SESSION III: WORKSHOPS

On-line computing with the Hewlett-Packard 2116B moving-head disk operating system

E. C. CARTERETTE, A. BARNEBEY, J. D. LOVELL
D. C. NAGEL, and M. P. FRIEDMAN
University of California, Los Angeles, California 90024

"The purpose of computing is insight,
not numbers"

R. W. Hamming
*Numerical Methods for
Scientists and Engineers*

1. INTRODUCTION AND DESCRIPTION OF THE OPERATING SYSTEM

A. W. Burks, H. H. Goldstine, and J. von Neumann in 1946 (Taub, 1963) gave a lucid preliminary discussion of the logical design of an electronic computing instrument: "To achieve a total electronic storage of about 4000 words we plan to use 40 Selectrons [storage tubes], thereby achieving a memory of 2^{12} words of 40 binary digits each. [We believe] that this memory capacity exceeds the capacities required for most problems

that one deals with at present by a factor of about 10. The precision [about 12 decimals] is also safely higher than what is required for the great majority of present day problems." Such a machine, having an add time of 5 microsec, would have put some 160,000 bits at their disposal, but only a group like the Army Ordnance Department (who sponsored the report) could have afforded to build and maintain this vacuum-tube instrument. The high-speed input and output proposed was magnetic wire. The transfer rate,

allowing for reading and conversion, was to be about 6,000 bits/sec. Only numerical calculation was considered in this article. In retrospect, these pioneers seem oddly conservative to us, to whom minicomputers with 500 nsec cycle time and 500,000 bits of high-speed store are not unusual.

Built but 22 years later, our modest computer has a core storage of 260,000 bits, adds two numbers in 3.2 microsec, and can transfer data to and from unlimited disk storage at about 1.5 megabits/sec. The essential features of our electronic computing

Fig. 1a. Schematic layout of the computer.

CHANNEL	DEVICE	CHANNEL	DEVICE
10	10-bit A/D; Dual 12-bit D/A.	20	16-bit relay register.
11	1 megacycle clock/preset counter.	21	High-speed punched-tape reader.
12	Time Base Generator (100 kHz base).	22	Teleprinter (10 characters/sec = 110
13	CRO Teleprinter Terminal (Datapoint 3300) (110-2400 BAUD)	23	SPARE
14	SPARE	24	Disc data channel
15	Dual 8-bit D/A	25	Disc command channel
16	16-bit input-output general purpose register	26	SPARE
17	16-bit input-output general purpose register	27	SPARE

Fig. 1b. The current allocation of input/output channels to external devices. Note that Channels 10 and 11 interact through a logical control for high-speed inbitting and outbitting.

instrument—a Hewlett-Packard HP2116B¹—are these: central processor (CP) store of 16,384 16-bit words, 1.6-microsec cycle time, 3.2-microsec store time, 3.2-microsec add time, 70 basic instructions, multilevel priority interruption, two addressable accumulators, two channels of direct memory access (DMA) having a transfer rate of 263,000 words/sec, and four language compilers—Assembler, ALGOL, Basic, and FORTRAN IV. This “analytical engine” has its life dedicated to serving the research and instruction in our laboratory only. Most of its computing is to the end of controlling experiments in perception and psychophysics. Figure 0.0 is a photograph of our computer and has not been included because virtually nothing about the machine’s hardware or software architecture could be deduced from a photograph.

Since August 1970, we have run the computer as a disk-operating system via the two high-speed DMA channels. The cartridge disk drive is a Hewlett-Packard Model HP2870A

interfaced to the CP by an HP2871A controller. One fixed disk and one moving disk are served by the same moving head. Each of the disks stores over 11×10^6 bits, making the total of active on-line store over 22 million bits, which translates to 1,247,232 16-bit words. Off-line store is unlimited. Each member of the laboratory has one or more removable cartridges. Each cartridge stores over 600,000 words and costs about \$80.

