C. Fletcher

PROGRAMMED DATA PROCESSOR-1

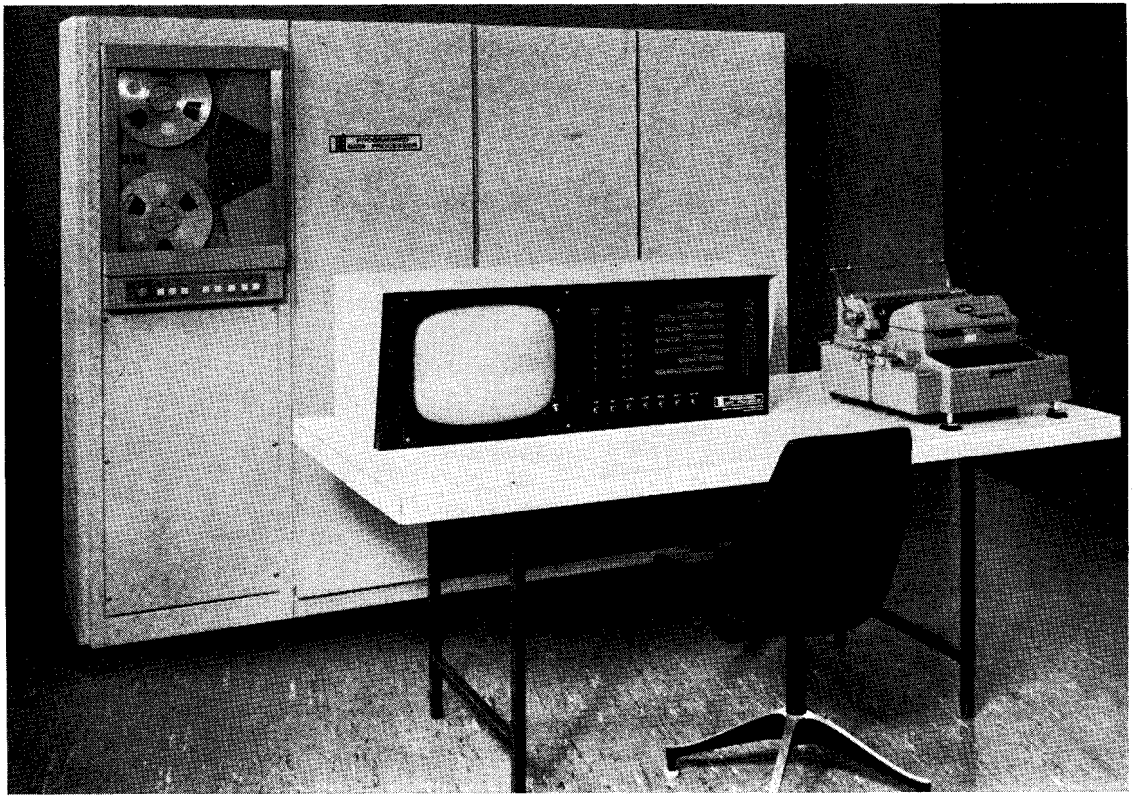
C. Fletcher
PK1/E33

PROGRAMMED DATA PROCESSOR-1

digital equipment corporation

MAYNARD, MASSACHUSETTS

TABLE OF CONTENTS



PROGRAMMED DATA PROCESSOR-1

I. INTRODUCTION

The Programmed Data Processor (PDP-1) is a high speed, solid state digital computer designed to operate with several types of input-output devices, with no internal machine changes. It is a single address, single instruction, stored program computer with powerful program features. Five-megacycle circuits, a magnetic core memory, and fully parallel processing make possible a computation rate of 100,000 additions per second (about 2.5 times the speed of most large computers in use today, and more than 100 times the speed of magnetic drum computers). The PDP-1 is unusually versatile. It is easy to install, operate and maintain. Conventional 110-volt power is used, neither air conditioning nor floor reinforcement is necessary, and preventive maintenance is provided for by built-in marginal checking circuits.

PDP-1 circuits are based on the designs of DEC's highly successful and reliable System Building Blocks. Flip-flops and most switches use saturating transistors. Primary active elements are Micro-Alloy and Micro-Alloy-Diffused transistors.

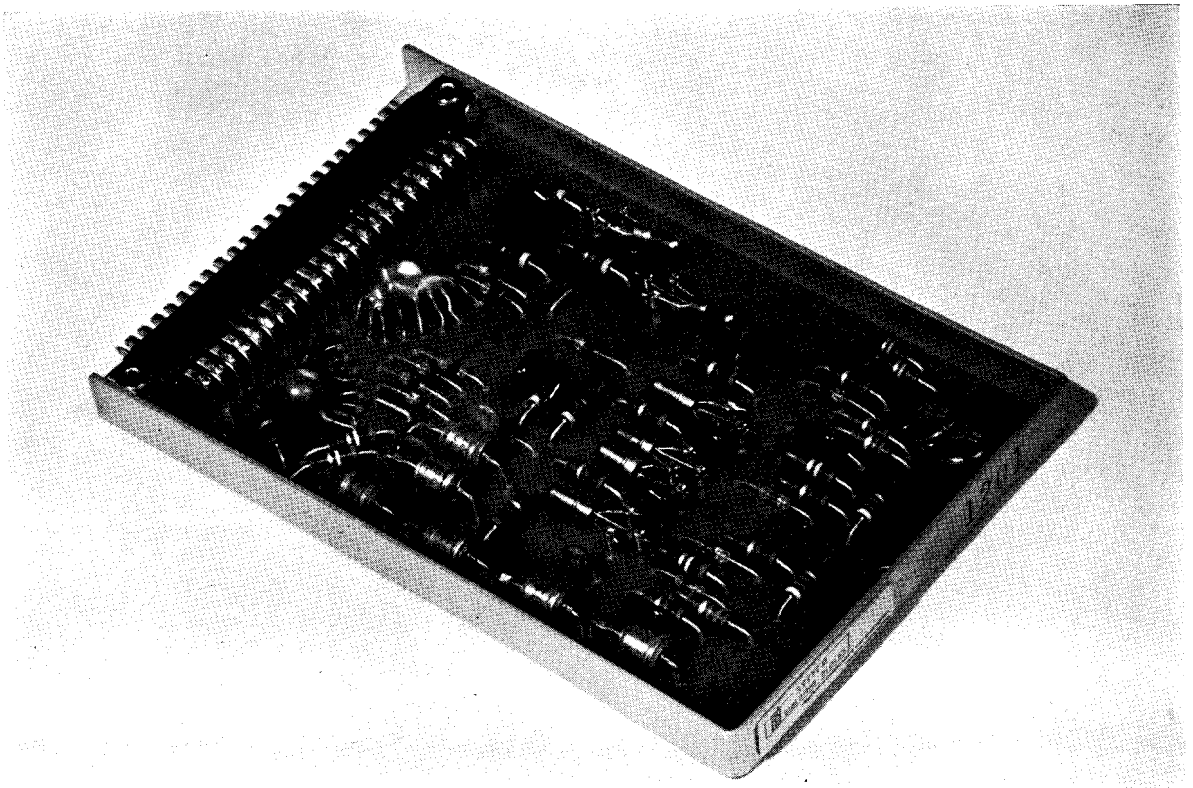
The entire computer occupies only 32 square feet of floor space. It consists of a seven-foot console-desk and three equipment frames.

