PDP-1 COMPUTER
ELECTRICAL ENGINEERING DEPARTMENT
M.I.T.
CAMBRIDGE, MASSACHUSETTS 02139

PDP-35

PDP-1 INSTRUCTION LIST

April 29, 1966

INDEX TO INSTRUCTIONS

INDEX TO INSTRUCTIONS (contd.)

PDP-1 INSTRUCTION LIST

This list includes the title of the instruction, the normal execution time (i.e., the time with no indirect address*), the mnemonic code of the instruction, the operation code number, and the description of the instruction's operation. In the following list, the Accumulator is abbreviated as AC, the In-Out Register as IO, and the contents of a register as C( ). Thus, C(Y) indicates the contents of memory at Address Y, C(AC), the contents of the Accumulator; and C(IO), the contents of the In-Out Register.

I. MEMORY REFERENCE INSTRUCTIONS

A. Information Transfer

1. 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.

2. 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.

3. 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. The 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.

* Add 5 microseconds for each indirect address level.

4. Deposit Instruction Part (10μ sec)

   dip Y Operation Code 30

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

5. Load In-Out Register (10μ sec)

   lio Y Operation Code 22

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

6. Deposit In-Out Register (10μ sec)

   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.

7. Deposit Zero in Memory (10μ sec)

   dzm Y Operation Code 34

   Clears (sets equal to plus zero) the C(Y).

B. Arithmetic Instructions

1. Add (10 μ sec)

add Y Operation Code 40

The sum of the C(Y) and the C(AC) replace the C(AC). The C(Y) are unchanged. The addition is performed with one's complement arithmetic. If the sum of two like-signed numbers yields a result of the opposite sign, the overflow flip-flop will be set (see szo instruction). A result of minus zero is changed to plus zero.

2. Subtraction (10 μ sec)

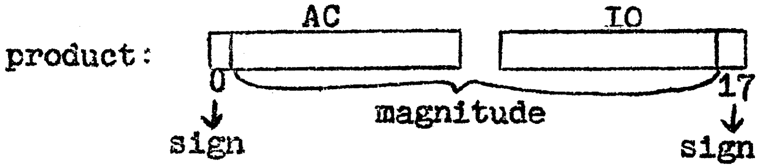
sub Y Operation Code 40

The C(AC) minus the C(Y) replace the C(AC). The C(Y) are unchanged. The subtraction is performed using one's complement arithmetic. When two unlike-signed numbers are subtracted, the sign of the result must agree with the sign of the original Accumulator, or the overflow flip-flop will be set (see szo instruction). A result of minus zero can exist in one instance only. (-0) - (+0) = (-0)

3. Multiply (14 to 25 μ sec)

mul Y Operation Code 54

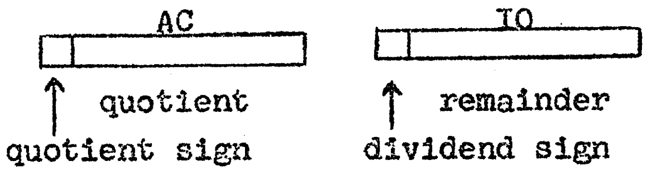
The product of the C(AC) and the C(Y) is formed in the AC and IO registers. The sign of the product is in the AC sign bit. IO bit 17 also contains the sign of the product. The magnitude of the product is the 34-bit string from AC Bit 1 through IO bit 16. The C(Y) are not affected by this instruction. If the entire product results in a minus zero, it is changed to a plus zero. The variation in execution time is caused by the number of one's in C(Y).



4. Divide (30 to 40μ.sec, except on overflow, 12μsec)

div Y Operation Code 56

The dividend must be in the AC and IO registers in the form indicated in the instruction, mul. IO bit 17 is ignored. The divisor is the C(Y). At the completion of the instruction, the C(AC) are the quotient and the C(IO) are the remainder. The sign of the remainder (in IO bit zero) is the sign of the dividend. The instruction that follows a div instruction will be skipped unless an overflow occurs. The C(Y) are not affected by this instruction. If the remainder or quotient result in minus zero, that value is changed to plus zero. If the magnitude of the high order part of the dividend is equal to or greater than the magnitude of the divisor, an overflow is indicated. In this case, the instruction following the div is not skipped. The original C(AC) and C(IO) are restored. The overflow flip-flop is not affected.



5. Index (10μ sec)

idx Y Operation Code 44

C(Y) + 1 replace the C(Y) and the C(AC). The previous C(AC) are lost. Overflow is not indicated. If the original C(Y) equals the integer, minus one (-1), the result after indexing is plus zero.

C. Logical Instruction

1. Logical AND (10μ sec)

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

2. Exclusive OR (10μ sec)

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 instruction.

EXCLUSIVE OR TABLE

AC bit	Y bit	Result
0	0	0
0	1	1
1	0	1
1	1	0

3. Inclusive OR (10μ sec)

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(AC) are unaffected by this instruction.

INCLUSIVE OR TABLE

AC bit	Y bit	Result
0	0	0
0	1	1
1	0	1
1	1	1

D. Decision-Making Instructions

1. Skip if Accumulator and Y Differ (10μsec)

   sad Y Operation Code 50

   The C(Y) are compared with 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.

2. Skip if Accumulator and Y are the Same (10μsec)

   sas Y Operation Code 52

   The C(Y) are compared with 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 the C(Y) are unaffected by this operation.

3. Index and Skip if Positive (10μsec)

   isp Y Operation Code 46

   The C(Y) +1 replace the C(Y) and the C(AC). The previous C(AC) are lost. If, after the addition, the Accumulator is positive, the Program Counter is advanced one extra position and the next instruction in the sequence is skipped. Overflow is not indicated. If the original C(Y) equals the integer, minus one (-1), the result after indexing is plus zero and the skip takes place.

E. Transfer Instructions

1. Jump (5 µ sec)

   jmp Y Operation Code 60

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

2. Jump and Save Program Counter (5 µ sec)

   jsp Y Operation Code 62

   The contents of the Program Counter are transferred to bits 6 through 17 of the Accumulator. The state of the overflow flip-flop is transferred to bit zero, the condition of the Extend flip-flop to bit 1, and the contents of the Extended Program Counter to bits 2, 3, 4, and 5 of the AC. When the transfer of the PC to the AC 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 executed will be taken from Memory Register Y. The original C(AC) are lost.

3. Jump and Deposit Accumulator (10 µ sec)

   jda Y Operation Code 17

   The contents of the AC are deposited in Memory Register Y. The contents of the Program Counter (holding the address of the instruction following the jda instruction) are transferred to bits 6 through 17 of the AC. The state of the overflow flip-flop is transferred to bit zero, the condition of the Extend flip-flop to bit 1, and the contents of the Extended Program Counter to bits 2, 3, 4, and 5 of the AC. The next instruction executed is taken from Memory Register Y+1. The jda instruction requires that the indirect bit be a one, but indirect addressing does not occur. The instruction is equivalent to the instruction dac Y followed by jsp Y+1.

4. Jump and Deposit Program Counter (10μ sec)

jdp Y Operation Code 14

The contents of the Program Counter (holding the address of the instruction following the jdp) are deposited in bits 6 through 17 of the Memory Register Y. The original contents of the AC remain in the AC unchanged. The state of the overflow flip-flop is transferred to bit zero, the condition of the Extend flip-flop to bit 1, and the contents of the Extended Program Counter to bits 2, 3, 4, and 5 of the Memory Register Y. The next instruction executed is taken from Memory Register Y+1.

