

Half-Duplex Interaction. The alternative to a full-duplex interface, known as a *half-duplex* interface, only allows transfer to proceed in one direction at a time. That is, a single set of wires that connects the processor and the external device must be shared. In the next chapter, we will see that sharing requires negotiation — before it can perform a transfer, a processor or device must wait for the current transfer to finish, and must obtain exclusive use of the underlying wires.

13.8 Interface Latency And Throughput

Because it can only send one bit at a time, a serial interface operates slower than a parallel interface. As we have seen with memories, however, we must be careful to distinguish between *latency* and *throughput*. Latency refers to the delay between the time a bit is sent and the time the bit is received (i.e., how long it takes to transfer a single bit). Thus, latency is usually measured in nanoseconds (ns). Throughput refers to the number of bits that can be transferred per unit time, and is usually measured in *Mega-bits per second (Mbps)* or *Megabytes per second (MBps)*.

We can summarize:

The latency of an interface is a measure of the time required to perform a transfer, the throughput of an interface is a measure of the data that can be transferred per unit time.

13.9 The Fundamental Idea Of Multiplexing

It may seem that choosing an interface is trivial: a full-duplex, parallel interface offers more performance than any other combination. The difference in performance between a serial and a parallel interface can be dramatic: a parallel interface with width N has a throughput that is N times higher than a serial interface. Similarly, increasing the width of a parallel interface increases performance (e.g., doubling the width of an interface doubles the throughput).

Despite higher performance, many other factors make the choice of interfaces complex. Recall, for example, each integrated circuit has a fixed number of *pins* that provide external connections. A wider interface uses more pins, which means fewer pins for other functions. Similarly, an interface that provides full-duplex capability uses approximately twice as many pins as an interface that provides half-duplex capability.

Most architects choose a compromise for external connections. The connection has *limited parallelism*, and the hardware uses a technique known as *multiplexing* to send data. Although details are complex, the concept of multiplexing is easy to understand. The idea is that the hardware breaks a large data transfer into pieces and sends one piece at a time. We use the terms *multiplexor* and *demultiplexor* to describe the hardware that handles data multiplexing. For example, Figure 13.3 illustrates the multi-