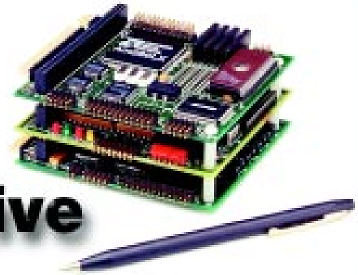
PC/104

The non-backplane alternative



By Rick Lehrbaum

Designers of tomorrow's embedded systems face a dazzling set of opportunities and challenges – both technological and logistical – in seeking to harness the potential offered by the dramatic evolution of computer hardware and software technologies over the last decade. Powerful microprocessors – coupled with exploding capacities of program and storage memory and supported by highly advanced software environments – enable vast increases in the sophistication of data monitoring, process control, and user interfaces. What in the past required the resources of a mini or mainframe computer, can now be embedded directly within many types of products. Highly sophisticated equipment can now be made "user friendly" by virtue of highly intelligent graphical user interface software.

What is the best way for embedded system designers to take advantage of these advances in technology, yet bring new products to market in a timely manner and within budget constraints? One solution is to cleverly combine product-specific proprietary hardware and software along with appropriate off-the-shelf embedded-computer hardware and software "building blocks."

Embedded computer evolution

Embedded computer technology has come a long way. Since the birth of the microprocessor in the mid-1970s, we have witnessed continuous exponential growth in CPU performance and memory capacity – for both program execution and data storage. By the '80s, a few megahertz and a few kilobytes were the norm. In the '90s, this became tens of megahertz and tens of megabytes. Currently, embedded computer CPUs are clocked at up to 200 MHz, RAM capacities are beginning to surpass 64 Mbytes, and mass storage is commonly measured in gigabytes!

As embedded computer speeds and memories grow ever larger, embedded applications become increasingly decoupled from the underlying embedded computer *architecture*. As this occurs, the real "magic" of the embedded system lies within its unique software, interface technology, peripherals, and packaging. As a result of this trend, developers now spend much more of their time being *product* architects rather than *embedded computer* architects.

The embedded computer is therefore increasingly perceived as a platform on which to run the application's software; and it is consequently most often *software*, not *hardware*, that represents the greatest technology concern and risk. The preferred embedded computer architecture is now typically that which optimizes the application's software development process, resulting in faster development cycles, reduced technical risks, and improved system sophistication. In today's fast-moving and competitive market for technology-based systems, any and all efficiencies are greatly welcomed.

An obvious way to increase the efficiency of embedded system development is to employ standardized – even off-the-shelf – hardware and software building blocks, if available. This would minimize the need to design from scratch. How can this be accomplished?

In response to this question, embedded system development teams have looked to the highly popular PC architecture to provide a standardized hardware and software toolkit. The enormous popularity of the PC architecture ("over 300 million sold") has generated a vast resource of *desktop* software and hardware. Can these be successfully adapted to embedded systems? If so, billions of dollars in R&D investment could be harnessed for the benefit of the embedded market.

The key desktop PC technologies of interest include CPUs, operating systems, user interface (hardware and software), mass storage (both magnetic and solid state), communications, and networking. Also – and not to be underestimated – is one of the greatest benefits of using a PC architecture: the enormously rich supply of easy-to-use, low-cost development tools, coupled with the fact that nearly every engineer, programmer, and technician is already knowledgeable on the use of PC hardware and software.

In short, a PC-compatible hardware/software *architecture* certainly brings the promise of great savings in development time and costs, reduced product material costs (due to less expensive chips and peripherals), and minimized maintenance and support headaches. But what about the idea of off-the-shelf *modules* that could be used in a Lego-like manner to rapidly develop and cost-effectively manufacture embedded computer-based products?

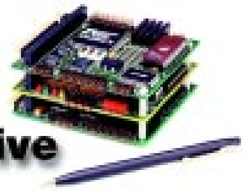
Unique requirements of the embedded market

Although PC technology offers many exciting possibilities to embedded system developers, such applications place severe demands and constraints on their internal electronics that are not applicable to desktop PCs. In the PC market, price pressures severely constrain such factors as:

- ❑ reliability
- ❑ ruggedness
- ❑ quality
- ❑ product longevity

PC/104

The non-backplane alternative



After all, the main objectives of desktop PC manufacturers are to minimize cost while continually incorporating the latest new technologies, in an effort to sell as many systems and system upgrades as possible. This results in low-cost, minimal quality, and extremely short product life cycles.

