IM3016 MULTI-MASTER PROM PROGRAMMER

OPERATING MANUAL

Revised 05/24/85

INTERNATIONAL MICROSYSTEMS, INC.

Corporate Headquarters:

790 E. Arques Ave
Sunnyvale, CA 94086

408-245-7180

IM3016
TABLE OF CONTENTS

SECTION 4: IMMEDIATE COMMAND DEFINITIONS

SECTION 5: RUN MODE DEFINITIONS

SECTION 6: SAMPLE PROGRAMMING SESSION

SECTION 7: INTERFACING THE IM3016

SECTION 8: TESTING THE IM3016

SECTION 9: GENERAL INFORMATION ON TERMINAL AND COMPUTER CONTROL

SECTION 1: OVERVIEW

1.0 IM3016 DESCRIPTION

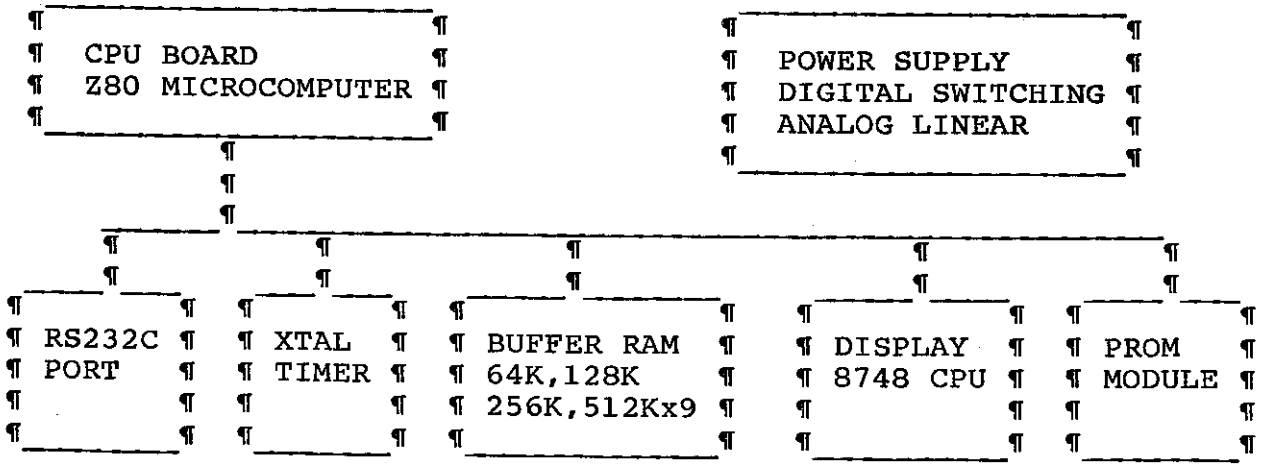
The IM3016 is a microprocessor based Multi-Master PROM Programmer in which from one to sixteen different master PROMs can be copied in one programming cycle. A complete IM3016 consists of two major elements: a Base Unit and a PROM module. Easily changed modules allow the IM3016 to program a full range of EPROMs and EEPROMs.

The IM3016 Base Unit includes: power supply, control electronics, 8-digit alphanumeric display, keyboard for entry of operating modes such as PROM types and SET size, error indicators, one RS232C serial port, and a 64Kx9 Buffer RAM. Expansion to 128Kx9, 256Kx9, and 512Kx9 Buffer RAM is optional.

The IM3016 modules are attached to the Base Unit using two pin hinges. Modules are designed to program certain classes of PROMs, and software selection is used for choosing a specific PROM type. For example, the IM3016-MOD-1V-28 programs all popular 5V-only EPROMs including the 2758, 2716, 2516, 2732, 2732A, 2564, 2764, 2764A, 27128, 27128A, 27256, and 68764/66.

All IM3016 module PROM sockets are electrically isolated from each other so that faulty conditions existing at one socket will not affect any other socket. This is an advanced feature that is not generally found on multi-socketed programmers. Isolated sockets allow for detection of faulty or mis-socketed PROMs. For example, the IM3016 can detect whether a 2532 type PROM has been inserted by mistake as a 2732 type PROM.

1.1 BLOCK DIAGRAM OF THE IM3016.



1.2 CPU

The IM3016 CPU incorporates a Z80 microprocessor. The program memory resides in PROM. The software is written in the high level "C" programming language which provides reliable code and superior documentation. All critical timing required for the PROM programming sockets is derived from a crystal time base and a LSI timer IC connected to the Z80.

1.3 BUFFER RAM

The Buffer RAM in the IM3016 is used to hold the data to be copied into the PROMs in the module sockets. 64Kx1 dynamic RAMs are used in a x9 configuration where one bit is a parity check bit. Because the data to be copied into the user's PROMs must be resident in the Buffer RAM, the size of the Buffer RAM is a limitation on the total number of different PROMs that may be programmed at one time.

In general, for duplicating 4Kx8 or smaller PROMs, the standard 64Kx9 Buffer RAM is sufficient. For duplicating 32Kx8 or smaller PROMs, the 512x9 Buffer RAM is preferable. To determine the RAM size to meet your specific requirements, use the following simple formula:

Required RAM size = SET size x PROM max address

where SET size = number of unique PROMs in the set.

EXAMPLE: User desires to copy 4 different 2764's (8Kx8).

For a 2764 the max address = 8K

Therefore:

Buffer RAM required = 4 (SET size) x 8K (max address)
= 32Kx9 BUFFER RAM

Since the IM3016 module has 16 sockets, the user can duplicate 4 such sets in one programming cycle.

1.4 DISPLAY AND KEYBOARD

The IM3016 display and keyboard are controlled by an 8748 type processor interfaced to the Z80 CPU. The display is made up of eight 14-segment 0.5 inch LED digits and is capable of displaying all standard letters and numbers. In addition, two large LEDs are used for LOAD and BLANK error indicators and a beeper is provided. The keyboard consists of a custom 16 position keyboard plus one separate START key for single key operation. A hinged cover is provided to allow for one key control during production PROM programming.

1.5 SERIAL INTERFACE

The IM3016 comes with one RS232C interface which is connected to a standard 25-pin "D" female connector. The serial parameters such as baud rate, parity, etc., are selected by an 8 position dip switch. The serial port is normally half duplex and accomodates baud rates from 110 baud to 19.2K baud. A request-to-send/clear-to-send protocol is not required as most formats can be loaded with no wait states. Optional current loop and parallel interfaces are possible.

The serial port is used for loading the Buffer RAM with data from an external data device such as a microcomputer development system, or for operation of the IM3016 by a computer. Various data formats are available including Intellec Hex, Motorola S-Record and TEKHEX.

1.6 POWER SUPPLY

The IM3016 power supply provides power for the base unit as well as the module electronics. The digital logic is powered by a switching supply. Two linear supplies provide power for the module analog circuitry but all necessary analog reference voltages are set on each individual module. The line power connector is filtered and a circuit breaker is used for line current protection.

1.7 MODULES

Because the IM3016 uses easily changed modules, it is adaptable to many present and future PROM types. At present the IM3016 has available two programming modules which are listed as follows:

IM3016-MOD-3V-24.....Programs three voltage PROMs including 2704, 2708, TMS2716.

IM3016-MOD-1V-28.....Programs single voltage 24 and 28 pin PROMs including 2758, 2516, 2716, 2532, 2732, 2732A, 2564, 2764, 2764A, 27128, 27128A, 68764/68766, and 27256.

Each module contains sixteen zero insertion/extraction force sockets.

1.8 EXTERNAL CONTROL

The IM3016 is capable of complete external control via its RS232C port. Section 13 describes over 35 commands which are available. In addition, IM has developed a software package written in "C" for installation by the user on his/her own computer. Please feel free to consult the factory concerning this package or other applications assistance in solving your PROM programming requirements.

1.9 RESET SWITCH

A momentary reset switch on the back of the IM3016 resets the processor and forces the user to redefine any parameters that have been previously entered. However, the data in Buffer RAM will not be changed unless the reset occurs during a Buffer RAM operation. The need to use the reset key should occur only under rare circumstances such as a power interruption. The switch is recessed and a two second time delay has been incorporated into the switch in order to prevent accidental depression.

1.10 MECHANICAL AND ELECTRICAL SPECIFICATIONS

Power: 120 VAC, 1.5 amp for standard IM3016
100, 200, 240 VAC are optional

Circuit Protection: 2 amp fuse and line filter.

Size: 5.5"(13.9cm)H x 19.5"(49.5cm)W x 10.0"(25.4cm)D

Weight: 20 lbs(9.1kg) including module

Shipping Weight: 23 lbs(10.4kg) including module

SECTION 2: UNPACKING AND INITIAL CHECKOUT

2.0 UNPACKING INSTRUCTIONS

Remove unit from shipping carton and inspect for shipping damage or missing items as detailed by packing slip. Notify carrier immediately of concealed damage.

Your IM3016 may or may not be shipped with a module installed. If the module received is not installed, refer to Section 2.1 on changing modules and attach the module. With the power cord disconnected, inspect the module and Base Unit for:

1. Loose ICs, especially in 8-, 14-, and 16-pin sockets.
2. Loose connectors on the power supply, CPU, display board, RAM board and module.
3. Check for loose or broken components.
4. Double check that the CPU-to-module cables are correctly connected together.

You are now ready to tighten the module to the Base Unit using the two push-pull fasteners on the module.

2.1 CHANGING MODULES

To change modules, the IM3016 should be turned off. The module is changed by pulling up on the push-pull fasteners, then swinging the module up and over to the left, placing the module horizontal to the case. Disconnect the 50-pin flat cable from the module, then disconnect the 10-pin power cable from the module. Be sure to note the orientation of both of these cables. Slide the module up and off the hinges. To attach a new module, reverse this procedure. When reconnecting cables, be sure the plugs are installed properly.

2.2 INITIAL PROGRAMMER CHECKOUT

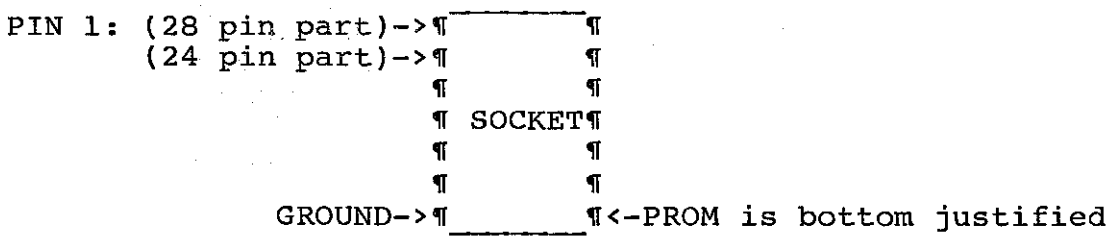
1. Install power cord.

2. Turn programmer ON. The red LED power light on the top left of the Base Unit should be on. The display should read I.M...3.0.1.6.

3. Define the TYPE of PROM to be programmed as given in the chart below: This chart is also found inside the keyboard cover and in Appendix A. (TYPE ENTER XXX ENTER).

TYPE	PROM	TYPE	PROM
310	2704	120	68764/68766
311	2708	121	2764
312	TMS2716	122	2564
		124	FAST2764
110	2758		
111	2716		
112	2732	131	27128
113	2732A	132	27256
114	2532	133	2764A
		134	27128A

4. Install programmed test PROM to be copied in Socket 1 according to diagram.



5. MOVE PROM into Buffer RAM. (MOVE ENTER ENTER).

6. Place test PROM in Socket 2 and verify. (VERIFY ENTER).

7. Repeat for each socket until all sockets have been verified.

8. (Optional) Now load 16 erased PROMs of the same type into all sockets. BLANK check, PROGRAM, and VERIFY parts using the Automatic Mode (RUN ENTER START). The display will show the checksum of the master EPROM set at the end of the cycle.

9. (EXIT)

SECTION 3: OPERATING THE IM 3016

3.0 DEFINITION OF A PROM SET

The IM3016 is designed to program sets of PROMs. A PROM set is defined as a group of PROMs, each with the same or different data, and in the most general case, each of different manufacturer's part numbers (i.e., different types). The maximum number of PROMs in a set is 16. If the PROM set consists of 8 or less PROMs, the IM3016 can program multiple sets with the restriction that the total number of PROMs is 16 or less. Also, during the same program cycle only identical type PROMs may be programmed.

3.1 BUFFER RAM

The IM3016 programs PROMs from data residing in an internal memory which will be referred to as the "Buffer RAM". The Buffer RAM may be loaded with data from master PROMs in the PROM sockets or by loading a PROM object file from an external mass storage device such as a computer. The size of the Buffer RAM is presently set at 64Kx9, expandable to 128Kx9, 256Kx9, and 512Kx9. Therefore, an address in Buffer RAM is specified by five hexadecimal digits (e.g., address = $1A34E).

3.2 MAPPING OF PROM SOCKETS TO BUFFER RAM

The IM3016 provides a number of different ways to map the PROM sockets to RAM. The default mapping is determined by the TYPE of PROM and the SET Size. Upon power up and after any definition of the TYPE and SET size, the PROM sockets are remapped with Socket 1 mapped to the bottom of RAM, Socket 2 mapped to the next location following Socket 1, etc., until a complete SET has been mapped. The first socket following a complete SET is mapped to the same address as Socket 1, etc., until all sockets have been mapped or a complete SET is impossible.

Example: Mapping of module sockets to Buffer RAM

Relevant Conditions:
1. TYPE = 2732 (max address = $FFF)
2. SET Size = 5.

Socket	Buffer RAM Address	
1	$0000-$0FFF	
2	$1000-$1FFF	
3	$2000-$2FFF	
4	$3000-$3FFF	
5	$4000-$4FFF	
6	$0000-$0FFF	Same as Socket 1
7	$1000-$1FFF	" 2
8	$2000-$2FFF	" 3
9	$3000-$3FFF	" 4
10	$4000-$4FFF	" 5
11	$0000-$0FFF	" 1
12	$1000-$1FFF	" 2
13	$2000-$2FFF	" 3
14	$3000-$3FFF	" 4
15	$4000-$4FFF	" 5
16	Not Mapped	

Note that Socket 16 is not mapped. This is because the SET Size of 5 does not divide evenly into 16. Socket 16 will remain "dead" for all operations until a different SET Size is defined. For instance, if a SET Size of 16 is chosen, all sockets are active and assigned a unique position in RAM. Likewise, if a SET Size of 1 is chosen, all sockets are active and mapped to the same data in RAM. In this latter case, the IM3016 is similar to "Gang" type PROM programmers.

Other commands that affect the mapping are special functions 3 and 5. SPEC 3 allows the user to specify a 16 bit data format (see Section 7.3) and SPEC 5 allows assigning the start address of Socket 1 to any 1K hex boundary in the IM3016 buffer RAM (see Section 7.5). When in External Control, the PROM sockets may be mapped in any arbitrary manner the user choses.

3.3 SOFTWARE MONITOR

The software in the IM3016 employs a monitor which controls the selection of the various commands that can be performed. The user is notified that the monitor is ready to accept a new keyboard command when all 8 display decimal points are visible on the display panel (e.g., I.M...3.0.1.6.).

3.4 COMMAND TYPES

There are two types of commands which are accepted by the monitor; these are Immediate Commands and RUN Mode Commands.

3.5 IMMEDIATE COMMANDS

Immediate Commands operate on the PROM sockets as individual entities and not as sets. They are initiated immediately after entry. An example of an Immediate Command is the key entry BLANK ENTER which results in the PROM sockets being tested for erased PROMs. Other Immediate Commands are: CKSUM, CLEAR, ENTER, EXIT, FUNC, LIST, MOVE, PROG, SET, SPEC, STEP, TEST, TYPE, and VERIFY.

3.6 RUN MODE COMMANDS

RUN Mode Commands operate only on complete sets of PROMs and only after the RUN Mode is entered. RUN Mode Command sequences are not executed after entry, but are stored for execution at a later time. The TYPE of PROM, PROM SET size, and a previous load of the Buffer RAM (using the MOVE command) must all have been completed before the PROM sockets are activated in the RUN Cycle Mode.

EXAMPLE: Entry and execution of a RUN Mode Command.

(1) Define PROM function to be performed in RUN Mode.

Key Entry	Display	Explanation
FUNC ENTER	F= BPVC	Displays previous function sequence. Any combination of the commands BLANK, PROG, VERIFY, CKSUM may be used to define the RUN Mode Command sequence.
BLANK	F=B	First operation on socket is to test PROMs for erased condition.
PROG	F=BPV	Program and verify active sockets. NOTE: Anytime the PROG key is entered, VERIFY is automatically appended. The VERIFY may be deleted by entry of a CLEAR.
ENTER	F.=.B.P.V....	Return to monitor. The RUN Mode sequence has been defined as BLANK, PROG, VERIFY.

(2) User may optionally perform any **Immediate Commands** that he so desires.

(3) User enters into RUN Mode operation.

RUN	RUN	Entry into RUN Mode using RUN key.
ENTER	R.U.N.....	START light goes on. IM3016 waits for depression of START key before beginning the RUN Mode Command sequence.

(4) Keyboard door is closed. The IM3016 is now ready for one key operation via the START key. The module sockets are loaded with the desired number of erased PROMs (i.e., complete sets). The RUN Mode is initiated with depression of the START key.

START	B BUSY	IM3016 performs a load test and then blank checks PROMs.
	P WAIT	All activated sockets have blank PROMs and the programming cycle begins.
	BUSYNNNN	IM3016 starts to program a set(s) of PROMs using data from the Buffer RAM. "NNNN" represents the PROM address locations as they are programmed.

(5) The blank check, program, verify sequence is now completed. The results are displayed via socket LEDs and the pass/fail totals appear on the display.

V.P..=.N.X.=.M.	Program cycle has been completed. "N" PROMs passed, "M" PROMS failed. Green socket LEDs indicate pass. Red socket LEDs indicate fail. White socket LEDs indicate inactive sockets.

(6) The user now removes all PROMs and reloads sockets for the next RUN cycle.

3.7 COMMAND PROCESSING

Most RUN Mode and Immediate Commands require certain parameters to be defined before execution. For instance, to LIST a PROM in a socket, the socket must be active and the TYPE of PROM defined. Also, before any socketed PROMs are read or programmed, various tests are performed which are referred to as "Load Sequence" tests. During a Load Sequence extensive checking and error processing is done by the IM3016 in order to insure that neither the IM3016 nor any PROMs are accidently damaged.

