

6809

fig-FORTH

ASSEMBLY SOURCE LISTING

RELEASE 1

WITH COMPILER SECURITY

AND

VARIABLE LENGTH NAMES

v 1.0

JUNE 1980

This public domain publication is provided through the courtesy of the FORTH Interest Group. Further distribution must include this notice.

FORTH INTEREST GROUP P.O. Box 1105 San Carlos, Ca. 94070

TTL (C)1980 TALBOT MICROSYSTEMS
STTL 68`FORTH for 6809 : FIG MODEL
OPT PAG,NOC,MAC,NOE

*
*
*** FORTH FOR 6809 by R. J. Talbot, Jr. 80.03.20
*
*** TALBOT MICROSYSTEMS

* This version of FORTH follows the model created by the
* The FORTH Interest Group (FIG)
* PO Box 1105, San Carlos, CA 94070
* (415) 962-8653
* The model is described in a document which may be obtained from
* them for \$15.00 entitled "fig-FORTH Installation Manual"
*
* This version was developed for a SWTPC 6809 system with FLEX, but
* all terminal I/O is done by internal code, so it is independent
* of the rom monitor or operating system such as FLEX.
* The only system dependent terminal I/O code which might need
* changing is the location of the control ACIA port in memory
* space -- the present assignment is to E004 and the data word is
* the control address + 1.
*
* All terminal I/O is done in three assembly language subroutines:
* PEMIT - emits a character to terminal
* PKEY - reads a character from terminal (no echo)
* PQTERM - tests terminal for a character having been typed
*
* The FORTH words for disk I/O follow the model of the FORTH
* Interest Group - there are both a RAM simulation of disk I/O and real
* disk I/O of standard FORTH SCREENS. Also, there is an interface
* which allows input or output using DOS format TEXT files, and
* there is a link to the DOS command structure so that
* DOS commands may be executed from FORTH, including read into
* or write from RAM simulated disk using TAPE or DISK SAVE or LOAD.
*
* This 68`FORTH Vers 1.1 assembled machine code program is available on
* a FLEX 9.0 soft-sectored 5-1/4" diskette or
* on a 300 baud KCS cassette from TALBOT MICROSYSTEMS.
* The cassette version may be used in conjunction with the
* RAM simulation of disk to implement a cassette-only version or to
* modify the DOS interface to something other than FLEX.
*
* Advanced versions are available (in
* diskette form only) which contains a full 6809 assembler in FORTH,
* a screen oriented FORTH source text editor, and many other
* useful vocabularies -- contact TALBOT Microsystems.
*
* This assembly source code is available (on FLEX 9.0 soft sectored
* 5 1/4" diskette only) -- contact TALBOT Microsystems.
*

PAG

* MEMORY MAP
* addr contents pointer init by
* **** *****
* 0000 COLD start entry point
* 0003 Warm start entry point
*
* 0006 start of FORTH KERNEL
* COLD startup parameters, WARM startup parameters
* common system variables
* start of FORTH code
* register Y <== IP ABORT
* (W = X after LDX ,Y++ at NEXT) <== W
*
* 1BEF end of FORTH KERNAL dict links to FORTH further up.
* 2000 -NBLK*(BUFSIZ+4) FIRST,VIRBGN
* NBLK buffer sectors of VIRTUAL MEMORY
* initialized with NBLK=4 so VIRBGN = 1BFO
* 2000 VIREND
* registers and pointers for FORTH
* 2020 USER #1 table of variables <== UP UPINT
*
* 2050 "FORTH" (a word) <= <==CONTEXT
* `=====CURRENT
* 207E "TASK" (a word marking end of dict.)
* 2XXX <== DP DPINIT
* 2XXX dictionary grows |
* up |
* |
* towards higher memory
* |
* towards lower memory
* down |
* 2F30 DATA stack grows | register U <== SP SPO,SINIT
* 2F30 <== IN TIB
* INPUT LINE BUFFER
* holds up to 132 characters and
* is scanned upward by IN starting
* at TIB
* 2FB4
* 3000 RETURN stack base register S <== RP RINIT
* LO,DSMBGN
* space to simulate a disk mass memory
* 4000 HI,MENTOP
* TOPMEM

PAG

0004	NBLK	SET	4	# of disc buffer blocks for virtual memory
0100	BUFSIZ	EQU	256	# of bytes per disk sector
0000	PRGBGN	EQU	\$0000	beginning of FORTII program, COLD entry point, * WARM entry point is PRGBGN + 3
1BFO	VIRBGN	SET	VIREND-NBLK*(BUFSIZ+4)	assigns space for 4 BUFFERS
2000	VIREND	EQU	\$2000	end of virtual memory buffers
				* each block is BUFSIZ+4 bytes in size, holding BUFSIZ characters
				* plus 4 bytes of control info
2000	USREGN	EQU	\$2000	beginning of user space
3000	USREND	EQU	\$3000	end of user space, above is for disc sim
3000	DSMBGN	EQU	\$3000	begin of space available for disc simulation
4000	DSMEND	EQU	\$4000	end of memory available for disc simulation
3000	MEMEND	EQU	DSMBGN	
4000	MEMTOP	EQU	DSMEND	
			*	

PAG

```
*** * * *
* CONVENTIONS USED IN THIS PROGRAM ARE -
*
* IP = register Y points towards the next word to execute
* SP = register U points to LAST BYTE on the data stack
* RP = register S points to LAST WORD on return stack
* register X is used as a general index register for pointing
* at things. For some indexing purposes, Y,U, or S are
* saved so X and Y, U, or S may be used at same time.
* W upon entry to a word, X = W = location of word containing
* address of code to execute.
*
*
* When A and B are used separately, in order to maintain compatibility
* with D register, A contains high byte, B the low byte.
*
*** * * *
```

```
***** MACRO for creating dictionary headers *****
0000 LASTNM SET 0
*
WORDM MACRO
NEXTNM SET *
    IFC &4,IMMEDIATE
    FCB &1+$C0
* 1st byte is no of char with sign and immed bit on if IMMEDIATE
    ELSE
    FCB &1+$80
    ENDIF
    IFNC &1,1
    FCC `&2`
    ENDIF
* if more than one char, then all but last in here
* then last has sign bit set
    FCB $80+`&3
    FDB LASTNM
LASTNM SET NEXTNM
IFC &5,USER
&6 FDB DOUSER    TSC assembler gives error message -- IGNORE
*** ILLEGAL LABEL
    FDB &7-UORIG
ENDIF
ENDM
*
PAG
```

2000 ORG USRECN variables
2000 N RMB 10 used as scratch
200A UP RMB 2 the pointer to base of current user's
* USER table (for multi-tasking)
* This system is shown for one user, but additional ones
* may be added by allocating additional user tables and
* words for switching the pointer between them.
* Alternatively, with SWTP SBUG dynamic memory assignment, it would
* be possible to have a memory management procedure in KERNEL which
* switches various USER 4k blocks in and out of this low space.
*
* Some of the next stuff is initialized during COLD and WARM starts.
* Names correspond to FORTH words of similar (no X) name.
*
200C UORIG RMB 6 3 reserved variables
* INIT ON COLD START
2012 XFENCE RMB 2 fence for FORGET
2014 XDP RMB 2 dictionary pointer
2016 XVOCL RMB 2 vocabulary linking
2018 XACIA RMB 2 address of acia port
201A XDELAY RMB 2 carriage return delay count (# of nulls)
201C XCOLUMN RMB 2 carriage width
201E XBKSP RMB 1 backspace character
201F XBKSPE RMB 1 backspace echo
2020 XLINDEL RMB 1 line delete character
2021 XLINDE RMB 1 line delete echo
* INIT BELOW ON COLD OR WARM
2022 XSPZER RMB 2 initial top of data stack for this user
2024 XTIB RMB 2 start of terminal input buffer
2026 XRZERO RMB 2 initial top of return stack
2028 XFINA RMB 2 address of input file FCB
202A XFOUTA RMB 2 address of output file FCB
202C XWIDTH RMB 2 name field width
202E XMSGBS RMB 2 Base SCreen number for messages and GO
2030 XWARN RMB 2 warning message mode (0 = no disk)
* END OF INITIALIZED PARAMETERS
2032 0000 XBLK FDB 0 disc block being accessed
2034 0000 XIN FDB 0 scan pointer into the block
2036 0002 XOUT FDB 2 cursor position
2038 0000 XSCR FDB 0 disc screen being accessed (0 = terminal)
203A 0000 XOFSET FDB 0 disc sector offset for multi=disc
203C 207E XCONT FDB TASK-7 last word in primary search vocabulary
203E 207E XCURR FDB TASK-7 last word in extensible vocabulary
2040 0000 XSTATE FDB 0 flag for 'interpret' or 'compile' modes
2042 000A XBASE FDB 10 number base for I/O numeric conversions
2044 0002 XDPL FDB 2 decimal point place
2046 0000 XFLD FDB 0
2048 0000 XCSP FDB 0 current stack position, for compile checks
204A 0000 XRNUM FDB 0
204C 0000 XHLD FDB 0
204E 0000 IOSTAT FDB 0 last acia status from write/read

* END OF USER TABLE
*
*** Beginning of variable dictionary entries

2050 C5		FCB	\$C5	5, IMMEDIATE
2051 46 4F 52 54		FCC	"FORT"	
2055 C8		FCB	\$80+H	
2056 1A34		FDB	NOOP-7	LINK "BACK"
2058 0700 105A	FORTH	FDB	DODES,DOVOC,\$81A0,TASKAA	
205C 81A0 207E				
2060 0000		FDB	0	
2062 28 43 29 20		FCC	"(C) Talbot Microsystems 1980"	
2066 54 61 6C 62				
206A 6F 74 20 4D				
206E 69 63 72 6F				
2072 73 79 73 74				
2076 65 6D 73 20				
207A 31 39 38 30				
207E 84	TASKAA	FCB	\$84	
207F 54 41 53		FCC	"TAS"	
2082 CB		FCB	\$80+K	
2083 2050		FDB	FORTH-S	link "back" to FORTH
2085 0073 0080	TASK	FDB	DOCOL,SEMIC	
2089 REND	EQU	*	(first empty location in dictionary)	
		PAG		

```
*      The FORTH program begins here;
0000          ORG      PRGBGN
* First, COLD and WARM entry points
0000 16 013F    KERNAL LBRA CENT
0003 16 018E    LBRA WENT
*****
*      Startup parameters ****
*
0006 6809      CPUTYP FDB    $6809    cpu
0008 0101      VERSON FDE    $0101    version wxyz print as wx.yz
000A 0000
000C 14
000D 52 2E 20 4A
0011 2E 20 54 41
0015 4C 42 4F 54
0019 2C 20 4A 52
001D 2E 20 20 20
0021 200C      UPINIT FDB    UORIG   initial user area
* FOLLOWING INITIALIZED ON COLD START ONLY
0023 207E      FENCIN FDB    TASKAA  initial fence at TASK
0025 2089      DPINIT FDB    REND    cold start value for DP location in dict.
0027 2060      VOCINT FDB    FORTH+8 cold start for VOC-LINK
0029 E004      ACIAI  FDB    SE004   initial location of acia port
002B 0008      DELINT FDB    8       initial carriage return delay
002D 0050      COLINT FDB    80     initial terminal carriage width
002F 08        BACKSP FCB    $08    character to indicate backspace
0030 08        BACKEC FCB    $08    character to echo for backspace
0031 18        LINDEL FCB    $18    character to indicate line delete
0032 18        LINDEC FCB    $18    character to echo for line delete
0033 1BF0      XViRbG FDB    ViRBGN
0035 2000      XViRED FDB    ViREND
0037 3000      XDSMBG FDB    DSMBGN
0039 4000      XDSMED FDB    DSMEND
* END COLD START INITIALIZATION AREA
*
* THE FOLLOWING USED TO INITIALIZE USER AREA ON WARM OR COLD START
003B 2F30      SINIT  FDB    USREND-$D0 initial top of data stack
003D 2F30      TIBINT FDB    USREND-$D0 terminal input buffer
003F 3000      RINIT  FDB    USREND  initial top of return stack
0041 0000      FINA   FDB    0       initialize no input file FCB
0043 0000      FCUTA  FDB    0       "       no output file FCB
0045 001F      WIDINT FDB    31    init name field width
0047 0028      MSGBAS FDB    40    init base SCreen number for messages and GO
0049 0001      WRNINT FDB    1       init warning mode (0= no disc)
* END WARM+COLD INITIALIZATION AREA
*
* system variables
004B           XUSE   RMB    2
004D           XPREV RMB    2
                  PAG
```

```
*  
* Start of FORTH Kernel  
*  
004F 37 06 PULLDX PULU D 15 cycles to NEXT  
0051 ED 84 STOREX STD ,X 8  
0053 20 22 BRA NEXT  
0055 EC 84 GETX LDD ,X 15 cycles to NEXT  
0057 36 06 PUSHD PSHU D 7  
0059 20 1C BRA NEXT  
*  
* Here is the IP pusher for allowing nested words  
* ;S is the equivalent unnester  
*  
005B WORDM 1,,:,IMMEDIATE  
005F 0073 0A51 COLON FDB DOCOL,QEXEC,SCSP,CURENT,AT,CONXT,STORE  
0063 0A0D 0877  
0067 06A7 0869  
006B 06BF  
006D 0F7D 0AD2 FDB CREATE,RBRAK,PSCODE  
0071 0E21  
0073 34 20 DOCOL PSHS Y save present IP on ret stack RP  
0075 31 02 LEAY 2,X kick Y up to first param after CFA in W=X  
* LBRA NEXT JUST DROP ON THROUGH T NEXT  
*  
* NEXT takes 14 cycles  
*  
**** BEGINNING OF SIMULATION OF VIRTUAL FORTH MACHINE  
*  
0077 AE A1 NEXT LDX ,Y++ get W to X and then increment Y=IP  
* the address of the pointer to the present code is in X now  
* if need it at any time, it may be computed by LDX -2,Y  
0079 6E 94 NEXT3 JMP [,X] jump indirect to code pointed to by W  
*  
**** END OF SIMULATION OF THE VIRTUAL FORTH MACHINE  
007B WORDM 2,;,S  
0080 0082 SEMIS FDB *+2  
0082 10AE E1 PSEMIS LDY ,S++ reset Y=IP to next addr and drop frm S=RP  
0085 20 F0 BRA NEXT  
PAG
```

0087		WORDM	7 ,EXECUT,E
0091 0093	EXEC	FDB	*+2
0093 37 10		PULU	X
0095 20 E2		BRA	NEXT3
0097		WORDM	3 ,MO,N
009D 1A59	MON	FDB	PMON
009F		WORDM	3 ,JS,R
00A5 00A7	JSR	FDB	*+2
00A7 AD D1		JSR	[,U++]
>00A9 16 FFCE		LBRA	NEXT
00AC		WORDM	4 ,EMI,T
00B3 0073 00E9	EMIT	FDB	DOCOL,CEMIT,SEMIS
00B7 0080			
00B9 00BB	CEMIT	FDB	*+2 this is a word with no header
00BB 37 06		PULU	D
00BD 1F 98		TFR	B,A
00BF 17 198E		LBSR	PEMIT
00C2 BE 2036		LDX	XOUT
00C5 30 01		LEAX	1,X increment by 1
00C7 BF 2036		STX	XOUT
>00CA 16 FFAA		LBRA	NEXT
00CD		WORDM	3 ,KE,Y
00D3 0073 00D9	KEY	FDB	DOCOL,CKEY,SEMIS
00D7 0080			
00D9 00DE	CKEY	FDB	*+2 this is a word with no header
00DB 17 1975		LBSR	PKEY
00DE 1F 89		TFR	A,B
00E0 4F		CLRA	
00E1 16 FF73		LBRA	PUSHD
00E4		WORDM	9 ,?TERMINA,L
00F0 00F2	QTERM	FDB	*+2
00F2 17 1961		LBSR	PQTER
00F5 1F 89		TFR	A,B
00F7 4F		CLRA	
00F8 16 FF5C		LBRA	PUSHD
00FB		WORDM	2 ,C,R
0100 0073 00F0	CR	FDB	DOCOL,QTERM,ZBRAN
0104 020B			
0106 0004		FDB	CR1-*
0108 10F2		FDB	QUIT
010A 01EE	CR1	FDB	CLITER
010C 0A		FCB	\$0A
010D 00E3 01EE		FDB	EMIT,CLITER
0111 0D		FCB	\$0D LF
0112 00B3 076B		FDB	EMIT,ZERO,OUT,STORE
0116 0844 06BF			
011A 01E7 201A		FDB	LIT,XDELAY,AT,ZBRAN
011E 06A7 020B			
0122 0014		FDB	CRE-*
0124 01E7 201A		FDB	LIT,XDELAY,AT,ZERO,XDO
0128 06A7 076B			

