# PC Computing & Control Software Tools



OPTIONS: HP-41 EMULATOR IEEE-488 RS-232 HP-IL 3D GRAPHICS DATA ACQUISITION



# ELI-41/41SX USER'S MANUAL



Version 2.0 June 1988

# **Table of Contents**

PREFACE : POLICIES AND PROCEDURES
Copyright And Trademark Notice
Technical Support
Acknowledgments
CHAPTER 1 : GETTING STARTED 1-1
What is ELI-41?
SYSTEM REQUIREMENTS
THE BASICS
CHAPTER 2 : USING ELI-41
LOADING ELI-41
THE FILES USED
FILES CONTAINED ON THE DISTRIBUTION DISK 2-4
SCREEN DISPLAY LAYOUT
THE POPUP
AUTORUN AND AUTOEND
CHAPTER 3 : ELI-41 MODES OF OPERATION
THE ALPHA MODE
NUMBER INPUT AND DISPLAY FORMATS
NUMBER INPUT
DECIMAL
HEXADECIMAL
BINARY
OCTAL
NUMBER DISPLAY

ELI-41	
--------	--

FIXED POINT NOTATION
SCIENTIFIC NOTATION
ENGINEERING NOTATION
CHAPTER 4 : ELI-41 PROGRAMMING FACILITY 4-1
PROGRAMMING THE ELI-41
THE EDITOR
Block Delete
Block Insert
Block Copy
Block Mark
Save File To Disk
Comments Editor
Search and Replace
Search Only
SUBROUTINES
CREATING AN ELI-41 PROGRAM WITH AN ASCII EDITOR 4-7
The label statement.
The first character.
Maximum characters.
Comments
Maximum program length.
The END statement
CHAPTER 5 · PRINTER AND DISK OPERATIONS 5-1
PRINTER UTILIZATION FUNCTIONS 5-1
FLAGS AFFECTING THE OPERATION OF THE PRINTER 5-1
CHADTED C. ON LINE CONTEDSION TADIES
WHAT IS A CONVERSION TABLE?
USING ELLAT'S CONVERSION TABLES
MAKING YOUR OWN CONVERSION TABLES 6-4
RULES FOR CODING CONVERSION TABLES

<b>CHAPTER 7 : OPTIONS AND DEFAULT</b>	SET	TINGS	• •		7-1
OPTIONS MENU	• •				7-1
SETTING OPTIONS					7-2
Catalog					7-2
All					7-2
Programs					7-2
Conversion	• •				7-3
Memory			• •		7-3
Internal Functions			•••		7-3
Quit	••		• •		7-3
Print	•••	• • • •	•••	• • •	7-3
ON/OFF					7-3
File (ELI-41SX)	•••		•••	•••	7-4
1 (ELI-41SX)	•••		• •		7-4
2 (ELI-41SX)	•••		• •		7-4
3 (ELI-41SX)	••		•••		7-4
Quit	•••		•••		7-4
Memory	•••		•••		7-4
Dump	• •				7-4
File	• •		•••		7-4
Printer	• •		• •		7-5
Quit	• •				7-5
Restore	•••		•••		7-5
No	• •		•••		7-5
Yes	• •		•••		7-5
Initialize	• •		• •		7-6
No	• •		•••		7-6
Yes	• •		•••		7-6
$Quit \ldots \ldots \ldots \ldots$	• •	• • • •	• •	• • •	7-6
Options	• •	• • • •	•••		7-6
Colors	• •		• •		7-6
Monochrome	• •		•••		7-6
Color	• •		• •		7-7
Filespecs	• •		• •		7-7
Utility Directory	• •		•••		7-8
Program Directory .	• •		•••		7-8
Memory Directory .	• •		• •		7-8
Program File Name	•••		•••	•••	7-8

Memory File Name
Load
No
Yes
Save
$NO \dots \dots$
Quit
gattering
CHAPTER 8 : THE ELI-41 FUNCTIONS
APPENDIX A : THE ELI-41 FLAGS
APPENDIX B • FUNCTION KEYBOARD USAGE B-1
APPENDIX C : FUNCTION TABLES
APPENDIX D : THE EXAMPLES
APPENDIX E : COMMAND LINE SWITCHES
FIGURES
Figure 2-1: CALCULATOR FACE
Figure 3-1: ALPHA MODE screen
Figure 3-2: SHIFT ALPHA mode
Figure 4-1: EDIT MODE
Figure 4-2: EDIT mode, comments entry screen
Figure 6-1: CATALOG 5 display
Figure 6-2: Conversion utility, CONVERT FROM screen 6-2
Figure 6-3: Conversion utility, CONVERT TO screen 6-3
Figure 7-1: OPTIONS MENU TREE

#### **TABLES**

Table 6-1:	CONVERSION	WORK TABLE		•••		•	•	•	•	. 6	5-4
Table 6-2:	CONVERSION	TABLE FILE L	AYOUT	`.	•		•		•	. 6	5-5

#### **EXAMPLES**

Example 4-1: SUE	ROUTINE CA	LLS	•••		•	•••	•	• •	••	•	•	4-6
Example 4-2: SOL	UTION TO QU	JADRA	ГІС	EÇ	JUA	TIC	DN	s	•	•	•	4-11
Example 4-3: ASC	II PROGRAM	FILE	•••		•			• •	••	•	•	4-12

(This page is intentionally left blank)

### PREFACE

### POLICIES AND PROCEDURES

#### **Terms And Conditions**

Eclipse Logic, Inc. makes no representations or warranties as to the merchantability or fitness of ELI-41/41SX for any particular purpose or application. The purchaser is given a lifetime license for the use of ELI-41/41SX and, as such, is permitted to make a copy for backup purposes only. Reproduction or copying of ELI-41/41SX or it's documentation for any other purpose is a violation of Eclipse Logic's Copyright. In no event shall Eclipse be liable for any loss of profit or any other commercial damage, including but not limited to special, incidental, consequential or other damages.

#### Copyright And Trademark Notice

Both the program and this manual are proprietary, copyrighted material, and no part of either may be reproduced, transmitted, stored, or translated into any other computer or foreign language without the express written permission of Eclipse Logic Incorporated.

Copyright<sup>©</sup> 1985-88 by Eclipse Logic Incorporated. The ELI-41 and ELI-41SX<sup>™</sup> are trademarks of Eclipse Logic Incorporated.

All Rights Reserved.

#### Software Registration

We urge you to register your software. Registration is free and is easy to do. Simply complete the Eclipse Logic Software Registration Form provided in the back of this manual and mail it back to us. Once we receive your completed registration form, you will be entitled to:

- Technical support for 45 days
- New product announcements and update information
- Extended support plan

#### **Technical Support**

Your acquisition of this product entitles you to a free 45 day initial support period that begins at the time of Registration. As noted previously this technical support is available only to registered owners.

During this period you may call our technical support representatives for answers to your questions about this software product.

Our standard technical support times are 8:00 AM - 5:00 PM, local Pacific time, Monday-Fridays, excluding holidays.

You may contact us at:

ECLIPSE LOGIC INC. Technical Support Division 1815 N.W. 169th Place, Suite 5040 Beaverton, Oregon 97006 TEL: (503) 629-0917

#### Acknowledgments

In this manual, references are made to a number of trademarks:

IBM<sup>®</sup> is a trademark of International Business Machines. HP-41<sup>®</sup> is a trademark of Hewlett-Packard company. HP-IL<sup>®</sup> is a trademark of Hewlett-Packard company. MSDOS<sup>®</sup> is trademark of Microsoft. PCDOS<sup>®</sup> is a trademark of IBM Corp. ELI-41<sup>™</sup>, ELI-41SX<sup>™</sup>, ELI-IL<sup>™</sup>, ELI-488<sup>™</sup> are trademarks of Eclipse Logic Inc. (This page is intentionally left blank)

### **CHAPTER 1**

### **GETTING STARTED**

#### What is ELI-41?

ELI-41 is an advanced scientific calculator program for use with IBM or compatible personal computers. The program simulates the Hewlett-Packard HP-41 calculator and supports all of the HP-41 functions and features. In addition, ELI-41 offers the following features which are not available with the HP-41:

- ELI-41 can operate in either of two modes: memory resident mode or execute-only mode. In memory resident mode, all of the ELI-41 commands are available any time the <Ctrl-~> keys are pressed. In execute only mode, ELI-41 operates as a "standard" application program.
- Not copy protected
- Over 150 functions for performing basic mathematics and advanced trigonometric and statistical computations
- Data Entry and Display in Decimal, Binary, Octal or Hexadecimal mode and performing all arithmetic functions in the respective mode with up to 16 digits of precision.
- Unique view of stack, flags and data registers.
- An On-line Program and Comments Editor.
- Unlimited number of program lines per file. (ELI-41SX)
- Unlimited number of program files.
- Creating programs of virtually unlimited size by nesting program calls.

- Unlimited number of programmed ASCII conversion tables.
- Support for majority of the HP-41 printer and disk commands. (ELI-41SX)
- Ability to save and restore to/from disk all data registers, stack, flags and user customized keyboard definitions.
- Print data storage elements, program listings and intermediate results
- User keyboard reassignments
- Full support for DOS multidirectory pathing
- Program source code stored in ASCII text files
- Import/Export data from/to another application while in Pop-Up mode
- Automatic sensing of 8087/287/387 math coprocessor chips

It is not the purpose of this manual to provide instruction on the basic HP-41 calculator operation, but to provide the necessary information for the proper use of this program. ELI-41 users that are not familiar with the HP-41 calculator should obtain one of the many texts available that cover the use and programming of Hewlett-Packard calculators. The READ.ME file on the distribution disk contains a partial list of texts available.

#### SYSTEM REQUIREMENTS

ELI-41 requires that you have an IBM-PC, XT, AT or a true compatible personal computer. Minimum system requirements are 256K bytes of random access memory, one floppy disk drive operating under MS-DOS (or PC-DOS) version 2.x or above operating system, an IBM or compatible Color/Composite or Mono/Graphics adapter card and a display monitor. The use of an 80x87 math co-processor is optional. ELI-41 automatically senses and uses 80x87 instructions if one is present. The 80x87 is recommended for high accuracy / precision computations.

The initial memory requirement at start-up is 128K. Initializing routines set aside various work areas and release the balance of the original 128K for system usage.

At start-up, ELI-41 performs an interrupt call to determine if a multitasking system is present. If a multitasking system such as DoubleDos or DESQview is present then it will use the screen memory address returned by the interrupt call. Otherwise, the calculator checks the machine's equipment configuration for the default display type, and directly writes to the video memory. The printer is assumed to be LPT1 in the ELI-41 program.

#### THE BASICS

The ELI-41 uses memory registers to store number entries and temporary results of calculations, as well as alphanumeric strings. The primary registers used in all calculations are referred to as the Stack. The Stack consists of five (5) basic data registers, which we will refer to as Stack-X, Stack-Y, Stack-Z, Stack-T and Last-X. Most calculations that change the contents of the Stack-X register, store the contents of the Stack-X in the Last-X register previous to any computations. Another basic register of the calculator is the Alpha-register which can contain up to 24 alphanumeric and special characters.

The ELI-41 is an RPN (Reverse Polish Notation) type calculator. Therefore, both of the numbers involved in a calculation must be present in the Stack-X and Stack-Y registers before the actual function can be performed. In order for the calculator to know that the entry of the first number is ended and the second number is being entered you must terminate the first number's entry with the <Enter> key. So to compute 3 \* 4 = 12 in RPN format press the following keys:

<4> <Enter> <3> <\*>

The result will be displayed both in the Data Entry area as well as in the Stack-X register.

Corrections to the number as it is being entered can be made by using the <Back Space> key. This key will delete the last digit of the entry and remove it from the display. Continued use of this key will remove each digit until all digits are removed. Use of the <Back Space> key, with no entry having been made, will set the Stack-X register to zero and update the display.



### **USING ELI-41**

#### LOADING ELI-41

Before using ELI-41, make a copy of the distribution disk and use it as your working copy. Put the original ELI-41 disk in a safe place. You may want to create sub-directories for the executable file, program files and conversion table files. On the distribution disk is a file called INSTALL.BAT that installs the files from the distribution disk to the drive specified on the command line (example: A:> install c:). You may want to modify it to fit your own system configuration. (Please refer to the DOS manual for information on creating directories and copying files.)

If you want to retain the current configuration settings, then the option file (ELI41.OPT) should be copied to the same directory as the executable file (ELI41.EXE). The calculator can create this file. When the ELI-41 is executed, the option file is loaded from the current directory, and with its absence ELI-41 will use default settings stored in the calculator itself. These settings specify the calculator display attributes and the directory paths that are searched for the program, memory, and conversion table files.

To run ELI-41 first change current directory to the directory where the executable program and the option file resides; This is necessary so the program will be able to find the option file, then type ELI41 in response to the DOS prompt for the execute-only mode and ELI41 /P for memory resident mode. An additional command line option that has been provided is "/n", which disables the calculator control of the <Num Lock> key. To exit ELI-41 press the <Alt-F1> keys.

NOTE: With pop-up use the foreground application may have disabled the use of the <Alt> key, like BASIC. Under such conditions an exit may be accomplished by the use of the XEQ OFF function.

NOTE: Some application programs operate in graphics page 1 instead of graphics page 0. To exit properly, while running these applications in the foreground, use the <Alt-F3> key.

#### THE FILES USED

ELI-41 accesses several different kinds of files. The type of file is identified by the extension of the filename on a directory listing. The file specifications are as follows:

#### EXT. CONTENTS

- \*.PGM: Files with this extension are program files. This type file is created by the ELI-41 Program/Comment Editor or any word processor or text editor (ASCII FORMAT). The contents of .PGM files are a series of ELI-41 commands.
- \*.OPT: The ELI-41 Option File is stored with .OPT extension. It contains information about the environment that the calculator is working in, such as directory paths, colors, file names, etc. The complete filename of the option file is ELI41.OPT. Screen attributes, file paths and other environmental functions can be modified on the option screen (Chapter7). This type file is generated by the calculator.

NOTE: If you are using a color card driving a composite monitor rename ELI41.OPT to something else (to save it) and rename COMPOSIT.OPT to ELI41.OPT.

The ELI-41SX product uses an Option file that is different than the ELI-41.OPT file. The options for this product are stored in binary format in a file called ELI-41SX.OPT.

NOTE: The ELI-41SX product has a command line option that would force the calculator to use screen attributes suited for Composite screen monitors. This feature is useful when you have a composite monitor attached to a color card. This feature can be used by specifying/B on the command line when ELI-41SX is executed.

\*.MEM: Files with this extension are memory files. At anytime while using ELI-41, you may save memory (Stack, All Registers, Flags and User Assigned Keys) to disk and retrieve the information at a later time. .MEM files are generated by the calculator.

The ELI-41SX doesn't use the memory file format of the ELI-41. It has three new file formats that are used to store Register contents, Status, and user key assignments.

- **\*.MST:** Files with this extension contain the status of the ELI-41SX when they were saved by the WRTS function.
- \*.MRG: Files with this extension contain the ASCII values of the ELI-41SX Memory registers when they were saved by the WRTR and WRTRX functions.
- **\*.MKY:** Files with this extension contain the ASCII strings of the ELI-41SX user key assignments when they were saved by the WRTK function.
- \*.CNV: Files with this extension are Conversion tables. Several conversion tables are provided with the ELI-41 system. You may also create your own conversion table file with a standard word processor or text editor that supports ASCII text format (Refer to chapter 6).

#### FILES CONTAINED ON THE DISTRIBUTION DISK

For a complete description of all files on the distribution disk please read the "READ.ME" file supplied on the distribution disk.

#### SCREEN DISPLAY LAYOUT



Figure 2-1: CALCULATOR FACE

Refer to Figure 2-1 for the following references regarding the calculator screen display. In the upper left hand corner is the Stack and Alpha-register display area. A highlight bar will highlight the last register used. Located below the Stack display area is the data register display area. Ten registers are visible at anytime. Again, the last register accessed is highlighted by a highlight bar. The data register display area may be scrolled by using the <Cursor>

direction keys direction keys (with the <Num Lock> key off). The action of the cursor keys is as follows:

<home></home>	- Display from register 0.
<end></end>	- Display the last ten registers.
<down></down>	- Down one register.
<up></up>	- Up one register.
<pgup></pgup>	- Previous ten registers.
<pgdn></pgdn>	- Next ten registers.

The flag status box is in the center of the screen and allows viewing of the sixty (60) internal flags. Flags in the OFF state are displayed as a small dot. Flags that are in the ON state are displayed as a small square. Refer to Appendix A for definition of flag settings.

The 25th line on the screen displays the function keys and their usage. Refer to Appendix B for a full description of their use.

On the right hand side of the display is the calculator itself. The top line is what we will refer to as the Data Entry Area. It is used as a data input area and also the display area for the results. Error messages also appear in this area. On the second line in this area is where instructional messages are displayed. We will refer to this area as the Instruction Area. Below this area is the Annunciator Area. This area displays the current calculator mode settings (CALC, SHIFT CALC, USER, SHIFT USER, ALPHA and SHIFT ALPHA), current trigonometric angle mode (DEG, RAD, GRAD) and the numeric display & entry mode (DEC, HEX, BIN, OCT).

Below the annunciator area is the main body of the calculator. To execute any of the functions, the assigned letter must be pressed. The assigned letter for each function is shown on the left hand side of the function. Some of the items do not have a letter assigned such as, numbers 0 through 9 and (+,-,/,\*) functions. In order to enter or execute any of the these items simply press the associated key on the keyboard or the Numeric entry keypad.

Each time you press a key, that key will be highlighted on the calculator face. This allows you to see what your last keypress was.

When the "NUM LOCK" function is active it is indicated by the word, "NUM" located on the bottom border of the calculator. The NUM LOCK function can be toggled (On/Off) By using the <Num Lock> key. "CAPS LOCK" function acts likewise, and is located just to the left of the "NUM" display. A third indicator that can resides in this area is the "PRT" indicator. It indicates that a printer is attached and is on line, and also the print enable flag (Flag 21) is set ON. The fourth indicator is "WAIT", and it will be visible when the calculator is performing an activity which is not immediate, such as printing a program or reading from disk.

Upon start up of ELI-41, the calculator enters the default CALC mode. The function keys  $\langle F8 \rangle$ ,  $\langle F9 \rangle$ ,  $\langle F10 \rangle$  allow you to change current mode to the USER mode, ALPHA mode, and the SHIFT mode operators, respectively. Like the  $\langle Num Lock \rangle$  key, they act as a toggle. For instance, if the calculator is currently in CALC mode and you want to change to SHIFT CALC mode. Simply press  $\langle F10 \rangle$ . The calculator Annunciator will change from CALC to SHIFT CALC. The contents of "keys" area will display SHIFT CALC type function keys and the keypress character availability will be for the SHIFT CALC mode. The SHIFT mode allows only one key press and then it will revert to the un-shifted mode. But if you want to return to CALC mode, without choosing a function, simply press the Shift key  $\langle F10 \rangle$  again.

While in CALC mode, you may switch to ALPHA mode by pressing the  $\langle F9 \rangle$  key. While within the ALPHA mode, you may enter up to 24 alphanumeric characters. Upon terminating the ALPHA mode by pressing either the  $\langle Enter \rangle$  key or the  $\langle F9 \rangle$  key the the content of the Alpha register will be replaced with the characters entered, and the calculator will revert back to the mode previous to the ALPHA mode (CALC or USER modes).

The USER mode can be initiated by pressing the <F8> key. while in the USER mode you can activate the functions or programs that had been assigned to any of the calculator keys. In USER mode the face of the calculator changes to a two-up vertical table display and the available functions and their assigned key will be displayed. Each function can be executed simply by pressing the assigned key. Specific information on each of the calculator modes will be discussed in later chapters.

#### THE POP-UP

ELI-41 is loaded as a pop-up by typing ELI41 followed by '/p' on the command line. The pop-up version of ELI-41 must always be loaded from the DOS prompt or a batch file ("??????.bat"). ELI-41 uses well behaved DOS functions for most operations and will coexist with almost any other memory resident program. Some memory resident programs must be loaded last, however the ELI-41 has no restrictions in this regard.

When the pop-up ELI-41 is initially loaded, a copyright notice along with the current version is displayed on the top left corner of the screen. If the pop-up version has been loaded previously, an error message will be displayed and the normal operations will resume. In the event that there is insufficient memory for the ELI-41 to load, a message will be displayed and ELI-41 will NOT be installed in memory resident mode.

To activate ELI-41, hold the <Alt> key down and press the back quote <'> key (the un-shifted < $\sim$ > key, located beneath the <]> key on standard IBM-PC keyboards). This key combination was designed so as not to conflict with other programs.

Upon using the activating keypress, the calculator screen display will appear and its operation is the same as the basic calculator. The <Alt-F1> (or <Alt-F3>) keys will terminate ELI-41 but the program will remain memory resident and your screen will be restored to what it was prior to entering the calculator. If the previous application was using the graphics page 0, then use the <Alt-F1> keys to exit, and if graphics page 1 was being used then use the <Alt-F3> key sequence. In the event that the screen did not restore correctly upon exiting, simply bring the calculator back up with the activating keypress and exit again with the opposite exit key sequence. The pop-up version can be removed from memory. From a DOS prompt, press the <Ctrl-Shift-'> key sequence (the un-shifted < $\sim$ > key) simultaneously. All memory used by the pop-up will be released back to the system. Any memory resident programs loaded after ELI-41 must be removed prior to removing ELI-41. Failure to do so will leave a 'hole' in low memory which DOS cannot handle and your machine will lock up. This requires a re-boot of the system.

Incorporated into the pop-up mode of ELI-41 is the option of importing data from the previous application and/or exporting the contents of the Stack-X register to the previous application.

The import function is activated by the use of <Alt-F2> function key while the calculator is being displayed. The press of this key sequence will display the screen that was present when the calculator was popped up. The acceptable key usage at this time is limited to the cursor movement keys: the <Cursor>, <Home>, and <End> keys.

The <Enter> key signals that the location of the cursor is positioned upon a character of the string of numeric characters (maximum 16 characters) that are intended to be loaded into the Stack-X register. A limitation to this numeric string is that it must be delimited on either end by a white space character (that is space, tab and line feed characters). The import feature is limited to pulling text from a text display of the previous application. The graphic display of numbers will not work.

The export function can be activated at anytime after an entry has been made within the calculator. The contents of the Stack-X register is the number that can be exported to another application. This function only works when the calculator is not being displayed. The key press sequence associated with this function is the <Ctrl-'> key (un-shifted < >). The last content of the Stack-X register is inserted at the current cursor location. The way that the number looked in the calculator's data entry area, less any commas, is the way that it will look when it is inserted. As often as this key sequence is used, that number will again be inserted at the current cursor location.

#### AUTOMATIC EXECUTION OF PGM'S UPON LOADING

The ELI-41SX program has an option to automatically execute a program named AUTORUN.PGM upon entry and also execute a program called AUTOEND.PGM upon terminating the ELI-41SX program. If the said programs exist in the same directory as the ELI-41SX.EXE then the automatic execution begins. This feature is useful when customizing the keyboard functions within USER mode, or to load customized keyboard assignments, data files and etc.