3.8 ERROR PROCESSING

There are a number of different types of errors that are recognized by the IM3016. The main types are Command Entry Errors, Formatted I/O Errors, Load Errors at the PROM sockets, Programming Related Errors and Parity RAM Error. Appendix C lists the various error messages which are possible.

3.9 COMMAND ENTRY AND FORMATTED I/O ERRORS

Command errors result from incorrect command sequences such as trying to program a set of PROMs before the TYPE of PROM is defined. Such errors are flagged with appropriate error messages such as TYPE ERR. After a command error is recognized, control is returned to the monitor. Formatted I/O Error messages are handled in a similar fashion. For specific information, refer to Section 7 on Formatted I/O.

3.10 LOAD SEQUENCE ERRORS

Load Sequence Errors can occur while an operation is being performed on the PROMs in the module sockets. Before a PROM is read or programmed, the socket is activated with a "Load Sequence". During a Load Sequence, each socket is checked for those electrical parameters which determine whether a PROM is correctly inserted into a socket. The possible errors that may occur are given below and in Appendix C.

Display	ERROR Condition
NONE ACT	All sockets are empty.
VCC ERR	A PROM is misaligned or upside down.
OPEN PIN	A data line is open.
STUCK 1	A data line is stuck to a high (+5V) when it should be disabled.
STUCK 0	A data line is stuck to a low (0V) when it should be disabled.
LSET ERR	In the RUN Mode, only complete sets of PROMs may be operated on.

If any of the sockets exhibit the above conditions, the LOAD ERR lights up, the display indicates the type of error, and a red LED over the socket(s) indicates the faulty PROM(s). Also, any command in progress is halted and will not continue until the problem is corrected and the START or ENTER key is re-entered. Following the completion of the Load Sequence, the desired PROM command will be performed.

Once the Load Sequence has been successfully completed and the desired PROM related function is being executed, none of the socketed PROMs should be touched. For example, while listing the contents of a PROM, do not disturb the socketed PROMs in any way. Failure to observe this simple rule may damage the IM3016.

3.11 PROGRAMMING RELATED ERRORS

Other than Load Sequence Errors, four types of programming related errors can occur. These are Blank Check Errors, Verify Errors, Vp Current Limit Errors and Initial Programming Data Errors.

Blank Check and Verify Errors are really results and not errors in that they do not indicate inherently faulty PROMs. These commands are discussed in detail in Section 4.

Vp Current Limit Errors can be detected at any time during the programming of a PROM. If a Vp Current Limit Error occurs at the beginning of a program cycle, the following steps are taken:

1. The Load Error light is on.
2. The red LED above the socket at which the Vp error occured is on.
3. The beeper is sounded.
4. The display reads BAD VP.

The user should remove and replace the bad part. The IM3016 will initiate a new Load Sequence routine after depression of the START or ENTER key.

If, on the other hand, a Vp Current Limit Error occurs during a program cycle, the socket is deactivated and failed, but no discernable action takes place. Another PROM can be programmed later in the Immediate Mode to replace the faulty part.

The last type of programming error that can occur is associated with the programming of 5V only PROMs; that is, those PROMs programmed with the IM3016-MOD-1V-28 module. A full discussion of these data verification errors is located in Section 4 under the PROG key. In brief, if a data error verification is detected at the start of programming a PROM, the PROM is flagged with a red LED and an error message is displayed.

3.12 RAM PARITY ERROR

The Buffer RAM includes a parity check bit and the IM3016 will display a "E PARITY" if an internal RAM problem should occur. Once a parity error occurs, the integrity of the data in the Buffer RAM is in question. Either the IM3016 is malfunctioning or a power line interruption has occured. In order to continue, one must reset the IM3016 by use of the RESET switch in the back left corner or the mainframe. The data in the IM3016 Buffer RAM will not change due to the Reset, but all other parameters such as the TYPE, SET size, etc., must be reentered.

SECTION 4: IMMEDIATE COMMAND DEFINITIONS

4.0 IMMEDIATE MODE DEFINITIONS

Immediate Commands are those commands which either treat the PROMs in the module sockets as individual entries or are used for manipulating the data in the Buffer RAM. In contrast, RUN Mode Commands treat the socketed PROMs as sets and are to be used in production PROM programming. In order to clarify the difference, commands such as BLANK, PROG, and VERIFY are given two different definitions; one for the Immediate Mode (Section 4) and one for the RUN Mode (Section 5).

The following Immediate Mode Commands are covered in this section:

BLANK, CKSUM, CLEAR, ENTER, EXIT, FUNC, LIST, MOVE, PROG, SET, STEP, TYPE, and VERIFY.

The SPEC key is described in Section 7 and the TEST Key in Section 8.

4.1 BLANK

BLANK check PROMs in the module sockets.

BLANK is normally used to test if the PROMs in the sockets are erased (not programmed). If the type of PROM is an electrically eraseable device, the user is also prompted as to whether s/he wishes to erase the PROMs. After BLANK has been executed, the total number of passes and fails are displayed along with the lighting of the appropriate socket LEDs. It is recommended that if a large number of PROMs are to be BLANK checked, a SET size of one be chosen so that all sockets are active. When used in conjunction with the FUNC key, BLANK becomes part of the RUN Mode cycle definition (see Section 5).

EXAMPLE: BLANK checking PROMs

Relevant conditions:

1. A TYPE has been defined.

2. The sockets must be active.

Key Entry	Display	Explanation
BLANK	BLANK	
ENTER	B..P.=.0.X.=.3.	A Load Sequence is executed. PROMs are checked for erased condition. 0 pass (blank); 3 fail (not blank). Return to monitor.

4.2 CKSUM

In the Immediate Mode, either individual sockets (PROMs) are checksummed or the Buffer RAM block defined by the SET function is checksummed (i.e., the SET checksum).

CHECKSUM of a PROM

A PROM in a module socket is checksummed by entering the CKSUM key. After this is done, the checksum for all active sockets are displayed sequentially by repeated use of the ENTER key. The socket number is displayed along with a 4-digit number which is the simple 8-bit sum of all bytes read from the socket. The CLEAR key may be used to back up as long as the last active socket has not been checksummed.

EXAMPLE: CKSUM of an active socket (PROM).

Relevant conditions:

1. A TYPE has been defined.
2. The socket must be active.
3. Assume PROMs are in Sockets 1, 4 and 9.

Key Entry	Display	Explanation
CKSUM	CKSUM	
ENTER	CK PROM?	
ENTER	C X NNNN C 1 NNNN	Displays Socket 1 checksum "NNNN".
ENTER	C 4 NNNN	Displays Socket 4 checksum.
ENTER	C 9 NNNN	Displays Socket 9 checksum.
CLEAR	C 1 NNNN	Goes back to Socket 1.
ENTER	C 4 NNNN	
ENTER	C 9= NNNN	Last socket checksum displayed.
ENTER	C..9..N.N.N.N.	Return to monitor.

CHECKSUM of the Buffer RAM: The SET CHECKSUM

Buffer RAM is checksummed by entering the CKSUM key and by using the STEP or CLEAR keys as a toggle between PROM and RAM. The checksum of a set is a 5 digit number associated with a block of Buffer RAM defined by the given set. If the SET size is 1, the RAM checksum is equal to the simple sum of the RAM. However, if the SET size is greater than 1, the SET checksum is NOT the simple sum of the individual RAM blocks that make up the given set. The individual socket checksums are modified so that incorrectly sequenced master PROMs of an otherwise correct set can be recognized.

The Set checksum for sets greater than 1 is calculated as follows. First, the simple checksum is calculated for each RAM block determined by the TYPE, SET size, and Mapping of the sockets. Then each of these individual checksums has its nth bit complemented where n is the position in the set. Finally, all individual checksums are added together. The resulting sum is the SET checksum. This manipulation helps to detect the insertion of master PROMs in the wrong order.

EXAMPLE: CKSUM of a set residing in Buffer RAM.

Relevant conditions:

1. A TYPE and SET size has been defined.
2. The Buffer RAM has been filled with data using the MOVE function or a SPEC function LOAD operation.
3. Assume that the SET size is 3 and the max address of the PROM is $1000.
4. Assume that the PROMs in the SET have the following checksums.

   PROM 1 CHECKSUM = $AAAA
   PROM 2 CHECKSUM = $0033
   PROM 3 CHECKSUM = $F0DD

Key Entry	Display	Explanation
CKSUM ENTER	CKSUM CK PROM?	
STEP **or CLEAR**	**CK RAM ?**	STEP or CLEAR keys toggle between PROM and RAM.
ENTER	**CK BUSY** **C.K.=.1.9.B.B.5.**	The set checksum "NNNNN" is displayed. The block of Buffer RAM checksummed is $00000 to $02FFF. Return to monitor.

The Calculation is $19BB5 = $AAAB + $0031 + $F0D9

EXAMPLE: Change of set checksum with a change of master PROM sequence.

Relevant conditions:

1. A TYPE has been defined.

2. A SET size of 3 has been chosen.

(1) Sockets 1, 2 and 3 are loaded with master PROMs.

Key Entry	Display	Explanation
MOVE	MOVE	
ENTER	C.K.0.2.A.6.3..	Set checksum is 02A63. Return to monitor.

(2) The PROMs in Sockets 1 and 2 are switched.

MOVE	MOVE	
ENTER	C.K.0.2.A.6.1..	Set checksum is 02A61. PROMs inserted in wrong set order change the set checksum. Return to monitor.

4.3 CLEAR

Used to clear last entry, choose between RAM or PROM function, or decrement displayed socket number.

4.4 ENTER

Used to execute presently displayed command.

4.5 EXIT

Stops any command cycle in progress and returns to monitor. When entered during a command definition, EXIT will cancel the command being defined.

4.6 FUNC (Default = BPVC)

The FUNC command defines the RUN Mode Command sequence which is any combination of BLANKCHECK (BLANK), PROGRAM (PROG), VERIFY, or CHECKSUM (CKSUM).

EXAMPLES: Defining RUN Mode sequences with the FUNC key.

Key Entry	Display	Explanation
FUNC	FUNC	
ENTER	F=XXXX	Previously defined RUN Mode sequence "XXXX" is displayed.
PROG	F=PV	Entry of a PROG key in the RUN Mode sequence definition will automatically add the VERIFY command. If after programming, a VERIFY is not desired, enter a CLEAR key.
CLEAR	F=P	
ENTER	F.=.P.....	Return to monitor.
FUNC	FUNC	
ENTER	F=P	
BLANK	F=B	BLANK.
PROG	F=BPV	BLANK, PROG, VERIFY.
CKSUM	F=BPVC	BLANK, PROG, VERIFY, CKSUM.
ENTER	F.=.B.P.V.C...	BLANK, PROG, VERIFY and CKSUM commands are now defined for the RUN Mode. Return to monitor.

4.7 LIST

There are two operations possible with the LIST key. One allows for listing a PROM in a chosen socket along with the data in the Buffer RAM associated with that socket. The second option lists the data in the Buffer RAM. When used to display the data in a PROM, a verify can be performed between Buffer RAM and the selected PROM.

Listing PROM data:

After a socket is chosen, the address relative to the PROM followed by the RAM data and finally the PROM data is displayed. The user may slew at different rates forward or backward by use of the STEP and CLEAR keys. STEP sets the slew direction UP, CLEAR sets the direction DOWN. The slew rate is changed by entry of any of the number keys 1 to 5. The ENTER key is used to stop slewing and single step in either direction. The EXIT key is used to exit the function at any time.

The LIST PROM function may also be used to verify a PROM against Buffer RAM. In order to observe differences between a PROM and the Buffer RAM image, a slew rate of 5 should be entered.

EXAMPLE: Listing a PROM

Relevant conditions:

1. A TYPE has been defined.

2. Assume a PROM is in Socket 2.

Key Entry	Display	Explanation
LIST	LIST	
ENTER	L PROM ?	
STEP or CLEAR	L RAM ?	STEP of CLEAR keys toggle between PROM and RAM.
STEP	L PROM ?	
ENTER	SOC= 1 ?	
STEP	SOC= 2 ?	STEP key increments 1.
STEP	SOC= 3 ?	STEP key increments 1.
CLEAR	SOC= 2 ?	CLEAR key decrements 1.
ENTER	AAAARRPP	Reminder message for display format. AAAA = PROM address, RR = Buffer RAM data, and PP = PROM data.
ENTER	0000RRPP	A Load Sequence is run on the sockets. Address 0000 is displayed along with Buffer RAM data RR and socket data PP.
ENTER	0001RRPP	Address is incremented by 1.
2	AA BLURR	Address slewed upwards.
5	01234567	Slew stopped at address 0123 where Buffer RAM data 45 is different from PROM data 67.
5	0156789A	Data of Buffer RAM and PROM between locations 0123 and 0156 match. Display shows mismatch at location 0156.
ENTER	01578888	Address and data displayed.
CLEAR	0156789A	Address is decremented by 1.
1	AA BLURR	Address is slewed downwards.
EXIT	I.M...3.0.1.6.	End function. Return to monitor.

Listing Buffer RAM data:

The IM3016 Buffer RAM appears to the user as having 8-bit data expressed as two hexadecimal digits in a 20-bit address space expressed as five hexadecimal digits. The LIST RAM function lists the five digit address and the two digit data of the buffer RAM. Slew rates are defined with the number keys 1 to 5 and the direction with STEP(up) and CLEAR(down). Initial entry into the LIST RAM mode is by a toggle using the STEP or CLEAR keys as shown in the following example.

EXAMPLE: Listing Buffer RAM

Key Entry	Display	Explanation
LIST	LIST	
ENTER	L PROM ?	
STEP or CLEAR	L RAM ?	STEP or CLEAR keys toggle between PROM and RAM.
ENTER	00000 12	Address 00000 and data 12 displayed.
ENTER 3	00001 34	Address incremented by 1. Address slewed upwards.
	AA BLUR	(Slew speed controlled by keys 0, 1, 2, 3, 4, 5)
ENTER	01234 56	Address and data displayed.
CLEAR	01233 78	Address decremented by 1.
4	AA BLUR	Address slewed downwards.
EXIT	I.M...3.0.1.6.	End function. Return to monitor.

4.8 MOVE

MOVE is used to load the Buffer RAM with the appropriate socket data associated with the desired set. At the end of the MOVE command, the SET checksum is displayed.

The socket data is moved starting with Socket 1 into Buffer RAM from where Socket 1 has been mapped (usually $00000) in a contiguous block whose size is determined by the size of the PROM TYPE times the SET size.

A Load Sequence occurs before the data in the PROM(s) is transferred to the Buffer RAM. A Load Set Error will occur if the correct number of PROMs are not present.

After the data has been moved into RAM, a Verify is performed. If an error is detected, a MOVE ERR message is displayed. This error will result if there are more PROMs in the sockets than specified in the SET size or if there is a malfunction in the IM3016 RAM. If no Verify error occurs, the SET Checksum is displayed.

EXAMPLE: Moving a PROM set into Buffer RAM

Relevant conditions:

1. A TYPE has been defined.

2. A SET size of 4 has been chosen.

3. Four master PROMs are loaded into the module sockets.

Key Entry	Display	Explanation
MOVE	MOVE	
ENTER	C.K.=.N.N.N.N.N.	Master set moved to Buffer RAM and the set checksum is displayed. Return to monitor.

The set checksum displayed upon completion of the MOVE command is the set checksum of the Buffer RAM. This value is NOT the simple sum of the individual Buffer RAM blocks that make up the given set. The individual socket checksums are modified so that a different set checksum is observed anytime incorrect sequencing of the master PROMs has occurred. See the definition of CHECKSUM for further details.

4.9 PROG

The PROG command is used to program data from the Buffer RAM into a loaded socket. The sequence of programming operations is as follows:

1. Load Sequence tests.

2. Display reads P WAIT.

3. A special test for program voltage (Vp) breakdown is executed.

4. Assuming the PROM TYPE programming algorithm allows individual location programming, the first 32 locations are programmed and verified. If the first 32 locations verify, the display shows the addresses being programmed until done.

5. Upon completion of the programming cycle, a verify cycle is executed.

If any activated socket fails during steps 2 to 4, the following actions are taken:

1. The display reads BAD VP for step 3 above or
2. The display reads BAD VER for step 4 above.
3. Red LEDs are on over the failed sockets.
4. The Load Error and START lights are turned on.

The user should now remove all failed parts and optionally reload the sockets. The program cycle will be re-initiated starting from step 1 (Load Sequence) after depression of the START or ENTER key.

4.10 SET (Default = 1)

SET defines the size of the PROM set to be programmed. Valid entries are 1 through 16. The default value is 1. During the definition of the SET size, the STEP and CLEAR keys are used to increment or decrement to the desired value. The SET function maps the module sockets to the Buffer RAM as explained in the definition of the MOVE key.

EXAMPLE: The SET Command

Key Entry	Display	Explanation
SET	SET	
ENTER	SET=16 ?	Displays previously defined Set size.
CLEAR	SET=15 ?	Decrement 1 place value.
CLEAR	SET=14 ?	Decrement 1 place value.
ENTER	S.E.T.=.1.4...	SET size of 14 is defined. Return to monitor.

4.11 SPEC

The SPEC key (special) is used to perform special I/O functions such as loading Intellec Hex data from the serial port. SPEC functions are defined by entry of the number keys 0 to 5, up to 3 digits long. See Section 7 for more information.

4.12 STEP

The STEP key is used by various commands to increment values or to toggle between two choices. See LIST, for example.

4.13 TEST

The TEST key is used to initiate IM3016 hardware tests. See Section 8 on TEST FUNCTIONS.

4.14 TYPE (Default = 0)

Define the TYPE of PROM to be loaded into the module PROM sockets. Each PROM TYPE is defined by a three-digit number which is referred to below as the "PROM Conversion Number". Entry of no numbers will display the previously defined PROM TYPE. A PROM TYPE must be defined before any other command, except the TEST and SPEC 2 (Computer Control) Commands. Appendix A contains a list of all currently defined PROM TYPEs.

Entry of a new TYPE will automatically map socket 1 to RAM address $00000.

Note: EPROM manufacturers have approved the use of a fast algorithm for programming the larger EPROMs (i.e., 2764,27128, and 27256). However, the IM3016 can program the smaller EPROMs (2732, 2716, etc.) with the fast algorithm as well. The fast algorithm is not an approved algorithm for the smaller parts and should **only be used in an engineering or development application.**To use the fast algorithm for the smaller parts, change the first digit of the type code from a 1 to a 0.