CENTRAL PROCESSOR

The Central Processor contains the control, arithmetic and memory addressing elements and the memory buffer register. The word length is 18 binary digits. Instructions are carried out in multiples of the memory cycle time of five microseconds. Add, subtract, deposit, and load, for example, are two-cycle instructions requiring 10 microseconds. Multiplication, by subroutine, requires 350 microseconds on the average. Program features include: single address instructions, multiple step indirect addressing and logical arithmetic commands. Console features include: flip-flop indicators grouped for convenient octal reading, six program flags for automatic setting and computer sensing and six sense switches for manual setting and computer sensing.

MEMORY SYSTEM

The coincident-current, magnetic core memory holds 4096 words of 18 bits each. Additional memory units of the same capacity may be readily added to the machine; a memory field switch instruction built into PDP-1 will then select the correct memory module. The



A Standard DEC System Building Block used in PDP-1

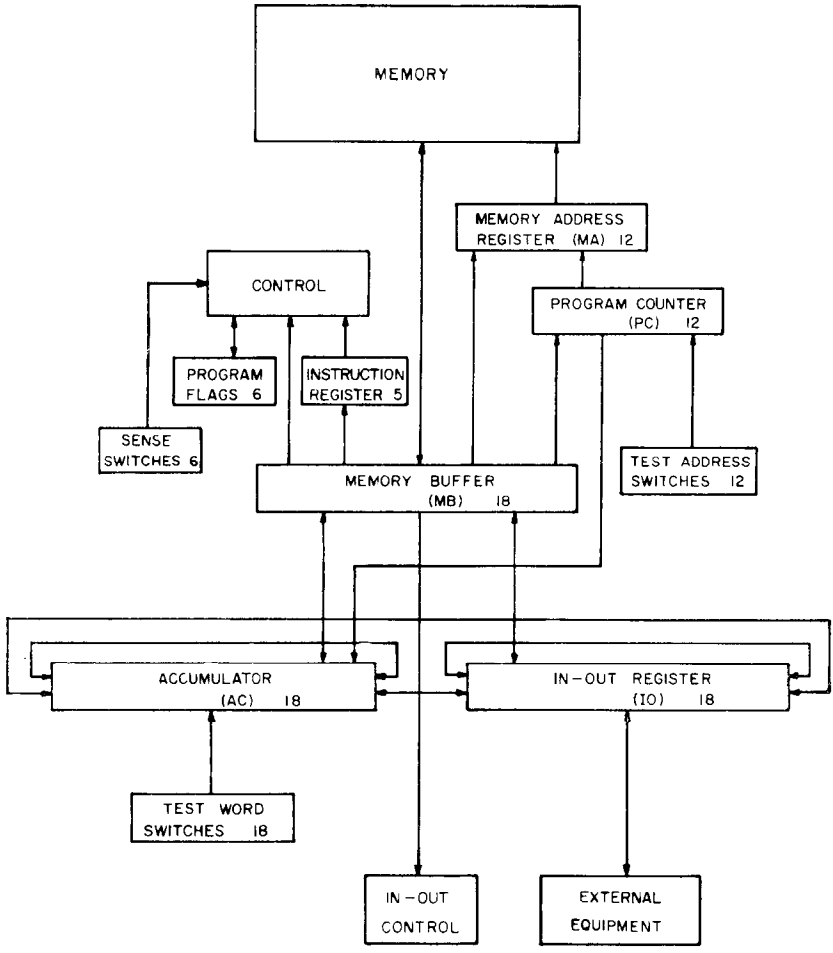
read-rewrite time of the memory is five microseconds, the basic computer rate. Driving currents are automatically adjusted to compensate for temperature variations between 55 and 100 degrees fahrenheit. The core memory storage may be supplemented by up to 64 magnetic tape transports and a tape control unit that serves them all.

INPUT-OUTPUT

PDP-1 is designed to operate a variety of input-output devices. Standard equipment consists of a paper tape reader with a read speed of 300 lines (100 18-bit words) per second, a typewriter for on-line operation in both input and output and a paper-tape punch (alphanumeric or binary) with a nominal speed of 20 characters per second. Optional external equipment includes: compatible magnetic tape (75 inches per second, alphanumeric or binary); 16-inch cathode ray tube for graphic or tabular displays; light pen input; line printer (600 lines per minute); analog to digital and digital to analog converters; and a real time clock. All in-out operations are performed through the In-Out Register.

Of particular interest is the ease with which new, and perhaps unusual, external equipment can be added to PDP-1. Space is provided for additional gates to, and buffers from, the In-Out Register. The in-out system is sufficiently simple so that little control circuitry is needed for additional devices.

The PDP-1 is also available with the optional Sequence Break System. This is a 16-channel (or more, when needed) automatic interrupt feature which permits concurrent operation of several in-out devices.



PDP-1 System Block Diagram

II. PROGRAMMING PDP-1

The Central Processor of PDP-1 contains the Control Element, the Memory Buffer Register, the Arithmetic Element, and the Memory Addressing Element. The Control Element governs the complete operation of the computer including memory timing, instruction performance and the initiation of input-output commands. The Arithmetic Element, which includes the Accumulator and the In-Out Register, performs the arithmetic operations. The Memory Addressing Element, which includes the Program Counter and the Memory Address Register, performs address bookkeeping and modification.

The powerful program features of PDP-1 include multiple step indirect addressing, Boolean operations, twelve variations of arithmetic and logical shifting, and ten conditional instructions. Six independent flip-flops, called "program flags," are available for use as program switches or special in-out synchronizers. Two special instructions, Multiply Step and Divide Step, are included in the Instruction List. Multiply and divide subroutines using these instructions operate in about 350 and 600 microseconds respectively.

NUMBER SYSTEM

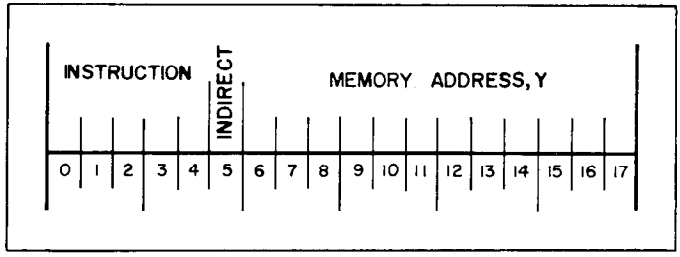
The PDP-1 is a "fixed point" machine using binary arithmetic. Negative numbers are represented as the 1's complement of the positive numbers. Bit 0 is the sign bit which is ZERO for positive numbers. Bits 1 to 17 are magnitude bits, with Bit 1 being the most significant and Bit 17 being the least significant.

