```
P                          --- addr              DICTIONARY
  A user variable, the dictionary pointer, which contains the
  address of the next free memory above the dictionary. The
  value may be read by HERE and altered by ALLOT.

PL                         --- addr              CONVERSION
  A user variable containing the number of digits to the right
  of the decimal on double integer input. It may also be used
  to hold output column location of a decimal point, in user
  generated formating. The default value on single number
  input is -1.

R0 DR1 R2                  ---                   DISK
  Command to select disk drives by presetting OFFSET. The
  contents of OFFSET is added to the block number in BLOCK to
  allow for this selection. OFFSET is suppressed for error
  text to that it may always originate from drive 0.

RIVE                   n ---                     DISK
  Adjusts OFFSET so that the drive number on the stack becomes
  the first drive in the system.

ROP                    n ---                     STACK
  Drop the top number from the stack.

UP                     n --- n n                 STACK
  Duplicate the value on the stack.

COUNT                      --- addr              ERROR
  A user variable which contains an error count. This is used
  to prevent error recursion.

LSE              addr1 n1 --- addr2 n2 (comp)    STRUCTURE
  Occurs within a colon-definition in the form:
                IF  ...  ELSE  ...  ENDIF
  At run-time, ELSE executes after the true part following
  IF. ELSE forces execution to skip over the following false
  part and resume execution after ENDIF. It has no stack
  effect.
  At compile-time, ELSE emplaces BRANCH reserving a branch
  offset and leaves the address addr2 and n2 for error
  testing. ELSE also resolves the pending forward branch from
  IF by calculating the offset from addr1 to HERE and storing
  it at addr1.
        !
MIT                    ch ---                    PRINT
  Transmit ascii character ch to the selected output device.
  OUT is incremented for each character output.

MIT8                   ch ---                    PRINT
  Transmit an 8-bit character to the selected output device.
  OUT is incremented for each character output.

MPTY-BUFFERS               ---                   DISK
  Mark all block-buffers as empty, not necessarily affecting
  the contents. Updated blocks are not written to disk.
  This is also an initialization procedure before first use of
  the disk.

NCLOSE           addr1 ch --- addr1 n1 n2 n3     MEMORY
  The text scanning primitive used by WORD. From the text
  address addr1 and an ascii delimiting character ch, is
  determined the byte offset to the first non-delimiter
  character n1, the offset to the delimiter after the text n2,
  and the offset to the first character not included. This
  procedure will not process past an ascii 'null', treating it
  as an unconditional delimiter.

ND                         f ---                 STRUCTURE
  This is an 'alias' or duplicate definition for UNTIL.

NDCASE                     ---                   STRUCTURE
  Terminates the CASE construct.

NDIF             addr n --- (compile)            STRUCTURE
  Occurs in a colon-definition in the form:
                IF  ...  ENDIF
                IF  ...  ELSE  ...  ENDIF
  At run-time, ENDIF serves only as the destination of a
  forward branch from IF or ELSE. It marks the conclusion of
  the conditional structure. THEN is another name for ENDIF.
  Both names are supported in fig-FORTH. See also IF and ELSE.
  At compile-time, ENDIF computes the forward branch offset
  from addr to HERE and stores it at addr. n is used for
  error tests.

NDOF                       ---                   STRUCTURE
  Terminates the OF construct within the CASE construct.

RASE                   addr n ---                MEMORY
  Clear a region of memory to zero from addr over n bytes.

RROR                   n1 --- n2 n3              ERROR
  Execute error notification and restart of system. WARNING
  is first examined. If 1, the text of line n1, relative to
  screen 4 of drive 0 is printed. This line number may be
  positive or negative, and beyond just screen 4. If
  WARNING=0, n1 is just printed as a message number (non-disk
  installation). If WARNING is -1, the definition (ABORT) is
  executed, which executes the system ABORT. The user may
  cautiously modify this execution by altering (ABORT).
  fig-FORTH saves the contents of IN (n2) and BLK (n3) to
  assist in determining the location of the error. Final
  action is execution of QUIT.

XECUTE                 cfa ---                   INTERPRET
  Execute the definition whose code field address is on the
  stack. The code field address is also called the
  compilation address.

XPECT              addr cnt ---                  MEMORY
  Transfer characters from the terminal to addr, until a
  "ENTER" or the count of characters has been received. One
  or more nulls are added at the end of the text.

NCE                        --- addr              DICTIONARY
  A user variable containing an address below which FORGETing
  is trapped. To FORGET below this point the user must alter
  the contents of FENCE.

LL                 addr cnt b ---                MEMORY
  Fill memory beginning at addr with the specified number
  (cnt) of bytes b.

RST                        --- addr              DISK
  A constant that leaves the address of the first (lowest)
  block buffer.

RST@                       --- addr              DISK
  A user variable which contains the first byte of the disk
  buffer area.

FLD                        --- addr              PRINT
  A user variable for control of number output field width.
  Presently unused in fig-FORTH and TI FORTH.

FLUSH                      ---                   EDITOR
  Rewrites to the disk all disk buffers that have
  been updated

FORGET                     ---                   DICTIONARY
  Executed in the form:
                FORGET cccc
  Deletes definition named cccc from the dictionary with all
  entries physically following it.

FORTH                      ---                   DICTIONARY
  The name of the primary vocabulary. Execution makes FORTH
  the CONTEXT vocabulary. Until additional user vocabularies
  are defined, new user definitions become a part of FORTH.
  FORTH is immediate, so it will execute during the creation
  of a colon-definition to select this vocabulary at compile
  time.

FORTH-LINK                 --- addr              DICTIONARY
  A user variable used for vocabulary linkage.

GOTOXY                 c r ---                   UOP
  Places the cursor at the designated column and row
  position. NOTE: Rows and columns are numbered from 0.

HERE                       --- addr              DICTIONARY
  Leave the address of the next available dictionary
  location.

HEX                        ---                   CONVERSION
  Set the numeric conversion base to sixteen (hexadecimal).

HLD                        --- addr              CONVERSION
  A user variable that holds the address of the latest
  character of text during numeric output conversion.

HOLD                       ch ---                CONVERSION
  Used between <# and #> to insert an ascii character into a
  pictured numeric output string.  e.g. 2E HOLD will place a
  decimal point.

I                          --- n                 STRUCTURE
  Used within a DO-LOOP to copy the loop index to the stack.
  Other use is implementation dependent. See R .

ID.                        nfa ---               DICTIONARY
  Print a definition's name from its name field address.

IF                         f --- (run-time)       STRUCTURE
                           --- addr n (compile)
  Occurs in a colon-definition in the form:
                IF  (tp)  ...  ENDIF
                IF  (tp)  ...  ELSE  (fp)  ...  ENDIF
  At run-time, IF selects execution based on a boolean flag.
  If f is true (non-zero), execution continues ahead thru the
  true part. If f is false (zero), execution skips to just
  after ELSE to execute the false part. After each part,
  execution resumes after ENDIF.  ELSE and its false part are
  optional; if missing, false execution skips to just after
  ENDIF.
  At compile time, IF compiles 0BRANCH and reserves space for
  an offset at addr.  addr and n are used later for resolution
  of the offset and error checking.

IMMEDIATE                  ---                   INTERPRET
  Mark the most recently made definition so that when
  encountered at compile time, it will be executed rather than
  being compiled.  i.e. the precedence bit in its header is
  set. This method allows definitions to handle unusual
  compiling situations, rather than build them into the
  fundamental compiler. The user may force compilation of an
  immediate definition by preceeding it with [COMPILE].

IN                         --- addr              KEYBOARD
  A user variable containing the byte offset within the
  current input text buffer (terminal or disk) from which the
  next text will be accepted. WORD uses and moves the value
  of IN.

INTERPRET                  ---                   INTERPRET
  The outer text interpreter which sequentially executes or
  compiles text from the input stream (terminal or disk)
  depending on STATE. If the word name cannot be found after
  a search of CONTEXT and then CURRENT it is converted into a
  number according to the current base. That also failing, an
  error message echoing the name with a "?" will be given.
  Text input will be taken according with the convention for
  WORD. If a decimal point is found as part of a number, a
  double number value will be left. The decimal point has no
  other purpose than to force this action. See NUMBER.

INTLNK                     --- addr              ISR
  A user variable which is a pointer to the Interrupt Service
  Linkage.

ISR                        --- addr              ISR
  A user variable that initially contains the address of the
  interrupt service linkage code to install an Interrupt
  Service Routine.  The user must modify ISR to contain the
  CFA of the routine to be executed each 1/60 second. Next,
  the contents of HEX 83C4 must be modified to point to this
  address. Note, the interrupt service linkage code address
  is also available in INTLNK.

J                          --- n                 STRUCTURE
  Copies the loop index of the second innermost loop to the
  stack.

KEY                        --- ch                KEYBOARD
  Leave the ascii value of the next terminal key struck.

KEY8                       --- ch                KEYBOARD
  Leave the 8-bit value of the next terminal key struck.

L/SCR                      --- n                 EDITOR
  Returns on the stack the number of lines per SCREEN.

LATEST                     --- nfa               DICTIONARY
  Leave the name field address of the topmost word in the
  CURRENT vocabulary.

LEAVE                      ---                   STRUCTURE
  Force termination of a DO-LOOP at the next opportunity by
  setting the loop limit equal to the current value of the
  index. The index itself remains unchanged, and execution
  proceeds normally until LOOP or +LOOP is encountered.
```

