

PROCESS TABLE CONFIGURATION

USER PROCESS TABLE		EXECUTIVE PROCESS TABLE	
0	USER PAGE 0	0	AVAILABLE TO SOFTWARE
17	USER PAGE 36	37	AVAILABLE TO SOFTWARE
20	USER PAGE 40	40	EXECUTIVE MUO STORED HERE
		41	MUO HANDLER INSTRUCTION
		42	
	AVAILABLE TO SOFTWARE IF SMALL USER		STANDARD PRIORITY INTERRUPT INSTRUCTIONS
		57	
		60	
			AVAILABLE TO SOFTWARE
177	USER PAGE 376	177	
200	USER PAGE 400	200	EXECUTIVE PAGE 400
			EXECUTIVE PAGE 401
217	USER PAGE 436		
220	USER PAGE 440		
	AVAILABLE TO SOFTWARE IF SMALL USER		
377	USER PAGE 776	377	EXECUTIVE PAGE 776
400	EXECUTIVE PAGE 340	400	EXECUTIVE PAGE 777
			AVAILABLE TO SOFTWARE
417	EXECUTIVE PAGE 376	417	
420	EXECUTIVE PAGE 377	420	EXECUTIVE PAGE FAILURE TRAP INSTRUCTION
421	EXECUTIVE PAGE FAILURE TRAP INSTRUCTION	421	EXECUTIVE ARITHMETIC OVERFLOW TRAP INSTRUCTION
422	USER ARITHMETIC OVERFLOW TRAP INSTRUCTION	422	EXECUTIVE PUSHDOWN OVERFLOW TRAP INSTRUCTION
423	USER PUSHDOWN OVERFLOW TRAP INSTRUCTION	423	EXECUTIVE TRAP 3 TRAP INSTRUCTION
424	USER TRAP 3 TRAP INSTRUCTION	424	
424	MUO STORED HERE		
425	PC WORD OF MUO STORED HERE		
426	EXECUTIVE PAGE FAILURE WORD		
427	USER PAGE FAILURE WORD		
430	KERNEL NO TRAP NEW MUO PC WORD		
431	KERNEL TRAP NEW MUO PC WORD		
432	SUPERVISOR NO TRAP NEW MUO PC WORD		
433	SUPERVISOR TRAP NEW MUO PC WORD		
434	CONCEALED NO TRAP NEW MUO PC WORD		
435	CONCEALED TRAP NEW MUO PC WORD		
436	PUBLIC NO TRAP NEW MUO PC WORD		
437	PUBLIC TRAP NEW MUO PC WORD		
440			
	AVAILABLE TO SOFTWARE		AVAILABLE TO SOFTWARE
777		777	

RADIX 50 REPRESENTATION

Radix 50 representation condenses 6-character symbols into 32 bits. The symbol characters are subscripted in the following manner

$$S_6 S_5 S_4 S_3 S_2 S_1$$

Determine the octal code (O_n) for each character and use the following formula to generate the 50_8 representation.

$$((((O_6 * 50) + O_5) * 50 + O_4) * 50 + O_3) * 50 + O_2) * 50 + O_1$$

OCTAL CODES

		Second Octal Digit							
		0	1	2	3	4	5	6	7
First Octal Digit	0	null	0	1	2	3	4	5	6
	1	7	8	9	A	B	C	D	E
	2	F	G	H	I	J	K	L	M
	3	N	O	P	Q	R	S	T	U
	4	V	W	X	Y	Z	•	\$	%

decsystem10

SYSTEM

REFERENCE CARD

(Including the DECsystem-1070)

digital

DEVICE MNEMONICS

SECOND AND THIRD OCTAL DIGITS

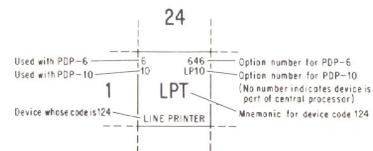
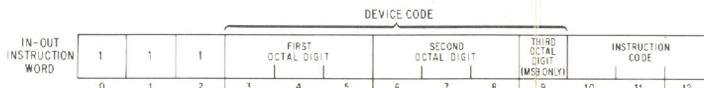
	00	04	10	14	20	24	30	34	40	44	50	54	60	64	70	74
FIRST OCTAL DIGIT	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
	APR CPA CENTRAL PROCESSOR	PI PRIORITY INTERRUPT	PAG* K110 PAGING	CCI PDP-8,9 INTERFACE	CCI2 PDP-8,9 INTERFACE	ADC ANALOG-DIGITAL CONVERTER	ADC2 ANALOG-DIGITAL CONVERTER						DLB PDP-11 DATA LINK	DLC PDP-11 DATA LINK	CLK REAL TIME CLOCK	CLK2 REAL TIME CLOCK
	6 10 261 10 6 10	6 10 760 10	10 6 10 6 10	10 6 10 6 10	10 6 10 6 10	10 6 10 6 10	10 6 10 6 10	10 6 10 6 10	10 6 10 6 10	10 6 10 6 10	10 6 10 6 10	10 6 10 6 10	10 6 10 6 10	10 6 10 6 10	10 6 10 6 10	10 6 10 6 10
	1 PTP TAPE PUNCH	1 PTR TAPE READER	1 CDP CARD PUNCH	1 CDR CARD READER	1 TTY CONSOLE TELETYPE	1 LPT LINE PRINTER	1 DIS DISPLAY	1 DIS2 DISPLAY	1 PLT PLOTTER	1 PLT2 PLOTTER	1 CR CARD READER	1 CR2 CARD READER	1 DLB2* PDP-11 DATA LINK	1 DLC2 PDP-11 DATA LINK	1 DSK DISK/DRUM	1 DSK2 DISK/DRUM
	6 10 136 6 10	6 10 136 6 10	6 10 136 6 10	6 10 551 6 10	6 10 551 6 10	6 10 551 6 10	6 10 551 6 10	6 10 551 6 10	6 10 551 6 10	6 10 551 6 10	6 10 551 6 10	6 10 551 6 10	6 10 551 6 10	6 10 551 6 10	6 10 551 6 10	6 10 551 6 10
	2 DC DATA CONTROL	2 DC2 DATA CONTROL	2 UTC DECTAPE	2 UTS DECTAPE	2 MTC MAGNETIC TAPE	2 MTS MAGNETIC TAPE	2 MTM1 MAGNETIC TAPE	2 LPT21 LINE PRINTER	2 DLS DATA LINE SCANNER	2 DLS2 DATA LINE SCANNER	2 DPC DISK PACK SYSTEM	2 DPC2 DISK PACK SYSTEM	2 DPC3 DISK PACK SYSTEM	2 DPC4 DISK PACK SYSTEM	2 RMC* DATA CONTROL	2 RMC2 DATA CONTROL
	6 10	6 10	6 10	6 10	6 10	6 10	6 10	6 10	6 10	6 10	6 10	6 10	6 10	6 10	6 10	6 10
	3 DCSA DATA COMMUNICATION	3 DCSB DATA COMMUNICATION			3 DTC DECTAPE	3 DTS DECTAPE	3 DTC2 DECTAPE	3 DTS2 DECTAPE	3 TMC MAGNETIC TAPE	3 TMS MAGNETIC TAPE	3 TMC2 MAGNETIC TAPE	3 TMS2 MAGNETIC TAPE				
	6 10	6 10	6 10	6 10	6 10	6 10	6 10	6 10	6 10	6 10	6 10	6 10	6 10	6 10	6 10	6 10
	4												4 DSS SINGLE SYNCHRONOUS LINE UNIT	4 DSI SINGLE SYNCHRONOUS LINE UNIT	4 DSS2 SINGLE SYNCHRONOUS LINE UNIT	4 DSI2 SINGLE SYNCHRONOUS LINE UNIT
	6 10	6 10	6 10	6 10	6 10	6 10	6 10	6 10	6 10	6 10	6 10	6 10	6 10	6 10	6 10	6 10
	5															
	6															
	7															

