Terminal Remote Control

3. TERMINAL REMOTE CONTROL

This section explains how to operate the 212 Multi Programmer from a remote terminal. Most of the operations that can be performed from the front panel are available in Terminal Remote Control (TRC) mode, as well as sumchecking commands, and commands for programming with data in 16-bit-wide word format and 32-bit-wide word format.

The commands for operation from a remote terminal are grouped into the following subsections:

- ☐ INTRODUCTION—This section includes a list of the symbols and conventions used in this section, a quick reference summary of all of the commands available in Terminal Remote Control, and some general operating notes. Read the general operating notes before attempting any of the Terminal Remote Control Commands.

- ☐ ENTERING AND EXITING TERMINAL REMOTE CONTROL—This subsection explains how to transfer control of the programmer to a remote terminal and how to return control of the programmer to the front panel.

- ☐ ON-LINE HELP—This subsection explains how to display the on-line help screens for EPROMs and Microprocessors and how to interpret the help screen symbols.

- ☐ SELECTING A DEVICE TYPE—This subsection explains how to select the device type using the 212 TRC menus or by entering the family/pinout code directly. The correct device type must be selected prior to performing programming operations, otherwise damage to the installed devices could result. This section also explains how to enable electronic ID checking. This feature causes the 212 to read the electronic ID of the installed device and to verify that the family/pinout code is correct.

- ☐ PROGRAMMING OPERATIONS—This subsection explains how to perform a blank check on a device prior to programming it, load the master data into RAM from a previously programmed device, and program a blank device with the master data.

- VERIFYING A PROGRAMMED DEVICE—This subsection explains how to verify that the data programmed into the installed devices matches the data in the programmer's RAM. Devices are automatically verified during the programming operation, but this command allows you to verify the device at another time.

- EDITING MEMORY—This subsection explains how to edit the contents of the programmer's data memory, or RAM.

- SUMCHECKING DATA—This subsection explains how to perform a sumcheck (hexadecimal summation) and a checksum (exclusive-OR summation) on the data stored in the programmer's RAM.

Symbols and Conventions

The following is a list of symbols and conventions used in the Terminal Remote Control section of this document:

<CR>	This symbol represents the carriage return key.
<ESC>	This symbol represents the escape key.
<CTRL-S>	CTRL- followed by a letter means that you should hold down the CONTROL (or CTRL) key and press the letter once. For example, to type <CTRL-S>, you would hold down the CONTROL key and press the S key once.
H	Any value followed by an upper-case H means that the value is in hexadecimal form.
<lower-case>	Lower-case letters enclosed in angle brackets describe the kind of parameter that you should fill in for the command. Replace the words and the brackets with the appropriate parameter value for the operation you are performing. For example, the Memory Modify command format is M <start address>. You would enter M 6 to modify memory starting at hexadecimal address 6.
[<lower-case>]	Words or characters enclosed in square braces represent optional entries.
Ø	Zeroes in this format are used to avoid confusion where letters and numbers are interspersed.
word-wide	Refers to data formatted in 16-bit-wide words.
long-word-wide	Refers to data formatted in 32-bit-wide words.

TERMINAL REMOTE CONTROL COMMAND SUMMARY

Command Format	Description	Command Format	Description
M	Enter TRC mode	P [a1 a2 a3]	Program single device
R	Exit TRC mode (Return)	PS [set no.]	Program Set of devices
<CR>	Execute command	PW [set no.]	Program Word-wide set
<ESC>	Abort command	PL [set no.]	Program Long-word-wide set
<BS>	Backspace (key)		
<CTRL-S>	Suspend display	V [Vcc] [a1 a2 a3]	Verify single device
<CTRL-Q>	Continue display	VS [Vcc] [set no.]	Verify Set of devices
		VW [Vcc] [set no.]	Verify Word-wide set
?	Display help screen	VL [Vcc] [set no.]	Verify Long-word-wide set
A?	Select device type	M a1 a2	Display Memory
ID on/off	Select electronic ID checking	M a1	Modify Memory
S ff/pp	Select family/pinout code	M a1 a2 d1 [d2...d8]	Fill Memory
		I a1 d1 [d2...d8]	Insert data
B	Blank check installed devices	D a1 [a2]	Delete data
		T a1 a2 a3	Transfer (copy) data
L [a1 a2 a3]	Load single master	MS a1 a2 d1 [d2...d8]	Search Memory
LS [set no.]	Load Set of masters	BS a1 a2	Byte Swap
LW [set no.]	Load Word-wide masters	CT a1 a2 [a3]	Sumcheck (Total)
LL [set no.]	Load Long-word-wide masters	CX a1 a2 [a3]	Checksum (EXOR)

List of Symbols

[ ] indicates optional entry
a1 = starting RAM address
a2 = ending RAM address
a3 = starting device or destination address
d1 d2...d8 = data bytes 1 through 8
on/off = on/off selection (1=on, 0=off)
ff/pp = four digit family/pinout code
set no. = set size
Vcc = Vcc selection (H or L)

GENERAL OPERATING NOTES

This subsection contains important information on how to type in the Terminal Remote Control (TRC) commands, as well as information on how to use special keys to control command execution and how to read the device error indicators. A list of the default parameter values for the TRC commands is given at the end of this subsection.

Entering Commands

Below are some general rules for entering Terminal Remote Control commands.

- ☐ Each command must be entered at the command prompt (>) and must be followed by a <CR>.
- ☐ All commands must be typed in upper-case letters.
- ☐ In most cases, the first parameter (such as a memory address) that follows the command letter(s) can either be separated from the command letter(s) by a space or it can be typed immediately following the command letter(s) with no space. For example, the Load Set command could be typed LS4 or LS 4 to load a set of four masters. The one exception is the command to verify a single device (or a set of devices), which requires a space between the command letter (V), and the Vcc selection (L or H) in order to distinguish it from the Verify Long Word command (VL).
- ☐ Each parameter following the first parameter must be separated from the preceding parameter by a space. For example, to display memory addresses Ø through F, the Display Memory command must be typed M Ø F or MØ F.
- ☐ You do not need to enter leading zeros when entering command parameters, such as memory addresses. For example, 0010 is the same as 10.
- ☐ All command parameters are entered as hexadecimal numbers, unless otherwise specified.

Controlling Command Execution

The following keys are special keys which can be used to control the execution of TRC commands.

BACKSPACE Key
While you are entering TRC commands, you can use the <BS> key (or <Delete>, <Rubout>, or <CTRL-H>, if your terminal keyboard has no <BS> key) to delete the previously entered character. You can use this key repeatedly until you have deleted the entire command line.

ESCAPE Key
You can abort the command currently being executed or the command you are currently typing by pressing the <ESC> key. If you press the <ESC> key while typing in a command (before pressing <CR>), the programmer will ignore the line you were typing, display the command prompt (>) on a new line, and await your input. If you press the <ESC> key while a command is being executed, the programmer will abort the command, display the command prompt (>), and await your input.

<CTRL-S>
Typing a <CTRL-S> will suspend the displaying of data on the terminal screen. The command that was being executed when the <CTRL-S> was typed is temporarily halted, but not aborted. To continue the display, type <CTRL-Q>.

<CTRL-Q>
Typing a <CTRL-Q> continues the displaying of data on the terminal screen exactly where it was suspended by the <CTRL-S> command.

Action Display

While an operation is being executed in TRC mode, advancing dots will appear on the bottom line of the programmer display, indicating that the programmer is performing the operation.

Error Indicators

The lamp (LED) above the socket is used to indicate successful or unsuccessful programming operations. If an operation is completed successfully on a device, a "pass" signal will sound (two beeps) and the LED above the socket will light green. If an operation is not completed successfully on a device, a "fail" signal will sound (three beeps) and the LED above the socket will light red.

A "SYNTAX ERROR" will occur when the format of a command is incorrect or when illegal characters are entered in a command. A caret (^) will appear below the first unrecognized or illegal character encountered by the programmer.

Default Parameter Values

Many of the Terminal Remote Control commands can be accompanied by optional parameters, such as starting and ending memory addresses. The power-up default values of these optional parameters are listed below. When a new set size is entered or a new device type is selected, the new value becomes the default for future operations. The default values of the <start address>, <end address>, <device address>, and <Vcc> will remain the same as listed below even if a new value has been specified in a previous command.

<start address>	ØØØØH
<end address>	3FFFFH (256k) or the size of the selected device for device operations
<device address>	ØØØØH
<set size>	1
<Vcc> for verify operations	device manufacturer's recommendation
Device type	reads Electronic ID of installed device (initial family/pinout code is FF/FF)

ENTERING AND EXITING TERMINAL REMOTE CONTROL

The following is an explanation of how to transfer control of the programmer to the remote terminal (enter TRC mode) and how to return control of the programmer to the front panel (exit TRC mode).

Entering Terminal Remote Control

Procedure	Example 212 Displays
1. Connect the programmer to the terminal as described in the RS-232C Port Cable Connections discussion in the Getting Started section of this manual.	
2. Power up the programmer by plugging the power cord into the back of the programmer and into a power outlet and pressing the power switch on the back of the programmer to the ON position.	SELF TESTING . . . . .
3. When the self test is complete and the display reads LOAD FROM MASTER, scroll through the main menu to RS232 PORT and press ENTER.	RS232 PORT COMPUTER CONTROL
4. Scroll through the RS232 PORT menu to TERMINAL CONTROL and press ENTER.	TERMINAL CONTROL

Procedure	Example 212 Displays
5. To establish communications between the programmer and the terminal and automatically select the proper communications protocol, type M<*CR*> on the terminal keyboard. (The "M" will not be displayed on the terminal screen.) You can also set the communications protocol manually, by executing the PORT SETTING function (see the Front Panel Operation section). If the correct port settings are selected prior to entering TRC mode, you can press any terminal key to establish communications. When communication is established, a command prompt (>) will appear on the terminal screen. The programmer will display TERMINAL CONTROL until control of the programmer is returned to the front panel.	TERMINAL CONTROL

Exiting Terminal Remote Control

You can exit Terminal Remote Control mode by using either the terminal keys or the programmer keys. To exit TRC using the terminal keys, type

R<*CR*>

(The R command stands for Return to local.) To exit TRC mode using the programmer keys, press any key.

