ISBN/ISSN: 9780138146467

**SECTION 13–2 CHECKUP**

1. What are the four basic elements in a microprocessor?
2. What are the three types of buses in a microprocessor?
3. What function does a microprocessor perform in a computer?
4. What are the three basic operations that a microprocessor performs?
5. What is pipelining?
6. What is multitasking?
7. What is multithreading?

## 13–3 BASIC MICROPROCESSOR OPERATION

The original Intel microprocessor family has undergone a tremendous change over the years from the 8086/8088 to the Pentium family and to multi-core processors, both in speed and in complexity. However, the concept of the basic register set and other features of the 8086/8088 have been retained (and expanded) throughout the evolutionary process so that all of the newer processors respond to the same instructions (as well as a number of new instructions) as the original devices. This section starts with a limited introduction of basic concepts of microprocessor architecture, operation, and programming using the 8086/8088 for illustration. The section ends with a brief overview of the principal changes to the register structure that forms the software model of the newer processors, such as the Pentium.

After completing this section, you should be able to

- Discuss the basic microprocessor operation
- Describe the bus interface unit
- State the purpose of the segment registers
- State the purpose of the instruction pointer
- Describe the execution unit
- Describe the general set of registers
- State the purpose of the flag register
- Discuss the software model of the Pentium processor

### Basic Operation

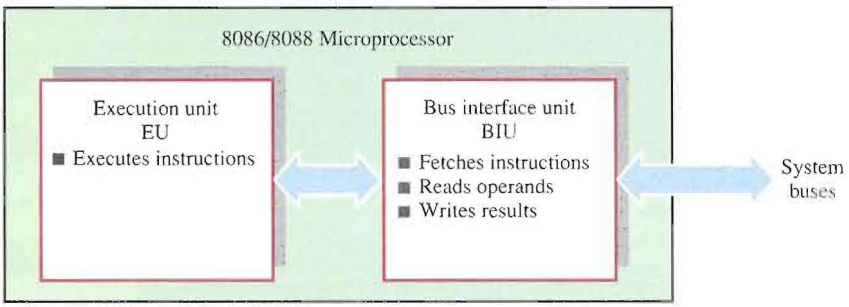
A microprocessor executes a program by repeatedly cycling through the following three steps:

1. Fetch an instruction from memory and place it in the CPU.
2. Decode the instruction; if other information is required by the instruction, fetch the other information. In the decode step, the program counter is updated to point to the next instruction.
3. Execute the instruction (do what the instruction says). Results are returned to registers and memory during this step.

The architecture of the 8086/8088 microprocessor provided for two separate internal units: the execution unit (**EU**), which executes instructions, and the bus interface unit (**BIU**), which interfaces with the system buses and fetches instructions, reads operands, and writes results. These units are shown in Figure 13–9.

▶ FIGURE 13–9

The 8086/8088 has two separate internal units, the EU and the BIU.




The BIU performs all the bus operations for the EU, such as data transfers from memory or I/O. While the EU is executing instructions, the BIU "looks ahead" and fetches more instructions from memory. This action is called **prefetching.** The concept of prefetching is to allow the processor to execute instructions at the same time as the next instruction was being fetched, eliminating idle time. The prefetched instructions are stored in an internal high-speed memory called the instruction **queue** (pronounced "Q"). The queue allows the BIU to keep the EU supplied with instructions. The EU does not have to wait for the next instruction to be fetched from memory; but instead it retrieves the next instruction directly from the queue in much less time. In the Pentium, this process is taken a step further. Two complete execution units enable two instructions to execute at the same time provided they are independent. Certain compilers are designed to take advantage of the two execution units by a process known as **instruction pairing** to remove dependencies.

## Basic 8086/8088 Architecture

Figure 13–10 is a block diagram of the architecture (internal organization) of an 8088 microprocessor. Externally, the 8088 had 20 address bits that could address 1 MB (1,048,576 bytes) of memory and used an 8-bit data bus. Internally, the 8088 had a 16-bit data bus and a 4-byte queue. The 8086 was identical except that it had an external 16-bit data bus and a 6-byte instruction queue.

## The Bus Interface Unit (BIU)

The major parts of the BIU are the 4-byte instruction queue, the segment registers (CS, DS, SS, and ES), the instruction pointer (IP), and the address summing block (Σ). The 16-bit internal data buses and the Q bus interconnect the BIU and the EU.

***Instruction Queue*** The instruction queue increases the average speed with which a program is executed (called the **throughput**) by storing up to four bytes (six in the 8086). As described earlier, this technique allowed the 8088 essentially to do two things, fetch and execute, at one time. This feature has been expanded in subsequent processors to include much larger and faster queues.

***Segment Registers*** The 8086/8088 processors had four segment registers (CS, DS, SS, and ES) that were all 16-bit registers used in the process of forming a 20-bit address. A **segment** is a 64 kB block of memory and can begin at any point in the 1 MB (1,048,576 bytes) of memory space, provided it begins on a 16-byte boundary (evenly divisible by 16).

In designing the 8086/8088 and subsequent processors, Intel chose a unique method of generating the required 20-bit physical address using two 16-bit registers. One of the registers