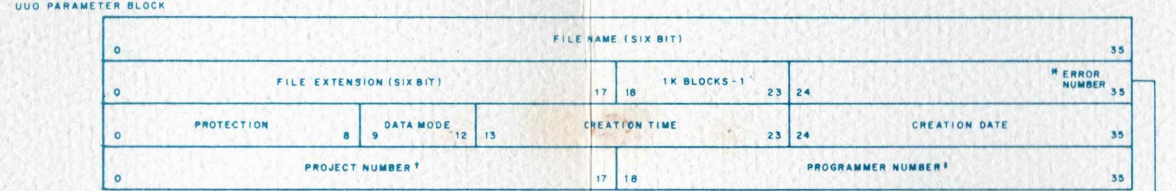
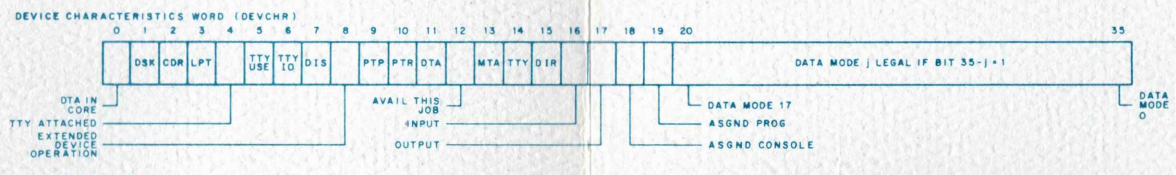
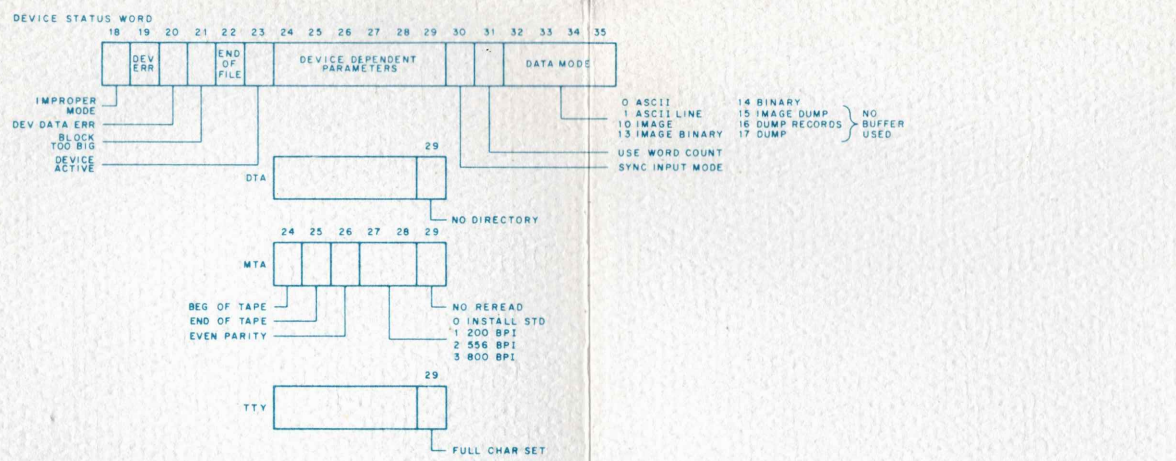
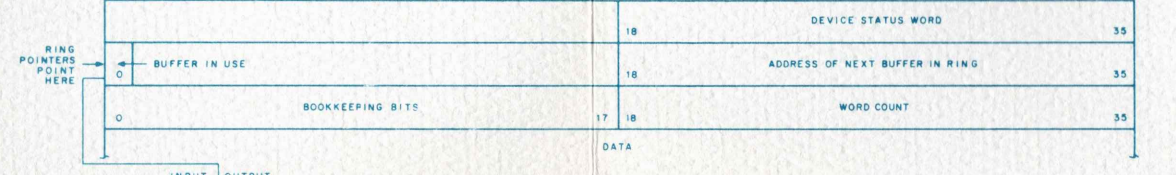
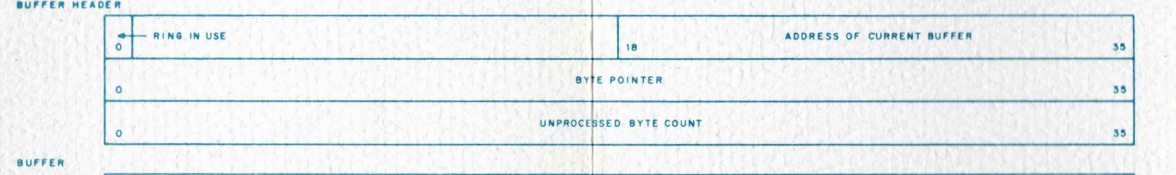
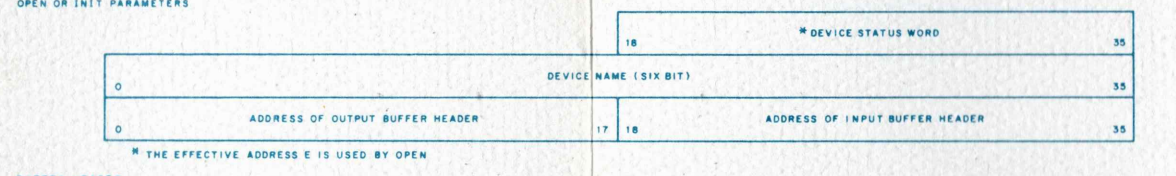


