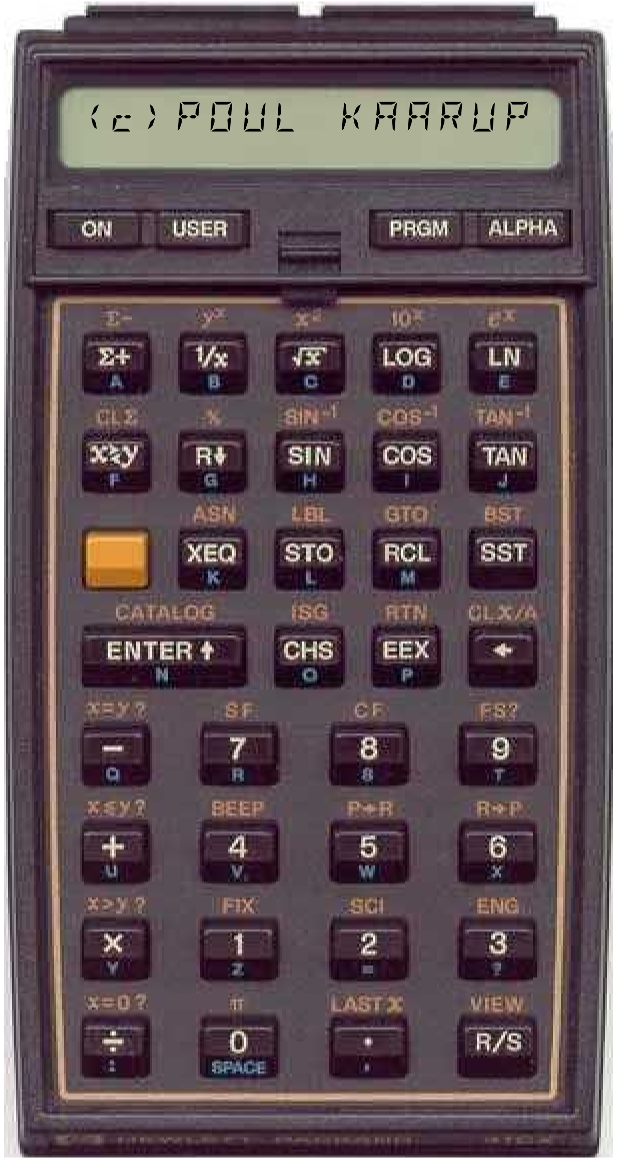
HP41CX

A PROGRAMMERS HANDBOOK



COMPILED
BY
POUL KAARUP

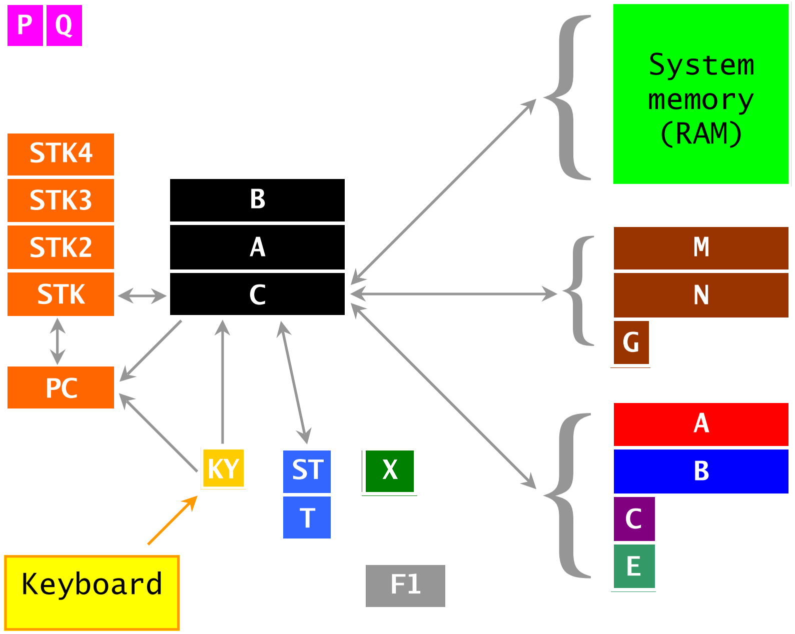
1992-2015

Index

3 Microprocessor registers
4 RAM structure & user flags
5 Status registers
6 Register structure
7 Flag register d
8 Key assign flag registers ⊦ & e
9 BLDSPEC print
10 Mcode class 0
11 -
12 Mcode class 1 & 2
13 Mcode class 3
14 Error messages
15 Interrupts & prompts
16 Partial key sequence
17 Card reader + tone generator
18 Wand
19 HP-IL interface
20 -
21 Printer
22 Timer
23 -
24 Display
25 -
26 Port dependent jumps
27 Entry points
28 -
29 -
30 -
31 -
32 -
33 -
34 -
35 Fat Addresses
36 ROM function names & DEBUG registers
37 Synthetic QRC 1
38 Synthetic QRC 2
39 Key code maps
40 -
41 -
42 -
43 Bankswitching & Module structure
44 ROM structure
45 GTO 00-14
46 GTO 15-99
47 XEQ
48 XROM# & END & XEQ/GTO IND
49 Alpha
50 HpCX41 mainframe functions
51 Module ID’s
52 XROM #’s
53 -
54 -
55 -
56 -
57 -
58 -
59 -
60 -
61 -
62 -
63 -
64 -
65 -
66 Buffer ID’s & Bytes pr. function
67 System Memory Map © Ángel Martin
68 -
69 -
70 -

HP 41 microprocessor

CPU register connections:



CPU:

C	56 bit	accumulator
A	56 bit	primary arithmetic register
B	56 bit	secondary arithmetic register
M	56 bit	storage register
N	56 bit	storage register
G	8 bit	storage register
PC	16 bit	program counter
STK	16 bit	4-level CPU return stack
KY	8 bit	keyboard buffer register
T	8 bit	beeper output register
ST	8 bit	CPU flag register (0-7)
X	6 bit	CPU flag register (9-13)
F1	14 bit	peripheral flag register
P	4 bit	pointer
Q	4 bit	pointer

DISPLAY:

A	48 bit	display register (bit 0-3 of each chr)
B	48 bit	display register (bit 4-7 of each chr)
C	12 bit	display register (bit 8 of each chr)
E	12 bit	display register (annunciators)

CPU FLAGS:

0	User flag (in the ST register)
1	User flag (in the ST register)
2	User flag (in the ST register)
3	User flag (in the ST register)
4	User flag (in the ST register)
5	User flag (in the ST register)
6	User flag (in the ST register)
7	User flag (in the ST register)
8	User flag
9	User flag
10	System flag (UserCode program pointer in ROM)
11	System flag (Stack lift enabled)
12	System flag (Program pointer in PRIVATE program)

RAM structure

3EF	Extended Memory #2	1007
301		769
2EF	Extended Memory #1	751
201		513
1FF	Top of Main Memory	511
	DATA register 00	
	Top of User programs	
	.END.	
	Free registers	
	I/O Buffer area	
	Alarm buffer area	
0C0	Key Assignments	192
0BF	Top of XF extended memory	191
040	Bottom of XF extended memory	64
	Nonexistent registers (void)	
00F		15
	Status registers	
000		0

User & system flags

🟥 set at turn on	🟦 cleared if printer is absent
🟩 cleared at turn on	🔷 matches flag 55 at turn on
🟪 system flags - not applicable	🟨 maintained by memory

00	🟨	general use
01	🟨	general use
02	🟨	general use
03	🟨	general use
04	🟨	general use
05	🟨	general use
06	🟨	general use
07	🟨	general use
08	🟨	general use
09	🟨	general use
10	🟨	general use
11	🟩	auto execute
12	🟩	double wide print
13	🟩	lower case print
14	🟩	overwrite card protection
15	🟩	IL-printer MAN / NORM
16	🟩	IL-printer TRACE
17	🟩	end of record
18	🟩	TINTR enable
19	🟩	general use
20	🟩	general use
21	🔷	printer enable
22	🟩	number entry
23	🟩	ALPHA entry
24	🟩	range error ignore
25	🟩	error ignore
26	🟥	audio enable
27	🟨	USER mode
28	🟨	decimal point
29	🟨	digit grouping
30	🟩🟪	CAT mode
31	🟪	timer MDY / DMY
32	🟪	IL manio
33	🟪	IL lock
34	🟪	ADRON / ADROFF
35	🟪	disable autostart
36	🟨🟪	digit number 8,9
37	🟨🟪	digit number 4,5,6,7
38	🟨🟪	digit number 2,3,6,7
39	🟨🟪	digit number 1,3,5,7,9
40	🟨🟪	display FIX / SCI
41	🟨🟪	display ENG /FIX-ENG
42	🟨🟪	trig mode DEG / GRAD
43	🟨🟪	trig mode RAD
44	🟩🟪	contiuuous ON
45	🟩🟪	system data entry
46	🟩🟪	partial key sequence
47	🟩🟪	SHIFT
48	🟩🟪	ALPHA
49	🟪	low BAT
50	🟩🟪	message
51	🟩🟪	SST
52	🟩🟪	PRGM mode
53	🟩🟪	I/O
54	🟩🟪	PSE
55	🟦🟪	Printer existence

Status registers

<table>
<tr><th></th><th colspan="2">6</th><th colspan="2">5</th><th colspan="2">4</th><th colspan="2">3</th><th colspan="2">2</th><th colspan="2">1</th><th colspan="2">0</th><th></th></tr>
<tr><td>e</td><td colspan="9">SHIFTED KEY ASSIGNMENTS</td><td colspan="2">PTEMP2</td><td colspan="3">PROGRAM LINE #</td><td>00F</td></tr>
<tr><td>d</td><td colspan="14">FLAG REGISTER</td><td>00E</td></tr>
<tr><td>c</td><td colspan="3">SREG ABS. ADR.</td><td colspan="2">printer use</td><td colspan="3">COLD START 169</td><td colspan="3">REG 00 ABS.ADR.</td><td colspan="3">.END. ABS.ADR.</td><td>00D</td></tr>
<tr><td>b</td><td colspan="2">3rd return</td><td colspan="4">2nd return</td><td colspan="4">1st return</td><td colspan="4">PRGM POINTER</td><td>00C</td></tr>
<tr><td>a</td><td colspan="4">6th return</td><td colspan="4">5th return</td><td colspan="4">4th return</td><td colspan="2">3rd return</td><td>00B</td></tr>
<tr><td>⊢</td><td colspan="9">UNSHIFTED KEY ASSIGNMENTS</td><td colspan="3">last instruction exec.</td><td colspan="2">PASN keycode</td><td>00A</td></tr>
<tr><td>Q</td><td colspan="14">TEMPORARY ALPHA SCRATCH</td><td>009</td></tr>
<tr><td>P</td><td colspan="8">ALPHA register (25-28)</td><td colspan="6">ALPHA register (22-24)</td><td>008</td></tr>
<tr><td>O</td><td colspan="14">ALPHA register (15-21)</td><td>007</td></tr>
<tr><td>N</td><td colspan="14">ALPHA register (8-14)</td><td>006</td></tr>
<tr><td>M</td><td colspan="14">ALPHA register (1-7)</td><td>005</td></tr>
<tr><td>L</td><td colspan="14">USER stack register L</td><td>004</td></tr>
<tr><td>X</td><td colspan="14">USER stack register X</td><td>003</td></tr>
<tr><td>Y</td><td colspan="14">USER stack register Y</td><td>002</td></tr>
<tr><td>Z</td><td colspan="14">USER stack register Z</td><td>001</td></tr>
<tr><td>T</td><td colspan="14">USER stack register T</td><td>000</td></tr>
<tr><td></td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td><td></td></tr>
</table>

RAM register structure

One RAM register is 56 bits:

55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

or 14 nybbles:

13	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**

or 7 bytes:

byte #6	**byte #5**	**byte #4**	**byte #3**	**byte #2**	**byte #1**	**byte #0**

Text register format:

1	0	char	char	char	char	char	char

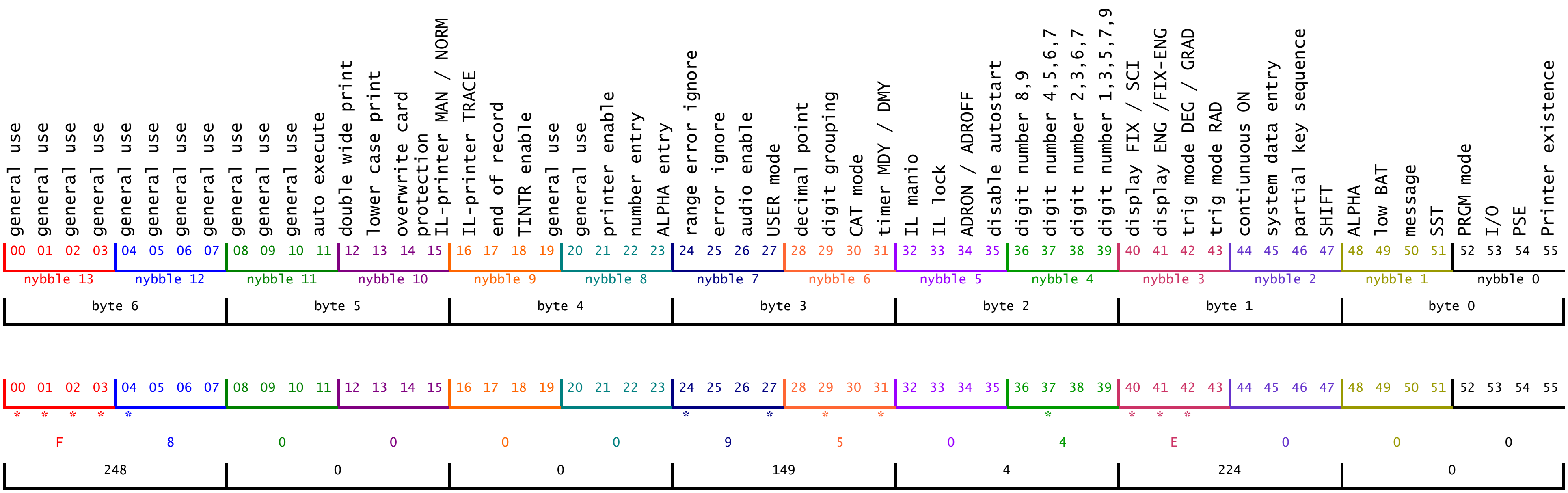
Numerical register format:

MS	Mantissa	XS	Exp.

<table>
<tr><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>MS</td><td colspan="10">M</td><td>XS</td><td colspan="2">XP</td></tr>
<tr><td colspan="7"></td><td colspan="4">ADR</td><td colspan="3">S&X</td></tr>
<tr><td colspan="9"></td><td colspan="2">KY</td><td colspan="3"></td></tr>
</table>

MS	13	sign of mantissa
M	3-12	mantissa
XS	2	sign of exponent
XP	0-1	exponent
S&X	0-2	sign of exponent and exponent
ADR	3-6	address field
KY	3-4	key buffer field

Flag register d



00	general use	*	nybble 13	F	byte 6	248
01	general use	*				
02	general use	*				
03	general use	*				
04	general use	*	nybble 12	8		
05	general use					
06	general use					
07	general use					
08	general use		nybble 11	0	byte 5	0
09	general use					
10	general use					
11	auto execute					
12	double wide print		nybble 10	0		
13	lower case print					
14	overwrite card protection					
15	IL-printer MAN / NORM					
16	IL-printer TRACE		nybble 9	0	byte 4	0
17	end of record					
18	TINTR enable					
19	general use					
20	general use		nybble 8	0		
21	printer enable					
22	number entry					
23	ALPHA entry					
24	range error ignore	*	nybble 7	9	byte 3	149
25	error ignore					
26	audio enable					
27	USER mode	*				
28	decimal point		nybble 6	5		
29	digit grouping	*				
30	CAT mode					
31	timer MDY / DMY	*				
32	IL manio		nybble 5	0	byte 2	4
33	IL lock					
34	ADRON / ADROFF					
35	disable autostart					
36	digit number 8,9		nybble 4	4		
37	digit number 4,5,6,7	*				
38	digit number 2,3,6,7					
39	digit number 1,3,5,7,9					
40	display FIX / SCI	*	nybble 3	E	byte 1	224
41	display ENG /FIX-ENG	*				
42	trig mode DEG / GRAD	*				
43	trig mode RAD					
44	continuous ON		nybble 2	0		
45	system data entry					
46	partial key sequence					
47	SHIFT					
48	ALPHA		nybble 1	0	byte 0	0
49	low BAT					
50	message					
51	SST					
52	PRGM mode		nybble 0	0		
53	I/O					
54	PSE					
55	Printer existence					

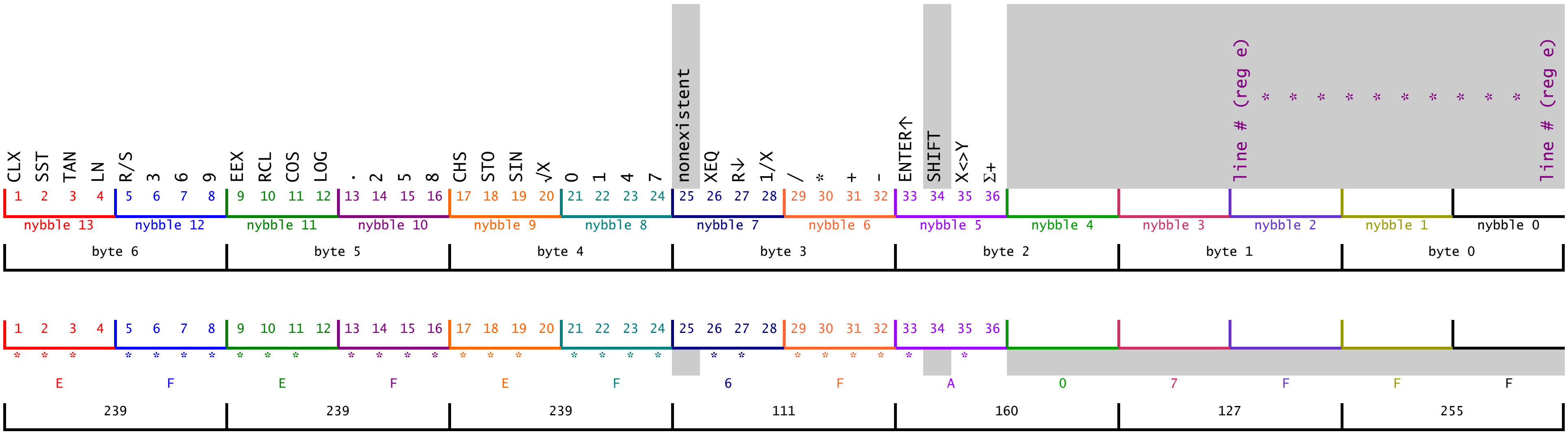
UserCode :

▩ ▩ ‾ ‾ ▩ ⊼ ▩ ‾	247,248,0,0,149,4,224,0
RCL M	144, 117
STO d	145, 126

Mcode :

04E	C=0 ALL
2DC	PT=13
3D0	LD@PT- F
210	LD@PT- 8
29C	PT=7
250	LD@PT- 9
150	LD@PT- 5
010	LD@PT- 0
110	LD@PT- 4
390	LD@PT- E
3A8	WRIT (14)d

Key assign flag registers ⊦ & e



nybble	bits	keys	flags set	value	byte
nybble 13	1 2 3 4	CLX SST TAN LN	* * *	E	byte 6: 239
nybble 12	5 6 7 8	R/S 3 6 9	* * * *	F	
nybble 11	9 10 11 12	EEX RCL COS LOG	* * *	E	byte 5: 239
nybble 10	13 14 15 16	. 2 5 8	* * * *	F	
nybble 9	17 18 19 20	CHS STO SIN √X	* * *	E	byte 4: 239
nybble 8	21 22 23 24	0 1 4 7	* * * *	F	
nybble 7	25 26 27 28	nonexistent XEQ R↓ 1/X	* *	6	byte 3: 111
nybble 6	29 30 31 32	/ * + -	* * * *	F	
nybble 5	33 34 35 36	ENTER↑ SHIFT X<>Y Σ+	* *	A	byte 2: 160
nybble 4		line # (reg e)		0	
nybble 3		* * * * * * * * * *		7	byte 1: 127
nybble 2				F	
nybble 1				F	byte 0: 255
nybble 0		line # (reg e)		F	

UserCode :

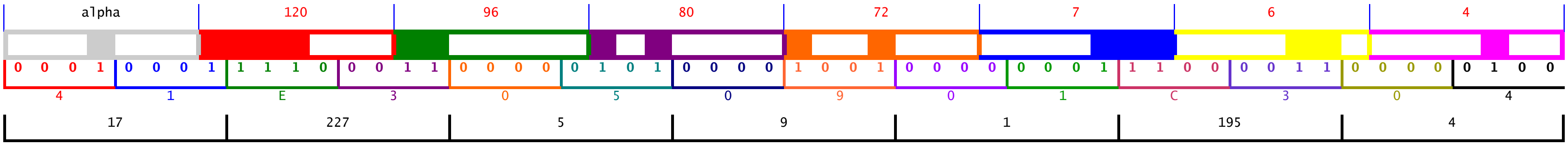
▩ ▩ ▩ ▩ ▩ ▩ ⊦ ▩	247,239,239,239,111,160,127,255
RCL M	144, 117
STO e	145, 127

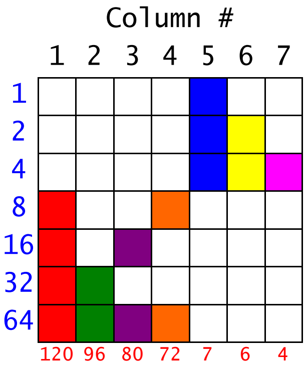
STO ⊦	145, 122

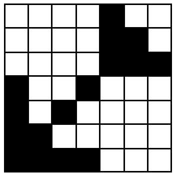
Mcode :

04E	C=0 ALL		
2DC	PT=13		
3D0	LD@PT- E		
210	LD@PT- F		
	LD@PT- E		
250	LD@PT- F		
150	LD@PT- E		
010	LD@PT- F		
110	LD@PT- 6		
390	LD@PT- F		
	LD@PT- A		
	LD@PT- 0		
	LD@PT- 7		
	LD@PT- F		
	LD@PT- F		
	LD@PT- F		
3E8	WRIT (15)e	2A8	WRIT (10)⊦

BLDSPEC print







UserCode :

▩▩▩⊼▩⊼▩⊼	247,17,227,5,9,1,195,4
RCL M	144, 117
ACSPEC	167, 68

PRBUF	167, 74

Mcode :

04E	C=0 ALL
2DC	PT=13
3D0	LD@PT- F
210	LD@PT- 8
29C	PT=7
250	LD@PT- 9
150	LD@PT- 5
010	LD@PT- 0
110	LD@PT- 4
390	LD@PT- E
3A8	WRIT (14)d

M-code (Class 0)

xx xxxx xx00

	T	Z	Y	X	L	M	N	O	P	Q	▩	a	b	c	d	e	
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
CLRF	384	304	204	004	044	084	144	284	104	244	0C4	184	344	2C4			clear CPU flag p
SETF	388	308	208	008	048	088	148	288	108	248	0C8	188	348	2C8			set CPU flag p
?FSET	38C	30C	20C	00C	04C	08C	14C	28C	10C	24C	0CC	18C	34C	2CC			set carry if CPU flag p is set
PT=	39C	31C	21C	01C	05C	09C	15C	29C	11C	25C	0DC	19C	35C	2DC			set pointer to digit p
?PT=	394	314	214	014	054	094	154	294	114	254	0D4	194	354	2D4			set carry if pointer is at p
LD@PT-	010	050	090	0D0	110	150	190	1D0	210	250	290	2D0	310	350	390	3D0	load C with p at pointer (PT=PT-1)
RCR		33C	23C	03C	07C	0BC	17C	2BC	13C	27C	0FC	1BC	37C	2FC			rotate C p digits right
WRIT	028	068	0A8	0E8	128	168	1A8	1E8	228	268	2A8	2E8	328	368	3A8	3E8	write C to selected user memory
READ		078	0B8	0F8	138	178	1B8	1F8	238	278	2B8	2F8	338	378	3B8	3F8	read selected user memory to C
HPIL=C	200	240	280	2C0	300	340	380	3C0									copy C[1;0] to HPIL register p
SELP	024	064	0A4	0E4	124	164	1A4	1E4	224	264	2A4	2E4	324	364	3A4	3E4	select peripheral to take control