The actual position of the binary point may be arbitrarily assigned to best suit the problem in hand. Two common conventions in the placement of the binary point are:

> The binary point is to the right of the least significant digit; thus, numbers represent integers.
>
> The binary point is to the right of the sign digit; thus, the numbers represent a fraction which lies between ± 1.

The conversion of decimal numbers into the binary system for use by the machine may be performed automatically by subroutines. Similarly the output conversion of binary numbers into decimals is done by subroutine. Operations for floating point numbers are handled by interpretive programming. The utility program system provides for automatic insertion of the routines required to perform floating point operations and number base conversion.

INSTRUCTION FORMAT



PDP-1 INSTRUCTION FORMAT

The Bits 0 through 4 define the instruction code; thus there are 32 possible instruction codes, not all of which are used. The instructions may be divided into two classes:

Memory reference instructions

Augmented instructions

In the memory reference instructions, Bit 5 is the indirect address bit. The instruction memory address, Y, is in Bits 6 through 17. These digits are sufficient to address 4096 words of memory.

The augmented instructions use Bits 5 through 17, to specify variations of the basic instruction. For example, in the shift instruction, Bit 5 specifies direction of shift, Bit 6 specifies the character of the shift (arithmetic or logical), Bits 7 and 8 enable the registers (01 = AC, 10 = IO, and 11 = both) and Bits 9 through 17 specify the number of steps.

INDIRECT ADDRESSING

A memory reference instruction which is to use an indirect address will have a ONE in Bit 5 of the instruction word. The original address, Y, of the instruction will not be used to locate the operand, jump location, etc., of the instruction, as is the normal case. Instead, it is used to locate a memory register whose contents in Bits 6 through 17 will be used as the address of the original instruction. Thus, Y is not the location of the operand but the location of the location of the operand. If the memory register containing the indirect address also has a ONE in Bit 5, the indirect addressing procedure is repeated and a third address is located. There is no limit to the number of times this process can be repeated.

OPERATING SPEEDS

Operating times of PDP-1 instructions are multiples of the memory cycle of 5 microseconds. Two-cycle instructions refer twice to

memory and thus require 10 microseconds for completion. Examples of this are add, subtract, deposit, load, etc. The jump instruction and the augmented instructions need only one call on memory and are performed in 5 microseconds.

In-Out Transfer instructions that do not include the optional wait function require 5 microseconds. If the in-out device requires a wait time for completion, the operating time depends upon the device being used.

Each step of indirect addressing requires an additional 5 microseconds.

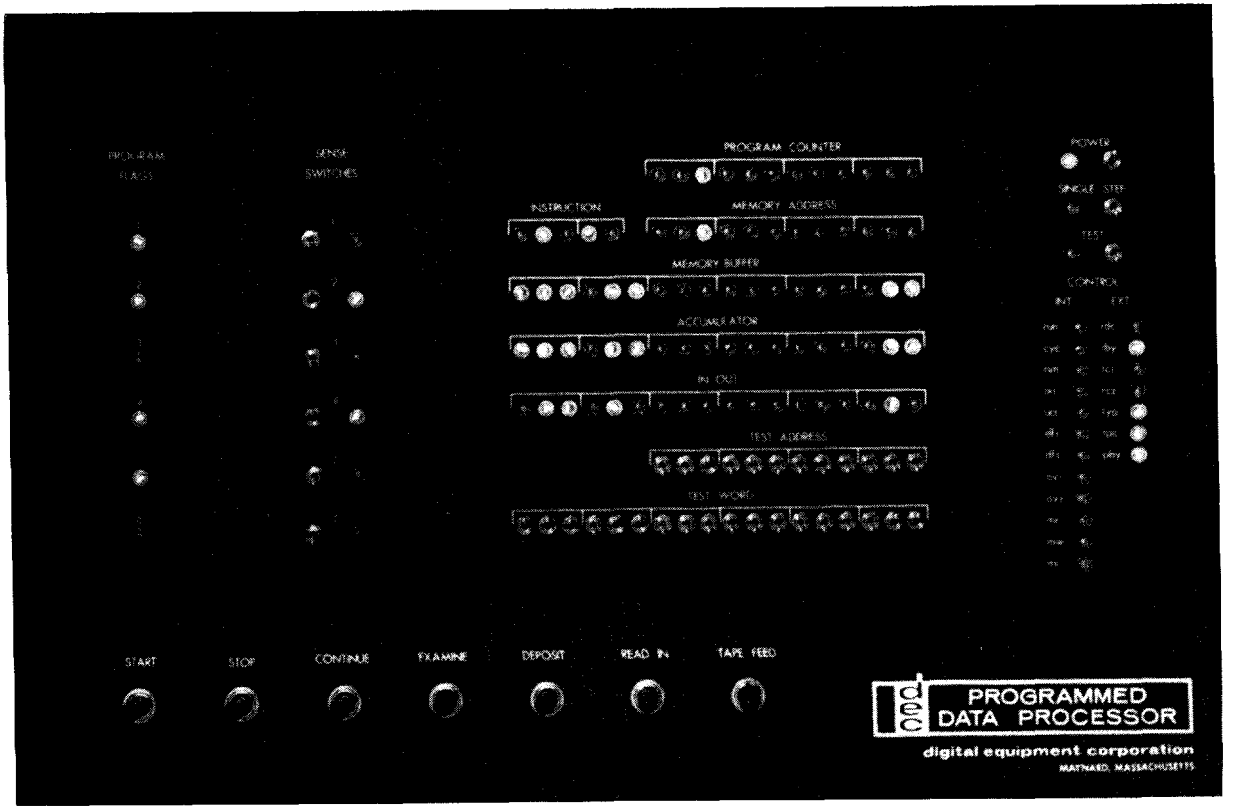
MANUAL CONTROLS

The Console of PDP-1 has controls and indicators for the use of the operator. All computer flip-flops have indicator lights on the Console. These indicators are primarily for use when the machine has stopped or when the machine is being operated one step at a time. While the machine is running, the brightness of an indicator bears some relationship to the relative duty factor of that particular flip-flop.

Three registers of toggle switches are available on the Console. These are the Test Address (12 bits), the Test Word (18 bits), and the Sense Switches (6 bits). The first two are used in conjunction with the operating push buttons. The Sense Switches are present for manual intervention. The use of these switches is determined by the program.

