generation logic.

13.14.11 Octal Bus Transceiver

Bus transceivers are devices with a high-output drive capability for interconnection with data buses. In a microprocessor-based system they provide an interface between the microprocessor bus and the system data bus. 8286 is an eight-bit bipolar transceiver with a three-state output that is used in a wide variety of buffering applications in microcomputer systems. It comes in a 20-pin DIP package.

Review Questions

1. Briefly describe the difference between a microprocessor and a microcomputer. What are the three main constituents of a microprocessor and what is the basic function performed by each one of them.
2. What are the different types of register found in a typical microprocessor? Briefly describe the function of each one of them.
3. Distinguish between the following
 - (a) address bus and data bus;
 - (b) direct addressing mode and indirect addressing mode;
 - (c) programmable timer and clock generator;
 - (d) programmable interrupt controller and DMA controller;
 - (e) RISC and CISC microprocessors.

4. Briefly describe the parameters that you would consider while choosing the right microprocessor for your application, emphasizing the significance of each parameter.
5. With the help of a labelled diagram, briefly describe the operational role of the three types of bus in a microcomputer system.
6. Briefly describe the primary functions of the following peripheral devices. Also, give at least one device type number for each of them:
 - (a) programmable timer;
 - (b) clock generator;
 - (c) programmable peripheral interface;
 - (d) DMA controller;
 - (e) programmable interrupt controller.
7. Briefly describe salient features of the Pentium series of microprocessors.
8. Compare and contrast:
 - (a) eight-bit microprocessors,
 - (b) 16-bit microprocessors,
 - (c) 32-bit microprocessors and
 - (d) 64-bit microprocessors
from Intel and Motorola.

Further Reading

1. Brey, B. B. (2000) *The Intel Microprocessors 8086/8088, 80186/80188, 80286, 80326, 80486, Pentium, and Pentium Pro Processor Architecture, Programming, and Interfacing*, Prentice-Hall, NJ, USA.
2. Floyd, T. L. (2005) *Digital Fundamentals*, Prentice-Hall Inc., USA.
3. Crisp, J. (2004) *Introduction to Microprocessors and Microcontrollers*, Newnes, Oxford, UK.
4. Tocci, R. J. and Ambrosio, F. J. (2002) *Microprocessors and Microcomputers: Hardware and Software*, Prentice-Hall, NJ, USA.
5. Rafiquzzaman, M. (1990) *Microprocessor and Microcomputer-based System Design*, CRC Press, FL, USA.