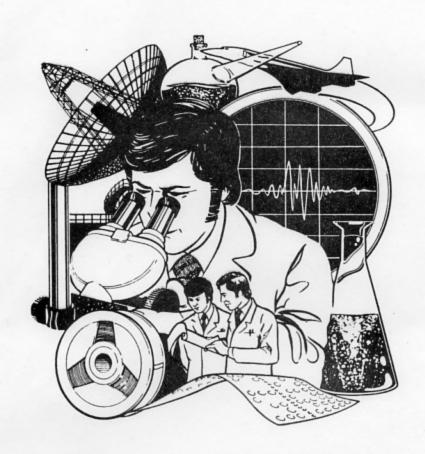
F.I.G FOR



SYNTAX PRODUCTION

A SYMTAX FOR PRODUCTION



F.I.G FORTH

F.I.G. FORTH

MTX VERSION BY KEITH JONES. PRODUCED BY K. HOOK ARTWORK SYNART

THIS VERSION IS PROVIDED THROUGH THE COURTESY OF THE FORTH INTEREST GROUP P.O. BOX 1105. SAN CARLOS, CA 94070

MTX fig-FORTH 1.12 (C) Keith Jones 1985

Provided through the courtesy of the FORTH INTEREST GROUP, PO Box 1105. San Carlos, CA 94070.

Thanks to:
Trevor for help and ideas.
Vaughan for patience, sustanance and understanding.

The aim of this manual is to give the new user a basic introduction to FORTH. It is not a complete teaching aid, as There are many books on the market which can do this better than I. The manual is divided into three parts. The first part describes the fig-FORTH words -a word in FORTH is the name of any definition. It is not to be confused with its meaning in machine code for a sixteen bit value.

The second section is devoted to the MTX extensions and the final part is devoted to the screen editor.

FORTH is a strange language, but it is fast, compact and efficient. I hope you have fun using it.At present work is commencing on a cross compiler so that your definitions can run independently of the main body. I will be pleased to answer any questions or enquires about this program, or FORTH in general.All mail should be addressed to me c/o GENPAT.

Next to each word you will see symbols e.g. next to FILL are the symbols addr quan b --- .Anything to the left of the three dashes is expected to be on the stack for the word to execute. Symbols to the right of the dashes, if any, are values left on the stack after the word has been executed. All the symbols are explained in the key before the definitions.

NUMBERS

FORTH deals with numerical values in four different ways.It will accept signed integers in the range of -32,768 to +32,767.Unsigned integers (0 to 65,535) are accepted and both of these types of numbers are called single length values.Double length numbers are either signed (-2,147,483,648 to +2,147,483,647) or unsigned (0 to 4,294,967,295).

MATHS

FORTH uses post-fix maths, also known as Reverse Polish Notation. It is worth working with the maths until you are fully aware of its operation.

To use post-fix you first place the values on the stack and then supply the maths operator <code>,e.g."4+2"</code> would be "4 2 +","1/2" would be "1 2 /".A space must be typed between each number and all operations. It may take some time to get used to this way of working, but it is faster as computers use post-fix within their own floating point maths.

The above examples were easy, but what about an expression like "((10+3)*5)/2". This is just as easy as the above, it just involves more operators! The stack is a last in first out stack which grows down in memory, so if you type 3 <RET> 4 <RET> the stack has 4 at the top and 3 below it. When you type + <RET> the maths interpreter takes 4 and 3 off the stack, adds them together and leaves the result ,7, on top of the stack. So now to our more complex equation you would type 10 <RET> 3 <RET> + <RET> 5 <RET> * <RET> (RET> and this would leave the result on top of the stack. You don't need to type all the <RET>'s in of course, 10 3 + 5 * 2 / <RET> would leave the same result on the stack. Dont't forget that FORTH is integer only maths. Floating point can be done in FORTH, but it slows everything down. FORTH does provide MOD operations, which are explained in the FORTH definitions section of this manual.

CASE CONSTRUCT

An additional feature of MTX-FORTH is the CASE statement. This replaces multiple IF...ELSE...THEN statements and makes the listing easier to follow.