Operating Push Buttons

START	The computer will start. The first instruction comes from the memory location indicated in the Test Address Switches.
STOP	The computer will come to a halt at the completion of the current memory cycle.
CONTINUE	The computer will resume operation starting at the state indicated by the lights.
EXAMINE	The contents of the memory register indicated in the Test Address will be displayed in the Accumulator and the Memory Buffer lights.
DEPOSIT	The word selected by the Test Word Switches will be put in the memory location indicated by the Test Address Switches.
READ-IN	The photoelectric paper tape reader will start operating in the Read-In mode.



PDP-1 Control Panel

Toggle Switches

SINGLE CYCLE SWITCH — When the Single Cycle Switch is on, the computer will halt at the completion of each memory cycle. This switch is particularly useful in debugging programs. Repeated operation of the Continue Switch Button will step the program one cycle at a time. The programmer is thus able to examine the state of the machine at each step.

TEST SWITCH — When the Test Switch is on, the computer will perform the instruction indicated in the Test Address location, repeating this instruction at either the normal or single cycle rate, if the Single Cycle Switch is up. This switch is primarily useful in maintenance.

INSTRUCTION LIST

This list includes the title of the instruction, the normal execution time of the instruction, *i.e.*, the time with no indirect address, the mnemonic code of the instruction, and the operation code number. In the following list, the contents of a register are indicated by C( ). Thus C(Y) means the contents of memory at Address Y; C(AC) means the contents of the accumulator; C(IO) means the contents of the in-out register. An alphabetical and numerical listing of the instructions is contained on Pages 27 to 29.

Memory Reference Instructions

ARITHMETIC INSTRUCTIONS

Add (*10 μsec*)
add Y Operation Code 40

The new C(AC) are the sum of C(Y) and the original C(AC). The C(Y) are unchanged. The addition is performed with 1's complement arithmetic. If the sum exceeds the capacity of the Accumulator Register, the overflow flip-flop will be set (see Skip Group instructions).

Subtract (*10 μsec*)
sub Y Operation Code 42

The new C(AC) are the original C(AC) minus the C(Y). The C(Y) are unchanged. The subtraction is performed using 1's complement arithmetic. If the difference exceeds the capacity of the Accumulator, the overflow flip-flop will be set (see Skip Group instructions).

Multiply Step (*10 μsec*)
mus Y Operation Code 54

If Bit 17 of the In-Out Register is a ONE, the C(Y) are added to C(AC). If IO Bit 17 is a ZERO, the addition does not take place. In either case, the C(AC) and C(IO) are shifted right one place. AC Bit 0 is made ZERO by this shift. This instruction is used in the multiply subroutine.

Divide Step (*10 μsec*)
dis Y Operation Code 56

The Accumulator and the In-Out Register are rotated left one place. IO Bit 17 receives the complement of AC Bit 0. If IO Bit 17 is ONE, the C(Y) are subtracted from C(AC). If IO Bit 17 is ZERO, C(Y) + 1 are added to C(AC). This instruction is used in the divide subroutine.

Index (*10 μsec*)
idx Y Operation Code 44

The C(Y) are replaced by C(Y) + 1. The C(Y) + 1 are left in the Accumulator. The previous C(AC) are lost. Overflow is not indicated.

Index and Skip if Positive (*10 μsec*)
isp Y Operation Code 46

The C(Y) are replaced by C(Y) + 1. The C(Y) + 1 are left in the Accumulator. The previous C(AC) are lost. If, after the addition, C(Y) + 1 are positive, the Program Counter is advanced one extra position and the next instruction in the sequence is skipped. Overflow is not indicated.

LOGICAL INSTRUCTIONS

Logical AND (*10 μsec*)
and Y Operation Code 02

The bits of C(Y) operate on the corresponding bits of the Accumulator to form the logical AND. The result is left in the Accumulator. The C(Y) are unaffected by this instruction.

LOGICAL AND TABLE

AC Bit	*Y Bit*	*Result*
0	0	0
0	1	0
1	0	0
1	1	1

Exclusive OR (*10 μsec*)
xor Y Operation Code 06

The bits of C(Y) operate on the corresponding bits of the Accumulator to form the exclusive OR. The result is left in the Accumulator. The C(Y) are unaffected by this order.

EXCLUSIVE OR TABLE

AC Bit	*Y Bit*	*Result*
0	0	0
0	1	1
1	0	1
1	1	0

Inclusive OR *(10 μsec)*
ior Y Operation Code 04

The bits of C(Y) operate on the corresponding bits of the Accumulator to form the inclusive OR. The result is left in the Accumulator. The C(Y) are unaffected by this order.

INCLUSIVE OR TABLE

AC Bit	*Y Bit*	*Result*
0	0	0
0	1	1
1	0	1
1	1	1

GENERAL INSTRUCTIONS

Load Accumulator *(10 μsec)*
lac Y Operation Code 20

The C(Y) are placed in the Accumulator. The C(Y) are unchanged. The original C(AC) are lost.

Deposit Accumulator *(10 μsec)*
dac Y Operation Code 24

The C(AC) replace the C(Y) in the memory. The C(AC) are left unchanged by this instruction. The original C(Y) are lost.

Deposit Address Part *(10 μsec)*
dap Y Operation Code 26

Bits 6 through 17 of the Accumulator replace the corresponding digits of memory register Y. C(AC) are unchanged as are the contents of Bits 0 through 5 of Y. The original contents of Bits 6 through 17 of Y are lost.

Deposit Instruction Part *(10 μsec)*
dip Y Operation Code 30

Bits 0 through 5 of the Accumulator replace the corresponding digits of memory register Y. The Accumulator is unchanged as are Bits 6 through 17 of Y. The original contents of Bits 0 through 5 of Y are lost.

Load In-Out Register *(10 μsec)*
lio Y Operation Code 22

The C(Y) are placed in the In-Out Register. C(Y) are unchanged. The original C(IO) are lost.

Deposit In-Out Register *(10 μsec)*
dio Y Operation Code 32

The C(IO) replace the C(Y) in memory. The C(IO) are unaffected by this instruction. The original C(Y) are lost.

Jump *(5 μsec)*
jmp Y Operation Code 60

The Program Counter is reset to Address Y. The next instruction that will be executed will be taken from Memory Register Y. The original contents of the Program Counter are lost.

Jump and Save Program Counter (*5 μsec*)
jsp Y Operation Code 62

The contents of the Program Counter are transferred to the Accumulator. When the transfer takes place, the Program Counter holds the address of the instruction following the jsp. The Program Counter is then reset to Address Y. The next instruction that will be executed will be taken from Memory Register Y. The original C(AC) are lost.

Skip if Accumulator and Y differ (*10 μsec*)
sad Y Operation Code 50

The C(Y) are compared with the C(AC). If the two numbers are different, the Program Counter is indexed one extra position and the next instruction in the sequence is skipped. The C(AC) and the C(Y) are unaffected by this operation.

Skip if Accumulator and Y are the same (*10 μsec*)
sas Y Operation Code 52

The C(Y) are compared with the C(AC). If the two numbers are identical, the Program Counter is indexed one extra position and the next instruction in the sequence is skipped. The C(AC) and C(Y) are unaffected by this operation.

