
OmniFORTH

Installation



INTERACTIVE COMPUTER SYSTEMS, INC.

6403 DIMARCO ROAD • TAMPA, FLORIDA 33614

First Edition 1979

All rights reserved. Reproduction or use, without express permission, of editorial or pictorial content, in any manner is prohibited. No patent liability is assumed with respect to the use of the information contained herein. While every precaution has been taken in the preparation of this manual, the publisher assumes no responsibility for errors or omissions. Neither is any liability assumed for damages from the use of the information contained herein.

© Copyright 1979,
Interactive Computer Systems, Inc.
6403 DiMarco Rd. Tampa, Fl 33614

0 OmniFORTH INSTALLATION

0.1	Master Disk Protection	0-1
0.2	File System	0-1
0.2.1	Master Disk Copy on Multi-Drive Systems . . .	0-2
0.2.2	Master Disk Copy on Single Drive Systems . .	0-2
0.2.3	Copying DOS and Utilities on YOUR FORTH Disk	0-4
0.3	Reconfigure System	0-5
0.3.1	Changing the Backspace Input Character . . .	0-6
0.3.2	Enabling the Control C and ?TERMINAL . . .	0-7
0.3.3	Reconfigure Memory Size (MAP)	0-8
0.3.4	Precompiled User Screens	0-10
0.3.5	Removing Precompiled Words	0-11
0.3.6	OmniFORTH Precompiled Screens	0-11
0.4	Source Listings for OmniFORTH	0-12
0.4.1	Index for OmniFORTH Screens 25 thru 59 . . .	0-13
0.4.2	Listing of Screens 25 thru 59	0-14

0.1 Master Disk Protection

Do not remove the protect tab on the master disk. The master disk should be copied to another work disk before attempting to use OmniFORTH. You may make copies of this software provided that such copies are used only on the computer designated on the Software License Agreement.

Save the original master disk in a safe place and never record or alter it in any way. The master disk will be requested from you when we supply updates and new releases of OmniFORTH. To qualify for updates and revisions, however, requires that the signed Software License Agreement be on file with Interactive Computer Systems, Inc. We must be able to read the master disk and validate your serial number before any new software can be shipped.

0.2 File System

OmniFORTH uses a file system that partitions the disk into areas called screens with 1024 bytes per screen. Screens are physically allocated on the disk starting with screen 0 being the first 1024 bytes on the disk. FORTH does not maintain a disk directory of file names on disk. FORTH computes the track and sector required for disk access as a function of screen number and allows the user to access any part of the disk. The user is given complete control of every byte of disk storage.

OmniFORTH for the North Star supports the following screen numbers:

Density	Drive 1	Drive 2	Drive 3	Drive 4
Single	0 : 86	87:173	174:260	261:347
Double	0 : 174	175:348	349:522	523:696
Quad	0 : 348	349:696	697:1044	1045:1392

Note: FORTH will return to DOS if you attempt to access a screen that is not available. Make sure drive is ready and on-line before accessing multiple drives.

Screens 0 thru 24 have been reserved for you to install the new OmniFORTH object program FORTH, your DOS, and any utilities that you may need such as CF, CD, etc.

Note: We do not supply the North Star DOS or any utilities on the master disk. You will have to copy your version to the working master copy. OmniFORTH will interface with your North Star DOS and you must have DOS on the system drive before you can boot and request FORTH to be executed.

0.2.1 Master Disk Copy on Multi-Drive Systems

Making a backup copy of the master disk is simple for users with more than one disk drive:

- 1) Insert a disk with your North Star DOS and copy disk utility CD in drive 1.
- 2) Place a spare disk in drive 2.
- 3) Initialize spare disk by typing: IN 2
- 4) Start CD (copy disk utility) by entering: GO CD 1 2
- 5) Remove your North Star DOS disk from drive 1 and replace it with the OmniFORTH master disk. Press carriage return to continue the disk copy.
- 6) Remove the master disk and save it in a safe place. The disk on drive 2 is your working copy of OmniFORTH.

0.2.2 Master Disk Copy on Single Drive Systems

Making a backup copy of the master disk takes a little more effort for users with only one disk drive:

- 1) Bring up your North Star DOS using drive 1.
- 2) Remove the DOS disk and replace it with your OmniFORTH master disk.
- 3) List the directory by typing: LI

For single density you should see:

```
##### 4      0 S    0
----- 4      32 S   0
FORTH   36     64 S    1  2000
SCR25-39 100   60 S    0
SCR40-49 160   40 S    0
SCR50-59 200   40 S    0
```

For double density you should see:

```
##### 4      0 D    0
----- 4      28 D    0
FORTH   18     64 D    1  2000
SCR25-39 50    60 D    0
SCR40-49 89    40 D    0
SCR50-59 100   40 D    0
```

The first file shown as ##### is the serial number of your new OmniFORTH disk. Do not alter this serial number in any way as it will be required if you need updates or have questions about your software.

The second file shown as ----- is used as a separator to allocate disk space for you to copy a few of your North Star utilities like CD, CF, etc. This separator also keeps the following files in place if you accidentally compact CD this disk.

FORTH is the executable program that starts at memory location 2D00 hex.

The files SCR25-39, SCR40-49, and SCR50-59 are for the convenience of the single drive user. They are allocated so that they cover screens 25 through 59 which is OmniFORTH source code supplied with this release.

- 4) Remove the OmniFORTH master and place a spare in drive.
- 5) Initialize the spare disk by typing: IN
- 6) Create the files (in the same place) that listed in step 3 and set the file type for FORTH.

For single density:

```
CR FORTH      64 36
TY FORTH      1 2D00
CR SCR25-39   60 100
CR SCR40-49   40 160
CR SCR50-59   40 200
```

For double density:

```
CR FORTH      64 18
TY FORTH      1 2D00
CR SCR25-39   60 50
CR SCR40-49   40 80
CR SCR50-59   40 100
```

- 7) You are now ready to copy the files between the OmniFORTH master and the spare disk using LF (Load File) and SF (Save File) DOS commands. The files to be copied are FORTH, SCR25-39, SCR40-49, and SCR50-59.

For each file to be copied:

- 1) Place master disk in drive and load file into memory using LF.
- 2) Replace with spare and use SF to write out file.

The following is an example of an actual copy:

```
LF FORTH 2D00
SF FORTH 2D00
LF SCR25-39 2D00
SF SCR25-39 2D00
LF SCR40-49 2D00
SF SCR40-49 2D00
LF SCR50-59 2D00
SF SCR50-59 2D00
```

Note that the disks were swapped between each command.

The above procedure only copies areas of the disk with data supplied by ICS, the user will have to expand or alter the procedure to meet his growing needs.

0.2.3 Copying DOS and Utilities on Your FORTH Disk

Note: We do not supply the North Star DOS or any utilities on the master disk. You will have to copy your version to the working master copy. OmniFORTH will interface with your North Star DOS and you must have DOS on the system drive before you can boot and request FORTH to be executed.