CODES IN THIS SECTION RESERVED FOR USER SPECIAL DEVICES

K110 UNRESTRICTED CODES RESERVED FOR USERS

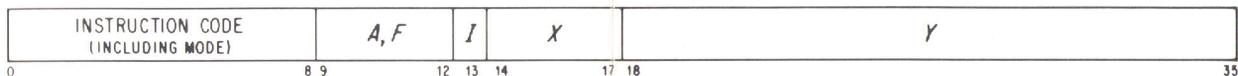
K110 UNRESTRICTED CODES RESERVED FOR DEC

* IN THE PDP-6 THESE CODES ARE USED FOR OTHER DEVICES † FOR A THIRD LINE PRINTER USE CODE 230
 010 DRUM PROCESSOR
 160 PDP-7,8 INTERFACE
 270 DISK FILE (DFF)

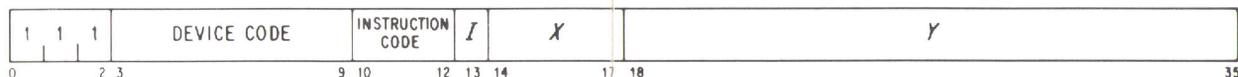


WORD FORMATS

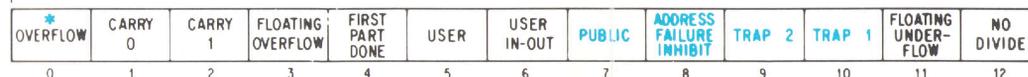
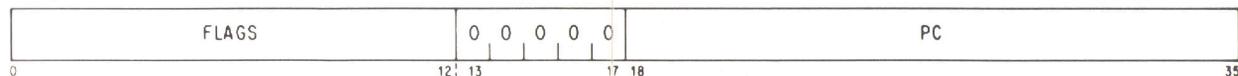
BASIC INSTRUCTIONS



IN-OUT INSTRUCTIONS



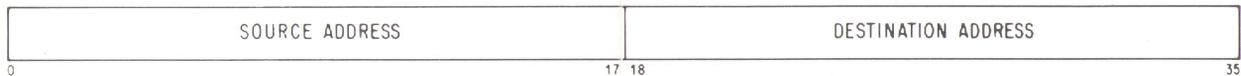
PC WORD



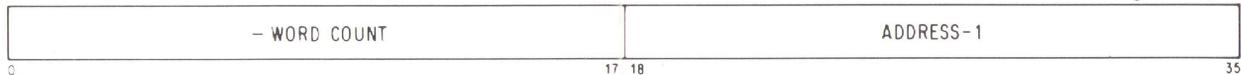
* DISABLE BYPASS IN K110 EXECUTIVE MODE

The items appearing in blue are K110 bits only.

BLT POINTER {XWD}



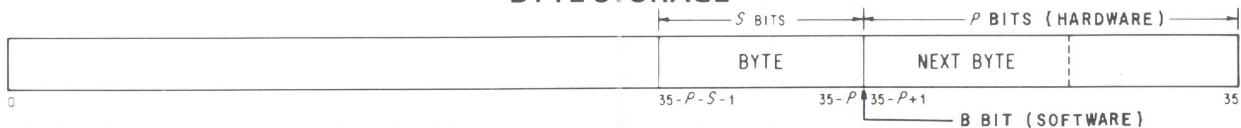
BLKI/BLKO POINTER, PUSHDOWN POINTER, DATA CHANNEL CONTROL WORD {IOWD}



BYTE POINTER



BYTE STORAGE

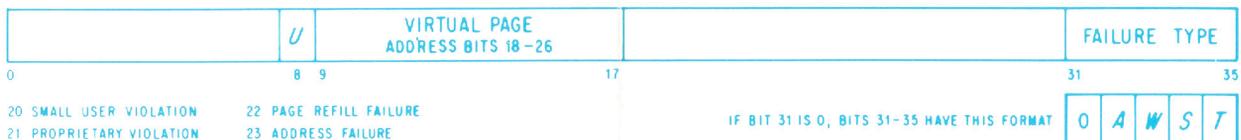


The MACRO POINT pseudo-op defines the b bit as the right-most bit number of the byte.

