Behavior Research Methods & Instrumentation
1974, Vol. 6, No. 2, 171-173

The BUS-ALL: An inexpensive set of interface for the SKED system

F. E. BUTLER
University of Iowa, Iowa City, Iowa 52242

A cost-effective relay-transistor interface has been developed for connecting the minicomputer with the experimental environment. The interface contains a 10-msec clock, input (response) channels, and output (stimulus channels.

The SKED software has been in existence for the last 6 years (Snapper, Knapp, Kushner, & Kadden, 1967). The software has been continuously upgraded, and it has used several types of interfaces (Snapper & Walker, 1971; Walker & Snapper, 1971; Snapper & Kadden, 1973). The new software has eliminated previous restrictions and has economized the processing time. The interface has not, until now, been made cost effective.

The present SKED software requires a clock that causes an interrupt every 10 msec, at which time input and output functions are performed. This is a change from the previous versions of SKED that required the responses to cause an interrupt.

THE NOTION OF INTERFACE

An interface is all the hardware that is needed to transform real-world events to and from computer events. At the speed of a computer, a switch closure is not a discrete all-or-none event; rather, the switch bounces. This can cause the computer to read many closures of the switch when there was only one closure. The switch closure must be filtered to eliminate switch bounce.

The power-driving capability of a computer is limited. A computer is not capable of activating a dipper by itself. The interface must contain devices, such as relays, capable of driving several amps.

The computer must know when it is time to process the information that comes in and the information that the computer puts out. This is done by means of an interrupt. Every 10 msec, an interrupt occurs. An interrupt can be caused by the clock, the Teletype, the high-speed reader, or the high-speed punch. When an interrupt has been caused by a device, a flip-flop is set. The computer decides which device caused the interrupt by means of the skip line. If the device is active, then it is serviced and the appropriate functions are performed.

The PDP-8 transfers all programmed interrupts through the 12-bit accumulator. There is an 8-bit device code (BMB 3 through BMB 8), which allows the computer to service 64 12-bit interfaces. There are three more bits that are used along with the device code, namely, the three IOP pulses. The IOP pulses are issued only on input or output command from the computer. When the IOP pulses along with the device code are true, the interface is serviced.

CONSIDERATIONS PROMPTING ANOTHER INTERFACE

There are several ways of interfacing the computer which will be considered. Each method has limitations.

One may interface the PDP-8 using the "systems" approach. This consists of purchasing a system from a company, such as the UDC from Digital Equipment Corporation. This option is close to a turn-key system. The cost of such a system is prohibitive for most.

A laboratory engineer can design and build the interface, but many laboratories are without such technical help. The psychologist can take the time (or have a graduate student take the time) to learn about the idiosyncrasies of the Flip-Chips offered by DEC. After the logic and the polarities are worked out, the interface may be designed and built.

In laboratories equipped with a PDP-8/e, -8/f, or -8/m, DR8 input and output cards may be used. This is a fine system, provided the number of inputs and outputs is small. If the number is large, the size of the cables entering the computer becomes a problem. (This method cannot be used with a PDP-8/L.)

Any new interface should meet certain criteria. The E should be able to interact with the interface at the computer site. This necessitates lights and switches on the interface. The interface should be as turn-key as possible. The interface should be isolated from the laboratory as much as possible. The use of relays on the inputs and outputs allows a high degree of isolation. The interface should be cost effective.

The interface presented here meets these criteria. It consists of an input, an output, and a clock card. The input and and output cards have the logic to handle 36 bits. Relays, with the associated switches and lights, are added in blocks of 12. The appropriate device code is

