MOSTEK®

Z80 MICROCOMPUTER SYSTEM

Operations Manual

BIOS-80
BASIC INPUT/OUTPUT SERVICES
Z80 SOFTWARE PACKAGE

MOSTEK BIOS-80
OPERATION MANUAL

Publication Number MK79784

TABLE OF CONTENTS

TABLE OF CONTENTS

TABLE OF CONTENTS

TABLE OF CONTENTS

TABLE OF CONTENTS

TABLE OF CONTENTS

TABLE OF CONTENTS

PARAGRAPH NUMBER TITLE PAGE NUMBER

TABLE OF CONTENTS

SECTION 1

GENERAL DESCRIPTION

1.1 INTRODUCTION

BIOS-80 is the MOSTEK Basic Input/Output Services Z80 software package. It provides a common set of routines to facilitate the design of I/O drivers as well as includes the I/O drivers for terminal, printer, and floppy disk. All of the BIOS-80 modules are designed to operate under MITE-80; the MOSTEK Multiple Independent Task Executive Z80 software package. The BIOS-80 package supplements MITE-80 in that the routines and drivers provided are designed for real-time multiple asynchronous applications.

1.1.1 FEATURES

The highlighted features of BIOS-80 are:

Terminal Driver
Printer Driver
Floppy Disk Driver
Common routines available for user developed drivers
Compatible with MITE-80

1.1.2 SOFTWARE CONFIGURATION

The various modules of BIOS-80 are designed to work with specific hardware as follows:

For Terminal Driver: MDX-EPROM/UART Card and one CTC Channel

(MK3883), available on an MDX-CPUX card, interfaced to a Hazeltine or equivalent terminal.

For Printer Driver: MDX-PIO Card interfaced to a Centronics or equivalent type printer.

For Floppy Disk Driver: MDX-FLP Card interfaced to a Shugart or equivalent type disk drive.

All of the above drivers operate under MITE-80 and require the MDX-CPU Card plus their respective memory requirements.

1.2 REFERENCE DOCUMENTS

MITE-80 Operation Manual	MK79726
MDX-CPU1 Card Operation Manual	MK79612
MDX-CPU2 Card Operation Manual	MK79711
MDX-EPROM/UART Card Operation Manual	MK79604
MDX-PIO Card Operation Manual	MK79606
MDX-FLP Card Operation Manual	MK79639
FLP-80DOS Operation Manual	MK78557
Micro Components Data Book	MK79801

1.3 DEFINITIONS OF SYMBOLS USED

The following conventions are used throughout this manual:

Most hexadecimal numbers are identified by the character 'H' following the hexadecimal numbers.

aaaa indicates any hexadecimal number.

Operator entries are indicated by underlined characters in string entries.

1.4 PRODUCT OVERVIEW

The BIOS-80 software package is provided on a floppy diskette in IBM 3740 single density format. The files can be read using the MOSTEK FLP-80DOS software package. The BIOS-80 software package contains the following files:

Terminal Driver
Printer Driver
Floppy Disk Driver
BIOS Routines
I/O Macros

A brief overview of each file follows:

1.4.1 TERMINAL DRIVER

The Terminal Driver provides asynchronous input and output interface between user tasks operating under MITE-80 and an operator's terminal interfaced to a MDX-EPROM/UART Card. The Driver operates as a task under MITE-80 and can accommodate one or multiple devices. Various parameters are user configurable which allows specifying unique characteristics for each terminal.

1.4.2 PRINTER DRIVER

The Printer Driver provides output control between user tasks operating under

MITE-80 and a printer interfaced to a MDX-PIO Card. The Driver operates as a task under MITE-80 and can accommodate one or multiple printer devices. Various Driver parameters are user configurable which allows specifying unique characteristics for each printer.

1.4.3 FLOPPY DISK DRIVER

The Floppy Disk Driver provides input and output control between user tasks operating under MITE-80 and a floppy disk drive interfaced to a MDX-FLP Card. The Driver operates as a task under MITE-80 and can accommodate one or multiple disk drives. The Driver can handle the standard 8 inch and the mini 5.25 inch disk drives. Various Driver parameters are user configurable which allows specifing unique characteristics for each disk drive.

