INTERNATIONAL ATOMIC ENERGY AGENCY
UNITED NATIONS EDUCATIONAL, SCIENTIFIC AND CULTURAL ORGANIZATION
INTERNATIONAL CENTRE FOR THEORETICAL PHYSICS
I.C.T.P., P.O. BOX 586, 34100 TRIESTE, ITALY, CABLE CENTRATOM TRIESTE

H4.SMR. 403/7

FIFTH COLLEGE ON MICROPROCESSORS: TECHNOLOGY AND APPLICATIONS IN PHYSICS

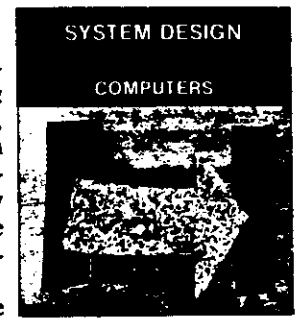
2- 27 October 1989

High-speed 32-bit buses for forward-looking computers

P.L. BORRILL
National Semiconductor Corporation, Santa Clara, U.S.A.

High-speed 32-bit buses for forward-looking computers

Allowing multiprocessing and scalability, these computer backbones offer many choices. . .and some headaches



Buses perform arguably the most important technical function in a computer after the processing unit. Originating simply as traces on a circuit board, they have by now become the stable platform on which to build two or more generations of computers, thus preserving users' investment in the many circuit boards to be plugged into them. They are therefore eminently desirable candidates for standardization.

Designers of computer systems of all sizes require them both to accommodate faster processors as semiconductor technology improves and to employ a standard architecture so that users may mix and match system components at their pleasure. Just to keep pace with current reduced-instruction-set computer (RISC) processors, it is necessary to choose among newer 32-bit buses that significantly outperform the old 16-bit buses. But to anticipate expected next-generation RISC requirements, an equally significant leap in bus performance must be made.

The newest wave of high-performance 32-bit system buses are in fact immensely more capable, but also more difficult to design with. The choices crowd in upon one. Should a system designer go for the performance of Futurebus, or settle for the compatibility of VMEbus? If the designer envisions multiprocessing systems down the road, what must the bus include to provide for that? Furthermore, while older buses were little more than extensions of a given processor's signal pins, their successors offer a multitude of essential new features. Among them are block transfers to RAM, cache coherence, and autoconfiguration: the ability of a bus to poll boards connected to it, identify them, and adjust the software interface accordingly.

Together, these features are capable of supporting the increased performance of current and future RISC processors. They also make the computer systems based on them scalable; in other words, systems of widely varying price and performance can be based on the same bus architecture because that architecture can handle different bit widths—not only 16 and 32, but in some cases (notably the Futurebus) 64, 128, and 256 bits of data.

Existing 32-bit buses include the VMEbus (IEEE P1014), Multibus II (IEEE P1296), and the NuBus (IEEE P1196), which is used in Apple Computer Inc.'s Macintosh II. If those make a second generation of buses, following a first generation of simpler 8-bit and 16-bit system buses, then the third generation is just starting to arrive—made up of far more capable and powerful 32-bit buses. These include Futurebus (IEEE P896), the Sun Microsystems Inc. S-bus, and the Scalable Coherent Interface, or SCI (IEEE P1596). SCI is not actually a bus at all, but an interface that is to allow complete interconnection among all system modules. SCI will define a connector interface that will plug a processor into any interconnection network, be it a crossbar, a ring, or a more exotic network.

Choosing the right bus has never been straightforward. Along with physical attributes such as board size and connector type, designers must evaluate arbitration methods, synchronization protocols, transfer protocols, and the semiconductor technologies available to drive the bus. All of these exert strong mutual influences on one another, to either the detriment or gain of a system's architecture, efficiency, and cost.

Paul L. Borrill National Semiconductor Corp.

Is bigger better?