It is desirable to have DOS and utilities such as CF, CD, etc. on your OmniFORTH working disk. ICS provides space between the directory and FORTH for this purpose. The file separator ----- has been provided to allocate an area big enough to hold a few North Star utilities.

The CF (copy file utility) will not work for this purpose, since your disk already has files on it and the DOS must start in sector 4.

Use CR (create file) from DOS using the optional start address (see DOS manual) and TY (type file). After the files are created and typed, you may use the CF utility or DOS commands LF and SF to copy the files over.

0.3 Reconfigure System

OmniFORTH provides a table that is used to configure the system at cold start entry (standard entry). An assembly listing of this table is given below.

```

2D00      ;<<<<< NORTH STAR SYSTEM : JUMP TABLE CONSTANTS >>>>>
2D08      DOS      EQU   2028H      ; SOFT DOS RE-ENTRY
2D0D      COUT     EQU   200DH      ; CHAR OUTPUT
2D10      CIN      EQU   2010H      ; CHAR INPUT
2D16      CONTC    EQU   2016H      ; CONTROL 'C' ROUTINE
E800      BOOT     EQU   0E800H     ; HARD DOS RE-ENTRY
2D22      DCOM     EQU   2022H      ; DISK READ/WRITE VIA DOS
2D00      ;
2D00      ;
2D00      ;<<<<< ORIGIN of OmniFORTH >>>>>
2D00      00       ORIG      NOP
2D01      C3253D   JMP      CLD      ; VECTOR TO COLD START
2D04      00       NOP
2D05      C3103D   JMP      WRM      ; VECTOR TO WARM START
2D08      02       DB      OMNIREL   ; OmniFORTH RELEASE #
2D09      03       DB      OMNIREV   ; OmniFORTH VERSION #
2D0A      00       DB      USRVER    ; USER VERSION #
2D0B      0E       DB      0EH       ; IMPLEMENTATION ATTRIBUTES
2D0C      5C46     DW      DOTCPU-7  ; TOPMOST WORD IN FORTH
2D0E      0800     DW      BSIN      ; BACKSPACE CHARACTER
2D10      ;
2D10      ;
2D10      ;<<<<< FOLLOWING USED BY COLD >>>>>
2D10      B87B     DW      INITR0    ; UP      USER POINTER
2D12      187B     DW      INITS0    ; S0      STACK POINTER
2D14      B87B     DW      INITR0    ; R0      RETURN STACK POINTER
2D16      187B     DW      INITS1    ; TIB     TERMINAL INPUT BUFFER
2D18      1F00     DW      1FH       ; WIDTH   MAX WIDTH OF WORDS
2D1A      0100     DW      1         ; WARNING ERROR WARNINGS
2D1C      8146     DW      INITDP    ; FENCE   PROTECTED VOCABULARY
2D1E      8146     DW      INITDP    ; DP      DICTIONARY POINTER
2D20      E63B     DW      FORTH+6   ; VOC-LINK VOCABULARY POINTER
2D22      050020B3 DW      5H,0B320H ; CPU NAME BASE 36 INTEGER
2D26      ;
2D26      ;
2D26      ;<<<<< SPECIAL RECONFIGURATION VALUES >>>>>
2D26      0080     DW      EM        ; INIT 'LIMIT'
2D28      F87B     DW      BUF1      ; INIT 'FIRST'
2D2A      0200     DW      NBUF      ; INIT '#BUFF'
2D2C      0100     DW      1         ; INIT 'DENSITY'
2D2E      3A2D     DW      NEXT      ; ADDRESS OF NEXT
2D30      00       KCIN     DB      0   ; KEYBOARD INPUT DEVICE #
2D31      00       KCOU    DB      0   ; KEYBOARD OUTOUT DEVICE #
2D32      01       KPOU    DB      1   ; PRINTER OUTPUT DEVICE #
2D33      00       ENACTLC DB      0   ; ?TERMINAL ENABLE (0=OFF, 1=ON)
2D34      ;
2D34      ;
NOTE: ?TERMINAL USES NORTH STAR
CONTROL C ENTRY.

```


The average user will be concerned only with a portion of this table. Values may be examined and/or changed using the North Star monitor.

OmniFORTH provides methods to change the configuration constants. The remainder of this section will discuss the use of OmniFORTH in changing:

- 1) Input character used for backspacing.
- 2) Enabling Control C and ?TERMINAL.
- 3) Maximum memory used by OmniFORTH.
- 4) Precompiling additional definitions into the FORTH object file.

All values used in this section are hexadecimal numbers.

This section is intended for the advanced FORTH user especially, since it should not be necessary for the new user to reconfigure except for the input backspace character which could be changed using the North Star monitor.

A very important OmniFORTH word used in this section is +ORIGIN. +ORIGIN adds an offset equivalent to the beginning of OmniFORTH to the top of the stack leaving the resulting memory address.

0.3.1 Changing the Backspace Input Character.

The procedure is to bring up OmniFORTH, make the change at address 0E +ORIGIN, exit OmniFORTH and return to DOS then save the object back on disk using the DOS SF command.

For example, to change the backspace code (08 HEX) to the rubout code (7F HEX).

```
GO FORTH
```

```
OMNIFORTH 2.3 (BASED ON FIG-FORTH)
```

```
COPYWRITE 1979 BY  
INTERACTIVE COMPUTER SYSTEMS, INC.  
6403 DIMARCO RD. TAMPA, FL. 33614
```

```
ALL RIGHTS RESERVED. S/N #####
```

```
HEX OK
```

```
7F 0E +ORIGIN ! BYE +SF FORTH 2D00
```

The input backspace character has now been changed from backspace code (08 HEX) to the rubout code (7F HEX).

0.3.2 Enabling the Control C and ?TERMINAL

OmniFORTH is released with the control-C option disabled so that terminals that do not support this mode can function. Users that have altered the control-C routines will need to generate code that will provide the equivalent status of the Zero flag specified in the North Star DOS manual section G:

CONTC

This routine detects if a control-C has been typed on the console terminal. No information is passed to the routine in any of the registers, and no registers need be saved or restored by CONTC. They are all available for unrestricted use by the routine. If a control-C has been typed, the routine should set the Zero flag. If no character has been typed or if the character typed was not a control-C, then the Zero flag should be cleared instead. As soon as the Zero flag is given its proper value, CONTC must do a RET. CONTC should not wait for a character to be typed. If no character has been typed, it should do a RET immediately after clearing the Zero flag.

The North Star CONTC routine also sets the carry flag if any character has been typed.

?TERMINAL

This OmniFORTH word uses the North Star CONTC routine to sense if any key depression has occurred by testing the carry flag after each call to CONTC. If no key has been depressed, ?TERMINAL will put a zero on top of the stack. If any key has been depressed, ?TERMINAL will put a one on the stack. ?TERMINAL is supplied disabled on this release of OmniFORTH. You can enable it by changing the enable control-C reconfiguration value shown for ENACTLC on the assembly listing at location 2D33 hex.