### **CHAPTER 3**

### **ELI-41 MODES OF OPERATION**

#### THE ALPHA MODE

The ALPHA mode contains the alphabetic characters that have been implemented in the ELI-41 calculator. The purpose of these characters are either to build prompt strings, or to specify the name of a program or an internal function for XEQ, LBL, COPY, PRP, etc. functions.

The available characters in the ALPHA mode are  $\langle A \rangle - \langle Z \rangle$  and = ? : ,(comma) SPACE  $\langle Sace Bar \rangle$  key but all characters 1 - 255 are available in the ELI-41SX.



Figure 3-1: ALPHA MODE screen.



Figure 3-2: SHIFT ALPHA mode.

To switch into the ALPHA mode the user must press the  $\langle F9 \rangle$  key, and to leave the alpha mode,  $\langle F9 \rangle$  or the  $\langle Enter \rangle$  key must be pressed. Upon exit from the ALPHA mode the string entered previously will be placed in the Alpha Register of the calculator which can contain a maximum of 26 characters. The previous statement holds true if the switch to alpha mode was made from one of the calculator modes, and not through a function that was receiving a program label or function name.

By pressing the <Back Space> key while in the ALPHA mode the last character entered will be erased.

#### NUMBER INPUT AND DISPLAY FORMATS

In contrast to hand-held scientific calculators, ELI-41's screen display affords high visibility of all pertinent information, easy keyboard entry and useful additional features.

#### NUMBER INPUT

There are four different number bases that could be used to input numbers into ELI-41: DECIMAL, BINARY, OCTAL and HEXADECIMAL. The choice of number entry and display mode (within the calculator data entry area) is made by pressing the appropriate function key. The current display mode is displayed in the Annunciator Area of the calculator face as "Dec", "Bin", "Oct", or "Hex". Number input and display always takes place in the current display mode. When the number input; in any base, is complete, then the number will be displayed in Stack display area as the Stack-X element in the Decimal mode, and is also displayed within the data entry area in the selected base. Any time before an entry is terminated, the number entry may be edited by use of the <Back Space> key. A brief description of each entry mode follows:

#### DECIMAL <F1>

The DECIMAL entry mode is the default upon program initialization. The acceptable number entry keys are zero <0> through <9> and the <.>. Any other characters entered will terminate number entry, set the Stack-X equal to the number entered, push the stack if stack lift was enabled, and finally perform the associated function.

The "EEX" function is used to enter an exponential factor for the current number and is obtained by pressing the <P> key.

To enter a negative number, simply key in the number and then press the  $\langle O \rangle$  key (CHS).

NOTE: the "-" is not used for negative number entry but for performing subtraction only.

#### HEXADECIMAL <F2>

HEXADECIMAL number entry mode allows input of numbers in base 16. The acceptable number entry keys are <0> through <9>, and A through F. The largest number that can be input in this mode is 16,777,215 decimal (FFFFFF in Hexadecimal). If the number entered is larger, an "OUT OF RANGE" message will be displayed. The gray <Back Space> key must be used to clear the error message. Afterward, the calculator automatically switches to the DECIMAL entry and display mode. The functions that are normally assigned to the <A>-<F> keys are disabled in this mode.

#### BINARY <F3>

#### OCTAL <F4>

OCTAL entry mode allows input of numbers in base 8 and consists of the digits zero through seven. Any other key press terminates the entry and performs the identified function for that key (if defined). The largest number that can be input in this mode is 16,777,215 decimal (77777777 in Octal). If a number is larger, an "OUT OF RANGE" message is displayed. The gray <Back Space> key must be pressed to clear the error message. Afterward the calculator automatically switches to the DECIMAL entry and display mode.

#### NUMBER DISPLAY

The contents of the Stack and Register display areas are always displayed in fixed decimal notation (with non- significant zero suppression) with 12 decimal digits. If the number is too small or too large to be displayed in fixed decimal format then it will automatically be displayed in SCIENTIFIC notation.

#### FIXED POINT NOTATION

This is the default number Entry/Display format while the calculator is in DECIMAL (base 10) mode. In this mode you can specify the number of digits to be displayed after the decimal point. The calculator normally displays numbers with commas separating groups of numbers such as: 123,456,789.0234. The group separation occurs if flag 29 is set. Also the separator (,) and the decimal point (.) can be swapped using flag 28. (European notation)

The ELI-41 automatically changes the display format from Fixed Point to Scientific notation when the number to be displayed is either too small or too large to be viewed with a fixed decimal point.

To change the default display of 9 decimal places, change to "SHIFT CALC" mode and press the  $\langle Z \rangle$  key (FIX). You are then prompted to enter the number of decimal places (0  $\langle = nn \rangle \langle = 15 \rangle$ ). This action affect only the display format of the calculator Data Entry Area.

#### SCIENTIFIC NOTATION

The scientific notation displays one digit to the left of the decimal point and up to a three digit exponent. The displayed number is rounded to the specified number of decimal positions. To change the number of displayed decimal places, change to "SHIFT CALC" mode and press the  $\langle = \rangle$  key (SCI). You are then prompted for the number of decimal digits ( $0 \langle = nn \rangle = 15$ ). This action affects only the display format of the calculator Data Entry Area.

#### **ENGINEERING NOTATION**

The engineering notation is similar to scientific notation except that the exponent is always forced to be in multiples of three. It is most useful in the scientific and engineering calculation where most units of measure are specified in multiples of three (Micro, Mega, Kilo, etc.). The displayed number is rounded after the specified number of decimal digits. To change the number of displayed decimal digits, change to "SHIFT CALC" mode and press the <?> key (ENG). You are then prompted for the number of decimal positions. This number must be between zero and 15. This action affects only the display format of the calculator Data Entry Area.

The following example assumes 4 decimal places were specified:

DECIMAL	SCIENTIFIC	ENGINEERING
123.4569	1.2346E+02	123.46E+00
1234.5679	1.2346E + 03	1.2346E + 03
0.1234567	1.2346E-01	123.46E-03

### **CHAPTER 4**

### **ELI-41 PROGRAMMING FACILITY**

#### **PROGRAMMING THE ELI-41**

ELI-41 programs are structured the same as Hewlett Packard calculator programs. The program instructions follow the same sequence as if the key strokes were being directly entered at the time of the calculation. It is beyond the scope of this manual to give detailed instruction on programming technique or structure; however, some simple examples will be given to illustrate the use of ELI-41's special functions. Users that are not familiar with programming in Hewlett-Packard's RPN format are advised to refer to the READ.ME file on the distribution disk for sources of detailed programming instruction.

#### THE EDITOR

To afford users an easy program entry and editing interface, the EDITOR has been developed. The EDIT mode can be selected by simply pressing the <F7> key. This environment allows building of programs by the mere press of the calculator keys.

In EDIT mode all key presses are trapped, and each key press will insert the corresponding instruction code into the program buffer. Intermediate testing or viewing of the current register contents can be done by exiting the EDIT mode by pressing the toggle key <F7>.



#### Figure 4-1: EDIT MODE

To begin an edit session, press the  $\langle F7 \rangle$  key. 'EDIT \_\_' will be displayed in the data entry area. This prompt waits for a program name to be entered (maximum seven characters). The gray  $\langle Back \rangle$ Space> key may be used to abort this prompt and also edit the program name entered. To enter the program name either go into the ALPHA mode by pressing the  $\langle F9 \rangle$  key and enter the program name, or press the  $\langle F9 \rangle$  key once more to display a catalog of the program names in the current program directory. Simply select the name and press  $\langle Enter \rangle$  to start editing that program. The program is loaded into the program buffer area, and the EDITOR window will open displaying the beginning instruction code of the program.

If the program name is not found on disk, then the EDITOR window will open and displayed the two lines of required code; the program label and an END statement. These two lines cannot be deleted. The highlighted line in the EDITOR identifies the current instruction. The EDITOR is an insert mode only editor, as any calculator functions will be inserted above/prior to the current instruction. The current instruction may be deleted by pressing the gray <Back Space> key. A message in the data entry area will appear waiting for confirmation of this deletion.
The ELI-41SX program is also able to operate in append mode, where, the instructions are appended after the current highlighted instruction. The append mode can be enabled by using the /A switch from the command line when the ELI-41SX is loaded. This feature is very similar to the method used by the HP-41 calculator because instructions are inserted after the current line.

Function key usage is displayed on the 25th line of the screen. The edited program can be saved to disk at any point during its development by pressing the <Alt-F7> key. The <Alt-F1> keypress will escape the ELI-41 environment and return to DOS prompt. A safety feature is included that if the program has been changed a message will display warning of this circumstance and allow you to save the resident program file at that point if desired.

The <Alt-F8> key will open a simple comment editor window which allows inserting descriptive comments along with the developed instruction code (Figure 4-2). The <UP Arrow> and <DOWN arrow> keys can be used to scroll the comment display. The <Esc> key will exit the comment editor and return to the program instruction EDITOR.

1 Z	#LBL ELI41 ;PROGRAM AND COMMENT	14,5	00	).0				
j	BIN : DOCUMENTATION OF YOUR			<b>-</b> 1 -				_
	DEC ; PROGRAMS.			Calc		D	eg	Dec
5	PRREG : PRINT ALL REGISTERS	Α Σ+	Б	1/x c	٩×	r	LOG	E LN
,	PRSTK :.STACK AND FLAGS.	<b>I</b>	Ĩ	+		Ì		- <b>I</b>
3	PRFLAG	Ëx*y	Ġ	R↓ Ĥ	SIN	i	COS	J TAN
)	AND ;VIEW DATA IN HEX,BIN,OCT,DEC.	<b>I</b>	+	+		-+	·	
0	OR ; BOOLEAN FUNCTIONS.	Shi		XEQ Ĺ	STO	۰.M	RCL	SST
1	XOR		1	+		-+		+
12	NOT	EN	TI	ERT Ó	CHS	P	EEX	.∢ .∢ .
13	<< 9					_		
14	>> 8	- 1		7		8	1	9
15	CONVERSION: LENGTH							
16	CONVERT FROM: 1 ; CONVERSION TABLES	•		4		5	, I	6
17	CONVERT TO: 4			+				
18	END ;END OF EL141 PROGRAM EDITOR.	*		1		Z	•	3
		1		0				R/S
iment V	IEW DATA IN HEX,BIN,OCT,DEC.							

Figure 4-2: EDIT mode, comments entry screen.

The EDITOR environment also doubles as a debugging environment by use of the single step key <;>. The message area on the calculator face will display the instruction just executed as in normal single step mode. The highlighted instruction in the EDITOR is the next instruction line after the executed instruction. BE WARNED though, if you use the <Cursor> keys to change the current EDITOR line, single step mode will be out of synchronization with program execution until a GTO or XEQ instructions is performed. If you want to execute the same series of instructions again, use the BREAK function to set a break point in the program, execute to the break point and single step from that point.

Our format for program files allows any number of comment lines to begin with a (;) comment character in column one. Some of the program files on the distribution disk use this format. It is recommended, of course, to work with a copy and not with the original distribution disk. The ELI-41 EDITOR does not support or generate this type of comment line and will not save it back out if an existing file using this format is edited. To save a file with a new name and to keep from overwriting the older one on disk, make the current file label line (line number one in the EDITOR) the current line and insert a new label with different name. The old label can then be deleted with the <Back Space> key. Another way is by using the OPTIONS screen to change the directory path that files are loaded or saved to.

The ELI-41SX has enhanced its EDITOR capabilities by adding the following features:

# **Block Delete**

<Alt-F3> : Deletes from the Beginning line of the marked block down to and including the current highlighted program line.

# **Block Insert**

<Alt-F4> : Inserts the contents of the holding buffer before the current line.

# **Block Copy**

<Alt-F5> : Copies the marked block into the temporary holding buffer for use with <Alt-F3> or <Alt-F4>. After the beginning line of the block is marked (<Alt-F6>), use the down arrow key to highlight the ending line of the block and then press the <Alt-F5> keys to copy the marked block into the temporary buffer.

# Block Mark

<Alt-F6> : Marks a block of program lines to be Copied to temporary holding buffer(<Alt-F5>), Inserted at any point within the current program or any other program edited(<Alt-F4>), or Deleted(<Alt-F3>). To use this feature, use the arrow keys to highlight the upper line of the block and then press the <Alt-F6> keys to begin marking the block. This action marks the beginning line of the Block.

# Save File To Disk

 $<\!$  Alt-F7> : Saves the contents of the program buffer currently edited to disk.

# **Comments Editor**

<Alt-F8> : This feature allows users to comment each and every line of the program code.

# Search and Replace

<Alt-F9> : This feature allows Replacement of any program line with a program line supplied by user.

# Search Only

<Alt-F10> : This feature searches for a string given by the user from the current line down to the end of the program buffer.

# SUBROUTINES

When a program encounters an XEQ label ('Xlabel') statement, the subroutine 'label' will be loaded from disk and executed. The stack and registers will not be effected during the load and may be used to pass values to and from the subroutine. When a RTN ('SRTN') statement is encountered in the subroutine, execution is returned to the calling program on the line immediately following the XEQ label ('Xlabel') statement.

To understand the way subroutines work, let's look at the operation of the programs that will be used in this example.

The program XEQTEST reads:

LBL XEQTEST	; Program name label
XEQ TONE1	; Calls the subroutine 'TONE1'
"MARY"	; Places the string 'MARY' in the
ASTO 00	; alpha register 00
XEQ IND 00	; This is an indirect subroutine call.
; When this instruc	tion is executed the
; program will brar	nch to the label contained
; in the referenced	register (00).
; In this case regis	ter 00 contains 'MARY'
; and the program	loads and executes the MARY program.
END	; Program ends

#### **Example 4-1: SUBROUTINE CALLS**

# CREATING AN ELI-41 PROGRAM WITH AN ASCII EDITOR

ELI-41 programs can be created with any text editor that outputs an ASCII file to disk. Program structure is identical to the HP-41 calculator with the addition of a program control character at the beginning of each line. When writing a program, the following syntax must be adhered to:

# 1. The label statement.

A program must start with a label statement that contains the program file name, less the file extension. Comment lines may come before the label statement.

# 2. The first character.

The first character in a statement must be either a semicolon (;) or a program control character. The program control characters and the ELI-41 command types that they are to be used with are listed in Table 4-1.

# 3. Maximum characters.

The maximum number of characters per line is 79.

# 4. Comments.

A comment must begin with a semicolon (;) in/or before the 26th column. A space or tab character must separate any executable statement or command and the semicolon.

# 5. Maximum program length.

The maximum program length is 300 executable statements for the ELI-41 program. Programs longer than this may be broken into segments or subroutines and linked together with the XEQ or GTO commands. This feature is illustrated in Example 4-1.

ELI-41SX has no limitations on the number of program statements per program file.

# 6. The END statement.

Every program must end with an END statement.

#### Table 4-1: PROGRAM PREFIXES

Control Char	Command Format Description
"L"	
	Label, Marks the beginning of a program or subroutine. The variable 'label' may be up to 7 alpha or numeric characters.
" <b>N</b> "	
	Nnnn, Places the numeric value 'nnn' in the Stack-X register. The variable 'nnn' may be up to 16 digits.
" <b>T</b> "	
	Tfunc, Designates a function, 'func' that requires additional parameters (i.e. STO).
" <b>A</b> "	
	Aaaa, Places the string 'aaa' in the ALPHA register. The variable 'aaa' may be up to 24 alpha characters. (Automatically places ELI-41 into ALPHA MODE, stores 'aaa' and cancels ALPHA MODE).

#### Table 4-1: PROGRAM PREFIXES (Continued)

"S"	
	Sfunc, Designates a single key function that operates directly with Stack registers with no additional parameters (i.e. COS).
"X"	
	Xlabel, Designates the XEQ command. It Branches program execution to the disk resident program or subroutine specified by 'label'. Program execution returns to the calling program on the line following the XEQ command when a RTN command is encountered in the called program/subroutine. XEQ subroutines may be nested to 30 levels deep.
" <b>G</b> "	
	Glabel, Designates the GTO command. Transfers program execution to the specified disk resident program, local alpha or numeric label.
" <b>C</b> "	
	Clabel, Loads the conversion file specified by 'label'. The file may be loaded at any time during program execution prior to a Fnn command.
" <b>F</b> "	
	Fnn, Specifies the "Convert From" matrix element, 'nn', of the conversion file previously loaded with the Clabel command. Must be immediately followed by a Dnn command.
" <b>D</b> "	
	Dnn, Specifies the "Convert To" matrix element, 'nn', of the previously loaded conversion file and performs the conversion. Must immediately follow a Fnn command.
"@"	
	@aaa, Appends string 'aaa' to Alpha register. The @ symbol is a substitute for the HP-41's Lazy T symbol.

The following examples will illustrate some programming basics and the proper program format. A common problem encountered in many disciplines is the solution to quadratic equations having the form:

 $AX^2 + BX + C = 0$  where x1 and x2 are the roots

$$-b + SQRT(b^2 - 4ac)$$
  
x1 = -----  
2a

and

A possible ELI-41 keystroke sequence to calculate the solution is:

ENTRY	DESCRIPTION
nn STO 01 nn STO 02 nn STO 03 RCL 02 CHS RCL 02 X†2	Enter coefficient 'a' Store in register 01 Enter coefficient 'b' Store in register 02 Enter coefficient 'c' Store in register 03 Recall 'b' coefficient Loads -'b' on the stack Recalls 'b' coefficient Calculates & loads 'b'
RCL 01 RCL 03 * 4 *	Recall 'a' coefficient Recall 'c' coefficient Calculates a * c
SQRT STO 04 + 2 / RCL 01 / X < > Y RCL 04 - 2 / RCL 01	Calculates SQRT(b <sup>2</sup> - 4ac) Stores intermediate result in 04
1	Calculates X2 root

### **Example 4-2: SOLUTION TO GUADRATIC EQUATIONS**

An ELI-41 program can be written directly from this keystroke sequence. By adding a Program Control Character to each function and saving the file, this program can be recalled and executed whenever desired. User prompts have been added to enhance the program operation.

; This program solv ; equation. ; ** CAUTION **	ves for the real roots of a quadratic
; This program will LROOTS	work only if both roots are real numbers. ;Program name must be ROOTS.PGM
AINPUT a SPROMPT	
TSTO 01 AINPLIT b	; Inputs 'a' to register 01
SPROMPT	
TSTO 02 AINPUT c	; Inputs 'b' to register 02
SPROMPT	
TRCL 02	; inputs 'c' to register 03
SCHS	; Loads Stack-X with -b
TRCL 02	; Pushes Stack-X register and stores register 02 in X
SX∱2	; Loads stack with b <sup>2</sup>
TRCL 01	
TRCL 03	
N4	
S*	
S-	•
SSQRT	; Calculates SQRT(b <sup>2</sup> - 4ac)
TSTO 04	; Stores the above intermediate result in register 04
S+	; Calculates -b + SQRT(b <sup>2</sup> - 4ac)
N2 S/	; (Recall that -b was loaded in the stack
TRCL 01	
S/	; Calculates X1
TSTO 05	; Stores X1 in register 05
SPSE	; Pauses to display X1

SX<>Y	; Moves -b from Stack-Y to Stack-X register
TRCL 04	
S-	; Calculates -b - SQRT(b2 - 4ac)
N2	
S/	
TRCL 01	
S/	; Calculates X2
TSTO 06	; Stores X2 in register 06
SRTN	Could be included to allow use as a subroutine
SEND	; All programs must conclude with END statement

#### **Example 4-3: ASCII PROGRAM FILE**

(This page is intentionally left blank)



# PRINTER AND DISK OPERATIONS

# PRINTER UTILIZATION FUNCTIONS

The ELI-41 utilizes the capabilities of the printer attached to the computer by providing functions for organized listings of the results of the computation. Facilities are also included that help simplify tracing and debugging of the user's programs. Certain flags affect the printer operations and are discussed in the following section. These flags can be set or reset by the SF or CF functions. The ELI-41 automatically assumes that the printer is attached to LPT1 port of the computer and therefore all the printer operations are done on that port.

# FLAGS AFFECTING THE OPERATION OF THE PRINTER

The flags that affect the operation of the printer I/O are as follows:

# **FLAG 55:**

This flag is a system flag that ELI-41 controls it's state of being On or Off by checking the printer status each time a function is executed and at powerup, to ensure that the printer is connected and is on-line. If the printer is off or is off-line then FLAG 55 and FLAG 21 will be reset.

### FLAG 21:

This flag enables or disables the printer operations. It is programmable by the user to control the operation of the printer during program execution.

# FLAG 15:

This flag in conjunction with FLAG 16 will determine the mode that printer will be used.

# **FLAG 16:**

If FLAG 15 is SET and FLAG 16 is CLEAR then the printer will be in TRACE mode. In TRACE mode each function name and the Stack-X register is printed after the function is executed to show the results of that function. While in TRACE mode each program statement and its result is printed during program execution.

If FLAG 15 is SET and FLAG 16 is SET then the printer will be in TRACE mode with Stack option. In this mode each function that changes the contents of the Stack-X register will automatically print the function name and the contents of the stack and alpha register, whether in command mode or during program execution.

If FLAG 15 is CLEAR and FLAG 16 is CLEAR then the printer will be in normal mode and only functions such as VIEW, AVIEW and PROMPT would utilize the printer.

# **CHAPTER 6**

# **ON-LINE CONVERSION TABLES**

# WHAT IS A CONVERSION TABLE?

A conversion table is a standard ASCII text file which contains information to change a value from one unit of measure to another. ELI supplies several standard tables with the ELI-41 System diskette.

# **USING ELI-41'S CONVERSION TABLES**

The value to be converted is assumed to reside in the Stack-X register. That being true, the conversion table is loaded into memory by one of two methods.

1. The conversion key is pressed < F6>.

2. The catalog function is called, (the <N> key, located in the SHIFT CALC mode) and the CATALOG 5 function is chosen.

The calculator face will reveal the .CNV files resident in the CONVERSION directory. Using the <Cursor> keys to place the lightbar over the desired conversion file, the <Enter> key is pressed to signify the choice (Figure 6-1).



Figure 6-1: CATALOG 5 display

In the following example Boldface signifies user input. Assume the Stack-X register contains a value of 3.345 cm, and the result looked for being in feet, the calculator face would contain from the LENGTH.CNV file:



Figure 6-2: Conversion utility, CONVERT FROM screen



Figure 6-3: Conversion utility, CONVERT TO screen

Convert From **01** Convert To **03**.

The result rendered as 0.109743875 feet would be placed in the Stack-X register.

The calculator displays CONVERT FROM \_\_\_\_\_ on the command line, seeking a two digit number. The number entered represents the type value of the Stack-X register contents as represented by the types showing on the calculator face. Place holders are necessary, so type one (1) would be chosen as 01 (Figure 6-2).

The prompt CONVERT TO \_\_\_\_\_ is then displayed in the calculator input area, requesting a two digit number corresponding to the type of the resulting conversion. That number will replace the contents of the Stack-X register. NOTE: At any time before the final key press the conversion can be aborted or the conversion 'from' or 'to' can be edited by use of the gray <Back Space> key. Repeated entry of this key backs out of this function (Figure 6-3).

# MAKING YOUR OWN CONVERSION TABLES

In addition to the supplied tables, others may be generated with a word processor or text editor by the user. Up to 90 different tables can be accessed (within a single directory path). Each table can contain up to a 10 by 10 matrix. An example of a linear measure conversion table follows.