5. Call Subroutine (10μsec)

cal Y Operation Code 16

The address part of the instruction, Y, is ignored. The contents of the AC are deposited in Memory Register 100. The contents of the Program Counter (holding the address of the instruction following the cal) are transferred to bits 6 through 17 of the AC. The state of the overflow flip-flop is transferred to bit zero, the condition of the Extend flip-flop to bit 1, and the contents of the Extended Program Counter to bits 2, 3, 4, and 5 of the AC. The next instruction executed is taken from Memory Register 101. The cal instruction requires that the indirect bit be zero. The instruction may be used as part of a master routine to call subroutines.

F. Miscellaneous

1. Execute (5μ sec plus time of instruction executed)

xct Y Operation Code 10

The instruction located in register Y is executed. The Program Counter remains unchanged (unless a jump or skip were executed). If a skip instruction is executed (by xct Y), depending on the skip condition either the Program Counter is advanced one extra position and the next instruction in the sequence (after the xct instruction) is skipped or the Program Counter remains unchanged and the next instruction in the sequence is executed. Execute may be indirectly addressed, and the instruction being executed may use indirect addressing. An xct instruction may execute other xct commands.

II. AUGMENTED INSTRUCTIONS

1. 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 (bit 5) is a one (1), (-N) is put in the Accumulator.

A. 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.

a. 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.

b. 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. As bits are shifted out from one end of a register they are replaced at the other end by ones if the number is negative and zeroes if the number is positive. The sign bit is not shifted. The number of shift or rotate steps to be performed (N) is indicated by the number of one's (1's) in bits 9 through 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)

1. Rotate Accumulator Right (5μsec)

   rar N   Operation Code 671

   Rotates the bits of the Accumulator right N positions, where N is the number of one's (1's) in bits 9 through 17 of the instruction word.

2. Rotate Accumulator Left (5μsec)

   ral N   Operation Code 661

   Rotates the bits of the Accumulator left N positions, where N is the number of one's (1's) in bits 9 through 17 of the instruction word.

3. Shift Accumulator Right (5μsec)

   sar N   Operation Code 675

   Shifts the contents of the Accumulator right N positions, where N is the number of one's (1's) in bits 9 through 17 of the instruction word. As bits are shifted out from the right end of the AC they are replaced at the left end by ones if the number is negative and zeroes if the number is positive. The sign bit is not shifted.

4. Shift Accumulator Left (5μsec)

   sal N   Operation Code 665

   Shifts the contents of the Accumulator left N positions, where N is the number of one's (1's) in bits 9 through 17 of the instruction word. As bits are shifted out from the left end of the AC they are replaced at the right end by ones if the number is negative and zeroes if the number is positive. The sign bit is not shifted.

5. Rotate In-Out Register Right (5μsec)

   rir N   Operation Code 672

   Rotates the bits of the In-Out Register right N positions, where N is the number of one's (1's) in bits 9 through 17 of the instruction word.

6. Rotate In-Out Register Left (5μsec)

ril N Operation Code 662

Rotates the bits of the In-Out Register left N positions, where N is the number of one's (1's) in bits 9 through 17 of the instruction word.

7. Shift In-Out Register Right (5μsec)

sir N Operation Code 676

Shifts the contents of the In-Out Register right N positions, where N is the number of one's (1's) in bits 9 through 17 of the instruction word. As bits are shifted out from the right end of the IO they are replaced at the left end by ones if the number is negative and zeroes if the number is positive. The sign bit is not shifted.

8. Shift In-Out Register Left (5μsec)

sil N Operation Code 666

Shifts the contents of the In-Out Register left N positions, where N is the number of one's (1's) in bits 9 through 17 of the instruction word. As bits are shifted out from the left end of the IO they are replaced at the right end by ones if the number is negative and zeroes if the number is positive. The sign bit is not shifted.

9. Rotate Accumulator and In-Out Right (5μsec)

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 one's (1's) in bits 9 through 17 of the instruction word.

10. Rotate Accumulator and In-Out Left (5μsec)

rcl N Operation Code 663

Rotates the bits of the combined register left in a single ring N positions, where N is the number of one's (1's) in bits 9 through 17 of the instruction word.

11. Shift Accumulator and In-Out Right (5μsec)

scr N Operation Code 677

Shifts the contents of the combined register right N positions, where N is the number of one's (1's) in bits 9 through 17 of the instruction word. As bits are shifted out from the right end of the IO they are replaced at the left end of the AC by ones if the number is negative and zeroes if the number is positive. The sign bit is not shifted.

12. Shift Accumulator and In-Out Left (5μsec)

scl N Operation Code 667

Shifts the contents of the combined registers right N positions, where N is the number of one's (1's) in bits 9 through 17 of the instruction word. As bits are shifted out from the left end of the AC they are replaced at the right end of the IO by ones if the number is negative and zeroes if the number is positive. The sign bit is not shifted.

B. Skip Group (5μsec)

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 0 (zero) or if the In-Out Register is positive.

a. The combined instruction would still take 5 microseconds.

b. The intents of any skip instruction can be reversed by making bit 5 (normally the Indirect Address bit) equal to one (1). For example, the skip on Zero Accumulator instruction, with bit 5 equal to one (1), becomes Do Not Skip on Zero Accumulator.

1. Skip on Zero Accumulator (5μsec)

sza Address 0100

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.

2. Skip on Plus Accumulator (5μ sec)

spa Address 0200

If the sign bit of the Accumulator is zero (0), the Program Counter is advanced one extra position and the next instruction in the sequence is skipped.

3. Skip on Minus Accumulator (5μsec)

sma Address 0400

If the sign bit of the Accumulator is a one (1), the Program Counter is advanced one extra position and the next instruction in the sequence is skipped.

4. Skip on Zero Overflow (5μsec)

szo Address 1000

If the overflow flip-flop is a zero (0), 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 only by an addition or subtraction that exceeds the capacity of the Accumulator. (See definition of add and subtract instructions.) 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 also cleared by the "start" switch.

5. Skip on Plus In-Out Register (5μsec)

spi Address 2000

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

6. Skip on Non-Zero In-Out Register (5μsec)

sni Address 4000

If the In-Out Register is not equal to plus zero (at least one bit is a one), the Program Counter is advanced one extra position and the next instruction in the sequence is skipped.

7. Skip on Zero Switch (5μsec)

szs Address 0010, 0020, 0030, ... 0070

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 0020, Switch 2, etc. Address 0070 senses all the switches. If Address 0070 is selected all 6 switches must be zero (0) to cause the skip. The instruction to skip on zero Sense Switch 1 would be szs 10, for sense switch 2, szs 20, etc.

8. Skip on Zero Program Flag (5μsec)

szf Addresses 0001 to 0007

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 0001 selects Program Flag 1, etc. Address 0007 selects all six program flags. All six must be zero to cause the skip. The instruction to skip on zero program flag 1 would be szf 1, for program flag 2, szf 2, etc.

The following skip group instructions are extended instructions; they are defined to represent the or-ing of existing skip instructions.

9. Skip on Zero or Minus Accumulator (5μsec)

szm

If the sign bit of the Accumulator is a one (1) or 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. This represents the or-ing of sza and sma.

10. Skip on Positive Quantity (5μsec)

spq

If the Accumulator is equal to plus zero (all bits are zero) or if the sign bit of the Accumulator is not a one (1), the Program Counter is advanced one extra position and the next instruction in the sequence is skipped. This represents the or-ing of sza and sma i.

