digital INTEROFFICE MEMORANDUM

DATE: February 4, 1969

SUBJECT: PDP-11 Instruction Set and Revised Page 5 NO: RC:69:08

TO: Distribution List A FROM: R. Cady

The attached is the firm, final and irrevocable instruction set for the PDP-11 family. Please destroy all information on this subject dated prior to January 31, 1969.

Attached also is a revised Page 5 for the BUS DESCRIPTION memo dated January 29, 1969. Please destroy previous Page 5.

Use of NPR line when making bus request:

If the use of the bus is requested by a device for data transfer or other operation which does not affect the processor (DIN, DOT, or DTR other than to processor) it may gain access to the bus in the middle of an instruction cycle by asserting ASAP along with its BR line. When the processor a) has control of the bus, b) does not have a slave device selected, and c) sees NPR asserted, it enters into a PTR sequence if the BR priority is greater or equal to the processor priority. Note that by means of the NPR line, a device may obtain the bus before completion of the current instruction. The processor may not be interrupted (DTR sequence) at this time. Thus, the NPT signal must inhibit a device from putting its minor priority on D 7:0 if it is requesting the bus for processor interrupt.

DTR - Direct Transfer

This is used to unconditionally transfer control of the bus to another device. It is the means by which the processor executes ~~XXX and XXX~~ instructions and by which a device may interrupt the processor.

1. Master sets C=6, sets D<7:0> to the Device Selection code (DS) of the device to which it wishes to transfer control, and sets D<15:8> to the Function Command (FC) being issued to the selected device, (the significance of the Device Command is specific to the selected device).
2. Master asserts MSYN and negates BBSY.
3. All devices see C=6 and MSYN, and examine D<7:0>. The selected device examines D<15:8> (FC) and performs accordingly. If the selected device requires the bus, it asserts BBSY.
4. The selected device asserts SSYN.
5. The master sees SSYN and negates MSYN.
6. The selected device sees MSYN, negates SSYN. If the selected device has asserted BBSY, it then becomes master; otherwise, the processor sees MSYN and SSYN and BBSY and becomes master.

NOTE:
The processor responds to DS=0. Processor interprets FC as the address in the first 256 bytes of memory at which an interrupt entry vector is located and performs an interrupt via that address.

The console responds to DS=FF_{16}. The console interprets any command as a Halt, and holds the bus until manual intervention restarts the system. In addition, if the Function Command is in the range of FO_{16} to FF_{16}, the console examines and displays the contents of the processor register with that address in the last 256

PDP-11 INSTRUCTION SET

January 31, 1969

I. Introduction

The PDP-11 is a 16-bit small computer. It operates on bytes (8 bits) or words (16 bits). The memory is byte addressable, to a maximum of 65536 bytes. Instructions are one, two, or three bytes (8, 16, 24 bits). Throughout this memo, several conventions will be adhered to:

The term "byte" refers to an 8-bit quantity.

The term "word" refers to two consecutive bytes. The low order byte is first in memory.

Numbers are hexadecimal except as noted and when specifying bit positions within a byte or word.

Bit numbering begins at Ø at the right or low order end of a byte or word and increases to the left to 7 for a byte or 15 for a word. For example, the high order bit of the accumulator is bit 15.

This memo describes the processor registers, the memory reference instruction addressing and the six groups of machine instructions - memory reference, operate, conditional jump, add to register, push/pop and external.

II. Storage Organization

The PDP-11 system has a single bus for memory and input/output devices. Addresses for memory locations or status registers in devices are referred to in a similar manner. There are 16 address bits, allowing locations 0 through $2^{16}-1$ to be assigned to memory units or input/output devices.

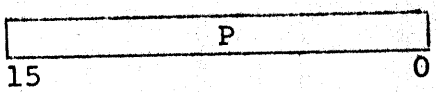
There is no paging as in the PDP-8, but certain locations are reserved for special purposes. Words 0 through 7F are called page0 and may be referred to from any location by a 2 byte instruction. Words 0 through FF may be used for interrupt status (described in the next section.