To enable the control-C and ?TERMINAL

GO FORTH

OMNIFORTH 2.3 (BASED ON FIG-FORTH)

COPYWRITE 1979 BY
INTERACTIVE COMPUTER SYSTEMS, INC.
6403 DIMARCO RD. TAMPA, FL. 33614

ALL RIGHTS RESERVED. S/N #####

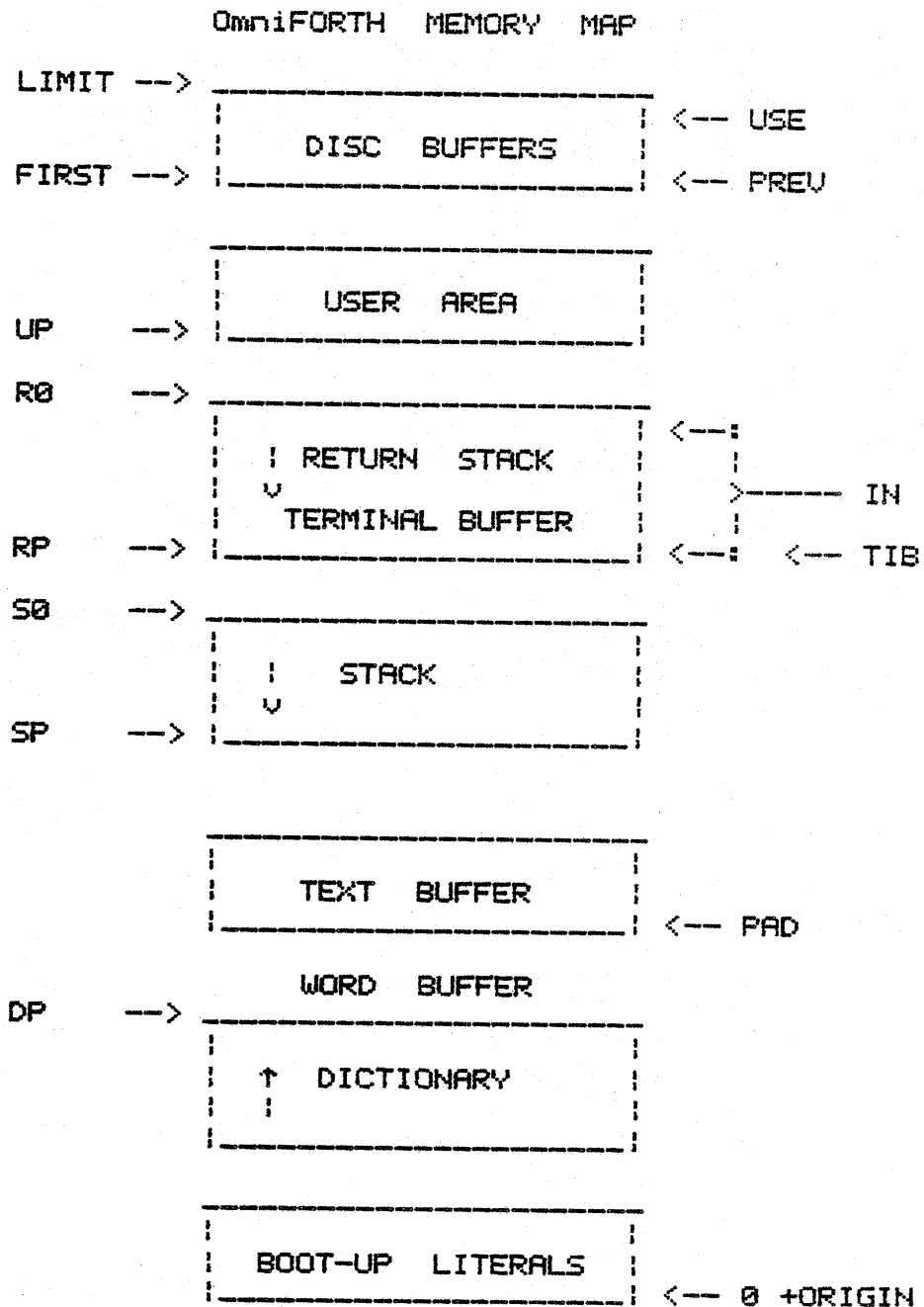
HEX OK

1 33 +ORIGIN C! BYE +SF FORTH 2D00

0.3.3 Reconfigure Memory Size (MAP)

OmniFORTH comes configured to run in 24k bytes of memory including DOS. This provides more than enough available work space for the average user.

The following is a diagram of the OmniFORTH memory map:



OmniFORTH starts at memory address 2000 for either the single or double density North Star DOS versions.

Remember that the OmniFORTH supplied on the master disk allocates 16k bytes of disk storage. When FORTH was saved on disk, a file was created allocating 64 (256 byte) blocks which allows enough space for 16k bytes for FORTH and most user applications. The user can extend the supplied FORTH vocabulary with applications beyond 16k bytes as needed. If your application requires more space, be sure to create a disk file larger than the 64 blocks to hold FORTH and save your version using screen 46 and DOS SF command.

Screens 25 thru 59 of the OmniFORTH master disk contain source for part of the FORTH vocabulary included in this release. A listing of screens 25 thru 59 is included at the end of this installation manual to assist you in making changes and learning more about OmniFORTH.

Screen 25 and 26 contain the algorithm to configure memory requirements from LIMIT (top of memory+1) down to 50 (top of the stack).

Note: The literals on the screen are the values used to configure your 24k system.

Most users will only wish to change the memory size which can be changed by changing the 8000 in line 5 of screen 25.

Line 7 of screen 25 contains a 2 which configures two 512 byte disk buffers. This value must be equal to 2 or greater.

The following is a procedure to reconfigure memory:

- 1) Bring up a copy of OmniFORTH.
- 2) Execute OmniFORTH by typing: GO FORTH
- 3) Use the EDITOR to adjust the configuration literals on screens 25 and 26.
- 4) Load screens 25 and 26 by typing: 25 Load (26 will load automatically). The present map has not been changed but the BOOT-UP literals (See OmniFORTH map) have been changed.
- 5) Return to North Star DOS by typing: BYE
- 6) Using DOS, save reconfigured OmniFORTH back on disk by typing: SF FORTH 2000
- 7) Bring up reconfigured OmniFORTH by entering: GO FORTH

0.3.4 Precompiled User Screens

Precompiling is the ability to load user screens (OmniFORTH source) and save the expanded OmniFORTH module so it will be available the next time that you bring up FORTH.

An OmniFORTH user will develop new words which he will want to always have available. The user can load the appropriate screens each time they bring up OmniFORTH, which can become very time consuming, or you may precompile the screens once and save them as object code within FORTH.

Procedure to Precompile

- 1) Bring up OmniFORTH by typing: GO FORTH
- 2) Load your new words from your screens.

