

decsystem10

SYSTEM

REFERENCE CARD

(Including the DECsystem-1070)

digital

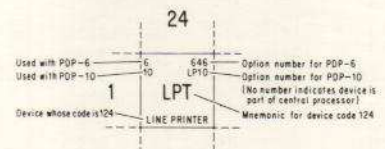
SECOND AND THIRD OCTAL DIGITS		DEVICE MNEMONICS															
		00	04	10	14	20	24	30	34	40	44	50	54	60	64	70	74
FIRST OCTAL DIGIT	0	6,10 APR CPA CENTRAL PROCESSOR	6,10 PI PRIORITY INTERRUPT	10 PAG* K110 PAGING	10 CCI PDP-8,9 INTERFACE	10 CCI2 PDP-8,9 INTERFACE	10 ADC ANALOG-DIGITAL CONVERTER	10 ADC2 ANALOG-DIGITAL CONVERTER						10 DLB PDP-11 DATA LINK	10 DLC DL10	10 CLK REAL TIME CLOCK	10 CLK2 REAL TIME CLOCK
		6 PTP PAPER TAPE PUNCH	761 PTR PAPER TAPE READER	760 CDP CARD PUNCH	CP10 CDR CARD READER	461 TTY CONSOLE TELETYPE	626 LPT LINE PRINTER	646 DIS DISPLAY	6,10 DIS2 DISPLAY	340 PLT PLOTTER	10 XY10 PLOTTER	10 CR CARD READER	10 CR2 CARD READER	10 DLB2* PDP-11 DATA LINK	10 DLC2 DISK/DRUM	10 DSK DISK/DRUM	10 DSK2 DISK/DRUM
1		6 DC DATA CONTROL	136 DC2 DATA CONTROL	630 UTC DECTAPE	551 UTS DECTAPE	6 MTC MAGNETIC TAPE	6 MTS MAGNETIC TAPE	516 MTM ¹ MAGNETIC TAPE	646 LPT2 ¹ LINE PRINTER	10 DLS DATA LINE SCANNER	10 DLS2 DATA LINE SCANNER	10 DPC DISK PACK SYSTEM	10 DPC2 DISK PACK SYSTEM	10 DPC3 DISK PACK SYSTEM	10 DPC4 DISK PACK SYSTEM	10 RMC* DATA CONTROL	10 RMC2 DATA CONTROL
		3 DCSA DATA COMMUNICATION	DCSB			10 DTC DECTAPE	10 DTS DECTAPE	10 DTC2 DECTAPE	10 DTS2 DECTAPE	10 TMC MAGNETIC TAPE	10 TMS MAGNETIC TAPE	10 TMC2 MAGNETIC TAPE	10 TMS2 MAGNETIC TAPE				
4														10 DSS SINGLE LINE UNIT	10 DSI SYNCHRONOUS LINE UNIT	10 DSS2 SINGLE LINE UNIT	10 DSI2 SYNCHRONOUS LINE UNIT
5																	
6																	
7																	

CODES IN THIS SECTION RESERVED FOR USER SPECIAL DEVICES

K110 UNRESTRICTED CODES
RESERVED FOR USERS

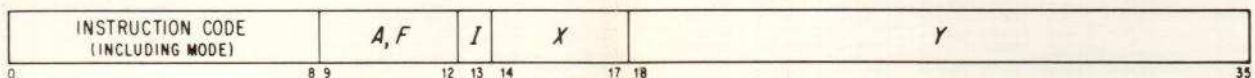
K110 UNRESTRICTED CODES
RESERVED FOR DEC

*IN THE PDP-6 THESE CODES ARE USED FOR OTHER DEVICES 1 FOR A THIRD LINE PRINTER USE CODE 230.
010 DRUM PROCESSOR
160 PDP-7,8 INTERFACE
270 DISK FILE (DFI)