EXAMPLE

EPROM 2732
TYPE=112 Approved algorithm
TYPE=012 FAST ALGORITHM

PROM Conversion Number	PROM TYPE
110	2758
111	2716(1V),2516
112	2732
113	2732A
114	2532
120	68764
121	2764
122	2564
124	FAST2764
131	27128
132	27256
133	2764A
134	27128A
310	2704
311	2708
312	TMS2716(3V)

EXAMPLE: The TYPE Command.

Key Entry	Display	Explanation
TYPE	TYPE	
ENTER	TYPE=114	Displays previously defined PROM TYPE.
122	TYPE=122	PROM TYPE Conversion Number 122 defined.
ENTER	T.=...2.5.6.4.	TYPE 122 identified as 2564. Return to monitor.

4.15 VERIFY

VERIFY compares data read from PROMs in the sockets to the data in Buffer RAM. Before the verification takes place, a Load Sequence is initiated. At the completion of the VERIFY command, the number of sockets that have both passed and failed are displayed. Note that this key can also be used in conjunction with the FUNC key.

The VERIFY function does not allow viewing any data discrepancies between Buffer RAM and socketed PROMs. However, this can be done using the LIST key with a slew rate of 5 (see Section 4.7).

EXAMPLE: The VERIFY Command

Relevant conditions:

1. A TYPE has been defined.
2. The Buffer RAM must have been correctly loaded with a MOVE command or SPEC load command.
3. Only active sockets are verified and the blocks of Buffer RAM used for comparison are those defined by the TYPE and SET keys.

Key Entry	Display	Explanation
VERIFY	VERIFY	
ENTER	V BUSY 1	Verifying active socket (PROM) data to Buffer Ram data.
	V.P.=.3.X.=.0..	3 pass, 0 fail. Green/Red LED light up over appropriate pass/fail sockets. Return to monitor.

SECTION 5: RUN MODE COMMAND DEFINITIONS

5.0 GENERAL

RUN Mode Commands are used when the operations to be performed on the socketed PROMs are set operations. RUN Mode Command sequences are created using the FUNC key and are executed after the RUN key is entered. Although the most common use of the RUN Mode is to program sets of PROMs, other possible set operations are blank checking sets, verifying sets and checksumming sets.

5.1 LOAD SEQUENCE IN RUN MODE

As with Immediate Mode Commands, the RUN Mode Commands which enable the PROMs in the module sockets are preceded with a Load Sequence. If an error condition exists at any socket, operation is temporarily halted until the condition is fixed and the START or ENTER key is depressed again. In addition to the normal Load Sequence tests on the socketed PROMs such as VCC errors and Open Pins, the Load Sequence in the RUN Mode requires that complete sets of PROMs be activated. If such is not the case, the following actions will take place:

1. The display reads LSET ERR.
2. A red LED appears over the socket creating the incomplete set.
3. The Load Error light goes on.

The user should correct the situation and re-initiate another Load Sequence by depression of the START key. An example of this is demonstrated in the BLANK command given below.

5.2 BLANK IN RUN MODE

PROMs in the module sockets are tested for erased condition.

EXAMPLE: BLANK function with Load Set Error

Relevant conditions:

1. RUN Mode sequence is BLANK as previously defined with FUNC key (F=B).
2. SET size = 3.

(1) User places 15 supposedly blank PROMs into Sockets 1 to 15 (5 sets).

Key Entry	Display	Explanation
RUN	RUN	
ENTER	R.U.N.....	RUN Mode has been entered. START light is on.
START		Load Sequence begins.
	VCC ERR	Red LED is on over Socket 3. Load Error light is on.

(2) User notices Socket 3 has an upside down PROM. He removes this device.

START		Load Sequence begins.
	LSET ERR	A Load Set Error has occurred because Socket 3 is empty. A red LED is on over Socket 3.

(3) User places a blank PROM in Socket 3.

START		Load Sequence begins. All devices pass Load Sequence and Load Set Test.
	B BUSY	Activated Sockets 1 to 15 are tested for blank condition.
	B.P.=.1.5.X=.0..	15 PROMs are blank. 0 PROMs not erased.

5.3 CKSUM

When executed as part of the RUN Mode sequence, CKSUM will result in the display of the set checksum of the Buffer RAM block defined by the chosen TYPE and SET. The use of CKSUM in the RUN Mode is to provide the operator with a check on the validity of the Buffer RAM and the previously defined TYPE and SET values.

5.4 PROG

When executed as part of the RUN Mode sequence, PROG will result in the programming of complete sets of socketed PROMs. As in the Immediate Mode, a programming cycle is preceded with a Load Sequence. However, the additional test of whether complete sets of active sockets are present is also applied. An example of what happens when this is not the case (i.e., a Load Set Error exists) is given in this section under BLANK.

5.5 RUN

The RUN key is used to enter into a RUN Mode sequence defined by the FUNC key. When in the RUN Mode, operations can only occur on complete sets of active sockets. If this is not the case, a Load Set Error condition exists (see BLANK in this section). Normally, after the RUN Mode has been entered, the keyboard door is closed leaving the START key as the only visible key. When the IM3016 has finished a prescribed function and is waiting for a key entry, the START key light will be on. The IM3016 will only recognize two other keys in the RUN Mode; the ENTER key which is interpreted as a START key depression, and the EXIT key. Depression of the EXIT key while in the RUN Mode will normally cause exit from the RUN Mode and will allow the user to enter Immediate Mode Commands. If, however, an EXIT key is entered when a program operation has resulted in a display of "BAD VP", the operator is again requested to reload the failed sockets with a display of "NO RELOD". This sequence takes place because it is likely that all PROMs in the module sockets have been partially programmed. However, a second entry of the the EXIT key will result in exit from the RUN Mode.

EXAMPLE: Use of EXIT key during RUN Mode.

Relevant conditions:

1. **RUN Mode sequence is BLANK, PROG, VERIFY (i.e. F=BPV).**
2. **SET size = 3.**
3. **User loads Sockets 1 to 15 with blank PROMs at beginning of each cycle.**

Key Entry	Display	Explanation
RUN	RUN	
ENTER	R.U.N.....	START light goes on.

(1) User closes keyboard door and loads 15 blank PROMs into sockets.

START	B BUSY	PROMs are blank checked.
	P WAIT	PROMs are partially programmed and verified.
	BUSYAAAA	PROMs are programmed.
	V BUSY	PROMs are verified.
	V.P.=.1.5.X.=.0.	15 PROMs (5 sets) are programmed correctly. START light is on.

(2) User decides to create different PROM sets and must exit from RUN Mode to do this.

(3) User opens keyboard door.

EXIT	I.M.3.0.1.6...	Return to monitor.

(4) User changes data in Buffer RAM by loading a different set with the MOVE command. He then re-enters RUN Mode with RUN ENTER, closes the keyboard door and reloads 15 new PROMs into Sockets 1 to 15.

START	B BUSY	PROMs are blank checked.
	P WAIT	PROMs are partially programmed and verified.
	BAD VP	Red LED is on over Socket 3. User replaces this PROM with a new one.
START	P WAIT	PROMs are partially programmed and verified.
	BUSYAAAA	PROMs are programmed.
	V BUSY	PROMs are verified.
	V.P.=.1.5.X.=.0.	15 PROMs (5 sets) programmed correctly. START light is on.

(5) User removes programmed PROMs and replaces these with 15 new PROMs.

START	B BUSY	PROMs are blank checked.
	P WAIT	PROMs are partially programmed and verified.
	BAD VER	Red LEDs are on over Sockets 1 to 15.

(6) User realizes that he has loaded 15 known bad parts into the IM3016. He decides to exit from the RUN Mode.

EXIT	NO RELOD	Display signifies that all PROMs may be partially programmed.
EXIT	I.M.3.0.1.6...	Return to monitor.

5.6 START

The START button duplicates the ENTER key in function. That is, it initiates the desired command. It is a lighted push button and is the only key visible when the keyboard door is closed. Thus, START provides for truly one key operation when the IM3016 is placed in the RUN Mode.

5.7 VERIFY

Socketed PROMs are verified against Buffer RAM data. A Load Set Error will result if there are incomplete sets of active PROMs in the module sockets before verification takes place (see BLANK in this section). If VERIFY is the last operation in the RUN Mode sequence, both the number of PROMs that have been found correct and incorrect will be displayed.

SECTION 6: SAMPLE PROGRAMMING SESSION

6.0 DEFINING THE RUN MODE PROGRAMMING SEQUENCE

The proper commands must be entered in order to program sets of PROMs. They are listed below in the probable order of their execution.

1. **TYPE:** PROM TYPE must be identified. See PROM TYPE Conversion Table. (Appendix A)

2. **SET:** PROM SET size must be defined.

3. **FUNC:** The RUN Mode command sequence must be defined by any combination of BLANK, PROG, VERIFY, and CKSUM commands.

4. **MOVE:** Master PROM set images copied into Buffer RAM.

4a. (Alternate) Load master file from computer.

5. **RUN:** Places the IM3016 in the RUN Mode. The Start light is on.

6. **START:** Depression of the START button will begin a RUN Mode program cycle.

6.1 SAMPLE PROGRAMMING SESSION

EXAMPLE: Sample Programming Session

Relevant Conditions:

1. TYPE = 2532.

2. SET size = 5.

3. Total desired # of sets = 5 (25 PROMs total)

(1) Defining the TYPE of PROM.

Key Entry	Display	Explanation
TYPE	TYPE	
ENTER	TYPE=NNN	NNN = previously defined TYPE if any.
114	TYPE=114	Define TYPE of PROM.
ENTER	T.=...2.5.3.2.	PROM TYPE 114 identified as 2532.

(2) Choosing the SET size.

SET	SET	
ENTER	SET= 6 ?	Previously defined SET size displayed.
CLEAR	SET= 5 ?	CLEAR decrements value to desired SET size.
ENTER	S.E.T.=.5.	SET size of 5 defined.

(3) Defining the RUN Mode sequence.

FUNC	FUNC	
ENTER	F=XXXX	XXXX = previous RUN Mode Command sequence.
BLANK	F=B	BLANK.
PROG	F=BPV	BLANK, PROG, VERIFY.
CKSUM	F=BPVC	BLANK, PROG, VERIFY, CKSUM.
ENTER	F.=.B.P.V.C...	RUN cycle defined for BLANK, PROG, VERIFY, CKSUM commands.

(4) The set Master PROMs are inserted into the module sockets and their data is transferred to the Buffer Ram.

MOVE	MOVE	
ENTER	C.K.=.N.N.N.N.N.	Master PROMs have been moved into Buffer RAM. The SET checksum is displayed.

(5) The Run Mode is entered.

RUN	RUN	
ENTER	R.U.N.....	START button lights. Previously defined RUN Mode Sequence is ready for execution.

(6) Master PROMs are removed and replaced by 15 blank PROMs to create 3 sets of 5 PROMs per set. The door is shut over keyboard.

(7) The RUN Mode sequence is initiated.

START	B BUSY	Load Sequence test, followed by blankcheck of socketed PROMs.
	P WAIT	The first 32 locations are programmed and verified. Any programming errors that occur will cause a reload sequence to begin.
	BUSYNNNN	"NNNN" represents the address locations as they are programmed.
	C.K.=.N.N.N.N.N.	The SET checksum is displayed at the end of program cycle. LEDs over PROM sockets indicate pass/fail. START button is on. Ready to execute next RUN Mode cycle.

(8) 3 sets of 5 PROMs each are removed and replaced by 10 blank PROMs (2 sets of 5 PROMs each).

START	B BUSY	Load test sequence is executed, then PROMs are checked for erased condition.
	P WAIT	First 32 locations are programmed and verified, program cycle continues.
	BUSYNNNN	"NNNN" represents the address locations as they are programmed.
	C.K.=.N.N.N.N.N.	The Set checksum is displayed at the end of programming cycle. LEDs over PROM sockets indicate pass/fail.

(9) 25 PROMs have been programmed and user wishes to leave the RUN Mode.

EXIT	I.M...3.0.1.6.	Return to monitor.

Suppose that during the above sequence two PROMs failed to program correctly; for example, additional copies of Master PROMs 1 and 4 are needed. Since the RUN Mode only operates on sets and a complete set is not needed, the user can use the Immediate Mode Commands to solve this problem. This is shown as in the following example.

EXAMPLE: Additional copies of Master PROMs are needed.

(1) User places two blank PROMs in sockets; one in Socket 1 and one in Socket 4.

Key Entry	Display	Explanation
BLANK		
ENTER	B BUSY	
	B..P.=.2.X.=.0.	Both PROMs are blank.
PROG		
ENTER	P WAIT	PROMs 1 and 4 are programmed.
	BUSYNNNN	
	V BUSY	PROMs are verified.
	V..P.=.2.X.=.0.	Both PROMs are programmed and verified.

The ability to deal with PROMs both as sets in the RUN Mode and as individual entities in the Immediate Mode is a uniquely powerful characteristic of the IM3016.

SECTION 7: INTERFACING THE IM3016

7.0 FORMATTED I/O

A number of formatted I/O programs are incorporated into the IM3016 to allow the loading and dumping of PROM data via the RS232C port on the rear of the IM3016. In addition, because the software of the IM3016 is written in an easily changed high level language, International Microsystems welcomes requests for formats not presently provided. Also, other types of interfaces besides RS232C are possible; notably, current loop and TTL parallel interfacing. Please consult the factory for further assistance.

7.1 DEFINED SPEC FUNCTIONS

Included with every IM3016 are SPEC functions 1, 2, 3, 4, and 5, and various formatted load and dump programs. The table below gives the special functions available with the present IM3016 software revision.

IM3016 Special Functions

SPEC #	
1	Program Time
2	Terminal/Computer Control
3	8 or 16-Bit Data
4	Fill Buffer RAM with either FF's or 00's
5	Map Socket 1 to Arbitrary Buffer RAM Address
10	Intellec Hex Load
11	Intellec Hex Dump
12	Motorola S Record Load
13	Motorola S Record Dump
14	TEK HEX Load
15	TEK HEX Dump
20	Rockwell Dev Load
21	Rockwell Dev Dump
22	TI Dev Load
24	Binary Load
25	Binary Dump
30	HP Binary Load
32	LDA Load
34	RCA COSMAC Load
35	RCA COSMAC Dump
42	Load AXCII Text

7.2 SPEC 1: PROGRAM TIME

SPEC 1 allows the user to change the default programming pulse width. This may be desireable for one of two reasons. In some instances, a given manufacturer may have more than one programming time for the same EPROM TYPE. For example, some 2732A's can be programmed with 20ms pulses rather than the typical 50ms pulse. Hence, being able to lower the program pulse length from 50ms to 20ms will result in approximately two and a half times faster program cycle. Such variations in programming times are becoming common with PROMs that are 4kx8 or smaller in size. The decreased times have resulted from the improved semiconductor processing methods which have produced smaller geometries and lower leakage program cells. Another possible reason one may wish to lower the program pulse width is to speed up the time wasted during program development on programming PROMs which are to be used only in testing. Standard programming times are determined so that the charge retention of the EPROM cell is guaranteed for virtually forever. When one lowers the programming pulse width, the charge stored in the EPROM cell may be reduced and hence the charge retention lowered. Thus the price paid for lowering the program time may be poor data retention by the EPROM and hence failure in operation. Therefore, the user is cautioned NEVER to lower the standard programming time on PROMs used in shippable product without first making sure that the program time is approved by the manufacturer of the EPROM.

The program pulse width may be varied between 0.1 ms and 50.9ms using SPEC 1. However, the user is cautioned that do to software and hardware constraints of the IM3016, the lower limit of most program pulses is normally about 2ms for 4kx8 and smaller EPROMs and 0.9 ms for 8kx8 or larger EPROMs. The user should use an oscilloscope and actually observe the program pulse width for any particular PROM that he lowers the pulse more than 5ms. This in some cases will require looking at both the VP line and a PGM or programming pulse line.

The values set in SPEC 1 are reset to the default value upon power up and whenever the TYPE of PROM is specified. The STEP and CLEAR keys are used to increase or decrease the display pulse width value. The user is prompted for two values; namely, PT1 which is the program time in milliseconds (0-50) and PT2 which is the program time in 0.1 milliseconds (0-9). The actual program time is the sum of PT1 and PT2.

7.3 SPEC 2: TERMINAL/COMPUTER CONTROL

SPEC 2 allows the user to enter the external control mode and operate the IM3016 with a terminal or computer. Also, external control may be entered on power-up or hardware reset if position 5 of the eight position dip switch on the rear of the IM3016 is in the ON position. External control allows many additional operations to be performed, including filtering and offsetting formatted data files during a download to the IM3016. However, the reader is advised to learn the commands available from the IM3016 keyboard before proceeding to external control.

7.4 SPEC 3: 8 OR 16 BIT DATA

Some I/O formats such as the Intel Hex or the TI Development System can be used for both 8-bit and 16-bit data words. In contrast, the data width of most EPROMs and the IM3016 RAM is 8-bits wide. Software in the IM3016 provides the means for separating 16-bit data in such a way as to allow the correct PROM images to be created. SPEC 3 lets the user choose between 8-bit and 16-bit data. The default format is 8-bit, and when this is chosen, the sockets are mapped and data loaded in the expected way. However, when a data format of 16-bits is chosen, both the method of mapping the sockets to the IM3016 RAM and certain I/O formats are affected. Fortunately, the use of SPEC 3 to choose 8-bit or 16-bit data is relatively straightforward.

Example: User wishes to load 32K x 16-bits of data using the TI Development Loader.

KEY ENTRY	DISPLAY	COMMENT
SPEC ENTER	SPEC= ?	Choose 16-bit data
3 ENTER	DATA= 8?	Show effect of STEP
STEP	DATA=16?	
ENTER	DATA=16	
SPEC ENTER	SPEC= ?	Choose TI DEV load format
22 ENTER	TEXAS IN	
ENTER	ADD=	
	ADD=1234	File is being loaded
	DONEFFFF	End of load at FFFFH

To summarize: to load 16-bit data, the user must execute the SPEC 3 function before entering the load format. Upon power up, the IM3016 will default to 8-bit data format and contiguous mapping.

Multiple Sets of 16-Bit Data

Multiple sets of 16-bit data are possible in the same manner as with 8-bit data. Assume that SPEC 3 has been executed for 16-bit data and the IM3016 RAM has been loaded as desired. Also assume that the set size is 4 and the type of PROM is a 2764 (2000H). The socket mapping is given below.