012C 0261				
012E 076B 00B3	CR2	FDB	ZERO,EMIT,XLOOP	
0132 0228				
0134 FFFA		FDB	CR2-*	
0136 0080	CRE	FDB	SEMIS	
0138 FF	IFCOLD	FCB	\$FF	
0139		WORDM	4,COL,D	
0140 0142	COLD	FDB	*+2	
0142 DE 25	CENT	LDU	DPINIT top of destination	
0144 8E 11C9		LDX	#ERAM top of stuff to move	
0147 A6 82	COLD2	LDA	,-X	
0149 A7 C2		STA	,-U	
014B 8C 1190		CPX	#RAM	
014E 26 F7		BNE	COLD2	
0150 86 FF		LDA	#\$FF	
0152 B7 0138		STA	IFCOLD	
0155 10DE 35		LDS	XVIRED put stack somewhere safe	
0158 9E 35		LDX	XVIRED	
015A BF 07A6		STX	LIMIT+2	
015D 9E 33		LDX	XVIRBG	
015F 9F 4B		STX	XUSE	
0161 9F 4D		STX	XPREV	
0163 BF 079A		STX	FIRST+2	
0166 86 00		LDA	#0	
0168 A7 80	COLD8	STA	,X+ ·	
016A 9C 35		CMPX	XVIRED	
016C 26 FA		BNE	COLD8	
016E A7 84		STA	,X	
0170 9E 39		LDX	XDSMED	
0172 BF 1862		STX	HI+2	
0175 9E 37		LDX	XDSMBG	
0177 BF 1859		STX	LO+2	
017A CE 2022		LDU	#XLINDE+1	
017D 8E 0033		LDX	#LINDEC+1	
0180 A6 82	COLDZ	LDA	,-X	
0182 A7 C2		STA	,-U	
0184 8C 0023		CPX	#FENCIN	
0187 26 F7		BNE	COLDZ	
0189 20 09		BRA	WENT	
018B		WORDM	4,WAR,M	
0192 0194	WARM	FDB	*+2	
0194 CE 2032	WENT	LDU	#XWARN+2	
0197 8E 004B		LDX	#WRNINT+2	
019A A6 82	WARM2	LDA	,-X	
019C A7 C2		STA	,-U	
019E 8C 003B		CPX	#SINIT	
01A1 26 F7		BNE	WARM2	
01A3 FE 2022		LDU	XSPZER U is SP	
01A6 9E 21		LDX	UPINIT	
01A8 BF 200A		STX	UP init user pointer	
01AB 108E 1122		LDY	#ABORT+2 Y is IP, init to first instruc in ABORT	
01AF 12	. INTSPC	NOP		

01B0 12		NOP	here is place to jump to special
01B1 12		NOP	initialization routines
>01B2 16 0025		LBRA RPSTOR+2	
01B5		WORDM 3,SP,@	
01BB 01BD	SPAT	FDB *+2	
01BD 30 C4		LEAX ,U	X = VALUE OF SP
01BF 36 10		PSHU X	
01C1 16 FEB3		LBRA NEXT	
01C4		WORDM 3,SP,!	
01CA 01CC	SPSTOR	FDB *+2	
01CC FE 2022		LDU XSPZER	
01CF 16 FEA5		LBRA NEXT	
01D2		WORDM 3,RP,!	
01D8 01DA	RPSTOR	FDB *+2	
01DA 10FE 2026		LDS XRZERO	initialize S=RP from constant
01DE 16 FE96		LBRA NEXT	
01E1		WORDM 3,LI,T	NOTE: this is different from LITERAL
01E7 01E9	LIT	FDB *+2	
01E9 EC A1		LDL ,Y++	get word pointed to by Y=IP and increment
01EB 16 FE69		LERA PUSHD	push D to data stack and then NEXT
01EE 01F0	CLITER	FDB *+2	this is an invisible word with no header
01F0 E6 A0		LDB ,Y+	
01F2 4F		CLRA	
01F3 16 FE61		LBRA PUSHD	
01F6		WORDM 6,BRANC,H	
01FF 0211	BRAN	FDB ZBYES	go steal code in ZBRANCH
0201		WORDM 7,0BRANC,H	
020B 020D	ZBRAN	FDB *+2	
020D EC C1		LDD ,U++	get quantity on stack and drop it
020F 26 09		BNE ZBNO	
0211 1F 20	ZBYES	TFR Y,D	puts IP = Y into D for arithmetic
0213 E3 A4		ADDD ,Y	adds offset to which IP is pointing
0215 1F 02		TFR D,Y	sets new IP
0217 16 FE5D		LBRA NEXT	
021A 31 22	ZBNO	LEAY 2,Y	skip over branch
021C 16 FE58		LBRA NEXT	
021F		WORDM 6,(LOOP,)	
0228 022A	XLOOP	FDB *+2	
022A CC 0001		LDD #1	set inc cntr to 1 and steal other code
022D 20 0E		BRA XPLCP2	
022F		WORDM 7,(+LOOP,)	
0239 023E	XPLOOP	FDB *+2	
023B 37 06		PULU D	
023D 4D	XPLCP2	TSTA	
023E 2A 0E		BPL XPLOF	forward loopint
0240 E3 E4		ADDD ,S	add D to counter on RP=S
0242 ED E4		STD ,S	
0244 1C 01		ANDCC #\\$1	set c bit
0246 E2 63		SBCB 3,S	
0248 A2 62		SBCA 2,S	
024A 2A C5		BPL ZBYES	
024C 20 08		BRA XPLONO	fall thru

024E E3	E4	XPLOF	ADDD	,S	
0250 ED	E4		STD	,S	
0252 A3	62		SUBD	2,S	
0254 2B	BB		BMI	ZBYES	
0256 32	64	XPLONO	LEAS	4,S	drop 4 bytes of counter and limit
0258 20	C0		ERA	ZBNO	use ZBRAN to skip over unused delta
025A			WORDM	4,(DO,)	
0261 0263		XDO	FDB	*+2	
0263 37	06		PULU	D	counter
0265 37	10		PULU	X	limit
0267 34	16		PSHS	X,D	X goes first, so becomes second on RP=S
0269 16	FE05		LBRA	NEXT	
026C			WORDM	1,,I	
0270 0272		I	FDB	*+2	
0272 EC	E4		LDD	,S	get counter from RP
0274 16	FDE0		LBRA	PUSHD	
0277			WORDM	1,,J	
027B 027D		J	FDB	*+2	
027D EC	64		LDD	4,S	get second counter above limit for first
027F 16	FDD5		LBRA	PUSHD	
0282			WORDM	1,,K	
0286 0288		K	FDB	*+2	
0288 EC	68		LDD	8,S	get third counter
028A 16	FDCA		LBRA	PUSHD	
028D			WORDM	5,DIGI,T	
0295 0297		DIGIT	FDB	*+2	
0297 A6	43		LDA	3,U	second item is char of interest
0299 80	30		SUBA	#\$30	ascii zero
029B 2B	1B		BMI	DIGIT2	if less than '0', ILLEGAL
029D 81	0A		CMPA	#\$A	
029F 2B	0A		BMI	DIGITO	if '9' or less
02A1 81	11		CMPA	#\$11	
02A3 2B	13		BMI	DIGIT2	if less than 'A'
02A5 81	2B		CMPA	#\$2B	
02A7 2A	0F		BPL	DIGIT2	if greater than 'Z'
02A9 80	07		SUBA	#7	translate 'A' thru 'Z'
02AB A1	41	DIGITO	CMPA	1,U	
02AD 2A	09		BPL	DIGIT2	if not less than base
02AF C6	01		LDB	#1	
02B1 A7	43		STA	3,U	
02B3 E7	41	DIGIT1	STAB	1,U	store flag
02B5 16	FDBF		LBRA	NEXT	
02B8 5F		DIGIT2	CLRB		
02B9 33	42		LEAU	2,U	pop top off
02BB E7	C4		STAB	0,U	make sure both bytes 0
02BD 20	F4		BRA	DIGIT1	
02BF			WORDM	6,(FIND,)	
02C8 02CA		PFIND	FDB	*+2	
2000	PD		EQU	N	
2002	PA0		EQU	N+2	
2004	PA		EQU	N+4	
2006	PCHR		EQU	N+6	

02CA 34 20		PSHS	Y	save Y
02CC 37 30	PFIND0	PULU	X,Y	
02CE 10BF 2002		STY	PAO	
	*	*	*	X is dict ptr Y is ptr to word that finding
02D2 E6 80	PFIND1	LDB	,X+	get count from dict
02D4 F7 2006		STAB	PCHR	
02D7 C4 3F		ANDB	#\$3F	mask sign and precedence
02D9 10BE 2002		LDY	PAO	
02DD E1 A0		CMPB	0,Y+	
02DF 26 18		BNE	PFIND4	not equal
02E1 A6 A0	PFIND2	LDA	,Y+	
02E3 6D 84		TST	,X	is dict entry neg?
02E5 2A 0E		BPL	PFIND8	
02E7 8A 80		ORA	#\$80	make A neg also
02E9 A1 80		CMPA	,X+	
02EB 27 12		BEQ	FOUND	
02ED AE 84	PFIND3	LDX	0,X	get new link in dict
02EF 26 E1		BNE	PFIND1	continue if new link not = 0
	*	not found :		
02F1 1F 10		TFR	X,D	
02F3 20 14		BRA	PFINDE	
	*			
02F5 A1 80	PFIND8	CMPA	,X+	
02F7 27 E8		BEQ	PFIND2	
02F9 E6 80	PFIND4	LDB	,X+	scan forward to end of name
02FB 2A FC		BPL	PFIND4	
02FD 20 EE		BRA	PFIND3	
	*			
	*	found :		
02FF 30 04	FOUND	LEAX	4,X	point to parameter field
0301 F6 2006		LDB	PCHR	
0304 4F		CLRA		
0305 36 16		PSHU	X,D	X goes first
0307 C6 01		LDB	#1	
0309 35 20	PFINDE	PULS	Y	
030B 16 FD49		LBRA	PUSHD	
030E		WORDM	7,ENCLOS,E	
	*	NOTE:	FC means offset (bytes) to First Character of next word	
	*	EW	" "	to End of next Word
	*	NC	" "	to Next Character to start next enclose at
0318 031A	ENCLOS	FDE	*+2	
031A 37 06		PULU	D	get char off stack to use as delim into B
031C AE C4		LDX	,U	addr to begin
031E 7F 2000		CLR	N	
0321 F7 2001		STB	N+1	save delim to use
	*	wait	for a non-delimiter or NUL	
0324 A6 84	ENCL2	LDA	0,X	
0326 27 2A		BEQ	ENCL6	
0328 B1 2001		CMPA	N+1	check for delim
032B 26 07		BNE	ENCL3	
032D 30 01		LEAX	1,X	
032F 7C 2000		INC	N	

0332 20 F0 BRA ENCL2
* found first character, Push PC
0334 F6 2000 ENCL3 LDB N found first character
0337 4F CLRA
0338 36 06 PSHU D
* wait for a delimiter or NUL
033A A6 80 ENCL4 LDA ,X+
033C 27 1C BEQ ENCL7
033E B1 2001 CMPA N+1 check for delim
0341 27 05 BEQ ENCL5
0343 7C 2000 INC N
0346 20 F2 BRA ENCL4
* found EW, Push it
0348 F6 2000 ENCL5 LDB N
034B 4F CLRA
034C 36 06 PSHU D
*advance and push NC
034E 5C INCB
034F 16 FD05 LBRA PUSHD
* found NUL before non delimiter, therefore, no word
0352 F6 2000 ENCL6 LDB N A is zero
0355 36 06 PSHU D
0357 5C INCB
0358 20 03 BRA ENCL7P
* found NUL following word instead of SPACE
035A F6 2000 ENCL7 LDB N
035D 36 06 ENCL7P PSHU D save EW
035F F6 2000 ENCL8 LDB N save NC
0362 16 FCF2 LBRA PUSHD
0365 WORDM 5,CMOV,E sourcead, destinationad, count
036D 036F CMOVE FDB *+2
036F 8D 03 BSR PCMOVE
0371 16 FD03 LBRA NEXT
0374 34 30 PCMOVE PSHS X,Y
0376 37 36 PULU D,X,Y D=ct, X=dest, Y=source
0378 34 40 PSHS U
037A 1F 23 TFR Y,U
037C 1F 02 TFR D,Y use Y as COUNTER
037E 31 21 LEAY 1,Y
0380 31 3F CMOV2 LEAY -1,Y
0382 27 06 BEQ CMOV3
0384 A6 C0 LDA ,U+
0386 A7 80 STA ,X+
0388 20 F6 BRA CMOV2
038A 35 40 CMOV3 PULS U
038C 35 30 PULS X,Y
038E 39 RTS
*
038F WORDM 2,U,*
0394 0396 USTAR FDB *+2
0396 8D 05 BSR USTARS
0398 33 42 LEAU 2,U

039A 16	FCBA	LBRA	PUSHD	
*				
* The following is a subroutine which multiplies top				
* 2 words on stack, leaving 32-bit result: high order in D				
* and low order word in 2ND word of stack.				
039D 8E	0011	USTARS	LDX #17	
03A0 CC	0000		LDD #0	
03A3 66	42	USTAR2	ROR 2,U	shift mult
03A5 66	43		ROR 3,U	
03A7 30	1F		LEAX -1,X	done ?
03A9 27	08		BEQ USTAR4	
03AB 24	02		BCC USTAR3	
03AD E3	C4		ADDD ,U	
03AF 46		USTAR3	RORA	
03B0 56			RORB	
03B1 20	F0		BRA USTAR2	
03B3 39		USTAR4	RTS	
03B4			WORDM 2,U,/	
03B9 03BB		USLASH	FDB *+2	
03BB EC	42		LDD 2,U	
03BD AE	44		LDX 4,U	
03BF AF	42		STX 2,U	
03C1 ED	44		STD 4,U	
03C3 68	43		ASL 3,U	
03C5 69	42		ROL 2,U	
03C7 8E	0010		LDX #\$10	
03CA 69	45	USLL1	ROL 5,U	
03CC 69	44		ROL 4,U	
03CE EC	44		LDD 4,U	
03D0 A3	C4		SUBD ,U	
03D2 1C	FE		ANDCC #\$FE	CLC
03D4 2B	04		BMI USLL2	
03D6 ED	44		STD 4,U	
03DS 1A	01		ORCC #1	SEC
03DA 69	43	USLL2	ROL 3,U	
03DC 69	42		ROL 2,U	
03DE 30	1F		LEAX -\$1,X	
03E0 26	E8		BNE USLL1	
03E2 33	42		LEAU 2,U	
03E4 16	FC90		LBRA NEXT	
03E7			WORDM 3,AN,D	
03ED 03EF		AND	FDB *+2	
03EF 37	06		PULU D	
03F1 E4	41		ANDB 1,U	
03F3 A4	C4		ANDA 0,U	
03F5 ED	C4	PUTD	STD ,U	
03F7 16	FC7D		LBRA NEXT	
03FA			WORDM 2,O,R	
03FF 0401		OR	FDB *+2	
0401 37	06		PULU D	
0403 EA	41		ORB 1,U	
0405 AA	C4		ORA 0,U	

0407 20 EC		BRA	PUTD
0409		WORDM	3,XO,R
040F 0411	XOR	FDB	*+2
0411 37 06		PULU	D
0413 E8 41		EORB	1,U
0415 A8 C4		EORA	0,U
0417 20 DC		BRA	PUTD
0419		WORDM	1,+,
041D 041F	PLUS	FDB	*+2
041F 37 06		PULU	D
0421 E3 C4		ADDD	,U
>0423 16 FFCF		LBRA	PUTD
0426		WORDM	2,D,+
042B 042D	DPLUS	FDB	*+2
042D EC 42		LDD	2,U
042F E3 46		ADDD	6,U
0431 ED 46		STD	6,U
0433 EC C4		LDD	,U
0435 E9 45		ADCB	5,U
0437 A9 44		ADCA	4,U
0439 33 44		LEAU	4,U
043B ED C4		STD	,U
043D 16 FC37		LBRA	NEXT
0440		WORDM	5,MINU,S
0448 044A	MINUS	FDB	*+2
044A 60 41		NEG	1,U
044C 25 05		BCS	MINUS2
044E 60 C4		NEG	,U
0450 16 FC24		LBRA	NEXT
0453 63 C4	MINUS2	COM	,U
0455 16 FC1F		LBRA	NEXT
0458		WORDM	6,DMINU,S
0461 0463	DMINUS	FDB	*+2
0463 63 C4		COM	0,U
0465 63 41		COM	1,U
0467 63 42		COM	2,U
0469 60 43		NEG	3,U
046B 26 0A		BNE	DMINX
046D 6C 42		INC	2,U
046F 26 06		BNE	DMINX
0471 6C 41		INC	1,U
0473 26 02		BNE	DMINX
0475 6C C4		INC	,U
0477 16 FBFD	DMINX	LBRA	NEXT
047A		WORDM	2,1,+,
047F 0481	ONEP	FDB	*+2
0481 EC C4		LDD	,U
0483 C3 0001		ADDD	#1
0486 16 FF6C		LBRA	PUTD
0489		WORDM	2,2,+,
048E 0490	TWOP	FDB	*+2
0490 CC 0002		LDD	#2

