

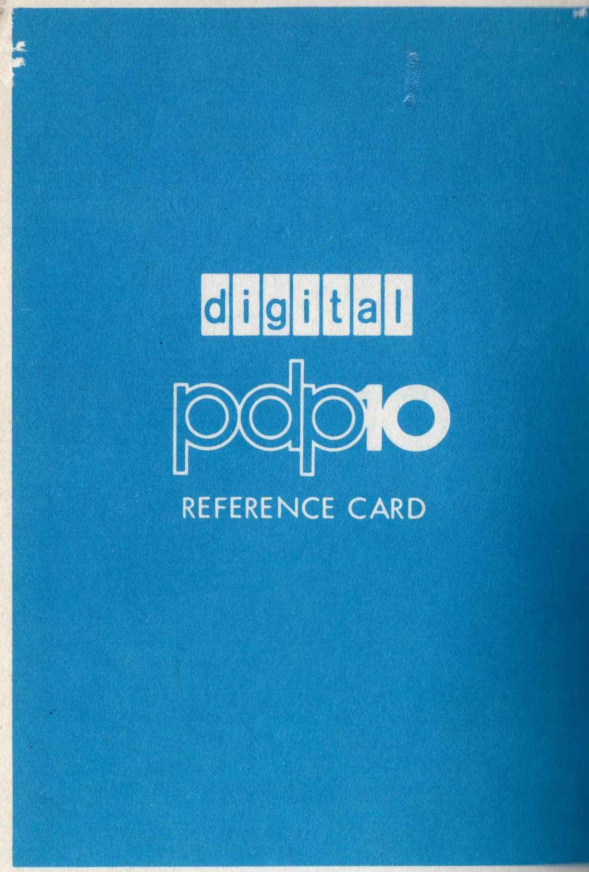
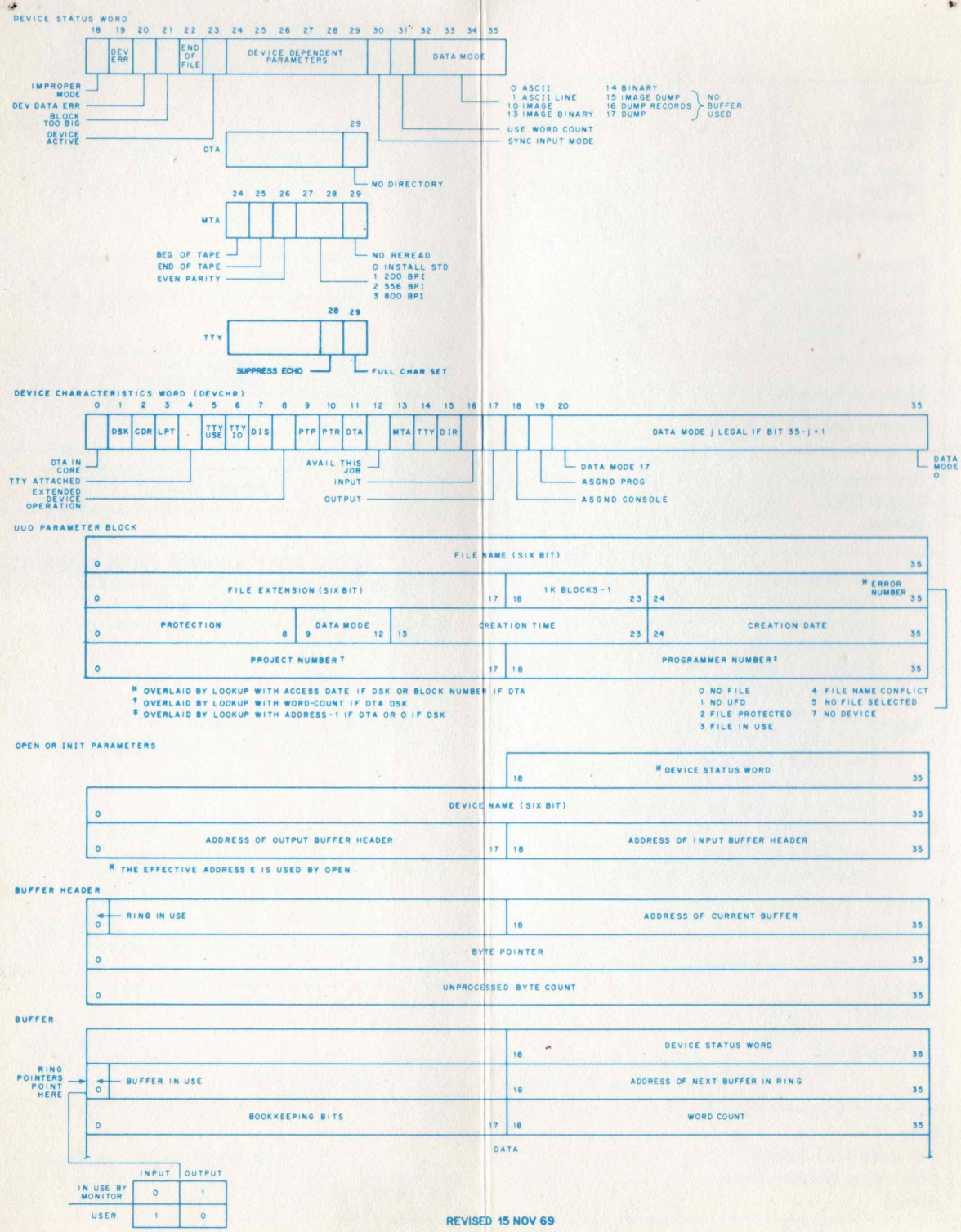
INSTRUCTION CODES