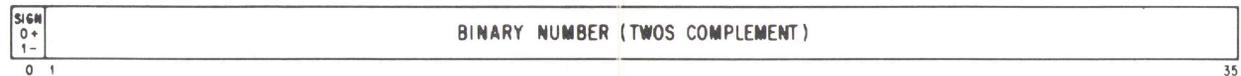
PAGE MAP WORD



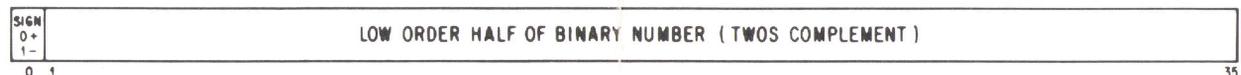
PAGE FAIL WORD



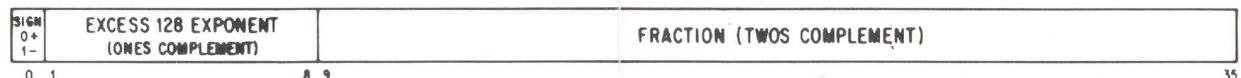
FIXED POINT OPERANDS



LOW ORDER WORD IN DOUBLE LENGTH FIXED POINT OPERANDS

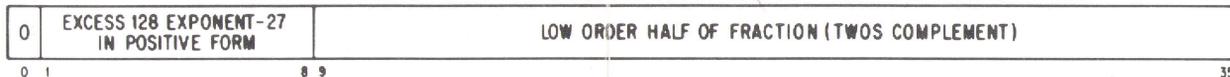


FLOATING POINT OPERANDS



The items appearing in blue are KI10 word formats only.

LOW ORDER WORD IN SOFTWARE DOUBLE LENGTH FLOATING POINT OPERANDS

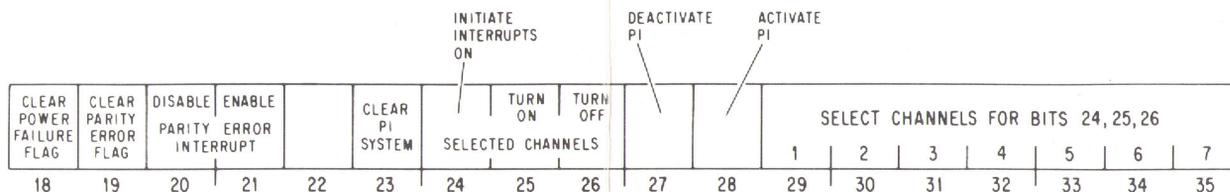


LOW ORDER WORD IN HARDWARE DOUBLE LENGTH FLOATING POINT OPERANDS

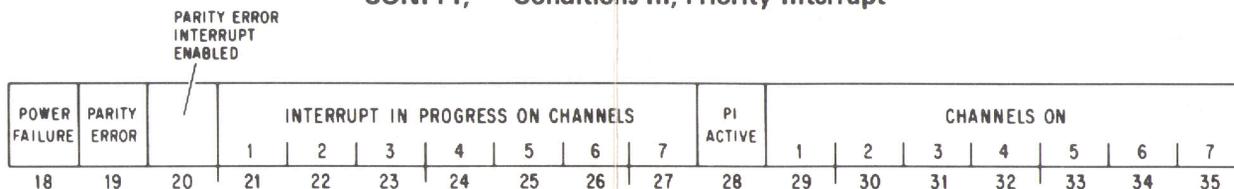


KA10 ONLY WORD FORMATS

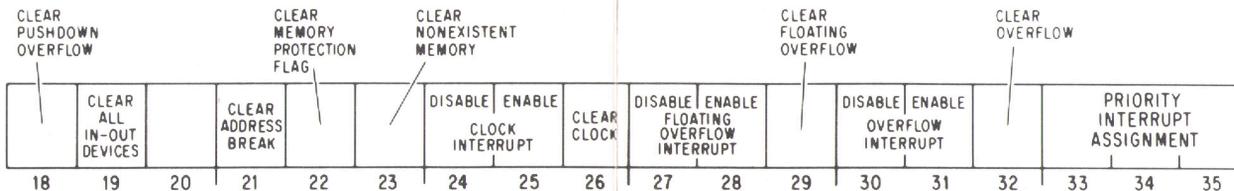
CONO PI, Conditions Out, Priority Interrupt



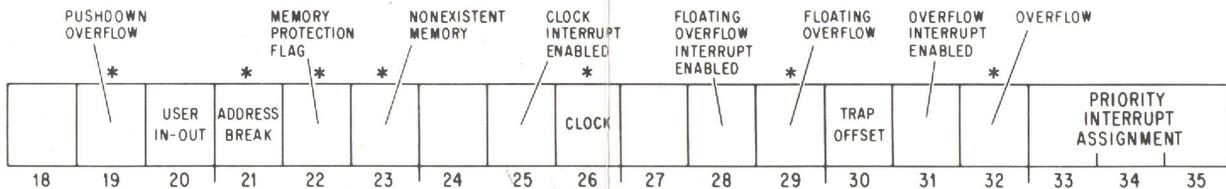
CONI PI, Conditions In, Priority Interrupt



CONO APR, Conditions Out, Arithmetic Processor



CONI APR, Conditions In, Arithmetic Processor

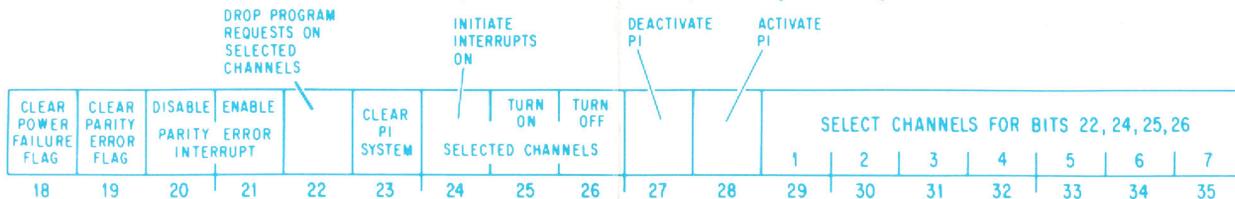


*These bits request interrupts.

The items appearing in blue are KI10 word formats only.