The last 128_{10} bytes (FF80 through FFFF) are called the external page and are normally assigned to hardware registers in the processor and input/output devices. The last 16 bytes (FFF0-FFFF) are internal processor registers.

III. Processor Registers

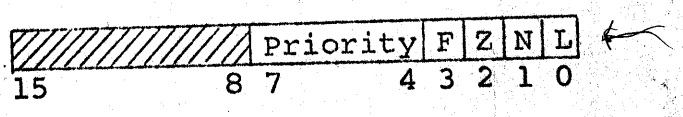
The PDP-11 has 8 addressable 16-bit hardware registers.

A. Program Counter (P)



Prior to each instruction, the program counter contains the address of the instruction. During the execution it contains the address of the next sequential instruction. The program counter itself is addressable at (FFFC) 16.

B. Status Register (S)



Bits 7 through 4 of the status register contain the current processor priority. Bits 3 through 0 contain the condition codes. Bits 15-8 are zero if referenced. The status register is addressable at (FFFE) 16.

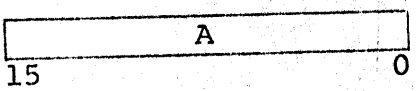
1. Condition Codes

Bits 3 through 0 of the status register are called the condition codes. They are set according to the results of most instructions. L (bit 0) is complemented as a result of carry out from the adder. N (bit 1) is set (reset) if the result of an arithmetic instruction is negative (non-negative). Z (bit 2) is set (reset) if the result of an arithmetic operation is zero (non-zero). Bit 3 is the input/output flag (F). It is set by certain devices on conditions determined by the device.

2. Priority

Bits 7 through 4 of the status register determine the priority of the currently running program. The higher order bits 7, 6 determine the processor major priority, while bits 5, 4 determine its minor priority. The processor will only service requests from other devices in the system if it is operating at a lower priority than the requesting device.

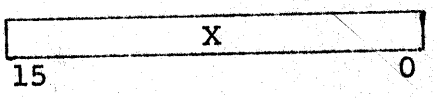
C. Accumulator (A)



This register is used both as an operand and as the result register for most arithmetic instructions. In addition to being implicitly addressed by these instructions, it also may be explicitly addressed

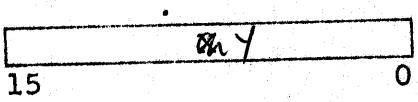
at location (FFFØ) 16.

D. X-Register (X)



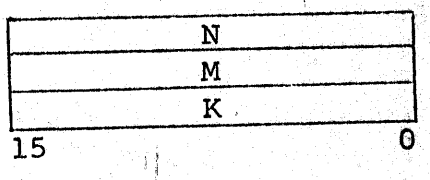
The X-Register is used to provide a hardware push down list facility. It may also be employed as a general index register. It is incremented or decremented by various instructions below. It may be explicitly addressed at (FFF8)16.

E. Y-Register (Y)



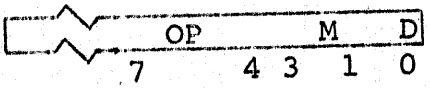
This register is a general index register. It may be explicitly addressed at (FFFA)16.

F. General Registers (K, M, N)



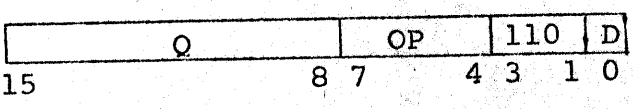
The general registers are three hardware 16-bit registers useable by the program for temporary address or data storage. They are normally addressed in the mode described below in Section IV (E). In addition, K, M, N may be explicitly addressed at (FFF2)16, (FFF4)16, (FFF6)16, respectively.