0493 E3 C4		ADDD	,U
0495 16 FF5D		LBRA	PUTD
0498		WORDM	2,1,-
049D 049F	ONEM	FDB	*+2
049F EC C4		LDD	,U
04A1 83 0001		SUBD	#1
04A4 16 FF4E		LBRA	PUTD
04A7		WORDM	2,2,-
04AC 04AE	TWON	FDB	*+2
04AE EC C4		LDD	,U
04B0 83 0002		SUBD	#2
04B3 16 FF3F		LBRA	PUTD
04B6		WORDM	2,M,*
04BB 0073 065D	MSTAR	FDB	DOCOL,OVER,OVER,XOR,TOR,ABS,SWAP,ABS,USTAR
04BF 065D 040F			
04C3 0639 057C			
04C7 0679 057C			
04CE 0394			
04CD 0647 05E8		FDB	FROMR,DSETSN,SEMS
04D1 0080			
04D3		WORDM	1,,*
04D7 0073 04BB	STAR	FDB	DOCOL,MSTAR,DROP,SEMS
04DB 066B 0080			
04DF		WORDM	2,M,/ signed double=-3,-2,signed divisor-1 --> signed rem -2 , quotient -1
04E4 0073 065D	MSLASH	FDB	DOCOL,OVER,TOR,TOR,DABS,R,ABS,USLASH,FROMR,R,XOR
04E8 0639 0639			
04EC 0591 0654			
04F0 057C 03B9			
04F4 0647 0654			
04F8 040F			
04FA 05D6 0679		FDB	SETS N,SWAP,FROMR,SETS N,SWAP,SEMS
04FE 0647 05D6			
0502 0679 0080			
0506		WORDM	4,/MO,D
050D 0073 0639	SLMOD	FDB	DOCOL,TOR,STOD,FROMR,MSLASH,SEMS
0511 05C1 0647			
0515 04E4 0080			
0519		WORDM	1,,/
051D 0073 050D	SLASH	FDB	DOCOL,SLMOD,SWAP,DROP,SEMS
0521 0679 066B			
0525 0080			
0527		WORDM	3,MO,D
052D 0073 050D	MOD	FDB	DCCOL,SLMOD,DROP,SEMS
0531 066B 0080			
0535		WORDM	5,*/MO,D
053D 0073 0639	SSMOD	FDB	DOCOL,TOR,MSTAR,FROMR,MSLASH,SEMS
0541 04BB 0647			
0545 04E4 0080			
0549		WORDM	2,*,/
054E 0073 053D	SSLASH	FDB	DOCOL,SSMOD,SWAP,DROP,SEMS
0552 0679 066B			

0556 0080			
0558		WORDM	5,M/MO,D
0560 0073 0639	MSMOD	FDB	DOCOL,TOR,ZERO,R,USLASH,FROMR,SWAP,TOR
0564 076B 0654			
0568 03B9 0647			
056C 0679 0639			
0570 03B9 0647		FDB	USLASH,FROMR,SEMIS
0574 0080			
0576		WORDM	3,AB,S
057C 0073 068A	AES	FDB	DOCOL,DUP,ZLESS,ZBRAN
0580 0611 020B			
0584 0004		FDB	ABS2-*
0586 0448		FDB	MINUS
0588 0080	ABS2	FDB	SEMIS
058A		WORDM	4,DAB,S
0591 0073 068A	DAES	FDB	DOCOL,DUP,ZLESS,ZBRAN
0595 0611 020B			
0599 0004		FDB	DABS2-*
059B 0461		FDB	DMINUS
059D 0080	DABS2	FDB	SEMIS
059F		WORDM	1,,<
05A3 05A5	LESS	FDB	*+2
05A5 37 06		PULU	D
05A7 A1 C4		CMPA	0,U
05A9 2E 09		BGT	LESST
05AB 26 04		BNE	LESSF
05AD E1 41		CMPB	1,U
05AF 22 03		BHI	LESST
05B1 5F	LESSF	CLRE	
05B2 20 02		BRA	LESSX
05B4 C6 01	LESST	LDB	#1
05B6 4F	LESSX	CLRA	
05B7 16 FE3B		LBRA	PUTD
05BA		WORDM	4,S->,D
05C1 05C3	STOD	FDB	*+2
05C3 CC 0000		LDD	#0
05C6 6D C4		TST	,U
05C8 2A 02		BPL	STOD2
05CA 43		COMA	
05CB 53		COMB	
05CC ED C3	STOD2	STD	--U
05CE 16 FAA6		LBRA	NEXT
05D1		WORDM	2,+, -
05D6 0073 0611	SETSN	FDB	DOCOL,ZLESS,ZBRAN
05DA 020B			
05DC 0004		FDB	SETS N2-*
05DE 0448		FDB	MINUS
05E0 0080	SETS N2	FDB	SEMIS
05E2		WORDM	3,D+,-
05E8 0073 0611	DSETS N	FDB	DOCOL,ZLESS,ZBRAN
05EC 020B			
05EE 0004		FDB	DSETS 2-*

05F0 0461		FDB	DMINUS
05F2 0080	DSETS2	FDB	SEMIS
05F4 33 42		LEAU	2,U
05F6 16 FA7E		LBRA	NEXT
05F9		WORDM	2,0,=
05FE 0600	ZEQU	FDB	*+2
0600 4F		CLRA	
0601 5F		CLRB	
0602 AE C4		LDX	,U
0604 26 01		BNE	ZEQU2
0606 5C		INCB	
0607 ED C4	ZEQU2	STD	,U
0609 16 FA6B		LBRA	NEXT
060C		WORDM	2,0,<
0611 0613	ZLESS	FDB	*+2
0613 86 80		LDA	#\$80 check sign bit
0615 A4 C4		ANDA	,U
0617 27 06		BEQ	ZLESS2
0619 4F		CLRA	
061A C6 01		LDE	#1
061C 16 FDD6		LBRA	PUTD
061F 5F	ZLESS2	CLRB	
0620 16 FDD2		LBRA	PUTD
	*		
0623		WORDM	5,LEAV,E
062B 062D	LEAVE	FDB	*+2
062D EC E4		LDD	,S
062F ED 62		STD	2,S
0631 16 FA43		LBRA	NEXT
0634		WORDM	2,>,R
0639 063B	TOR	FDB	*+2
063B 37 06		PULU	D
063D 34 06		PSHS	D
063F 16 FA35		LBRA	NEXT
0642		WORDM	2,R,>
0647 0649	FROMR	FDB	*+2
0649 35 06		PULS	D
064B 36 06		PSHU	D
064D 16 FA27		LBRA	NEXT
0650		WORDM	1,,R
0654 0272	R	FDB	I+2 steal code from I
0656		WORDM	4,OVE,R
065D 065F	OVER	FDB	*+2
065F EC 42		LDD	2,U
0661 16 F9F3		LBRA	PUSHD
0664		WORDM	4,DRO,P
066B 066D	DROP	FDB	*+2
066D 33 42		LEAU	2,U
066F 16 FA05		LBRA	NEXT
0672		WORDM	4,SWA,P
0679 067B	SWAP	FDB	*+2
067B 37 16		PULU	D,X

067D 1E 01		EXG	D,X	swap order
067F 36 16		PSHU	D,X	
0681 16 F9F3		LBRA	NEXT	
0684		WORDM	3,DU,P	
068A 068C	DUP	FDB	*+2	
068C EC C4		LDD	,U	
068E 16 F9C6		LBRA	PUSHD	
0691		WORDM	2,+,!	
0696 0698	PSTORE	FDB	*+2	
0698 AE C1		LDX	,U++	
069A EC C1		LDD	,U++	
069C E3 84		ADDD	,X	
069E ED 84		STD	,X	
06A0 16 F9D4		LERA	NEXT	
06A3		WORDM	1,,@	
06A7 06A9	AT	FDB	*+2	
06A9 EC D4		LDD	[,U]	U points to address on stack, get # there
06AB 16 FD47		LBRA	PUTD	replace stack add with #
06AE		WORDM	2,C,G	
06B3 06B5	CAT	FDB	*+2	
06B5 E6 D4		LDB	[,U]	
06B7 4F		CLRA		
06B8 16 FD3A		LBRA	PUTD	
06BB		WORDM	1,,!	
06BF 06C1	STORE	FDE	*+2	
06C1 37 10		PULU	X	
06C3 37 06		PULU	D	forced to do this because in wrong order
06C5 ED 84		STD	,X	
06C7 16 F9AD		LBRA	NEXT	
06CA		WORDM	2,C,!	
06CF 06D1	CSTORE	FDB	*+2	
06D1 37 10		PULU	X	
06D3 37 06		PULU	D	
06D5 E7 84		STB	,X	
06D7 16 F99D		LBRA	NEXT	
06DA		WORDM	7,<BUILD,S	
06E4 0073 076B	BUILDS	FDB	DOCOL,ZERO,CON,SEMIS	
06E8 0740 0080				
06EC		WORDM	5,DOES,>	
06F4 0073 0647	DOES	FDB	DOCOL,FROMR,LATEST,PFA,STORE,PSCODE	
06F8 09B7 09F9				
06FC 06BF 0E21				
0700 34 20	DODOES	PSHS	Y	push return address to RP=S
0702 10AE 02		LDY	2,X	get new IP
0705 30 04		LEAX	4,X	get address of parameter
0707 36 10		PSHU	X	
0709 16 F96B		LBRA	NEXT	
070C		WORDM	6,TOGGL,E	
0715 0073 065D	TOGGLE	FDB	DOCOL,OVER,CAT,XOR,SWAP,CSTORE,SEMIS	
0719 06B3 040F				
071D 0679 06CF				
0721 0080				

0723		WORDM	1,,;,IMMEDIATE
0727 0073 0A79	SEMI	FDB	DOCOL,QCSP,COMPIL,SEMIS,SMUDGE,LBRAK,SEMIS
072B 0AAE 0080			
072F 0AE6 0AC4			
0733 0080			
0735		WORDM	8,CONSTAN,T
0740 0073 0F7D	CON	FDS	DOCOL,CREATE,SMUDGE,COMMA,PSCODE
0744 0AE6 08F4			
0748 0E21			
074A EC 02	DOCON	LDD	2,X
074C 16 F908		LBRA	PUSHD
074F		WORDM	8,VARIABL,E
075A 0073 0740	VAR	FDB	DOCOL,CON,PSCODE
075E 0B21			
0760 30 02	DOVAR	LEAX	2,X gets address after CFA in W=X
0762 36 10		PSHU	X
0764 16 F910		LBRA	NEXT
0767		WORDM	1,,0
076B 074A	ZERO	FDB	DOCON
076D 0000		FDB	0
076F		WORDM	1,,1
0773 074A	ONE	FDB	DOCON
0775 0001		FDB	1
0777		WORDM	1,,2
077B 074A	TWO	FDB	DOCON
077D 0002		FDB	2
077F		WORDM	1,,3
0783 074A	THREE	FDB	DOCON
0785 0003		FDB	3
0787		WORDM	2,B,L
078C 074A	BL	FDB	DOCON
078E 0020		FDB	\$20 ascii blank
0790		WORDM	5,FIRS,T
0798 074A	FIRST	FDB	DOCON
079A 1BF0		FDB	VIRBGN
079C		WORDM	5,LIMI,T
07A4 074A	LIMIT	FDB	DOCON
07A6 2000		FDB	VIREND
07A8		WORDM	4,USE,R
07AF 0073 0740	USER	FDB	DOCOL,CON,PSCODE
07B3 0B21			
07B5 EC 02	DOUSER	LDD	2,X gets offset to user's table
07B7 F3 200A		ADDD	UP add to users base address
07BA 16 F89A		LBRA	PUSHD
07BD		WORDM	7,+ORIGI,N
07C7 0073 01E7	PORIG	FDB	DOCOL,LIT,PRGBGN,PLUS,SEMIS
07CB 0000 041D			
07CF 0080			
07D1		WORDM	2,S,0
07D6 07B5	SZERO	FDB	DOUSER
07D8 0016		FDB	XSPZER-UORIG

07DA		WORDM	2,R,0
07DF 07B5	RZERO	FDB	DOUSER
07E1 001A		FDB	XRZERO-UORIG
07E3		WORDM	3, TI,B,,USER,TIB,XTIB
07ED		WORDM	5,WIDT,H,,USER,WIDTH,XWIDTH
07F9		WORDM	7,WARNIN,G,,USER,WARN,XWARN
0807		WORDM	5,FENC,E,,USER,FENCE,XFENCE
0813		WORDM	2,D,P,,USER,DP,XDP
081C		WORDM	8,VOC-LIN,K,,USER,VOCLIN,XVOCL
082B		WORDM	3,BL,K,,USER,ELK,XELK
0835		WORDM	2,I,N,,USER,IN,XIN
083E		WORDM	3,OU,T,,USER,OUT,XOUT
0848		WORDM	3,SC,R,,USER,SCR,XSCR
0852		WORDM	6,OFFSE,T,,USER,OFFSET,XOFFSET
085F		WORDM	7,CONTEX,T,,USER,CONTEXT,XCONTEXT
086D		WORDM	7,CURREN,T,,USER,CURRENT,XCURRENT
087B		WORDM	5,STAT,E,,USER,STATE,XSTATE
0887		WORDM	4,BAS,E,,USER,BASE,XBASE
0892		WORDM	3,DP,L,,USER,DPL,XDPL
089C		WORDM	3,FL,D,,USER,FLD,XFLD
08A6		WORDM	3,CS,P,,USER,CSP,XCSP
08B0		WORDM	2,R,#,,USER,RNUM,XRNUM
08B9		WORDM	3,HL,D,,USER,HLD,XHLD
08C3		WORDM	7,COLUMN,S,,USER,COLUMNS,XCOLUMNS
*			
08D1		WORDM	4,HER,E
08D8 0073 0818	HERE	FDB	DOCOL,DP,AT,SEMIC
08DC 06A7 0080		WORDM	5,ALLO,T
08E0		FDB	DOCOL,DP,PSTORE,SEMIC
08E8 0073 0818	ALLOT	WORDM	1,","
08EC 0696 0080		FDB	DOCOL,HERE,STORE,TWO,ALLOT,SEMIC
08F0		WORDM	2,C,","
08F4 0073 08D8	COMMA	FDB	DOCOL,HERE,CSTORE,ONE,ALLOT,SEMIC
08F8 06EF 077B		WORDM	1,-
08FC 08E3 0080		FDB	DOCOL,MINUS,PLUS,SEMIC
0900		WORDM	1,=
0905 0073 08D8	CCOMMA	FDB	DOCOL,HERE,CSTORE,ONE,ALLOT,SEMIC
0909 06CF 0773		WORDM	1,,>
090D 08E8 0080		FDB	DOCOL,SWAP,LESS,SEMIC
0911		WORDM	5,SPAC,E
0915 0073 0448	SUB	FDB	DOCOL,EMIT,SEMIC
0919 041D 0080		WORDM	3,MI,N
091D		FDB	DOCOL,ZEQU,SEMIC
0921 0073 0915	EQUAL	WORDM	1,,>
0925 05FE 0080		FDB	DOCOL,SWAP,LESS,SEMIC
0929		WORDM	5,SPAC,E
092D 0073 0679	GREAT	FDB	DOCOL,EMIT,SEMIC
0931 05A3 0080		WORDM	3,MI,N
0935		FDB	DOCOL,ZEQU,SEMIC
093D 0073 078C	SPACE	WORDM	1,,>
0941 00E3 0080		FDB	DOCOL,SWAP,LESS,SEMIC
0945		WORDM	5,SPAC,E

094B 0073 065D	MIN	FDB	DOCOL,OVER,OVER,GREAT,ZBRAN
094F 065D 092D			
0953 020B			
0955 0004		FDB	MIN2-*
0957 0679		FDB	SWAP
0959 066B 0080	MIN2	FDB	DROP,SEMIS
095D		WORDM	3,MA,X
0963 0073 065D	MAX	FDB	DOCOL,OVER,OVER,LESS,ZBRAN
0967 065D 05A3			
096B 020B			
096D 0004		FDB	MAX2-*
096F 0679		FDB	SWAP
0971 066B 0080	MAX2	FDB	DROP,SEMIS
0975		WORDM	4,-DU,P
097C 0073 068A	DDUP	FDB	DOCOL,DUP,ZBRAN
0980 020B			
0982 0004		FDB	DDUP2-*
0984 068A		FDB	DUP
0986 0080	DDUP2	FDB	SEMIS
0988		WORDM	8,TRavers,E
0993 0073 0679	TRAV	FDB	DOCOL,SWAP
0997 065D 041D	TRAV2	FDB	OVER,PLUS,CLITER
099B 01EE			
099D 7F		FCB	\$7F
099E 065D 06B3		FDB	OVER,CAT,LESS,ZBRAN
09A2 05A3 020B			
09A6 FFF1		FDB	TRAV2-*
09A8 0679 066B		FDB	SWAP,DROP,SEMIS
09AC 0080			
09AE		WORDM	6,LATES,T
09B7 0073 0877	LATEST	FDB	DOCOL,CURENT,AT,AT,SEMIS
09BB 06A7 06A7			
09BF 0080			
09C1		WORDM	3,LF,A
09C7 0073 01EE	LFA	FDB	DOCOL,CLITER
09CB 04		FCB	4
09CC 0915 0080		FDB	SUB,SEMIS
09D0		WORDM	3,CF,A
09D6 0073 077B	CFA	FDB	DOCOL,TWO,SUB,SEMIS
09DA 0915 0080			
09DE		WORDM	3,NF,A
09E4 0073 01EE	NFA	FDB	DOCOL,CLITER
09E8 05		FCB	5
09E9 0915 0773		FDB	SUB,ONE,MINUS,TRAV,SEMIS
09ED 0448 0993			
09F1 0080			
09F3		WORDM	3,PF,A
09F9 0073 0773	PFA	FDB	DOCOL,ONE,TRAV,CLITER
09FD 0993 01EE			
0A01 05		FCB	5
0A02 041D 0080		FDB	PLUS,SEMIS
0A06		WORDM	4,!CS,P