MONITOR SYSTEM DATA FORMATS

top

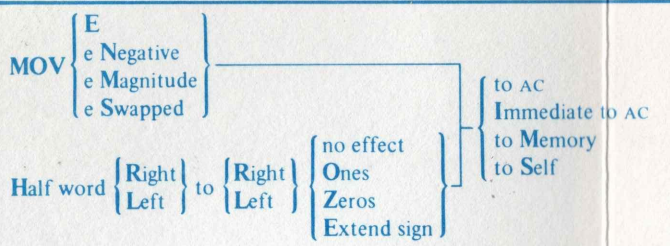
	--0	--1	--2	--3	--4	--5	--6	--7
00	(ILLEGAL)							
01								
02	USER DEFINED UO'S (UNIMPLEMENTED USER OPERATIONS)							
03								
04	CALL OPEN	INIT TDCALL	LEFT FOR SPECIAL	SPECIAL	MONITORS	CALLI		
05	OPEN	TDCALL	RESERVED FOR DEC	INBUFB	RENAME IN	OUT		
06	SETSTS	STATO	GETSTS	STATZ	INBUFB	OUT		
07	CLOSE	RELEASES	MTAPE	UGETF	USETI	USETO	INPUT	OUTPUT
10	USE							
11								
12								
13	UFA	DFN	FSC	IBP	ILDB	LDB	IDPB	DPB
14	FAD	-L	-M	-B	FADR	-I	-M	-B
15	FSB	-L	-M	-B	FSBR	-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	JSR	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	HLLZ	-I	-M	-S	HRLZ	-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	HLR0	-I	-M	-S
57	HRRZ	-I	-M	-S	HLRZ	-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	TRC	TRCE	TRCE	TRCA	TRCA	TRCN	TRCN
65	TDC	TSC	TDCE	TSC	TDCA	TSCA	TDCN	TSCN
66	TRO	TLO	TROE	TLOE	TROA	TLOA	TRON	TLON
67	TDO	TDO	TDOE	TDOE	TDOA	TDOA	TDON	TSON
7--	INPUT - OUTPUT INSTRUCTIONS							

7--00-BLKI the device number is inserted in bits 3 to 9 of each I/O instruction.
 7--04-DATAI
 7--10-BLKO
 7--14-DATAO
 7--20-CONO
 7--24-CONI
 7--30-CONSE
 7--34-CONSO