11. Clear the Overflow Flip-Flop (5μsec)

clo

The overflow flip-flop is set only by addition or subtraction that exceeds the capacity of the AC. The overflow flip-flop is cleared by the instruction szo; it is not cleared by any arithmetic operations. This instruction clears the overflow flip-flop and then proceeds to the next instruction in the sequence. This represents the or-ing of szo, sma, spa, and i.

C. Operate Group (5μsec)

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. For example, the instruction opr 3200 will clear the AC, put the contents of the Test Word into the TW to AC, and complement the AC.

a. The combined instruction would still take 5 microseconds.

b. The order of operate class instructions is:

EFFECT	^->AC
	O->IO
		TW->AC, stf
		PC->AC, clf
			$\overline{AC}$->AC
			O->RUN
				O->MB
				(lai)
					IO->MB(lai)
					O->IO(lia)
						AC->MB(lia)
						MB->AC(lai)
							MB->IO
							(lia)

TIME	7	8	9	10	0	1	2
	Time pulses of this instruction				TPs of next instruction		

1. Clear the In-Out Register (5μsec)

cli Address 4000

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

2. Load the Accumulator from the Test Word (5μsec)

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 0200 (clear Accumulator), so that C(AC) will equal the contents of the Test Word Switches. (Thus, lat is defined as 762200 initially in POSSIBLE symbol table and ID symbol table.)

3. Load the Accumulator with the Program Counter (5μsec)

   lap   Address 0100

   Forms the inclusive OR of the C(AC) and the contents of the Program Counter (which contains the address of the instruction following the lap) in AC bits 6 through 17. Also, the inclusive OR of AC bit zero and the state of the overflow flip-flop is formed in AC bit zero. This instruction is usually combined with address 0200 (clear accumulator) so that the C(AC) will equal the contents of the overflow flip-flop (in AC bit zero) and the contents of the Program Counter (in AC bits 6 through 17). The contents of the Extend flip-flop are transferred to AC bit 1, the contents of the Extended Program Counter to bits 2, 3, 4, and 5. (Lap is initially defined as 760300 in POSSIBLE symbol table and ID symbol table.

4. Complement the Accumulator (5μsec)

   cma   Address 1000

   Complements (changes all ones to zeroes and all zeroes to ones) the contents of the Accumulator.

5. Halt

   hlt   Address 0400

   Stops the computer. When the computer is in Time-Sharing, this instruction is treated as an illegal instruction when executed.

6. Clear the Accumulator (5μsec)

   cla   Address 0200

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

7. Clear Selected Program Flag (5μsec)

clf   Address 0001 to 0007

Clears the selected Program Flag. Address 0001 clears Program Flag 1, 0200 clears Program Flag 2, etc. Address 0007 clears all program flags. Thus, the instruction to clear Program Flag 1 is clf 1, for Program Flag 2, clf 2, etc.

8. Set Selected Program Flag (5μsec)

stf   Address 0011 to 0017

Sets the selected program flag. Address 0011 sets Program Flag 1; 0012 sets Program Flag 2, etc. Address 0017 sets all program flags. Thus, if stf is defined as 760010, the instruction to set Program Flag 1 is stf 1, for Program Flag 2, stf 2, etc. (Stf is initially defined in POSSIBLE symbol table and ID symbol table as 760010.)

9 No Operation (5μsec)

nop   Address 0000

The state of the computer is unaffected by this operation, and the Program Counter continues in sequence.

10. Load the Accumulator from the In-Out Register (5μsec)

lai   Address 0040

This instruction copies the contents of the in-out register into the Accumulator. It happens after all normal operate class options. If the computer is stopped at the end of this instruction, the Memory Buffer Register will contain zero, and the old contents of the Accumulator will be shown. (See the order of implementation of the operate class instructions at the beginning of this section.)

11. Load the In-Out Register from the Accumulator (5μsec)

lia Address 0020

This instruction copies the contents of the Accumulator into the in-out register. It happens after all normal operate class options. If the computer is stopped at the end of this instruction, the Memory Buffer Register will contain zero, and the old contents of the In-Out Register will be shown. (See the order of implementation of the operate class instructions at the beginning of this section.)

The following operate group instructions are extended instructions; they are defined to represent the or-ing of existing operate instructions.

12. Swap Accumulator with the In-Out Register (5μsec)

swp

This instruction is the combination of lia and lai. It copies the contents of the Accumulator into the In-Out Register and the original contents of the In-Out Register is copied into the Accumulator. It happens after all normal operate class options. If the computer is stopped at the end of this instruction the Memory Buffer Register will contain zero, and the swap will not yet have occurred.

13. Clear and Complement the Accumulator (5μsec)

clc

This instruction represents the or-ing of cla and cma. It clears (sets equal to plus zero) the contents of the Accumulator and then complements (changes all ones to zeroes and all zeroes to ones) the contents of the Accumulator.

III. IN-OUT TRANSFER GROUP (5μ sec without in-out wait)

iot Operation Code 72

The in-out transfer command is used to perform all the in-out control and information transfer functions. The address of the iot instructions is used to select various devices. The decoding for the instructions is as follows:

Bits	Use
0-4	11101 instruction bit code for in-out transfer
5-6	Used for device synchronization
7-11	Useful for modification of iot instructions
12-17	Addresses 1 of 64 possible devices

NOTE: Quite often, input-output operations must be synchronized. That is, information is transferred which may cause the computer (or device) to "wait", and then proceed in synchronism. When several in-out devices operate simultaneously, the synchronization is essential. The control for this is coded in each in-out transfer command:

IN-OUT TRANSFER Command Bits		Wait for Completion Pulse for Restart/Continue without wait	Enable/Disable Completion (Done) Pulse Signal
5	6		
0	0	continue, no wait	Disable
0	1	continue, no wait	Enable
1	0	wait, then continue	Enable
1	1	wait, then continue	Disable

Bit 5 of the in-out transfer command designates whether the program is to wait for a completion pulse before continuing. The exclusive OR of bits 5,6 of the command specify whether the completion pulse return signal is to be enabled or disabled.

The description of each in-out instruction below is divided into two parts, NON-TS and TS, so that the user can understand the differences. Generally, in time sharing mode the executive routine buffers characters for the typewriter, punch, and reader so the user does not worry about synchronization. Almost all iot instructions trap to the executive routine where they are interpreted and serviced. Control is quickly returned to the user's program except when an output instruction would cause an overflow of an executive routine buffer. In this case, the program is dismissed and remains inactive until the buffer becomes almost empty.

A. Reader

The perforated tape reader of the PDP-1 is a photoelectric device capable of reading 400 lines per second. Three lines form the standard 18-bit word when reading binary punched 8-hole tape. Five, six, and seven-hole tape may also be read.

1. Read Paper Tape, Alphanumeric

   rpa Address 0001

   a. NON+TS: This instruction reads one line of tape (all eight channels) and transfers the resulting 8-bit code to the Reader Buffer. (NOTE: rpa is initially defined as 730001 in ID symbol table and in POSSIBLE symbol table.

      1) If bits 5 and 6 of the rpa instruction are different [730001≡(rpa) or 724001] the 8-bit code read from tape is automatically transferred to the IO Register via the Reader Buffer and appears as follows:

IO BITS	10	11	12	13	14	15	16	17
TAPE CHANNELS	8	7	6	5	4	3	2	1

The remaining bits of the IO are set to zero. In the case of 730001, the program waits until the material has been read and transferred and a completion pulse returned before continuing. In the case of 724001, the program is continued and a completion pulse is generated but the program waits for it later when a synchronizing ioh instruction is given.