0A0D 0073 01BB	SCSP	FDB	DOCOL,SPAT,CSP,STORE,SEMIS
0A11 08AC 06EF			
0A15 0080			
0A17		WORDM	6,?ERRO,R
0A20 0073 0679	QERR	FDB	DOCOL,SWAP,ZBRAN
0A24 020B			
0A26 0008		FDB	QERR2-*
0A28 0F18 01FF		FDB	ERROR,BRAN
0A2C 0004		FDB	QERR3-*
0A2E 066B	QERR2	FDB	DROP
0A30 0080	QERR3	FDB	SEMIS
0A32		WORDM	5,?COM,P
0A3A 0073 0883	QCOMP	FDB	DOCOL,STATE,AT,ZEQU,CLITER
0A3E 06A7 05FE			
0A42 01EE			
0A44 11		FCB	\$11
0A45 0A20 0080		FDB	QERR,SEMIS
0A49		WORDM	5,?EXE,C
0A51 0073 0883	QEXEC	FDB	DOCOL,STATE,AT,CLITER
0A55 06A7 01EE			
0A59 12		FCB	\$12
0A5A 0A20 0080		FDB	QERR,SEMIS
0A5E		WORDM	6,?PAIR,S
0A67 0073 0915	QPAIRS	FDB	DOCOL,SUB,CLITER
0A6B 01EE			
0A6D 13		FCB	\$13
0A6E 0A20 0080		FDB	QERR,SEMIS
0A72		WORDM	4,?CS,P
0A79 0073 01BD	QCSP	FDB	DOCOL,SPAT,CSP,AT,SUB,CLITER
0A7D 08AC 06A7			
0A81 0915 01EE			
0A85 14		FCB	\$14
0A86 0A20 0080		FDB	QERR,SEMIS
0A8A		WORDM	8,?LOADIN,G
0A95 0073 0831	QLOAD	FDB	DOCOL,BLK,AT,ZEQU,CLITER
0A99 06A7 05FE			
0A9D 01EE			
0A9F 16		FCB	\$16
0AA0 0A20 0080		FDB	QERR,SEMIS
0AA4		WORDM	7,COMPIL,E
0AAE 0073 0A3A	COMPIL	FDB	DOCOL,QCOMP,FROMR,DUP,TWOP,TOR,AT,COMMA,SEMIS
0AB2 0647 068A			
0AB6 048E 0639			
0ABA 06A7 08F4			
0ABE 0080			
0AC0		WORDM	1,,[,IMMEDIATE
0AC4 0073 076B	LBRAK	FDB	DOCOL,ZERO,STATE,STORE,SEMIS
0AC8 0863 06EF			
0ACC 0080			
0ACE		WORDM	1,,],NOIM
0AD2 0073 01EE	RBRAK	FDB	DOCOL,CLITER
0AD6 C0		FCE	\$CO

0AD7 0883 06BF	FDB	STATE,STORE,SEMIS
0ADB 0080		
0ADD	WORDM	6,SMUDG,E
0AE6 0073 09B7	SMUDGE	FDB DOCOL,LATEST,CLITER
0AEA 01EE		
0AEC 20	FCB	\$20
0AED 0715 0080	FDB	TOGGLE,SEMIS
0AF1	WORDM	3,HE,X
0AF7 0073 01EE	HEX	FDB DOCOL,CLITER
0AFB 10	FCB	16
0AFC 088E 06BF	FDB	BASE,STORE,SEMIS
0B00 0080		
0B02	WORDM	7,DECIMA,L
0B0C 0073 01EE	DEC	FDB DOCOL,CLITER
0B10 0A	FCB	10
0B11 088E 06BF	FDB	BASE,STORE,SEMIS
0B15 0080		
0B17	WORDM	7,(;CODE,)
0B21 0073 0647	PSCODE	FDB DOCOL,FROMR,LATEST,PFA,CFA,STORE,SEMIS
0B25 09B7 09F9		
0B29 09D6 06BF		
0B2D 0080		
0B2F	WORDM	5,;COD,E,IMMEDIATE
0B37 0073 0A79	SEMIC	FDB DOCOL,QCSP,COMPIL,PSCODE,SMUDGE,LBRAK,QSTACK,SEMIS
0B3B 0AAE 0B21		
0B3F 0AE6 0AC4		
0B43 0C5D 0080		
* NOTE : QSTACK is replaced by ASSEMBLER in versions with one.		
0B47	WORDM	5,COUN,T,NOIM
0B4F 0073 068A	COUNT	FDB DOCOL,DUP,ONEP,SWAP,CAT,SEMIS
0B53 047F 0679		
0B57 06B3 0080		
0B5B	WORDM	4,TYP,E
0B62 0073 097C	TYPE	FDB DOCOL,DDUP,ZERAN
0B66 020B		
0B68 0018	FDB	TYPE3-*
0B6A 065D 041D	FDB	OVER,PLUS,SWAP,XDO
0B6E 0679 0261		
0B72 0270 06B3	TYPE2	FDB I,CAT,EMIT,XLOOP
0B76 00B3 0228		
0B7A FFF8	FDB	TYPE2-*
0B7C 01FF	FDB	ERAN
0B7E 0004	FDB	TYPE4-*
0B80 066B	TYPE3	FDB DROP
0B82 0080	TYPE4	FDB SEMIS
0B84	WORDM	9,-TRAILIN,G
0B90 0073 068A	DTRAIL	FDB DOCOL,DUP,ZERO,XDO
0B94 076B 0261		
0B98 065D 065D	DTRAL2	FDB OVER,OVER,PLUS,ONE,SUB,CAT,BL
0B9C 041D 0773		
0BA0 0915 06B3		
0BA4 078C		

OBA6 0915 020B	FDB	SUB,ZBRAN
OBAA 0008	FDS	DTRAL3-*
OEAC 062E 01FF	FDB	LEAVE,BRAN
OBBO 0006	FDB	DTRAL4-*
OBDB 0773 0915	DTRAL3	FDB ONE,SUB
OBEB 0228	DTRAL4	FDE XLOOP
OBEB FFE0	FDB	DTRAL2-*
OBBA 0080	FDB	SEMIS
OBBC C1	OBEC NEXTNM	SET *
OBBD A2	FCB	SCI
OBBE 0B84	FCB	\$80+''
OBBC LASTNM	SET	LASTNM
OBCO 0073 01EE	QUOTE	FDB DOCOL,CLITER
OBG4 22	FCB	\$22 quote
OBG5 0883 06A7	FDB	STATE,AT,ZBRAN
OBG9 020B		
OBGB 0014	FDB	QUOTE1-*
OBGD 0AAE 0BF9	FDB	COMPIL,PQUOTE,WORD,HERE,CAT,ONEP,ALLOT,BRAN
OBDD 0DED 08D8		
OBDD 06B3 047F		
OBDD 08E8 01FF		
OBDD 0014	FDB	QUOTE2-*
OBDF 0DED 08D8	QUOTE1	FDB WORD,HERE,HERE,CAT,ONEP,PAD,SWAP,CMOVE,PAD
OBEE 08D8 06B3		
OBEE 047F 0DDB		
OBED 0679 036D		
OBEF 0DDB		
OBF1 0080	QUOTE2	FDB SEMIS
OBF3 NEXTNM	SET	*
OBF3 83	FCB	\$83
OBF4 28 22	FCC	/("./
OBF6 A9	FCB	\$80+')
OBF7 OBBC	FDB	LASTNM
OBF3 LASTNM	SET	NEXTNM
OBF9 0073 0654	PQUOTE	FDB DOCOL,R,DUP,CAT,ONEP,FROMR,PLUS,TOP,SEMIS
OBFD 068A 06B3		
OC01 047F 0647		
OC05 041D 0639		
OC09 0080		
OCOB NEXTNM	SET	*
OCOB 84	FCB	\$84
OCOC 28 2E 22	FCC	/("./
OCOF A9	FCB	\$80+')
OC10 OBF3	FDB	LASTNM
OCOB LASTNM	SET	NEXTNM
OC12 0073 0654	PDOTQ	FDB DOCOL,R,COUNT,DUP,ONEP,FROMR,PLUS,TOR,TYPE,SEMIS
OC16 0B4F 068A		
OC1A 047F 0647		
OC1E 041D 0639		
OC22 0E62 0080		
OC26 NEXTNM	SET	*

0C26 C2	FCB	\$C2	IMMEDIATE
0C27 2E	FCB	'.	
0C28 A2	FCB	\$80+""	
0C29 0C0B	FDB	LASTNM	
0C26 LASTNM	SET	NEXTNM	
0C2B 0073 01EE	DOTQ	FDB	DOCOL,CLITER
0C2F 22		FCB	\$22 quote
0C30 0883 06A7		FDB	STATE,AT,ZERAN
0C34 020B			
0C36 0014		FDB	DOTQ1-*
0C38 0AAE 0C12		FDB	COMPIL,PDOTQ,WORD,HERE,CAT,ONEP,ALLOT,BRAN
0C3C 0DED 08D8			
0C40 06E3 047F			
0C44 08E8 01FF			
0C48 000A		FDB	DOTQ2-*
0C4A 0DED 08D8	DOTQ1	FDB	WORD,HERE,COUNT,TYPE
0C4E 0B4F 0B62			
0C52 0080	DOTQ2	FDB	SEMIS
0C54		WORDM	6,?STAC,K maachine dependent
0C5D 0073 01E7	QSTACK	FDB	DOCOL,LIT
0C61 003B		FDB	SINIT-PRGBGN
0C63 07C7 06A7		FDB	PORIG,AT,SPAT,LESS,ONE,QERR
0C67 01BB 05A3			
0C6B 0773 0A20			
0C6F 01BB	QSTAC2	FDB	SPAT
0C71 08D8 01EE		FDB	HERE,CLITER
0C75 80		FCB	\$80 want 128 spaces higher than dict
0C76 041D 05A3		FDB	PLUS,LESS
0C7A 077B 0A20		FDB	TWO,QERR full stack
0C7E 0080	QSTAC3	FDB	SEMIS
			* WORDM 5,?FRE,E is done by ?STACK in this version
			*QFREE FDB DOCOL,SPAT,HERE,CLITER
			* FCB \$80
			* FDB PLUS,LESS,TWO,QERR,SEMIS
0C80		WORDM	3,RO,T
0C86 0073 0639	ROT	FDB	DOCOL,TOR,SWAP,FROMR,SWAP,SEMIS
0C8A 0679 0647			
0C8E 0679 0080			
0C92		WORDM	6,EXPEC,T
0C9B 0073 065D	EXPECT	FDB	DOCOL,OVER,PLUS,OVER,XDO
0C9F 041D 065D			
0CA3 0261			
0CA5 00D3 068A	EXPEC2	FDB	KEY,DUP,LIT
0CA9 01E7			
0CAB 2020 06B3		FDB	XLINDL,CAT,EQUAL,ZERAN
0CAF 0921 020B			
0CB3 0018		FDB	EXPECZ-*
0CB5 066B 01E7		FDB	DROP,LIT,XLINDE,CAT,FROMR,DROP,OVER,ONEM,TOR,BRAN
0CB9 2021 06B3			
0CBD 0647 066E			
0CC1 065D 049D			
0CC5 0639 01FF			

OCC9 0055		FDB	EXPEC6-*
OCGB 068A 01E7	EXPECZ	FDB	DUP,LIT,XBKSP,CAT
OCCF 201E 06B3			
OCD3 0921 020B		FDB	EQUAL,ZBRAN
OCD7 0022		FDB	EXPEC3-*
OCD9 066B 01E7		FDB	DROP,LIT
OCDD 201F 06B3		FDB	XBKSP,CAT
OCE1 065D 0270		FDB	OVER,I,EQUAL,DUP,FROMR,TWO,SUB,PLUS,TOR,SUB,ERAN
OCE5 0921 068A			
OCE9 0647 077B			
OCED 0915 041D			
OCF1 0639 0915			
OCF5 01FF			
OCF7 0027		FDB	EXPEC6-*
OCF9 068A 01EE	EXPEC3	FDB	DUP,CLITER
OCFD 0D		FCB	\$D (CR)
OCFE 0921 020B		FDB	EQUAL,ZBRAN
OD02 000E		FDB	EXPEC4-*
OD04 062B 066B		FDB	LEAVE,DROP,BL,ZERO,BRAN
OD08 078C 076B			
OD0C 01FF			
OD0E 0004		FDB	EXPEC5-*
OD10 068A	EXPEC4	FDB	DUP
OD12 0270 06CF	EXPEC5	FDB	I,CSTORE,ZERO,I,OMEP,STORE
OD16 076B 0270			
OD1A 047F 06BF			
OD1E 00B3 0228	EXPEC6	FDB	EMIT,XLOOP
OD22 FF83		FDB	EXPEC2-*
OD24 066B 0080		FDB	DROP,SEMIS
OD28		WORDM	5,QUER,Y
OD30 0073 07E9	QUERY	FDB	DOCOL,TIB,AT,COLUMNS,AT,EXPECT,ZERO,IN,STORE,SEMIS
OD34 06A7 08CD			
OD38 06A7 0C9B			
OD3C 076B 083A			
OD40 06BF 0080			
OD44 NEXTNM	SET	*	
OD44 C1	FCB	\$C1	IMMEDIATE
OD45 80	FCB	\$80	(NULL)
OD46 0D28		FDB	LASTNM
OD44 LASTNM	SET	NEXTNM	
OD48 0073 0831	NULL	FDB	DOCOL,BLK,AT,ZBRAN
OD4C 06A7 020B			
OD50 0026		FDB	NULL2-*
OD52 0773 0831		FDB	ONE,BLK,PSTORE,ZERO,IN,STORE,BLK,AT,BSCR,MOD,ZEQU
OD56 0696 076B			
OD5A 083A 06BF			
OD5E 0831 06A7			
OD62 17C5 052D			
OD66 05FE			
	* check for end of screen		
OD68 020B		FDB	ZBRAN
OD6A 0008		FDB	NULL1-*

0D6C 0A51 0647	FDB	QEXEC, FROMR, DROP
0D70 066B		
0D72 01FF	NULL1	FDB BRAN
0D74 0006		FDB NULL3-*
0D76 0647 066B	NULL2	FDB FROMR, DROP
0D7A 0080	NULL3	FDB SEMIS
0D7C		WORDM 4, FIL, L
0D83 0073 0679	FILL	FDB DOCOL, SWAP, TOR, OVER, CSTORE, DUP, ONEP, FROMR, ONE
0D87 0639 065D		
0D8B 06CF 068A		
0D8F 047F 0647		
0D93 0773		
0D95 0915 036D	FDB	SUB, CMOVE, SEMIS
0D99 0080		
0D9B		WORDM 5, ERAS, E
0DA3 0073 076B	ERASE	FDB DOCOL, ZERO, FILL, SEMIS
0DA7 0D83 0080		
0DAB		WORDM 6, BLANK, S
0DB4 0073 078C	BLANKS	FDB DOCOL, BL, FILL, SEMIS
0DB8 0D83 0080		
0DBC		WORDM 4, HOL, D
0DC3 0073 01E7	HOLD	FDB DOCOL, LIT, \$FFFF, HLD, PSTORE, HLD, AT, CSTORE, SEMIS
0DC7 FFFF 08BF		
0DCB 0696 08BF		
0DCF 06A7 06CF		
0DD3 0080		
0DD5		WORDM 3, PA, D
0DDB 0073 08DS	PAD	FDB DOCOL, HERE, CLITER
0DDF 01EE		
0DE1 44		FCB \$44
0DE2 041D 0080		FDB PLUS, SEMIS
0DE6		WORDM 4, WOR, D
0DED 0073 0831	WORD	FDB DOCOL, BLK, AT, ZBRAN
0DF1 06A7 020B		
0DF5 000C		FDB WORD2-*
0DF7 0831 06A7		FDB BLK, AT, BLOCK, BRAN
0DFB 12B0 01FF		
0DFF 0006		FDB WORD3-*
0E01 07E9 06A7	WORD2	FDB TIB, AT
0E05 083A 06A7	WORD3	FDB IN, AT, PLUS, SWAP, ENCLOS, HERE, CLITER
0E09 041D 0679		
0E0D 0318 08D8		
0E11 01EE		
0E13 22	FCB	34
0E14 0DB4 083A	FDB	BLANKS, IN, PSTORE, OVER, SUB, TOR, R, HERE, CSTORE, PLUS
0E18 0696 065D		
0E1C 0915 0639		
0E20 0654 08D8		
0E24 06CF 041D		
0E28 08D8 047F		FDB HERE, ONEP, FROMR, CMOVE, SEMIS
0E2C 0647 036D		
0E30 0080		