Augmented Instructions

Load Accumulator with N (*5 μsec*)
law N Operation Code 70

The number in the memory address bits of the instruction word is placed in the Accumulator. If the indirect address bit is ONE, the complement of N (−N) is put in the Accumulator.

Shift Group (*5 μsec*)
sft Operation Code 66

This group of instructions will rotate or shift the Accumulator and/or the In-Out Register. When the two registers operate combined, the In-Out Register is considered to be an 18-bit magnitude extension of the right end of the Accumulator.

Rotate is a non-arithmetic cyclic shift. That is, the two ends of the register are logically tied together and information is rotated as though the register were a ring.

Shift is an arithmetic operation and is, in effect, multiplication of the number in the register by $2^{\pm N}$, where N is the number of shifts; plus is left and minus is right.

The number of shift or rotate steps to be performed (N) is indicated by the number of ONES in Bits 9 thru 17 of the instruction word. Thus, Rotate Accumulator Right nine times is 671777. A shift or rotate of one place can be indicated nine different ways. The usual convention is to use the right end of the instruction word (rar 1 = 671001).

Rotate Accumulator Right (*5 μsec*)
rar N Operation Code 671

Rotates the bits of the Accumulator right N positions, where N is the number of ONES in Bits 9-17 of the instruction word.

Rotate Accumulator Left (*5 μsec*)
ral N Operation Code 661

Rotates the bits of the Accumulator left N positions, where N is the number of ONES in Bits 9-17 of the instruction word.

Shift Accumulator Right (*5 μsec*)
sar N Operation Code 675

Shifts the contents of the Accumulator right N positions, where N is the number of ONES in Bits 9-17 of the instruction word.

Shift Accumulator Left (*5 μsec*)
sal N Operation Code 665

Shifts the contents of the Accumulator left N positions, where N is the number of ONES in Bits 9-17 of the instruction word.

Rotate In-Out Register Right (*5 μsec*)
rir N Operation Code 672

Rotates the bits of the In-Out Register right N positions, where N is the number of ONES in Bits 9-17 of the instruction word.

Rotate In-Out Register Left (*5 μsec*)
ril N Operation Code 662

Rotates the bits of the In-Out Register left N positions, where N is the number of ONES in Bits 9-17 of the instruction word.

Shift In-Out Register Right (*5 μsec*)
sir N Operation Code 676

Shifts the contents of the In-Out Register right N positions, where N is the number of ONES in Bits 9-17 of the instruction word.

Shift In-Out Register Left (*5 μsec*)
sil N Operation Code 666

Shifts the contents of the In-Out Register left N positions, where N is the number of ONES in Bits 9-17 of the instruction word.

Rotate AC and IO Right (*5 μsec*)
rcr N Operation Code 673

Rotates the bits of the combined registers right in a single ring N positions, where N is the number of ONES in bits 9-17 of the instruction word.

Rotate AC and IO Left (*5 μsec*)
rcl N Operation Code 663

Rotates the bits of the combined registers left in a single ring N positions, where N is the number of ONES in Bits 9-17 of the instruction word.

Shift AC and IO Right (*5 μsec*)
scr N Operation Code 677

Shifts the contents of the combined registers right N positions, where N is the number of ONES in Bits 9-17 of the instruction word.

Shift AC and IO Left (*5 μsec*)
scl N Operation Code 667

Shifts the contents of the combined registers left N positions, where N is the number of ONES in Bits 9-17 of the instruction word.

Skip Group (*5 μsec*)
skp Operation Code 64

This group of instructions senses the state of various flip-flops and switches in the machine. The address portion of the instruction selects the particular function to be sensed. All members of this group have the same operation code.

The instructions in the Skip Group may be combined to form the inclusive OR of the separate skips. Thus, if Address 3000 is selected, the skip would occur if the overflow flip-flop equals ZERO or if the In-Out Register is positive. The combined instruction would still take 5 microseconds.

Skip on ZERO Accumulator (*5 μsec*)
sza Address 100

If the Accumulator is equal to plus ZERO (all bits are ZERO), the Program Counter is advanced one extra position and the next instruction in the sequence is skipped.

Skip on Plus Accumulator (*5 μsec*)
spa Address 200

If the sign bit of the Accumulator is ZERO, the Program Counter is advanced one extra position and the next instruction in the sequence is skipped.

Skip on Minus Accumulator (*5 μsec*)
sma Address 400

If the sign bit of the Accumulator is ONE, the Program Counter is advanced one extra position and the next instruction in the sequence is skipped.

Skip on ZERO Overflow (*5 μsec*)
szo Address 1000

If the overflow flip-flop is a ZERO, the Program Counter is advanced one extra position and the next instruction in the sequence will be skipped. The overflow flip-flop is cleared by the instruction. This flip-flop is set by an addition or subtration that exceeds the capacity of the Accumulator. The overflow flip-flop is not cleared by arithmetic operations which do not cause an overflow. Thus, a whole series of arithmetic operations can be checked for correctness by a single szo. The overflow flip-flop is cleared by the "Start" Switch.

Skip on Plus In-Out Register (*5 μsec*)
spi Address 2000

If the sign digit of the In-Out Register is ZERO, the Program Counter is indexed one extra position and the next instruction in sequence is skipped.

Skip on ZERO Switch (*5 μsec*)
szs Addresses 10, 20. 70

If the selected Sense Switch is ZERO, the Program Counter is advanced one extra position and the next instruction in the sequence will be skipped. Address 10 senses the position of Sense Switch 1, Address 20 Switch 2, etc. Address 70 senses all the switches. If 70 is selected all 6 switches must be ZERO to cause the skip.

Skip on ZERO Program Flag (*5 μsec*)
szf Addresses 0 to 7 inclusive

If the selected program flag is a ZERO, the Program Counter is advanced one extra position and the next instruction in the sequence will be skipped. Address 0 is no selection. Address 1 selects Program Flag 1, etc. Address 7 selects all program flags. All flags must be ZERO to cause the skip.

Operate Group (*5 μsec*)
opr Operation Code 76

This instruction group performs miscellaneous operations on various Central Processor Registers. The address portion of the instruction specifies the action to be performed.

The instructions in the Operate Group can be combined to give the union of the functions. The instruction opr 3200 will clear the AC, put TW to AC, and complement AC. If the number minus zero is interpreted as an instruction, the IO is cleared, AC gets the complement of the TW switches, all program flags are set and the computer halts.

Clear In-Out Register (*5 μsec*)
cli Address 4000

Clears (sets equal to plus zero) the In-Out Register.

Load Accumulator from Test Word (*5 usec*)
lat Address 2000

Forms the inclusive OR of the C(AC) and the contents of the Test Word. This instruction is usually combined with address 200 (clear Accumulator), so that C(AC) will equal the contents of the Test Word Switches.