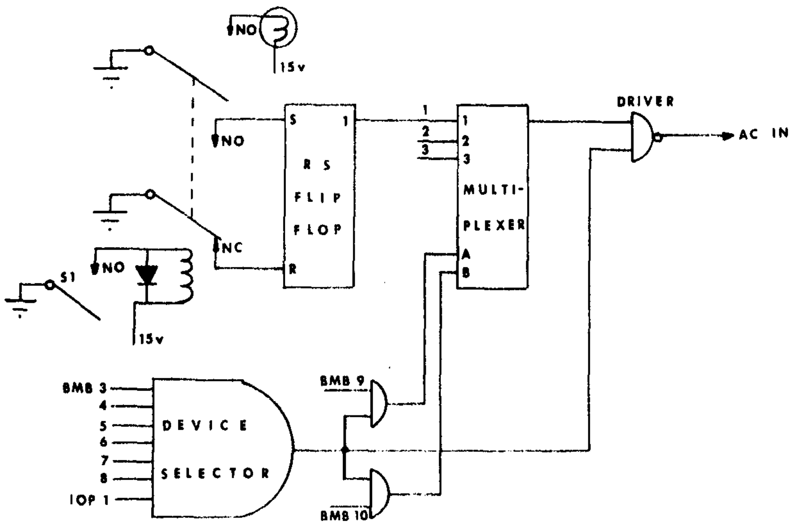

Fig. 1. Simplified diagram of input interface.

wired on the input, output, and clock cards. The relay cards are connected to the input and output cards. The input, output, and clock cards are inserted into slots. All that remains for the E is to attach the external switches and devices to the relays, load SKED, and start the experiments. The cost per bit (exclusive of the relay, switch, and light) is $4.17. The least expensive alternative to this interface design is with Flip-Chips (one M735, one M050, and two M203, for interfacing 12 in and 12 out). The cost of the Flip-Chip approach is $8.54 per bit.

SYSTEMS ANALYSIS

Input

A simplified diagram of the input interface is given in Fig. 1. A lever (S1) activates a relay. The relay indicates its operation by turning on a light; the relay also sets an R/S flip-flop. The switch will bounce, but it never bounces all the way back to reset the flip-flop. When the appropriate device code and IOP pulses are selected, the output of the flip-flop is multiplexed into a bus driver that goes into the corresponding accumulator bit. The multiplexer gates three inputs.

The input interface is a scanning type. The information is not stored, but is scanned on every clock pulse. The relay has been derated from 24 to 15 V. This slows the speed of the relay, so that no storage of information is necessary.

If the switch closure lasts less than 10 msec, as may be the case in work with pigeons, the normally closed contact of the relay may be disconnected and a one-shot attached to the reset of the flip-flop. The one-shot is triggered by the device code and another IOP pulse. Such a feature is incorporated in the interface.

Output

When the appropriate device code and IOP pulse are selected, information from the accumulator is transfered to a D flip-flop, as shown in Fig. 2. The output of the flip-flop drives a 7406 intergrated circuit. This device is capable of driving 30 V at 40 mA. A relay is driven by the 7406. A light indicates the status of the relay.

Clock

A M404 crystal clock is used. The lowest frequency of the clock is 5 kHz. SKED requires a clock rate of 100 Hz. Therefore, the rate must be divided by 50. The 50th clock pulse sets a flip-flop. The output of the flip-flop causes an interrupt. The computer will then execute an input/output command. If this command has the appropriate device code and IOP, it will cause the computer to skip the next instruction. The command also clears the flip-flop. A simplified diagram is seen in Fig. 3.

A switch may be added to the M404 to select 5 or

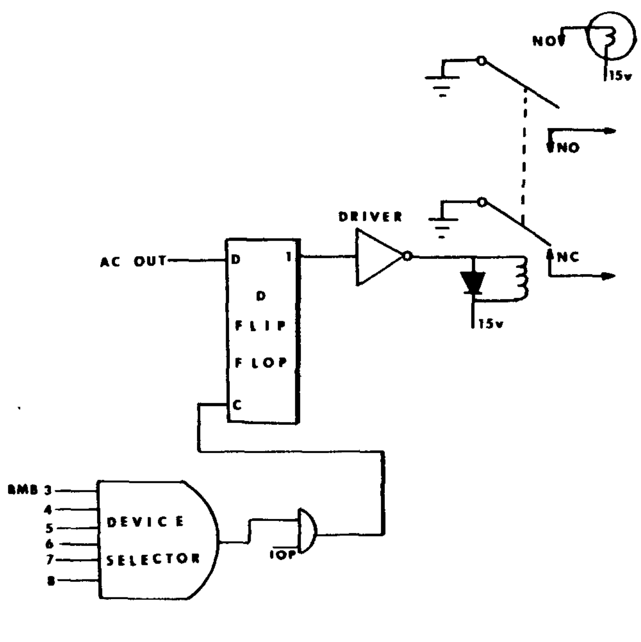

Fig. 2. Simplified diagram of the output interface.

