CENTRAL RESEARCH LIBRARY
DOCUMENT COLLECTION

OAK RIDGE NATIONAL LABORATORY
operated by
UNION CARBIDE CORPORATION
NUCLEAR DIVISION
for the
U.S. ATOMIC ENERGY COMMISSION

LOCKHEED MARTIN ENERGY RESEARCH LIBRARIES
3 4456 0513477 0

ORNL-TM-2013
COPY NO. - 57
DATE - Oct. 4, 1967

Neutron Physics Division

A PDP-8 INTERFACE FOR A CHARGED-PARTICLE NUCLEAR PHYSICS EXPERIMENT

W. R. Burrus, E. Madden,* C. O. McNew,* and R. W. Peelle
(Written by R. W. Peelle)

DOCUMENT COLLECTION
LIBRARY LOAN COPY
DO NOT TRANSFER TO ANOTHER PERSON
If you wish someone else to see this document, send in name with document and the library will arrange a loan.

ABSTRACT

An interface has been constructed to use a PDP-8 computer with a charged particle detector system employing three solid-state detectors and flight-time analysis. Up to 48 bits from each randomly-occurring event are transferred through the data (break) channel to a hardware-selected buffer region in the core of a PDP-8 computer. Designed for use as a magnetic tape analyzer for the most complex cases, the system assumes that the 48 bits originate in flag bits set by fast logic and in (presently four) amplitude digitizers, all of which are assumed to contain information for the same event. The system includes some limited capability for controlling the course of the experiment, and provides for readout through the computer of a series of external fast counters. This report summarizes the design concepts, shows schematic flow diagrams, defines the computer instructions associated with the interface system, and gives simple model programs to illustrate methods of application.

Research partially supported by the
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
under contract R-104(1)

*Instrumentation and Controls Division.

NOTICE This document contains information of a preliminary nature and was prepared primarily for internal use at the Oak Ridge National Laboratory. It is subject to revision or correction and therefore does not represent a final report.

LEGAL NOTICE

This report was prepared as an account of Government sponsored work. Neither the United States, nor the Commission, nor any person acting on behalf of the Commission:

A. Makes any warranty or representation, expressed or implied, with respect to the accuracy, completeness, or usefulness of the information contained in this report, or that the use of any information, apparatus, method, or process disclosed in this report may not infringe privately owned rights; or

B. Assumes any liabilities with respect to the use of, or for damages resulting from the use of any information, apparatus, method, or process disclosed in this report.

As used in the above, "person acting on behalf of the Commission" includes any employee or contractor of the Commission, or employee of such contractor, to the extent that such employee or contractor of the Commission, or employee of such contractor prepares, disseminates, or provides access to, any information pursuant to his employment or contract with the Commission, or his employment with such contractor.

TABLE OF CONTENTS

		Page No.
I.	Introduction	4
II.	Overall System Configuration	5
III.	The Experimental Data Interface	7
	A. The Data Break Control	8
	B. Pulse-Height Analyzer (or other data) Logic Adaptor	11
IV.	Experiment and Display Control	13
	A. The Hardware Clock	13
	B. External Counter Readout	15
	C. Output Levels and Pulses	18
	D. Decimal Thumbwheel and Push-Switch Interrupts	19
	E. The Display System and Light Pen	22

LOCKHEED MARTIN ENERGY RESEARCH LIBRARIES

3 4456 0513477 0

I. INTRODUCTION

The hardware system described in this report was intended to be useful in conjunction with typical nuclear experiments, but the design was predicated almost completely on a single experiment.[1]

In this experiment charged particles from nuclear reactions of 40-60 MeV protons or α-particles are detected in a telescope consisting of three semiconductor detectors. Particles strike the counter stack at random rates up to 1000 per second. The time of arrival of each particle detected in the thinnest (100 micron) detector is compared to a timing signal derived from the cyclotron accelerating voltage to produce a pulse from a time-to-amplitude converter, and charge pulses from each detector are measured. Since all particle ranges are of interest, the amplitudes of four 1-microsec signals must be digitized unless complex linear gating and summing circuitry is employed. Summing circuits were altogether avoided by summing the absorbed energies in the computer. Fast logic decisions are made outside the equipment discussed here to determine the counter in which the particle stopped, whether the event in question was compromised by pileup, etc.

The system was designed to serve the above experiment by employing a PDP-8 computer[2] as a magnetic tape analyzer with event-by-event recording on IBM-compatible magnetic tape of the ~48 bits corresponding to each event. Somewhat similar computer configurations have surely been used elsewhere. This choice was dictated by the large size of the arrays needed to represent the three two-dimensional systems of pulse amplitude combinations required for identification of the particle type without losing the advantage of the 0.3 - 0.5% resolution inherent in the detector system. For the simpler experiments involved in detector calibration and lineup, the system allows on-line single-dimension pulse-height analysis. The computer also provides limited control over the course of the experiment, and provides for on-line low precision partial analysis of results. The system could have been programmed for on-line rather than delayed-time analysis, with good efficiency for the experimenter but with less efficiency in (scarce) cyclotron time.

The most important stipulation in the design which restricts the diverse application of the system is the one which requires that all the amplitude digitizers respond to coincident events. That is, the four digitizers connected to the system cannot now be employed with detectors sensing uncorrelated radiations unless some additional work is done on triggering and dead-time arrangements.

The primary purpose of this report is to provide a manual for users of the system, but the material may interest persons concerned with system design.

II. OVERALL SYSTEM CONFIGURATION

Figure 1 represents the hardware configuration employed. The Type 580 tape system is a minimum IBM-compatible design which performs all its data transfers through the computer's arithmetic register, so that tape reading or writing generally forecloses other program activity. Since fast paper tape equipment is not provided, the magnetic tape needs to be used for program storage as well as data dumping. This is accomplished through the use of "PUT" and "TAKE" routines described by Rust and Burrus.[2] Appendix A indicates the simple but important modifications made to the 580 tape system.

Data from randomly-occurring nuclear events enter the computer memory through the cycle-stealing data break. Data are stored in a single end-around buffer region within the computer memory, with hardware-controlled storage address. Six microsec of computer time are required for each 4-word (48 bit) transfer. A hardware up-down counter maintains a record of the number of words of unprocessed data in the buffer, and the program maintains cognizance of the situation by noting where data has been entered or that the up-down counter is full. The simulator replaces real data with a switch register for program and hardware testing.

Display facilities are based on a standard DEC Type 34 display system with a 8- x 10-cm screen and three intensity levels. The Moseley