KI10 ONLY WORD FORMATS

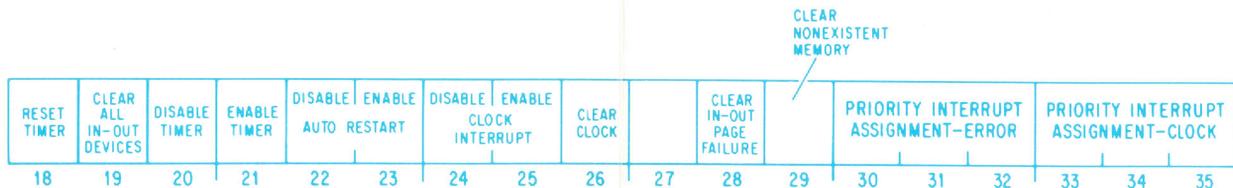
CONO PI, Conditions Out, Priority Interrupt



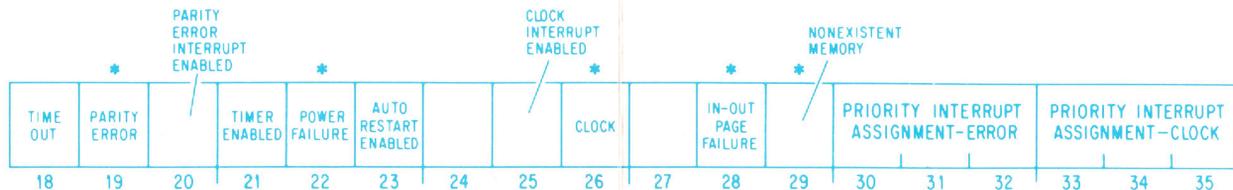
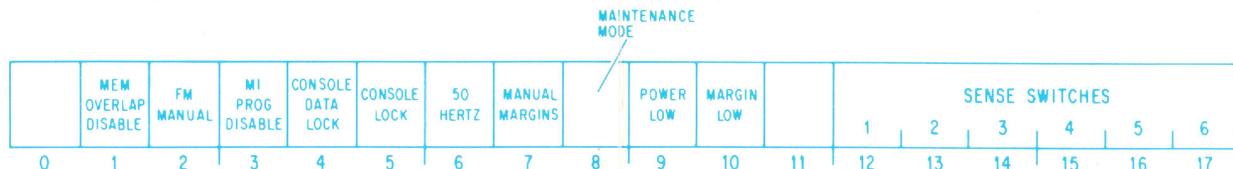
CONI PI, Conditions In, Priority Interrupt



CONO APR, Conditions Out, Arithmetic Processor

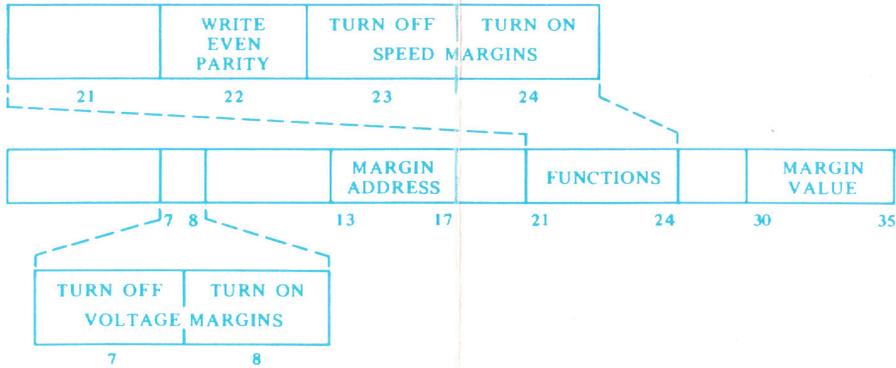


CONI APR, Conditions In, Arithmetic Processor

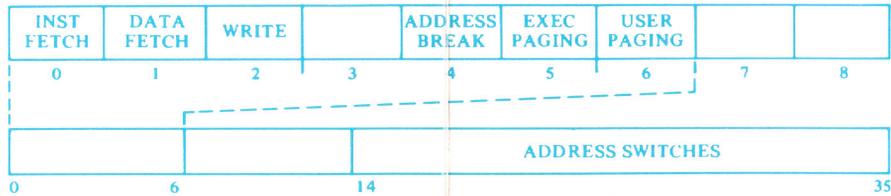


*These bits cause interrupts.

DATAO APR, Maintenance Data Out, Arithmetic Processor



DATAO PTR, Operating Data Out, Console



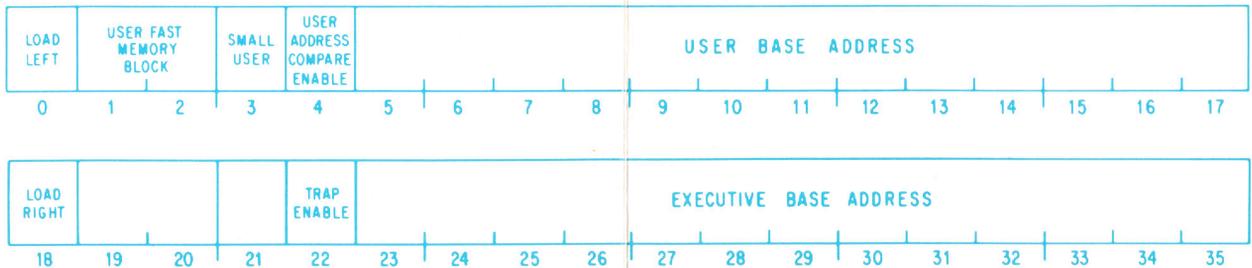
CONO PAG, Conditions Out, Paging



CONI PAG, Conditions In, Paging



DATAO PAG, Conditions Out, Paging



The items appearing in blue are KI10 word formats only.