For instance :

: EXAMPLE

DUP 0=IF ." ZERO " ELSE

DUP 1=IF ." ONE " ELSE

DUP 2=IF ." TWO " ELSE

DUP 3=IF ." THREE " ELSE

THEN THEN THEN THEN DROP ;

Can be replaced by ;

: EXAMPLE

O OF ." ZERO " ENDOF

1 OF ." ONE " ENDOF

2 OF ." TWO " ENDOF

3 OF ." THREE " ENDOF

ENDCASE ;

As you can see, this is much more compact and easier to follow. If it fails all tests then it will not execute any of the statements.

SYNTAX

Syntax in FORTH will seem curious at first. Having to type 4 VS rather than VS 4 seems silly and unnecessary, but this is not so. Some commands need a value on the stack before they can operate, hence the number must come first.

The same syntax applies to conditional statements e.g.the BASIC statement IF A=1 THEN "condition1" ELSE "condition2" translates in FORTH to A 1 = IF "condition1" else "condition2" THEN. THEN in FORTH can also be called ENDIF.

ERROR MESSAGES

Should the compiler find an error at any point then it clears all stacks and returns an error message with a numerical value. These values mean;

Meaning
Word does not exist.
Stack is empty.
Dictionary full.
Incorrect address mode.
Definition for this word already exists.
Outside disc range (Pages 0-4).
Stack full.
Use in a colon word only.
Execution only.
Condition not paired.
Definition not finished.
In protected dictionary.
Use only when LOADING.
Off current editing screen.

- 24 Declare vocabulary.
- ! n addr --- LO Store 16 bits of n at address.Pronounced "store".
- !CSP
 Save the stack position in CSP.Used as part of the compiler security
- # d1 --- d2 LO

 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 count LO Terminates numeric output conversion by dropping d,leaving the text address and character count suitable for TYPE.
- #BUF --- n
 A constant which leaves the number of buffers on the stack.
- #S d1 --- d2
 Generates ascii text in the text output buffer, by the use of #,until a zero double number n2
 results.Used between <# and #>.
 - --- addr P,LO
 Leaves the parameter field address of dictionary word nnnn.As a compiler directive, executes in a colon definition to compile the address as a literal .If the word is not found after a search of CONTEXT and CURRENT, an appropriate error message is given.Pronounced "tick".
- Used in the form (cccc)

 Ignore a comment that will be delimited by a right parenthesis on the same line. May occur during execution or in a colon definition. A blank after the leading paranthesis is required.
- (.")

 C+

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

 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.
- (+LOOP) n --- C2

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

 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.
- (DD)

 The run-time procedure compiled by DO which moves the loop control parameters to the return stack. See DO
- (FIND) addr1 addr2 --- pfa b tf (ok)
 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) n1 n2 --- addr count

Convert the line number n1 and the screen n2 to the disc buffer address containing the data.A count of 64 indicates the full line text length.

(LOOP) C2

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

(NUMBER) d1 addr1 --- d2 addr2

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.

n1 n2 --- prod L0

Leaves the signed product of two signed numbers.

*/ n1 n2 n3 --- n4 L0

Leaves 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 L0

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

n1 n2 --- sum L0

Leave the sum of n1 and n2.

+! n addr --- L0

Add n to the value at address.Pronounced "plus store".

- n1 n2 --- n3

Apply the sign of n2 to n1, which is left in n3.

+BUF addr1 --- addr2 f

Advance the disc 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.

+L00P n1 --- (run)

addr n2 --- (compile) P,C2,L0

Used in a colon-definition in the form:

DO ... n1 +LOOP

At run-time, +L00P selectively controls branching back to the corresponding DO based on n1, the loop 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.

+ORIGIN n --- addr

Leave the memory address relative by n to the origin parameter area.n is the minimum address unit, either byte or word. This definition is used to access or modify the boot-up parameters at the origin area.

Store n into the next available dictionary memory cell,advancing the dictionary

pointer.(comma).

n1 n2 --- diff

Leave the difference of n1-n2.

--> P,LO

Continue interpretation with the next disc screen. (pronounced next-screen).

