

Program Area Format

1

40	00
----	----

3

nn	nn
----	----

 * Entries, Symbol Tbl

5

--	--

7*

--	--

 Program Length Less Initial 12 bytes

9

11*

--	--

 Next avail Symbol table insert pt
length of Symbol Table

13

--

14*

--	--

 Length of Prog + Symbol Tbl { #7 + #11 => #14

16

8

 Program Name - space filled

24

--

 Function Table # Entries - May be zero
Each = 3 bytes

--	--	--

stmt #

Letter Index (A=0 Z=25)

Program Area

program
stmts

Trailer

05 FF FF 34 F2 System Supplied End Stmt
03 00 00 Trailer

Symbol Table

--

 1 byte length

(main Symbols)

⚡

 Symbol

∞

 End of table

Symbol Addenda

FF	FE
----	----

next entry
pos

Long Function

--

 def

⚡

 name

nn	nn
----	----

 Statement # of def

Statement Format

--

 1 byte - length

--	--

 2 bytes - Stmt #

--

 1 byte - op code

} details

F2

 End of Stmt Flag

E ϕ xx xx Stmt # Literal

EB xx Integer

E3 ll x - x Text literal (Short)

CA 00 01 Symbol Table Reference - Numeric
(to π) A or A%

CE 00 01 Symbol Table Ref - String

C8 00 01 STR - numeric - source

F4 00 03 31 30 30 long Text literal
length start literal

CC 00 01 ~~Program Name~~ ^{STR} ~~literal~~ - A \$

CE 00 01 store STR String

AB 00 00 01 Format Reference

A9

EB ⁽⁴⁾ xx xx xx xx Long Integer

E9

EA (7 bytes) v. Long Integer or float
integer w/ implied decimal pt

14 digit * 9 - 9 = 5 bytes long

Integer value		
Exp		
00 = 0	01 = 10^1	+127
FF = 1	02 = 10^2	-128
FE = -2	03 = 10^3	

C9 xx xx

User Function Symbol Table Ref

A1 xx xx

Statement Label Table Reference

Arithmetic Operators

5E +

5B / divide

5D - Subtract

60 - negate (ie, $A = -B$)

5C * multiply

5F + String Concatenation

5A ↑ Exponentiation ($**$ or \uparrow)

54 = Equals Test Compare for Equal

56 <> Compare <>

55 >

59 <

53 = All "A"

8D B1 LIKE

8D BA Test as Logical (AND)

8D BA 00 06 AND

8D BA 40 05 OR

36 GOTO

GOTO

E4 00 14
Start & Literal

F2

37 GOSUB

GOSUB

E4 00 0A
Start & List

F2

01 GOSUB

02 START

EB 2D

F2

Integer
& Pages

03 RELEASE

F2

04 RUN

08 LOAD

E3 03 41 42 43
↑
text

05 CALL

E3 01 41 F2

06 ENTER

F4 F2

06 ENTER
A

F4 CA 00 01
Symbol
Tbl Ref

06 ENTER
A, A

F4 CA 00 01 F4 CA 00 02

~~49 LET~~

07 EXIT

07 EXIT EB 05 Exit 5

08 SETTRACE

~~F3-11.~~

09 Endtrace.

0A Escape

0B LOAD

0C Delete E4 0001 E0 FFFE

0D EDIT F4 00 00
+not lateral of command

0E MERGE EB 01 E1 Merge (1)

0F { LIST E4 0001 E4 FFFE
LIST

10 SAVE

11 PROGRAM

12 INDEXED

13 SERIAL

14 SORT

15 DIRECT

16 SYSTEM

17 FILE

18 ERASE

19 ADD

1A ADDR

1B DRDP

1C ENABLE

1D DISABLE

1E RESERVE

20 OPEN

21 CLOSE

IF IGNORE (unused)

E4 01
channel

E1

CC 0001

OPEN(1) A#

end of () list

P "A", 1, 1, 1

22 LOCK

23 UNLOCK

24 PUT

25 PRINT

$\overbrace{EB \ 01 \ E1}^{\text{channel}} \ F4 \ F2$
EOS

26 WRITE

$F4 \ F1 \ \overbrace{E-3 \ 01 \ 41}^{\text{"A"}} \ \text{WRITE "A"}$

27 REMOVE

$\overbrace{EB \ 01 \ E1}^{\text{CHAN}} \ \overbrace{CC \ 00 \ 01}^{\text{A\$}} \ BB \ \text{KEY=}$