M-code (Class 0) xx xxxx xx00

NOP	000	no operation
LDI S&X	130	load the 10-bit word at next address to C[2;0]
PT=PT+1	3DC	decrement pointer (if PT=0 then PT=13)
PT=PT-1	3D4	increment pointer (if PT=13 then PT=0)
SLCT P	0A0	select P as active pointer
SLCT Q	0E0	select Q as active pointer
?P=Q	120	set carry if P and Q have the same value
C=M ALL	198	copy M register to C
M=C ALL	158	copy C register to M
C<>M ALL	1D8	exchange C and M registers
C=N ALL	0B0	copy N register to C
N=C ALL	070	copy C register to N
C<>N ALL	0F0	exchange C and N registers
C=G @PT,+	098	copy G register to C register digits at PT and PT+1
G=C @PT,+	058	copy C register digits at PT and PT+1 to G register
C<>G @PT,+	0D8	exchange C register digits at PT and PT+1 with G register
C=ST XP	398	copy ST to C[1;0]
ST=C XP	358	copy C[1;0] to ST
C<>ST XP	3D8	exchange ST and C[1;0]
ST=0	3C4	clears ST (CPU flags 0-7)
ST=T	298	copy T to ST
T=ST	258	copy ST to T
ST<>T	2D8	exchange ST and T
READ DATA	038	copy active user memory register to C
WRIT DATA	2F0	copy C to active user memory register
FETCH S&X	330	fetches the word at system memory given in C[6;3] to C[2;0]
WRIT S&X	040	writes word in C[2;0] to system memory given in C[6;3]
RAM SLCT	270	select user memory register specified in C[2;0]
PRPH SLCT	3F0	select peripheral unit specified in C[2;0]
CLRKEY	3C8	clears the keydown flag (immediately set if key is down)
?KEY	3CC	set carry if keydown flag is set
C=KEY KY	220	copy key code from KY to C[4;3]
GOTO KEY	230	KEY register is written into lowets byte of PC

A=B=C=0	1A0	clear A, B and C registers
SETDEC	2A0	set CPU to decimal mode
SETHEX	260	set CPU to hexadecimal mode
C=C AND A	3B0	do logical AND on C and A registers and store result in C
C=C OR A	370	do logical OR on C and A registers and store result in C
RTN	3E0	return to address in STK
?C RTN	360	return to address in STK if carry is set
?NC RTN	3A0	return to address in STK if carry is clear
POP ADR	1B0	copy bottom STK to C[6;3] and STK drops
PUSH ADR	170	push STK up and store C[6;3] in bottom STK
GOTO ADR	1E0	jumps to address in C[6;3]
XQ>GO	020	pop return stack, turns the latest XQ into a GO
DSPOFF	2E0	turns display off
DSPTOG	320	toggles display on/off
?LOWBAT	160	set carry if battery is low
POWOFF	060	disp on: stop CPU, disp off: turn HP41 off, must be followed by NOP
?ALM	36C	set carry if an alarm from the timer has occured
?CRDR	32C	used with card reader
?EDAV	0AC	set carry if the diode of IR module is available
?FRAV	12C	set carry if a frame is available from HP-IL interface
?FRNS	26C	set carry if the frame transmitted (HPIL) not returns as sent
?IFCR	16C	set carry if HP-IL interface is ready
?ORAV	0EC	set carry if output register is available
?PBSY	3AC	set carry if HP82143 printer is busy
?SERV	2EC	set carry if any peripheral unit needs service
?SRQR	2AC	set carry if HPIL interface needs service
?WNDB	22C	set carry if there is data in the wand buffer
ENBANK1	100	enables primary bank
ENBANK2	180	enables secondary bank
ENBANK3	140	enables third bank
ENBANK4	1C0	enables forth bank
WPTOG	1F0	toggles write protection of HEPAX RAM specified in C[0]
ROM BLK	030	moves HEPAX ROM to memroy block specified in C[0]

M-code (Class 1) xx xxxx xx01

Jump to adress ABCD :

1.word : ccccdddd01
2.word : aaaabbbbtt

tt is **00**=?NC XQ
01=?C XQ
10=?NC GO
11=?C GO

M-code (Class 2) xx xxxx xx10

	ALL	M	S&X	MS	XS	@PT	PT←	P-Q	
A=0	00E	01A	006	01E	016	002	00A	012	Clear A
B=0	02E	03A	026	03E	036	022	02A	032	Clear B
C=0	04E	05A	046	05E	056	042	04A	052	Clear C
A=C	10E	11A	106	11E	116	102	10A	112	Copy C to A
C=B	0CE	0DA	0C6	0DE	0D6	0C2	0CA	0D2	Copy B to C
B=A	08E	09A	086	09E	096	082	08A	092	Copy B to A
A<>C	0AE	0BA	0A6	0BE	0B6	0A2	0AA	0B2	Exchange A and C
C<>B	0EE	0FA	0E6	0FE	0F6	0E2	0EA	0F2	Exchange C and B
A<>B	06E	07A	066	07E	076	062	06A	072	Exchange A and B
C=C+A	20E	21A	206	21E	216	202	20A	212	Add A to C
A=A+C	14E	15A	146	15E	156	142	14A	152	Add C to A
A=A+B	12E	13A	126	13E	136	122	12A	132	Add B to A
C=C+C	1EE	1FA	1E6	1FE	1F6	1E2	1EA	1F2	Shift C 1 bit left
C=A-C	24E	25A	246	25E	256	242	24A	252	Subtract C from A
A=A-C	1CE	1DA	1C6	1DE	1D6	1C2	1CA	1D2	Subtract C from A
A=A-B	18E	19A	186	19E	196	182	18A	192	Subtract B from A
C=C+1	22E	23A	226	23E	236	222	22A	232	Increment C
A=A+1	16E	17A	166	17E	176	162	16A	172	Increment A
C=C-1	26E	27A	266	27E	276	262	26A	272	Decrement C
A=A-1	1AE	1BA	1A6	1BE	1B6	1A2	1AA	1B2	Decrement A
?C≠0	2EE	2FA	2E6	2FE	2F6	2E2	2EA	2F2	Carry if C≠0
?A≠0	34E	35A	346	35E	356	342	34A	352	Carry if A≠0
?B≠0	2CE	2DA	2C6	2DE	2D6	2C2	2CA	2D2	Carry if B≠0
?A≠C	36E	37A	366	37E	376	362	36A	372	Carry if A≠C
?A<C	30E	31A	306	31E	316	302	30A	312	Carry if A<C
?A<B	32E	33A	326	33E	336	322	32A	332	Carry if A<B
RSHFC	3CE	3DA	3C6	3DE	3D6	3C2	3CA	3D2	Shift C 1 digit right
RSHFA	38E	39A	386	39E	396	382	38A	392	Shift A 1 digit
RSHFB	3AE	3BA	3A6	3BE	3B6	3A2	3AA	3B2	Shift B 1 digit right
LSHFA	3EE	3FA	3E6	3FE	3F6	3E2	3EA	3F2	Shift A 1 digit left
C=0-C	28E	29A	286	29E	296	282	28A	292	1’s complement
C=-C-1	2AE	2BA	2A6	2BE	2B6	2A2	2AA	2B2	2’s complement

M-code (Class 3)   xx xxxx xx11

	JNC+	JC+	JNC-	JC-
01	00B	00F	3FB	3FF
02	013	017	3F3	3F7
03	01B	01F	3EB	3EF
04	023	027	3E3	3E7
05	02B	02F	3DB	3DF
06	033	037	3D3	3D7
07	03B	03F	3CB	3CF
08	043	047	3C3	3C7
09	04B	04F	3BB	3BF
0A	053	057	3B3	3B7
0B	05B	05F	3AB	3AF
0C	063	067	3A3	3A7
0D	06B	06F	39B	39F
0E	073	077	393	397
0F	07B	07F	38B	38F
10	083	087	383	387
11	08B	08F	37B	37F
12	093	097	373	377
13	09B	09F	36B	36F
14	0A3	0A7	363	367
15	0AB	0AF	35B	35F
16	0B3	0B7	353	357
17	0BB	0BF	34B	34F
18	0C3	0C7	343	347
19	0CB	0CF	33B	33F
1A	0D3	0D7	333	337
1B	0DB	0DF	32B	32F
1C	0E3	0E7	323	327
1D	0EB	0EF	31B	31F
1E	0F3	0F7	313	317
1F	0FB	0FF	30B	30F
20	103	107	303	307
21	10B	10F	2FB	2FF
22	113	117	2F3	2F7
23	11B	11F	2EB	2EF
24	123	127	2E3	2E7
25	12B	12F	2DB	2DF
26	133	137	2D3	2D7
27	13B	13F	2CB	2CF
28	143	147	2C3	2C7
29	14B	14F	2BB	2BF
2A	153	157	2B3	2B7
2B	15B	15F	2AB	2AF
2C	163	167	2A3	2A7
2D	16B	16F	29B	29F
2E	173	177	293	297
2F	17B	17F	28B	28F
30	183	187	283	287
31	18B	18F	27B	27F
32	193	197	273	277
33	19B	19F	26B	26F
34	1A3	1A7	263	267
35	1AB	1AF	25B	25F
36	1B3	1B7	253	257
37	1BB	1BF	24B	24F
38	1C3	1C7	243	247
39	1CB	1CF	23B	23F
3A	1D3	1D7	233	237
3B	1DB	1DF	22B	22F
3C	1E3	1E7	223	227
3D	1EB	1EF	21B	21F
3E	1F3	1F7	213	217
3F	1FB	1FF	20B	20F
40			203	207

Jump distance **D** :

```
rddddddn11

r is 0=forward or 1=backwards
 dddddd is the distance
       n is 0=JC or 1=JNC
```

Custom error message

```
3A1  ?NC XQ
088  ->22E8   ERRSUB
3C1  ?NC XQ
0B0  ->2CF0   ENLCD
3BD  ?NC GO
01C  ->07EF   MESSL
00D  M
005  E
013  S
013  S
001  A
007  G
205  E
3DD  ?NC XQ
0AC  ->2BF7   DSPLFT
201  ?NC XQ
070  ->1C80   MSG105
3ED  ?NC XQ
08A  ->22FB   ERR110
```

Error message table

```
3D5  ?NC XQ
088  ->22F5   ERROR
XXX
```

018	ALPHA DATA
022	DATA ERROR
02D	MEMORY LOST
038	NONEXISTENT
03C	NULL
043	PRIVATE
04F	OUT OF RANGE
056	PACKING
05F	TRY AGAIN
062	YES
064	NO
067	RAM
06A	ROM

Error messages

MAINFRAME:

ALPHA DATA	14E2	PRIVATE	2184	RAM	2172
DATA ERROR	282D	OUT OF RANGE	00A2	ROM	21F0
NONEXISTENT	02E0	TRY AGAIN	2F72		

XF:

DIR EMPTY	3C7B	KEYCODE ERR	310A	REC TOO LONG	3A2D
DUP FL	30E8	NAME ERR	3565	*ROM 5 Bank 2:*	
END OF FL	3E7E	NO DRIVE	3675	CHKSUM ERR	5219
FL NOT FOUND	3D2D	NO ROOM	30CF	FL SIZE ERR	5391
FL TYPE ERR	3D22	NO SUCH ALM	394A	END OF REC	5651

TIME:

CAT EMPTY	3790	DATA ERROR Z		NO ROOM	
DATA ERROR X		ERROR = D*nn*		TIMER ALARM	5EAE
DATA ERROR Y		ERROR = Rnn			

PRINTER:

OUT OF PAPER	PRINT ERROR	PRINTER OFF

CARD READER:

CARD	MALFUNCTION	SIZE ERR
CARD ERR	MRG ERR	TYPE t tr nn
CHECKSUM ERR	NO ROOM	WORKING
LOW BAT	RDY kk OF nn	

HP-IL:

ADR ERR	72D7	FL TYPE ERR	788A	NO ROOM	79CE
DIR FULL	7726	MEDM ERR	742E	NO MEDM	
DRIVE ERR	7426	MEDM FULL	771F	READ ERR	7AA6
DUP FL NAME	7692	NAME ERR	72AE	SIZE ERR	7B9C
END OF FILE	773C	NO DRIVE	7CF4	TRANSMIT ERR	77E9
FL NOT FOUND	7855	NO KEYS	7DF1		
FL SECURED	7D10	NO PRINTER			

CMT-200:

BSIZE>1771	B6AA	NO ROOM	B6D7	REACH EOB	B1D1
NO BUFFER	B12C	NO TIMER	B1C2		

Using interrupts

xFF4	Interrupt checked during Pause
xFF5	Interrupt checked if system flag 53 set
xFF6	Interrupt checked on wakeup /not ON key
xFF7	Interrupt checked when HP41 is turned off
xFF8	Interrupt checked just before CPU stops
xFF9	Interrupt checked on wakeup /ON key
xFFA	Interrupt checked on MEMORY LOST

Before interrupt routine :

1. Save nybble 3 to 10 of register C

After interrupt routine :

1. Restore nybble 3 to 10 of register C
2. Have P as selected pointer
3. Load user flags 48 to 55 into register ST (Status set zero SS0)
4. Have chip 0 selected
5. CPU must be in HEX mode
6. GOTO address 27F3

```
25D  ?NC XQ
01C  ->0797   LDSSTO
260  SETHEX
0A0  SLCT P
3CD  ?NC GO
09E  ->27F3   RMCK10
```

This entry point selects chip 0 and places the user flags 48 to 55 in C and then in ST register

This entry point returns to continue interrupt polling

M-CODE prompts

	Leftmost **digit** of		
	1st chr	**2nd** chr	
Example	**op1**	**op2**	Type of prompt
SIN	0	-	none
COPY	1	0	Alpha input only (null input OK)
DEL	1	1	3 digits (4 by pressing EEX)
	1	2	Same as COPY (but no null input)
FIX	1	3	1 digit (+ indirect register/indirect stack)
STO	2	0	2 digits (+indirect reg/stack + stack) (*1)
ASTO	2	1	Same as STO (*2)
FS?C	2	2	2 digits (+ indirect register/stack)
	2	3	Same as FS?C
LBL	3	0	Non-null alpha or 2 digits
XEQ	3	1	Same as LBL (+ stack/indirect stack)
	3	2	Same as LBL
GTO	3	3	Same as XEQ (*3)

1 When the +/- keys are pressed at the double prompt, the functions defaults to the storeage arithmetic function*

2 The storage arithmetic part does not work

3 If the decimal key is pressed at the double prompt the function changes to GTO.

Numeric entry :	S&X of CPU A-register
Indirect numeric entry :	+080h in S&X of A-reg.
ALPHA entry :	Q status register

Partial key sequence

```
NEXT1     ?NC XQ 0E45            NEXT3     ?NC XQ 0E4B
NEXT2     ?NC XQ 0E48            NEXT      ?NC XQ 0E50
```

```
PTEMP1                                    in ST
PTEMP2                                    in G
Rightmost digit of Partial key code       in mantissa sign of A
Logical key code                          in nybble 1 & 2 of N
```

PTEMP1							
bit 7	**bit 6**	**bit 5**	**bit 4**	**bit 3**	**bit 2**	**bit 1**	**bit 0**
Decimal point	SHIFT	ALPHA	key from row 1 or 2	digit key	bit 0 of op1	bit 1 of op2	bit 0 of op2
					bit 8 of 1st chr of fcn name	bit 9 of 2nd chr of fcn name	bit 8 of 2nd chr of fcn name

PTEMP2							
bit 7	**bit 6**	**bit 5**	**bit 4**	**bit 3**	**bit 2**	**bit 1**	**bit 0**
not used	INDIRECT bit	XROM bit	INSERT bit	0	bit 0 of op1	bit 1 of op2	bit 0 of op2
	set if indirect operand	set if XROM fcn	set if PRGM mode & PRGMable		bit 8 of 1st chr of fcn name	bit 9 of 2nd chr of fcn name	bit 8 of 2nd chr of fcn name

```
115   ?NC XQ
038   0E45      NEXT1
07B   JNC +0F   if backarrow key !
04C   ?FSET 4   all other keys goes here
         :
         :
215   ?NC XQ
00C   0385      RSTSQ
```

Peripherals : The Card Reader

130	LDI S&X	
010	CON **16**	
270	RAMSLCT	de-selects user memory
130	LDI S&X	
0FC	CON **252**	
3F0	PRPH SLCT	selects the card reader

32C	?CDRD	sets carry if CardReader interrupt flag is set
038	READ DATA	reads data one record from CardReader to C[13;7] and C[6;0]
2F0	WRIT DATA	write on record from C[13;7] to CardReader buffer if card is present and motor is running
028	WRIT 0(T)	end write cycle
068	WRIT 1(Z)	start write cycle when motor is running
0A8	WRIT 2(Y)	end read cycle
0E8	WRIT 3(T)	set read mode
168	WRIT 5(M)	set CardReader interrupt flag if card is write protected
1E8	WRIT 7(O)	set CardReader interrupt flag if there is a card and the motor is running
2E8	WRIT 11(a)	read mode: clears interrupt flag if a record can be read write mode: sets interrupt flag if a record can be written to buffer
328	WRIT 12(b)	stop the motor
368	WRIT 13(c)	start the motor
3E8	WRIT 15(e)	set CardReader interrupt flag if the CardReader external flag is set

Peripherals : The wand

130	LDI S&X	
010	CON **16**	
270	RAMSLCT	de-selects user memory
130	LDI S&X	
0FE	CON **254**	
3F0	PRPH SLCT	selects the wand

22C	?WNDB	sets carry if data in wand buffer
038	READ DATA	reads data one byte to C[1;0]

Peripherals : The tone generator

258	T=ST	exchange the T value between 00h and FFh
298	ST=T	frequency is 158.000.000:(number of FF cycles + number of 00 cycles)
2D8	ST<>T	

Peripherals : The HP-IL interface

16C	?IFCR	set carry if interface ready
2AC	?SRQR	set carry if interface request service
12C	?FRAV	set carry if a frame is available from the loop
26C	?FRNS	set carry if frame not returned as it was sent
0EC	?ORAV	set carry if an output register is available

200	HPIL=C 0(T)	copies C[1;0] to HP-IL register 0
240	HPIL=C 1(Z)	copies C[1;0] to HP-IL register 1
280	HPIL=C 2(Y)	copies C[1;0] to HP-IL register 2
2C0	HPIL=C 3(X)	copies C[1;0] to HP-IL register 3
300	HPIL=C 4(L)	copies C[1;0] to HP-IL register 4
340	HPIL=C 5(M)	copies C[1;0] to HP-IL register 5
380	HPIL=C 6(N)	copies C[1;0] to HP-IL register 6
3C0	HPIL=C 7(O)	copies C[1;0] to HP-IL register 7

SELP r cccccccc01b	copies cccccccc to HP-IL register r

024	SELP 0(T)
064	SELP 1(Z)
0A4	SELP 2(Y)
0E4	SELP 3(X)

124	SELP 4(L)
164	SELP 5(M)
1A4	SELP 6(N)
1E4	SELP 7(O)

024	SELP 0(T)	
03A	C=PREG 0	
003	?PFSET	copies HP-IL register 0 TO C[1;0]
064	SELP 1(Z)	
07A	C=PREG 1	
043	?PFSET	copies HP-IL register 1 TO C[1;0]
0A4	SELP 2(Y)	
0BA	C=PREG 2	
083	?PFSET	copies HP-IL register 2 TO C[1;0]
0E4	SELP 3(X)	
0FA	C=PREG 3	
0C3	?PFSET	copies HP-IL register 3 TO C[1;0]

124	SELP 4(L)	
13A	C=PREG 4	
103	?PFSET	copies HP-IL register 4 TO C[1;0]
164	SELP 5(M)	
17A	C=PREG 5	
143	?PFSET	copies HP-IL register 5 TO C[1;0]
1A4	SELP 6(N)	
1BA	C=PREG 6	
183	?PFSET	copies HP-IL register 6 TO C[1;0]
1E4	SELP 7(O)	
1FA	C=PREG 7	
1C3	?PFSET	copies HP-IL register 7 TO C[1;0]

Peripherals : The HP-IL interface

	Register 0 : **status**
bit 0	master clear
bit 1	clear IFC received
bit 2	write:set local ready read: RFC received
bit 3	send receive request
bit 4	listener active
bit 5	talker active
bit 6	controller active
bit 7	system controller

	Register 1 : **control interrupt**	
bit 0	w:enable FI line	r:output register available
bit 1	w:not used	r:frame received not as sent
bit 2	w:not used	r:frame available
bit 3	w:not used	r:service request received
bit 4	w:not used	r:interface clear received
bit 5	w:output control bits	r:input control bits
bit 6	w:output control bits	r:input control bits
bit 7	w:output control bits	r:input control bits

	Register 2 : **data bits**	
bit 0	w:input data bits	r:output data bits
bit 1	w:input data bits	r:output data bits
bit 2	w:input data bits	r:output data bits
bit 3	w:input data bits	r:output data bits
bit 4	w:input data bits	r:output data bits
bit 5	w:input data bits	r:output data bits
bit 6	w:input data bits	r:output data bits
bit 7	w:input data bits	r:output data bits

	Register 3 : **parallel poll**
bit 0	parallel poll pesponse bit designation
bit 1	parallel poll pesponse bit designation
bit 2	parallel poll pesponse bit designation
bit 3	parallel poll polarity
bit 4	parallel poll enable
bit 5	parallel poll individual status
bit 6	automatic IDY sourcing in idle mode
bit 7	oscillator disable

	Register 4 : **lopp address**
bit 0	address bits
bit 1	address bits
bit 2	address bits
bit 3	address bits
bit 4	address bits
bit 5	scratch
bit 6	scratch
bit 7	scratch

	Registers 5,6 & 7 : **scratch**
bit 0	scratch
bit 1	scratch
bit 2	scratch
bit 3	scratch
bit 4	scratch
bit 5	scratch
bit 6	scratch
bit 7	scratch

Peripherals : The printer

3AC	?PBSY	set carry if the printer is busy
264	SELP 9	transfers control to the printer until an instruction with the rightmost bit set

while in control these instructions are understood by the printer:

003	BUSY?	set carry if the printer is busy
083	ERROR?	set carry if printer error
043	POWON?	set carry if printer is on
007	BUF=BUF+C	copy C[1;0] to printer buffer
03A	C=STATUS	copy printer status word to C[1;0] - next instruction must be 001h

printer status word:

bit	meaning
15-14	printer mode: 00 = MAN 10 = TRACE 01 = NORM
13	PRINT key is down
12	PAPER ADVANCE key is down
11	printer is OUT OF PAPER
10	printer battery is low
9	printer is idle (not printing)
8	printer buffer is empty
7	printer is using lower case (SF13)
6	printer is in graphics mode (column mode)
5	printer is using double width (SF12)
4	printer is printing right justified
3	last byte sent was End-Of-Line
2	printer error is occuring
1-0	always set

Peripherals : The timer

```
130   LDI  S&X
010   CON  16
270   RAMSLCT     de-selects user memory
130   LDI  S&X
0FB   CON  21
3F0   PRPH SLCT   selects the timer
```

status register bits:

bit	meaning
12	timer is in TEST A mode
11	timer is in TEST B mode
10	set if the interval timer is running
9	set if ALARM B is enabled (usually clear)
8	set if ALARM A is enabled (usually set)
7	set if CLOCK B is counting forwards
6	set if CLOCK A is counting forwards
5	set if timer chip supply voltage has been low
4	set if the interval timer has counted a whole interval
3	set if an overflow has occurred in CLOCK B
2	set if ALARM B register is the same as CLOCK B
1	set if an overflow has occurred in CLOCK A
0	set if ALARM A register is the same as CLOCK A

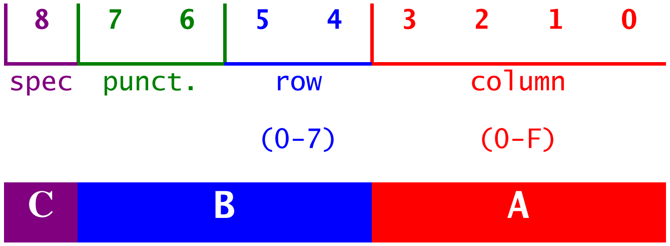
CLOCK REGISTER A	CURRENT TIME (1/100 SECONDS SINCE JAN 1 1900, DECIMALLY, RIGHT ALIGNED)
ALARM REGISTER A	TIME OF NEXT ALARM
SCRATCH REGISTER A	TIME WHEN CLOCK WAS LAST ADJUSTED
CLOCK REGISTER B	STOPWATCH TIME
ALARM REGISTER B	COLDSTART CONSTANT 09999999999000h
SCRATCH REGISTER B	BIT5 SET IF CLK24 FORMAT, BIT6 SET IF CLKTD FORMAT
A/B POINTER	
INTERVAL TIMER	
13 BIT STATUS REGISTER	
ACCURACY FACTOR REGISTER	

