Behavior Research Methods & Instrumentation
1980, Vol. 12 (6), 607-613

TUTORIAL

On choosing an inexpensive microcomputer for the experimental psychology laboratory

ADAM V. REED
Graduate Faculty of Social and Political Science, New School for Social Research
New York, New York 10011

This report compares currently available, inexpensive (minimum stand-alone disk-based configuration under $2,000), mass-produced microcomputers from the viewpoint of their applicability as experiment control, stimulus generation, and data collection devices for the experimental psychology laboratory.

In the last several years, many psychological researchers have found that experiments once thought to require a minicomputer, or at least a 16-bit microcomputer such as the DEC LSI-11, could be quite satisfactorily carried out on relatively inexpensive 8-bit microcomputers. The growth of a mass market for these personal computers has resulted in a dramatic decrease in prices, and 8-bit systems now sell for less than half the price of 16-bit systems equipped with comparable hardware. On the basis of prices advertised in the December 1980 issue of *Byte*, an APPLE II system with 64K bytes of RAM, UCSD PASCAL, two assemblers, two BASICs, two disk drives, a video monitor, a serial/parallel interface, and a direct-connect modem for off-loading data to a remote main frame, plus a matrix printer/plotter, costs a total of $3,400. A laboratory that already has a video monitor and can use a remote main frame for sophisticated data processing and for hard copy, eliminating the need for PASCAL, printer, and second disk drive, can get the needed system for $1,875. For comparison, a maximally discounted single-drive LSI-11-based system costs $5,650. Moreoever, an 8-bit microcomputer may be used as a network controller and development system for single-board subsystems using the same microprocessor (Reed, Lewart, & Schneider, in press). The KIM-1, a typical satellite subsystem with six seven-segment LED displays, 16 software-addressable keys, two programmable timers, and 30 TTL-compatible I/O lines, currently sells for $159 new or about $100 used.

There are over a dozen inexpensive (under $1,000 for a 16K-byte unit) microcomputers in today's market. When these are considered for laboratory applications, however, the choice narrows considerably. Since exact timing is often essential in laboratory research (see Reed, 1979), computers with no provision for machine language programming (the Texas Instruments TI 99/4, for example) have no place in the laboratory. The same is true for all computers with inadequate hardware documentation, especially if the manufacturer regards essential technical information as a trade secret, as happens to be the case for the otherwise very competent Atari machines. Finally, computers with bus structures unsupported by independent peripheral manufacturers (e.g., those made by Ohio Scientific, OSI) involve the risk of no alternative to the possibly high prices and unreliable supply of peripherals from the original manufacturer. This restricts the laboratory user's choice to three manufacturers: Apple, Commodore, and Radio Shack.

PROCESSORS

The integrated circuit that performs the computation inside a microcomputer is the microprocessor. The microcomputers under consideration in this article are based on three different microprocessors: the 6502 in the Apple and Commodore products, the Z-80 in the older TRS-80 (Radio Shack) designs, and the 6809 in the recent color-series TRS-80s.

The Z-80 microprocessor used in traditional Radio Shack products belongs to the triple-bus processor family, which also includes the 8008, the 8080, the 8085, and the 8088. The 8008, introduced by Intel in the early 1970s (there is some controversy about the exact date), was the first modern microprocessor and is sometimes called a zero-generation design. Most of the programmers who were available in the labor pool at the time the 8008 was introduced had been trained on the large computers of the 1950s and 1960s. In these early machines peripherals, memory components, and processor registers operated at vastly different speeds. For this reason, traditional computers used different sets of instructions to address memory, I/O, and general-purpose registers. The general-purpose registers were physically located inside the central processing unit to permit fast access. This arrangement facilitated the use of fast flip-flop technology for the general-purpose

0005-7878/80/060607-07$00.95/0

registers, but it limited the number of those registers to a dozen or so. In the 1970s, the situation changed: With the introduction of semiconductor memories and I/O interfaces, all computer components could operate at the speed of the processor, so that the use of different instruction sets for memory, I/O, and general-purpose register operations was no longer required. Nevertheless, in order to use available programmers and to simplify the adaptation of existing software to the new processor, the designers of the 8008 decided to follow the traditional tripartite architecture.