#### Table 6-1: CONVERSION WORK TABLE

CONVERT CONVERT FROM TO Inches Feet Yards Miles Inches 1 0.833333 0.277777778 0.0000157828 Feet 12.0 1.0 0.333333333 0.0001893939 Yards 36.0 3.0 0.0005681818 1.0 Miles 63360.0 5280.0 1760.0 1.0

The following represents a disk file derived from this conversion table. It would be created with the use of a text editor and saved to disk. Naming the file must be done according to standard DOS conventions. Consult your DOS manual for further information about DOS naming conventions.

The ELI-41SX product contains a built-in ASCII text editor that can be used to edit or create a conversion table file. For more information about QEDIT function please refer to chapter 8.

#### Table 6-2: CONVERSION TABLE FILE LAYOUT

(2); Note that comments always start with a semicolon(3); and may be included on statement lines.(4)4;Matrix Size(5)Inches;Description of Units(6)Feet;(7)Yards;(8)Miles;(9)1.0;Inches to Inches(10)0.833333;Inches to Feet(11)0.27777778;Inches to Yards(12)0.0000157828;Inches to Miles(13)12.0;Feet to Inches(14)1.0;Feet to Yards(15)0.333333333;Feet to Yards(16)0.0001893939;Feet to Miles(17)36.0;Yards to Inches(18)3.0;Yards to Yards(20)0.0005681818;Yards to Miles(21) $63360.0$ ;Miles to Inches(22)5280.0;Miles to Feet(23)1760.0;Miles to Yards(24)1.0;Miles to Miles	(1)	; Linear Conv	ersion Table
(3); and may be included on statement lines.(4)4; Matrix Size(5)Inches; Description of Units(6)Feet;(7)Yards;(8)Miles;(9)1.0; Inches to Inches(10)0.833333; Inches to Feet(11)0.27777778; Inches to Yards(12)0.0000157828; Inches to Miles(13)12.0; Feet to Inches(14)1.0; Feet to Feet(15)0.333333333; Feet to Yards(16)0.0001893939; Feet to Miles(17)36.0; Yards to Inches(18)3.0; Yards to Yards(20)0.0005681818; Yards to Miles(21) $63360.0$ ; Miles to Inches(22)5280.0; Miles to Yards(23)1760.0; Miles to Yards(24)1.0; Miles to Miles	(2)	; Note that con	mments always start with a semicolon
(4)       4       ;Matrix Size         (5)       Inches       ;Description of Units         (6)       Feet       ;         (7)       Yards       ;         (8)       Miles       ;         (9)       1.0       ;Inches to Inches         (10)       0.833333       ;Inches to Feet         (11)       0.277777778       ;Inches to Yards         (12)       0.0000157828       ;Inches to Miles         (13)       12.0       ;Feet to Inches         (14)       1.0       ;Feet to Feet         (15)       0.333333333       ;Feet to Yards         (16)       0.0001893939       ;Feet to Miles         (17)       36.0       ;Yards to Inches         (18)       3.0       ;Yards to Yards         (20)       0.0005681818       ;Yards to Miles         (21)       63360.0       ;Miles to Inches         (22)       5280.0       ;Miles to Feet         (23)       1760.0       ;Miles to Yards         (24)       1.0       ;Miles to Miles	(3)	; and may be i	ncluded on statement lines.
(5)       Inches       ;Description of Units         (6)       Feet       ;         (7)       Yards       ;         (8)       Miles       ;         (9)       1.0       ;Inches to Inches         (10)       0.833333       ;Inches to Feet         (11)       0.277777778       ;Inches to Yards         (12)       0.0000157828       ;Inches to Miles         (13)       12.0       ;Feet to Inches         (14)       1.0       ;Feet to Feet         (15)       0.3333333333       ;Feet to Yards         (16)       0.0001893939       ;Feet to Miles         (17)       36.0       ;Yards to Inches         (18)       3.0       ;Yards to Yards         (20)       0.0005681818       ;Yards to Yards         (21)       63360.0       ;Miles to Inches         (22)       5280.0       ;Miles to Feet         (23)       1760.0       ;Miles to Yards         (24)       1.0       ;Miles to Miles	(4)	4	;Matrix Size
(6)       Feet       ;         (7)       Yards       ;         (8)       Miles       ;         (9)       1.0       ;Inches to Inches         (10)       0.833333       ;Inches to Feet         (11)       0.277777778       ;Inches to Yards         (12)       0.0000157828       ;Inches to Miles         (13)       12.0       ;Feet to Inches         (14)       1.0       ;Feet to Feet         (15)       0.333333333       ;Feet to Yards         (16)       0.0001893939       ;Feet to Miles         (17)       36.0       ;Yards to Inches         (18)       3.0       ;Yards to Yards         (20)       0.0005681818       ;Yards to Yards         (21)       63360.0       ;Miles to Inches         (22)       5280.0       ;Miles to Feet         (23)       1760.0       ;Miles to Yards         (24)       1.0       ;Miles to Miles	(5)	Inches	;Description of Units
(7)Yards;(8)Miles;(9)1.0;Inches to Inches(10)0.833333;Inches to Feet(11)0.27777778;Inches to Yards(12)0.0000157828;Inches to Miles(13)12.0;Feet to Inches(14)1.0;Feet to Feet(15)0.33333333;Feet to Yards(16)0.0001893939;Feet to Miles(17)36.0;Yards to Inches(18)3.0;Yards to Yards(20)0.0005681818;Yards to Miles(21)63360.0;Miles to Inches(22)5280.0;Miles to Yards(23)1760.0;Miles to Yards(24)1.0;Miles to Miles	(6)	Feet	•
(8)       Miles       ;         (9)       1.0       ;Inches to Inches         (10)       0.833333       ;Inches to Feet         (11)       0.277777778       ;Inches to Yards         (12)       0.0000157828       ;Inches to Miles         (13)       12.0       ;Feet to Inches         (14)       1.0       ;Feet to Feet         (15)       0.3333333333       ;Feet to Yards         (16)       0.0001893939       ;Feet to Miles         (17)       36.0       ;Yards to Inches         (18)       3.0       ;Yards to Yards         (20)       0.0005681818       ;Yards to Miles         (21)       63360.0       ;Miles to Inches         (22)       5280.0       ;Miles to Feet         (23)       1760.0       ;Miles to Yards         (24)       1.0       ;Miles to Miles	(7)	Yards	•
(9)1.0;Inches to Inches(10)0.833333;Inches to Feet(11)0.27777778;Inches to Yards(12)0.0000157828;Inches to Miles(13)12.0;Feet to Inches(14)1.0;Feet to Feet(15)0.333333333;Feet to Yards(16)0.0001893939;Feet to Miles(17)36.0;Yards to Inches(18)3.0;Yards to Yards(20)0.0005681818;Yards to Miles(21)63360.0;Miles to Inches(22)5280.0;Miles to Feet(23)1760.0;Miles to Yards(24)1.0;Miles to Miles	(8)	Miles	, ,
(10)0.833333;Inches to Feet(11)0.27777778;Inches to Yards(12)0.0000157828;Inches to Miles(13)12.0;Feet to Inches(14)1.0;Feet to Feet(15)0.333333333;Feet to Yards(16)0.0001893939;Feet to Miles(17)36.0;Yards to Inches(18)3.0;Yards to Feet(19)1.0;Yards to Yards(20)0.0005681818;Yards to Miles(21)63360.0;Miles to Inches(23)1760.0;Miles to Yards(24)1.0;Miles to Miles	(9)	1.0	;Inches to Inches
(11)0.27777778;Inches to Yards(12)0.0000157828;Inches to Miles(13)12.0;Feet to Inches(14)1.0;Feet to Feet(15)0.333333333;Feet to Yards(16)0.0001893939;Feet to Miles(17)36.0;Yards to Inches(18)3.0;Yards to Feet(19)1.0;Yards to Yards(20)0.0005681818;Yards to Miles(21)63360.0;Miles to Inches(23)1760.0;Miles to Yards(24)1.0;Miles to Miles	(10)	0.833333	;Inches to Feet
(12)0.0000157828;Inches to Miles(13)12.0;Feet to Inches(14)1.0;Feet to Feet(15)0.333333333;Feet to Yards(16)0.0001893939;Feet to Miles(17)36.0;Yards to Inches(18)3.0;Yards to Feet(19)1.0;Yards to Yards(20)0.0005681818;Yards to Miles(21)63360.0;Miles to Inches(22)5280.0;Miles to Feet(23)1760.0;Miles to Yards(24)1.0;Miles to Miles	(11)	0.277777778	;Inches to Yards
(13)12.0;Feet to Inches(14)1.0;Feet to Feet(15)0.333333333;Feet to Yards(16)0.0001893939;Feet to Miles(17)36.0;Yards to Inches(18)3.0;Yards to Feet(19)1.0;Yards to Yards(20)0.0005681818;Yards to Miles(21)63360.0;Miles to Inches(22)5280.0;Miles to Feet(23)1760.0;Miles to Yards(24)1.0;Miles to Miles	(12)	0.0000157828	;Inches to Miles
(14)1.0;Feet to Feet(15)0.333333333;Feet to Yards(16)0.0001893939;Feet to Miles(17)36.0;Yards to Inches(18)3.0;Yards to Feet(19)1.0;Yards to Yards(20)0.0005681818;Yards to Miles(21)63360.0;Miles to Inches(22)5280.0;Miles to Feet(23)1760.0;Miles to Yards(24)1.0;Miles to Miles	(13)	12.0	;Feet to Inches
(15)0.3333333333;Feet to Yards(16)0.0001893939;Feet to Miles(17)36.0;Yards to Inches(18)3.0;Yards to Feet(19)1.0;Yards to Yards(20)0.0005681818;Yards to Miles(21)63360.0;Miles to Inches(22)5280.0;Miles to Feet(23)1760.0;Miles to Yards(24)1.0;Miles to Miles	(14)	1.0	;Feet to Feet
(16)0.0001893939;Feet to Miles(17)36.0;Yards to Inches(18)3.0;Yards to Feet(19)1.0;Yards to Yards(20)0.0005681818;Yards to Miles(21)63360.0;Miles to Inches(22)5280.0;Miles to Feet(23)1760.0;Miles to Yards(24)1.0;Miles to Miles	(15)	0.3333333333	;Feet to Yards
(17)36.0;Yards to Inches(18)3.0;Yards to Feet(19)1.0;Yards to Yards(20)0.0005681818;Yards to Miles(21)63360.0;Miles to Inches(22)5280.0;Miles to Feet(23)1760.0;Miles to Yards(24)1.0;Miles to Miles	(16)	0.0001893939	;Feet to Miles
(18)3.0;Yards to Feet(19)1.0;Yards to Yards(20)0.0005681818;Yards to Miles(21)63360.0;Miles to Inches(22)5280.0;Miles to Feet(23)1760.0;Miles to Yards(24)1.0;Miles to Miles	(17)	36.0	;Yards to Inches
(19)1.0;Yards to Yards(20)0.0005681818;Yards to Miles(21)63360.0;Miles to Inches(22)5280.0;Miles to Feet(23)1760.0;Miles to Yards(24)1.0;Miles to Miles	(18)	3.0	;Yards to Feet
(20)         0.0005681818         ;Yards to Miles           (21)         63360.0         ;Miles to Inches           (22)         5280.0         ;Miles to Feet           (23)         1760.0         ;Miles to Yards           (24)         1.0         ;Miles to Miles	(19)	1.0	;Yards to Yards
(21)       63360.0       ;Miles to Inches         (22)       5280.0       ;Miles to Feet         (23)       1760.0       ;Miles to Yards         (24)       1.0       ;Miles to Miles	(20)	0.0005681818	;Yards to Miles
(22)         5280.0         ;Miles to Feet           (23)         1760.0         ;Miles to Yards           (24)         1.0         ;Miles to Miles	(21)	63360.0	;Miles to Inches
(23)         1760.0         ;Miles to Yards           (24)         1.0         ;Miles to Miles	(22)	5280.0	;Miles to Feet
(24) 1.0 ;Miles to Miles	(23)	1760.0	;Miles to Yards
	(24)	1.0	;Miles to Miles

# **RULES FOR CODING CONVERSION TABLES**

#### 1. Conversion table file names:

Conversion Table files must be named with a ".CNV" file extension.

#### 2. Matrix sizes:

Maximum of a 10 by 10 matrix is allowed.

# 3. Required Statements.

A: Matrix Size: This statement defines the size of the matrix.

B: Descriptions: There must be the exact number of description statements as the number specified in the Matrix Size statement. Description names greater than seven ASCII characters long will be truncated for display purposes.

C: Conversion Factors: The number of conversion factor statements must be the value of the Matrix Size statement squared. Values may be recorded in either decimal format or scientific notation.

# 4. Line lengths:

The length of either a Comment Line or a Statement Line cannot exceed 79 characters.

# 5. Comment lines:

Comments within the file always begin with a semicolon. There are two types of comments. Comment lines and comment statements. A Comment Line is an individual record within the file containing a semicolon in position 1 of the record followed by any text. A Comment Statement is any text preceded by a semicolon and a space on any of the required statement records. A space or tab character must always separate the required statement's value and the semicolon.

# **CHAPTER 7**

# **OPTIONS AND DEFAULT SETTINGS**

# **OPTIONS MENU**

You invoke the Options Menu from the calculator display screen by pressing <F5>. This system allows you to make changes to ELI-41 to fit your machine environment and preferences. The highlighting, intensity and characteristics of various elements of the display can be changed from the defaults set at startup for the monochrome screen, as an instance. The colors, of course, can be changed for a color card and monitor.



Figure: 7-1 OPTIONS MENU TREE

The display is changed to a rectangular window containing a menu selection line and a description line which describes each function of the menu. This description is of the presently highlighted item.

You can make a selection in either of two ways. The first, overlay the menu item with the highlight bar using the <Cursor> direction keys and press <Enter>. And the second, press the first letter of any menu item. The program treats this just as if you had moved the highlight bar over the menu item and had pressed <Enter>.

This menu system is multi-layered, meaning, a selection may take you to a nested level of menus. At any time you may press the <Esc>key to return to the preceding menu. To return to the calculator display screen simply press the <Esc> key or select Quit from the top menu level.

# **SETTING OPTIONS**

The following is a detailed description of the various levels to the OPTIONS screen:

# Catalog

This main menu option obtains directory listings of files or internal functions according to the following levels.

#### **All**

Provides a catalog of all files from the "Utility Path" regardless of file extension. Sub directories will also be listed.

#### Programs

Provides a catalog of ELI-41 programs contained in the "Program Path" with file extension of ".PGM".

#### Conversion

Provides a catalog of Conversion table files in the "Program Path" with file extension ".CNV". (ELI-41)

Provides a catalog of Conversion table files in the "Conversion Path" with file extension ".CNV". (ELI-41SX)

#### Memory

Provides a catalog of ELI-41 "Dumped Memory" files in the "Memory Path" with file extension ".MEM". (ELI-41)

Provides a catalog of ELI-41 "Dumped Memory" files in the "Memory Path" with file extensions of ".MRG", ".MST", ".MKY". (ELI-41SX)

#### **Internal Functions**

Displays a list of all functions internal to ELI-41/41SX.

#### Quit

Returns to the previous menu level.

# Print

Activates and deactivates printer. This turns on or off the printer flag 21.

#### ON/OFF

Toggles the printer on or off. On the lower edge of the calculator face, an annunciator will inform of the condition of printer.

#### File (ELI-41SX)

Redirects the printer output to an ASCII text file name ELI41SX.PRT in the default Print path.

#### 1 (ELI-41SX)

Redirects the printer output to LPT1.

#### 2 (ELI-41SX)

Redirects the printer output to LPT2.

#### 3 (ELI-41SX)

Redirects the printer output to LPT3.

#### Quit

Returns to the previous menu level.

#### Memory

The memory management features the ability to save the present state of all registers and flags for reinstatement at a later time.

#### Dump

Saves or prints ELI-41 memory contents. The Stack registers and flags are saved as they are but only memory registers whose contents are not zero are saved. Any program resident in memory is not saved.

#### File

Saves the contents of memory to a disk file whose name must be specified in "Options/Filespecs". If

none has been supplied the default of ELI.MEM is used.

Saves the contents of data register, status, and all user key assignments in ELI-41.MRG, ELI-41.MST, and ELI-41.MKY files accordingly. This is similar to executing the WRTA function. (ELI-41SX)

#### Printer

Prints the memory contents, Stack, memory registers, and flags. Does not depend on the state of the print flag (21), but a printer must have been on-line when the calculator was loaded. A status check on the printer was done at that time.

#### Quit

Returns to the previous menu level.

#### Restore

Retrieves a previously dumped memory file ("?.MEM") from disk and places it in memory. If no other file name has been supplied on the "Options/Filespecs" level, the default file name of ELI.MEM is loaded, if existent.

#### No

States no, do not retrieve file, do not disturb the contents of memory.

#### Yes

Confirms, continue with operation, replace memory with data contained in the file.

#### Initialize

No

States no, do not initialize data or Stack registers to zero.

Yes

Confirms, continue with operation, initialize data and Stack registers to zero.

#### Quit

Returns to the previous menu level.

# Options

Miscellaneous environment settings are supplied on this level. They take effect immediately.

#### Colors

An equipment check was made at startup. If it showed no color equipment, changing colors will have no effect. The monochrome attributes , of course, would.

#### Monochrome

Changes attributes for use of a monochrome monitor. Select the screen display area you want to change (selections "A" through "N" on the right side of the screen) and you will be prompted to select a color attribute. The possible attributes are:

A - Normal Intensity B - Bright Intensity

- C Reverse Highlight
- D Black

#### Color

Change attributes for use with a color monitor. Select the screen display area you want to change (selections "A" through "N") and you will be prompted to select a color attribute. The available colors are:

- A Black
- B Blue
- C Green
- D Cyan
- E Red
- F Magenta
- G Brown
- H Light Gray
- I Dark Gray
- J Light Blue
- K Light Green
- L Light Cyan
- M Light Red
- N Light Magenta
- O Yellow
- P White

Note: Background color cannot be changed from Black. Any color attribute identified by "???" is a special combination of a foreground and background color. A combination of foreground and background colors (i.e.: Blue Foreground on a Red Background) cannot be accomplished.

#### Filespecs

Allows changing directory paths for program, utility, conversion and memory files. The file name for the memory file is specified here as well.

#### Utility Directory

A general purpose directory path can be supplied to display a catalog of any drive and path.

#### **Program Directory**

A directory path must be supplied for accessing ELI-41 Programs (".PGM" files) and Conversion Tables (".CNV" files).

#### **Memory Directory**

The Directory path must be supplied to access saved memory files (".MEM"). This is used for saving or restoring ELI-41 memory.

#### Program File Name

The name of the most current program file loaded, is displayed.

#### Memory File Name

The name of the memory file to either save to or restore from must be supplied. A default of ELI.MEM is provided.

#### Load

Requests the loading of a disk option file ("ELI41.OPT") from here.

#### No

Do not load options file, use option settings currently in memory.

#### Yes

If ELI41.OPT is existent in the memory file directory, that file will be loaded into memory. The replacement takes the place of the options that are currently in memory.

If ELI41SX.OPT is existent in the current directory it is loaded and the replacement takes the place of the options that are currently in memory. (ELI-41SX)

#### Save

Requests the saving of the current options to an options file ("ELI41.OPT") on disk. The current directory for memory files is used.

Requests the saving of the current options to an options file ("ELI41SX.OPT") on disk. The directory where the program was loaded from is used as the default directory. (ELI-41SX)

No

Do not save options to disk.

#### Yes

Saves the current options to disk, overwriting any existing "ELI41.OPT" file. The next time ELI-41 is started these newly saved options will become the defaults.

#### Quit

Returns to the calculator display screen.

(This page is intentionally left blank)

# **CHAPTER 8**

# **THE ELI-41 FUNCTIONS**

This chapter contains the description of all mathematical and miscellaneous functions that have been implemented on the ELI-41/41SX scientific calculator software packages. Each function is presented in the following format:

NAME XXXXXXX KEYPRESS STACK LAST-X DESCRIPTION:

**RELATED FUNCTIONS:** 

The NAME section contains the executable name of the function and KEYPRESS is the keyboard <key> within the proper mode to activate the function. Almost all functions can be executed using the XEQ function or be executed through the CATALOG 3 two column function directory.

The STACK section gives clues as to the action performed on the 5 element stack (X,Y,Z,T,L) of the calculator such as enabling the stack lift, which would cause the stack to be pushed before the next number entry is made. The LAST-X section shows the assignment action to the Last-X register of the stack.

The DESCRIPTION section is a brief description of the function and some clues to how it could be useful in the programming facility of the calculator. Finally the RELATED FUNCTIONS section provides the name of functions that have similar properties.

NAME	%
KEYPRESS	<g> (SHIFT CALC mode)</g>
STACK	Enables Stack-lift
LAST-X	Precomputation Stack-X
DESCRIPTION:	-

The PERCENT function uses the Stack-X register as the percent rate and the Stack-Y register as the base number and returns in The Stack-X register the Stack-X percentage of the Stack-Y register. The contents of the Stack-Y register remains unchanged.

RELATED FUNCTIONS: %CH

NAME	%CH
KEYPRESS	(XEQ or CATALOG 3)
STACK	Enables Stack-lift
LAST-X	Precomputational Stack-X
DESCRIPTION:	-

The PERCENT CHANGE function calculates the percent increase or decrease from the contents of Stack-Y to the contents of the Stack-X registers.

(Stack-X = ((Stack-X - Stack-Y) \* 100) / Stack-Y)

**RELATED FUNCTIONS: %** 

NAME	1/X
KEYPRESS	<b> (CALC Mode)</b>
STACK	Enables Stack-lift
LAST-X	Precomputation Stack-X
DESCRIPTION:	-

The RECIPROCAL function calculates the reciprocal of the Stack-X register and places the result in the Stack-X register, thus overwriting its previous contents.

NAME KEYPRESS STACK LAST-X DESCRIPTION:

10↑X <D> (SHIFT CALC Mode) Enables Stack-lift Precomputation Stack-X

The COMMON EXPONENTIAL function returns in the Stack-X register the value of 10 raised to the power of the Stack-X register. This is the reverse action of the COMMON LOGARITHM.

**RELATED FUNCTIONS:** LOG, LN,  $E \uparrow X$ ,  $E \uparrow X$ -1, LN1+X

NAME	<<
KEYPRESS	(XEQ or CATALOG 3)
STACK	Enables Stack-lift, Allows indirection
LAST-X	Precomputational Stack-X
DESCRIPTION:	-

The SHIFT LEFT 'n' function shifts the integer portion of the Stack-X register left by 'n' bits.

RELATED FUNCTIONS: >>, AND, OR, NOT, XOR

NAME	>>
KEYPRESS	(XEQ or CATALOG 3)
STACK	Enables Stack-lift
LAST-X	Precomputational Stack-X
DESCRIPTION:	-

The SHIFT RIGHT 'n' function shifts the integer portion of the Stack-X register right by 'n' bits.

RELATED FUNCTIONS: <<, AND, OR, XOR, NOT

8-3

NAME	ABS
KEYPRESS	(XEQ or CATALOG 3)
STACK	Enables Stack-lift
LAST-X	Precomputational Stack-X
DESCRIPTION:	-

The ABSOLUTE function converts the Stack-X register to a positive number.