LO

-DUP n1 --- n1 (if zero) 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 b tf (found)
--- ff (not found)

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 is left. Otherwise, only a boolean false is left.

-TRAILING addr n1 --- addr n2

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

1 --- LO

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

." P.LO

Used in the form:

." cccc"

Compiles an in-line string cccc (delimited by a 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 ". The maximum number of characters may be an installation dependent value. See (.").

.LINE line scr ---

Print on the terminal device, a line of text from the disc by its line and screen number. Trailing blanks are suppressed.

.R n1 n2 ---

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

/ n1 n2 --- quot LC
Leave the signed quotient of n1/n2

/MOD n1 n2 --- rem quot L0

Leave the remainder and signed quotient of n1/n2. The remainder has the sign of the dividend.

0 1 2 3 --- n

These small numbers are used so often that it is attractive to define them by name in the dictionary as constants.

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

O= n --- f LO

Leave a true flag if the number is equal to zero, otherwise leave a false flag.

OBRANCH f ---

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

C2

1+ n1 --- n2 L1

Increment n1 by 1.

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

2! d addr --Stores 32 bits of d at addr.d is split into two 16-bit values (nlow and nhigh) with nhigh
stored at addr and nlow stored at addr+2.

20 addr --- d
Fetches 32-bit value from addr.d is placed on the stack as two 16-bit values (nlow and nhigh).

2DROP d --Drops a 32-bit value from the stack.

20UP d --- d d

Duplicates a 32-bit value.

25WAP d1 d2 --- d2 d1
Will swap two 32-bit values when on the stack.

P,E,LO
Used in the form called a colon-definition:

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.

P,C,LO

Terminate a colon-definition and stop further compilation. Compiles the run-time ;S.

:CODE P,C,LO

Used in the form:

: cccc ;CODE

assembly mnemonics

Stop compilation and terminate a new defining word cccc by compiling (;CODE). Set the CONTEXT vocabulary to ASSEMBLER, assembling to machine code the following mnemonics. When cccc later executes in the form:

cccc onno the word norm will be created with its execution procedure given by the machine code following cccc. That is, when norm is executed, it does so by jumping to the code after norm. An existing defining word must exist in cccc prior to ;CODE.

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

n1 n2 --- f L0 Leave a true flag if n1 is less than n2; otherwise leave a false flag.

<#

Setup for pictured numeric output formatting using the words: <# # #S SIGN #>

The conversion is done on a double number producing text at PAD.

<BUILDS

C,LO

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 <BUILDS to create a dictionary entry for nnnn with a call to the DOES> part 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

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

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

>R C.LO

> 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.

Print the value contained at the address in free format according to the current base.

?COMP

Issue error message if not compiling.

?CSP

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

?ERROR f n ---

Issue error message number n, if the boolean flag is true.

?EXEC

Issue an error message if not executing.

?LOADING

Issue an error message if not loading.

n1 n2 ---?PAIRS

> Issue an error message if n1 does not equal n2. The message indicates that compiled conditionals do not match.

?STACK

Issue an error message if the stack is out of bounds. This definition may be installation dependent.

LO addr --- n Leave the 16 bit contents of address.

ABORT

LO Clear the stacks and enter the execution state. Return control to the operators terminal, printing a message appropriate to the installation.

ABS

n --- u

LO

Leave the absolute value of n as u.

AGAIN

addr n --- (compiling) P,C2,L0

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

LO n ---Add the signed number to the dictionary pointer DP.May be used to reserve dictionary space or re-origin memory.n is with regard to computer address type (byte or word).

AND

n1 n2 --- n2

Leave the bitwise logical and of n1 and n2 as n3.

B/BUF

This constant leaves the number of bytes per disc buffer, the byte count read from disc by BLOCK.

B/SCR

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

BACK

addr ---

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

BASE

--- addr

U,LO

A user variable containing the current number base used for input and output conversion.

BEGIN

--- addr n (compiling) P,LO

Occurs in a colon-definition in 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 or BEGIN will occur if the top of the stack is false; for AGAIN and REPEAT a return to BEGIN will always occur.

At compile time BEGIN leaves its return address and n for compiler error checking.