The first microprocessor to see widespread application was the successor of the 8008, Intel's 8080. The 8080 is generally referred to as a first-generation microprocessor. It had been kept software and bus compatible with the 8008. Intel's second-generation 8085 changed the specification of some bus signals but retained the triple-bus, tripartite instruction set. The Z-80, a second-generation microprocessor developed by Zilog, a competing company, was designed with a superset of the 8080 instruction set and the same architecture. This is also the case with Intel's third-generation 8-bit processor, the 8088.

Unlike the 8000 series, the 6000 series processors were designed from the start to take advantage of a single-bus architecture made possible by integrated circuit technology. The 6800, a first-generation microprocessor introduced by Motorola, was the first to permit the placement of I/O interfaces anywhere in the computer's memory space; I/O content could be operated on directly, like the content of any other memory location. High-speed general-purpose registers were placed physically outside the microprocessor chip, on the same bus as other memory. This permitted the use of an entire page of memory, 256 bytes, as a bank of 128 16-bit general-purpose registers. By contrast, the tripartite architecture processors remain limited to about three dozen registers. The single-bus architecture was combined with a more powerful instruction set in Motorola's second-generation processors, 6802 through 6805. A competitor, MOS Technology, introduced the 6502, a second-generation single-bus processor with a very clean and consistent instruction set. The 6502, now also manufactured by Rockwell and by Synertech, could be learned easily by engineers who had not used computers before. It soon accounted for the majority of all microprocessors used in process-control applications, as well as in small computers made by Apple, Commodore, OSI, Atari, and others. Motorola, in turn, improved on the 6502 with an even more powerful third-generation microprocessor, the 6809. The 6800, 6502, and 6809 are not instruction-set compatible, but they have a common conceptual structure and a common hardware bus. Because the 6809 became available only recently, I am not sufficiently familiar with its instruction set to discuss it at greater length. The remainder of this comparison, therefore, will be restricted to the Z-80 and the 6502.

Since the main memories of pre-solid state computers were relatively slow, the instruction sets of these machines discouraged the waste of time that would result from too many memory fetch cycles. Their instruction sets required that the operand, or its address, already reside in one of the processor registers. The Z-80 follows this convention, so that the only instructions addressable directly to memory are fetch (into a register) and store (register content in memory). The 6502, on the other hand, permits direct addressing of any single-operand instruction, including not only fetch and store, but also add, subtract, and compare with the accumulator, compare with either index register, perform a logical "AND," "OR," and "exclusive OR" with the accumulator, test 1 or more bits, increment and decrement, shift or rotate right or left, and jump to content (jump indirect). Moreover, since the 6502 does not need separate instruction codes for loading and examining the I/O ports, its designers were able to include a variety of addressing modes not available on triple-bus machines.

The 6502 addressing modes most relevant to the laboratory user are indexed indirect and indirect indexed. With these addressing modes, it is usually possible to access the indexed component of a list or matrix in one or two instructions. On the Z-80, which does not permit the combination of indexed and indirect addressing, the address of each cell in a table of operands must be computed explicitly before it can be used. The only exceptions are tables on Page 0 (not available on the TRS-80 because Page 0 is dedicated to read-only memory), programs that deal with at most two tables (one per index), and tables not longer than 255 bytes (the maximum indexed displacement permissible on the Z-80). Most psychological experiments use lists of stimuli and store data in tabular form, so that ease of access to tables and matrix elements is an important consideration. The following example compares routines implementing a typical routine that requires such access.

The routine shown in Tables 1 and 2 counts the number of errors made by each of n participants (the number in the group is not known in advance) in a classification experiment, leaving the count in 16-bit form in a table called ERCT (assumed to have been

Table 1

6502	Instructions		Bytes	T Cycles
	LDY	STIMLS	2	4
	LDA	(CORRSP),Y	2	5
	LDX	#N*2	2	2
LOOP	CMP	(RSPRTS,X)	2	6
	BEQ	DONT	2	6
	INC	ERCT,X	2	6
	BNE	DONT	2	6
	INC	ERCT+1,X	2	6
DONT	DEX		1	2
	DEX		1	2
	BNE	LOOP	2	6
Total:	11		20	51