28 ~~REMOVE~~ GET

29 INPUT

$F4 \ F0 \ \overbrace{CE \ 00 \ 01}^{\text{A\$}}$

2A READ

(1, KEY = A\$)
2B EXTRACT $\overbrace{EB \ 01 \ E1}^{\text{chan}} \ \overbrace{CC \ 00 \ 01}^{\text{A\$}} \ BB \ F4 \ \overbrace{CE \ 00 \ 01}^{\text{key=}} \ \text{End of (L)}$

2C FIND

2D RENAME

2E FLOATING POINT

2F PRECISION

30 RESET

31 CLEAR

32 BEGIN

33 STOP

34 END

35 ON/GOTO/GOSUB

36 ?GOTO

37 GOSUB
38 SETERR
39 SETESC
3A FOR (for/next)
3B EXIT TO
3C RETURN
3D NEXT
3E RETRY
3F EXECUTE
40 REM
41 DIM
42 TABLE
43 IOLIST
44 DEF FN_x
45 SETTIME
46 SETDAY
47 WAIT
48 OR IF
49 LET
4A INPUTRECORD
4B READRECORD
4C EXTRACTRECORD
4D FINDRECORD
4E PRINTRECORD
4F WRITERECORD

50 (
51)

8E Extended Op Code

8E - Extended Codes

- 50 ENCRYPT
- 51 PSAVE
- 52 BOOT
- 53 SYNC
- 54 PREAD
- 55 PREADRECORD
- 56 PEXTRACT
- 57 PEXTRACTRECORD
- ~~58~~ 58 IF
- ~~59~~ 59 INPUT
- 59 INPUT
- 5A INIT FILE
- 5B ?
- 5C *blw memory*
- 5D SHORTVAR
- 5E LONGVAR
- 5F INPUT EDT
- 60 INPUT EDT RECORD
- 61 WINDOW
- 62 MSORT
- 63 ADDSORT
- 64 REMSORT
- 65 <cd>
- 66 DROP ALL
- 67 WHILE
- 68 WEND

Below 50, acts like normal Codes

- 69 ESC ON
- 6A ESC OFF
- 6B
- 6C
- 6D
- 6E TISAM
- 6F EDITF
- 70 SET PREFIX
- 71 SET DIR
- 72 TEXT
- 73 SET DATE STRINGS
- 74 FIXUP
- 75 SYMTAB
- 76 ~~FORMAT~~ FCREATE
- 77
- 78 DUMP
- 79 SET DRIVE
- 7A { SET TRACEMODE
 STM
- 7B ESCAPE WHEN
- 7C XCALL
- 7D FORMAT INCLUDES
 - INCLUDE EB00
 - DELETE EB 03
 - INIT EB 01
 - DEFAULT EA 02
- 7E SET PRM
- 7F LOG OPEN
- 80 LOG CLOSE

8E Extended Op Codes

pg 2

- 81 TRANSACTION BEGIN
- 82 COMMIT
- 83 ROLLBACK
- 84 LET FMD
- 85 SET DATE MASK
- 86 SET CMASK
- 87 SET HOTKEY
- 88 SET ERR ON
- 89 " " OFF
- 8A LET FMT
- 8B PACK ARDAY
- 8C UNPACK ARDAY
- 8D END ESCAPE WHEN
- 8E GLOBAL ESCAPE WHEN
- 8F SET ERC
- 90 CLEAR ERC

Result type
N S E C U
Functions

Function's

Op	CD	Function
	61	ABS
8D	95	ACS Arc Cosine
	53	= ALL "x"
	62	AND
8D	CD	ARG
8D	CE	ARGE
	63	ASC
8D	96	ASN - Arc Sine
	64	ATH Ascii to hex
8D	93	ATN - Arc Tangent
8D	94	ATQ - Arc Tan Quo
8D	DA	ATR - Attribute
	65	BIN
		BSZ (not rec'd)
8D	AD	CDN
8D	AE	CD S
8D	C5	CGV
	67	CHR
8D	DD	CMASK
8D	91	COS
	68	CPL Compile Line
8D	CA	CPP Compile Program
	69	CRC Cyclic Red Code
	92	CTL
8D	9C	CVT
8D	DC	DATEMASK
8D	CB	DATESTRINGS