1.4.4 BIOS ROUTINES

The BIOS routines provide a common set of I/O services which can be used to facilitate the design of user developed drivers. The routines provide the functions that all drivers need, such as user buffer pointer maintenance, hardware initialization and character interpretation. All of the routines are re-entrant and can be used by as many Driver tasks as required by the application.

1.4.5 I/O MACROS

A set of I/O macros is provided to assist in constructing the Driver's Task Control Block along with their unique device characteristics. A general purpose I/O task macro is also included for constructing the mandatory TCB parameters on user developed Drivers.

1.4.6 DOCUMENT FORMAT

The following sections detail the areas of functional description, setting up the configuration, and use of each of the BIOS modules.

SECTION 2

BASIC INPUT/OUTPUT SERVICES

2.1 INTRODUCTION

This section outlines the functional characteristics and user interface to the Basic Input/Output Services (BIOS). BIOS provides a common set of routines which can be used to facilitate writing I/O drivers. The MOSTEK supplied drivers for Terminal, Printer, and Floppy Disk all use many of the BIOS routines. A memory savings can be realized by using the BIOS functions since common functions such as buffer pointer maintenance and special character interpretation (carriage return, line feed, tab) are required by many I/O drivers and are centralized in BIOS.

2.1.1 FEATURES

The features provided by BIOS are:

Input and output buffer maintenance
Special character interpretation
System or user error recovery
Common routines for service of inputting and outputting data
Common routines for services such as device open and interrupt handling

2.1.2 CONFIGURATION

The BIOS module requires about 820 bytes of program memory and can reside anywhere in the Z-80 address range.

2.2 FUNCTIONAL DESCRIPTION

2.2.1 OPERATION

BIOS operates as a set of task callable subroutines. All BIOS routines are re-entrant and can be used by as many tasks as required of the application. BIOS works off of data provided in the calling task's Task Control Block and the Message Block currently being serviced by that task.

BIOS provides seven callable routines for accessing the common services and are:

IO?WHY	Command decode
IO?GNC	Get next character, for output
IO?PNC	Put next character, for input
IO?ISR	Interrupt service handling
IO?OPN	Open device
IO?EOB	End of Block Routine
IO?ILL	Illegal command processor

Each of these services are described in detail in the following sections.

2.2.2 IO?WHY SERVICE

2.2.2.1 FORMAT

```
LD      IX,<Driver TCB Address>
LD      HL,<Message Block Address>
CALL    IO?WHY
JR      <Get Next Request>-$
DEFB    <N>                     ;Number of Valid Requests
JR      <Open Routine>-$        ;REQUEST 0
JR      <Close Routine>-$       ;REQUEST 1
JR      <Read Routine>-$        ;REQUEST 2
JR      <Write Routine>-$       :REQUEST 3
.
.
.
JR      <Request N-1 Routine>-$ ;REQUEST N-1
```

2.2.2.2 DESCRIPTION

The IO?WHY Service provides the command decode functions for BIOS designed drivers. This service is the first BIOS service called by the driver task. IO?WHY determines why the driver task is requested by decoding the Message Block's I/O service request code. Once the service request code is known, BIOS will give CPU control to the appropriate driver routine that handles the code.

The above format is part of the first series of program instructions that is executed by the Driver. Before calling the service the IX register is set-up with the address of the Driver's TCB and the HL registers are set-up with the address of the Message Block to be processed. Immediately following the service call is a relative jump to the point in the Driver where the next Message Block is retrieved for processing. The cycle of IX and HL register set-up and call to IO?WHY is repeated within this program loop. The DEFB value is the number of valid service requests handled by the Driver. Following this value is a relative jump to the appropriate Driver's entry point for each of the service requests handled by the driver. This relative jump table must be in sequential order by service request code number.

The BIOS IO?WHY service saves the address of the current Message Block, option flags, and address and length of the current user buffer for all the other BIOS services. The IO?WHY Service sets-up the DE registers with the address of the current position in the user buffer prior to entry into any of the driver service request routines.