IV. Memory Reference Instruction Addressing



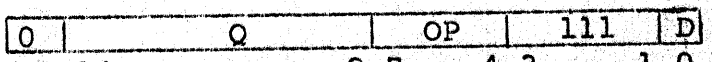
Each memory reference instruction consists of one, two or three bytes. The first byte is called the instruction byte. Successive bytes, if any, contain either address information or data. The operation code (OP) is in the range of Ø through (C)16. The mode bits (M) determine the address computation. The deferred bit (D) specifies whether the address is to be deferred one level indirectly.

A. Relative Addressing



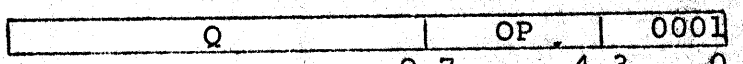
The 2's complement offset quantity Q is added to the program counter to form an effective address which is in the range (-80,+7F)16 relative to the program counter.

B. Page Ø Addressing



Q is used to form an address in the range (0, ~~7F~~ FF)16.

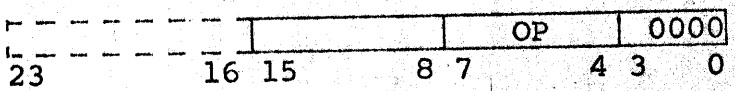
C. Full Addressing



or immediate deferred !

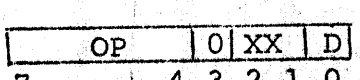
Q is taken as a full 16-bit address. This address mode cannot be deferred

D. Immediate Addressing



The operand is taken directly from the byte or word following the instruction byte. The operand is a byte or a word depending on the operation code (see Section V). This addressing mode may not be deferred

E. General Register Addressing



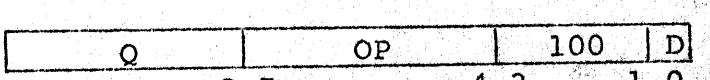
The effective address before deferral is K, M, or N depending on whether XX is 01, 10, or 11 respectively. If D is set, and "auto-increment" is added to the register ~~to determine the effective address~~. This new effective address also replaces the contents of the register. The autoincrement is 0, 1, or 2 depending on the instruction (see Section V).

effective address before inc.

F. External Addressing

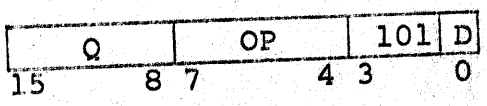
In this mode, the high order 9 bits of the effective address before deferral are all ones. The low order 7 bits are Q. Thus, the address range before deferral is (FF8Ø-FFFF)16. Input/output and external devices are typically assigned in this area allowing input/output or external references in a two byte instruction.

G. X-Indexed Addressing



The effective address before deferral is the sum of the 2's complement number Q and the contents of the Y-register. This provides a range of (-80,+7F)16 relative to the contents of X.

H. Y-Indexed Addressing

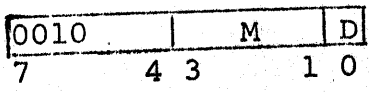


The effective address, before deferral, is the sum of the 2's complement number Q and the contents of the Y-register. This provides a range of (-80,+7F)16 relative to the contents of Y.

V. Memory Reference Instruction Group

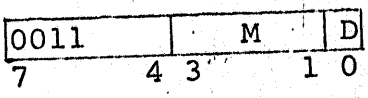
Bits 3-0 are defined by the address mode as discussed in Section IV above.

A. Load Byte (LDB)



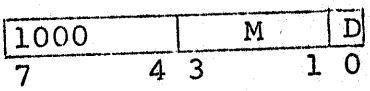
The effective byte is loaded into bits 7 through 0 of the accumulator. The high order bit of the effective byte is extended through bits 15-8 of the accumulator. L is not affected. N is set to bit 15 of the effective byte. Z is set to one (zero) if the effective byte is zero (non-zero). The autoincrement is one.

B. Load Word (LDW)



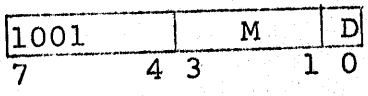
The effective word is loaded into the accumulator. L is not changed. N is set to bit 15 of the effective word. Z is set to one (zero) if the effective word is zero (non-zero). The autoincrement is two.