Socket#	1	2	3	4
St Add	0000	8000	2000H	A000H
Byte H/L	H	L	H	L

Sockets 5,9,13 are the same as 1.
Sockets 6,10,14 are the same as 2.
Sockets 7,11,15 are the same as 3.
Sockets 8,12,16,are the same as 4.

The above mapping is the logical extension of that used for 8-bit data. Thus, with a set size of 4, four sets of 4 PROMs can be created at one time regardless of the data width.

Detailed Explanation of 16-Bit Mapping

Before an example is given on how 16-bit data is split in the IM3016, a note on 16-bit processors is in order. CPU's such as the Intel 8086, DEC 11's, and the TI 9900's address memory on byte boundries even though two bytes are read during one memory fetch. In effect, memory may be viewed in either 32K x 16-bit blocks or 64K x 8-bit blocks depending on the function being performed. Thus, when loading 16-bit formatted data, the IM3016 RAM buffer can be viewed as having blocks of 32K x 16-bits of memory. Formatted data being loaded into the IM3016 appears as a stream of bytes which must be separated into high and low blocks of data. The PROM sockets are remapped such that each pair of sockets make up 16-bit wide data blocks. An example of how this is actually done is given below.

Example: Splitting up byte data for 16-bit CPU's.

Assume that SPEC 3 has been executed and that 16-bit data was chosen. Also assume that the actual formatted load file (e.g. Intel Hex) specifies that four bytes of data are at the following locations:

Address	Data
$F0000	$00
$F0001	$01
$F0002	$02
$F0003	$03

First, the IM3016 determines which 64K RAM bank it will place the data into by examining the bank address (in this case $F) and truncating its value to the maximum available RAM in the IM3016. Thus, if the IM3016 has 128K x 8 of RAM, $F will be truncated to $1, and the data will be placed in the second IM3016 RAM bank.

Next, the data is split into odd and even bytes with the odd bytes going to the bottom half of the 64K bank and the even bytes going to the top half of the 64K bank.

Address	IM3016 Add	Even	Odd
$F0000	$18000	00	
$F0001	$10000		01
$F0002	$18001	02	
$F0003	$10001		03

Mathematically, the formatted 64K in bank address is shifted logically to the right one bit, the carry is complemented and made the MSB of the resulting address. The choice to complement the carry was made to accomodate the Intel convention of even bytes being the low byte.

The above process explains how the data is loaded into the IM3016 RAM. The corresponding mapping of the sockets after the SPEC 3 choice of 16-bit data is as follows: Assume that the Set size is 16, Socket 1 is mapped to $10000 and the PROM size is $1000. Then Socket 2 is mapped to $18000, Socket 3 is mapped to $11000, Socket 4 is mapped to $19000, etc. In terms of our example,

Socket 1		Socket 2	
Address	Data	Address	Data
0000	01	0000	00
0001	03	0001	02

Although all of this manipulation seems rather confusing, the net result is simple. Namely, when 16-bit data is being loaded with an appropriate format, odd and even bytes are separated in 64K blocks and the sockets are mapped using one command, SPEC 3.

7.5 SPEC 4: CLEARING THE BUFFER RAM BEFORE AN OBJECT LOAD

Many times an object load file will not completely fill the address space that data is loaded into. To allow reprogramming the unused portions at a later time, it is necessary to first fill the Buffer RAM with the erased condition of the PROM. Also, it is sometimes desirable to fill unused locations with 0's which is generally the NOP instruction for microcomputers. The SPEC 4 command provides this function. On power up, all locations are filled with 0's.

EXAMPLE: Filling the Buffer RAM with 1's or 0's.

Relevant conditions:

1. A TYPE has been defined.

Key Entry	Display	Explanation
SPEC	SPEC	
ENTER	SPEC= ?	
4	SPEC= 4?	Choose function SPEC 4.
ENTER	FILL=FF?	Fill Buffer RAM with 1's ?
STEP	FILL=00?	Fill Buffer RAM with 0's ?
ENTER	FIL BUSY	Yes.
	D.O.N.E..F.I.L.	Return to monitor.

7.6 SPEC 5: MAPPING SOCKET 1 TO ARBITRARY START ADDRESS

SPEC 5 provides the user with a means of mapping Socket 1 to any 1K hex block of IM3016 RAM. However care must be exercised if SPEC 3 has been used to set the data format to 16 as in this case any 64K address for Socket 1 equal to or greater than $8000 will produce strange results (see Section 7.3 SPEC 3) . Also SPEC 5 can be used to set the start address for a formatted dump to any 1K hex boundry in the IM3016 RAM buffer (see Section 7.11). Whenever the TYPE of PROM is changed, Socket 1 is remapped to $00000!.

Example: User has data in IM3016 RAM starting at 19000H and desires to program four 2716's.

KEY ENTRY	DISPLAY	COMMENTS
SPEC ENTER	SPEC= ?	Choose SPEC 5
5 ENTER	S1AD= 0?	Map Socket 1 to 0000H?
STEP	S1AD= 1?	Increment to desired
"		value
STEP	S1AD= 9?	Map Socket 1 to 9000H?
ENTER	S1BK= 0?	Socket 1 mapped to 9000H in Bank 0 of RAM
STEP	S1BK= 1?	Socket 1 mapped to 19000H in RAM
ENTER	S.1.B.K.=..1..	Done

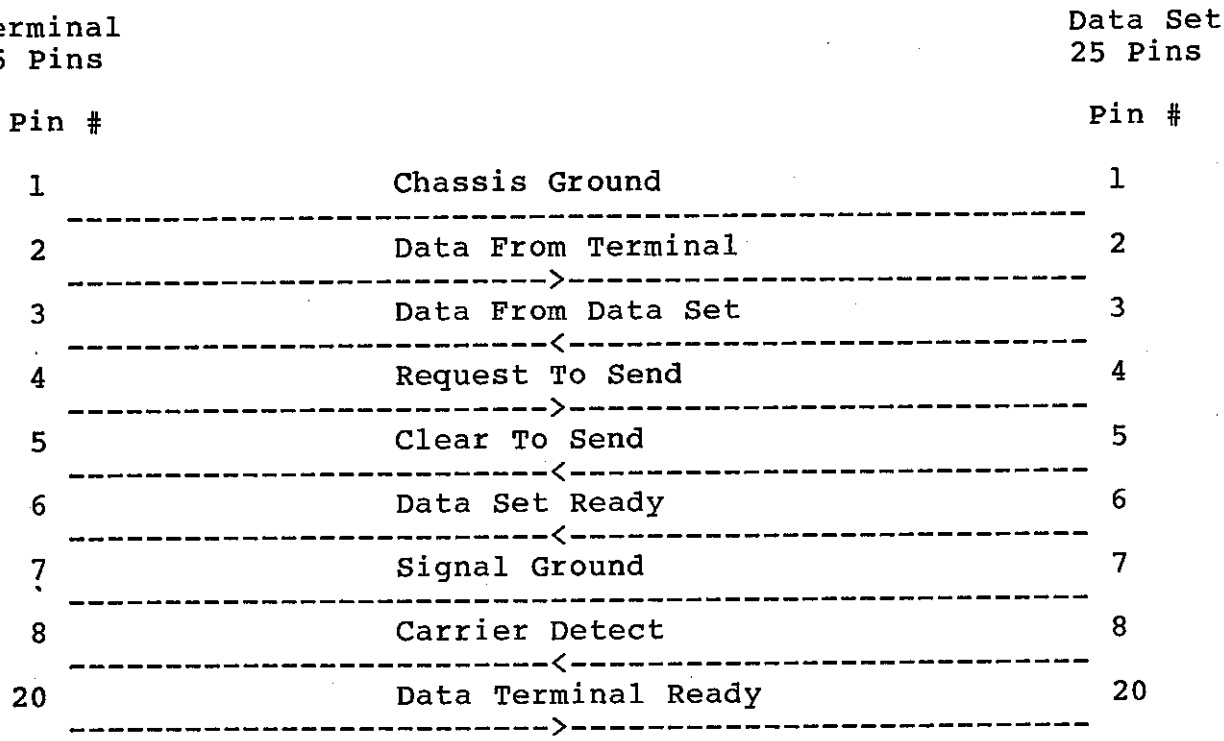
Example: Socket Map With 2716's, Set Size = 4, Socket 1 =19000H.

Socket #	8-Bit Data Bank	Address
1	1	9000H
2	1	9800H
3	1	A000H
4	1	A800H
5,9,13	same as Socket 1	
6,10,14	same as Socket 2	
7,11,15	same as Socket 3	
8,12,16	same as Socket 4	

The default mapping of Socket 1 on power up is 0000H.

7.7 STANDARD RS232C PIN ASSIGNMENTS

Due to the confusion surrounding the RS232C standard, a short review is applicable. The RS232C standard recognizes two types of devices: a terminal and a data set. The figure below shows a typical connection. A brief description of these lines and their use follows.



Grounds:

Pin 1: Chassis Ground

Pin 7: Signal Ground

Data Transmission:

Pin 2: Data from Terminal

Pin 3: Data from Data Set

On Line Status:

Pin 20: The Data Terminal Ready (DTR) line is pulled active by the terminal (> + 3V) to indicate that it is on line.

Pin 6: The Data Set Ready (DSR) line is pulled active by the data set to indicate that it is on line.

Pin 8: The Carrier Detect Line (DCD) is pulled active by the data set to indicate that a carrier signal is available.

Hand Shake Status:

Pin 4: The Request To Send (RTS) line is pulled active by the terminal to indicate it is ready to send data to the data set.

Pin 5: The Clear To Send (CTS) line is pulled active by the data set to signal the terminal that the terminal may initiate a data transfer. This line may only go high in response to an active RTS from the terminal.

The ground lines (Pin 1 and 7) should always be connected. The data lines (Pin 2 and 3) also must be connected for bi-directional data transfers. The status lines may or may not be needed depending on the peripheral device.

7.8 IM3016 RS232C PORT CONNECTIONS

Pin #	Name	Direction
1	Ground	------->
2	IM3016 Data In	<-------
3	IM3016 Data Out	------->
4	$\overline{\text{RTS}}$	<-------
5	$\overline{\text{CTS}}$	------->
6	$\overline{\text{DSR}}$	------->
7	Ground	------->
8	$\overline{\text{CDC}}$	------->
20	$\overline{\text{DTR}}$	ignored

NOTES:

1. The truth levels and corresponding voltages of the RS232C lines are:

   1 (true) = +3V to +15V

   0 (false) = -3V to -15V

2. The bars occur over the RTS, CTS, DSR, and DTR lines above for the following reason. These signals are originated or terminated at a TTL level device. The RS232C TTL level shifter IC's used in the IM3016 are inverters. Thus, a true RS232C level (> +3V) will result in a 0V TTL signal.

3. The IM3016 RS232C port generally operates in half duplex. Thus upon receiving a character, no return character is transmitted. Also, because request-to-send : clear-to-send protocol is supported, the user should pull this line to a 1 level by some means such as connecting pins 4 and 5 together.

4. The serial encoding formats supported are seven level ASCII data and 8-bit binary. No special ASCII codes are assigned that are unusable or result in special operations.

5. For 3 wire connections without hardware status, it is recommended that the user tie 1 and 7, 4 and 6 together and leave pin 5 open on the IM3016.

7.9 The IM3016 RS232C SWITCH

The eight position switch on the rear of the IM3016 is used to set various RS232C parameters such as the baud rate, parity, etc. Also, switch position 5 may be set to enable the IM3016 to be put directly into external control upon receiving an upper case "C" through the RS232 port. The assignments are given below and in Appendix B.

DEFINITION OF SWITCH 1 (SW 1)

X = ON
O = OFF

BAUD RATE	POS 8	POS 7	POS 6
110	X	X	X
150	O	X	X
300	X	O	X
1200	O	O	X
2400	X	X	O
4800	O	X	O
9600	X	O	O
19.2K	O	O	O

POSITION	5	4	3	2	1
USE	POWER ON MODE	PARITY ENABLE	PARITY	DATA BITS	STOP BITS
OFF	NORMAL	ON	EVEN	8 BITS	2 STOP
ON	EXT CNTL	OFF	ODD	7 BITS	1 STOP

COMMON SWITCH SETTINGS

EXAMPLE: 1200 BAUD RS232C, EVEN PARITY, 7 BITS (ASCII), 2 STOPS.

Switch posistions 2 and 6 are on. All others off.

EXAMPLE: 9600 BAUD RS232C, NO PARITY, 7 BITS (ASCII), 1 STOP.

Switch positions 1, 2, 4, and 8 are on. All others off.

7.10 GETTING THE RS232C PORT UP AND RUNNING

Because interfacing a sophisticated device such as the IM3016 to a computer or other instrument is often time consuming, a special test package has been provided. This test is normally run between a terminal provided by the user and the IM3016. In the first part, the IM3016 continuously outputs the character "A". In the second part, a character transmitted by the terminal is displayed on the IM3016 display.

Test of IM3016 RS232C Port

(1) User connects a terminal to the IM3016 via an RS232C cable.

(2) The terminal and IM3016 are set for the desired baud rate, parity, etc. (See Section 7.3)

Key Entry	Display	Explanation
TEST	TEST	
ENTER	TEST = ?	
11	TEST =11	
ENTER	SW=22	The value of the IM3016 switch is displayed. Switch Positions 2 and 6 are on.
ENTER	OUT=A	A continuous stream of upper case "A"'s are transmitted at 1200 baud.
ENTER*	IN = X	Where X equals the ASCII character sent by the terminal. Only upper case letters will be displayed correctly.

7.11 LOADING AN OBJECT FILE

There are various formats which may be used to load data into the IM3016 RAM. Some of the attributes which separate the different formats are the maximum available memory that can be addressed in the record, the efficiency of the format, and the error detection capabilities of the format. It should be noted that the Offset and Filter commands as well as SPEC 3 can be used to change what data is loaded into the IM3016 and where it is put in the Buffer RAM. Of course, the user will probably choose the format which is most compatible with his or her development system. However, a brief review of the advantages and disadvantages of the various formats may be of some use to those who wish to optimize the loading of data into the IM3016.

Summary: Intel Hex
Advantages: Addresses greater than 64K. Can be used with 16-bit processors.

Summary: Tekhex
Advantages: Processor Independent. Handshake between sender and reciever for reliable phone line transmission.

Summary: Binary
Advantage: More than twice as efficient than ASCII formatted data.
Disadvantage: Starting address is assumed to always be 0000. Note however, that the Offset Command can change the starting address.

EXAMPLE: Loading an Intellec Hex File.

Key Entry	Display	Explanation
SPEC	SPEC	
10	SPEC =10	
ENTER	INTEL IN	
ENTER	ADD=	IM3016 waits for start of text character and address information.
	ADD=AAAA	Line address AAAA information received. Data loaded into RAM.
	F.I.N. .A.A.A.A.	End of text. Character received.

7.12 DUMPING AN OBJECT FILE

Various formats are available for dumping an object file from the IM3016. However, 16-bit data dumps are not available at this time. In effect, the data format is assumed to be 8 for all data dumps. The Intel format allows dumping more than 64K bytes; all other formats are restricted to an upper limit of 64K. If a dump is initiated from the IM3016 keyboard, the start address of the dump is determined by where Socket 1 is mapped in the IM3016 buffer RAM (i.e., the value determined by SPEC 5).

EXAMPLE: Dump of 72Kx8 of Buffer RAM.

```
TYPE = 8Kx8 PROM
SET  = 9
S1AD = 08000, S1BK = 0
```

Key Entry	Display	Explanation
SPEC	SPEC	
ENTER	SPEC = ?	
11	SPEC =11	
ENTER	INTELOUT	Choose Intellec Out.
ENTER	OUT RDY?	Request to send data.
ENTER	ADD=AAAA	File is transmitted with each line address displayed.
	E.N.D. .A.A.A.A.	Dump is completed. Data from $08000 to $1BFFF transmitted.

In addition to the format end-of-file indicator, the IM3016 also sends a ^Z to terminate the file.

7.13 SPEC 10 & 11: INTELLEC HEX FORMAT

1. RECORD MARK FIELD: Frame 0.

   The ASCII code for a colon (:) is used to signal the start of a record.

2. RECORD LENGTH: Frames 1 and 2.

   The number of data bytes in the record is represented by two ASCII hexadecimal digits in this field. The high-order digit is in Frame 1. The maximum number of data bytes in a record is 255 ($FF). An end-of-file record contains two ASCII zeroes in this field.

3. LOAD ADDRESS FIELD: Frames 3 to 6.

   The four ASCII hexadecimal digits in frames 3 to 6 give the address in whcih the data is loaded. The high-order digit is in Frame 3. The first data byte is stored in the location indicated by the load address. This field, in an end-of-file record, contains zeroes.

4. RECORD TYPE FIELD: Frames 7 and 8.

   The ASCII hexadecimal digits in this field specify the record.

RECORD TYPE	Description
00	Data Record
01	End-of-File Record
02	Extended Address Record
03	Start Address Record (ignored)

5. DATA FIELD: Frames 9 to 9+2*(record length)-1.

   A data byte is represented by two frames containing the ASCII characters 0-9 or A-F, which represent a hexadecimal value between 0 and $F.

6. CHECKSUM FIELD: Frames 9+2*(record length) to
   9+2*(record length) +1.

   The checksum field contains the ASCII hexadecimal representation of the two's complement of the 8-bit sum of the 8-bit bytes that result from converting each pair of ASCII hexadecimal digits to one byte of binary, from the record length field up to and including the last byte of the data field. When all bytes of a record are added together, including the checksum, the result will be 0 modulo 256.

EXAMPLE: Intellec Hex.

```
:02319400923176
:00000001FF
```

USING THE IM3016 WITH INTELLEC RECORD TYPE 02 FILES.

If the user wishes to load Intel Hex files with type 02 records, they should refer to the following publication for a complete understanding of what follows: Intel's MCS-86 ABSOLUTE OBJECT FILE FORMATS, AN INTEL TECHNICAL SPECIFICATION, order # 9800821A. In brief, a type 02 record contains a 4 digit address which is transformed into a 5 digit address by adding a 0 on the end of the specified address. This is referred to as the USBA address. The USBA is then added to all subsequent type 00 record addresses to determine the absolute load address of any data byte in the type 00 record.