LFA pfa --- lfa DICTIONARY
Convert the parameter field address of a dictionary definition to its link field address.

LIMIT --- addr DISK
A constant which leaves the address just above the highest memory available for a disk buffer.

LIMIT$ --- addr DISK
A user variable which contains the address just above the highest memory available for a disk buffer.

LIST scr# --- PRINT
Lists the specified SCREEN to the output device. See PAUSE.

LIT --- n INTERPRET
Within a colon-definition, LIT is automatically compiled before each 16 bit literal number encountered in input text. Later execution of LIT causes the contents of the next dictionary address to be pushed on the stack.

LITERAL n --- (compiling) INTERPRET
If compiling, then compile the stack value n as a 16 bit literal. This will execute during a colon-definition. The intended use is:
: xxx [calculate] LITERAL ;
Compilation is suspended for the compile-time calculation of a value. Compilation is resumed and LITERAL compiles this value.

LOAD n --- INTERPRET
Begin interpretation of SCREEN n. Loading will terminate at the end of the SCREEN or at ;S. See ;S and --).

LOOP addr n --- (compiling) STRUCTURE
Occurs in a colon-definition in the form:
DO ... LOOP
At run-time, LOOP selectively controls branching back to the corresponding DO based on the loop index and limit. The loop index is incremented by one and compared to the limit. The branch back to DO occurs until the index equals or exceeds the limit; at that time, the parameters are discarded and execution continues ahead.
At compile-time, LOOP compiles (LOOP) and uses addr to calculate an offset to DO. n is used for error testing.

M* n1 n2 --- d ARITHMETIC
A mixed magnitude math operation which leaves the double number signed product of two signed numbers.

M/ d n1 --- n2 n3 ARITHMETIC
A mixed magnitude math operator which leaves the signed remainder n2 and signed quotient n3, from a double number dividend and divisor, n1. The remainder takes its sign from the dividend.

M/MOD ud1 u2 --- u3 ud4 ARITHMETIC
An unsigned mixed magnitude math operation which leaves a double quotient ud4 and remainder u3, from a double dividend ud1 and a single divisor u2.

MAX n1 n2 --- n3 ARITHMETIC
Leave the greater of the two numbers.

MESSAGE n --- PRINT
Print on the selected output device the text of line n relative to screen 4 of drive 0. n may be positive or negative. MESSAGE may be used to print incidental text such as report headers. If WARNING is zero, the message will simply be printed as a number (disk un-available).

MIN n1 n2 --- n3 ARITHMETIC
Leave the smaller of the two numbers.

MINUS n1 --- n2 ARITHMETIC
Leave the two's complement of a number.

MOD n1 n2 --- mod ARITHMETIC
Leave the remainder of n1/n2, with the same sign as n1.

MOVE addr1 addr2 n --- MEMORY
Move the contents of n memory cells (16 bit contents) beginning at addr1 into n cells beginning at addr2. The contents of addr1 is moved first.

MYSELF --- INTERPRET
Used in a colon definition. Places the CFA of a routine into itself. This permits recursion.

NFA pfa --- nfa DICTIONARY
Convert the parameter field address of a definition to its name field address.

NOP --- INTERPRET
A do nothing instruction. NOP is useful for patching as in assembly code.

NUMBER addr --- d CONVERSION
Convert a character string left at addr with a preceeding count, to a signed double number, using the current numeric base. If a decimal point is encountered in the text, its position will be given in DPL, but no other effect occurs. If numeric conversion is not possible, an error message will be given.

OF n --- STRUCTURE
Initiates the OF ... ENDOF construct inside of the CASE construct. n is compared to the value which was on top of the stack when CASE was executed. If the numbers are identical, the words between OF and ENDOF will be executed.

OFFSET --- addr DISK
A user variable which may contain a block offset to disk drives. The contents of OFFSET is added to the stack number by BLOCK. Messages issued by MESSAGE are independent of OFFSET. See BLOCK, DR0, MESSAGE.

OR n1 n2 --- n3 LOGICAL
Leave the bit-wise logical OR of two 16 bit values.

OUT --- addr PRINT
A user defined variable that contains a value incremented by EMIT and EMIT8. The user may alter and examine OUT to control display formating.

OVER n1 n2 --- n1 n2 n1 STACK
Copy the second stack value, placing it as the new top.

PAB0 --- addr UDP
A user variable which points to a region in UDP RAM which has been set aside for creating PABs.

PAD --- addr DICTIONARY
Leave the address of the text output buffer, which is a fixed offset above HERE.

PAUSE --- f PRINT
The words LIST, INDEX, DUMP and VLIST all call the word PAUSE. Pause allows the user to temporarily halt the output by pressing any key. Pressing another key will allow continuation. To exit one of these routines prematurely, press BREAK.

PFA nfa --- pfa DICTIONARY
Convert the name field address of a compiled definition to its parameter field address.

PREV --- addr DISK
A variable containing the address of the disk buffer most recently referenced. The UPDATE command marks this buffer to be later written to disk.

QUERY --- KEYBOARD
Input 80 characters of text (or until a "enter") from the operator's terminal. Text is positioned at the address contained in TIB with IN set to zero.

QUIT --- INTERPRET
Clear the return stack, stop compilation, and return control to the operator's terminal. No message is given.

R --- n STACK
Copy the top of the return stack to the parameter stack.

R# --- addr PRINT
A user variable which may contain the location of an editing cursor, or other file related function.

R->BASE --- STACK
Restore the current base from the return stack. See BASE->R.

R/W addr n1 f --- DISK
The fig-FORTH standard disk read-write linkage. addr specifies the source or destination block buffer, n1 is the sequential number of the referenced block, and f is a flag for f=0 write and f=1 read. R/W determines the location on mass storage, performs the read-write and performs error checking.

R0 --- addr STACK
A user variable containing the initial location of the return stack. Pronounced "R zero". See RP!.