Peripherals : The timer

028	WRIT 0(T)	copy the C register to the active clock register
038	READ DATA	copy the active clock register to the C register
068	WRIT 1(Z)	as WRIT 0(T) but used after READ 1(Z), takes into account the time used since reading the time
078	READ 1(Z)	as READ DATA, used when correcting the time using T+X
0A8	WRIT 2(Y)	copy the C register to the active alarm register
0B8	READ 2(Y)	copy the active alarm registerto the C register
0E8	WRIT 3(X)	if pointer set to **A**: copy C[5;0] to the timer status register (can only clear bits) if pointer set to **B**: copy C[16;4] to timer accuracy factor register (0 to 99.9 , C[16] = sign)
0F8	READ 3(X)	if pointer set to **A**: copy status register to C[12;0] if pointer set to **B**: copy the accuracy factor register to C[16;4]
128	WRIT 4(L)	copy C register to active scratch register
138	READ 4(L)	copy active scratch register to C register
168	WRIT 5(M)	copy C[4;0] to interval timer and start timer (0.01 to 999.99 seconds) each time the interval period has passed the timer interrupt flag is set (CLOCK function uses this)
178	READ 5(M)	copy the value of the interval timer to C[4;0]
1E8	WRIT 7(O)	stop the interval timer
228	WRIT 8(P)	clear test mode (A or B)
268	WRIT 9(Q)	set test mode (A or B)
2A8	WRIT 10(⊦)	disable the active alarm, but does not clear it ALARM A is re-enabled at turn-off, timer alarms (negative stopwatch alarms) can’t be disabled
2E8	WRIT 11(a)	re-enable the disabled alarm
328	WRIT 12(b)	stop the clock in the active clock register (CLOCK A restarts as soon as CPU stops)
368	WRIT 13(c)	start the clock in the active clock register
3A8	WRIT 14(d)	set the A/B pointer to B
3E8	WRIT 15(e)	set the A/B pointer to A

Peripherals : The display

```
130  LDI  S&X
010  CON  16
270  RAMSLCT    de-selects user memory
130  LDI  S&X
0FD  CON  253
3F0  PRPH SLCT  selects display
```



Bits:

0-3 column of table

4-5 row of table

6-7 punctuation

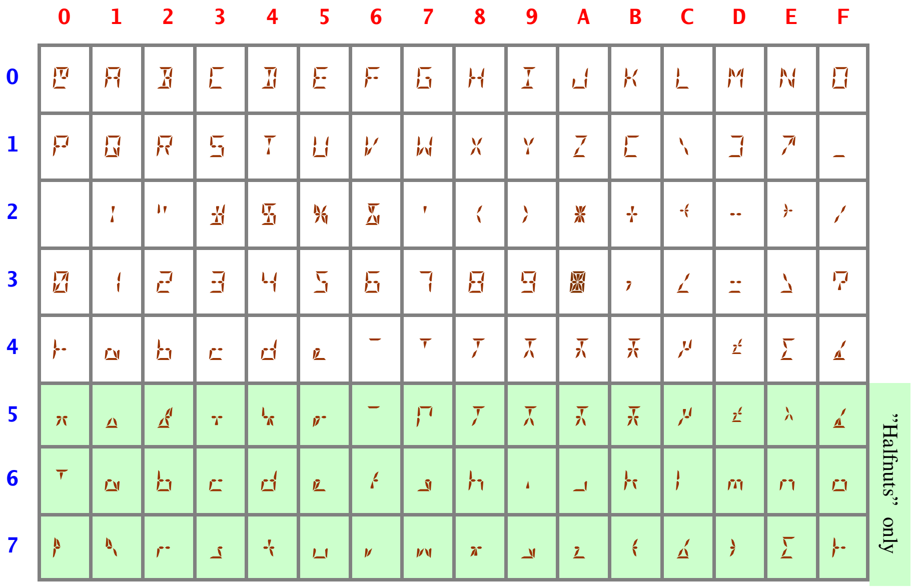
7	6	
0	0	none
0	1	.
1	0	:
1	1	,

8 special character

BAT	USER	G	RAD	SHIFT	0	1	2	3	4	PRGM	ALPHA
11	10	9	8	7	6	5	4	3	2	1	0

E

```
2F0  WRIT DATA  writes C S&X to annunciators
178  READ 5(M)  reads annunciators to C S&X
```



0
1
2
3
4
5
6
7
8
9
A
B
C
D
E
F
0
1
2
3
4
5
6
7
"Halfnuts" only

DISPLAY	PRINT	BYTE
e	e	7F
d	d	7E
c	c	7D
b	b	7C
a	a	7B
⊢	⊤	7A
Q	_	79
P	↑	78
O	]	77
N	\	76
M	[	75

DISPLAY	PRINT	BYTE
:		3A
,		3B
<		3C
=		3D
>		3E
?		3F

Peripherals : The display

Instruction	Hex	# of chrs	Rotation	Digs in C
READ DATA	038	12	←	12 (1 each)
WRIT 0 (T)	028	12	→	12 (1 each)
READ 1 (Z)	078	12	←	12 (1 each)
WRIT 1 (Z)	068	12	→	12 (1 each)
READ 2 (Y)	0B8	12	←	12 (1 each)
WRIT 2 (Y)	0A8	12	→	12 (1 each)
READ 3 (X)	0F8	6	←	12 (2 each)
WRIT 3 (X)	0E8	6	→	12 (2 each)
READ 4 (L)	138	4	←	12 (3 each)
WRIT 4 (L)	128	4	→	12 (3 each)
WRIT 5 (M)	168	6	←	12 (2 each)
READ 6 (N)	1B8	1	←	1
WRIT 6 (N)	1A8	4	←	12 (3 each)
READ 7 (O)	1F8	1	→	1
WRIT 7 (O)	1E8	1	→	1
READ 8 (P)	238	1	→	1
WRIT 8 (P)	228	1	→	1
READ 9 (Q)	278	1	→	1
WRIT 9 (Q)	268	1	→	1
READ 10 (⊦)	2B8	1	←	1
WRIT 10 (⊦)	2A8	1	←	1
READ 11 (a)	2F8	1	←	1
WRIT 11 (a)	2E8	1	←	1
READ 12 (b)	338	1	→	2
WRIT 12 (b)	328	1	→	2
READ 13 (c)	378	1	←	2
WRIT 13 (c)	368	1	←	2
READ 14 (d)	3B8	1	→	3
WRIT 14 (d)	3A8	1	→	3
READ 15 (e)	3F8	1	←	3
WRIT 15 (e)	3E8	1	←	3

C	B	A
		▨
		▨
	▨	
	▨	
▨		
▨		
	▨	▨
	▨	▨
▨	▨	▨
▨	▨	▨
	▨	▨
▨		
▨	▨	▨
		▨
		▨
	▨	
	▨	
▨		
▨		
		▨
		▨
	▨	
	▨	
	▨	▨
	▨	▨
	▨	▨
	▨	▨
▨	▨	▨
▨	▨	▨
▨	▨	▨
▨	▨	▨

A	bit	3 2 1 0	3 2 1 0	3 2 1 0	3 2 1 0	3 2 1 0	3 2 1 0	3 2 1 0	3 2 1 0	3 2 1 0	3 2 1 0	3 2 1 0	3 2 1 0
	char	12th	11th	10th	9th	8th	7th	6th	5th	4th	3rd	2nd	1st
B	bit	7 6 5 4	7 6 5 4	7 6 5 4	7 6 5 4	7 6 5 4	7 6 5 4	7 6 5 4	7 6 5 4	7 6 5 4	7 6 5 4	7 6 5 4	7 6 5 4
	char	12th	11th	10th	9th	8th	7th	6th	5th	4th	3rd	2nd	1st
C	bit	8	8	8	8	8	8	8	8	8	8	8	8
	char	12	11	10	9	8	7	6	5	4	3	2	1
E	ann.	B	U	G	R	S	0	1	2	3	4	P	A
	bit	12	11	10	9	8	7	6	5	4	3	2	1

Port dependent jumps

341 08C qXX	?NC XQ ->23D0	PORT DEP: GO ->pQXX	1st quarter p000-p3FF
365 08C qXX	?NC XQ ->23D9	PORT DEP: GO ->pQXX	2nd quarter p400-p7FF
389 08C qXX	?NC XQ ->23E2	PORT DEP: GO ->pQXX	3rd quarter p800-pBFF
3AD 08C qXX	?NC XQ ->23EB	PORT DEP: GO ->pQXX	4th quarter pC00-pFFF
369 03C qXX	?NC XQ ->0FD9	PORT DEP: GO ->pQXX	same quarter

349 08C qXX	?NC XQ ->23D2	PORT DEP: XQ ->pQXX	1st quarter p000-p3FF
36D 08C qXX	?NC XQ ->23DB	PORT DEP: XQ ->pQXX	2nd quarter p400-p7FF
391 08C qXX	?NC XQ ->23E4	PORT DEP: XQ ->pQXX	3rd quarter p800-pBFF
3B5 08C qXX	?NC XQ ->23ED	PORT DEP: XQ ->pQXX	4th quarter pC00-pFFF
379 03C qXX	?NC XQ ->0FDD	PORT DEP: XQ ->pQXX	same quarter

NB! Musts be HEX mode. Uses 2 places in RTN-stack

HP41CX ENTRY POINTS

ENTRY	ADR	MCODE

A

ABS	1076	
ABTS10	0D16	
ABTSEQ	0D12	149 034
ACOS	107D	
AD1_10	1809	
AD2_10	1807	01D 060
AD2_13	180C	
ADD1	1CE0	
ADD2	1CE3	
ADDONE	1800	001 060
ADRFCH	0004	
ADVNCE	114D	
AFORMT	0628	
AGTO	1085	
AJ2	0DD4	
AJ3	0DD0	
ALCL00	06C9	
ALLOK	02CD	
ALPDEF	03AE	
ANN_14	075B	
ANNOUT	075C	
AOFF	1345	
AON	133C	
AOUT15	2C2B	
APHST_	2E62	
APND_	1FF3	
APND10	1FF5	
APNDDG	1FFA	
APNDNW	2D14	
APPEND	2D0E	
ARCL	108C	
ARGOUT	2C10	
ASCLCA	2C5E	
ASCLCD	2C5D	
ASHF	1092	
ASIN	1098	
ASN	109E	
ASN15	27C2	
ASN20	27CC	
ASRCH	26C5	
ASTO	10A4	
ATAN	10AA	
AVAIL	28C4	
AVAILA	28C7	
AVIEW	10B2	
AXEQ	10B5	

B

BAKAPH	09E3	
BAKDE	09A5	295 024
BCDBIN	02E3	38D 008
BEEP	10BB	
BIGBRC	004F	
BKROM2	2A91	
BLANK	05B7	
BLINK	0899	265 020
BLINK1	0899	
BRT100	1D80	
BRT140	1DEC	
BRT160	1DA8	
BRT200	1E0F	
BRT290	1DAC	
BRTS10	1D6B	
BST	10C2	
BSTCAT	0BBA	
BSTE	290B	
BSTE2	2AF2	
BSTEP	28DE	
BSTEPA	28EB	

C

CAINC	00D7	35D 000
CALDSP	29C3	
CAT	10C8	
CAT1	0BC3	
CAT2	0B53	
CAT3	1383	
CF	10CC	
CHK_NO_S	14D8	
CHK_NO_S1	14D4	
CHK_NO_S2	14D9	
CHKAD4	1686	
CHKADR	166E	
CHKFUL	05BA	
CHKRPC	0222	
CHRLCD	05B9	
CHS	123A	
CHSA	1CDA	
CHSA1	1CDC	
CLA	10D1	345 040
CLCTMG	03C9	
CLDSP	10E0	381 040
CLLCDE	2CF0	
CLP	10E7	
CLR	1733	
CLREG	10ED	
CLRLCD	2CF6	
CLRPGM	228C	
CLRREG	2155	
CLRSB2	0C00	
CLRSB3	0C02	
CLSIG	10F3	
CLST	10F9	
CLX	1101	
CNTLOP	0B9D	
COLDST	0232	
COPY	1109	
COS	127C	
CPGM10	067F	1FD 018

HP41CX ENTRY POINTS

ENTRY	ADR	MCODE

CPGMHD	067B	

D

D_R	110E	
DAT106	2D4C	
DAT231	2D77	
DAT260	2D94	
DAT280	2D98	
DAT300	2D9B	
DAT320	2DA2	
DAT400	2E05	
DAT500	2E10	
DATENT	2D2C	
DATOFF	0390	
DCPL00	2EC3	
DCPLRT	2F0B	
DCRT10	2F0D	
DEC	132B	
DECAD	29C7	
DECADA	29CA	
DECMPL	2EC2	
DEEXP	088C	
DEG	1114	
DEGDO	172A	
DEL	1124	
DELETE	1127	
DELLIN	2306	
DELNNN	22A8	
DEROVF	08EB	
DEROW	04AD	
DERUN	08AD	
DERW00	04B2	
DF060	0587	
DF150	0482	
DF160	0485	
DF200	04E7	
DFILLF	0563	
DFKBCK	0559	
DFRST8	0562	
DFRST9	0561	
DGENS8	0836	0D9 020
DIGENT	0837	0DD 020
DIGST_	08B2	2C9 020
DIV110	18A5	
DIV120	18AF	
DIV15	18A9	
DIVIDE	106F	
DOSKP	1631	
DOSRC1	24E3	
DOSRCH	24E4	
DROPST	00E4	
DROWSY	0160	
DRSY05	0161	
DRSY25	0173	
DRSY50	0190	
DRSY51	0194	
DSE	112D	
DSPCA	0B35	
DSPCRG	0B26	
DSPLN_	0FC7	
DSWKUP	01AD	
DTOR	1981	205 064
DV1_10	189A	
DV2_10	1898	261 060
DV2_13	189D	

E

E_TO_X	1147	
E_TO_X_MIN	1163	
ENCP00	0952	149 024
END	1132	
END2	03B6	
END3	03BE	
ENG	1135	
ENLCD	07F6	3D9 01C
ENTER	113E	
ERR0	18C3	
ERR110	22FB	
ERR120	22FF	
ERRAD	14E2	
ERRDE	282D	
ERRIGN	00BB	
ERRNE	02E0	
ERROF	00A2	
ERROR	22F5	
ERRPR	2184	
ERRRAM	2172	
ERRSUB	22E8	
ERRTA	2F17	
EXP10	1A0A	029 068
EXP13	1A0D	
EXP400	1A21	
EXP500	1A61	
EXP710	1A4C	
EXP720	1A50	
EXSCR	192A	

F

FACT	1154	
FC	115A	
FC_C	116B	
FCNTBL	1400	
FDIG20	0E3D	
FDIGIT	0E2F	
FILLXL	00EA	
FIND_NO_1	1775	
FIX	1171	
FIX57	0AC3	
FIXEND	2918	
FLGANN	1651	
FLINK	2928	

HP41CX ENTRY POINTS

ENTRY	ADR	MCODE

FLINKA	2927	
FLINKM	2929	
FLINKP	2925	
FNDEND	1730	
FORMAT	0A7B	
FRAC	117C	
FS	1182	
FS_C	1188	
FSTIN	14C2	

G

GCP112	2BB5	
GCPK04	2BBC	
GCPK05	2BBE	
GCPKC	2B80	
GCPKC0	2B89	
GENLNK	239A	
GENNUM	05E8	3A1 014
GETLIN	1419	
GETN	1CEA	
GETPC	2950	
GETPCA	2952	
GETX	1CEF	
GETXSQ	1CEE	
GETXY	1CEB	
GETY	1CED	
GETYSQ	1CEC	
GOL0	23D0	
GOL1	23D9	
GOL2	23E2	
GOL3	23EB	
GOLNGH	0FD9	
GOLONG	0FDA	
GOSUB	0FDE	
GOSUB0	23D2	
GOSUB1	23DB	
GOSUB2	23E4	
GOSUB3	23ED	
GOSUBH	0FDD	
GOTINT	02F8	3E1 008
GRAD	111A	
GSB000	23FA	
GSB256	23FA	
GSB512	23FA	
GSB768	23FA	
GSUBS1	23C9	
GT3DBT	0FEB	
GTACOD	1FDB	
GTAI40	0341	
GTAINC	0304	
GTBYT	29B0	
GTBYTA	29BB	
GTBYTO	29B2	
GTCNTR	0B8D	
GTFEN1	20EB	
GTFEND	20E8	
GTLINK	224E	
GTLNKA	2247	
GTO	1191	
GTO_5	29AA	
GTOL	118C	
GTONN	2959	
GTRMAD	0800	
GTSRCH	24DF	

H

H_HMS	1199	
HMS_H	1193	
HMS_MINUS	1045	
HMS_PLUS	1032	
HMSDV	19E5	
HMSMP	19E7	

I

IN3B	2A65	
INBCHS	2E0A	
INBYT	29E6	
INBYT0	29E3	
INBYT1	29EA	
INBYTC	29E4	
INBYTJ	2E0C	
INBYTP	29E5	
INCAD	29CF	
INCAD2	29D3	
INCADA	29D6	
INCADP	29D1	
INCGT2	0286	
IND	0DB2	
IND21	0DC4	
INEX	2A4A	
INLIN	2876	
INLIN2	29F6	
INPTDG	08A0	
INSHRT	2A17	
INSLIN	29F4	
INSSUB	23B2	
INSTR	2A73	
INT	1177	
INTARG	07E1	
INTFRC	193B	0ED 064
INTINT	02FB	3ED 008
INTXC	2A7D	
IORUN	27E4	
ISG	119E	

K

KEYOP	068A	
KYOPCK	0693	

L

LASTX	1228	
LBL	11A4	
LD90	1995	

HP41CX ENTRY POINTS

ENTRY	ADR	MCODE
LDD_P_	0B1D	
LDDP10	0B1E	
LDSST0	0797	
LEFTJ	2BF7	
LINN1A	2A93	
LINNM1	2A90	
LINNUM	2A8B	
LN	11A6	
LN1_PLUS_X	1220	
LN10	1B45	115 06C
LN560	1BD3	
LNAP	1A8A	
LNC10	1AAE	
LNC10_	1AAD	
LNC20	1ABD	
LNSUB	19F9	
LNSUB_MINUS	19F8	
LOAD3	14FA	
LOG	11AC	
LSWKUP	0180	

M

ENTRY	ADR	MCODE
MASK	2C88	
MEAN	11B9	
MEMCHK	0205	
MEMLFT	05A1	
MESSL	07EF	3BD 01C
MIDDIG	0DE0	
MINUS	1054	
MOD	104F	
MOD10	195C	171 064
MODE	134D	
MODE1	134F	
MOVREG	215C	
MP1_10	184F	
MP2_10	184D	135 060
MP2_13	1852	
MPY150	1865	
MSG	1C6B	
MSG105	1C80	201 070
MSG110	1C86	219 070
MSGA	1C6C	1B1 070
MSGAD	1C18	
MSGDE	1C22	
MSGDLY	037C	
MSGE	1C71	
MSGML	1C2D	
MSGNE	1C38	
MSGNL	1C3C	
MSGNO	1C64	
MSGOF	1C4F	
MSGPR	1C43	
MSGRAM	1C67	
MSGROM	1C6A	
MSGTA	1C5F	
MSGWR	1C56	
MSGX	1C75	
MSGYES	1C62	
MULTIPLY	105C	

N

ENTRY	ADR	MCODE
NAM40	0F34	
NAM44_	0F7D	
NAME20	0EE6	
NAME21	0EE9	
NAME33	0EEF	
NAME37	0F09	
NAME4A	0FA4	
NAME4D	0FAC	
NAMEA	0ED9	
NBYTA0	2D04	
NBYTAB	2D06	
NEXT	0E50	141 038
NEXT1	0E45	115 038
NEXT2	0E48	121 038
NEXT3	0E4B	12D 038
NFRC	00F1	3C5 000
NFRENT	00C4	
NFRFST	00F7	3DD 000
NFRKB	00C7	31D 000
NFRKB1	00C6	
NFRNC	00A5	295 000
NFRNIO	0106	
NFRPR	00EE	3B9 000
NFRPU	00F0	3C1 000
NFRSIG	00C2	
NFRST_PLUS	0BEE	
NFRX	00CC	331 000
NFRXY	00DA	369 000
NLT000	0E91	
NLT020	0EA0	
NLT040	0EAA	
NM44_5	0F7E	
NOPRT	015B	
NOREG9	095E	179 024
NOSKP	1619	065 058
NOTFIX	0ADD	
NRM10	1870	
NRM11	1871	
NRM12	1872	
NRM13	1884	
NROOM3	28C2	
NULT_	0E65	
NULT_3	0E7C	
NULT_5	0E8F	
NULTST	0EC6	
NWGOOS	07D4	
NXBYT3	29B7	
NXBYTA	29B9	
NXBYT0	2D0B	

HP41CX ENTRY POINTS

ENTRY	ADR	MCODE

NXL1B	2B23	
NXL3B2	2B63	
NXLCHN	2B49	
NXLDEL	2AFD	
NXLIN	2B14	
NXLIN3	2B5F	
NXLINA	2B1F	
NXLSST	2AF7	
NXLTX	2B77	
NXTBYT	2D07	

O

OCT	1330	
OFF	11C8	321 044
OFFSHF	0750	
OFSHFT	0749	
ONE_BY_X	11D6	
ONE_BY_X10	188B	
ONE_BY_X13	188E	
OPROMT	2E4C	
OUTLCD	2C80	
OUTROM	2FEE	
OVFL10	1429	

P

P_R	11DC	
P10RTN	02AC	
P6RTN	1670	
PACH10	03EC	

PACH11	03F5	
PACH12	03FC	
PACH4	03E2	
PACK	11E7	
PACKE	2002	
PACKN	2000	
PAK200	2055	
PAKEND	20AC	
PAKSPC	20F2	
PAR111	0CED	
PAR112	0CF5	
PARA06	0D22	
PARA60	0D35	
PARA61	0D37	
PARA75	0D49	
PARB40	0D99	
PARS05	0C34	
PARS56	0C93	
PARS75	0CCD	
PARSDE	0C90	
PARSE	0C05	
PARSEB	0D6D	
PATCH1	21DC	
PATCH2	21E1	
PATCH3	21EE	
PATCH5	21F3	
PATCH6	1C06	

PATCH9	1C03	
PCKDUR	16FC	
PCT	1061	
PCTCH	11EC	
PGMAON	0956	
PI	1242	
PI_BY_2	199A	
PKIOAS	2114	051 084
PLUS	104A	
PMUL	1BE9	
POWER_OF_TEN	12CA	
PR10RT	0372	
PR14RT	1365	
PR15RT	22DF	
PR3RT	0EDD	
PROMF1	05CB	
PROMF2	05D3	
PROMFC	05C7	
PROMPT	1209	
PSE	11FC	
PSESTP	03AC	
PTBYTA	2323	
PTBYTM	2921	
PTBYTP	2328	
PTLINK	231A	
PTLNKA	231B	
PTLNKB	2321	

PUTPC	2337	
PUTPCA	2339	
PUTPCD	232C	
PUTPCF	2331	
PUTPCL	2AF3	
PUTPCX	232F	
PUTREG	215E	

Q

QUTCAT	03D5	

R

R_D	120E	
R_P	11C0	
R_SCAT	0BB7	
R_SUB	14ED	
RAD	111F	
RAK06	0C7F	
RAK60	06FA	
RAK70	070A	
RCL	122E	
RCSCR	1934	
RCSCR_	1932	
RDN	1252	
RDNSUB	14E9	
REGLFT	059A	
RFDS55	0949	
RG9LCD	08EF	3BD 020
RMCK05	27EC	

HP41CX ENTRY POINTS

ENTRY	ADR	MCODE
RMCK10	27F3	
RMCK15	27F4	
RND	1257	
ROLBAK	2E42	
ROLLUP	1260	
ROMCHK	27E6	
ROMH05	066C	1B1 018
ROMH35	0678	
ROMHED	066A	
ROUND	0A35	
ROW0	2766	
ROW10	02A6	
ROW11	25AD	
ROW12	2743	
ROW120	0519	
ROW933	0467	
ROW940	0487	
RST05	009B	26D 000
RSTANN	0759	
RSTKB	0098	261 000
RSTMS0	038E	239 00C
RSTMS1	0390	
RSTMSC	0392	
RSTSEQ	0384	
RSTSQ	0385	215 00C
RSTST	08A7	
RTJLBL	14C9	
RTN	125C	
RTN30	272F	
RTOD	198C	231 064
RUN	07C2	
RUN_STOP	1218	
RUNING	0108	
RUNNK	011D	
RW0110	04E9	
RW0141	04F1	

