

# Hardware debugging for on-line psychologists

ROBERT S. McLEAN

*Department of Computer Applications  
The Ontario Institute for Studies in Education  
252 Bloor Street West, Toronto, Ontario, Canada M5S 1V6*

As computers become more prevalent as laboratory equipment, psychologists and graduate assistants will be more likely to be involved with interfacing the experiment to the computer. This paper presents some general ideas about the design and construction of digital computer interfaces and ways to facilitate their check-out and integration into the computer system. The suggestions center around the identification of functional modules within the interface or device from the beginning of the design process. These modules are preserved in the design and construction and are tested separately where possible. Specification of the interface signals between these functional units speeds debugging of a new device and facilitates maintenance of the device at a later date if adequately documented. It is important to ensure that these signals are readily available through test points and/or indicators. The paper suggests minimal equipment necessary to construct and debug an interface. It is suggested that an interactive construction procedure may be the most successful. The device is constructed in stages, with each part being verified as it is built. A simple interface is suggested for the beginner which gives him practice and a useful debugging aid as well.

In an ideal world, the computer-user psychologist need not concern himself with the hardware either of his computer or of the interface between his computer and his experiments. The psychologist uses the on-line computer as a tool, leaving the tool designing to specialists. His computer is repaired by someone who specializes in maintenance. He has available people and funds to let someone else worry about connecting his experimental equipment to the computer either by using standard interfaces or by such special arrangements as might be necessary.

The world is seldom ideal; the psychologist or his graduate assistant is often involved with the process of connecting—"interfacing"—the experiment to the computer. This happens for a number of reasons—often economics, but sometimes because these people are the ones that understand the problem and can spend the time involved.

Whatever the reason, it is often easy to get help with the design phase of the interface. Assistance is available from numerous books and manufacturer's literature as well as from sales persons for both computers and logic modules. A good overview of the popular TTL integrated circuits is given by Morris and Miller (1971), including sample designs of counters, decoders, arithmetic circuits, shift registers, and other circuits. A shorter discussion of IC use can be found with manufacturer's data sheets in collections such as those of

Texas Instruments (1971) or Motorola (1971). Logic board manufacturers provide introductions to interfacing in their catalogs and explanatory materials for their logic lines (e.g., Hughes, 1969; Digital, 1970).

Logic design of a special interface or device is often given in technical articles in journals; in this case, the psychologist may be tempted to try to produce his own copy of the device from the schematic.

Wiring may be done fairly easily, or an assistant may be found to do the job. Sometimes vendor's wire-wrapping services may be employed. Actually building the interface or device from the drawings is not difficult, although it may be tedious.

This paper discusses some considerations of design and construction of devices that will make the check-out and debugging processes easier. Here there is less help available from salesmen or casual help; it often takes someone who knows both computer and experiment.

Given that the psychologist or assistant is going to get involved, there are some steps he can take to assure that he minimizes the difficulty of his task. Most of the suggestions here will be familiar to anyone who has done interfacing before, but may be useful to the person who is a beginner.

## DEBUGGING STARTS WITH DESIGN

Just as programs are easier to get running if they are well thought out during the design stage, so hardware can be designed to facilitate debugging and test. Perhaps the single most important design key to easy check-out and performance is the notion of modularity. This concept can be applied at many levels of circuitry and packaging, yielding circuits that are separable into clearly defined functions.

### Modularity of Circuits

During the design phase, the goal should be to provide clean separation of functional parts of the device. The input and output conditions of each such subunit should be clearly defined. Ideally, any subunit should be capable of test in isolation from the rest of the device. If this is not practical because of the number of interface connections with surrounding units, separate testing on the conceptual level should still be possible. One way to accomplish this is to involve two or more people in the design effort; each should be able to understand the other's circuits with a simple explanation. An even better idea is to commit the description to writing at this point. This serves to demonstrate the clearness of the thinking that went into the design and serves as a basis for maintenance documentation for the completed device.

Examples of circuit modularity in interfaces can be

given. Consider a typical interface for a Teletype terminal. The data path from the unit through the serial to parallel converter should be clearly separated from the addressing circuits for decoding the I/O commands from the computer. Less obvious is the need for separation of the gating structures that transfer data to or from the I/O bus of the computer, although they can be usefully delimited also. Any special character recognition circuits and interrupt facilities should be separated in the design. Wherever possible, optional circuits should be capable of exclusion from the circuit, both electrically and mechanically. Thus, it should be possible to run the Teletype interface without the interrupt or special character circuits.

Indicators can be placed on a few critical signals to indicate visually the state of functional modules. The most useful signals usually originate in the control section of the device, such as interrupt request, status, transfer direction, or busy flip-flops. It is rarely necessary to use indicators on data registers, since data usually move too fast to verify visually. When necessary, a scope and signal source can be used to verify data circuits. The primary purpose of indicators should be to show major conditions of the interface.

Similarly, manual controls should be provided where suitable to circumvent certain hardware functions. A common example is the write protect switch on tape and disk controllers; interrupt disable switches are also common.