INSTRUCTION SET

<p>MOV $\left\{ \begin{array}{l} \text{E} \\ \text{e Negative} \\ \text{e Magnitude} \\ \text{e Swapped} \end{array} \right\}$ $\left\{ \begin{array}{l} \text{to AC} \\ \text{Immediate to AC} \\ \text{to Memory} \\ \text{to Self} \end{array} \right\}$</p> <p>Half word $\left\{ \begin{array}{l} \text{Right} \\ \text{Left} \end{array} \right\}$ to $\left\{ \begin{array}{l} \text{Right} \\ \text{Left} \end{array} \right\}$ $\left\{ \begin{array}{l} \text{no effect} \\ \text{Ones} \\ \text{Zeros} \\ \text{Extend sign} \end{array} \right\}$</p> <p>BLOCK Transfer</p> <p>EXCHANGE AC and memory</p>	<p>ADD</p> <p>SUBtract</p> <p>MULTiPLY</p> <p>Integer MULTiPLY</p> <p>DIVide</p> <p>Integer DIVide</p> <p>$\left\{ \begin{array}{l} \sim \\ \text{Immediate to Memory} \\ \text{to Both} \end{array} \right\}$</p> <p>Floating $\left\{ \begin{array}{l} \text{AdD} \\ \text{SuBtract} \\ \text{MultiPly} \\ \text{DiVide} \end{array} \right\}$ $\left\{ \begin{array}{l} \text{and Round} \\ \sim \\ \text{Long to Memory} \\ \text{to Both} \end{array} \right\}$</p> <p>Floating Scale</p> <p>Double Floating Negate</p> <p>Unnormalized Floating Add</p> <p>FIX</p> <p>FIX and Round</p> <p>FLoaT and Round</p> <p>Double Floating $\left\{ \begin{array}{l} \text{AdD} \\ \text{SuBtract} \\ \text{MultiPly} \\ \text{DiVide} \end{array} \right\}$</p> <p>Double MOV $\left\{ \begin{array}{l} \text{E} \\ \text{e Negative} \end{array} \right\}$ $\left\{ \begin{array}{l} \sim \\ \text{to Memory} \end{array} \right\}$</p>
<p>use present pointer $\left\{ \begin{array}{l} \text{and} \\ \text{Load Byte into AC} \\ \text{Increment pointer} \end{array} \right\}$ and $\left\{ \begin{array}{l} \text{Load Byte into AC} \\ \text{DePosit Byte in memory} \end{array} \right\}$</p> <p>Increment Byte Pointer</p>	<p>FIX and Round</p> <p>FLoaT and Round</p>
<p>PUSH down $\left\{ \begin{array}{l} \sim \\ \text{and Jump} \end{array} \right\}$</p> <p>POP up $\left\{ \begin{array}{l} \sim \\ \text{and Jump} \end{array} \right\}$</p>	<p>Double Floating $\left\{ \begin{array}{l} \text{AdD} \\ \text{SuBtract} \\ \text{MultiPly} \\ \text{DiVide} \end{array} \right\}$</p> <p>Double MOV $\left\{ \begin{array}{l} \text{E} \\ \text{e Negative} \end{array} \right\}$ $\left\{ \begin{array}{l} \sim \\ \text{to Memory} \end{array} \right\}$</p>
<p>SET to $\left\{ \begin{array}{l} \text{Zeros} \\ \text{Ones} \\ \text{AC} \\ \text{Memory} \\ \text{Complement of AC} \\ \text{Complement of Memory} \end{array} \right\}$ $\left\{ \begin{array}{l} \text{to AC} \\ \text{Immediate} \\ \text{Memory} \\ \text{Both} \end{array} \right\}$</p> <p>AND inclusive OR $\left\{ \begin{array}{l} \sim \\ \text{with Complement of AC} \\ \text{with Complement of Memory} \\ \text{Complements of Both} \end{array} \right\}$ $\left\{ \begin{array}{l} \text{to AC} \\ \text{Immediate} \\ \text{Memory} \\ \text{Both} \end{array} \right\}$</p> <p>Inclusive OR</p> <p>EXclusive OR</p> <p>EQuivalence</p>	<p>Jump $\left\{ \begin{array}{l} \text{to SubRoutine} \\ \text{and Save Pc} \\ \text{and Restore AC} \\ \text{if Find First One} \\ \text{on Flag and CLear it} \\ \text{on OVerflow (JFCL 10,)} \\ \text{on CaRrY 0 (JFCL 4,)} \\ \text{on CaRrY 1 (JFCL 2,)} \\ \text{on CaRrY (JFCL 6,)} \\ \text{on Floating OVerflow (JFCL 1,)} \\ \text{and ReSTore} \\ \text{and ReSTore Flags (JRST 2,)} \\ \text{and ENable PI channel (JRST 12,)} \end{array} \right\}$</p> <p>HALT (JRST 4,)</p> <p>PORTAL (JRST 1,)</p> <p>eXeCuTe</p> <p>MAP</p>
<p>SKIP if memory</p> <p>JUMP if AC</p> <p>Add One to Subtract One from $\left\{ \begin{array}{l} \text{memory and Skip} \\ \text{AC and Jump} \end{array} \right\}$ if $\left\{ \begin{array}{l} \text{never} \\ \text{Less} \\ \text{Equal} \\ \text{Less or Equal} \\ \text{Always} \\ \text{Greater} \\ \text{Greater or Equal} \\ \text{Not equal} \end{array} \right\}$</p> <p>Compare AC $\left\{ \begin{array}{l} \text{Immediate} \\ \text{with Memory} \end{array} \right\}$ and skip if AC $\left\{ \begin{array}{l} \text{never} \\ \text{Less} \\ \text{Equal} \\ \text{Less or Equal} \\ \text{Always} \\ \text{Greater} \\ \text{Greater or Equal} \\ \text{Not equal} \end{array} \right\}$</p> <p>Add One to Both halves of AC and Jump if $\left\{ \begin{array}{l} \text{Positive} \\ \text{Negative} \end{array} \right\}$</p>	<p>DATA</p> <p>BLocK $\left\{ \begin{array}{l} \text{In} \\ \text{Out} \end{array} \right\}$</p> <p>CONDitions $\left\{ \begin{array}{l} \text{in and Skip if} \\ \text{all masked bits Zero} \\ \text{some masked bit One} \end{array} \right\}$</p>
<p>Arithmetic SHiFT</p> <p>Logical SHiFT</p> <p>ROTate $\left\{ \begin{array}{l} \sim \\ \text{Combined} \end{array} \right\}$</p>	<p>Test AC $\left\{ \begin{array}{l} \text{with Direct mask} \\ \text{with Swapped mask} \\ \text{Right with E} \\ \text{Left with E} \end{array} \right\}$ $\left\{ \begin{array}{l} \text{No modification} \\ \text{set masked bits to Zeros} \\ \text{set masked bits to Ones} \\ \text{Complement masked bits} \end{array} \right\}$ and skip $\left\{ \begin{array}{l} \text{never} \\ \text{if all masked bits Equal 0} \\ \text{if Not all masked bits equal 0} \\ \text{Always} \end{array} \right\}$</p>

NOTE: The instructions in blue are K110 instructions that are unassigned on the KA10.