2.2.3 IO?GNC SERVICE

2.2.3.1 FORMAT

```
        LD      DE,<address of current position in user buffer>
        LD      IX,<address of Driver TCB>
        CALL    IO?GNC
        JR      NZ,<end of record routine>-$
```

2.2.3.2 DESCRIPTION

The IO?GNC Service provides the "get next character" functions for BIOS output type drivers. The Service gets the next character from the user's buffer, maintains the user buffer pointer, and handles the various I/O options. Each of the options is described in detail in section 2.3.2.7 I/O Service Options (FLGS).

Before calling the service the DE registers are set-up with the address of the current position in the user's buffer and the IX register is set-up with the address of the Driver's TCB. These registers are set-up by the IO?WHY Service.

The IO?GNC Service will return in the A register the next character from the user's buffer to be outputted. The character retrieved from the user's buffer may not be the same character which is provided on return to the Driver for output. For example, when the EDIT option is specified and the character from the buffer is TAB (ASCII 09H), the Service will return BLANKS (ASCII 20H) until the next "tab stop" position is reached.

On return from the services the result of the service operation is determined by values in the A and the F registers. The following table defines the service operation results:

Z Flag	S Flag	Indicates
0	0	End of record reached, completed without error, completion code in A register
0	1	End of record reached, error code in A register
1	0	Character available in A register

If the Z Flag is set a character is available for output. Otherwise, either an end of record has been reached or an error has occurred.

2.2.4 IO?PNC SERVICE

2.2.4.1 FORMAT

```
        LD      DE,<address of current position in user buffer>
        LD      IX,<address of Driver TCB>
        LD      A,<character>
        CALL    IO?PNC
        JR      NZ,<end of record routine>-$
```

2.2.4.2 DESCRIPTION

The IO?PNC Service provides the "put next character" functions for BIOS input type drivers. The Service takes the character from the Driver and puts it in the next position in the user's buffer. The Service also maintains the user buffer pointer and handles the various I/O options.

Before calling the service the DE registers are set-up with the address of the current position in the user buffer, the IX register is set-up with the address of the Driver's TCB, and the A register is set-up with the character to be placed into the user's buffer. If the IX and DE registers are not altered by user code, these registers will have already been set-up from the IO?WHY Service.

The IO?PNC Service will take the character in the A register and place it in the user's buffer.

On return from the Service the result of the service operation is determined by the values in the A and F registers. The following table defines the service operation results:

Z Flag	S Flag	Indicates
0	0	End of record reached, completed without error, completion code in A register
0	1	End of record reached, error code in A register
1	0	Character was taken from A register

If the Z Flag is set the character was taken from the A register and placed into the user's buffer, ready for next character to input. Otherwise, either the last character which was returned caused end of record to be reached or an error has occurred.

2.2.5 IO?EOB SERVICE

2.2.5.1 FORMAT

```
        LD      A,<completion_code>
        JP      IO?EOB
```

2.2.5.2 DESCRIPTION

The IO?EOB Service provides the general end of block functions for an I/O operation. The routine places the completion code that is in the A register into the status byte of the MB, updates the USIZ field of the MB (if necessary), and either returns the MB to the caller or forwards it to the Error Handler task (if the SYStem Error recovery option is specified). IO?EOB then returns to the next instruction after the call to IO?WHY.

2.2.6 IO?ILL SERVICE

2.2.6.1 FORMAT

```
        JP      IO?ILL
```

2.2.6.2 DESCRIPTION

The IO?ILL routine provides a convenient entry point for handling illegal I/O requests (i.e. READ requests to the Line Printer driver). The routine simply loads the A register with the illegal operation error code and 'jumps' to IO?EOB.

2.2.7 IO?ISR SERVICE

2.2.7.1 FORMAT

```
        CALL    IO?ISR
        DEFB    TCBSTATUS               ; DRIVER TCB STARTS HERE
        DEFB    TCBPRIORITY
          .
          .
          .
```

2.2.7.2 DESCRIPTION

The IO?ISR Service provides the general housekeeping functions for an interrupt handler. The interrupts for BIOS type drivers are normally handled by this Service. When the interrupt vectors to this Service, the interrupted task's registers are pushed onto the task's stack, the IX register is set-up with the address of the driver's TCB, and a jump to MITE-80 Post Interrupt System Service (M8PINT) is performed. There are no user set-up requirements since each BIOS type driver's interrupt vector branches to this service whenever an interrupt occurs.