C. Store Byte (STB)



The contents of accumulator bits 7 through 0 are stored in the effective byte. The condition codes are not affected. The autoincrement is one.

D. Store Word (STW)

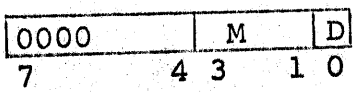


The contents of the accumulator are stored in the effective word. The condition codes are not affected. The autoincrement is two.

no auto inc. on immediates?

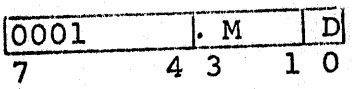
02's masked compares.

E. Add Byte (ADB)



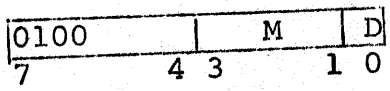
The effective byte is treated as a 2's complement 8-bit quantity and added, with sign extended, to the contents of the accumulator. L is complemented if the operation causes a carry out of bit zero of the adder. N is set to the high-order bit of the result. Z is set to one (zero) if the result is zero (non-zero). The autoincrement is one.

F. Add Word (ADW)



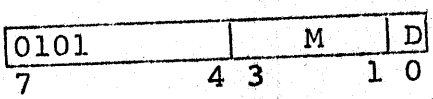
The effective word is added to the contents of the accumulator. L is complemented if the operation causes a carry out of bit zero of the adder. N is set to the high order bit of the result. Z is set to one (zero) if the result is zero (non-zero). The autoincrement is two.

G. Compare Byte (CPB)



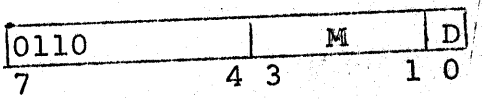
The 2's complement of the accumulator and the effective byte are added together. The effective byte is treated as a 2's complement 8-bit quantity. The result is not stored in any register, but is used to set the condition codes. L is complemented if the operation causes a carry out of bit zero of the adder. N is set to the high order bit of the result. Z is set to one (zero) if the result is zero (non-zero) The autoincrement is one.

H. Compare Word (CPW)



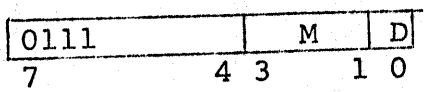
The 2's complement of the accumulator and the effective word are added together. The result is not stored in any register, but is used to set the condition codes. L is complemented if the operation causes a carry out of bit zero of the adder. N is set to the high order bit of the result. Z is set to one (zero) if the result if zero (non-zero). The autoincrement is two.

I. ~~And Byte~~ (ANB) IOR of word



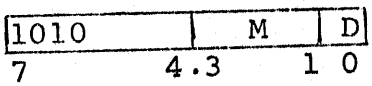
The effective byte is anded to the contents of the accumulator bits 7 through 0. Bits 15 through 8 of the accumulator are not affected. L is not changed. N is set to the sign of the accumulator. Z is set to one (zero) if the accumulator is zero (non-zero) after the operation. The autoincrement is one.

J. And Word (ANW)



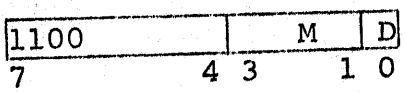
The effective word is anded to the contents of the accumulator., L is not changed. N is set to one (zero) if the result is negative (non-negative). Z is set to one (zero) if the result is zero (non-zero). The autoincrement is two.

K. Increment (INC)



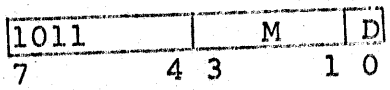
The effective word is incremented by one. L is complemented if the operation causes a carry out of bit zero of the adder. N is set to the high order bit of the result. Z is set to one (zero) if the result is zero (non-zero). The autoincrement is two.

L. Jump (JMP)



The effective address replaces the contents of the program counter. The condition codes are not affected. The autoincrement is zero.