ORNL DWG 67-11229

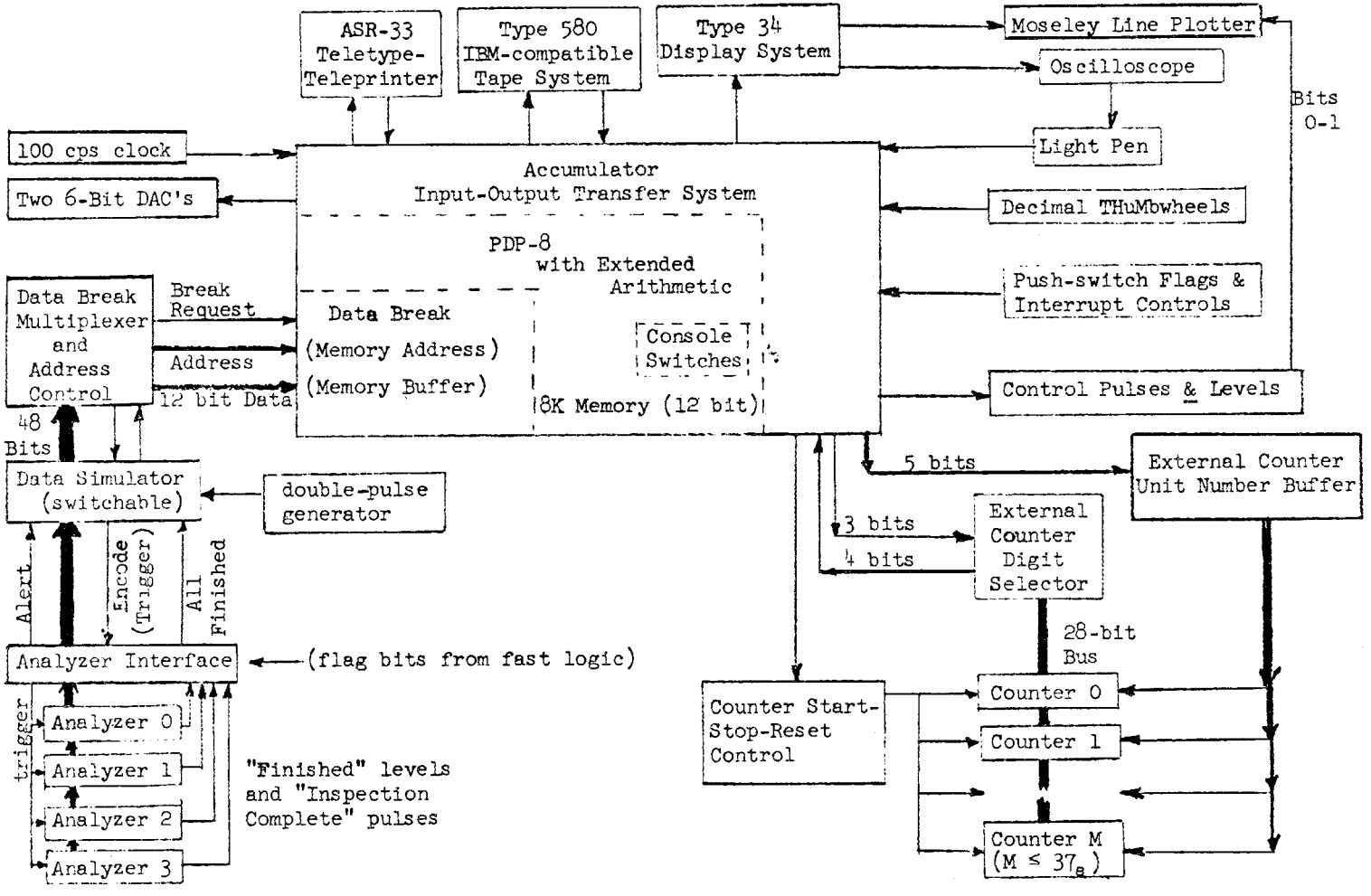


Figure 1. Computer and Interface Configuration

plotter is simply wired in parallel with the oscilloscope, though of course the programming is special. The "light pen" is a home-made unit with standard operating characteristics except that the moveable head, the case from a ball-point pen, is very light and has no shutter.

Operator interaction with the program, and therefore with the experiment, can take place through the teletype, the light pen, and the 12 bits of the console switch register. Two three- (decimal) digit thumbwheel switches are also provided, which may be read into the computer's accumulator on program demand. These are more convenient than the console switches for transfer of decimal numbers. Finally, five flags (toggles) are provided which allow the operator to signal specific conditions or requests to the program by means of pulses or push switches.

For reading into the computer the contents of external fast counters, provision is made to accept into the computer's accumulator four bits from the selected decimal digit of any one of 32 external counters.

Experiment control equipment, included to allow for rapid expansion of capability, includes a one-word program-controlled buffer with six available level outputs, three program-controlled pulse outputs, and two six-bit digital-to-analog converters which may be set via the computer's accumulator.

Detailed wiring prints of the locally-provided portion of the interface system are available in the Q-2929 series of the Oak Ridge National Laboratory Instrumentation and Controls Division. ["A Multi-Analyzer Stored Program Data Acquisition System".] Appendix B lists the computer instructions defined for the completed system.

III. THE EXPERIMENTAL DATA INTERFACE

As indicated in Fig. 1, the pulse-height analysis data (along with flags from the fast logic) enter the computer through an interface divided into two sections, one of which is essentially a control and multiplexer for the data break, and the other of which is a logic adaptor for four pulse-height analyzers. Interposed is a set of simulator switches which allow analyzer data bits to be set with a 48-bit switch

register and which provides dummy (adjustable) analysis delays. This simulator is invaluable for program testing. (To be optimally useful, such a device should also simulate the computer to the analyzers.) The sections below describe the two main segments of the data interface.

A. The Data Break Control

The data break control system (DEC series 100330) was designed by D. Vanada of Digital Equipment to meet our general specifications, and has undergone only minor revision to improve compatibility with external hardware. The ("single cycle") data break system employed has as its purpose the parallel deposition of a 12-bit word from an external source into the computer memory location determined by the external hardware of the Control, without interference with the program. Such transfers require 1 machine cycle (1.5 microsec) per 12-bit word. In the present case the Data Break Control is designed to perform the functions illustrated in Fig. 2 and described in the paragraphs below.

Provided that the resulting data break would not write over unanalyzed data in the computer memory, the Control gives the computer a a Break Request level when an appropriate event is ready for analysis, when data storage has been enabled both by hardware and software controls, and when the pulse-height analyzers have completed encoding (or rejection) of all signals relevant to that event. The Break Request is terminated when enough successive break cycles have been taken to transfer the number of words (per event) determined by hardware switches.

The Control multiplexes 12 to 48 bits into successive 12-bit words in synchronization with the PDP-8 timing pulses; i.e., successive 12-bit level combinations are made available to the computer's Memory Buffer register.

The Control determines the 12-bit address for each transfer, and makes it available to the Memory Address Buffer at the proper time. Data words are written into a region of memory of length 256 (400_8) or 512 (1000_8) words long, as determined by a hardware switch. Each word from the multiplexer is placed in the next successive core location, with end-around carry. (The program does not automatically have cognizance of the location of the hardware "pointer".) The 8 or 9 less-significant

ORNL DWG 67-11230

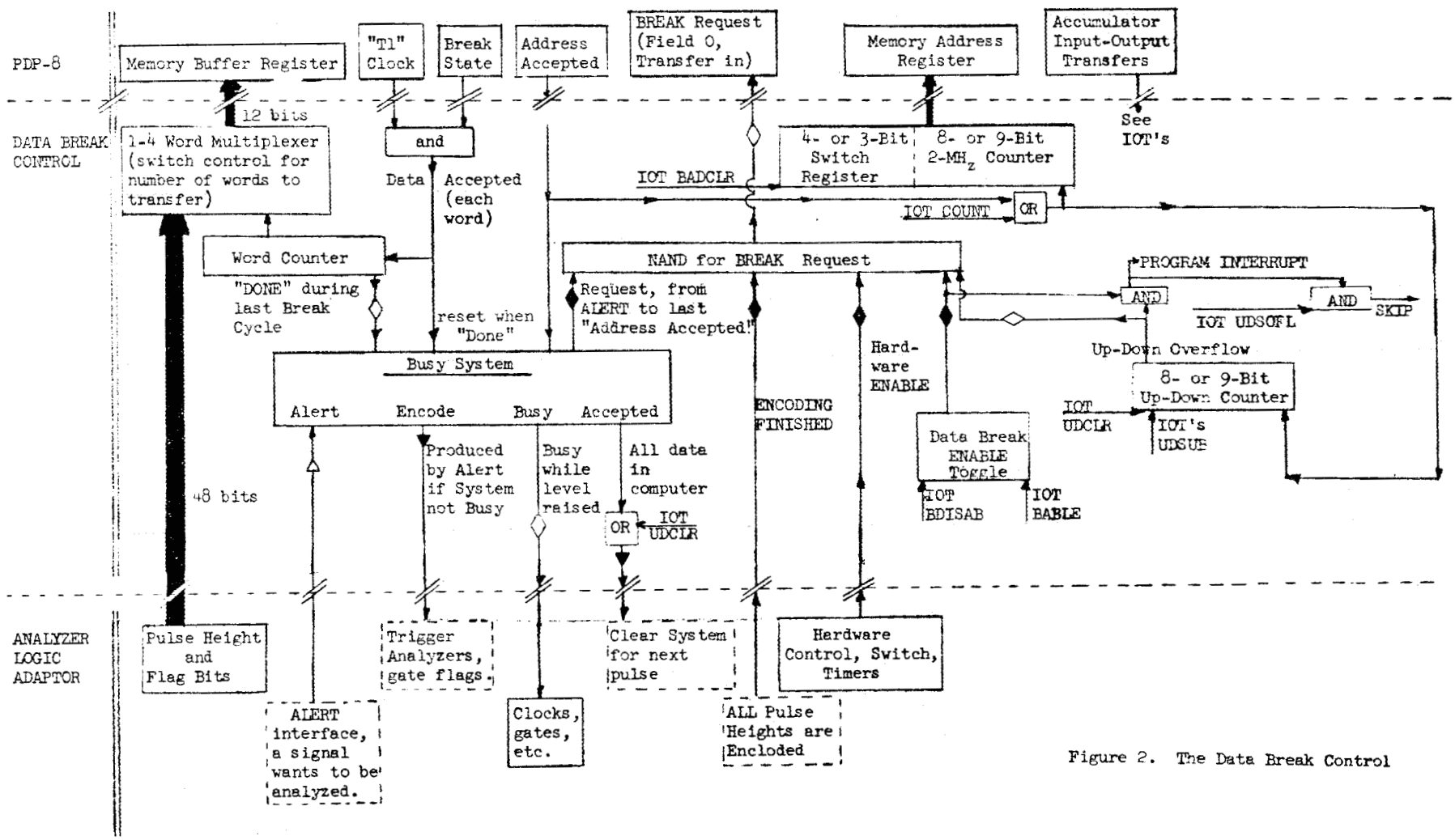


Figure 2. The Data Break Control

bits of the break address are determined by a binary counter, while the 4 or 3 more significant bits are determined by hardware switches. All transfers are now into memory field 0.