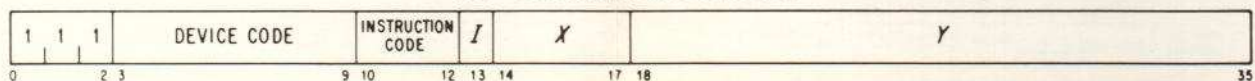


WORD FORMATS

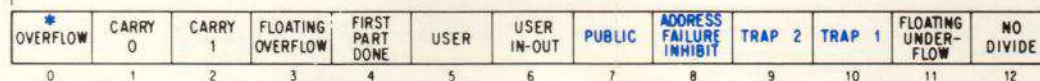
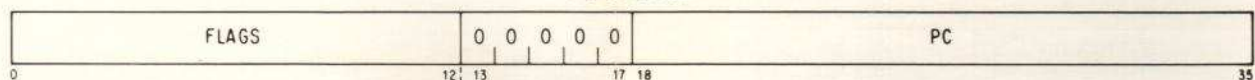
BASIC INSTRUCTIONS



IN-OUT INSTRUCTIONS



PC WORD



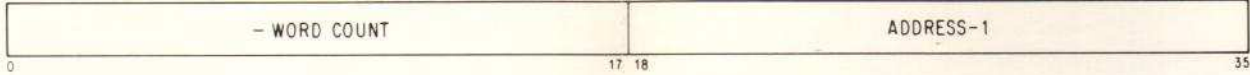
*DISABLE BYPASS IN K110 EXECUTIVE MODE

The items appearing in blue are K110 bits only.

