NFCROM-1B

PRODUCT DESCRIPTION

WRITTEN BY NELSON CROWLE

PROTOTECH, INC.
P. O. BOX 12104
BOULDER, CO. 80303
(303)-447-9883

NFCROM-1A PROGRAM LIST

NFCROM-1A	Displays copyright message
BOOT	Loads ProtoCODER from data registers
DUMP	Dumps any ROM contents into data registers
LEFT	Rotates display one character to the left
,	Appends a left-goose to the display
DISASM	Assembly language disassembler
CL	Clears system flag 12
RCLA	Non-normalised recall from absolute register
CODE	Converts hex characters in ALPHA into non-normalised number in X
DECODE	Converts non-normalised number in X into hex characters in ALPHA
+1	Increments X from 0 until any key is pressed
AK	Hex code key assignments
.	Appends a right-goose to the display
DIS	Appends the character in X (1:0) to the display
ROM?	Displays the revisions of system ROMs 0, 1, 2
DISTST	Turns on most of the display
LOAD	Load single data word into the ProtoCODER
X=1?	Compares X to floating point 1
PROMT	Allows prompting for 1 to 8 hex digits using hex keyboard
LODE	Loads sequential words into the ProtoCODER
CAT	Lists on-line ROMs or CAT 2 starting at any ROM
POW2	Extended precision powers of 2
LODB	Program byte loader (unfinished)
MNEM	Provides assembly language mnemonics
MANT	Returns the mantissa of X to X
ROMSUM	Calculates ROM checksum
M10INT	Returns INT(X) MOD 10 to X
STA	Non-normalised store to absolute register
BJUMP	Byte-jumper
TOGF	Toggles user flag
DEC-HEX	Converts up to 12 decimal digits to a hex number
HEX-DEC	Converts up to 8 hex digits to a decimal number
<>	Exchanges IND X with IND Y
INIT	Initializes ProtoCODER and sets up a catalog table

NFCROM-1A displays copyright message. It uses no user registers.

BOOT loads the ProtoCODER. Five 10 bit words of ROM data are stored in each user register, right justified, with the first (leftmost) 6 bits as 000100 so that the register will not be normalised. BOOT must be used in a program, and loops back on itself until the last register in the calculator (absolute address 1FF) has been loaded. To use, load registers sequentially up to 1FF with data to be loaded. DUMP can be used for this. Enter the following program:

```
LBL 01
BOOT
END
```

Then set up the stack as follows and XEQ 01. BOOT will load the data sequentially until the contents of register 1FF has been loaded. Before execution, set up the stack as:

Z (2:0) contains CODEd absolute address of next ProtoCODER word to be loaded (000 - FFF)
Y set to zero
L (2:0) contains CODEd absolute address of next register to use for data

DUMP dumps 5 ROM words into one user register. Also increments Y and adds 5 to X. By looping, a series of ROM words can be dumped into consecutive registers. To use, set up:

Y (2:0) contains CODEd absolute address of register to load with the data word
X (3:0) contains CODEd absolute address of first ROM word to be dumped

LEFT rotates the display one character to the left during program execution. For example:

```
LBL 01
CLA
AVIEW
10
,
DIS
.
LBL 00
0
ENTER
0
LEFT
GTO 00
```

appends left-goose to the display. See LEFT.

DISASM returns disassembled ROM lines. Input CODEd ROM address in X (3:0). DISASM increments X, returns the result of CXISA in Y, and returns "aaaa ddd c " in ALPHA where aaaa is the ROM address, ddd is the ROM data, and c is the character interpretation. Use with MNEM to list a fully disassembled list of a ROM. See MNEM.

CL clears system flag 12.

RCLA recalls the contents of the register whose absolute address is in X. For example, 511 RCLA will recall the contents of the register with absolute address 1FF, into X. No normalisation occurs.

CODE encodes the hex characters in ALPHA into the X register. For example, "10000000010101" CODE will display X as containing 3 full-man symbols.

DECODE decodes the X register into hex characters in ALPHA. For example, 289 DECODE will place "02890000000002" in ALPHA, which is the floating point representation of 289.

+1 Demonstrates the speed of microcode. Upon execution, it counts starting at zero, incrementing by 1 in each loop, until any key is pressed. The resulting count is returned in X. It runs about 125 times as fast as a user program to do the same thing.

AK Makes key assignments. Any two-byte hex code can be assigned using AK. Upon execution, AK displays 4 underlines for prompts. Type in the hex code to be assigned (eg. 9075 for RCL M) by pressing the digit keys (0-9) or the A-F keys. Backarrow works. After entering the four hex digits, another prompt will be displayed. Press the key to be assigned to, or press SHIFT and the key to be assigned. Example: AK A440 LN to assign NFCROM-1A to the LN key.

. appends right-goose to the display. See LEFT.

DIS appends the character contained in X (1:0) to the display. See LEFT.

ROM? displays which system ROMs are in your calculator.

DISTST turns on all display segments except the top dot on the colon.

LOAD Loads one word of data into the ProtoCODER. Y (2:0) contains the address (000 - FFF) and X (2:0) contains the data to be loaded.

X=1? Compares the contents of the X register with 1. Works just like X=0?.

PROMT Universal hex digit prompting routine. Allows the input of 1 to 8 hex digits (0-9,A-F). The result is returned in the X register, right justified and zero filled to the left. If the user inputs a complete string of digits (filling all the prompts), then the sign bit of X is set to 1. If the user terminates the input early by pressing "." (terminates using what has already been input, eg, for a prompt of 4 digits, inputting 1F. returns 00 00 00 00 00 00 1F in X) then the sign digit is set to zero. Backarrow can be used to backspace or to cancel prompt (returns 0 in X if prompt is cancelled). All other keys are ignored. To use, put the prompt string in the display, set X to 10 ** (# digits - 1 to prompt for) then PROMT. Example: "ADDRESS" AVIEW 100 PROMT will display the prompt "ADDRESS ___ _".