Table 2

Z80	Instructions		Bytes	T Cycles
	LD	BC,STIMLS	4	20
	LD	HL,#CORRSP	3	10
	ADD	HL,BC	1	11
	LD	A,(HL)	1	7
	LD	IX,#RSPRTS	4	14
	LD	IY,#ERCT	4	14
	LD	B,#N	2	7
LOOP	LD	C,(IX+0)	3	19
	PUSH	AF	1	11
	IN	D,(C)	2	11
	POP	AF	1	11
	CP	D	1	4
	JR	Z,DONT	2	12
	INC	(IY+0)	3	23
	JR	NZ,DONT	2	12
	INC	(IY+1)	3	23
DONT	INC	IX	2	10
	INC	IY	2	10
	INC	IY	2	10
	DEC	B	1	4
	JR	NZ,LOOP	2	12
Total:	21		46	255

preset to zero during initialization). The responses are collected with ASCII keyboards interfaced to an arbitrary set of parallel input ports, the addresses of which are found in table RSPRTS. The stimulus code for the current trial is found in memory location STIMLS, and the table of correct responses, classified by stimulus code, starts in location CORRSP. The 6502 code for this routine runs as shown in Table 1. The corresponding Z-80 code is shown in Table 2.

The instruction count, 11 for the 6502 routine and 21 for the Z-80, is a rough measure of relative programming effort. The byte count ratio, 20 for the 6502 to 46 for the Z-80, corresponds to the ratio of the amount of memory needed for equivalent programs in computers based on the two microprocessors. Finally, the cycle time totals correspond to the speed of execution, when adjusted for different clock rates in computers using the two processors. With equivalent technology, a Z-80 can be made to execute four times as many cycles per second as a 6502. The 4-MHz Z-80 used in the TRS-80 Model II, for example, is the technological equivalent of the 1-MHz 6502 used in the APPLE II and in the 2000 series PET/CBM from Commodore. The Z-80s used in TRS-80 Models I and III, on the other hand, run at about 1.5 MHz. Thus, equivalent programs can be expected to execute 1.25 times slower on a TRS-80 Model II than on a 1-MHz 6502-based computer. On a Model I or Model III TRS-80, the computation takes about three times as long as it does on a 6502 system.

The 6809 processor used in the recently introduced TRS-80 color-series computers is likely to be considerably faster than a 6502, but the clock rate used by Radio Shack in its latest computers was not available to the author as this article was being written.

The Z-80's disadvantage of less efficient optimal code is balanced by two advantages, due to its traditional architecture. Until recently, these advantages more than compensated for its relative inefficiency. First, professional programmers are still learning heuristics appropriate to the single-bus architecture, and old habits from years of working with triple-bus machines are likely to persist. Many published and commercially distributed programs for 6000 series machines are suboptimal, sometimes taking two, three, or four times as much code as necessary to perform their functions. The obverse of this is that once the programmer acquires the skills necessary to use all features of single-bus processors, programming on traditional triple-bus processors begins to be experienced as slow and cumbersome.

Single-bus processors also suffer from the impossibility of translating traditional software to run on the new architecture. Thus a language processor, for example, has to be written from scratch to run on a 6000 series machine, whereas the Z-80 version can be adapted from language processors originally written for traditional main-frame computers. Thus, until recently, the only languages available for the 6502 were BASIC, the mass-market beginner's language, and computer enthusiasts' languages such as LISP and FORTH. This situation has improved considerably since the introduction of the UCSD p-system. Under the p-system, computer operations are broken down into a set of machine-independent p-codes, or primitives. A highly optimized p-code interpreter is then written for each target processor separately. In the case of multichip microprocessors, such as those manufactured by Western Digital, the p-code interpreter may be microprogrammed into the processor chip set. With single-chip processors, the p-code interpreter is generally kept in software. Processors for higher level languages, such as PASCAL and FORTRAN, are written in p-code and can be executed on any computer equipped with the p-code interpreter.

With the availability of the p-code system on the Apple microcomputers, the main appeal of the Z-80 is for applications that require COBOL, the only popular high-level language not supported by the UCSD p-system. A great deal of commercial and business software is available only in COBOL, so Z-80-based computers will continue to be marketed for the forseeable future. Many computer vendors believe that only computer systems capable of running COBOL deserve the adjective "professional," and I have even met a psychologist who was sold a very expensive Z-80-based computer system after telling the salesman that he wanted a professional machine. There is, of course, one remaining good reason to buy a Z-80: to run Z-80 programs written on machines acquired before FORTRAN was available for 6000 series computers. Even then, the best choice may be a machine whose backplane can support multiple processors, so that a user might normally run on a 6502 or 6809, switching to a Z-80 to run existing Z-80 software.