R> --- n STACK
Remove the top value from the return stack and leave it on the parameter stack. See >R and R .

RDISK addr n1 n2 --- n3 DISK
The primitive routine that performs disk reads. addr is the address where the block is to be written in CPU RAM. n1 is the block number, n2 is the number of bytes per block, and n3 is the returned error code.

REPEAT addr n --- (compiling) STRUCTURE
Used within a colon-definition in the form:
BEGIN ... WHILE ... REPEAT
At run-time, REPEAT forces an unconditional branch back to just after the corresponding BEGIN.
At compile-time, REPEAT compiles BRANCH and the offset from HERE to addr. n is used for error testing.

ROT n1 n2 n3 --- n2 n3 n1 STACK
Rotate the top three values on the stack, bringing the third to the top.

RP! --- STACK
A procedure to initialize the return stack pointer from user variable R0.

S->D n --- d CONVERSION
Sign extend a single number to form a double number.

S->F n --- f CONVERSION
Converts a single precision number on the stack to a floating point number.

S->FAC n --- CONVERSION
Takes a single precision number from the stack, converts it to floating point, and leaves it in FAC.

S0 --- addr STACK
Pronounced "S zero". See SP!.

SCR --- addr EDITOR
A user variable containing the screen number most recently referenced by LIST or EDIT.

SCRN_END --- addr UDP
A user variable containing the address of the byte immediately following the last byte of the screen image table to be used as the logical screen.

SCRN_START --- addr UDP
A user variable containing the address of the first byte of the screen image table to be used as the logical screen.

SCRN_WIDTH --- addr UDP
A user variable which contains the number of characters which will fit accross the screen. (32 or 40) Used by the screen scroller.

SIGN n d --- d CONVERSION
Stores an ascii "-" sign at the current location in a converted numeric output string in the text output buffer when n is negative. n is discarded, but double number d is maintained. Must be used between <# and #> .

SLA n1 cnt --- n2 ARITHMETIC
Arithmetically shifts the number on the stack cnt bits to the left, leaving the result on the stack. cnt will be modulo 16, except when cnt=0, causing 16 bits to be shifted. To create a word which permits shifts when cnt could be zero, use the following definition: : SLA@ -DUP IF SLA ENDIF ;

SMUDGE --- DICTIONARY
Used during word definition to toggle the "smudge bit" in a definition's name field. This prevents an uncompleted definition from being found during dictionary searches, until compiling is completed without error.

SP! --- STACK
A procedure to initialize the stack pointer from S0.

CPU MEMORY MAP

FROM (HEX)	TO (HEX)	SIZE (HEX)	USAGE
0000	1FFF	2000	CONSOLE ROM
2000	3FFF	2000	LOW MEMORY EXPANSION -Loader, Program, REF/DEF Table
4000	5FFF	2000	PERIPHERAL ROMs FOR DSRs
6000	7FFF	2000	UNAVAILABLE - ROM IN COMMAND MODULES
8000	9FFF	2000	MEMORY MAPPED DEVICES FOR VDP, GROM, SOUND, SPEECH CPU RAM AT 8300-83FF
A000	FFFF	6000	HIGH MEMORY EXPANSION

MEMORY EXPANSION

	FROM (HEX)	TO (HEX)	SIZE (HEX)	USAGE
LOW MEMORY	2000	200F	0010	XML VECTORS
	2010	3423	1414	DISK BUFFERS
	3424	397F	055C	99/4A SUPPORT FOR FORTH
	3980	39FF	0080	USER VARIABLE AREA
	3A00	3CD9	02DA	ASSEMBLER SUPPORT
	3CDA	3FFF	0326	RETURN STACK
HIGH MEMORY	A000	BC7F	1C80	RESIDENT FORTH VOCABULARY
	BC80	FF9F	4320	USER DICTIONARY SPACE STARTS AT BC80, MOVES TOWARD HIGH MEMORY
				PARAMETER STACK STARTS AT FF9F, MOVES TOWARD LOW MEMORY
	FFA0	FFF1	0052	TERMINAL INPUT BUFFER

USER VARIABLES

Name	Offset (HEX)	Initial Value	Description
UCON$	06		Base of User Var Initial value table
0	08		Base of Stack
R0	0A		Base of Return Stack
U0	0C		Base of User Variables
TIB	0E		Terminal Input Buffer addr
WIDTH	10	31	Name length in dictionary
DP	12		Dictionary Pointer
SYS$	14		Addr of System Support
CURPOS	16		Cursor location in VDP RAM
INTLNK	18		Pointer to Interrupt Service Linkage
WARNING	1A	1	Message Control
C/L$	1C	64	Characters per Line
FIRST$	1E		Beginning of Dist Buffers
LIMIT$	20		End of Disk Buffers
B/BUF$	22	1024	Bytes per Buffer
B/SCR$	24	1	Blocks per Screen
DISK_LO	26	1	Low end Dist Fence
DISK_HI	28	90	High end Dist Fence
DISK_SIZE	2A	90	Logical Disk Size in Screens
DISK_BUF	2C	>1000	VDP location of 1K Buffer
PABS	2E	>460	VDP location for PABs
SCRN_WIDTH	30	40	Screen Width in Characters
SCRN_START	32	0	Screen Image Start in VDP
SCRN_END	34	960	Screen Image End in VDP
ISR	36		Interrupt Service Pointer
ALTIN	38	0	Alternate Input Pointer
ALTOUT	3A	0	Alternate Output Pointer
FENCE	3C		Dictionary Fence
BLK	3E		Block being interpreted
IN	40		Byte offset in text buffer
OUT	42		Incremented by EMIT
SCR	44		Last Screen referenced
OFFSET	46		Block offset to disks
CONTEXT	48		Pointer to Context Vocabulary
CURRENT	4A		Pointer to Current Vocabulary
STATE	4C		Compilation State
BASE	4E		Number Base for Conversions
DPL	50		Decimal Point Location
FLD	52		Field Width (unused)
CSP	54		Stack Pointer for error checking
R#	56		Editing Cursor location
HLD	58		Holds addr during numeric conversion
USE	5A		Next Block Buffer to Use
PREV	5C		Most recently accessed disk buffer
	5E		Don't use
	60		Don't use
FORTH_LINK	62		FORTH Vocabulary base
ECOUNT	64		Error control
VOC-LINK	66		Vocabulary Linkage
	68		Available for new USER Variables
	70		Available for new USER Variables
	72		Available for new USER Variables
	74		Available for new USER Variables
	76		Available for new USER Variables
	78		Available for new USER Variables
	7A		Available for new USER Variables
	7C		Available for new USER Variables
	7E		Available for new USER Variables