2) If bits 5 and 6 of the rpa instruction are both zero [(rpa-i)=720001], the contents of the Reader Buffer must be transferred to the IO Register by executing a rrb instruction. When the Reader Buffer has information ready to be transferred to the IO Register, Status Register bit 1 is set to one. (Sequence break can be used in this situation to perform the read.)

b. TS: The reader must be assigned to the user's console before his program can execute a rpa instruction. (When the reader is not assigned, the rpa instruction is an illegal one.) The instruction is the same as in NON-TS but the line read is put into a pseudo reader buffer and bit 6 is ignored. The executive reader buffer routine is then filled with succeeding lines of tape. The contents of the pseudo reader buffer (prb) is then placed in the IO if the instruction was a rpa. Otherwise, on rpa-i, the material goes into the prb but isn't transferred to the IO until a rrb instruction is executed by the user. Material is taken from the reader buffer in the executive routine on the next rpa instructions. When the buffer is almost empty more tape is read in to refill it.

NOTE: The code of the off-line tape preparation typewriter (Friden FIO-DEC Recorder - Reproducer) contains an odd parity bit. This bit may be checked by read-in programs. The FIO-DEC code can then be converted to the Concise (6-bit) Code used by the PDP-1 merely by dropping the eighth bit (parity). A list of characters and their FIO-DEC and Concise Codes can be found in the appendix.

2. Read Paper Tape, Binary

rpb Address 0002

a. NON-TS: This instruction reads three lines of tape (six channels per line) and assembles the resulting 18-bit word in the Reader Buffer. For a line to be recognized by this instruction, Channel 8 must be punched (lines with Channel 8 not punched will be skipped over). Channel 7 is ignored. The instruction sub 5137, for example, appears on tape and is assembled by rpb as follows:

CHANNEL	8	7	6	5	4	3	2	1
LINE 1	x		x				x	
LINE 2	x		x		x			x
LINE 3	x			x	x	x	x	x

READER BUFFER 100 010 101 001 011 111

NOTE: Vertical dashed line indicates sprocket holes and the symbol "x" indicates holes punched on tape.

NOTE: rpb is initially defined as 730002 in ID symbol table and POSSIBLE symbol table.

1) If bits 5 and 6 of the rpb instruction are different [730002 ≐ rpb or 724002], the 18-bit word read from tape is automatically transferred to the IO Register via the Reader Buffer. In the case of 730001, the program waits until the material has been read and transferred and a completion pulse is generated but the program waits for it later when a synchronizing ioh instruction is given.

2) If bits 5 and 6 of the rpb instruction are both zero [rpb-i ≐ 720002], the contents of the Reader Buffer must be transferred to the IO Register by executing a rrb instruction. When the Reader Buffer has information ready to be transferred to the IO Register, status Register bit 1 is set to one. (Sequence break can be used in this situation to perform the read.)

b. TS: The reader must be assigned to the user's console before his program can execute a rpb instruction. (When the reader is not assigned, rpb is an illegal instruction.) This instruction is the same as in NON-TS but the 3 lines read are put into a pseudo reader buffer and bit 6 is ignored. The executive routine reader buffer is then filled with succeeding lines of tape. The contents of the pseudo reader buffer (prb) is then placed in the IO if the instruction was a rpb. Otherwise, on a rpb-i, the material goes into the prb but isn't transferred to the IO until a rrb instruction is executed by the user. Material is taken from the reader buffer in the executive routine on the next rpb instructions. When the buffer is almost empty more tape is read in to refill it.

3. Read Reader Buffer
rrb Address 0030

a. NON-TS: When the rpa or rpb instructions are given with bits 5 and 6 both zero (720001 ≡ rpa-i or 720002 ≡ rpb-i) information read from tape fills the Reader Buffer, but is not automatically transferred to the IO Register. To accomplish this transfer, these instructions must be followed by a rrb instruction. In addition, the rrb instruction clears Status Register bit 1.

b. TS: Same as above, but the information is stored in the pseudo reader buffer. It is transferred from there to the IO. Executing an rrb instruction is illegal when not preceded by iot 1 or iot 2.

B. Perforated Tape Punch

The standard PDP-1 Perforated Tape Punch operates at a speed of 63 lines per second. It can operate in either the alphanumeric or binary mode.

1. Punch Perforated Tape, Alphanumeric

ppa Address 0005

a. NON-TS: This instruction takes information from the IO Register and punches one line of tape in the following format:

IO Bits	10	11	12	13	14	15	16	17
Holes	8	7	6	5	4	3	2	1

NOTE: ppa is initially defined as 730005 in ID symbol table and POSSIBLE symbol table.

1) If bits 5 and 6 of the ppa instruction are different [730005 ≡ ppa or 724005], a completion pulse is generated.

a) In the case of bit 5 being on and bit 6 off, the program waits until the material has been punched and a completion pulse returned before continuing.

b) In the case of bit 5 being off and bit 6 on, the program is continued, a completion pulse is generated and the program will wait for it later when a synchronizing ioh instruction is given.

2) If bits 5 and 6 are both zero in the ppa instruction [ppa-i ≡ 720005], the program is continued and no completion pulse will be given. In this case, the Status bit must be checked for. [Sequence break can be used in this situation to perform the punching.]

b. TS: Same as above but transfers one character from IO to the executive routine punch buffer for transfer to the punch when it is ready. If the buffer is full, the user's quantum ends. Both bit 5 and 6 are completely ignored. Time - about 500 microseconds.

C. Alphanumeric On-Line Typewriter

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

1. Type Out

tyo Address 0003

a. NON-TS: For each in-out transfer instruction, tyo, one character is typed. The character is specified by the right six bits of the IO Register.

NOTE: tyo is initially defined as 730003 in ID symbol table and POSSIBLE symbol table.

1) If bits 5 and 6 of the tyo instruction are different [730003 ≡ tyo or 724003] a completion pulse is generated.

a) In the case of bit 5 being on and bit 6 off, the program waits for the completion pulse before continuing.

b) In the case of bit 5 being off and bit 6 on, the program is continued, a completion pulse is generated and the program will wait for it later when an ioh instruction is given.

2) If bits 5 and 6 are both zero in the tyo instruction [tyo-i ≡ 720003], the program is continued and no completion pulse will be given. In this case, the status bit must be checked for. [Sequence Break can be used in this situation to perform the typing.]

b. TS: Same as above but places a character from the IO register into the executive routine user's console buffer to be printed when the typewriter is ready. If his buffer is full, the user's quantum ends. The console is placed in print status by the executive routine and characters are printed from the buffer until it is empty. Then the console returns to type status. Both bits 5 and 6 are completely ignored. Time: about 500 microseconds.

2. Type In

tyi Address 0004

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

a. NON-TS: When a typewriter key is stuck, the 6-bit code for the struck key is placed in the typewriter buffer, program flag 1 is set, and the type-in status bit, bit 3, is set to one. A program designed to accept typed-in data would periodically check program flag 1 or bit 3 of the status register and if found to be set to a one, a tyi instruction could be executed for the information to be transferred to the In-Out Register. This in-out transfer should not use the optional in-out wait. The information contained in the typewriter buffer is then transferred to the right six bits of the In-Out Register. The tyi instruction clears the type-in Status bit and program flag 1.

b. TS: When a typewriter key is struck, the 6-bit code for the struck key is placed in the console's typewriter buffer, the type-in Status, bit 3, is set to one. [Program Flag 1 is not set when a typewriter key is struck.] The tyi instruction alone can replace the listen loop described above to accept typed-in data. Executing this in-out transfer instruction causes a character from the typewriter buffer to be transferred to the right six bits of the In-Out Register. If there are no characters in the buffer, the program is made inactive until a key has been struck.* Programs which perform other operations while waiting for type-in must use a check Status loop for detecting the striking of a key. This in-out transfer should not use the optional in-out wait. The tyi instruction clears the type-in status bit. Time: about 500 microseconds.

