Controllers

What is the difference between a microprocessor, a microcontroller, and a digital signal processing chip? Why would you select one of those devices over either of the others?

Before you can find the answers to those questions, you should first look into a concept called *address space*. Figure 3 shows some details. The address space is the sum total "reach" of a programmable IC. As in a city directory, an address space has *locations*. Into each location, you can put one "unit" of information.

This unit of information is often called a *byte*. Any given byte can be a computer command, one piece of data, one character in a document, a musical note, or an I/O "window" to the outside world. Bytes are eight bits in length. One byte is capable of representing as many as 256 different values. Multiple bytes can be combined for additional values.

The hardware located in any particular address space location is often RAM, ROM, I/O, or nothing. RAM is any memory that is fast and easy to change. ROM is any memory that stays more or less permanent, even on power-down. These days, of course, RAM is becoming very ROM-like, and vice-versa. There is a continuum of intelligently chosen options here.

I/O is short for *input* or *output*, and is the way your computer circuitry can reach the outside world. And finally, not all of the address space must actually be used. Some of it could be empty; other portions can be available but not in use.

The size of the address space can be as little as 256 bytes in certain controllers. A classic 8-bit microcomputer has an address space of 64K, or 65536 locations. Newer chips have

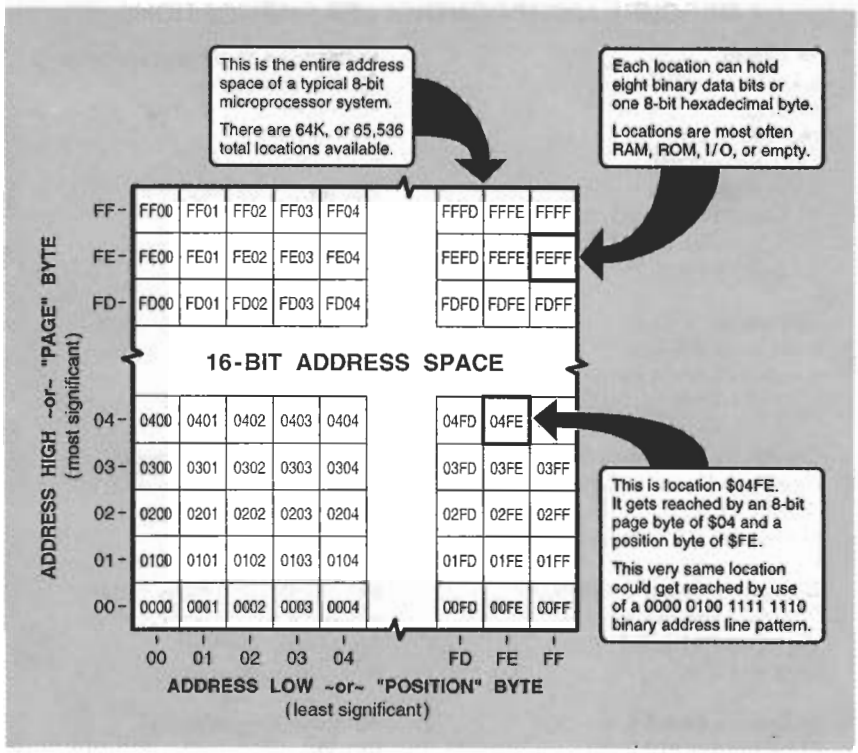

FIG. 3—THE ADDRESS SPACE of a classic microprocessor.

address spaces of 24 bits (for a grand total of 16,777,216 locations), or 32 bits (for a vast 4,294,967,296 possible locations).

The big tradeoff between microprocessors and microcontrollers is chip count. A microcontroller with internal-only address space often allows your design to get by with a single integrated circuit. But the RAM and ROM provided is both fixed and restricted. An external address space lets you add nearly anything you want. Almost any way you like, for much greater flexibility and performance at higher system cost.

Some single-chip solutions let you have the best of both worlds. While their address space stays fixed and internal, they easily let you add low-cost *serial* memory chips for extra data storage and whatever. Although serial memory is much slower than memory inside the "real" address space, it is more than fast enough for many real-world applications. Figure 4 summarizes the key differences between these main three computing options.

A *microprocessor* normally has a large address space, a generalized instruction set, and requires additional chips for a complete system. A *microcontroller* normally has a small internal address space, and its instruction set is optimized for control and other bit manipulations. Microcontrollers usually provide a single-chip solution. A *digital signal processor* (or DSP) is just a different name for a microprocessor, except that it uses fewer, more specialized commands.

The instructions for a DSP chip are carefully optimized for digital filter uses and related tasks, and thus aren't all that great for anything else. The specialized DSP instructions might include an ultra fast multiplier, a multiplier/accumulator that adds a small value to a large total, or some *barrel shifter* that instantly multiplies or divides by powers of two.

There are certain things that DSP chips do very well. But their highly specialized nature and their military heritage has kept them expensive and out of the mainstream. Sadly, DSP program development and emulation remain as costly hassles. But DSP popularity is very much on the upswing, especially for applications in sound generation, fancy filters, and video compressors.

By far the finest choice in any microcontroller today is the PIC from *Microchip Technology*. The best way to get started using microcontrollers is with the *Basic Stamp* from *Parallax* and the *Scott Edwards Tools* for full machine code access and speed.

By special arrangement, I've made the complete Basic Stamp manuals available on my *GEnie* PSRT. See BASTAMP1.PDF for the intro, BASTAMP2.PDF for the instruction set, and BASTAMP4.PDF for updated and expanded applications.

Most **MICROPROCESSOR** chips have a large and external address space. Into which you place your choice of additional RAM, ROM, I/O or nothing. The instruction set is a general one, often optimized for file manipulation and complex computing tasks.

Instead, a **MICROCONTOLLER** offers a single chip solution with a smaller and fixed internal address space. All resources are defined on chip, with the possible exception of some slower and serial EEPROM memory. The usual instruction set has additional commands in it that favor simple I/O tasks. Such as bit sets, clears, or tests.

Finally, a **DIGITAL SIGNAL PROCESSOR** also usually has a large and an externally accessible address space. But its instruction set is often strictly limited to specialized tasks. Such as a very rapid multiplier, one or more adder/accumulators, or a barrel shifter.