0E32		WORDM	8,(NUMBER,)
0E3D 0073	PNUMB	FDB	DOCOL
0E3F 047F 068A	PNUMB2	FDB	ONEP,DUP,TOR,CAT,BASE,AT,DIGIT,ZBRAN
0E43 0639 06E3			
0E47 088E 06A7			
0E4B 0295 020B			
0E4F 002C		FDB	PNUMB4-*
0E51 0679 088E		FDB	SWAP,BASE,AT,USTAR,DROP,ROT,BASE
0E55 06A7 0394			
0E59 066B 0C86			
0E5D 088E			
0E5F 06A7 0394		FDB	AT,USTAR,DPLUS,DPL,AT,ONEP,ZBRAN
0E63 042B 0898			
0E67 06A7 047F			
0E6B 020B			
0E6D 0008		FDB	PNUMB3-*
0E6F 0773 0898		FDB	ONE,DPL,PSTORE
0E73 0696			
0E75 0647 01FF	PNUMB3	FDB	FROMR,BRAN
0E79 FFC6		FDB	PNUMB2-*
0E7B 0647 0080	PNUMB4	FDB	FROMR,SEVIS
0E7F		WORDM	6,NUMBE,R
0E88 0073 076B	NUMB	FDB	DOCOL,ZERO,ROT,DUP,ONEP,CAT,CLITER
0E8C 076B 0C86			
0E90 068A 047F			
0E94 06B3 01EE			
0E98 2D		FCB	'- minus sign
0E99 0921 068A		FDB	EQUAL,DUP,TOR,PLUS,LIT,\$FFFF
0E9D 0639 041D			
0EA1 01E7 FFFF			
0EA5 0898 06EF	NUMB1	FDB	DPL,STORE,PNUMB,DUP,CAT,BL,SUB,ZBRAN
0EA9 0E3D 068A			
0EAD 06B3 078C			
0EB1 0915 020B			
0EB5 0015		FDB	NUMB2-*
0EB7 068A 06B3		FDB	DUP,CAT,CLITER
0EBB 01EE			
0EBD 2E		FCB	'.
0EBE 0915 076B		FDB	SUB,ZERO,QERR,ZERO,BRAN
0EC2 0A20 076B			
0EC6 01FF			
0EC8 FFDD		FDB	NUMB1-*
0ECA 066B 0647	NUMB2	FDB	DROP,FROMR,ZBRAN
0ECE 020B			
0ED0 0004		FDB	NUMB3-*
0ED2 0461		FDB	DMINUS
0ED4 0080	NUMB3	FDB	SEVIS
0ED6		WORDM	5,-FIN,D
0EDE 0073 078C	DFIND	FDB	DOCOL,BL,WORD,HERE,CONTXT,AT,AT,PFIND,DUP,ZEQU,ZBRAN
0EE2 0DED 08D8			
0EE6 0869 06A7			
0EEA 06A7 02C8			

0EEE 068A 05FE		
0EF2 020B		
0EF4 000A	FDB	DFIND2-*
0EF6 066B 08D8	FDB	DROP,HERE,LATEST,PFIND
0EFA 09E7 02C8		
0EFE 0080	DFIND2	FDB SEMIS
0F00		WORDM 7,(ABORT,)
0FOA 0073 1120	PABORT	FDB DOCOL,ABORT,SEMIS
0FOE 0080		
0F10		WORDM 5,ERRO,R
0F18 0073 0803	ERROR	FDB DOCOL,WARN,AT,ZLESS,ZBRAN
0F1C 06A7 0611		
0F20 020B		
* WARNING is -1 to abort, 0 to print error #, and >1 to print		
* error message from the message SCreen on disk		
0F22 0004	FDB	ERROR2-*
0F24 0FOA	FDB	PABORT
0F26 08D8 0B4F	ERROR2	FDB HERE,COUNT,TYPE,PDOTQ
0F2A 0B62 0C12		
0F2E 04 07	FCB	4,7 (BELL)
0F30 20 3F 20	FCC	" ? "
0F33 1372 01CA	FDB	MESS,SPSTOR,IN,AT,BLK,AT,QUIT,SEMIS
0F37 083A 06A7		
0F3B 0831 06A7		
0F3F 10F2 0080		
0F43		WORDM 3, ID, .
0F49 0073 0DDB	IDDOT	FDB DOCOL,PAD,CLITER
0F4D 01EE		
0F4F 20	FCB	32
0F50 01EE	FDB	CLITER
0F52 5F	FCB	\$5F
0F53 0D83 068A	FDB	FILL,DUP,PFA,LFA,OVER,SUB,PAD,SWAP,CMOVE
0F57 09F9 09C7		
0F5B 065D 0915		
0F5F 0DDB 0679		
0F63 036D		
0F65 0DDB 0B4F		FDB PAD,COUNT,CLITER
0F69 01EE		
0F6B 1F	FCB	31
0F6C 03ED 0B62	FDB	AND,TYPE,SPACE,SEMIS
0F70 093D 0080		
0F74		WORDM 6, CREAT,E
0F7D 0073 0EDE	CREATE	FDB DOCOL,DFIND,ZBRAN
0F81 020B		
0F83 001A	FDB	CREAT2-*
0F85 066B 0C12	FDB	DROP,PDOTQ
0F89 08 07	FCB	8,7 (BELL)
0F8B 72 65 64 65	FCC	"redef: "
0F8F 66 3A 20		
0F92 09E4 0F49		FDB NFA, IDDOT,CLITER
0F96 01EE		
0F98 04	FCB	4

0F99 1372 093D	FDB	MESS,SPACE
0F9D 08D8 068A	CREAT2 FDB	HERE,DUP,CAT,WIDTH,AT,MIN,ONEP,ALLOT,DUP,CLITER
0FA1 06B3 07F5		
0FA5 06A7 094B		
0FA9 047F 08E8		
0FAD 068A 01EE		
0FB1 A0	FCB	\$AO
0FB2 0715 08D8	FDB	TOGGLE,HERE,ONE,SUB,CLITER
0FB6 0773 0915		
0FEA 01EE		
0FBC 80	FCB	\$80
0FBD 0715 0937	FDB	TOGGLE,LATEST,COMMA,CURENT,AT,STORE,HERE,TWOP
0FC1 08F4 0877		
0FC5 06A7 06BF		
0FC9 08D8 048E		
0FCD 08F4 0080	FDB	COMMA,SEVIS
0FD1	WORDM	9,[COMPILE,],IMMEDIATE
0FDD 0073 0EDE	BCOMP	DOCOL,DFIND,ZEQU,ZERO,QERR,DROP,CFA,COMMA,SEVIS
0FE1 05FE 076B		
0FE5 0A20 066B		
0FE9 09D6 08F4		
0FED 0080		
0FFF	WORDM	7,LITERA,L,IMMEDIATE
0FF9 0073 0883	LITER FDB	DOCOL,STATE,AT,ZBRAN
0FFD 06A7 020B		
1001 0008	FDB	LITER2-*
1003 0AAE 01E7	FDB	COMPIL,LIT,COMMA
1007 08F4		
1009 0080	LITER2 FDB	SEVIS
100B	WORDM	8,DLITERA,L,IMMEDIATE
1016 0073 0883	DLITER FDB	DOCOL,STATE,AT,ZBRAN
101A 06A7 020B		
101E 0008	FDB	DLITE2-*
1020 0679 OFF9	FDB	SWAP,LITER,LITER
1024 OFF9		
1026 0080	DLITE2 FDB	SEVIS
1028	WORDM	9,INTERPRET,NOIM
1034 0073	INTERP FDB	DOCOL
1036 0EDE 020B	INTER2 FDB	DFIND,ZBRAN
103A 001E	FDB	INTER5-*
103C 0883 06A7	FDB	STATE,AT,LESS
1040 05A3		
1042 020B	FDB	ZBRAN
1044 000A	FDB	INTER3-*
1046 09D6 08F4	FDB	CFA,COMMA,BRAN
104A 01FF		
104C 0006	FDB	INTER4-*
104E 09D6 0091	INTER3 FDB	CFA,EXEC
1052 0C5D 01FF	INTER4 FDB	QSTACK,BRAN
1056 001A	FDB	INTER7-*
1058 08D8 0E88	INTER5 FDB	HERE,NUMB,DPL,AT,ONEP,ZBRAN
105C 0898 06A7		

1060 047F 020B			
1064 0008	FDB	INTER6-*	
1066 1016 01FF	FDB	DLITER,BRAN	
106A 0006	FDB	INTER7-*	
106C 066B OFF9	INTER6 FDB	DROP,LITER	
1070 0C5D 01FF	INTER7 FDB	QSTACK,BRAN	
1074 FFC2	FDB	INTER2-*	
	* FDB SEMIS never executed		
1076	WORDM	9,IMMEDIAT,E	
1082 0073 09E7	IMMED	FDB DOCOL,LATEST,CLITER	
1086 01EE			
1088 40	FCB	\$40	
1089 0715 0080	FDB	TOGGLE,SEMIS	
108D	WORDM	10,VOCABULAR,Y	
109A 0073 06E4	VOCAB FDB	DOCOL,BUILDS,LIT,\$21AO,COMMA,CURENT,AT,CFA,COMIA	
109E 01E7 81A0			
10A2 08F4 0377			
10A6 06A7 09D6			
10AA 08F4			
10AC 08D8 0827	FDB	HERE,VOCLIN,AT,COMMA,VOCLIN,STORE,DOES	
10B0 06A7 08F4			
10B4 0327 06BF			
10B8 06F4			
10BA 048E 0869	DOVOC FDB	TWOP,CONTXT,STORE,SEMIS	
10BE 06BF 0080			
10C2 0000	FDB	0	
10C4	WORDM	11,DEFINITION,S	
10D2 0073 0869	DEFIN FDB	DOCOL,CONTXT,AT,CURENT,STORE,SEMIS	
10D6 06A7 0877			
10DA 06BF 0080			
10DE	WORDM	1,,(,IMMEDIATE	
10E2 0073 01EE	PAREN FDB	DOCOL,CLITER	
10E6 29	FCB	')	
10E7 0DED 0080	FDB	WORD,SEMIS	
10EB	WORDM	4,QUI,T,NOIM	
10F2 0073 076B	QUIT FDB	DOCOL,ZERO,BLK,STORE,LBRAK	
10F6 0831 06BF			
10FA 0AC4			
	* Here is outer interpreter which gets line of input, does it, and		
	* then prints "OK" and repeats.		
10FC 01D8 0100	QUIT2 FDB	RPSTOR,CR,QUERY,INTERP,STATE,AT,ZEQU,ZBRAN	
1100 0D30 1034			
1104 0883 06A7			
1108 05FE 020B			
110C 0008	FDB	QUIT3-*	
110E 0C12	FDB	PDOTQ	
1110 03	FCB	3	
1111 20 4F 4B	FCC	" OK"	
1114 01FF	QUIT3 FDB	BRAN	
1116 FFE6	FDB	QUIT2-*	
	* FDB SEMIS never executed		
1118	WORDM	5,ABOR,T	

1120 0073 01CA	ABORT	FDB	DOCOL,SPSTOR,DEC,DRZERO,CR,PDOTQ
1124 0E0C 1812			
1128 0100 0C12			
112C 12	FCB	18	
112D 36 38 27 46	FCC	"68'FORTH-09 VERS #"	
1131 4F 52 54 48			
1135 2D 30 39 20			
1139 56 45 52 53			
113D 20 23			
113F 01E7 0008	FDB	LIT,VERSON,DUP,CAT,DOT,PDOTQ	
1143 068A 06B3			
1147 167C 0C12			
114B 01	FCB	1	
114C 2E	FCB	'.	
114D 047F 06B3	FDB	ONEP,CAT,DOT	
1151 167C			
1153 076B 083A	FDB	ZERO,IN,STORE,ZERO,BLK,STORE	
1157 06BF 076B			
115B 0831 06BF			
115F 2058 10D2	FDB	FORTH,DEFIN,LIT,IFCOLD,CAT,ZERAN	
1163 01E7 0138			
1167 06B3 020B			
116B 000C	FDB	ABORTC-*	
116D 076B 01E7	FDB	ZERO,LIT,IFCOLD,CSTORE,GO	
1171 0138 06CF			
1175 117E			
1177 10F2	ABORTC FDB	QUIT	
	* FDB SEMIS never executed		
1179		WORDM 2,G,O	
117E 0073 01E7	GO	FDB	DOCOL,LIT,XMSGES,AT,THREE,PLUS,DRZERO,LOAD,SEMIS
1182 202E 06A7			
1186 0783 041D			
118A 1812 13C8			
118E 0080			

PAG

*

* Here is stuff which gets copied to ram in user space

1190 C5	RAM	FCB	\$C5	5, IMMEDIATE
1191 46 4F 52 54		FCC	"FORT"	
1195 C8		FCB	\$80+'H	
1196 1A34		FDB	NOOP-7	LINK "BACK"
1198 0700 10EA	RFORTH	FDB	DODOES,DOVOC,\$81A0,TASK-7	
119C 81A0 207E				
11A0 0000		FDB	0	
11A2 28 43 29 20		FCC	"(C) Talbot Microsystems 1980"	
11A6 54 61 6C 62				
11AA 6F 74 20 4D				
11AE 69 63 72 6F				
11B2 73 79 73 74				
11B6 65 6D 73 20				
11BA 31 39 38 30				
11BE 84		FCB	\$84	
11BF 54 41 53		FCC	"TAS"	
11C2 CB		FCB	\$80+'K	
11C3 2050		FDB	FORTH-S link "back" to FORTH	
11C5 0073 0080	RTASK	FDB	DOCOL,SEMIS	
11C9 52 2E 20 4A	ERAM	FCC	"R. J. Talbot, Jr."	
11CD 2E 20 54 61				
11D1 6C 62 6F 74				
11D5 2C 20 4A 72				
11D9 2E				

PAG

*
* Disc primitives :
11DA WORDM 3,US,E
11E0 074A 004B USE FDB DOCOM,XUSE
11E4 WORDM 4,PRE,V
11EB 074A 004D PREV FDB DOCON,XPREV
11EF WORDM 4,+EU,F
11F6 0073 17B9 PBUF FDB DOCOL,BEUF
11FA 01EE FDB CLITER
11FC 04 FCB 4
11FD 041D FDB PLUS
11FF 041D 068A FDB PLUS,DUP,BEUF,PLUS,CLITER
1203 17B9 041D
1207 01EE
1209 04 FCB 4
120A 041D 07A4 FDB PLUS,LIMIT,GREAT,ZERAN
120E 092D 020B
1212 0006 FDB PBUF2-*
1214 066B 0798 FDB DROP,FIRST
1218 068A 11EB PBUF2 FDB DUP,PREV,AT,SUB,SEVIS
121C 06A7 0915
1220 0080
1222 WORDM 6,UPDAT,E
122B 0073 11EB UPDATE FDB DOCOL,PREV,AT,AT,LIT,\$8000,OR,PREV,AT,STORE,SEVIS
122F 06A7 06A7
1233 01E7 8000
1237 03FF 11EB
123B 06A7 06BF
123F 0080
1241 WORDM 13,EMPTY-BUFFER,S
1251 0073 0798 MTBUF FDB DOCOL,FIRST,LIMIT,OVER,SUB,ERASE,SEVIS
1255 07A4 065D
1259 0915 0DA3
125D 0080
125F WORDM 6,BUFFE,R
1268 0073 11E0 BUFFER FDB DOCOL,USE,AT,DUP,TOR
126C 06A7 068A
1270 0639
1272 11F6 020B BUFFR2 FDB PBUF,ZBRAN
1276 FFFC FDB BUFFR2-*
1278 11E0 06BF FDB USE,STORE,R,AT,ZLESS,ZBRAN
127C 0654 06A7
1280 0611 020B
1284 0014 FDB BUFFR3-*
1286 0654 048E FDB R,TWOP,R,AT,LIT,\$7FFF,AND,ZERO,RW
128A 0654 06A7
128E 01E7 7FFF
1292 03ED 076B
1296 186A
1298 0654 06BF BUFFR3 FDB R,STORE,R,PREV,STORE,FROMR,TWOP,SEVIS
129C 0654 11EB

12A0 06BF 0647			
12A4 048E 0080			
12A8	WORDM	5 ,BLOC,K	
12B0 0073 085B	BLOCK	FDB DOCOL,OFFSET,AT,PLUS,TOR,PREV,AT,DUP,AT,R,SUB	
12B4 06A7 041D			
12B8 0639 11EB			
12BC 06A7 068A			
12C0 06A7 0654			
12C4 0915			
12C6 068A 041D		FDB DUP,PLUS,ZBRAN	
12CA 020B			
12CC 0034		FDB BLOCK5-*	
12CE 11F6 05FE	BLOCK3	FDB PBUF,ZEQU,ZBRAN	
12D2 020B			
12D4 0014		FDB BLOCK4-*	
12D6 066B 0654		FDB DROP,R,BUFFER,DUP,R,ONE,RW,TWO,SUB	
12DA 1268 068A			
12DE 0654 0773			
12E2 186A 077B			
12E6 0915			
12E8 068A 06A7	BLOCK4	FDB DUP,AT,R,SUB,DUP,PLUS,ZEQU,ZBRAN	
12EC 0654 0915			
12F0 068A 041D			
12F4 05FE 020B			
12F8 FFD6		FDB BLOCK3-*	
12FA 068A 11EB		FDB DUP,PREV,STORE	
12FE 06BF			
1300 0647 066B	BLOCK5	FDB FROMR,DROP,TWOP,SEmis	
1304 048E 0080			
1308	WORDM	5 ,FLUS,H	
1310 0073 07A4	FLUSH	FDB DOCOL,LIMIT,FIRST,SUB,EBUF,CLITER	
1314 0798 0915			
1318 17B9 01EE			
131C 04	FCB	\$04	
131D 041D 051D		FDB PLUS,SLASH,ZERO,XDO	
1321 076B 0261			
1325 01E7	FLUSH1	FDB LIT	
1327 7FFF		FDB \$7FFF	
1329 1268 066B		FDB BUFFER,DROP	
132D 0228		FDB XLOOP	
132F FFF6		FDB FLUSH1-*	
1331 0080		FDB SEMIS	
1333	WORDM	6 ,(LINE,)	
133C 0073 0639	PLINE	FDB DOCOL,TOR,CLITER	
1340 01EE			
1342 40	FCB	\$40	
1343 17B9 053D		FDB BEUF,SSMOD,FROMR,SCRBLK,PLUS,BLOCK,PLUS,CLITER	
1347 0647 17DB			
134B 041D 12B0			
134F 041D 01EE			
1353 40	FCB	\$40	
1354 0080		FDB SEMIS	