NAME	ACA
KEYPRESS	(XEQ or CATALOG 3)
STACK	Enables Stack-lift
LAST-X	Not applicable
DESCRIPTION:	

The ACCUMULATE ALPHA function sends the entire Alpha-register to the printer's line buffer. Note that this string is appended to the string that is already in the printer's line buffer and will not be printed unless either PRBUF or ADV command is issued.

NAME	ACCBYTE
KEYPRESS	(XEQ or CATALOG 3)
STACK	<b>Enables Stack-lift</b>
LAST-X	Not applicable
DESCRIPTION:	

The ACCumulate BYTE function converts the integer portion of the Stack-X register in to an ASCII character and then appends it to the Alpha register. It could be used to place ASCII characters that are not available through the ALPHA mode into the Alpha register.

#### Example:

"APPEND 2 M	ORE"
25	; 0X19 Hexadecimal (SPD)
ACCBYTE	; Append integer portion of Stack-X to the Alpha register
95	; ('_') UNT
ACCBYTE	; Append integer portion of Stack-X to the Alpha register

RELATED FUNCTIONS: ATOX, XTOA, POSA, ALENG, ATOF

NAME
KEYPRESS
STACK
LAST-X
DESCRIPTION:

ACCHR (XEQ or CATALOG 3) Enables Stack-lift Not applicable

The ACCUMULATE CHARACTER function sends the integer portion of Stack-X register ( $0 \le X \le 255$ ) to the printer's line buffer. This character is appended to the contents of the line buffer and will not be printed unless either PRBUF or ADV command is issued.

NAME	ACOS
KEYPRESS	<i> (SHIFT CALC Mode)</i>
STACK	Enables Stack-lift
LAST-X	Precomputation Stack-X
DESCRIPTION:	-

The ARCCOSINE function computes an angle in which the Stack-X register is the COSINE of. The angle is computed based on the current trigonometric angle mode (DEG, RAD, GRAD) and the result is placed in the Stack-X register.

RELATED FUNCTIONS: SIN, COS, TAN, ASIN, ATAN

NAME	ACX
KEYPRESS	(XEQ or CATALOG 3)
STACK	Enables Stack-lift
LAST-X	Not applicable
DESCRIPTION:	

The ACCUMULATE Stack-X register function sends a string which represents the contents of the Stack-X register according to the current display format and number of digits to the printer's line buffer.

RELATED FUNCTIONS: ACA

NAME	ADV
KEYPRESS	(XEQ or CATALOG 3)
STACK	Enables Stack-lift
LAST-X	Not applicable
DESCRIPTION:	

The ADVANCE function sends a carriage return, line feed to the printer's line buffer which causes the contents of the line buffer to be printed. The line feed causes the paper to advance one line.

NAME	ALENG (ELI-41SX)
KEYPRESS	(XEQ or CATALOG 3)
STACK	Enables Stack-lift
LAST-X	Precomputation Stack-X
DESCRIPTION:	-

The ALPHA LENGTH function computes the length of the text string in the Alpha-register by counting all the characters in it and places the resulting number in the Stack-X register, lifting the stack.

#### Example:

1) "ABCDEFGHIJKLMO" 2) ALENG ; After execution of ALENG function the Stack-X register ; will contain 14.0

NAME	AND
KEYPRESS	(XEQ or CATALOG 3)
STACK	Enables Stack-lift
LAST-X	Precomputational Stack-X
DESCRIPTION:	-

The AND function performs a boolean AND on the integer portion of Stack-X and Stack-Y registers and places the result in the Stack-X register.

RELATED FUNCTIONS: XOR, OR, NOT, <<, >>
NAME	ANUM (ELI-41SX)
KEYPRESS	(XEQ or CATALOG 3)
STACK	Lifts stack, Enables Stack-lift
LAST-X	Precomputation Stack-X
DESCRIPTION:	-

The ANUM function searches the Alpha register for a number and brings that number into the Stack-X register, lifting the stack first if stack lift is enabled.

#### Example:

1) "TEMP = 45.23"? 2) ANUM ; After execution of ANUM function the Stack-X register ; will contain 45.23

NAME	AOFF
KEYPRESS	(XEQ or CATALOG 3)
STACK	Enables Stack-lift
LAST-X	Not applicable
DESCRIPTION:	

The ALPHA OFF function turns the ALPHA mode OFF, and switches to the mode previous to going into ALPHA mode.

RELATED FUNCTIONS: AON

NAME	AON
KEYPRESS	(XEQ or CATALOG 3)
STACK	Enables Stack-lift
LAST-X	Not applicable
DESCRIPTION:	

The ALPHA ON function switches into the ALPHA mode (flag 48). It functions similar to the  $\langle F9 \rangle$  key, however it could be called from within user programs.

RELATED FUNCTIONS: AOFF

NAME	APP
KEYPRESS	<k> (SHIFT ALPHA Mode)</k>
STACK	Not applicable
LAST-X	Not applicable
DESCRIPTION:	

The APPend to alpha function allows appending additional characters to the Alpha-register. After executing the function any other character entered will be appended to the Alpha-register.

NAME	ARCL
KEYPRESS	<m> (SHIFT ALPHA Mode)</m>
STACK	Enables Stack-lift, Allows indirection
LAST-X	Not applicable
DESCRIPTION:	

The ALPHA RECALL 'nn' function appends the contents of the register 'nn' to the Alpha-register. If the contents of the memory register is not alphanumeric then it will be converted to alphanumeric characters based on the current display format (FIX, ENG, SCI).

RELATED FUNCTIONS: ASTO, STO, AVIEW, etc.

NAME	AROT (ELI-41SX)
KEYPRESS	(XEQ or CATALOG 3)
STACK	Enables Stack-lift
LAST-X	Not applicable
DESCRIPTION:	

Rotates the characters in the Alpha-register left or right by the number of places given in the Stack-X register. If Stack-X contains a positive number, the rotation is to the left, and if the number is negative, the rotation is to the right.

# Example:

1) "SCAT" 2) 1 3) AROT ; After execution of AROT function the Alpha-register ; will contain "CATS".

RELATED FUNCTIONS: ATOX, XTOA, ALENG, POSA

NAME	ASHF
KEYPRESS	(XEQ or CATALOG 3)
STACK	Enables Stack-lift
LAST-X	Not applicable
DESCRIPTION:	

The ALPHA SHIFT function shifts the Alpha-register contents left six (6) characters. The first six (6) characters are thus shifted out of the Alpha-register. It is used to store contents of the Alpha-register in the data registers, because they can only hold up to six characters at a time.

RELATED FUNCTIONS: ASTO

NAME	ASIN
KEYPRESS	<h> (SHIFT CALC Mode)</h>
STACK	Enables Stack-lift
LAST-X	Precomputation Stack-X
DESCRIPTION:	-

The ARCSINE function computes an angle in which the Stack-X register is the SINE of. The angle is computed based on the current trigonometric angle mode (DEG, RAD, GRAD) and the result is placed in the Stack-X register.

RELATED FUNCTIONS: SIN, COS, TAN, ACOS, ATAN

NAME	ASN
KEYPRESS	<k> (SHIFT CALC Mode)</k>
STACK	Enables Stack-lift
LAST-X	Not applicable
DESCRIPTION:	

The ASSIGN LABEL TO KEY function is used to assign a particular function or program name to a specific key to be used in the USER mode. Any of the calculator functions or user programs which reside in the current program directory can be assigned to the available calculator keys.

Executing the ASSIGN function will Prompt the user to enter a 7 character alpha name representing the function or program name, followed by the key to which the function will be assigned. The user must press the  $\langle F9 \rangle$  (ALPHA) key in response to ASN \_\_ message to change in to alpha mode and enter up to seven characters followed by either the  $\langle Enter \rangle$  key or the  $\langle F9 \rangle$  (ALPHA) key. The calculator in turn will respond with the ASN \*\*\*\*\*\* \_ message requesting a single key to which it will assign the function to. User has the option of assigning functions or program names to SHIFT USER mode keys by pressing the  $\langle F10 \rangle$  (SHIFT) key before the single assignable key is entered.

As an example the CLST function could be assigned to the  $\langle B \rangle$  key in the USER or the SHIFT USER modes by pressing the following keystrokes:

```
<F10> <K>
<F9>
<"CLST"> : ASN CLST_
<F9> or <Enter> key : ASN CLST_
<B> : Assign "CLST" to <B> KEY (USER mode)
or <F10> <B> : <B> KEY (SHIFT USER mode)
```

RELATED FUNCTIONS: None

NAME	ASTO
KEYPRESS	<l> (SHIFT ALPHA Mode)</l>
STACK	Enables Stack-lift, Allows indirection
LAST-X	Not applicable
DESCRIPTION:	

The ALPHA STORE 'nn' function will copy the first six (6) characters of the Alpha-register into the memory register 'nn'.

RELATED FUNCTIONS: ASHF, ARCL, STO, RCL, AVIEW

NAME	ATAN
KEYPRESS	<j> (SHIFT CALC Mode)</j>
STACK	Enables Stack-lift
LAST-X	Precomputation Stack-X
DESCRIPTION:	

The ARCTANGENT function computes an angle in which the Stack-X register is the TANGENT of. The angle is computed based on the current trigonometric angle mode (DEG, RAD, GRAD) and the result is placed in the Stack-X register.

RELATED FUNCTIONS: SIN, COS, TAN, ASIN, ACOS

NAME	ATOF
KEYPRESS	(XEQ or CATALOG 3)
STACK	Enables Stack-lift
LAST-X	Precomputation Stack-X
DESCRIPTION:	-

The ATOF function converts the character string contained in the Alpha register into a double precision floating point number and stores it in the Stack-X register. The character string must be a sequence of ASCII characters that can be interpreted as a numerical value of the floating point type.

The conversion stops at the first character that is not recognized as part of a number. The ATOF function expects the string to have the following form: [whitespace][sign][digits][.digits]{d|D|e|E}[sign and digits]

A whitespace character consists of space and/or tab character, which are ignored; sign is either '+' or '-'; and digits are one or more decimal digits. If no digits appear before the decimal point, at least one must appear after the decimal point. The decimal digits may be followed by an exponent, which consists of an introductory letter (d,D,e, or E) and an optional signed decimal integer.

# Example:

"OVEN1" IBFIND "TEMP?"	; Select the OVEN1
IBWRT	; Ask for current oven temperature
IBRD	; read temperature
ATOF	; convert to float
; now the Sta	ack-X contains the temperature in floating point
; format	
•	

RELATED FUNCTIONS: ACCBYTE, XTOA, ATOX, CHG2ASC

NAME	ATOX (ELI-41SX)
KEYPRESS	(XEQ or CATALOG 3)
STACK	Enables Stack-lift
LAST-X	Not applicable
DESCRIPTION:	

The ATOX function removes the leftmost character from the Alpha-register, interprets it as a number between 1 to 255 and places the result into the Stack-X register lifting the stack if stack-lift is enabled.

# Example:

1) "HELLO" 2) 38 ; "& CHARACTER" 3) XTOA ; After execution of XTOA function the Alpha-register ; will contain "HELLO&".

RELATED FUNCTIONS: XTOA, AROT, POSA, ALENG

NAME	AVIEW
KEYPRESS	<>> (SHIFT ALPHA Mode)
STACK	Enables Stack-lift
LAST-X	Not applicable
DESCRIPTION:	

The ALPHA VIEW 'nn' function prints contents of Alpha-register if flag 21 is set and the printer is on, otherwise it will halt program execution and display the current content of the alpha register.

RELATED FUNCTIONS: VIEW

NAME	Back Space
KEYPRESS	<back space=""> (CALC Mode)</back>
STACK	Disables Stack-lift with no previous entries
LAST-X	Not applicable
DESCRIPTION:	

The <Back Space> key is used for editing the current number or alpha entry in the data entry area. Repeated pressing of this key will remove the last entry character from the display area until the display is cleared; at which time, the Stack-X register will be cleared to zero.

RELATED FUNCTIONS: CLX, CLA

NAME	BEEP
KEYPRESS	<v> (SHIFT CALC</v>
STACK	Enables Stack-lift
LAST-X	Not applicable
DESCRIPTION:	

The BEEP function sounds 4 tones with frequencies of approximately 525, 394, 788, and 629 Hertz.

Mode)

RELATED FUNCTIONS: TONE

NAME	BIN
KEYPRESS	(XEQ or CATALOG 3)
STACK	Enables Stack-lift
LAST-X	Not applicable
DESCRIPTION:	

The BINARY function will change the present data entry and display modes to binary. The acceptable entry digits in this mode are 0 and 1. Refer to Chapter 3 for further information.

RELATED FUNCTIONS: HEX, DEC, OCT

NAME	BREAK
KEYPRESS	(XEQ or CATALOG 3)
STACK	Not Applicable
LAST-X	Not Applicable
DESCRIPTION:	

The BREAK function is similar to the STOP function, except that stack\_lift is not enabled. It could be used to set breakpoints in the program without disturbing stack\_lift status.

RELATED FUNCTIONS: STOP, END

NAME
KEYPRESS
STACK
LAST-X
DESCRIPTION:

CAT (CATALOG) <N> (SHIFT CALC Mode) Not applicable Not applicable

The CATALOG 'n' function is used to view the various Internal functions of the calculator, Program files or the Conversion table files. The user also has the option to run any of functions or programs by using the <Cursor> keys to place the highlight bar over the desired function name and pressing the <Enter> key to perform the function or to execute the desired program.

The CATALOG function will prompt the user to enter a single digit by displaying the CATALOG \_ message. A single digit between 1 and 5 must be entered. The associated modes for the response to CATALOG function are as follows:

- 1. Program files.
- 2. Program files.
- 3. Internal Functions of the ELI-41 calculator.
- 4. Various Memory files. (status, keys, data registers)
- 5. Conversion table files.

An entry of one of the above numbers will display a two column directory of the associated names. The lightbar will reside on the first entry and could be moved to view the rest of the names by any of the <Cursor> keys.

Pressing the <Enter> key with the lightbar on a function or program file executes that function or program. The <Esc> key simply clears the display and returns to the previous display.

RELATED FUNCTIONS: XEQ

NAME	CF
KEYPRESS	<s> (SHIFT CALC Mode)</s>
STACK	Enables Stack-lift, Allows indirection
LAST-X	Not applicable
DESCRIPTION:	

The CLEAR FLAG 'nn' function is used to Clear flags 00 - 29. For example, to Clear the printer flag (21) execute the CF function by pressing  $\langle$ F10 $\rangle$  (SHIFT)  $\langle$ S $\rangle$  and then enter 21 in response to the CF \_ prompt.

RELATED FUNCTIONS: SF, FC?, FC?C, FS?, FS?C

NAME	CHG2ASC
KEYPRESS	(XEQ or CATALOG 3)
STACK	Enables Stack-lift
LAST-X	Not applicable
DESCRIPTION:	

The CHG2ASC function is used to convert the contents of the Alpha register to ASCII representation. This function would be useful in a case where an instrument has sent bytes that are not ASCII characters in response to the IBRD function.

An example for this situation would be the PPG3500 programmable pressure generator manufactured by the SCHWIEN ENGINEERING INC., it sends the current pressure transducer frequency back in a string of byte values such as 0020040030000100506 for frequency = 2430156. By calling the CHG2ASC function after the above number is received in the Alpha register, the contents of the Alpha register will be converted to "2430156" ASCII so that it could be converted into a floating point number using the ATOF function.

### Example:

LBL "PPG3500" ;S	CHWIEN PROGRAMMABLE PRESSURE
SF 17	;GENERATOR 3500-500A-IEEE
SF 18	;FLAGS 17 AND 18 ON = NULL EOS
"PPG1"	;FIND PPG2 ON THE ADDRESS TABLE
IBFIND	

IBCLR		
CLA		
2		
ACCBYTE IBWRT	; READ F	PERIOD COUNTER MODE
IBRD CHG2ASC	; RECEIV	/E 7 BYTES FROM PPG3500
; Stack-X now	contains the f	requency counter reading
RELATED FU	NCTIONS:	ATOX, XTOA, ALENG, ATOF, ACCBYTE

NAME	CHS
KEYPRESS	<o> (CALC Mode)</o>
STACK	Enables Stack-lift
LAST-X	Not applicable
DESCRIPTION:	

The CHANGE SIGN function can be used for three different purposes:

- 1. At any point after entry of a digit into the data entry area, the number currently there can have a sign change of negative to positive, or positive to negative.
- 2. After any operation that terminates a digit entry, the present contents of the Stack-X register can be switched from negative to positive, or from positive to negative.
- 3. This is the only facility for entering a negative exponential value. This must be used anytime before terminating the entry of the exponent and anytime after calling the ADD EXPONENT function (below).

The CHS function acts as a toggle so a number can be switched to the opposite sign.

# EXAMPLE: See Example D-6.

RELATED FUNCTIONS: EEX

NAME	CLA
KEYPRESS	<back space=""> (SHIFT ALPHA Mode)</back>
STACK	Enables Stack-lift
LAST-X	Not applicable
DESCRIPTION:	

The CLEAR ALPHA function clears the present contents of the Alpha-register.

RELATED FUNCTIONS: CLST, CLX, CLP, CLRG, CLD

NAME	CLD
KEYPRESS	(XEQ or CATALOG 3)
STACK	Enables Stack-lift
LAST-X	Not applicable
DESCRIPTION:	

The CLEAR DISPLAY function displays the alpha register if ALPHA mode is on, otherwise it will display the Stack-X register.

RELATED FUNCTIONS: AVIEW

NAME	CLKEYS (ELI-41SX)
KEYPRESS	(XEQ or CATALOG 3)
STACK	Enables Stack-lift
LAST-X	Not applicable
DESCRIPTION:	

This function clears all key assignments. It provides an alternative to clearing the user keys one by one with the PASN function.

RELATED FUNCTIONS: ASN, PASN

NAME KEYPRESS STACK LAST-X DESCRIPTION: **CLP** (XEQ or CATALOG 3) Enables Stack-lift Not applicable

The CLEAR PROGRAM function clears the program buffer.

**RELATED FUNCTIONS:** CLA, CLST, CLRG,  $CL\Sigma$ 

NAME KEYPRESS STACK LAST-X DESCRIPTION: CLRG (XEQ or CATALOG 3) Enables Stack-lift Not applicable

The CLEAR REGISTER function will clear all of the memory registers to zero. It does not affect the flags or the Stack.

**RELATED FUNCTIONS:** CLA, CLST, CLP,  $CL\Sigma$ 

NAME	CLRGX (ELI-41SX)
KEYPRESS	(XEQ or CATALOG 3)
STACK	Enables Stack-lift
LAST-X	Not applicable
DESCRIPTION:	

The CLRGX function lets you clear data registers selectively according to a control word of the form bbb.eeeii in the Stack-X register. This control word has the same form as the control word for ISG and DSE functions; bbb is the first(beginning) address to be cleared, eee is the last address and ii is the increment, ii is treated as 01 if it is zero.

RELATED FUNCTIONS: CLRG, CLST

NAME	CLST
KEYPRESS	(XEQ or CATALOG 3)
STACK	Enables Stack-lift
LAST-X	Not applicable
DESCRIPTION:	

The CLEAR STACK function clears all of the Stack registers (X,Y,Z,T,L).

RELATED FUNCTIONS: CLA, CLP, CLRG,  $CL\Sigma$ 

NAME	CLX
KEYPRESS	<back space=""> (SHIFT CALC Mode)</back>
STACK	Disables Stack-lift
LAST-X	Not applicable
DESCRIPTION:	

The CLEAR STACK-X function clears the contents of the Stack-X register.

RELATED FUNCTIONS: CL<sub>2</sub>, CL<sub>P</sub>, CL<sub>RG</sub>, CL<sub>ST</sub>

NAME	$CL\Sigma$
KEYPRESS	<f> (SHIFT CALC Mode)</f>
STACK	Enables Stack-lift
LAST-X	Not applicable
DESCRIPTION:	

The CLEAR STATISTICAL REGISTER BLOCK function clears the 6 data registers that are currently being used as the statistical block. The default block starts at memory register 11 and may be changed by the  $\Sigma$ REG function.

**RELATED FUNCTIONS:**  $\Sigma$ -,  $\Sigma$ +,  $\Sigma$ REG, CL $\Sigma$ , MEAN, SDEV

NAMECOPYKEYPRESS(XEQ or CATALOG 3)STACKNot applicableLAST-XNot applicableDESCRIPTION:

The COPY 'label' function loads a copy of the program specified ('label') in to the program buffer.

RELATED FUNCTIONS: EDIT, XEQ, CLP

NAME	COS
KEYPRESS	<i> (CALC Mode)</i>
STACK	Enables Stack-lift
LAST-X	Precomputation Stack-X
DESCRIPTION:	-

The COSINE function calculates the cosine of the Stack-X register according to current angle mode (DEG, RAD, GRAD). Result is placed in the Stack-X register.

RELATED FUNCTIONS: SIN, TAN, ASIN, ACOS, ATAN

NAME	COSH (ELI-41SX)
KEYPRESS	(XEQ or CATALOG 3)
STACK	Enables Stack-lift
LAST-X	Not applicable
DESCRIPTION:	

The COSH function computes the hyperbolic cosine of the number in the Stack-X register and places the result in the Stack-X register.

RELATED FUNCTIONS: SIN, COS, TAN, SINH, TANH

NAME KEYPRESS STACK LAST-X DESCRIPTION: **CREATE (ELI-41SX)** (XEQ or CATALOG 3) Enables Stack-lift Not applicable

The CREATE function uses the memory file name represented by the Alpha-register and creates a zero filled memory file by that name in the default memory directory. The number in the Stack-X register specifies the number of data registers to be allocated in the new data file. If a file with the specified name already exists on the medium, FILE ALREADY EXISTS! error occurs and no new data file is created.

# Example:

- 1) "C:\ELI\MEMFILES\"
- 2) DIRMEM
- 3) "FILE1"
- 4) 500
- 5) CREATE
- ; creates a file named FILE1.MRG in the
- ; C:\ELI\MEMFILES\ sub-directory and reserves room
- ; for 500 data registers.

RELATED FUNCTIONS: READR, etc.

NAME	D-R
KEYPRESS	(XEQ or CATALOG 3)
STACK	Enables Stack-lift
LAST-X	Precomputational Stack-X
DESCRIPTION:	-

The DEGREES TO RADIANS function converts the Stack-X register from degrees to radians.

RELATED FUNCTIONS: R-D

NAME	DATE
KEYPRESS	(XEQ or CATALOG 3)
STACK	Enables Stack-lift, Pushes Stack
LAST-X	Precomputational Stack-X
DESCRIPTION:	

The DATE function will display the current system date. After pressing the <Back Space> key the decimal representation of the date is placed in the Stack-X register. When DATE function is executed in program mode it simply places the numeric representation of the system date into the Stack-X register.

RELATED FUNCTIONS: TIME

NAME	DEC
KEYPRESS	(XEQ or CATALOG 3)
STACK	Enables Stack-lift
LAST-X	Not applicable
DESCRIPTION:	

The DECIMAL function switches The current data entry and display modes to the default decimal format.

RELATED FUNCTIONS: BIN, HEX, OCT

NAME	DEG
KEYPRESS	(XEQ or CATALOG 3)
STACK	Enables Stack-lift
LAST-X	Not applicable
DESCRIPTION:	

The DEGREES function will set the current trigonometric angle mode to the default degree mode.