S

ENTRY	ADR	MCODE
SAR021	2640	
SAR022	2641	
SAROM	260D	
SAVR10	27D5	
SAVRC	27DF	
SAVRTN	27D3	
SCI	1265	
SCROL0	2CDE	
SCROLL	2CDC	
SD	1D10	
SEARC1	2434	
SEARCH	2433	
SEPXY	14D2	
SERR	24E8	
SETQ_P	0B15	
SETSST	17F9	
SF	1269	
SGT019	25C9	
SHF10	186D	
SHF40	186C	
SHIFT	1348	
SIGMA	1C88	
SIGMA_MINUS	1271	
SIGMA_PLUS	126D	
SIGN	1337	
SIGREG	1277	
SIN	1288	
SINFR	1947	
SINFRA	194A	
SIZE	1292	
SIZSUB	1797	
SKP	162E	0B9 058
SKPDEL	2349	
SKPLIN	2AF9	
SNR10	243F	
SNR12	2441	
SNROM	2400	
SQR10	18BE	2F9 060
SQR13	18C1	
SQRT	1298	
SRBMAP	2FA5	
SST	129E	
SSTBST	22DD	
SSTCAT	0BB4	
STATCK	1CC8	
STAYON	12A3	
STBT10	2EA3	
STBT30	2FE0	
STBT31	2FE5	
STDEV	11B2	
STFLGS	16A7	
STK	0DF3	
STK00	0DFA	
STK04	0E00	
STMSGF	037E	
STO	10DA	
STO_DIVIDE	12C1	
STO_MINUS	12B9	
STO_MULTIPLY	12A8	
STO_PLUS	12B0	
STOLCC	2E5B	
STOP	1215	
STOPS	03A7	
STOPSB	03A9	
STORFC	07E8	
STOST0	013B	
STSCR	1922	
STSCR_	1920	
SUBONE	1802	009 060
SUMCHK	1667	
SUMCK2	1669	

HP41CX ENTRY POINTS

ENTRY	ADR	MCODE

T

ENTRY	ADR	MCODE
TAN	1282	
TBITMA	2F7F	
TBITMP	2F81	
TEN_TO_X	1BF8	
TEXT	2CAF	
TGSHF1	1FE7	
TODEC	1FB3	
TOGSHF	1FE5	
TONE	12D0	
TONE7	1716	
TONE7X	16DB	
TONEB	16DD	
TONSTF	0054	
TOOCT	1F79	
TOPOL	1D49	
TOREC	1E75	
TRC10	19A1	285 064
TRC30	1E38	
TRCS10	1E57	
TRG100	1E78	
TRG240	1ED1	
TRG430	1F5B	
TRGSET	21D4	
TSTMAP	14A1	
TXRW10	04F6	
TXTLB1	2FC6	
TXTLBL	2FC7	
TXTROM	04F5	
TXTROW	04F2	
TXTSTR	04F6	

U

ENTRY	ADR	MCODE
UPLINK	2235	

V

ENTRY	ADR	MCODE
VIEW	12D6	

W

ENTRY	ADR	MCODE
WKUP10	0184	
WKUP21	01A7	
WKUP25	01BA	
WKUP70	01F5	
WKUP80	01FF	

X

ENTRY	ADR	MCODE
X_BY_Y13	1893	
X_EQ_0	130E	
X_EQ_Y	1314	
X_GT_0	131A	
X_GT_Y	1320	
X_LE_0	12EF	
X_LE_Y	12F6	
X_LT_0	12E8	
X_LT_Y	1308	
X_NE_0	12DC	
X_NE_Y	12E2	
X_TO_2	106B	
X_XCHNG	124C	
X_XCHNG_Y	12FC	
XARCL	1696	
XASHF	1748	
XASN	276A	
XASTO	175C	
XAVIEW	0364	
XBAR	1CFE	
XBAR_	1D07	
XBEEP	16D1	
XBST	2250	
XCAT	0B80	
XCF	164D	
XCLSIG	14B0	
XCLX1	1102	
XCOPY	2165	
XCUTB1	0091	
XCUTE	015B	
XCUTEB	0090	
XDEG	171C	
XDELET	22AF	
XDSE	159F	
XECROM	2F4A	
XEND	2728	
XEQ	1328	
XEQC01	24EA	
XFS	1645	
XFT100	18EC	
XGA00	248D	
XGI	24C7	
XGI07	24DA	
XGI57	24C1	
XGNN10	2512	
XGNN12	2514	
XGNN40	255D	
XGOIND	1323	
XGRAD	1726	
XGTO	2505	
XISG	15A0	
XLN1_PLUS_X	1B73	
XMSGPR	056D	
XNNROW	0026	
XPACK0	2000	001 080
XPRMPT	03A0	
XR_S	079D	
XRAD	1722	
XRDN	14BD	
XRND	0A2F	
XROLLUP	14E5	
XROM	2FAF	
XROMNF	2F6C	
XROW1	0074	
XRS45	07BE	
XRTN	2703	

HP41CX ENTRY POINTS

ENTRY	ADR	MCODE

XSCI	16C0	
XSF	164A	
XSGREG	1659	
XSIGN	0FF4	
XSIZE	1795	
XSST	2260	
XSTYON	1411	
XTOHRS	19B2	
XTONE	16DE	
XVIEW	036F	
XX_EQ_0	1606	
XX_EQ_Y	1614	
XX_GT_0	15F1	
XX_GT_Y	15F8	
XX_LE_0	160D	
XX_LE_0A	1609	
XX_LE_Y	1601	
XX_LT_0	15FA	
XX_LT_Y	15EF	
XX_NE_0	1611	
XX_NE_Y	1629	
XXEQ	252F	
XY_TO_X	1B11	

Y

Y_MINUS_X	1421	
Y_TO_X	102A	

1

1/X10	188B	22D 060
10↗X	1BF8	3E1 06C

FAT-addresses

ADR1 | XROM#
ADR2 | FCN#

000	XROM ID	020	15	040	31	060	47	080	63
001	# FCN's	021	16	041	32	061	48	081	64
002	00	022	16	042	32	062	48	082	NOP
003	01	023	17	043	33	063	49	083	NOP
004	01	024	17	044	33	064	49		
005	02	025	18	045	34	065	50		
006	02	026	18	046	34	066	50		
007	03	027	19	047	35	067	51		
008	03	028	19	048	35	068	51		
009	04	029	20	049	36	069	52		
00A	04	02A	20	04A	36	06A	52		
00B	05	02B	21	04B	37	06B	53		
00C	05	02C	21	04C	37	06C	53		
00D	06	02D	22	04D	38	06D	54		
00E	06	02E	22	04E	38	06E	54		
00F	07	02F	23	04F	39	06F	55		
010	07	030	23	050	39	070	55		
011	08	031	24	051	40	071	56		
012	08	032	24	052	40	072	56		
013	09	033	25	053	41	073	57		
014	09	034	25	054	41	074	57		
015	0A	035	26	055	42	075	58		
016	10	036	26	056	42	076	58		
017	0B	037	27	057	43	077	59		
018	11	038	27	058	43	078	59		
019	0C	039	28	059	44	079	60		
01A	12	03A	28	05A	44	07A	60		
01B	0D	03B	29	05B	45	07B	61		
01C	13	03C	29	05C	45	07C	61		
01D	0E	03D	30	05D	46	07D	62		
01E	14	03E	30	05E	46	07E	62		
01F	0F	03F	31	05F	47	07F	63		

ROM function names

Address	Code	Description
x00E	004	**M-code**
x00F	002	Function address x402
	:	
x39E	094	T
x39F	013	S
x400	005	E
x401	014	T
x402	08C	?FSET 5 ← **entry point**
	:	

x39E	094	T	**+080**	last character of name
x39F	013	S		
x400	**o**05	E	**o=op2**	used for
x401	**o**14	T	**o=op1**	prompting
x402	000	**NOP =**		**NOT PRGMable**
x403	000	**NOP =**		**NOT NULLable**

Address	Code	Description
x00E	204	**UserCode**
x00F	002	Function address x402
	:	
x400	009	9 regs to copy
x401	220	
x402	1C2	LBL
x403	001	← **entry point**
x404	0F5	⊤
x405	000	
x406	054	T
x407	045	E
x408	053	S
x409	054	T
x40A	19C	FIX
x40B	000	0
x40C	1A6	XROM
x40D	C5	27,05
	:	

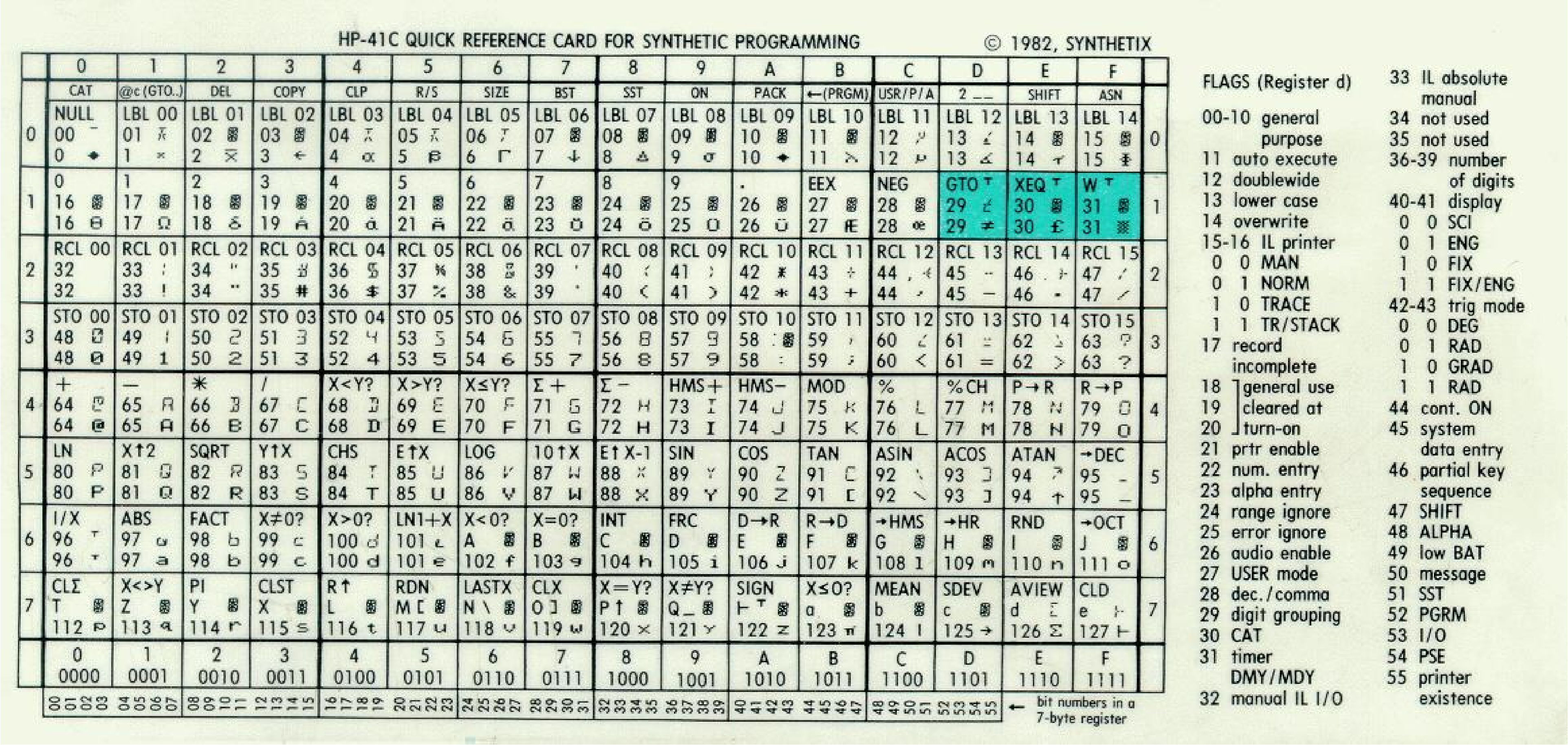
DEBUG registers

nybbles

ABS reg #	13	12	11	10	09	08	07	06	05	04	03	02	01	00
487	0	0	RTN 3				RTN 2				RTN 1			
488	KY		RTN 4				**XY**		P	Q	G		ST	
489	CPU register C													
490	CPU register A													
491	CPU register B													
492	CPU register M													
493	CPU register N													
494	STATUS register T													
495	STATUS register Z													
496	STATUS register Y													
497	STATUS register X													
498	STATUS register L													
499	STATUS register M													
500	STATUS register N													
501	STATUS register O													
502	STATUS register P													
503	STATUS register Q													
504	STATUS register ⊦													
505	STATUS register a													
506	STATUS register b													
507	STATUS register c													
508	STATUS register d													
509	STATUS register e													
510								BREAK address				word		
511								BREAK address+1				word		
#	13	12	11	10	09	80	07	06	05	04	03	02	01	00

XY bit	07	06	05	04	03	02	01	00
CPU flag #	13	12	11	10	9	8	v	w
							↓	↓
0 =							hex mode	SLCT P
1 =							dec mode	SLCT Q

Synthetic QRC



HP-41C QUICK REFERENCE CARD FOR SYNTHETIC PROGRAMMING

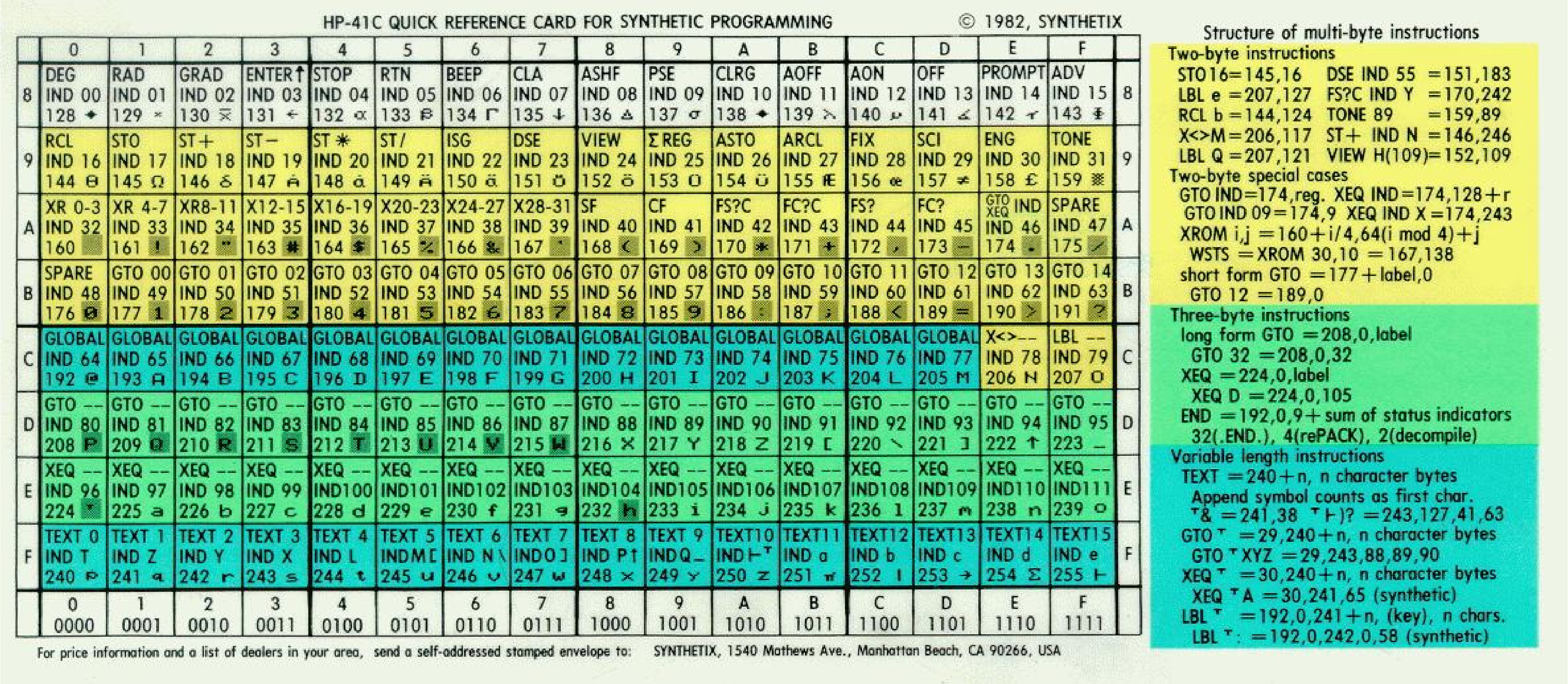
© 1982, SYNTHETIX

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
	CAT	@c (GTO..)	DEL	COPY	CLP	R/S	SIZE	BST	SST	ON	PACK	←(PRGM)	USR/P/A	2 __	SHIFT	ASN	
0	NULL 00 ¯ 0 ◆	LBL 00 01 x̄ 1 ×	LBL 01 02 ▩ 2 x̄	LBL 02 03 ▩ 3 ←	LBL 03 04 x̄ 4 α	LBL 04 05 x̄ 5 β	LBL 05 06 Γ 6 Γ	LBL 06 07 ▩ 7 ↓	LBL 07 08 ▩ 8 Δ	LBL 08 09 ▩ 9 σ	LBL 09 10 ▩ 10 ◆	LBL 10 11 ▩ 11 λ	LBL 11 12 µ 12 µ	LBL 12 13 ∡ 13 ∡	LBL 13 14 ▩ 14 τ	LBL 14 15 ▩ 15 Φ	0
1	0 16 ▩ 16 Θ	1 17 ▩ 17 Ω	2 18 ▩ 18 δ	3 19 ▩ 19 Á	4 20 ▩ 20 à	5 21 ▩ 21 Ä	6 22 ▩ 22 ä	7 23 ▩ 23 Ö	8 24 ▩ 24 ö	9 25 ▩ 25 Ü	. 26 ▩ 26 ü	EEX 27 ▩ 27 Æ	NEG 28 ▩ 28 œ	GTO ᵀ 29 ≠ 29 ≠	XEQ ᵀ 30 ▩ 30 £	W ᵀ 31 ▩ 31 ▒	1
2	RCL 00 32 32	RCL 01 33 ! 33 !	RCL 02 34 " 34 "	RCL 03 35 # 35 #	RCL 04 36 $ 36 $	RCL 05 37 % 37 %	RCL 06 38 & 38 &	RCL 07 39 ' 39 '	RCL 08 40 (40 (	RCL 09 41) 41)	RCL 10 42 * 42 *	RCL 11 43 + 43 +	RCL 12 44 , , 44 ,	RCL 13 45 - 45 —	RCL 14 46 . ⊦ 46 ·	RCL 15 47 / 47 /	2
3	STO 00 48 0 48 0	STO 01 49 1 49 1	STO 02 50 2 50 2	STO 03 51 3 51 3	STO 04 52 4 52 4	STO 05 53 5 53 5	STO 06 54 6 54 6	STO 07 55 7 55 7	STO 08 56 8 56 8	STO 09 57 9 57 9	STO 10 58 : ▩ 58 :	STO 11 59 ; 59 ;	STO 12 60 < 60 <	STO 13 61 = 61 =	STO 14 62 > 62 >	STO 15 63 ? 63 ?	3
4	+ 64 @ 64 @	— 65 A 65 A	✳ 66 B 66 B	/ 67 C 67 C	X<Y? 68 D 68 D	X>Y? 69 E 69 E	X≤Y? 70 F 70 F	Σ+ 71 G 71 G	Σ− 72 H 72 H	HMS+ 73 I 73 I	HMS− 74 J 74 J	MOD 75 K 75 K	% 76 L 76 L	%CH 77 M 77 M	P→R 78 N 78 N	R→P 79 O 79 O	4
5	LN 80 P 80 P	X↑2 81 Q 81 Q	SQRT 82 R 82 R	Y↑X 83 S 83 S	CHS 84 T 84 T	E↑X 85 U 85 U	LOG 86 V 86 V	10↑X 87 W 87 W	E↑X-1 88 X 88 X	SIN 89 Y 89 Y	COS 90 Z 90 Z	TAN 91 [91 [	ASIN 92 \ 92 \	ACOS 93] 93]	ATAN 94 ↑ 94 ↑	→DEC 95 _ 95 _	5
6	1/X 96 ᵀ 96 ᵀ	ABS 97 a 97 a	FACT 98 b 98 b	X≠0? 99 c 99 c	X>0? 100 d 100 d	LN1+X 101 e 101 e	X<0? A ▩ 102 f	X=0? B ▩ 103 g	INT C ▩ 104 h	FRC D ▩ 105 i	D→R E ▩ 106 j	R→D F ▩ 107 k	→HMS G ▩ 108 l	→HR H ▩ 109 m	RND I ▩ 110 n	→OCT J ▩ 111 o	6
7	CLΣ T ▩ 112 p	X<>Y Z ▩ 113 q	PI Y ▩ 114 r	CLST X ▩ 115 s	R↑ L ▩ 116 t	RDN M [▩ 117 u	LASTX N \ ▩ 118 v	CLX O] ▩ 119 w	X=Y? P ↑ ▩ 120 x	X≠Y? Q _ ▩ 121 y	SIGN ⊢ ᵀ ▩ 122 z	X≤0? a ▩ 123 π	MEAN b ▩ 124 |	SDEV c ▩ 125 →	AVIEW d Σ 126 Σ	CLD e ⊦ 127 ⊢	7
	0 0000	1 0001	2 0010	3 0011	4 0100	5 0101	6 0110	7 0111	8 1000	9 1001	A 1010	B 1011	C 1100	D 1101	E 1110	F 1111	
	00 01 02 03	04 05 06 07	08 09 10 11	12 13 14 15	16 17 18 19	20 21 22 23	24 25 26 27	28 29 30 31	32 33 34 35	36 37 38 39	40 41 42 43	44 45 46 47	48 49 50 51	52 53 54 55	← bit numbers in a 7-byte register		

FLAGS (Register d)

00-10	general purpose
11	auto execute
12	doublewide
13	lower case
14	overwrite
15-16	IL printer
0 0	MAN
0 1	NORM
1 0	TRACE
1 1	TR/STACK
17	record incomplete
18	general use cleared at turn-on
19	
20	
21	prtr enable
22	num. entry
23	alpha entry
24	range ignore
25	error ignore
26	audio enable
27	USER mode
28	dec./comma
29	digit grouping
30	CAT
31	timer DMY/MDY
32	manual IL I/O
33	IL absolute manual
34	not used
35	not used
36-39	number of digits
40-41	display
0 0	SCI
0 1	ENG
1 0	FIX
1 1	FIX/ENG
42-43	trig mode
0 0	DEG
0 1	RAD
1 0	GRAD
1 1	RAD
44	cont. ON
45	system data entry
46	partial key sequence
47	SHIFT
48	ALPHA
49	low BAT
50	message
51	SST
52	PGRM
53	I/O
54	PSE
55	printer existence

Synthetic QRC



HP-41C QUICK REFERENCE CARD FOR SYNTHETIC PROGRAMMING © 1982, SYNTHETIX

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
8	DEG IND 00 128 ◆	RAD IND 01 129 ×	GRAD IND 02 130 x̄	ENTER↑ IND 03 131 ←	STOP IND 04 132 α	RTN IND 05 133 β	BEEP IND 06 134 Γ	CLA IND 07 135 ↓	ASHF IND 08 136 Δ	PSE IND 09 137 σ	CLRG IND 10 138 ◆	AOFF IND 11 139 λ	AON IND 12 140 µ	OFF IND 13 141 ∡	PROMPT IND 14 142 τ	ADV IND 15 143 Φ	8
9	RCL IND 16 144 Θ	STO IND 17 145 Ω	ST+ IND 18 146 δ	ST− IND 19 147 Å	ST * IND 20 148 å	ST/ IND 21 149 Ä	ISG IND 22 150 ä	DSE IND 23 151 Ö	VIEW IND 24 152 ö	Σ REG IND 25 153 Ü	ASTO IND 26 154 ü	ARCL IND 27 155 Æ	FIX IND 28 156 æ	SCI IND 29 157 ≠	ENG IND 30 158 £	TONE IND 31 159 ▒	9
A	XR 0-3 IND 32 160	XR 4-7 IND 33 161 !	XR8-11 IND 34 162 "	X12-15 IND 35 163 #	X16-19 IND 36 164 $	X20-23 IND 37 165 %	X24-27 IND 38 166 &	X28-31 IND 39 167 '	SF IND 40 168 (	CF IND 41 169)	FS?C IND 42 170 *	FC?C IND 43 171 +	FS? IND 44 172 ,	FC? IND 45 173 -	GTO XEQ IND IND 46 174 .	SPARE IND 47 175 /	A
B	SPARE IND 48 176 0	GTO 00 IND 49 177 1	GTO 01 IND 50 178 2	GTO 02 IND 51 179 3	GTO 03 IND 52 180 4	GTO 04 IND 53 181 5	GTO 05 IND 54 182 6	GTO 06 IND 55 183 7	GTO 07 IND 56 184 8	GTO 08 IND 57 185 9	GTO 09 IND 58 186 :	GTO 10 IND 59 187 ;	GTO 11 IND 60 188 <	GTO 12 IND 61 189 =	GTO 13 IND 62 190 >	GTO 14 IND 63 191 ?	B
C	GLOBAL IND 64 192 @	GLOBAL IND 65 193 A	GLOBAL IND 66 194 B	GLOBAL IND 67 195 C	GLOBAL IND 68 196 D	GLOBAL IND 69 197 E	GLOBAL IND 70 198 F	GLOBAL IND 71 199 G	GLOBAL IND 72 200 H	GLOBAL IND 73 201 I	GLOBAL IND 74 202 J	GLOBAL IND 75 203 K	GLOBAL IND 76 204 L	GLOBAL IND 77 205 M	X<>__ IND 78 206 N	LBL -- IND 79 207 O	C
D	GTO -- IND 80 208 P	GTO -- IND 81 209 Q	GTO -- IND 82 210 R	GTO -- IND 83 211 S	GTO -- IND 84 212 T	GTO -- IND 85 213 U	GTO -- IND 86 214 V	GTO -- IND 87 215 W	GTO -- IND 88 216 X	GTO -- IND 89 217 Y	GTO -- IND 90 218 Z	GTO -- IND 91 219 [	GTO -- IND 92 220 \	GTO -- IND 93 221]	GTO -- IND 94 222 ↑	GTO -- IND 95 223 _	D
E	XEQ -- IND 96 224 `	XEQ -- IND 97 225 a	XEQ -- IND 98 226 b	XEQ -- IND 99 227 c	XEQ -- IND100 228 d	XEQ -- IND101 229 e	XEQ -- IND102 230 f	XEQ -- IND103 231 g	XEQ -- IND104 232 h	XEQ -- IND105 233 i	XEQ -- IND106 234 j	XEQ -- IND107 235 k	XEQ -- IND108 236 l	XEQ -- IND109 237 m	XEQ -- IND110 238 n	XEQ -- IND111 239 o	E
F	TEXT 0 IND T 240 p	TEXT 1 IND Z 241 q	TEXT 2 IND Y 242 r	TEXT 3 IND X 243 s	TEXT 4 IND L 244 t	TEXT 5 INDM[245 u	TEXT 6 IND N \ 246 v	TEXT 7 INDO] 247 w	TEXT 8 IND P↑ 248 x	TEXT 9 INDQ_ 249 y	TEXT10 IND⊢⊤ 250 z	TEXT11 IND a 251 π	TEXT12 IND b 252 |	TEXT13 IND c 253 →	TEXT14 IND d 254 Σ	TEXT15 IND e 255 ⊢	F
	0 0000	1 0001	2 0010	3 0011	4 0100	5 0101	6 0110	7 0111	8 1000	9 1001	A 1010	B 1011	C 1100	D 1101	E 1110	F 1111	