MOVSI T, MTTYLN ;LENGTH OF
 SKIPL DEVDAT, TTYTAB(T)
 AOBJN T, .-1
 JUMPG T, COM2 ;NONE FOUR
 HRRZS TTYTAB(T) ;NO, CLEAR
 MOVSI DAT, 440700 ;FORM OUT
 ADDI DAT, TTYBUF(DEV DAT)
 MOVE TAC, DAT ;SAME AS
 PUSHJ PDP, CTEXT ;RETURN C
 JUMPE TAC1, COM2 ;IGNORE BU
 MOVSI T, -DISPL ;SEARCH FO
 CAME TAC1, COMTAB(T)
 AOBJN T, .-1
 LDB ITEM, JPOINT ;GET JOB
 JUMPN ITEM, COM1 ;HAS A JO
 MOVSI ITEM, MJOBN ;NO, SEAR
 AOBJN ITEM, .+1 ;SKIP NUL

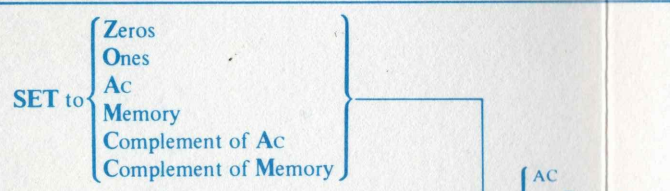
INSTRUCTION LIST



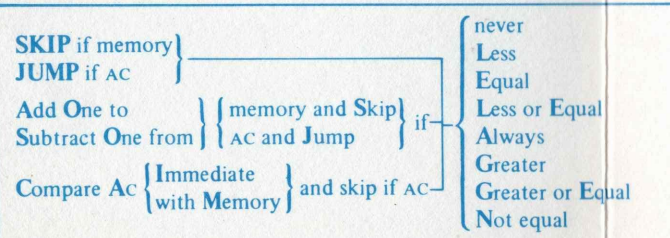
Block Transfer
EXCHange AC and memory

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

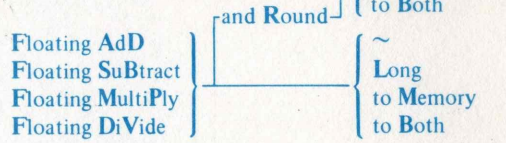
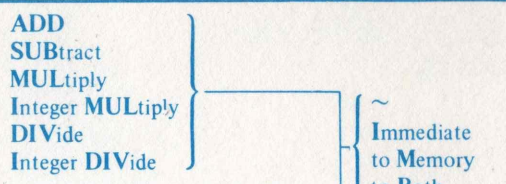
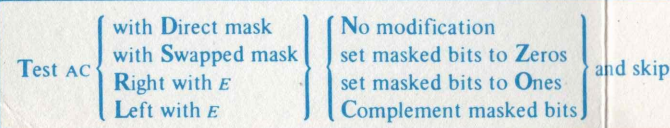
PUSH down } {
POP up } {
 ~
 and Jump



Inclusive OR
exclusive OR
EQivalence



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

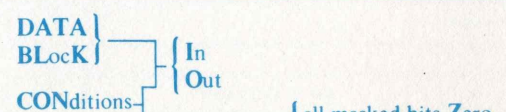


Floating **SC**ale
 Double Floating **Ne**gate
 Unnormalized Floating **ADD**



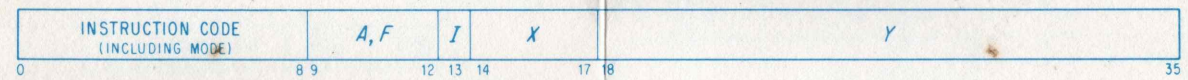
Jump {
 to SubRoutine
 and Save PC
 and Save AC
 and Restore AC
 if Find First One
 on **FL**ag and **CL**ear it
 on **CARRY 0** (JFCL 4,)
 on **CARRY 1** (JFCL 2,)
 on **CARRY** (JFCL 6,)
 on Floating **OV**erflow (JFCL 1,)
 and **ReST**ore
 and **ReST**ore **FL**ags (JRST 2,)
 and **EN**able PI channel (JRST 12,)