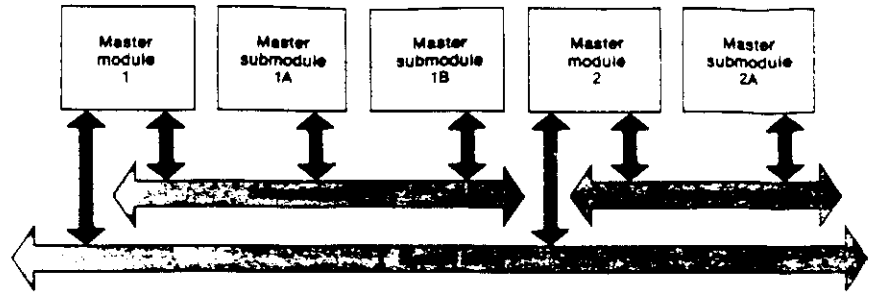
Board size, for example, can have far-reaching implications for the scalability of a bus. Only so much functionality will fit on a small board before spilling over onto others, which must then all be designed to communicate over the bus, inevitably at lower speeds than if the signals could remain on one board. Therefore, although large boards are individually more expensive, they may prove more economical in the context of the overall system design because the bus interface cost is amortized over the greater complement of resources on each board. Smaller boards do tend to be more modular, however, making it easier to mix and match components toward a specific application. Larger boards, on the other hand, may be more suited to higher-performance applications. Witness Sun Microsystems workstations, which contain boards as large as 366.7 by 400 millimeters, whereas IBM PC boards measure no more than 90 by 305 mm.

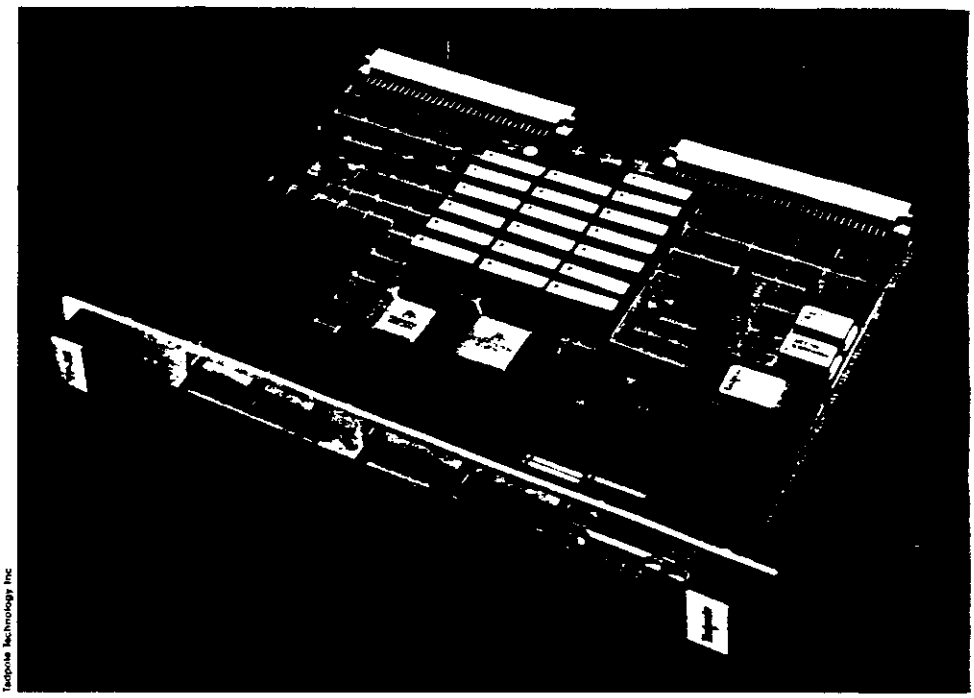
The applications to be served by the bus must also be considered. A memory bus, for example, must optimize communication between the processor and memory. It may allow interleaving, in which successive read requests go to different banks of memory so one bank will be recovering from a previous access while another bank is accessed with the latest request for data. I/O buses, however, tend to be narrower—have lower bit-widths

Defining terms

Arbitration: a protocol for selecting a single master module from a number of equal, competing modules requesting resources.
Autoconfiguration: a mechanism and protocol to allow software to configure system parameters automatically; a method of uniquely identifying each circuit board on a bus, as by geographical addressing, must be in place to achieve this.
Cache: a small intelligent memory close to the processor, which retains copies of recently referenced main memory locations in case they are needed by subsequent instructions.
Cache coherence: a software feature that keeps multiple copies of main memory contents identical, so that any processor that requests data receives the latest version.
Geographical addressing: a method of allowing each module in a computer system to uniquely identify itself by its position along the backplane.
Master: the module that currently controls the bus; in multiprocessor systems, modules take turns at being master.