Example: Determining the Absolute Intel Load Address

```
:02000002F123E8      USBA = $F1230
:02456700A1A20F      REC AD = $4567; Data = $A1 and $A2

                     Thus the address of data byte $A1 is
                     $F1230 + $04567 = $F5759
                     And the address of data byte $A2 is $F575A
```

If the user has specified an 8-bit data format (e.g., an 8088 processor file) and the IM3016 has 128K bytes of RAM, the data $A1 will be placed at $15759 and $A2 will be placed at $1575A (See Section 7.3 on SPEC 3). However, if the user used SPEC 3 to specify 16-bit data (e.g., an 8086 processor file), life gets a little more complicated. In this case, data byte $A1 ends up at $12BAC and data $A2 ends up at $1ABAD. To see how this comes about, let's assume that the absolute load address formed from the USBA and the type 00 data record address is X5 X4 X3 X2 X1 . As explained in Section 7.3, X5 will be truncated to the size of the IM3016 RAM and form the 64K IM3016 bank value. The remaining hex digits, X4 X3 X2 X1, are now manipulated to split odd and even bytes into the corresponding lower and upper halves of the chosen 64K bank. Specifically, X4 X3 X2 X1 is logically shifted right one bit, the carry complemented, and then placed in the MSB of the shifted address.

Let X4 X3 X2 X1 have an original binary value of

A15 A14 A13 A12 A11 A10 A9 A8 A7 A6 A5 A4 A3 A2 A1 A0

After manipulation, the binary value becomes

Ac A15 A14 A13 A12 A11 A10 A9 A8 A7 A6 A5 A4 A3 A2 A1

where Ac stands for the complement of A0.

This manipulation has the desired effect of separating odd and even bytes in such a way as to produce pairs of PROMs in adjacent sockets with the high (odd bytes) value to the left.

Example: Manipulation of X4X3X2X1 for 16 Data.

Address X4X3X2X1	1405H
binary before address	0001 0100 0000 0101
and complemented carry after	0000 1010 0000 0010
IM3016 RAM address	0A02H

Example: Other sample Intel hex files with 16-bit data.

```
:02000002F0000C
:1000F4000002D1E383BFACDF01751F8A1E79E0B72C
```

The data is loaded as follows;

ADDRESS	DATA	RAM ADDRESS	DATA
1007A	02	1807A	00
1007B	E3	1807B	D1
1007C	BF	1807C	83
1007D	DF	1807D	AC
1007E	75	1807E	01
1007F	8A	1807F	1F
10080	79	18080	1E
10081	B7	18081	E0

```
:02000002E123F8
:1001D500DFA079E0B40089C6C6840EE002C787DADD
```

RAM ADDRESS	DATA	RAM ADDRESS	DATA
00A02	DF	08A02	
00A03	79	08A03	A0
00A04	B4	08A04	E0
00A05	89	08A05	00
00A06	C6	08A06	C6
00A07	0E	08A07	84
00A08	02	08A08	E0
00A09	87	08A09	C7
00A0A		08A0A	DA

Example: Loading an Intel Hex file for the 8086.

Relevant conditions: TYPE = 2764. Set Size = 16.

1. Set PROM type. (see section 4.14)

KEY ENTRY	DISPLAY	EXPLANATION
TYPE	TYPE	TYPE SELECT
ENTER	TYPE= 0	
121	TYPE=121	
ENTER	T= 2764	2764 SELECTED

2. Set data size.

KEY ENTRY	DISPLAY	EXPLANATION
SPEC	SPEC	ENTER SPEC MODE
ENTER	SPEC= ?	
3	SPEC= 3	SELECT SPEC 3
ENTER	DATA=8 ?	
STEP	DATA=16?	SET DATA SIZE TO 16 BITS
ENTER	D.A.T.A.=.1.6.	

3. Set the set size to 16 (see section 4.10)

KEY ENTRY	DISPLAY	EXPLANATION
SET	SET	SET SIZE SELECT
ENTER	SET= 1 ?	
CLEAR	SET=16 ?	SET OF 16 SELECTED
ENTER	S.E.T.=.1.6...	

4. Fill the IM3016 RAM with 00's so that non-loaded areas are NOP's.

KEY ENTRY	DISPLAY	EXPLANATION
SPEC	SPEC	ENTER SPEC MODE
ENTER	SPEC= ?	
4	SPEC=4	SELECT SPEC 4
ENTER	FILL=FF?	
STEP	FILL=00?	SELECT 00'S
ENTER	F BUSY	RAM FILLING WITH 00'S
	D.O.N.E..F.I.L.L.	

5. Set for Intel format (see section 7.9)

KEY ENTRY	DISPLAY	EXPLANATION
SPEC	SPEC	ENTER SPEC MODE
ENTER	SPEC= ?	
10	SPEC=10	SELECT SPEC 10
ENTER	INTL IN	
ENTER	ADD=	IM3016 WAITING FOR INTEL FILE

7.14 SPEC 12 AND 13: MOTOROLA S RECORD (S1S9)

The Motorola Exorcisor Microcomputer Development System transfers data in a format designated by Motorola as the MICBUG Hexadecimal format.

1. RECORD TYPE FIELD: Frame 0 and 1.

   The ASCII character "S" followed by a decimal digit signifies the type of record.

   An S-Record-format module may contain S-Records of the following types:

   S0 The header record for each block of S-Records. The code/data field may contain any descriptive information identifying the following block of S-Records. Under VERSAdos, the resident linker's IDENT command can be used to designate module name, version number, and description information which will make up the header record. The address field is normally zeroes.

   S1 A record containing code/data and the 2-byte address at which the code/data is to reside.

   S2 A record containing code/data and the 3-byte address at which the code/data is to reside.

   S3 A record containg code/data and the 4-byte address at which the code/data is to reside.

   S5 A record containing the number of S1, S2, and S3 records transmitted in a particular block. This count appears in the address field. There is no code/data field.

   S7 A termination record for a block of S3 records. The address field may optionally contain the 4-byte address of the instruction to which control is to be passed. There is no code/data field.

   S8 A termination record for a block of S2 records. The address field may optionally contain the 3-byte address of the instruction to which control is to be passed. There is no code/data field.

   S9 A termination record for a block of S1 record. The address field may optionally contain the 2-byte address of the instruction to which control is to be passed. Under VERSAdos, the resident linker's ENTRY command can be used to specify this address. If not specified, the first entry point specification encountered in the object module input will be used. There is no code/data field.

Only one termination record is used for each block of S-records. S7 and S8 records are usually used only when control is to be passed to a 3- or 4-byte address. Normally, only one header record is used, although it is possible for multiple header records to occur.

2. RECORD LENGTH FIELD:

The number of data bytes in the record is represented by two ASCII hexadecimal digits in this field. The high-order digit is in frame 1. The maximum number of data bytes in a record is 255 ($FF). An end-of-file record contains two ASCII zeroes in this field.

3. LOAD ADDRESS FIELD:

The four ASCII hexadecimal digits in frames 3 to 6 give the address at which the data is loaded. The high-order digit is in frame 3, the lower-order digit is in frame 6. The first data byte is stored in the location indicated by the load address; successive bytes are stored in successive memory locations. This field in an end-of-file record contains zeroes or the starting address of the program.

4. DATA FIELD: Frames 8 to 8+2* (record length)-1.

A data byte is represented by two frames containing the ASCII characters 0 to 9 or A to F, which represent a hexadecimal value between 0 and $FF (0 and 255 decimal). The high-order digit is in the first frame of each pair. If the data is 4-bit, then the low-order digit represents the data and the other digit of the pair may be any ASCII hexadecimal digit. Data transmitted by the IM3016 always has the high order digit set to an ASCII H'F' when data is 4-bit.

5. CHECKSUM FIELD: Frames 8+2* (record length) to
8+2* (record length) +1.

The checksum field contains the ASCII hexadecimal representation 8-bit sum of the 8-bit bytes that result from converting each pair of ASCII hexadecimal digits to one byte of binary, from the record length field to and including the last byte of the data field. Therefore, the sum of all the ASCII pairs in a record after converting to binary, from the record length field to and including the checksum field, is hexadecimal 'FF'.

EXAMPLE: S-RECORD hexadecimal format.

```
S00600004844521B
S1130000285F245F2212226A000424290008237C2A
S11300100002000800082629001853812341001813
S113002041E900084E42234300182342000824A952
S107003000144ED492
S9030000FC
```

SAMPLE TEST FILE FOR MOTOROLA S RECORDS

```
      ADDRESS  DATA

       0C2905  75
       000030  00 14 4E D4
     99511234  56
```

THE LAST RECORD CREATES A CHECKSUM ERROR WITH THE DISPLAY=EC123555

```
       S RECORDS

     S2050C2905754B
     S107003000144ED492
     S3069951123456C9
     S9
```

7.15 SPEC 14 AND 15: TEKHEX

The Tektronix 8001 and 8002 microprocessor labs transfer data in a unique format, designated as Tektronix Hexadecimal Format (TEKHEX).

The user is requested to familiarize him/herself with the COMM command section of the Tektronix user's manual. The COMM command details how the user is to upload or download data from/to the IM3016. The IM TEKHEX format utilizes the handshaking routine for acknowledging the correct receipt of a formatted line of data.

A correctly transmitted line of data will result in the development system or the IM3016 transmitting an ASCII zero (0). A transmission error will result in an ASCII seven (7) being sent with the expectation that the incorrect line will be retransmitted.

1. RECORD MARK FIELD: Frame 0.

   The ASCII code for a slash (/) is used to signal the start of a record.

2. LOAD ADDRESS FIELD: Frames 1 to 4.

   The starting location in memory from which data will be output from the IM3016 consists of four hexadecimal digits.

3. RECORD LENGTH FIELD: Frames 5 and 6.

   The number of data bytes in the record is represented by two ASCII hexadecimal digits in this field. The high order digit is in frame 5. The maximum number of data bytes in a TEKHEX record output by an IM3016 is $10 (16 bytes decimal). An end-of-file record contains two ASCII zeros in this field.

4. FIRST CHECKSUM FIELD: Frames 7 and 8.

   The "First Checksum" is the eight-bit sum of the four-bit hexadecimal values of the six digits that make up the location counter and the byte count. The checksum is a two-digit hexadecimal number.

5. DATA FIELD: Frames 9 to 9+ (Record Length) -1.

   A standard IM record length consists of 16 bytes (32 hexadecimal digits).

6. SECOND CHECKSUM FIELD: Frames 9+ (Record Length) to (Record Length) +1.

The "Second Checksum" is the eight-bit sum, modulo 256, of the four-bit hexadecimal values of the digits that make up the 16 bytes of data. The checksum is a two digit hexadecimal number.

7. END OF RECORD FIELD.

The last frames of every record are a carriage return, line feed character sequence. Following the CR LF sequence, the device accepting data will transmit the acknowledgement character (0=no errors, retransmission not required; 7=error detected, retransmission required).

EXAMPLE: Standard Record

/000005051A023F7B633A CR
/001010029EB70358AFFF094C5681110292F8A695E1 CR
/00200A0C1C7B5432F8A6B695963287 CR

Terminating Block Format

/03FF0021 CR

Note: The load address for the terminating block is variable.

Abort Block Format

// (error information) CR

Note: The abort block is indicated by a double header character being transmitted by the Tektronix microprocessor lab.

7.16 SPEC 20 AND 21: ROCKWELL 6502

The format described below is used by the Rockwell Development System and other 6502-based software.

1. RECORD MARK FIELD: Frame 0

   The ASCII code for a semicolon (;) is used to signal the start of a record.

2. RECORD LENGTH FIELD: Frame 1 and 2

   The number of data bytes in a record is represented by two ASCII hexadecimal digits in this field. The high-order digit is in frame 1. The maximum number of data bytes in a record is 255 ($FF). An end-of-file record contains two ASCII zeroes in this field.

3. LOAD ADDRESS FIELD: Frames 3-6

   The four ASCII hexadecimal digits in frames 3-6 give the address at which the data is to be loaded. The high-order digit is in frame 3, the lower order digit in frame 6. The first data byte is stored in the location indicated by the load address; successive bytes are stored in successive memory locations. This field in an end-of-file record contains zeroes.

4. DATA FIELD: Frames 7 to 7+2* (Record length)-1

   A data byte is represented by two frames containing the ASCII characters 0-9 or A-F, Which represent a hexadecimal value between 0 and FF (0 and 255 decimal). The high order digit is in the first frame of each pair. There are no data bytes in an end-of-file record.

5. CHECKSUM FIELD: Frames 7+2* (Record length) to
   7+2* (record length) +3

   The checksum field contains the 16-bit sum of each pair of converted ASCII to hex characters, not including the checksum and the starting semicolon.

EXAMPLE:

```
;100410A9048DFFFFD858A2FF9AA2281890FEAD09E4
;0000000000
```

7.17 SPEC 22 TI DEVELOPMENT SYSTEM

SPEC 22 loads object records as described in TI's publication 943441-9701, section 10.5.1, and publication 949617-9701, Appendix F. With reference to the above Appendix F, the following tags are recognized and processed by SPEC 22.

TAG	Function	Total Field Characters	IM3016 Response
0	PSEC length and program ID	12	Ignore
7	Checksum	4	Checksum
9	Absolute load address	4	Fix address
B	Absolute data	4	Load data
F	End of line	0	Wait for CR/LF
:	End of file	0	End processing

In general, if any other tags appear in the record, they will be ignored. All characters received as possible tags with hex values less than 30H ('0') will be ignored. Also, entry of the EXIT key from the IM3016 keyboard will abort the program.

CKSUM Calculation

The checksum occurring after Tag 7 is the two's complement of the sum of all printable ASCII values record from the first tag of the record through the checksum tag, "7". Thus, after tag 7, the checksum is cleared and activated by any of the valid tags given. Every character including the first valid tag until and including Tag 7 has its ASCII value added to the checksum. After Tag 7 is received, the 4 digit hex representation of the succeeding 4 characters is added to the checksum and the resulting 4 digit hex value should be zero. If it is not, a checksum error will occur and program execution will be aborted.

8 or 16-Bit Data

The TI format can be used with both 8- and 16-bit data. The choice is made depending upon the value specified by SPEC 3 (see Section 7.3). If 16-bit data is chosen, odd bytes are placed in the bottom half of a 64K block and even bytes in the top half. The method used is exactly the same as the Intel Hex format (Section 7.11). The user should study this section for a complete description of how the odd and even bytes are placed in the IM3016 buffer RAM.

Example: TI format

00000CLINKR 90000BA3DCB0040BA3DCB271ABA510B28DEBA2E8B0044BA9A27F187F

Where start address = 0000
First data = A3DC
Checksum = F187

For 8-bit data, the IM3016 RAM locations have the values:

00000 = A3
00001 = DC

For 16-bit data, the IM3016 RAM locations have the values:

00000 = DC
08000 = A3

PROM Socket 1 will map to 00000; Socket 2 will map to 08000.

7.18 Binary Load (SPEC 24) and Binary Dump(SPEC 25)

The binary format described below transfers bytes as opposed to the ASCII formats which transfer ASCII encoded nibbles. The user must set the 8 position dip switch in the rear of the IM3016 for 8-bit data. The start address is assumed to be 0, although this may be changed by use of the Offset command under computer control. The format consists of a start of text header, a 16-bit byte count, data, and a 16-bit byte checksum.

Start of Text Header: An arrow head $08,$1c,$2a,$49,$08,$00

4 Nibbles specifying the Byte Count, followed by $FF

Load Data with Add to Checksum

Two Null Bytes

Two Byte Checksum MSD first

Note: All values in Hex.

Example 1:

08,1C,2A,49,08,00,00,00,00,04,FF,01,02,03,04,00,00,00,0A

Where Arrow Head = 08,1C,2A,49,08,00
Byte Count = 0004
First Data = 01
Last Data = 04
Checksum = 000A

Example 2:

08,1C,2A,49,08,00,00,00,02,00,FF,01,02,04,08,10,20,40,80,55,AA,
01,02,03,04,05,06,07,08,09,0A,0B,0C,0D,0E,0F,10,11,12,13,14,15,16,
00,00,02,FB

Where Arrow Head = 08,1C,2A,49,08,00
Byte Count = 0020
First Data = 01
Last Data = 16
Checksum = 02FB

During the Load of the Binary File, the IM3016 will display "Ready".

7.19 HP Binary (SPEC 30)

The HP Binary format is used by HP development systems. All files start with a header byte($04), followed by ten bytes which are ignored. Next is a two byte data byte count with the most significant byte first, followed by a four byte start address, msb first; data bytes and then a single checksum byte. The checksum byte is the simple sum modulo 256 of all bytes starting with the first byte starting with the data byte count msb and ending with last data byte. An end of file is signified by a record with a zero byte count.

Example : HP Binary

04,00,01,02,03,04,05,06,07,08,09,12,34,87,65,43,21,...data..,CK

where : St of File = 04
Ignore = 00-09
Byte Count = 1234
St Address = 87654321
CK = Data Sum + 12+34 +87+65+43+21

7.20 SPEC 32: LDA Load

The LDA (Load Absolute) File Format is used by Digital Equipment Corporation (DEC) computers, including the VAX computers. All the files begin with a start-of-header byte specifying the size of the address (1 for 16-bit addresses and 2 for 32-bit addresses). These are followed by two bytes specifying the byte-count, with the least significant bytes first; followed by either two or four load address bytes (2 for 16-bit addresses and 4 for 32-bit addresses),again least significant bytes first. The data follows the address and finally a checksum byte equal to the 2's complement of all previous bytes excluding the start of text byte modulo 256. An End of File is signified by the record with no data bytes.

LDA File Formats:

1. 16-Bit Address Mode

Example: 01 00 07 00 34 12 56 5B places a 56 at location 1234:
Address: 12 34
Data: 56
Byte Count: 0007 = 6 + # of data bytes = 6 + 1
= # of bytes in record excluding checksum

Byte		Example
0 -	01 (16-bit address mode)	01
1 -	00	00
2 -	Byte Count: least significant byte (n & 00FF)	07*
3 -	Byte Count: most significant byte	00
4 -	Load Address: least significant byte	34
5 -	Load Address: most significant byte	12
6->n -	Data	56
n+1 -	Checksum: 2's complement of previous 'n' bytes	5C

*If the byte-count = 06, this signifies the end of file.