RELATED FUNCTIONS: RAD, GRAD

NAME KEYPRESS STACK LAST-X DESCRIPTION: **DIRCNV (ELI-41SX)** (XEQ or CATALOG 3) Enables Stack-lift Not applicable

The DIRCNV function uses the string in the Alpha-register to set the default Conversion file directory. The mentioned directory holds the ASCII conversion files, and with this programmable function it can be used to change the default directory during program execution.

### Example:

1) "C:\ELI\CNV" 2) DIRCNV

RELATED FUNCTIONS: DIRUTL, DIRMEM, DIRPGM

NAME	DIRUTL (ELI-41SX)
KEYPRESS	(XEQ or CATALOG 3)
STACK	Enables Stack-lift
LAST-X	Not applicable
DESCRIPTION:	

The DIRUTL function uses the string in the Alpha-register to set the default utility file directory. The mentioned directory holds the files that the user wishes to edit with the QEDIT function.

RELATED FUNCTIONS: DIRCNV, DIRMEM, DIRPGM

NAME	DIRMEM (ELI-41SX)
KEYPRESS	(XEQ or CATALOG 3)
STACK	Enables Stack-lift
LAST-X	Not applicable
DESCRIPTION:	

The DIRMEM function uses the string in the Alpha-register to set the default memory file directory. The mentioned directory holds the data register memory files, status memory files and the key assignments memory files.

### Example:

1) "C:\ELI\MEMFILES" 2) DIRMEM 3) "FILE1" 4) WRTR

RELATED FUNCTIONS: DIRUTL, DIRCNV, DIRPGM

NAME KEYPRESS STACK LAST-X DESCRIPTION: **DIRPGM (ELI-41SX)** (XEQ or CATALOG 3) Enables Stack-lift Not applicable

The DIRPGM function uses the string in the Alpha-register to set the default program file directory. The mentioned directory holds all the PGM files that the program can call as subroutines.

# Example:

1) "C:\ELI\PGMS" 2) DIRPGM 3) XEQ "MARY"

RELATED FUNCTIONS: DIRUTL, DIRMEM, DIRCNV

NAME	DIRPRT (ELI-41SX)
KEYPRESS	(XEQ or CATALOG 3)
STACK	Enables Stack-lift
LAST-X	Not applicable
DESCRIPTION:	

The DIRPRT function uses the string in the Alpha-register to set the default print file directory. The mentioned directory holds the print file that have been created when using the Print To File utility in the Options menu.

RELATED FUNCTIONS: DIRUTL, DIRMEM, DIRPGM

NAMEDSEKEYPRESS(XEQ or CATALOG 3)STACKEnables Stack-lift, Allows indirectionLAST-XNot applicableDESCRIPTION:

The DECREMENT AND SKIP IF EQUAL 'nn' function is used to generate controlled loops. A control number in the form of 'bbbbb.eeeii' must have been placed in register 'nn' previous to execution of this function. The 'bbbbb' part of the control number represents the beginning count for initiating the loop and 'eee' portion represents the ending value of the count. The DSE function first decrements the 'bbbbb' portion by 'ii' and then tests the result against the 'eee' portion, if it is less than or equal then the next line of program will be skipped, otherwise it is executed.

If the 'ii' portion of the control number is left blank, then the calculator automatically defaults to 1, and if the 'eee' portion is left blank then the end count will be 0. To put it in a formula: if (bbbbb - ii)  $\langle =$  eee skip next instruction. In a program, the register containing the control number must be supplied in DSE 'nn'; where, 'nn' is the register containing the control number.

RELATED FUNCTIONS: ISG

NAME	DWELL (ELI-41SX)
KEYPRESS	(XEQ or CATALOG 3)
STACK	Enables Stack-lift
LAST-X	Not applicable
DESCRIPTION:	

The DWELL function uses the number in the Stack-X register to pause the program execution by that amount times 50 milliseconds.

RELATED FUNCTIONS: PSE

NAME KEYPRESS STACK LAST-X DESCRIPTION: **EEX** <P> (CALC Mode) Not applicable Not applicable

The ADD EXPONENT function allows adding an exponent to an Engineering notation type number. The current number in the data entry area will have an E appended to the number and an underscore (cursor) will follow. The calculator then awaits further input. The exponent portion of the number can be any number above -300 and below 300. In order to input a negative exponent, first enter the exponent and then press the <O> (CHS) key.

# EXAMPLE: See Example D-6.

RELATED FUNCTIONS: CHS

NAME	END
KEYPRESS	(XEQ or CATALOG 3)
STACK	Enables Stack-lift
LAST-X	Not applicable
DESCRIPTION:	

The END statement marks the end of a program or subroutine.

RELATED FUNCTIONS: RTN

NAME	ENG
KEYPRESS	(SHIFT CALC Mode)
STACK	Enables Stack-lift, Allows indirection
LAST-X	Not applicable
DESCRIPTION:	••

The ENG function is similar to SCI function except that all exponents are displayed in multiples of three.

RELATED FUNCTIONS: FIX, SCI

NAME	<b>ENTER</b> †
KEYPRESS	<enter> key or <n> (CALC Mode)</n></enter>
STACK	Disables Stack-lift, Pushes the stack
LAST-X	Not applicable
DESCRIPTION:	

The ENTER function will push the contents of the stack down one register. Each register is copied into the next register with the Stack-X register being duplicated in the Stack-Y register. The ENTER function terminates the entry of a number into the Stack-X register, Thus any number entry made after this function will overwrite the contents of the Stack-X register.

RELATED FUNCTIONS: RDN, R↑, X<>Y, X<> nn

NAME	EXEC (ELI-41SX)
KEYPRESS	(XEQ or CATALOG 3)
STACK	Enables Stack-lift
LAST-X	Not applicable
DESCRIPTION:	

The EXEC function uses the current utility directory and the contents of the Alpha-register to run the DOS program specified by the contents of the Alpha-register.

### EXAMPLE:

"C:\DOS" DIRUTL "DIR \*.EXE" EXEC

RELATED FUNCTIONS: DIRUTL

NAME	E↑X
KEYPRESS	<e> (SHIFT CALC Mode)</e>
STACK	Enables Stack-lift
LAST-X	Precomputation Stack-X
DESCRIPTION:	-

The NATURAL EXPONENTIAL function returns in the Stack-X register the value of e (2.718...) raised to the power of the Stack-X register. This is the reverse action of the NATURAL LOGARITHM function.

RELATED FUNCTIONS: LOG, LN,  $10\uparrow X$ , LN1+X, E $\uparrow X$ -1

NAME	<b>E</b> ↑ <b>X-1</b>
KEYPRESS	(XEQ or CATALOG 3)
STACK	Enables Stack-lift
LAST-X	Precomputational Stack-X
DESCRIPTION:	-

This function is used for natural antilogarithms of numbers whose value is close to zero.

RELATED FUNCTIONS: LN1+X, LN, LOG,  $E\uparrow X$ ,  $10\uparrow X$ 

NAME	FACT
KEYPRESS	(XEQ or CATALOG 3)
STACK	Enables Stack-lift
LAST-X	Precomputational Stack-X
DESCRIPTION:	-

The FACT function computes the factorial of the numbers less than 150. It returns in the Stack-X register, the factorial of the number in the Stack-X register.

RELATED FUNCTIONS: None

NAME	FC?
KEYPRESS	(XEQ or CATALOG 3)
STACK	Enables Stack-lift, Allows indirection
LAST-X	Not applicable
DESCRIPTION:	

The IS FLAG CLEAR? function tests the particular flag specified ( 0 <= nn <=55 ) and skips the next sequential program line unless the flag is clear.

RELATED FUNCTIONS: FC?C, FS?, FS?C

NAME	FC?C
KEYPRESS	(XEQ or CATALOG 3)
STACK	Enables Stack-lift, Allows indirection
LAST-X	Not applicable
DESCRIPTION:	

The IS FLAG CLEAR?, CLEAR 'nn' function tests the condition of a specified flag ( $0 \le nn \le 29$ ) and skips the next sequential program line if the flag is set and then clears it.

RELATED FUNCTIONS: FC?, FS?, FS?C

NAME	FIX
KEYPRESS	<z> (SHIFT CALC Mode)</z>
STACK	Enables Stack-lift, Allows indirection
LAST-X	Not applicable
DESCRIPTION:	

The FIX digits format function sets the current display mode to fixed decimal format with 'nn' ( $0 \le nn \le 15$ ) digits viewed after the decimal point. Note that if the Stack-X register is too big or too small to be displayed in FIX format, then the display format will be changed to SCI mode automatically. In FIX 00 mode if flags 29 is clear then no decimal point will be displayed.

RELATED FUNCTIONS: ENG, SCI

NAME	FRC
KEYPRESS	(XEQ or CATALOG 3)
STACK	Enables Stack-lift
LAST-X	Precomputational Stack-X
DESCRIPTION:	-

The FRACTION function replaces contents of the Stack-X register with its decimal portion.

RELATED FUNCTIONS: INT

NAME	FS?
KEYPRESS	<t> (SHIFT CALC Mode)</t>
STACK	Enables Stack-lift, Allows indirection
LAST-X	Not applicable
DESCRIPTION:	

The IS FLAG SET? 'nn' function tests the condition of the flag 'nn' ( $0 \le nn \le 55$ ) and skips the next sequential program line if the flag is clear, otherwise program execution continues.

RELATED FUNCTIONS: CF, FC?, FC?C, SF, FS?C

NAME	FS?C
KEYPRESS	(XEQ or CATALOG 3)
STACK	Enables Stack-lift, Allows indirection
LAST-X	Not applicable
DESCRIPTION:	

The IS FLAG SET? CLEAR 'nn' function tests the condition of the flag 'nn' ( $0 \le nn \le 29$ ) and skips the next sequential program line if the flag is clear, otherwise it will clear the specified flag.

RELATED FUNCTIONS: FS?, FC?, FC?C

NAME KEYPRESS STACK LAST-X DESCRIPTION: **GETKEY (ELI-41SX)** (XEQ or CATALOG 3) Enables Stack-lift Not applicable

The GETKEY function is used to get a key from the keyboard. The function returns the keycode in the Stack-X register using the row and column identification of the HP-41 keyboard. It waits for ten seconds for a key to be pressed and then puts the keycode in the Stack-X register and lifts the stack if stack lift is enabled. If no key is pressed within the 10 seconds dwell time then a 0 is put into Stack-X.

RELATED FUNCTIONS: GETKEYX

NAME	GETKEYX (ELI-41SX)
KEYPRESS	(XEQ or CATALOG 3)
STACK	Enables Stack-lift
LAST-X	Not applicable
DESCRIPTION:	

Same as GETKEY except that the 10 second dwell time is programmable using the value in the Stack-X register.

**RELATED FUNCTIONS:** GETKEY

NAME	GRAD
KEYPRESS	(XEQ or CATALOG 3)
STACK	Enables Stack-lift
LAST-X	Not applicable
DESCRIPTION:	••

The GRADIANS function changes the current trigonometric angle mode to gradians.

RELATED FUNCTIONS: RAD, DEG

NAME G KEYPRESS < STACK E LAST-X N DESCRIPTION:

**GTO** <M> (SHIFT CALC Mode) Enables Stack-lift, Allows indirection Not applicable

The GTO function is used to direct program execution to a known numeric (00 - 99) or alpha label. It is similar to the BASIC'S GO TO function except that execution will continue from the named label instead of a line number.

ELI-41 also supports GTO IND nn, in which the data register nn will contain either a two digit label number or a seven character alphanumeric label which could reside within the internal program buffer or the external program directory.

While in the command mode user has the option of going to a particular line number by pressing the  $\langle M \rangle$  (GTO) key followed by the period key  $\langle . \rangle$  to generate the GTO \_\_\_\_\_ prompt at which time user will enter the 3 digit desired line number (300 maximum). Upon entering the desired line number, the internal program line pointer is set to the 3 digit number entered, and the user will have the option of continuing program execution from that point or going into the EDIT mode to view and edit the program (NOTE: line number 1 should be entered as 001).

RELATED FUNCTIONS: XEQ, SST, LBL

NAME	HEX
KEYPRESS	(XEQ or CATALOG 3)
STACK	Enables Stack-lift
LAST-X	Not applicable
DESCRIPTION:	

The HEXADECIMAL function switches the data entry and display modes to hexadecimal format. The acceptable input characters are <0>-<9> and <A>-<F>. Functions whose key is within these limits are disabled until the calculator is switched to another input mode. Programs in execution are not affected by this limitation.

NAMEHMSKEYPRESS(XEQ or CATALOG 3)STACKEnables Stack-liftLAST-XPrecomputational Stack-XDESCRIPTION:

The HOURS MINUTES SECONDS function converts the decimal equivalent of degrees or hours to 'HH.mmsshh' format, where: 'HH' indicates hours or degrees, 'mm' indicates minutes, 'ss' indicates seconds, 'hh' indicates hundredths of seconds.

RELATED FUNCTIONS: HMS+, HMS-, HR

This function adds the contents of the Stack-X register to the Stack-Y register. The addition assumes that the contents of both registers are in 'HH.mmsshh',Seconds.

RELATED FUNCTIONS: HMS, HMS-, HR

NAME	HMS-
KEYPRESS	(XEQ or CATALOG 3)
STACK	Enables Stack-lift, Stack popped
LAST-X	Precomputational Stack-X
DESCRIPTION:	-

This function subtracts the contents of the Stack-X register from the contents of the Stack-Y register. The subtraction assumes the contents of both registers are in 'HH.mmsshh' format.

RELATED FUNCTIONS: HMS, HMS+, HR

NAME	HR
KEYPRESS	(XEQ or CATALOG 3)
STACK	Enables Stack-lift
LAST-X	Precomputational Stack-X
DESCRIPTION:	-

The HOUR function converts the contents of the Stack-X register from 'HH.mmsshh' format to its decimal equivalent.

RELATED FUNCTIONS: HMS, HMS-, HMS+

NAME KEYPRESS STACK LAST-X DESCRIPTION: **INPORT** (XEQ or CATALOG 3) Enables Stack-lift Precomputation Stack-X

The INPORT function reads a byte value from the currently selected I/O port and places the number into the Stack-X register.

### Example:

-	
LBL "RELAY"	
768	; 0x300 I/O base
11	; CMD register offset
+	
PORTSEL	; select as current port
129	; 0x81 port $A = out B = out Clow = in$
OUTPORT	; write 0x81 to CMD register
768	
9	; port A offset
+	
PORTSEL	; select port A as current port
0	
OUTPORT	; output 0x00 to port A
LBL 00	
"VALUE TO WRITE	TO PORT?"
PROMPT	
OUTPORT	; output value to port A

GTO 00 END

RELATED FUNCTIONS: OUTPORT, PORTSEL

NAME	INT
KEYPRESS	(XEQ or CATALOG 3)
STACK	Enables Stack-lift
LAST-X	Precomputational Stack-X
DESCRIPTION:	-

The INTEGER function returns in the Stack-X register the integer portion of the Stack-X register.

RELATED FUNCTIONS: FRC

NAME	ISG
KEYPRESS	<o> (SHIFT CALC Mode)</o>
STACK	Enables Stack-lift, Allows indirection
LAST-X	Not applicable

The Increment and Skip if Greater 'nn' function is used to generate controlled loops. A control number in the form of bbbbb.eeeii must have been placed in register 'nn' prior to execution of this function. The 'bbbbb' part of the control number represents the beginning count for initiating the loop and 'eee' portion represents the ending value of the count. The ISG function first increments the 'bbbbb' portion by 'ii' and then tests the result against the eee portion, if it is greater then the next line of program will be skipped, otherwise it is executed.

If the 'ii' portion of the control number is left blank, then the calculator automatically defaults to 1 and if the 'eee' portion is left blank then the end count will be 0. To put it in a formula: if (bbbbb + ii) >= eee skip next instruction. In a program, the register containing the control number must be supplied in ISG 'nn'; where, 'nn' is the register containing the control number.

RELATED FUNCTIONS: DSE

NAME
KEYPRESS
STACK
LAST-X
DESCRIPTION:

LASTX <,> (SHIFT CALC Mode) Enables Stack-lift, Stack is pushed Not applicable

The LASTX function sets the Stack-X register equal to the Last-X register. First the contents of stack is pushed if the Stack lift was previously enabled, then Last-X register is copied in to Stack-X.

NAME	LBL
KEYPRESS	<l> (SHIFT CALC Mode)</l>
STACK	Enables Stack-lift
LAST-X	Not applicable
DESCRIPTION:	

The LABEL function is used to define a Program or Subroutine name. Each program must begin with the LABEL and the program name followed by the body of the program and ended with the END function. For a more complete description of the LABEL function and its uses within programs please refer to HP41's Programmer's reference manual.

RELATED FUNCTIONS: XEQ, GTO

NAME	LN
KEYPRESS	<e> (CALC Mode)</e>
STACK	Enables Stack-lift
LAST-X	Precomputation Stack-X
DESCRIPTION:	-

The NATURAL LOGARITHM function calculates the base e (2.718...) log of the Stack-X register and places the result in the Stack-X register.

### EXAMPLE: See Example D-3.

**RELATED FUNCTIONS:**  $10\uparrow X$ ,  $E\uparrow X$ , LOG,  $E\uparrow X$ -1, LN1+X

NAME	LN1+X
KEYPRESS	(XEQ or CATALOG 3)
STACK	Enables Stack-lift
LAST-X	Precomputational Stack-X
DESCRIPTION:	-

This LN1+X function returns in the Stack-X register the natural logarithm of numbers close to one.

RELATED FUNCTIONS:  $E\uparrow X-1$ , LN, LOG,  $10\uparrow X$ ,  $E\uparrow X$ 

NAME	LOG
KEYPRESS	<d> (CALC Mode)</d>
STACK	Enables Stack-lift
LAST-X	Precomputation Stack-X
DESCRIPTION:	-

The LOGARITHM function calculates the base 10 log of the Stack-X register and places the result in the Stack-X.

# **EXAMPLE: See Example D-3.**

**RELATED FUNCTIONS:**  $10\uparrow X$ ,  $E\uparrow X$ , LN,  $E\uparrow X$ -1, LN1+X

NAME	LPT# (ELI-41SX)
KEYPRESS	(XEQ or CATALOG 3)
STACK	Enables Stack-lift
LAST-X	Not applicable
DESCRIPTION:	

The LPT# function uses the number in the Stack-X register ( $1 \le X \le 3$ ) to set the currently accessed line printer. Therefore the printer access can be changed during program execution.

RELATED FUNCTIONS: NONE

NAME	MEAN
KEYPRESS	(XEQ or CATALOG 3)
STACK	Enables Stack-lift
LAST-X	Precomputational Stack-X
DESCRIPTION:	-

The MEAN function computes the arithmetic average of the accumulated values in the statistical block.

**RELATED FUNCTIONS:** SDEV,  $\Sigma$ +,  $\Sigma$ -,  $\Sigma$ REG

NAME	MOD
KEYPRESS	(XEQ or CATALOG 3)
STACK	Enables Stack-lift, Stack popped
LAST-X	Precomputational Stack-X
DESCRIPTION:	-

The MODULO function computes the remainder of Stack-Y divided by Stack-X and places the result in the Stack-X.

(Stack-X = Stack-Y - floor(Stack-Y/Stack-X) \* Stack-X)

RELATED FUNCTIONS: None

NAME	NDIG? (ELI-41SX)
KEYPRESS	(XEQ or CATALOG 3)
STACK	Enables Stack-lift
LAST-X	Not applicable
DESCRIPTION:	

The NDIG? function places current setting of the number of digits displayed into the Stack-X register.

RELATED FUNCTIONS: None

NAMENOTKEYPRESS(XEQ or CATALOG 3)STACKEnables Stack-liftLAST-XPrecomputational Stack-XDESCRIPTION:

The NOT function computes the boolean negation of the integer portion of the Stack-X register.

RELATED FUNCTIONS: XOR, AND, OR, <<, >>

NAME	OCT
KEYPRESS	(XEQ or CATALOG 3)
STACK	Enables Stack-lift
LAST-X	Not applicable
DESCRIPTION:	

The OCTAL function switches the data entry and display modes to octal format. The acceptable input digits are 0 through 8.

RELATED FUNCTIONS: DEC, BIN, HEX

NAME	OFF
KEYPRESS	(XEQ or CATALOG 3)
STACK	Not applicable
LAST-X	Not applicable
DESCRIPTION:	

The OFF function will terminate the calculator. If used with the pop-up version, it will terminate the action of the calculator but leave it resident.

**RELATED FUNCTIONS:** None

NAME	OR
KEYPRESS	(XEQ or CATALOG 3)
STACK	Enables Stack-lift
LAST-X	Precomputational Stack-X
DESCRIPTION:	-

The OR function, performs a boolean OR between the contents of the Stack-Y register and the Stack-X register. The result is placed in the Stack-X register.

RELATED FUNCTIONS: XOR, AND, NOT, <<, >>

NAME	OUTPORT
KEYPRESS	(XEQ or CATALOG 3)
STACK	Enables Stack-lift
LAST-X	Not applicable
DESCRIPTION:	

The OUTPORT function writes the byte value represented by the integer portion of the Stack-X register to the currently selected I/O port. The Stack-X register must contain a NUMBER between 0 and 255.

### Example:

LBL "RELAY"	
768	; 0x300 I/O base
11	; CMD register offset
+	-
PORTSEL	; select as current port
129	; 0x81 port A = out B = out Clow = in
OUTPORT	; write 0x81 to CMD register
768	
9	; port A offset
+	
PORTSEL	; select port A as current port
0	
OUTPORT	; output 0x00 to port A
LBL 00	
<b>"VALUE TO WRIT</b>	E TO PORT?"

PROMPT OUTPORT ; output value to port A GTO 00 END

RELATED FUNCTIONS: INPORT, PORTSEL

NAMEP-RKEYPRESS<W> (SHIFT CALC Mode)STACKEnables Stack-liftLAST-XPrecomputation Stack-XDESCRIPTION:

The Polar to Rectangular function converts polar (Stack-X= magnitude, Stack-Y= angle) coordinate system to rectangular (Stack-X = 'x', Stack-Y= 'y') coordinate system. Note that the angle is assumed to be in the current angle mode (DEG, RAD, GRAD).

RELATED FUNCTIONS: R-P

NAME	PASN (ELI-41SX)
KEYPRESS	(XEQ or CATALOG 3)
STACK	Enables Stack-lift
LAST-X	Not applicable
DESCRIPTION:	

The PASN function allows each program to program the key assignments. User must place the keycode in the Stack-X register, a function or routine name in the Alpha-register and then execute the PASN function. the key codes are specified by: The first digit is a row number, starting from 1 at the top row and the second digit is the key number in the row, starting from 1 at the left. A minus sign denotes a shifted key. PASN cannot be used to make assignments to the toggle keys or to the SHIFT key (F10).

RELATED FUNCTIONS: ASN, CLKEYS
NAME	PI
KEYPRESS	<space bar=""> (SHIFT CALC Mode)</space>
STACK	Enables Stack-lift, Stack is pushed
LAST-X	Not applicable
DESCRIPTION:	

The PI function sets the Stack-X register equal to 3.14159265359.