Care should be used in designing clocking circuits for interfaces. Delay circuits should be used only when absolutely necessary and then should have a large margin for timing error allowable before errors occur in the data. Parallel delays should be avoided since errors can cause timing race conditions.

Occasionally it is possible to design test circuits into a device or interface with little additional effort. Where feasible, these circuits will help both original check-out and routine diagnostics. Examples include the facility to read contents of output buffers and control flip-flops into the computer or to force input registers into particular data patterns. In some applications, simple plugs can be used to return output signals along similar input lines, allowing the computer to verify the data path in both directions.

### **Modularity in Construction**

Once the circuits are electrically and functionally divided into logical groups and construction is to start, consideration should be given to preserving these modular boundaries in the hardware implementation. Whether this is possible often depends upon the type of logic building blocks being used and the complexity of the circuit. If the circuit is being assembled from small units such as individual integrated circuits, it should be possible to avoid the use of one circuit package (one IC) in more than one functional module. If the ICs are already packaged on circuit cards, economics may

dictate the use of one card in several modules.

The mounting arrangements for the logic packages can facilitate debugging by making changes easy to accomplish. Wire wrapping, instead of soldering, seems to be nearly universal for connecting present-day logic packages and is recommended. Along with its inherent good mechanical and electrical connections, it ensures that the logic packages will be mounted in sockets (resist any temptation to wrap directly to ICs). This facilitates removal of neighboring logic functions to isolate particular circuits of interest. In the rare instance of component failure, it makes replacement easier.

There are several options available in the selection of logic packages. The main ones include use of ICs direct from a supplier or the use of ICs mounted in groups on logic cards. In the latter case, the packaging will be selected by the manufacturer and may have several advantages in terms of general mechanical and electrical compatibility with the computer being used. For the constructor electing to start with ICs, there are several options and advantages. One of the advantages of this approach is that it is often possible to obtain substantial portions of an interface circuit already in modular form through medium-scale integration techniques. Thus, it is possible to buy a chip that contains the entire shift register for the Teletype interface example rather than constructing it from flip-flops and gates. In general these advantages of MSI should be employed where available. It is even possible to obtain the entire receiver or transmitter for the example, complete with buffer register, on a single chip, although this may not be desirable for other complicating reasons such as power supplies required.

When using ICs directly, it is possible to use special wire-wrap panels that consist of a single circuit plane thus making all connections available on the same surface rather than having some hidden on circuit board; on the other side of the mounting block. An alternative available from some manufacturers of logic boards, is a board containing wire-wrap sockets for custom interconnection; it can be connected to the same mounting plane as the logic modules supplied by the company. In this way, the best mixture of the two approaches to packaging can sometimes be obtained.

### **Interchangeability**

Other things being equal, it is desirable to use a relatively small number of different types of components, all of which should be common, readily available types. This will facilitate future repairs and ease of maintaining spares inventory. It also makes possible substitution of suspected defective components during the debugging stages even without spares.

Logic families (TTL, RTL, DTL) should not be mixed due to differing power and logic levels and speed characteristics.

### Accessibility

Access to signals within the interface is important for debugging. When a single-plane wire-wrap board is used, all signals are presented and can be monitored with a scope. A similar situation exists in a slightly more limited extent when using prepared logic boards. When the combination of wire-wrap boards and printed circuit boards are used, not all of the interconnections on the wire-wrap board are available on the backplane, so advance planning of the signals brought out to the backplane will facilitate debugging, as noted below.

Consideration should be given to the mechanical location and ease of access to components for substitution of spares and ease of making connections. If the logic used is such that extender boards can be employed to move the logic boards out from the rest for testing in the circuit, care should be exercised in placement not only of the backplane mounting, but also of the cables that interconnect the device with other parts of the system. It is possible to position cables in such a way as to create a jungle in which to work later on. When wire wrapping, wires should be at the same level (near the plane or further out on the pin) at both ends; this insures that no more than three wires need be removed to change one wire.

The power source for the device should provide sufficient current within the variations specified by the manufacturer of the logic. Where power is to be drawn from other equipment, careful checking of the power specifications is in order. Grounding and power bypass capacitors are quite important in high-speed logic systems, particularly TTL. Morris and Miller (1971), as well as various logic data books (e.g., Motorola, 1971), contain explicit suggestions in this regard.

## ACTUAL CHECK-OUT OF THE DEVICE

### Bench Check

When the device is wired, considerable debugging can be accomplished before approaching the computer. This is an important strategy when the computer is in regular use or computer time is otherwise expensive. It is also useful in simplifying early stages of check-out. One can devise simple checks of small portions of the circuit that will verify the operation of much of the circuitry. This confirms much of the design and provides some confidence in the accuracy of the wiring. A few basic pieces of equipment will be helpful for this check-out; they are described below.