2. 32-Bit Address Mode

Example: 02 00 09 00 67 45 23 01 89 99 places 89 at the location
Address: 01 23 45 67
Data: 89
Byte Count: 0009 = 8 + # of data bytes

Byte		Example
0	- 02 (32-bit address mode)	02
1	- 00	00
2	- Byte Count:least significant byte (n & 0FF)	09*
3	- Byte Count:most significant byte (n & -100/100)	00
4	- Load Address: A7 - AO	67
5	- Load Address: A15 - A8	45
6	- Load Address: A23 - A16	23
7	- Load Address: A31 - A 24	01
8->n	- Data	89
n+1	- Checksum: 2's complement of previous 'n' bytes.	9C

*If the byte-count = 08, this signifies the end of file.

7.21 RCA COSMAC Format Load (SPEC 34) and RCA COSMAC Dump (SPEC 35)

RCA COSMAC Format utilizes a driver (UT4) that requires that data be transferred in the following format:

1. RECORD MARK FIELD: Frames 0 and 1

The ASCII code for a carriage return (CR) is used to signal the start of a record.

2. LOAD ADDRESS FIELD: Frames 2-5

The four ASCII hexadecimal digits in frames 2-5 give the address at which the data is loaded. The high-order digit is in frame 2, the lower order digit in frame 5. The first data byte is stored in the location indicated by the load address; successive bytes are stored in successive memory locations.

3. SEPARATOR FIELD: Frame 6

The value for an ASCII space is used to separate the address field from the data field. An ASCII space is also used to separate every four data digits in the data field.

4. DATA FIELD:

A data byte is represented by two frames with hexadecimal digits representing a value between 0 and FF. The high-order digit is in the first frame of each pair. If the data is 4-bit, then the low-order digit represents the data and the other digit of the pair may be any ASCII hexadecimal digit. Data transmitted by the IM3016 always has the high-order digit set to an ASCII H'F' when data is 4-bit.

5. END-OF-LINE FIELD:

Line termination is represented by an ASCII semicolon (;). The ASCII value for a carriage return (CR) must follow the end-of-line field before a new line can be transferred.

6. END-OF-RECORD-BLOCK FIELD:

The termination of data transfer is ended by a carriage return (CR) in the end-of-line field rather than a semicolon.

RCA COSMAC Format Example:

```
<CR>
0100 311A 320E 0311 7E31 CD40 003A 9231 B7C2 ;  <CR>
0110 6031 0E00 1170 31CD 4000 3A92 31B7 C260 ;  <CR>
0120 312A 7E31 227A 310E 0311 7E31 CD40 003A ;  <CR>
0130 9231 B7C2 5031 2A8C 317C B5CA 5031 0E04<CR>
```

7.22 SPEC 42: LOAD ASCII TEXT

SPEC 42 loads ASCII text into the IM3016. The default start address is 0. The start of text is a linefeed (LF) or carraige return (CR). The End of File is a control Z ($1A). Thus any 7 bit value is a valid data byte other than CNTL Z.

To Use from Keyboard:

1. Key in the following

Key Entry	Display
SPEC	
ENTER	
4	
2	
ENTER	ASCII IN
ENTER	ADD=

2. Connect up terminal

3. There is no start of text character. The first character entered will go into RAM location 00000. The second character into 00001, etc.

4. To terminate, Type in a Control Z (ASCII Hex 1AH).

5. All characters can be loaded except Control Z. Any 8 bit value received with its msb set, will have its msb reset. For example, if a FFH is received, it will be changed to a 7FH.

6. Upon receiving the CNTL Z end of text character, the display will read FIN AAAA where AAAA is the last location + 1 written to.

7. As each character is entered, the IM3016 will attempt to show it in digit 8 of its display. However, many characters will not be recognizable due to the limited character set of the 14 segment display. In particular, lower case alpha characters are not recognizable.

To Use from Computer Control:

The computer command is L0:2A. The first character following CR or LF or CRLF will be loaded into location 00000. Note that a CRLF combo must not have any time delay between these characters. The user may direct the text to a different start address other than zero by use of the OFFSET command.

Example: **OF:1,2345 LO:2ACRLF Pause**
678 (CNTLZ)

will load the string 678 starting in location 12345 assuming the user has at least 128K or RAM.

SECTION 8: TESTING THE IM3016

8.0 INTRODUCTION

The IM3016 is provided with a number of software aids to allow testing in the field. These are accessed via the TEST key. Tests may be run on all lighted indicators, the Buffer RAM memory and the module sockets. The testing of the RS232C port (TEST 11) is covered in Section 7.8.

Listing of all Tests:

Test #	Use
1	All lighted indicators.
2	Socket address, control, and data lines.
3	Reading socket data lines.
4	Socket Vp lines.
5	Buffer RAM test.
11	RS232C Port.

8.1 TEST 1: TESTING LIGHTED INDICATORS

RUNNING TEST 1.

Key Entry	Display	Explanation
TEST	TEST	Enter test mode.
ENTER	TEST= ?	
1	TEST=1	Choose Light Test.
ENTER	LED TEST	All lights off.
ENTER	LED TEST	All green LEDs on, all else off.
ENTER	LED TEST	All red sockets LEDs, START, LOAD ERROR and BLANK ERROR on.
ENTER	LED TEST	All green LEDs on.
EXIT	I.M...3.0.1.6.	EXIT and return to monitor.

8.2 TESTING THE MODULE

Testing any PROM programming socket consists of six basic elements. These are the power supply lines, the address lines, the control lines, the data lines driving the socket, the data lines reading the socket, and the program pulse lines. The methods for testing each of these elements are given in the following sections.

8.3 TESTING THE SOCKET POWER SUPPLY LINES

All IM3016 modules present a "dead" socket in standby. That is, in standby no pin should be at any voltage other than ground. Before a socket is accessed for reading or programming, a Load Sequence is initiated to test for correct PROM insertion. Thus, the Load Sequence provides a convenient means for watching the power lines switch. The following steps describe the general method of power supply line testing using a Load Sequence.

EXAMPLE: Testing the Vcc line for the 2532 PROM.

Necessary Test Equipment:

1. An oscilloscope and a load resistor of suitable wattage.

   The value of the resistor should be:

   load resistor = 5V/PROM max current

   For the 2532:

   RL = 5V/160ma = 30 ohms

2. A 24 pin header prepared as follows. Connect the resistor between ground (Pin 12) and Vcc (Pin 24) of the 24 pin header. **DO NOT under any circumstances put resistors, scope leads, etc., directly into the zero insertion force sockets as the pins are easily damaged.**

RUNNING THE VCC TEST.

(1) Choose the correct TYPE with the SET size = 1.

(2) Insert the 24 pin header with the correct load resistor into Socket 1.

(3) Place the scope on the side of RL connected to Pin 24 and the scope ground on Pin 12. Note the bottom left hand corner of each socket is the ground connection. This will be Pin 12 for 24 pin PROMs and Pin 14 for 28 pin PROMs.

(4) You are now ready to observe the Vcc signal.

Key Entry	Display	Explanation
VERIFY	VERIFY	
ENTER	NONE ACT	Each entry of START or ENTER will produce a pulse from ground to Vcc.
.		
.		
.		
ENTER		

(5) After observing this waveform for Socket 1, move the test head to all other sockets and repeat the test.

(6) If any socket does not have the correct waveforms, consult the factory.

8.4 TEST 2: TESTING THE CONTROL, ADDRESS, AND DATA OUT LINES

Test 2 exercises all lines driving the module sockets other than ground and power lines. Test 2 software will pulse each line while all others are grounded.

Necessary Test Equipment:

1. An oscilloscope of bandwidth 25mhz and a socket header.

RUNNING TEST 2.

(1) Enter TEST, ENTER, 2, ENTER. The display reads TOG A D.

(2) Place an appropriate socket connector (the one used for Vcc test is OK) in Socket 1.

(3) Connect the ground of the scope to the socket ground.

(4) Using the scope, check that a short positive pulse (TTL level) appears on all pins but the power supply pins. The length of the pulses may be different (e.g., the data line pulses are generally longer).

(5) Test all sockets in a similar manner.

8.5 TEST 3: TESTING THE DATA IN LINES

Test 3 is used for verifying that data in a PROM is correctly read by the IM3016. (Note: A PROM type must be defined for Test 3 to operate.)

Necessary Test Equipment:

1. A test header and 33 ohm resistor.

RUNNING TEST 3.

(1) Enter TEST, ENTER, 3, ENTER. The display reads DIN FFFF. A green LED should appear over all sockets.

(2) Insert the test header in Socket 1. In turn, connect the 33 ohm resistor between ground and each data line. The display will read DIN FFXX where XX is FE, FD, FB, F7, EF, DF, BF, and 7F depending on which data line is connected to the 33 ohm resistor. A red LED will also appear over the SUT ("Socket Under Test").

(3) Test all other sockets in a similar manner.

8.6 TEST 4: TESTING THE PROGRAM LINES

Test 4 tests the program lines. In general, the high voltage pulses necessary to program a PROM are characterized by rise and fall times, voltage limits, current limits and program pulse lengths. Since the IM3016 module is designed to program many different types of PROMs, Test 4 has been written to pulse all socket programming pins with all required program pulse levels in one simple test, while ignoring the program pulse length requirements. In order to test the IM3016 module for a specific type, the user may observe the waveforms of a device actually being programmed with the PROG key. However, Test 4 provides a quick way to test all program pins on all sockets.

Necessary Test Equipment:

1. An oscilloscope and a 24 or 28 pin socket header.

2. An appropriate load resistor where:

   RL = max Vp for module / max Ip for module

   For example: The IM3016-1V-28 module has:

   Vp max = 25V
   Ip max = 50ma
   RL = 25V / 50ma

   RL = 500 ohms

RUNNING TEST 4.

(1) Enter TEST, ENTER, 4, ENTER. The display reads ALL VP.

(2) Using the test header and the scope, but not the load resistor, place the scope on all program pins of interest. The Vp voltage level should be within a volt of the nominal value given in the PROM manufacturers' data sheet.

(3) Connect the 500 ohm resistor between ground and the appropriate Vp pins on the test header. Connect the scope across the resistor and observe the Vp pulses. Do this for each socket in turn.

8.7 TEST 5: TESTING THE BUFFER RAM

The IM3016 Buffer RAM utilizes 64Kx1 dynamic RAMs in a x9 configuration. The ninth bit is used for a parity bit which should always be high (1). Test 5 can be used to test the Buffer RAM both at the board level and at the system level. Only the system level test will be given here.

EXAMPLE: Extended testing of the IM3016 Buffer RAM.

RUNNING THE TEST.

Key Entry	Display	Explanation
TEST	TEST	
ENTER	TEST= ?	
5		Choose Memory Test.
ENTER	MEM TES1	Board Test on RAS and CAS lines.
ENTER	$AAAA=AA	Data Line Test.
ENTER	$D555=55	Data Line Test.
ENTER	Write B3 XXXX	BANK 3 being written.
	Write B2 XXXX	BANK 2 being written.
	Write B1 XXXX	BANK 1 being written.
	Write B0 XXXX	BANK 0 being written.
	Read B0 XXXX	BANK 0 being read.
	.	Etc.
	.	"
	.	"

The last test will continue indefinitely until an error is detected or the EXIT key is entered. If an error is detected, the following will take place:

Display: P=X YYZZ

YY = Data that should be read.
ZZ = Data that is read.

The next ENTER will produce the following:

Display: AD=BAAAA

where B = the Buffer RAM Bank (0 to 3).
AAAA = the bank address.

Subsequent ENTERs will continue the test.

This final Memory Test is a good test for extended time testing of the IM3016 as it will continue indefinitely until an error occurs.

8.8 TEST 10: LIST IM3016 RAM IN ASCII

Test 10 is used to list the contents of the RAM both in hexadecimal binary and in ASCII. The format is

AAAAABBC Where

AAAAA = Hexadecimal address of the IM3016 RAM
BB = Hexidecimal representation of the data
C = ASCII representation of the data.

Because of the restricted character set of the LED digits on the IM3016 display, many non alpha characters will display a non recognizable symbol. For instance, a null byte, $00, in location 00 will be displayed as below

00000 00 G

Add=0 Data=00

Also, lower case alpha characters will be displayed as upper case.

Example: location 01234 holds $61 which is a lower case "a".

IMDisplay = 0123461A

As in the standard LIST Function, there are various speeds of listing available using the numbers 1-5; and the STEP and CLEAR keys are used to list increasing or decreasing addresses.

Example: List locations 0 and 1 with data $31 and $32 respectively.

Key Entry	Display
TEST	
ENTER	
1	
0	
ENTER	LIST ASC
ENTER	00000311
ENTER	00001322

8.9 TEST 11: TESTING THE RS232C PORT INTERFACE.

RUNNING TEST 11.

Key Entry	Display	Explanation
TEST	TEST	Enter test mode.
ENTER	TEST= ?	
11	TEST=11	
ENTER	SW=98	The value of the IM3016 switch is displayed. Switch positions 4, 5, and 8 are on.
ENTER	OUT=A	A continuous stream of upper-case "A"s are transmitted at 9600 baud.
ENTER*	IN = X	Where X equals the ASCII character sent by the terminal. Only upper-case letters will be displayed correctly.

8.10 TEST 12: TEST BINARY IN

Test 12 loads into the IM3016's RAM all characters recieved via the RS232C port with the first character going into location 0. This is meant solely for testing and not as a load format. Test 12 may only be exited using the EXIT key.

Test 12 is useful for debugging any ASCII or Binary Transmission to the IM3016. Used in conjunction with Test 10 (List IM3016 RAM in ASCII) Test 12 allows the user to view exactly what the IM3016 receives at its RS232C port.

Key Entry	Display	
TEST		
ENTER		
1		
0		
ENTER	TEST BIN	
ENTER	ADD= X	Where X is ASCII value of received character

SECTION 9: GENERAL INFORMATION ON TERMINAL & COMPUTER CONTROL

9.0 INTRODUCTION

The IM3016 terminal and computer package allows the user to control the IM3016 programmer from a terminal or host computer. By setting a particular control byte, the user can configure the IM3016 to operate in a multitude of different modes. These include: half or full duplex, Control Q - Control S response, enable or disable prompt messages, CR only or LFCR responses, and enable or disable IM3016 keyboard operator responses. The ability to dynamically control the IM3016's mode of response allows the user to freely mix computer directed commands and IM3016 operator interaction.

This flexible control structure has been provided so that the user can choose or mix a number of different methods in controlling the IM3016. These include the following:

1. Terminal control.
2. Computer set up followed by IM3016 keyboard operator programming.
3. Same as 2 with pass/fail reporting and computer interruption.
4. Complete computer control except for the manual operation of loading and unloading the IM3016 PROM sockets.

There are 6 major categories of commands:

1. Socket related commands; e.g., PROM List.
2. Buffer RAM commands; e.g., RAM List.
3. Mapping Sockets to Buffer RAM commands; e.g., Map RAM to Sockets.
4. Status commands; e.g., Socket Status.
5. I/O related commands; e.g., Load Hex File.
6. Miscellaneous; e.g., Quit.

9.1 LEARNING THE COMMANDS

Because of the wealth of commands, it is recommended that the first time user learn the command formats first by hooking a terminal to the IM3016 and experimenting with all the commands provided. To get the IM3016 up and running with your terminal, proceed as follows:

1. Connect the IM3016 to a terminal set for Full Duplex, Upper Case.

2. Use TEST 11 (see Section 8.8) to make sure the terminal and IM3016 are communicating correctly.

3. Key in SPEC ENTER 2 ENTER to enter the computer control mode.

4. Type in CO:C9 CR from the terminal keyboard.

5. Type in DI:HELLO CR. This "HELLO" should appear on the IM3016 display.

6. Go through all the commands.

7. Do not hesitate to call the factory for help.

SECTION 10: USING THE COMMANDS

10.0 COMMANDS AVAILABLE:

CA	CANCEL COMMAND LINE
CO	CONTROL BYTE DEFINITION
DA	DATA WIDTH CONTROL
DI	DISPLAY MESSAGE
DU	DUMP A FORMATTED FILE
FT	FILTER DATA LOADED INTO RAM
FU	DEFINE RUN FUNCTION
LI	LIGHT ERROR LIGHTS
LL	LIGHT LEDS OVER PROM SOCKETS
LO	LOAD A FORMATED FILE
LT	LOAD TEST
MA	MAP RAM TO SOCKET IN STANDARD MANNER
ML	MAP LIST OF SOCKETS
MR	MAP RAM TO SPECIFIED SOCKETS
MS	MAP SOCKETS TO SOCKETS
OF	SET LOAD OFFSET BANK ADDRESS
PB	PROM BLANK CHECK
PC	SOCKET CHECKSUM
PE	ERASE EEPROMS IN SOCKETS
PL	PROM LIST
PM	PROM MOVE INTO RAM
PP	PROGRAM PROMS
PS	PROM SIZE
QU	QUIT COMPUTER CONTROL
RC	RAM CHECKSUM
RF	FILL ALL OF RAM
RL	RAM LIST
RM	RAM MOVE
RP	RESET RAM PARITY
RR	REPLACE RAM DATA
RU	ENTER RUN MODE
SA	SOCKET ACTIVE STATUS RETURNED
SE	SET SIZE
SL	LOAD TEST STATUS RETURNED
SM	MACHINE STATUS RETURNED
SS	SOCKET STATUS RETURNED
TN	TRANSMISSION TRAILING NULL DEFINITION
TY	TYPE DEFINITION
VE	VERIFY PROMS AGAINST RAM
WA	WAIT ON START KEY

10.1 IM3016 RESPONSES

All responses from the IM3016 are prefixed by the following headers.

*E>	Command excution error condition.
*I>	Ready for a new command.
*P>	RAM parity error detected.
*R>	Result of operation or request follows.
*?>	Bad command syntax.

In addition, it is possible for terminal operation to force an additional prompt of *I>. This will be discussed later.

10.2 DEFINITION NOMENCLATURE

The following definitions use the nomenclature given below.

A,B,C	ASCII letter
D,E,F,G,H	Decimal digit
W,X,Y,Z	Hexadecimal ASCII digits
CR	ASCII carriage return $0D
LF	ASCII line feed $0A
ESC	Escape $1B
;	Command Separator, ASCII semicolon ";"
,	Command Parameter Separator, ASCII comma ","
:	Separator between Command Name and Command parameters, ASCII colon ":"
.	Used by Display Command for SP display, ASCII "."
...	Indicates repeated Command Parameters
$	Hex value follows
#	Command Terminator for variable length commands, ASCII pound "#"
DC3	ASCII $13 (^S)
DC1	ASCII $11 (^Q)

10.3 GENERAL COMMAND RULES

All commands are defined by two letters followed by a colon.