* The executive routine accepts manually typed characters and enters then in the buffer while the console is in type status. If the buffer becomes filled, the type-in light will go out and further characters typed will be lost.

D. Precision CRT Display (Type 30)

The sixteen-inch cathode ray tube display is intended to be used as an on-line output device for the PDP-1. It is useful for high speed presentation of graphs, diagrams, drawings, and alphanumerical information. The unit is solid state, self-powered and completely buffered. It has magnetic focus and deflection. The cathode ray tube has a P7 phosphor; thus a point displayed persists for a relatively long time. Some other display characteristics are

1024 by 1024 addressable locations
fixed origin at center of CRT
Plots 20,000 points per second
Ones complement binary arithmetic
Random point plotting
Accuracy of points ± 3 per cent of raster size
Raster size 9.25 by 9.25 inches.

Resolution is such that 512 points along each axis are discernible on the face of the tube.

1. Display One Point on CRT

   dpy Address 0007

   a. NON-TS: This instruction clears the light pen status bit and displays one point using bits 0 through 9 of the AC to represent the (signed) x coordinate of the point and bits 0 through 9 of the IO as the (signed) Y coordinate.

   (NOTE: dpy is initially defined as 730007 in ID symbol table and POSSIBLE symbol table.)

   1) If bits 5 and 6 of the dpy instruction are different [730007 or 724007] a completion pulse is generated.

      a) In the case of bit 5 being on and bit 6 off [730007], the program waits until the point is displayed and a completion pulse returns before continuing calculations in the program.

b) In the case of bit 5 being off and bit 6 on [724007], the program proceeds with the calculations, a completion pulse is generated when the point has been displayed, and the program gives later a synchronizing "wait" instruction which synchronizes the display.

2) If bits 5 and 6 are both zero in the dpy instruction [720007], the program does not halt and the completion pulse from the display is not enabled. Thus, the program must only refrain from giving the display commands too frequently; each dpy instruction takes 50 microseconds.

b. TS: The display has been assigned to all user's; if more than one active program uses the scope, it will display for whichever program is in core. The dpy instruction is the same as above. Display instructions do not trap to the executive routine, but are interpreted by the hardware. The instruction dpy (730007) displays a point and waits 50 microseconds before resuming operation of the program. There is no waiting when a display instruction does not have an indirect bit (i.e. dpy-i or dpy-400), but the hardware will delay execution of the next display command, so that display instructions cannot be given too rapidly in time-sharing.* When a console's Display Lever is depressed, the program running at that console is given an extended running time in core. This facilitates observing information on the scope.

* If an interrupt occurs during the in-out halt of a display instruction, the current point will be displayed a second time upon dismissal. This can be avoided by using the sequence

```
dpy-i
ioh
```

instead of

```
dpy
```

E. Light Pen (Type 32)

NON-TS AND TS: The light pen is designed to be used with the CRT DISPLAY TYPE 30. By "writing" on the face of the CRT, stored or displayed information can be expanded, deleted, or modified. Specifically, each time a light-pulse strikes the pen, the Light Pen status bit is set to one and Program Flag 3 is set to one. At the time that status bit is set, x-y coordinates of the point just "seen" by the pen are in bits 0 through 9 of the AC and bits 0 through 9 of the IO, respectively (if the Display Instruction was given with the optional in-out wait). A program designed to accept Light Pen input would periodically check the Light Pen status bit and when found to be set, would store the C(AC) and C(IO) thereby defining the point just "seen" by the pen.

F. Drum

The PDP-1 drum is a high-speed magnetic drum used for storage. It is divided into 22_{10} fields of 4096 words each. Words are transferred between the drum and the core memory under automatic control. The drum runs at 30 revolutions per second; thus, each word on the drum is available once every 33 1/3 milliseconds. When a drum operation has begun, words are transferred at a rate of 8.16 microseconds each. In a single operation, information can be written on the drum, read from the drum or both simultaneously.

1. Drum Initial Address

   dia Address 0060

   a. NON-TS: This instruction causes the $C(IO)_{1-5}$ to be sent to the drum write field buffer. These bits specify which field of the drum will be written on during the next dcc instruction or if $C(IO)_{1-5}$ is zero, that no write operation is to occur. The $C(IO)_{6-17}$ are sent to the drum initial address register to specify the first drum address to be transferred.

   b. TS: The field number employed in a user's program is a pseudo field number unless specified as absolute by the sign bit being on. The pseudo field number is translated into its absolute field number and checked for validity by the executive routine. Time: about 150 microseconds.

2. Drum Break on Address

   dba Address 0061

   a. NON-TS: This instruction causes the $C(IO)_{6-17}$ to be sent to the drum initial address register. When the current drum address becomes equal to the contents of the initial address register, a sequence break request is indicated. Bit 5 of the status word is set by the break, and is cleared by the next dcc instruction.

b. TS: The address given will be advanced by approximately 50 words to allow for processing by the executive routine. Occurrence of the sequence break will interrupt the user's program and turn on the drum status bit. The status bit is cleared as above. Time: about 200 microseconds.

3. Drum Count and Commence

dcc **Address 0062**

a. NON-TS: This instruction causes the $C(IO)_{1-5}$ to be sent to the drum read field buffer. These bits specify which field will be read, or if $C(IO)_{1-5}$ is zero, that no read operation is to occur. The $C(AC)_{2-5}$ specify which core will be used for the data transfer; $C(AC)_{6-17}$ specify the first core memory address of the data to be transferred. The $C(IO)_{6-17}$ specify the number of words to be transferred. If the $C(IO)_{6-17}$ is zero, 4096 words are transferred. While the dcc instruction is being executed, the computer stops and the drum system takes full control of the core memory. Successive words are transferred from sequential locations until the operation is complete. If no errors* occurred during the drum operation, the instruction following the dcc will be skipped. The C(AC) and the C(IO) are lost during this operation. If both read field and write field are non-zero (both reading and writing operations are specified) the contents of memory are written on the write field; then the read field data are read into memory. The read field must not equal the write field. In order to avoid passing a given drum address, and hence losing 33 milliseconds, the dcc instruction must be given at least 250 microseconds before the drum address reaches the initial address.

b. TS: The field number employed in a user's program is a pseudo field number (unless specified by the sign bit being on). The pseudo field number is translated into its absolute field number and checked for validity by the executive routine. Time: about 300 microseconds plus access and transfer time.

4. Drum Read Address

dra iot 63

a. NON-TS: This instruction causes the current drum address to be read into IO_{6-17}. The parity error flag is read into IO_0; and the timing error flag is read into IO_2. Two cycles elapse before this information is placed in the IO.

b. TS: The address read will be about 50 words in advance of the actual position of the drum to allow for processing by the executive routine. Time: about 100 microseconds.

* The various error conditions are as follows:

1. Read Error: When a word is written on the drum, its parity is generated and is also written on the drum. Whenever this word is read off the drum, a new parity is generated and is checked against the parity just read. A mismatch sets the read error flip-flop.
2. Selection Error: A five-bit address can specify 32 drum fields out of which 23 are legal and 9 are illegal. Field 0 means no selection, fields 1 through 22 are legal fields and fields 23 through 32 are illegal fields. A selection error flip-flop will be set if a user selects an illegal field.
3. Timing Error: This error monitors the drum clock circuit malfunctions. The timing flip flop will be set when (a) the time period between consecutive clock pulses is not equal to ~8.2 microseconds, (b) number of clock pulses available on the drum is not equal to 4096, and (c) the reference index pulse is lost.

