IBM PC because of its simplicity and low cost. The 8048 was manufactured in a 40-pin DIP and could be expanded with external memory and peripherals via an optional external address/data bus. However, when operated as a nonexpanded single-chip computer, the pins that would otherwise function as its bus were available for general I/O purposes—a practice that is fairly standard on microcontrollers.

Motivated by the popularity of the 8048, Intel introduced the 8051 microcontroller in 1980, which is substantially more powerful and flexible. The 8051's basic architecture is shown in Fig. 6.3. It contains 128 bytes of RAM, 4 kB of ROM, two 16-bit timer/counters, and a serial port. Registers within the microprocessor are 8 bits wide except for the 16-bit data pointer (DPTR) and program counter (PC). Memory is divided into mutually exclusive program and data sections that each can be expanded up to 64 kB in size via an external bus. Expansion is accomplished by borrowing pins from two of the four 8-bit I/O ports. Intel manufactured several variants of the 8051. The 8052 doubled the amount of on-chip memory to 256 bytes of RAM and 8 kB of ROM and added a third timer. The 8031/8032 are 8051/8052 chips without on-board ROM. The 8751/8752 are 8051/8052 devices with EPROM instead of mask ROM. As time went by and the popularity of the 8051 family increased, other companies licensed the core architecture and developed many variants with differing mixes of memory and peripherals.

Ports 0 through 3 are each eight-bit bidirectional I/O structures that can be used as either general-purpose signals or as dedicated interface signals according to the system configuration. In a single-chip configuration where all memory is contained on board, the four ports may be assigned freely. Some peripheral functions use these I/O pins, but if a specific function is not required, the pins may be used in a generic manner. Port 3 is the default peripheral port where pins are used for the serial port's transmit and receive, external interrupt request inputs, counter increment inputs, and external bus expansion control signals. Port 1 is a general-purpose port that is also assigned for additional peripheral support signals when an 8051 variant contains additional peripheral functions beyond what can be supported on port 3 alone.

In a multichip configuration where memory and/or additional peripherals are added externally, ports 0 and 2 are used for bus expansion. Port 0 implements a multiplexed address/data bus where the 8051 first drives the lower eight address bits and then either drives write-data or samples read-data in a conventional bidirectional data bus scheme. In this standard configuration, the lower address bits, A[7:0], are latched externally by a discrete logic chip (generally a 74LS373 or similar), and the 8051 drives an address latch enable (ALE) signal to control this latch as shown in Fig. 6.4. This multiplexed address/data scheme saves precious pins on the microcontroller that can be used

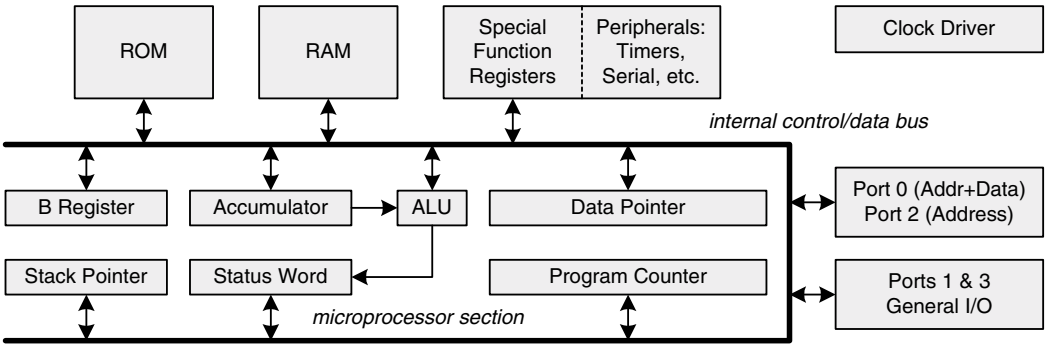

FIGURE 6.3 8051 overall architecture.

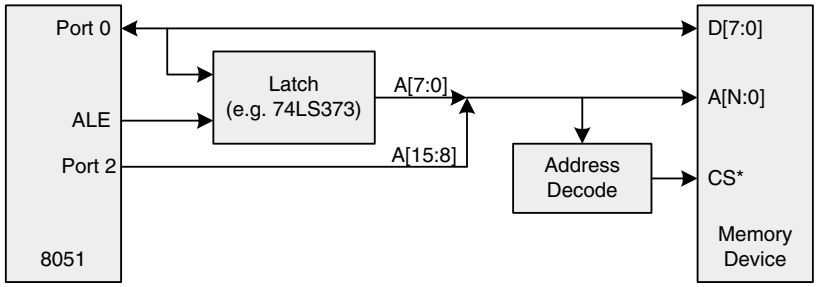

FIGURE 6.4 8051 system with external address latch.

for valuable I/O functions. Some applications may suffice with just an eight-bit external address bus. For example, if the only expansion necessary were a special purpose I/O device, 256 bytes would probably be more than enough to communicate with the device. However, some applications demand a fully functional 16-bit external address bus. In these situations, port 2 is used to drive the upper address bits, A[15:8].

The 8051's microprocessor is very capable for such an early microcontroller. It includes integer multiply and divide instructions that utilize eight-bit operands in the accumulator and B register, and it then places the result back into those registers. The stack, which grows upward in memory, is restricted to on-board RAM only (256 bytes at most), so only an eight-bit stack pointer is implemented. Aside from the general-purpose accumulator and B registers, the 8051 instruction set can directly reference 8 byte-wide general-purpose registers, numbered R0 through R7, that are mapped as 4 banks in the lower 32 bytes of on-board RAM. The active register bank can be changed at any time by modifying two bank-select bits in the status word. The map of on-board data memory is shown in Table 6.2. At reset, register bank 0 is selected, and the stack pointer is set to 0x07, meaning that the stack will actually begin at location 0x08 when the first byte is eventually pushed. Above the register banks is a 16-byte (128-bit) region of memory that is bit addressable. Microcontroller applications often involve reading status information, checking certain bits to detect particular events, and then triggering other events. Using single bits rather than whole bytes to store status information saves precious memory in a microcontroller. Therefore, the 8051's bit manipulation instructions can make efficient use of the chip's resources from both instruction execution and memory usage perspectives. The remainder of the lower 128-byte memory region contains 80 bytes of general-purpose memory.

The upper 128 bytes of data memory are split into two sections: special-function registers and RAM. Special-function registers are present in all 8051 variants, but their definitions change according to the specific mix of peripherals in each variant. Some special-function registers are standard across all 8051 variants. These registers are typically those that were implemented on the original 8051/8052 devices and include the accumulator and B registers; the stack pointer; the data pointer; and serial port, timer, and I/O port control registers. Each time a manufacturer adds an on-board peripheral to the 8051, accompanying control registers are added into the special-function memory region.

On variants that incorporate 256 bytes of on-board RAM, the upper 128 bytes are also mapped into a parallel region alongside the special-function registers. Access between RAM and special-function registers is controlled by the addressing mode used in a given instruction. Special-function registers are accessed with direct addressing only. Therefore, such an instruction must follow the opcode with an eight-bit address. The upper 128 bytes of RAM are accessed with indirect addressing only. Therefore, such an instruction must reference one of the eight general-purpose registers (R0