ASCII CODE TABLE

ASCII CODE HEX	ASCII CODE DECIMAL	CHARACTER	ASCII CODE HEX	ASCII CODE DECIMAL	CHARACTER
00	0	NUL c-,	40	64	@ f-J
01	1	SOH c-A f-7	41	65	A f-K
02	2	STX c-B f-4	42	66	B f-L
03	3	ETX c-C f-1	43	67	C f-M
04	4	EOT c-D f-2	44	68	D f-N
05	5	ENQ c-E f-=	45	69	E
06	6	ACK c-F f-8	46	70	F f-Y
07	7	BEL c-G f-3	47	71	G
08	8	BS c-H f-6	48	72	H
09	9	HT c-I f-0	49	73	I
0A	10	LF c-J f-X	4A	74	J
0B	11	VT c-K f-E	4B	75	K
0C	12	FF c-L f-5	4C	76	L
0D	13	CR c-M	4D	77	M
0E	14	SO c-N f-5	4E	78	N
0F	15	SI c-O f-9	4F	79	O
10	16	DLE c-P	50	80	P
11	17	DC1 c-Q	51	81	Q
12	18	DC2 c-R	52	82	R
13	19	DC3 c-S	53	83	S
14	20	DC4 c-T	54	84	T
15	21	NAK c-U	55	85	U
16	22	SYN c-U	56	86	U
17	23	ETB c-W	57	87	W
18	24	CAN c-X	58	88	X
19	25	EM c-Y	59	89	Y
1A	26	SUB c-Z	5A	90	Z
1B	27	ESC c-.	5B	91	[
1C	28	FS c-;	5C	92	\
1D	29	GS c-=	5D	93	]
1E	30	RS c-8	5E	94	^
1F	31	US c-9	5F	95	_
20	32	SPACE	60	96	`
21	33	!	61	97	a
22	34	"	62	98	b
23	35	#	63	99	c
24	36	$	64	100	d
25	37	%	65	101	e
26	38	&	66	102	f
27	39	'	67	103	g
28	40	(	68	104	h
29	41	)	69	105	i
2A	42	*	6A	106	j
2B	43	+	6B	107	k
2C	44	,	6C	108	l
2D	45	-	6D	109	m
2E	46	.	6E	110	n
2F	47	/	6F	111	o
30	48	0	70	112	p
31	49	1	71	113	q
32	50	2	72	114	r
33	51	3	73	115	s
34	52	4	74	116	t
35	53	5	75	117	u
36	54	6	76	118	v
37	55	7	77	119	w
38	56	8	78	120	x
39	57	9	79	121	y
3A	58	:	7A	122	z
3B	59	;	7B	123	ä
3C	60	<	7C	124	ö
3D	61	=	7D	125	å
3E	62	>	7E	126	ü
3F	63	?	7F	127	DEL

EXPLANATION OF ABBREVIATIONS

ABBREVIATIONS	MEANING
addr, addr1, ...	memory address
b	byte
c	column position
cccc	string representation
cfa	code field address
ch	ascii character code
cnt	count (length)
d, d1, d2, ...	double precision number (32 bits)
dc, dc1, dc2, ...	dot column position
dr, dr1, dr2, ...	dot row position
dst	refers to DSK1, DSK2, or DSK3
f	boolean flag
ff	boolean false flag
f1, f11, f12, ...	floating point number
lfa	link field address
mod	module
n, n1, n2, ...	single precision signed number (16 bits)
nfa	name field address
nnnn	string representation
pfa	parameter field address
r	row position
rem	remainder
scr#	screen number
spr#	sprite number
tf	boolean true flag
tol	tolerance limit
u	unsigned single precision number (16 bits)
ud	unsigned double precision number (32 bits)
vaddr	VDP address

TI FORTH
QUICK REFERENCE CARD
Resident Words

: Par Mario Jonathan Stephanie & Liz Beaulieu (1986) ; ok

! n addr --- MEMORY
Store 16 bits of n at address. Pronounced "STORE".

!CSP --- STACK
Save the stack position in CSP. Used as part of the compiler security.

d1 --- d2 CONVERSION
Generate from a double number d1, the next ASCII character which is placed in an output string. Result d2 is the quotient after division by BASE, and is maintained for further processing. Used between (# and #). See #S.

#) d --- addr cnt CONVERSION
Terminates numeric output conversion by droping d, leaving the text address and character count suitable for TYPE.

#S d1 --- d2 CONVERSION
Generates ASCII text in the text output buffer, by the use of #, until a zero double number d2 results. Used between (# and #).

' --- pfa DICTIONARY
Used in the form:
' nnnn
Leaves the parameter field address of dictionary word nnnn. As a compiler directive, executes in a colon definition to compile the address of a literal. If the word is not found after a search of CONTEXT and CURRENT, an appropriate error message is given. Pronounced "TICK".

(--- COMMENT
Used in the form:
(cccc)
Ignore a comment that will be delimited by a right parenthesis on the same screen. May occur during execution or in a colon definition. A blank after the leading parenthesis is required.

(+LOOP) n --- STRUCTURE
The run-time procedure compiled by +LOOP, which increments the loop index by n and tests for loop completion. See +LOOP.

(.") --- PRINT
The run-time procedure, compiled by ." which transmits the following in-line text to the selected output device. See ."

(;CODE) --- ASSEMBLER
The run-time procedure, compiled by ;CODE, that rewrites the code field of the most recently defined word to point to the following machine code sequence. See ;CODE.

(ABORT) --- ERROR
Executes after an error when WARNING is -1. This word normally executes ABORT, but may be altered (with care) to a user's alternative procedure.

(DO) --- STRUCTURE
The run-time procedure compiled by DO which moves the loop control parameters to the return stack. See DO.

(DOES>) --- DEFINING WORDS
The run-time procedure compiled by DOES>.

(FIND) addr1 addr2 --- pfa b tf (ok) DICTIONARY
addr1 addr2 --- ff (bad)
Searches the dictionary starting at the name field address addr2, matching to the text at addr1. Returns parameter field address, length byte of name field, and boolean true for a good match. If no match is found, only a boolean false is left.

(LINE) n scr# --- addr cnt EDITOR
Convert the line number n and the screen scr# to the disk buffer address containing the data. A count of 64 indicates the full line text length.

(LOOP) --- STRUCTURE
The run-time procedure compiled by LOOP which increments the loop index and tests for loop completion. See LOOP.

(NUMBER) d1 addr1 --- d2 addr2 CONVERSION
Convert the ASCII text beginning at addr1+1 with regard to BASE. The new value is accumulated into double number d1, being left as d2. Addr2 is the address of the first unconvertable digit. Used by NUMBER.

(OF) --- STRUCTURE
The run-time procedure compiled by OF.

***** n1 n2 --- n3 ARITHMETIC
Leave the signed product of two signed numbers.

***/** n1 n2 n3 --- n4 ARITHMETIC
Leave the ratio n4=n1*n2/n3 where all are signed numbers. Retention of an intermediate 31 bit product permits greater accuracy than would be available with the sequence : n1 n2 * n3 /

***/MOD** n1 n2 n3 --- n4 n5 ARITHMETIC
Leave the quotient n5 and remainder n4 of the operation n1*n2/n3 . A 31 bit intermediate product is used as for */.

+ n1 n2 --- n3 ARITHMETIC
Leave the sum of n1 + n2.

+! n addr --- ARITHMETIC
Add n to the value at the address. Pronounced "PLUS STORE".

+- n1 n2 --- n3 ARITHMETIC
Apply the sign of n2 to n1, which is left as n3.

+BUF addr1 --- addr2 f DISK
Advance the disk buffer address addr1 to the address of the next buffer addr2. Boolean f is false when addr2 is the buffer presently pointed to by variable PREV.

+LOOP n1 --- (run) STRUCTURE
addr n2 --- (compile)
Used in a colon-definition in the form:
DO ... n1 +LOOP
At run time, +LOOP selectively controls branching back to the corresponding DO based on n1. the loop index and the loop limit. The signed increment n1 is added to the index and the total compared to the limit. The branch back to DO occurs until the new index is equal to or greater than the limit (n1>0), or until the new index is equal to or less than the limit (n1<0). Upon exiting the loop, the parameters are discarded and execution continues ahead. At compile time, +LOOP compiles the run-time word (+LOOP) and the branch offset computed from HERE to the address left on the stack by DO. n2 is used for compile time error checking.

, n --- DICTIONARY
Store n into the next available dictionary cell, advancing the dictionary pointer. (comma)

- n1 n2 --- n3 ARITHMETIC
Leave the difference of n1 - n2.

--) --- EDITOR
Continue interpretation with the next disk screen. Pronounced "NEXT SCREEN".