Embedded computers, on the other hand, must satisfy a whole different set of objectives – most of which run counter to the priorities of the desktop market. These include:

- ❑ minimized size and weight (so they can fit within space-limited products)
- ❑ reduced power consumption (to satisfy limited power budgets or even run off batteries)
- ❑ resistance to shock and vibration (to survive harsh embedded system environments)
- ❑ extended operating temperature (for operation in non-office applications)
- ❑ enhanced functional reliability (to eliminate the likelihood of unallowable system failures)

Furthermore, unlike desktop PCs, embedded computers are not general-purpose, user-programmable systems, but have highly specialized requirements. So they must be easily adaptable to application-specific requirements which often require custom interface electronics.

Also, product life cycles of embedded products are much longer than those of desktop-PCs. In fact, desktop-PC vendors consciously strive to continually introduce new technologies, in the hope of selling upgrades to their existing customer base. It's often said that the typical "half-life" (to obsolescence) of PC chipsets is around three "Comdexes" (1 Comdex = 6 months). Clearly, while it may be in the best interest of PC manufacturers to sell everybody a new PC (motherboard, video card, disk controller, network controller, etc.) every year or so, this situation represents an unacceptable risk for developers and manufacturers of embedded systems. Since embedded system products typically take 18-24 months to develop – and in some cases several more years to gain agency approvals – embedded system designs cannot be sensibly based on components with life cycles as short as 18 months!

The PC/104 alternative

Given the potential benefits of using a PC-compatible hardware/software *architecture* in an embedded system design – but in light of the shortcomings of normal desktop PC systems relative to the needs of the embedded electronics market – is there anything else that can allow embedded system designers to take advantage of PC technology?

Fortunately, the answer is yes – and it's spelled "PC/104" (see Figure 1). The PC/104 Embedded Computer Modules standard was introduced in 1992, to provide a modular building-block method of incorporating PC-compatible hardware and software technologies into embedded systems. PC/104 modules are expressly intended for a wide range of application environments – including fixed, portable, and mobile environments – in a broad variety of non-desktop embedded system markets.

Basically, PC/104 defines how to repackage desktop PC functions in a manner that satisfies the ruggedness, reliability, and size constraints of embedded systems. PC/104 offers full hardware and software compatibility with the desktop PC architecture, but in the form of compact (3.6" x 3.8"), self-stacking, modules (see Figure 2). Therefore, PC/104 offers a way to incorporate a PC-compatible architecture into an embedded system, based on off-the-shelf building blocks.

The non-backplane paradigm

Prior to the availability of PC/104, the options for embedding a PC architecture were to use a motherboard- or backplane-based approach (which is bulky and unreliable), or to create a custom embedded-PC based on individual chips (which is costly and time consuming). PC/104 modules are small enough to fit where a backplane-based approach won't, so they provide an excellent space-efficient "middle ground" for many embedded applications.