The programmer will display

```
RS232 PORT
```

You can now continue operating the programmer using the front panel keys.

ON-LINE HELP -

EPROMs Only

Help screens displaying the currently selected device type and a list of all of the TRC commands and their formats can be displayed on the terminal screen by typing

?<*CR*>

at the command prompt (>). The help screen shown on your terminal will be for either EPROM devices or microprocessor devices depending on the module connected to the base unit. The help screen for **EPROM** devices is shown below:

```
                  ***  212 MULTI PROGRAMMER COMMAND HELP  ***

SELECT DEVICE TYPE  A?                  \  SUMCHECK (TOTAL)   CT a1 a2 [ a3]
BLANK CHECK         B                   \  CHECKSUM (EXOR)    CX a1 a2 [ a3]
LOAD SINGLE         L  [a1 a2 a3]       \  MEMORY DISPLAY     M  a1 a2
LOAD SET            LS [ # ]            \  MEMORY MODIFY      M  a1
LOAD WORD           LW [ # ]            \  MEMORY FILL        M  a1 a2 d1[..d8]
LOAD LONG WORD      LL [ # ]            \  INSERT             I  a1 d1[..d8]
PROGRAM SINGLE      P  [a1 a2 a3]       \  DELETE             D  a1[ a2]
PROGRAM SET         PS [ # ]            \  TRANSFER           T  a1 a2 a3
PROGRAM WORD        PW [ # ]            \  MEMORY SEARCH      MS a1 a2 d1[..d8]
PROGRAM LONG WORD   PL [ # ]            \  ENABLE ID          ID <1/0>
VERIFY SINGLE       V  [H/L][a1 a2 a3]  \  SET F/P CODE       S  code
VERIFY SET          VS [H/L][ # ]       \  BYTE SWAP          BS a1 a2
VERIFY WORD         VW [H/L][ # ]       \  RETURN TO LOCAL    R
VERIFY LONG WORD    VL [H/L][ # ]       \  HELP               ?

DEVICE TYPE: *                  F/P: FF/FF       SETSIZE #: 01
```

List of Symbols

[ ] indicates optional entry
a1 = starting RAM address
a2 = ending RAM address
a3 = starting device or destination address
d1...d8 = data bytes 1 through 8
# = set size
H/L = Vcc selection (H or L)
on/off = on/off selection (1=on, 0=off)
code =4-6 digit family/pinout code

ON-LINE HELP -
Microprocessors Only

The help screen for **Microprocessor** devices is shown below:

```
                *** 212 MULTI PROGRAMMER COMMAND HELP ***

SELECT DEVICE TYPE  A?                  \  SUMCHECK (TOTAL)    CT a1 a2[ a3]
BLANK CHECK         B                   \  CHECKSUM (EXOR)     CX a1 a2[ a3]
LOAD SINGLE         L  [a1 a2 a3]       \  MEMORY DISPLAY      M  a1 a2
PROGRAM SINGLE      P  [a1 a2 a3]       \  MEMORY MODIFY       M  a1
VERIFY SINGLE       V  [a1 a2 a3]       \  MEMORY FILL         M  a1 a2 d1 [..d8]
RETURN TO LOCAL     R                   \  INSERT              I a1 d1[..d8]
SET F/P CODE        S  [code]           \  DELETE              d a1 [a2]
SECURITY OPTION     SO                  \  TRANSFER            T a1 a2 a3
ENCRYPTION DISPLAY  E  a1 a2            \  MEMORY SEARCH       MS a1 a2 d1[..d8]
ENCRYPTION MODIFY   E  a1               \  BYTE SWAP           BS a1 a2
ENCRYPTION FILL     E  a1 a2 d1[..d8]   \  ENABLE ID           ID (1/0)
USER SIGNATURE      U                   \  ENABLE DEVICE TEST  DT (1/0)
HELP                ?

DEVICE TYPE: *                  F/P: FF/FF
```

List of Symbols

[ ] indicates optional entry
a1 = starting RAM address
a2 = ending RAM address
a3 = starting device or destination address
d1...d8 = data bytes 1 through 8
# = set size
H/L = Vcc selection (H or L)
on/off = on/off selection (1=on, 0=off)
code = 4-6 digit family/pinout code

SELECTING A DEVICE TYPE

Use the commands described in the following paragraphs to select the type of the device you are loading from, blank checking, programming or verifying. The device type can be selected in three ways — by selecting the correct manufacturer and part number from the TRC menus, by entering the Data I/O family/pinout code directly, or by using the electronic ID on devices that contain the electronic ID option. Selecting the device type tells the programmer which algorithm and voltages to use to program the part. ***You must select the correct device type before blank checking, loading, programming, or verifying a device, otherwise damage to the device could result.*** To make use of the electronic ID feature, for example, when a family/pinout code of ØFF/ØFF is selected, the 212 will read the electronic ID of the installed device and automatically select the correct family/pinout code for the device. You may also enable the Electronic ID checking feature (explained in this subsection), which causes the 212 to display an error message if the selected device type and the electronic ID contained in the installed device do not match. The current device type and family/pinout code selected can be viewed by typing `?<CR>`.

Selecting the Device Type from the TRC Menus

Use the **Select Device Type** command to select the manufacturer and part number of the device you are loading from, blank checking, programming, or verifying from the TRC menus. You do not need to know the device's assigned family/pinout code to select the device type using this command. After you have selected the device's manufacturer and part number using this command, the 212 Multi Programmer will automatically supply the proper family and pinout codes for the device.

Command Format: A?

Procedure	Example Key Sequence
1. Type in the Select Device Type command letters, A?, and press carriage return.	A?<CR>
2. After the list of device manufacturers is displayed on the screen, type in the number that corresponds to the manufacturer of the device you are blank checking, loading, programming or verifying and press carriage return. If you press carriage return without entering a number, device manufacturer 0 will be selected.	8<CR>
3. After the list of device part numbers is displayed on the screen, type in the number that corresponds to the correct device part number and press carriage return. If you press carriage return without entering a number, part number 0 will be selected. The device type selected will remain the default device type until a new device type is selected.	7<CR>

Example

Select Intel device 2764:

```
>A?

DEVICE MANUFACTURERS

0.AMD                8.INTEL          16.OKI           24.TEXAS INSTRUMENT
1.ATMEL              9.ICT            17.RICOH         25.TOSHIBA
2.CYPRESS           10.MATSUSHITA     18.ROCKWELL      26.TRISTAR SEMICON
3.EUROTECHNIQUE     11.MITSUBISHI     19.SAMSUNG       27.VTI
4.EXEL              12.MOSTEK         20.SEEQ          28.XICOR
5.FUJITSU           13.MOTOROLA       21.SGS-ATES
6.GENERAL INSTRU.   14.NATIONAL       22.SIGNETICS
7.HITACHI           15.NEC            23.SMOS SYES

ENTER MANUFACTURER NUMBER AND RETURN 8<CR>

INTEL DEVICES

0. 2716          5. 2732B         10. P2764         15. 27256
1. 2816          6. P2732A        11. P2764A        16. 27C256
2. 2816A         7. 2764          12. 27128         17. 27512
3. 2732          8. 2764A         13. 27128A        18. 27513
4. 2732A         9. 27C64         14. P27128A

ENTER PART NUMBER AND RETURN 7<CR>

FAMILY AND PINOUT CODE IS 79/33
```

NOTE

The above list is only an example. The number of devices available at the time you purchase your programmer may be different.

Selecting the Family/Pinout Code

Use the **Set Family/Pinout Code** command to select the correct device type by entering the Data I/O family/pinout code. The correct family/pinout code for each device programmable by the 212 is printed in the device list shipped with your programmer or module. Perform the following procedure to select the family/pinout code.

Command Format: S <ffpp>

Procedure	Example Key Sequence
1. Type in the Set Family/Pinout Code command letter, S, followed by a space.	S
2. Type in the six-digit family/pinout code printed in the device list for the device you will be loading, programming, or verifying and then press carriage return.	S 079033<CR> or S<CR> 079033<CR>

NOTE

If you type an "S" followed by a carriage return (without entering a family/pinout code), the programmer will prompt you to enter the family/pinout code.

After you enter the family/pinout code, the programmer will return a message indicating whether the family/pinout code combination you entered was a legal code or was an illegal combination. If you receive a failure message, check your device list for the correct code and try entering it again.

Enabling/Disabling Electronic ID Checking

Use the **Electronic ID Checking** feature to cause the programmer to read the electronic ID contained in the installed device and make sure that the device type specified by the electronic ID matches the device type you selected. If this feature is enabled, the programmer will read the electronic ID of the installed device at the beginning of a blank check, load, program, or verify operation. If the device type you selected does not match the installed device electronic ID type, the programmer will not allow the operation to be performed. Electronic ID checking is enabled upon power up.

Procedure	Example Key Sequence
1. Type in the Electronic ID Checking command letters, ID, followed by a space.	`ID`

2. If you want to enable the Electronic ID Checking feature, type a 1 and press carriage return. If you want to disable the Electronic ID Checking feature, type a 0 (zero) and press carriage return.
 	