0018-9235/89/0700-0034$1.00©1989 IEEE





In a multiple-bus system, a subsystem bus allows several smaller modules to behave as a single large virtual module, while maintaining the physical characteristics of more modular smaller board sizes (diagram). It may also, however, allow higher-density boards to be created (photograph). This processor board from one of the leading manufacturers—Tadpole Technology Inc., Waltham, Mass.—contains a Motorola 88000 RISC processor (center of board toward the front), memory (behind the processor), an I/O subsystem, and user-configurable circuitry—essentially an entire system on a board. The board shown uses both the VMEbus and the VSB subsystem bus, but the same system also comes with Multibus II. Boards like this are designed to be plugged into the appropriate backplane bus in concert with other processor and special-function boards, perhaps as many as 20 forming a highly multiprocessing system.

—since most I/O requirements are character oriented. But they must also adapt to the peculiar timing requirements of peripheral devices and manage the interrupts they generate.

Almost all new 32-bit buses use boards that conform to the expandable Eurocard (IEEE 1011) standard. This gives mechanical specifications for a range of standard board sizes, starting at 100 mm in height (growing in 133.35-mm increments) and 160 mm in width (broadening in 60-mm increments). Thus VMEbus cards, at 233.35 by 160 mm, are known as "double Eurocard" height, while Sun's 366.7-by-400-mm boards have a "triple Eurocard" height.

The first version of Futurebus used triple-Eurocard sizes, but the latest revision of the standard also permits double-card sizes. This option makes Futurebus + a likelier recruit for military use; the triple size was thought to be too large to withstand flexing from the shock and vibration of the battleground.

One is not enough

A single general-purpose bus that tries to meet all requirements is often optimal for none. Typically designers resort to two or more buses in order to interconnect system modules cost-effectively [see figure above]. Supercomputers, for instance, often use VMEbus for I/O rather than their proprietary backplane bus.

Designers like to distinguish between physical and virtual module size. The size of the physical module—usually that of the board—is chosen to optimize the modularity of the system. A second bus, often called an extension bus, can then be used to tie together two or more physical modules into a single virtual module having a single interface to the system bus. The LBX II (Local Bus Extension) from Intel Corp. and the VSB (VME Subsystem Bus) from Motorola Inc. are examples.

Local extension buses have the logical characteristics of onboard buses, but the physical characteristics (connector, board size, terminators, and so on) of backplane buses. By requiring an extension bus to carry the additional traffic, designers can add bandwidth to their system without saturating the main system bus. To illustrate, using a subsystem bus to connect a graphics engine to its dedicated RAM rids the main system bus of repetitive high-bandwidth memory transfers of graphic images, so that the overall system performs at a much higher level.

When processor speeds rise to 25 megahertz or more, even short interconnections of just a few centimeters exhibit transmission line behavior that must be dealt with if the system bus is to perform adequately and reliably. Thus the next choice a designer must make is among CMOS, TTL, emitter-coupled logic (ECL), and the newer backplane transceiver logic (BTL), also bipolar, which is rapidly becoming a standard in newer bus designs [see "The bus-driving problem," p. 37].

Other parameters that characterize buses (and consequently optimize them for certain application areas) are their data width, address width, and multiplexing ability [see table]. Data width often depends on the bit width of the processors supported. Processors of 16 bits generally need 16-bit or wider buses; 32-bit processors need 32 bits or more, and so forth.

Designers can avoid inefficient use of bus pins by selecting a bus that multiplexes addresses and data. Nonmultiplexed buses (for example, VMEbus) read data by using the same set of bus lines alternately for addresses and data, thus circumventing access delays in the memory system. Multiplexing utilizes those lines more efficiently, at least for read operations; among others, Multibus II (IEEE P1296) and Futurebus (IEEE P896) multiplex address and data lines.