By means of a Datapoint 3300 cathode-ray display console operating at 2400 baud (21.8 times TTY rate) connected to a DOS-M, the creation, checkout, and execution of programs is quite simple. The computer can be run from the keyboard or under batch monitor control, as from a high-speed paper-tape reader. Programs can be automatically assembled, compiled, loaded, and executed without intermediate paper tapes. The complex matter of input and output control is centralized. In addition to the disk manager and the job processor, all I/O drivers and certain executive modules are core resident, which increases

efficiency and speed. The overhead of our DOS-M is about 5,100 words, leaving more than 11K of the 16K core memory. Even though DOS-M allows segmentation and overlay of programs, we have only once required this feature. A main program can be coded in FORTRAN, ALGOL, or Assembly and can call subprograms coded in any or all of these three languages.

2. APPLICATIONS: ON-LINE, OFF-LINE, AND MAIN LINE

From this system, we run on-line experiments in psychological and physiological acoustics, as well as others in vision and perception. A great deal of this work requires very precise timing, on the order of a millisecond in an auditory or visual backward masking experiments or on the order of a few microseconds, as in binaural lateralization experiments. The presentation of stimuli conditional upon previous responses is routine in virtually every experiment. Auditory and visual displays are controlled by means of external logic modules and interfaces (see Fig. 1): digital-to-analog converters (DAC), analog-to-digital converters (ADC), digital attenuators, switches, voltage-controlled generators, cathode-ray oscilloscopes (CRO), microelectrode drives, and the like. The physical plan of the laboratory is shown in Fig. 2.

An important feature of DOS-M is its power as a synthesizer of complex waveforms and noises and as a Fast Fourier Transform analyzer and synthesizer. The combination of 16-bit-word, moderately high-speed, and very large disk store (almost a virtual memory!) gives us a complete digital filtering system with all the versatility that the term implies.

We have made considerable use of the DOS-M for dynamic modeling. Recently we have written simulation models of the basilar membrane (Lovell, Carterette, & Nagel, 1971), the auditory neurone, and the psychophysical observer (Nagel, 1970). The basilar membrane model allows us to display the time-varying displacement in response to such auditory inputs as sinusoids, square waves and impulses, noise, speech—indeed to any complex stimulus. A cinema of the behavior of the model on precision-CRO displays was shown at the Conference.

The most recent setup is an auditory physiology laboratory for interactive on-line stimulation of and recording from individual neurons of the auditory system, and the binaural system in particular. In addition to displays of single spikes and statistical features such as response histograms,

PERCEPTUAL SYSTEMS LABORATORY

1. COMPUTER (HP-2116B) AND DISC.
2. INSTRUMENTS, CONTROL AND SWITCHING.
3. LARGE SUBJECT ROOM.
4. ANECHOIC CHAMBER.
5. IAC QUIET ROOM.
6. PISSOIR.

Fig. 2. The physical plan of the Perceptual Systems Laboratory. The distance from the CP 1 to the physiological acoustics setup is about 40 m by cable.

amplitude-frequency response areas ("tuning curves") can be automatically computed and displayed. A pulse-amplitude window discriminator has been designed to aid in finding and holding single units.

Another kind of work that we are tooling up for will involve audiometric testing of children and running them with interactive programs in order to measure their ability to process information in reading, hearing, and speaking.

3. SOME ADVANTAGES OF CYCLIC STORE

As we have tried to show, the DOS-M is an exceedingly general system for programming, numerical computation, and process control. It is relatively slow in the sense that track-to-track access time is 30 msec, average rotational delay is 20 msec, and an average random move takes 70 msec (these figures are halved in the latest HP7900 Cartridge Disc, and the storage density is doubled). The maximum number of words that can be read or written before a new track must be sought is 3,072. However, data can be transferred between disk and core at nearly 50K words/sec. By alternately reading from two buffers in core, 20,000 samples/sec can be output over a dual 12-bit DAC. Equally high sampling rates are possible with analog-to-digital conversion.