2.2.8 IO?OPN SERVICE

2.2.8.1 FORMAT

```
        LD      IX,<address of Driver's TCB>
        LD      C,<Driver's port address>
        CALL    IO?OPN
```

2.2.8.2 DESCRIPTION

The IO?OPN Service provides a general device open service for BIOS type drivers. The Service will set-up the driver's port interrupt vector in the specified port and sets up the vector in the vector page. On return from the Service the Driver's interrupt vector will have been set-up.

2.3 SERVICE REQUEST MESSAGE BLOCK

All task requests for driver services such as open, close, read, and write are all initiated by a separate Message Block sent from the requesting task to the appropriate Driver task. For BIOS type drivers a Message Block structure must be adhered to.

2.3.1 DATA STRUCTURE

The Message Block consists of a minimum 16 bytes and a maximum length determined by the application. The structure is:

Field	# of					
#	Bytes	Offset	Name	Value	Data Type	Source
1	1	0	STAT	Status	Bit encoded	Driver
2	1	1	PRIO	Priority	Binary	User
3	2	2	LINK	Link	Binary	MITE-80
4	2	4	RPTR	Rcvr. Ptr.	Binary	User
5	2	6	SPTR	Sender Ptr.	Binary	User
6	1	8	RQST	I/O Request	Binary & Bit encoded	User
7	1	9	FLGS	I/O Options	Bit encoded	User
8	2	10	BFFR	User buffer	Binary	User
9	2	12	USIZ	Byte count	Binary	User
10	2	14	RSIZ	Record count	Binary	User
11	N	16	DATA	Data	User defined	User

The purpose of Fields 1 thru 5 are identical to that described for a MITE-80 Message Block. Fields 6 thru 11 are unique to the BIOS Drivers and comprise the Message Block's Data field.

2.3.2 FIELD DESCRIPTIONS

The Message Block's field descriptions are described in detail in the following sections. The fields of STAT, PRIO, LINK, RPTR, and SPTR are summarized and additional information on these fields can be found in the MITE-80 Operations Manual (MK79726).

2.3.2.1 Status (STAT)

The Status field is used by the Driver to return a coded response indicating the results of the service requested. The response codes are listed in Appendix B.

2.3.2.2 Priority (PRIO)

The Priority field is supplied by the user to specify the priority level of this Message Block.

2.3.2.3 Link (LINK)

The Link field is used by MITE-80 for linking the Message Block into a task's Message Block queue.

2.3.2.4 Receiver Pointer (RPTR)

The Receiver Pointer field is supplied by the user and specifies the receiving task's TCB address of where the Message Block is to be sent.

2.3.2.5 Sender Pointer (SPTR)

The Sender Pointer field is supplied by either the user or MITE-80 and specifies the sending task's TCB address.

2.3.2.6 Service Request (RQST)

The Service Request field is supplied by the user and specifies the type of I/O service operation to be performed in bits 4-0, bits 7-5 are available to the driver for additional options. BIOS provides common routines which handle four different service requests. A User defined service request can be added to the service request list as application need requires (e.g. status, rewind). However, the BIOS service requests must be assigned the following codes:

Code	Service	Service Request
00H	Open	open the device for I/O service use
01H	Close	close the device from any further I/O service use
02H	Read	read data from an input device to the user's buffer
03H	Write	write data to an output device from the user's buffer

Before the first read or write operation is requested, an Open Service Request must be sent to the Driver Task. An error code will be returned to the calling task if either an I/O operation is requested without the Driver Task first being issued an Open request, or the request code is an illegal service request for this Driver.