1356		WORDM	5,.LIN,E
135E 0073 133C	DLINE	FDB	DOCOL,PLINE,DTRAIL,TYPE,SEMS
1362 0B90 0B62			
1366 0080			
1368		WORDM	7,MESSAG,E
1372 0073 0803	MESS	FDB	DOCOL,WARN,AT,ZBRAN
1376 06A7 020B		FDB	MESS3-*
137A 0028		FDB	DDUP,ZBRAN
137C 097C 020B		FDB	MESS4-*
1380 003F		FDB	LIT,XMSGBS,AT
1382 01E7 202E		FDS	
1386 06A7		FDB	OFFSET,AT,TOR,ZERO,OFFSET,STORE,DLINE,FROMR,OFFSET,STORE
138C 0639 076B			
1390 085B 06BF			
1394 135E 0647			
1398 085B 06BF			
139C 0100 01FF		FDE	CR,BRAN
13A0 001F		FDB	MESS4-*
13A2 0C12	MESS3	FDB	PDOTQ
13A4 04		FCB	4
13A5 65 72 72 20		FCC	"err "
13A9 01EE		FDB	CLITER
13AB 23		FCB	'#
13AC 088E 06A7		FDB	BASE,AT,CLITER
13B0 01EE			
13B2 0A		FCB	10 DECIMAL
13B3 0921 05FE		FDB	EQUAL,ZEQU,PLUS if = 10, add 0, if = 16, add 1 TO MAKE '\$
13B7 041D			
13B9 00B3 093D		FDB	EMIT,SPACE
13BD 167C		FDE	DOT
13BF 0080	MESS4	FDB	SEMS
13C1		WORDM	4,LCA,D input: scr #
13C8 0073 0831	LOAD	FDB	DOCOL,BLK,AT,TOR,IN,AT,TOR,ZERO,IN,STORE,SCREBLK,BLK
13CC 06A7 0639			
13D0 083A 06A7			
13D4 0639 076B			
13D8 083A 06BF			
13DC 17DB 0831			
13E0 06BF 1034		FDB	STORE,INTERP,FROMR,IN,STORE,FROMR,BLK,STORE,SEMS
13E4 0647 083A			
13E8 06BF 0647			
13EC 0831 06BF			
13F0 0080			
13F2		WORDM	3,--,>,IMMEDIATE
13F8 0073 0A95	ARROW	FDB	DOCOL,QLOAD,ZERO,IN,STORE,BSCR,BLK,AT,OVER,MOD
13FC 076B 083A			
1400 06BF 17C5			
1404 0831 06A7			
1408 065D 052D			
140C 0915 0831		FDB	SUB,BLK,PSTORE,SEMS
1410 0696 0080			

1414		WORDM	1,,`IMMEDIATE
1418 0073 0EDE	TICK	FDB	DOCOL,DFIND,ZEQU,ZERO,QERR,DROP,LITER,SEMIC
141C 05FE 076B			
1420 0A20 066B			
1424 0FF9 0080			
1428		WORDM	6,FORGE,T,NOIM
1431 0073 0877	FORGET	FDB	DOCOL,CURRENT,AT,CONTEXT,AT,SUB,CLITER
1435 06A7 0869			
1439 06A7 0915			
143D 01EE			
143F 18		FCB	\$18
1440 0A20 1418		FDB	QERR,TICK,DUP,FENCE,AT,LESS,CLITER
1444 068A 080F			
1448 06A7 05A3			
144C 01EE			
144E 15		FCB	\$15
144F 0A20 068A		FDB	QERR,DUP,LIT,SINIT,AT,GREAT,CLITER
1453 01E7 003B			
1457 06A7 092D			
145B 01EE			
145D 15		FCB	\$15
145E 0A20 068A		FDB	QERR,DUP,NFA,DP,STORE,LFA,AT,CONTEXT,AT,STORE,SEMIC
1462 09E4 0818			
1466 06BF 09C7			
146A 06A7 0869			
146E 06A7 06EF			
1472 0080	*		
1474		WORDM	4,BAC,K
147B 0073 08D8	BACK	FDB	DOCOL,HERE,SUB,COMMA,SEMIC
147F 0915 08F4			
1483 0080			
1485		WORDM	5,BEGI,N,IMMEDIATE
148D 0073 0A3A	BEGIN	FDB	DOCOL,QCOMP,HERE,ONE,SEMIC
1491 08D8 0773			
1495 0080			
1497		WORDM	5,ENDI,F,IMMEDIATE
149F 0073 0A3A	ENDIF	FDB	DOCOL,QCOMP,TWO,QPAIRS,HERE,OVER,SUB,SWAP,STORE,SEMIC
14A3 077B 0A67			
14A7 08D8 065D			
14AB 0915 0679			
14AF 06EF 0080			
14B3		WORDM	4,THE,N,IMMEDIATE
14BA 0073 149F	THEN	FDB	DOCOL,ENDIF,SEMIC
14BE 0080			
14C0		WORDM	2,D,O,IMMEDIATE
14C5 0073 0AAE	DO	FDB	DOCOL,COMPIL,XDO,HERE,THREE,SEMIC
14C9 0261 08D8			
14CD 0783 0080			
14D1		WORDM	4,LOO,P,IMMEDIATE
14D8 0073 0783	LOOP	FDB	DOCOL,THREE,QPAIRS,COMPIL,XLOOP,BACK,SEMIC
14DC 0A67 0AAE			

14E0 0228 147B			
14E4 0080			
14E6		WORDM	5,+LOO,P,IMMEDIATE
14EE 0073 0783	PLOOP	FDB	DOCOL,THREE,QPAIRS,COMPIL,XPLOOP,BACK,SEMIS
14F2 0A67 0AAE			
14F6 0239 147B			
14FA 0080			
14FC		WORDM	5,UNTL,L,IMMEDIATE
1504 0073 0773	UNTIL	FDB	DOCOL,ONE,QPAIRS,COMPIL,ZBRAN,BACK,SEMIS
1508 0A67 0AAE			
150C 020B 147B			
1510 0080			
1512		WORDM	3,EN,D,IMMEDIATE
1518 0073 1504	END	FDB	DOCOL,UNTIL,SEMIS
151C 0080			
151E		WORDM	5,AGAI,N,IMMEDIATE
1526 0073 0773	AGAIN	FDB	DOCOL,ONE,QPAIRS,COMPIL,BRAN,BACK,SEMIS
152A 0A67 0AAE			
152E 01FF 147B			
1532 0080			
1534		WORDM	6,REPEA,T,IMMEDIATE
153D 0073 0639	REPEAT	FDB	DOCOL,TOR,TOR,AGAIN,FROMR,FROMR,TWO,SUB,ENDIF,SEMIS
1541 0639 1526			
1545 0647 0647			
1549 077B 0915			
154D 149F 0080			
1551		WORDM	2,I,F,IMMEDIATE
1556 0073 0AAE	IF	FDB	DOCOL,COMPIL,ZBRAN,HERE,ZERO,COMMA,TWO,SEMIS
155A 020B 08D8			
155E 076B 08F4			
1562 077B 0080			
1566		WORDM	4,ELS,E,IMMEDIATE
156D 0073 077B	ELSE	FDB	DOCOL,TWO,QPAIRS,COMPIL,BRAN,HERE,ZERO,COMMA,SWAP
1571 0A67 0AAE			
1575 01FF 08D8			
1579 076B 08F4			
157D 0679			
157F 077B 149F		FDB	TWO,ENDIF,TWO,SEMIS
1583 077B 0080			
1587		WORDM	5,WHL,E,IMMEDIATE
158F 0073 1556	WHILE	FDB	DOCOL,IF,TWOP,SEMIS
1593 048E 0080	*		
1597		WORDM	6,SPACE,S
15A0 0073 076B	SPACES	FDB	DOCOL,ZERO,MAX,DDUP,ZBRAN
15A4 0963 097C			
15A8 020B			
15AA 000C		FDB	SPACE3-*
15AC 076B 0261		FDB	ZERO,XDO
15B0 093D 0228	SPACE2	FDB	SPACE,XLOOP
15B4 FFFC		FDB	SPACE2-*
15B6 0080	SPACE3	FDB	SEMIS

15B3		WORDM	2,<,#
15BD 0073 0DDB	EDIGS	FDB	DOCOL,PAD,HLD,STORE,SEVIS
15C1 08BF 06BF			
15C5 0080			
15C7		WORDM	2,#,>
15CC 0073 066B	EDIGS	FDB	DOCOL,DROP,DROP,HLD,AT,PAD,OVER,SUB,SEVIS
15D0 066B 08BF			
15D4 06A7 0DDB			
15D8 065D 0915			
15DC 0080			
15DE		WORDM	4,SIG,N
15E5 0073 0C86	SIGN	FDB	DOCOL,ROT,ZLESS,ZBRAN
15E9 0611 020B			
15ED 0007		FDB	SIGN2-*
15EF 01EE		FDB	CLITER
15F1 2D		FCB	'-
15F2 0DC3		FDB	HOLD
15F4 0080	SIGN2	FDB	SEVIS
15F6		WORDM	1,,#
15FA 0073 08SE	DIG	FDB	DOCOL,BASE,AT,MSMOD,ROT,CLITER
15FE 06A7 0560			
1602 0C86 01EE			
1606 09		FCB	9
1607 065D 05A3		FDB	OVER,LESS,ZBRAN
160B 020B			
160D 0007		FDB	DIG2-*
160F 01EE		FDB	CLITER
1611 07		FCB	7
1612 041D		FDB	PLUS
1614 01EE	DIG2	FDB	CLITER
1616 30		FCB	'0 ascii zero
1617 041D 0DC3		FDB	PLUS,HOLD
161B 0080		FDB	SEVIS
161D		WORDM	2,#,S
1622 0073	DIGS	FDB	DOCOL
1624 15FA 065D	DIGS2	FDB	DIG,OVER,OVER,OR,ZEQU,ZBRAN
1628 065D 03FF			
162C 05FE 020B			
1630 FFF4		FDB	DIGS2-*
1632 0080		FDB	SEVIS
1634		WORDM	3,D.,R
163A 0073 0639	DDOTR	FDB	DOCOL,TOR,SWAP,OVER,DABS,EDIGS,DIGS,SIGN
163E 0679 065D			
1642 0591 15BD			
1646 1622 15E5			
164A 15CC 0647		FDB	EDIGS,FROMR,OVER,SUB,SPACES,TYPE,SEVIS
164E 065D 0915			
1652 15A0 0B62			
1656 0080			
1658		WORDM	2,,,R
165D 0073 0639	DOTR	FDB	DOCOL,TOR,STOD,FROMR,DDOTR,SEVIS
1661 05C1 0647			

1665 163A 0080				
1669		WORDM	2,D,.	
166E 0073 076B	DDOT	FDB	DOCOL,ZERO,DDOTR,SPACE,SEMIC	
1672 163A 093D				
1676 0080				
1678		WORDM	1,..	
167C 0073 05C1	DOT	FDB	DOCOL,STOD,DDOT,SEMIC	
1680 166E 0080				
1684		WORDM	1,,?	
1688 0073 06A7	QUEST	FDB	DOCOL,AT,DOT,SEMIC	
168C 167C 0080				
1690	*			
		WORDM	4,LIS,T	
1697 0073 0E0C	LIST	FDB	DOCOL,DEC,CR,DUP,SCR,STORE,PDOTQ	
169B 0100 068A				
169F 084E 06BF				
16A3 0C12				
16A5 06		FCB	6	
16A6 53 43 52 20		FCC	"SCR # "	
16AA 23 20				
16AC 167C 01EE		FDB	DOT,CLITER	
16B0 10		FCB	16	
16B1 076B 0261		FDB	ZERO,XDO	
16B5 0100 0270	LIST2	FDB	CR,I,THREE	
16B9 0783				
16BB 165D 093D		FDB	DOTR,SPACE,I,SCR,AT,PLINE,TYPE,CLITER	
16BF 0270 084E				
16C3 06A7 133C				
16C7 0B62 01EE				
16CB 3C		FCB	\$3C	
16CC 00B3 0228		FDB	EMIT,XLOOP	
16D0 FFE5		FDB	LIST2-*	
16D2 0100 0080		FDB	CR,SEMIC	
16D6		WORDM	4,DUM,P	
16DD 0073 065D	DUMP	FDB	DOCOL,OVER,PLUS,SWAP,XDO	
16E1 041D 0679				
16E5 0261				
16E7 0270 0100	DUMP1	FDB	I,CR,HEX,DOT,I,CLITER	
16EB 0AF7 167C				
16EF 0270 01EE				
16F3 10		FCB	16	
16F4 041D 0270		FDB	PLUS,I,XDO	
16F8 0261				
16FA 093D 0270	DUMP2	FDB	SPACE,I,CAT,TWO,DOTR,XLOOP	
16FE 06B3 077B				
1702 165D 0223				
1706 FFF4		FDB	DUMP2-*	
1708 0783 15A0		FDB	THREE,SPACES,I,CLITER	
170C 0270 01EE				
1710 10		FCB	16	
1711 041D 0270		FDB	PLUS,I,XDO	
1715 0261				

1717 0270 06B3	DUMP3	FDB	I,CAT,DUP,CLITER
171E 068A 01EE			
171F 20		FCB	\$20
1720 05A3 020B		FDB	LESS,ZERAN
1724 0007		FDB	DUMP31-*
1726 066B 01EE		FDB	DROP,CLITER
172A 5F		FCB	-
172B 00B3 0228	DUMP31	FDB	EMIT,XLOOP
172F FFE8		FDB	DUMP3-*
1731 01EE		FDB	CLITER
1733 10		FCB	16
1734 0239		FDB	XPLOOP
1736 FFB1		FDB	DUMP1-*
1738 0080		FDB	SEMIS
173A		WORDM	5,VLIS,T
1742 0073 01EE	VLIST	FDB	DOCOL,CLITER
1746 80		FCB	\$80
1747 0844 06BF		FDB	OUT,STORE,CONTXT,AT,AT
174B 0869 06A7			
174F 06A7			
1751 0844 06A7	VLIST1	FDB	OUT,AT,COLUMNS,AT,CLITER
1755 08CD 06A7			
1759 01EE			
175B 10		FCB	16
175C 0915 092D		FDB	SUB,GREAT,ZBRAN
1760 020B			
1762 000A		FDB	VLIST2-*
1764 0100 076B		FDB	CR,ZERO,OUT,STORE
1768 0844 06BF			
176C 068A 0F49	VLIST2	FDB	DUP,IDDOT,SPACE,SPACE,PFA,LFA,AT,DUP,ZEQU,QTERM
1770 093D 093D			
1774 09F9 09C7			
1778 06A7 068A			
177C 05FE 00F0			
1780 03FF 020B		FDB	OR,ZBRAN
1784 FFCD		FDB	VLIST1-*
1786 066B 0080		FDB	DROP,SEMIS
*			
*			
***** FILE FDISK.TXT			
*<<< DISK I/O WORDS >>> SYSTEM DEPENDENT			
*			
178A		WORDM	3,#D,R
1790 074A	NUMDR	FDB	DOCON
1792 0002		FDB	2 the number of disk drives
1794		WORDM	8,TRK/DIS,K tracks per disk
179F 074A	TRKDSK	FDB	DOCON
17A1 0023		FDB	35
17A3		WORDM	7,SEC/TR,K sectors per track == block = sector
17AD 074A	SECTRK	FDB	DOCON
17AF 000A		FDB	10
17B1		WORDM	5,B/BU,F

17B9 074A	BBUF	FDB	DOCOM
17BB 0100		FDB	256
17BD		WORDM	5,B/SC,R
17C5 0073 01E7	BSCR	FDB	DOCOL,LIT,1024,BBUF,SLASH,SEMIS
17C9 0400 17B9			
17CD 051D 0080			
17D1		WORDM	7,SCR>BL,K
17DB 0073 17C5	SCRBLK	FDB	DOCOL,BSCR,STAR,USEBLK,SLMOD,SECTRK,STAR
17DF 04D7 17FA			
17E3 050D 17AD			
17E7 04D7			
17E9 179F 04D7		FDB	TRKDSK,STAR,PLUS,SEMIS converts SCR# TO BLOCK #
17ED 041D 0080	*		ALLOWING FOR THE NON INTEGER # OF SCR PER DISK
17F1		WORDM	6,USEBL,K no of blocks per disk useable as SCReens
17FA 0073 17AD	USEBLK	FDB	DOCOL,SECTRK,TRKDSK,STAR,BSCR,SLASH,BSCR,STAR,SEMIS
17FE 179F 04D7			
1802 17C5 051D			
1806 17C5 04D7			
180A 0080			
180C		WORDM	3,DR,0
1812 0073 076B	DRZERO	FDB	DOCOL,ZERO,OFSET,STORE
1816 085B 06BF			
181A 0080		FDB	SEMIS
181C		WORDM	3,DR,1
1822 0073 0773	DRONE	FDB	DOCOL,ONE,DRIVE,SEMIS
1826 1842 0080			
182A -		WORDM	5,DRSI,M
1832 0073 1790	DRSIM	FDB	DOCOL,NUMDR,DRIVE,SEMIS
1836 1842 0080			
183A		WORDM	5,DRIV,E drive number is arg on stack
1842 0073 17AD	DRIVE	FDB	DOCOL,SECTRK,TRKDSK,STAR,STAR,OFSET,STORE,SEMIS
1846 179F 04D7			
184A 04D7 085B			
184E 06BF 0080	*		
		PAG	