Complement Accumulator (*5 μsec*)
cma Address 1000

Complements (makes negative) the contents of the Accumulator.

Halt
hlt Address 400

Stops the computer.

Clear Accumulator (*5 μsec*)
cla Address 200

Clears (sets equal to plus zero) the contents of the Accumulator.

Clear Selected Program Flag (*5 μsec*)
clf Address 01 to 07 inclusive

Clears the selected program flag. Address 01 clears Program Flag 1, 02 clears Program Flag 2, etc. Address 07 clears all program flags.

Set Selected Program Flag (*5 μsec*)
stf Addresses 11 to 17 inclusive

Sets the selected program flag. Address 11 sets Program Flag 1; 12 sets Program Flag 2, etc. Address 17 sets all program flags.

In-Out Transfer Group (*5 μsec without in-out wait*)
iot Operation Code 72

The variations within this group of instructions perform all the in-out control and information transfer functions. If Bit 5 (normally the Indirect Address bit) is a ONE, the computer will halt and wait for the completion pulse from the device activated. When this device delivers its completion, the computer will resume operation of the instruction sequence.

An incidental fact which may be of importance in certain scientific or real time control applications is that the time origin of operations following an in-out completion pulse is identical with the time of that pulse.

Most in-out operations require a known minimum time before completion. This time may be utilized for programming. The appropriate In-Out Transfer is given with no in-out wait (Bit 5 a ZERO). The instruction sequence then continues. This sequence must include an iot instruction which performs nothing but the in-out wait, and the instruction must occur before the safe minimum time. A table of minimum times for all in-out devices is delivered with the computer: it lists minimum time before completion pulse and minimum In-Out Register free time.

III. INPUT-OUT EQUIPMENT

STANDARD EQUIPMENT

Paper Tape Reader

The paper tape reader of the PDP-1 is a photoelectric device capable of reading 300 lines per second. Three lines form the standard 18-bit word when reading binary punched eight-hole tape. Five, six, and seven-hole tape may also be read.

Read Paper Tape, Alphanumeric
rpa iot 1

In this mode, one line of tape is read for each In-Out Transfer. All eight holes of the line are read. The information is left in the right eight bits of the In-Out Register, the remainder of the register being left clear.

The code of the off-line tape preparation typewriter (Friden FPC-8 "Flexowriter") contains an odd parity bit. This bit may be checked by the read-in program. The Friden Code is then converted to a concise six-bit code. This conversion squeezes out the fifth bit (parity) and drops the eighth bit. The carriage return character (Friden 200) is converted to 77.

The more concise code is used by the on-line typewriter, printer, and magnetic tape. A list of characters and their codes is found on Pages 30 and 31.

Read Paper Tape Binary
rpb iot 2

For each In-Out Transfer instruction, three lines of paper tape are read and assembled in the In-Out Register to form a full computer word. For a line to be recognized in this mode, the eighth hole must be punched; *i.e.*, lines with no eighth hole will be skipped over. The seventh hole is ignored. The pattern of holes in the binary tape is arranged so as to be easily interpreted visually in terms of machine instruction.

Read-In Mode

This is a special mode activated by the "Read-In" switch on the console. It provides a means of entering programs which neither rely on programs in memory nor require a plug board. Pushing the "Read-In" switch starts the reader in the binary mode. The first group of three lines, and alternate succeeding groups of three lines, are interpreted as "Read-In" mode instructions. Even-numbered groups of three lines are data. The "Read-In" mode instructions must be either "deposit in-out" (dio Y) or "jump" (jmp Y). If the instruction is dio Y, the next group of three binary lines will be stored in memory location Y and the reader continues moving. If the instruction is jmp Y, the "Read-In" mode is terminated and the computer will commence operation at the address of the jump instruction.

Paper Tape Punch

The standard PDP-1 paper tape punch has a nominal speed of 20 lines per second. It can operate in either the alphanumeric mode or the binary mode.

Punch Paper Tape, Alphanumeric

ppa iot 5

For each In-Out Transfer instruction one line of tape is punched. In-Out Register Bit 17 conditions Hole 1. Bit 16 conditions Hole 2, etc. Bit 10 conditions Hole 8.

Punch Paper Tape, Binary

ppb iot 6

For each In-Out Transfer instruction one line of tape is punched. In-Out Register Bit 5 conditions Hole 1. Bit 4 conditions Hole 2, etc. Bit 0 conditions Hole 6. Hole 7 is left blank. Hole 8 is always punched in this mode.

Typewriter

The typewriter will operate in the input mode or the output mode.

Type Out

tyo iot 3

For each In-Out Transfer instruction one character is typed. The character is specified by the right six bits of the In-Out Register.

Type In

tyi iot 4

This operation is completely asynchronous and is therefore handled differently than any of the preceding in-out operations.

When a typewriter key is struck, Program Flag 1 is set. At the same time the code for the struck key is presented to gates connected to the right six bits of the In-Out Register. This information will remain at the gate for a relatively long time by virtue of the slow mechanical action. A program designed to accept typed-in data would periodically check the status of Program Flag 1. If at any time Program Flag 1 is found to be set, an In-Out Transfer instruction with Address 4 must be executed for information to be transferred. This In-Out Transfer should not use the optional in-out halt. The information contained in the typewriter's coder is then read into the right six bits of the In-Out Register. tyi does not clear the IO. The tyi is usually preceded by cli and clf-1.

OPTIONAL EQUIPMENT

Magnetic Tape

The magnetic tape system consists of the magnetic tape control unit and one or more tape transport units which contain the read and write circuits. The tape control unit contains the equipment to select the active transport and the logic necessary to control the system.

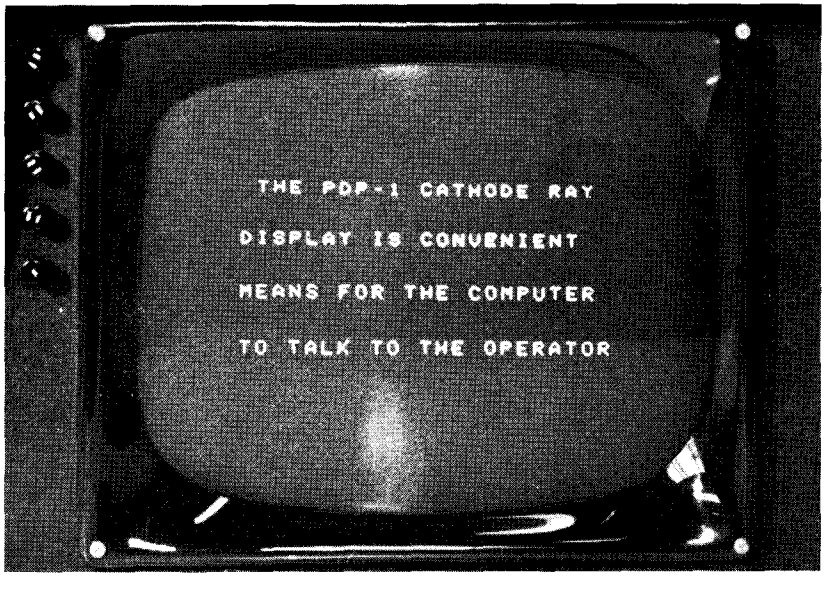
The method of recording is non-return-to-zero. Each flux change represents a binary ONE. The reading is done at two levels of sensitivity. A check is performed at each level. Thus the high level of sensitivity detects the presence of excessive noise and the low sensitivity detects weak ONES.