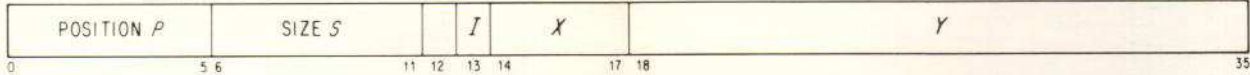
BLT POINTER {XWD}



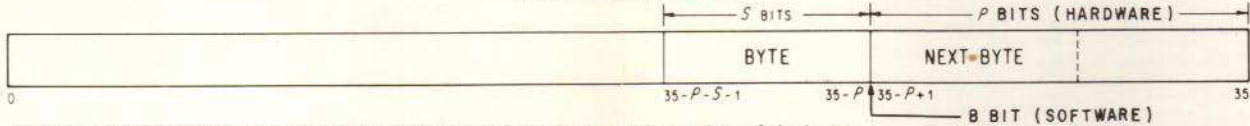
BLKI/BLKO POINTER, PUSHDOWN POINTER, DATA CHANNEL CONTROL WORD {IOWD}



BYTE POINTER

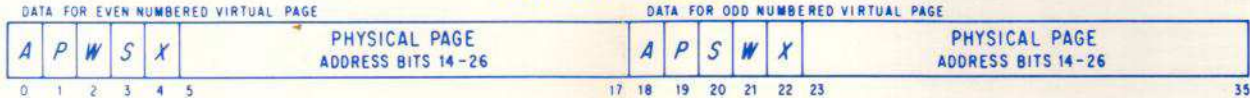


BYTE STORAGE



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

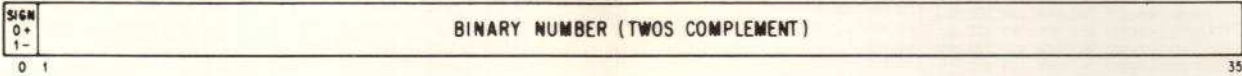
PAGE MAP WORD



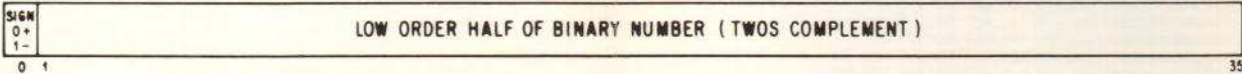
PAGE FAIL WORD



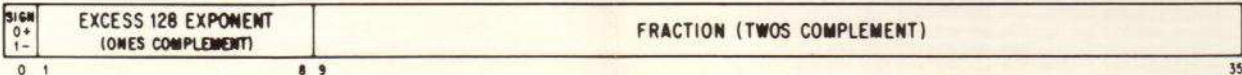
FIXED POINT OPERANDS



LOW ORDER WORD IN DOUBLE LENGTH FIXED POINT OPERANDS

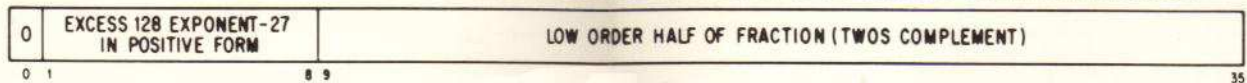


FLOATING POINT OPERANDS

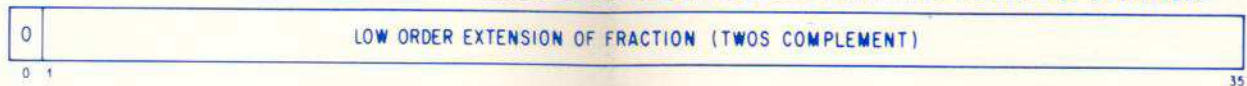


The items appearing in blue are KI10 word formats only.

LOW ORDER WORD IN SOFTWARE DOUBLE LENGTH FLOATING POINT OPERANDS

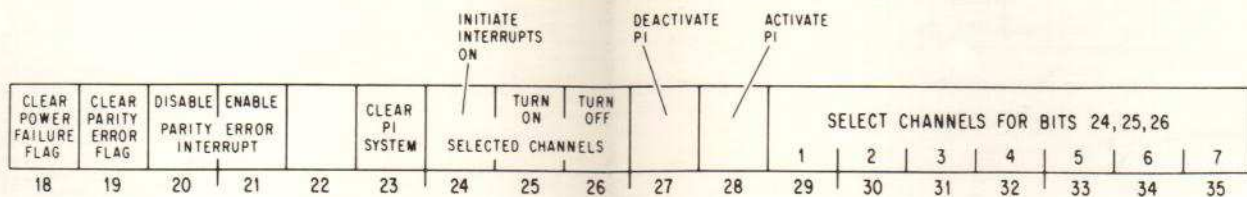


LOW ORDER WORD IN HARDWARE DOUBLE LENGTH FLOATING POINT OPERANDS

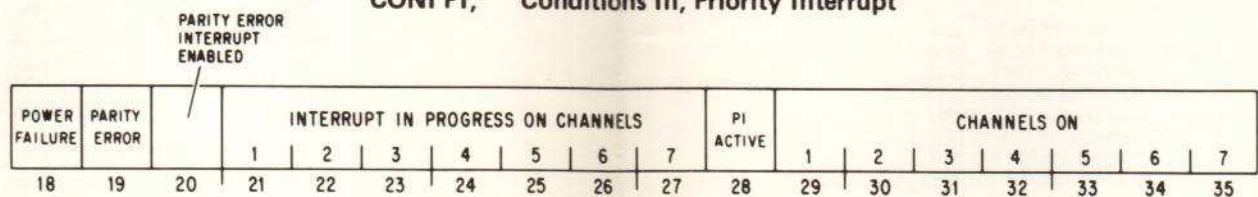


KA10 ONLY WORD FORMATS

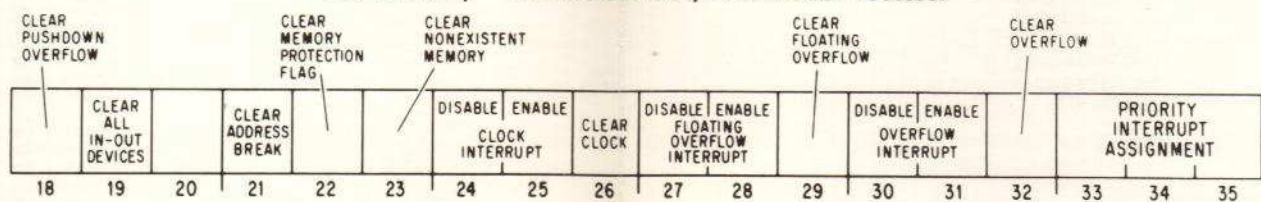
CONO PI, Conditions Out, Priority Interrupt