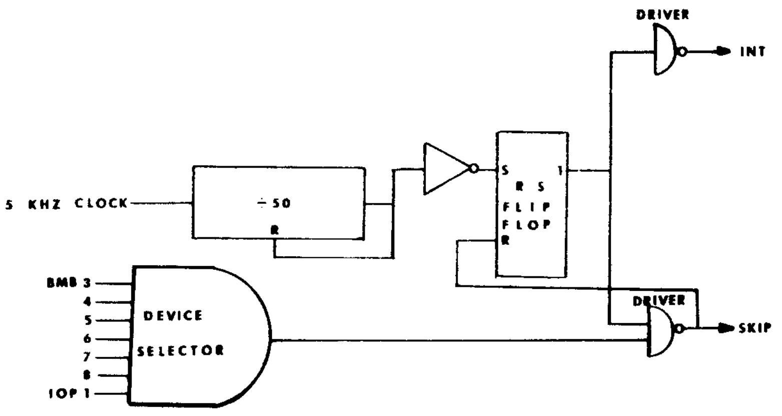

Fig. 3. Simplified diagram of the clock interface.

50 kHz. This selection translates into 100 or 1,000 Hz. The switch can also open the circuit between the clock and the divider. This effectively disables the interrupt so that other programs (i.e., FOCAL, etc.) may be run.

THE BUS-ALL APPROACH

The quad size input, output, and clock cards are placed in a H933 mounting panel. The signals, as they come from a PDP-8/L or the KA8-A, are bussed across the H933. The device code is wired on the input, output, and clock cards. The input and output cards may be placed in any slot. The clock card has resistors to terminate the IOP pulses, therefore it must be placed in the last slot, furthest from the cables.

REFERENCES

Snapper, A. G., & Kadden, R. M. Time-sharing in a small computer based on the use of a behavioral notation system. In B. Weiss (Ed.), *Digital computers in the behavior laboratory.* New York: Appleton-Century-Crofts, 1973. Pp. 41-97.

Snapper, A. G., Knapp, J. Z., Kushner, H. K., & Kadden, R. M. A notation system and computer program for behavioral experiments. Paper presented at the meeting of the Digital Equipment Computer Users Society, New York, June 1967.

Snapper, A. G., & Walker, A. The SKED software system. DECUS Program Library, DECUS 8-465, 1971.

Walker, A., & Snapper, A. G. Improvements to the SKED processor central software system. In DECUS Proceedings, Spring 1971. Pp. 7-12.

Behavior Research Methods & Instrumentation
1974, Vol. 6, No. 2, 173-176

Programming special functions in the SKED system

ARTHUR SNAPPER
Western Michigan University, Kalamazoo, Michigan 49001

and

BRUCE HAMILTON
American University, Washington, D.C. 20016

Machine language subroutines can be integrated with the SKED system. These subroutines can shorten lengthy programs that could otherwise be handled by SKED, and can provide complex decision functions, data recording schemes, and software for new peripheral devices. Rules and examples for each function will be presented.

SKED is a higher level, general language used to simplify programming the PDP-8 for experimental control and data recording. An unfortunate restriction shared by many user-oriented languages is their inflexibility when faced with a specific requirement unforeseen by the creators of the system. Some user finds his application needs 8.5K of core in an 8K machine, or that it is impossible to apply a program that was originally designed for schedules of reinforcement to verbal behavior experiments or to automated psychoanalysis.

Although SKED has more generality than most process control languages, for some applications the system will be either inefficient or impossible to use. For this reason, the Run Time System (RTS) has a built-in method for referencing machine language subroutines, so that the special-purpose programs can be readily accessed at the appropriate time or by the specified response in the state program.

The special-purpose program can be written (by an experienced programmer) in machine language, assembled by PAL III, and loaded along with the RTS in