	--0	--1	--2	--3	--4	--5	--6	--7
00	(ILLEGAL)							
01	USER DEFINED UO'S (UNIMPLEMENTED USER OPERATIONS)							
02								
03								
04	CALL	INIT	LEFT FOR	SPECIAL	MONITORS	CALLI		
05	OPEN	RESERVED	STATO	STATZ	INBUF	OUTBUF	INPUT	OUTPUT
06	SETSTS	RELEAS	GETSTS	STATZ	INBUF	OUTBUF	LOOKUP	ENTER
07	CLOSE	MTAPE	UGETF	USETI	USETO	LOOKUP		
10								
11								
12								
13	UFA	DFN	FSC	IMP	ILDB	LDB	IDPB	DPB
14	FAD	-L	-M	-B	FADR	-I	-M	-B
15	FSB	-L	-M	-B	FADR	-I	-M	-B
16	FMP	-L	-M	-B	FMPR	-I	-M	-B
17	FDV	-L	-M	-B	FDVR	-I	-M	-B
20	MOVE	-I	-M	-S	MOV5	-I	-M	-S
21	MOVN	-I	-M	-S	MOV5	-I	-M	-S
22	IMUL	-I	-M	-B	MUL	-I	-M	-B
23	IDIV	-I	-M	-B	DIV	-I	-M	-B
24	ASH	ROT	LSH	JFFO	ASHC	ROTC	LSHC	
25	EXCH	BLT	AOBJP	AOBJN	JRST	JFCL	XCT	
26	PUSHJ	PUSH	POP	POPJ	JSP	JSA	JRA	
27	ADD	-I	-M	-B	SUB	-I	-M	-B
30	CAI	-L	-E	-LE	-A	-GE	-N	-G
31	CAM	-L	-E	-LE	-A	-GE	-N	-G
32	JUMP	-L	-E	-LE	-A	-GE	-N	-G
33	SKIP	-L	-E	-LE	-A	-GE	-N	-G
34	AOJ	-L	-E	-LE	-A	-GE	-N	-G
35	AOS	-L	-E	-LE	-A	-GE	-N	-G
36	SOJ	-L	-E	-LE	-A	-GE	-N	-G
37	SOS	-L	-E	-LE	-A	-GE	-N	-G
40	SETZ	-I	-M	-B	AND	-I	-M	-B
41	ANDCA	-I	-M	-B	SETM	-I	-M	-B
42	ANDCM	-I	-M	-B	SETA	-I	-M	-B
43	XOR	-I	-M	-B	IOR	-I	-M	-B
44	ANDCB	-I	-M	-B	EQV	-I	-M	-B
45	SETCA	-I	-M	-B	ORCA	-I	-M	-B
46	SETCM	-I	-M	-B	ORCM	-I	-M	-B
47	ORCB	-I	-M	-B	SETO	-I	-M	-B
50	HLL	-I	-M	-S	HRL	-I	-M	-S
51	HLLZ	-I	-M	-S	HRLZ	-I	-M	-S
52	HLL0	-I	-M	-S	HRL0	-I	-M	-S
53	HLL1	-I	-M	-S	HRL1	-I	-M	-S
54	HRR	-I	-M	-S	HLR	-I	-M	-S
55	HRRZ	-I	-M	-S	HLRZ	-I	-M	-S
56	HRR0	-I	-M	-S	HURO	-I	-M	-S
57	HRR1	-I	-M	-S	HURE	-I	-M	-S
60	TRN	TLN	TRNE	TLNE	TRNA	TLNA	TRNN	TLNN
61	TDN	TSN	TDNE	TSNE	TDNA	TSNA	TDNN	TSNN
62	TRZ	TLZ	TRZE	TLZE	TRZA	TLZA	TRZN	TLZN
63	TDZ	TSZ	TDZE	TSZE	TDZA	TSZA	TDZN	TSZN
64	TRC	TLC	TRCE	TLCE	TRCA	TLCA	TRCN	TLCN
65	TDC	TSC	TDCE	TSCE	TDCA	TSCA	TDCN	TSCN
66	TRO	TLO	TROE	TLOE	TROA	TLOA	TRON	TLON
67	TDO	TSO	TDOE	TSOE	TDOA	TSOA	TDON	TSON