BL

A constant that leaves the ascii value for "blank".

BI ANKS

addr count ---

Fill an area of memory beginning at addr with blanks.

BLK

--- addr

U.LO

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

BLOCK

LO

Leave the memory address of the block buffer containing block n.If the block is not already in memory, it is transferred from disc to which ever buffer was least recently written. If the

block occupying that buffer has been marked as updated, it is rewritten to disc before block n is read into the buffer. See also BUFFER, R/W UPDATE FLUSH.

BRANCH

C2.L0

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

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

C! b addr ---

Store 8 bits at address.

C/L --- n

Constant which leaves the number of characters per line on the stack. By FORTH convention this is 64

C, b ---Store 8 bits of b into the next available dictionary byte, advancing the dictionary pointer.

C0 addr --- b
Leave the 8 bit contents of memory address

CASE --- n

Must be used within a colon definition. For use see introduction. The value on the stack is for compiler error checking.

CFA pfa --- cfa

Convert the parameter field address of a definition into its code field address

C2

CMOVE from to count ---

Move the specified number of bytes beginning at address from to address to. The contents of address from is moved first preceding toward high memory.

COLD

The cold start procedure to adjust the dictionary pointer to the minimum standard and restart via ABORT.

COMPILE

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).

CONSTANT n == LO

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 onto the stack.

CONTEXT --- addr U.LO

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

COUNT addr1 --- addr2 n LO

Leave the byte address addr2 and byte count n for a message text beginning at address addr1.It

is preassumed that the first byte at addr1 contains the text byte count and the actual text

9

starts with the second byte. Typically COUNT is followed by TYPE.

CR

LO

U

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

CREATE

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 words parameter field. The new word is created in the CURRENT vocabulary.

CSP --- addr

A user variable temporarily storing the stack pointer position, for compilation error checking.

D+ d1 d2 --- dsum

Leave the double number sum of two double numbers.

D+- d1 n --- d2

Apply the sign of n to the double number d1, leaving it as d2.

D. d --- L1

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 dn---

Print a signed double number d right aligned in a field n characters wide.

DABS d --- uc

Leave the absolute value ud of a double number.

DECIMAL LO

Set the numeric conversion BASE for decimal input-output.

DEFINITIONS

L1

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 c n1

Converts the ascii character c (using base n1) to its binary equivalent n2,accompanied by a true flag. If the conversion is invalid, leaves only a false flag.

DLITERAL d --- d (executing)
d --- (compiling)

If compiling, compile a stack double number into a literal. Later execution of the definition containing the literal will push it to the stack. If executing, the number will remain on the stack.

DMINUS d1 --- d2

Convert d1 to its double number two's complement.

DO

n1 n2 --- (execute) addr n --- (compile) P,C2,LD

Occurs in a colon-definition in form:

DO ... LOOP

At run-time,00 begins a sequence with repetitive execution controlled by a loop limit n1 and an index with initial value n2.00 removes these from the stack.Upon reaching LOOP the index is incremented by one.Until the new index equals or exceeds the limit, execution loops back to just after the 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.See I,LOOP,+LOOP,LEAVE.

When compiling within the colon-definition,DO compiles (DO),leaves the following address addr and n for later error checking.

DOES>

LO

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>.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 or its contents.Typical uses include the FORTH assembler,multidimensional arrays,and compiler generation.

DP

11.1

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

DPL

--- addr U,LO

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 the output column location of a decimal point, in user generated formatting. The default value on single number input is -1.

DROP

n ---

Drop the number from the stack

--- addr

DUP

n --- n n

Duplicate the value on the stack.

EDITOR

Make current vocabulary EDITOR, for screen editing see section 3

ELSE

addr1 n1 --- addr2 n2

(compiling)

P,C2,L0

LO

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 resumes execution after the ENDIF.It has no stack effect.

At compile time ELSE emplaces BRANCH reserving a branch offset, 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 at addr1

EMIT

L

Transmit ascii character c to the selected output device.OUT is incremented for each character output.