The programmer will display a message acknowledging that the Electronic ID Checking feature has either been enabled or disabled, depending upon whether you entered a 0 or a 1.
	`ID 1<CR>`
	`or`
	`ID 0<CR>`

PROGRAMMING OPERATIONS

The following commands allow you to perform programming operations from the remote terminal. With these commands you can check to make sure a device is blank, set security options, load data into RAM from a master device, and program blank devices with the master data. The commands are presented in the following order:

Command Name	Command Format
Blank Check	B
Set Security Options	SO (microprocessors only)
Load Single	L [<start address> <end address> <device address>]
Program Single	P [<start address> <end address> <device address>]
Load Set	LS [<set size>]
Program Set	PS [<set size>]
Load Word	LW [<set size>]
Program Word	PW [<set size>]
Load Long Word	LL [<set size>]
Program Long Word	PL [<set size>]

Checking for Non-blank Devices

The **Blank Check** command allows you to check a device prior to programming it to make sure that it is blank.

Command Format: B

Procedure	Example Key Sequence
1. Make sure that the correct device type is selected.	
2. Insert the device to be checked in the socket and lock it into place by pushing the socket lever down.	

3. Type in the Blank Check command letter, B, and press carriage return.
 	
If the blank check is successful, the terminal will display
 	
`CHECK BLANK . . . . . .`
`BLANK CHECK OK`
 	
and the socket LED will light green.
 	
If the blank check is unsuccessful, the terminal will display an error message below the `CHECK BLANK . . . . . .` message and the socket lamp will light red. See the Error Messages section for an explanation of the error message.
	`B<CR>`

4. Lift the socket lever and remove the device.	

Setting Security Options -
Microprocessors Only

Use the **Security Options** command to program the security bit of devices that support security data protection; the security bit is programmed so that the data contained in the device cannot be read by a programmer. Security options must be set before a device is programmed; if using Electronic ID, the device must be inserted in the programmer before selecting the security options. The available settings are selected by typing the number associated with the option as explained below:

Family Code D5 and 54
1. Array Only
2. Array and Security
3. Security Only

Family 5A
1. Array Only
2. Array and Lock Bit 1 & Lock Bit 2
3. Lock Bit 1 and Lock Bit 2
4. Encryption Table Only (32 Bytes)
5. Lock Bit 1 Only
6. Array and Encryption Table
7. Array and Lock Bit 1
8. Array and Encryption Table & Lock Bit 1

Family 51
1. Array and User Signature
2. Array and User Signature and Security Bit
3. Security Bit Only

Command Format: SO

Procedure	Example Key Sequence
1. Make sure that the correct device type and family/pinout code is selected. Programming a device with the wrong family and pinout codes selected could permanently damage the device. If you are using Electronic ID, the device must be inserted in the programmer socket before selecting the security option.	
2. Type in the Security Options command letter, SO.	SO<CR>
3. After the list of Security Options is displayed on the screen, type the number that corresponds to the option you wish to select and press carriage return. If you press carriage return without entering a number, security option #1 will be selected.	2<CR>
4. Insert the master device into the socket and lock it into place by pushing the socket lever down.	
5. Proceed with one of the programming operations as described on the following pages.	

Loading a Single Master

Use the **Load Single** command to load (or copy) data from a single master device into the programmer's data RAM for subsequent programming of blank devices, uploading or editing. You can load data starting at programmer RAM address 0, or you can specify a different block of programmer RAM to be loaded with data. You can also select the device address that the first byte of data will be loaded from. Loading the master data into RAM is the first step in programming devices. The second step is to copy the data from RAM to a blank device using the Program Single command (which is explained following this subsection).

Command Format: L [<start address> <end address> <destination address>]

Procedure	Example Key Sequence
1. Make sure that the correct device type is selected.	
2. Insert the master device into the socket and lock it into place by pushing the socket lever down.	
3. Type in the Load Single command letter, L.	L
4. Type in the first address of RAM that you want data written to, or, to load RAM starting with the default device address (Ø) and the default programmer RAM address (Ø), go to step 7. (The space after "L" is optional.)	L Ø

Procedure	Example Key Sequence
5. Type a space and then type in the last address of RAM that you want data written to. If you entered a start address, you must enter an end address.	L Ø 1FFF
6. Type a space and then type in the device address that you want the first byte of data to be loaded from.	L Ø 1FFF
7. Press carriage return to execute the command. 0<*CR*> When the load operation is complete, the terminal will display LOADING..... THE SUMCHECK (TOTAL) IS: HHHHHH where "HHHHHH" is the sumcheck of the data that was just loaded into programmer memory, and the socket lamp will light green.	L Ø 1FFF or L<CR>
8. Lift the socket lever and remove the master device.	

Programming Single Devices

Use the **Program Single** command to program a device with the master data loaded into the programmer's RAM with a Load Single command or a download operation. You can program the device with programmer RAM starting with the first byte of programmer RAM, or you can specify another range of programmer RAM to be programmed into the device. You can also select the device address that the first data byte will be written to.

Command Format: P [<start address> <end address> <destination address>]

Procedure	Example Key Sequence
1. Make sure that the correct device type is selected. Programming a device with the wrong family and pinout codes selected could permanently damage the device.	
2. Insert the blank device in the socket and lock the device into place by pressing the socket lever down.	
3. Type in the Program Single command letter, P.	P
4. Type in the first address of RAM that you want programmed into each device, or, to program each device starting with the default programmer RAM address (Ø) and the default device address (0), go to step 7. (The space after "P" is optional.)	P Ø

Procedure	Example Key Sequence
5. Type a space and then type in the last address of RAM that you want programmed into each device. If you entered a start address, you must enter an end address.	P Ø 1FFF
6. Type a space and then type in the address of the device memory location that you want the first byte of data to be written to (destination address).	P Ø 1FFF Ø
7. Press carriage return to execute the command. The terminal will display or PROGRAMMING If the programming operation is successful, the terminal will display PROGRAMMING..... THE SUMCHECK (TOTAL) IS: HHHHHH where "HHHHHH" is the hexadecimal sumcheck of the programmed device. The sumcheck should match the sumcheck of the master data loaded into the programmer's RAM.	P Ø 1FFF Ø<CR> P<CR>

Procedure	Example Key Sequence
If the programming operation is unsuccessful, the terminal will display an error message below the PROGRAMMING. . . . . message and the socket lamp will light red. See the Error Messages section for an explanation of the error message.	

8. Lift the socket lever and remove the device.

Loading a Set of Masters

Use the **Load Set** command to load data from a set of master devices into programmer RAM for subsequent programming into a set of blank devices, uploading, or editing. The data of each successive device is loaded into a successive block of programmer RAM. See the table following the procedure steps for the block of RAM assigned to each device loaded. After loading the set of master devices, you can use the Program Set command (described following this subsection) to program the data into a set of blank devices.

Command Format: LS [<number of devices>]

Procedure	Example Key Sequence
1. Make sure that the correct device type is selected.	
2. Type in the Load Set command letters, LS.	LS
3. Type in the number of devices in the set to be loaded and press carriage return, or, to load the default number of devices (1), press carriage return without entering a number of devices. (The space after "LS" is optional.) The terminal will display `INSERT DEVICE NO: 01`	LS 4<CR> or LS<CR>

Procedure	Example Key Sequence

4. Insert the correct device of the set into the socket and press carriage return to execute the load command for the device number displayed, or, to load a different device, type in the number of the device you want to load before pressing carriage return. This feature allows you to load data into any specified block of RAM. When the load operation for the installed device is complete, the terminal will display
 	
`LOADING......`
`INSERT DEVICE NO: NN`
 	
where "NN" is the number of the next device to be loaded. ("NN" is always the number of the device previously loaded plus 1.)	

5. Lift the socket lever and remove the device just loaded. To load another device, return to step 4. If all devices have been loaded, note the displayed sumcheck for later verification of sets of devices programmed with this data.	

Example

Load a set of four devices, but skip device number 2 and load device number 3 instead.

```
>LS 4<CR>
INSERT DEVICE NO: 01  <CR>
LOADING......
INSERT DEVICE NO: 02  3<CR>
LOADING......
INSERT DEVICE NO: 04  <CR>
LOADING......
THE SUMCHECK (TOTAL) IS: 001E6F
```

Programmer RAM blocks assigned to set devices being loaded (Assuming a 256k byte RAM size):

	Device Size (no. of bits)						
	16K	**32K**	**64K**	**128K**	**256K**	**512K**	**1M**
Device Number	**Device Size (no. of bytes) 2K**	**4K**	**8K**	**16K**	**32K**	**64K**	**128K**
1	0000 07FF	0000 0FFF	0000 1FFF	0000 3FFF	0000 7FFF	0000 FFFF	0000 1FFFF
2	0800 0FFF	1000 1FFF	2000 3FFF	4000 7FFF	8000 FFFF	10000 1FFFF	20000 3FFFF
3	1000 17FF	2000 2FFF	4000 5FFF	8000 BFFF	10000 17FFF	20000 2FFFF	
4	1800 1FFF	3000 3FFF	6000 7FFF	C000 FFFF	18000 1FFFF	30000 3FFFF	
	•	•	•	•	•		
	•	•	•	•	•		
	•	•	•	•	•		
8	3800 3FFF	7000 7FFF	E000 FFFF	1C000 1FFFF	38000 3FFFF		
	•	•	•	•			
	•	•	•	•			
	•	•	•	•			
16	7800 7FFF	F000 FFFF	1E000 1FFFF	3E000 3FFFF			
	•	•	•				
	•	•	•				
	•	•	•				
32	F800 FFFF	1F000 1FFFF	3E000 3FFFF				
	•	•					
	•	•					
	•	•					
64	1F800 1FFFF	3F000 3FFFF					

Programming Sets of Devices

Use the **Program Set** command to program a set of devices with the contents of the programmer's RAM. This command fills each device with successive blocks of programmer RAM, starting with RAM address Ø and device address Ø. See the following table to determine which block of RAM will be programmed into which device. You can use this command to produce sets of devices identical to the masters or you can use this command to merge the data into larger devices. To merge the data into larger devices, select the device type of the larger blank devices before executing the Program Set command.

