Microchip extended the 16C5x's architecture and features with the 16C6x and 16C7x families. The 16C5x's advantages of low power consumption, wide operating voltage range, and small size are retained. Improvements include more memory (up to 368 bytes of RAM and 8 kB of ROM), a more versatile microprocessor core with interrupt capability, an eight-level stack, and a wider selection of on-board peripherals including additional timers, serial ports, and *analog-to-digital* (A/D) converters. (An A/D converter is a circuit that converts an analog voltage range into a range of binary values. An 8-bit A/D converter covering the range of 0 to 5 V would express voltages in that range as a byte value with a resolution of 5 V ÷ (2^8 – 1) increments = 19.6 mV per increment.) Between four and eight A/D converters are available in 'C7x devices.

Some PIC microcontrollers contain two serial ports on the same chip: an asynchronous port suitable for RS-232 applications and a synchronous port capable of SPI or I^2C operation in conjunction with other similarly equipped ICs in a system. At the other end of the spectrum, very small PIC devices are available in eight-pin packages—small enough to fit almost anywhere.

6.5 INTEL 8086 16-BIT MICROPROCESSOR FAMILY

Intel moved up to a 16-bit microprocessor, the 8086, in 1978—just two years after introducing the 8085 as an enhancement to the 8080. The "x86" family is famous for being chosen by IBM for their original PC. As PCs developed during the past 20 years, the x86 family grew with the industry—first to 32 bits (80386, Pentium) and more recently to 64 bits (Itanium). While the 8086 was a new architecture, it retained certain architectural characteristics of the 8080/8085 such that assembly language programs written for its predecessors could be converted over to the 8086 with little or no modification. This is one of the key reasons for its initial success.

The 8086 contains various 16-bit registers as shown in Fig. 6.9, some of which can be manipulated one byte at a time. AX, BX, CX, and DX are general-purpose registers that have alternate functions and that can be treated as single 16-bit registers or as individual 8-bit registers. The accumulator, AX, and the flags register serve their familiar functions. BX can serve as a general pointer. CX is a loop iteration count register that is used inherently by certain instructions. DX is used as a companion register to AX when performing certain arithmetic operations such as integer division or handling long integers (32 bits).

The remaining registers are pointers of various types that index into the 8086's somewhat awkward segmented memory structure. Despite being a 16-bit microprocessor with no register exceeding 16 bits in size, Intel recognized the need for more than 64 kB of addressable memory in more advanced computers. One megabyte of memory space was decided upon as a sufficiently large address space in the late 1970s, but the question remained of how to access that memory with 16-bit pointers. Intel's solution was to have programmers arbitrarily break the 1 MB address space into multiple 64-kB special-purpose segments—one for instructions (code segment), two for data (primary data and "extra" data), and one for the stack. Memory operations must reference one of these defined segments, requiring only a 16-bit pointer to address any location within a given segment. Segments can be located anywhere in memory, as shown in Fig. 6.10, and can be moved at will to provide flexibility for different applications. Additionally, there is no restriction on overlapping of segments.

Each segment register represents the upper 16 bits of a 20-bit pointer (2^{20} = 1 MB) where the lower 4 bits are fixed at 0. Therefore, a segment register directly points to an arbitrary location in 1 MB of memory on a 16-byte boundary. A pointer register is then added to the 20-bit segment address to yield a final 20-bit address, the effective address, with which to fetch or store data. Algebraically, this relationship is expressed as: effective address = (segment pointer × 16) + offset pointer.

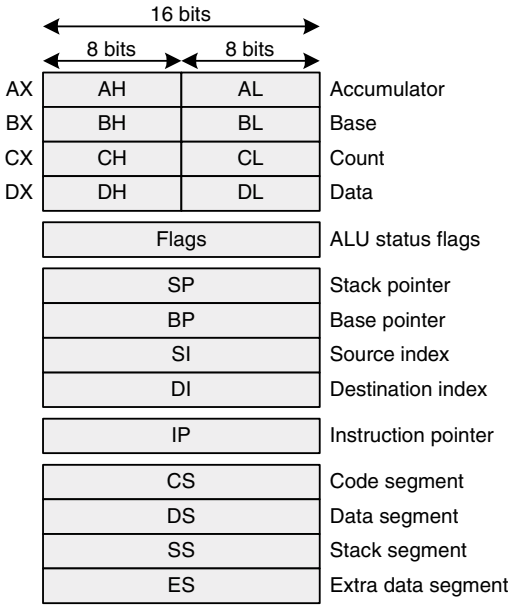

FIGURE 6.9 8086 register set.

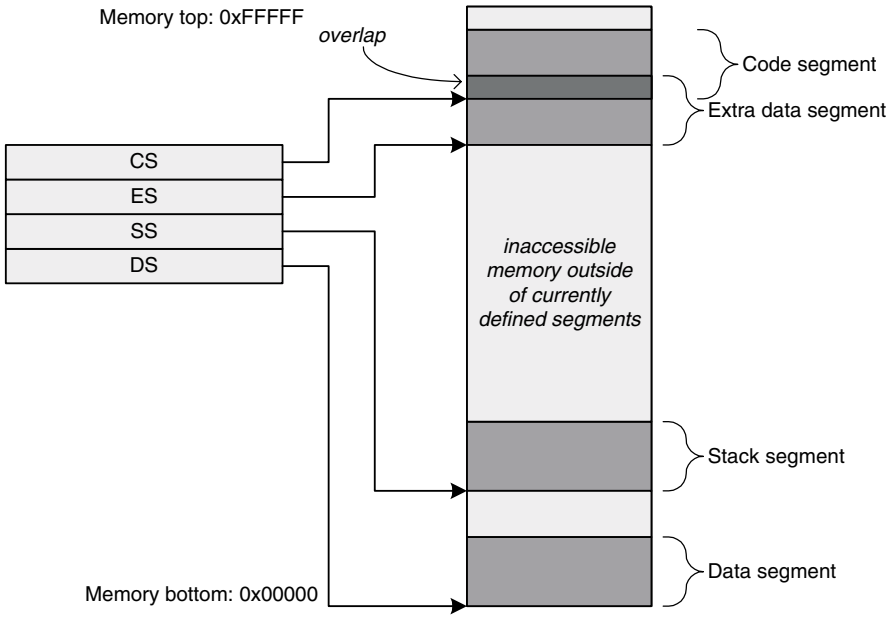

FIGURE 6.10 8086 segments.