The Control signals the program when the data buffer has no room for additional data by use of the overflow level from an up-down counter which is incremented by the hardware each time the break address is incremented, and is decremented by the program when a data word has been handled so that the corresponding memory location can be re-used. (The program inspects and then clears a fixed bit position in a word from each event to avoid re-analysis of the same event.) Having a fixed bit which was wired in proved to be a great aid in checkout when it was not clear whether troubles were due to hardware or software. Since the UDC is a 2 MHz counter, "up" and "down" counts presumably should not occur within 0.4 μsec of each other, so present programs disable the data break for 10 microsec to perform the count-down operation. Analysis of the break timing implies that this precaution is unnecessary if the UDSUB3 instruction is avoided. While the UDC is in overflow, no more data breaks can occur. If the data break is enabled, the UDC overflow is a level which causes program interrupt and which is available for testing with a skip instruction.

The Control accepts ALERT signals corresponding to analyzable events, and determines which of them occur when the system is ready for handling a new pulse. These produce 70 nsec ENCODE pulses which are returned to the Analyzer Logic Adaptor.

When the last data word from a given event is in the computer memory, an ACCEPTED signal is issued to the Analyzer Logic Adaptor to allow preparation for the next appropriate signal.

A BUSY level is produced which stretches from any ENCODE until a tunable delay after the corresponding ACCEPTED. No additional ENCODE signals can be produced by the interface until the BUSY interval is concluded. The tunable delay has been set at 5 microsec, but the appropriate number depends on the Analyzer Logic Adaptor and the analyzers themselves, and should be checked by studies with random signals.

An ACCEPTED pulse is produced by the UDCLR instruction for initialization of the Analyzer Logic Adaptor. (The Data Break Control is similarly initialized by the BADCLR instruction.)

Three documented programs[4-6] are available for use with the Break Control, and a variety of others could be produced by adapting routines from these. While the system was based on the use of the end-around buffer, one-event or few-event buffers have been successfully used by depending upon receipt of an up-down counter overflow before counting down the UDC. In this manner the core data buffer becomes synchronized (and may be made any length), since the address count is reset or known each time the UDC is initialized after an overflow.

The following instructions are utilized by the Data Break Control.

6301	COUNT	Increment the address counter and the up-down counter.
6311	BADCLR	Break ADdress CLeaR, clears the address counter and initializes the Data Break Control.
6312	BABLE	Break enABLE to allow transfers to computer memory.
6314	BDISAB	Break DISABle so that no data break requests can be made.
6321	UDSOFL	Up Down counter, Skip on OverFLow, if the Data Enable toggle is set.
6324	UDCLR	Up Down counter, CLeaR. UDCLR also issues an ACCEPTED pulse to the Analyzer Logic Adaptor.
6331	UDSUB1	
6333	UDSUB2	
6337	UDSUB3	Up Down counter SUBtract 1, 2, or 3. The corresponding IOT pulses are simply OR'ed and applied to the "down" input of the UDC. Note that the last of the three UDSUB3 pulses occurs about 0.1 or 0.2 microsec before the Address Accepted pulse which is used to increment the UDC in case the following machine cycle is a data break.

B. Pulse-Height Analyzer (or other data) Logic Adaptor

The adaptor in use was designed by E. Madden and is described by the ORNL Q-2929 prints. The main functions are summarized in the paragraphs below.

The logic levels available from the data source are converted as necessary to match the multiplexer inputs of the Data Break Control. (-3v is logical unity, ground a zero.)

In the externally ALERTed mode generally used so far, triggers are produced for all digitizers when an ENCODE signal is received from the

Data Break Control. In the internally ALERTed mode, the Data Break Control is provided an ALERT pulse when an analyzer's inspection circuitry reports an input pulse.

In the internally ALERTed mode, all enabled analyzers are blocked so they cannot respond to additional signals until the BUSY interval is completed. (Some units perform this function internally.)

Switch controls are provided to determine which analyzers are enabled, i.e. from which analyzers will we await signals indicating completion of digitization. The ENCODING FINISHED level is produced when all enabled analyzers have responded.

When pulses timed with an ENCODE on an enabled analyzer are outside the range of amplitudes which can be digitized, the hardware sets the flag which asserts that the analyzer has completed its digitization. When such overflows or underflows are detected, a flag bit is transferred to the computer with the suspect data. (The analyzers have their "slope intercepts" adjusted so that an ENCODE unaccompanied by a signal will store a "baseline" amplitude of several channels.)

For each 12-bit data word, three of the bits are (hardware) switchable between input data lines and toggles for which pulse inputs are provided. That is, there are 12 such flag toggles. Arrangements have been included to assure that toggles cannot be set except by events which produce ENCODE signals in the Data Break Control.

When any step in the logic chain fails, this interface system "freezes", usually with the BUSY level set but no ENCODING FINISHED level forthcoming. Features are included in the design to aid trouble-shooting of such freezes. Freezes are rare indeed now that logical flaws have been removed from the interpretation of signals from the pulse analyzers. Our analyzers utilize some rather ambiguous logic signals, and one must take care not to have circuits which make two independent decisions as to whether a logic level is present or not. (One inside the analyzer and one in the interface.)

IV. EXPERIMENT AND DISPLAY CONTROL

This section describes all parts of the system except for the input through the data break of information from random events.

A. The Hardware Clock

Figure 3 represents the circuit associated with the 100 Hz hardware clock, which is serviced with a single instruction:

6322 = KKKS, Kludge Klock flag Klear if Skip.

A standard commercial 100 Hz oscillator drives once per cycle the Schmidt trigger which sets the clock flag. This flag gates the IOP (input-output pulse) corresponding to the KKKS instruction into the Skip Bus (grounding it), and drives the Program Interrupt if the corresponding hardware switch is enabled. When the interrupt is not used, some program loop must check the clock flag at appropriate intervals. Note that a pulse amplifier is used to assure that the flag is reset if and only if the skip bus is grounded. Otherwise the time measure cannot be as accurate as the oscillator.

Supposing that the program interrupt is off or the clock connection to it is disabled, the following subprogram might be called at convenient (100 microsec to 5 msec) intervals. KLOCK assumes that half a year is the longest period to be recorded, and that storage in 1 sec increments is appropriate.

```
KLØCK, 0                / Stores Time in 1 sec Units.  Does Not
                          Disturb AC.
       KKKS             / Test Clock Flag
       JMP I KLØCK      / Return, No Flag
       ISZ C            / Flag Found and Cleared
       JMP I KLØCK
       DCA TEMP         / Protect Accumulator
       TAD M100D
       DCA C            / Restore Counter Which Counted to 100
       ISZ SECIØ        / Counts Seconds
       JMP .+3
       ISZ SECHI        / Counts Overflows
       NØP
       TAD TEMP         / Restore Accumulator
       JMP I KLØCK      / Return
M100D, -144
```