Although the cost per bit stored on a moving-head disk (3×10^{-6} \$/bit) approaches that of magnetic tape (2×10^{-7} \$/bit) (Bell & Newell,

1971), there is a great difference in the speed with which a given random location can be accessed. This difference in speed is fundamental, since a wider class of experiments is possible by means of the disk. In work carried out in our laboratory, large auditory arrays such as tonal complexes or special noises are stored on the disk. The noises are created using a simple but versatile Fourier synthesizer coded in ASA FORTRAN. The program (Lovell & Carterette, 1972) constructs acoustic arrays from recipes entered by keyboard.

Some experiments (see, e.g., Carterette, Friedman, & Lovell, 1969; Lovell, Carterette, & Barnebey, 1971; Carterette & Lovell, 1971) require that a given sample of noise be repeated exactly within 125 msec or that a random sample from one noise be presented simultaneously or immediately following a sample chosen randomly from one of a set of other noises. Such experiments are elementary to carry out from DOS-M but present major problems using a linear store on magnetic tape. To appreciate this, consider that a 3-sec 90-dB SPL noise generated at a 10-kHz sampling rate is 30,000 16-bit words long, i.e., 480,000 bits long. A magnetic-tape drive reading and writing at 80 bpi and run at 45 ips requires at least 30 in. of store. In the worst case, that of two samples separated by half the length of the tape, access time (at 150 ips) could be on the order of 60,000 msec. For DOS-M, the average access time is about 100 msec independently of location of a sample.

4. USING DOS-M IN INSTRUCTION AND RESEARCH

For the purpose of training students (and faculty!), the value of a DOS-M like ours cannot be overestimated. A program can be entered from the keyboard and instantly displayed for editing (where instantly means up to 2400 baud, or 21.8 times the 110-baud rate of the Teletype ASR-33). Programs of 100 lines are edited in 2 or 3 sec after the entire set of edit instructions have been entered. Compilation is rapid. A FORTRAN program of 15 lines takes about 12 sec, 67 lines about 18 sec, and 140 lines about 28 sec. A short Assembly program of 33 lines requires some 7 sec and one of 493 lines about 35 sec.

In fact, on the basis of about 15 randomly selected source programs, we found that the compilation time, C(L), was a linear function of the number of lines, L. Thus for FORTRAN IV programs,

$$C(L) \div (L/7) + 8 \text{ sec} \quad (1)$$

and for Assembly programs

$$C(L) \div (L/16.5) + 5 \text{ sec} \quad (2)$$

ALGOL programs go faster than FORTRAN because ALGOL is a one-pass compiler.

The FORTRAN program and the two Assembler subroutines illustrated in Fig. 3 (a), (b), and (c) took a total of 27 sec to compile and 55 sec to load all library routines. At the end of this time, the program could be run at once from the keyboard by simply typing :RUN, VENUS.

With the essentially unlimited bulk storage of programs and data and the relatively rapid turn-around time, learning how to program is enormously facilitated. In the psychoacoustics laboratory, skill in programming is a powerful impetus to knowledge. It becomes possible, for example, to analyze and synthesize waveforms and display them quickly by means of Fast Fourier Transform programs based on the Cooley-Tukey (1965) algorithm. A new version of an on-line controlled experiment can be quickly written, compiled, and run.

This speed of revision is made possible by virtue of a driver,² DVR37, discussed below, and by the use of standard input/output utility routines stored on the disk in relocatable form. The names of these routines are simply entered from the keyboard at load time, and DOS-M automatically collects and links them with the calling programs. One of these relocatable routines is shown in Fig. 3(c). Some of the most important utility routines are described in Table 1. They are controlled by statements in the calling program.