For price information and a list of dealers in your area, send a self-addressed stamped envelope to: SYNTHETIX, 1540 Mathews Ave., Manhattan Beach, CA 90266, USA

Structure of multi-byte instructions

Two-byte instructions

STO 16=145,16 DSE IND 55 =151,183
LBL e =207,127 FS?C IND Y =170,242
RCL b =144,124 TONE 89 =159,89
X<>M=206,117 ST+ IND N =146,246
LBL Q =207,121 VIEW H(109)=152,109

Two-byte special cases

GTO IND=174,reg. XEQ IND=174,128+r
GTO IND 09=174,9 XEQ IND X =174,243
XROM i,j =160+i/4,64(i mod 4)+j
WSTS =XROM 30,10 =167,138
short form GTO =177+label,0
GTO 12 =189,0

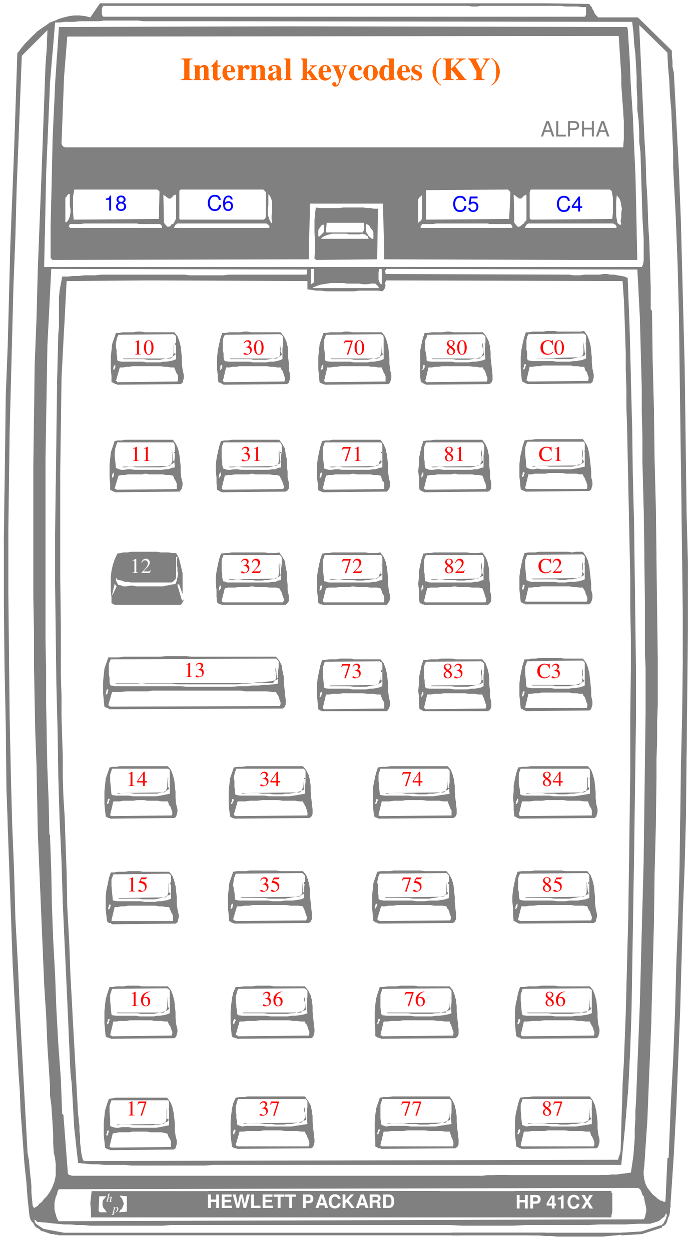
Three-byte instructions

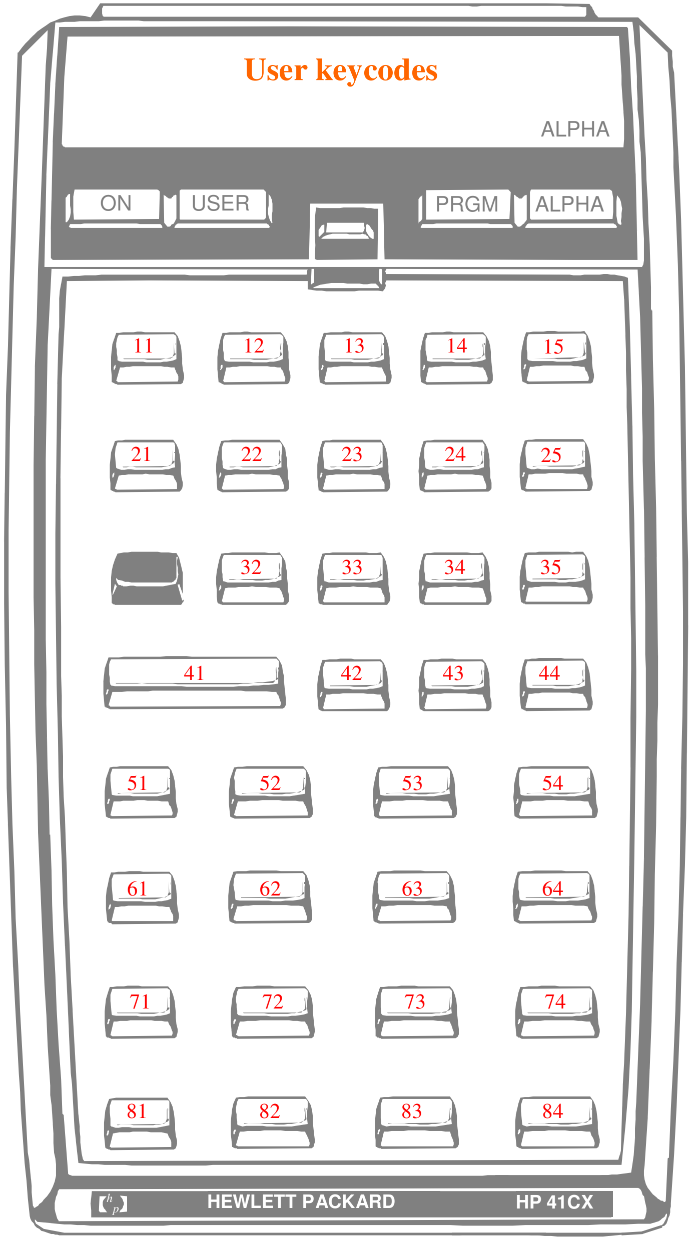
long form GTO =208,0,label
GTO 32 =208,0,32
XEQ =224,0,label
XEQ D =224,0,105
END =192,0,9+ sum of status indicators
32(.END.), 4(rePACK), 2(decompile)

Variable length instructions

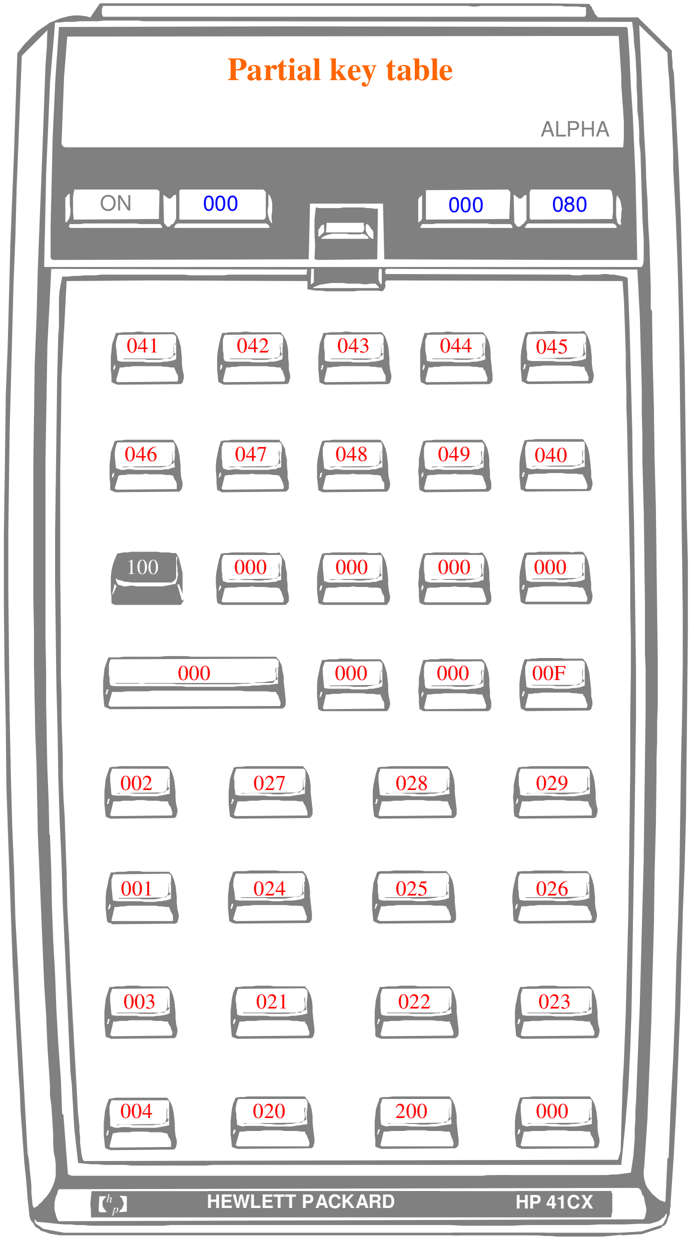
TEXT =240+n, n character bytes
Append symbol counts as first char.
⊤& =241,38 ⊤⊢)? =243,127,41,63
GTO ⊤ =29,240+n, n character bytes
GTO ⊤XYZ =29,243,88,89,90
XEQ ⊤ =30,240+n, n character bytes
XEQ ⊤A =30,241,65 (synthetic)
LBL ⊤ =192,0,241+n, (key), n chars.
LBL ⊤: =192,0,242,0,58 (synthetic)

Keycode maps



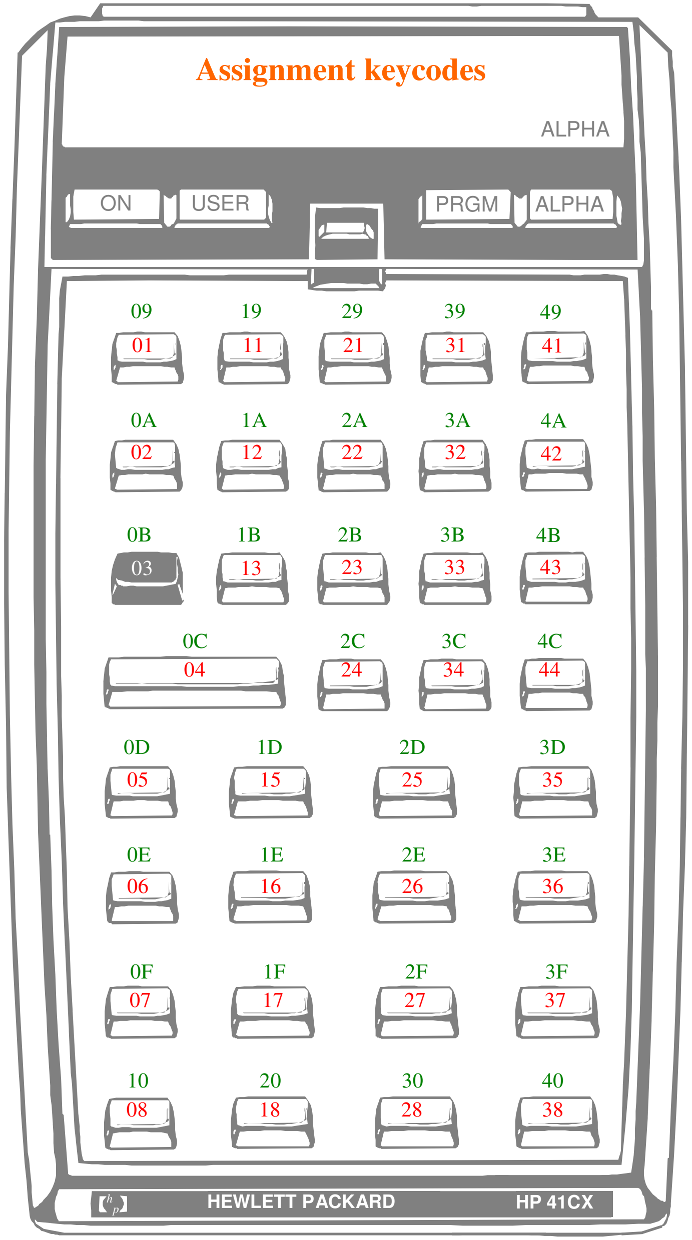


Keycode maps



Partial key table

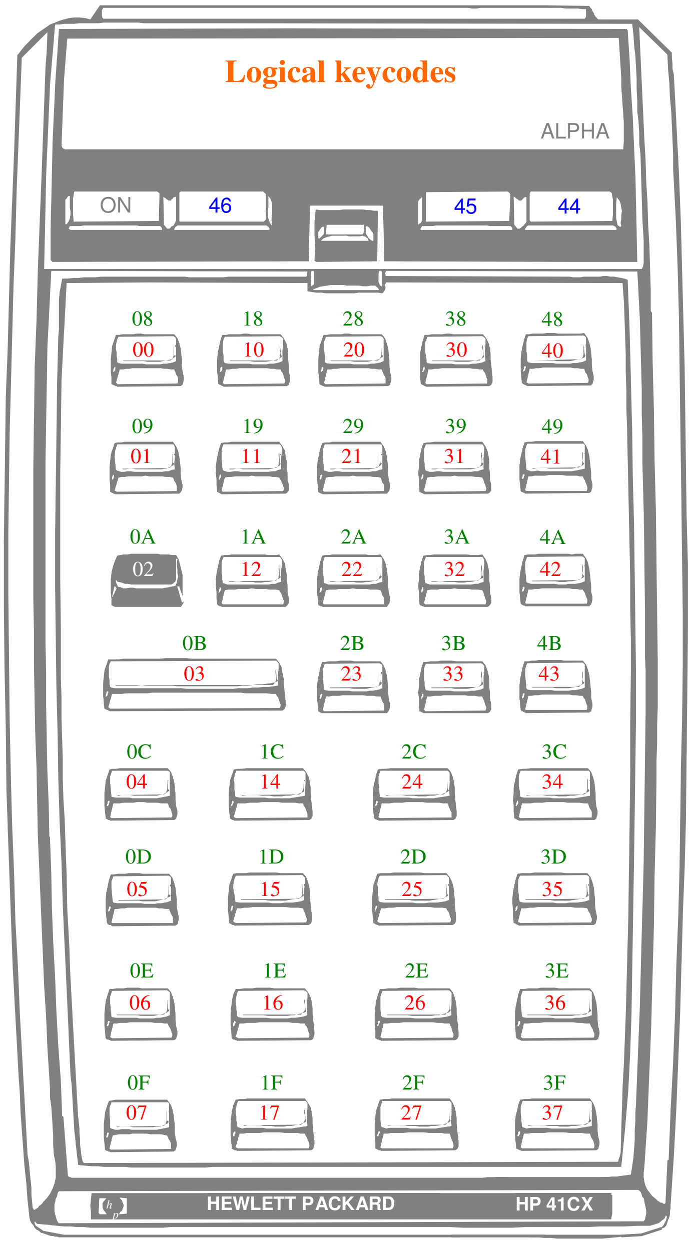
ON	000		000	080
041	042	043	044	045
046	047	048	049	040
100	000	000	000	000
000		000	000	00F
002	027	028	029	
001	024	025	026	
003	021	022	023	
004	020	200	000	



Assignment keycodes

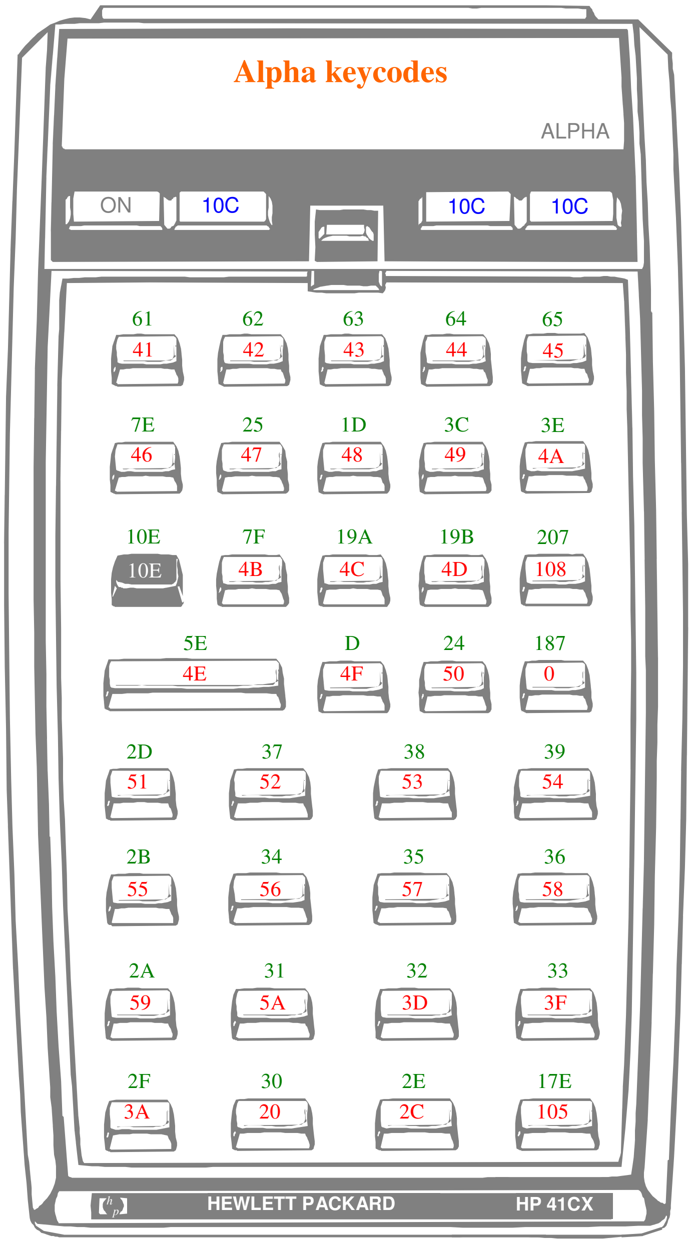
ON	USER		PRGM	ALPHA
09 01	19 11	29 21	39 31	49 41
0A 02	1A 12	2A 22	3A 32	4A 42
0B 03	1B 13	2B 23	3B 33	4B 43
0C 04		2C 24	3C 34	4C 44
0D 05	1D 15	2D 25	3D 35	
0E 06	1E 16	2E 26	3E 36	
0F 07	1F 17	2F 27	3F 37	
10 08	20 18	30 28	40 38	

Keycode maps



Logical keycodes

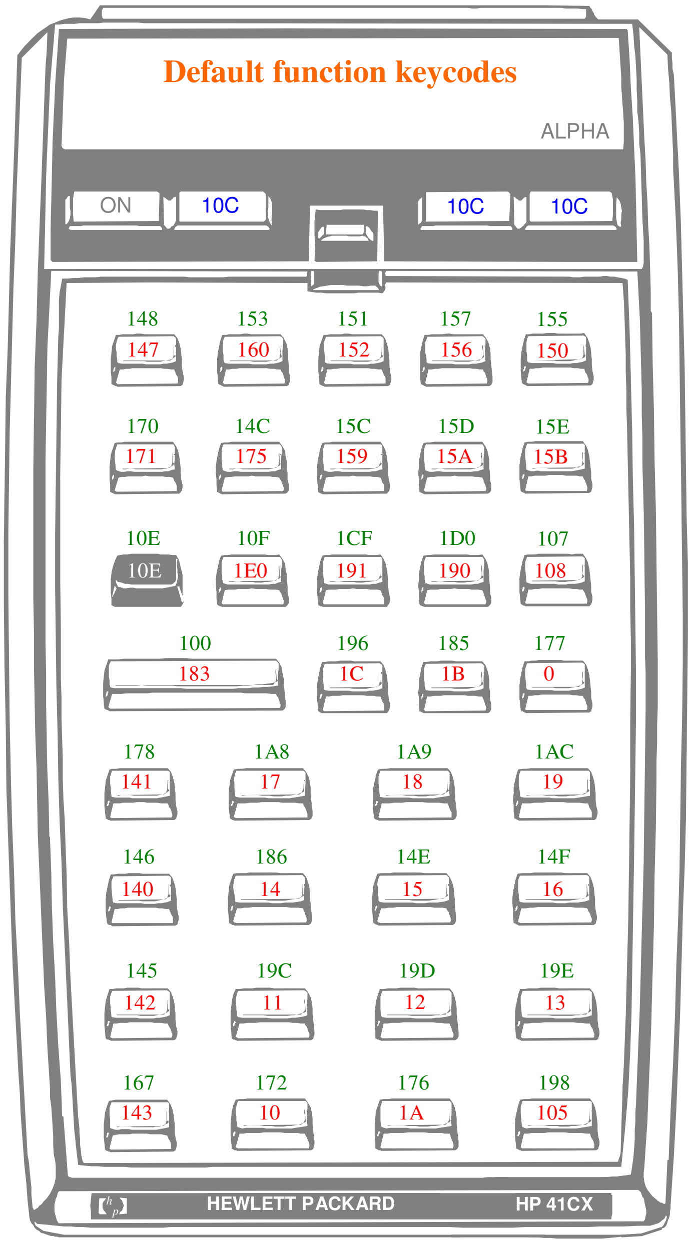
ON	46		45	44
08 00	18 10	28 20	38 30	48 40
09 01	19 11	29 21	39 31	49 41
0A 02	1A 12	2A 22	3A 32	4A 42
0B 03		2B 23	3B 33	4B 43
0C 04	1C 14	2C 24	3C 34	
0D 05	1D 15	2D 25	3D 35	
0E 06	1E 16	2E 26	3E 36	
0F 07	1F 17	2F 27	3F 37	



