

# IN-SITU EMULATION PACES NEW MICROS

Evolutionary development improves accuracy of simulation by moving the target processor closer to the application

by David McCracken

Support for microprocessor application development must grow rapidly to keep up with advances in very large scale integration technology. The increasing speed and complexity of semiconductor devices have especially taxed in-circuit emulation, which strives to accurately simulate the functional and electrical characteristics of a microprocessor. Present in-circuit systems position the target microprocessor as much as several feet away from the application circuit. Buffer devices separate it from the application electrically. The physical and electrical separation of microprocessor and application circuitry makes conventional in-circuit emulation a less adequate way than ever to simulate semiconductor devices.

Following the current trend toward moving the target microprocessor closer to the application, in-situ emulation meets the requirements of very large scale integra-

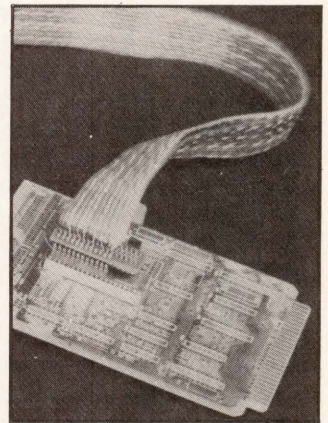
*David McCracken is a microprocessor applications consultant and principal engineer at Thera Institute, PO Box 1090, Aptos, CA 95003. His design work includes automated laboratory equipment for biochemical research, multiprocessor controlled test environments, and miniature computerized field data acquisition systems. He was also an industrial control systems consultant. Mr McCracken studied civil engineering at the University of California, Berkeley.*

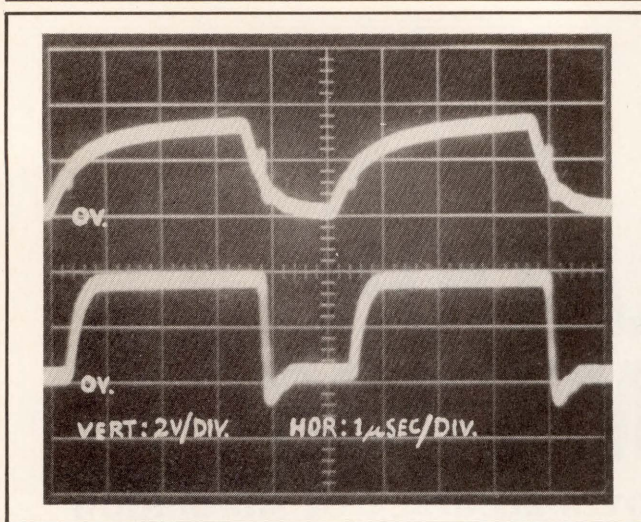
tion (VLSI) simulation by placing the microprocessor directly in the application physically and, as much as possible, electrically. This often requires hybrid microcircuits to reduce the size of buffer and control circuitry, producing an

emulation plug that can comfortably fit into most applications. However, the concept is initially demonstrated with a Z8™ in-situ emulator that has been built using standard integrated circuits (ICs), as only two ICs are needed for cable driving buffers in the emulation plug. The Z8 resides physically in the application with no buffering, producing complete electrical transparency.

## Challenges of in-circuit emulation

In-circuit emulation (ICE) basically time-multiplexes a target device between development and application environments. The target microprocessor communicates with a host computer while in the development environment, performing program development functions specified by the operator, such as beginning execution at a particular address or sending breakpoint information for use in a display assembled by the host. While in the application environment, the target executes programs that are being developed, interacting only with the application hardware.





**Fig 1** Effects of emulation. Oscilloscope trace at top shows z8 output port driving highly capacitive load. Bottom trace shows typical ICE circuit driving same load through TTL buffer and short length of twisted pair. Its undershoot below 0 V can damage circuits

Transparency of emulation, or the degree to which the target microprocessor in the development environment resembles an unattached application, must be measured in both functional and electrical terms. Functional transparency has generally been achieved to a high degree through various implementations of memory bank switching so that development programs reside only in the target's virtual address space and need not impinge on actual potential facilities.<sup>1</sup> Electrical transparency, however, has generally not been achieved because target microprocessors have been separated from the application both physically [up to several feet ( $\pm 1$  m)] and electrically by buffer devices.<sup>2</sup> The resulting signal delays and inaccurate drive/load simulations have not been an overwhelming problem in most situations because, until recently, most microprocessors operated at speeds low enough so that the signal delays became unimportant, and bus-oriented microprocessors usually are buffered in the application by devices similar to those used in the ICE circuitry.