Example: AA:

Commands may be upper or lower case, however, hexadecimal values must all be numbers or upper case letters. Spaces are optional and are ignored. One command line of 80 characters maximum ignoring spaces can be entered at one time and the commands will be sequentially executed. The command line is not prescanned for correctness.

Example: *I>AA: BB:DDD CC:DD,DD

Assuming AA and CC are correct commands but BB is not, AA would be executed, an error would be generated at BB, and a new command line would be required. CC would not be executed.

Leading zeros cannot be omitted in parameter definitions.

Example: TYpe:012 is correct
TYpe:12 is incorrect

10.4 CONTROL OPTIONS

The COntrol command defines the type of handshaking between the IM3016 and the computer and controls the keyboard and the display of the IM3016. The COntrol command may be set for both terminal and computer control. In general, the COntrol command should be the first command sent by the computer. Each bit of the command control byte controls a separate function.

The format is: COntrol:XX

Bit 8 Computer Control

This bit is automatically set upon entering computer control. The user cannot change its value.

Bit 7 Enable *I> Prompt

This bit is normally set for terminal control and reset for computer control. If set, upon the completion of a command, a CR is transmitted and then an *I> is sent.

Bit 6 Enable DC1/DC3

If this bit is set, then the receiver may stop transmission from the sender by sending a DC1 ($11) and restart transmission by sending a DC3 ($13). If this bit is set, the IM3016 will send a DC1 at the beginning of each command execution and will send a DC3 upon termination of a command. It should be remembered that the RS232C port on the IM3016 supports RTS-CTS line control, which, if available, may eliminate the DC1/DC3 requirement.

Bit 5 Enable Diagnostics

If Bit 5 is set, every IM3016 response will be tagged with a

: DD,EE

where DD = Command number
EE = line point value

Setting this bit allows processing of multiple commands per command line.

Bit 4 Reserved

Bit 3 Ignore Keyboard and Operator

If Bit 3 is set, the operator cannot affect operation of the machine from the IM3016 keyboard. Note that the EXIT key is also disabled and therefore if this bit is set, be careful that subsequent commands do not hang up the IM3016. The reset key on the back of the IM3016 is always available if needed.

Bit 2 Suppress Display Messages

If Bit 2 is set, the normal messages displayed on the IM3016 display will be suppressed. The only messages possible are those sent using the DI command.

Bit 1 Enable Full Duplex

If Bit 1 is set, all characters sent to the IM3016 will be echoed back. This bit should be set if operating with a terminal set for full duplex.

Common Settings

Computer Control	$80	The default mode
	$86	Ignore operator and suppress display
	$A0	Enable DC1/DC3
	$84	Ignore operator
Terminal Control	$C9	Enable Prompt, Full Duplex
	$CD	Enable Prompt, Ignore Operator, Full Duplex
	$C8	Enable Prompt, Half Duplex
	$CC	Enable Prompt, Ignore Operator, Half Duplex

10.5 SOCKET RELATED COMMANDS

Before a PROM can be acted upon in a socket, two conditions must first be satisfied. The first of these is referred to as "activating a socket" and the second is that the socketed PROM must pass a "Load Test". To activate a socket, a valid PROM TYPE must be sent to the IM3016 (e.g., TY:121) and then the socket must be mapped to the IM3016 buffer RAM (e.g., MA:). A socket will remain active until another TYPE command is sent. The TYPE command begins by deactivating all sockets. The active status of each socket is returned by the IM3016 in response to the SA: command.

The commands which involve the PROM sockets are PBlank, PChecksum, PMove, PProgram , PList, and PVerify. These commands are preceded by a Load Test. A detailed description of the Load Test is described in the Section 8. In brief, all potentially active sockets are tested for correct insertion of a PROM. Thus, in essence, the above commands each perform two operations; first a Load Test (LT: command) and then the desired function.

Also, commands which access the PROM sockets will normally require at least one second to execute. In particular, the PRogram command may take hundreds of seconds depending on the PROM type.

10.6 ERROR RESPONSES

Errors can be caused by five different conditions. These are unrecognizable commands, incorrect command parameters, command line too long, command execution errors, and a RAM parity error. Appendix D lists all error responses.

Syntax Errors

Command errors such as unrecognizable commands, bad commmand parameters, and a command line greater than 80 characters long, all cause an error response of *?> followed by certain parameters.

The Format is:

*?>DD,FF,GG

D=01 for an unrecognizable command
D=02 for bad command syntax
FF= command number (00 if D=01)
GG= command line pointer value, ignoring spaces

Command Execution Errors

Once a command has been recognized and the syntax of the command parameters has been checked, the command is executed. If a failure occurs during execution of a command, the response is *E> followed by a unique error number.

*E>DD,FF,GG

DD= the unique command error number
FF= the command number
GG= the value of the command line pointer

The command error codes are given in Appendix D. The most common command error codes are:

Error code	Explanation
01	Load Test Error
03	Unknown Command
06	Non Active Socket
07	Unknown Type Error
09	Known Command But Bad Data

Parity Error Response

The IM3016 uses a parity bit for checking RAM integrity. Every time RAM is accessed, the RAM parity circuit is enabled. If a Parity Error is detected, a flag is set which must be reset before normal operation may proceed. If the response to any command is

*P> CR

the computer initiating the response must issue a RParity command before further commands may be processed. Thus, in general, the command response processor software in the controlling computer must allow for the *P> response in all commands.

10.7 DIRECT ENTRY INTO COMPUTER CONTROL

IM3016A programmers use switch position 5 on the rear panel dip switch for power up and reset initialization into external control.

SW-5 Off : The IM3016 initializes with I.M.3.0.1.6 on the display.

SW-5 On : The IM3016 initializes with EX CNTL on the display and a sign-on message of *I> is sent out the RS232C port.

SECTION 11: TERMINAL OPERATION

11.0 TERMINAL OPERATION OF THE IM3016

It is recommended that terminals be set for Full Duplex operation. This makes the operation of the RL and PL commands somewhat cleaner. The user must then decide if he wants to disable the IM3016 keyboard. Assuming the above conditions, the choice of the control word will be:

$C9	Keyboard Enabled
$CD	Keyboard Disabled

Terminal operation of the IM3016 is relatively straightforward except for understanding the mapping of the sockets to the RAM. Note should be made that execution of the TYPE command (TY:) will deactivate all sockets. Until some mastery is gained with the IM3016 commands, it is recommended that after every TY or SE command the user enter the MA command.

SECTION 12: COMPUTER CONTROL

12.0 COMPUTER CONTROL

There are at least three variations of computer control. These are

1. Initial Computer Setup followed by total keyboard control.
2. Initial Computer Setup followed by limited keyboard control.
3. Complete Computer Control.

Engineering environments will normally use number 1.
Manufacturing environments will choose number 2 or 3.

Whichever method is desired, it is recommended that each succeeding step be mastered before progressing to the higher control method.

12.1 STEP 1: COMPUTER SETUP FOLLOWED BY OPERATOR CONTROL

The basic philosophy for Step 1 is that the computer control will set up the IM3016 and then the operator will perform any functions involving load tests. The user will normally wish to have one command line with all commands and then leave computer control.

Example: Load an Intel Hex file, then set up for correct TYPE and Set Size.

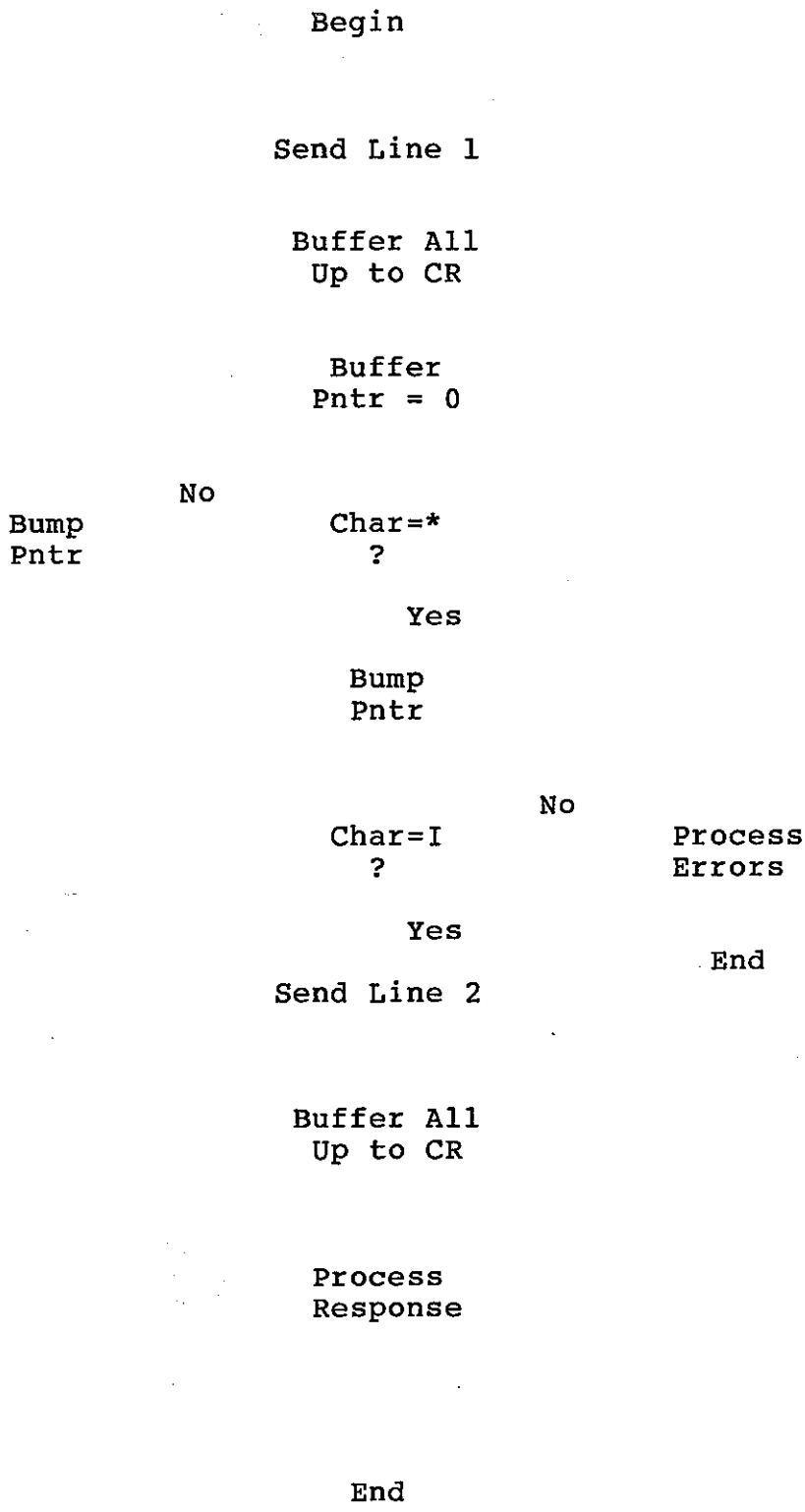
1. LO:0A TY:113 SE:16 MA:QU:CR

The above command line will load an Intel Hex formatted file, set the TYPE to 113 (2732A's), the Set Size to 16, Map the sockets in the standard manner and then Quit computer control.

Suppose, however, that the user wishes to fill the IM3016 RAM with $FF before loading the File. In this case, the command lines and program might be as shown below.

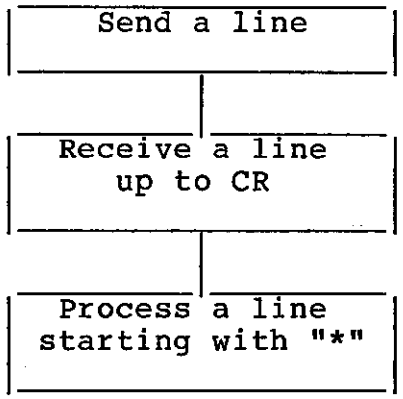
2. FI:FF LO:0A TY:113 SE:16:QU: CR

If error processing is desired, the flow chart should be modified as shown in Figure 1.


Begin
Send Line 1
Buffer All
Up to CR
Buffer
Pntr = 0
No
Bump
Pntr
Char=*
?
Yes
Bump
Pntr
No
Char=I
?
Process
Errors
Yes
End
Send Line 2
Buffer All
Up to CR
Process
Response
End

(FIGURE 1)

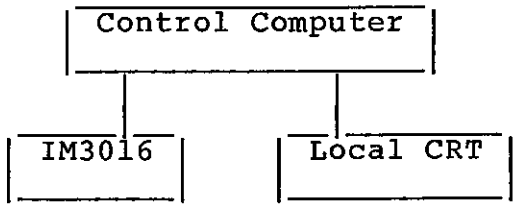
The last program shows an important concept. Namely, interactive programs should first buffer the IM response up to a CR; then process the response starting with an "*".



Failure to start processing with the "*" has been known to cause problems especially when the DC1/DC3 protocol is being used.

12.2 STEP 2: COMPUTER SETUP AND RUN MODE CONTROL AND MONITORING

Once Step 1 is mastered, the controlling computer can also determine how many PROMs of a given type need to be programmed. The hardware set up is assumed to be as below.



The user may desire to use the local CRT to enter a job number which calls up a program to determine how many and what type of PROMs need to be programmed, etc. The necessary information needed by the IM3016 is as follows.

1. Set up TYPE, Set Size, Map the Sockets to the RAM.
2. Load Data.
3. Enter RU command with total pass/fail responses.
4. Process pass/fail info for correct number of PROMs.
5. Go on to the next job.

The above extension from Step 1 is relatively easy vis-a-vis the IM3016. All that has to be mastered is processing the RU command responses. The work left to the user is in the job set up and initial operator interaction.

12.3 STEP 3: COMPLETE COMPUTER CONTROL

The IM3016 can be controlled to such an extent that all functions including those involving Load Tests are initiated by the computer. In this mode, the operator is prompted by a local terminal to perform only insertion and extraction of PROMs. IM has a C program available for installation on suitable computers which supports this mode of operation. For further details, consult the factory.

12.4 DETAILED DESCRIPTION OF COMPUTER COMMAND RESPONSE

The IM3016 will respond to ^Q ($11) and ^S ($13) if Bit 5 in the control word is set. Thus, the control word "$A0" will cause the IM3016 to recognize a ^S ($13) as "Stop Transmission" and a ^Q ($11) as "Start Transmission". This will <u>not</u> cause the IM3016 to transmit ^Q and ^S, however.

```
Example:  CO:80 <CR>      Response: *I>CRLF
          TY:111 ^S <CR>            *I>CRLF
          CO:A0 ^S <CR>             *I>CRLF
          TY:111 ^S <CR>            (no response)
             ^Q                     *I>CRLF
```

SECTION 13: COMMAND DEFINITIONS

13.0 COMMAND DEFINITIONS

The command definitions are presented in the following format.

Command letters (Command number) Brief Description

Command Syntax

Command Response

Typical Errors

Command Description

Each command has a unique number given in ( ) after its letters. If no response other than *I> or a Syntax Error is possible, the Response and Error information is omitted.

The following definitions use the nomemclature given below:

A	ASCII letter
D,E,F,G,H	Decimal digit
W,X,Y,Z	Hexadecimal ASCII digits
CR	ASCII carriage return $0D
LF	ASCII line feed $0A
ESC	Escape $1B
;	Command Separator, ASCII semicolon ";"
,	Command Parameter Separator, ASCII comma ","
:	Separator between Command Name and Command parameters, ASCII colon ":"
.	Used by Display Command for SP display, ASCII "."
...	Indicates repeated Command Parameters
$	Hex value follows
#	Command Terminator for variable length commands, ASCII pound "#"
DC3	ASCII $13,^S
DC1	ASCII $11,^Q

CA (1) CANCEL COMMAND LINE

CA:

The CA command is used to cancel all previous entries in the command line. It performs the same function as an entry of ESC.

CO (2) CONTROL BYTE DEFINITION

CO:XX

D8 of XX = Computer Control
D7 = Enable Prompt *I>
D6 = DC1/DC3 Enable
D5 = Enable Diagnostics
D4 = Enable LF after CR
D3 = Ignore Keyboard and Operator
D2 = Suppress Display
D1 = Enable Full Duplex

The CO command should usually be the first command executed. Bit D8 is always set. Upon initial entry into the computer control mode, the control byte is set to $80. Subsequent entries into computer control will restore the last user-determined value. Typical values for terminal operation are $C8 for half duplex and $C9 for full duplex. It is recommended that terminal operation use full duplex.

DA (4) DATA WIDTH CONTROL

DA:08
DA:16

08 for 8-bit data
16 for 16-bit data

The DA command affects the execution of two other functions; the MA: command and the LO: command. Refer to Section 7 if a DA:16 command is to be used.

DI (5) DISPLAY MESSAGE

DI:AAAAAAAA

A = Upper case letter, number, or "."

If A = ".", a space is displayed.There should always be eight values sent.

DU (6) **DUMP A FORMATED FILE**

```
DU:WW,XXXXX,YYYYY
```

W = Format Number
X = Start Address
Y = End Address

```
Errors - *?>
         *?>
```

W is not odd or is less than $0B.
(XXXX & $FFE0) => (YYYYY | $1F)

Data is dumped in $20 byte blocks. Thus the start address XXXXX is ANDed with $FFFE0 and the end address YYYYY is ORed with $0001F. The format number W is expressed in hex which differs from the decimal keyboard entry. Thus a keyboard entry of 11 (Intel hex dump) is expressed as $0B.

FT (40) **FILTER DATA LOADED INTO RAM**

```
FT:XXXXX,YYYYY
```

Load Data if Record Address
=> X, <= Y

For the Intel Hex files, only the first three nibbles of X and Y are significant. The address of the data being loaded with the LO: command must satisfy the below comparisons:

RRR00 => XXX00
RRRFF <= YYYFF

For Motorola(0CH), Extended Tek Hex(28H), HP Binary (1EH), and LDA Binary(20H) the FT command is interpreted differently.

With FT:X1234,X5678 (X is usually zero but is ignored)
The Address of the data being loaded satisfies

ADD => 12340000 and ADD <= 5678FFFF

In other words, the FT command specifies an acceptable address within a 64K block.
The FT: command is good for only one load. After each execution of a LO: command, the filter parameters are cleared.