Table 1
Some Basic Utility Routines for Controlling Inputs and Outputs*

DCLK (Assembler)	Time base, preset clock, or counter.
UDTR (Algol and Fortran IV Versions)	Wetherill (1966) up-down transformed response adaptive method of obtaining thresholds. Variables accepted at run time are initial intensity, step size of intensity changes, and a value that sets a stopping rule.
DRLY (Assembler)	Opens and closes arrays of relays. Used with switches, digital attenuators, warning signals, feedback lights or tones, etc.
DBX1 (Assembler)	Initiates a trial and collects responses from S's panel.
DESW (Assembler)	Controls Grason-Stadler electronic switches.
DSP (Fortran)	Outputs to CRO an X-Y display of waveforms stored on the disk.
RAND (Assembler)	Generates pseudo-random numbers according to a conventional multiplicative generator. ⁴
SRAND (Assembler)	Passes a starting random number to RAND.

*The language in which each is coded is shown in parentheses.

Thus, in FORTRAN, CALL CLOCK (0, 3) sets up a 30-microsec interval and CALL EXEC (2, 8, CLKON) turns the clock on for 30 microsec. An example with context is given below in Section 5.

5. HOW TO HAVE YOUR DOS-M AND BEAT IT, TOO

For certain classes of experiments, it was important to bypass momentarily the control of DOS-M in the interests of speed, precision, and simplicity. Since only one program is run at any one time by DOS-M, bypassing was simple to arrange by means of a dummy driver.

Consider as an example the exceedingly simple driver, DVR37 (written by JDL). It is activated by a directive in the FORTRAN main program, CALL EXEC (2, 8, XXXXX), as in Lines 33, 35, and 37 [Fig. 3(a)]. The effect of the call is just to turn over the system to a subroutine. When the subroutine completes its task, control is returned to DOS-M. The versatility of our system ultimately inheres in the fact that DOS-M can be subverted by several special drivers like DVR37 when interference from DOS-M cannot be tolerated, as in the case of synchronous, dense, high-speed inputs and outputs of auditory signals or visual displays.

By means of this elementary device—calling a dummy driver—we have been able to enjoy all of the benefits of the DOS-M while escaping from its complex priority interrupts and memory-protection³ schemes whenever problems of speed or timing of I/O require or when programming is simplified. Thus, entirely with keyboard commands, we can edit, compile, load, store, and access and run dynamic interactive on-line

experiments with the great generality and convenience that the DOS-M's sophisticated Disc Manager and Job Processor afford.

We illustrate the elegant simplicity of the use of DVR37 in Fig. 3. A fragment of a FORTRAN program is shown in Fig. 3(a). The program was written by us in order to carry out the luminous calibration of a cathode-ray-oscilloscope (CRO) visual display (Sperling, 1971). Lines 1-25 (not shown) compute an $N \times N$ matrix of integer pairs. Each pair is packed into a single 16-bit word that specifies a point on the face of the CRO. Lines 26-31 read in the duration and intensity of the display. Line 32 calls subroutine BRIT [Fig. 3(c)] and passes to it a value (LUMEN), stored finally in location IBRIT,⁵ that controls the voltage level of the z-axis of the CRO. Likewise, in Fig. 3(b), Line 34 calls FLUX and passes the number of seconds (ISEC, up to 32, 677) of duration of the display, storing it in ILUX.⁶ Note that intensity and duration have only been passed. Neither subroutine was activated by the call. This finesse in separating parameters from action requires two, instead of one, calls but allows for flexibility and accuracy of timing. The parameter call can precede the action call at any distance.

Line 33, CALL EXEC (2, 8, BRIL), of the main program is a call to DOS-M, and it, in effect, instructs DOS-M ("2") to get the driver DVR37 associated with Equipment Table Entry 10 ("8") and to go to the entry point ("BRIL") for action. Now BRIL is Line 10 of subroutine BRIT, Fig. 3(c). The return address to the calling program is stored in BRIL, displacing NOP = 000000. The A-register is loaded with the number whose address is contained in IBRIT

(Line 5) and is output to DAC, a 12-bit converter located in I/O Channel 10 of the central processor. The DAC voltage sets the intensity of the CRO electron beam. Lines 36 and 37 put a zero voltage out to the z-axis of the CRO and (now that these two calls are used regularly) protects the phosphor from further burning.