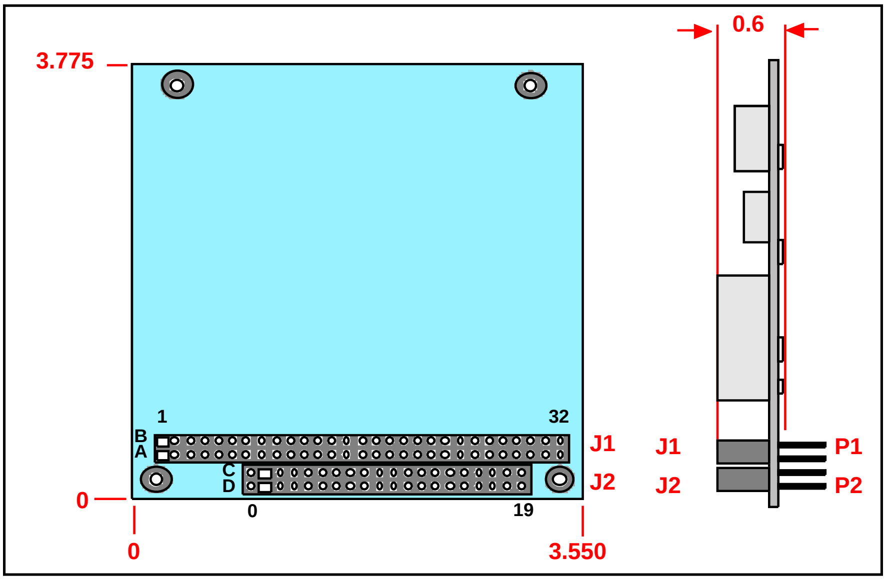


Figure 1

Figure 2

The differences between PC/104 and the "normal" PC are primarily mechanical; there are no software differences. The PC/104 specification (available from the PC/104 Consortium) defines a compact form factor, a self-stacking 104-signal pin-and-socket connector bus (to eliminate backplanes and card cages), and reduced bus drive (for lower power consumption and minimized components).

Another important advantage of PC/104 modules lies in the fact that they are not at all intended for the *desktop* market, but are explicitly meant to be used as *embedded* components within a wide range of OEM applications. Therefore, the suppliers of PC/104 modules typically concentrate on quality, reliability, service, and support – as demanded by this customer base and their widely varying applications.

Although configuration and application possibilities are practically limitless, PC/104 modules are generally used in three ways. One approach is to use PC/104 modules like ultra-compact backplane-bus boards, except without the backplanes and card cages. Highly compact PC/104 module stacks are thus "bolted" directly within a system's enclosure, in an otherwise empty space.

In a second approach, the PC/104 modules are distributed *horizontally* – plugged into custom, application baseboards like multi-chip "macrocomponents" (see Figure 3). The PC/104 application baseboard in such a system can include all interfaces and logic that are not available on (or, for whatever reason, are not desired on) PC/104 modules. The baseboard might include power conversion components, signal conditioning logic, specialized "real-world" interfaces and connectors, etc. It's important to note that devices on the baseboard don't actually have to *interface* with the PC/104 bus, but could be included simply to eliminate unnecessary additional assemblies.

As for what size and shape the application baseboard should be, it tends to take the shape of the *system* – which may be square, rectangular, or even round – whatever the application dictates! A spare PC/104 "socket" can be included on the application baseboard for future add-ons, but all that is needed to allow system expansion is 0.6 inch of vertical space for a piggybacked additional module.

The evolving PC/104 standard

Recently, a PCI-extended version of PC/104 called "PC/104-*Plus*" was adopted by the PC/104 Consortium (see Figure 4). The purpose of PC/104-*Plus* is to support high performance CPUs (e.g., Pentium) and PCI bus throughput for tomorrow's performance-intensive applications, which increasingly incorporate high speed graphics, net-working, and data crunching. PC/104-*Plus* preserves full backward compatibility with PC/104, including the ability to coexist within a stack with PC/104 modules. This further enhances the flexibility of PC/104 by allowing it to be used in increasingly performance-intensive applications.

Another approach to structuring a compact, reliable, fully compatible, and cost effective embedded-PC is to incorporate all the essential ingredients of a PC-compatible system onto a highly integrated single-board computer, or SBC. A highly integrated SBC can eliminate some of the bulk, weight, and costs of using multiple boards or modules. However, although the SBC may