ASCII CHARACTER SET
ASCII-1968 (ANSI X3.4-1968)

To obtain octal ASCII, decimal ASCII, or DECsystem-10 SIXBIT representation of a character, add the row value to the column value.

Column Value Row Value	000	008	016	024	032	040	048	056	064	072	080	088	096	104	112	120
	000	010	020	030	040	050	060	070	100	110	120	130	140	150	160	170
0	NUL	BS	DLE	CAN	space	(0	8	@	H	P	X	grave	h	p	x
1	SOH	HT	DC1	EM	!)	1	9	A	I	Q	Y	a	i	q	y
2	STX	LF	DC2	SUB	"	*	2	:	B	J	R	Z	b	j	r	z
3	ETX	VT	DC3	ESC	#	+	3	;	C	K	S	[c	k	s	{
4	EOT	FF	DC4	FS	\$,	4	<	D	L	T	\	d	l	t	
5	ENQ	CR	NAK	GS	%	-	5	=	E	M	U]	e	m	u	}
6	ACK	SO	SYN	RS	&	.	6	>	F	N	V	([†])	f	n	v	(ESC)
7	BEL	SI	ETB	US	'apos	/	7	?	G	O	W	([†]) (-)	g	o	w	DEL

64 character graphic subset
 95 character graphic subset

Differences in the ASCII Standard

Octal	(ASCII 1963)	ASCII 1968
136	↑	^ (circumflex)
137	←	_ (underline)
176	ESC	~

NUL	NULL	DLE	DATA LINK ESCAPE (†P)
SOH	START OF HEADING (†A)	DC1	DEVICE CONTROL 1 (†Q)
STX	START OF TEXT (†B)	DC2	DEVICE CONTROL 2 (†R)
ETX	END OF TEXT (†C)	DC3	DEVICE CONTROL 3 (†S)
EOT	END OF TRANSMISSION (†D)	DC4	DEVICE CONTROL 4 (STOP) (†T)
ENQ	ENQUIRY (†E)	NAK	NEGATIVE ACKNOWLEDGE (†U)
ACK	ACKNOWLEDGE (†F)	SYN	SYNCHRONOUS IDLE (†V)
BEL	BELL (†G)	ETB	END OF TRANSMISSION BLOCK (†W)
BS	BACKSPACE (†H)	CAN	CANCEL (†X)
HT	HORIZ. TABULATION (†I)	EM	END OF MEDIUM (†Y)
LF	LINE FEED (†J)	SUB	SUBSTITUTE (†Z)
VT	VERT. TABULATION (†K)	ESC	ESCAPE (†)
FF	FORM FEED (†L)	FS	FILE SEPARATOR (†\)
CR	CARRIAGE RETURN (†M)	GS	GROUP SEPARATOR (†])
SO	SHIFT OUT (†N)	RS	RECORD SEPARATOR (†^)
SI	SHIFT IN (†O)	US	UNIT SEPARATOR (†_)
		DEL	DELETE (RUBOUT)

On most teleprinters, the † x character is produced by depressing the CTRL key and at the same time depressing the x character key.

NOTES

- SIXBIT is not part of any ASCII standard. It is used by DECsystem-10 programs as a code compression technique for the 64 character graphic subset of ASCII.
- Teleprinters manufactured by Teletype Corporation, Skokie, Illinois, have used codes 175 (ALT) and 176 for ESC. Programs may forgo the use of } (175) and ~ (176) in order to use these codes as ESC on older teleprinters.
- ASCII is a seven bit character code with an optional odd parity bit (200) added for many devices. Programs normally use just seven bits internally; the 200 bit is either stripped or added so the program will operate with either parity or non-parity generating devices.
- ISO Recommendation R646 and CCITT Recommendation V.3 (International Alphabet No. 5) is identical to ASCII except that number sign (043) is represented as £ instead of # and certain characters are reserved for national use.