90 DAY	8D.C9 User FN (long)
8D.B4 DCM	6E FPT
6A DEC	8D.CB FST
8D.A7 DIM	6F GAP
8D.C4 DIR	71 HSH
8D.D6 DNE	72 HTA
86 DSD	73 IND
8D.CF DSK	8D.D0 INF
94 DSZ	74 INT
8D.AA DTN	75 IOR
8D.BC DTR	76 KEY
6B EPT	77 LEN
8D.E1 ERC	8D.9A LKY
8D.CB GRM	89 LOG
93 ERR	78 LRC
8D.D8 ERRBUF	79 LST
99 ESC	8D.B3 MAX
8C EXP	8D.B2 MIN
8D.B6 FDT	8D.9F MNE
6D FID	7A MOD
8D.97 FIX	8D.B8 NEA
8D.9B FKY	8A NLG
8D.DB FMD	8D.C7 NMV
8D.DF FMT	7B NOT
8D.D9 FMTNL	
8F.00 / F0F8 FNx	A=0 B=1
8F.80 / F08F FNxb	

1A9 NTD
7C NUM
3D.04 NUM (NTP=)
9D OCH
D.C1 PAD
87 PCK
D.9D PFL
3D.9E PFP
8D.D1 PGCHARBASE
84 PGM
9B PGN
3D.99 PKY
7D POS
9F PRC
8D.B9 PREFIX
8D.D5 PRM
95 PSZ
8D.B7 PTN
7E PUB
98 QVO
8D.98 RND
8D.BD RTD
8D.CC SDX
9A SEP
7F SGN
8D.90 SIN
8B SQR
96 SSN

83 SSZ
8D.BB STL
80 STR
8D.D3 STR (NTP,SIZE)
8D.BE SWP
9C SHS
8D.92 TAN
8D.BF TBL
85 TCB
91 TIM
8D.D2 TRACEMODE (S)
81 TSIC
97 TSM
8D.B5 UCM
8D.E0
95 UNT (JH)
9E UNT
88 UPK (N)
8D.B4 WIN (GET) EB03 EB01 8DB0
8D.B4 WIN (GETCURSOR) EB01 EB01 8DB0
8D.B4 WIN (GETLIST) EB00 EB01 8DB0
8D.B4 WIN (GETSAVEDLIST) EB03
8D.B4 WIN (GETSCREEN) EB04
8D.AC XFD
82 XOR

STATEMENT IN ERROR

F3 14 00 OF Stmt text
Fixed

Continuation Stmt

Instead of F2 @ End

Use E7 at end of 1st Stmt

In let Stmt,

Separate let segments w/ EC

System Functions

95 PSZ

94 DSZ

8D Extended Functions

7C NUM

6A DEC

Extended Functions 8D

AD

CDN

AE

CDS

BF ATR =

BB TIM =

BZ DOM =

B1 Prior is ~~OPT~~

BE OPT =

B4 BE - Prior is DEV =

BC Prior Arg is
Error Jump

B4 Prior Arg is
Error Jump

B7 Prior is ISZ =

B8 KEY =

DUM(AS, ZRR = 100)
CC0002 Exp 0004 BC 7C
↑
err jump

OPEN(1, ZRR = 100) A#
EB 01 E1 EQ 00 64 B4 CC0001
↑ channel ↑

F0 Parameter Variable Name Follows { pre Variable, can be Input or Output }

eg: F0 CA 00 01

masked by 2nd Variable

F0 EA 00 01 CE 00 02

↑
numeric

↑ string var

no Fx value implies masking

F1 Function (Output only)

~~Parameter Variable~~ ~~Refers~~ ~~Masked~~ ~~By~~ ~~Literal~~

eg F1 CA 00 01 E3 05 # # 00

↑
no Fx value implies masking

F4 input start, output ; end if options ()

F5 List Follows (Start List)

eg: F5 F0 CE 00 01

F5 F0 CA 00 01

F7 Screen Psn Function, eg (2 vars)

IDLIST @ (10, 20), A#

43 F5 F1 EB 0A EB 14 F7 F0 CG 00 01
↑ ↑
IDLIST List 10 20 C A#

F6 Line Psn Function, eg (1 var)

IDLIST @ (20), A#

43 F5 F1 EB 14 F6 F0 CG 00 01
↑ ↑
var 20 1C A#

F8 ~~Force End of list~~ End of Function Call Param List