ORNL DWG 67-11231

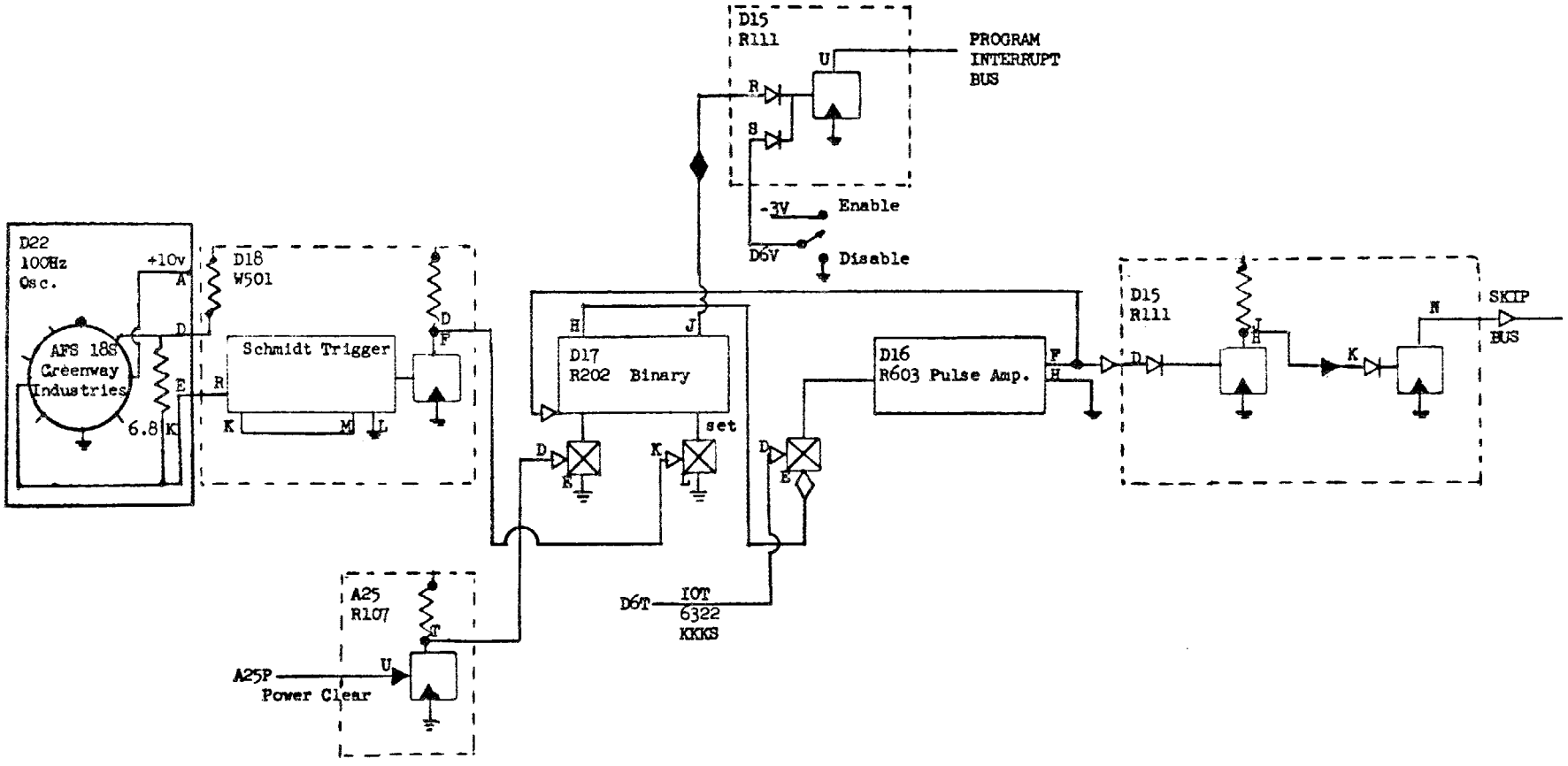


Figure 3. The Hardware Clock. The circuit card numbers generally refer to devices manufactured by the Digital Equipment Corp.

```
 TEMP, 0
SECLO, 0
SECHI, 0
    C, 0
```

A more practical program requires printout and initialization of SECLO and SECHI. A simple test program for KLØCK, which displays the time in seconds in octal in the extended AC-MQ arithmetic register, is given below:

```
        TAD M100D        / Test Program for KLØCK
        DCA C
        DCA SECLØ        / Reset Storage
        DCA SECHI
    L,  TAD M100D        /Set Delay Length
        DCA DELC
        JMS KLØCK
        TAD SECLØ
        MQL              / Load Display
        TAD SECHI
        ISZ DELC         / Delay for 600μsec
        JMP .-1
        JMP L
 DELC,  0
```

B. External Counter Readout

Figure 1 illustrates the system for readout of external counters. The units actually used are TSI 10 MHz counters with 7 decimal digits each, and the scheme is to bus the (28) BCD output lines from one counter to the next and finally to the counter control. The 5-bit counter Unit Number Buffer governs which counter unit affects the levels on the (28) bit lines. Similarly, the 3-bit Decade Number Buffer controls which digit of the selected counter unit is available to the computer. (The decades, and the corresponding decimal digits, are numbered from 1 through 7, with decade 1 being the most significant.) The selected decimal digit is read on command into the PDP-8 accumulator's bits 8-11. The Unit Number and Decade Number buffers can be cleared, loaded from the PDP-8 accumulator, or incremented. The contents of the selected decade can be sampled by the accumulator. The count units themselves can be started, stopped, or cleared. The following input-output instructions are available in the system, and the useful ones have been assigned mnemonics.

6341	CCCUB	Counter Control Clear Unit Buffer.
(6342)		OR the AC, bits 7-11, into the Unit Buffer, Clear AC.
6343	CCLUB	Counter Control Load Unit Buffer.--This instruction loads the Unit Buffer with 7-11 of the AC, and then clears the AC.
6361	CCIUB	Counter Control Increment Unit Buffer.
(6351)		Clears the Decade Buffer, but is presently useless by itself.
(6352)		OR's bits 0-2 of the AC into the Decade Buffer then clears AC.
6353	CCLDB	Counter Control Load Decade Buffer. This instruction loads the Decade Number Buffer with bits 0-2 of the AC, then clears the AC.
6354	CCIDB	Counter Control Increment Decade Buffer.
6355	CCSDB	Counter Control Set Decade Buffer. (to 001)
		- - - - - -
6344	CCLDA	Counter Control Load Digit to Accumulator. This is not a true load instruction, but OR's the decimal digit (four bits) from the selected decade of the selected counter unit into bits 8-11 of the PDP-8 accumulator. CCLDA must not be microprogrammed with 6341 or 6342 because of the time required for charging the data lines.
		- - - - - -
6371	CCSTRT	Counter Control STaRT, all external counters.
6372	CCSTØP	Counter Control STOP, all external counters.
6374		(spare)
6401	CCRZ	Counter Control Reset counter Zero.
(6402)		Resets all counters except counter zero.
6403	CCRALL	Counter Control Reset ALL counters.
6404	CCRREG	Counter Control Reset REGisters.

Note that the contents of the selected decade of the selected counter unit is always available to the AC without delay. This feature is convenient to preset counting problems.

For a sample program, suppose that there is no need to decode the contents of the counter units to binary or to store them, but only the need to print them out sequentially on the teleprinter. The following sample program can accomplish this limited objective, and does so after external counter 0 reaches the preset number originally read in through the switch register. TTYN is assumed to be a teletype routine which adds a base of 0260 to the number in the accumulator, prints, and clears

the AC. LINEF is a subroutine which produces the carriage return and line feed combination, and leaves the AC cleared.

```
        LAS             / switch register contains number of units
                          to print.
        CIA
        DCA UC          / counter for printout is prepared.
        HLT
        LAS             / when bits 8-11 of SR = contents of the unit
                          zero decade given by bits 0-2 of SR, the
                          program will stop data and start print-
                          out.
        CCLDB           / load decade number for preset count.
        LAS
        AND M0017
        CIA
        DCA PRESET      / comparison for preset count.
        CCCUB           / clear unit buffer.
        CCRREG CCRALL   / reset all counts and registers.
        JMS DBINIT      / call data break initialization routine.
        CCSTRT          / start count.
        BABLE
AWAY,   JMS LØØP        / LØØP subroutine produces delay of suitable
                          length (1 ms) and perhaps processes ana-
                          lyzer data obtained during a run.
        JMS KLØCK       / 100 cps clock routine (see sect. A).
        CLA
        CCLDA           / bring digit from selected decade of unit
                          0.
        TAD PRESET
        SZA CLA         / skip if we have hit preset counts.
        JMP AWAY
        CCSTØP          / run is over! Print out counter contents.
        BDISAB
   L,   CCSDB           / set to read first digit.
        JMS LINEF       / carriage return and line feed.
        TAD N7
        DCA DC          / set decade counter.
  LL,   CCLDA           / selected digit in accumulator.
        JMS TTYN        / type digit.
        CCIDB
        ISZ DC          / done with one unit?
        JMP LL          / no.
        CCIUB           / yes, go to next unit.
        ISZ UC          / printed enough units?
        JMP L           / no.
        JMS LINEF       / yes.
        TAD SECLØ       / displays Klock time.
        MQL
        TAD SECHI
        HLT
```

```
    UC, 0
    DC, 0
    N7, -7
 M0017, 0017
PRESET, 0
```

C. Output Levels and Pulses

For general purpose interaction between the computer and experimental hardware, a few program-controlled pulses, logic levels, and analog levels are provided; serviced by the instructions below.