EMPTY-BUFFERS

10

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

11

ENCLOSE addr1 c --- addr1 n1 n2 n3

> The text scanning primitive used by WORD. From the text address addr1 and an ascii delimiting character c, is determined the byte offset to the first non-delimiter character n1, the offset to the first 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.

END P.C2.LO

addr n ---

This is an 'alias' or duplicate definition for UNTIL.

Must be used within a colon-definitipn. For use see introduction. The values on the stack are for compiler error checking.

ENDIF addr n --- (compile) P.CO.LO

Occurs in a colon definition in 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.

ENDOF addr n ---

ENDCASE

Must be used within a colon-definition. For use see introduction. The values on the stack are for compiler error checking.

FRASE addr n ---

Clear a region of memory to zero from addr over n addresses.

ERROR line --- in blk

> Execute error notification and restart os system.WARNING is first examined. If 1, the text of line n.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,n is just printed as a message number (non disc 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 and BLK to assist in determining the location of the error. Final action is execution of QUIT

EXECUTE addr ---

> Execute the defition whose code field address is on the stack. The code field address is also called the compilation address.

EXPECT 10 addr count ---

Transfer characters from the terminal to address, until a 'return' or the count of charaters have been received. One or more nulls are added to the end of the text.

FENCE

A user variable containing an address below which FORGETting is trapped. To forget below this point the user must alter the contents of FENCE.

FILL addr quan b ----Fill memory at the address with the specified quantity of bytes b.

FIRST

A constant that leaves the address of the first (lowest) block buffer

FLD

--- addr

A user variable for control of number output field width. Presently unused in fig-FORTH.

FLUSH

Will write all screens to RAM-DISC area. Must be used before SAVET. It is recommended that FLUSH is used after every EDITOR session.

FORGET

E,LO

Expected in the form:

FORGET cccc

Deletes definition marked cccc from the dictionary with all entries physically following it. In fig-FORTH, an error message will occur if the CURRENT and CONTEXT vocabularies are not currently the same.

FORTH

P.L1

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.

HERE

--- addr

--- addr

LO

Leave the address of the next available dictionary location.

HEX

LO

LO

LO

Set the numeric conversion base to sixteen (hexadecimal).

HLD

A user variable that holds the addresses of the latest character of text during numeric output conversion.

HOLD

C ---

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

I

C.LO

Used within a 00--L00P to copy the loop index to the stack.Other use is implementation dependent.See R.

ID.

addr ---

Print a definition's name from its name field address.

IF

f --- (run-time)

--- addr n (compile) P,C2,L0

Occurs in a colon-definition in 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 through the true part. If f is false (zero), execution skips till just after the ELSE to execute the false part. After either 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 OBRANCH and reserves space for an offset at addr.addr and n are used later for resolution of the offset and error testing.

IMMEDIATE

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 preceding it with [COMPILE].

IN

--- addr

LO

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

INDEX

from to ---

Print the first line of each screen over the range from, to. This is used to view the comment lines of an area of text on disc screens.

INTERPRET

The outer text interpreter which sequentially executes or compiles text from the input stream (terminal or disc) depending on STATE. If the word name cannot be found after a search of CONTEXT and then CURRENT it is converted to 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 to 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.

KEY

Leave the ascii value of the next terminal key pressed.

LATEST

--- addr

Leave the name field address of the top most word in the CURRENT vocabulary.

LEAVE

C.LO

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

Convert the parameter field address of a dictionary definition to its link field address.

LIMIT

A constant leaving the address just above the highest memory available for a disc buffer. Usually this is the highest system memory.

LINK

If n>O then anything which is printed on the screen will also be sent to the printer attached to the centronics interface.

LIST

n ---

Display the ascii text of a screen n on the selected output device.SCR contains the screen number during and after this process.

LIT

C2,L0

LO

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 to the stack.

LITERAL

n --- (compiling)

P.C2.LO

If compiling, then compile the stack value of n as a 16-bit literal. This definition is immediate so that it 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

LO Begin interpretation of screen n.Loading will terminate at the end of the screen or at ;s. See \$5 and -->.

LOOP

addr n --- (compiling) F,C2,L0

Occurs in a colon definition in 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

A mixed magnitude math operation which leaves the double number signed product of two signed numbers.