Now, VLSI technology challenges traditional ICE electrical non-transparency in two major respects. One is that the operating speeds of new microprocessors are entering regions where transistor-transistor logic (TTL) buffer delays become significant, especially when combined with signal delays through the connecting ribbon

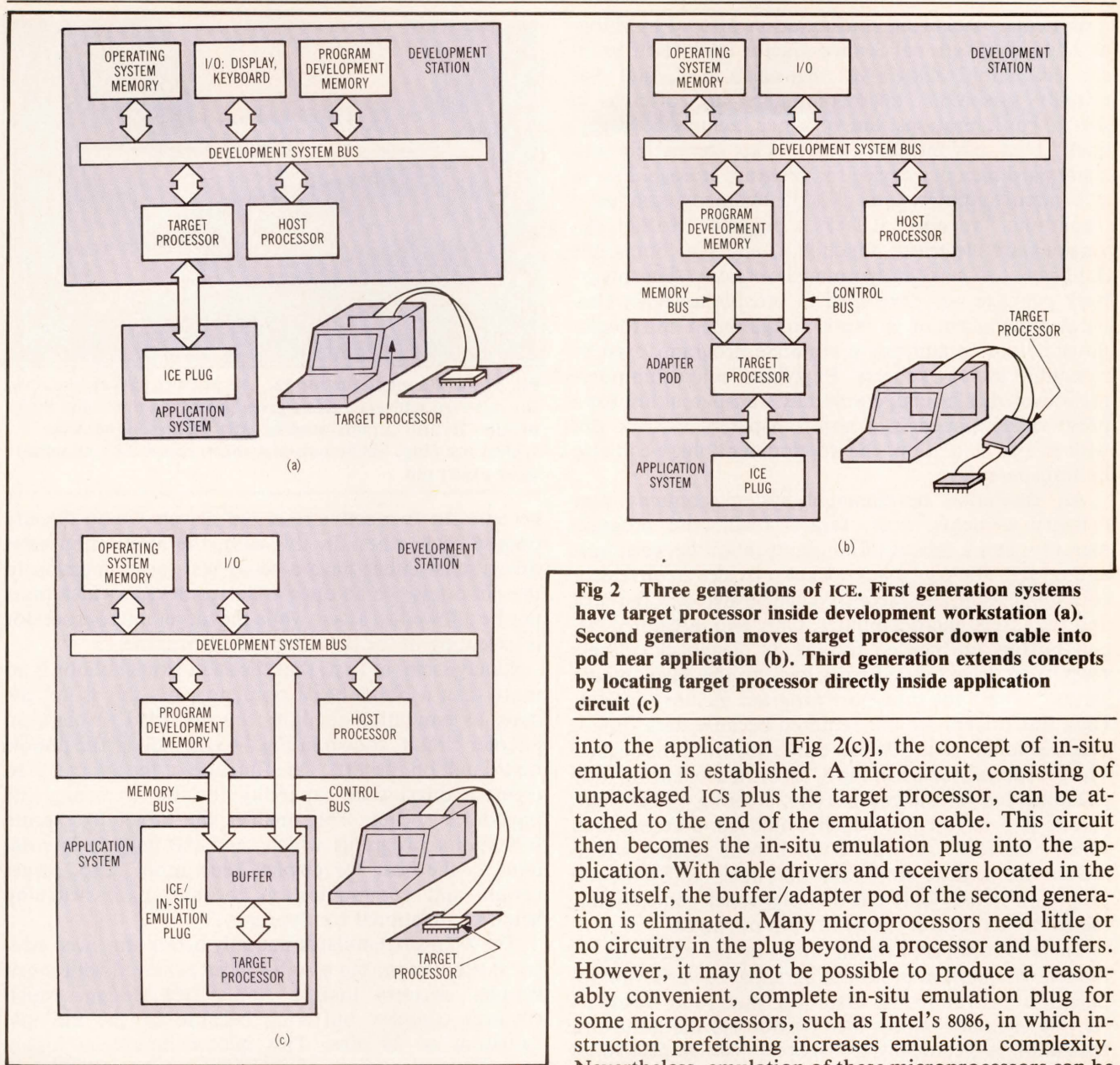
cables. The other is the proliferation of increasingly complex single-chip microcomputers for which electrical, as well as functional, simulation is complicated. Metal oxide semiconductor I/O ports are particularly difficult to simulate using remote, cable-driving TTL devices. With more specialized functions, such as analog to digital conversion, moving into increasingly dense microcomputers, present emulation techniques may not be able to provide adequate simulation.