2.3.2.7 I/O Service Options (FLGS)

The Service Options field is supplied by the user and specifies the options to be used in performing the I/O service request of the RQST field. The options available to each BIOS type Driver are:

LABEL	BIT	VALUE
SYSE	7	1 = System error recovery
		0 = User error recovery
EDIT	6	1 = Edit mode
		0 = No edit mode
BLCK	5	1 = Block mode
		0 = No block mode
FOLD	4	1 = Fold long line
		0 = Don't fold long line
NPAR	3	1 = Strip parity bit
		0 = Leave parity bit as is
ONCR	2	1 = Don't detect carriage return as end of operation
		0 = Detect carriage return as end of operation
----	1-0	XX = Driver dependent options

Each of the bits are further defined as follows:

SYSE (Bit 7) - If set (1), any errors detected by a driver will cause appropriate device dependent error recovery procedures to be executed by the driver. If the procedures fail, the error condition is reported to the operator as well as the requesting task. If reset (0), any errors detected will be reported to the requesting task, and any error recovery procedure is the responsibility of the requesting task.

EDIT (Bit 6) - If set (1), the interpretation of certain ASCII characters is performed; End of Text, Backspace, Tab, Line Feed, Form Feed, Control-U, and Rubout. If reset (0), there is no interpretation of these characters. When set, the interpretation of each character is as follows:

ETX (03H) - for both read and write operations, interpreted as an end-of-record indicator.

BS (08H) - on read operations, will cause buffer pointer to be repositioned backward one character position. On write operations, the character is output.

HT (09H) - on write operations, cause 8 - (N mod 8) spaces to be output, where N is the current buffer pointer. On read operations, the character is not interpreted by BIOS, but may be interpreted by a driver if required, such as a console driver.

LF (0AH) - on write operations, the character is output. On read operations, the character is ignored and not passed to the user's buffer.

FF (OCH) - on write operations, causes N line feeds to be output, where N is the number of lines remaining on the current page. On read operations, the character is not interpreted.

NAK (15H) - on write operations, causes output of a three ASCII character sequence <, CR, LF. On read operations, causes the user's buffer pointer to be reset to it's initial position.

DEL (7FH) - on read operations, will cause buffer pointer to be repositioned backward one character position. On write operations, the character is output.

BLCK (Bit 5) - If set (1), inhibits end of record detection (such as End of Text and Carriage Return) and causes the transfer of the number of bytes specified by the record size field, RSIZ, limited by the DSIZ field of the driver's TCB. If reset (0), causes End of Text or Carriage Return character to terminate transfer operation and be processed as an end of record indication.

FOLD (Bit 4) - Interpreted only in write operations. If set (1), causes a Carriage Return and Line Feed characters to be inserted after every Device Size (DSIZ) character length is reached. For example, if outputting to a console device whose DSIZ is 80 characters, if the output exceeds 80 characters, a carriage return and line feed characters will be output after the 80th character, and then the remainder of the output will be transferred. If reset (0), no fold operation will occur.

NPAR (Bit 3) - If set (1), the parity bit will be stripped (set to a zero) for any character transferred. If reset (0), the parity bit position will remain intact.

ONCR (Bit 2) - If set (1), a detection of the Carriage Return character will be processed as an end of record indication. If reset (0), no detection of the Carriage Return character is performed.

NOTE: On all of the I/O Option (FLGS) bits, the operation will not be valid unless the Driver's I/O Option (BOPT) supported function bits are set to the corresponding Message Block's I/O Option (FLGS) request.

2.3.2.8 User Buffer (BFFR)

The User Buffer field is supplied by the user and specifies the starting address of the buffer to be used in the I/O operation. Data transfers will start at this buffer location and proceed to the next higher memory address, an increment buffer pointer process.

2.3.2.9 User Size (USIZ)

The User Size field is supplied by the Driver at the completion of an I/O operation. The field contains the actual number of bytes transferred during the requested service operation.

2.3.2.10 Record Size (RSIZ)

The Record Size field is supplied by the user and specifies the logical record size for this I/O transfer. The value in this field determines the maximum number of bytes which will be transferred to or from the user's buffer.

2.3.2.11 Data (DATA)

The Data field is supplied by the user and contains any other information required by the Driver from the requesting task. This field is optional and is dependent on each Driver's requirements.