Performance the great debate

Perhaps the most revered—and yet most widely misunderstood—parameter of a bus is its bandwidth rating. But proponents of different buses continually compare apples to oranges in touting the overall performance of their bus. Some manufacturers quote the theoretical peak bandwidth of a bus operating under unrealistic assumptions—assuming processors and memories far faster than any built to date. Others quote bandwidth levels sustainable only when the bus is part of a single-processor system in which all functions are synchronized with the same clock.

Just as processor benchmarks are elusive metrics, so is the bandwidth of a bus. A valid comparison requires measuring the performance of the final system in a realistic application.

A more useful metric for modern computer buses is obtained by dividing the sustainable bandwidth—measured in megabytes per second—by the number of pins in the connector, giving the transfer-per-second rate per pin. This indicates to a designer the tradeoff between cost and performance.

Numbers alone will not reveal how a bus will perform in a system design. So far, there is no adequate way to simulate in software the performance of a proposed bus design and all components that interact with it. Prototyping is still required.

But designers can look for features in a bus specification that allow them to improve overall system throughput—not to the level of unrealistic benchmarks, but certainly above that obtained by designers who work solely by the book. They can, for instance, create a multiprocessing system that exploits some buses' block-transfer modes, requesting data only in blocks so that transaction overheads can be amortized across multiple bus transfers. Message-passing architectures—including Multibus II and Futurebus—encourage the use of sequential blocks, rather than random single transfers over the bus. Programmers must then package their data into blocks for transfer, but bus traffic is much reduced.

Newer bus designs also enable multiprocessing through the use of cache memory. Whereas caches were traditionally seen just as a way to improve the memory access time, for the new generation of buses they also filter out duplicate memory accesses to reduce bus traffic and dramatically improve the efficiency of bus transfers by utilizing the bus's block transfer mechanism.

Controlling complexity

System designers need more than a heap of chips and a product idea to build a successful system. In hardware as well as software design, complexity may increase exponentially with system size if the system is not carefully structured.

This complexity is often best controlled by hierarchical organization. In hardware, modularity is represented by constructing the system as a set of modules, connected over a bus. Modules connected by one level of bus can be treated as a single module at the next, and the structure can be applied recursively.

As higher levels of integration arise, systems that formerly required multiple backplane buses, connected by bus repeaters, can now be implemented by a handful of processors on each board. The boards themselves are then connected by a single backplane bus, which supplants the older system's inter-repeater bus.

Only the newest bus designs—Futurebus and SCI—are explicitly designed to work with multiple buses connected by repeaters. The bus protocols can maintain cache coherence not only on a single backplane bus, but among different racks of boards attached to different backplanes, cabled together through repeaters. While Futurebus is limited to tens of systems, SCI can scale up to several thousand.

Failing safe

Systems that require very high availability rarely have components that fail less often. Rather, they are designed to degrade gracefully when failures do occur. For example, other processors may assume the workload of a failed processor in multiprocessor systems. But the failed module remains in the system.

Consequently, live insertion and withdrawal of boards is a prerequisite to building fault-tolerant systems. Typical of such facilities are a logical connect and disconnect protocol, to disengage the module from the system, and methods to sequence the power supplies as the boards are plugged in. The only IEEE standard bus to support this facility is Futurebus.

Another perennial problem for systems based on board-level products is the number of jumpers, bit switches, and other paraphernalia that users must manually adjust to configure them. Matching the boards to a system and an application is especially difficult in PCs installed and configured by nontechnical users.

Newer buses, however, let a board uniquely identify itself when the system is powered up, so the system software can configure each board without jumpers or bit switches. The system automatically polls the bus to determine which slots are occupied, then reads a ROM present on each board to learn what resources it carries, its configuration, and even its serial number.

The hottest trend in the computer industry today is that for open systems, reflecting the customers' desires for many competing product sources. The new backplane buses afford strong anchorage for systems built from off-the-shelf boards with varying ratios of price to performance.