RELATED FUNCTIONS: None

NAME	PORTSEL
KEYPRESS	(XEQ or CATALOG 3)
STACK	Enables Stack-lift
LAST-X	Not applicable
DESCRIPTION:	

The PORTSEL function uses the integer portion of the Stack-X register to specify the port number that will be used in subsequent calls to the INPORT and OUTPORT functions.

Warning: Use of this function with unknown port addresses could result in severe hardware and software damage to your system. Please be very cautious in using this function.

#### Example:

LBL "RELAY"	
768	; 0x300 I/O base
11	; CMD register offset
+	
PORTSEL	; select as current port
129	; 0x81 port A = out B = out Clow = in
OUTPORT	; write 0x81 to CMD register
768	-
9	; port A offset
+	
PORTSEL	; select port A as current port
0	•
OUTPORT	; output 0x00 to port A
LBL 00	• •

"VALUE TO WRITE TO PORT?" PROMPT OUTPORT ; output value to port A GTO 00 END

RELATED FUNCTIONS: INPORT, OUTPORT

NAME	POSA (ELI-41SX)
KEYPRESS	(XEQ or CATALOG 3)
STACK	Enables Stack-lift
LAST-X	Not applicable
DESCRIPTION:	

This function searches the Alpha-register for the text beginning with the string or character code contained in the Stack-X. Lets assume you are looking for the letter A and the Alpha-register contains the text string MANIA. You can put the number 65 (ASCII A) in the Stack-X and then execute POSA. Positions in the Alpha-register are counted from left to right, starting at 0, so you will see the result 1 in the Stack-X register.

RELATED FUNCTIONS: AROT, ATOX, XTOA, ALENG

NAME	PRA
KEYPRESS	(XEQ or CATALOG 3)
STACK	Enables Stack-lift
LAST-X	Not applicable
DESCRIPTION:	

The PRINT ALPHA function prints the contents of the ALPHA register and advances to the next line.

**PRBUF** (XEQ or CATALOG 3) Enables Stack-lift Not applicable

The PRINT BUFFER function simply sends a carriage return to the printer to print the current contents of the printer's line buffer.

RELATED FUNCTIONS: None

NAMEPRFLAGKEYPRESS(XEQ or CATALOG 3)STACKEnables Stack-liftLAST-XNot applicableDESCRIPTION:

The PRINT FLAGS function prints a listing of all the flags that are currently set on.

RELATED FUNCTIONS: None

NAME**PROMPT**KEYPRESS(XEQ or CATALOG 3)STACKEnables Stack-liftLAST-XNot applicableDESCRIPTION:

The PROMPT function is used in a program to display the Alpha-register; usually containing a prompt for the user to make an entry of some kind, and halts the program execution. To continue execution, the <> (RUN/STOP) key must be pressed.

RELATED FUNCTIONS: PSE

NAMEPRPKEYPRESS(XEQ or CATALOG 3)STACKEnables Stack-liftLAST-XNot applicableDESCRIPTION:

The PRINT PROGRAM function requests a program name to be printed by displaying PRP \_ message. A program name can be entered either by pressing the  $\langle$ F9 $\rangle$  (ALPHA) key and the name followed by an  $\langle$ Enter $\rangle$  or  $\langle$ F9 $\rangle$  key or by pressing the  $\langle$ F9 $\rangle$  key twice to display a catalog of the available programs to be listed. In the later case simply move the highlight bar to the desired program name and press the  $\langle$ Enter $\rangle$  key to begin listing the program on to the printer. Each line of the program will be preceded by the program line number and followed by comments if any had been entered.

NOTE: Both flags 55 and 21 must be Set before this function is used.

RELATED FUNCTIONS: None

NAMEPRREGKEYPRESS(XEQ or CATALOG 3)STACKEnables Stack-liftLAST-XNot applicableDESCRIPTION:

The PRINT REGISTERS function prints a listing of all the 500 registers that have a non zero value.

**PRSTK** (XEQ or CATALOG 3) Enables Stack-lift Not applicable

The PRINT STACK function prints a listing of the Stack-X, Stack-Y, Stack-Z, Stack-T, LAST-X registers and the Alpha-register.

RELATED FUNCTIONS: None

NAMEPRXKEYPRESS(XEQ or CATALOG 3)STACKEnables Stack-liftLAST-XNot applicableDESCRIPTION:

The PRX function prints the contents of the Stack-X register according to the current display format and number digits and advances to the next line.

RELATED FUNCTIONS: None

NAME	PSE
KEYPRESS	(XEQ or CATALOG 3)
STACK	Enables Stack-lift
LAST-X	Not applicable
DESCRIPTION:	

The PAUSE function displays the Stack-X register and then delays the program execution for approximately 1 second.

RELATED FUNCTIONS: PROMPT

## PSIZE (ELI-41SX)

NAME KEYPRESS STACK LAST-X DESCRIPTION: **PSIZE (ELI-41SX)** (XEQ or CATALOG 3) Enables Stack-lift Not applicable

The PSIZE function allows for adjusting the size of data storage registers dynamically during program execution. It uses the contents of Stack-X register and tries to adjust the maximum number of data registers to that number.

RELATED FUNCTIONS: None

NAME	<b>GEDIT</b> (ELI-41SX)
KEYPRESS	(XEQ or CATALOG 3)
STACK	Enables Stack-lift
LAST-X	Not applicable
DESCRIPTION:	

The QEDIT function uses the default utility directory and the filename placed in the Alpha-register to bring up an ASCII text editor on the left hand side of the display.

A complete ASCII text editor has been supplied on the ELI-41SX master disk and is called QEDIT. To use QEDIT simply use the DIRUTL function to set the utility directory to the drive and directory that the ASCII is residing or to be created and then place the name of the ASCII file into the Alpha-register and execute the QEDIT function. The Text Editor will be displayed on the left half of the display and then you can start editing the file.

<Ctrl-Y> keys will delete the entire line that the cursor is placed upon.

<Alt-F1> keys will save any changes made to the same file if one already exists, otherwise it will create the file and save all entries to that file.

<ESC> key will abort editing without saving any changes made to the text file.

**R-D** (XEQ or CATALOG 3) Enables Stack-lift Precomputational Stack-X

The RADIANS TO DEGREES function converts the contents of the Stack-X register from radians to degrees.

RELATED FUNCTIONS: D-R

NAME	R-P
KEYPRESS	<x> (SHIFT CALC Mode)</x>
STACK	Enables Stack-lift
LAST-X	Precomputation Stack-X
DESCRIPTION:	-

The R-P function converts rectangular (Stack-X = 'x', Stack-Y = 'y') coordinate system to polar (Stack-X = magnitude, Stack-Y = Angle) coordinate system. Note that angle is computed based on current angle mode (DEG, RAD, GRAD).

RELATED FUNCTIONS: P-R

NAME	RAD
KEYPRESS	(XEQ or CATALOG 3)
STACK	<b>Enables Stack-lift</b>
LAST-X	Not applicable
DESCRIPTION:	

The RADIANS function changes the current trigonometric angle mode to radians.

RELATED FUNCTIONS: DEG, GRAD

NAMERAN# (ELI-41SX)KEYPRESS(XEQ or CATALOG 3)STACKEnables Stack-liftLAST-XPrecomputation Stack-XDESCRIPTION:

The RAN# places a pseudo-random integer in the range of 0 to 32,767 into the Stack-X register.

RELATED FUNCTIONS: SEED

NAMERCLKEYPRESS<M> (CALC Mode)STACKEnables Stack-lift, Allows indirection, Stack pushedLAST-XNot applicableDESCRIPTION:

The RECALL 'nn' function is the reverse of the STORE function discussed above. The register nn is copied into the Stack-X register and all of the options noted with STORE are available with RECALL, except for (+,-,/,\*). If Stack-lift was enabled previously, it will push the Stack and then continue with its normal operation.

#### EXAMPLE See Example D-5.

RELATED FUNCTIONS: STO, ASTO, ARCL, ST+, ST-, ST\*, ST/

NAME	RCLFLAG (ELI-41SX)
KEYPRESS	(XEQ or CATALOG 3)
STACK	Enables Stack-lift
LAST-X	Not applicable
DESCRIPTION:	

Not Implemented Yet.

**RDN** <G> (CALC Mode) Enables Stack-lift Not applicable

**READA (ELI-41SX)** (XEQ or CATALOG 3)

**Enables Stack-lift** 

Not applicable

The ROLL DOWN function moves the contents of the various stack registers downward one register, that is, the contents of the Stack-X register is moved to the Stack-T register, the Stack-T register is moved to the Stack-Z register, etc.

RELATED FUNCTIONS: R↑, X<> nn, X<>Y

NAME KEYPRESS STACK LAST-X DESCRIPTION:

The READA function uses the default memory directory set by DIRMEM and the file name set by the Alpha-register to read all status information, key assignments, and data storage registers from the disk storage.

RELATED FUNCTIONS: None

NAME	READK (ELI-41SX)
KEYPRESS	(XEQ or CATALOG 3)
STACK	Enables Stack-lift
LAST-X	Not applicable
DESCRIPTION:	

The READK function reads and sets the key assignments from the memory file name specified in the Alpha-register.

READR (ELI-41SX) (XEQ or CATALOG 3) Enables Stack-lift Not applicable

The READR function reads the contents of the data storage registers from the .MRG file specified by the Alpha-register.

RELATED FUNCTIONS: None

NAME	READRX (ELI-41SX)
KEYPRESS	(XEQ or CATALOG 3)
STACK	Enables Stack-lift
LAST-X	Not applicable
DESCRIPTION:	

The READRX function uses the number in the Stack-X to specify the registers that the contents will be filled with data from the disk storage. Before executing this function the memory file storage and the pointers must be positioned by SEEKR function or a previous WRTRX, WRTR, READRX, or READR. The contents of the Stack-X registers are interpreted as bbb.eee, where bbb is the first register in the calculator to be filled. If eee is less than bbb, only the bbb register is read. Reading stops when the last storage register or end-of-file is reached.

RELATED FUNCTIONS: None

NAME	READS (ELI-41SX)
KEYPRESS	(XEQ or CATALOG 3)
STACK	Enables Stack-lift
LAST-X	Not applicable
DESCRIPTION:	

The READS functions reads and sets the status of the calculator from the .MST file specified by the Alpha-register.

**READSUB (ELI-41SX)** (XEQ or CATALOG 3) Enables Stack-lift Not applicable

The READSUB function uses the filename specified by the Alpha-register to append it to the end of the program in the program buffer.

RELATED FUNCTIONS: None

NAME	<b>REGMOVE (ELI-41SX)</b>
KEYPRESS	(XEQ or CATALOG 3)
STACK	Enables Stack-lift
LAST-X	Not applicable
DESCRIPTION:	

The REGMOVE function copies the contents of a block of data registers to another block of registers. The Stack-X contains a number of the type sss.dddnnn, where the sss is the source block address of the data registers and ddd is the destination block address and nnn is the number of data registers to be copied.

RELATED FUNCTIONS: None

NAME	<b>REGSWAP</b> (ELI-41SX)
KEYPRESS	(XEQ or CATALOG 3)
STACK	Enables Stack-lift
LAST-X	Not applicable
DESCRIPTION:	

The REGSWAP function works exactly like the REGMOVE function except that the contents of the source and destination blocks of data registers are swapped.

NAME	RND
KEYPRESS	(XEQ or CATALOG 3)
STACK	Enables Stack-lift
LAST-X	Precomputational Stack-X
DESCRIPTION:	

The RND function rounds the Stack-X register to the number of decimal digits specified by FIX, SCI, ENG functions.

RELATED FUNCTIONS: FIX, SCI, ENG

NAME	RTN
KEYPRESS	<p> (SHIFT CALC Mode)</p>
STACK	Enables Stack-lift
LAST-X	Not applicable
DESCRIPTION:	

The RETURN function is used in a program to denote the end of a subroutine. Once the RTN instruction is executed the program will return to the line following the XEQ instruction that had called the subroutine. If the subroutine had not been called previously by a XEQ function then the RTN function will halt the program execution and act as if the end of the program had been reached.

RELATED FUNCTIONS: XEQ, GTO, STOP, SST, END

NAME	RUN/STOP (R/S)
KEYPRESS	<>> (CALC Mode)
STACK	Enables Stack-lift
LAST-X	Not applicable
DESCRIPTION:	

The Run/Stop function is used to perform one of the two functions as the name implies. First, continuation of a program that was halted because of a PROMPT, STOP or BREAK instruction. Secondly, halting a program that is currently being executed.

RELATED FUNCTIONS: GTO, SST, EDIT, COPY, BREAK, STOP

NAME	$\mathbf{R}_{\uparrow}$
KEYPRESS	(XEQ or CATALOG 3)
STACK	Enables Stack-lift
LAST-X	Not applicable
DESCRIPTION:	

The ROLL UP function shifts the contents of the Stack registers up one register. The Stack-T is placed in to Stack- X and so on.

RELATED FUNCTIONS: ENTER↑, RDN, X<>Y, X<>

NAME	SCI
KEYPRESS	<=> (SHIFT CALC Mode)
STACK	Enables Stack-lift, Allows indirection
LAST-X	Not applicable
DESCRIPTION:	

The SCI function sets the current display mode to scientific format. Each number will be displayed with one digit before the decimal point and 'nn' ( $0 \le nn \le 15$ ) digits after the decimal point multiplied by a power of 10.

RELATED FUNCTIONS: ENG, FIX

NAME	SDEV
KEYPRESS	(XEQ or CATALOG 3)
STACK	Enables Stack-lift
LAST-X	Precomputational Stack-X
DESCRIPTION:	-

The STANDARD DEVIATION function computes the standard deviation sample of the statistical block. It places the sample of the X values in the Stack-X and the Y values in the Stack-Y.

**RELATED FUNCTIONS:** MEAN,  $\Sigma$ +,  $\Sigma$ -,  $\Sigma$ REG

## SEED (ELI-41SX)

NAME KEYPRESS STACK LAST-X DESCRIPTION: **SEED (ELI-41SX)** (XEQ or CATALOG 3) Enables Stack-lift Not applicable

The SEED function can be used to set the random starting point for the RAN# function. It uses the contents of the Stack-X register to set the seed.

NAME	SEEKR (ELI-41SX)
KEYPRESS	(XEQ or CATALOG 3)
STACK	Enables Stack-lift
LAST-X	Not applicable
DESCRIPTION:	

The SEEKR function positions the data storage file pointer to the number specified by the Stack-X register. This permits data to be stored and retrieved from individual registers within a file using the WRTRX and READRX functions. The Alpha-register specifies the name of the memory file.

NAME	SF
KEYPRESS	<r> (SHIFT CALC Mode)</r>
STACK	Enables Stack-lift, Allows indirection
LAST-X	Not applicable
DESCRIPTION:	••

The Set Flag 'nn' function is used to set flags 00 - 29 ON. For example, to set the printer flag (21) on execute the SF function by pressing  $\langle F10 \rangle$  (SHIFT)  $\langle R \rangle$  and then enter 21 in response to the SF \_\_ prompt.

Certain flags are reserved for the user and are referred to in Appendix A. These flags can be tested within programs and specific actions can be taken accordingly.

RELATED FUNCTIONS: CF, FC?, FC?C, FS?, FS?C

NAME	SIGN
KEYPRESS	(XEQ or CATALOG 3)
STACK	Enables Stack-lift
LAST-X	Precomputational Stack-X
DESCRIPTION:	-

The SIGN function replaces the Stack-X register with a one (1) using the sign of the previous Stack-X register.

**RELATED FUNCTIONS:** None

NAME	SIN
KEYPRESS	<h> (CALC Mode)</h>
STACK	Enables Stack-lift
LAST-X	Precomputation Stack-X
DESCRIPTION:	

The SINE function calculates the sine of the Stack-X register according to current angle mode (DEG, RAD, GRAD). Result is placed in the Stack-X register.

RELATED FUNCTIONS: COS, TAN, ASIN, ACOS, ATAN

NAME	SINH (ELI-41SX)
KEYPRESS	(XEQ or CATALOG 3)
STACK	Enables Stack-lift
LAST-X	Not applicable
DESCRIPTION:	

The SINH function computes the hyperbolic sine of the number in the Stack-X register and places the result in the Stack-X register.

RELATED FUNCTIONS: SIN, COS, TAN, COSH, TANH

## SIZE? (ELI-41SX)

NAME KEYPRESS STACK LAST-X DESCRIPTION: SIZE? (ELI-41SX) (XEQ or CATALOG 3) Enables Stack-lift Not applicable

The SIZE? function returns in the Stack-X the current setting for the maximum number of data registers.

RELATED FUNCTIONS: PSIZE

NAME	SKPCHR
KEYPRESS	(XEQ or CATALOG 3)
STACK	Enables Stack-lift
LAST-X	Not applicable
DESCRIPTION:	

The SKIP CHARACTER function sends a series of spaces to the printer's line buffer according to the contents of the Stack-X register.

RELATED FUNCTIONS: None

NAME	SQRT
KEYPRESS	<c> (CALC Mode)</c>
STACK	Enables Stack-lift
LAST-X	Precomputation Stack-X
DESCRIPTION:	-

The SQUARE ROOT function calculates the square root of the Stack-X register and places the result in the Stack-X register, thus overwriting its previous contents. Attempted use of this function on a negative number will result in an error message.

#### EXAMPLE: See Example D-2.

RELATED FUNCTIONS: X<sup>1</sup>2

NAMESSTKEYPRESS<;> (CALC Mode)STACKNot applicableLAST-XNot applicableDESCRIPTION:

The primary purpose of the SINGLE STEP function is in the debugging of programs. A program must be first loaded into the internal program buffer using either the COPY, XEQ or EDIT functions, however the SST function must be used before the END statement is reached.

The instruction area will display the instruction and its line number after the function is executed and the Stack-X register will contain the latest results. Continued pressing of the <;> (SST) key will progressively step through the program until the END statement is reached. Any errors made during execution of the program instruction will display a pertinent error message and the line number of the statement in error.

RELATED FUNCTIONS: XEQ, GTO, COPY

NAMEST+KEYPRESS(XEQ or CATALOG 3)STACKEnables Stack-lift, Allows indirectionLAST-XNot applicableDESCRIPTION:

The STORE WITH ADDITION 'nn' function adds the contents of the Stack-X register to the register number supplied.

RELATED FUNCTIONS: STO, RCL, ASTO, ARCL, ST-, ST\*, ST/

SST

NAME	ST-
KEYPRESS	(XEQ or CATALOG 3)
STACK	Enables Stack-lift, Allows indirection
LAST-X	Not applicable
DESCRIPTION:	

The STORE WITH SUBTRACT 'nn' function subtracts the contents of the Stack-X register from the register number specified

RELATED FUNCTIONS: STO, RCL, ASTO, ARCL, ST+, ST\*, ST/

NAME	ST*
KEYPRESS	(XEQ or CATALOG 3)
STACK	Enables Stack-lift, Allows indirection
LAST-X	Not applicable
DESCRIPTION:	

The STORE WITH MULTIPLY 'nn' function multiplies the contents of the Stack-X register by the register number specified.

RELATED FUNCTIONS: STO, RCL, ASTO, ARCL, ST+, ST-, ST/

NAME	ST/
KEYPRESS	(XEQ or CATALOG 3)
STACK	Enables Stack-lift, Allows indirection
LAST-X	Not applicable
DESCRIPTION:	••

The STORE WITH DIVIDE 'nn' function divides the contents of the register number supplied by the Stack-X register.

RELATED FUNCTIONS: STO, RCL, ASTO, ARCL, ST+, ST-, ST\*

**STO** <L> (CALC Mode) Enables Stack-lift. Allows indirection Not applicable

The STORE 'nn' function is used to store intermediate results in memory registers 0 through 500. The STO function will request a two digit register number by displaying STO \_\_\_\_\_ message.

At this point some additional acceptable characters can be entered to allow indirect storage of numbers in registers 100 through 499 or to perform math operations on any of the data registers. A list of these options are as follows:

- 1. A modifying character such as +,-,\*,/ which signifies an addition, subtraction, multiplication or division to the register number that will follow. The data entry area will display ST+\_\_\_\_\_ in the case of the <+> key entry. The two underscores signifying further entry is needed.
- 2. The <F10> (SHIFT) key will allow indirect storage of the Stack-X register in any of the 500 data registers. The STO IND \_\_ message will be displayed to request entry of a two digit number. If a two digit number is entered, the integer contents of the named register will point to the memory register which will receive a copy of the contents of the Stack-X register.
- 3. The <.> period key will allow storage of the Stack-X register on any of the stack registers (X,Y,Z,T and L). The STO STK \_ message will be displayed upon pressing the <.> period key. At this point a single character that represents the stack registers (X,Y,Z,T or L) can be entered.

Any or all of these special modifications to the basic STORE function can be used but only in the order that they were presented here. Some Examples are as follows:

## ST+ IND STK Y

Would add the Stack-X register to the register pointed to by the integer portion of the Stack-Y register.

ST+ IND 02

Would add the Stack-X register to the contents of register number pointed to by the contents of the memory register 02.

Any entry made out of place will not trigger an error response with the exception of the stack character entry. The illegitimate key will simply be ignored. Pressing the <Back Space> key will terminate the function with no action being taken but to redisplay the contents of the Stack-X register.

## EXAMPLE: See Example D-5.

RELATED FUNCTIONS: RCL, ST+, ST-, ST\*, ST/

NAMESTOFLAG (ELI-41SX)KEYPRESS(XEQ or CATALOG 3)STACKEnables Stack-liftLAST-XNot applicableDESCRIPTION:

Not Implemented Yet.

**RELATED FUNCTIONS:** None

NAMESTOPKEYPRESS(XEQ or CATALOG 3)STACKEnables Stack-liftLAST-XNot applicableDESCRIPTION:

The STOP function halts the execution of a program.

RELATED FUNCTIONS: BREAK, XEQ, R/S, END

NAME	TAN
KEYPRESS	<j> (CALC Mode)</j>
STACK	Enables Stack-lift
LAST-X	Precomputation Stack-X
DESCRIPTION:	-

The TANGENT function calculates the tangent of the Stack-X register according to current angle mode (DEG, RAD, GRAD). Result is placed in the Stack-X register.

RELATED FUNCTIONS: SIN, COS, ASIN, ACOS, ATAN

NAME	TANH (ELI-41SX)
KEYPRESS	(XEQ or CATALOG 3)
STACK	Enables Stack-lift
LAST-X	Not applicable
DESCRIPTION:	

The TANH function computes the hyperbolic tangent of the number in the Stack-X register and places the result in the Stack-X register.

RELATED FUNCTIONS: ACOS, ATAN, SIN, COS, TAN, COSH, SINH

NAME	TIME
KEYPRESS	(XEQ or CATALOG 3)
STACK	Enables Stack-lift, Pushes Stack
LAST-X	Precomputational Stack-X
DESCRIPTION:	-

The TIME function displays the current time, and after pressing the <Back Space> key places the time in the Stack-X register based on 'HH.mmsshh' format. When TIME function is executed in program mode it simply places the numeric representation of the system time into the Stack-X register.

RELATED FUNCTIONS: DATE

NAME	TONE
KEYPRESS	(XEQ or CATALOG 3)
STACK	Enables Stack-lift, Allows indirection
LAST-X	Not applicable
DESCRIPTION:	

The TONE 'n' function sounds a tone corresponding to the number supplied. This supplied number must be within the range 0 through 9, the lower the number, the lower the tone generated.