Command Format: PS [<number of devices>]

Procedure	Example Key Sequence
1. Make sure that the correct device type is selected. Programming a device with the wrong family and pinout codes selected could permanently damage the device.	
2. Type in the Program Set command letters, PS.	PS
3. Type in the number of devices to be programmed and press carriage return, or, to program the default number of devices, press carriage return without entering a number of devices. (The space after "PS" optional.) The terminal will display	PS 4<CR> or PS<CR>

Procedure	Example Key Sequence

4. Insert a blank device into the socket and press carriage return to program the device with the block of RAM corresponding to the device number displayed, or, to program the device with a different block of RAM, type in the device number corresponding to the block of RAM that you want to program into the blank device before pressing carriage return. This feature allows you to program a device with any specified block of RAM. (See the table following the example to determine which block of RAM will be programmed into the device.) The terminal will display

   ```
   PROGRAMMING
   ```

   If the program operation is successful for the installed device, the terminal will display

   ```
   PROGRAMMING......
   INSERT DEVICE NO: NN
   ```

   where "NN" is the number of the next device to be programmed. ("NN" is always the number of the device previously programmed plus 1.) If the last device of the set was just programmed the terminal will display

   ```
   THE SUMCHECK (TOTAL) IS: HHHHHH
   ```

   The sumcheck (HHHHHH) should match the sumcheck of the set data loaded into the programmer's RAM.

Procedure	Example Key Sequence

If the program operation is unsuccessful, the terminal will display an error message below the PROGRAMMING..... message and the socket lamp will light red. See the Error Messages section for an explanation of the error message.

5. Lift the socket lever and remove the programmed device. To program another device, return to step 4.

Example

Program a set of two devices.

```
>PS<CR>
INSERT DEVICE NO: 01  <CR>
PROGRAMMING......
INSERT DEVICE NO: 02  <CR>
PROGRAMMING......
THE SUMCHECK (TOTAL) IS: 0096D4


>
```

Programmer RAM blocks programmed into set devices (Assuming a 256K byte RAM size):

	Device Size (no. of bits)						
	16K	**32K**	**64K**	**128K**	**256K**	**512K**	**1M**
Device Number	**Device Size (no. of bytes)**						
	2K	**4K**	**8K**	**16K**	**32K**	**64K**	**128K**
1	0000 07FF	0000 0FFF	0000 1FFF	0000 3FFF	0000 7FFF	0000 FFFF	0000 1FFFF
2	0800 0FFF	1000 1FFF	2000 3FFF	4000 7FFF	8000 FFFF	10000 1FFFF	20000 3FFFF
3	1000 17FF	2000 2FFF	4000 5FFF	8000 BFFF	10000 17FFF	20000 2FFFF	
4	1800 1FFF • • •	3000 3FFF • • •	6000 7FFF • • •	C000 FFFF • • •	18000 1FFFF • • •	30000 3FFFF	
8	3800 3FFF • • •	7000 7FFF • • •	E000 FFFF • • •	1C000 1FFFF • • •	38000 3FFFF		
16	7800 7FFF • • •	F000 FFFF • • •	1E000 1FFFF • • •	3E000 3FFFF			
32	F800 FFFF • • •	1F000 1FFFF • • •	3E000 3FFFF				
64	1F800 1FFFF	3F000 3FFFF					

Loading Word-Wide Masters

Use the **Load Word** command to load pairs of 8-bit-wide devices in 16-bit-wide word format. This command loads the data from the first device into even-numbered addresses, starting with address Ø, and loads the data from the second device into odd-numbered addresses (see illustration). You can load more than one pair of 8-bit-wide devices (with the data for each pair being loaded into the next available RAM block), as long as the total size of all the devices does not exceed 512K bits. See the table following the procedure for the maximum number of pairs that can be loaded with one Load Word command.

Command Format: LW [<number of devices>]

Procedure	Example Key Sequence
1. Make sure that the correct device type is selected.	
2. Type in the Load Word command letters, LW.	`LW`

3. Type in the number of devices to be loaded (which must be a multiple of 2) and press carriage return, or, to load the default number of devices, press carriage return without entering a number of devices. (The space after "LW" is optional.) The terminal will display
 	
`INSERT DEVICE NO: 01`
	`LW 4<CR>`
	or
	`LW<CR>`

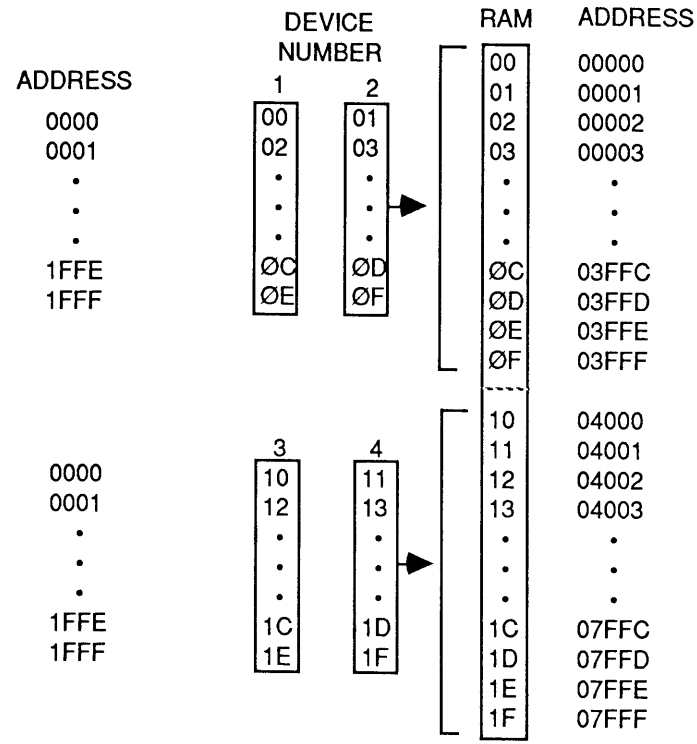
Procedure	Example Key Sequence

4. Insert the correct device of the pair into the socket (the device containing the low-order bytes for the pair should be loaded first and then the device for the same pair containing the high-order bytes should be loaded). Press carriage return to execute the load command for the device number displayed, or, to load a different device, type in the number of the device you want to load before pressing carriage return. This feature allows you to load any device of any pair. When the load operation for the installed device is complete, the terminal will display
 	
`LOADING • • • • • •`
`INSERT DEVICE NO: NN`
 	
where "NN" is the number of the next device to be loaded. ("NN" is always the number of the device previously loaded plus 1.)	

5. Lift the socket lever and remove the device just loaded. To load another device, return to step 4. If all devices have been loaded, note the displayed sumcheck for verification of sets of devices programmed with this data.	

Maximum number of pairs that can be loaded with one Load Word Command:

Device Size in Bits	Maximum No. Pairs	
	RAM size in Bytes 1 Meg	256K
16K	256	64
32K	128	32
64K	64	16
128K	32	8
256K	16	4
512K	8	2
1M	4	1



Example

Load two pairs of 64K devices (four devices total).

```
>LW 4<CR>
INSERT DEVICE NO: 01  <CR>
LOADING • • • • • •
INSERT DEVICE NO: 02  <CR>
LOADING • • • • • •
INSERT DEVICE NO: 03  <CR>
LOADING • • • • • •
INSERT DEVICE NO: 04  <CR>
LOADING • • • • • •
THE SUMCHECK (TOTAL) IS: 00E8F4
>
```

Programming Devices Using a Word-Wide Format

Use the **Program Word** command to program 8-bit-wide devices with data loaded into the programmer's RAM in 16-bit-wide word format. The programmer programs the low bytes (even-addressed bytes) into the first device and the high bytes (odd-addressed bytes) into the second device. You can program as many unique pairs of word-wide devices as were loaded into RAM with the Load Word command (or down-loaded) and you can also program as many copies of each pair as you want.

Command Format: PW [<number of devices>]

Procedure	Example Key Sequence
1. Make sure that the correct device type is selected. Programming a device with the wrong family and pinout codes selected could permanently damage the device.	
2. Type in the Program Word command letters, PW.	PW
3. Type in the number of devices to be programmed (which must be a multiple of two) and press carriage return, or, to program the default number of devices, press carriage return without entering a number of devices. (The space after "PW" is optional.) The terminal will display INSERT DEVICE NO: 01	PW 4<CR> or PW<CR>

Procedure	Example Key Sequence

4. Insert a blank device into the socket and press carriage return to program the device with the data loaded into RAM that corresponds to the device number displayed; or, to program the device with data loaded into RAM from a different master device, type in the device number corresponding to the data that you want to program into the blank device before pressing carriage return. This feature allows you to program a device with data loaded into RAM from any specified master device. Odd-numbered devices will be programmed with the low-order bytes and even-numbered devices will be programmed with the high-order bytes for the word-wide pair (see the previous illustration). The terminal will display

   ```
   PROGRAMMING
   ```

   If the program operation is successful for the installed device, the terminal will display

   ```
   PROGRAMMING . . . . . .
   INSERT DEVICE NO: NN
   ```

   where "NN" is the number of the next device to be programmed. ("NN" is always the number of the device previously programmed plus 1.) If the last device of all pairs was just programmed the terminal will display

   ```
   THE SUMCHECK (TOTAL) IS: HHHHHH
   ```

   The sumcheck (HHHHHH) should match the sumcheck of all of the word-wide data loaded into the programmer's RAM.