### On-Line Check-Out

Regardless of the care that goes into building a device, very few devices of any complexity work the first time they are connected to the computer. It may be just as well not to try the device all at once, but rather to check each function, systematically, one by one. Start with the simple ones, such as decoding of the I/O device address from the computer. With a scope, it should be easy to verify that only legal device codes and their associated

I/O pulses get through the gating structures. Given that successfully finished, the data paths controlled by those I/O commands can be checked. jumpers can be used to force certain "data" conditions for checking input commands and the effects of output commands can be observed on a scope.

This process is facilitated by the inclusion of test points in the wiring of the device. By deciding what signals are indicative of correctly functioning modules, potential bugs can be located readily. By selecting these test points in a formal manner and documenting their existence, one can provide valuable help to the person who is later faced with a malfunctioning device.

### Equipment

The specific equipment needed by the E who is about to begin interfacing will depend on the extent to which he is going to get involved. For casual interfacing in the university setting, most of the following equipment might be borrowable from the electrical engineering or computer science departments. This list represents basic tools for interfacing and development of special devices.

#### *An Oscilloscope*

The oscilloscope should preferably be dual trace, capable of displaying 100-nsec pulses. Not only must the time base have this capability, but on plug-in units it is important to check the vertical amplifier's bandwidth. A storage scope is sometimes useful in observing signals that have a very low repetition rate.

#### *Wire-Wrapping Tool and Unwrapping Tool*

The wrapping tool should be at least the hand-operated gun variety, although the pencil types are usable for minor corrections while debugging. The proper-sized bit (usually No. 30 gauge) must be obtained. Gardner-Denver has a monopoly on the tools for this technique, although most logic vendors distribute for them.

#### *An IC Insertion Tool*

The IC insertion tool, a simple and inexpensive plastic device, is very useful for those working with individual ICs.

#### *Signal and Power Source*

If much construction of logic circuits and interfaces is contemplated, the laboratory should be equipped with a few additional devices to facilitate check-out on the bench. A power supply suitable for providing the logic power is necessary to operate most interfaces away from the computer. A signal generator capable of providing fairly high-frequency signals for input to portions of a circuit is quite useful. This can be constructed fairly easily and need not cost more than a few dollars in parts. A simple one is illustrated in Fig. 1. Care should be used to insure that signals are inserted into input circuits only; do not apply this signal in parallel with another

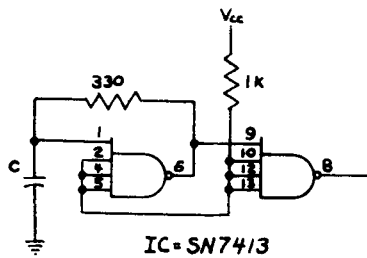


Fig. 1. Signal source. Select C for frequency desired. Vcc (5 V) can usually be obtained from equipment under test.

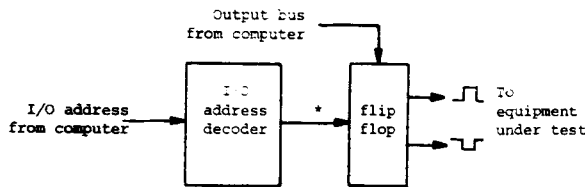


Fig. 2. Signal generator using computer program. The I/O pulse that sets the flip-flop (\*) may be useful for triggering scope.

TTL circuit output, since damaging currents could develop.

If the computer is available for check-out use, particularly if it is time-shared with its I/O bus or relay buffer available in the lab, simple programs can be written to simulate various signal generator outputs for checking digital circuits.

### Practice Interface

The best way to learn about a particular computer

and interfacing to it is to build an interface. Perhaps a suitable first project for anyone starting into interfacing is to build a simple interface that will allow the computer to serve as the signal generator for debugging further interfaces. Such a circuit can consist of the simplest examples of an I/O addressing decoder and a data register. This circuit is outlined in Fig. 2. In solving several small problems associated with its construction, the E will obtain useful insights into the operation of custom interfaces as well as a great deal of confidence. Additionally, the device will be useful.

### Stepwise Construction

Perhaps the best way to proceed with construction of a device is to build the functional blocks of the interface one at a time, checking the behavior of each as they are constructed. The interactive procedure allows for unforeseen errors in design and layout, since each modular part is verified as it is constructed.

This suggestion is very important when the device being constructed is composed of many identical circuits, as is often the case in parallel data transfers. In this case, one circuit (e.g., for 1 bit of data) should be designed, wired, and tested before constructing the remaining bits.

### REFERENCES

- Digital Equipment. *Small computer handbook*. Maynard, Mass: Digital Equipment Corp., 1970.
- Hughes, J. L. *Computer lab workbook*. Maynard, Mass: Digital Equipment Corp., 1969.
- Morris, R. L., & Miller, J. R. *Designing with TTL integrated circuits*. New York: McGraw-Hill, 1971.
- Motorola. *TTL integrated circuits data book*. Phoenix, Ariz: Motorola Semiconductor Products, 1971.
- Texas Instruments. *The integrated circuits catalog for design engineers*. Dallas, Tex: Texas Instruments, 1971.