The transports operate at 75 inches per second with a recording density of 200 bits to the inch. The format is the same as for the IBM 729 I. Seven tracks are written: six are binary or alphanumeric bits, and a seventh is used as a lateral parity. At the completion of a record, which may be of arbitrary length, a longitudinal parity is written.

Real Time Clock

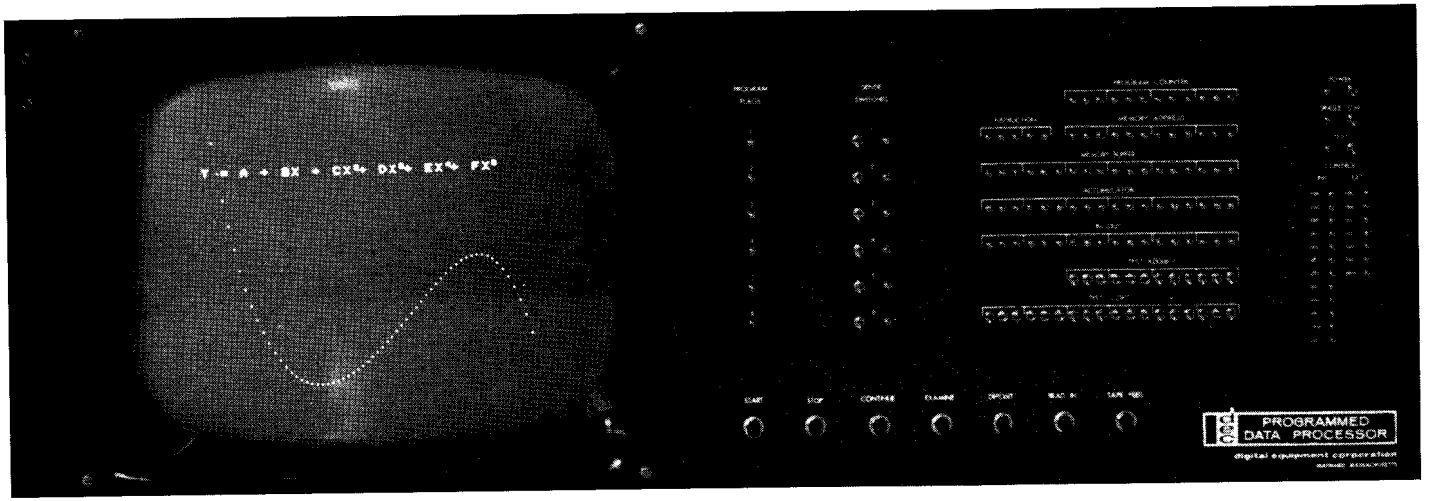
A special input register may be connected to operate as a real time clock. This is a counting register operated by a crystal controlled oscillator.

The state of this counter may be read at any time by the appropriate In-Out Transfer instruction. The computer stops only long enough to provide synchronization with the clock oscillator, then resumes operation in phase with it.



Cathode Ray Tube Display

The PDP-1 cathode ray tube display is useful for presentation of graphical or tabular data to the operator. For each In-Out Transfer instruction, one point is displayed. The first 10 bits of the In-Out Register, Bits 0-9, are the X coordinate of the point. Bits 0-9 of the Accumulator are the Y coordinate of the point.



Cathode Ray X-Y Point Plotter

An additional display option is a light pen. By use of this device the computer is signaled that the operator is interested in the last point displayed. Thus the program can take appropriate action such as changing the display or shifting operation to another program.

Line Printer

A 72-column line printer is available as an on-line printing station. The operating speed is 450 lines per minute. A simple one-line buffer is part of this equipment. The appropriate In-Out Transfer instruction is repeated to fill the buffer. The order to print is then given. Following the completion of the line print, the printer returns a completion pulse and spaces the paper.

Analog Equipment

Equipment providing analog input to and output from the computer can be provided. This equipment can take the form either of high speed electronic equipment or shaft position conversion equipment. In either case, multiplexing can be provided.

Other Optional Equipment

Additional in-out devices may be added to PDP-1 with, at most, a few hour's work on the machine. Sockets for several In-Out Transfer variation pulse commands are prewired. Space is provided for additional gates to and buffers from the In-Out Register. The in-out system is sufficiently simple so that the control circuitry needed for any additional device is minimal.

Sequence Break System

An optional in-out control is available for PDP-1. This control, termed the Sequence Break System, allows concurrent operation of several in-out devices and the main sequence. The system has, nominally, 16 automatic interrupt channels arranged in a priority chain.

A break to a particular sequence may be initiated by the completion of an in-out device, the program, or any external signal. If this sequence has priority, the C(AC), C(IO), C(PC), and the contents of the memory field flip-flops (if present) are stored in adjacent fixed locations unique to that sequence. The Program Counter is reset to the address contained in a fourth fixed location. The program is now operating in the new sequence. This new sequence may be broken by a higher priority sequence. A typical program loop for handling an in-out sequence would contain 3 to 5 instructions, including the appropriate iot. These are followed by load AC and load IO from the fixed locations and an indirect jump to location of the previous C(PC). This last instruction terminates the sequence.

The Sequence Break System provides PDP-1 with much of the power of a multiple sequence machine or of a computer having in-out synchronizers or automatic trunks.

IV. UTILITY PROGRAMS

The Utility Programs for PDP-1 are designed to provide the nucleus of a growing system of programs. Programs available upon delivery of the machine are:

SYMBOLIC ADDRESS ASSEMBLY PROGRAM is the basic element in the utility system. It is designed for maximum flexibility consistent with adequate indication of program errors. Numerous macro instructions are included such as floating point add, subtract, multiply, divide, decimal-to-binary conversion, and binary-to-decimal conversion.

MEMORY PRINT-OUT can appear either on the typewriter or on the line printer, if connected.

BINARY PUNCH will punch out a specified region of memory. Check characters are included on the tape. The program is available in both the binary read-in format and the read-in mode format.

BINARY READ-IN reads the tapes prepared by the Binary Punch Program. Several versions of this program are available. They differ in the region of memory in which the read-in program is written. Thus, various read-in programs are available in both the binary read-in format and the read-in mode format.

