ist today. The 8080 was housed in a 40-pin DIP, featured a 16-bit address bus and an 8-bit data bus, and ran at 2 MHz. It also implemented a conventional stack pointer that enabled deep stacks in external memory (Intel's earlier microprocessors had internal stacks with very limited depth). The 8080 became extremely popular as a result of its performance and rich, modern instruction set. This popularity was evidenced two years later, in 1976, with Intel's enhanced 8085 and competitor Zilog's famous Z80. Designed by former Intel engineers, the Z80 was based heavily on the 8080 to the point of having a partially compatible instruction set.

Both the 8085 and Z80 were extremely popular in a variety of computing platforms from hobbyists to mainstream commercial products to video arcade games. The 8085 architecture influenced the famous 16-bit 8086 family whose strong influence continues to this day in desktop PCs. The Z80 eventually lost the mainstream microprocessor war and migrated to microcontrollers that are still available for new designs from Zilog.

As microprocessors progress, technologies that used to be leading edge first become mainstream and then appear quite pedestrian. Along the way, some microprocessor families branch into multiple product lines to suit a variety of target applications. The high-end computing market gets most of the publicity and accounts for the major technology improvements over time. Lower-end microprocessors are either made obsolete after some time or find their way into the *embedded* market. Embedded microprocessors and systems are those that may not appear to the end user as a computer, or they may not be visible at all. Instead, embedded microprocessors typically serve a control function in a machine or another piece of equipment. This is in contrast to the traditional computer with a keyboard and monitor that is clearly identified as a general-purpose computer.

Integrated microprocessor products are called *microcontrollers*, a term that has already been introduced. A microcontroller is a microprocessor integrated with a varying mix of memory and peripherals on a single chip. Microcontrollers are almost always found in embedded systems. As with many industry terms, *microcontrollers* can mean very different things to different people. In general, a microcontroller contains a relatively inexpensive microprocessor core with a complement of onboard peripherals that enable a very compact, yet complete, computing system—either on a single chip or relatively few chips. There is a vast array of single-chip microcontrollers on the market that integrate quantities of both RAM and ROM on the same chip along with basic peripherals including serial communications controllers, timers, and general I/O signal pins for controlling LEDs, relays, and so on. Some of the smallest microcontrollers can cost less than a dollar and are available in packages with as few as eight pins. Such devices can literally squeeze a complete computer into the area of a fingernail. More complex microcontrollers can cost tens of dollars and provide external microprocessor buses for memory and I/O expansion. At the very high end, there are microcontrollers available for well over $100 that include 32-bit microprocessors running at hundreds of megahertz, with integrated Ethernet controllers and DMA. Manufacturers typically refer to these high-end microcontrollers with unique, proprietary names to differentiate them from the aforementioned class of inexpensive devices.

6.2 MOTOROLA 6800 EIGHT-BIT MICROPROCESSOR FAMILY

As the microprocessor market began to take off, Motorola jumped into the fray and introduced its eight-bit 6800 in 1974, shortly after the 8080 first appeared. While no longer available as a discrete microprocessor, the 6800 is significant, because it remains in Motorola's successful 68HC05/68HC08 and 68HC11 microcontroller families and also serves as a vehicle with which to learn the basics of computer architecture. Like the 8080, the 6800 is housed in a 40-pin DIP and features a 16-bit address bus and an 8-bit data bus. All of the basic register types of a modern microprocessor are

implemented in the 6800, as shown in Fig. 6.1: a program counter (PC), stack pointer (SP), index register (X), two general-purpose accumulators (ACCA and ACCB), and status flags set by the ALU in the condition code register (CCR). ACCA is the primary accumulator, and some instructions operate only on this register and not ACCB. A half-carry flag is included to enable efficient binary coded decimal (BCD) operations. After adding two BCD values with normal binary arithmetic, the half-carry is used to convert illegal results back to BCD. The 6800 provides a special instruction, decimal adjust ACCA (DAA), for this specific purpose. A somewhat out-of-place interrupt mask bit is also implemented in the CCR, because this was an architecturally convenient place to locate it. Bits in the CCR are modified through either ALU operations or directly by transferring the value in ACCA to the CCR.

The 6800 supports three interrupts: one nonmaskable, one maskable, and one software interrupt. More recent variants of the 6800 support additional interrupt sources. A software interrupt can be used by any program running on the microprocessor to immediately jump to some type of maintenance routine whose address does not have to be known by the calling program. When the software interrupt instruction is executed, the 6800 reads the appropriate interrupt vector from memory and jumps to the indicated address. The 6800's reset and interrupt vectors are located at the top of memory, as listed in Table 6.1, which generally dictates that the boot ROM be located there as well. For example, an 8-kB 27C64 EPROM (8,192 bytes = 0x2000 bytes) would occupy the address range 0xE000 through 0xFFFF. Each vector is 16 bits wide, enough to specify the full address of the associated routine. The MSB of the address, A[15:8], is located in the low, or even, byte address, and the LSB, A[7:0] is located in the high, or odd, byte address.

TABLE 6.1 6800 Reset and Interrupt Vectors

Vector Address	Purpose
0xFFFE/0xFFFF	Reset
0xFFFC/0xFFFD	Nonmaskable interrupt
0xFFFA/0xFFFB	Software interrupt
0xFFF8/0xFFF9	Maskable interrupt

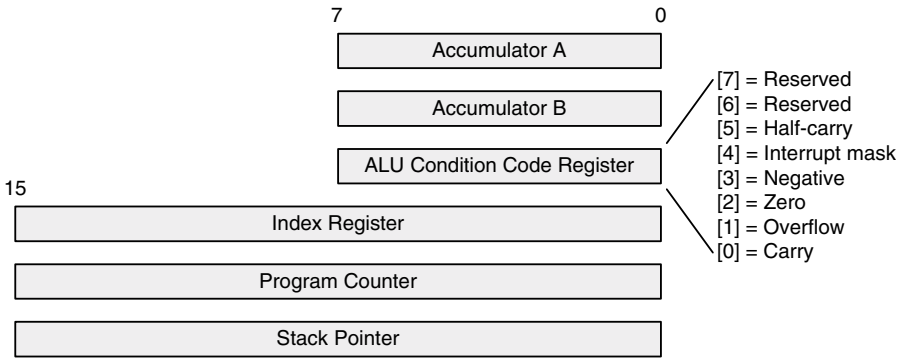

FIGURE 6.1 6800 registers.