

PDP-6

0xx UUO Traps/System calls 7xx IO Instructions

Byte Manipulation

133 IBP Increment byte pointer
134 ILDB IBP, LDB
136 IDPB IBP, DPB
135 LDB Load byte into AC
137 DPB Deposit byte from AC

Floating Point Arithmetic (Mode 2)

140 FAD $AC+(E) \rightarrow DST$
150 FSB $AC-(E) \rightarrow DST$
160 FMP $AC \times (E) \rightarrow DST$
170 FDV $AC \div (E) \rightarrow DST$
4 R round
132 FSC $AC \times 2^E \rightarrow AC$

Full Word Transfer (Mode 3)

200 MOVE $SRC \rightarrow DST$
204 MOVS $SRC_S \rightarrow DST$
210 MOVN $\neg SRC \rightarrow DST$
214 MOVN $|SRC| \rightarrow DST$

Fixed Point Arithmetic (Mode 1)

220 IMUL $AC \times M \rightarrow DST$
224 MUL $AC \times M \rightarrow DST_2$
230 IDIV $AC \div M \rightarrow DST_2$ AC_{+1} : remainder
234 DIV $AC_2 \div M \rightarrow DST_2$

270 ADD $AC+M \rightarrow DST$
274 SUB $AC-M \rightarrow DST$

Shift & Rotate

240 ASH $AC \times 2^E \rightarrow AC$
241 ROT Rotate AC E times
242 LSH Shift AC E times
244 ASHC $AC_2 \times 2^E \rightarrow AC_2$
245 ROTC Rotate AC_2 E times
246 LSHC Shift AC_2 E times

Miscellaneous

250 EXCH $AC \leftrightarrow (E)$
251 BLT Block transfer
252 AOBJP $AC+[1,1] \rightarrow AC$, Jump if ≥ 0
253 AOBJN $AC+[1,1] \rightarrow AC$, Jump if < 0
254 JRST Jump E [reset proc. flags]
255 JFCL Jump E if flags, clear flags
256 XCT Execute (E)

Pushdown List, Jumps

260 PUSHJ Push PC, Jump E
261 PUSH Push (E)
262 POP Pop into (E)
263 POPJ Pop address, Jump there
264 JSR $PC \rightarrow (E)$, Jump E+1
265 JSP $PC \rightarrow AC$, Jump E
266 JSA $AC \rightarrow (E)$, $[E, PC] \rightarrow AC$, Jump E+1
267 JRA Jump E, $(AC_L) \rightarrow AC$

Mode 1	M	DST	DST ₂	Mode 2	DST	Mode 3	SRC	DST
0	–	(E)	AC	0	–	0	–	(E)
1	I	[0,E]	AC	1	L	1	I	[0,E]
2	M	(E)	(E)	2	M	2	M	AC
3	B	(E)	(E),AC	3	B	3	S	(E),AC _{≠0}

Arithmetic Testing and Modification

300 CAI	if cond(AC,E): skip	0	–	cond(A,B)
310 CAM	if cond(AC,(E)): skip	1	L	A < B
320 JUMP	if cond(AC,0): jump E	2	E	A = B
330 SKIP	(E) $\rightarrow AC_{\neq 0}$	3	LE	A ≤ B
340 AOJ	AC+1 $\rightarrow AC$	4	A	always
350 AOS	(E)+1 $\rightarrow (E), AC_{\neq 0}$	5	GE	A ≥ B
360 SOJ	AC–1 $\rightarrow AC$	6	N	A ≠ B
370 SOS	(E)–1 $\rightarrow (E), AC_{\neq 0}$	7	G	A > B

Bitwise Boolean (Mode 1)

400 boolean	bool(AC,M) $\rightarrow DST$	000 SETZ	040 ANDCB
		004 AND	044 EQV
	m a bool(a,m)	010 ANDCA	050 SETCA
	0 0 Inst ₃	014 SETM	054 ORCA
	0 1 Inst ₄	020 ANDCM	060 SETCM
	1 0 Inst ₅	024 SETA	064 ORCM
	1 1 Inst ₆	030 XOR	070 ORCB
		034 IOR	074 SETO

Half Word Transfer (Mode 3)

NB: Mode S writes full word to $AC_{\neq 0}$

500 HLL	$SRC_L \rightarrow DST_L$	00	–
504 HRL	$SRC_R \rightarrow DST_L$	10	Z $0 \rightarrow (DST)_O$
540 HRR	$SRC_R \rightarrow DST_R$	20	O $777777 \rightarrow (DST)_O$
544 HLR	$SRC_L \rightarrow DST_R$	30	E $Sign \times 777777 \rightarrow (DST)_O$

Logical Testing and Modification

600 T	00 R	M: 0,E	if cond(AC∧M,0): skip	00	N
	01 L	M: [E,0]	(only –,E,A,N)	20	Z $AC \wedge \sim M \rightarrow AC$
	10 D	M: (E)		40	C $AC \vee M \rightarrow AC$
	11 S	M: (E) _S		60	O $AC \vee M \rightarrow AC$

Instruction Format

Inst	ac	i	x	y
0	9	13	14	18
				35

Notation

[x,y]: word with x in left half, y in right half
 x_L : left half of x x_R : right half of x
(x): word at address x x_S : x swapped, $[x_R, x_L]$
AC: (ac) AC_{+1} : (ac+1)
 AC_2 : high word AC, low word AC_{+1}

Address calculation

$y \rightarrow E$
if $x \neq 0$: $E + (x) \rightarrow E$
if $i = 1$: $(E) \rightarrow i, x, y$
repeat