MAINTENANCE PROGRAMS include programs for checking memories, input-output equipment, and operation of the Central Processor.

V. APPENDIX

ABBREVIATED INSTRUCTION LIST

Basic Instructions

Instruction	*Code #*	*Explanation*	*Oper. Time* (μsec)	*Page Ref.*
add Y	40	Add C(Y) to C(AC)	10	13
and Y	02	Logical AND C(Y) with C(AC)	10	14
dac Y	24	Put C(AC) in Y	10	15
dap Y	26	Put contents of address part of AC in Y	10	15
dio Y	32	Put C(IO) in Y	10	15
dip Y	30	Put contents of instruction part of AC in Y	10	15
dis Y	56	Divide step	10	14
idx Y	44	Index (add one) C(Y), leave in Y & AC	10	14
ior Y	04	Inclusive OR C(Y) with C(AC)	10	15
iot Y	72	In-out transfer, see below		20
isp Y	46	Index and skip if result is positive	10	14
jmp Y	60	Take next instruction from Y	5	15
jsp Y	62	Jump to Y and save program counter in AC	5	16
lac Y	20	Load the AC with C(Y)	10	14
law N	70	Load the AC with the number N	5	16
law $-$N	71	Load the AC with the number $-$N	5	16
lio Y	22	Load IO with C(Y)	10	15
mus Y	54	Multiply step	10	14
opr	76	Operate, see below	5	19
sad Y	50	Skip next instruction if C(AC) $\neq$ C(Y)	10	16
sas Y	52	Skip next instruction if C(AC) = C(Y)	10	16
shift	66	See below	5	16
skp	64	Skip, see below	5	18
sub Y	42	Subtract C(Y) from C(AC)	10	13
xor Y	06	Exclusive OR C(Y) with C(AC)	10	14

Operate Group

Instruction	*Code #*	*Explanation*	*Oper. Time (μsec)*	*Page Ref.*
cla	760200	Clear AC	5	19
clf	760001-7	Clear selected Program Flag	5	20
cli	764000	Clear IO	5	19
cma	761000	Complement AC	5	19
hlt	760400	Halt	5	19
lat	762200	Load AC from Test Word switches	5	19
stf	760011-7	Set selected Program Flag	5	20

In-Out Transfer Group

ppa	730005	Punch paper tape alphanumeric		22
ppb	730006	Punch paper tape binary		22
rpa	730001	Read paper tape alphanumeric		21
rpb	730002	Read paper tape binary		21
tyi	720004	Read typewriter input switches	5	22
tyo	730003	Type out		22

Skip Group

sma	640400	Skip on minus AC	5	18
spa	640200	Skip on plus AC	5	18
spi	642000	Skip on plus IO	5	18
sza	640100	Skip on ZERO (+0) AC	5	18
szf	64000f	Skip on ZERO flag (f = flag #)	5	19
szo	641000	Skip on ZERO overflow (and clear overflow)	5	18
szs	6400S0	Skip on ZERO sense switch (S = switch #)	5	19

Shift/Rotate Group

ral	661	Rotate AC left	5	17
rar	671	Rotate AC right	5	16
rcl	663	Rotate combined AC & IO left	5	17
rcr	673	Rotate combined AC & IO right	5	17
ril	662	Rotate IO left	5	17
rir	672	Rotate IO right	5	17
sal	665	Shift AC left	5	17

SHIFT/ROTATE GROUP (Continued)

Instruction	*Code #*	*Explanation*	*Oper. Time (μsec)*	*Page Ref.*
sar	675	Shift AC right	5	17
scl	667	Shift combined AC & IO left	5	18
scr	677	Shift combined AC & IO right	5	17
sil	666	Shift IO left	5	17
sir	676	Shift IO right	5	17

NUMERICAL INSTRUCTION LIST

Code	*Instruction*	*Code*	*Instruction*
00	*	40	add
02	and	42	sub
04	ior	44	idx
06	xor	46	isp
10	*	50	sad
12	*	52	sas
14	*	54	mus
16	*	56	dis
20	lac	60	jmp
22	lio	62	jsp
24	dac	64	skp
26	dap	66	Shift
30	dip	70	law
32	dio	72	iot
34	*	74	*
36	*	76	opr

* spare code, computer will halt

ALPHANUMERIC CODES

TABLE I

Character	*Friden Code*	*Concise Code*	*Character*	*Friden Code*	*Concise Code*
a A	141	61	y Y	070	30
b B	142	62	z Z	051	31
c C	163	63	0)	040	20
d D	144	64	1 '	001	01
e E	165	65	2 @	002	02
f F	166	66	3 #	023	03
g G	147	67	4 =	004	04
h H	150	70	5 %	025	05
i I	171	71	6 ¢	026	06
j J	121	41	7 ?	007	07
k K	122	42	8 *	010	10
l L	103	43	9 (	031	11
m M	124	44	Space	020	00
n N	105	45	, ,	073	33
o O	106	46	. .	153	73
p P	127	47	/ :	061	21
q Q	130	50	& ;	160	60
r R	111	51	$ —	133	53
s S	062	22	- "	100	40
t T	043	23	Upper Case	174	74
u U	064	24	Lower Case	172	72
v V	045	25	Tab.	076	36
w W	046	26	Carr. Ret.	200	77
x X	067	27	Tape Feed	177	—

TABLE II

Friden	*Character*	*Friden*	*Character*
001	1 '	043	t T
002	2 @	045	v V
004	4 =	046	w W
007	7 ?	051	z Z
010	8 *	061	/ :
020	Space	062	s S
023	3 #	064	u U
025	5 %	067	x X
026	6 ¢	070	y Y
031	9 (	073	, ,
040	0)	076	Tab.

TABLE II (Continued)

Friden	*Character*	*Friden*	*Character*
100	– "	144	d D
103	l L	147	g G
105	n N	150	h H
106	o O	153	. .
111	r R	160	& ;
121	j J	163	c C
122	k K	165	e E
124	m M	166	f F
127	p P	171	i I
130	q Q	172	Lower Case
133	$ —	174	Upper Case
141	a A	177	Tape Feed
142	b B	200	Carr. Ret.

TABLE III

Concise Code	*Character*	*Concise Code*	*Character*
00	Space	42	k K
01	1 '	43	l L
02	2 @	44	m M
03	3 #	45	n N
04	4 =	46	o O
05	5 %	47	p P
06	6 ¢	50	q Q
07	7 ?	51	r R
10	8 *	53	$ —
11	9 (	60	& ;
20	0)	61	a A
21	/ :	62	b B
22	s S	63	c C
23	t T	64	d D
24	u U	65	e E
25	v V	66	f F
26	w W	67	g G
27	x X	70	h H
30	y Y	71	i I
31	z Z	72	Lower Case
33	, ,	73	. .
36	Tab.	74	Upper Case
40	– "	77	Carr. Ret.
41	j J		