WARNING: OmniFORTH was saved on disk in an area allocated for 16k bytes. FORTH disk file was created with 64 blocks of 256 bytes per block giving 16k bytes of user program storage. It is possible for a user to create new vocabularies that extend the dictionary beyond the 16k byte storage allocated for FORTH on disk. To determine the size of the current memory and storage necessary to save FORTH type:

```
HERE 0 +ORIGIN - ..
```

If current memory requirement is greater than 16k bytes you will have to go back to DOS and create a file that will be large enough to save your new FORTH.

- 3) Load screen 46 (Precompiler) by typing: 46 LOAD
- 4) Return to DOS by typing: BYE
- 5) Save FORTH using DOS command: SF FORTH 2000

The next time you request FORTH it will come up with any new words that you have loaded.

0.3.5 Removing Precompiled Words

Part or all of the users precompiled vocabulary may be removed from the FORTH object file by:

- 1) Dropping the memory protect fence by typing: 0 FENCE !
- 2) Forgetting back to and including the word you want to remove by entering:

FORTH FORGET word you want to forget

- 3) Load screen 46 (Precompiler) by typing: 46 LOAD
- 4) Return to DOS by typing: BYE
- 5) Save FORTH using DOS command: SF FORTH 2000

0.3.6 OmniFORTH Precompiled Screens

We have precompiled most of the source screens supplied on the OmniFORTH disk and have saved them as part of FORTH. These include screens for the EDITOR, DUMP, special formatting, TYPE, LIST, TRIAD, SHOW, and INDEX that are supplied on screens 29 thru 44.

Screen 29 has been created as a user load screen and will load all of the standard OmniFORTH software. Note that screen 29 contains a USER-MARK and TASK dummy words which mark the lower and upper limits of the OmniFORTH vocabulary. The user can alter words supplied in screens 29 thru 44 as well as create new screens to extend the vocabulary using the EDITOR.

The user can then forget a part or all of the OmniFORTH vocabulary and reload words that suits his application. For example, the standard OmniFORTH is supplied with a special word CEMIT on screen 30 that maps any character (even control or non printable) to be printed into upper case ASCII only. If your console can print lower case characters you may want to change CEMIT to translate any character to either upper or lower case. To do this requires that you use the EDITOR to change the mask constant from 95 to 127. Then dropping the memory protect fence, forgetting the OmniFORTH vocabulary, reloading screen 29, and saving the new precompiled version by loading screen 46 and returning to DOS to save FORTH.

To change CEMIT to print both upper and lower case ASCII:

1) Use the EDITOR to change CEMIT on screen 30

```
DECIMAL EDITOR 30 LIST
```

```
TOP X 95
```

```
C 127
```

```
FLUSH
```

2) Drop the protected memory fence: 0 FENCE !

3) Forget lower vocabulary: FORTH FORGET USER-MARK

4) Reload user screens: 29 LOAD

5) Load the precompiler screen: 46 LOAD

6) Return to DOS: BYE

7) In DOS save file FORTH: SF FORTH 2000

0.4 Source Listings for OmniFORTH

Screens 25 thru 59 of the OmniFORTH master disk contain source for part of the FORTH vocabulary included in this release. A listing of screens 25 thru 59 is included on the following pages of this installation manual to assist you in making changes and learning more about OmniFORTH.

Screens 25 and 26 contain the source to reconfigure memory requirements to suit your system.

Screens 27 and 28 hold error messages that are referenced by OmniFORTH. You may extend the messages for your application.

Screen 29 is the user source screen that was used to load OmniFORTH sources supplied in this release. You may alter this load screen to include additional definitions and screens as you require. Screen 29 is released to load special console routines, utilities, and the EDITOR.

Screens 36 thru 44 contain the source for the text EDITOR.

Screen 46 is the precompiler load screen that enables saving your application as part of FORTH.

Screens 48 thru 51 are optional Z80 assembler instructions that may be loaded if needed. These screens are not loaded with this release of OmniFORTH.

Screens 57 thru 59 are supplied as example and work areas to test and learn more about the EDITOR and OmniFORTH.

INTERACTIVE COMPUTER SYSTEMS, INC. OMNIFORTH 2.3 INSTALLATION

- 25 < MEMORY MAP RECONFIGURATION SCREEN 1 OF 2 > HEX
- 26 < MEMORY MAP RECONFIGURATION SCREEN 2 OF 2 > HEX
- 27 < ERROR MESSAGES >
- 28 < ERROR MESSAGES >
- 29 < USER SOURCE LOAD SCREEN > DECIMAL
- 30 < QUIT, CEMIT, TYPE, LIST, TRIAD, SHOW, INDEX > DECIMAL
- 31 < SPECIAL FORMATTING: #D, #H, .2H, .4H, .. > HEX
- 32 < UTILITIES: DUMP, ?S, MODE > HEX
- 33
- 34
- 35
- 36 < TEXT, LINE >
- 37 < LINE EDITOR >
- 38 < LINE EDITING COMMANDS >
- 39 < LINE EDITING COMMANDS >
- 40 < LINE EDITING COMMANDS >
- 41 < SCREEN EDITING COMMANDS >
- 42 < STRING EDITING COMMANDS > HEX
- 43 < STRING EDITING COMMANDS > HEX
- 44 < STRING EDITING COMMANDS > HEX
- 45
- 46 < PRE-COMPILER > DECIMAL
- 47
- 48 < Z80 ASSEMBLER -- OPTIONAL INSTRUCTION SET > HEX
- 49 < Z80 ASSEMBLER -- EXCHANGE, BLOCK TRANSFER, SEARCH GROUP > HEX
- 50 < Z80 ASSEMBLER -- ROTATE, SHIFT, & BIT GROUPS > HEX
- 51 < Z80 ASSEMBLER -- JMP OPERATORS > HEX
- 52
- 53
- 54
- 55
- 56
- 57 < EXAMPLE EDIT SCREEN. USE SCREENS 57, 58, AND 59 TO TEST >
- 58 < QUIT, CEMIT, TYPE, LIST, TRIAD, SHOW, INDEX > DECIMAL
- 59 < USE THIS SCREEN TO HOLD A WORK COPY >

INDEX OF SCREENS SUPPLIED WITH OmniFORTH 2.3 INSTALLATION

SCR # 24

0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15

SCR # 25

0 < MEMORY MAP RECONFIGURATION SCREEN 1 OF 2 > HEX
1 < BY ADJUSTING THE LITERALS AND LOADING, THEN SAVING THE >
2 < OBJECT ON DISK THE USER CAN CHANGE MEMORY UTILIZATION. >
3 < *** REFER TO DOCUMENTATION AND USE EXTREME CAUTION. *** >
4
5 8000 DUP 26 +ORIGIN ! < "LIMIT" TOP OF MEMORY + 1 *)
6
7 2 DUP 2A +ORIGIN ! < "#BUFF" # OF DISK BUFFERS *)
8
9 200 < BYTES/BUFFER -- FIXED FOR NS SYSTEMS *)
10 4 + * - DUP 28 +ORIGIN ! < "FIRST" ADDR OF 1ST BUFFER *)
11
12 -->
13
14
15

SCR # 26

0 < MEMORY MAP RECONFIGURATION SCREEN 2 OF 2 > HEX
1
2 40 < USER VARIABLE SPACE *)
3 - DUP 10 +ORIGIN ! < "UP" START OF USER AREA *)
4
5 DUP 14 +ORIGIN ! < "R0" ALSO TOP OF RETURN STACK *)
6
7 00A0 < RETURN STACK & TERMINAL BUFFER SPACE *)
8 - DUP 12 +ORIGIN ! < "S0" TOP OF PARAMETER STACK *)
9
10 16 +ORIGIN ! < "TIB" -- TERMINAL INPUT BUFFER *)
11
12 DECIMAL ;S
13
14
15

SCR # 27

- 0 < ERROR MESSAGES >
- 1 EMPTY STACK
- 2 DICTIONARY FULL
- 3 HAS INCORRECT ADDRESS MODE
- 4 ISN'T UNIQUE
- 5
- 6 DISC RANGE ?
- 7 FULL STACK
- 8 DISC ERROR !

