A typical DMAC supports multiple channels, each of which controls a different DMA transfer. While only one transfer can execute at any given moment, multiple transfers can be interleaved to prevent one peripheral from being starved for data while another is being serviced. Because a typical peripheral transfer is not continuous, each DMA channel can be assigned to each active peripheral. A DMAC can have one channel configured to load incoming data from a serial controller, another to store data to a disk drive controller, and a third to move data from one region of memory to another. Once initialized by the microprocessor, the exact order and interleaving of multiple channels is resolved by the individual DMA request signals, and any priority information is stored in the DMAC.

When a DMAC channel has completed transferring the requested quantity of data, the DMAC asserts an interrupt to the microprocessor to signal that the data has been moved. At this point, the microprocessor can restart a new DMA transfer if desired and invoke any necessary routines to process data that has been moved.

External DMA support logic may be necessary, depending on the specific DMAC, microprocessor, and peripherals that are being used. Some microprocessors contain built-in DMAC arbitration logic. Some peripherals contain built-in DMA request logic, because they are specifically designed for these high-efficiency memory transfers. Custom arbitration logic typically functions by waiting for the DMAC to request the bus and then pausing the microprocessor's bus transfers until the DMAC relinquishes the bus. This pause operation is performed according to the specifications of the particular microprocessor. Custom peripheral control logic can include DMAC read/write interface logic to assert the correct peripheral address when a transfer begins and perform any other required mapping between the DMAC's transfer enable signaling and the peripheral's read/write interface.

3.8 EXTENDING THE MICROPROCESSOR BUS

A microprocessor bus is intended to directly connect to memory and I/O devices that are in close proximity to the microprocessor. As such, its electrical and functional properties are suited for relatively short interconnecting wires and relatively simple device interfaces that respond with data soon after the microprocessor issues a request. Many computers, however, require some mechanism to extend the microprocessor bus so that additional hardware, such as plug-in expansion cards or memory modules, can enhance the system with new capabilities. Supporting these modular extensions to the computer's architecture can be relatively simple or quite complex, depending on the required degree of expandability and the physical distances across which data must be communicated.

Expansion buses are generally broken into two categories, memory and I/O, because these groups' respective characteristics are usually quite different. General-purpose memory is a high-bandwidth resource to which the microprocessor requires immediate access so that it can maintain a high level of throughput. Memory is also a predictable and regular structure, both logically and physically. If more RAM is added to a computer, it is fairly certain that some known number of chips will be required for a given quantity of memory. In contrast, I/O by nature is very diverse, and its bandwidth requirements are usually lower than that of memory. I/O expansion usually involves cards of differing complexity and architecture as a result of the wide range of interfaces that can be supported (e.g., disk drive controller versus serial port controller). Therefore, an I/O expansion bus must be flexible enough to interface with a varying set of modules, some of which may not have been conceived of when the computer is first designed.

Memory expansion buses are sometimes direct extensions of the microprocessor bus. From the preceding 8-bit computer example, the upper 16 kB of memory could be reserved for future expansion. A provision for future expansion could be as simple as adding a connector or socket for an extra memory chip. In this case, no special augmentation of the microprocessor bus is required. However, in a larger system with more address space, provisions must be made for more than one

additional memory chip. In these situations, a simple *buffered* extension of the microprocessor bus may suffice. A buffer, in this context, is an IC that passes data from one set of pins to another, thereby electrically separating two sections of a bus. As shown in Fig. 3.12, a buffer can extend a microprocessor bus so that its logical functionality remains unchanged, but its electrical characteristics are enhanced to provide connectivity across a greater distance (to a multichip memory expansion module). A unidirectional address buffer extends the address bus from the microprocessor to expansion memory devices. A bidirectional data buffer extends the bus away from the microprocessor on writes and toward the microprocessor on reads. The direction of the data buffer is controlled according to the state of read/write enable signals generated by the microprocessor.

More complex memory structures may contain dedicated memory control logic that sits between the microprocessor and the actual memory devices. Expanding such a memory architecture is generally accomplished by augmenting the "back-side" memory device bus as shown in Fig. 3.13 rather than by adding additional controllers onto an extended microprocessor bus. Such an expansion scheme may or may not require buffers, depending on the electrical characteristics of the bus in question.

I/O buses may also be direct extensions of the microprocessor bus. The original expansion bus in the IBM PC, developed in the early 1980s, is essentially an extended Intel 8088 microprocessor bus that came to be known as the *Industry Standard Architecture* (ISA) bus. Each I/O card on the ISA bus is mapped in a unique address range in the microprocessor's memory. Therefore, when software wants to read or write a register on an I/O card, it simply performs an access to the desired location. The ISA bus added a few features beyond the raw 8088 bus, including DMA and variable *wait states* for slow I/O devices. A wait state results when a device cannot immediately respond to the microprocessor's request and asserts a signal to stretch the access so that it can respond properly.

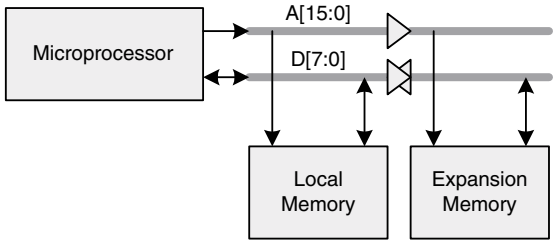

FIGURE 3.12 Buffered microprocessor bus for memory expansion.

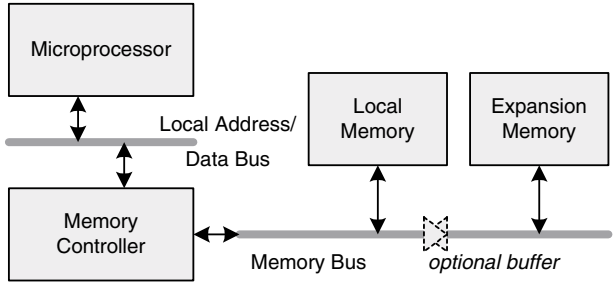

FIGURE 3.13 Extended memory controller bus.