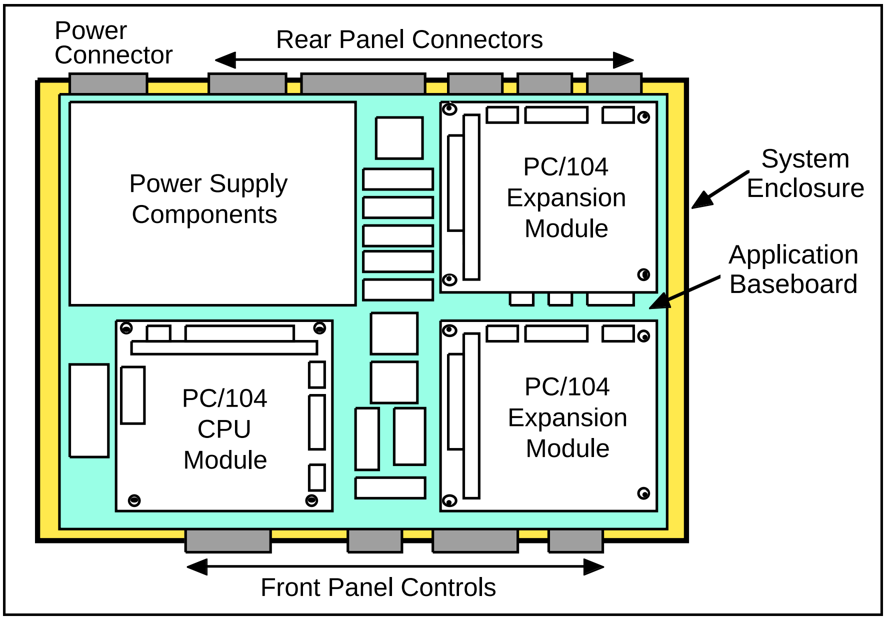


Figure 3

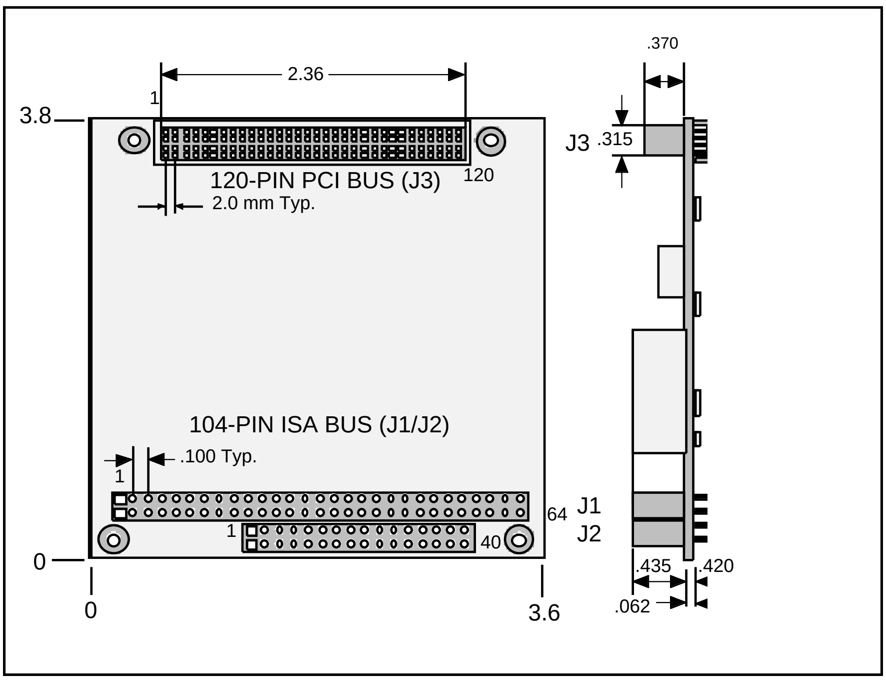


Figure 4

contain all of the normal PC-compatible functions (e.g., interfaces for keyboard, speaker, serial, parallel, disk, and network), a means is usually needed for adding the application-specific functions and interfaces that adapt the embedded-PC to its intended purpose in the particular embedded application. For this reason, it is important that a PC/104 (or PC/104-*Plus*) expansion stack location be provided on any embedded-PC SBC to allow installation of off-the-shelf PC/104 modules, or to serve as a standardized interface to whatever custom electronics the application may require.

Although such non-backplane SBCs have been around for over 15 years, no multi-vendor standards have existed for these kinds of products. Recently, however, Ampro and Motorola collaborated to generate the industry's first actual multi-vendor standard for highly integrated *non-backplane* SBCs. This new standard, called EBX (for "Embedded Board, eXpandable"), evolved from Ampro's "Little Board form factor" (5.75" x 8.0"), providing an SBC form factor that is large enough to accommodate a high level of functional integration and CPU performance, yet small enough to be "deeply embedded" within a wide variety of applications.

An important feature of EBX (see Figure 5) is its inclusion of an onboard PC/104-*Plus* expansion location, which facilitates adapting the SBC to various specific applications. Obvious benefits of using an EBX SBC is reduction in the number of required modules, and elimination of the electrical and mechanical interface "glue" required when using multiple modules. Another benefit lies in the ability of onboard devices such as memory, video, SCSI, or network controllers to take advantage of local bus data rates rather than being constrained by bus interface speeds (e.g., ISA or PCI). Cost savings may also result from a reduction in glue logic, connectors, and independent assemblies.