M/

d n1 --- n2 n3

A mixed magnitude math operator which leaves the signed remainder n2 and signed quotient n3, from a double number dividend and dividor n1. The remainder takes its sign from the dividend.

m/mod

ud1 u2 --- u3 ud4

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

MAX

n1 n2 --- max

LO

Leave the greater of two numbers.

MESSAGE

n --

Print on the selected output device the text of line n relative to screen 4 of drive O.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 (disc un-available).

MIN

n1 n2 --- min

LO

Leave the smaller of two numbers.

MINUS

n1 --- n2

10

Leave the two's complement of a number.

MOD

n1 n2 --- mod

ın

Leave the remainder of n1/n2, with the same sign as n1.

NFA

pfa --- nfa

Convert the parameter field address of a definition to its field name.

NOOP

A FORTH no operation. Basically does nothing !

NUMBER

addr --- c

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

OF

n --- f

Used within a colon-definition. The value n is supplied by the user and leaves a flag on the stack for conditional branching.

OFF

Turns the cursor off

OFFSET --- addr

A user variable which may contain a block offset to disc drives. The contents of OFFSET is added to the stack number by BLOCK. Messages by MESSAGE are independent of OFFSET. See BLOCK. MESSAGE.

ON

Turns the cursor on

OR n1 n2 --- or L0

Leave the bit-wise logical or of two 16 bit values.

OUT --- addr

A user variable that contains a value incremented by EMIT. The user may alter and examine OUT to control display formatting.

OVER n1 n2 --- n1 n2 n1 L0

Copy the second stack value, placing it as the new top.

P! b port# --Outputs byte b to port#.So 255 2 P! is equivalent to OUT (2),255

P@ port# --- b
Equivalent to IN (port#).

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

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

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

QUERY

Input 80 characters of text (or until a "return") from the operators terminal. Text is positioned at the address contained in TIB with IN set to zero.

QUIT

Clear the return stack, stop compilation, and return control to the operators terminal. No message is given.

R --- n
Copy the top of the return stack to the computation stack.

R# --- addr

A user variable which may contain the location of an editing cursor, or other file related function.

R/W addr blk f --The fig-FORTH standard disc read-write linkage.addr specifies the source or destination block buffer,blk 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 all error checking.

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

RO

--- addr

Ш

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

REPEAT

addr n --- (compiling) P,C2

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 L0

Rotate the top three values on the stack, bringing the third to the top

RPS

--- addr

Leaves the current value in the return stack register.

RP!

Initializes the return stack pointer from user variable RO.

S->D

n --- d

--- addr

Sign extend a single number to form a double number.

SO

- 11

A user variable that contains the initial value for the stack pointer.Pronounced S-zero. See SP!

SAVET

Will output to tape the contents of the RAM-DISC area. N.B. Never press BRK whilst SAVEing.Always FLUSH before using SAVET.

SCR

--- addi

U

A user variable containing the screen number most recently referenced by LIST.

SIGN

n d --- d

LO

Stores an ascii "-" sign just before 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 #.

SMUDGE

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

SP!

A computer dependant procedure to initialize the stack pointer from SO.

SPO

addr

A computer dependant 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

10

LO

LO,U

Transmit an ascii blank to the output device.

SPACES

Transmit n ascii blanks to the output device.

STATE

addr

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 L0

Exchange the top two values on the stack.

TASK

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

THEN P,CO,LO

An alias for ENDIF.

TIB --- addr U

A user variable containing the address of the terminal input buffer.

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

TRAVERSE addr1 n --- addr2

Move across the name field of a fig-FORTH variable length field. addr1 is the address of either the length byte or the last letter. If n=1, the motion is toward hi 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 --Display on the selected output device the three screens which include that numbered scr,beginning with a screen evenly divisable by three.Output is suitable for source text

records, and includes a reference line at the bottom taken from line 15 of screen 4 .

TYPE addr count --- LO

Transmit count characters from addr to the selected output device.

UK u1 u2 --- f
Will leave a true flag if u1 is less than u2.An unsigned "less-than" operation.