USER SOFTWARE/SYSTEM SOFTWARE INTERACTION

In most microcomputers, the limited resources of a single microprocessor have to be shared between the system software (built into the computer's read-only memory or the disk-based operating system), necessary to handle the interactions between the computer and its peripherals (including the human user's display and keyboard interfaces), and user software, the programs actually employed by the user to perform operations of the user's choice. Conflicts between user and system software may arise from (1) timesharing the processor between user and system operations or (2) reservation of some resources for the system, making them unavailable to the user.

The timesharing conflict is of considerable importance in the psychological laboratory, in which exact and repeatable timing is often essential. The one system facility that must be continuously serviced, even when a user program is being executed, is the video display. This is handled differently in different computers (see Reed, 1979). On the APPLE, the main memory runs at twice the speed of the processor, and alternate cycles are used to service the display. Thus, processor and display operations are mutually "transparent" (i.e., they do not affect each other at all). On the Commodore and Radio Shack machines, processor and memory cycle times are equal. During a video scan, the segment of memory used by the display is logically disconnected from the processor.

On Radio Shack computers, display memory cannot be accessed by the processor during the display cycle. If an attempt to access a display location is made during the display cycle, the processor is made to wait as long as it takes for the display cycle to be completed. On Commodore computers, the processor has priority of memory access. The display circuit is momentarily disconnected from the memory if the processor requests access to a byte within the display segment. Thus, the display will blink if the memory display segment is accessed directly by the user's software. The blinking may be avoided by using system firmware to access the screen. This firmware performs the same function as the corresponding hardware in Radio Shack computers: It makes the user's program wait until the end of the display cycle before resuming operation. The effect of this, like the effect of trying to access screen memory directly on Radio Shack machines, is the introduction of unpredictable delays lasting up to one frame time (17 msec) into software timing. Commodore does provide the user with hardware timers that are not affected by this variability (although software access to those timers may be affected). On the TRS-80s, precisely replicable timing cannot be achieved without additional user-supplied hardware.

The second possible conflict between system and user operations concerns resources that are not available to the user as a result of having been reserved for system software. The TRS-80s and the PET/CBM both use the nonmaskable interrupt (NMI) line of the processor in ways that make it unavailable to the user. On the APPLE, both the NMI and the maskable interrupt (IRQ) are available for user requirements. The availability of the NMI line often simplifies application software considerably. For example, a precise millisecond real-time clock may be implemented with less than one page of software and a 1-kHz multivibrator connected to the NMI line. The NMI line is usually essential for real-time microcomputer processing of very rapid events, for example, in neurophysiological unit recording applications. Most human experimentation does not require use of the NMI, although doing without it tends to increase the complexity of application software.

SECURITY AND PORTABILITY

Microcomputers, unlike traditional laboratory equipment, are popular consumer products. They are therefore easy to pawn or sell, and attractive to thieves. The APPLE and the traditional TRS-80s are packaged in light, easily portable, and easy-to-steal modules. The television monitors and cassette recorders used with those microcomputers are also likely targets of theft. The PET/CBM and the recently introduced TRS-80 Model III, on the other hand, each consists of a large integrated assembly housing the video monitor as well as the computer modules, and sometimes also including a disk or cassette drive. Such an assembly can be readily anchored in place. Even if not bolted to its location, its bulky shape and weight (30-50 lb) would be enough to make it difficult to get away with. While physical security need not be a major consideration if the laboratory is inaccessible to passers-by and can be reliably locked, the PET/CBM and the TRS-80 Model III would seem to be the only acceptable choices for a laboratory faced with a less ideal security situation.

The obverse of the above considerations would apply if the experimenter needs to carry the microcomputer to locations outside the laboratory. Fitted luggage is available for all portable microcomputers.

HARDWARE

Electromagnetic Interference

The cabinets of Radio Shack and Commodore microcomputers are designed to act as Faraday cages, shielding the computer from electromagnetic interference (EMI). EMI is not an important consideration in most laboratories, but it may be important in on-site studies of human performance in military and industrial environments, ornithoethological studies that use radar equipment, and so on. The APPLE II is not effectively shielded against EMI and should not be used in environments in which EMI is likely.