*** The next 4 words are written to create a substitute for
* disc mass memory, located in DSMBGN to DSSEND in RAM

1852	WORDM	2,L,0	low address for simulated disk
1857	LO	FDB	DOCON
1859	3000	FDB	DSMBGN
185B	WORDM	2,H,I	high address for simulated disk
1860	074A	FDB	DOCON
1862	4000	FDB	DSSEND
1864	RW	WORDM	3,R/W
186A	0073	FDB	DCCOL,SWAP now have BLOCK NO ON STACK
186E	068A	FDB	DUP,ZLESS,ZEQU,ZBRAN cant have block < 0
1872	05FE	FDB	RWDE-*
1876	0014	FDB	SECTRK,TRKDSK,STAR,SLIOD now have block-2,dr-1
1878	17AD	FDB	DUP,NUMDR,GREAT,ZBRAN
187C	04D7	FDB	RWD1-* > RWD1 IF DRIVE < = #DR
1880	068A	FDB	CR,DOT,PDOTQ drive error
1884	092D	FDB	3
1888	001D	FCC	" Drive ?"
188A	C100	RWDE	FCC
188E	0C12	RWDEL	LIT,\$7FFF,PREV,AT,STORE,QUIT
1890	08	FDB	DUP,NUMDR,EQUAL,ZBRAM
1891	20	FDB	RWD2-* --> RWD2 IF < #DR
1895	76	FDB	DROP,TWON,TWON,DUP,ZLESS,ZBRAN USE SIM DUFF
1899	65	FDB	RWSL-* ONLY IF SCR>0
189D	20	FDB	CR,DOT,PDOTQ
18A1	01E7	FDB	FCC
18A1	06BF	FDB	" Range ?"
18A5	063A	FDB	BRAN
18A9	0921	FDB	RWDEL-*
18AD	0049	FDB	BZUF,STAR,LO,PLUS,DUP,HI,BZUF,SUE,GREAT,ZEQU,ZBRAN
18AF	066B	FDB	18B3 04AC 068A
18B7	0611	FDB	18BB 0015 18ED 0100 18C1 0C12
18C3	08	FDB	18C4 20 52 61 6E
18C8	67	FDB	18CC 01FF
18CE	FFCB	FDB	18DD 17B9 04D7
18D0	041D	FDB	18D4 1857 041D
18D8	068A	FDB	18DC 17E9 0915
18E0	092D	FDB	18E4 020B
18E6	FFD7	FDB	18E8 0679 020B
18EC	0004	FDB	18EE 0679
		FDB	RW44-* SWAP

18F0 17B9 036D	RW44	FDB	BEOF,CMOVE,SEMIS
18F4 0080			
18F6 0639 17AD	RWD2	FDB	TOR,SECTRK,SLMOD,SWAP,ONEP,SWAP,FRMR
18FA 050D 0679			
18FE 047F 0679			
1902 0647			
1904 1911 0080		FDB	DISKRW,SEMIS
1908		WORDM	6,DISKRW,W
1911 1913	DISKRW	FDB	*+2
1913 17 014D		LBSR	DSKRW0
1916 16 E75E		LBRA	NEXT
1919		WORDM	3,(_,)
191F 0073 0654	PDOS	FDB	DOCOL,R,COUNT,DUP,ONEP,FRMR,PLUS,TOR,GODOS,SEMIS
1923 0E4F 068A			
1927 047F 0647			
192B 041D 0639			
192F 1933 0080			
1933 1935	GODOS	FDB	*+2
1935 17 0128		LBSR	GODOS0
1938 16 E73C		LBRA	NEXT
	193B	NEXTNM	SET *
193B C1		FCB	\$C1 immediate
193C DF		FCB	\$80+-
193D 1919		FDB	LASTNM
	193B	LASTNM	SET NEXTNM
193F 0073 01EE	DOSQ	FDB	DOCOL,CLITER
1943 22		FCB	\$22 ascii quote
1944 0883 06A7		FDB	STATE,AT,ZBRAN
1948 020B			
194A 0014		FDB	DOS1-*
194C 0AAE 191F		FDB	COMPIL,PDOS,WORD,HERE,CAT,ONEP,ALLOT,BRAN
1950 0DED 08D8			
1954 06B3 047F			
1958 08E8 01FF			
195C 000A		FDB	DOS2-*
195E 0DED 08D8	DOS1	FDB	WORD,HERE,COUNT,GODOS
1962 0B4F 1933			
1966 0080	DOS2	FDB	SEMIS
1968		WORDM	3,DO,S
196E 1A5C	DOS	FDB	PDOSW
	2C80	FCBIN	EQU USREND-\$100-640
	2DC0	FCEOUT	EQU FCBIN+320
1970		WORDM	6,DISKI,N
1979 074A 2C80	DISKIN	FDB	DOCON,FCBIN
197D		WORDM	7,DISKOU,T
1987 074A 2DC0	DISKOUT	FDB	DOCON,FCEOUT
198B		WORDM	6,REWIN,D
1994 1996	RENDF0	FDB	*+2
1996 17 00D3		LBSR	REWNDF
1999 16 E6DE		LBRA	NEXT
199C		WORDM	6,DELETE,E
19A5 19A7	DELTFO	FDB	*+2

19A7 17 00C5	LBSR	DELETF
19AA 16 E6CA	LBRA	NEXT
19AD	WORDM	4,OPE,N
19B4 19B6	OPENFO	FDB *+2
19B6 17 00AD	LBSR	OPENF expects filenameaddr,ioccode,fcbadr on STACK
19E9 16 E6BB	LBRA	NEXT
19EC	WORDM	4,REA,D
19C3 0073 0773	READ	FDB DOCOL,ONE,DISKIN,OPENFO,DISKIN,LIT,XFINA
19C7 1979 19B4		
19CB 1979 01E7		
19CF 2028		
19D1 06BF 0080	FDB	STORE,SEMIS
19D5	WORDM	5,WRIT,E
19DD 0073 076E	WRITE	FDB DOCOL,ZERO,DISKOUT,OPENFO,DISKOUT,LIT,XFOUTA
19E1 1987 19B4		
19E5 1987 01E7		
19E9 202A		
19EB 06BF 0080	FDB	STORE,SEMIS
19EF	WORDM	5,CLOS,E
19F7 19F9	CLOSFO	FDB *+2
>19F9 17 006D	LBSR	CLOSEF expects fcb adr on stack
19FC 16 E678	LBRA	NEXT
19FF	WORDM	7,CLOSEI,N
1A09 0073 076E	CLOSIN	FDB DOCOL,ZERO,LIT,XFINA,STORE
1A0D 01E7 2028		
1A11 06BF		
1A13 1979 19F7	FDB	DISKIN,CLOSFO,SEMIS
1A17 0080		
1A19	WORDM	8,CLOSEOU,T
1A24 0073 076B	CLOSOT	FDB DOCOL,ZERO,LIT,XFOUTA,STORE
1A28 01E7 202A		
1A2C 06BF		
1A2E 1987 19F7	FDB	DISKOUT,CLOSFO,SEMIS
1A32 0080	*	
1A34	WORDM	4,NOO,P a noop
1A3B 0077	NOOP	FDB NEXT
	*	
1A3D FDOSBG	EQU	* CHECK TO SEE IF SPACE OK FOR FDOS
	*	
	*	* FOLLOWING ARE SYSTEM DEPENDENT MACHINE LANGUAGE ROUTINES
	PAG	

*** * * *

* TALBOT MICROSYSTEMS 68'FORTH

*

TTL (c)1980 TALBOT MICROSYSTEMS
STTL 68'FORTH I/O DRIVERS
OPT PAG,NOC,MAC,NOE

*

* FDOS IS A FILE CONTAINING THE ASSEMBLY LANGUAGE ROUTINES WHICH
* INTERFACE 68'FORTH WITH A DISK OPERATING SYSTEM
* THIS IS VERSION 1.1 (80.3.8)
*

* IT IS SUPPLIED FOR TSC FLEX 9.0

*

* THERE ARE ADDRESSES IN HERE WHICH REFER BACK INTO THE CODE
* 68'FORTH AND THESE MUST NOT BE CHANGED
* THERE ARE ENTRY POINTS AT WHICH 68'FORTH EXPECTS TO FIND
* VARIOUS ROUTINES, AND THESE ADDRESSES MUST NOT BE CHANGED
* THE STARTING POINT IS FBGNIO
* THE LAST BYTE OF THESE ROUTINES MUST NOT GO BEYOND \$1BEF
*

* IF NECESSARY TO USE MORE SPACE, YOUR MUST ALLOCATE IT SOMEWHERE
* UP ABOVE THE MEMORY SPACE USED FOR VIRTUAL MEMORY DISK BUFFERS
* STACKS, AND SIMULATED DISK.

*

*

* THE NEXT WORDS ARE SYSTEM-DEPENDENT I/O SUBROUTINES

*

*

* FBGNIO this is the address where these I/O routines are to start.

*

* FBYTSC the addr of # of bytes in a sector in the disk IO
in FLEX9.0 this is 256

*

* FFINA location for storing address of input FCB
* FFOUTA location for storing address of output FCB
* FACIA location of address of terminal ACIA status word
data byte is 1+

*

*<<<<<< FROM HERE TO >>>>> THE ADDRESSES CAN NOT BE CHANGED

*

1A50 FBGNIO SET \$1A50
1A50 ORG FBGNIO

*

17BB FBYTSC SET \$17BB

*

2028 FFINA SET \$2028
202A FFOUTA SET \$202A
2018 FACIA SET \$2018

*

*** * * *

*

* NOW JUMP VECTORS FOR FORTH - 3 BYTES EACH

*

>1A50 16 0048 PEMIT LBRA PPEMIT emit char in A to terminal
>1A53 16 0062 PKEY LBRA PPKEY get char from termnl - put in A, NO ECHO!
>1A56 16 0079 PQTER LBRA PPQTER query terminal to see if char typed -
* ret 0 if not, ret char if so - ESC is treated as a
* request to pause, another ESC will resume as if no
* key had been pressed.
1A59 16 016D PMON LBRA RESMON close any open files and return to MONITOR
1A5C 6E 9D 0027 PDOSW JMP [DOSWRM,PCR] return to DOS
1A60 16 0091 GODOSO LBRA GODOSI routine to set up DOS command call
1A63 16 00AD DSKRWO LBRA DSKRWI disk sector IO - args on U stack
* FORTH-BUFFER-ADDRESS -5
* READ/WRITE CODE - 1=READ, 0=WRITE -4
* SECTOR NUMBER -3
* TRACK NUMBER -2
* DRIVE NUMBER -1
1A66 16 0109 OPENF LERA OPENFI OPEN file - args on the U stack
* ADDRESS OF FIRST CHAR (COUNT FIELD) OF STRING WITH
* NAME OF FILE -3
* READ/WRITE FLAG 1=READ,0=WRITE -2
* ADDRESS OF FCB -1
*
1A69 16 0151 CLOSEF LBRA CLOSFI arg is on stack ADDRESS OF FCB -1
1A6C 16 0152 REWNDF LBRA REWNDI " " " " " " "
1A6F 16 0153 DELETEF LERA DELETI " " " " " " "
*
1A72 RMB 9 reserve space for 3 more vectors
*
* >>>>>>>>>> THE ABOVE CODE CAN NOT BE CHANGED
*
* <<<<<<<<< THE CODE BELOW MAY BE CHANGED, BUT THE LAST ADDRESS MAY
* NOT BE LARGER THAN 1BEF - 1BFO TO 2000 IS USED FOR DISK
* VIRTUAL MEMORY BUFFERS
*
* THIS VERSION IS FOR FLEX 9.0 WITH MF-68 DUAL DISK
* The following are variable depending upon the DOS system
* or the monitor
*
F802 NXTMON EQU \$F802 MONITOR LOC of addr to restart,
* i.e., JMP [NXTMON,PCR]
*
* NOT ALL OF THESE ARE ACTUALLY USED AT PRESENT. THE ONES WHICH ARE
* ARE MARKED WITH < IN COMMENTS AND SO MUST BE SET TO DOS
* LOCATION WHICH DOES EQUIVALENT FLEX FUNCTION OR YOU MUST
* CREATE YOUR OWN ROUTINE TO DO EQUIVALENT.
*
1A7B C840 DOSFCB FDB \$C840 <address of FLEX system FCB
1A7D C080 DOSIBF FDB \$C080 <beginning of input line buffer for FLEX
1A7F CC0B DOSSDN FDB \$CC0B address of system drive number
1A81 CC0C DOSWDN FDB \$CC0C <address of working drive number

1A83 CC14	DOSBPT	FDB	\$CC14	<address of DOS line buffer pointer
1A85 CC20	DOSDET	FDB	\$CC20	<address of FMS error type number
1A87 CD03	DOSWRM	FDB	\$CD03	<FLEX WARMS warm start entry
1A89 CD2D	DOSGFL	FDB	\$CD2D	<FLEX GETFIL get file specification
1A8B CD33	DOSEXT	FDB	\$CD33	<FLEX SETEXT set extension for file
1A8D CD3F	DOSRER	FDB	\$CD3F	<FLEX RPTERR reprt File Managemnt Sys error
1A8F CD4B	DOSCMD	FDB	\$CD4B	<FLEX call as subroutine
1A91 D403	DOSFCL	FDB	\$D403	<FLEX FMS CLOSE close all open files
1A93 D406	DOSFMS	FDB	\$D406	<FLEX FMS
	*			
1A95		RMB	6	reserve space for 3 more system parameters
1A9B 34 14	PPEMIT	PSHS	B,X	
1A9D 7D 202A		TST	FFOUTA	test to see if file output add set
1AA0 27 08		BEQ	PEMITO	if not, do terminal IO
1AA2 BE 202A		LDX	FFOUTA	get output file FCB address
1AA5 17 010E	FLAIO	LBSR	FMSCAL	call DOS FMS
1AA8 20 0B		BRA	PEMIT2	
1AAA BE 2018	PEMITO	LDX	FACIA	
1AAD E6 34	PEMIT1	LDB	,X	get status
1AAF C5 02		BITE	#2	check ready bit
1AB1 27 FA		BEQ	PEMIT1	
1AB3 A7 01		STA	1,X	send character in A
1AB5 35 14	PEMIT2	PULS	B,X	
1AB7 39		RTS		
1AE8 34 14	PPKEY	PSHS	B,X	
1ABA 7D 2028		TST	FFINA	test if input file address is set
1ABD 27 05		BEQ	PKEY0	if not, read from terminal
1ABF BE 2028		LDX	FFINA	get input address
1AC2 20 E1		BRA	FLAIO	go to file io routine
1AC4 BE 2018	PKEY0	LDX	FACIA	
1AC7 E6 84	PKEY2	LDB	,X	get status
1AC9 57		ASRB		
1ACA 24 FB		BCC	PKEY2	no incomming data yet
1ACC A6 01		LDA	1,X	
1ACE 84 7F		ANDA	#\$7F	strip parity
1ADO 20 E3		BRA	PEMIT2	
1AD2 34 10	PPQTER	PSHS	X	terminal query routine
1AD4 BE 2018		LDX	FACIA	
1AD7 A6 84		LDA	,X	look at status
1AD9 47		ASRA		
1ADA 25 03		BCS	PQTER2	if key has been pressed, get it and return in A register;
	*	CLRA		if not, return 0 - note cntl @ = NULL will
1ADC 4F		BRA	PQTER3	be regarded as no key
1ADD 20 12		PQTER2	LDA	puts character into A
1ADF A6 01			1,X	
1AE1 81 1B		CMPA	#\$1B	test if it was ESCAPE KEY
1AE3 26 0C		BNE	PQTER3	if not, return and just signal that key presse
1AE5 A6 84		PQTER3	LDA	look for another key
1AE7 47		ASRA		
1AE8 24 FB		BCC	PQTER2	loop until find one
1AEA A6 01		LDA	1,X	get it
1AEC 81 1B		CMPA	#\$1B	test to see if escape