2.4 I/O DRIVER TCB MACROS

To configure a Driver into a system the user must define the Driver's characteristics and provide for certain dependent modules. The Driver's characteristics are user specified in the Driver's TCB. The dependent modules required of the Driver are determined by the Driver characteristics specified.

To assist in constructing a Driver TCB, two macros are provided; MIOTCB and EIOTCB. The primary difference between these two macros is that the MIOTCB macro is for RAM based systems while the EIOTCB macro is for ROM/EPROM based systems. The latter macro will create executable code that will transfer the TCB from ROM to RAM, set-up the Driver's stack, allocate internal Driver required RAM area, and Link the TCB into the MITE-80 system. The MIOTCB does not create this executable code and the user, once the TCB is loaded into RAM, must link the TCB into the MITE-80 system.

Both macros have identical user-specifiable parameters with the exception of the EIOTCB macro having one additional parameter.

2.4.1 FORMAT

The formats of both macros are:

```
MIOTCB  1,2,3,4,5,6,7,8,9,10,11,12,13,14[,15,...,N]
EIOTCB  1,2,3,4,5,6,7,8,9,10,11,12,13,14,15[,16,...,N]
```

The parameter positions are defined as:

MIOTCB PARAM #	EIOTCB PARAM #	PARAM	PARAMETER DESCRIPTION	RANGE (HEX)	DEFAULT VALUE
-	1	ADDR	RAM Load address	0000-FFFF	Current Addr.
1	2	NAME	TCB name	2-3 char.	Error
2	3	TIME	Device	00-FF	No timer message
3	4	VECT	Device I/O Vector	00-FF	(NM)VECT*
4	5	PORT	Device I/O Port	00-FF	(NM)PORT
5	6	STAT	Device TCB Status	M or A	M
6	7	PRIO	Device priority	00-7F	(NM)PRIO
7	8	STKL	Stack length	00-FFFF	(NM)STKL
8	9	DTYP	Device type	Bit Encoded	(NM)DTYP
9	10	DSTA	Device status	Bit Encoded	(NM)DSTA
10	11	DOPT	Device Options	Bit Encoded	(NM)DOPT
11	12	BOPT	Device BIOS Options	Bit Encoded	(NM)BOPT
12	13	DSIZ	Device record size	00-FFFF	(NM)DISZ
13	14	DLEN	Device form length	00-FF	(NM)DLEN
14	15	DPAD	Device pad count	00-FF	(NM)DPAD
15 - N	16 - N		Add'l. parameters	User Defined	Omitted

* NM corresponds to the first 2 characters of the NAME parameter.

The above parameters must be defined in the specified order. If no value is specified for a parameter, the default value will then be used. A comma (,) must separate all parameter positions, including those parameters not elected to be defined.

The default values for most of the parameters represent a label whose first two characters are the user defined TCB NAME, such as PTVECT where PT was the user specified TCB name and VECT being the default for the vector parameter.

2.4.2 PARAMETERS

An error message will appear if any macro parameter is incorrectly specified. The parameters are further defined as follows:

2.4.2.1 Address (ADDR)

The starting memory address of where the TCB and stack is to be loaded. The memory area used is from the specified starting address and progresses toward the high end of memory.

2.4.2.2 Name (NAME)

The 2-3 character TCB identifier. The TCB name field will contain the first and last characters of this parameter. The default parameter labels will consist of the first 2 characters of this parameter followed by the parameter name.

2.4.2.3 Timer Value (TIME)

The I/O watchdog timer value for device. The timer value is usually specified in 50 millisecond ticks. The value specified determines the duration of time the device is allowed to wait for an I/O operation completion before an I/O timeout error occurs. For example, on a 120 line per minute printer, the timer value would be set above 500 milliseconds, possibly around 600-750 milliseconds. For timer values, refer to the Timer Handler Section of the MITE-80 Operations Manual. If the time value is omitted, no watchdog timer function is then specified.

2.4.2.4 Vector (VECT)

The interrupt vector displacement for this device. This value is loaded into the hardware's interrupt vector register.