Procedure	Example Key Sequence
If the program operation is unsuccessful, the terminal will display an error message below the PROGRAMMING • • • • • message and the socket lamp will light red. See the Error Messages section for an explanation of the error message.	

5. Lift the socket lever and remove the programmed device. To program another device, return to step 4.

Example

Program a pair of word-wide devices.

```
>PS<CR>
INSERT DEVICE NO: 01  <CR>
PROGRAMMING • • • • • •
INSERT DEVICE NO: 02  <CR>
PROGRAMMING • • • • • •
THE SUMCHECK (TOTAL) IS: 0096D4


>
```

Loading Long-Word-Wide Masters

Use the **Load Long Word** command to load a set of four 8-bit-wide devices in 32-bit-wide word format. See the illustration following the procedure steps for the order that device data is loaded into RAM. You can load more than one long-word-wide set of 8-bit-wide devices, as long as the total size of all of the devices does not exceed RAM size. See the table following the procedure for the maximum number of pairs that can be loaded with one Load Long Word command.

Command Format: LL [<number of devices>]

Procedure	Example Key Sequence
1. Make sure that the correct device type is selected.	
2. Type in the Load Long Word command letters, LL.	LL<*CR*>
3. Type in the number of devices to be loaded (which must be a multiple of 4) and press carriage return, or, to load the default number of devices, press carriage return without entering a number of devices. (The space after "LL" is optional.) The terminal will display `INSERT DEVICE NO: 01`	LL 8<*CR*> or LL<*CR*>

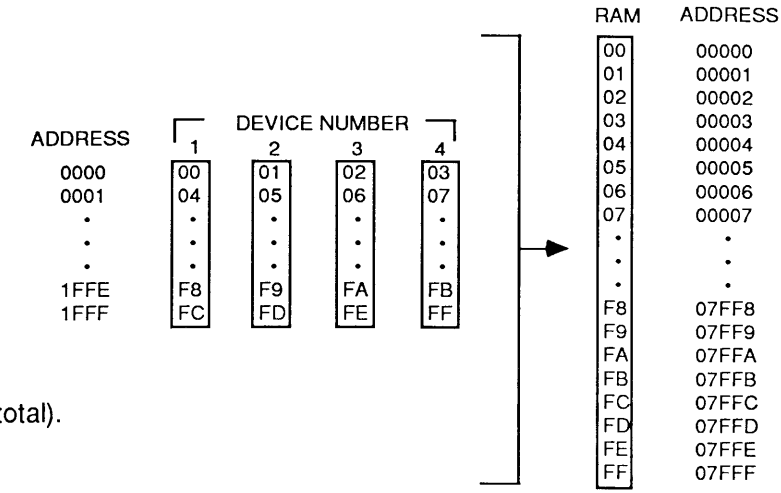
Procedure	Example Key Sequence

4. Insert the correct device into the socket (the devices of each set should be loaded in order, starting with the device containing the lowest-order bytes and ending with the device containing the highest-order bytes). Press carriage return to execute the load command for the device number displayed, or, to load a different device, type in the number of the device you want to load before pressing carriage return. This feature allows you to load any device of any set. When the load operation for the installed device is complete, the terminal will display
 	
`LOADING . . . . . .`
`INSERT DEVICE NO: NN`
 	
where "NN" is the number of the next device to be loaded. ("NN" is always the number of the device previously loaded plus 1.)	

5. Lift the socket lever and remove the device just loaded. To load another device, return to step 4. If all devices have been loaded, note the displayed sumcheck for verification of sets of devices programmed with this data.	

Maximum number of four-device sets that can be loaded with one Load Long Word command:

Device Size In Bits	Maximum No. of Sets	
	RAM Size in Bytes	
	1M	256K
16K	128	32
32K	64	16
64K	32	8
128K	16	4
256K	8	2
512K	4	1



Example

Load one set of 64K devices (four devices total).

```
>LL 4<CR>
INSERT DEVICE NO: 01  <CR>
LOADING • • • • • •
INSERT DEVICE NO: 02  <CR>
LOADING • • • • • •
INSERT DEVICE NO: 03  <CR>
LOADING • • • • • •
INSERT DEVICE NO: 04  <CR>
LOADING • • • • • •
THE SUMCHECK (TOTAL) IS: 00E8C6
>
```

Programming Devices Using a Long-Word-Wide Format

Use the **Program Long Word** command to program 8-bit-wide devices with data loaded into the programmer's RAM in 32-bit-wide word format. This command treats four 8-bit-wide devices as a single 32-bit-wide device. The devices of a set are programmed in order; the first device is programmed with the lowest-order bytes of the set and the fourth device of the set is programmed with the highest-order bytes of the set (see the previous illustration). You can program as many unique sets of long-word-wide devices as were loaded into RAM with the Load Long Word command (or a download operation) and you can also program as many copies of each long-word-wide set as you want.

Command Format: PL [<number of devices>]

Procedure	Example Key Sequence
1. Make sure that the correct device type is selected. Programming a device with the wrong family and pinout codes selected could permanently damage the device.	
2. Type in the Program Long Word command letters, PL.	PL
3. Type in the number of devices to be programmed (which must be a multiple of 4) and press carriage return, or, to program the default number of devices, press carriage return without entering a number of devices. (The space after "PL" is optional.) The terminal will display `INSERT DEVICE NO: 01`	PL 8<CR> or PL<CR>

Procedure	Example Key Sequence

4. Insert a blank device into the socket and press carriage return to program the device with the data loaded into RAM that corresponds to the device number displayed; or, to program the device with data loaded into RAM from a different master device, type in the device number corresponding to the data that you want to program into the blank device before pressing carriage return. This feature allows you to program a device with data loaded into RAM from any specified master device. The first device for each set will be programmed with the lowest-order bytes for the set and subsequent devices will be programmed with higher-order bytes, with the fourth device for each set being programmed with the highest order bytes (see the previous illustration). The terminal will display

   ```
   PROGRAMMING
   ```

   If the program operation is successful for the installed device, the terminal will display

   ```
   PROGRAMMING • • • • • •
   INSERT DEVICE NO: NN
   ```

   where "NN" is the number of the next device to be programmed. ("NN" is always the number of the device previously programmed plus 1.) If the last device of all of the sets was just programmed the terminal will display

   ```
   THE SUMCHECK (TOTAL) IS: HHHHHH
   ```

Procedure	Example Key Sequence
The sumcheck (HHHHHH) should match the sumcheck of all of the long-word-wide set data loaded into the programmer's RAM. If the program operation is unsuccessful, the terminal will display an error message below the PROGRAMMING • • • • • • message and the socket lamp will light red. See the Error Messages section for an explanation of the error message.	
5. Lift the socket lever and remove the programmed device. To program another device, return to step 4. To exit the Program Long Word operation, press *<ESC>*. (The programmer will continue programming sets of devices until you press *<ESC>*.)	

Example

Program one set of long-word-wide devices.

```
>PL<CR>
INSERT DEVICE NO: 01  <CR>
PROGRAMMING • • • • • •
INSERT DEVICE NO: 02  <CR>
PROGRAMMING • • • • • •
INSERT DEVICE NO: 03  <CR>
PROGRAMMING • • • • • •
INSERT DEVICE NO: 04  <CR>
PROGRAMMING • • • • • •
THE SUMCHECK (TOTAL) IS: 00C6D8

INSERT DEVICE NO: 01  <ESC>
>
```

VERIFYING PROGRAMMED DEVICES

Use the following commands to verify device data against data loaded into the programmer's RAM. Each device is automatically verified after it is programmed; however, if you wish to verify devices at another time, you can use these commands. Before you use any of the following verify commands, you must load the programmer's RAM with the data which the devices are to be verified against using either a load or a download command. The verify commands are presented in the following order:

Command Name	Command Format
Verify Single	V [<Vcc Selection>] [<start address> <end address> <destination address>]
Verify Set	VS [<Vcc Selection>] [<number of devices>]
Verify Word	VW [<Vcc Selection>] [<number of devices>]
Verify Long Word	VL [<Vcc Selection>] [<number of devices>]

Verifying Single Devices

The **Verify Single** command allows you to verify a single device against the master data loaded into the programmer's memory. You can verify a device beginning with master data at programmer RAM address Ø, or you can select a different block of RAM to verify the device against. You can also select the address of the device memory location where the verify operation will begin.

Command Format: V [<Vcc Selection>] [<start address> <end address> <destination address>]

Procedure	Example Key Sequence
1. Load the master data into the programmer's RAM using either a Load Single operation or a download operation.	
2. Make sure that the correct device type is selected.	
3. Insert the device to be verified into the socket.	
4. Type in the Verify Single command letter, V.	V

Procedure	Example Key Sequence
5. Type a space and then type the letter (L or H) that corresponds to the proper Vcc high and low settings shown in the following table (in this case, the space following the command letter V is required), or, to verify the installed device with the voltage settings recommended by the manufacturer, do not type anything for this parameter.	V L
6. Type in the first address of RAM that you want the device to be verified against, or, to verify the entire device data, go to step 9.	V L Ø
7. Type a space and then type in the last address of RAM that you want the device to be verified against. If you entered a start address, you must enter an end address.	
8. Type a space and then type in the first address of device memory that you want to be verified against the selected block of RAM.	V L Ø 1FFF Ø

Selection	High Voltage	Low Voltage
L	5.2 V	4.8 V
H	5.5 V	4.5 V

Procedure	Example Key Sequence

9. Press carriage return to execute the verify command.
 	
If the operation is successful, the terminal will display
 	
`FUNCTION COMPLETED`
`THE SUMCHECK (TOTAL) IS: HHHHHH`
 	
where "HHHHHH" is the hexadecimal summation of the data contained in the verified device.
 	