-DUP n1 --- n1 (if zero) STACK
n1 --- n1 n1 (non-zero)
Reproduce n1 only if it is non-zero. This is usually used to copy a value just before IF, to eliminate the need for an ELSE part to drop it.

-FIND --- pfa cnt f (ok) DICTIONARY
--- ff (bad)
Accepts the next text word (delimited by blanks) in the input stream to HERE, and searches the CONTEXT and then CURRENT vocabularies for a matching entry. If found, the dictionary entry's parameter field address, its length byte, and a boolean true are left. Otherwise, only a boolean false is left.

-TRAILING addr n1 --- addr n2 MEMORY
Adjusts the character count n1 of a text string beginning at addr to suppress the output of trailing blanks. i.e. the characters at addr+n1 to addr+n2 are blanks.

. n --- PRINT
Print a number from a signed 16 bit two's complement value, converted according to the numeric BASE. A trailing blank follows. Pronounced "DOT".

." --- PRINT
Used in the form:
." cccc"
Compiles an in-line string cccc (delimited by the trailing ") with an execution procedure to transmit the text to the selected output device. If executed outside a definition, ." will immediately print the text until the final ". See (.").

.LINE n scr# --- PRINT
Print on the terminal device, a line of text from the disk by its line (n) and screen number. Trailing blanks are supressed.

.R n1 n2 --- PRINT
Print the number n1 right aligned in a field whose width is n2. No following blank is printed.

/ n1 n2 --- n3 ARITHMETIC
Leave the signed quotient of n1/n2.

/MOD n1 n2 --- rem n3 ARITHMETIC
Leave the remainder and signed quotient of n1/n2. The remainder has the sign of the dividend.

0 1 2 3 --- n ARITHMETIC
These small numbers are used so often that it is attractive to define them by name in the dictionary as constants.

0< n --- f STRUCTURE
Leave a true flag if the number is less than zero (negative), otherwise leave a false flag.

0= n --- f STRUCTURE
Leave a true flag if the number is equal to zero, otherwise leave a false flag.

0BRANCH f --- STRUCTURE
The run-time procedure to conditionally branch. If f is false (zero), the following in-line parameter is added to the interpretive pointer to branch ahead or back. Compiled by IF, UNTIL, and WHILE.

1+ n1 --- n2 ARITHMETIC
Increment n1 by 1.

1- n1 --- n2 ARITHMETIC
Decrement n1 by 1.

2+ n1 --- n2 ARITHMETIC
Leave n1 incremented by 2.

2- n1 --- n2 ARITHMETIC
Leave n1 decremented by 2.

: --- INTERPRET
Used in the form called a colon-definition:
: cccc ... ;
Creates a dictionary entry defining cccc as equivalent to the following sequence of FORTH word definitions '...' until the next ';' or ';CODE'. The compiling process is done by the text interpreter as long as STATE is non-zero. Other details are that the CONTEXT vocabulary is set to the CURRENT vocabulary and that words with the precedence bit set (P) are executed rather than being compiled. When colon definitions are compiled under the TRACE option, : takes on an alternate definition which allows the colon definition to be traced.

; --- INTERPRET
Terminates a colon-definition and stops further compilation. Compiles the run-time ;S.