2.4.2.5 I/O Port (PORT)

The port address for this device. For a device with multiple ports, this value is the lowest (first) port address.

2.4.2.6 Status (STAT)

The device's TCB status of register set usage; M for main set only or, A for full set.

2.4.2.7 Priority (PRIO)

The priority level at which this device is to execute.

2.4.2.8 Stack Length (STKL)

The length in bytes of the stack required by this device. The stack size must be large enough to accommodate the devices stack needs plus 16 or 24 bytes for saving the register set used in the event of an interrupt suspending this device task.

2.4.2.9 Device Type (DTYP)

The identification of the type of device that this Driver handles. The byte is bit encoded as follows:

Bit	Definition
7	1 = Disk device
	0 = No disk device
6	1 = Card Reader device
	0 = No Card Reader device
5	1 = Printer device
	0 = No Printer device
4	1 = Paper Tape device
	0 = No Paper Tape device
3	1 = Terminal device
	0 = No Terminal device
2-0	000 = Not used, reserved for future use

An example of DTYP field settings would be; for a teletype with keyboard, printer, and paper tape, the DTYP field would be 34H. For a CRT with keyboard and printer the DTYP field would be 24H.

2.4.2.10 Device Status (DSTA)

This field is used by a Driver to keep information on the current state of an I/O operation. The byte is driver dependent and is user defined with the exception of Bit 7, which is used by BIOS for an OPEN flag. This field can be pre-conditioned to a user defined state when the TCB is installed into the MITE-80 System.

2.4.2.11 Device Options (DOPT)

This field identifies the device options for this device. This field is user defined and is used to describe how the Driver is to handle I/O operations for this device, such as even or odd parity. The entire byte is user defined except for Bit 2 which is used to indicate if an I/O timeout feature is to exist with this driver. The field is defined as:

Bit	Definition
7-3	User defined
2	1 = I/O timeout feature exists 0 = No I/O timeout feature exists
1-0	User defined

2.4.2.12 BIOS Options (BOPT)

This field identifies the BIOS options supported by this Driver. The definition of this field is the same as the Message Block's I/O Service Options (FLGS) field. If the option is not set in the BOPT field, the Message Block's I/O Service Options request will not be handled for the corresponding option function. The BOPT byte is summarized here, refer to section 2.3.2.7 I/O Service Options (FLGS) for a detailed description:

B7 = SYSE, system error recovery
B6 = EDIT, edit mode
B5 = BLCK, block mode
B4 = FOLD, fold long lines
B3 = NPAR, parity bit disposition
B2 = ONCR, carriage return detect
B1-0 = not used, reserved for future use.

2.4.2.13 Device Size (DSIZ)

This field defines the maximum number of bytes, in binary, per record to be transferred by this device if both the IOTCB and the Message Block's I/O Service Options (FLGS) field's block option (BLCK) bit are set.

2.4.2.14 Device Length (DLEN)

This field defines the maximum number of output lines, in binary, on a page that this device is to handle.

2.4.2.15 Device Pad Count (DPAD)

This field defines the number of Null (ASCII 00H) characters, in binary, that this device is to output after each Carriage Return or Line Feed Character is outputted.

2.4.2.16 Additional Parameters

Any additional required parameters for this device follow DPAD. This field or fields is user defined. The specified parameters are placed at the end of the IOTCB in the order that they are specified.

2.4.3 MULTIPLE LIKE DEVICES

For a Driver designed to support multiple devices, each device must be identified to the MITE-80 system by a unique TCB. Each TCB can identify identical device characteristics but the TCB NAME, PORT, and VECT parameters must be unique. The IOTCB macros can be used to construct each device's TCB along with the device's required characteristics.

2.4.4 DEPENDENT MODULES

The driver must execute as a task under MITE-80. In addition to MITE-80, the driver optionally depends on the MITE-80 Timer Handler (M80TH). If the Driver's TCB "TIME" parameter is specified, then this timer module must be included in the user's system in order to provide the desired device time-out feature. The Timer

Handler must have a TCB named 'TH' in order for the driver's watchdog timer to function properly.