* OVERLAID BY LOOKUP WITH ACCESS DATE IF DSK OR BLOCK NUMBER IF DTA
 † OVERLAID BY LOOKUP WITH WORD-COUNT IF DTA DSK
 ‡ OVERLAID BY LOOKUP WITH ADDRESS-1 IF DTA OR 0 IF DSK
 0 NO FILE, 1 NO UFD, 2 FILE PROTECTED, 3 FILE IN USE
 4 FILE NAME CONFLICT, 5 NO FILE SELECTED, 7 NO DEVICE



IN USE BY MONITOR	0	1
USER	1	0

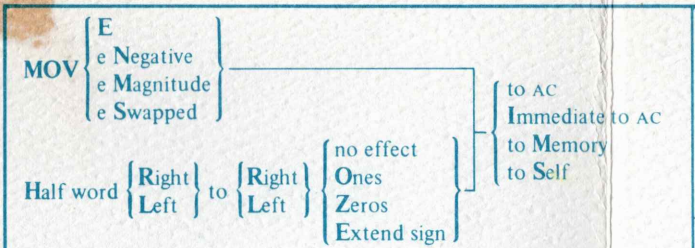
digital

PDP-10

REFERENCE CARD

MOVSI T, MTTYLN	;LENGTH OF
SKIPL DEVDAT, TTYTAB(T)	
AOBJN T, .-1	
JUMPG T, COM2	;NONE FOUN
HRRZS TTYTAB(T)	;NO, CLEAR
MOVSI DAT, 440700	;FORM OUTP
ADDI DAT, TTYBUF(DEV DAT)	
MOVE TAC, DAT	;SAME AS I
PUSHJ PDP, CTEXT	;RETURN CO
JUMPE TAC1, COM2	;IGNORE BL
MOVSI T, -DISPL	;SEARCH FO
CAME TAC1, COMTAB(T)	
AOBJN T, .-1	
LDB ITEM, JPOINT	;GET JOB N
JUMPN ITEM, COM1	;HAS A JOB
MOVSI ITEM, MJOBN	;NO, SEARC
AOBJN ITEM, .+1	;SKIP NULL

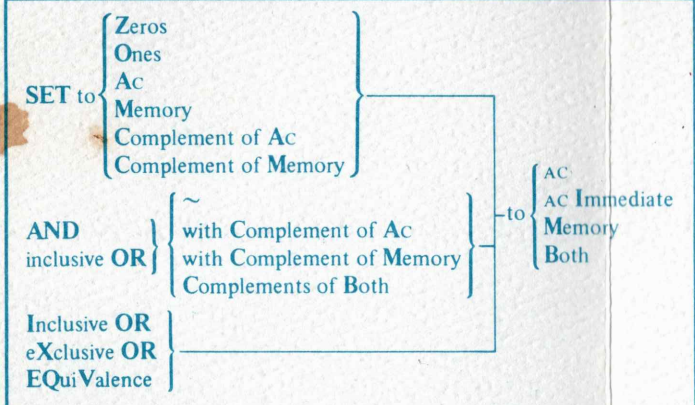
0123456789101112
AC



Block Transfer
EXCHange AC and memory

use present pointer } and {
Increment pointer } {
Increment Byte Pointer } {
Load Byte into AC
DePosit Byte in memory

PUSH down } {
POP up } {
and Jump



SKIP if memory }
JUMP if AC }
Add One to } {
Subtract One from } {
memory and Skip }
AC and Jump } if {
never
Less
Equal
Less or Equal
Always
Greater
Greater or Equal
Not equal

Compare AC {
Immediate
with Memory } and skip if AC {
Positive
Negative

Add One to Both halves of AC and Jump if {
Positive
Negative

Test AC {
with Direct mask
with Swapped mask
Right with E
Left with E } {
No modification
set masked bits to Zeros
set masked bits to Ones
Complement masked bits } and skip {
never
if all masked bits Equal 0
if Not all masked bits equal 0
Always