;S --- INTERPRET
Stop interpretation of a screen. ;S is also the run-time word compiled at the end of a colon-definition which returns execution to the calling procedure.

```
<                    n1 n2 --- f                    STRUCTURE
  Leave a true flag if n1 is less than n2 otherwise leave a
  false flag.

<#                      ---                         CONVERSION
  Setup for pictured numeric output formatting using the
  words:
        <#  #  #S  SIGN  #>
  The conversion is done on a double number producing text at
  PAD.

<BUILDS                 ---                         INTERPRET
  Used within a colon-definition:
        : cccc <BUILDS  ...
                DOES>   ...  ;
  Each time cccc is executed, <BUILDS defines a new word with
  a high level execution procedure. Executing cccc in the
  form:
        cccc  nnnn
  uses <BUILD to create a dictionary entry for nnnn. When
  nnnn is later executed, it has the address of its parameter
  area on the stack and executes the words after DOES> in
  cccc. <BUILDS and DOES> allow run-time procedures to be
  written in high-level rather than in assembler code (as
  required by ;CODE).

=                    n1 n2 --- f                    STRUCTURE
  Leave a true flag if n1=n2; otherwise leave a false flag.

=CELLS              addr --- n2                     STACK
  This instruction expects an address or an offset to be on
  the stack. If this number is odd, it is incremented by 1 to
  put it on the next even word boundary. Otherwise, it
  remains unchanged.

>                    n1 n2 --- f                    STRUCTURE
  Leave a true flag if n1 is greater than n2; otherwise leave
  a false flag.

>R                     n ---                        STACK
  Remove a number from the computation stack and place as the
  most accessable on the return stack. Use should be balanced
  with R> in the same definition.

?                   addr ---                        PRINT
  Print the value contained at the address in free format
  according to the current BASE. This word must precede the
  address.

?COMP                   ---                         ERROR
  Issue error message if non-compiling.

?CSP                    ---                         ERROR
  Issue error message if stack position differs from value
  saved in CSP.

?ERROR                f n ---                       ERROR
  Issue an error message number n, if the boolean flag is
  true.

?EXEC                   ---                         ERROR
  Issue an error message if not executing.

?KEY                    --- ch                      KEYBOARD
  Scans the keyboard for input. If no key is pressed, a 0 is
  left on the stack. Else, the ascii code of the key pressed
  is left on the stack.

?KEY#                   --- n                       KEYBOARD
  Scans the keyboard for input. If no key is pressed, a 0 is
  left on the stack. Else, the 8-bit code of the key pressed
  is left on the stack.

?LOADING                ---                         ERROR
  Issue an error message if not loading.

?PAIRS               n1 n2 ---                      ERROR
  Issue an error message if n1 does not equal n2. The message
  indicates that compiled conditionnals do not match.

?STACK                  ---                         ERROR
  Issue an error message if the stack is out of bounds.

?TERMINAL               --- f                       KEYBOARD
  Perform a test on the terminal keyboard for actuation of the
  break key. A true flag indicates actuation. On the TI
  99/4A, the CLEAR key is used as the BREAK key.

@                   addr --- n                      MEMORY
  Leave the 16 bit contents of addr.

ABORT                   ---                         INTERPRET
  Clears the stacks and enter the execution state. Return
  control to the operator's terminal, printing an appropriate
  message.

ABS                  n1 --- n2                      ARITHMETIC
  Leave the absolute value of n1 as n2.

AGAIN              addr n --- (compiling)           STRUCTURE
  Used in a colon-definition in the form:
                BEGIN  ...  AGAIN
  At run-time, AGAIN forces execution to return to
  corresponding BEGIN. There is no effect on the stack.
  Execution cannot leave this loop (unless R) DROP is executed
  one level below).
  At compile time, AGAIN compiles BRANCH with an offset from
  HERE to addr. n is used for compile-time error checking.

ALLOT                  n ---                        DICTIONARY
  Add the signed number to the dictionary pointer DP. May be
  used to reserve dictionary space or re-origin memory.

ALTIN                   --- addr                    KEYBOARD
  A user variable whose value is 0 if input is coming from the
  keyboard else its value is a pointer to the UDP address
  where the PAB for the alternate input device is located.

ALTOUT                  --- addr                    PRINT
  A user variable whose value is 0 if output is going to the
  monitor else its value is a pointer to the UDP address where
  the PAB for the alternate output device is located.

AND                  n1 n2 --- n3                   LOGICAL
  Leave the bitwise logical AND of n1 and n2 as n3.

B/BUF                   --- n                       DISK
  This constant leaves the number of bytes per disk buffer,
  the byte count read from disk by BLOCK.

B/BUF#                  --- addr                    DISK
  A user variable which contains the number of bytes per
  buffer.

B/SCR                   --- n                       EDITOR
  This constant leaves the number of blocks per editing
  screen. By convention, an editing screen is 1024 bytes
  organized as 16 lines of 64 characters each.

B/SCR#                  --- addr                    EDITOR
  A user variable which contains the number of blocks per
  SCREEN.

BACK                addr ---                        DICTIONARY
  Calculate the backward branch offset from HERE to addr and
  compile into the next available dictionary memory address.

BASE                    --- addr                    CONVERSION
  A user variable containing the current number base used for
  input and output conversion.

BASE->R                 ---                         STACK
  Place the current base on the return stack. See R->BASE.

BEGIN                   --- addr n (compile)        STRUCTURE
  Occurs in a colon-definition in the form:
        BEGIN  ...  UNTIL
        BEGIN  ...  AGAIN
        BEGIN  ...  WHILE  ...  REPEAT
  At run-time, BEGIN marks the start of a sequence that may be
  repetitively executed. It serves as a return point from the
  corresponding UNTIL, AGAIN, or REPEAT. When executing
  UNTIL, a return to BEGIN will occur if the top of the stack
  is false; for AGAIN and REPEAT a return to BEGIN always
  occurs.
  At compile time, BEGIN leaves its return address and n for
  compiler error checking.

BL                      --- ch                      CONVERSION
  A constant that leaves the ASCII value for "blank".

BLANKS             addr cnt ---                     MEMORY
  Fill an area of memory beginning at addr with cnt blanks.

BLK                     --- addr                    INTERPRET
  A user variable containing the block number being
  interpreted. If zero, input is being taken from the
  terminal input buffer.

BLOAD               scr# --- f                      DISK
  Loads the binary image at scr# which was created by BSAVE.
  BLOAD returns a true flag (1) if the load was NOT successful
  and a false flag (0) if the load WAS successful.

BLOCK                  n --- addr                   DISK
  Leave the memory address of the block buffer containing
  block n. If the block is not already in memory, it is
  transferred from disk to whichever buffer was least recently
  written. If the block occupying that buffer has been marked
  as updated, it is rewritten to disk before block n is read
  into the buffer. See also BUFFER, R/W, UPDATE, and FLUSH.

BOOT                    ---                         INTERPRET
  Examines the SCREEN designated as the booting SCREEN (SCR
  3). If it contains only displayable characters (32-127) it
  performs a LOAD on that SCREEN.

BRANCH                  ---                         STRUCTURE
  The run-time procedure to unconditionally branch. An
  in-line offset is added to the interpretive pointer IP to
  branch ahead or back. BRANCH is compiled by ELSE, AGAIN,
  and REPEAT.

BUFFER                 n --- addr                   EDITOR
  Obtain the next memory buffer, assigning it to block n. If
  the contents of the buffer is marked as updated, it is
  written to disk. The block is not read from the disk.
  The address left is the first cell within the buffer for
  data storage.

C!                  b addr ---                      MEMORY
  Store 8 bits at addr. Bytes always occupy the low order
  bits when on the stack.

C,                     b ---                        DICTIONARY
  Store 8 bits of b into the next available dictionary byte,
  advancing the dictionary pointer. This instruction should
  be used with caution on byte addressing, word oriented
  computers such as the TI 9900.

C/L                     --- n                       EDITOR
  Returns on the stack the number of characters per line.

C/L#                    --- addr                    EDITOR
  A user variable whose value is the number of characters per
  line.

C@                  addr --- b                      MEMORY
  Leave the 8 bit contents of the memory address on the
  stack.

CASE                   n ---                        STRUCTURE
  Initiates the construct:
        CASE...OF...ENDOF...ENDCASE

CFA                 pfa --- cfa                     DICTIONARY
  Convert the parameter field address of a definition to its
  code field address.

CLEAR               scr# ---                        EDITOR
  Fills the designated screen with blanks.

CMOVE     addr1 addr2 cnt ---                       MEMORY
  Move the specified quantity of bytes beginning at addr1 to
  addr2. The contents of addr1 is moved first proceeding
  toward high memory.

COLD                    ---                         INTERPRET
  The COLD start procedure to adjust the dictionary pointer to
  the minimum standard and restart via ABORT. May be called
  from the terminal to remove application programs and
  restart. COLD calls BOOT prior to calling ABORT.

COMPILE                 ---                         INTERPRET
  When the word containing COMPILE executes, the execution
  address of the word following COMPILE is copied (compiled)
  into the dictionary. This allows specific compilation
  situations to be handled in addition to simply compiling an
  execution address (which the interpreter already does).
```

VDP MEMORY MAP

VDP MODES	FROM (HEX)	TO (HEX)	SIZE (HEX)	USAGE
TEXT	0000	03BF	03C0	SCREEN IMAGE TABLE
	03C0	03DF	0020	VDP ROLLOUT AREA
	03E0	045F	0080	STACK FOR VSPTR
	0460	077F	0320	PABS ETC
	0780	07FF	0080	SPRITE MOTION TABLE
	0800	0FFF	0800	PATTERN DESCRIPTOR TABLE &SPRITE DESCRIPTOR TABLE
	1000	13FF	0400	FORTH'S DISK BUFFER
	1400	3507	2108	UNUSED
	3508	3FFF	0A28	DISK BUFFERING REGION FOR 3 SIMULTANEOUS DISK FILES
GRAPHICS	0000	02FF	0300	SCREEN IMAGE TABLE
	0300	037F	0080	SPRITE ATTRIBUTE LIST
	0380	039F	0020	COLOR TABLE
	03A0	03BF	0020	UNUSED
	03C0	03DF	0020	VDP ROLLOUT AREA
	03E0	045F	0080	STACK FOR VSPTR
	0460	077F	0320	PABS ETC
	0780	07FF	0080	SPRITE MOTION TABLE
	0800	0FFF	0800	PATTERN DESCRIPTOR TABLE &SPRITE DESCRIPTOR TABLE
	1000	13FF	0400	FORTH'S DISK BUFFER
	1400	3507	2108	UNUSED
	3508	3FFF	0A28	DISK BUFFERING REGION FOR 3 SIMULTANEOUS DISK FILES
MULTI-COLOR	0000	02FF	0300	SCREEN IMAGE TABLE
	0300	037F	0080	SPRITE ATTRIBUTE LIST
	0380	039F	0020	COLOR TABLE
	03A0	03BF	0020	UNUSED
	03C0	03DF	0020	VDP ROLLOUT AREA
	03E0	045F	0080	STACK FOR VSPTR
	0460	077F	0320	PABS ETC
	0780	07FF	0080	SPRITE MOTION TABLE
	0800	0FFF	0800	PATTERN DESCRIPTOR TABLE &SPRITE DESCRIPTOR TABLE
	1000	13FF	0400	FORTH'S DISK BUFFER
	1400	3507	2108	UNUSED
	3508	3FFF	0A28	DISK BUFFERING REGION FOR
GRAPHICS2 (BIT MAP)	0000	17FF	1800	BIT MAP COLOR TABLE
	1800	1AFF	0300	BIT MAP SCREEN IMAGE TABLE
	1B00	1BFF	00C0	PABS ETC
	1BC0	1BFF	0040	STACK FOR VSPTR
	1C00	1FFF	0400	FORTH'S DISK BUFFER
	2000	37FF	1800	BIT MAP PATTERN DESC. TABLE
	3800	387F	0080	SPRITE ATTRIBUTE LIST
	3880	39D9	015A	SPRITE DESCRIPTORS
	39DA	3FFF	0626	DISK BUFFERING REGION FOR 2 SIMULTANEOUS DISK FILES