9
10
11
12
13
14

15 INTERACTIVE COMPUTER SYSTEMS, INC. OMNIFORTH 2.3 INSTALLATION

SCR # 28

- 0 < ERROR MESSAGES >
- 1 COMPILATION ONLY, USE IN DEFINITION
- 2 EXECUTION ONLY
- 3 CONDITIONALS NOT PAIRED
- 4 DEFINITION NOT FINISHED
- 5 IN PROTECTED DICTIONARY
- 6 USE ONLY WHEN LOADING
- 7 OFF CURRENT EDITING SCREEN
- 8 DECLARE VOCABULARY

9
10
11
12
13
14
15

SCR # 29

- 0 < USER SOURCE LOAD SCREEN > DECIMAL
- 1
- 2 : USER-MARK ; < DUMMY WORD USED TO MARK USER LOADS * >
- 3
- 4 30 LOAD < SPECIAL: TYPE, LIST, TRIAD, SHOW, INDEX >
- 5 31 LOAD < SPECIAL: FORMATTING >
- 6 32 LOAD < UTILITY: DUMP, ?S, MODE >
- 7 36 LOAD < EDITOR >

8
9 : TASK ; < TO FORGET PRESENT TASK, DROP FENCE FIRST * >
10
11 DECIMAL ;S
12
13
14
15

INTERACTIVE COMPUTER SYSTEMS, INC. OMNIFORTH 2.3 INSTALLATION

```

SCR # 30
0 < QUIT, CEMIT, TYPE, LIST, TRIAD, SHOW, INDEX > DECIMAL
1 : QUIT          < TEST TERMINAL INTERRUPT, IF CONTROL C QUIT *)
2   ?TERMINAL IF KEY 3 = IF SP! QUIT ENDIF ENDIF ; CR
3 : CEMIT        < MAP CHAR TO EMIT, USE 95 FOR UPPER, 127 FOR BOTH *)
4   95 AND DUP 32 < IF 32 + ENDIF EMIT ;
5 : TYPE        QUIT -DUP IF OVER + SWAP          < TYPE UNTIL CONTROL C *)
6   DO I C@ CEMIT LOOP ELSE DROP ENDIF ;
7 : .LINE       <LINE> -TRAILING TYPE ;
8 : LIST        CR DUP SCR ! ." SCR # " .          < LIST UNTIL CONTROL C *)
9   16 0 DO CR I 3 .R SPACE I SCR @ .LINE LOOP CR ;
10 : .FF        12 EMIT ; CR                       < EMIT FORM FEED *)
11 : .M15       CR 15 MESSAGE CR ;                 < TYPE MESSAGE 15 *)
12 : TRIAD      .FF 3 / 3 * 3 OVER + SWAP DO CR I LIST LOOP .M15 ;
13 : SHOW       1+ SWAP 3 / 3 * DO I TRIAD 3 +LOOP ;
14 : INDEX      1+ 55 SWAP ROT DO 1+ DUP 55 > IF DROP 0 .FF .M15 ENDIF
15             CR I 3 .R SPACE 0 I .LINE LOOP DROP .FF ; CR

```

```

SCR # 31
0 < SPECIAL FORMATTING: #D, #H, .2H, .4H, .. > HEX
1 : #D          < CONVERT ONE DECIMAL DIGIT, HOLDING IN PAD *)
2   0A M/MOD ROT 9 OVER < IF 7 + ENDIF 30 + HOLD ;
3 : #H          < CONVERT ONE HEX DIGIT, HOLDING IN PAD *)
4   10 M/MOD ROT 9 OVER < IF 7 + ENDIF 30 + HOLD ;
5 : .2H 0 <# #H #H #> TYPE SPACE ;                < 2 DIGIT HEX OUTPUT *)
6 : .4H 0 <# #H #H #H #H #> TYPE SPACE ;          < 4 DIGIT HEX OUTPUT *)
7 : .. 0 <# #S #> TYPE SPACE ;                    < UNSIGNED EQUIV OF . *)
8 DECIMAL ;S
9
10
11
12
13
14
15

```

```

SCR # 32
0 < UTILITIES: DUMP, ?S, MODE > HEX
1 : DUMP CR     < MEMORY HEX/ASCII DUMP ROUTINE, ADDR-2, COUNT-1 *)
2   ." BYTE->  0 1 2 3 4 5 6 7   8 9 A B C D E F"
3   ."      0123456789ABCDEF"          0 DO CR DUP .4H
4   10 0 DO I 7 AND 0= 2 * SPACES DUP C@ .2H 1+ LOOP 2 SPACES
5   10 - 10 0 DO DUP C@ CEMIT 1+ LOOP      10 +LOOP DROP CR ;
6 : ?S         < NON-DESTRUCTIVE PRINT OUT OF STACK *)
7   S0 @ SP@ - 2 - 2 / -DUP CR          IF ." STACK  HEX  DEC"
8   0 DO CR SP@ I 2 * + @ ." S" I 1+ 0 <# #D #D #> TYPE ." = "
9   DUP .4H SPACE BASE @ DECIMAL SWAP 6 .R BASE ! LOOP CR
10  ELSE ." NOTHING ON STACK " CR ENDIF ;
11 DECIMAL
12 : MODE      < PRINTS THE NUMBER BASE *)
13  BASE @ 10 - IF BASE @ 16 - IF ." BASE<10> = " BASE @
14  0 <# #D #D #D #> TYPE SPACE ELSE ." HEX " ENDIF
15  ELSE ." DECIMAL " ENDIF ;

```

SCR # 33

0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15

SCR # 34

0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15

SCR # 35

0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15

```

SCR # 36
0 < TEXT, LINE >
1 FORTH DEFINITIONS HEX
2 : TEXT                                < ACCEPT FOLLOWING TEXT TO PAD * >
3   HERE C/L 1+ BLANKS WORD HERE PAD C/L 1+ CMOVE ;
4
5 : LINE                                < RELATIVE TO SCR, LEAVE ADDRESS OF LINE * >
6   DUP FFF0 AND 17 ?ERROR < KEEP ON THIS SCREEN >
7   SCR @ <LINE> DROP ;
8 -->
9
10
11
12
13
14
15