Depending on the electrical characteristics of a particular application, the accuracy of present ICE simulation varies. The oscilloscope display in Fig 1 demonstrates several of the problems. The top trace is created by a z8 output port driving a  $0.01\text{-}\mu\text{F}$  capacitive load directly, as it would in an actual application. The bottom trace is produced by a typical ICE circuit with a TTL buffer driving a 1.5-ft (46-cm) twisted pair into the same application environment. Beyond obvious differences in waveforms, a particularly troublesome phenomenon is the voltage undershoot caused by impedance mismatches between buffer, cable, and application circuitry. In some cases, the signal excursion below zero volts could damage devices in the application, but this would not happen outside the development system. Impedance mismatch problems will increase as ICE speeds rise to accommodate faster VLSI microprocessor targets. Possibly the only reasonable solution is the use of line drivers and receivers at both ends of the cable so that whether or not the target microprocessor is located *in situ*, some active circuitry must still be located in the emulation plug.

#### Evolution of in-circuit emulation

The problems posed by VLSI are not new to ICE. Increasing microprocessor speeds, in particular, have prompted steady change in emulation methods. In first generation hardware [Fig 2(a)], the target processor was located inside the development station, with a relatively short ribbon cable leading to the application microprocessor socket. Originally, the target processor accessed program memory (as well as other development facilities) through the main system bus shared with a host processor. The host moved programs that were to be tested into system memory under operator direction. It then relinquished bus control to allow target access to the memory. Unfortunately, this memory was often not fast enough to test program execution in real time, especially when faster microprocessors became available. Also, in some designs, the host lost much of its system control while emulation proceeded, and the target microprocessor, having control of the main bus, could cause system crashes by improperly accessing program development and control facilities. These problems were mostly solved by using a 2-port memory, accessed separately by host and target, and dedicated to emulation.

As microprocessors continued to increase in speed, and as development system users began to demand more flexible hardware, ICE designers moved the target processor into a buffer/adaptor pod located in the cable between the development station and application system [Fig 2(b)]. This move required much more complex ICE



**Fig 2 Three generations of ICE. First generation systems have target processor inside development workstation (a). Second generation moves target processor down cable into pod near application (b). Third generation extends concepts by locating target processor directly inside application circuit (c)**

into the application [Fig 2(c)], the concept of in-situ emulation is established. A microcircuit, consisting of unpackaged ICs plus the target processor, can be attached to the end of the emulation cable. This circuit then becomes the in-situ emulation plug into the application. With cable drivers and receivers located in the plug itself, the buffer/adapter pod of the second generation is eliminated. Many microprocessors need little or no circuitry in the plug beyond a processor and buffers. However, it may not be possible to produce a reasonably convenient, complete in-situ emulation plug for some microprocessors, such as Intel's 8086, in which instruction prefetching increases emulation complexity. Nevertheless, emulation of these microprocessors can be improved by active line drivers and receivers in the plug, even if the target must still be in a remote pod.

#### In-situ emulator

Because electrical transparency is especially desirable in single-chip microcomputer emulation, it will be used as the area to demonstrate the in-situ concept. In addition,

circuitry, with both memory and target control signals delivered to the buffer/adapter from the host through another ribbon cable. However, the relatively small pod could be placed close to the application, reducing the effects of long cables on signal delay and impedance mismatches (even one extra foot of cable can severely aggravate problems caused by impedance mismatches). This second generation hardware has not dealt with electrical transparency in that buffers are still required between the target processor and application. In addition, application circuitry still must drive both the cable to the pod and the standard device loads within the pod, instead of the actual microprocessor.

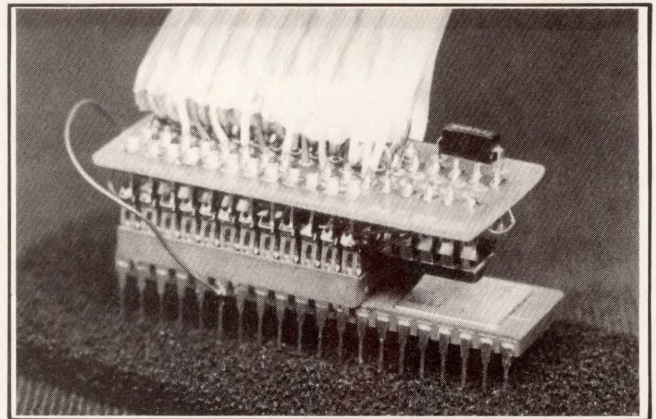
To meet the challenges posed by VLSI in the form of greater speeds and unique functional and electrical characteristics of microcomputers, a third generation emulation can be extrapolated from the existing trend toward moving the target processor closer to the application. As the target processor finally moves directly