CPU RAM PAD

FROM (HEX)	TO (HEX)	SIZE (HEX)	USAGE
8300	831F	0020	FORTH'S WORKSPACE
832E	8347	001A	FORTH'S INNER INTERPRETER ETC.
834A	8351	0008	FAC floating point accumulator
8356	8357	0002	SUBROUTINE POINTER FOR DSR's
835C	8363	0008	ARG floating point argument register
8370	8371	0002	HIGHEST AVAILABLE ADDRESS OF VDP RAM
8372	8372	0001	LEAST SIGNIFICANT BYTE OF DATA STACK PTR
8373	8373	0001	LEAST SIGNIFICANT BYTE OF SUBR STACK PTR
8374	8374	0001	KEYBOARD NUMBER TO BE SCANNED
8375	8375	0001	ASCII KEYCODE DETECTED BY SCAN ROUTINE
8376	8376	0001	JOYSTICK Y-STATUS
8377	8377	0001	JOYSTICK X-STATUS
8379	8379	0001	VDP INTERRUPT TIMER
837A	837A	0001	NUMBER OF SPRITES THAT CAN BE IN AUTOMOTION
837B	837B	0001	VDP STATUS BYTE BIT 0 ON DURING VDP INTERRUPTS BIT 1 ON WHEN 5 SPRITES ON A LINE BIT 2 ON WHEN SPRITE COINCIDENCE BIT 3-7 NO. OF 5TH SPRITE ON A LINE
837C	837C	0001	GPL STATUS BYTE BIT 0 HIGH BIT BIT 1 GRATER THAN BIT BIT 2 ON WHEN KEYSTROKE DETECTED (COND) BIT 3 CARRY BIT BIT 4 OVERFLOW BIT
8380	8380	0001	THE DEFAULT SUBROUTINE STACK ADDRESS
83A0	83A0	0001	THE DEFAULT DATA STACK ADDRESS
83C0	83C0	0001	RANDOM NUMBER SEED
83C2	83C2	0001	BEGIN INTERRUPT WORKSPACE BIT 0 DISABLE ALL OF THE FOLLOWING BIT 1 DISABLE SPRITE MOTION BIT 2 DISABLE AUTO SOUND BIT 3 DISABLE SYSTEM RESET KEY (QUIT)
83C4	83C4	0001	LINK TO ISR HOOK
83D4	83D4	0001	CONTENTS OF VDP REGISTER 1
83E0	83FF	0020	BEGIN GPL WORKSPACE

SP@ --- addr STACK
A procedure to return the address of the stack position to the top of the stack, as it was before SP@ was executed. (e.g. 1 2 SP@ @ . . . would type 2 2 1).

SPACE --- PRINT
Transmit an ascii blank to the output device.

SPACES n --- PRINT
Transmit n ascii blanks to the output device.

SRA n1 cnt --- n2 ARITHMETIC
Arithmetically shifts n1 cnt bits to the right and leaves the result on the stack. cnt will be modulo 16, except when cnt=0, when 16 bits will be shifted. To create a word which permits shifts when cnt could be zero, use the following definition: : SRA0 -DUP IF SRA ENDIF ;

SRC n1 cnt --- n2 ARITHMETIC
Performs a circular right shift of cnt bits on n1 leaving the result on the stack.

SRL n1 cnt --- n2 LOGICAL
Performs a logical right shift of cnt bits and leaves the result on the stack. cnt will be modulo 16, except when cnt=0, when 16 bits will be shifted. To create a word which permits shifts when cnt could be zero, use the following definition: : SRL0 -DUP IF SRL ENDIF ;

STATE --- addr INTERPRET
A user variable containing the compilation state. A non-zero value indicates compilation. The value itself may be implementation dependent.

SWAP n1 n2 --- n2 n1 STACK
Exchange the top two values on the stack.

SWPB n1 --- n2 MEMORY
Reverses the order of the two bytes in n1 and leaves the new number as n2.

SYS@ --- addr INTERPRET
A user variable that contains the address of the system support entry point.

SYSTEM n --- INTERPRET
Calls a system synonyms. You must specify an offset n into a jump table for the routine you wish to call. n must be one of the predefined even numbers.

TASK --- DICTIONARY
A no-operation word which can mark the boundary between applications. By forgetting TASK and re-compiling, an application can be discarded in its entirety.

THEN --- STRUCTURE
An alias for ENDIF.

TIB --- addr KEYBOARD
A user variable containing the address of the terminal input buffer.

TOGGLE addr b --- MEMORY
Complement the contents of the byte at addr by the bit pattern b.

TRAVERSE addr1 n --- addr2 DICTIONARY
Move accross the name field of a fig-FORTH variable length name field. addr1 is the address of either the length byte or the last letter. If n=1, the motion is toward high memory; if n=-1, the motion is toward low memory. The addr2 resulting is the address of the other end of the name.

TRIAD scr# --- PRINT
Display on the RS232 the three SCREENS which include that number scr#, beginning with a SCREEN evenly divisible by three. Output is suitable for source text records, and includes a reference line at the bottom taken from line 15 of screen 4.

TRIADS scr# scr# --- PRINT
May be thought as a multiple TRIAD. You must specify a SCREEN range. TRIADS will perform as many TRIAD's as necessary to cover that range.

TYPE addr cnt --- PRINT
Transmit count characters from addr to the selected output device.

U --- n STACK
Places the contents of register U on the stack. Register U contains the base address of the user variable area.

U* u1 u2 --- ud ARITHMETIC
Leave the unsigned double number product of two unsigned numbers.

U. u --- PRINT
Prints an unsigned number to the output device.

U.R u n --- PRINT
Prints an unsigned number right justified in a field of width n.

U/ ud u1 --- u2 u3 ARITHMETIC
Leave the unsigned remainder u2 and unsigned quotient u3 from the unsigned double dividend ud and unsigned divisor u1.

U0 --- addr INTERPRET
A user variable that points to the junction between the user variable area and the return stack.

UK u1 u2 --- f STRUCTURE
Leaves a true flag if u1 is less than u2, else leaves a false flag.

UCONS@ --- addr INTERPRET
A user variable which contains the base address of the user variable default area which is used to initialize the user variables at COLD.

UD. ud --- PRINT
Prints an unsigned double number to the output device.

UD.R ud n --- PRINT
Prints an unsigned double number right justified in a field of length n.

UNFORGETABLE addr --- f DICTIONARY
Decides whether or not a word can be forgotten. A true flag is returned if the address is not located between FENCE and HERE.

UNTIL f --- (run-time) STRUCTURE
addr n --- (compile)
Occurs within a colon-definition in the form:
BEGIN ... UNTIL
At run-time, UNTIL controls the conditional branch back to the corresponding BEGIN. If f is false, execution returns to just after BEGIN; if true, execution continues ahead. At compile-time, UNTIL compiles (0BRANCH) and an offset from HERE to addr. n is used for error tests.

UPDATE --- DISK
Marks the most recently referenced block (pointed to by PREV) as altered. The block will subsequently be transferred automatically to disk should its buffer be required for storage of a different block.

USE --- addr EDITOR
A variable containing the address of the block buffer to use next, as the least recently written.

USER n --- MEMORY
A defining word used in the form:
n USER cccc
which creates a user variable cccc. The parameter field of cccc contains n as a fixed offset relative to the user pointer register UP for this user variable. When cccc is later executed, it places the sum of its offset and the user area base address on the stack as the storage address of that particular variable.