M. Jump to Subroutine (JSR)



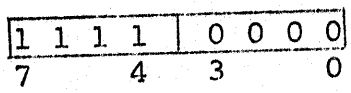
13 mv 1 opr.

P->X

The program counter is stored in the word which has the address specified by the contents of the X-register. The X-register is then incremented by two. The effective address then replaces the contents of the program counter. The condition codes are not affected. The autoincrement is zero.

VI. Operate Instruction Group

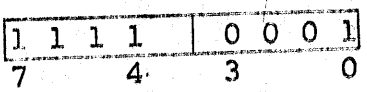
A. No Operation (NOP)



add to S

The program counter is advanced one byte. The condition codes are not changed.

B. Increment Accumulator (IAC)

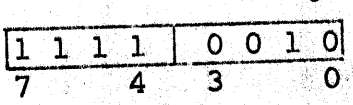


what

Use ADB immediate!

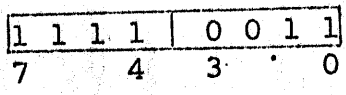
The accumulator is incremented by one. L is complemented if the operation causes a carry out of bit zero of the adder. N is set to the high order bit of the result. Z is set to one (zero) if the result is zero (non-zero).

C. Complement Accumulator (CMA)



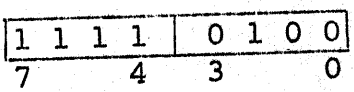
The contents of the accumulator and the L bit are replaced by the ones complement of the original contents. N is set to the high order bit of the result. Z is set to one (zero) if the result is zero (non-zero).

D. Negate (NEG)



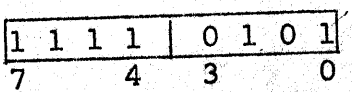
The 2's complement of the accumulator replaces the original accumulator contents. L is complemented if the original contents of the accumulator are non-zero. N is set to the high order bit the result. Z is set to one (zero) if the result is zero (non-zero).

E. Clear Accumulator (CLA)



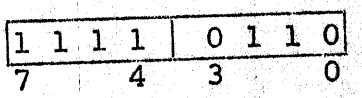
The contents of the accumulator are set to zero. L is not affected. N is set to zero and Z is set to one.

F. Plus One to Accumulator (PAC)



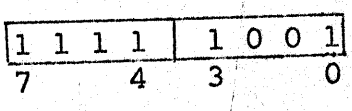
The accumulator is set to plus one. L is not affected. Both N and Z are set to zero.

G. Minus One to Accumulator (MAC)



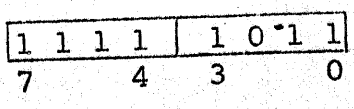
The accumulator is set to minus one. L is complemented. N is set to one and Z is set to zero.

H. Rotate Accumulator Right (RAR)



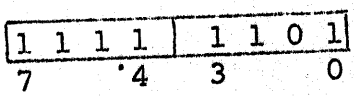
The L bit and the accumulator are treated as a single 17 bit circular register. The combination is rotated right one bit position (L goes into accumulator bit 15 and accumulator bit Ø goes into L. N is set to the high order bit of the result. Z is set to one (zero) if the result is zero (non-zero).

I. Rotate Accumulator Left (RAL)



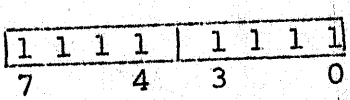
The L bit and the accumulator are treated as a single 17 bit register. The combination is rotated left one bit position (L goes to accumulator bit Ø and accumulator bit 15 goes to L). N is set to the high order bit of the result. Z is set to one (zero) if the result is zero (non-zero).

J. Complement Link (CML)



The L bit is complemented. N, Z, and F are not affected.

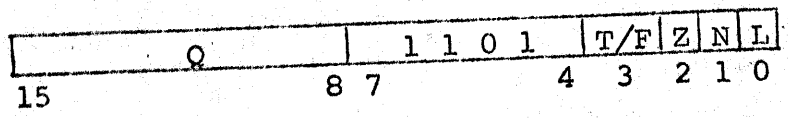
K. Clear Condition Codes (CCC)



The four condition codes (F, Z, N, L) are set to zero.

