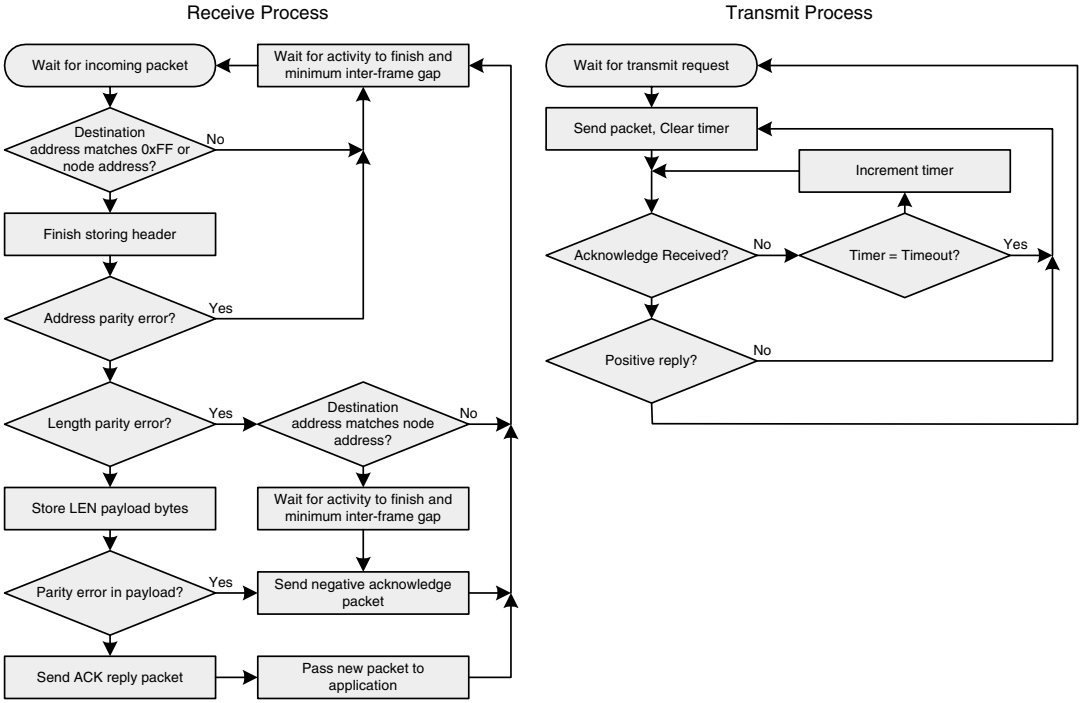

FIGURE 5.17 Hypothetical network driver flowchart.

to traditional parallel buses. Computer architectures often include a variety of microprocessor peripheral devices with differing bandwidth requirements. Main memory, both RAM and ROM, is a central part of computer architecture and is a relatively high-bandwidth element. The fact that the CPU must continually access main memory requires a simple, high-bandwidth interface—a parallel bus directly or indirectly driven by the CPU. Other devices may not be accessed as often as main memory and therefore have a substantially lower bandwidth requirement. Peripherals such as data acquisition ICs (e.g., temperature sensors), serial number EEPROMs, or liquid crystal display (LCD) controllers might be accessed only several times each second instead of millions of times per second. These peripherals can be directly mapped into the CPU's address space and occupy a spot on its parallel bus, but as the number of these low-bandwidth peripherals increases, the complexity of attaching so many devices increases.

Short-distance serial data links can reduce the cost and complexity of a computer system by reducing interchip wiring, minimizing address decoding logic, and saving pins on IC packages. In such a system, the CPU is connected to a serial controller via its parallel bus, and most other peripherals are connected to the controller via several wires in a bus topology as shown in Fig. 5.18.