FU (8) **DEFINE RUN FUNCTION**

```
FU:AAAAAA
FU:A...#
```

A = B for Blank Check
P for Program
V for Verify
C for Display Set Checksum
E for EEProm Erase

The FU command has a variable length data field. Unless six operations are given, the command must be terminated with a '#'.

LI (9) LIGHT ERROR LIGHTS

LI:XY

If X = 1 Load Error light on
Y = 1 Blank Error light on

The LI command controls the Load Error light and Blank Error light. After these lights are turned on, the WA: command is called. Thus the command will not terminate unless the START or ENTER key is depressed.

LL (10) LIGHT LEDS OVER PROM SOCKETS

LL:DDDDDDDDDDDDDDDD

D = 0 Red on
D = 1 Green on
D = 2 None on

There must be exactly 16 entries per command. The first D corresponds to socket 1. This command resets the socket status to the values given.

LO (11) LOAD A FORMATTED FILE

LO:XX

XX = Format number

Errors - *?> X is not even or is less than $0A.

The format number is expressed in hex which differs from the decimal key entry. Thus, the computer command LO:0A corresponds to the keyboard entry SPEC 10.

If an FT: and or an OF: command has preceeded the LO: command, these conditional parameters are only good for one load.

The various errors in the load file which may produce a response of *E>XX are given in Appendix D.

LT (12) LOAD TEST

LT:

Response - *R>XX,XX,XX,XX,XX,XX,XX,XX,XX,XX,XX,XX,XX,XX,XX,XX

If Set (1),

D8 of X =	Inactive Soc Error	Socketed PROM in an Inactive Socket.
D7 =	Vcc Power Fail	Vcc Current fail when all sockets enabled.
D6 =	Vcc Soc Fail	Socket Vcc current is > 200ma.
D5 =	Soc Disable 1	Stuck bit at socket is high.
D4 =	Soc Disable 0	Stuck bit at socket is low.
D3 =	Soc Enable Fail	Open socket data pin but socket not empty.
D2 =	Soc Empty	
D1 =	Load Set Error	

The LT command is the most important command to understand if the user chooses to completely control the IM3016 from an external computer. All commands which read or program a socketed PROM must necessarily execute a load test. D3 of the Control Word (Ignore Keyboard and Operator) is temporarily forced true during this command. If an operation is desired on a particular socket such as Blank Check, the corresponding LT value must be 00. Load Test errors with bits D7, D6, D5, D4, D3, or D1 Set (1) can be considered hard errors in that any other command such as PB: will not be performed. If an operation is directed at a particular socket, such as PC:01, bits D8 and D2 will also prevent further processing.

MA (13) MAP RAM TO SOCKET IN STANDARD MANNER

MA:

The RAM sockets are mapped in the standard manner. Socket 1 is mapped to 00000, socket 2 is mapped to 0 + prom size, etc. up to the set size. Then this is repeated. Please refer to Section 4.8 for a complete explanation. Each socket mapped is activated. All non-mapped sockets are deactivated. For example, if the Set Size equals 3, socket 16 is deactivated.

ML (39) LIST ALL SOCKET MAPPINGS

ML:

Response -

```
01,XXXXX 02,XXXXX 03,XXXXX  etc.
09,XXXXX 10,XXXXX 11,XXXXX  etc.
```

MR (15) MAP RAM TO SPECIFIED SOCKETS

MR:DD,XXXXX;EE,YYYYY;...#

Socket D is mapped to XXXXX
Socket E is mapped to YYYYY

MR:01,01000;02,02000#

Socket 1 is mapped to 01000
Socket 2 is mapped to 02000

As each socket is mapped, it is activated. Since MR has a variable length data field, the command must be terminated with a "#". To map all 16 sockets uniquely, this command must be executed twice because the IM3016 has an 80-character buffer. The MR command is often used in conjunction with the MS command. To deactivate all sockets not defined in the MR: command, the user should issue a TY: command before using MR:.

MS (16) MAP SOCKETS TO SOCKETS

MS:DD,EE,...;FF,GG,...#

Where DD and FF are previously mapped sockets. Sockets EE, GG, etc., are mapped to the respective addresses of DD and FF.

The MS command has a variable length data field. Therefore, it must be terminated with a "#". Each socket mapped is activated.

OF (17) SET LOAD OFFSET BANK ADDRESS

OF:X,YYYY

X = Offset 64K bank
Y = Offset 64K address

The 64K record address specified during a format load is added to Y modulo 64K. X is added to the record bank address. Thus, there is wrap around in any given 64K bank.

This command is good for one load only. Thus, after every load, the offset parameters are set to 0.

PB (18) PROM BLANK CHECK

PB:

Response - *R>DD,EE
DD = Number of Sockets that Pass
EE = Number of Sockets that Fail

Errors - *E>01 Load Test Error

The command PB operates on the PROM sockets. Thus, a load test is executed before the blank check is actually performed. If a load test error occurs and D3 of the control word is not set, the operator must correct the load error before operation can continue. If D3 is set and an *E>01 is returned, then the socket load test errors may be found using the SL command. The sockets contributing to the totals in the *R> response may be found using the SS command.

PC (19) SOCKET CHECKSUM

PC:DD
DD is the socket to be checksummed

Response - *R>XXXX
XXXX is the simple sum of all eight bit data values

Errors - *E>01 Load Test Error
*E>06 Socket Not Active

The command PC operates on the PROM sockets. Thus, a load test is executed before the checksum is actually performed. If a load test error occurs and D3 of the control word is not set, the operator must correct the load error before operation can continue. If D3 is set and an *E>01 is returned, then the socket load test errors may be found using the SL command.

PE (20) ERASE EEPROMS IN SOCKETS

PE:

Response - *R>DD,EE

DD = number of sockets successfully erased
EE = number of sockets which failed to erase

Errors - *E>01 Load Test Error

The command PE operates on the PROM sockets. Thus, a load test is executed before the erase function is actually performed. If a load test error occurs and D3 of the control word is not set, the operator must correct the load error before operation can continue. If D3 is set and an *E>01 is returned, then the socket load test errors may be found using the SL command. The sockets contributing to the totals in the *R> response may be found using the SS command.

PL (21) PROM LIST

PL:DD,XXXX

D = Socket #
X = PROM Start Address

Response - *R>YYYY;ZZ,ZZ,...

Y = Address
Z = Data

Errors - *E>01 Load Test Error
***E>06** Socket Not Active

Address X is ANDed with $FFF0 and data is listed in $10 byte blocks. For terminal operation, the listing will stop on any character entered and restart on the next character entered. The entry of ESC will terminate the listing. Every 256 byte data block boundary has an extra *R>.

The command PL operates on the PROM sockets. Thus, a load test is executed before the List PROM function is actually performed. If a load test error occurs and D3 of the control word is not set, the operator must correct the load error before operation can continue. If D3 is set and an *E>01 is returned, then the socket load test errors may be found using the SL command. The sockets contributing to the totals in the *R> response may be found using the SS command.

PM (22) **PROM MOVE INTO RAM**

PM:

Errors - *E>01 Load Test Error

The PM command performs in the same manner as a keyboard MOVE command. See Section 4.8 for a full explanation. The command PM operates on the PROM sockets. Thus, a load test is executed before the Move PROM into RAM function is actually performed. If a load test error occurs and D3 of the control word is not set, the operator must correct the load error before operation can continue. If D3 is set and an *E>01 is returned, then the socket load test errors may be found using the SL command. The sockets contributing to the totals in the *R> response may be found using the SS command. A complete Set of PROMs must be active otherwise a Load Test Error will occur.

PP (23) **PROGRAM PROMS**

PP:

Response - *R>DD,EE

D = number programmed correctly
E = number not programmed correctly

Error - *E>01 Load Test Error

The command PP automatically performs a post verify of all programmed PROMs. The result of this post verify is what is expressed in the *R> response. The command PP operates on the PROM sockets. Thus, a load test is executed before the Program PROMs function is actually performed. If a load test error occurs and D3 of the control word is not set, the operator must correct the load error before operation can continue. If D3 is set and an *E>01 is returned, then the socket load test errors may be found using the SL command. The sockets contributing to the totals in the *R> response may be found using the SS command.

PS **PROM SIZE**

PS:

Response - *R>DDDDP

D = size in hex of PROM

The PS command returns the hexadecimal value of the size of the current PROM type.

PT **PROGRAM PULSE TIME**

PT:DD,EE Where DD is Pulse Time in ms, EE in 0.1 ms. The Pulse Time is Reset to default time after any TY: command. Thus PT: must always follow a TY: command and never precede it.

QU (25) QUIT COMPUTER CONTROL

QU:

The QU command returns control to the IM3016 keyboard operator. The CO command word is saved and is automatically in force upon the next SPEC 2 key entry.

RC (3) RAM CHECKSUM

RC:XXXXX,YYYYY

X = Start Address of Checksum
Y = End Address of Checksum

Response - *R>ZZZZ

Z = Simple Sum of all 8-bit bytes from X to, and including, Y.

RF (7) FILL ALL OF RAM

RF:XX

XX = Fill Byte

The RF command allows clearing RAM before a LO: command is executed.

RL (27) RAM LIST

RL:XXXXX

X = RAM Start Address

Response - *R>YYYYY;ZZ,ZZ,...

Y = RAM Address
Z = RAM Data

Address X is ANDed with $FFF0 and data is listed in $10 byte blocks. For terminal operation the listing will stop on any character entered and restart on the next character entered. The entry of ESC will terminate the listing. Every 256 byte data block boundary has an extra *R>.

RM () RAM MOVE

RM:XXXXX,YYYYY,ZZZZ

X = RAM Start Address
Y = RAM Destination Address
Z = Number of Bytes - 1

RM moves the specified number of bytes from the RAM starting address to the RAM destination address.

RP (28) **RESET RAM PARITY**

RP:

If a response *P> is ever returned by the IM3016, a RAM Parity Error has occurred. This may be caused by the user trying to access IM3016 RAM which does not exist or some hardware error. In any event, a RP command should be executed before further processing and possibly other remedial action taken. The implication of a parity error is the the IM3016 buffer RAM data is possibly corrupted.

RR (26) **REPLACE RAM DATA**

RR:XXXXX,YYZZYYZZ...#

X = RAM adress
Y,Z = RAM data

The RR command has a variable length data field. Therefore, it must be terminated with a "#".

RU (29) **ENTER RUN MODE**

RU:X,DD

If X = Odd, pass/fail responses are returned. D number of cycles are performed.

The RU command performs as from the keyboard with the addition of returned responses and multiple executions. Early termination can be requested by transmission of a ESC. D3 of the control word is temporarily reset. Upon termination of the Run command the original value of D3 is restored.

SA (30) **SOCKET ACTIVE STATUS RETURNED**

SA:

Response - *R>D,D,D,D,D,D,D,D,D,D,D,D,D,D,D,D

D = 0 Socket is Inactive
D = 1 Socket is Active

The SA command reflects the current ACTIVE status of all sockets. A socket with a '0' status can not be acted upon.

SE (31) **SET SIZE**

SE:DD

DD = 01 to 16

The Set Size is determined. No other parameters are affected.

SL (32) LOAD TEST STATUS

SL:

Response - *R>XX,XX,XX,XX,XX,XX,XX,XX,XX,XX,XX,XX,XX,XX,XX,XX

If Set (1),

D8 of X =	Inactive Soc Error	Socketed PROM in an Inactive Socket.
D7 =	Vcc Power Fail	Vcc Current fail when all sockets enabled.
D6 =	Vcc Soc Fail	Socket Vcc Current is > 200ma.
D5 =	Soc Disable 1	Stuck bit at socket is high.
D4 =	Soc Disable 0	Stuck bit at socket is low.
D3 =	Soc Enable Fail	Open socket data pin but socket not empty.
D2 =	Soc Empty	
D1 =	Load Set Error	

The current load test status is returned. However, no load test is run as in the LT command. The individual load tests are sequentially run. If any failure occurs, the test is terminated.

SM (33) MACHINE STATUS

SM:

Response - *R>TY:DDD,SE:DD,RAM Size, Software Version No.

SS (35) SOCKET STATUS

SS:

Response - *R>SS:D,D,D,D,D,D,D,D,D,D,D,D,D,D,D,D

D = 0 Socket Failed
D = 1 Socket Passed
D = 2 Socket Inactive

The SS command is used to determine which sockets contributed to the totals in a previous pass/fail response.

TN () **TRANSMISSION TRAILING NULL DEFINITION**

TN:DD D = Number of Trailing Nulls

The TN command specifies the number of trailing nulls necessary for the IM3016 to properly transmit information to the host computer.

TY (36) **TYPE DEFINITION**

TY:DDD

D = Type Number

Execution of the TY command sets all sockets inactive. It is the only command capable of this function.

VE (24) **VERIFY PROMS AGAINST RAM**

VE:

Response - *R>DD,EE

D = Number Passed
E = Number Failed

The command VE operates on the PROM sockets. Thus, a load test is executed before the verify function is actually performed. If a load test error occurs and D3 of the control word is not set, the operator must correct the load error before operation can continue. If D3 is set and an *E>01 is returned, then the socket load test errors may be found using the SL command. The sockets contributing to the totals in the *R> response may be found using the SS command.

WA (37) **WAIT ON START KEY**

WA:

Regardless of D3 in the Control Byte, the START key light is lit and execution is commenced after depression of a START or ENTER key.

APPENDIX A : PROM TYPE CONVERSION TABLE

TYPE	PROM	ARRAY	PINS	IM MOD
310	2704	512x8	24	3V
311	2708	1024x8	24	3V
110	2758	1024x8	24	1V
312	TMS2716	2048x8	24	3V
111	2716	2048x8	24	1V
112	2732	4096x8	24	1V
113	2732A	4096x8	24	1V
114	2532	4096x8	24	1V
120	68764/66	8192x8	24	1V
121	2764	8192x8	28	1V
122	2564	8192x8	28	1V
130	2528	16384x8	28	1V
124	2764	8192x8 (Intel Fast Algorithm)	28	1V
131	27128	16384x8 (Intel Fast Algorithm)	28	1V
132	27256	32768x8 (Intel Fast Algorithm)	28	1V
133	2764A	8192x8 (Lower VPP)	28	1V
134	27128A	16384x8 (Lower VPP)	28	1V

APPENDIX A : PROM TYPE CONVERSION TABLE

APPENDIX B : SERIAL SWITCH SETTINGS

BAUD RATE	POS 8	POS 7	POS 6
110	X	X	X
150	O	X	X
300	X	O	X
1200	O	O	X
2400	X	X	O
4800	O	X	O
9600	X	O	O
19.2K	O	O	O

X = ON

O = OFF

POSITION	5	4	3	2	1
USE	POWER ON MODE	PARITY ENABLE	PARITY	DATA BITS	STOP BITS
OFF	NORMAL	ON	EVEN	8 BITS	2 STOP
ON	EXT CNTL	OFF	ODD	7 BITS	1 STOP

COMMON SWITCH SETTINGS

EXAMPLE: 1200 BAUD RS232C, EVEN PARITY, 7 BITS (ASCII), 2 STOPS.

Switch posistions 2 and 6 are on. All others off.

EXAMPLE: 9600 BAUD RS232C, NO PARITY, 7 BITS (ASCII), 1 STOP.

Switch positions 1, 2, 4, and 8 are on. All others off.

APPENDIX C : ERROR MESSAGES

GENERAL ERROR MESSAGES

DISPLAY	ERROR CONDITION
TYPE ERR	No TYPE defined
MOVE ERR	Wrong Set Size or Bad Buffer RAM
E PARITY	Bad Buffer RAM
BAD VP	Bad PROM; Vp current limit during program
BAD VER	Bad PROM; Initial Program Verify Error

LOAD ERROR MESSAGES

NONE ACT	Socket Flippers up or wrong PROM TYPE
VCC ERR	Upside down PROM
OPEN PIN	Open data line; Misaligned PROM
STUCK 1	Bad PROM; Misaligned PROM
STUCK 0	Bad PROM; Misaligned PROM
LSET ERR	Incomplete Set in RUN Mode; Misaligned PROM

FORMATTED I/O ERRORS

ECXXXXYY	Checksum error;X=Address, Y=Checksum
ERECXX	Bad Record Type; X=Type recognized
EHEX	Non-ASCII Hex Character recieved
ETAG	Invalid Tag recieved

APPENDIX D: ERROR CODES

ERROR	EXPLANATION
01	Load Test Error
02	Line Overrun
03	Unknown Command
05	Function Error
06	Non Active Socket
07	Unknown TYPE
08	MOVE Error - Bad RAM or too many Masters
09	Known Command But Bad Data
11	Exit during programming
12	Bad VP during Programming
13	Verify Error at beginning of Programming
32	Load Checksum Error
33	Load Record Error
34	Load bad Hexadecimal Number
35	Load Tag Error
36	Load Error undefined

LIMITED WARRANTY

International Microsystems, Inc. (Seller) warrants that all equipment sold pursuant to any resultant agreement shall be free from defects in material or workmanship at the time of delivery. Such warranty shall extend for the period of one (1) year from date of shipment except zero insertion force PROM sockets.

Buyer must notify Seller within the above prescribed warranty period of any defect. Seller shall either replace or repair the defective part or parts of equipment or replace the equipment or refund the purchase price at Seller's option after return of such equipment by Buyer to Seller. During the first ninety (90) days of the warranty period, Buyer ships one way and Seller will pay return shipping costs. After ninety (90) days, Buyer pays for shipping to Seller and incurs a $20 shipping and handling charge or the shipping costs, whichever is greater for return of the goods. The liability of Seller hereunder shall be limited to replacing or repairing, at its option, any defective units which are returned FOB Seller's plant. Equipment or parts which have been subject to abuse, misuse, accident, alteration, neglect, unauthorized repair or installation are not covered by warranty. In no event will Seller be liable for special or consequential damages as a result of any alleged breach of this warranty provision. As to items repaired or replaced, the warranty shall continue in effect for the remainder of the warranty period or for ninety (90) days following date of shipment by Seller of the repaired or replaced part, whichever period is longer.

No liability is assumed for expendable items such as lamps or fuses. No warranty is made with respect to custom equipment or products produced to Buyer's specification except as specifically stated in writing by Seller and contained in the contract. The warranty set forth herein is the only warranty, oral or written, made by Seller on any of its standard products and is in lieu of or replaces all other warranties, expressed or implied. Programmable device manufacturers occasionally change programming algorithms. Seller accepts no liability for programming equipment changes necessitated in connection therewith.