```

```

SCR # 37
0 < LINE EDITOR >
1 VOCABULARY EDITOR IMMEDIATE HEX
2 : WHERE                                < PRINT SCREEN # AND IMAGE OF ERROR * >
3   DUP B/SCR / DUP SCR ! ." SCR # " DECIMAL .
4   SWAP C/L /MOD C/L * ROT BLOCK + CR C/L TYPE
5   CR HERE C@ - SPACES 5E EMIT [COMPILE] EDITOR QUIT ;
6
7 EDITOR DEFINITIONS
8 : #LOCATE                              < LEAVE CURSOR OFFSET-2, LINE-1 * >
9   R# @ C/L /MOD ;
10 : #LEAD                                < LINE ADDRESS-2, OFFSET-1, LINE-1 * >
11   #LOCATE LINE SWAP ;
12 : #LAG                                 < CURSOR ADDRESS-2, COUNT-1 AFTER CURSOR * >
13   #LEAD DUP >R + C/L R> - ;
14 : -MOVE                                < MOVE IN BLOCK BUFFER ADDR FORM-2, LINE TO-1 * >
15   LINE C/L CMOVE UPDATE ; -->

```

```

SCR # 38
0 < LINE EDITING COMMANDS >
1 : H                                    < HOLD NUMBERED LINE AT PAD * >
2   LINE PAD 1+ C/L DUP PAD C! CMOVE ;
3
4 : E                                    < ERASE LINE-1 WITH BLANKS * >
5   LINE C/L BLANKS UPDATE ;
6
7 : S                                    < SPREAD MAKING LINE # BLANK * >
8   DUP 1 - < LIMIT > 0E < FIRST TO MOVE >
9   DO I LINE I 1+ -MOVE -1 +LOOP E ;
10
11 : D                                    < DELETE LINE-1, BUT HOLD IN PAD * >
12   DUP H 0F DUP ROT
13   DO I 1+ LINE I -MOVE LOOP E ;
14
15 -->

```

```

SCR # 39
0 ( LINE EDITING COMMANDS )
1
2 : M      ( MOVE CURSOR BY SIGNED AMOUNT-1, PRINT ITS LINE *)
3   R# +! CR SPACE #LEAD TYPE 5F EMIT
4   #LAG TYPE #LOCATE . DROP ;
5
6 : T      ( TYPE LINE BY #-1, SAVE ALSO IN PAD *)
7   DUP C/L * R# ! DUP H 0 M ;
8
9 : L      ( RE-LIST SCREEN *)
10  SCR @ LIST 0 M ;
11 -->
12
13
14
15

```

```

SCR # 40
0 ( LINE EDITING COMMANDS )
1 : R      ( REPLACE ON LINE#-1, FROM PAD *)
2   PAD 1+ SWAP -MOVE ;
3
4 : P      ( PUT FOLLOWING TEXT ON LINE-1 *)
5   1 TEXT R ;
6
7 : I      ( INSERT TEXT FROM PAD ONTO LINE # *)
8   DUP S R ;
9
10 : TOP   ( HOME CURSOR TO TOP LEFT OF SCREEN *)
11   0 R# ! ;
12 -->
13
14
15

```

```

SCR # 41
0 ( SCREEN EDITING COMMANDS )
1 : CLEAR      ( CLEAR SCREEN BY NUMBER-1 *)
2   SCR ! 10 0 DO FORTH I EDITOR E LOOP ;
3
4 : COPY      ( DUPLICATE SCREEN-2, ONTO SCREEN-1 *)
5   B/SCR * OFFSET @ + SWAP B/SCR * B/SCR OVER + SWAP
6   DO DUP FORTH I BLOCK 2 - ! 1+ UPDATE LOOP
7   DROP FLUSH ;
8 -->
9
10
11
12
13
14
15

```

```

SCR # 42
0 ( STRING EDITING COMMANDS ) HEX
1 : 1LINE      ( SCAN LINE WITH CURSOR FOR MATCH TO PAD TEXT, *)
2              ( UPDATE CURSOR, RETURN BOOLEAN *)
3      #LAG PAD COUNT MATCH R# +! ;
4
5 : FIND      ( STRING AT PAD OVER FULL SCREEN RANGE, ELSE ERROR *)
6      BEGIN 3FF R# @ <
7      IF TOP PAD HERE C/L 1+ CMOVE @ ERROR ENDIF
8      1LINE UNTIL ;
9
10 : DELETE   ( BACKWARDS AT CURSOR BY COUNT-1 *)
11      >R #LAG + FORTH R - ( SAVE BLANK FILL LOCATION )
12      #LAG R MINUS R# +! ( BACKUP CURSOR )
13      #LEAD + SWAP CMOVE
14      R> BLANKS UPDATE ; ( FILL FROM END OF TEXT )
15 DECIMAL -->

```

```

SCR # 43
0 ( STRING EDITING COMMANDS ) HEX
1 : N          ( FIND NEXT OCCURANCE OF PREVIOUS TEXT *)
2      FIND @ M ;
3
4 : F          ( FIND OCCURANCE OF FOLLOWING TEXT *)
5      1 TEXT N ;
6
7 : B          ( BACKUP CURSOR BY TEXT IN PAD *)
8      PAD C@ MINUS M ;
9
10 : X         ( DELETE FOLLOWING TEXT *)
11      1 TEXT FIND PAD C@ DELETE @ M ;
12
13 : TILL      ( DELETE ON CURSOR LINE, FROM CURSOR TO TEXT END *)
14      #LEAD + 1 TEXT 1LINE @ = @ ?ERROR
15      #LEAD + SWAP - DELETE @ M ; DECIMAL -->

```

```

SCR # 44
0 ( STRING EDITING COMMANDS ) HEX
1 : C          ( SPREAD AT CURSOR AND COPY IN THE FOLLOWING TEXT *)
2      1 TEXT PAD COUNT
3      #LAG ROT OVER MIN >R
4      FORTH R R# +! ( BUMP CURSOR )
5      R - >R      ( CHARS TO SAVE )
6      DUP HERE R CMOVE ( FROM OLD CURSOR TO HERE )
7      HERE #LEAD + R> CMOVE ( HERE TO CURSOR LOCATION )
8      R> CMOVE UPDATE ( PAD TO OLD CURSOR )
9      @ M ( LOOK AT NEW LINE ) ;
10 TOP FORTH DEFINITIONS DECIMAL FORTH
11
12 ( EDITOR 6 + 32 +ORIGIN ! ( VOC-LINK )
13
14 ;S
15