a particular packaging configuration offered by Zilog, Mostek, and National Semiconductor lends itself to in-situ emulation. Zilog's Z8<sup>3</sup>, Mostek's 3870<sup>4</sup>, and National's 8048/49/50<sup>5</sup> microcomputers are offered in several configurations, one of which includes a "piggyback" read only memory (ROM) socket compatible with a standard memory device for program storage. For its 8022 microcomputer, Intel calls the analogous product a "bond-out" version.<sup>6</sup> It lacks a piggyback socket. All versions are electrical, functional, and pin-compatible duplicates of mask-programmed versions. The piggyback package was developed to provide an exact electrical simulation of a mask-programmed microcomputer before committing a tentative program to an irreversible factory process. Piggyback microcomputers are intended as a final, electrically transparent test for a program developed in a non-transparent system. But with in-situ emulation, the development phase can also be transparent.\*

An emulation development system contains four primary elements: host, target, read/write program memory, and a means of communication between host and target. The piggyback socket provides a convenient communication and memory channel between the host and the target microcomputer. Only two integrated circuit buffers are needed for the Z8 piggyback address outputs to drive twisted pairs back to the development station, where the dual-port program memory resides. Data line drivers are not required because data flow is unidirectional, as the Z8 can only read from the socket. The prototype, shown in Fig 3, is constructed using fully packaged ICs for the two buffers, creating a fairly high profile [0.8 in (2 cm)], which would be undesirable in some situations. A production hybrid version would reduce this to little more than that achieved by the ICE plugs now in use. Only one non-piggyback connection is required—to control the RESET pin of the Z8.

The RESET pin connection brings up for ICE designers the critical question of how to anticipate a variety of application environments, assuming that total transparency is virtually impossible. Obviously, the Z8 in-situ emulation is electrically transparent with respect to all pin functions other than RESET. But the RESET function, as well, can be made nearly transparent by anticipating the most practical application environments. This is not difficult, as the application RESET circuitry will take only two different forms. One is a standalone microprocessor configuration in which the Z8 is allowed to reset itself on power-up, using an external capacitor for time delay (all single-chip microcomputers have this function on the chip to reduce low-end product cost). The other is an external driver to synchronize with other devices in a multiprocessing configuration.

The emulator can anticipate the self-reset circuit simply by using a high impedance 3-state driver to control the pin. Then, during emulation, when the emulator controller need not exercise its reset control, the driver is placed in its high impedance state, allowing the application reset to function normally. In the second configuration—which is much less common—the application circuitry must be designed to accommodate the emulator



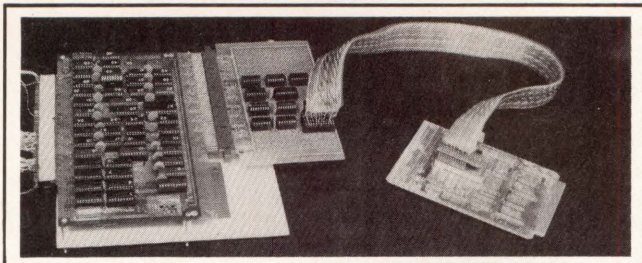
**Fig 3 In-situ emulator for Z8. Use of fully packaged chips for onboard address buffers gives prototype unusually high profile. Hybrid circuits need be only slightly higher than typical ICE plug. Single non-piggyback connection attaches to Z8 RESET pin**

because the two active reset pin drivers would directly oppose each other. In this case, the application reset driver must either have a series resistor connection to the pin or else be an open collector device, which must not be allowed to assert while the target microprocessor is executing in the development environment.

It may seem, at first, that the Z8 in-situ emulator is no more than a simple ROM simulator, but this is not the case, as emulation facilities are provided through an adapter circuit, shown in Fig 4 (right side of the double board unit on the left). One function of the adapter is to create a quasi-write capability for the Z8 through its piggyback socket. This enables the target to execute breakpoint programs which send the target's internal data to the host for display generation. Also, target restart and communication through bank-switching achieves functional transparency.<sup>1</sup>

The Z8 in-situ emulator not only offers improved electrical simulation, but also has been achieved with much simpler circuitry than a similar ICE design, which requires complex buffering because of the multiple functions of Z8 pins. This microcomputer is highly reconfigurable; most of its pins can be individually changed from input to output under software control. One final achievement is that this circuitry is compatible with Mostek's 3870 microcomputer, which provides the same pinout in its piggyback ROM socket. Thus, with no hardware changes other than connecting the cable to a different target processor, two different emulators can be produced. This in-situ emulator has one deficiency not found in second generation ICE. The Z8 can be reconfigured for external memory access through pins which would otherwise be I/O ports. With full control over all target pins, the ICE can provide substitutes for nonexistent application memory, which the in-situ emulator cannot. A complete development system could offer both forms of emulation.<sup>6</sup>

\*Zilog has introduced a universal peripheral controller microcomputer with a piggyback package, the Z8090-UPC; Phillip's German subsidiary, Valvo GmbH, has introduced a piggyback series similar to the 8048 called the MAB8400.