Socketed vs. Soldered Construction

The APPLE II is constructed with plug-in sockets for nearly all integrated circuits (the only exception is the keyboard assembly, made for Apple by a different manufacturer). Plug-in sockets make possible a very simple, reversible hardware modification technique (Reed, 1979), using plug-in wire-wrap sockets. Besides making hardware modifications easy to implement and to change, all-socketed construction simplifies hardware diagnostics and repairs: One can simply pull out any suspect part, plug in a replacement chip, and determine whether the malfunction has been corrected. Unless a laboratory has a skilled electronics technician always on call, simplicity of modification and repair saves a great deal of time. On Radio Shack and Commodore computers, only a few integrated circuits (those most likely to need replacement) are socketed. This makes almost every repair or modification into a sophisticated and delicate procedure, to be attempted only by skilled personnel with proper equipment. It is always a good practice to install a socket, if not originally present, for any integrated circuit that has had to be replaced.

Apart from simplifying the manufacturing process, soldered construction improves reliability in the presence of mechanical vibration and shock. This means that Commodore and Radio Shack machines will be more reliable than the APPLE in mechanically adverse environments, although they will still be more difficult to repair once they do break down.

BUILT-IN VIDEO DISPLAYS

The following discussion applies to the display facilities found in the less expensive models from the major manufacturers: the APPLE II, PET/CBM 2000 series, and TRS-80s other than Model II. Although each manufacturer also markets models with more extensive display capabilities (APPLE III, CBM Series 8000, and TRS-80 Model II, respectively), these machines are oriented primarily to business applications and are expensive enough not to be competitive with traditional LSI-11-based systems for laboratory applications.

Multiple Display Pages

Multiple display pages simplify considerably the task of programming video displays for psychological experiments (see Reed, 1979). The APPLE II comes with two built-in text pages of 1K (1,024) bytes each and two bit-map graphics pages of 8K each. Other inexpensive microcomputers do not have multiple display pages built-in, although all can be modified for multipage operation if software solutions become inadequate to deal with display problems.

Color

The APPLE II and the new 6809-based Radio Shack color computer have built-in color capability, which is essential in experiments on cognitive aspects of color (such as the Stroop task) and may be useful in holding the attention of young children in developmental psychology experiments. The user of the APPLE II may choose between block mode (using the 1K display pages with 16 colors in a 40 by 48 grid) and point mode (using the 8K pages with 6 colors and a 140 by 192 grid) graphics (note that black-and-white graphics use of the 8K pages also gives 192 lines, but with up to 280 points in 560 positions/line). The number of usable colors may be increased to 136 in block mode and 15 in point mode by interlacing the two display pages available in each mode.

Although a radio-frequency modulator may permit the use of microcomputer-generated color displays with an ordinary color television set, especially in entertainment and educational applications, a direct video color monitor is usually necessary in meeting the convergence and stability requirements of psychological experimentation. Even with a direct video color monitor, the much stricter requirements of experiments on the psychophysics and perception of color cannot be met by the built-in circuitry of any inexpensive microcomputer. For the latter type of application, it will usually be necessary to use three separate high-resolution peripheral display boards (see under Expansion) and a high-resolution RBG (separate red, blue, and green inputs) monitor.

Standard Character Text

All three microcomputers have built-in character generators for numbers, uppercase English letters, and most of the punctuation marks and special symbols found on typewriter keyboards. The TRS-80 displays 16 rows of 64 characters each (32 characters each on the TRS-80 color computer). The PET/CBM displays 25 rows of 40 characters each. The APPLE II displays 24 rows of 40 characters from either display page. A plug-in accessory for the APPLE II permits the display of 80-character lines, although with individual characters too compressed for use as stimuli in most psychological experiments.

Nonstandard Characters

Two different approaches to the problem of displaying nonstandard characters have been taken. One is to permit the substitution of a second character generator with the nonstandard characters for the standard generator, under software control. A circuit permitting this is built into the PET/CBM and TRS-80 Model III and is available as an accessory for the APPLE II. Although supplied by the manufacturer with a lowercase English character-generator read-only memory chip, other character-generator read-only memories may be substituted if available (e.g., Greek, Hebrew, Katakana, APL).