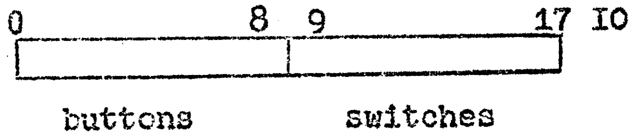
C. Buttons and Switches

Four consoles of nine buttons and nine switches each can be added to the PDP-1 time-sharing system to facilitate the communication between the user and the computer.

1. Read Buttons

rbt Address x237 where $0 \leq x \leq 3$

TS: This instruction is available only in Time-Sharing. The button console must be assigned to the user's console before his program can execute rbt instruction. [When the button console is not assigned, the rbt is an illegal instruction.] The x in the instruction address corresponds to bits 8-9 of the rbt instruction and indicates one of the four possible sets of buttons and switches.* This instruction reads the value of the buttons and switches into the IO register. The right nine bits of the IO register will contain the state of the switches from left to right (1 meaning on, 0 meaning off); the resulting left nine bits of the IO will contain the state of the buttons from left to right.



* For the programmer, sets of buttons are numerated from 0 up, regardless of whether the sets are contiguous or not during assignment.

H. Knobs

Four consoles of four analog-devices each can be added to the PDP-1 time sharing system to facilitate the communication between the user and the computer. One such console, a console of four knobs, has been added. Analog voltage from 0 to -10 volts from these devices can be converted into a binary word which can be placed in the IO register.

1. Check Knobs

ckn Address xx27 where $0 \leq x \leq 3$

TS: This instruction is available only in Time-Sharing. The Analog console must be assigned to the user's console before his program can execute a ckn instruction. [When the device is not assigned, the ckn is an illegal instruction.] This instruction reads the value of the selected analog device into the lower 8 bits (bits 10-17) of the IO register. The first "x" of the instruction (bits 8-9) determines which of the four possible analog consoles is selected while the second "x" (bits 10-11) indicate the specific knob device (numbered 0 to 3 from left to right) on the selected console that is to be read. Other bits are unused. For the programmer, sets of analog-devices are numerated from 0 up, regardless of whether the sets are contiguous or not during assignment.

I. External Register

A 18-bit register has been added to the Time Sharing facilities of the PDP-1 computer to aid communication between programs and in-out devices. This External Register can be assigned in two modes, shared or absolute. When it is absolutely assigned, only one user may have it; on shared assignment, it is possible for more than one user to use it.

1. Load External Register from the Accumulator

   lea   Address 4677

   This instruction copies the contents of the Accumulator into the External Register. This instruction is available only in Time Sharing. The external register must be assigned to the user's console before his program can execute a lea instruction. [When the external register is not assigned, the lea is an illegal instruction.]

2. Load External Register from the In-Out Register

   lei   Address 4577

   This instruction copies the contents of the External Register into the IO Register. This instruction is available only in Time-Sharing. The external register must be assigned to the user's console before his program can execute a lei instruction. [When the external register is not assigned, the lei is an illegal instruction.]

3. Read External Register into the In-Out Register

   rer   Address 4777

   This instruction copies the contents of the External Register into the IO Register. This instruction is available only in Time-Sharing. The external register must be assigned to the user's console before his program can execute a rer instruction. [When the external register is not assigned, the rer is an illegal instruction.]

J. Sequence Break Mode

The purpose of the Sequence Break Mode (or program interrupt) is to allow concurrent operation of several in-out devices and the main program sequence. It also provides a means of indicating to the computer than an in-out device is ready to accept or furnish data.

a. NON-TS: The standard one channel sequence break mode is the only mode available in NON-TS. Interrupt requests can be received from any number of in-out devices. Each such request sets a unique status bit. If the channel is free, the main program sequence is interrupted after completion of the current memory cycle and the C(AC) are automatically stored in memory location 0, the C(PC) in location 1, and the C(IO) in location 2. The time required to accomplish this is 15μsec. The C(PC) as stored in location 1 includes the state of the overflow flip-flop in bit 0. The Program Counter is then reset to the address 0003 and the program begins operating in the new sequence. The program beginning at location 0003 is usually designed to inspect the Status bits, through the use of the Check Status Instruction, to determine which in-out device caused the interrupt. A jump to the appropriate in-out subroutine can then be executed. Each such subroutine is terminated by the following instructions:

```
lac   0000  /to restore the AC
lio   0002  /to restore the IO
jmp i 0001  /to resume the main program
```

The last of these three instructions restores the overflow and the PC flip-flops and frees the channel thus, allowing the next interrupt request received by the system to be processed. Interrupt requests that occurred while the channel was busy set status bits, and cause interrupts when the channel next becomes free. The read, punch, typewriter, light pen, and drum are attached to the one-channel Sequence Break System and seven status bits are defined (see Check Status Instruction). Three instructions are directly associated with the One-Channel Sequence Break System on the standard PDP-1:

1. Enter Sequence Break Mode

   esm Address 0055

   This instruction turns on the Sequence Break System, allowing automatic interrupts to the main sequence to occur.

2. Leave Sequence Break Mode

   lsm Address 0054

   This instruction turns off the Sequence Break System, thus preventing interrupts to the main sequence. Should interrupt requests occur while the system is off, the status bit will, nevertheless, continue to be set. This instruction should be given in the main sequence, not in the program servicing the break.

3. Clear Sequence Break System

   cbs Address 0056

   This instruction clears certain control flip-flops in the Sequence Break System thus nullifying the effect of any interrupt requests just granted or about to be granted (i.e. just prior to the transfer of the C(AC) to lo:ation zero).

b. TS:

1) The standard one channel sequence break system described above can be used in time-sharing also. The sequence break instructions do not trap to the executive routine. The features are the same as above except for the following notes below. The reader, punch, typewriter, light pen, drum, and external equipment level are attached to the one channel sequence break system in TS and seven status bits are defined (see check status instruction, cks). The table below lists the events which will interrupt a user's program through the sequence break system. All interrupts to a user's program will be delayed until his program is granted a quantum, placed in operation, and the machine is in sequence break mode. When a user's quantum ends the contents of the break waiting, break started, and sequence break mode flip flops are preserved together with the status of the AC, IO, PC, and program flags, and are restored before the program is placed in operation for its next quantum.

Device	Event Causing Interrupt
Light Pen	None
Paper Tape Reader	Information becomes available in the reader buffer and the reader is assigned.
Paper Tape Punch	The punch is assigned and space in the executive routine punch buffer becomes available for another character, or a ppa or ppb trap occurs and the buffer does not just become full.
Console	The console is not in print status and a typed-in character enters the previously empty executive routine console buffer, or a tyi trap occurs and the buffer does not just become empty.
	The console is in print status and space in the executive routine console buffer becomes available for another character, or a tyo trap occurs and the buffer does not just become full.
Drum	A drum break return occurs.
External Equipment	An interrupt signal is received from the external equipment for which the assignment level is on.

One additional instruction for Time-Sharing Operation of the one channel sequence break system exists. The instructions explained in the NON-TS section are the same.

4. Initiate Sequence Break

   isb Address

   This instruction turns on the break waiting flip flop to indicate that the program would like an interrupt to service a device.