If the operation is unsuccessful, the terminal will display an error message below the `VERIFYING.....` message and the socket lamp will light red. See the Error Messages section for an explanation of the error message.
	`VLØ 1FFF Ø<CR>`
	or
	`VL<CR>`

10. Remove the verified device. To verify another device of the same type against the same master data, return to step 3.	

Verifying Sets of Devices

The **Verify Set** command allows you to verify sets of devices (each containing different data) against the master set data stored in the programmer's RAM.

Command Format: VS [<Vcc Selection>] [<number of devices>]

Procedure	Example Key Sequence
1. Load the master set data into the programmer's RAM using either a Load Set command or a download operation.	
2. Make sure that the correct device type is selected.	
3. Type in the Verify Set command letters, VS.	VS
4. Type in the letter (L or H) that corresponds to the proper Vcc high and low settings shown in the following table, or, to verify the installed device with the voltage settings recommended by the manufacturer, do not type anything for this parameter. (The space after "VS" is optional.)	VS L

Selection	High Voltage	Low Voltage
L	5.2 V	4.8 V
H	5.5 V	4.5 V

Procedure	Example Key Sequence
5. Type a space, type in the number of devices to be verified and press carriage return, or, to verify the default number of devices, press carriage return without entering a number of devices. The terminal will display INSERT DEVICE NO: 01	VS L or VS L<*CR*>

6. Insert the correct device of the set into the socket (see the table in Programming Sets of Devices for information on which RAM address corresponds to which device) and press carriage return to execute the command for the device number displayed, or, to verify a different device, type in the number of the device before pressing carriage return. This feature allows you to verify a partial set.

   If the operation is successful, the terminal will display

   ```
   VERIFYING . . . . . .
   INSERT DEVICE NO: NN
   ```

   where "NN" is the number of the next device to be verified. ("NN" is always the number of the device previously verified plus 1.)

Procedure	Example Key Sequence
If the last device of the set was just verified, the terminal will display `THE SUMCHECK (TOTAL) IS: HHHHHH` where "HHHHHH" is the hexadecimal summation of the data contained in the set that was just verified. If the operation is unsuccessful, the terminal will display an error message below the `VERIFYING.....` message and the socket lamp will light red. See the Error Messages section for an explanation of the error message.	
7. Remove the verified device. To verify another device, return to step 6. To exit the Verify operation, press *<ESC>*. The programmer will continue to verify device sets until you press *<ESC>*.	

Verifying Word-Wide Devices

Use the **Verify Word** command to verify devices against data that is loaded into RAM in 16-bit-wide word format. This command compares two 8-bit-wide devices to 16-bit-wide data stored in the programmer's RAM. The data in odd-numbered devices is compared to low-order bytes (even-addressed bytes) stored in RAM, and the data in even-numbered devices is compared to high-order bytes (odd-addressed bytes) stored in RAM. More than one pair of devices can be compared, provided the master data for all pairs to be verified is loaded into the programmer's RAM. Also, multiple copies of each pair can be verified with one command.

Command Format: VW [<Vcc Selection>] [<number of devices>]

Procedure	Example Key Sequence
1. Load the master set data into the programmer's RAM using either a Load Word command or a download operation.	
2. Make sure that the correct device type is selected.	
3. Type in the Verify Word command letters, VW.	VW

Procedure	Example Key Sequence
4. Type in the letter (L or H) that corresponds to the proper Vcc high and low settings shown in the following table, or, to verify the installed device with the voltage settings recommended by the manufacturer, do not type anything for this parameter. (The space after "VW" is optional.)	VW L
5. Type a space, type in the number of devices to be verified (which must be a multiple of 2) and press carriage return, or, to verify the default number of devices, press carriage return without entering a number of devices. The terminal will display INSERT DEVICE NO: 01	VW L4<CR> or VW L<CR>
6. Insert the correct device of the set into the socket; odd-numbered devices will be verified against low-order bytes for the pair and even-numbered devices will be verified against high-order bytes for the pair. Press carriage return to execute the command for the device number displayed, or, to verify a different device, type in the number of the device before pressing carriage return. This feature allows you to verify any device of any pair against the loaded data.	

Selection	High Voltage	Low Voltage
L	5.2 V	4.8 V
H	5.5 V	4.5 V

Procedure	Example Key Sequence

If the operation is successful, the terminal will display

```
VERIFYING • • • • • •
INSERT DEVICE NO: NN
```

where "NN" is the number of the next device to be verified. ("NN" is always the number of the device previously verified plus 1.)

If the last device of all of the pairs was just verified, the terminal will display

```
THE SUMCHECK (TOTAL) IS: HHHHHH
```

where "HHHHHH" is the hexadecimal summation of all of the data contained in all of the pairs.

If the operation is unsuccessful, the terminal will display an error message below the `VERIFYING • • • • • •` message and the socket lamp will light red. See the Error Messages section for an explanation of the error message.

7. Remove the verified device. To verify another device, return to step 6. To exit the Verify operation, press *<ESC>*. The programmer will continue to verify devices until you press *<ESC>*.

Verifying Long-Word-Wide Devices

Use the **Verify Long Word** command to verify device data against data that is loaded into RAM in 32-bit-wide word format. This command compares four 8-bit-wide devices to 32-bit-wide data in the programmer's RAM. (See Loading Long Word-Wide Masters for information on which device number corresponds to which block of RAM data.) More than one unique set of devices can be verified with one Verify Long Word command, provided that the data for all devices is loaded into the programmer's RAM, and multiple copies of each set can be verified with one Verify Long Word command.

Command Format: VL [<Vcc Selection>] [<number of devices>]

Procedure	Example Key Sequence
1. Load the master data into the programmer's RAM using either a Load Long Word command or a download operation.	
2. Make sure that the correct device type is selected.	
3. Type in the Verify Long Word command letters, VL.	VL

Procedure	Example Key Sequence
4. Type in the letter (L or H) that corresponds to the proper Vcc high and low settings shown in the following table, or, to verify the installed device with the voltage settings recommended by the manufacturer, do not type anything for this parameter. (The space after "VL" is optional.)	VL L
5. Type a space, type in the number of devices to be verified (which must be a multiple of 4) and press carriage return, or, to verify the default number of devices, press carriage return without entering a number of devices. The terminal will display INSERT DEVICE NO: 01	VL L 8 <CR> or VL L <CR>
6. Insert the correct device of the set into the socket; the devices will be verified in the same order that they were programmed in originally. Press carriage return to execute the command for the device number displayed, or, to verify a different device, type in the number of the device before pressing carriage return. This feature allows you to verify a partial set.	

Selection	High Voltage	Low Voltage
L	5.2 V	4.8 V
H	5.5 V	4.5 V

Procedure	Example Key Sequence

If the operation is successful, the terminal will display

```
VERIFYING • • • • • •
INSERT DEVICE NO: NN
```

where "NN" is the number of the next device to be verified. ("NN" is always the number of the device previously verified plus 1.)

If the last device of all of the sets was just verified, the terminal will display

```
THE SUMCHECK (TOTAL) IS: HHHHHH
```

where "HHHHHH" is the hexadecimal summation of all of the data contained in the sets just verified.

If the operation is unsuccessful, the terminal will display an error message below the `VERIFYING • • • • • •` message and the socket lamp will light red. See the Error Messages section for an explanation of the error message.

7. Remove the verified device. To verify another device, return to step 6. To exit the Verify operation, press *<ESC>*. The programmer will continue to verify device sets until you press *<ESC>*.

EDITING MEMORY

The following commands allow you to edit data that has been loaded into the Model 212 Programmer RAM. With these commands, you can display the data, edit individual addresses, fill a segment of memory with data, insert new data, delete data, copy data, byte swap data, and search memory for up to 8 bytes of data. The commands are presented in the following order:

Command Name	Command Format
Memory Display	M <start address> <end address>
Memory Modify	M <start address>
Memory Fill	M <start address> <end address> <data1> [<data2...data8>]
Insert	I <start address> <data1> [<data2...data8>]
Delete	D <start address> [<end address>]
Transfer (Copy)	T <start address> <end address> <destination address>
Byte Swap	BS <start address> <end address>
Memory Search	MS <start address> <end address> <data1> [<data2...data8>]
	Microprocessors Only
Encryption Display	E [<start address><end address>]
Encryption Modify	E [address]
Encryption Fill	E <start address><end address><data1>[<data2>...<data8>]

Displaying Memory

Use the **Memory Display** command to display the contents of RAM on the terminal screen. The RAM addresses are arranged on the screen in a table format, with each row containing 16 bytes of consecutive data (see example). To determine the address of a byte of RAM, add the 6-character RAM address shown at the very left of the line containing the data to the 2-character address shown at the top of the column containing the data. The ASCII-CODE column on the far right contains the ASCII equivalent of the hexadecimal data on each line.

Command Format: M <start address> <end address>

Procedure	Example Key Sequence
1. Type the Memory Display command letter, M.	M
2. Type in the address of the first byte of RAM that you want to display on the screen.	M Ø
3. Type a space, type the address of the last byte of RAM that you want to display on the screen and press carriage return.	M Ø 1F<*CR*>
4. Type CTRL-S to halt the display and CTRL-Q to resume the display.	

Example

Display the contents of RAM addresses 0 through 1F:

```
>M 0 1F<CR>
        00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F    ASCII-CODE
000000  32 1A 5B 47 43 52 03 4B CB 20 54 33 4A 11 20 52  2.[GCR.K. T3J. R
000010  52 20 11 4A 33 54 20 CB 4B 03 AA 52 43 47 4B 1A  R.J3T .K..RCGK..
```

NOTE

To determine the memory address of a data byte, add the 2-character column heading of the data byte column to the row heading (6-character hexadecimal number at the far left of each line).