Timing and output of the display is done in the same way by subroutine LUX [Fig. 3(b)]. Lines 5-13 store the duration of the display in location WAIT as the negative of the number of seconds specified. Negation is the two's-complement operation "CMA, INA" of Line 11: "Complement the value in A, add 1 to it." Line 14 is action, initiated by CALL EXEC (2, 8, LUX). Lines 15-28 are a timing loop. The time-base code for seconds (=D4) is given the time-base generator TBG (Lines 15-16), the clock (TBG) is started in Line 17, and the display is repeated for the duration stored in WAIT (Lines 17-28). Each time the buffer is displayed, the clock's flag is checked. Each time the flag is set (in this case, once each second), the value of WAIT is decreased by one. When the value of WAIT is zero, control is returned to the FORTRAN calling program at Line 29. The instruction of Line 27, ISZ WAIT, means "add 1 to WAIT and if the result is zero, skip the next instruction; otherwise, do the next instruction."

Several complete on-line experimental programs form prototypes and often can be easily adapted to other experiments. One such, FLIKØ, controls visual backward masking or metacontrast displays but can also display arbitrary sets of point arrays having general temporal relations with one another.

Another, THIN (Threshold in Noise), is a general calling program for obtaining temporal two-alternative forced-choice masked thresholds using the Wetherill UDTR adaptive psychophysical procedure (1966). Parameters read from the keyboard at run times control such factors as timing, sequences, and duration of signals, noises, warning lights, feedback events, disk-file name, initial intensity, stopping rule, and the like. A complete stimulus-response history may be printed out, as well as the estimate of threshold. As used in its present form, THIN selects randomly from arbitrary noises generated by SOUND and stored in a disk file.

The family of experiments controlled by such versatile programs as THIN use a wide variety of devices. As Fig. 1 shows, these devices include relay registers, digital-to-analog and analog-to-digital converters, time-base generators, preset clocks, counters, digital attenuators, voltage-control

generators, and active filters. Because of the precision and speed required of the instruments used in our experiments, we usually buy the best we can afford.

However, we make many devices—for instance, logic modules, active filters, digital attenuators, and amplitude window discriminators. We have made three digital-attenuator sets using fast reed or mercury relays. These have a range of 0 to 63.75 dB in .25-dB steps, and, by merely adding another stage, the range can be increased to 123.75 dB or the precision can be increased to .125-dB steps, leaving the range at 63.75 dB. Their precision exceeds that of the standard Hewlett-Packard attenuator sets, and settling times are well under a millisecond.