LODE — Loads data sequentially into the ProtoCODER. When executed initially, it prompts ADDRESS _ _ _ _. Type in the address of the first word to be loaded into the ProtoCODER. Then each address to be loaded will display a prompt, eg, FF4 _ _ _. For each address, type in the data to be loaded. To change addresses or stop loading, type backarrow to the first prompt.

CAT — Prompts for one hex character. For inputs 0 - 3, executes the standard CAT 0 - CAT 3. CAT 4 lists the on-line ROM labels (ie, XROM nn,00). CAT 5 - F will list a normal CAT 2, but starting at the ROM page specified. For example, assuming you have the card reader (page E) and the printer (page 6) plugged in, running a CAT E will display the card reader catalog, but will skip the printer catalog.

POW2 — Extended precision powers of two. Soon to be rewritten.

LODB — Program byte loader. Unfinished, however, it works for 1, 2, and 3 byte instructions. Prompts LOAD _ _. Type in the first byte. If the instruction is a two or three byte instruction, LODB will prompt for additional bytes. Example: in PRGM mode, at the place where you wish to load a RCL M instruction, execute LODB. The calculator will prompt LOAD _ _. Type in 90 for RCL. The calculator will now display LOAD 9 0 _ _. Type in 75 for M. The calculator will now display RCL M as a program line.

MNEM — Provides mnemonics for disassembler listings of ROMs. It uses data from the X and Y registers as provided by DISASM. It returns the first half of the mnemonic in Z and the second half in T. It also uses L for two-word ROM instructions (class 1 and LDI). The following program will prompt for a starting address (0000 - FFFF) then list ROM contents from that address until it is stopped.

```
LBL "LIST"
"START"
AVIEW
E 3
PROMT
LBL 00
DISASM
MNEM
ARCL Z
"+ "
ARCL T
PRA
GTO 00
```

MANT — Returns the mantissa of X to X. Example, 345.45E12 MANT returns 3.4545.

ROMSUM — Computes ROM checksum. Starts at CODEd address in X (3:0), and adds ROM words until the end of the 4K ROM is reached. To calculate the checksum for your ProtoCODER or EPROM, set the last word (address FFF) to 000, then set up X to contain the first address of the ProtoCODER (eg, if your ProtoCODER is addressed to page E, enter "E000" in ALPHA then execute CODE). Then execute ROMSUM. This returns the sum in X (2:0). Subtract this result from 3FF in hex to get the checksum to be stored in word EFFF.

M10INT — Returns INT(X) MOD 10 to X. Examples:

M10INT of	returns
136	6
E 38	0
34.9	4
-4.3	5

STA — Stores the contents of Y into the absolute register addressed by X. For example, with SIZE 5:
300 ENTER 511 STA will store 300 into absolute address 511 (hex 1FF) which is register 04. RCL 04 to check. No normalisation occurs when STA is executed.

BJUMP — This is a byte jumper that prompts for the number of bytes to be jumped. Enter the following program, then PACK:
LBL 01
"ABCDE"
The display will show "ABCDE". Execute BJUMP, which will then display BJUMP _. To use BJUMP, it should be assigned to a key. If you type 1 to this prompt, you will see the program line - which is the A in the string interpreted as a standalone line.

TOGF — Toggles (off to on or on to off) the flag specified in the X register. Example: 49 TOGF to see the BAT indicator.

DEC-HEX — Converts up to 12 decimal digits contained in X (11:0) to hexadecimal. The resulting hex value is returned in the right end of X, with the sign bit of X set to 1 so that the value will not be normalised. Example: "1048575" CODE DEC-HEX DECODE will display "10 00 00 00 0F FF FF".

HEX-DEC — Converts up to 8 hex digits contained in X(7:0) to decimal. The resulting decimal value is returned in the right end of X, with the sign bit of X set to 1 so that the value will not be normalised. Example: "FFFFF" CODE HEX-DEC DECODE will display "10 00 00 01 04 85 75".

<> — This function exchanges two registers, pointed to by the contents of the X and Y registers. Example, SIZE 200, then store 12 in 135 and 9 in 45. If you then execute the following: 135 ENTER 45 <> and check the registers, register 135 will contain 9 and register 45 will contain 12.

INIT — Used to initialize the ProtoCODER. It takes several minutes to run, and BEEPs after each third of the ProtoCODER is loaded. It clears all words then sets up a catalog with one entry. To use, set your ProtoCODER to page address 3 (0011) then execute INIT. Change the page address to F (1111) and run a catalog (use CAT F above) to see that the ProtoCODER was initialized properly.

NFCROM-1B ADDENDUM

The following routines were added to NFCROM-1B:

COPEE Copies a 4K ROM into the ProtoCODER. COPEE must be used as a program line. CODE the starting ROM address into L(3:0). Example: To copy the contents of the ROM in page 8, execute the following program:

```
"8000"
CODE
SIGN
COPEE
END
```

SST? Performs SST in program mode only, and continues to SST through program until key is released.

BST? Performs BST in program mode only, and continues to BST through program until key is released.

X+Y Binary addition of X with Y. Result is in X. Y, Z, T, L unchanged.

OR Boolean logical OR of X with Y (bit by bit). Result in X. Y, Z, T, L unchanged.

AND Boolean logical AND of X with Y (bit by bit). Result in X. Y, Z, T, L unchanged.

XOR Boolean logical exclusive-or of X with Y (bit by bit). Result in X. Y, Z, T, L unchanged.

NOT Boolean logical complement of X (bit by bit). Result in X. Y, Z, T, L unchanged.