1AEE 26	01		BNE	PQTER3	if not, then pass it on	
1AF0 4F			CLRA		if so, then treat as if no key pressed	
1AF1 35	10	PQTER3	PULS	X		
1AF3 39		PQTER4	RTS			
1AF4 37	16	GODOSI	PULU	D,X		
1AF6 34	60		PSHS	U,Y		
1AF8 FE	1A7D		LDU	DOSIBF		
1AFB EF	9C 85		STU	[DOSEBPT,PCR] init	LINE BUFFER POINTER	
1AFE 1F	02		TFR	D,Y	use Y as counter	
1B00 A6	80	GODOSI	LDA	,X+		
1B02 A7	C0		STA	,U+		
1B04 31	3F		LEAY	-1,Y		
1B06 26	F8		BNE	GODOSI		
1B08 86	0D		LDA	#\$0D		
1B0A A7	C4		STA	,U		
1B0C AD	9C 80		JSR	[DOSCMD,PCR]		
1B0F 35	60		PULS	U,Y		
1B11 39			RTS			
		DE0C	DRSEL	EQU	\$DE0C	
		DE00	DRREAD	EQU	\$DE00	
		DE03	DRWRIT	EQU	\$DE03	
		DE06	DRVERF	EQU	\$DE06	
1B12 00		NUMTRY	FCB	0	holds number of tries	
1B13 BE	1A7B	DSKRWI	LDX	DOSFCB	buffered rwcode sector track drive	
1B16 EC	C1		LDD	,U++	get drive	
1B18 E7	03		STB	3,X	drive byte of FCB	
1B1A BD	DE0C		JSR	DRSEL		
1B1D 86	0A		LDA	#10	number of tries	
1B1F B7	1B12		STA	NUMTRY		
1B22 A6	41	DSKRC1	LDA	1,U		
1B24 E6	43		LDB	3,U		
1B26 6D	45		TST	5,U	test rwcode	
1B28 27	10		BEQ	DSKRWW	0==WRITE	
1B2A AE	46		LDX	6,U	buff ad into X	
1B2C BD	DE00	DSKRWI	JSR	DRREAD		
1B2F 27	1C		BEQ	DSKRTS		
1B31 7A	1B12		DEC	NUMTRY		
1B34 26	EC		BNE	DSKRC1	try read again	
1B36 86	52		LDA	#`R		
1B38 20	16		BRA	DSKRWE		
1B3A AE	46	DSKRWW	LDX	6,U	buff ad into X	
1B3C BD	DE03	DSKRWL	JSR	DRWRIT		
1B3F BD	DE06		JSR	DRVERF		
1B42 27	09		BEQ	DSKRTS		
1B44 7A	1B12		DEC	NUMTRY		
1B47 26	D9		BNE	DSKRC1		
1B49 86	57		LDA	#`W		
1B4B 20	03		BRA	DSKRWE		
1B4D 33	43	DSKRTS	LEAU	S,U		
1B4F 39			RTS			
1B50 17	FF48	DSKRWE	LBSR	PPEMIT	type io type	
1B53 33	48		LEAU	8,U		

1B55 16	E4AB		LBRA	3	warm restart
1B58 AD	9D FF31	RWDSEO	JSR	[DOSRER,PCR]	report error
1B5C 6E	8D E4A3		JMP	3,PCR	warm start entry point
1B60 26	01	CHKERR	BNE	CHKERO	if any error consider what it is
1B62 39			RTS		otherwise return
1B63 E6	01	CHKERO	LDB	1,X	get error code
1B65 C1	08		CMPB	#\$8	is it EOF?
1B67 26	EF		BNE	RWDSEO	go report error and warm restart
1B69 7F	2028		CLR	FFINA	clear input file FCB address so that input
1B6C 7F	2029		CLR	FFINA+1	will be from terminal
1B6F 86	0D		LDA	#\$D	return a car ret and continue
1B71 39			RTS		
1B72 34	20	OPENFI	PSHS	Y	
1B74 10AE	44		LDY	4,U	get addr of count byte of string for
	*				name of file
1B77 31	21		LEAY	1,Y	move Y up to first character of name
1B79 AE	8D FF00		LDX	DOSIBF,PCR	address of DOS input line buffer
1B7D E6	3F		LDB	-1,Y	get number of characters in name of file
1B7F 27	07	OPNL1	BEQ	OPNL2	br down when out of characters
1B81 A6	A0		LDA	,Y+	get next char
1B83 A7	80		STA	,X+	store in next buff loc
1B85 5A			DEC8		decr ctr
1B86 20	F7		BRA	OPNL1	
1B88 86	0D	OPNL2	LDA	#\$D	carriage ret denotes end of name
1B8A A7	S4		STA	,X	
1B8C AE	8D FEED		LDX	DOSIBF,PCR	get buffer address again
1B90 AF	9D FEEF		STX	[DOSEPT,PCR]	set buffer ptr to pt to buffer beginning
1B94 AE	C4		LDX	0,U	get address of FCB to use for this file
1B96 A6	9D FEE7		LDA	[DOSWDN,PCR]	get DCS working file no to use as default
1B9A A7	03		STA	3,X	
1B9C AD	9D FEE9		JSR	[DOSGFL,PCR]	call DOS GETFIL rtn to parse file name
	*				and set up FCB
1BA0 86	01		LDA	#1	set default extinsion to TKT
1BA2 AD	9D FEE5		JSR	[DOSEXT,PCR]	sets extinsion to default if not given.
1BA6 A6	43		LDA	3,U	get READ (=1) or WRITE (=0) code from stack
1BA8 33	46		LEAU	6,U	drop all arguments from stack
1BAA 35	20		PULS	Y	
1BAC 26	06		BNE	FMSCL1	
1BAE 86	02		LDA	#2	0 = WRITE , IN FLEX, WRITE IS COMMAND 2
1BB0 20	02		BRA	FMSCL1	
1BB2 AE	C1	FMSCLL	LDX	,U++	get FCB address from stack and drop it
1BB4 A7	84	FMSCL1	STA	0,X	
1BB6 AD	9D FED9	FMSCLL	JSR	[DOSFMS,PCR]	
1BBA 8D	A4		BSR	CHKERR	
1BBC 39			RTS		
1BBD 86	04	CLOSFI	LDA	#\$04	FLEX CLOSE FILE
1BBF 20	F1		BRA	FMSCLL	
1BC1 86	05	REWNDI	LDA	#\$5	FLEX REWIND AN OPEN FOR READ FILE
1BC3 20	ED		BRA	FMSCLL	
1BC5 86	0C	DELETI	LDA	#\$OC	DELETE FILE FROM DISK (CLOSE FIRST)
1BC7 20	E9		BRA	FMSCLL	
1BC9 AD	9D FEC4	RESMON	JSR	[DOSFCL,PCR]	close all open files

(c)1980 TALBOT MICROSYSTEMS
68'FORTH I/O DRIVERS

4-20-80 TSC ASSEMBLER PAGE 52

```
1BCD 6E  9D DC31      JMP  [NXTMON,PCR] next monitor command processor
    1BD0  FDOSIN  EQU  *-1      this address FDOSIN must assemble to be <= 1BEF
    *                               END      PRGBGN
```

1 ERROR(S) DETECTED

SYMBOL TABLE:

ABORT	1120	ABORTC	1177	ABS	057C	ABS2	0588	ACIAI	0029
AGAIN	1526	ALLOT	08E8	AND	03ED	ARROW	13F8	AT	06A7
BACK	147B	BACKEC	0030	BACKSP	002F	BASE	088E	BBUF	17B9
ECOMP	0FDD	BDIGS	15BD	BEGIN	148D	BL	078C	BLANKS	0DB4
BLK	0831	BLOCK	12B0	BLOCK3	12CE	BLOCK4	12E3	BLOCK5	1300
BRAN	01FF	BSCR	17C5	BUFFER	1268	BUFFR2	1272	BUFFR3	1298
BUFSIZ	0100	BUILDS	06E4	CAT	06B3	CCOMM	0905	CEMIT	00B9
CENT	0142	CFA	09D6	CHKERO	1B63	CHKERR	1B60	CKEY	00D9
CLITER	01EE	CLOSEF	1A69	CLOSFO	19F7	CLOFSI	1BBD	CLOSIN	1A09
CLOSOT	1A24	CMOV2	0380	CMOV3	038A	CMOVE	036D	COLD	0140
COLD2	0147	COLD8	0168	COLDZ	0180	COLINT	002D	COLON	005F
COLUMNS	08CD	COMMA	08F4	COMPIL	0AAE	CON	0740	CONTXT	0869
COUNT	0E4F	CPUTYP	0006	CR	0100	CRI	010A	CR2	012E
CRE	0136	CREATE	0F9D	CREATE	0F7D	CSP	08AC	CSTORE	06CF
CURRENT	0877	DABS	0591	DABS2	059D	DDOT	166E	DDCTR	163A
DDUP	097C	DDUP2	0986	DEC	0E0C	DEFIN	10D2	DELETF	1A6F
DELETI	1EC5	DELINT	002B	DELTFO	19A5	DFIND	0EDE	DFIND2	0EFE
DIG	15FA	DIG2	1614	DIGIT	0295	DIGITO	02AB	DIGIT1	02B3
DIGIT2	02B8	DIGS	1622	DIGS2	1624	DISKIN	1979	DISKOU	1987
DISKRW	1911	DLINE	135E	DLITE2	1026	DLITER	1016	DMINUS	0461
DMINX	0477	DO	14C5	DOCOL	0073	DOCON	074A	DODOES	0700
DOES	06F4	DOS	196E	DOS1	195E	DOS2	1966	DOSBPT	1A83
DOSCMD	1A8F	DOSDET	1A85	DOSEXT	1A8B	DOSFCB	1A7B	DOSFCL	1A91
DOSFMS	1A93	DOSGFL	1A89	DCSIBF	1A7D	DOSQ	193F	DOSFER	1A8D
DOSSDN	1A7F	DCSWDN	1A81	DOSWRM	1A87	DOT	167C	DOTQ	0C2B
DOTQ1	0C4A	DOTQ2	0C52	DOTR	165D	DOUSER	07B5	DOVAR	0760
DOVOC	10EA	DP	0818	DPINIT	0025	DPL	0898	DPLUS	042B
DRIVE	1842	DRONE	1822	DROP	066B	DRREAD	DE00	DRSEL	DE0C
DRSIM	1832	DRVERF	DE06	DRWRIT	DE03	DRZERO	1812	DSETS2	05F2
DSETSN	05E8	DSKRC1	1B22	DSKRTS	1B4D	DSKRWO	1A63	DSKRW1	1B2C
DSKRWE	1B50	DSKRWI	1B13	DSKRWL	1B3C	DSKRWV	1B3A	DSMBGN	3000
DSMEND	4000	DTRAIL	0B90	DTRAL2	0B98	DTRAL3	0BE2	DTRAL4	0BE6
DUMP	16DD	DUMP1	16E7	DUMP2	16FA	DUMP3	1717	DUMP31	172B
DUP	068A	EDIGS	15CC	ELSE	156D	EMIT	00B3	ENCL2	0324
ENCL3	0334	ENCL4	033A	ENCL5	0348	ENCL6	0352	ENCL7	035A
ENCL7P	035D	ENCL8	035F	ENCLOS	0318	END	1518	ENDIF	149F
EQUAL	0921	ERAM	11C9	ERASE	0DA3	ERROR	0F18	ERROR2	0F26
EXEC	0091	EXPEC2	0CA5	EXPEC3	0CF9	EXPEC4	0D10	EXPEC5	0D12
EXPEC6	0D1E	EXPECT	0C9B	EXPECZ	0CCE	FACIA	2018	FBGNIO	1A50
FBYTSC	17B8	FCBIN	2C80	FCBOUT	2DC0	FDCSBC	1A3D	FDOSIN	1B00
FENCE	080F	FENCIN	0023	FFINA	2028	FFOUTA	202A	FILL	0D83
FINA	0041	FIRST	0798	FLAIO	1AA5	FLD	08A2	FLUSH	1310
FLUSH1	1325	FMSCAL	1BB6	FMSCLI	1BE4	FMSCLL	1BE2	FORGET	1431
FORTH	2058	FOUND	02FF	FOUTA	0C43	FROMIR	0647	GETX	0055
GO	117E	GODOS	1933	GODOS0	1A60	GODOS1	1B00	GODOS1	1AF4
GREAT	092D	HERE	08D8	HEX	0AF7	HI	1860	HLD	08BF
HOLD	0DC3	I	0270	IDDOT	0F49	IF	1556	IFCOLD	0138
IMMED	1082	IN	083A	INTER2	1036	INTER3	104E	INTER4	1052

INTER5	1058	INTER6	106C	INTER7	1070	INTERP	1034
IOSTAT	204E	J	027D	JSR	00A5	K	0236
KEY	00D3	LASTNM	1A34	LATEST	09E7	LEBAK	0AC4
LESS	05A3	LESSF	05B1	LESSST	05B4	LESSX	05E6
LIMIT	07A4	LINDEC	0032	LINDEL	0031	LIST	1697
LIT	01E7	LITER	0FFF9	LITER2	1009	LO	1857
LOOP	14D8	MAX	0963	MAX2	0971	MENEND	3000
MESS	1372	MESS3	13A2	MESS4	13BF	MIN	094B
MINUS	0448	MINUS2	0453	NOD	052D	MON	009D
MSLASH	04E4	MSMOD	0560	MSSTAR	04BB	MTBUF	1251
NBLK	0004	NEXT	0077	NEXT3	0079	NEXTNN	1A34
NOOP	1A3B	NULL	0D48	NULL1	0D72	NULL2	0D76
NUMB	0E88	NUMBER1	0EA5	NUMBER2	0ECA	NUMB3	0ED4
NUMTRY	1B12	NXTMON	F802	OFFSET	085D	ONE	0773
ONEP	047F	OPENF	1A66	OPENFO	19B4	OPENFI	1372
OPNL2	1B88	OR	03FF	OUT	0344	OVER	065D
PAO	2002	PABORT	0F0A	PAD	0DD8	PAREN	10E2
PBUF2	1218	PCHR	2006	PCMOVE	0374	PDOS	2000
PDSW7	1A5C	PDOTQ	0C12	PEMIT	1A50	PEMIT0	1AAA
PEMIT2	1A55	PFA	09F9	PEIND	02C8	PEIND0	02CC
PFIND2	02E1	PFIND3	02ED	PFIND4	02F9	PFIND8	02F5
PKEY	1A53	PKEY0	1AC4	PKEY2	1AC7	PLINE	133C
PLUS	041D	PION	1A59	PNUMB	0E3D	PNUMB2	0E3F
PNUMB4	0E7B	PORIG	07C7	PEMIT	1A9B	PPKEY	1AB8
PQTER	1A56	POTER2	1ADF	POTER3	1AF1	POTER4	1AF3
PQUOTE	0BF9	FREV	1EB	PRGBGN	0000	PSCODE	0B21
PSTORE	0696	PULLDX	004F	PUSHD	0057	PUTD	03F5
QCSP	0A79	QERR	0A20	QERR2	0A2E	QERR3	0A30
QLOAD	0A95	QPAIRS	0A67	QSTAC2	0C6F	QSTAC3	0C7E
QTERN	00F0	QUERY	0D30	QUEST	1688	QUIT	10F2
QUIT3	1114	QUOTE	0BC0	QUOTE1	0EDF	QUOTE2	0BF1
RAM	1190	RBREAK	0AD2	READ	19C3	REID	2089
RESN0N	1BC9	RENDFC	1994	RENDFD	1A6C	REMNDI	1BC1
RINIT	003F	RNUM	08E5	ROT	0C66	RPSTOR	01D8
RW	186A	RW4	13E8	RW44	18F0	RWDI	18A5
RWDE	186A	RWDE1	1399	RWDEO	1B58	RWRE	18BD
RZERO	07DF	SCR	034E	SCRELK	17DB	SCSP	0A0D
SEMI	0727	SEMIC	0B37	SEMS	0080	SETSN	05D6
SIGN	1E5	SIGN2	15F4	SINIT	003B	SLASH	051D
SMUDGE	0AE6	SPACE	093D	SPACE2	15B0	SPACE3	1536
SPAT	01B5	SPSTOR	01CA	SSLASH	054E	SSLD	053D
STATE	0C83	STOD	05C1	STOD2	05CC	STORE	06EF
SUB	0915	SNAP	0679	SZERO	07D6	TASK	2085
TREN	14BA	THREE	0783	TIB	07E9	TIBINT	003D
TOGGLE	0715	TOR	0639	TRAV	0993	TRAV2	0997
TWO	077B	TWOM	04AC	TWOF	048E	TYPE	0B62
TYPE3	0B80	TYPE4	0B82	UNTIL	1504	WORDG	200C
UPDATE	122B	UPINIT	0021	USE	11E0	USEBLK	17EA
USLASH	03B9	USLL1	03CA	USLL2	03DA	USBEGN	2000
USTAR	0394	USTAR2	03A3	USTAR3	03AF	USTARS	03B3
VAR	075A	VERSON	0008	VIRBGN	1EFO	VIREND	2000
VLIST1	1751	VLIST2	176C	VOCAB	109A	VOCINT	0027

WARM 0192	WARM2 019A	WARN 0803	WENT 0194	WHILE 158F
WIDINT 0045	WIDTH 07F5	WORD 0DED	WORD2 0E01	WORD3 0E05
WRITE 19DD	WRMINT 0049	XACIA 2018	XBASE 2042	XBKSP 201E
XBKSPE 201F	XBLK 2032	XCOLUMN 201C	XCONT 203C	XCSP 2048
XCURR 203E	XDELAY 201A	XDO 0261	XDP 2014	KDPL 2044
KDSMBG 0037	XDSMED 0039	XFENCE 2012	XFINA 2028	XFLD 2046
XFOUTA 202A	XHLD 204C	XIN 2034	XLINDE 2021	XLINDL 2020
XLOOP 0228	XMSGBS 202E	XOFSET 203A	XOR 040F	XOUT 2036
XPLOF 024E	XPLONO 0256	XPLOOP 0239	XPLOP2 023D	XPREV 004D
XRNUM 204A	XRZERO 2026	XSCR 2038	XSPZER 2022	XSTATE 2040
XTIB 2024	XUSE 004B	XVIREG 0033	XVIRED 0035	XVOCL 2016
XWARN 2030	XWIDTH 202C	ZBNO 021A	ZERAN 020B	ZYES 0211
ZEQU 05FE	ZEQU2 0607	ZERO 076B	ZLESS 0611	ZLESS2 061F