REFERENCES

- BELL, C. G., & NEWELL, A. *Computer structures*. New York: McGraw-Hill, 1971.
- CARTERETTE, E. C., FRIEDMAN, M. P., & LOVELL, J. D. Mach bands in hearing. *Journal of the Acoustical Society of America*, 1969, 45, 986-998.
- CARTERETTE, E. C., & LOVELL, J. D. Auditory Mach bands under several types of filtering. Paper presented at the 12th Annual Meeting of the Psychonomic Society, St. Louis, Missouri, November 1971.
- COOLEY, J. W., & TUKEY, J. W. An algorithm for the machine calculation of complex Fourier series. *Mathematics of Computation*, 1965, 19, 297-301.
- HAMMING, R. W. *Numerical methods for scientists and engineers*. New York: McGraw-Hill, 1962.
- HEWLETT-PACKARD CO. *A pocket guide to Hewlett-Packard computers*. Cupertino, Calif: Hewlett-Packard, 1970.
- LOVELL, J. D., & CARTERETTE, E. C. Digital generation of acoustic stimuli. *Behavior Research Methods & Instrumentation*, 1972, in press.
- LOVELL, J. D., CARTERETTE, E. C., & BARNEBEY, A. Threshold shifts at the edge of noise bands. Paper presented at the 82nd Meeting of the Acoustical Society of America, Denver, Colorado, October 1971.
- LOVELL, J. D., CARTERETTE, E. C., & NAGEL, D. C. A digital simulation model of the basilar membrane. Paper presented at the 82nd Meeting of the Acoustical Society of America, Denver, Colorado, October 1971.
- NAGEL, D. C. Adaptive psychophysical procedures. Paper presented at the Workshop on Mathematical Models of perception, sponsored by the National Science Foundation, Miami Beach, Florida, August 1970.
- SPERLING, G. The description and luminous calibration of cathode ray oscilloscope visual displays. *Behavior Research Methods & Instrumentation*, 1971, 3, 148-151.
- TAUB, A. H. (Ed.) J. von Neumann, *Collected works*, Vol. 5. New York: Macmillan, 1963, P. 34ff.
- TOOTILL, J. P. R., ROBINSON, W. D., & ADAMS, A. G. The runs up-and-down performance of Tausworthe pseudo-random number generators. *Journal of the Association of Computing Machinery*, 1971, 18, 399.
- WETHERILL, G. B. *Sequential methods in statistics*. New York: Wiley, 1966, 1966.

NOTES

1. See *A pocket guide to Hewlett-Packard*

```

a. 0001 FTN
    0002 PROGRAM VENUS
    0003 *
    0004 *
    0026 WRITE(6,1111)
    0027 1111 FORMAT("SECONDS AWAIT?")
    0028 READ(1,*)ISEC
    0029 WRITE(6,707)
    0030 707 FORMAT("HOW BRIT?")
    0031 READ(1,*)LUMEN
    0032 CALL BRIT(LUMEN)
    0033 CALL EXEC(2,8,BRIL)
    0034 CALL FLUX(ISEC)
    0035 999 CALL EXEC(2,8,LUX)
    0036 CALL BRIT(4000)
    0037 CALL EXEC(2,8,BRIL)
    0038 GO TO 1
    0039 END
    **** LIST END ****

b. 0001 ASMB
    0002 NAM LIJX,7
    0003 COM YY(1200)
    0004 ENT LIJX,FLUX
    0005 EXT .ENTR
    0006 ILIJX NOP
    0007 FLUX NOP
    0008 JSR .ENTR
    0009 DEF ILUX
    0010 LDA ILUX,I
    0011 CMA,INA
    0012 STA WAIT
    0013 JMP FLUX,I
    0014 LIJX NOP
    0015 LDA =D4
    0016 OTA TBG
    0017 STC TBG,C
    0018 LWAIT LDB LOC
    0019 SCENE LDA I,I
    0020 OTA 15B
    0021 INB
    0022 SZA
    0023 JMP SCENE
    0024 SFS TBG
    0025 JMP LWAIT
    0026 CLF TBG
    0027 ISZ WAIT
    0028 JMP LWAIT
    0029 JMP LIJX,I
    0030 LOC DEF YY
    0031 WAIT NOP
    0032 TBG EQU 12B
    0033 END
    **** LIST END ****

c. 0001 ASMB
    0002 NAM BRIT,7
    0003 ENT BRIT,BRIL
    0004 EXT .ENTR
    0005 IRRIT NOP
    0006 BRIT NOP
    0007 JSR .ENTR
    0008 DEF IRRIT
    0009 JMP BRIT,I
    0010 BRIL NOP
    0011 LDA IRRIT,I
    0012 ALF
    0013 OTA DAC
    0014 STC DAC
    0015 CLC DAC
    0016 JMP BRIL,I
    0017 DAC EQU 10B
    0018 END
    **** LIST END ****

d. 0001 ASMB,R,B,L
    0002 NAM DVR37,0
    0003 ENT 1.37,C.37
    0004 1.37 NOP
    0005 LDB EQT10,I
    0006 JSR 1,I
    0007 LDA =B4
    0008 JMP 1.37,I
    0009 C.37 NOP
    0010 CLA
    0011 JMP C.37,I
    0012 EQT10 EQU 214B
    0013 END
    **** LIST END ****