**Fig 4 Prototype hardware for in-situ emulation. Complete system has emulation plug to replace application microprocessor (at right). Adapter circuit allows target processor to access dual port memory and channel for communicating with host**

### Summary

As accurate VLSI simulation demands increasing ICE performance, the evolutionary development of ICE design suggests placing the target processor directly in the application, creating in-situ emulation. This can provide the improved speed and signal quality required by faster processors and more accurate electrical simulation for single-chip microcomputers. Even where the complexity of microprocessors precludes the ideal in-situ form, active circuitry in the emulation plug will produce valuable improvements.

### References

1. D. H. McCracken, "Hybrid Tool for Universal Microprocessor Development," *Computer Design*, Apr 1980, pp 119-126
2. "Evolution or Revolution to Next Generation Development Systems," 1980 Electro Professional Program, Session 23; "But How Does Your Microprocessor Development System Develop My System?" 1980 Electro Professional Program, Session 27
3. "Z8-03 MPE Microcomputer Protopack Emulator Product Preview," Zilog, Inc, Cupertino, Calif, Sept 1979
4. "3870 Single Chip Micro Family MK3870 and MK38P70," Mostek Corp, Carrollton, Tex
5. "INS 87P50 Piggyback Microcomputer," Preliminary Product Description, National Semiconductor Corp, Santa Clara, Calif
6. "ICE-22 Unit Cuts Microcontroller Design Risks," *Preview*, Intel Corp, Santa Clara, Calif, Nov/Dec 1980, p 13

*Please rate the value of this article to you by circling the appropriate number in the "Editorial Score Box" on the Inquiry Card.*

High 701

Average 702

Low 703

# Data Cable Encyclopedia

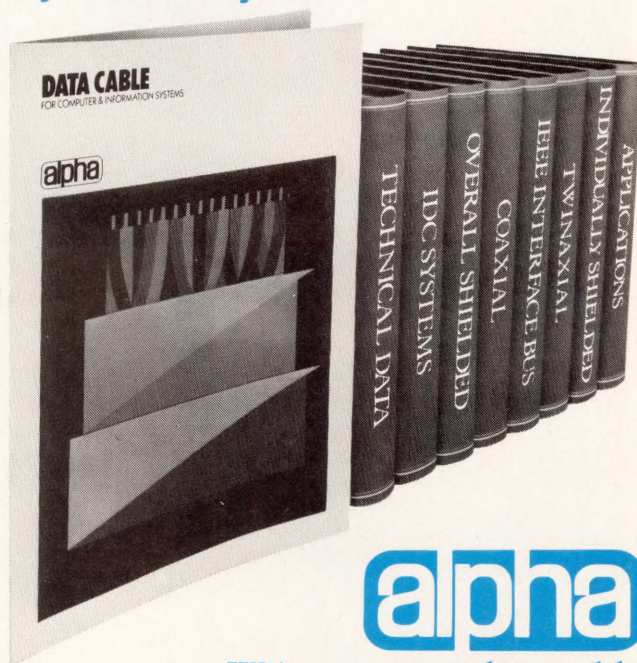
Your comprehensive guide to quality data cable...  
manufactured by Alpha and available everywhere. Rely on it.

If you buy, design, or install computer and information systems, the Alpha Data Cable Catalog should be on your desk right now.

Whether you're involved in communications interface, network cabling, original equipment design, data entry and display, or computer peripheral interconnects you'll find all the products you need—presented clearly, described fully, and backed by solid technical information.

Within its 60 pages are complete specs on coaxial and twinaxial cable, in-house and remote transmission cable, shielded and unshielded cable, IEEE interface bus cable, IDC systems and more. All items are manufactured to meet necessary certifications, and all are in stock. Which means your local Alpha distributor can put the cable in your hands wherever and whenever you need it.

Send for your Alpha data cable "encyclopedia" today. Just contact Alpha. In the U.S.A., it's Alpha Wire Corporation, P.O. Box 711, Elizabeth, NJ, 07207, (201) 925-8000. In Europe, it's Alpha Wire Limited, Alpha House, N. Feltham Trading Estate, N. Feltham, Middlesex, UK (01) 751-0261. Or, contact your local Alpha distributor. He's not far away.



**alpha**

We've got your data cable