2) Also a pseudo sixteen channel sequence break system is available in time sharing. This system has sixteen automatic interrupt channels arranged in a priority chain, the lowest channel having the highest priority. A break to a particular sequence can be initiated by the completion of an in-out device, the program, or any external signal. Breaks cannot occur within breaks; they are stored and serviced after the present one is dismissed. If more than one break occurs at the same time, the break with the lowest channel member will be serviced first. When a break occurs, the C(AC) are automatically stored in memory location 0, the C(PC) in location 1, the C(IO) in location 2, and the number of the channel that caused the break is placed in the AC. The C(PC) as stored in location 1 includes the state of the overflow flip-flop in bit 0. The Program Counter is then reset to the address 0003 and the program begins operating in the new sequence. For a break to occur, the computer must be in sequence break mode, the new mode must be on, and the status bit and the corresponding enable channel associated with this device must be on. The program beginning at location 0003 usually contains a dispatch table since the C(AC) indicates which in-out device caused the interrupt. A jump to the appropriate in-out subroutine can then be executed. Each such subroutine is terminated by the following instructions:

```
lac 0000  /to restore the AC
lio 0002  /to restore the IO
jmp i 0001  /to resume the main program
```

The last of these three instructions restores the overflow and the PC flip-flops and frees the sequence break system thus allowing another interrupt to be processed. When a service request is completed, the corresponding status bit should be OFF.

The reader, punch, typewriter, light pen, drum and a set of switches and knobs are each assigned a unique channel in the pseudo 16 channel sequence break system. Provision has also been made for the connection of a user's special external equipment to be two higher and the two lower priority channels. The channels and their corresponding IO devices in the order of their priority are:

Channel	Device
0 1	High priority for user's external equipment
4	buttons
5	light pen
6	type-in
7	type-out
10	punch
11	drum
12	reader
16 17	low priority for user's external equipment

The three instructions associated with the one channel sequence break system of the non-time sharing system are supplemented by six additional instructions for using the new sequence break system.

1. New Mode One
   nmn Address 5377
   This sets the Sequence Break System to new 16 channel mode.
2. New Mode Off
   nmf Address 5477
   This sets the Sequence Break System to old 1 - channel mode.
3. Deactivate Sequence Channel N
   dsc N × 100 Address 0050 + (N × 100)
   This turns off channel N in the new mode Sequence Break System, thus not making it possible for an interrupt to occur on this channel.

4. Activate Sequence Break Channel N
   asc Nx100 Address 50+(Nx100)
   This instruction turns on channel N in the new mode sequence break system, thus making it possible for an interrupt to occur on this channel.
5. Clear All Channels
   cac Address 53
   This instruction turns off all sixteen channels in the new mode sequence break system, thus not making it possible for any further interrupts to occur until a channel is activated.
6. Skip On Break Request
   sbr Address 2577
   If a break has been requested and is waiting to be serviced, the Program Counter is advanced one extra position and the next instruction in the sequence is skipped.

NOTE: Sequence Break operations with extended memory use Memory Module zero, locations 00 through 03 to store the C(AC), C(PC), and C(IO) and to jump to the new sequence. The C(PC) as stored in location 01 includes the state of the overflow flip-flop in bit zero, the state of the indirect address mode (extend=1, normal=0) in bit 1, and the contents of the extended Program Counter in bits 2 through 17. At the beginning of servicing a sequence break, the overflow and indirect address mode flip-flops are automatically set to zero. The indirect jump that terminates a sequence break requires that the Sequence Break System be on, temporarily places the computer in the extend mode so that a 16-bit exit address may be obtained then restores the overflow, indirect address mode, and PC flip-flops to their previous stages (i.e., just prior to the beginning of the sequence break).

K. Memory Module and Memory Extension Control

The standard PDP-1 has a 4096, 18-bit word Memory Module. Additional such modules are connected to the PDP-1 for expanding memory capacity. A Memory Extension Control is needed to expand memory in increments of 4096-word modules beyond the standard 4096 words. This control provides a single-level, indirect address mode called "extend", in addition to the normal multiple-level, indirect address mode of the standard PDP-1. A toggle switch labelled "extend", which is on the main control console, provides for initial selection of the extend or normal mode in conjunction with the use of the START OR READ-IN push buttons. During the operation of a program or of time-sharing, the extend or normal mode can be selected as required through the use of two instructions:

1. Enter Extend Mode (5μsec)

   eem   Address 4074

   NON TS and TS: This instruction places the computer in the single-level, indirect address mode called "extend". In this mode, all memory reference instructions that are indirectly addressed refer to the location of a word which is taken as a 16-bit effective address. This address is contained in bits 2 through 17 of the specified word. The Program Counter (PC) and the Memory Address Register (MA) both become 16-bit registers. As in multiple-level, indirect address mode, the instructions jsp, jda, cal, and lap (with address 300) supply the AC with the state of the overflow flip-flop in bit zero, the state of the indirect address mode (extend=1, normal=0) in bit 1, and the contents of the extended Program Counter in bits 2 through 17. Instructions not indirectly addressed are executed as in the standard PDP-1, but refer to the 4096 words in the memory module designated by the program counter extension, PC bits 2 through 5. Only bits 6 through 17 of the extended Program Counter act as a counter. Therefore, unless a transfer of control is indicated, an instruction in location 7777 is followed by the instruction in location 0000 of the same memory module, as specified by PC bits 2 through 5. In the extend mode, the cal instruction uses memory location 0100 and 0101 in memory module designated by the extended Program Counter, PC bits 2 through 5.

2. Leave Extend Mode (5μ sec)

   lem Address 0074

   NON TS and TS: This instruction places the computer in the multiple-level, indirect address mode called "normal". In this mode, the PDP-1 operates as usual and all addressing refers to the 4096 words in the memory module designated by the program counter extension, PC bits 2 through 5. As in the extend mode, the instructions jsp, jda, cal, and lap (with address 300) supply the AC with the contents of the overflow, indirect address mode, and PC flip-flops. In the normal mode, the cal instruction uses memory locations 0100 and 0101 in the memory module designated by the program counter extension, PC bits 2 through 5.

NOTE: Sequence Break operations with extended memory use Memory Module zero, locations 00 through 03 to store the C(AC), C(PC), and C(IO) and to jump to the new sequence. The C(PC) as stored in location 01 includes the state of the overflow flip-flop in bit zero, the state of the indirect address mode (extend=1, normal=0) in bit 1, and the contents of the extended Program Counter in bits 2 through 17. At the beginning of servicing a sequence break, the overflow and indirect address mode flip-flops are automatically set to zero. The indirect jump that terminates a sequence break requires that the Sequence Break System be on, temporarily places the computer in the extend mode so that a 16-bit exit address may be obtained then restores the overflow, indirect address mode, and PC flip-flops to their previous stages (i.e., just prior to the beginning of the sequence break).

I. Check Status

cks Address 0033

This instruction checks the status of various in-out devices and sets IO Bits 0 through 6 for subsequent program interrogation as follows:

a. NON-TS:

IO Bit Position	Status Register Definitions
0	Set to 1 when light-pulse from a dpy instruction strikes pen. Set to 0 at the start of each dpy instruction.
1	Set to 1 when Paper Tape Reader Buffer has information ready to be transferred to IO Register. Set to 0 by the reader return pulse or by the rrb instruction.
2	Set to 1 when typewriter is free to receive a tyo instruction. Set to 0 at the start of each tyo instruction.
3	Set to 1 when typewriter key is struck. Set to 0 by the completion of tyi instruction.
4	Set to 1 when paper tape punch is free to receive a ppa or ppb instruction. Set to 0 at the start of each ppa or ppb instruction.
5	Set to 1 when drum address equals address specified by dba instruction. Set to 0 by the dcc instruction.
6	Set to 1 on entering the Sequence Break Mode. Set to 0 on leaving the Sequence Break Mode.