```

Fig. 3. Fragment of a FORTRAN main program (a), assembler subprograms for timing and display (b), and electron-beam intensity (c). Dummy driver DRV37 (d) is shown in its entirety. This set of programs is used for the luminous calibration of a cathode-ray oscilloscope.

computers. Cupertino, Calif: Hewlett-Packard Company, 1970. The company's first minicomputers were delivered about 1967. In 1971, the HP 2100 was introduced with virtually identical hardware and completely compatible software. The essential advances were a decrease in cycle time from 1.6 microsec to 980 nsec, an integral read-only-memory microprocessor for controlling execution of and integral hardware for extended arithmetic (multiplication, division, double load, double store, long shifts, and rotations). In November 1971, Hewlett-Packard announced the System/3000 as "the first small-scale computer system with true multiprogramming and multilingual capabilities." It is a sophisticated machine, having inherent reentrant code, dynamic

storage allocation, relative addressing, virtual memory, and allowing concurrent input/output and central processing. But it is by no means a minicomputer, since the smallest model at 32K bytes costs \$100,000.

2. A driver is a routine coded in Assembly language. Its purpose is to manage one or more I/O channels. It is a part of the operating system. For example, the familiar FORTRAN directive WRITE (6, 999) calls up the driver that controls Unit 6, teleprinter, or line printer, as output is written according to Format Statement 999.

3. Our version of the 2116B has several options which become essential for high-speed interaction between disk, CP, and I/O such as DMA and Memory Parity Check. Memory protect secures a selected block (or all) of memory against alteration

by memory reference instructions, giving a program in memory exclusive control of the I/O. Since the DOS-M Disk Manager and Job Processor are always in core and protect Locations 1-0100, it was necessary to have a device for wresting I/O control from DOS-M.

4. For many purposes, this kind of pseudorandom generator will serve. However, for simulation work bias and failure of runs up and down to be random

may arise. For simulation we use linear shift-register methods (on related Tausworthe generators, see Tootill, Robinson, & Adams, 1971).

5. It is actually the address of a parameter that is stored in IBRIT and ILUX. The instruction of Line 11(c) LDA IBRIT,I loads the A register indirectly, that is, with the contents of the address stored in IBRIT.

6. Note 2 applies also to ILUX [see (b) Line 10].

peripheral to the computer have their own interrupt circuits and are plugged into a hard-wired bus, in which their location establishes their interrupt priority. Changing priorities is simply a matter of moving the interface cards around.

We have response levers and a Scientific Prototype tachistoscope, in addition to the above, hooked up to the computer system.

COGLAB: A computer system designed for human research*

RICHARD L. TAYLOR

Memorial University of Newfoundland, St. John's, Newfoundland, Canada

The COGLAB system, consisting of a modular on-line Executive program with equally modular utility routines, was designed to drive a large fast phosphor CRT as an n-channel tachistoscope and to service a human observer whose task during a session is to process displayed information. It is currently operational on a Hewlett-Packard 2114 computer in my information-processing laboratory.

The objectives in developing COGLAB were: (1) to permit studies requiring tachistoscopic timing but which would not be constrained during conception by a fixed number of available stimulus "channels," (2) to permit the presentation of both static and dynamic visual stimuli such as, for example, alphanumeric symbols and rotating two-dimensional projections of three-dimensional solids, (3) to permit both conventionally conceived studies in which values on stimulus parameters are predetermined before an experiment occurs, as well as less frequently employed paradigms in which important parameters are systematically varied, contingent on a S's performance during a task (an example here being the attempt to hold d' constant within and across sessions in a visual detection task by manipulating stimulus exposure duration, and (4) to enable experiments to proceed either fully S-paced, where the S is able to run as many trials for as long and as rapidly as he can without fatigue, or else to proceed under precise timing controls throughout a session. This paper describes my solutions to these four objectives and outlines some general principles which are important for similar systems.