```

SCR # 45

0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15

SCR # 46

0 < PRE-COMPILER > DECIMAL
1 < THIS SCREEN IS USED TO PRECOMPILE OR SAVE LOADED >
2 < SCREENS SO THAT AT THE NEXT COLD START THE SCREENS >
3 < WILL ALREADY BE LOADED : >
4 < 1. LOAD ALL DESIRED SCREENS >
5 < 2. LOAD THIS SCREEN >
6 < 3. TYPE BYE >
7 < 4. IN DOS TYPE -- SF FORTH 2000 >
8
9 LATEST 12 +ORIGIN ! < TOP NFA >
10 HERE 28 +ORIGIN ! < FENCE >
11 HERE 30 +ORIGIN ! < DP >
12 HERE FENCE !
13 ;S
14
15

SCR # 47

0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15

SCR # 48

```

0 ( Z80 ASSEMBLER -- OPTIONAL INSTRUCTION SET ) HEX
1 ASSEMBLER DEFINITIONS
2
3 3 CONSTANT E      2 CONSTANT D      1 CONSTANT C      0 CONSTANT B
4
5 047 CONSTANT IV   04F CONSTANT RF
6
7 ( 16 BIT LOAD GROUP )
8 4BED XDP LBCD     5BED XDP LDED
9 2ADD XDP LIXD     21FD XDP LIYD     7BED XDP LSPD
10 43ED XDP SBCD     53ED XDP SDED
11 22DD XDP SIXD     22FD XDP SIYD     73ED XDP SSPD
12 F9DD DOP SPIX     F9FD DOP SPIY
13
14 DECIMAL -->
15

```

SCR # 49

```

0 ( Z80 ASSEMBLER -- EXCHANGE, BLOCK TRANSFER, SEARCH GROUP ) HEX
1
2 ( EXCHANGE )
3 0008 OPC EX      00D9 OPC EXX
4 E3DD DOP XTIX     E3FD DOP XTIY
5 ( BLOCK TRANSFER )
6 A0ED DOP LDI      A8ED DOP LDD
7 ( SEARCH )
8 A9ED DOP CPD      B9ED DOP CPDR     A1ED DOP CPII
9
10 ( 8 BIT ARITH, LOCIC IMMEDIATE )
11 C6 ALI ADI      CE ALI ACI      D6 ALI SUI      DE ALI SBI
12 E6 ALI ANI      EE ALI XRI      F6 ALI ORI      FE ALI CPI
13
14 DECIMAL -->
15

```

SCR # 50

```

0 ( Z80 ASSEMBLER -- ROTATE, SHIFT, & BIT GROUPS ) HEX
1
2 0000 ROP SLC      0010 ROP RL      0008 ROP SRC      0018 ROP RR
3 0020 ROP SLA      0028 ROP SRA      0038 ROP SRL
4 6FED DOP RLD      67ED DOP RRD
5
6 0040 BOP BIT      0080 BOP RES      00C0 BOP SET
7
8 ( INPUT/OUTPUT INSTRUCTIONS )
9
10
11 A2ED DOP INI      B2ED DOP INIR     AAED DOP IND      BAED DOP INDR
12
13 A3ED DOP OUTI     B3ED DOP OUTIR     ABED DOP OUTD     BBED DOP OUTDR
14 DECIMAL -->
15

```

SCR # 51

0 (Z80 ASSEMBLER -- JMP OPERATORS) HEX

1

2 00D2 ADR JNC 00DA ADR JC 00C2 ADR JNZ 00CA ADR JZ

3 00E2 ADR JPO 00EA ADR JPE 00F2 ADR JF 00FA ADR JM

4 0018 ALI JR 0038 ALI JRC 0030 ALI JRNC 0028 ALI JRZ

5 0020 ALI JRNZ

6 E9DD DOP PCIY E9FD DOP PCIY 0010 ALI DJNZ

7

8 (CALL AND RETURN INSTRUCTIONS)

9 00D4 ADR CNC 00DC ADR CC 00C4 ADR CNZ 00CC ADR CZ

10 00EC ADR CPE 00E4 ADR CPO 00F4 ADR CP 00FC ADR CM

11 00C9 OPC RET 00D8 OPC RNC 00D8 OPC RC 00C8 OPC RNZ

12 00C8 OPC RZ 00E8 OPC RPE 00E8 OPC RPO 00F8 OPC RP

13 00F8 OPC RM

14

15 FORTH DEFINITIONS DECIMAL ;S

SCR # 52

0

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

SCR # 53

0

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

SCR # 54

0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15

SCR # 55

0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15

SCR # 56

0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15

SCR # 57

```
0 < EXAMPLE EDIT SCREEN. USE SCREENS 57,58, AND 59 TO TEST >
1
2 : TAX 5 * 100 / . ; < EXAMPLE 1.4.1 >
3
4 : YES/NO IF ." YES" ELSE ." NO" THEN ; < EXAMPLE 1.5.3 >
5
6 : MIN OVER OVER > IF SWAP ENDIF DROP ; < EXAMPLE 1.5.4 >
7
8 : ABC 10 0 DO I . LOOP ; < EXAMPLE 1.5.7 >
9
10 : XYZ 10 0 DO I . 2 +LOOP ; < EXAMPLE 1.5.7 >
11
12 : DUMP CR 1+ SWAP DO I C@ . LOOP ; < EXAMPLE 1.5.8 >
13
14 ;S
15
```

SCR # 58

```
0 < CQUIT, CEMIT, TYPE, LIST, TRIAD, SHOW, INDEX > DECIMAL
1 : CQUIT < TEST TERMINAL INTERRUPT, IF CONTROL C QUIT * >
2 ?TERMINAL IF KEY 3 = IF SP! QUIT ENDIF ENDIF ; CR
3 : CEMIT < MAP CHAR TO EMIT. USE 95 FOR UPPER, 127 FOR BOTH * >
4 95 AND DUP 32 < IF 32 + ENDIF EMIT ;
5 : TYPE CQUIT -DUP IF OVER + SWAP < TYPE UNTIL CONTROL C * >
6 DO I C@ CEMIT LOOP ELSE DROP ENDIF ;
7 : .LINE <LINE> -TRAILING TYPE ;
8 : LIST CR DUP SCR ! ." SCR # " . < LIST UNTIL CONTROL C * >
9 16 0 DO CR I 3 .R SPACE I SCR @ .LINE LOOP CR ;
10 : .FF 12 EMIT ; CR < EMIT FORM FEED * >
11 : .M15 CR 15 MESSAGE CR ; < TYPE MESSAGE 15 * >
12 : TRIAD .FF 3 / 3 * 3 OVER + SWAP DO CR I LIST LOOP .M15 ;
13 : SHOW 1+ SWAP 3 / 3 * DO I TRIAD 3 +LOOP ;
14 : INDEX 1+ 55 SWAP ROT DO 1+ DUP 55 > IF DROP 0 .FF .M15 ENDIF
15 CR I 3 .R SPACE 0 I .LINE LOOP DROP .FF ; CR
```