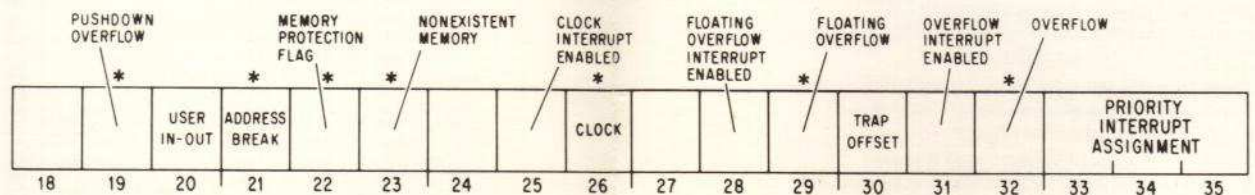
CONI PI, Conditions In, Priority Interrupt



CONO APR, Conditions Out, Arithmetic Processor



CONI APR, Conditions In, Arithmetic Processor

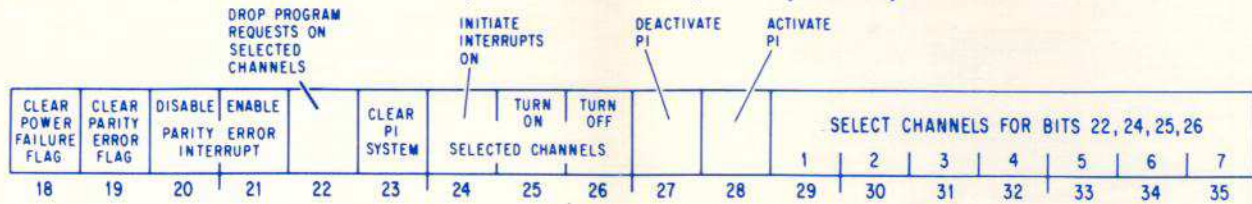


The items appearing in blue are KI10 word formats only.

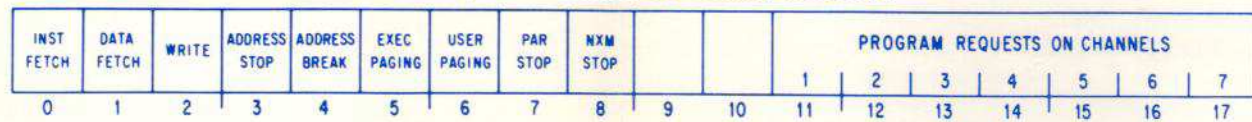
*These bits request interrupts.

KI10 ONLY WORD FORMATS

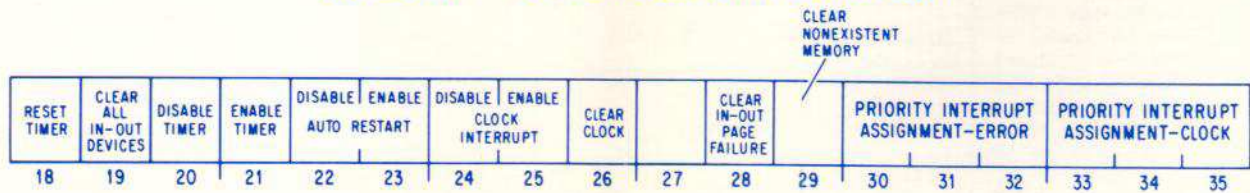
CONO PI, Conditions Out, Priority Interrupt



