the hardware. A RISC microcontroller can execute instructions very fast, but in a given application it may not be as fast as a CISC (complex instruction set computer) with an instruction set that can perform complex operations. For example, multiplying two 16-bit numbers may take one instruction and only a few clock cycles on a CISC processor or a single cycle on a DSP with multiplier hardware. On a RISC processor that has no multiply instruction or multiply hardware, this operation must be implemented in some kind of loop that uses several instructions and a large number of clock cycles. On the other hand, an application that does a lot of bit flipping and sensor reading, with little or no complex math, may get better performance from a RISC processor.

Not evaluating the architecture. The ADSP-2100 family parts from Analog Devices are DSPs that lend themselves well to embedded applications. These parts are optimized for signal processing, which means that they have some powerful data manipulation capabilities such as hardware multiply and barrel shifters. However, they also have some limitations. Some operations require an extra instruction to move a value from RAM to a register before it can be used, whereas other, slower processors allow the value in RAM to be manipulated or tested directly.

These are typical and by no means unique. Every processor has its quirks, and these are not dark secrets. You just must understand the data sheets on the part before you use it. Take the data book or CD-ROM home. Read it. Study the timing diagrams, especially the worst-case numbers. Understand how everything in your system will connect to and be controlled by this processor. If you do not understand something, you are not ready to start the design.

ROMability

This consideration applies only to those devices that execute their programs from internal ROM. These devices usually are chosen for an application where cost, rather than being no object, is a key factor. If the finished design is going to be a very high-volume (thousands per year) product, it may be worthwhile to select a processor that has a ROM version.

Most engineering projects use EPROM or flash memory for their development phases. These erasable and reprogrammable memories allow a part to be reused instead of thrown away. When the part goes to production, the EPROM parts can be replaced with one-time programmable (OTP) devices. These usually are just EPROM-based parts in a plastic package with no erasure window. Since the expensive ceramic package and quartz window are not required, the OTP parts are cheaper than the EPROM parts to manufacture, thus reducing product costs.

If the production volume is high enough, the EPROM part can be replaced with a mask ROM version. The designer supplies the finished program to the IC manufacturer, who creates a mask for the version of the IC that has an internal ROM. This provides the lowest production cost. However, the following caveats exist:

- There is a mask charge to produce the ROM. This charge is usually several thousand dollars and is usually tied to a minimum purchase requirement. If the product volumes are less than expected or (get your résumé ready) a bug is discovered in the program after the ROM is created, the mask charge is not recoverable. A new NRE (nonrecurring expense) is required for a new mask, and all the old parts must be scrapped because the ROM program cannot be changed.
- Some manufacturers are so swamped with mask order requests that they have stopped accepting them. This can be disastrous if your entire product pricing strategy is based on the availability of mask ROM parts. A list of these manufacturers, even assuming I knew who they all were this week, would be useless by the time this book reaches print. Check into this before deciding to use a ROM part.

Even though the production costs are low, the high upfront costs prevent many designers from using mask ROM parts. If your volume is too low or you know the design will change before the end of product life, then mask ROMs usually are a poor choice.

One additional consideration is that not all devices are available in all flavors. For example, the Motorola 68HC05 series parts are designed for extremely high-volume applications. Not all parts in the series (and there seem to be more every month) are available in the EPROM version. Some parts are available *only* in the ROM version. Development is done on a similar part for which an EPROM version is available. The catch is, if you cannot justify the ROM costs, you cannot select these ROM-only devices, and the nearest equivalent EPROM part may be too costly for your use.

Another example is the 8031 family parts, which are available in EPROM, OTP, and ROM versions. As of this writing, the cost of the EPROM version is about 10 times the cost of the ROM version, and the OTP is about 60 percent of the EPROM version, depending, of course, on your volume and where you buy the parts. The basic ROM 8031 may be the cheapest choice, but if you will not have the volume to use it, the OTP version of a different processor may be cheaper than the OTP 8031. The device with the lowest cost in a ROM version may not be the cheapest in the OTP. In addition, for some devices, the OTP is not available. Your choices are EPROM or ROM, which can make these parts a real cost problem in low-volume applications. Be sure to research which varieties of a part are available based on your volume and other product requirements.

Finally, remember that once a design is committed to mask ROM, it has the same inflexibility as a non-microprocessor-based hardware design. Once you go to ROM,