6302	DACCLR	Digital-to-Analog Converter CLeaR. Clears the buffer registers for two six-bit converters. When the buffers are cleared the DAC's each go to -10v.
(6304)		The contents of accumulator bits 0-5 are OR'ed into the buffer of the first DAC, and contents of accumulator 6-11 are OR'ed into the buffer of the second. Clears the AC.
6306	DACLD	Digital-to-Analog Converters, LoaD from accumulator. Clears the AC.
6411	FPIØT 10	First, second, and third Front Panel IOT's
6421	FPIØT 20	respectively. These are negative 3v, 1 μsec
6431	FPIØT 30	long (W640) pulse with a positive overload. They may be shortened to 400 nsec, or to 70 nsec by substituting a (W607) module.
6362	CBC	Control Buffer Clear, clears the 6-bit (expandable to 12 bits) control buffer which makes levels available at the front panel.
(6364)		OR's the accumulator into the control buffer, then clears the AC.
6366	CBLD	Control Buffer LoaD. This clears the control buffer, loads it from the accumulator, and then clears the accumulator.

Good application of these external signals and levels will depend on the problem. The DAC's were thought useful, for instance, for control of gain or zero shift. Note that in addition to the front panel IOT's listed above, instructions 6374 and 6404 are currently not used.

The Control Buffer can have 12 bits, though only 6 (bits 6-11) are now externally available on level output 6 through 1, respectively. Bits 0-1 of the AC load the Moseley Pen Control as part of the Control Buffer. See sect. IV. E.

The following test program images the accumulator switches into the control buffer, issues successive front panel IOT pulses, and produces stairsteps out of the two DAC's which run counter-current to each other.

```
        DCA DACHI        / clear high and low order DAC buffer images.
        DCA DACLØ
      S, LAS             / take bits from SR for control buffer.
        FPIØT 10
        FPIØT 20
        FPIØT 30
        CBLD             / should leave AC cleared.
        SZA              / see if it did.
        HLT              / error stop.
        TAD DACHI
        TAD N100
        DCA DACHI
        ISZ DACLØ
        NØP
        TAD DACLØ
        AND P0077
        TAD DACHI
        DACLD            / puts AC into DAC buffers.
        ISZ C; JMP .-1
        JMP S
      C, 0
  P0077, 0077
   N100, -100
  DACHI, 0               / image of high-order DAC buffer.
  DACLØ, 0               / image of low-order DAC buffer.
```

D. Decimal THuMbwheels and Push-Switch Interrupts

Two three-digit decimal thumbwheel switches are provided at the console, to simplify transfer of decimal numbers to the accumulator when the teletype is not convenient to use. Standard 1-2-4-8 weighting is employed. This leads to the following instructions:

```
6412   THMØR 10        / Logical OR the contents of THuMbwheel
                         switch 1, in BCD, to bits 0-11 of the
                         accumulator.
6422   THMØR 20        / Logical OR the contents of THuMbwheel
                         switch 2, in BCD, to bits 0-11 of the
                         accumulator.
```

Access is provided at the console to set five separate "push-switch flags" which may be tested by SKIP instructions and individually cleared

under program control. Any set flag causes a program interrupt unless that interrupt is disabled by the manual switch provided. Flags 1-4 may be set by mechanical push switches, and flags 2-5 may be set by -2.5 v pulses (>100 nsec long). The setting of flag 5 can be enabled or disabled by program control, and is disabled by the power clear pulse which occurs when the PDP-8 is started. All flags are reset by the power clear pulse. Push switches 1 and 2 are physically associated with the corresponding thumbwheels, to call attention to new values placed there, but there is no hardware connection between switches and thumbwheels so the programmer may use them as he wills. The software enabling of flag 5 may be used to allow an external signal to control program logic as conditioned by the program. A typical use might be to await the start of an experimental run until receipt of the first pulse from a current integrator which happens to occur after program initialization.

The following instructions apply.

6404	PSCF MO	Push Switch Clear Flag M, i.e. PSCF 50 clears flag 5.
6441	PS1SNF	Push Switch 1, Skip No Flag
6442	PS2SNF	Push Switch 2, Skip No Flag
6461	PS3SNF	Push Switch 3, Skip No Flag
6462	PS4SNF	Push Switch 4, Skip No Flag
6432	PS5SNF	Push Switch 5, Skip No Flag
6451	PS5DIS	Push Switch 5, DISable pulse input
6452	PS5EN	Push Switch 5, ENable pulse input.

The following routine tests the skip-no-flag instructions associated with the push switches (and any others listed in the instruction buffer which starts at 261), displays the raw data from either THuMBwheel in the MQ register, and prints out the number of any flag which it finds set. If the flag is discovered through the program interrupt a "P" preceeds the printed number. Pressing "5" on the teletype enables setting of flag 5, while pressing any other key causes a PS5DIS instruction to be issued.

```
*1400                   / test routine for THuMBwheel switches, push
                          switches, and other SNF instructions.
        UDCLR
        JMS CRLF        / resets teletype carriage.
BEGIN,  LAS             / SR(0) = 0 to bring THuMBwheel 1 contents
                          to MQ.
```

```
        IØN                 / program interrupt on.
        RAL                 / test bit into link.
        CLA
        TAD THMØRP
        SZL
        TAD P10
        DCA .+1
        0                   / gets an instruction to bring selected
                              THuMBwheel to AC.
        MQL                 / and displays bits in MQ.
        UDCLR               / use pulse to set Push Switch 5, if desired.
        ISZ INP
        JMP .-1             / delay to catch interrupt signal.
        IØF
        JMS CHECK
        JMP BEGIN
 CHECK, 0                   / routine to check flags.
        TAD INBUFF
        DCA INP             / set pointer in instruction buffer.
        TAD PSCFP
        DCA CLR
     L, TAD I INP           / bring first instruction.
        SNA
        JMP I CHECK         / we are through buffer of instructions.
        DCA .+1
        0                   / gets successive skip instructions.
        TAD INP             / no skip, there was a flag.
        JMP NØRMAL          / AC clear if no flag.
        JMS TYPE            / a double skip! Print instruction number.
        TAD D               / double skip label.
   PRN, JMS TYPE
        JMS CRLF
   CLR, 0
NØRMAL, SZA                 / no print if normal skip.
        JMP PRN
        ISZ INP
        TAD P10
        TAD CLR
        DCA CLR             / set instruction to clear flag.
        JMP L               / return for next test.
     D, 304
THMØRP, THMØR 10
   P10, 10
 PSCFP, PSCF 10
   INP, 0
INBUFF, 261
    *0
  ZERØ, 0
        JMP RUP
   *20                      / interrupt routine.
   RUP, DCA SAVE
        KSF
```

```
        SKP
        JMP TEST5
        TSF
        JMP .+2
        JMP ØUT
        TAD P
        JMS TYPE
        JMS I CHECKP
   ØUT, TCF
        NØP
        NØP
   RET, TAD SAVE
        IØN
        JMP I ZERO
  SAVE, 0
     P, 0320
CHECKP, CHECK
  TYPE, 0                 / subroutine for teletype
        TLS
        TSF
        JMP .-1
        CLA
        JMP I TYPE
  CRLF, 0                 / subroutine to give carriage return line
                            feed combination
        TAD CR
        JMS TYPE
        TAD LF
        JMS TYPE
        JMP I CRLF
 TEST5, KRB
        TAD MASK
        PS5EN
        SNA CLA
        JMP RET
        PS5DIS
        JMP RET
  MASK, -265
    CR, 215
    LF, 212
  *261
        PS1SNF            / list of test instructions.
        PS2SNF
        PS3SNF
        PS4SNF
        PS5SNF
        0
```

E. <u>The Display System and Light Pen</u>

The computer was delivered with a standard DEC Type 34B display

system, but adjustments and improvements have been made which require documentation. Ten-bit digital-to-analog converters are loaded from the less significant bits of the accumulator, and three intensity levels are available on a standard Tektronix RM503 oscilloscope with a P-7 phosphor by use of intensify pulses of various lengths. We have used a non-standard light pen, but it operates logically through the standard system. We have used the analog outputs of the DAC's to drive a standard Moseley Type 2D-2 x-y recorder in the pen mode, with programmed delays to give a rapid and accurate histogram plot.