When comparing EBX with PC/104, an EBX form factor SBC is often preferred if the features of the EBX SBC closely match the needs of the application. For example, if a CPU, video, and Ethernet are all needed, an EBX SBC that contains them all makes a great fit. On the other hand, a PC/104 form factor SBC may be a better fit for applications requiring only a CPU, memory, and a some serial and parallel I/O. Typically, applications where the embedded-PC is to perform the functions of a traditional embedded *microcontroller* are well-suited to a PC/104 form factor SBC.

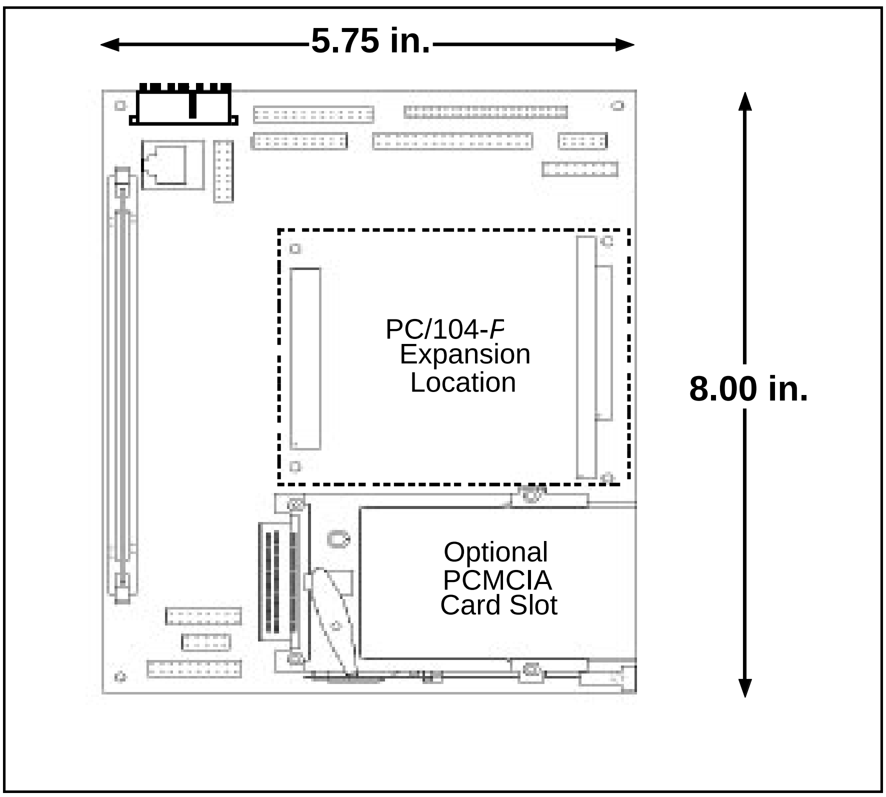
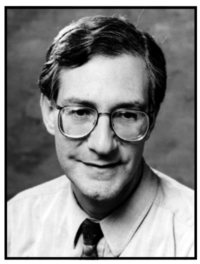


Figure 5

Summing it up

There are many good reasons to use the PC architecture as the hardware and software basis for embedded systems. However, standard desktop-PCs don't satisfy the space, power, ruggedness, quality, reliability, and longevity requirements of most embedded designs. The compact, modular PC-compatible PC/104 standard was developed specifically to provide PC architecture compatibility for *embedded* applications, and modules designed to this standard possess the important characteristics required by embedded system manufacturers.

Designers of embedded systems can use PC/104 modules in a variety of ways, including simple module stacks or plugging the modules into application baseboards like macrocomponents in a flexible, component-like design. Recent extensions to PC/104 (including the PCI-enhanced PC/104-*Plus* and the new EBX SBC form-factor standard) add to the design flexibility of such approaches. The good news is that these *non-backplane* embedded computer technologies provide a means to greatly simplify the development of embedded systems for a wide range of applications, resulting in faster project completion, reduced development costs and risks, and improved system features and sophistication thanks to increased focus on the application itself. Ω



Rick Lehrbaum co-founded Ampro in 1983and was the company's Vice President of Engineering until 1991. In his current role as Executive Vice President of Strategic Development, he serves as the company's chief evangelist – developing strategic alliances and communicating its technology vision to customers, strategic partners, and the media.