VII. Conditional Jump Instructions

These instructions test for specific patterns of condition codes. There are 16 such instructions, formed as micro-codings of the three conditions and a true/false bit. The format of these instructions is as below:



Where T/F specifies JCT (Jump on Conditions True) or JCF (Jump on Conditions False) and Z, N and L specify the condition code pattern under test. If Z, N, L are all zeroes, then the I/O Flag bit (F) is tested. If the T/F bit is zero, then the test is performed on the OR of the indicated conditions; e.g., JCT Z L; is "Jump on either Z=1 or L=1". If the T/F bit is one (JCF), then the test is performed on the AND of the falsity of the indicated conditions. Thus, JCF Z L is "Jump on Z=Ø and L=Ø.

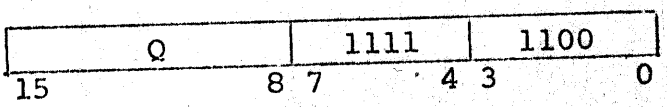
If the specified condition is met, the offset Q is added as a 2's complement number to the program counter. This provides a range of (-80,+7F)16 relative to the program counter. If the condition is not met, the next instruction is executed. The condition codes themselves are not affected.

Mnemonics, for more commonly used conditions tested are:

MNEMONIC	MEANING	CONDITION
JEQ	jump on equal	Z
JNE	jump on not equal	not Z
JLT	jump on less than	N
JGT	jump on greater than	not N and not Z
JLE	jump on less than or equal	N or Z
JGE	jump on greater or equal	not N
JLS	jump on link set	L
JLR	jump on link reset	not L
JFS	jump on F flag set	F
JFR	jump on F flag reset	not F

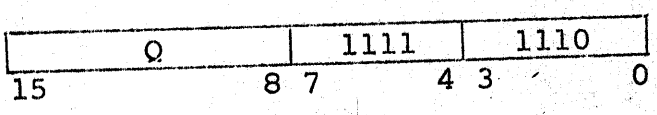
VIII. Add to Register Group

A. Add to M (ATM)



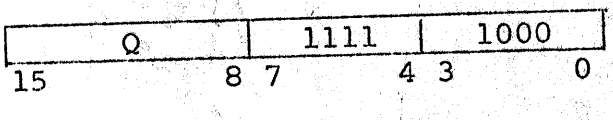
Q is treated as an 8-bit 2's complement number and added to the contents of M. L is complemented if the operation causes a carry out of bit zero of the adder. N is set to the high order bit of the result. Z is set to one (zero) if the result is zero (non-zero).

B. Add to N (ATN)



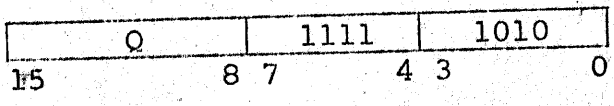
Q is treated as an 8-bit 2's complement number and added to N. L is complemented if the operation causes a carry out of bit zero of the adder. N is set to the high order bit of the result. Z is set to one (zero) if the result is zero (non-zero).

C. Add to X-Register (ATX)



Q is treated as an 8-bit 2's complement number and added to X. L is complemented if the operation causes a carry out of bit zero of the adder. N is set to the high order bit of the result. Z is set to one (zero) if the result is zero (non-zero).

D. Add to Y-Register (ATY)

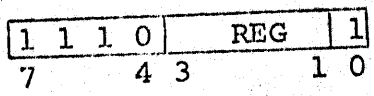


Q is treated as an 8-bit 2's complement number and added to Y. L is complemented if the operation causes a carry out of bit zero of the adder. N is set to the high order bit of the result. Z is set to one (zero) if the result is zero (non-zero).

IX. Push/Pop Group

This instruction group provides for moving data words between the general registers and the push down list.