Figure 4 shows the slightly-modified display system in schematic form. The modifications from standard are below:

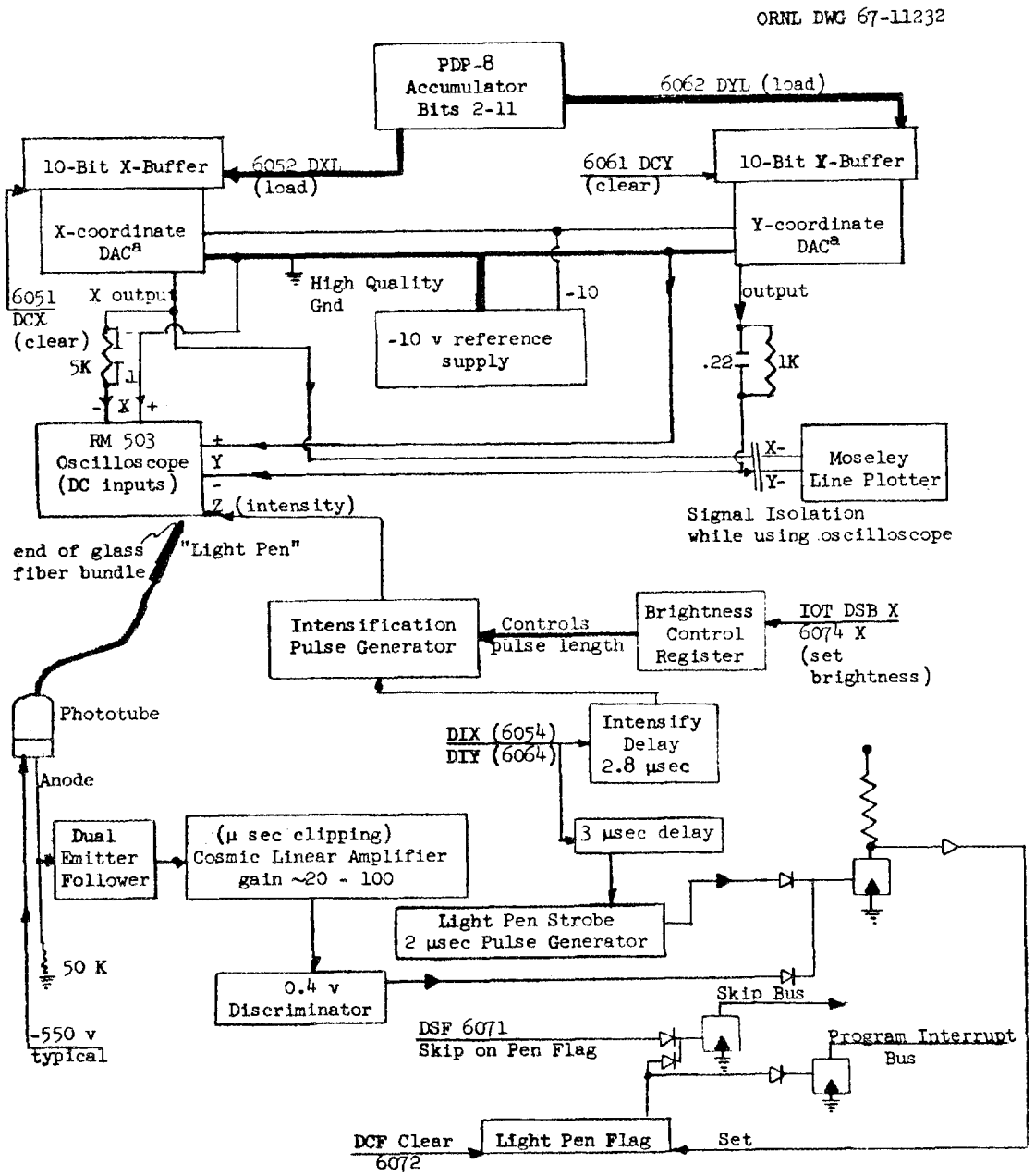
a) The lengths of the intensify pulses for the oscilloscope z-axis are more widely dispersed to increase contrast. Between start of rise and start of fall the lengths are now 1, 2.5, and 13 μsec.

b) The driver for the oscilloscope z-axis (DEC W681) was modified by replacing all PNP transistors with type 2N3638's and all NPN transistors with type 2N3568. Reliable 25 volt pulses are now obtained which reach 23 volts when constrained by a ~1 microsec length.

c) The delay between the intensify (e.g. DIX) IOT pulse and the intensification pulse, required in case one of the DAC's is loaded in the same IOT instruction, has been adjusted to about 2.8 μsec to give adequate settling time. Also, the light pen strobe pulse, which is required in coincidence with a light pen signal to set the light pen flag, is now obtained with a (R302) delay card in position BØD29 to produce a strobe pulse whose length (2.2 μsec) is independent of intensity level at the desired delay (3 μsec) from the intensify IOT-pulse.

d) The circuits joining the DAC outputs to the oscilloscope were altered in an attempt to minimize pickup and noise.

e) Although the DAC's used in the display system seem to have the specified fast settling times, the RM503 oscilloscope (on DC input) does not display a time-independent value. The last 0.2 to 0.5% of a displacement on the screen appears to require a millisecond or so. Therefore,



[a]DAC's are connected so that 0000 in buffer gives 0 volts analog, and 1777_8 in buffer gives -10 volts analog.

Figure 4. Display System

the system has a memory so that the displayed value a given time after the setting of the buffer appears to depend on the DAC output voltage prior to resetting the buffer. The result is blurring of the display for some types of programs. This can be attacked by clearing the DAC's after each displayed point, or by insertion of the shaping network shown in the figure. The effect is not understood, but solutions are reasonably satisfactory. This effect is now at the 0.1 or 0.2% level which line ripple and other noise sources also induce.

Figure 5 illustrates the timing of the pulses relevant to programming the system. To allow understanding of the timing, the computer I/O instructions are reviewed below.