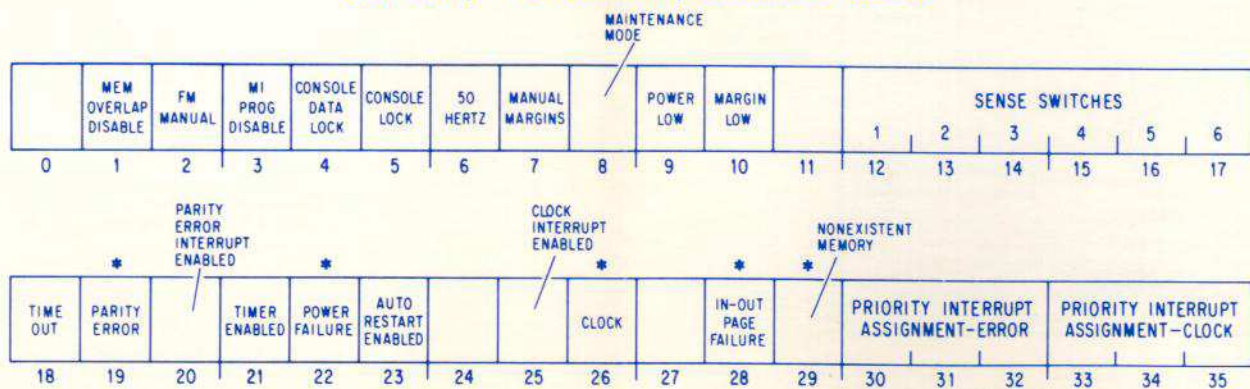
CONI PI, Conditions In, Priority Interrupt



CONO APR, Conditions Out, Arithmetic Processor

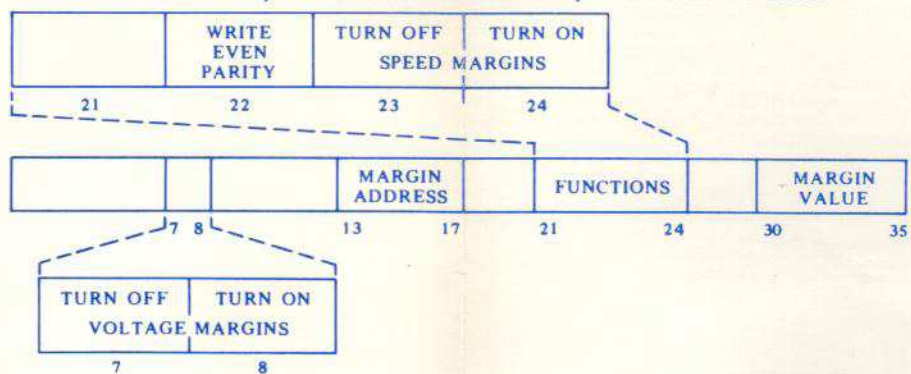


CONI APR, Conditions In, Arithmetic Processor

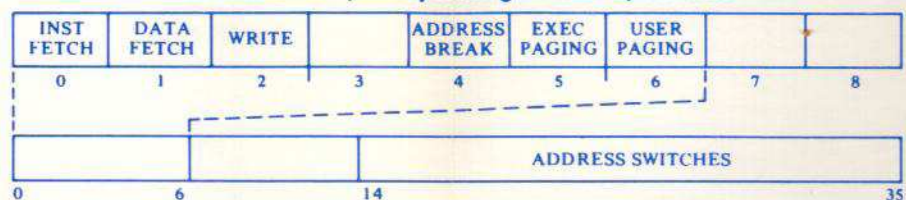


*These bits cause interrupts.

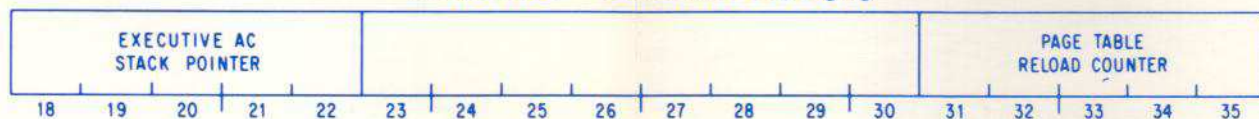
DATAO APR, Maintenance Data Out, Arithmetic Processor