ADD
SUBtract
MULTiply
Integer MULTiply
DIVide
Integer DIVide

and Round

{
~
Immediate
to Memory
to Both

Floating Add
Floating SuBtract
Floating MultiPly
Floating DiVide

{
~
Long
to Memory
to Both

Floating SCale
Double Floating Negate
Unnormalized Floating Add

Arithmetic Shift } {
Logical Shift } {
ROTate } {
Combined

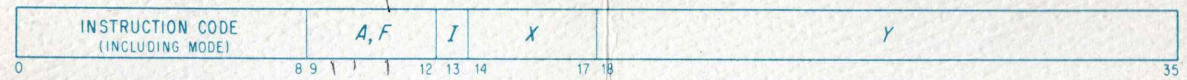
Jump {
to SubRoutine
and Save Pc
and Save Ac
and Restore Ac
if Find First One
on Flag and Clear it
on **OV**erflow (JFCL 10,)
on **CaRrY 0** (JFCL 4,)
on **CaRrY 1** (JFCL 2,)
on **CaRrY** (JFCL 6,)
on **Floating OV**erflow (JFCL 1,)
and **ReST**ore
and **ReST**ore **Fl**ags (JRST 2,)
and **EN**able pi channel (JRST 12,)

HALT (JRST 4,)
eXeCuTe

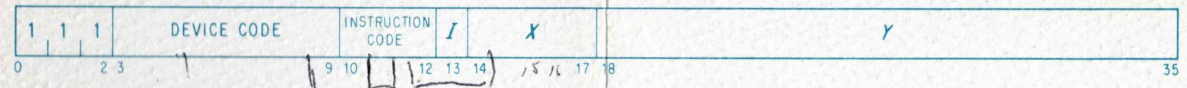
DATA }
BLocK } {
In
Out

CONDitions {
in and Skip if {
all masked bits Zero
some masked bit One

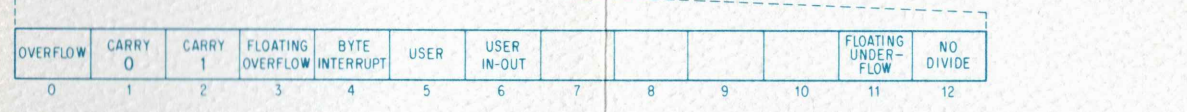
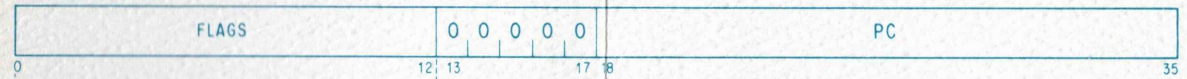
WORD FORMATS
BASIC INSTRUCTIONS



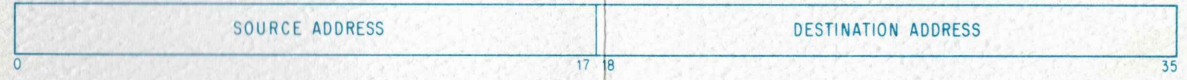
IN-OUT INSTRUCTIONS



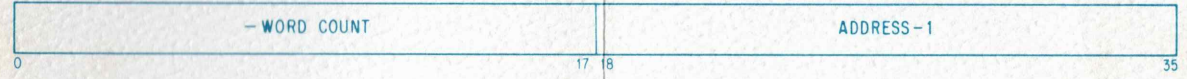
PC WORD



BLT POINTER {XWD}



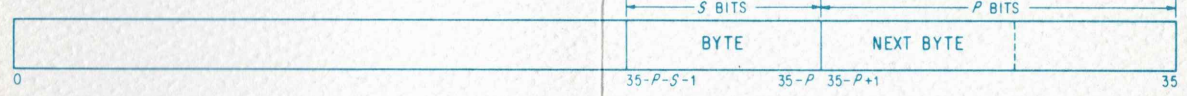
BLKI/BLKO POINTER, PUSHDOWN POINTER, DATA CHANNEL CONTROL WORD {IOWD}



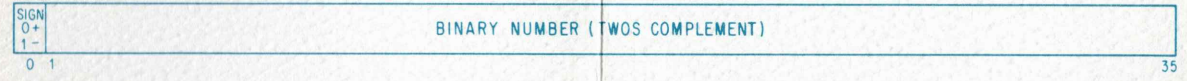
BYTE POINTER



BYTE STORAGE



FIXED POINT OPERANDS



FLOATING POINT OPERANDS



LOW ORDER WORD IN DOUBLE LENGTH FLOATING POINT OPERANDS