Standardization will undoubtedly occur, with or without the influence of a standards organization. There are three principal avenues toward it:

- A dominant manufacturer in a particular market may impose a *de facto* standard on users. One example is the IBM PC and PC AT buses, although IBM's more recent effort—the Micro Channel—has met some resistance, in the form of the rival Extended Industry Standard Architecture (EISA) bus proposed by a consortium of competing PC manufacturers.
- A dominant purchaser may decide to base all its applications on a single bus standard. The U.S. Navy, for example, recently chose to base all future mission-critical computers on Futurebus.
- If competing manufacturers create incompatible standards, try-

Current and future 32-bit system buses

Bus	IEEE standard number	Address width (bits)	Data width (bits)	Timing	Multiplexed protocol?	Maximum quoted performance (Mbytes/s)	Driver technology	Maximum number of connector pins
Earlier 32-bit								
Fastbus	960	32	32	Asynchronous	Yes	165	ECL	130*
VMEbus	P1014	16,32	16,32	Asynchronous	No	40	TTL	96 or 128
NuBus	P1196	32	32	Synchronous	Yes	37.5	TTL	96
Multibus II	P1296	32	16,32	Synchronous	Yes	40	TTL	96
IBM Micro Channel	None	16,24,32	8,16,32	Asynchronous	No	17†	TTL	198
Extended Industry Standard Architecture (EISA)	None	16,24	8,16,32	Synchronous	No	33	TTL	198
New								
Sun Microsystems S-bus	None	32	16,32	Synchronous	No	57	CMOS	96
Futurebus+	P896.1	32,64	32,64, 128,256	Asynchronous (source synchronized)	Yes	400 (32 bits)–3200 (256 bits)	BTL	192 (64 bits)
Scalable Coherent Interface (SCI)	P1596	64	64 (logical), 16 (physical)	Not yet defined	Yes	1000 per node	ECL	Not yet defined

Source: author

*An additional 195-pin auxiliary connector is also defined in the Fastbus standard.

†This is the the quoted maximum for systems that have actually been built and tested.

The bus-driving problem

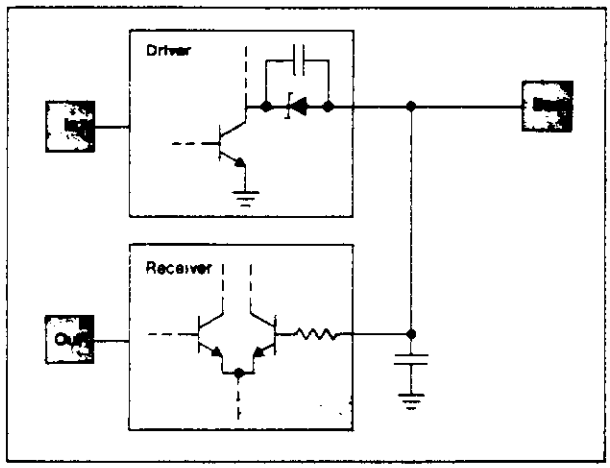
Most computer buses must be treated as transmission lines because bus-driver signals rise and fall faster than the round-trip propagation delay. The capacitive loads presented by connectors, board traces, and particularly transceivers at each slot along the backplane reduce the characteristic impedance of the signal lines, requiring more current to launch the same voltage step down the line.

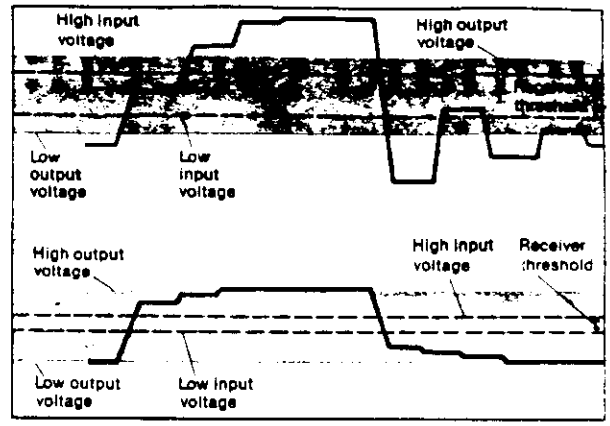
Unfortunately, the typical capacitive loading presented by a module to each slot position on the backplane can be as high as 20 or 25 picofarads. The lion's share is contributed by TTL bus drivers. Its large output capacitance presents TTL with the impossible task of having to drive bus impedances down to approximately 10 ohms. For a signal to clear this receiver threshold region without having to wait out reflections, the transceiver would have to switch a current calculated as the difference between the highest input and lowest output voltages, divided by the loaded impedance.