POWERS OF TWO AND EIGHT

$2^N, 8^M$ N, M $2^{-N}, 8^{-M}$

	1	0 0	1 0
	2	1	0.5
	4	2	0.25
	8	3 1	0.125
	16	4	0.062 5
	32	5	0.031 25
	64	6 2	0.015 625
	128	7	0.007 812 5
	256	8	0.003 906 25
	512	9 3	0.001 953 125
	1 024	10	0.000 976 562 5
	2 048	11	0.000 488 281 25
	4 096	12 4	0.000 244 140 625
	8 192	13	0.000 122 070 312 5
	16 384	14	0.000 061 035 156 25
	32 768	15 5	0.000 030 517 578 125
	65 536	16	0.000 015 258 789 062 5
	131 072	17	0.000 007 629 394 531 25
	262 144	18 6	0.000 003 814 697 265 625
	524 288	19	0.000 001 907 348 632 812 5
	1 048 576	20	0.000 000 953 674 316 406 25
	2 097 152	21 7	0.000 000 476 837 158 203 125
	4 194 304	22	0.000 000 238 418 579 101 562 5
	8 388 608	23	0.000 000 119 209 289 550 781 25
	16 777 216	24 8	0.000 000 059 604 644 775 390 625
	33 554 432	25	0.000 000 029 802 322 387 695 312 5
	67 108 864	26	0.000 000 014 901 161 193 847 656 25
	134 217 728	27 9	0.000 000 007 450 580 596 923 828 125
	268 435 456	28	0.000 000 003 725 290 298 461 914 062 5
	536 870 912	29	0.000 000 001 862 645 149 230 957 031 25
	1 073 741 824	30 10	0.000 000 000 931 322 574 615 478 515 625
	2 147 483 648	31	0.000 000 000 465 661 287 307 739 257 812 5
	4 294 967 296	32	0.000 000 000 232 830 643 653 869 628 906 25
	8 589 934 592	33 11	0.000 000 000 116 415 321 826 934 814 453 125
	17 179 869 184	34	0.000 000 000 058 207 660 913 467 407 226 562 5
	34 359 738 368	35	0.000 000 000 029 103 830 456 733 703 613 281 25
	68 719 476 736	36 12	0.000 000 000 014 551 915 228 366 851 806 640 625
	137 438 953 472	37	0.000 000 000 007 275 957 614 183 425 903 320 312 5
	274 877 906 944	38	0.000 000 000 003 637 978 807 091 712 951 660 156 25
	549 755 813 888	39 13	0.000 000 000 001 818 989 403 545 856 475 830 078 125
	1 099 511 627 776	40	0.000 000 000 000 909 494 701 772 928 237 915 039 062 5
	2 199 023 255 552	41	0.000 000 000 000 454 747 350 886 464 118 957 519 531 25
	4 398 046 511 104	42 14	0.000 000 000 000 227 373 675 443 232 059 478 759 765 625
	8 796 093 022 208	43	0.000 000 000 000 113 686 837 721 616 029 739 379 882 812 5
	17 592 186 044 416	44	0.000 000 000 000 056 843 418 860 808 014 869 689 941 406 25
	35 184 372 088 832	45 15	0.000 000 000 000 028 421 709 430 404 007 434 844 970 703 125
	70 368 744 177 664	46	0.000 000 000 000 014 210 854 715 202 003 717 422 485 351 562 5
	140 737 488 355 328	47	0.000 000 000 000 007 105 427 357 601 001 858 711 242 675 781 25
	281 474 976 710 656	48 16	0.000 000 000 000 003 552 713 678 800 500 929 355 621 337 890 625
	562 949 953 421 312	49	0.000 000 000 000 001 776 356 839 400 250 464 677 810 668 945 312 5
	1 125 899 906 842 624	50	0.000 000 000 000 000 888 178 419 700 125 232 338 905 334 472 656 25
	2 251 799 813 685 248	51 17	0.000 000 000 000 000 444 889 209 850 062 616 169 452 667 236 328 125
	4 503 599 627 370 496	52	0.000 000 000 000 000 222 044 604 925 031 308 084 726 333 618 164 062 5
	9 007 199 254 740 992	53	0.000 000 000 000 000 111 022 302 462 515 654 042 363 166 809 082 031 25
	18 014 398 509 481 984	54 18	0.000 000 000 000 000 055 511 151 231 257 827 021 181 583 404 541 015 625
	36 028 797 018 963 968	55	0.000 000 000 000 000 027 755 575 615 628 913 510 590 791 702 270 507 812 5
	72 057 594 037 927 936	56	0.000 000 000 000 000 013 377 787 807 814 456 755 295 395 851 135 253 906 25
	144 115 188 075 855 872	57 19	0.000 000 000 000 000 006 938 893 903 907 228 377 647 697 925 567 626 953 125
	288 230 376 151 711 744	58	0.000 000 000 000 000 003 469 446 951 953 614 188 823 848 962 783 813 476 562 5
	576 460 752 303 423 488	59	0.000 000 000 000 000 001 734 723 475 976 807 094 411 924 481 391 906 738 281 25
	1 152 921 504 606 846 976	60 20	0.000 000 000 000 000 000 867 361 737 988 403 547 205 962 240 695 953 369 140 625
	2 305 843 009 213 693 952	61	0.000 000 000 000 000 000 433 680 868 994 201 773 602 981 120 347 976 684 570 312 5
	4 611 686 018 427 387 904	62	0.000 000 000 000 000 000 216 840 434 497 100 886 801 490 560 173 988 342 285 156 25
	9 223 372 036 854 775 808	63 21	0.000 000 000 000 000 000 108 420 217 248 550 443 400 745 280 086 994 171 142 578 125
	18 446 744 073 709 551 616	64	0.000 000 000 000 000 000 054 210 108 624 275 221 700 372 640 043 497 085 571 289 062 5
	36 893 488 147 419 103 232	65	0.000 000 000 000 000 000 027 105 054 312 137 610 850 186 320 021 748 542 785 644 531 25
	73 786 976 294 838 206 464	66 22	0.000 000 000 000 000 000 013 552 527 156 068 805 425 093 160 010 874 271 392 822 265 625
	147 573 952 589 676 412 928	67	0.000 000 000 000 000 000 006 776 263 578 034 402 712 546 580 005 437 135 696 411 132 812 5
	295 147 905 179 352 825 856	68	0.000 000 000 000 000 000 003 388 131 789 017 201 356 273 290 002 718 567 848 205 566 406 25
	590 295 810 358 705 651 712	69 23	0.000 000 000 000 000 000 001 694 065 894 508 600 678 136 645 001 359 283 924 102 783 203 125
	1 180 591 620 717 411 303 424	70	0.000 000 000 000 000 000 000 847 032 947 254 300 339 068 322 500 679 641 962 051 391 601 562 5
	2 361 183 241 434 822 606 848	71	0.000 000 000 000 000 000 000 423 516 473 627 150 169 534 161 250 339 820 981 025 695 800 781 25
	4 722 366 482 869 645 213 696	72 24	0.000 000 000 000 000 000 000 211 758 236 813 575 084 767 080 625 169 910 490 512 847 900 390 625

OCTAL-DECIMAL CONVERSION TABLE

Bits 18-20		Bits 21-23		Bits 24-26		Bits 27-29		Bits 30-32		Bits 33-35	
Oct	Dec	Oct	Dec	Oct	Dec	Oct	Dec	Oct	Dec	Oct	Dec
0	0	0	0	0	0	0	0	0	0	0	0
100000	32768	10000	4096	2000	512	100	64	10	8	1	1
200000	65536	20000	8192	2000	1024	200	128	20	16	2	2
300000	98304	30000	12288	3000	1536	300	192	30	24	3	3
400000	131072	40000	16384	4000	2048	400	256	40	32	4	4
500000	163840	50000	20480	5000	2560	500	320	50	40	5	5
600000	196608	60000	24576	6000	3072	600	384	60	48	6	6
700000	229376	70000	28672	7000	3584	700	448	70	56	7	7

Octal to Decimal

For each position of the octal number, locate the octal digit and its decimal equivalent. Add the decimal equivalents to obtain the decimal number.

Example: $53702_8 = ?_{10}$

$50000_8 = 20480_{10}$
$3000_8 = 1536_{10}$
$700_8 = 448_{10}$
$00_8 = 0_{10}$
$2_8 = 2_{10}$
<hr/>
$53702_8 = 22466_{10}$

Decimal to Octal

Locate the decimal number, or the next lower decimal number if the desired number is not found, in the table. Record the octal equivalent. Subtract the decimal number appearing in the table from the original decimal number. Using the difference obtained, repeat the process and add the octal numbers obtained until the entire number is generated.