HALT (JRST 4,)
exEcuTe

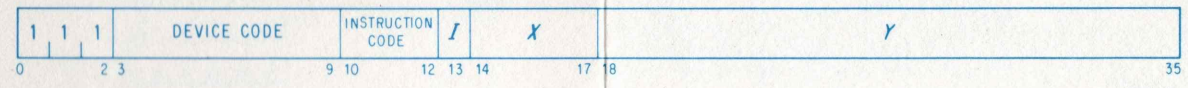


CONDitions {
 in and Skip if {
 all masked bits Zero
 some masked bit One

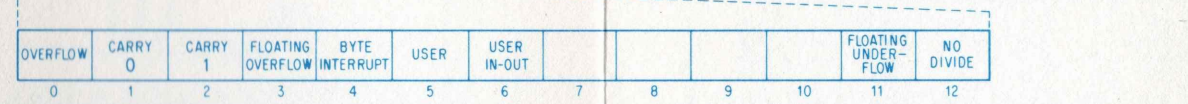
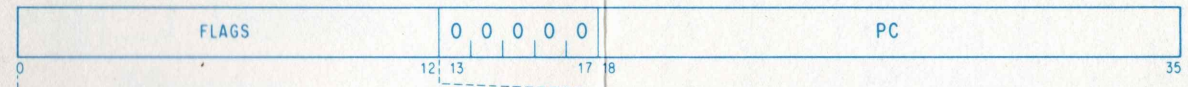
WORD FORMATS
 BASIC INSTRUCTIONS



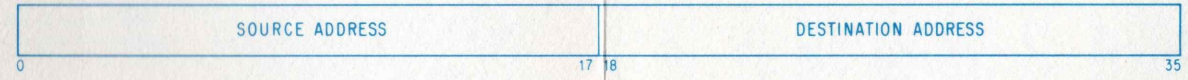
IN-OUT INSTRUCTIONS



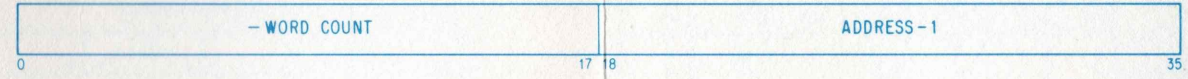
PC WORD



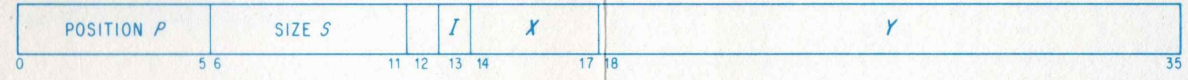
BLT POINTER {XWD}



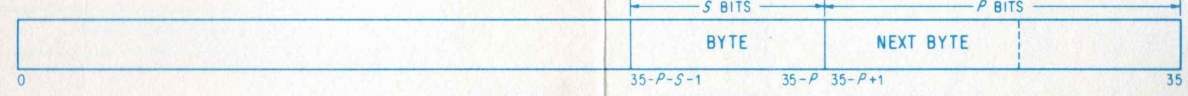
BLKI / BLKO POINTER, PUSHDOWN POINTER, DATA CHANNEL CONTROL WORD {IOWD}



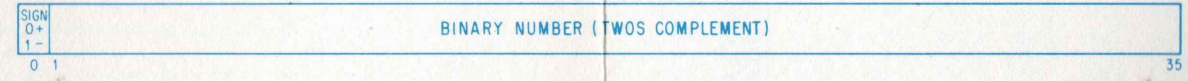
BYTE POINTER



BYTE STORAGE



FIXED POINT OPERANDS



FLOATING POINT OPERANDS



LOW ORDER WORD IN DOUBLE LENGTH FLOATING POINT OPERANDS