Alpha keycodes

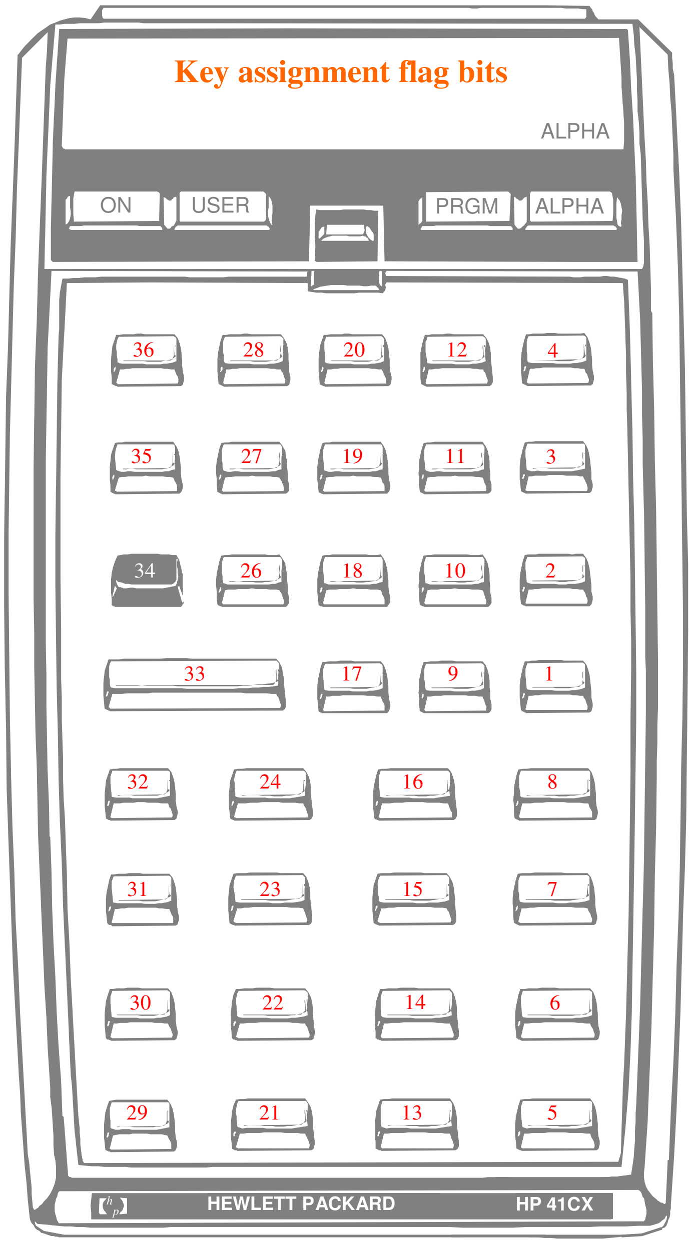
ON	10C		10C	10C
61 41	62 42	63 43	64 44	65 45
7E 46	25 47	1D 48	3C 49	3E 4A
10E 10E	7F 4B	19A 4C	19B 4D	207 108
5E 4E		D 4F	24 50	187 0
2D 51	37 52	38 53	39 54	
2B 55	34 56	35 57	36 58	
2A 59	31 5A	32 3D	33 3F	
2F 3A	30 20	2E 2C	17E 105	

Keycode maps



Default function keycodes

148 147	153 160	151 152	157 156	155 150
170 171	14C 175	15C 159	15D 15A	15E 15B
10E 10E	10F 1E0	1CF 191	1D0 190	107 108
100 183		196 1C	185 1B	177 0
178 141	1A8 17	1A9 18	1AC 19	
146 140	186 14	14E 15	14F 16	
145 142	19C 11	19D 12	19E 13	
167 143	172 10	176 1A	198 105	



Key assignment flag bits

36	28	20	12	4
35	27	19	11	3
34	26	18	10	2
33		17	9	1
32	24	16	8	
31	23	15	7	
30	22	14	6	
29	21	13	5	

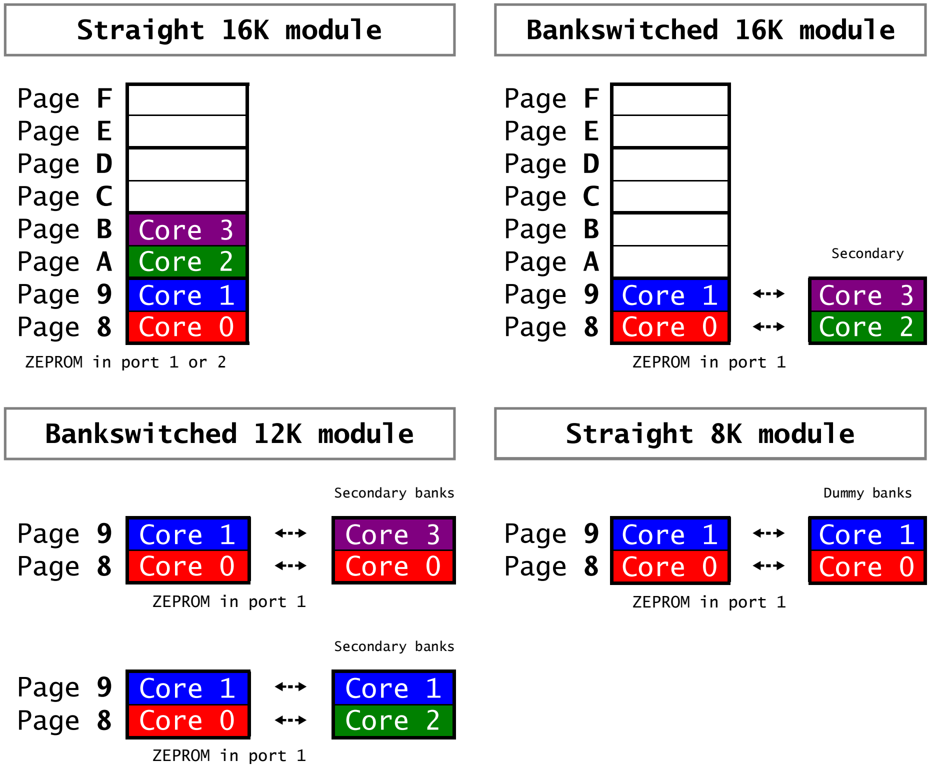
ROM module structure

Address	Code	Description
x000		XROM ID #
x001		# of FAT entries (n)
x002	t0a	Address of 1st function in page
x003	0aa	t is 0=Mcode or 2=UserCode
:		
x(2n)		Address of last function in page
x(2n+1)		
x(2n+2)	000	End of FAT
x(2n+3)	000	
x(2n+4)		Start of functions or program code
:		
:		
:		
:		
:		
:		
:		
xFC7	100	Enable bank 1
xFC8	3E0	
xFC9	180	Enable bank 2
xFCA	3E0	
:		
xFF4		Interrupt checked during Pause
xFF5		Interrupt checked if system flag 53 set
xFF6		Interrupt checked on wakeup /not ON key
xFF7		Interrupt checked when HP41 is turned off
xFF8		Interrupt checked just before CPU stops
xFF9		Interrupt checked on wakeup /ON key
xFFA		Interrupt checked on MEMORY LOST
xFFB		ROM revision ID
xFFC		ROM revision ID
xFFD		ROM revision ID (bit 8 & 9 is bankswitch-bits)
xFFE		ROM revision ID
xFFF		Page checksum

ROM bankswitching

Word at xFFD :

bit 9	bit 8	
0	0	page not bankswitched
0	1	bankswitched, primary bank
1	0	bankswitched, indeterminate bank (HP)
1	1	bankswitched, secondary bank



Straight 16K module

Page F
Page E
Page D
Page C
Page B Core 3
Page A Core 2
Page 9 Core 1
Page 8 Core 0
ZEPROM in port 1 or 2

Bankswitched 16K module

Page F
Page E
Page D
Page C
Page B
Page A
Secondary
Page 9 Core 1 ↔ Core 3
Page 8 Core 0 ↔ Core 2
ZEPROM in port 1

Bankswitched 12K module

Secondary banks
Page 9 Core 1 ↔ Core 3
Page 8 Core 0 ↔ Core 0
ZEPROM in port 1

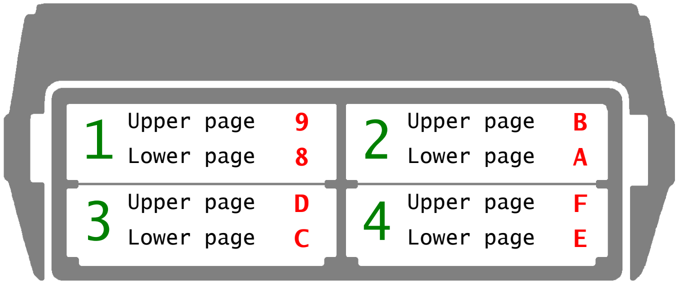
Secondary banks
Page 9 Core 1 ↔ Core 1
Page 8 Core 0 ↔ Core 2
ZEPROM in port 1

Straight 8K module

Dummy banks
Page 9 Core 1 ↔ Core 1
Page 8 Core 0 ↔ Core 0
ZEPROM in port 1

ROM structure

Primary Bank		Page	Secondary Bank		Page
PORT 4	Upper page	F	PORT 4	Upper page	F
	Lower page	E		Lower page	E
PORT 3	Upper page	D	PORT 3	Upper page	D
	Lower page	C		Lower page	C
PORT 2	Upper page	B	PORT 2	Upper page	B
	Lower page	A		Lower page	A
PORT 1	Upper page	9	PORT 1	Upper page	9
	Lower page	8		Lower page	8
IL Mass Storage		7			
Printer ROM		6	IR printer		6
Timer ROM		5	CX Extended		5
Take-over ROM		4			
CX Extended		3			
System ROM 2		2			
System ROM 1		1			
System ROM 0		0			



1	Upper page	9	2	Upper page	B
	Lower page	8		Lower page	A
3	Upper page	D	4	Upper page	F
	Lower page	C		Lower page	E

GTO 00 - 14 forward

Bx xx

RAM		ROM
LBL 01		LBL 01
B2 00	2 bytes	1B2 080
B2 10	3 bytes	1B2 081
B2 20	4 bytes	1B2 082
.. ..	..	.. ..
B2 60	8 bytes	1B2 086
B2 01	9 bytes	1B2 087
B2 11	10 bytes	1B2 088
.. ..	..	.. ..
B2 5F	112 bytes	1B2 0EE
B2 6F	113 bytes	1B2 0EF

```
1011 1111 0bbb rrrr

1011 is B
     1111 is label# + 1
          0 is forward
           bbb is number of bytes
               rrrr is number of registers
```

GTO 00 - 14 backwards

Bx xx

RAM		ROM
LBL 01		LBL 01
B2 B0	2 bytes	1B2 003
B2 C0	3 bytes	1B2 004
.. ..	..	.. ..
B2 F0	6 bytes	1B2 007
B2 81	7 bytes	1B2 008
B2 91	8 bytes	1B2 009
B2 A1	9 bytes	1B2 00A
B2 B1	10 bytes	1B2 00B
.. ..	..	.. ..
B2 DF	108 bytes	1B2 06E
B2 EF	109 bytes	1B2 06F

```
1011 1111 1bbb rrrr

1011 is B
     1111 is label# + 1
          1 is backwards
           bbb is number of bytes
               rrrr is number of registers
```

GTO 15 - 99 forward

Dx xx xx

RAM		ROM
LBL 15		LBL 15
D4 00 0F	2 bytes	1D0 002 08F
D6 00 0F	3 bytes	1D0 003 08F
D8 00 0F	4 bytes	1D0 004 08F
DA 00 0F	5 bytes	1D0 005 08F
DC 00 0F	6 bytes	1D0 006 08F
D0 01 0F	7 bytes	1D0 007 08F
D2 01 0F	8 bytes	1D0 008 08F
.. .. ..	..	.. .. ..

```
1101 bbbr rrrr rrrr 0111 1111

1101 is D
     bbb is number of bytes
        r rrrr rrrr is number of registers
                  0 is forward
                   111 1111 is label#
```

GTO 15 - 99 backwards

Dx xx xx

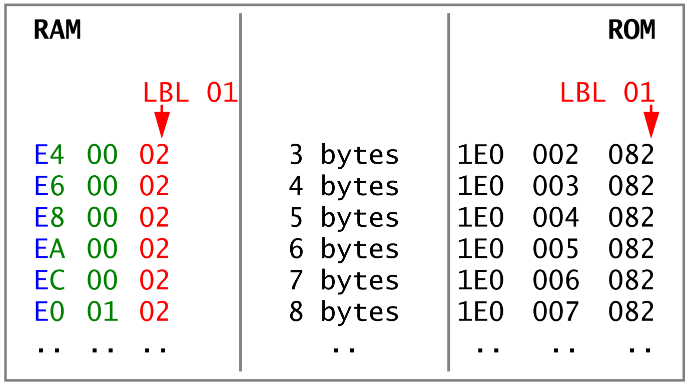
RAM		ROM
LBL 15		LBL 15
D6 00 8F	3 bytes	1D0 003 00F
D8 00 8F	4 bytes	1D0 004 00F
DA 00 8F	5 bytes	1D0 005 00F
DC 00 8F	6 bytes	1D0 006 00F
D0 01 8F	7 bytes	1D0 007 00F
D2 01 8F	8 bytes	1D0 008 00F
D4 01 8F	9 bytes	1D0 009 00F
.. .. ..	..	.. .. ..

```
1101 bbbr rrrr rrrr 1111 1111

1101 is D
     bbb is number of bytes
        r rrrr rrrr is number of registers
                  1 is backwards
                   111 1111 is label#
```

XEQ forward

Ex xx xx



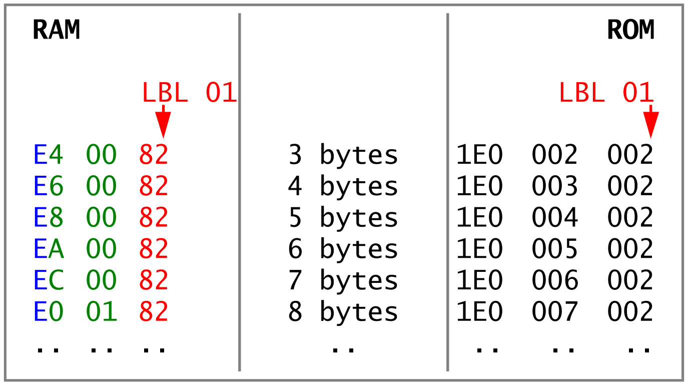
RAM		ROM
LBL 01		LBL 01
E4 00 02	3 bytes	1E0 002 082
E6 00 02	4 bytes	1E0 003 082
E8 00 02	5 bytes	1E0 004 082
EA 00 02	6 bytes	1E0 005 082
EC 00 02	7 bytes	1E0 006 082
E0 01 02	8 bytes	1E0 007 082
.. .. ..	..	.. .. ..

```
1110 bbbr rrrr rrrr 0111 1111

1110 is E
     bbb is number of bytes
        r rrrr rrrr is number of registers
                  0 is forwards
                   111 1111 is label#
```

XEQ backwards

Ex xx xx



RAM		ROM
LBL 01		LBL 01
E4 00 82	3 bytes	1E0 002 002
E6 00 82	4 bytes	1E0 003 002
E8 00 82	5 bytes	1E0 004 002
EA 00 82	6 bytes	1E0 005 002
EC 00 82	7 bytes	1E0 006 002
E0 01 82	8 bytes	1E0 007 002
.. .. ..	..	.. .. ..

```
1110 bbbr rrrr rrrr 1111 1111

1110 is E
     bbb is number of bytes
        r rrrr rrrr is number of registers
                  1 is backwards
                   111 1111 is label#
```

XROM

Ax xx

```
1010 0ddd ddff ffff

1010 is A
      ddd dd is device#
            ff ffff is function#
```

END

Cx xx xx

```
1100 bbbr rrrr rrrr 0ne0 1pc1

1100 is C
     bbb is number of bytes to next END
        r rrrr rrrr   is number of registers to next END
                     n is 0=NOT PRIVATE or 1=PRIVATE
                      e is 0=END or 1=.END.
                          p is 0=packed or 1=unpacked
                           c is 0=compiled or 1=not c
```

XEQ & GTO IND

AE xx

```
1010 1110 trrr rrrr

1010 1110 is AE
          t is 0=GTO IND or 1=XEQ IND
           rrr rrrr  is the ”indirect” register
```

alpha

```
Fx xx xx ....  xx

1111 nnnn aabb ccdd .... eeff

1111 is F
     nnnn is number of letters in the text
          aabb ccdd .... eeff is the text
```

XEQ alpha

```
1E Fx xx xx ....

0001 1110 1111 nnnn aabb ccdd .... eeff

0001 1110 1111 er 1E F
               nnnn is number of letters in the text
                    aabb ccdd .... eeff is the text
```

GTO alpha

```
1D Fx xx xx ....

0001 1101 1111 nnnn aabb ccdd .... eeff

0001 1101 1111 is 1D F
               nnnn is number of letters in the text
                    aabb ccdd .... eeff is the text
```

LBL alpha

```
Cx xx Fx xx xx xx .. xx

1010 bbbr rrrr rrrr 1111 nnnn kkkk aabb ccdd .... eeff

1010 is C
     bbb is number of bytes to next LBL
        r rrrr rrrr  is number of registers to next LBL
                    1111 is F
                         nnnn is number of letters + 1
                              kkkk is assign keycode
                                   aabb ccdd .... eeff
                                                 is the text
```

HP41 CX mainframe functions

ID = 004

000	CAT
001	GTO..
002	DEL
003	COPY
004	CLP
005	R/S
006	SIZE
007	BST
008	SST
009	ON
010	PACK
014	SHIFT
015	ASN
064	+
065	-
066	*
067	/
068	X<Y?
069	X>Y?
070	X<=Y?
071	Σ+
072	Σ-
073	HMS+
074	HMS-
075	MOD
076	%
077	%CH
078	P-R
079	R-P
080	LN
081	X↑2
082	SQRT
083	Y↑X
084	CHS
085	E↑X
086	LOG
087	10↑X
088	E↑X-1
089	SIN
090	COS
091	TAN
092	ASIN
093	ACOS
094	ATAN
095	DEC
096	1/X
097	ABS
098	FACT
099	X≠0?
100	X>0?
101	LNX+1
102	X<0?
103	X=0?
104	INT
105	FRC
106	D-R
107	R-D
108	HMS
109	HR
110	RND
111	OCT
112	CLΣ
113	X<>Y
114	PI
115	CLST
116	R↑
117	RDN
118	LASTX
119	CLX
120	X=Y?
121	X≠Y?
122	SIGN
123	X<=0?
124	MEAN
125	SDEV
126	AVIEW
127	CLD
128	DEG
129	RAD
130	GRAD
131	ENTER↑
132	STOP
133	RTN
134	BEEP
135	CLA
136	ASHF
137	PSE
138	CLRG
139	AOFF
140	AON
141	OFF
142	PROMPT
143	ADV
144	RCL
145	STO
146	ST+
147	ST-
148	ST*
149	ST/
150	ISG
151	DSE
152	VIEW
153	ΣREG
154	ASTO
155	ARCL
156	FIX
157	SCI
158	ENG
159	TONE
168	SF
169	CF
170	FS?C
171	FC?C
172	FS?
173	FC?
206	X<>
207	LBL
208	GTO
224	XEQ

MODULE ID's

The colors represent the way I have the module: green = actual module; purple, black & blue = programable module; orange = digital

XROM#	◆MODULE					◆NoVRAM	◆CLONIX41	◆.ROM		◆ZEPROM
1	MATH 1					41Z ROM				LAND NAV
2	STAT 1						EXTENDED IL			
3	SURVEYING 1					SANDMATH				
4	FINANCIAL 1	CMT-200							ES-41	CO-OP ROM
5	STANDARD	PANAME	ZENROM 1						BLDROM	RAMBOX32
6	CIRCUITS 1							TOMSROM	ALPHA ROM	
7	STRC ANAL X	STD HEPAX				HEPAX 1D				
8	STRESS					SANDBOX			SKWIDBC	
9	CCD MODULE	PANAME	HOME MGMT				ZEP PROGRAM		TEST	
10	PPC ROM	GAMES	KOELN3	FORECAST1	FORECASTER2			ERAMCO MLDL	AUTO/DUP	
11	CCD MODULE	REAL ESTATE								
12	MACHINE									
13	THERMAL					SANDBOX			TOOLBOX ROM	
14	NAVIGATION								PROTOPARIO	
15	PETROLEUM							MC-EPROM	DISASM4C	
16	PETROLEUM							SIMPLEX		
17	PLOTTER						NFCROM		BLDROM2	
18	PLOTTER	AECROM					AECROM			
19	STRC ANAL X	SECURITIES X	AVIATION X	CLIN LAB X				Advantage Application ROM		HP-IL DIAG
20	PPC ROM									
21	DEL.SERV.1	SCHENK MODUL	KOELN4	SEA KING MK5	ML-IDC	BG/UG IDC	MLROM	DATA LOGGER	HYDRACOMP	ASSEMBLER 3
22	ADVANTAGE	HP-IL DEVEL			ML-IDC					
23	EXTENDED I/O									
24	ADVANTAGE	HP-IL DEVEL								
25	X FUNCTIONS									
26	TIME MODULE									
27	WAND						DAVID ASSEM		PROFISET L	
28	HP-IL									
29	PRINTER	I/R PRINTER	IL-PRINTER							
30	CARD READER									
31	DEL.SERV.1	SCHENK MODUL	KOELN4			RAMBOXII	SUP-R-ROM	DATA LOGGER	PROFISET U	CO-OP ROM
◆NO ID				Laitram XQ2	PRINTERSERVICE		FORTH 41	Service-C	Service-CX	TEST ROM

XROM

◆HEADER ◆MCODE ◆USERCODE

01
MATH 1

00	MATH 1C
01	MATRIX
02	SIMEQ
03	VCOL
04	VMAT
05	PVT
06	DET
07	INV
08	EDIT
09	SOLVE
10	SOL
11	POLY
12	ROOTS
13	INTG
14	DIFEQ
15	FOUR
16	C+
17	C-
18	C*
19	C/
20	MAGZ
21	CINV
22	Z↗N
23	Z↗1/N
24	e↗N
25	LNZ
26	a↗Z
27	LOGZ
28	Z↗W
29	Z↗1/W
30	SINZ
31	COSZ
32	TANZ
33	SINH
34	COSH
35	TANH
36	ASINH
37	ACISH
38	ATANH
39	SSS
40	ASA
41	SAA
42	SAS
43	SSA
44	TRANS
45	*FN

02
STAT 1

00	STAT 1B
01	ΣBSTAT
02	ΣBSTG
03	*BE
04	ΣMMTUG
05	ΣMMTGD
06	*MT
07	*MD
08	ΣAOVONE
09	ΣANOTWO
10	ΣANOCOV
11	ΣLIN
12	ΣEXP
13	ΣLOGVONE
14	ΣPOW
15	ΣPOLYP
16	ΣPOLYC
17	ΣMLRXY
18	ΣMLRXYZ
19	ΣPTST
20	ΣTSTAT
21	ΣXSQEV
22	ΣEEFXSQ
23	ΣCTKK
24	ΣCTKKK
25	ΣSPEAR
26	ΣNORMD
27	ΣCHISQD
28	*a
29	*b

02
DAVID ASSEM

00	DAVID-ASSEM
01	ASSM
02	BEG/END
03	BUF>REG
04	DISTOA
05	REG>BUF

03
SURVEYING 1

00	SURVEY 1B
01	TRAV
02	COMP
03	TRANSIT
04	ADJUST
05	INTER
06	CURVE
07	HORIZ
08	VERT
09	RESECT
10	PREAREA
11	ENDVOL
12	PIT
13	COORD
14	ACRES
15	AZ
16	BRG
17	CIR
18	INVERSE
19	NE
20	TS
21	*AO
22	*∡
23	*BR
24	*DL
25	*DS
26	*EL
27	*H
28	*S
29	*A1
30	*B1
31	*IN
32	*YN

04
CMT-200

00	-DIGIO 1B
01	ANDXY
02	A>BUF
03	BUF>AX
04	BUF>TX
05	BUF>X
06	BUFX
07	FLAG>X
08	FLAG<>X
09	HS
10	IA
11	IAX
12	IBUFX
13	ICX
14	IDX
15	INTMASK
16	INTOFF
17	INTON
18	INVOFF
19	INVON
20	MATCHI
21	NOTX
22	OA
23	OBUFX
24	OCX
25	ODX
26	ORXY
27	PT>X
28	RATE
29	STROBE
30	TIMEI
31	TIMEO
32	TX>BUF
33	VIEWIN
34	WAITIX
35	X>A
36	X>BUF
37	X>FLAG
38	X>PT

XROM

◆HEADER ◆MCODE ◆USERCODE

04 CO-OP MODULE

00	SUN
01	ALM 1
02	AV
03	GRID
04	SETS
05	BS
06	KC
07	KD
08	TS
09	BB
10	BD
11	DD
12	PERPO
13	CR
14	TR
15	SS
16	COORTR
17	PINV
18	SSSTO
19	PNO
20	TMS
21	T41
22	PP
23	IN
24	CX
25	TRASTO
26	PTRA
27	NEXTNO
28	RECALL
29	RMM
30	REM
31	INV
32	ATB
33	BTA
34	BAZ
35	DMS
36	T
37	TRA
38	POB
39	PBEG
40	PRCL
41	ASWC
42	BSWC
43	ATOC
44	ASWB
45	PRTPT
46	F2
47	ACRE
48	STASTO
49	∡ RST
50	STAKE
51	PRTC
52	RPS
53	RS
54	RPR
55	RR
56	Z
57	PPP
58	NEXTN

