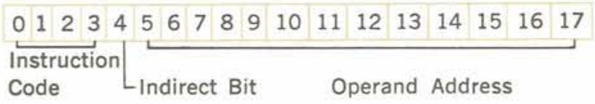
PROGRAMMING

The PDP-4 instruction format includes 4 bits for instruction code, 1 bit for indirect modified addressing and 13 bits for memory address or variations of the basic instructions.



When the indirect bit is a ONE, indirect addressing (or deferring) is specified. A defer memory cycle is required during which time the contents of the memory cell addressed are selected and the address part of this cell is used as the effective address of the original instruction. The instruction part of the cell and the indirect bit are ignored when obtaining the effective address. In addition, if the cell indirectly addressed is 10_8-17_8, a ONE is added to the contents of that cell before the address part is used as the effective address (auto-indexing).

Operating times of PDP-4 instructions are in multiples of the 8 microsecond memory cycle. Add, deposit, and load for example, are two-cycle instructions completed in 16 microseconds. Input-output connections are programmed by specifying iot instructions which affect the state of selected devices. The instructions may be microprogrammed to allow one basic instruction to handle one or more devices by changing the bits of the command.

MEMORY REFERENCE INSTRUCTIONS

Addressable or memory reference instructions which contain a memory address. The address portion of the instruction word specifies the location of an operand in the memory.

Mnemonic Code	Octal Code	Time (μsec)	Operation
cal	00	16	Same as jms 20. The address portion of this instruction is ignored. The cal instruction may be used for calling subroutines via a master central program which keeps track of exit addresses, allocates storage, and supplies parameters to the subroutines.
dac Y	04	16	Deposit Accumulator. C(AC)* are deposited in memory register Y. The C(AC) are unaffected by this operation.
jms Y	10	16	Jump to Subroutine. C(PC) are deposited in memory register Y. The next instruction will be taken from Y + 1, the beginning of the subroutine.
dzm Y	14	16	Deposit zero in memory. The contents of register Y are changed to zero. The original contents of Y are lost.
lac Y	20	16	Load AC. The C(Y) replace the C(AC). The previous C(AC) are lost. The C(Y) are unaffected.
xor Y	24	16	Exclusive OR. The exclusive "OR" logical function is performed on a bit-by-bit basis between the C(AC) and C(Y). The result is left in the AC and the original C(AC) are lost.
add Y	30	16	Add (ONE's Complement). The C(Y) are added to the C(AC) in ONE's complement arithmetic. The result is left in the AC and the original C(AC) is lost. This type add instruction is commonly used for most arithmetic. The Link bit is set to a ONE if the sum of the magnitude of C(Y) and C(AC) is greater than $2^{17} — 1$.
tad Y	34	16	Add (TWO's Complement). The C(Y) are added to C(AC) in TWO's complement arithmetic. If there is a carry out of bit 0, the Link will be set to ONE. This type of add instruction is useful in multiple precision arithmetic.

* C(AC): contents of the accumulator.

PDP-4 Instructions

Mnemonic Code	Octal Code	Time (μsec)	Operation
xct Y	40	8+ instruction execute time	The instruction in register Y will be executed. The computer will act as if the instruction located in Y were in the place of the xct Y.
isz Y	44	16	Index and Skip if zero. The C(Y) are replaced by C(Y) + 1. The C(AC) are unaffected by this instruction. The addition is done using two's complement arithmetic. If the sum is +0, the next instruction is skipped.
and Y	50	16	Logical AND. The logical "AND" function is performed on a bit-by-bit basis between C(AC) and C(Y). The result is left in the AC and the original C(AC) are lost.
sad Y	54	16	C(Y) are compared with the C(AC). If the two numbers are different, the next instruction in the sequence is skipped. The C(AC) and C(Y) are both unaffected by the instruction.
jmp y	60	8	Jump. The C(PC) are reset to address Y. The next instruction to be executed is taken from memory register Y. The original contents of the PC are lost.

AUGMENTED INSTRUCTIONS

Augmented instructions provide micro programming capability by using the address portion of the instruction to select logical operations. These instructions do not address a memory register.

The following instruction loads itself into the AC.

law	76	8	The address postion of this instruction may be used to specify a constant.

OPERATE GROUP

The operate instructions use bits 5 through 17 to specify variations of the basic instructions.

opr	74xxxx	8	Operate. The operate instruction is also the conditioning (skip) instruction. When a particular condition is present, the following instruction will be skipped. The various micro program events occur at different times to allow several events to be programmed which affect the same element. This is a micro program instruction using bits 4-17 to specify the desired operations. Combinations of the individual operations can be made. The operations are specified by bits as follows

Mnemonic Code	Octal Code	Operation	Sequence of Occurrence
cma	1	Complement AC.	3
cml	2	Complement Link.	3
oas	4	Inclusive OR AC switches with AC.	3
ral	10	Rotate AC and Link left one place.	3
rtl	2010	Rotate AC and Link two places left.	2, 3
rar	20	Rotate AC and Link right one place.	3
rtr	2020	Rotate AC and Link two places right.	2, 3
hlt	40	Halt the machine	4
sma	100	Skip on minus AC. If $AC_0 = 1$, the next instruction in sequence is skipped.	1
spa	1100	Skip on plus AC. If $AC_0 = 0$, the next instruction in sequence is skipped.	1
sza	200	Skip if AC = 0.	1
sna	1200	Skip if AC $\neq$ 0.	1
snl	400	Skip if Link $\neq$ 0.	1
szl	1400	Skip if Link = 0.	1
skp	1000	Skip unconditionally.	1
cll	4000	Clear Link.	2
cla	10000	Clear AC.	2

IN-OUT TRANSFER GROUP

The instructions in this group are similar to the Operate Group instructions except they pertain to the transfer of information between the Central Processor and various input-output devices. Bits 4 through 17 select and control input-output devices.

Mnemonic Code	Octal Code	Time (μsec)	Operation
iot	70xxxx	8	In Out Transfer . This instruction which forms a micro program is used to select an input or output device. The instruction forms a micro program and has the following format:

Function	Command Bits
Specifies the in-out instruction (Operation Code 1110)	0-3
May be used to select sub-device	4-5
Selects the device	6-11
May be used to select sub-device	12-13
Clears the AC at event time 1 if a ONE	14
Transfers an IOT pulse at event time 3 if a ONE	15
Transfers an IOT pulse at event time 2 if a ONE	16
Transfers an IOT pulse at event time 1 if a ONE	17

Bits 13-17 may be used together in any combination to allow various types of in-out command structures, and these may handle 1, 2, or 3 devices per selection (bits 4-12) depending upon the requirements of the devices.