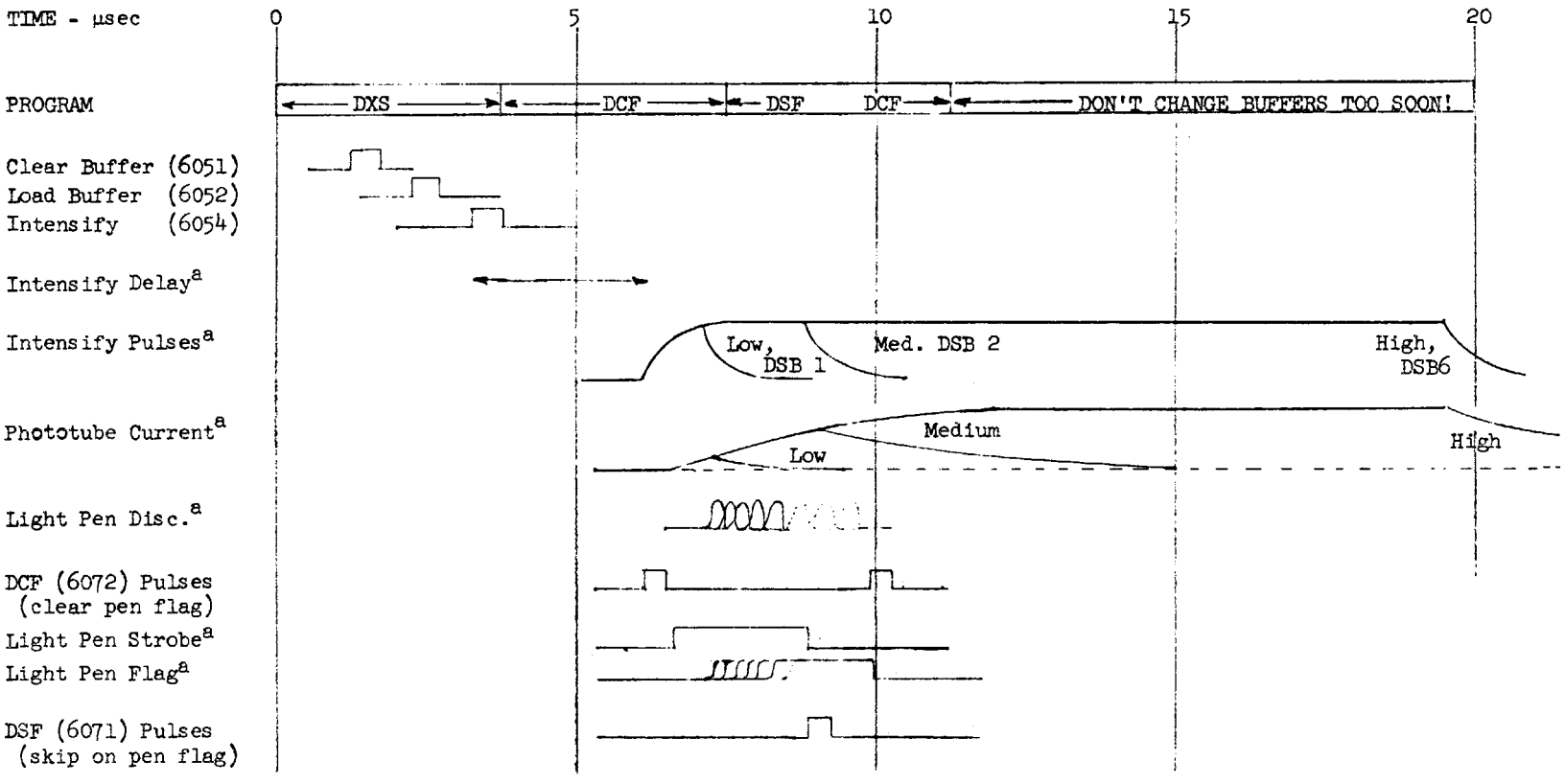
6051	DCX	Clear X-coordinate buffer.
6061	DCY	Clear Y-coordinate buffer.
6053	DXL	Clear and Load X-coordinate buffer.
6063	DYL	Clear and Load Y-coordinate buffer.
6054 6064	DIX DIY	These identical instructions intensify point defined by buffers. After delay, a strobe pulse is provided to the light pen control.
6057	DXS	Clear and load X buffer and then intensify point.
6067	DYS	Clear and load Y buffer and then intensify point. (Note that if the DXS or DYS commands are used, only 1 microsec set-up time would be available to the DAC's before intensification if no set-up delay were provided in the hardware.)
6075 6076 6077	DSB 1 DSB 2 DSB 3	Set Brightness Control Register from bits 10-11 of the instruction. These bits determine the length of the intensification pulse, for all following DIX or DIY instructions until the next DSB instruction.
6071	DSF	Skip on light pen Flag.
6072	DCF	Clear light pen Flag. (Note that if the DSF instruction follows a DIX, the system has had only 1.8 microsec to respond to the <u>initiation</u> of the light flash, less any delay introduced to allow the DAC's additional setup time. Therefore, program delays are required.)

In considering the possible instructions, several questions concerning timing arise.

a) As indicated in Fig. 5, the delay between the intensify (DIX or DIY) instruction and the DSF instruction should be at least two computer cycles (3 microsec).

ORNL DWG 67-11233

Figure 5. Display and Light Pen Timing Diagram. The time scale and the corresponding instructions are shown for a typical program. If the DCF instruction is not used, about 3μsec minimum program delay must be inserted in its place. There is some "walk" in the light pen discriminator output. On low intensity it ranges through the region illustrated by solid-line pulses, but for higher intensities a broader range is observed. Nothing is lost if some of these are missed by the strobe pulse, since they correspond to peripheral "vision" by the pen.



[a]The timing of all these pulses depends upon delays initiated by the DIX (intensify) IOT pulse.

b) If the maximum brightness has been set with a DSB 3 instruction, the coordinate buffers should not be changed for 16 microsec or so. Moreover the univibrator which produces the intensify signal has a deadtime so that adjacent intensify instructions should be separated by at least 19 microsec if both are at the maximum intensity. This easing of the restriction comes about because each intensity level has its own timing univibrator, the outputs of which are OR'ed to produce the input to the z-axis pulse amplifier (W681) card. If medium or low brightness has been selected, intensify instructions may follow each other directly.

It is interesting to consider the problem of contrast on the P-7 oscilloscope screen. Study of the output of the preamplifier connected with the light pen indicates that light production is nearly constant over the length of a 20 microsec intensification if the intensity control is set sufficiently low that an unintensified spot cannot be seen, except for the apparent $\sim$ 4 μsec decay time of the oscilloscope phosphor. Higher intensities can therefore be produced by repeated intensification of the same point by the software. About a factor of five in integrated light output is required to yield a clearly apparent brightness difference.

Bit position 0 and 1 of the Control Buffer, described in the previous section, control the pen of the Moseley recorder, provided the hardware control switch on the recorder is in the STANDBY position. The code used is given below.

Bit 0	Bit 1	Operation
0	0	Standby. Pen lifted and no servo. X and Y inputs disconnected.
0	1	Pen raised, servo motors on ("up").
1	1	Pen lowered, servo motors on ("record").

The following test routines SCOPE and MOS allow testing of the display system. The MOSeley recorder program allows time for setting the diagonal corners of the plotter for zero and the number in the switch register, then plots a box. The SCOPE routine plots points along the diagonal of the screen. The end points are plotted at medium brightness. A point is plotted at mid-scale with brightness taken from SR 10-11 after each end point is plotted, to allow hystersis detection. A fourth point

is plotted above or below the center point by 1/128 full scale, depending upon recognition of the center-scale point by the light pen. The delay between points depends upon the SR contents. An oscilloscope trigger occurs before each mid-scale point.

```
SCØPE, DSB 2            / set medium brightness.
       JMS PLØT         / plot origin, no light pen recognition.
       JMS TSTMID       / plot midscale and test for light pen flag,
                          give trigger.
       DSB 2
       TAD FULL
       JMS PLØT         / plot full scale, no light pen recognition.
       JMS TSTMID       / repeat midpoint to look at hysteresis.
       JMP SCØPE
 FULL, 1777

 PLØT, 0                / plots AC in both coordinates, if link set
                          gives osc trigger. Link cleared if no light
                          pen flag.
       SZL
       FPLØT 10         / trigger pulse for external gear.
       DXL; DYS         / load display buffers and intensify.
       DCF
       DSF DCF
       CLL              / clear link if no flag.
       JMS DELAY        / delay, clears AC.
       JMP I PLØT

 DELAY, 0               / delay routine, length given by SR.
       LAS              / load compliment of delay in 6 microsec
                          units.
       DCA C
       ISZ C; JMP .-1
       JMP I DELAY
     C, 0

TSTMID, 0               / routine for mid-scale points.
       JMS BR           / gets brightness value.
       TAD MID
       STL              / want to test light pen at midscale point.
       JMS PLØT
       TAD MIDP10       / plot higher point unless link set.
       SZL
       TAD N20          / point moves down if flag was seen.
       CLL
       JMS PLØT         / plot a point 1/128 of full scale away.
       JMP I TSTMID
   MID, 1000            / produces half-scale deflection.
MIDP10, 1010            / just over half scale.
   N20, -20
```

```
   BR, 0                  / routine gets brightness from SR 10-11.
       LAS
       AND P0003
       TAD DSBP
       DCA .+1
       0                  / gets DSB instruction.
       JMP I BR
 DSBP, DSB
P0003, 0003

  MØS, PSCF 30            / Moseley Test Program, start calibration at
                            SR value.
       PSCF 40
       LAS
       DXL; DYS           / load display converters
       STL RTR CLA
       CBLD               / correct code to turn on servo with pen
                            lifted.
       PS3SNF             / hit switch 3 to proceed.
       JMP .+2
       JMP .-2            / no flag, stay in loop at SR value.
       PSCF 30            / clear flag.
       DXL; DYS           / load zero into display buffers.
     Z, PS4SNF            / hit switch 4 if done calibrating.
       JMP DRAW
       PS3SNF             / hit switch 3 to try full scale again.
       JMP MØS
       JMP Z              / no flag stay at zero.

 DRAW, CLA CMA            / start to draw box.
       CBLD               / drop pen at zero.
       STL                / count display buffers up.
       JMS LINE           / draws across bottom.
       DXS                / LINE argument.
       JMS LINE; DYS      / draws up right side.
       CLL                / now count buffers down.
       TAD FULL
       JMS LINE; DXS      / draws back across top.
       TAD FULL
       JMS LINE; DYS      / back across left side to origin.
       CBLD               / pen on standby.
       HLT
       JMP MØS            / for restart.

 LINE, 0                  / routine for horizontal or vertical line.
                            call with initial coordinate value in AC
                            and proper display buffer load instruction
                            as argument.  Link set to increase
                            coordinate.
       DCA VALUE
       TAD I LINE
       DCA INST
```

```
       ISZ LINE        / prepare return.
       TAD FULL
       CIA
       DCA CTR         / set line length counter.
    LP, TAD VALUE
       SNL             / link set to add one unit.
       CMA             / subtract one if zero link.
       IAC
       SNL
       CMA
  INST, 0              / gets a coordinate load instruction like
                         DXS.
       DCA VALUE
       JMS DELAY
       ISZ CTR
       JMP LP
       JMS DELAY       / line finished, extra delay.
       JMP I LINE
   CTR, 0
 VALUE, 0
```

APPENDIX A

Modifications to the Type 580 Tape System

The system was modified by Frank Eagen of Digital Equipment Corp., at our request, to provide proper operation of the "file protect ring" logic which prevents writing on the tape when the plastic ring is removed from the back of the tape reel. The 580 system handbook now shows a circuit with this same effect.