04 FINANCIAL 1

00	FINANCE 1D
01	MONEY
02	IRR
03	MIRR
04	NPV
05	AMORT
06	SL
07	DB
08	SOYD
09	BOND
10	DAYS
11	*N
12	*I
13	*PV
14	*PMT
15	*FV
16	*IRR
17	*MIRR
18	*NPV
19	*AMORT
20	*SL
21	*DB
22	*SOYD
23	*PRC
24	*YLD
25	*DAYS
26	*BGN
27	*SIZE
28	*DATA
29	*DATA1
30	*OUT
31	*TGL
32	*TGL1
33	*Y/N

05 PANAME

00	-ADV PRT 3C
01	AID
02	ID
03	FINDAID
04	OUTAX
05	OUTCR
06	OUTLF
07	OUTLFX
08	OUTSPX
09	OUTXB
10	OUTYBX
11	OUTa
12	OUTaX
13	RCLSEL
14	-82163 FCNS
15	CLEAR
16	CLEARO
17	CSRDN
18	CSRHX
19	CSRL
20	CSROFF
21	CSRON
22	CSRR
23	CSRVX
24	CSRUP
25	CTYPE
26	HOME
27	SCRLDN
28	SCRLUP
29	SCRLX
30	XYTAB
31	-82162 FCNS
32	CLBUF
33	8BIT
34	ESCAPE
35	PARSE
36	STATUS
37	TABCOL
38	UNPARSE
39	-82905 FCNS
40	BELL
41	CHARSET
42	FFEED
43	FORMLEN
44	GRAPHX
45	MODE
46	SKIPOFF
47	SKIPON
48	TEXTLEN
49	VSPAC
50	-PLOT FCNS
51	AXIS
52	BACKSP
53	BACKSPX
54	BOX
55	COLOR
56	*CSIZE
57	*DRAW
58	*HOME
59	*LABEL
60	*LDIR
61	*LTYPE
62	*MOVE
63	*PLREGX

05 ZENROM 1

00	ZENROM 3B
01	CLMM
02	CLXM
03	CODE
04	DECODE
05	LASTP
06	MCED
07	NOP
08	NRCLM
09	NRCLX
10	NSTOM
11	RAMED
12	TOGF

XROM

◆HEADER ◆MCODE ◆USERCODE

05
STANDARD

00	STRD 1A	39	FF
01	CLSTK	40	HEX
02	1	41	FIN
03	2	42	ROOT
04	3	43	LIN
05	4	44	EXP
06	5	45	LOG
07	6	46	POW
08	7	47	INIT
09	8	48	CSUB
10	9	49	CADD
11	0	50	CDIV
12	STACK	51	CMULT
13	E↗	52	UV
14	RDWN	53	CRD
15	SWAP	54	Q
16	RUP	55	10
17	PL	56	SH
18	MI	57	DL
19	MU	58	S
20	DI	59	HT
21	CLR	60	DB
22	CHSN	61	PH
23	ST	62	DH
24	RC	63	P
25	LSTK		
26	CLNDR		
27	WORDS		
28	LTTR		
29	DESPEL		
30	TEACH		
31	AGN		
32	TRY		
33	YES		
34	+		
35	-		
36	*		
37	/		
38	RNDM		

06
CIRCUIT 1

00	CIR ANL 1A	39	STUBO=
01	GNAP	40	STUBS=
02	*J	41	PV
03	PHASE	42	P∡V
04	H<F>dB	43	PVdB
05	H<F>	44	∡V2/V1
06	GM=	45	V2/1dB
07	R=	46	V2/V1
08	L=	47	PI
09	C=	48	P∡I
10	LNAP	49	PIdB
11	RS	50	∡I2/I1
12	RP	51	I2/1dB
13	CS	52	I2/I1
14	CP	53	PZIN
15	LS	54	P∡ZIN
16	LP	55	ZIN
17	TF	56	∡ZIN
18	GY	57	PP2/P1
19	LCS	58	P2/P1
20	LCP	59	ALL
21	LINE	60	*AN
22	STUBO	61	*MAT*
23	STUBS	62	*C*
24	ICIS	63	*C+
25	VCIS		
26	RS=		
27	LS=		
28	CS=		
29	LCS=		
30	LINE=		
31	ICIS=		
32	VCIS=		
33	TF=		
34	GY=		
35	RP=		
36	LP=		
37	CP=		
38	LCP=		

07
HEPAX

00	-HEPAX 1D	39	PRIVATE
01	HAPPCHR	40	CLRAM
02	HAPPREC	41	CODE
03	HARCLRC	42	COPYROM
04	HASROOM	43	DECODE
05	HCLFL	44	DECODYX
06	HRFLAS	45	DISASM
07	HCRFLD	46	HEPAX
08	HDELCRH	47	HEPAXA
09	HDELREC	48	HEXEDIT
10	HEPDIR	49	HPROMPT
11	HEPDIRX	50	RAMTOG
12	HEPROOM	51	READROM
13	HFLSIZE	52	WRTROM
14	HGETA	53	XF
15	HGETK	54	XFA
16	HGETR		
17	HGETREC		
18	HGETRX		
19	HGETX		
20	HINSCHR		
21	HINSREC		
22	HPOSFL		
23	HPURFL		
24	HRCLPT		
25	HRCLPTA		
26	HREADFL		
27	HRENAME		
28	HSAVEA		
29	HSAVEK		
30	HSAVEP		
31	HSAVER		
32	HSAVERX		
33	HSAVEX		
34	HSEC		
35	HSEKPT		
36	HSEKPTA		
37	HUNSEC		
38	HWRTFL		

XROM

◆HEADER ◆MCODE ◆USERCODE

07
STRC ANAL X

00	STRCTA 1B
01	SECTION
02	SIMPLE
03	CANT
04	FIXED
05	PROPPED
06	SPAN
07	NSPAN
08	FIXL
09	FIXR
10	MOMENTS
11	SETTLE
12	CFRAME
13	VECTOR
14	SIZE?
15	SZ?
16	ATANY/X
17	BEAM
18	*B
19	*AI
20	*L1
21	*P1
22	*M1
23	*W1
24	*P
25	*M
26	*L
27	*W

08
STRESS

00	STRESS 1B
01	SIMPLE
02	CANT
03	FIXED
04	PROPPED
05	SPAN
06	NSPAN
07	FIXL
08	FIXL
09	MOMENTS
10	COLUMN
11	SECTION
12	MOHR
13	DELTA
14	REC
15	SODER
16	VECTOR
17	SIZE?
18	ATANY/X
19	BEAM
20	*B
21	*AI
22	*LI
23	*L
24	*P1
25	*P
26	*M1
27	*M
28	*W1
29	*W
30	*ROS
31	*MO
32	*COL

09
PANAME

00	RDRAW	39	RGINIT
01	RESET	40	RGNb
02	REVLF	41	RGORD
03	REVLFX	42	RGXTR
04	RMOVE	43	RGSUM
05	SETORG	44	RGVIEW
06	-UTILITIES	45	SAVERGX
07	/MOD	46	SIZE?
08	AD-LC	47	SORT
09	ALENG	48	STO>L
10	ANUM	49	SUB$
11	ANUMDEL	50	TF55
12	APPX	51	VKEYS
13	AROT	52	WRTEM
14	ATOXL	53	X=NN?
15	ATOXR	54	X NN?
16	ATOXX	55	X<NN?
17	BLDPT	56	X<=NN?
18	BRKPT	57	X>NN?
19	CHFLAG	58	X>=NN?
20	CLINC	59	X<>F
21	COLPT	60	XTOAL
22	GETRGX	61	XTOAR
23	LC-AD	62	Y/N
24	LINPT	63	YTOAX
25	NOP		
26	OUT		
27	POSA		
28	PSIZE		
29	READEM		
30	RG		
31	RG+-		
32	RG*		
33	RG/		
34	RG+Y		
35	RG*Y		
36	RG/Y		
37	RGAX		
38	RGCOPY		

09
CCD

00	-W&W CCD A	39	R>R?
01	B?	40	RMAXAB
02	CAS	41	RNRM
03	CLB	42	RSUM
04	RNDM	43	SUM
05	SAS	44	SUMAB
06	SEED	45	SWAP
07	SORT	46	YC+C
08	-ARR FNS	47	-HEX FNS
09	>C+	48	1CMP
10	>R+	49	2CMP
11	?IJ	50	AND
12	?IJA	51	bC?
13	C<>C	52	bS?
14	C>+	53	Cb
15	C>-	54	NOT
16	CMAXAB	55	OR
17	CNRM	56	R<
18	CSUM	57	R>
19	DIM	58	S<
20	FNRM	59	S>
21	IJ=	60	Sb
22	IJ=A	61	UNS
23	M+	62	WSIZE
24	M-	63	XOR
25	M*		
26	M*M		
27	M/		
28	MAX		
29	MAXAB		
30	MDIM		
31	MIN		
32	MOVE		
33	PIV		
34	R-PR		
35	R-QR		
36	R<>R		
37	R>+		
38	R>-		

XROM

◆HEADER ◆MCODE ◆USERCODE

09 HOME MGMT

00	HOME MN 1A
01	BUDGET
02	TRAVEL
03	STOCKS
04	FINANCE
05	BAL
06	IRA
07	INS
08	CHECK
09	HOME
10	BUY?
11	STORE
12	RELOAD
13	CL
14	*
15	*0
16	*e
17	N
18	*I
19	PV
20	PMT
21	FV
22	*J
23	*H

09 ZEP PROG

00	-ZEP PROG
01	ADDBSW
02	ADDMCF
03	ADDUCF
04	BLANK?
05	BNKSW?
06	BURNUC
07	BURNWD
08	CHKSUM
09	COMPUC
10	COPYPG
11	CPXYZ
12	DECHEX
13	ENABLEP
14	ENABLES
15	FRSPC?
16	HEXDEC
17	ILBURN
18	ILSAVE
19	INIT
20	INITP
21	INITPG
22	PB01
23	PGX=Y?
24	PRGMLN
25	READWD
26	REBURN
27	RRBURN
28	SB01
29	UCBURN

10 GAMES

00	GAMES 1A
01	BAGELS
02	BIOR
03	BIOF
04	CRAPS
05	HANG
06	PINBALL
07	SWAR
08	SUBHUNT
09	BOOM
10	INI
11	P
12	SIZE?
13	RNDM
14	RNDMW

10 FORECAST 1

00	*FORECASTER
01	START
02	EDIT
03	UP
04	LS
05	FC
06	CS
07	MR
08	T1
09	T2
10	TL
11	C1
12	C2
13	C3
14	SB
15	DS
16	SS
17	SZ
18	CC
19	*ASTRO
20	JD
21	CD
22	DBD
23	DAD
24	DOW
25	DOCY
26	DOLY
27	DLIST
28	PREC
29	EQ-H
30	RISE
31	SET
32	ANG
33	LMT
34	ST
35	UT
36	GST
37	LST
38	FW
39	BW
40	QLST
41	OUT
42	ACXR

10 FORECASTER 2

00	FORECASTER2
01	START
02	CS
03	EDIT
04	UP
05	LS
06	FC
07	CA
08	PS
09	TF
10	DFC
11	PT
12	RC
13	WC
14	TL
15	SS
16	DS
17	SB
18	BX
19	SZ
20	C1
21	C2
22	MS
23	FCA
24	FC0
25	S
26	EN
27	PPV
28	PSS
29	FT
30	PP
31	NFT
32	GINT

XROM

◆HEADER ◆MCODE ◆USERCODE

10
PPC

00	C PPC 1981	39	SU
01	MK	40	NH
02	1K	41	HN
03	+K	42	BL
04	F?	43	FL
05	LF	44	BI
06	CK	45	IP
07	VA	46	PS
08	UD	47	T1
09	PK	48	MS
10	A?	49	IF
11	DC	50	CB
12	ML	51	RT
13	RF	52	PD
14	Σ?	53	DP
15	S?	54	QR
16	C?	55	2D
17	DT	56	SX
18	AD	57	RX
19	XE	58	OM
20	HD	59	PA
21	ΣC	60	GE
22	LB	61	Ab
23	L-	62	E?
24	-B	63	FI
25	XD		
26	VM		
27	EX		
28	MT		
29	DS		
30	VS		
31	EP		
32	TN		
33	CX		
34	CU		
35	CD		
36	VK		
37	AL		
38	NC		

11
CCD

00	-I/O FNS	39	SAVEB
01	ABSP	40	SAVEK
02	ACAXY	41	SORTFL
03	ACLX		
04	ARCLE		
05	ARCLH		
06	ARCLI		
07	CLA-		
08	F/E		
09	INPT		
10	PMTA		
11	PMTH		
12	PMTK		
13	PRAXY		
14	PRL		
15	VIEWH		
16	XTOAH		
17	-ADV FNS		
18	A+		
19	A+B		
20	A-		
21	A-A		
22	DCD		
23	PC<>RTN		
24	PC>X		
25	PEEKB		
26	PEEKR		
27	PHD		
28	PLNG		
29	POKEB		
30	POKER		
31	PPLNG		
32	X>PC		
33	X>RTN		
34	XR>RTN		
35	-XF/M FNS		
36	GETB		
37	GETK		
38	MRGK		

11
REAL ESTATE

00	REAL EST 1B	39	R
01	$	40	CH
02	N	41	*EQ
03	*I	42	*RT
04	PV	43	OUT
05	PMT	44	SIZE?
06	FV	45	START
07	12*	46	CLK
08	12/		
09	BEG/END		
10	LIST		
11	CLFIN		
12	AMORT		
13	NPV		
14	IRR		
15	MIRR		
16	DEPR		
17	IPA		
18	GPMT		
19	WRAP		
20	SUMMARY		
21	EQ		
22	RENT		
23	*N		
24	I*		
25	*PV		
26	*PMT		
27	*FV		
28	*AMORT		
29	*AMT		
30	*SL		
31	*SOYD		
32	*DB		
33	*NPV		
34	*IRR		
35	*MIRR		
36	CALC		
37	SALE		
38	Y		

XROM

◆HEADER ◆MCODE ◆USERCODE

12
MACHINE

00	MACHINE 1A
01	CAM
02	HAR
03	CYC
04	PAR
05	DWECC
06	GEN4BAR
07	4BAR
08	CRANK
09	GEAR
10	SPUR
11	INV
12	SPRING
13	MOTION
14	COORD
15	POINTS
16	3POINTS
17	FCON
18	BCON
19	INI
20	*O
21	*IN
22	+360
23	*OUT
24	KEY

13
THERMAL

00	THRML 1A
01	KWONG
02	IDEAL
03	POLYTRP
04	ISNFLOW
05	FLOW
06	FLOW2
07	H2O
08	Re
09	ENERGY
10	HEATEX
11	A0
12	E0
13	ACNT
14	ECNT
15	APAR
16	EPAR
17	APRC
18	EPRC
19	ACRS
20	ECRS
21	BLKBODY
22	EbL
23	UNIT?
24	SZ?
25	INPUT
26	OUTPUT
27	KEY
28	-SI
29	SI-

14
NAVIGATION

00	NAVIG 1B
01	NA
02	*NA
03	STAR
04	*STAR
05	SUN
06	*SUN
07	*SUNL
08	*SUNU
09	MOON
10	*MOON
11	*MOONU
12	*MOONL
13	VENUS
14	*VENUS
15	P
16	MARS
17	*MARS
18	JUPITER
19	*JUPIT
20	SATURN
21	*SATUR
22	SRT
23	*SRT
24	JD
25	D+T
26	LBRYZX
27	ZYXdHA
28	DSPHAd
29	*IN
30	*IN1
31	DMT
32	DMS
33	HR
34	*T
35	FA
36	GST
37	BODY
38	*BODY
39	SIGHT
40	RM
41	DR
42	*DR
43	GCPOS
44	*GCPOS
45	GC
46	*GC
47	GCPLAN
48	*GCPLAN
49	GCPLOT
50	*GCPLOT
51	DSPP2
52	LOTOL
53	*LOTOL
54	DSPL
55	DSPLO
56	RLPOS
57	*RLPOS
58	RL
59	*RL
60	*M/a
61	*ec

15
PETROLEUM

00	PETROL 1A
01	Z
02	CG
03	CCG
04	BG
05	CBG
06	UG
07	CUG
08	TcPc
09	CTcPc
10	CCWA
11	PROP
12	WO
13	COMP
14	SOUR
15	IN
16	CGASG
17	CTPC
18	W1
19	CHV
20	CCB
21	W2
22	W3
23	CO
24	CCOb
25	CCO
26	COS
27	W4
28	W5
29	BO
30	CBOb
31	CBO
32	CBT
33	UO
34	CUOd
35	CUOb
36	CUO
37	CW
38	CCW
39	BW
40	CBW
41	UW
42	CUW
43	CPSAT
44	CFR
45	CCFR

XROM

◆HEADER ◆MCODE ◆USERCODE

16
PETROLEUM

00	RS
01	CRSb
02	Pb
03	CPb
04	BT
05	CBTb
06	RSW
07	CRSW
08	CT
09	CCT
10	CCTb
11	TITLE
12	W6
13	Y/N?
14	ITcPc
15	STDTP
16	SEPTP
17	T
18	P
19	GASG
20	OILG
21	IRS
22	RSI
23	%NACL
24	%POR
25	W7
26	W8
27	W9
28	X0
29	X1
30	X2
31	X3
32	X4
33	X5
34	X6
35	X7
36	X8
37	OUTK
38	OUTU
39	OUT
40	INK
41	INU
42	CZ
43	CCR
44	CON
45	INCON

17
PLOTTER

00	-PLOTTER 1A-
01	CLIPUU
02	CSIZE
03	CSIZEO
04	DGTIZE
05	DRAW
06	FRAME
07	GCLEAR
08	IDRAW
09	IMOVE
10	IPLOT
11	LABEL
12	LDIR
13	LIMIT
14	LOCATD
15	LOCATE
16	LORG
17	LTYPE
18	LTYPEO
19	LXAXIS
20	LYAXIS
21	MOVE
22	PEN
23	PENDN
24	PENUP
25	PINIT
26	PLOT
27	PLREGX
28	RATIO
29	RPLOT
30	SCALE
31	SETGU
32	SETUU
33	TICLEN
34	UNCLIP
35	WHERE
36	XAXIS
37	XAXISO
38	YAXIS

18
PLOTTER

00	-PLOTTER 2A-
01	PCLBUF
02	PDIR
03	PRCL
04	NEWPLOT
05	REPLOT
06	PLINIT
07	PLTUXY
08	PLANOT
09	Y?
10	X?
11	BC
12	BCA
13	BCAA
14	BCCKSM
15	BCO
16	BCP
17	BCREGX
18	BCSIZE
19	BCX
20	BCXS

19
CLINICAL LAB X

00	CLINLAB 1A
01	BEERS
02	BRA
03	CREAT
04	BLOOD
05	O2SAT
06	RCI
07	TBV
08	THY
09	RADCORR
10	RIA
11	BSTAT
12	CHI
13	TSTAT
14	TDIST
15	*
16	*0
17	*1
18	*2
19	*3
20	*4
21	*5
22	*6
23	*7
24	*8
25	*9
26	*A
27	*C
28	*CO
29	*CR
30	*CS
31	*F
32	*H
33	*HG
34	*IN
35	*I125
36	*I131
37	*P
38	*Pt
39	*PR
40	*R
41	*SE
42	*SR
43	*TC
44	*X
45	*XE
46	*YN

XROM

◆HEADER ◆MCODE ◆USERCODE

19 SECURITIES X

00	SECUR 1B
01	BONDS
02	STOCK
03	CALL
04	OPTION
05	HEDGING
06	BFLY
07	BULL
08	CSEC
09	CBOND
10	SPEC
11	PRC
12	YLD
13	ATY
14	ATP
15	JDAY
16	BEP
17	*BFLY
18	DBEP
19	UBEP
20	CONV

19 STRC ANAL X

00	STRCTB 1A
01	COLE
02	COLSI
03	TBEAM
04	ABEAM
05	CONCOL
06	ITCON
07	IRCON

19 AVIATION

00	AVIATIN 1A
01	FM
02	WB
03	CG
04	CLWB
05	PLAN
06	DP
07	WA
08	PERF
09	FLY
10	CLIMB
11	CRUISE
12	360+
13	WALT
14	DISP
15	IFW
16	HCW
17	1VOR
18	R2
19	2VOR
20	TAS
21	*T

20 PPC

00	SR
01	Sb
02	LR
03	SD
04	SK
05	RD
06	RK
07	BV
08	CV
09	IG
10	SV
11	FD
12	FR
13	DF
14	NP
15	GN
16	RN
17	BD
18	TB
19	PM
20	CM
21	CJ
22	JC
23	CA
24	LG
25	HA
26	HS
27	CP
28	MP
29	HP
30	BA
31	M2
32	M3
33	M1
34	BE
35	M4
36	M5
37	IR
38	DR
39	BM
40	BR
41	BX
42	BΣ
43	BC
44	UR
45	PR
46	S1
47	S3
48	S2
49	NS
50	NR
51	PO
52	Rb
53	AM
54	MA
55	SM
56	SE
57	XL
58	VF

21 SEA KING MK5

00	STAC ROM
01	TONE
02	CPA
03	L
04	K
05	DEPTH
06	ANGLE
07	UPD
08	TURN
09	HYFIX
10	UPDATE
11	HARM
12	TRACKEX
13	ESM
14	LESM
15	COG
16	VMX
17	CC
18	CBY
19	CST
20	CP
21	CW
22	CF
23	BRF
24	CLKEYS
25	LOGBOOK
26	BATHY
27	N
28	P
29	M
30	IF

XROM

◆HEADER ◆MCODE ◆USERCODE

21
ASSEMBLER 3

00	ASSEMBLER3	39	SXL
01	AND	40	SXR
02	OR	41	X>$
03	APPFN	42	X+Y
04	ASSEM	43	Y-X
05	DISASM	44	1CMP
06	A>X	45	2CMP
07	X>A	46	1-D
08	BCD>BIN	47	2-D
09	BIN>BCD	48	3-D
10	CF55	49	4-D
11	SF55		
12	CLROM		
13	CODE		
14	DECODE		
15	COMPILE		
16	COPYROM		
17	CVIEW		
18	VIEWA		
19	DISS		
20	GETPC		
21	PUTPC		
22	HEXKB		
23	INSBYTE		
24	RCLBYTE		
25	STOBYTE		
26	LOADP		
27	MLDL?		
28	NEXTFN		
29	NRCL		
30	NSTO		
31	PCWRT		
32	REG>ROM		
33	ROM>REG		
34	ROM>X		
35	X>ROM		
36	ROM?		
37	RXL		
38	RXR		

21
DEL.SERV.1

00	U
01	K
02	L
03	RN
04	DI
05	CC
06	2
07	Q
08	P
09	M1
10	RP
11	3
12	O
13	9
14	7
15	QF
16	3C
17	TC
18	SP
19	11
20	W
21	CLFL
22	CRFLD
23	GETX
24	SAVEX
25	SEEKPTA
26	V
27	CL
28	S

22
ADVANTAGE

00	-ADV CONV A	39	MRC-
01	BININ	40	MRIJ
02	BINVIEW	41	MRIJA
03	OCTIN	42	MRR+
04	OCTVIEW	43	MRR-
05	HEXIN	44	MS
06	HEXVIEW	45	MSC+
07	NOT	46	MSIJ
08	AND	47	MSIJA
09	OR	48	MSR+
10	XOR	49	MSWAP
11	ROTXY	50	MSYS
12	BIT?	51	PIV
13	-ADV MTRX	52	R<>R
14	C<>C	53	R>R?
15	CMAXAB	54	RMAXAB
16	CNRM	55	RNRM
17	CSUM	56	RSUM
18	DIM?	57	SUM
19	FNRM	58	SUMAB
20	I+	59	TRNPS
21	I-	60	YC+C
22	J+	61	MEDIT
23	J-	62	CMEDIT
24	M*M	63	MP
25	MAT*		
26	MAT+		
27	MAT-		
28	MAT/		
29	MATDIM		
30	MAX		
31	MAXAB		
32	MDET		
33	MIN		
34	MINV		
35	MMOVE		
36	MNAME?		
37	MR		
38	MRC+		

XROM

◆HEADER ◆MCODE ◆USERCODE

22
HP-IL DEVEL

00	-HP-IL DEV	39	PRFRMS
01	A-BUF	40	REN
02	A=BUF?	41	RFRM
03	A=BUFX?	42	RG-BUFX
04	AAD	43	RG=BUF?
05	AAU	44	RREG
06	AIPT	45	SAI
07	BSIZE?	46	SCOPE
08	BSIZEX	47	SDA
09	BUF-AX	48	SDC
10	BUF-RGX	49	SDI
11	BUF-XA	50	SF33
12	BUF-XB	51	SRQR?
13	CF33	52	SST
14	CMD	53	TAD
15	DDL	54	TCT
16	DDT	55	UNL
17	FRAV?	56	UNT
18	FRNS?	57	WFRM
19	GET	58	WREG
20	GTL	59	X-BUF
21	IDY	60	X=BUF?
22	IFCR?	61	X<>FLAG
23	IFC		
24	INBIN		
25	INBUFX		
26	LAD		
27	LPD		
28	MIPT		
29	MONITOR		
30	NRD		
31	NRE		
32	ORAV?		
33	OUTBIN		
34	OUTBINY		
35	OUTBUFX		
36	PT=		
37	PT?		
38	PRBYTES		