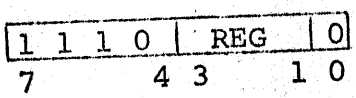
A. Push Instructions



The contents of the addressable hardware register specified by REG are stored at the address in the X-register. The contents of X are then incremented by two. The contents of the REG is not altered. The instructions specifying the registers are:

REG	MNEMONIC	OPERATION
000	PUA	Push Contents of A
001	PUK	Push Contents of K
010	PUM	Push Contents of M
011	PUN	Push Contents of N
100	PUX	Push Contents of X
101	PUY	Push Contents of Y
110	PUP	Push Contents of P
111	PUS	Push Contents of S

B. Pop Instructions



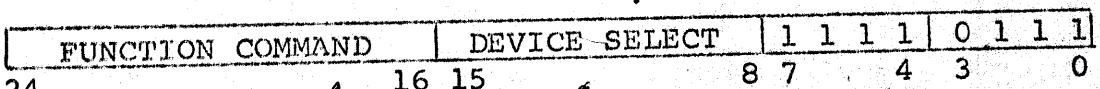
The contents of X are decremented by two. Then the word at the address in the X-register is stored in the addressable hardware register specified by REG. The instructions specifying the registers are:

REG	MNEMONIC	OPERATION
000	POA	Pop Into A
001	POK	Pop Into K
010	POM	Pop Into M
011	PON	Pop Into N
100	POX	Pop Into X
101	POY	Pop Into Y
110	POP	Pop Into P
111	POS	Pop Into S

POP is the return instruction corresponding to JSR (jump to subroutine). The instruction sequence POS, POP causes a return from an interrupt routine.

X. External Transfer (XTR)

This instruction allows the program to control certain devices in the system. The processor is device zero - a reference to the processor in an external instruction causes a programmed interrupt. The console is device (FF)16 - an external instruction to the console causes a halt. Other device numbers may be assigned to extended arithmetic elements, input/output devices or user equipment. The function of the external instruction in these cases depends on the device.



A. Interrupt of Processor

This instruction causes the processor to "interrupt itself" - that is, respond as if the interrupt had occurred from an external device. It occurs if the contents of DEVICE SELECT are zero. The program counter is stored in the location specified by the contents of the X-register. The status byte is stored in the location which is two greater than the contents of the X-register. The FUNCTION COMMAND code (bits 24-16) of the instruction is stored in the location which is three greater than the contents of the X-register. The X-register, itself, is incremented by four.

The processor resumes operation as determined by a three byte "status block" which begins at the address with bits 15-8 zero and bits 7-0 FUNCTION COMMAND. The first two bytes of the status block replace the program counter and the third byte becomes the new status byte.

B. Halt

If DEVICE is (FF)16, the system stops. Condition codes are not affected.

C. Input/Output Control

If DEVICE refers to an input/output device, the operation of the instruction is determined by the device.

D. Extended Arithmetic References

If the referenced device is an extended arithmetic unit, that unit performs the appropriate operation and returns control. The instruction format is determined by the referenced unit.

E. Trap

If DEVICE specifies a non-existant unit, and if the "time-out" option is installed, the effect is the same as an external control instruction with DEVICE and FUNCTION zero. This is used to simulate unimplemented extended operation codes by software or firmware programs. The contents of the word at location zero must be the starting address of a software trap handler.

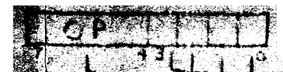
PDP11 INSTRUCTIONS

MEMORY REFERENCE

CODE	OP	INSTRUCTION
0	0000	ADB
1	0001	ADW
2	0010	SDB
3	0011	SBW
4	0100	CPB
5	0101	CPW
6	0110	ANB
7	0111	ANW
8	1000	STB
9	1001	STW
A	1010	INC
B	1011	JSR
C	1100	JMP

