put will cause unacceptable skew to develop with respect to the other pins that are driving only a single load. Whether this occurs depends on each situation's skew tolerance, clock buffer drive strength, and input impedance of the clock load.

As the number of clock loads in a system increases, there comes a point at which clock distribution cannot be performed using a single clock buffer IC. Using multiple clock buffers introduces a new source of skew: part-to-part propagation delay variation. It is easier to manage skew on a single chip, because each output circuit has very similar physical properties to its neighbors, and the temperature variation across a small chip is very low. Neither of these assumptions automatically holds true when dealing with separate ICs. Some low-skew clock buffers are specially designed and fabricated for low part-to-part skew specifications. Newer devices are available with part-to-part skews of 0.5 ns, and more mature devices with 1 ns part-to-part skews are available as well. Figure 16.6 shows how two clock buffers might be connected to drive twice the number of loads as shown previously. Note that the oscillator directly drives both buffers in parallel without any termination, because it is assumed that the wire length is short enough to minimize reflections. If more than a couple of buffers are necessary to distribute clocks to the system, an intermediate buffer should probably be placed between the oscillator and final buffers so that the oscillator does not have to drive too many loads.

16.3 ZERO-DELAY BUFFERS: THE PLL

Low-skew buffers do an excellent job of distributing clocks with minimal relative delays when there is no constraint on the absolute propagation delay from the clock tree input to its outputs. However, there are circumstances where the absolute delay is as important as the relative delay. Many microprocessors with synchronous bus interfaces generate their own bus clock internally and emit this clock on an output pin. All bus signals must be timed relative to that output clock to meet the microprocessor's timing rules of t_{CO}, t_{SU}, and t_H. If the microprocessor communicates with just one pe-

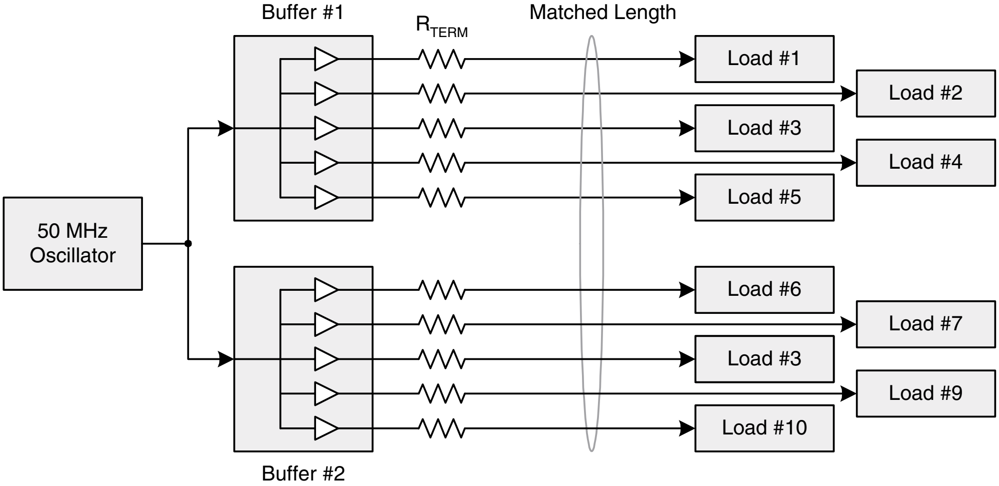

FIGURE 16.6 Dual-buffer clock tree.

ripheral IC that is located close to it, the peripheral's clock input can be driven directly by the microprocessor's clock output, and all should be okay. Chances are great, however, that the microprocessor must communicate with several peripheral and memory ICs that are located a significant distance away, as shown in Fig. 16.7. Placing a low-skew buffer between the microprocessor's clock output and the other ICs will not solve the problem, because the clocks at the loads will have a large skew relative to the microprocessor clock.

Another common clock distribution problem with an absolute delay constraint is driving common clocks to an expansion board. Consider the system shown in Fig. 16.8 in which a base board contains multiple ICs connected via a common synchronous bus. An expansion board can be plugged in to allow ICs on that card to communicate over the common bus on the base board. How can all the ICs on the base and expansion boards receive low-skew clocks? If the expansion board design is fixed ahead of time, a low-skew buffer tree on the base board can send the required number of clock signals to the expansion card and achieve length matching on both boards. This works as long as the expansion board's design never changes. A problem appears if a later design is intended to add more ICs with more clocks, or if the wiring of the expansion board can no longer maintain a length match with the wires on the base board. In all cases, as the number of individual clock loads increases on

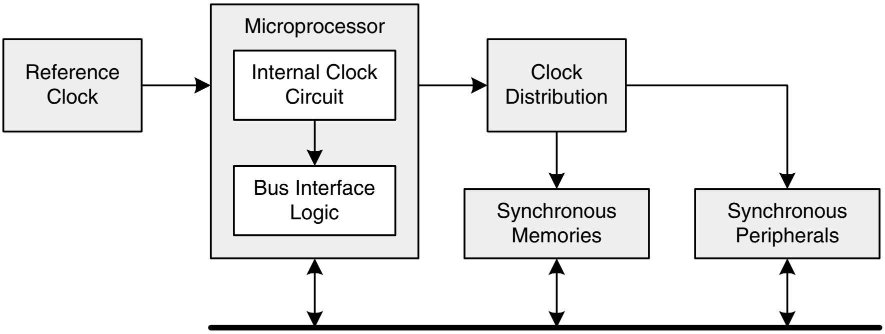

FIGURE 16.7 Microprocessor bus-clock master.

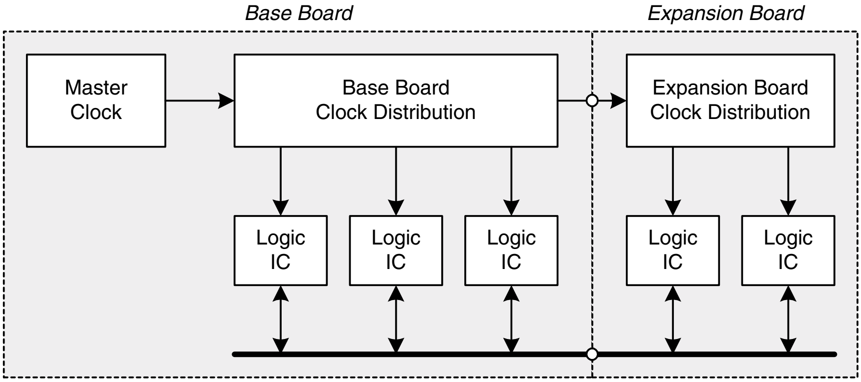

FIGURE 16.8 Clock distribution to an expansion board.