Such peripherals must be specifically designed with serial interfaces, and many are. It is common for low-bandwidth peripheral ICs to be designed in both parallel and serial variants. In fact, some devices are manufactured with only serial interfaces, because their economics overwhelmingly favors the reduction in logic, wiring, and pins of a serial data link. A temperature sensor with a serial interface can be manufactured with just one or two signal pins plus power. That same sensor might

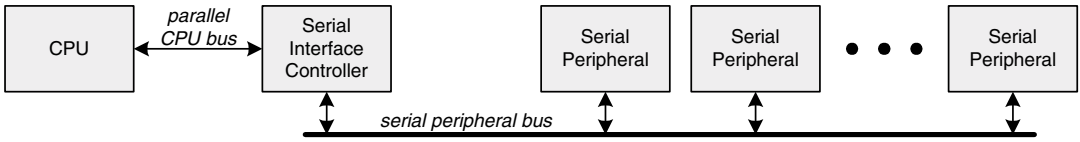

FIGURE 5.18 Generic interchip serial bus topology.

require 16 or more signal pins with a byte-wide parallel interface. Not only is the package cost reduced, its greatly reduced size enables the IC to be located in very confined spaces. Products including cell phones and handheld computers benefit tremendously from small IC packages that enable small, consumer-friendly form factors.

Interchip serial interfaces must be kept fairly simple to retain their advantages of low cost and ease of use. Industry standard interfaces exist so that semiconductor manufacturers can incorporate mainstream interfaces into their ICs, and engineers can easily connect multiple manufacturers' ICs together without redesigning the serial interface for each application. Many of these standard interfaces are actually proprietary solutions, developed by individual semiconductor manufacturers, that have gained wide acceptance. Two of the most commonly used industry standards for interchip serial communications are Philips' *inter-IC bus* (I^2C) and Motorola's *serial peripheral interface* (SPI). Both Philips and Motorola have long been leaders in the field of small, single-chip computers called *microcontrollers* that incorporate microprocessors, small amounts of memory, and basic peripherals such as UARTs. It was therefore a natural progression for these companies to add inexpensive interchip serial data links to their microcontrollers and associated peripheral products.

I^2C and SPI support moderate data rates ranging from several hundred kilobits to a few megabits per second. Because of their target applications, these networks usually involve a single CPU master connected to multiple slave peripherals. I^2C supports multiple masters and requires only two wires, as compared to SPI's four-plus wires.

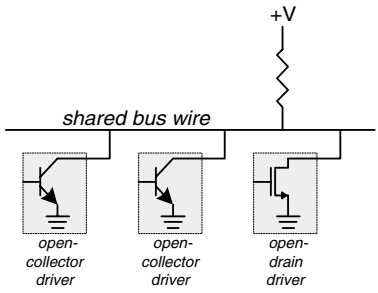

FIGURE 5.19 I^2C open-collector schematic representation.

I^2C consists of a clock signal, SCL, and a data signal, SDA. Both are *open-collector* signals, meaning that the ICs do not actively drive the signals high, only low. An open-collector driver is similar to a tri-state buffer, although no active high state is driven. Instead, the output is at either a low- or high-impedance state. The open-collector configuration is schematically illustrated in Fig. 5.19. The term *open-collector* originates from the days of bipolar logic when NPN output transistors inside the chips had no element connected to their collectors to assert a logic high. This terminology is still used for CMOS logic, although *open-drain* is the technically correct term when working with MOSFETs. A *pullup resistor* is required on each signal (e.g., SCL and SDA) to pull it to a logic 1 when the ICs release the actively driven logic 0. This open-collector arrangement enables multiple IC drivers to share the same wire without concern over electrical contention.

Under an idle condition, SCL and SDA are pulled high by their pullup resistors. When a particular IC wants to communicate, it drives a clock onto SCL and a pattern of data onto SDA. SCL may be as fast as 100 kHz for standard I^2C and up to 400 kHz for fast I^2C buses. I^2C is a real network that assigns a unique node address to each chip connected to the bus. As such, each transfer begins with a start sequence followed by seven-bit destination address. A read/write flag and data follow the ad-