A period under the ASCII-CODE heading represents hexadecimal data without a printable ASCII equivalent.

Modifying Single Memory Locations

Use the **Memory Modify** command to display and edit the contents of RAM locations one-by-one. The Memory Modify command causes the programmer to enter an editing mode which allows you to display the contents of RAM addresses one-by-one and type in new data for each address displayed. To exit Memory Modify mode and return to the command prompt (>), press *<ESC>*.

Command Format: M <start address>*<CR>*
<new data>*<CR>*
.
.
.
<ESC>

Procedure	Example Key Sequence
1. Type the Memory Modify command letter, M.	M
2. Type in the address of the first memory location you want to modify and press carriage return.	M 6*<CR>*
3. After the RAM address and its contents have been displayed, type in the new data and press carriage return to accept it. The new data will appear to the right of the old data. If you do not want to modify the data at the address displayed, but want to display the next RAM address for modification, press carriage return without entering new data.	E5*<CR>* or *<CR>*

Procedure	Example Key Sequence
4. To modify the next memory location, return to step 3.	
5. To exit Memory Modify mode and return to the command prompt (>), press <*ESC*>. If you press <*ESC*> after typing in new data but before pressing carriage return to accept the new data, the new data for the last address displayed will be ignored.	

Example

Enter Memory Modify mode starting at address 6 and type in new data for addresses 6, 7, and 8:

```
>M 6<CR>
000006   F1   E5<CR>
000007   FF   21<CR>
000008   00   21<CR>
000009   43   <ESC>
```

Filling a Segment of Memory

Use the **Memory Fill** command to fill (replace) a segment of RAM with up to eight bytes of new data. The original data in these RAM locations are overwritten with the new data.

Command Format: M <start address> <end address> <data1> [<data2>...<data8>]

Procedure	Example Key Sequence
1. Type the Memory Fill command letter, M.	M
2. Type in the address of the first byte of RAM you want to fill with new data.	M 20
3. Type a space and then type in the address of the last byte of RAM you want to fill with new data.	M 20 2F
4. Type a space, type in the data to be input into the specified memory range and press carriage return. Separate each hexadecimal byte with a space. You can insert up to eight hexadecimal bytes (or eight ASCII characters) with one Memory Fill command. If you input more data than can fit in the range specified, the excess data is ignored.	M 20 2F 11 22<*CR*>

Example

Display memory contents before fill:

```
>M 20 3F<CR>
        00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F    ASCII-CODE
000020  FF FF FF FF EE EE EE EE DD DD DD DD CC CC CC CC ................
000030  BB BB BB BB AA AA AA AA 99 99 99 99 88 88 88 88 ................
```

Fill memory:

```
>M 20 2F 11 22<CR>
```

Display memory contents after fill:

```
>M 20 3F<CR>
        00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F    ASCII-CODE
000020  11 22 11 22 11 22 11 22 11 22 11 22 11 22 11 22 ."."."."."."."."
000030  BB BB BB BB AA AA AA AA 99 99 99 99 88 88 88 88 ................
```

Inserting New Data

Use the **Insert** command to insert new data into RAM. The original data is not overwritten by this command, as it is with the Memory Modify and the Memory Fill commands, but is shifted up (to a higher address) to accommodate the new data. Data at the end of memory is lost.

Command Format: I <start address> <data1> [<data2>...<data8>]

Procedure	Example Key Sequence
1. Type the Insert command letter, I.	I
2. Type in the starting address where you want the new data to be inserted.	I 5
3. Type a space, type in the new data and then press carriage return. Separate each hexadecimal byte with a space. You can insert up to eight hexadecimal bytes (or eight ASCII characters) with one Insert command.	I 5 11 11 11 11<*CR*>

Example

Display memory contents before insertion of new data:

```
>M 0 1F<CR>
       00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F    ASCII-CODE
000000 FF FF FF FF EE EE EE EE DD DD DD DD CC CC CC CC ................
000010 BB BB BB BB AA AA AA AA 99 99 99 99 88 88 88 88 ................
```

Insert new data:

```
>I 5 11 11 11 11<CR>
```

Display memory contents after insertion of new data:

```
>M 0 1F<CR>
       00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F    ASCII-CODE
00000  FF FF FF FF EE 11 11 11 11 EE EE EE DD DD DD DD ................
00010  CC CC CC CC BB BB BB BB AA AA AA AA 99 99 99 99 ................
```

Deleting Data

Use the **Delete** command to delete data bytes from memory. Data following the deleted addresses are shifted down (to a lower address) to fill in the deleted addresses. "00" is filled in the vacated addresses at the end of RAM.

Command Format: D <start address> [<end address>]

Procedure	Example Key Sequence
1. Type the Delete command letter, D.	D
2. Type in the address where you want to begin deleting data.	D 4
3. To delete a single byte of data, press carriage return without entering an end address. To delete a range of data, type a space, type the address of the last memory byte that you want to delete and then press carriage return.	D 4<*CR*> or D 4 7<*CR*>

Example

Display memory contents before deletion:

```
>M 0 1F<CR>
        00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F    ASCII-CODE
000000  66 66 66 66 11 11 11 11 77 77 77 77 77 77 77 77 FFFF....WWWWWWWW
000010  88 88 88 88 88 88 88 88 88 88 88 88 88 88 88 88 ................
```

Delete data:

```
>D 4 7<CR>
```

Display memory contents after deletion:

```
>M 0 1F<CR>
        00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F    ASCII-CODE
00000   66 66 66 66 77 77 77 77 77 77 77 77 88 88 88 88 FFFFWWWWWWWW....
00010   88 88 88 88 88 88 88 88 88 88 88 88 00 00 00 00 ................
```

Transferring (Copying) Memory

Use the **Transfer** command to copy a block of RAM data from one RAM location to another. This command copies the block specified by the start address and end address to the destination address. The original data starting at the destination address is overwritten by the copied block.

Command Format: T <start address> <end address> <destination address>

Procedure	Example Key Sequence
1. Type the Transfer command letter, T.	T
2. Type in the address of the first byte of RAM to be copied.	T Ø
3. Type a space and then type in the address of the last byte of RAM to be copied.	T Ø F
4. Type a space and then type in the address that you want the first byte of the specified data range to be copied to, or the destination address, and press carriage return. The data will be copied into consecutive RAM addresses, starting with the destination address.	T Ø F 25<*CR*>

Example

Display memory contents before transfer:

```
>M 0 3F<CR>
        00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F    ASCII-CODE
000000  99 99 99 99 99 99 99 99 99 99 99 99 99 99 99 99 ................
000010  77 77 77 77 77 77 77 77 77 77 77 77 77 77 77 77 WWWWWWWWWWWWWWWW
000020  66 66 66 66 66 44 44 44 44 44 44 44 44 44 44 44 FFFFFDDDDDDDDDDD
000030  44 44 44 44 44 11 11 11 11 11 11 11 11 11 11 11 DDDDD...........
```

Transfer data:

```
>T 0 F 25<CR>
```

Display memory contents after transfer:

```
M 0 3F<CR>
        00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F    ASCII-CODE
00000   99 99 99 99 99 99 99 99 99 99 99 99 99 99 99 99 ................
00010   77 77 77 77 77 77 77 77 77 77 77 77 77 77 77 77 WWWWWWWWWWWWWWWW
```

Byte Swapping Memory

Use the **Byte Swap** command to swap the contents of adjacent memory byte pairs over a specified range.

Command Format: BS <start address> <end address>

Procedure	Example Key Sequence
1. Type the Byte Swap command letters, BS.	BS
2. Type in the address of the first byte of RAM that you want to swap.	BS Ø
3. Type a space, type the address of the last byte of RAM that you want to swap or press carriage return.	BS Ø 1F <CR>

Example

Display memory contents before Byte Swap:

```
>M0 1F <CR>

        00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F    ASCII-CODE
000000  AB FE 5A 67 00 F7 10 12 13 4B A4 5C CA 21 27 5A  ..Zg.....K.\.!'Z
000010  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
```

Byte Swap data:

```
>BS  0  1F  <CR>
```

Display memory contents after Byte Swap:

```
>M0  1F  <CR>

        00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F    ASCII-CODE
000000  FE AB 67 5A F7 00 12 10 4B 13 5C A4 21 CA 5A 27  ..gZ....K.\.!.Z'
000010  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
```

Searching Memory

Use the **Memory Search** command to find a block of hexadecimal data up to eight bytes long.

Command Format: MS <start address> <end address> <data1> [<data2>...<data8>]

	Procedure	Example Key Sequence
1.	Type the Memory Search command letters, MS.	MS
2.	Type in the address of the memory location where you want the search to begin.	MS Ø
3.	Type a space and then type in the memory address where you want the search to end.	MS Ø 3F
4.	Type a space and then type in the data that you want to search for and press carriage return. Separate each hexadecimal byte with a space. You can type in up to eight hexadecimal bytes. If the data is found, the terminal will display MEMORY MATCH AT ADDRESS HHHH, where "HHHH" is the address of the first byte of the search string that was typed in. If the data is not found, the terminal will display MEMORY SEARCH FAILURE.	MS Ø 3F 12 34<*CR*>

Example

Search memory block 0 through 3F for the hexadecimal data 12 34:

```
>MS 0 3F 12 34<CR>
MEMORY MATCH AT ADDRESS 000038
```

Display memory block searched:

```
>M 0 3F<CR>
        00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F    ASCII-CODE
000000  77 77 77 77 77 77 77 77 77 77 77 77 77 77 77 77  WWWWWWWWWWWWWWWW
000010  77 77 77 77 77 77 77 77 77 77 77 77 77 77 77 77  WWWWWWWWWWWWWWWW
000020  77 77 77 77 77 77 77 77 77 77 77 77 77 77 77 77  WWWWWWWWWWWWWWWW
000030  77 77 77 77 77 77 77 77 12 34 77 77 77 77 77 77  WWWWWWWW.4WWWWWW
```

Displaying Memory Reserved for Encryption - Microprocessors Only

Use the **Encryption Display** command for a screen display of the contents of the 32 bytes of Key Data reserved in RAM. The RAM addresses are arranged on the screen in a table format, with each row containing 16 bytes of consecutive data (see example). To determine the address of a byte of RAM, add the 6-character RAM address shown at the very left of the line containing the data to the 2-character address shown at the top of the column containing the data. The ASCII-CODE column on the far right contains the ASCII equivalent of the hexadecimal data on each line.