LO

U* u1 u2 --- ud

Leave the unsigned double number product of two unsigned numbers.

U. u --Will print the unsigned 16-bit value at the top of the stack.

U/ ud u1 --- u2 u3

Leave the unsigned remainder u2 and unsigned quotient u3 from the unsigned double dividend ud and unsigned divisor u1.

UNTIL f --- (run-time)

addr n --- (compile) P,C2,L0

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 (OBRANCH) and an offset from HERE to addr.n is used for error tests.

UPDATE

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

USE

--- addr

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

USER

--- LO

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

E.LO

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

L

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 FORGETting through multiple vocabularies.

VOCABULARY

E,L

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.

In fig-FORTH,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.

VLIST

List the names of the definitions in the context vocabulary. "BREAK" will terminate this listing.

WARNING

--- addr

U

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

WHERE

n1 n2 ---

When attempting to LOAD a screen if an ERROR is detected then n1 and n2 are left on the stack.WHERE uses these values to print the line and an indication of the error.It then makes. EDITOR the current vocabulary.

WHILE

f --- (run-time)

ad1 n1 --- ad1 n1 ad2 n2 P,C2

Occurs in a colon-definition in the form:

BEGIN ... WHILE (tp) ... REPEAT

At run-time, WHILE selects conditional execution based on a Boolean flag f.If f is true (non-zero), WHILE continues execution of the true part through to REPEAT, which then branches back to BEGIN.If f is false (zero), execution skips to just after REPEAT, exiting the structure.

At compile-time, WHILE emplaces (OBRANCH) and leaves ad2 of the reserved offset. The stack values will be resolved by REPEAT.

WIDTH

--- addr

U

In fig-FORTH,a user variable containing the maximimum number of letters saved in the compilation of a definition name. It must be between 1 and 31, with a default value of 31. The name character count and its natural characters are saved, up to the value of WIDTH . The value may be changed at any time within the above limits.

WORD

Read the next text characters from the input stream being interpreted,until a delimiter character c 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 c are ignored. If BLK is zero, text is taken from the terminal input buffer, otherwise from the disc block stored in BLK. See BLK, IN.

XOR

n1 n2 --- xor

L1

Leave the bitwise logical exclusive-or of two values.

[

P.L1

Used in a colon-definition in the form:

: xxx [words] more ;

Suspend compilation. The words after [are executed, not compiled. This allows calculation or compilation exceptions before resuming compilation with]. See LITERAL,].

[COMPILE]

P.C

Used in a colon-definition in the form:

: xxx [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 xxx executes, rather than at compile-time.

]

L1

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

MEMOTECH SPECIFIC COMMANDS

All symbols used are the same as in the MEMOTECH manual, reference section, as are the results of the command

ADJSPR

p n v ---

ATTR

p state ---

CLS

COLOUR

p n ---

CRVS

shwyxtn ---

CSR

x y ---

CTLSPR

p x ---

GENPAT

d1 d2 d3 d4 d5 d6 d7 d8 n p ---

GR

x y p --- f

This is an equivalent of the MTX GR\$. The flag will be 1 if the pixel at x y is on, and 0 if off.

INK

colour ---

INKEY

--- C

Will leave the ascii value of any key pressed during the execution of the word. If no key pressed will leave a value of O. N.B. This word has an extremely fast read time.

LINE

x1 y1 x2 y2 ---

MVSPR

p n d ---

PAPER

colour ---

PLOT

x y ---

SOUND

channel freq volume ---

SPK

--- C

Equivalent to SPK\$: Will read the characters at the cursor location, and leave this as an ascii value. So its format is :-

x y CSR SPK.

SPRITE

n pat xp yp xs ys col ---

VIEW

dir dis ---

VS

n ---

MTX FORTH only uses VS 4 and VS 5.All others may be redefined.

SECTION 3. SCREEN EDITOR.

TEXT

c ---

Accept following text to PAD.c is delimiter.

?LINE

n --- addr

Will leave address of line n of current screen.

R#

--- addr

A user variable which contains the offset of the editing cursor from the start of the screen.

#LOCATE

--- n1 n2

From the cursor position leave the line number (n2), and the offset into the line (n1).