Example: $22466_{10} = ?_8$

$20480_{10} = 50000_8$	Subtract	22466	
		<hr/>	1986
$1536_{10} = 3000_8$	Subtract	1986	
		<hr/>	1536
			<hr/>
			450
$448_{10} = 700_8$	Subtract	450	
		<hr/>	448
			<hr/>
			2
			<hr/>
			2
$2_{10} = 2_8$	Subtract	2	
		<hr/>	0
$22466_{10} = 53702_8$			

CARD CODES (ANSI X3.26-1970)

Zone	12		11		10		9		8		7		6		5		4		3		2		1	
	Digit																							
			0	space	{		}																	
1	A	J	/	1	a	j	~	SOH	DC1															
2	B	K	S	2	b	k	s	STX	DC2			SYN												
3	C	L	T	3	c	l	t	ETX	DC3															
4	D	M	U	4	d	m	u																	
5	E	N	V	5	e	n	v	HT		LF														
6	F	O	W	6	f	o	w		BS	ETB														
7	G	P	X	7	g	p	x	DEL		ESC	EOT													
8	H	Q	Y	8	h	q	y		CAN															
9	I	R	Z	9	i	r	z																	
8-1					grave																			
8-2	[\	:																				
8-3	.	\$	#						VT															
8-4	<	*	%	@					FF	FS		DC4												
8-5	()	_	'					CR	GS	ENQ	NAK												
8-6	+	:	>	=					SO	RS	ACK													
8-7	!	^	?	"					SI	US	BEL	SUB												

NOTES

To determine the card punch for a particular character, locate the character in the table and read the corresponding zone punch and then digit punch. For example, the card punch for a % is 0-8-4.

To obtain the character corresponding to a particular card punch, locate the junction of the zone punch and the digit punch. For example, the character corresponding to the card punch 12-11-9 is r.

Slots that do not contain characters represent card punches for which there are no ASCII equivalents.

The end-of-file card is one containing 12-11-0-1-6-7-8-9 in column 1.

INSTRUCTION CODES

	--0	--1	--2	--3	--4	--5	--6	--7
00-01-02-03	(ILLEGAL)							
04-05-06-07	CALL OPEN SET/STS CLOSE	INIT ITCALL STATO RELEASE	LEFT RESERVED FOR DFC GET/STS METAPE	RESERVED FOR DFC STA/IZ UG/TF	SPECIAL MONITORS INBUF USE/IT	MONITORS RENAME OUTBUF USE/TO	IN INPUT LOOKUP	CALLI OUTPUT FNTR
10-11-12-13	UFEN DFAD DMOVE/ UFA	DESB DMOVN DFN	DEMP FIX FSC	DI/DV IBP	DMOVE/M ILDB	DMOVN/LDB	FIXR IDPB	FLTR DPB
14-15-16-17	FAD FSB FMP FDV	-L -L -L -L	-M -M -M -M	-B -B -B -B	FADR FSBR FMPR FDVR	-I -I -I -I	-M -M -M -M	-B -B -B -B
20-21-22-23	MOVE MOVN IMUL IDIV	-I -I -I -I	-M -M -M -M	-S -S -B -B	MOVS MOVN MUL DIV	-I -I -I -I	-M -M -M -M	-S -S -B -B
24-25-26-27	ASH EXCH PUSHJ ADD	ROT BLT PUSHJ	LSH AOBJP POP	JFFO AOBJN POPJ	ASHC JRST JSR SUB	ROTC JFCCL JSP	LSHC XCT JSA	MAP JRA
30-31-32-33	CAI CAM JUMP SKIP	-L -L -L -L	-E -E -E -E	-LE -LE -LE -LE	-A -A -A -A	-GE -GE -GE -GE	-N -N -N -N	-G -G -G -G
34-35-36-37	AOJ AOS SOJ SOS	-L -L -L -L	-E -E -E -E	-LE -LE -LE -LE	-A -A -A -A	-GE -GE -GE -GE	-N -N -N -N	-G -G -G -G
40-41-42-43	SETZ ANDCA ANDCM XOR	-I -I -I -I	-M -M -M -M	-B -B -B -B	AND SETM SETA JOR	-I -I -I -I	-M -M -M -M	-B -B -B -B
44-45-46-47	ANDCB SETCA SETCM ORCB	-I -I -I -I	-M -M -M -M	-B -B -B -B	EOV ORCA ORCM SETO	-I -I -I -I	-M -M -M -M	-B -B -B -B
50-51-52-53	HLL HLLZ HLL0 HLL1	-I -I -I -I	-M -M -M -M	-S -S -S -S	HLR HRLZ HRL0 HRL1	-I -I -I -I	-M -M -M -M	-S -S -S -S
54-55-56-57	HRR HRRZ HRR0 HRR1	-I -I -I -I	-M -M -M -M	-S -S -S -S	HLR HLRZ HLRO HLRE	-I -I -I -I	-M -M -M -M	-S -S -S -S
60-61-62-63	TRN TDN TRZ TDZ	TLN TSN TLZ TSZ	TRNE TDNE TRZE TDZE	TLNE TSNE TLZE TSZE	TRNA TDNA TRZA TDZA	TLNA TDNA TLZA TDZA	TRNN TDNN TRZN TDZN	TLNN TSNN TLZN TSNZ
64-65-66-67	TRC TDC TRO TDO	TLC TSC TLO TSO	TRCE TDCE TROE TDOE	TLCE TDCE TLOE TSOE	TRCA TDCA TROA TDOA	TLCA TDCA TLOA TSOA	TRCN TD CN TRON TDON	TLCN TSCN TSCN 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 The instructions in blue are K110 instructions that are unassigned on the KA10.
- 7--20-CONO
- 7--24-CONI
- 7--30-CONSZ
- 7--34-CONSO

Order No. DEC-10-XSRCA-B-D. For single copies of this card, contact a DIGITAL sales office. Multiple copies may be ordered in lots of 25 from the Program Library, Digital Equipment Corporation, Maynard, Mass. 01754.

Copyright © 1972, 1973, 1974 by Digital Equipment Corporation

Created by DECsystem-10 Software Documentation Group