DATAO PTR, Operating Data Out, Console



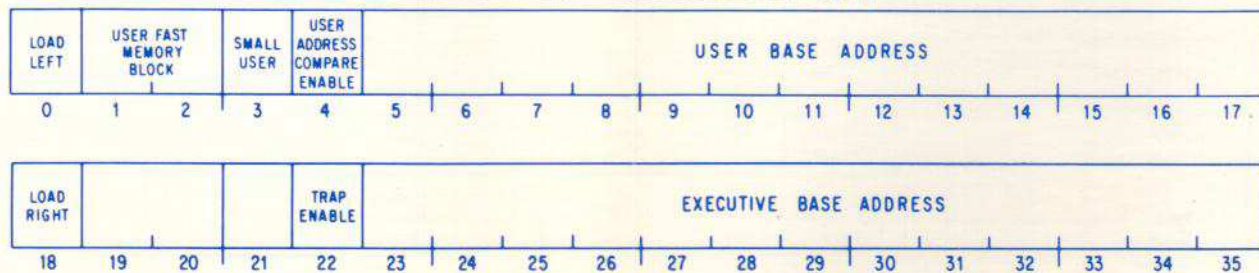
CONO PAG, Conditions Out, Paging



CONI PAG, Conditions In, Paging



DATAO PAG, Conditions Out, Paging



The items appearing in blue are KI10 word formats only.

PROCESS TABLE CONFIGURATION

USER PROCESS TABLE		EXECUTIVE PROCESS TABLE	
0	USER PAGE 0	0	AVAILABLE TO SOFTWARE
17	USER PAGE 36	37	
20	USER PAGE 40	40	EXECUTIVE LUID STORED HERE
		41	LUID 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 FAILURE TRAP INSTRUCTION	420	EXECUTIVE PAGE FAILURE TRAP INSTRUCTION
421	EXECUTIVE ARITHMETIC OVERFLOW TRAP INSTRUCTION	421	EXECUTIVE ARITHMETIC OVERFLOW TRAP INSTRUCTION
422	EXECUTIVE PUSHDOWN OVERFLOW TRAP INSTRUCTION	422	EXECUTIVE PUSHDOWN OVERFLOW TRAP INSTRUCTION
423	EXECUTIVE TRAP 3 TRAP INSTRUCTION	423	EXECUTIVE TRAP 3 TRAP INSTRUCTION
424	MUO STORED HERE	424	
425	PC WORD OF MUO STORED HERE		
426	EXECUTIVE PAGE FAILURE WORD		
427	EXECUTIVE PAGE FAILURE WORD		
428	KERNEL NO TRAP NEW MUO PC WORD		
429	KERNEL TRAP NEW MUO PC WORD		
430	SUPERVISOR NO TRAP NEW MUO PC WORD		
431	SUPERVISOR TRAP NEW MUO PC WORD		
432	CONCEALED NO TRAP NEW MUO PC WORD		
433	CONCEALED TRAP NEW MUO PC WORD		
434	PUBLIC NO TRAP NEW MUO PC WORD		
435	PUBLIC TRAP NEW MUO PC WORD		
436			
437			
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.

$$((((O_6 \cdot 50) + O_5) \cdot 50 + O_4) \cdot 50 + O_3) \cdot 50 + O_2) \cdot 50 + O_1$$

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	•	\$	%

INSTRUCTION SET

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

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	SIXBIT octal
000	010	020	030	040	050	060	070	100	110	120	130	140	150	160	170	170	decimal ASCII
0	NUL	BS	DLE	CAN	space	(0	8	@	H	P	X	grave	h	p	x	octal ASCII
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	↑	ˆ (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 (↑I)
FF	FORM FEED (↑L)	FS	FILE SEPARATOR (↑\)
CR	CARRIAGE RETURN (↑M)	GS	GROUP SEPARATOR (↑I)
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	0.000 000 000 000 000 444 089 209 850 062 616 169 452 667 236 328 125
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	0.000 000 000 000 000 013 877 787 807 814 456 755 295 395 851 135 253 906 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