D. TS:

IO Bit Position	Status Register Definitions
0	Set to 1 when the light-pulse from a dpy instruction strikes the pen. Set to 0 at the start of each dpy instruction.
1	Set to 1 by interrupt which occurs when information becomes available in the reader buffer. Set to 0 by rpa or rpb command.
2	Set to 1 when executive routine can accept a character via a tyo trap. Set to 0 at the start of each tyo instruction.
3	Set to 1 when executive routine can supply a character via a tyi trap. Set to 0 by the completion of tyi instruction.
4	Set to 1 when executive routine can accept a character via a ppa or ppb trap. Set to 0 at the start of each ppa or ppb instruction.
5	Set to 1 when drum break return occurs. Set to 0 by the dcc instruction.
6	Set to 1 on entering the Sequence Break Mode. Set to 0 on leaving the Sequence Break Mode.

M. Miscellaneous

The following in-out transfer instruction is an extended instruction; it is defined to wait for any completion pulse.

1. In-Out Halt

   ioh 730000

   a. NON-TS: This instruction performs nothing but an in-out wait. The computer will enter the special waiting state until a device which previously gave an in-out transfer with no in-out wait (bit 5 zero and bit 6 one) returns the in-out restart pulse. If the device has already returned the completion pulse before the ioh instruction, the computer will proceed immediately.

   b. TS: The ioh instruction waits for the display completion after a dpy-400 has been executed; otherwise, it is equivalent to a nop in the time-sharing system since either the hardware or the executive routine maintains efficient timing of in-out devices.

IV. INSTRUCTIONS FOR COMMUNICATION BETWEEN USER AND THE TIME-SHARING SYSTEM

The following instructions were added to the time-sharing PDP-1 system so that the user could communicate to the system itself. These are available only in time-sharing mode and are transparent to the accumulator and the in-out register. All these instructions trap to the executive routine where they are interpreted and serviced. Control is returned to the user's program upon completion of the instruction.

1. Administrative Request

a arq Address 2277

This instruction is used in the time sharing system to assign and deassign in-out equipment and additional drum fields to users. (If the device is not assigned to the user, the instructions corresponding to that device are treated as illegal.) The particular assignment or deassignment requested by this instruction is indicated by mnemonic codes. Concise codes for these mnemonics are placed in the AC and any additional information necessary for the request is placed in the IO before the arq instruction is executed. On the next two pages is the table of possible requests.

If the assignment or deassignment of fields is successful, the instruction following the arq will be skipped. For other assignments and deassignments, the instruction following the arq will be skipped only on successful assignment. An assignment will be successful if the field(s) or device requested is not already assigned or if the assignment is already in effect.

MNEMONIC WHOSE CONCISE CODE IS CONTENTS OF AC	CONTENTS OF IO	REQUEST
-r	---	dismiss reader
r	---	assign reader
-p	---	dismiss punch
p	---	assign punch
-x	---	dismiss external register
ax	---	assign external register absolutely.
sx	---	assign external register in shared mode.
k	M	assign or deassign analog-to-digital consoles. M is a 4-bit mask for consoles to be assigned (or left assigned, in the case of deassignment) to the user.
b	M	assign or deassign button consoles. M is a 4-bit mask for the consoles to be assigned (or left assigned, in the case of deassignment) to the user.
q1	---	assign external level 1
q2	---	" " " 2
q3	---	" " " 3
q4	---	" " " 4
q5	---	" " " 5
q6	---	" " " 6
q7	---	" " " 7

MNEMONIC WHOSE CONCISE CODE IS CONTENTS OF AC	CONTENTS OF IO	REQUEST REQUEST
-q	---	deassign external level
-f	---	dismiss all fields
f	N×10000	get a total of N fields
-1f	---	dismiss one field
1f	---	assign one field; returns with pseudo field just assigned in high part of AC.
af	A×10000+P	assign absolute field A (or the first available field if A=0) to pseudo field P (or the first available pseudo field if P=0). Returns with pseudo field in high part of AC.
-af	A×10000+P	<u>case 1</u>: P=x, A=0. deassign pseudo field x and the absolute field assigned to it. <u>case 2</u>: P=0, A=y. deassign absolute field y and the pseudo field assigned to it. <u>case 3</u>: P=x, A=y. deassign pseudo field x and the corresponding absolute field y. If x does not correspond, no deassignment is done and the request is unsuccessful. <u>case 4</u>: P=0, A=0. No deassignment is done, but the request is successful.
tf	---	translate pseudo field P and returns with its absolute field number in high part of AC.
0	---	dismiss to the administrative routine, MYSTIC

2. Dismiss

dsm Address 2377

This instruction is available only in time-sharing so that a user can dismiss his program and can return control to ID or the Administrative Routine. ID is brought back into control and types a carriage return if a ddt field has been requested previously. Otherwise, the Administrative Routine is given control and types out "dismissed", indicating that the program was dismissed because of a dsm instruction.

3. Breakpoint

bpt Address 2177

This instruction is available only in TS. It is provided so that ID can insert it into the user's program where breakpoints are requested for debugging. When the user transfers control from ID to his program, a bpt instruction replaces the instruction previously at the breakpoint location. When this instruction is encountered, the address at which the trap occurred is typed and is followed by a ")" and a tab. Then the previous contents of that register are typed out and the current location pointer is set to that address. If a bpt is encountered at a location where a breakpoint was not assigned to the user through ID, then ID interprets the instruction as illegal.

NOTE: Do not attempt to break at a program-modified instruction or at an instruction in the middle of a chain of indirect addressing.

4. Wait

wat Address 2477

This instruction is available only in time-sharing and is provided so that the user may deactivate his program and wait until an external device is available. When the latter is available, the program will automatically be activated again.

Alphanumeric Codes By Character

CHARACTER LOWER	CHARACTER UPPER	FIO DEC CODE	CONCISE CODE
a	A	61	61
b	B	62	62
c	C	263	63
d	D	64	64
e	E	265	65
f	F	266	66
g	G	67	67
h	H	70	70
i	I	271	71
j	J	241	41
k	K	242	42
l	L	43	43
m	M	244	44
n	N	45	45
o	O	46	46
p	P	247	47
q	Q	250	50
r	R	51	51
s	S	222	22
t	T	23	23
u	U	224	24
v	V	25	25
w	W	26	26
x	X	227	27
y	Y	230	30
z	Z	31	31

CHARACTER LOWER	CHARACTER UPPER		FIO DEC CODE	CONCISE CODE
0	→	(right arrow)	20	20
1	"	(double quotes)	01	01
2	'	(single quote)	02	02
3	~	(not)	203	03
4	⊃	(implies)	04	04
5	∨	(or)	205	05
6	∧	(and)	206	06
7	<	(less than)	07	07
8	>	(greater than)	10	10
9	↑	(up arrow)	211	11
(	[		57	57
)	]		255	55
‾	|	(non-spacing overstrike and vertical)	256	56
−	+	(minus and plus)	54	54
·	_	(non-spacing middle dot and underline)	40	40
,	=		233	33
.	X	(period and multiply)	73	73
/	?		221	21

	FIO-DEC CODE	CONCISE CODE
Lower Case	272	72
Upper Case	274	74
Space	200	00
Backspace	75	75
Tab	236	36
Carriage Return	277	77
Tape Feed	00	00
Red*	—	35
Black*	—	34
Stop Code	13	—
Delete	100	—

*Used on Type-Out only, not on keyboard