can generate. In large programs, the slight inefficiency of the compiler is well worth the trade-off for ease of programming in a high-level language. However, time-critical routines such as I/O drivers or ISRs may benefit from manual assembly language coding.

CHAPTER 4
Memory

Memory is as fundamental to computer architecture as any other element. The ability of a system's memory to transact the right quantity of data in the right span of time has a substantial impact on how that system fulfills its design goals. Digital engineers struggle with innovative ways to improve memory density and bandwidth in a way that is tailored to a specific application's performance and cost constraints.

Knowledge of prevailing memory technologies' strengths and weaknesses is a key requirement for designing digital systems. When memory architecture is chosen that complements the rest of the system, a successful design moves much closer to fruition. Conversely, inappropriate memory architecture can doom a good idea to the engineering doldrums of impracticality brought on by artificial complexity.

This chapter provides an introduction to various solid-state memory technologies and explains how they work from an internal structural perspective as well as an interface timing perspective. A memory's internal structure is important to an engineer, because it explains why that memory might be more suited for one application over another. Interface timing is where the rubber meets the road, because it defines how other elements in the system can access memory components' contents. The wrong interface on a memory chip can make it difficult for external logic such as a microprocessor to access that memory and still have time left over to perform the necessary processing on that data.

Basic memory organization and terminology are introduced first. This is followed by a discussion of the prevailing read-only memory technologies: EPROM, flash, and EEPROM. Asynchronous SRAM and DRAM technologies, the foundations for practically all random-access memories, are presented next. These asynchronous RAMs are no longer on the forefront of memory technology but still find use in many systems. Understanding their operation not only enables their application, it also contributes to an understanding of the most recent synchronous RAM technologies. (High-performance synchronous memories are discussed later in the book.) The chapter concludes with a discussion of two types of specialty memories: multiport RAMs and FIFOs. Multiport RAMs and FIFOs are found in many applications where memory serves less as a storage element and more as a communications channel between distinct logic blocks.

4.1 MEMORY CLASSIFICATIONS

Microprocessors require memory resources in which to store programs and data. Memory can be classified into two broad categories: volatile and nonvolatile. Volatile memory loses its contents when power is turned off. Nonvolatile memory retains its contents indefinitely, even when there is no power present. Nonvolatile memory can be used to hold the boot code for a computer so that the microprocessor can have a place to get started. Once the computer begins initializing itself from nonvolatile memory, volatile memory is used to store dynamic variables, including the stack and other programs that may be loaded from a disk drive. Figure 4.1 shows that a general memory device consists of a bit-storage array, address-decode logic, input/output logic, and control logic.