#LEAD

--- line-addr offset

#LAG

--- cursor-addr till EOL

-MOVE

addr n ---

Move a line of text from addr to line n of current screen.

н

n ---

Hold numbered line at PAD.

E

n ---

Erase line with n blanks.

S n ---

Spread at line n.Line n and following lines move down one.Line n becomes blank.

D n ---

Delete line n but hold in PAD.

M n ---

Move cursor by a signed amount and then print it. N.B. The cursor is represented by the underline character _.

T n ---

Type line n and save it in PAD

List current screen

REP

Replace line n with text in PAD.

P n ---

Put the following text on line n.

INS n ---

Spread at line n and INSert text from PAD.

TOP

Position cursor at top of current screen.

CLEAR

n ----

Clear current screen. Can be used to select screen n for editing.

FLUSH

See section 1 for definition.

COPY

n1 n2 ---

Copy screen n1 to screen n2.

-TEXT

addr1 count addr2 --- flag

True flag if text at addr1 matches text at addr2.Count is the number of characters to search from starting at addr2.

MATCH

cursor-addr till-EOL str-addr str-count --- tf
cursor-count-till-end-of-matching-text

Match the text at str-addr with all strings on the cursor line forward from the cursor. The values left allow the cursor to be updated either to the end of the matching text or to the start of the next line.

1LINE

--- f

Scan the cursor line for a match to PAD text.f represents boolean flag. If true then the cursor will be placed at text, if false then cursor will be placed at start of next line.

FIND

Search for match of text in PAD ,from the cursor position to the end of the screen. If no match is found then cursor is positioned at top of screen and an error message is given.

DELETE

n ---

Delete n characters prior to the cursor.

N Find next occurance of PAD text.

Input following text to PAD and search for a match from the cursor position to the end of the screen.

B Back up cursor by text in PAD.

X

Delete next occurance of following text.

TILL
Delete on cursor line until end of following text.

C Spread at cursor and copy the following text at the cursor line.

PROGRAMMED BY KEITH JONES. PRODUCED BY SYNTAXSOFT. ..n15

The code for FORTH is located between #4100 and #6100. save program is therefore:-

LD HL, #4100 ;START OF BLOCK. LD DE. #2000 ; LENGTH OF BLOCK XOR A LD (#FD67).A LD (#FD68),A CALL #OAAE RET

N.B. Should you exit from FORTH then type (CIRL) L in order to clear VS 0. You may then list the program, where two lines are contained which will permit a jump to WARM or COLD start. Do not type (RET) straight away as this may overwrite some of the FORTH code.

In the section on Memotech commands, the syntax for the CRVS command has been changed to

CRVS

ntxywhs ---

as per the Memotech manual.

fig-FORTH GLOSSARY

This glossary contains all of the word definitions in Release 1 of fig-FORTH. The definitions are presented in the order of their ascil aort.

The first line of each entry shows a symbolic description of the action of the proceedure on the parameter stack. The symbols indicate the order in which input parameters have been placed on the stack. Three dashes "---" indicate the execution point; any parameters left on the stack are listed. notation, the top of the stack is to the richt.

The symbols include:

nddr menory address 8 bit byte (%.e. hi 8 bits zero) 7 bit sscii character (hi 9 bits zero) ь 32 bit signed double integer, most significant portion with sign d on top of stack. boolean flag. O-false, non-zero-true 11 boolean falme flag-0 16 bit signed integer number 16 bit unsigned integer t f boolean true flag-non-zero

The capital letters on the right show definition characteristics:

- May only be used within a colon definition. A digit indicates number of memory addresses used, if other than one. Intended for execution only.
- LO Level Zero definition of FORTH-78 LI Level One definition of FORTH-78 Has precedence bit set. Will execute
- even when compiling.

A user variable.

Unless otherwise noted, all references to numbers are for 16 bit signed integers. On 8 bit data bus computers, the high byte of a number is on top of the stack, with the sign in the leftmost bit. For 32 bit signed double numbers, the most significant part (with the sign) is on top.

All arithemetic is implicitly 16 bit signed integer math, with error and under-flow indication unspecified.