NAME	VIEW
KEYPRESS	<>> (SHIFT CALC Mode)
STACK	Enables Stack-lift, Allows indirection
LAST-X	Not applicable
DESCRIPTION:	

The VIEW register 'nn' function is used to display the selected register and to print it's value, if flag 21 is set and the printer is on-line.

RELATED FUNCTIONS: ARROW KEYS (with NUM LOCK off)

NAME	WRTA (ELI-41SX)
KEYPRESS	(XEQ or CATALOG 3)
STACK	Enables Stack-lift
LAST-X	Not applicable
DESCRIPTION:	

The WRTA function uses the default memory directory set by DIRMEM and the file name set by the Alpha-register to save all status information, key assignments, and data storage registers on the disk storage. All existing files with the same name will be replaced otherwise new files will be created. The file name specified with the extension of .MRG will contain all data register contents, .MST files will contain status information and .MKY memory files will contain the ASCII formatted program key assignments.

WRTK (ELI-41SX) (XEQ or CATALOG 3) Enables Stack-lift Not applicable

The WRTK function writes the key assignments onto the memory file name specified in the Alpha-register in an ASCII format.

RELATED FUNCTIONS: None

NAME KEYPRESS STACK LAST-X DESCRIPTION: WRTR (ELI-41SX) (XEQ or CATALOG 3) Enables Stack-lift Not applicable

The WRTR function write the contents of all data storage registers to the .MRG file specified by the Alpha-register.

RELATED FUNCTIONS: None

NAME	WRTRX (ELI-41SX)
KEYPRESS	(XEQ or CATALOG 3)
STACK	Enables Stack-lift
LAST-X	Not applicable
DESCRIPTION:	

The WRTRX function uses the number in the Stack-X to specify the registers that the contents will be written to disk storage. Before executing this function the memory file storage and the pointers must be positioned by SEEKR function or a previous WRTRX, WRTR, READRX, or READR. The contents of the Stack-X registers are interpreted as bbb.eee, where bbb is the first register in the calculator to be written.

RELATED FUNCTIONS: WRTR

## WRTS (ELI-41SX)

NAME V KEYPRESS (2) STACK E LAST-X N DESCRIPTION:

WRTS (ELI-41SX) (XEQ or CATALOG 3) Enables Stack-lift Not applicable

The WRTS function writes the full status of the calculator to the memory file specified by the Alpha-register. The status of the calculator consists of :

flags 00 - 43 Stack-X register Stack-Y register Stack-Z register Stack-T register Stack-T register Alpha-register Statistical block register size of data registers no of digits display mode LPT number financial register

RELATED FUNCTIONS: None

NAME	<b>X</b> ↑ <b>2</b>
KEYPRESS	<c> (SHIFT CALC Mode)</c>
STACK	Enables Stack-lift
LAST-X	Precomputation Stack-X
DESCRIPTION:	-

The X SQUARED function multiplies the contents of the Stack-X register by itself and places the result in The Stack-X register.

RELATED FUNCTIONS: SQRT,  $Y \uparrow X$ 

**X#0?** (XEQ or CATALOG 3) Enables Stack-lift Not applicable

The is X NOT EQUAL 0? function causes the next sequential program line to be skipped if the Stack-X register is equal to zero during program execution.

RELATED FUNCTIONS: X<0?, X>0?, X#Y?, etc.

NAME	X#Y?
KEYPRESS	(XEQ or CATALOG 3)
STACK	Enables Stack-lift
LAST-X	Not applicable
DESCRIPTION:	

The is X NOT EQUAL TO Y? function causes the next sequential program line to be skipped if the Stack-X register is equal to Stack-Y during program execution.

RELATED FUNCTIONS: X#0?, X>0?, X<0?, etc.

NAME	X<0?
KEYPRESS	(XEQ or CATALOG 3)
STACK	Enables Stack-lift
LAST-X	Not applicable
DESCRIPTION:	••

The is X LESS THAN 0? function causes the next sequential program line to be skipped if the Stack-X register is greater than or equal to zero during program execution.

RELATED FUNCTIONS: X#0?, X>0?, X#Y?, etc.

NAME	X < = 0?
KEYPRESS	(XEQ or
STACK	Enables
LAST-X	Not appl
DESCRIPTION:	

X<=0? (XEQ or CATALOG 3) Enables Stack-lift Not applicable

The is X LESS THAN OR EQUAL TO 0? function causes the next sequential program line to be skipped if the Stack-X register is greater than 0 during program execution.

RELATED FUNCTIONS: X#0?, X>0?, X<0?, etc.

NAME	X<=Y?
KEYPRESS	<u> (SHIFT CALC Mode)</u>
STACK	Enables Stack-lift
LAST-X	Not applicable
DESCRIPTION:	

The is X LESS THAN OR EQUAL TO Y? function causes the next sequential program line to be skipped if the Stack-X register is greater than Stack-Y register during program execution.

RELATED FUNCTIONS: X=Y?, X=0?, etc.

NAME	X<>
KEYPRESS	(XEQ or CATALOG 3)
STACK	Enables Stack-lift
LAST-X	Not applicable
DESCRIPTION:	

The Stack-X EXCHANGE WITH 'nn' function exchanges the contents of the Stack-X register with specified register.

RELATED FUNCTIONS: X<>Y

X<>F (ELI-41SX) (XEQ or CATALOG 3) Enables Stack-lift Not applicable

Not Implemented Yet.

RELATED FUNCTIONS: N/A

NAME	X<>Y
KEYPRESS	<f> (CALC Mode)</f>
STACK	Enables Stack-lift
LAST-X	Not applicable
DESCRIPTION:	

The EXCHANGE X WITH Y function swaps the contents of the Stack-X and Stack-Y registers.

RELATED FUNCTIONS: X<> nn, R↑, RDN

NAME	X <y?< th=""></y?<>
KEYPRESS	(XEQ or CATALOG 3)
STACK	Enables Stack-lift
LAST-X	Not applicable
DESCRIPTION:	

The is X LESS THAN Y? function causes the next sequential program line to be skipped if the Stack-X register is greater than or equal to Stack-Y during program execution.

RELATED FUNCTIONS: X#0?, X>0?, X<0?, etc.

NAME	X=0?
KEYPRESS	<:> (SHIFT CALC Mode)
STACK	Enables Stack-lift
LAST-X	Not applicable
DESCRIPTION:	

The is X EQUAL 0? function causes the next sequential program line to be skipped if the Stack-X register is not equal to zero during program execution.

RELATED FUNCTIONS:  $X \le Y$ ?, X = Y?, etc.

NAME	X=Y?
KEYPRESS	<q> (SHIFT CALC Mode)</q>
STACK	Enables Stack-lift
LAST-X	Not applicable
DESCRIPTION:	

The IS X EQUAL TO Y? function tests the contents of the Stack-X register against the contents of the Stack-Y register for equality. The X=Y? function can also check for equality of alpha strings as well as numbers in both the Stack-X and Stack-Y registers. If both registers are equal then the program execution continues sequentially with the next program line, otherwise the next line of program will be skipped.

RELATED FUNCTIONS:  $X \le Y$ ?, X = Y?, X = 0?, etc.

NAME	X>0?
KEYPRESS	(XEQ or CATALOG 3)
STACK	Enables Stack-lift
LAST-X	Not applicable
DESCRIPTION:	

The is X GREATER THAN 0? function causes the next sequential program line to be skipped if the Stack-X register is less than or equal to zero during program execution.

RELATED FUNCTIONS: X#0?, X<0?, X#Y?, etc.

NAME	X>Y?
KEYPRESS	<y> (SHIFT CALC Mode)</y>
STACK	Enables Stack-lift
LAST-X	Not applicable
DESCRIPTION:	

The IS X GREATER THAN Y? function causes the next sequential program line to be skipped if the Stack-X register is less than or equal to Stack-Y register during program execution.

RELATED FUNCTIONS: X<=Y?, X=Y?, X=0?, etc.

NAME	XEQ
KEYPRESS	<k> (CALC Mode)</k>
STACK	Enables Stack-lift. Allows indirection
LAST-X	Not applicable
DESCRIPTION:	

The EXECUTE function allows the user to execute any of the calculator's internal functions or the user's generated program files. XEQ function needs additional parameters to perform its function. Upon pressing the  $\langle K \rangle$  key, the data entry area will display XEQ \_\_\_\_\_ message requesting a two digit subroutine label designator. By pressing the  $\langle F9 \rangle$  (ALPHA) key the calculator face will switch to ALPHA mode and user will be able to enter a program or function name, composed of up to seven alphanumeric characters. The alpha entry must be terminated by pressing the  $\langle F9 \rangle$  key or simply pressing the  $\langle Enter \rangle$  key.

The XEQ function will search the internal function table or the program directory in the following sequence in order to be able to execute the function or program requested:

1. Available keystroke functions that could be executed by a keypress function in CALC, SHIFT CALC, and SHIFT ALPHA modes.

X>Y?

- 2. Internal calculator functions that have not been assigned to any of the keys in the 4 calculator modes. Refer to internal functions section of this chapter for the available functions.
- 3. Any program file residing in the PROGRAM directory by the same name as that entered (without the file extension).

In the event that the function or program is not found according to the above rules, the NONEXISTENT error message will be displayed. The error message can only be cleared by pressing the Back Space key, which is indicated in the instruction area.

The XEQ function also allows for indirect execution of programs by storing the program name in any of the memory registers and issuing the XEQ IND nn command by pressing  $\langle K \rangle \langle F10 \rangle$  (SHIFT) key and the register number that contains the program name or the subroutine label number.

#### EXAMPLE:

<k> XEQ <f9> XEQ CHANGE TO ALPHA MOD &lt;"MARY"&gt; XEQ MARY_ DESIRED PROGRAM NAM</f9></k>	Y	ENTS
<f9> OR <enter<sup>†&gt; Executing Program!</enter<sup></f9>	K> F9> "MARY" > F9> OR < EN	SE TO ALPHA MODE ED PROGRAM NAME ing Program!

#### **PROGRAMMING EXAMPLE:**

LEXMPL11	; ARBITRARY LABEL
XMARY	; DENOTES XEQ MARY
AMARY	; PLACES MARY IN ALPHA REGISTER
TASTO 00	; STORES MARY IN REGISTER 00
XIND 00	; DENOTES XEQ IND 00
AMARY	; PLACES MARY IN ALPHA REGISTER
TASTO STK T	; STORES MARY IN STACK-T
XIND STK T ; I SEND	NDIRECTLY EXECUTES MARY THROUGH STACK-T

RELATED FUNCTIONS: GTO, SST, CATALOG, LBL, COPY, EDIT

NAME	XOR
KEYPRESS	(XEQ or CATALOG 3)
STACK	Enables Stack-lift
LAST-X	Precomputational Stack-X
DESCRIPTION:	-

The EXCLUSIVE OR function performs a boolean XOR between the integer portion of the Stack-X and Stack-Y registers and places the result in the Stack-X register.

RELATED FUNCTIONS: AND, OR, NOT, <<, >>

NAME	XTOA (ELI-41SX)
KEYPRESS	(XEQ or CATALOG 3)
STACK	Enables Stack-lift
LAST-X	Not applicable
DESCRIPTION:	

The XTOA function uses the contents of the Stack-X register to append one character to the right-hand end of the Alpha-register. If the Stack-X register contains a text string, then the text string is appended to the Alpha-register.

RELATED FUNCTIONS: ATOX, AROT, ALENG, POSA

NAME	Y↑X
KEYPRESS	<b> (SHIFT CALC Mode)</b>
STACK	Enables Stack-lift, Stack popped
LAST-X	Precomputation Stack-X
DESCRIPTION:	-

The RAISE Y TO POWER OF X function raises the contents of the Stack-Y register to the power of the Stack-X register and result is placed in the Stack-X register. This function could also be used to find the nth Root of the X by using the following method :

```
X
<Enter>
n
```

[1/X] [Y†X]

RELATED FUNCTIONS: X<sup>1</sup>2

NAMEΣ+KEYPRESS<A> (CALC Mode)STACKDisables Stack-liftLAST-XPrecomputation Stack-XDESCRIPTION:

The SUM PLUS function is a statistical function that utilizes six of the memory registers called the "Statistical Block". The default beginning register of the statistical block (stat\_reg) is the memory register 11, and can be changed prior to the use of this function by the  $\Sigma$ REG function.

This function assumes two entries are being made. The "x" entry is the Stack-X register and the "y" entry is the Stack-Y register. Performing this function will accumulate the registers mentioned into the statistical block leaving the number in the Stack-Y register (the "y" component) unchanged, and placing the number of the current entry sets in the Stack-X register. The following formulas reflect the actions taken upon the statistical block:

= X
= (X↑2)
= Y
= (Y↑2)
$= (X \star Y)$
= number of accumulated entries

## EXAMPLE: See Example D-1.

RELATED FUNCTIONS:  $\Sigma$ -,  $\Sigma$ REG,  $CL\Sigma$ , MEAN, SDEV

NAME	Σ-
KEYPRESS	<a> (SHIFT CALC Mode)</a>
STACK	Disables Stack-lift
LAST-X	Precomputation Stack-X
DESCRIPTION:	-

The SUM MINUS function removes an x,y pair from the accumulated statistical block registers. Stack-X must contain the x element and Stack-Y must contain the y element of the data pair to be removed.

## EXAMPLE: See Example D-1.

RELATED FUNCTIONS:  $\Sigma$ +,  $\Sigma$ REG, CL $\Sigma$ , MEAN, SDEV

NAME	∑REG
KEYPRESS	(XEQ or CATALOG 3)
STACK	Enables Stack-lift, Allows indirection
LAST-X	Not applicable
DESCRIPTION:	••

The SUM REG 'nn' function sets the beginning register of the six (6) register statistical block to 'nn'. Any statistical functions used from that point on will use the new statistical block rather than the default statistical block beginning at register 11.

RELATED FUNCTIONS:  $\Sigma$ +,  $\Sigma$ -, MEAN, CL $\Sigma$ , SDEV

NAME	$\Sigma REG?$ (ELI-41SX)
KEYPRESS	(XEQ or CATALOG 3)
STACK	Enables Stack-lift
LAST-X	Not applicable
DESCRIPTION:	

The  $\Sigma$ REG? function returns the current setting for the beginning register of the six register statistical block in the Stack-X register.

RELATED FUNCTIONS:  $\Sigma$ +,  $\Sigma$ -, MEAN, CL $\Sigma$ , SDEV,  $\Sigma$ REG

(This page is intentionally left blank)

# **APPENDIX A**

## THE ELI-41 FLAGS

Sixty flags are continuously displayed on the calculator screen. An "ON" status is indicated by a small box while an "OFF" status is displayed as a centered dot.

Flags are Set by the "SF" command and are Cleared by the "CF" command. Both of these commands are available in the SHIFT-CALC mode. If you attempt to manipulate a flag that is either reserved for internal use, or is not available for use by the user a "NONEXISTENT" error message will be displayed. Flags 0 through 20 are not pre-defined and are available throughout the program usage.

Other flags that can be Set/reSet by the user but have been pre-defined for special usage are as follows:

Flag	Usage
15	TRACE
16	TRACE
17	CRLF1
18	CRLF2
21	Printer Activation
24	Range Error Ignore
25	Error Ignore
26	Audible Tone Enable/Disable
27	USER mode
28	Separator (Used for Comma Editing)
32	MANIO flag

47	SHIFT
48	ALPHA mode

The remaining flags are used by ELI-41 for a number of purposes. They can be tested for "ON/OFF" status, but cannot be Set ON by any user commands.

The following is a complete list of the flags used by the calculator. Those, that are user changeable, are marked by an asterisk.

*00	User
*01	User
*02	User
*03	User
*04	User
*05	User
*06	User
*07	User
*08	User
*09	User
*10	User
*11	User
*12	User
*13	User
*14	User
*15	User
*16	User

## \*17 CRLF1

\*18 CRLF2

The CRLF1 and CRLF2 flags specify the end of string charaters that will be used when transmitting strings in and out for the HP-IL, IEEE-488, and the RS232 options.

To specify the eos string user must set or reset flags 17 and 18 as follows:
17	18	eos string appended
OFF	OFF	CR-LF (Default)
OFF	ON	LF
ON	OFF	Byte represented by the contents of the Stack-X register
ON	ON	NULL (Nothing appended)

\*20 User

#### \*21 Printer Enable

This flag controls printer operation and could be Set with SF 21 function, or using the Options Print On menu. Once Set and if flag 55 is Set meaning that a printer is attached and is On-Line, then AVIEW and VIEW functions print their results, or stop a programs execution if the flag 55 is Cleared.

#### \*22 Numeric Data Input

Set if a number has been entered into the Stack-X register, even if Stack-X is then Cleared. It is used to test for number entry, but should be Cleared before next use.

#### \*23 Alpha Data Input

Set if a character has been entered into the Alpha-register, even if Alpha-register is then Cleared. It is used to test for character entry, but should be Cleared before next use.

#### \*24 Range Error Ignore

If Set, the OUT OF RANGE errors will be ignored. (ELI-41SX)

#### \*25 Error Ignore

If Set, any errors that occur will be ignored. After an error is ignored this flag is reSet automatically.

#### \*26 Audible Tone Enable

If Cleared, the TONE/BEEP functions will be disabled. (ELI-41SX)

#### \*27 USER Mode

If Set, the calculator will automatically switch to the USER mode.

#### \*28 Separator

If Set, a decimal point separates the integer and fractional parts of numbers and a comma acts as a digit grouping mark. If Cleared a comma will be the separator of fractional and integer portion of numbers and decimal point will mark digit grouping of numbers.

#### \*29 Digit grouping

If Set, groups of 3 digits are separated by grouping marks (comma or decimal point depending on state of flag 28). If reSet, the grouping marks are suppressed. If this flag is reSet and the number display mode has been Set with FIX 00 function then grouping marks and the decimal point separator are suppressed.

#### 30 Catalog Activated

Set internally once the two column catalog display is activated using the CATALOG function.

- 31 ..Internal Use..
- 32 ..Internal Use..
- 33 ..Internal Use..
- 34 ..Internal Use..
- 35 ..Internal Use..
- 36 NO OF DIGITS (LSB)
- **37** NO OF DIGITS
- **38** NO OF DIGITS
- 39 NO OF DIGITS (MSB)

The current setting for number of decimal digits is encoded in the hexadecimal format using flags 36,37,38, and 39(MSB).

### 40 Display Format (FIX,ENG,SCI)

### 41 Display Format (FIX,ENG,SCI)

The number display mode is effected by status of the flags 40 and 41. If both flags are Cleared then SCI mode is in effect. Flag 40 Set and 41 Clear makes FIX mode active and 40 Clear and 41 Set make ENG mode active.

## 42 Angle Mode (DEG, RAD, GRADS)

## 43 Angle Mode (DEG, RAD, GRADS)

The angle mode is represented by the flags 42 and 43. DEGREE mode is in effect if flags 42 and 43 are Clear. RADIANS mode is in effect if flag 42 is Clear and 43 is Set and also if both flags are Set. GRADIANS mode is in effect if flag 42 is Set and 43 is Clear.

- 44 ..Internal Use..
- 45 ..Internal Use..
- 46 ..Internal Use..

#### \*47 Shift Mode

Set when the calculator is in SHIFT mode (<F10> key).

#### \*48 Alpha Mode

Set when the calculator is in ALPHA mode (<F10> key).

#### 49 ..Internal Use..

#### 50 Message Display

Internally Set when a message is displayed by the user via PROMPT, VIEW, or AVIEW function and when the ELI-41 is displaying a message such as an error. (ELI-41SX)

#### 51 SST

Internally Set during Single Step mode but it is always Cleared during program execution. It can be tested using the FC? 51 function. (ELI-41SX)

#### 52 EDIT Mode

Internally Set during program editing, but not during program execution. (ELI-41SX)

#### 53 ..Internal Use..

#### 54 Pause

Internally Set during a pause.

#### 55 Printer Existence

At any time after any function is executed or a change of modes occur the printer attached to the PC is tested to see if it is On-Line and active. If it is, then flag 55 is Set automatically otherwise, flag 55 is Cleared. This feature is overridden when the output is redirected to an ASCII file in the ELI-41SX program.

- 56 Numeric Lock
- 57 Caps Lock
- 58 ..Internal Use..
- 59 Execute

(This page is intentionally left blank)

# **APPENDIX B**

# FUNCTION KEYBOARD USAGE

<f1></f1>	
	Display contents of Stack-X in BASE 10 (Decimal).
<f2></f2>	
	Display contents of Stack-X in BASE 16 (Hex).
<f3></f3>	Display contents of Stack-X in BASE 2 (Binary).
<f4></f4>	
	Display contents of Stack-X in BASE 8 (Octal).
<f5></f5>	
	Go to System Options Screen.
<f6></f6>	Perform Data Value Conversions (using " CNN/" disk file)
~ <b>D7 b</b>	Perior in Data value conversions (using .Civv disk me)
< ſ / >	Program Editor.
<f8></f8>	
	Switch to User mode.

#### <F9>

Switch to Alpha mode.

#### <F10>

Mode Shift Key (By pressing the Mode Shift Key in any mode, the mode will be shifted. A second press of the Mode Shift key will return to the un-shifted mode).

#### <Alt-F1>

Exit from the calculator.

#### <Alt-F2>

Import number from foreground application screen. (ELI-41 POP-UP mode)

Status display. (ELI-41SX -01, -02, -03, -05)

#### <Alt-F3>

(Block Delete) Deletes from the Beginning line of the marked block down to and including the current highlighted program line. (ELI-41SX)

#### <Alt-F4>

(Block Insert) Inserts the contents of the holding buffer before the current line. (ELI-41SX)

#### <Alt-F5>

(Block Copy) Copies the marked block into the temporary holding buffer for use with <Alt-F3> or <Alt-F4>. After the beginning line of the block is marked (<Alt-F6>), use the down arrow key to highlight the ending line of the block and then press the <Alt-F5> keys to copy the marked block into the temporary buffer. (ELI-41SX)

#### <Alt-F6>

(Block Mark) Marks a block of program lines to be Copied to temporary holding buffer(<Alt-F5>), Inserted at any point within the current program or any other program edited(<Alt-F4>), or Deleted(<Alt-F3>). To use this feature, use the arrow keys to highlight the upper line of the block and then press the <Alt-F6> keys to begin marking the block. This action marks the beginning line of the Block. (ELI-41SX)

#### <Alt-F7>

(Save File To Disk) Saves the contents of the program buffer currently being edited to disk.

#### <Alt-F8>

(Comments Editor) This feature allows users to comment each and every line of the program code.

#### <Alt-F9>

(Search and Replace) This feature allows Replacement of any program line with a program line supplied by user. (ELI-41SX)

#### <Alt-F10>

(Search Only) This feature searches for a string given by the user from the current line down to the end of the program buffer. (ELI-41SX)

#### <Ctrl-F1>

Exit from the calculator and return to graphics page 1. (POP-UP mode)

(This page is intentionally left blank)

## **APPENDIX C**

## **FUNCTION TABLES**

The following abbreviations are used in the tables:

- D = Stack-lift is disabled.
- E = Stack-lift is enabled.
- I = The function can use indirect parameter using the specified register.
- L = The precomputational Stack-X is stored in the Last-X register.
- P = The Stack registers are popped.
- U = The Stack registers are pushed.
- N/A = Not appicable.
- nn = Memory register nn.
- 1 = Local or Global Label.