VARIABLE n --- MEMORY
A defining word used in the form:
n VARIABLE cccc
When VARIABLE is executed, it creates the definition cccc with its parameter field initialized to n. When cccc is later executed, the address of its parameter field (containing n) is left on the stack, so that a fetch or store may access this location.

VOC-LINK --- addr DICTIONARY
A user variable containing the address of a field in the definition of the most recently created vocabulary. All vocabulary names are linked by these fields to allow control for FORGETing thru multiple vocabularies.

VOCABULARY --- DICTIONARY
A defining word used in the form:
VOCABULARY cccc
to create a vocabulary definition cccc. Subsequent use of cccc will make it the CONTEXT vocabulary which is searched first by INTERPRET. The sequence "cccc DEFINITIONS" will also make cccc the CURRENT vocabulary into which new definitions are placed.
cccc will be so chained as to include all definitions of the vocabulary in which cccc is itself defined. All vocabularies ultimately chain to FORTH. By convention, vocabulary names are to be declared IMMEDIATE. See VOC-LINK.

WARNING --- addr ERROR
A user variable containing a value controlling messages. If =1 disk is present, and screen 4 of drive 0 is the base location for messages. If =0, no disk is present and messages will be presented by number. If =-1, execute (ABORT) for a user specified procedure. See MESSAGE, ERROR.

WDISK addr n1 n2 --- n3 DISK
The primitive routine which performs a disk write. addr is the CPU RAM location of the block to be written. n1 is the block number, n2 is the number of bytes per block, and n3 is the returned error code.

WHILE f --- (run-time) STRUCTURE
addr1 n1 --- adr1 n1 adr2 n2
Occurs in a colon-definition in the form:
BEGIN ... WHILE(tp) ... REPEAT
At run-time, WHILE selects conditional execution based on boolean flag f. If f is true (non-zero), WHILE continues execution of the true part thru to REPEAT, which then branches back to BEGIN. If f is flase (zero), execution skips to just after REPEAT, exiting the structure.
At compile-time, WHILE emplaces (0BRANCH) and leaves addr2 of the reserved offset. The stack values will be resolved by REPEAT.

WIDTH --- addr DICTIONARY
A user variable containing the maximum number of letters saved in the compilation of a definiton's name. It must be 1 thru 31, with a default value of 31. The name character count and its natural characters are saved, up to the value in WIDTH. The value may be changed at any time within the above limits.

WORD ch --- MEMORY
Read the text characters from the input stream being interpreted, until a delimiter ch is found, storing the packed character string beginning at the dictionary buffer HERE. WORD leaves the character count in the first byte, the characters, and ends with two or more blanks. Leading occurances of ch are ignored. If BLK is zero, text is taken from the terminal input buffer, otherwise from the dist block stored in BLK. See BLK, IN.

XOR n1 n2 --- n3 LOGICAL
Leave the bitwise logical EXCLUSIVE OR of two values.

[--- INTERPRET
Used in a colon-definition in the form:
: xxxx [words] more ;
Suspend compilation. The words after [are executed, not compiled. This allows calculation or compilation exceptions before resuming compilation with]. See LITERAL,].

[COMPILE] --- INTERPRET
Used in a colon-definition in the form:
: xxxx [COMPILE] FORTH ;
[COMPILE] will force the compilation of an immediate definition, that would otherwise execute during compilation. The above example will select the FORTH vocabulary when xxxx executes, rather than at compile time.

] --- INTERPRET
Resume compilation, to the completion of a colon-definition. See [.

CONSTANT n --- MEMORY
A defining word used in the form:
n CONSTANT cccc
to create word cccc, with its parameter field containing n. When cccc is later executed, it will push the value of n to the stack.

CONTEXT --- addr INTERPRET
A user variable containing a pointer to the vocabulary within which dictionary searches will first begin.

COUNT addr1 --- addr2 n MEMORY
Leave the byte address (addr2) and byte count (n) of a message text beginning at addr1. It is presumed that the first byte at addr1 contains the text byte count and the actual text starts with the second byte. Typically, COUNT is followed by TYPE.

CR --- PRINT
Transmit a carriage return and a line feed to the selected output device.

CREATE --- DICTIONARY
A defining word used in the form:
CREATE cccc
by such words as CODE and CONSTANT to create a dictionary header for a FORTH definition. The code field contains the address of the word's parameter field. The new word is created in the CURRENT vocabulary.

CSP --- addr STACK
A user variable temporarily storing the stack pointer position, for compilation error checking.

CURPOS --- addr UDP
A user variable that stores the current UDP cursor position.

CURRENT --- addr INTERPRET
A user variable pointing to the vocabulary into which new definitions will be compiled.

D+ d1 d2 --- d3 ARITHMETIC
Leave a double number sum of two double numbers.

D+- d1 n --- d2 ARITHMETIC
Apply the sign of n to the double number d1, leaving it as d2.

D. d --- PRINT
Print a signed double number from a 32 bit two's complement value. The high order 16 bits are most accessable on the stack. Conversion is performed according to the current BASE. A blank follows. Pronounced "D DOT".

D.R d n --- PRINT
Print a signed double number d right aligned in a field n characters wide.

DABS d1 --- d2 ARITHMETIC
Leave the absolute value of a double number.

DECIMAL --- CONVERSION
Set the numeric conversion BASE for decimal input/output.

DEFINITIONS --- INTERPRET
Used in the form:
cccc DEFINITIONS
Set the CURRENT vocabulary to the CONTEXT vocabulary. In the example, executing vocabulary name cccc made it the CONTEXT vocabulary and executing DEFINITIONS made both specify vocabulary cccc.

DIGIT ch n1 --- n2 tf (ok) CONVERSION
ch n1 --- ff (bad)
Convert the ascii character ch (using BASE n1) to its binary equivalent n2, accompanied by a true flag. If the conversion is invalid, leave only a false flag.

DISK_BUF --- addr DISK
A user variable that points to the first byte in UDP RAM of the 1K disk buffer.

DISK_HI --- addr DISK
A user variable which contains the SCREEN number immediately above the SCREEN range wherein SCREEN writes are permitted.

DISK_LO --- addr DISK
A user variable which contains the first SCREEN number of the range wherein disk writes are permitted.

DISK_SIZE --- addr DISK
A user variable whose value is the number of SCREENS logically assigned to a diskette.

DLITERAL d --- d (executing) INTERPRET
d --- (compiling)
If compiling, compile a stack double number into a literal. Later execution of the definition containing the literal will push it on the stack. If executing, the number will remain on the stack.

DMINUS d1 --- d2 ARITHMETIC
Convert d1 to its double number two's complement.

DO n1 n2 --- (execute) STRUCTURE
addr n --- (compile)
Occurs in a colon-definition in the form:
DO ... LOOP or DO ... +LOOP
At run-time, DO begins a sequence with repetitive execution controlled by a loop limit n1 and an index with initial value n2. DO removes these from the stack. Upon reaching LOOP, the index is incremented by 1. Until the new index equals or exceeds the limit, execution loops back to just after DO; otherwise the loop parameters are discarded and execution continues ahead. Both n1 and n2 are determined at run-time and may be the result of other operations. Within a loop, I will copy the current value of the index to the stack. When compiling within the colon-definition, DO compiles (DO), leaving the following address (addr) and n for later error checking. See I, LOOP, +LOOP and LEAVE.

DOES> --- INTERPRET
A word which defines the run-time action within a high-level defining word. DOES> alters the code field and first parameter of the new word to execute the sequence of compiled word addresses following DOES>. It is always used in combination with <BUILDS. When the DOES> part executes it begins with the address of the first parameter of the new word on the stack. This allows interpretation using this area of its contents. Typical uses include the FORTH assembler, multi-dimensional arrays, and compiler generation.