The other approach requires the use of a relatively dense bit-map point grid and therefore is only available to users of the APPLE. In this approach, characters are designed by the user and output to the screen with bit-mapping software. This permits not only the display of

standard character sets as above, but also the display of characters in various sizes, nonstandard fonts, Gibson figures, languages not available on character-generator read-only memories, connected text (English cursive, Farsi, Arabic), characters that require a larger than usual grid (such as Chinese ideographs), musical scores, and so on. This approach also permits the rotation of characters or lines (upside down, mirror image, etc.) and movement of characters to locations not on the standard text grid.

Bit-Map Graphics

In the APPLE II, each byte in the 8K bit-map display pages corresponds to seven consecutive display points in a 280 by 192 grid. The 9th bit may be used to displace the projected location of the byte .5 point to the left, giving the user 560 point positions to choose from (560 by 192 position grid). The point raster is exactly the same for bit-map mode as for standard text-mode displays, and text may be combined with bit-mode graphics under software control.

The PET/CBM and the TRS-80 provide the user with a relatively coarse bit-map display capability based on letter-size "graphics characters." Each letter is subdivided into four (PET/CBM) or six (TRS-80) bit-map fields. The corresponding bit-map display grid is 80 by 50 on the PET/CBM and 128 by 48 on the TRS-80.

OTHER BUILT-IN I/O

Keyboards

All microcomputers under consideration have built-in keyboards that can be used for interactive response collection if response latency measurement is not critical. These keyboards have a "smooth" response to increasing mechanical pressure, rather than the "crisp" or "snappy" response necessary for very precise latency measurements. In experiments in which response latency measurement with precision better than ±5 msec is required, low-force switches interfaced to digital input lines should be used in preference to the standard keyboards.

Digital Inputs and Outputs

The Radio Shack computers have no built-in digital I/O, but TTL-level (0 and +5 V) I/O may be added readily by the user. The APPLE II has three input and four output lines. If one of the input lines is reserved for video synchronization, the remaining two may be multiplexed with the four output lines in configurations of up to 32 switches (e.g., 2 switches/subject for up to 16 subjects). A plug-in board with two 6522 versatile interface adapter (VIA) chips is also available. It provides 32 directly addressable digital I/O lines, as well as four hardware timers, two shift registers for waveform generation, and so on. The PET/CBM is equipped with a built-in chip similar to the VIA, and 15 of its 16 I/O lines are available to the user. Either of the expanded configurations permits virtually unlimited flexibility in experiment control and data collection without additional hardware other than, perhaps, a few driver and multiplexer integrated circuits.

Audio Outputs

All the microcomputers considered above have audio cassette outputs, which, although designed for data and program recording, may also be used in conjunction with an external amplifier and speakers or headphones to generate rudimentary auditory stimuli. The APPLE II also has a built-in audio amplifier and speaker separate from the cassette interface. This speaker may be used to generate auditory cues and signals, even when the cassette interface is connected for its designed use. Additional hardware (see Expansion) is available for the production of higher quality musical or speech-like sound.

Analog Inputs

The APPLE II is equipped with four timer-based analog inputs. These may be used in experiments concerned with motor tracking or in any other setting in which the continuously variable input can be made to control a potentiometer. The sampling rate depends on the software used and could be made as high as once every 2 msec/input (e.g., every 6 msec with three inputs, etc.). If higher sampling rates are needed, hardware-based analog-to-digital converters are available for all microcomputers.

Instrument Control

Commodore microcomputers have a built-in controller for the IEEE 488 standard instrument control interface. After-market controllers for the IEEE 488 interface can be purchased for the Apple and Radio Shack machines.

EXPANSION

Buses

In electronics terminology, a bus is a cable or backplane specification for a group of electrical signals traveling together. Some familiarity with buses is essential for informed choice among microcomputer expansion peripherals.

Communication Buses

Communication buses are buses designed to carry information from one place to another. Serial (more exactly, bit-serial) buses transfer information one binary digit at a time. Standard serial buses include the 60-mA current loop, 20-mA current loop, and the now-dominant EIA RS-232 voltage-level bus.

Parallel (more exactly, bit-parallel byte-serial) buses transfer information 1 byte (7 or 8 bits) at a time. Two such buses were developed by Centronics and by Hewlett-Packard. The Hewlett-Packard bus, since standardized as IEEE 488, is widely used for control of physical, biological, and chemical instrumentation. The Centronics

bus was never officially codified into a standard, but it is now even more widely used than the IEEE 488, primarily for control of printers and plotters in computer systems and for high-speed communication in computer networks.