23
CLINICAL LAB X

00	-X MASS 1A	39	OUTXB
01	COPYFL	40	OUTP
02	DIRX	41	POLL
03	FLLENG	42	POLLD
04	FLTYPE	43	POLLE
05	MCOPY	44	POLLUNC
06	MCOPYPV	45	RCLSEL
07	MVERIFY	46	SRQ?
08	-X EXT FCN	47	STAT
09	ALENGIO	48	XFER
10	ANUMDEL	49	XFERC
11	ATOXL	50	XFERCL
12	ATOXR	51	XFERE
13	ATOXX	52	XFERN
14	XTOAL	53	-ADV CTL FN
15	XTOAR	54	ADROFF
16	X<>FIO	55	ADRON
17	YTOAX	56	DDL
18	-X CTL FNS	57	DDT
19	AID	58	LAD
20	CLRDEV	59	SEND
21	CLRLOOP	60	TAD
22	DEVL	61	UNL
23	DEVT	62	UNT
24	FINDAID		
25	ID		
26	INAC		
27	INACL		
28	INAE		
29	INAN		
30	INXB		
31	INP		
32	LOCK		
33	NLOOP		
34	NOTREM		
35	OUTAC		
36	OUTACL		
37	OUTAE		
38	OUTAN		

23
EXTENDED I/O

00	-X MASS 1A	39	OUTXB
01	COPYFL	40	OUTP
02	DIRX	41	POLL
03	FLLENG	42	POLLD
04	FLTYPE	43	POLLE
05	MCOPY	44	POLLUNC
06	MCOPYPV	45	RCLSEL
07	MVERIFY	46	SRQ?
08	-X EXT FNS	47	STAT
09	ALENGIO	48	XFER
10	ANUMDEL	49	XFERC
11	ATOXL	50	XFERCL
12	ATOXR	51	XFERE
13	ATOXX	52	XFERN
14	XTOAL	53	-ADV CTL FN
15	XTOAR	54	ADROFF
16	X<>FIO	55	ADRON
17	YTOAX	56	DDL
18	-X CTL FNS	57	DDT
19	AID	58	LAD
20	CLRDEV	59	SEND
21	CLRLOOP	60	TAD
22	DEVL	61	UNL
23	DEVT	62	UNT
24	FINDAID		
25	ID		
26	INAC		
27	INACL		
28	INAE		
29	INAN		
30	INXB		
31	INP		
32	LOCK		
33	NLOOP		
34	NOTREM		
35	OUTAC		
36	OUTACL		
37	OUTAE		
38	OUTAN		

XROM

◆HEADER ◆MCODE ◆USERCODE

24 ADVANTAGE

00	MATRX
01	MATR
02	-ADV MATH
03	SOLVE
04	INTEG
05	SILOOP
06	SIRTN
07	Z↗N
08	MAGZ
09	e↗Z
10	LNZ
11	Z↗1/N
12	SINZ
13	COSZ
14	TANZ
15	a↗Z
16	LOGZ
17	Z↗1/W
18	Z↗W
19	C+
20	C-
21	CINV
22	C*
23	C/
24	PLY
25	RTS
26	DIFEQ
27	CFIT
28	AΣ
29	DΣ
30	BFIT
31	FIT
32	Y?X
33	SZ?
34	VC
35	CROSS
36	VS
37	VR
38	DOT
39	VE
40	V-
41	V+
42	VXY
43	UV
44	V∡
45	VD
46	V*
47	TR
48	CT
49	AIP
50	-ADV TVM
51	TVM
52	N
53	PV
54	PMT
55	FV
56	*I

24 HP-IL DEVEL

00	---
01	AND
02	ASIZE?
03	A-XL
04	A-XR
05	A-XX
06	BININ
07	BINVIEW
08	BIT?
09	HEXIN
10	HEXVIEW
11	NOT
12	OR
13	OCTIN
14	OCTVIEW
15	ROMCHKX
16	ROTXY
17	XOR
18	X-AL
19	X-AR
20	Y-AX

25 XF

00	-EXT FCN 2D
01	ALENG
02	ANUM
03	APPCHR
04	APPREC
05	ARCLREC
06	AROT
07	ATOX
08	CLFL
09	CLKEYS
10	CRFLAS
11	CRFLD
12	DELCHR
13	DELREC
14	EMDIR
15	FLSIZE
16	GETAS
17	GETKEY
18	GETP
19	GETR
20	GETREC
21	GETRX
22	GETSUB
23	GETX
24	INSCHR
25	INSREC
26	PASN
27	PCLPS
28	POSA
29	POSFL
30	PSIZE
31	PURFL
32	RCLFLAG
33	RCLPT
34	RCLPTA
35	REGMOVE
36	REGSWAP
37	SAVEAS
38	SAVEP
39	SAVER
40	SAVERX
41	SAVEX
42	SEEKPT
43	SEEKPTA
44	SIZE?
45	STOFLAG
46	X<>F
47	XTOA
48	-CX EXT FCN
49	ASROOM
50	CLRGX
51	ED
52	EMDIRX
53	EMROOM
54	GETKEYX
55	RESZFL
56	ΣREG?
57	X=NN?
58	X≠NN?
59	X<NN?
60	X<=NN?
61	X>NN?
62	X>=NN?

26 TIME

00	-TIME 2C
01	ADATE
02	ALMCAT
03	ALMNOW
04	ATIME
05	ATIME24
06	CLK12
07	CLK24
08	CLKT
09	CLKTD
10	CLOCK
11	CORRECT
12	DATE
13	DATE+
14	DDAYS
15	DMY
16	DOW
17	MDY
18	RCLAF
19	RCLSW
20	RUNSW
21	SETAF
22	SETDATE
23	SETIME
24	SETSW
25	STOPSW
26	SW
27	T+X
28	TIME
29	XYZALM
30	-CX TIME
31	CLALMA
32	CLALMX
33	CLRALMS
34	RCLALM
35	SWPT

XROM

◆HEADER ◆MCODE ◆USERCODE

27 WAND

00	- WAND 1F -
01	WNDDTA
02	WNDDTX
03	WNDLNK
04	WNDSUB
05	WNDSCN
06	WNDTST

27 EXT IL

00	EXTILROM
01	CLRBUF
02	DIRLEFT
03	DIRSIZE
04	NAMEMED
05	READBUF
06	READCAL
07	READXM
08	RECLEFT
09	SCOPYFL
10	SCREATE
11	SDIR
12	SNEWM
13	SWRTA
14	SWRTK
15	SWRTP
16	SWRTPV
17	SWRTS
18	WRTBUFX
19	WRTCAL
20	WRTXM
21	PRINTFCNS
22	ATOBUFX
23	MCLIST
24	MCPRP
25	PRTAID
26	SACA
27	X>AR

28 HP-IL

00	-MASS ST 1H
01	CREATE
02	DIR
03	NEWM
04	PURGE
05	READA
06	READK
07	READP
08	READR
09	READRX
10	READS
11	READSUB
12	RENAME
13	SEC
14	SEEKR
15	UNSEC
16	VERIFY
17	WRTA
18	WRTK
19	WRTP
20	WRTPV
21	WRTR
22	WRTRX
23	WRTS
24	ZERO
25	–
26	-CTL FNS
27	AUTOIO
28	FINDID
29	INA
30	IND
31	INSTAT
32	LISTEN
33	LOCAL
34	MANIO
35	OUTA
36	PWRDN
37	PWRUP
38	REMOTE
39	SELECT
40	STOPIO
41	TRIGGER

29 PRINTER

29 IR PRINTER

00	-PRINTER 2E
01	ACA
02	ACCHR
03	ACCOL
04	ACSPEC
05	ACX
06	BLDSPEC
07	LIST
08	PRA
09	PRAXIS
10	PRBUF
11	PRFLAGS
12	PRLEYS
13	PRP
14	PRPLOT
15	PRPLOTP
16	PRREG
17	PRREGX
18	PRΣ
19	PRSTK
20	PRX
21	REGPLOT
22	SKPCHR
23	SKPCOL
24	STKPLOT

00	-PRINTER 3B
01	ACA
02	ACCHR
03	ACCOL
04	ACSPEC
05	ACX
06	BLDSPEC
07	LIST
08	PRA
09	PRAXIS
10	PRBUF
11	PRFLAGS
12	PRLEYS
13	PRP
14	PRPLOT
15	PRPLOTP
16	PRREG
17	PRREGX
18	PRΣ
19	PRSTK
20	PRX
21	REGPLOT
22	SKPCHR
23	SKPCOL
24	STKPLOT
25	FMT
26	--
27	DELAY
28	MAN
29	MAPOFF
30	MAPON
31	NORM
32	PRTOFF
33	PRTON
34	RESETP
35	STARTU
36	STOPU
37	TESTP
38	TRACE

XROM

◆HEADER ◆MCODE ◆USERCODE

30
CARD READER

00	CARD READER
01	MRG
02	RDTA
03	RDTAX
04	RSUB
05	VER
06	WALL
07	WDTA
08	WDTAX
09	WPRV
10	WSTS
11	7CLREG
12	7DSP0
13	7DSP1
14	7DSP2
15	7DSP3
16	7DSP4
17	7DSP5
18	7DSP6
19	7DSP7
20	7DSP8
21	7DSP9
22	7DSPI
23	7DSZ
24	7DSZI
25	7ENG
26	7FIX
27	7GSBI
28	7GTOI
29	7ISZ
30	7ISZI
31	7P<>S
32	7PRREG
33	7PRSTK
34	7PRTX
35	7RCLΣ
36	7SCI

31
DEL.SERV.1

00	SR
01	Z
02	US
03	T
04	2C
05	M
06	SG
07	6
08	R
09	N
10	5
11	I1
12	1
13	L1
14	X
15	Y

31
RAMBOX32

00	-RAMBOX 32
01	BUFLNG?
02	CLLSTFL
03	CLPG
04	CLRFL
05	COPYPG
06	CRDIR
07	CRFLBUF
08	CRFLDTA
09	CRFLKEY
10	ENDPG
11	FNC?
12	FRBYT?
13	GTBUF
14	GTKEY
15	GTREG
16	GTREGX
17	GTREGXY
18	INITPG
19	KEYAS?
20	LDBUF
21	LDKEY
22	LDPGM
23	LDREG
24	LDREGX
25	LDREGXY
26	PG?
27	PGSUM
28	PTCT
29	READPG
30	SETPRV
31	UNPTCT
32	WRTPG
33	XQ>XR

31
CO-OP MODULE

00	CO-OP ROM
01	SC
02	S
03	BA
04	TRAVRS
05	SIDS
06	O
07	R
08	∡ LT
09	W
10	∡ RT
11	DEF-LT
12	DEF-RT
13	ZS
14	ZSL
15	VSLOP
16	ZSLOP
17	STORE
18	SMM
19	INC
20	SEM
21	Y
22	MS
23	LOAD
24	SAVE
25	PREM
26	CONTOUR
27	SEARCH
28	CARD
29	K
30	KT
31	KB
32	N-E
33	S-E
34	S-W
35	N-W
36	CH
37	ABO
38	S1
39	S2
40	S3
41	S4
42	NN
43	ID
44	IE
45	IT
46	LR
47	IR
48	CURV
49	*D
50	AUTO
51	REP
52	PPTS
53	CRIS

BUFFER ID's

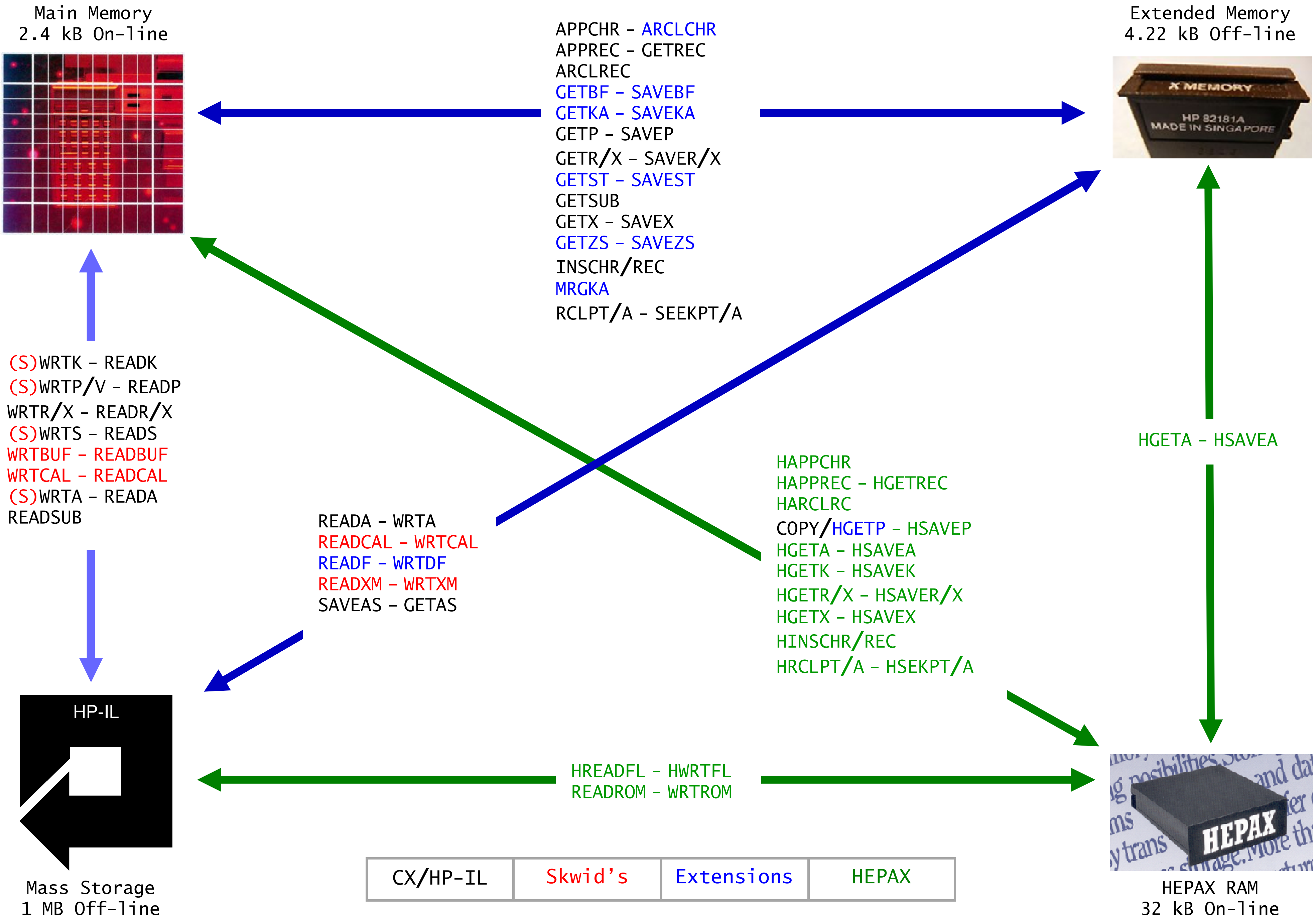
ID	Module	Function
1		
2		
3		
4		
5	CCD	WSIZE / SEED
6		
7	PK PROG	RTN
8		
9		
10	TIME	ALARM
11	PKROM1	KEY
12	CMT / HPIL-DEV	I/O
13		
14		

See PK Collection Manual for an updated list

Bytes pr. function

Bytes	Function
1	AOFF, AON, etc
2	XROM
	ASTO nn, ARCL nn, etc
3	END
	GTO 15 - GTO 99
	XEQ 00 - XEQ 99
	GTO A - GTO J
4	LBL alpha (+alpha)

System Memory Map © Ángel M. Martin - May 2014



Main Memory
2.4 kB On-line

Extended Memory
4.22 kB Off-line

APPCHR - ARCLCHR
APPREC - GETREC
ARCLREC
GETBF - SAVEBF
GETKA - SAVEKA
GETP - SAVEP
GETR/X - SAVER/X
GETST - SAVEST
GETSUB
GETX - SAVEX
GETZS - SAVEZS
INSCHR/REC
MRGKA
RCLPT/A - SEEKPT/A

(S)WRTK - READK
(S)WRTP/V - READP
WRTR/X - READR/X
(S)WRTS - READS
WRTBUF - READBUF
WRTCAL - READCAL
(S)WRTA - READA
READSUB

READA - WRTA
READCAL - WRTCAL
READF - WRTDF
READXM - WRTXM
SAVEAS - GETAS

HAPPCHR
HAPPREC - HGETREC
HARCLRC
COPY/HGETP - HSAVEP
HGETA - HSAVEA
HGETK - HSAVEK
HGETR/X - HSAVER/X
HGETX - HSAVEX
HINSCHR/REC
HRCLPT/A - HSEKPT/A

HGETA - HSAVEA

HREADFL - HWRTFL
READROM - WRTROM

HP-IL

Mass Storage
1 MB Off-line

HEPAX RAM
32 kB On-line

CX/HP-IL	Skwid's	Extensions	HEPAX

System Memory Map © Ángel M. Martin - May 2014

Main RAM

7P<>S
CLMM
CLP
CLRAM
CLRG/X
CLS
CLST
CLX
ENTER^
PCLPS
COPY
LASTX
NRCLX - NSTOY
PC>X - X>PC
PC<>RTN
PACK
PEEKR - POKER
R↗ - RDN
RAMED/EDIT
REGMOVE/SWAP
RCLBM - STOBM
ST>RG - RG>ST
ST<>RG _ _
ST>S - S>ST
ST<>S
(A)STO - (A)RCL
STOFLAG - RCLFLAG
SWAPR
(P)SIZE
X<>
X<>BM
X<>F
X<>Y
X<I>Y

ALPHA

A<>RG _ _
A<>ST
A>ST/ST>A
A>RG - RG>A
ABSP
AINT/AIP/ARCLI
ALENG/X/XY
ANUM/DEL
AREV
ASHF/X - AROT/X
ASWAP/USWAP
AXEQ?
(L)ATOX - XTOAL
CLA/-/C/?
LADEL/X
LEFT$ - MID$ - RIGHT$
LOW$ - UPR$
M<>N/O/P
N<>O/P
O<>P
POSA
PRESP/FX - POSTSP/FX
RATOX - XTOA(R)
REMZER
XATOX - ASTOXX
YTOAX/ASUB

I/O Area

ARCLBF - ASTOBF
B?/BUF?
BF>RGX - RGX>BF
BF>ST - ST>BF
BFHD
BHEAD
BFLNG
BRFCL - BFSTO
CLB/DELBUF
CLRBF
CRBUF
KACLR
KALNG
KAPCK
REIDBF
RESZBF

General XM

CLEM/CLXM
FLCOPY
RENMFL
RETPFL
FLDH
FLSIZE
PURFL
RESZFL
WORKFL

GETBF - SAVEBF
GETKA - SAVEKA
GETST - SAVEST
GETZS - SAVEZS
MRGKA

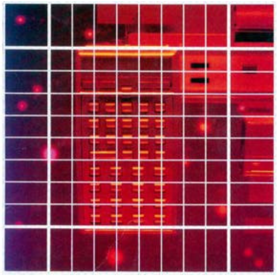
PRGM/DATA/ASCII

GETP - SAVEP
GETSUB
RSTCHK
XQXM

CRFLD
CLFL
GETR/X - SAVER/X
GETX - SAVEX
RCLPT/A - SEEKPT/A
WRTDF - READF

APPCHR - ARCLCHR
APPREC - GETREC
ARCLREC
ASROOM
CRFLAS
DELCHR/REC
ED
GETAS - SAVEAS
INSCHR/REC
POSFL
RCLPT/A - SEEKPT/A

Main Memory
2.4 kB On-line



Extended Memory
4.22 kB Off-line

System Memory Map © Ángel M. Martin - May 2014

-HEPAX 1F”

- CLRAM
- CODE
- COPYROM
- DECODE/YX
- DISASM
- HAPPCHR/REC
- HARCLRC
- HASROOM
- HCLFL
- HCRFLAS
- HCRFLD
- HDELCHR/REC
- HEPDIR/X
- HEPROOM
- HEXEDIT
- HFLADR
- HFLSIZE
- HFLTYP
- HGETREC
- HINSCHR/REC
- HPOSFL
- HPROMPT
- HPURFL
- HRCLPT/A
- HRENAME
- HSAVEA - HGETA
- HSAVEK - HGETK
- HSAVEP - HGETP
- HSAVER/X - HGETR/X
- HSAVEX - HGETX
- HSEC
- HSEKPT/A
- HUNSEC
- HWRKFL
- HWRTFL - HREADFL
- PRIVATE
- RAMTOG
- READROM - WRTROM
- RLSRAM

XF(A)

- *ALENG*
- *ANUM*
- *AROT*
- *ATOX*
- *CLEM*
- *CLKEYS*
- *CLRGX*
- *GETKEY*
- *GEYTKEYX*
- *PASN*
- *PCLPS*
- *POSA*
- *PSIZE*
- *RCLFLAG*
- *REGMOVE*
- *REGSWAP*
- *RTN?*
- *SIZE?*
- *SREG?*
- *STOFLAG*
- *X#NN?*
- *X<=NN?*
- *X<>F*
- *X<NN?*
- *X=NN?*
- *X>=NN?*
- *X>NN?*
- *XTOA*
- *XQ>GO*
- *XQXM*

HEPAX(A)

- *AND*
- *BCAT*
- *BCD-BIN*
- *BIN-BCD*
- *CTRAST*
- *DELETE*
- *INSERT*
- *NOT*
- *OR*
- PGROOM
- PGSUM
- *ROTYX*
- *SHIFTYX*
- *X-$*
- *X+Y*
- *XOR*
- *Y-X*

RAMBOX

- BUFLNG?
- CLLSTFL
- CLPG
- CLRFL
- COPYPG
- CRDIR
- CRFLDTA
- CRFLBUF
- CRFLKEY
- ENDPG
- FNC?
- FRBYT?
- INITPG
- KEYAS?
- LDPGM
- LDREG/X - GTREG/X
- LDREGXY - GTREGXY
- LDKEY - GTKEY
- LDBUF - GTBUF
- PG<>
- PG?
- PGSUM
- PTCT
- PG01/10
- SETPRV
- UNPTCT
- WRTPG - READPG
- XQ>XR

HEPAX RAM
32 kB On-line

RAMBOX
16 kB On-line /
16 kb Off-line

W&W SOFTWARE PRODUCTS GMBH

ProfiSET

CLDF
CLEM
CLPR
CLRSU
CLSEC
CRDF
DELFL
DF+/-/*//
DPTDF
EXALL
EXEM
EXREG/X
EXST
FIND/I
FLDIR
INIDATA
INIPRGM
IPTDF
LOADP
MPTDF
RCLALL
RCLDF/X
RCLEM
RCLIDF
RCLREG/X

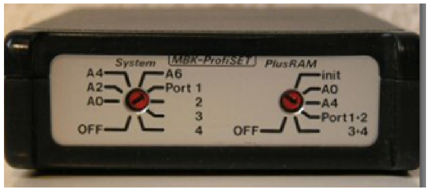
RCLST
READRAM
RENFL
RESDF
RLEFT?
RPTDF
RSTDF/X
SETSEC
SPTDF
STOALL
STODF/Y
STOEM
STOREG/X
STOST
SZ?DF
WRTRAM
X<>DF/Y
X=DF?
X<>DF?
X<DF?
X<=DF?
X>DF?
X>=DF?
X=PT?

ACHR/REC
AFLD
ALEN?
ASRM?
A=F? - A<>F?
A<F? - A>F?
A<=F? - A>=F?
CLAF
CRAF
DCHPT
DCHR/REC
DFLD
DRCPT
EDT
FINDPS
FLEN?
GCHR/REC
GFLD
ICHPT/R
IFLD
IRCPT
IREC
MCHPT
MRCPT
NAMAF/DF

NFLD?
NREC?
NUMREC
OUTCHR/REC
OUTFLD
RCHPT/R
RESAF
RFLD
RLEN?
RRCPT/REC
RSTAF
SCHPT
SFPT
SRCPT
ST<>DF
SWPREC
SZ?AF
X<>CHR
?EOF

EP=BL?
EP=FF?
EPROM?
PROM
PBL
CLEAR
CLRN
COPYBL
IDBL
INIT
NAMEBL
NEWBL
NEXTBL
PACKBL/KC
REVBL
SLCT
SUMBL
ADR>ID
BL?/A?/1A?
BLCAT/X
BLF?
CATF?
CATS?
ID?

LENG?
CHKBL
ID>ADR
REV?
SRV
TYP?
LADEP
PRVBL
PRVP
XROMBL/P
DATE2/5/7/N
DATI2/4/7/9
X>DATY
GETBL - SAVEBL
GETF - SAVEF
CLRS
EPACK
J?
L*/-/=
LADEK
LADES
N?
PKEY
PRM



Revision 2.07
April 27th 2015