SCR # 59

```
0 < USE THIS SCREEN TO HOLD A WORK COPY >
1
2 < WHILE YOU LEARN MORE ABOUT THE EDITOR >
3
4
5
6 < COPY THE SCREEN YOU WOULD LIKE TO EDIT INTO THIS SCREEN >
7
8 < THEN USE THE EDITOR TO MAKE CHANGES.... GOOD LUCK >
9
10
11
12
13
14
15
```

```

SCR # 60
0 < CD COPY DISK EXAMPLE USING RWDSK. FROM TO CD >
1 HEX D000 CONSTANT TOP < TOP OF MEMORY ABOVE LIMIT >
2 DECIMAL 350 CONSTANT LAST < LAST SECTOR: QUAD USE 700 >
3 @ VARIABLE DR @ VARIABLE DW @ VARIABLE #SEC
4
5 : NEXT-SEC #SEC @ SEC +! LAST SEC @ #SEC @ + -
6 DUP @< IF #SEC +! ELSE DROP THEN #SEC @ 0= ;
7 : WRITE LIMIT SEC @ #SEC @ 0 DW @ RWDSK DISK-ERROR ! ;
8 : READ LIMIT SEC @ #SEC @ 1 DR @ RWDSK DISK-ERROR ! ;
9 : INIT-SEC @ SEC ! TOP LIMIT - 256 DENSITY @ 1+ * / #SEC ! ;
10 : COPY INIT-SEC BEGIN READ WRITE NEXT-SEC UNTIL ." DONE " ;
11 : D&D DUP 128 DENSITY @ * + ; < COMPUTE DENSITY & DRIVE >
12
13 : CD < FROM-DRIVE TO-DRIVE CD COPY DISK USING RWDSK >
14 SWAP D&D DR ! ." COPY DISK FROM " D&D DW ! ." TO " CR
15 ." RETURN TO CONTINUE " KEY 13 = IF COPY ELSE ." OMIT " THEN ;

```

```

SCR # 61
0 < NORTH STAR DOS HARD DISK ERROR PATCH FOR OMNIFORTH > HEX
1
2 : HDERR CR ." HARD DISK ERROR" ABORT ;
3
4 ASSEMBLER HERE < SAVE ADDRESS OF ENTRY POINT ON STACK >
5 HDERR I LXI NEXT JMP < CODE TO ENTER FORTH HDERR >
6 201A ! < POKE A JMP HERE AT DOS ENTRY POINT FOR HDERR >
7
8 DECIMAL ;S
9
10 THIS ROUTINE ENABLES OMNIFORTH TO STAY ACTIVE AFTER A HARD
11 DISK ERROR. REVIEW YOUR NORTH STAR DOS SYSTEM SOFTWARE MANUAL
12 SECTION G FOR THE DOS ENTRY POINT TO HDERR. OUR RELEASE HAD
13 HDERR AT 2019 HEX LOADED WITH A C3 (JMP) FOLLOWED BY TWO BYTES
14 OF ZEROS. THIS ROUTINE WILL STORE THE ADDRESS OF HERE IN THE
15 ZERO BYTES (LOCATIONS 201A AND 201B).

```

```

SCR # 62
0 < NORTH STAR DISK READ/WRITE INTERRUPT PATCH > HEX
1
2 CODE :DI DI NEXT JMP < DISABLE INTERRUPTS >
3 CODE :EI EI NEXT JMP < ENABLE INTERRUPTS >
4
5 : %R/W < NEW READ/WRITE DISK ALLOWS INTERRUPTS >
6 SWAP IT@SCALC >R SEC @ DENSITY @ 0= 1+ R>
7 DENSITY @ IF 80 ELSE 0 THEN
8 DRIVE @ OR :DI RWDSK :EI DISK-ERROR ! ;
9
10 : PATCH-R/W < PATCH NEW R/W INTO OLD R/W IN OMNIFORTH >
11 / %R/W CFA / R/W !
12 / ;S CFA / R/W 2+ ! ; PATCH-R/W
13
14 DECIMAL ;S
15

```

```

SCR # 63
0 < NORTH STAR REAL TIME CLOCK INTERRUPT EXAMPLE >          HEX
1 < ASSUMES CLOCK WIRED TO UI0, AND MEMORY AT 0-2 >
2
3 0 VARIABLE INTC          < VARIABLE FOR INTERRUPT COUNTER >
4
5 ASSEMBLER HERE          < SAVE ENTRY POINT FOR LATER POKE >
6 H PUSH PSW PUSH        < SAVE REGISTERS USED IN INTERRUPT TASK >
7   INTC LHL  H INX   INTC SHLD   < INCREMENT INTC COUNTER >
8   50 A MVI  6 OUT          < RESET REAL TIME CLOCK FLAG >
9   PSW POP  H POP  EI  RETI   < RESTORE REGISTERS ENABLE RETURN >
10  C3 0 C!  1 !           < POKE JMP TO ENTRY POINT AT LOCATIONS 0-2H >
11
12 : ARM-CLOCK C0 6 P! :EI ;   < ARM NORTH STAR CLOCK INTERRUPT >
13
14 DECIMAL ;5
15

```

```

SCR # 64
0 < INTERRUPT WITH FORTH RESPONSE EXAMPLE 8080 / 280 MODE 0 > HEX
1
2 CODE :RETI          < DISABLE, RESTORE REGISTERS, ENABLE, & RETURN >
3 DI PSW POP  I POP  W POP  H POP  EI  RETI
4
5 : INT1 ." INTERRUPT 1" CR :RETI ; < FORTH RESPONSE TO 1 RST >
6
7 ASSEMBLER HERE      < SAVE VECTOR ADDRESS ON STACK, POKE LATER >
8 H PUSH  W PUSH  I PUSH  PSW PUSH  EI          < SAVE REGISTERS >
9 / INT1 I LXI NEXT JMP          < POINT TO INT1 FORTH RESPONSE >
10 C3 8 C!  9 !           < POKE JMP TO VECTOR ADDR AT LOCATION 8-10H >
11
12 CODE :RST1 1 RST NEXT JMP          < USE TO TEST INTERRUPT 1 >
13
14 DECIMAL ;5
15

```

```

SCR # 65
0 < NONMASKABLE INTERRUPT TO FORTH ABORT ROUTINE >          HEX
1
2 : NMI CR ." NMI" ABORT ; < NONMASKABLE INTERRUPT TASK: ABORT >
3
4 ASSEMBLER HERE      < SAVE VECTOR ADDRESS ON STACK, POKE LATER >
5 EI / NMI I LXI NEXT JMP < ENABLE AND POINT TO NMI >
6 C3 66 C!  67 ! < POKE JMP TO VECTOR ADDR AT LOCATION 66-68H >
7
8 DECIMAL ;5
9
10 NOTE: REGISTERS ARE NOT SAVED AND RESTORED BY THIS ROUTINE
11 BECAUSE ABORT CREATES A NEW ENVIRONMENT.
12
13 THE ENABLE INTERRUPT SERVES TO KEEP OTHER TASKS THAT YOU
14 MAY HAVE, LIKE A REAL TIME CLOCK, ENABLED AND ACTIVE.
15

```