System Buses

System buses are designed for communication among subsystems, usually mounted on separate printed circuit boards, of a single computer system. Microcomputer system buses include two categories, "professional" and "amateur." Professional buses include signal lines for in-circuit emulation functions, used by engineers in the development of microprocessor-based hardware. The major professional buses are the MULTIBUS, developed by Intel and used for the 8080/Z-80 family of microprocessors, and the EXORCISER bus, developed by Motorola and used for the 6000 series single-bus processors. The plug-in boards for these buses tend to be more expensive than are the corresponding boards for the "amateur" buses, but they may provide functions not available on amateur-bus plug-ins. Although in general the "professional" buses are not used in systems that do not need in-circuit emulation capability, they are mentioned here because S100-to-MULTIBUS and KIM-bus-to-EXORCISER-bus adapters can be obtained and may be used in laboratory applications for special functions not otherwise available.

The earliest "amateur" bus was the S100, still widely used and recently codified, after extensive revisions, as the IEEE S100 standard. The designers of the original S100 bus had no previous experience in system bus design, and the original S100 configuration proved to be extremely susceptible to electromagnetic noise and to cross talk between different signals. This gave rise to a lively after-market in active termination devices designed to eliminate noise, which would not have been present had the bus been properly designed in the first place. Being first, however, gave S100 a lasting advantage over better designs, and a wide variety of both processor boards (with 8080, Z-80, 8085, 8088, 6800, 6502, 6809, and several other microprocessors) and peripheral boards for that bus remains available. Since the separate I/O addressing lines incorporated in the S100 bus were not needed for 6000 series microprocessor systems, a similar bus without those lines, called SS-50, also attracted several manufacturers, but it never became as popular as the S100. Adapters that allow the use of S100 peripherals are available for the Z-80-based TRS-80s and for the PET/CBM.

KIM-1, the first rudimentary 6502-based microcomputer, incorporated an interface for a professionally designed "amateur" bus, which became known as the KIM bus. Since then, several other microcomputers incorporating interfaces for the KIM bus have appeared, including the SYM-1, the AIM-65, and the SUPERKIM. KIM-bus computers have remained rudimentary relative to the video microcomputers discussed here, but highly sophisticated peripherals for the KIM bus tend to be relatively inexpensive. A KIM-bus adapter is available for the Commodore microcomputers.

Radio Shack developed a proprietary expansion bus for the Z-80-based TRS-80s, but peripherals designed for this bus tend to be less versatile (although often less expensive) than their S100 counterparts. Either bus will work well with a Z-80-based TRS-80.

The Apple peripheral bus, found on the APPLE II, is a very sophisticated design that owes more to DEC's minicomputer buses than to other 8-bit designs. Like the LSI-11's Q-bus, the Apple bus supports daisy-chain arbitration of DMA and interrupt priority. It also supports distributed intelligence, auxiliary processors (a Z-80 card and a fast arithmetic processor card are already available, and a 6809 card is expected shortly), and memory overlays to a theoretical maximum of 16 million bytes (although at this point, no memory management cards are available). The Apple bus is a superset of the KIM bus. No Apple-to-KIM bus adapters are available commercially, probably because most functions of interest to the typical user are already available on Apple-bus plug-ins, but it would be a very simple matter to build one. An Apple-bus adapter is available for the PET/CBM. It is very likely that an Apple-bus adapter for the new 6809-based TRS-80 color computer will appear in the near future.

The TRS-80 color computer itself is still undocumented, and it will be a while before enough is known about it to recommend it to the laboratory user. Relative to the APPLE II, the first single-bus machine from Radio Shack has a more powerful processor and a much lower entry-level price. Radio Shack's documentation has been a strong point in the past; so has the availability of peripherals. The technical aspects of hardware design, on the other hand, have been rather poor. The choice of a modern processor for the new machine may be a sign that things are changing for the better in that department, and it is possible that psychologists shopping for an inexpensive laboratory microcomputer will have an even wider choice in the future.

REFERENCES

REED, A. V. Microcomputer display timing: Problems and solutions. *Behavior Research Methods & Instrumentation,* 1979, **11**, 572-576.

REED, A. V., LEWART, D., & SCHNEIDER, L. H. Low-cost microcomputer networking: How to get high throughput on a low budget. *Behavior Research Methods & Instrumentation,* in press.

(Received for publication December 24, 1980; accepted December 26, 1980.)