M	ADDRESS MODE	D=0	D=1
000	IMMEDIATE/FULL	OPERAND = Q_{15-B} OR Q_{23-B}	EFA = Q_{23-B}
001	K REGISTER	EFA = K	EFA = (K) + $\alpha \rightarrow$ K
010	M REGISTER	EFA = M	EFA = (M) + $\alpha \rightarrow$ M
011	N REGISTER	EFA = N	EFA = (N) + $\alpha \rightarrow$ N
100	X INDEX	EFA = (X) + Q_{15-B}	EFA = [(X) + Q_{15-B}]
101	Y INDEX	EFA = (Y) + Q_{15-B}	EFA = [(Y) + Q_{15-B}]
110	RELATIVE TO NEXT P	EFA = (P) + Q_{15-B}	EFA = [(P) + Q_{15-B}]
111	RELATIVE TO ZERO	EFA = ϕ + Q_{15-B}	EFA = [ϕ + Q_{15-B}]
	PAGE ZERO		Q_{15} = 0
	EXTERNAL		Q_{15} = 1

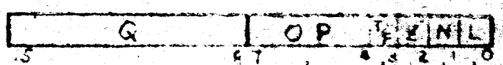
OPERATE



CODE	OP	INSTRUCTION	INSTR CODE
F	1111	NOP	0000 0
F	1111	IAC	0001 1
F	1111	CMA	0010 2
F	1111	NEG	0011 3
F	1111	CLA	0100 4
F	1111	PAC	0101 5
F	1111	MAC	0110 6
F	1111	RAR	1001 9
F	1111	RAL	1011 B
F	1111	CMR	1101 D
F	1111	SCC	1111 F

INTERNAL REGISTERS ARE THE OPERANDS OF ABOVE INSTRUCTIONS.

CONDITIONAL JUMP



CODE	OP	INSTRUCTION
D	1101	JUMP RELATIVE TO P, (P) + $Q_{15-8} \rightarrow$ P, CONDITIONS: MICRO-CODED

T/F TEST CONDITIONS: JCT, T/F = 0 → JUMP ON THE "OR" OF TRUTH OF CONDITIONS. JCF, T/F = 1 → JUMP ON THE "AND" OF FALSITY OF CONDITIONS.

CONDITIONS: Z=1, ZERO A; N=1, NEGATIVE A; L=1, LINK SET; Z+N+L=0, TEST TO FALSE

PUSH/POP

CODE	OP	REG	D=0	D=1
E	1110	000	PCA	PUA
E	1110	001	PCK	PUK
E	1110	010	PCM	PUM
E	1110	011	PCN	PUN
E	1110	100	PCX	PUX
E	1110	101	PCY	PUY
E	1110	110	PCP	PUP
E	1110	111	PCS	PUS

PUSH, D=1: (REG) → (X), (X)+2 → X

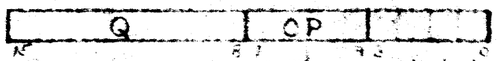
POP, D=0: (X)-2 → X, (X) → REG

INTERNAL REGISTERS ARE SPECIFIED BY REG

POP IS THE RETURN INSTR. FOR JSR

PCS, PUS IS THE RETURN INSTR. FOR INTERRUPT

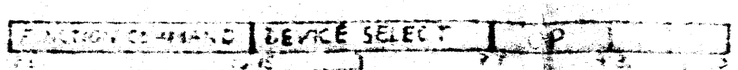
ADD TO REGISTER



CODE	OP	INSTRUCTION	INSTR CODE
F	1111	ATX	1000 8
F	1111	ATY	1010 A
F	1111	ATM	1100 C
F	1111	ATN	1110 E

THE ADDRESSED INTERNAL REGISTERS ARE INCREMENTED BY Q_{15-8}.

EXTERNAL TRANSFER



DEVICE SELECT:

00-$_{16}$ - INTERRUPT (P) → (X), (S) → (X)+2, INPUT (DEV) → (X)+3, (X)-4 → X, (PC NEXT CONT) → P, (PS NEXT CONT) → S

XX-$_{16}$ - OPERATION A FUNCTION OF DEV SELECTED

CODE	OP	INSTRUCTION	INSTR CODE
F	1111	ATF	0111 7

JFOL 2/3/69