Command Format: E <start address> <end address>

Procedure	Example Key Sequence
1. Type the Encryption Display command letter, E.	E
2. Type in the address of the first byte of RAM that you want to display on the screen.	E 0
3. Type a space, type the address of the last byte of RAM that you want to display on the screen and press carriage return.	E 0 1F<*CR*>
4. Type CTRL-S to halt the display and CTRL-Q to resume the display.	

Example

Display the contents of RAM addresses 0 through 1F:

```
>E 0 1F<CR>
        00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F    ASCII-CODE
000000  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ................
000010  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ................
```

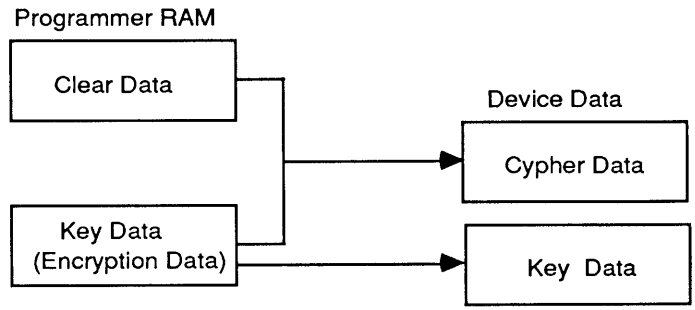
NOTE

To determine the memory address of a data byte, add the 2-character column heading of the data byte column to the row heading (6-character hexadecimal number at the far left of each line).

A period under the ASCII-CODE heading represents hexadecimal data without a printable ASCII equivalent.

Modifying Single Encryption Locations - Microprocessors Only

Use the **Encryption Modify** command to display and edit the contents of the 32 bytes of Key Data reserved in RAM. The Encryption Modify command causes the programmer to enter an editing mode which allows the contents of RAM reserved for encryption to be edited address-by-address.
To exit Encryption Modify and return to the command prompt (>), press *<ESC>*.



Command Format: E <start address>*<CR>*
<new data>*<CR>*
.
.
.
<ESC>

Procedure	Example Key Sequence
1. Type theEncryption Modify command letter, E.	E
2. Type in the address of the first encryption location you want to modify and press carriage return.	E 6*<CR>*

Procedure	Example Key Sequence
3. After the RAM address and its contents have been displayed, type in the new data and press carriage return to accept it. The new data will appear to the right of the old data. If you do not want to modify the key data at the address displayed, but want to display the next RAM address for modification, press carriage return without entering new data.	E5<*CR*> or <*CR*>
4. To modify the next memory location, return to step 3.	
5. To exit the Encryption Modify mode and return to the command prompt (>), press <*ESC*>. If you press <*ESC*> after typing in new data but before pressing carriage return to accept the new data, the new data for the last address displayed will be ignored.	

Example

Enter Encryption Modify mode starting at address 6 and type in new data for addresses 6, 7, and 8:

```
>E 6<CR>
000006   F1   E5<CR>
000007   FF   21<CR>
000008   00   21<CR>
000009   43   <ESC>
```

Filling a Segment of Memory Reserved for Encryption- Microprocessors Only

Use the **Encryption Fill** command to fill (replace) up to 32 bytes of RAM reserved for encryption with up to eight bytes of new data. The original data in these RAM locations are overwritten with the new data.

Command Format: E <start address> <end address> <data1> [<data2>...<data8>]

Procedure	Example Key Sequence
1. Type the Encryption Fill command letter, E.	E
2. Type in the address of the first byte of RAM you want to fill with new data.	E 20
3. Type a space and then type in the address of the last byte of RAM you want to fill with new data.	E 20 2F
4. Type a space, type in the data to be input into the specified memory range and press carriage return. Separate each hexadecimal byte with a space. You can insert up to eight hexadecimal bytes (or eight ASCII characters) with one Encryption Fill command. If you input more data than can fit in the range specified, the excess data is ignored.	E 20 2F 11 22<*CR*>

Example

Display memory contents before fill:

```
>E 00 1F<CR>
       00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F    ASCII-CODE
000000 FF FF FF FF EE EE EE EE DD DD DD DD CC CC CC CC ................
000010 BB BB BB BB AA AA AA AA 99 99 99 99 88 88 88 88 ................
```

Fill encryption:

```
>E 00 0F 55 AA 00 FF<CR>
```

Display memory contents after fill:

```
>M 00 1F<CR>
       00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F    ASCII-CODE
000000 55 AA 00 FF 55 AA 00 FF 55 AA 00 FF 55 AA 00 FF U...U...U...U...
000010 BB BB BB BB AA AA AA AA 99 99 99 99 88 88 88 88 ................
```

SUMCHECKING DATA

Use the following commands to cause the programmer to sumcheck (total) or checksum (EXOR) the data stored in the programmer's RAM. These commands are useful for obtaining a new sumcheck of any range of RAM, or verifying that the data stored in RAM is the same as data programmed into a device or data downloaded through the I/O port. The 3-byte sumcheck may also be stored in RAM. This feature allows you to program the sumcheck into a device so that software which calculates a sumcheck during a self test can verify its own sumcheck against the sumcheck stored in the device.

Performing a Sumcheck (Total)

Use the **Sumcheck (Total)** command to calculate a 3-byte hexadecimal summation of data stored in the programmer's RAM. This sumcheck can be used to verify that data was downloaded to the programmer correctly, to sumcheck only a portion of memory, or to produce a current sumcheck after you have edited memory. A sumcheck is automatically performed after most programming operations, so you should not need to use this command when programming or verifying devices using master devices.

Command Format: CT <start address> <end address> [<destination address>]

Procedure	Example Key Sequence
1. Type in the Sumcheck (Total) command letters, CT.	`CT`
2. Type in the address of the first byte of RAM to be checked by the sumcheck operation.	`CT 0`
3. Type a space and then type in the address of the last byte of RAM to be checked.	`CT 0 3FFF`

Procedure	Example Key Sequence
4. To perform the sumcheck without storing the sumcheck value in RAM, press carriage return, or, to store the sumcheck value in RAM, type a space, type in the RAM address that you want the sumcheck value to be stored in and press carriage return. If you store the sumcheck value in RAM, it will overwrite any data previously stored in the RAM locations to which the sumcheck is written. Make sure that there is no data that you want to retain located in the addresses that the sumcheck will be written to. The sumcheck (total) is calculated and then displayed on the screen. The sumcheck is also stored in memory if you specified a destination address. *NOTE* *The three-byte sumcheck is stored in memory addresses 4000 through 4003. The low byte is stored at address 4000; the middle byte is stored at 4001; and the high byte is stored at 4002. A "dummy" byte of ØØ is stored at 4003 to keep the sumcheck value an even number of bytes.*	CT 0 3FFF<*CR*> or CT 0 3FFF 4000<*CR*>

Example

Calculate the sumcheck total of memory addresses 0 through 3FFF and store the sumcheck value in address 4000:

```
>CT 0 3FFF 4000<CR>
THE SUMCHECK (TOTAL) IS: 030201
```

Display the RAM data

```
>M 4000 4003 <CR>
              00  01  02  03
    004000 01  02  03  00
                           ↑
                  (Dummy Byte)
```

Performing an Exclusive-OR Checksum

Use the **Checksum (EXOR)** command to calculate an exclusive-OR checksum of RAM. You may wish to use this command instead of the sumcheck (total) command if data being downloaded to the programmer includes an exclusive-OR checksum, or if you prefer to work with an exclusive-OR summation. The exclusive-OR checksum is a 1-byte hexadecimal value representing the result of exclusive-ORing each successive byte of the data range specified. For example,

```
  0000 0001
+ 1111 1111
  1111 1110
```

Command Format: CX <start address> <end address> [<destination address>]

Procedure	Example Key Sequence
1. Type in the Checksum (EXOR) command letters, CX.	CX
2. Type in the address of the first byte of RAM to be checked by the checksum operation.	CX 0
3. Type a space and then type in the address of the last byte of RAM to be checked by the checksum operation.	CX 0 3FFF

Procedure	Example Key Sequence
4. To perform the checksum operation without storing the checksum value in RAM, press carriage return. To store the checksum in RAM, type a space, type the RAM address that you want the checksum to be stored in and press carriage return. If you store the checksum value in RAM, it will overwrite any data previously stored in that RAM location. The checksum (EXOR) value is calculated and then displayed on the screen. The checksum is also stored in RAM if you specified a destination address.	CX 0 3FFF<*CR*> or CX 0 3FFF 4000<*CR*>

Example

Calculate the checksum of memory addresses 0 through 3FFF and store the checksum value in address 4000:

```
>CX 0 3FFF 4000<CR>
THE CHECKSUM (EXOR) IS:  FE
```