Func Name	Description	Stack
%	Takes percent (Stack-X) of Stack-Y	. E,L
%CH	Percent change from Stack-Y to Stack-X	. E,L
*	Stack-Y times Stack-X	. E,P,L
+	Stack-Y plus Stack-X	. E,P,L
	Stack-Y minus Stack-X	. E,P,L
/	Stack-Y divided by Stack-X	. E,P,L
1/X	Reciprocal of Stack-X	. E,L
10†X	Common Exponential	. E,L
<< n	Shift Stack-X left n bits (boolean)	. E,L,I
>> n	Shift Stack-X right n bits (boolean)	. E,L,I
ABS	Absolute value of Stack-X	. E,L
ACA	Copy Alpha register to the print buffer	. E

ACCBYTE	Append unprintable ASCII characters to the Alpha
	register
ACCHR	Append a character to the print buffer E
ACOS	Arc cosine function
ACX	Append X-register to the print buffer E
ADV	Advances paper and prints the buffer E
ALENG	(ELI-41SX)
AND	Stack-X = Stack-X ANDed with Stack-Y $\ldots E,L$
ANUM	(ELI-41SX)
AOFF	Alpha mode OFF
AON	Alpha mode ON
ARCL nn	Appends contents of reg nn to ALPHA reg E,I
AROT	(ELI-41SX)
ASHF	Shift the ALPHA reg Left 6 characters E
ASIN	Arc sine function
ASN 1,k	Assign function to a key
ASTO nn	Stores first 6 char of ALPHA in reg nn E,I
ATAN	Arc tangent function
ATOF	Convert the ASCII string in Alpha register to float E,L
ATOX	(ELI-41SX)
AVIEW	Views contents of ALPHA register
BEEP	Sound four tones
BIN	Switches data entry/display to binary E
BREAK	Sets a breakpoint in program (debug) N/A
CAT n	Displays catalog of PGM, MEM, CNV files N/A
CF nn	Clear flag nn (OFF) $00 \le nn \le 29$ E,I
CHG2ASC	Convert the bytes in Alpha register to ASCII E,L
CHS	Change the sign of Stack-X or the exp E
CLA	Clears contents of ALPHA register
CLD	Clears the display
CLKEYS	(ELI-41SX)
CLP	Clears program Buffer
CLRG	Clears the memory registers
CLRGX	(ELI-41SX)
CLST	Clears the Stack registers
CLX	Clear Stack-X register
CLΣ	Clears the stat block
COPY 1	Load program from disk into PGM buffer N/A
COS	Cosine of Stack-X
COSH	(ELI-41SX)

CREATE	(ELI-41SX)
D-R	Degrees to radians conversion
DATE	Displays the current date
DEC	Switches data entry/display to decimal E
DEG	Switches angle measurement to degrees E
DIRCNV	(ELI-41SX)
DIRUTL	(ELI-41SX)
DIRMEM	(ELI-41SX)
DIRPGM	(ELI-41SX)
DIRPRT	(ELI-41SX)
DSE nn	Decrement and skip if Equal
DWELL	(ELI-41SX)
EEX	Add the exponent to number in data entry N/A
END	Marks End of Program
ENG nn	Display nn significant digits Eng
ENTER†	Copy content of Stack-X into Stack-Y D,U
EXEC	(ELI-41SX)
E↑X	Natural Exponential
E∱X—1	Natural Exponential of number close to 0 E,L
FACT	Stack-X! factorial ( $0 \le X \le 150$ ) E,L
FC? nn	Skips prog line Unless flag nn is clear E,I
FC?C nn	Skips line if flag nn is set, & clears
FIX nn	Display nn significant digits Fixed
FRC	Stack-X equals decimal part of Stack-X E,L
FS? nn	Skip unless flag nn is ON (00 $\leq$ nn $\leq$ 55) E,I
FS?C nn	Skips line if flag nn is clear, & clears E,I
GETKEY	(ELI-41SX)
GETKEYX	(ELI-41SX)
GRAD	Switches angle measurement to gradians E
GTO nn	Set internal pointer to line or label E,I
HEX	Switches data entry/disp to hexadecimal E
HMS	Decimal hours to hours-minutes-seconds E,L
HMS+	Hours-min-sec addition
HMS—	Hours-min-sec subtraction
HR	Hours-min-sec to decimal hours
INPORT	Read from I/O port
INT	Integer portion of Stack-X
ISG nn	Increment And Skip If Greater
LASTX	Stack-X register equals LastX
LBL nn	Global or Local Label (pgm mode only) E

LN	Natural Logarithm of the Stack-X
LN1+X	Natural logarithm of numbers close to 1 E,L
LOG	Common Logarithm of the Stack-X
LPT#	(ELI-41SX)
MEAN	Mean of stat block
MOD	Stack-Y mod Stack-X (Remainder)
NDIG?	(ELI-41SX)
NOT	Complement Stack-X register (boolean) E,L
OCT	Switches data entry/display to octal
OFF	Exits calculator (pop-up remains res) N/A
OR	Stack-X = Stack-X ORed with Stack-Y $\ldots \ldots E,L$
OUTPORT	Write to I/O port
P-R	Polar to Rectangular conversion
PASN	(ELI-41SX)
PI	Stack-X equal to PI
PORTSEL	Select the current port for I/O reads and writes . E
POSA	(ELI-41SX)
PRA	Print the contents of the Alpha-register E
PRBUF	Print the print buffer and clear it
PRFLAG	Print the condition of all 60 flags
PROMPT	Displays Alpha register and Halts pgm E
PRP I	Print the program (1) with line numbers E
PRREG	Prints all nonzero registers
PRSTK	Prints Stack X, Y, Z, T, L registers
PRX	Prints the Stack-X register
PSE	Holds display (1 sec) during prog exec E
PSIZE	(ELI-41SX)
QEDIT	(ELI-41SX)
R-D	Convert Stack-X from radians to degrees E,L
R-P	Rectangular to Polar conversion
RAD	Switches angle measurement to radians E
RAN#	Generate random number
RCL nn	Copy content of register nn into Stack-X E,I,U
RCLFLAG	(ELI-41SX)
RDN	Rolls the contents of Stack regs down E
READA	(ELI-41SX)
READK	(ELI-41SX)
READR	(ELI-41SX)
READRX	(ELI-41SX)
READS	(ELI-41SX)

READSUB	(ELI-41SX)
REGMOVE	(ELI-41SX)
REGSWAP	(ELI-41SX)
RND	Round of Stack-X
RTN	Return from Subroutine or End Program E
R†	Rolls stack up 1 register E
SĊI nn	Display nn significant digits Scientific E,I
SDEV	Standard deviation of stat block
SEED	(ELI-41SX)
SEEKR	(ELI-41SX)
SF nn	Set flag nn (ON) $00 \le nn \le 29 \dots E, I$
SIGN	Stack-X register's sign
SIN	Sine of Stack-X
SINH	(ELI-41SX)
SIZE?	(ELI-41SX)
SKPCHR	Sends spaces to printer line buffer E
SQRT	Square root of Stack-X
SST	Single step mode
ST* nn	Multiply Register nn by Stack-X
ST+ nn	Add Stack-X to register nn
ST— nn	Subtract Stack-X from register nn
ST/ nn	Divide register nn by Stack-X
STO nn	Stores content of Stack-X in register nn E,I
STOFLAG	(ELI-41SX)
STOP	Halt or continue program execution E
TAN	Tangent of Stack-X
TANH	(ELI-41SX)
TIME	Displays the current time
TONE n	Sound tone (0 to 9)
VIEW nn	View the nn register
WRTA	(ELI-41SX)
WRTK	(ELI-41SX)
WRTR	(ELI-41SX)
WRTRX	(ELI-41SX)
WRTS	(ELI-41SX)
X†2	Squares the Stack-X register
X#0?	Skip line unless Stack-X not equal 0 E
X#Y?	Skip line unless Stack-X not equal Y E
X<0?	Skip line unless Stack-X less than 0 E
X<=0?	Skip line unless Stack-X less or equal 0 E

X<=Y?	Skip unless Stack-X less or equal to Y E
X<> nn	Exchange Stack-X with Register nn
X<>F	(ELI-41SX)
X<>Y	Exchange contents of Stack-X w/ Stack-Y E
X <y?< td=""><td>Skip line unless Stack-X less than Y E</td></y?<>	Skip line unless Stack-X less than Y E
X=0?	Skip line unless Stack-X equal to zero E
X=Y?	Skip unless Stack-X equal to Stack-Y E
X>0?	Skip line unless Stack-X greater than 0 E
X>Y?	Skip unless Stack-X Greater than Stack-Y E
XEQ nn,l	Execute a function or program
XOR	Stack-x = Stack-X XORed with Stack-Y E,L
XTOA	(ELI-41SX)
Y∱X	Raises the Stack-Y to power of Stack-X E,P,L
Σ+	Accumulates X & Y regs into stat block D,L
Σ	Removes last entries from stat block D,L
ΣREG nn	Set stat block beginning register
ΣREG?	(ELI-41SX)

# **APPENDIX D**

## THE EXAMPLES

#### Example D-1:

A series of measurements of the length of a concrete runway is made using a steel tape. The results (in meters) are tabulated below:

1363.7	1364.5	1364.0
1363.8	1364.0	1364.1

Compute the mean and the standard deviation for the measurements.

KEY PRESS	DISPLAY	DESCRIPTION
<f10></f10>	0.0	SHIFT CALC
<f></f>	0.0	$CL\Sigma$
1363.7	1363.7	
<a></a>	1.0 -	$\Sigma$ +
1364.5	1364.5	
<a></a>	2.0 -	$\Sigma +$
1364.0	1364.0	
<a></a>	3.0 -	$\Sigma$ +
1363.8	1363.8	
<a></a>	4.0 -	$\Sigma$ +
1364.0	1364.0	
<a></a>	5.0 -	$\Sigma$ +
1464.1	1464.1_	
<a></a>	6.0	<b>OH! TYPING ERROR</b>
1464.1	1464.1	
<f10></f10>	1464.1	SHIFT CALC
<a></a>	5.0	Σ-

1364.1	1364.1_	
<a></a>	6.0	$\Sigma$ +
<k></k>	XEQ	
"MEAN"	XEQ MEAN	
<enter></enter>	1,364.01666667	MEAN OF THE DATA
<k></k>	XEQ	
"SDEV"	XEQ SDEV	
<enter></enter>	0.278687398	STANDARD DEVIATION
		OF DATA

#### **PROGRAMMING EXAMPLE:**

LSTATTST TΣREG 20 SCLΣ N1363.7  $S\Sigma +$ N1364.5  $S\Sigma +$ N1364.0  $S\Sigma +$ N1363.8  $S\Sigma +$ N1364.0  $S\Sigma +$ N1364.1  $S\Sigma +$ SMEAN AMEAN= TARCL STK X SPROMPT SSDEV ASDEV= TARCL STK X SPROMPT SEND

#### Example D-2:

Find the square root of 16 and the square of the result. NOTE: Square root only accepts positive numbers.

KEY PRESS	DISPLAY	DESCRIPTION
16	16	
<c></c>	4.0	SQUARE ROOT
<f10></f10>	4.0	SHIFT CALC
<c></c>	16.0	SQUARE
<0>	-16.0	CHANGE SIGN
<c></c>	DATA ERROR!	NUMBER MUST $BE = 0.0$

#### Example D-3:

Find the logarithm and the natural logarithm of 13 also 10 and (e) to the power of Stack-X.

KEY PRESS	DISPLAY	DESCRIPTION
13	13	NON NEGATIVE NUMBER
<d></d>	1.113943352	LOG BASE 10 OF 13
<f10></f10>	1.113943352	SHIFT CALC
<d></d>	13.0	10 TO THE POWER OF X
<e></e>	2.564949357	LOG BASE e(2.718) OF 13
<f10></f10>	2.564949357	SHIFT CALC
<e></e>	13.0	e TO THE POWER OF X

#### Example D-4:

Find the sine of PI/4 in radians. What is the angle whose sine is 0.7854?

KEY PRESS	DISPLAY	DESCRIPTION
<k></k>	XEQ	
<rad></rad>	0.0	CHANGE MODE TO RADIANS
<f10></f10>	0.0	SHIFT CALC
<space bar=""></space>	3.141592654	PI

4	4_	
	0.785398163	
<h></h>	0.707106781	SINE OF PI/4
<f10></f10>	0.707106781	SHIFT CALC
<h></h>	0.785398163	ARC SINE OF 0.707106781

#### Example D-5:

Following is an example of different forms of executing store and recall shown in ASCII format.

LSTOTEST	
N10	
TSTO 00	; STORE 10 IN REG 00
TSTO IND 00	; STORE 10 IN REG 10
TSTO STK T	; STORE 10 IN STACK-T
TSTO IND STK	T; STORE 10 IN REG 10
TST+ 00	; ADD STACK-X TO REG 00
TST+ IND 00	; ADD STACK-X TO REG POINTED TO BY REG 00
TST+ STK T	; ADD STACK-X TO STACK-T
TST+ IND STR	(T; ADD STACK-X TO REG POINTED TO BY
TRCL 00	; STACK-T
TRCL IND 00	
TRCL STK T	
TRCL IND STK	Ϋ́Τ
SEND	

#### Example D-6:

The following example is supplied to show the use of EEX and CHS in data entry.

KEY PRESS	DISPLAY	DESCRIPTION
234.567	234.567_	
<0>	-234.567	CHANGE SIGN
<p></p>	-234.567E	EEX
32	-234.567E32	EXPONENT ENTRY
<0>	-234.567E-32	NEGATIVE EXPONENT
<enter></enter>	-2.345670000E-30	<u> </u>

# **APPENDIX E**

## **COMMAND LINE SWITCHES**

Upon running the ELI-41/41SX program certain command line switches could be included to alter the functionality of the calculator. The available switches are as follows:

/A : Forces the insert after the current line while the calculator is in the EDIT mode. With this switch set the calculator's program editor will behave exactly like the HP41 which inserts the program line after the viewed instruction. (ELI-41SX)

/B : Forces the calculator to use screen attributes suited for Composite screen monitors. This feature is useful when you have a composite monitor attached to a color card. (ELI-41SX)

/P : This switch causes the ELI-41 program to be loaded as a pop-up. Note that the ELI-41SX program is not a pop-up application program.

/R : This switch causes the stack display to be reversed so that Stack-X, Y, Z, and T are shown similar to the views that HP41 users are accustomed to. It also effects the Stack printout while tracing program execution. (ELI-41SX)

/N : This switch disables the calculator control of the <Num Lock> key.

/X : Hides the initial copyright screen upon program entry.

(This page is intentionally left blank)

# Index

### !

"CAPS"	2-6	ABS	8-4
"NUM"	2-6	ACA	8-4
"PRT"	2-6	ACCBYTE	8-4
"WAIT"	2-6	ACCHR	8-5
%	8-2	ACOS	8-5
%CH	8-2	activating keypress	2-7
*.CNV	2-3, 6-1	ACX	8-5
*.MEM	2-3	ADV	8-6
*.MKY	2-3	ALENG	8-6
*.MRG	2-3	ALPHA	2-5 - 2-6, 3-1
*.MST	2-3	Alpha-register	1-3, <b>2-4</b>
*.OPT	2-2	Alt-F1	2-2, 2-7, 4-3
*.PGM	2-2	Alt-F10	4-6
/В	2-3	Alt-F2	2-8
/n	2-1	Alt-F3	2-2, 2-7, 4-4
/P	2-1, 2-7	Alt-F4	4-4
1/X	8-2	Alt-F5	4-5
10 <b>↑</b> X	8-3	Alt-F6	4-5
$\Sigma +$	8-74	Alt-F7	4-3, 4-5
Σ-	8-75	Alt-F8	4-3, 4-5
ΣREG	8-75	Alt-F9	4-5
$\Sigma REG?$	8-75	AND	8-6
<<	8-3	Annunciator Area	2-5
>>	8-3	ANUM	8-7
		AOFF	8-7
		AON	8-7
		APP	8-8
		ARCL	8-8
		AROT	8-8
		ASCII	2-2, 6-1

A

ASHF	8-9	COS	8-21
ASIN	8-9	COSH	8-21
ASN	8-10	CREATE	8-22
ASTO	8-11	Ctrl-'	2-8
ATAN	8-11	Ctrl-~	1-1
ATOF	8-11	Ctrl-Shift-'	2-8
ATOX	8-12	Cursor	2-5, 6-1, 7-2
AUTOEND.PGM	2-9		
AUTORUN.PGM	2-9	n	
AVIEW	8-13	D	
		D-R	8-22
B		Data Entry Area	2-5
		DATE	8-23
Back Space	1-4, 3-2 ,	DEC	2-5, 8-23
-	4-2, 6-3, 8-13	BDECIMAL	3-3
BEEP	8-14	DEG	2-5, 8-23
BIN	2-5, 8-14	DESQview	1-3
BREAK	8-14	DIRCNV	8-24
		DIRMEM	8-24
C		DIRPGM	8-25
C		DIRPRT	8-25
CALC	9.5	DIRUTL	8-24
CALC	2-0	DoubleDos	1-3
catalog function	6-15 6-1	Down	2-5
	0-1	DSE	8-26
CHCOASC	0-10 9 16	DWELL	8-26
CHGZASC	0-10		
	0.10		
	8-18		
CLKEVS	8-18	F	
CLP	8-19	L	
CLRG	8-19	F*X-1	8-29
CLRGX	8-19	E A Y	8-29
CLST	8-10 8-20	EDIT mode	4-1
CT2	8-20	EEX	3-3 8-97
CLX	8-20	ELI-41	1-1
COMPOSITOPT	2-20	ELI41 EXE	<b>2-1</b>
conversion table	6-1	ELI41 OPT	2-1 - 2-2
COPY	3-1 8-21	End	2.5 2.8 8.97
~~	- 1, - 21		2 0, 2 0, 0 21

ENG	3-6, 8-27	graphics page 1	2-2, 2-7
Enter	1-3, 2-6, 2-8	GTO	8-33
ENTER+	, 3-2, 0-1, 7	2 TT	
Esc	7-2	H	
EXEC	8-28		
exit ELI-41	2-2	HEX	2-5, 8-33
		HEXADECIMAL	3-4
Г		HMS	8-34
Γ		HMS+	8-34
E1		HMS-	8-34
	3-3	Home	2-5, 2-8
FIU	2-0	HR	8-35
FZ F2	3-4 2 4		
го Га	3-4	Ι	
r4 r5	3-4 7 1		
FD FC	7-1 6 1	import function	2-8
F0 F7	0-1	INPORT	8-35
	4-1	INT	8-36
FO	2-0	ISG	8-36
r9	2-0, 3-2,		
FACT	4-2, 0-7	T.	
FAC I	0-29 8 30	L	
FC2C	8 30	Last-X	1-3
FUY	3-5 8-30	LASTX	8-37
flag status boy	0-0, 0-00 2-5	LBL	3-1 8-37
Flage	2-5	LN	8-37
FRC	2-3 8-31	LN1+X	8-38
FS?	8-31	LOG	8-38
FS?C	8-31	LPT1	5-1
function keys	2-5	LPT#	8-38
nunetion nego	20		
G		Μ	
CETKEV	8-30	MEAN	8-39
GETKEVY	8-32	memory resident	1-1. 2-1. 2-7
GRAD	2-5 8-32	MOD	8-39
draphics page 0	2-0, 0-02		2.00
Problem Page o	<i>∟ ∟</i> , <i>∟</i> -1		

Ν		Q	
NDIG? NOT	8-39 8-40	<b>QEDIT</b>	8-48
Num Lock	1-1, 2-1 ,2-5 - 2-6	R	
		R-D	8-49
0		R-P	8-49
		RAD	2-5, 8-49
OCT	2-5, 8-40	RAN#	8-50
OFF	2-2, 8-40	RCL	8-50
Options Menu	7-1	RCLFLAG	8-50
OR	8-41	RDN	8-51
OUTPORT	8-41	READ.ME	1-2, 2-4, 4-1
		READA	8-51
D		READK	8-51
I		READR	8-52
D_D	8-42	READRX	8-52
PASN	8-42	READS	8-52
PoDn	2-5	READSUB	8-53
Polln	2-5	REGMOVE	8-53
PI	8-43	REGSWAP	8-53
PORTSEL	8-43	RND	8-54
POSA	8-44	RPN	1-3
PRA	8-44	RTN	8-54
PRBUF	8-45	RUN/STOP	8-54
PRFLAG	8-45	Rţ	8-55
PROMPT	8-45		
PRP	3-1. 8-46	S	
PRREG	8-46	~	
PRSTK	8-47	SCI	3-6, 8-55
PRX	8-47	SDEV	8-55
PSE	8-47	SEED	8-56
PSIZE	8-48	SEEKR	8-56
		SF	1-1, 1-3 - 1-5, 8-56
		SHIFT ALPHA	2-5
		SHIFT CALC	2-5

SHIFT USER	2-5		
SIGN	8-57	W	
SIN	8-57	**	
SINH	8-57	WRTA	8-64
SIZE?	8-58	WRTK	8-65
SKPCHR	8-58	WRTR	8-65
SQRT	8-58	WRTRX	8-65
SST	8-59	WRTS	8-66
ST*	8-60	WICIO	0.00
ST+	8-59	<b>X</b> 7	
ST-	8-60	X	
ST/	8-60		
Stack	2-4	X#0?	8-67
Stack-T	1-3	X#Y?	8-67
Stack-X	1-3 - 1-4, 2-8	X<0?	8-67
Stack-Y	1-3	X∱2	8-66
Stack-Z	1-3	X<>F	8-69
STO	8-61	X<>Y	8-69
STOFLAG	8-62	X>Y?	8-71
STOP	8-62	X=0?	8-70
		X=Y?	8-70
т		X<>	8-68
1		XEQ	2-2, 3-1
ጥላ እ፤	0 62		, 8-71 - 8-72
TANH	8.63	X>0?	8-70
TIME	0-03	X <y?< td=""><td>8-69</td></y?<>	8-69
TONE	8-03 8-64	X<=0?	8-68
IONE	0-04	$X \le Y?$	8-68
		XOR	8-73
U		ΧΤΟΑ	8-73
Up	2-5	V	
USER	2-5 - 2-6	•	
<b>X</b> 7		Y∱X	8-73
V			
VIEW	8-64		

(This Page is intentionally left blank)

## BUSINESS AND SCIENTIFIC CALCULATOR SOFTWARE TOOLS FOR YOUR PC

Eclipse brings the power and flexibility of the HP41 series calculator to the PC World. A full range of products provides you with complete functionality for computational and instrumentational control on your PC. These products use the full screen graphics capability of your PC display screen thus providing an extremely user friendly environment.

Eclipse Logic, Inc. Software tools include: HP-41/41SX Emulator/ Scientific Calculator Software... Program and Data Upload/Download... HP-IL Device Emulator Software... HP-41 to ELI-41 Program Upload... ELI-41 with IEEE-488 for GPIB-PC Data Acquisition and Analysis...2D & 3D Graphics...Control for RS-232 Instruments and Modems...HP-IL Instrumentation control ...IEEE-488 Instrumentation Control.

Call our toll free number (800) 842-6888, extension 100 for more information today!