*This project is supported by the National Research Council of Canada through Operating Grant A-7811 and Special Equipment Grant E-3329 to the author.

EQUIPMENT

The hardware for COGLAB consists of: a Hewlett-Packard Model 2114 computer with 8K of 16-bit-word core memory; an ASR-33 Teletype; a 600-character/sec paper-tape reader; a programmable 10-kHz crystal clock; an 8 x 10 in. CRT display subsystem equipped with a P31 aluminized phosphor, which decays to 10% of its peak luminance in less than 1 msec for the levels employed; and a 16-bit general-purpose interface card that is connected to a rack of solid-state BRS modular logic. All of the devices

THE SOFTWARE

Each device that is driven by the computer has one or more interrupt drivers, which are subroutines designed to accomplish specific tasks. Each driver has one or more entry points which can be accessed by a simple FORTRAN subroutine call which specifies the name of the entry point and the parameters, if any, requiring transfer. For example, the 16-bit general-purpose interface card has an interrupt flip-flop which is connected through the solid-state external logic to levers available to a S such that, when a lever is pressed, the interface-interrupt flag is set. If one wishes to know when a S presses a lever, it is necessary only to include "CALL RYSET" in the proper place of the Executive program. When a response occurs, the subroutine RYSET, in addition to interrupt housekeeping chores, stores a constant

Table 1

PROGRAM LAB 14	
COMMON JJ (832), S (64)	
CALL NREAD	; Reads in pattern library
MSA = 3000	
MSB = 5000	
CALL CLOCK (1)	; Sets up session timer for ½ hr
9 CALL READY	; Displays the word "READY?" and waits for S to respond
N = N + 1	
CALL RESP (K1, K2, K3)	; Selects three digits as stimuli
CALL MESS (2, 0, 0, 0)	; Displays markers on screen where the key stimuli will appear later
CALL WAIT (MSA)	
IF (1 - K3) 10, 10, 11	
10 K2 = K3	
11 CALL MESS (3, N1, N2, 0)	; Displays words describing digits to appear
CALL STCLK (1)	; Starts millisecond timer
CALL PULL (K1, K2)	; Extracts digit patterns from library
12 CALL DIGIT (JA, JB)	; Selects one of four scanning directions
IF (5 - JA) 12, 12, 13	
13 CALL DIGIT (JD, JB)	; Selects one of four transformations
IF (5 - JA) 13, 13, 14	
14 CALL NFIND (JA, JB)	; Rearranges stored digit patterns
CALL RDCLK (NR)	; Reads current value of millisecond timer
MSB = MSB - NR	; Subtracts time for above from programmed delay
CALL WAIT (MSB)	
CALL MESS (4, 0, 0, 0)	; Displays actual digits selected above
CALL TRIAL (K3, JRT)	; Returns with error marker or S's RT
IF (1 - JRT) 16, 15, 16	; Tests to see if S made an error
15 CALL MESS (5, 0, 0, 0)	; Displays "ERROR" message on scope
CALL WAIT (MSA)	
GO TO 20	
16 CALL MESS (6, 0, 0, JRT)	; Displays an S's correct RT on screen
CALL WAIT (MSB)	
20 CALL DATUM (N, JRT, JA, JB, K3)	; Processes the data for this trial
CALL CLOCK (2)	; Tests to determine if S needs a rest
CALL TADJ (MSA, MSB, JRT)	; Resets time interval constants according to S's RT for this trial
GO TO 9	
END	