The effects of high transceiver capacitance—which depresses the backplane impedance—and insufficient drive current to develop the required voltage step over this depressed impedance are known as the bus-driving problem.

CMOS, with its reduced power dissipation, offers an alternative but often results in lower effective bandwidth. The limited availability of drive current with CMOS prevents it from adequately terminating the bus signal lines that make up transmission lines in the system, and so decreases its performance in a system. Termination requires a dc component with a high static current until the signal stops being driven, in conflict with CMOS's low static-state power use.

This leaves either emitter-coupled logic (ECL) or a new transceiver family, known as backplane transceiver logic (BTL), which uses an innovative scheme to reduce the capacitance of the transceivers. It isolates the large capacitance of the open-collector transistor from the backplane with a lower-capacitance series Schottky diode. A low-capacitance receiver input circuit with a precision receiver threshold completes the arrangement [below left].





BTL is rapidly becoming a *de facto* standard for higher-performance systems. It needs only a 5-volt supply, unlike ECL, and dissipates less power as well. BTL transceivers present the signal lines with a much lower capacitance, typically less than 5 pF. If the signal voltage is also made to swing between approximately 1 V and approximately 2 V, the resulting configuration is far better suited to the physics of modern computer backplanes.

In fully loaded backplane signaling viewed at the terminator, the first step on the rising edge of the TTL signal cannot clear the receiver threshold until reflections have been received from the terminators [above, top]. Also, ringing on the falling edge enters the threshold region until reflections abate.

A typical waveform for the initial BTL driver signal step, however, exceeds the receiver threshold without having to wait for reflections [above, bottom]. This is a fundamental requirement for achieving the kind of bandwidths required by higher-performance buses like Futurebus.

All the early 32-bit buses—including VMEbus, Multibus II, and NuBus—use TTL, but newer buses have abandoned it because it cannot drive the backplane with sufficient current. Roughly a third of the latest buses use ECL, and the rest use BTL, which seems likely to dominate the field.

Sun's S-bus expansion bus is the exception: it uses CMOS to reduce the power consumption on the main board and its connected expansion boards. However, the S-bus can do without ECL or BTL only because it has no backplane to drive and is limited to at most three slots. —*P.L.B.*

ing to lock customers into using their products, they may come to see the futility of such competition and ally to compose a common standard. EISA is an example here too.

The resulting standards in each case may be practical, but they also may reflect self-interest. True standards, however, are not only practical but objective and universal. Those developed as open standards from the start—VMEbus, Futurebus, and SCI included—more often approach the optimum solution.

But such success occurs only when the concerns, interests, and experience of all those affected by the standard are taken into account, and it requires all parties to discard the not-invented-here attitude that may afflict engineers. Here, IEEE standards committees have played a major and productive role.

To probe further

An overview of system buses is contained in "A framework for computer design," by W. Kenneth Dawson and Robert W. Dobinson, *IEEE Spectrum*, October 1986, pp. 49–54. *IEEE Micro* contains news about new and revised bus standards; it is available from the IEEE Computer Society, 10662 Los Vaqueros Circle, Los Alamitos, Calif. 90720 (714-821-8380), or the IEEE Service Center, Box 1331, Piscataway, N.J. 08855 (800-678-4333). For further information on IEEE standards activities, contact the Secretary, IEEE Standards Board, also in Piscataway.

About the author

Paul L. Borrill (M) is vice president of standards for the IEEE Computer Society and a member of its board of governors. Since 1988, he has been a staff scientist at National Semiconductor Corp., Santa Clara, Calif. He has worked on designs for buses and multiprocessor systems for 10 years, and has chaired the Multibus II and Futurebus committees. He earned a B.Sc. with honors in physics from the University of Manchester in 1977 and a Ph.D. in physics from University College, London, in 1986. ◆