A more subtle modification was required so that the TCPI (Tape control Clear Program Interrupt) instruction does not fail if the test is given near the same time as the end-of-record occurs. The program interrupt is set when a record gap is sensed during a SPACE operation, and the TCPI instruction is intended to sense the record gap after a space, and thereupon clear the program interrupt flag, the stop toggle, and also produce a SKIP pulse. In the standard circuit the TCPI pulse separately clears the flags and produces the skip (if flag is set). If the flag has just been set, it may be cleared without ever producing the skip. The possible near-coincidence of these signals can be avoided only if the program interrupt is turned on and only one device (the tape unit) can set it--an unlikely situation. The referenced programs[3,5] test the PI flag without having the interrupt on, and were sensitive to the problem. If other devices can cause program interrupt the same difficulty occurs in any case.

The cure can be effected by complicated programming to assure that the flag is set before it can be tested, but a circuit similar to that described below is required if the system is to work as intended. A pulse amplifier is inserted at the point formerly connected to the skip bus, and the clear signal to the program interrupt flag is given only if this amplifier fires.

This problem is to be considered whenever skip and clear commands are issued with the same instruction. The 100 cps clock (sect. IV A) was a previous example.

APPENDIX B

Summary of Instructions

DATA BREAK CONTROL (Section IV A)

COUNT = 6301	/count up ADDR UDC	BADCLR = 6311	/clear ADDR control
BABLE = 6312	/enable break control	BDISAB = 6314	/disable break control
UDSOFL = 6321	/skip on UDC overflow	UDCLR = 6324	/clear up-down counter
UDSUB1 = 6331	/subtract 1 from UDC	UDSUB2 = 6333	/subtract 2 from UDC
UDSUB3 = 6337	/subtract 3 from UDC (watch timing)	KKKS = 6322	/KlocK clear if skip

EXTERNAL COUNTER READOUT (Section IV B)

CCCUB = 6341	/clear unit buffer	CCLUB = 6343	/load unit buffer CLA
CCIUB = 6361	/increment unit buffer	CCLDB = 6353	/load decade buffer CLA
CCIDB = 6354	/increment decade buffer	CCSDB = 6355	/set decade buffer = 1
CCLDA = 6344	/OR digit to accumulator (do not microprogram)		
CCSTRT = 6371	/start external counters	CCSTOP = 6372	/stop external counters
CCRF = 6401	/reset counter zero	CCRALL = 6403	/reset all counters
CCREG== 6404	/reset registers		

OUTPUT LEVELS AND PULSES (Section IV C)

DACCLR = 6302	/clear DAC buffers	DACLD = 6306	/load DAC buffers
FPIOT = 6401	/front panel IOT (base)	CBC = 6362	/clear control buffer
CBLD = 6366	/load control buffer CLA thumb-wheels and push switches		

THUMBWHEELS AND PUSH SWITCHES (Section IV D)

THMR = 6402	/OR thumbwheel to AC (base)	PSCF = 6404	/clear push-switch flag (base)
PS1SNF = 6441	/skip no switch 1 flag	PS2SNF = 6442	/skip no switch 2 flag
PS3SNF = 6461	/skip no switch 3 flag	PS4SNF = 6462	/skip no switch 4 flag
PS5SNF = 6432	/skip no switch 5 flag	PS5DIS = 6451	/disable switch 5 flag input
PS5EN = 6452	/enable switch 5 flag input		

APPENDIX B - cont.

OSCILLOSCOPE DISPLAY AND LIGHT PEN (Section IV E)

DCX = 6051	/clear X buffer	DCY = 6061	/clear Y buffer
DXL = 6053	/load X buffer	DYL = 6063	/load Y buffer
DIX = 6054	/intensify	DIY = 6064	/intensify
DXS = 6057	/X-coordinate sequence	DYS = 6067	/Y-coordinate sequence
DSF = 6071	/skip on display flag	DCF = 6072	/clear the display flag
DSB = 6074	/set brightness register (OR with brightness level)		

REFERENCES

1. F. E. Bertrand et. al., Neutron Phys. Div. Ann. Progr. Rept., 1967, ORNL-4134 (Vol. II),

2. Digital Equipment Corporation, Maynard, Mass.

3. B. W. Rust and W. R. Burrus, "PUT-5" and "TAKE-5" -- Two Magnetic Tape Systems Routines for the PDP-8, ORNL-TM-1805 (March 22, 1967).

4. B. W. Rust and W. R. Burrus, "BIG HOOP" and "LITTLE HOOP" -- Two "Peter T. Hooper" Routines for Checking the Data Break on a PDP-8, ORNL-TM-1795 (March 22, 1967).

5. B. W. Rust and W. R. Burrus, "SUPER FLICKERS -- An On-Line Data Acquisition System for the PDP-8, ORNL-TM-1878 (June 1, 1967).

6. B. W. Rust and W. R. Burrus, FLEXI-KLUDGE -- A Multi-parameter Nuclear Spectrum Summing System for the PDP-8, ORNL-TM-1879, in press.

INTERNAL DISTRIBUTION

1-3. L. S. Abbott
4. R. G. Alsmiller, Jr.
5. F. E. Bertrand
6-10. W. R. Burrus
11. V. R. Cain
12. C. E. Clifford
13. O. W. Christian
14. J. B. Davidson
15. J. D. Drischler
16. W. A. Gibson
17. N. W. Hill
18. R. W. Ingle
19. T. A. Love
20-24. E. Madden
25-26. F. C. Maienschein
27. J. McConnell
28. E. McDaniel
29-33. C. O. McNew
34. J. T. Mihalczo
35. H. M. Nicholson
36-40. R. W. Peelle
41. F. G. Perey
42. D. Putzulu
43. J. Reynolds
44. R. T. Rosenberry
45. B. Rust
46. R. T. Santoro
47. H. A. Todd
48. D. K. Trubey
49. L. W. Weston
50. J. W. Woody
51. W. Zobel
52. G. Dessauer (consultant)
53. B. C. Diven (consultant)
54. W. N. Hess (consultant)
55. M. H. Kalos (consultant)
56. L. V. Spencer (consultant)
57-58. Central Research Library
59. Document Reference Section
60-258. Laboratory Records Department
259. Laboratory Records ORNL RC
260. ORNL Patent Office

EXTERNAL DISTRIBUTION

261. A. J. Cassette, DECUS, Digital Equipment Corp., Maynard. Mass.
262. E. H. Cooke-Yarborough, Electronics and Applied Physics Branch, AERE, Harwell, Didcot, Berkshire, England
263. R. L. Heath, Philips Petroleum Co., Idaho Falls, Idaho
264-265. J. A. Jones, Digital Equipment Corp., Maynard, Mass.
266. E. Norbeck, Dept. of Physics and Astronomy, University of Iowa, Iowa City, Iowa
267. A. Pearson, Development Engineering Division, Chalk River Nuclear Laboratories, Chalk River, Ontario, Canada
268. D. Vanada, Digital Corp., Maynard, Mass.
269. R. Wolfe, Digital Equipment Corp., Maynard, Mass.
270. P. B. Hemmig, Division of Reactor Development and Technology, U. S. Atomic Energy Commission, Washington, D. C.
271. I. F. Zartman, Division of Reactor Development, U. S. Atomic Energy Commission, Washington, D. C.
272-286. Division of Technical Information Extension (DTIE)
287. Laboratory and University Division (ORO)