2.4.5 LINKING INTO A SYSTEM

The Driver should be designed to reside anywhere in the Z80 address range. The Driver is linked directly with the user programs. The Driver's TCB macro(s) should be grouped, assembled, and linked with all of the other task TCBs.

To link the Driver the following procedure is used:

```
        $LINK UPROG,UDRIV,BIOS,MITE80,M8TCBQ TO USYS
```

The above command string will link a user program (UPROG) and the Driver (UDRIV) into an executable load module of user system (USYS).

SECTION 3

TERMINAL DRIVER

3.1 INTRODUCTION

This section outlines the functional characteristics and user interface to the Terminal Driver. The Terminal Driver is designed to facilitate asynchronous input and output between user tasks operating under MITE-80 and an operator's terminal interfaced to a MDX-EPROM/UART Card.

3.1.1 FEATURES

The features provided by the Terminal Driver are:

Input/output control of terminal's keyboard and display.

Operates as a task under MITE-80.

User service requests queued by priority.

User type ahead capability.

Output alarm feature for pre-empt of current request.

User-configurable parameters for specifying unique terminal characteristics.

Accommodates one or multiple terminals.

3.1.2 CONFIGURATION

The Terminal Driver requires about 800 bytes of program memory and about 120 bytes of RAM. The RAM is for the Driver's task control block and stack requirements. The Driver is designed to operate with the MDX-EPROM/UART Card which is interfaced to a Hazeltine or equivalent type terminal. Refer to the Terminal Driver's Configuration Requirements Section for information on user configurable terminal parameters. The Driver requires one CTC Timer Channel (MK 3883), the MDX-CPU Card can provide this timer requirement.

3.2 FUNCTIONAL DESCRIPTION

3.2.1 OPERATION

The Terminal Driver operates as a task executing under MITE-80. User tasks communicate with the Terminal Driver by Message Blocks. The Terminal Driver provides four user service requests; open, read, write, and close. Each of the service requests are initiated by a separate Message Block issued from the user task. The Driver will return to the calling task a status response at the completion of the service request. With each service request the calling task has several options with which to direct the Driver's operation, such as edit mode, echo option, type of error recovery, and number of characters to transfer. All of these options are described in the Message Block Section.

The open request conditions the hardware for I/O requests. The read request allows characters to be read in from the terminal and placed into the user's buffer. The output request will write characters from the user buffer to the terminal. The close request will disable the hardware from any further operation until an open request is performed.

After an open request is issued, the Driver will accept characters from the terminal even if a read request has not been issued. When characters are inputted from the terminal prior to a read request, the Driver will place the characters in a type ahead buffer. The buffer's contents will then be transferred to the user's buffer at read request time. The length of the type ahead buffer is specified by the user at configuration time.

The Driver provides an alarm feature for emergency message output. When the alarm feature is used, the message will be immediately output to the terminal regardless of whether the Driver is currently in a read or write request operation. The alarm message will take priority over all other non-alarm messages in the Driver's Message Block queue. A bell character will be outputted with the alarm message to alert the operator.

3.2.2 ERROR HANDLING

Two options exist for the handling of errors which can occur within the Driver during request operations. The options are user specified within the Message Block at service request time. The options allow for the error to be handled by the system, or by the calling task. The system error handling option will output to the terminal the error code, and will also return the error code to the calling task. The calling task error handling option will cause the Driver to return the error code to the calling task for process disposition. In that case, the Driver will not output the error code to the terminal.

3.3 SERVICE REQUEST MESSAGE BLOCK

A Message Block is required for each terminal operation request of open, read, write, and close. Calling tasks must adhere to the Terminal Driver's Message Block structure in order to assure proper service request operation.

3.3.1 DATA STRUCTURE

The Terminal Driver's Message Block is 16 to 20 bytes in length and its structure is as follows:

Field #	# of Bytes	Byte offset	Name	Field	Data Type	Source
0	1	0	STAT	Status	Binary	Driver
1	1	1	PRIO	Priority	Binary	User
2	2	2	LINK	Link	Binary	MITE-80
3	2	4	RPTR	Receiver Pointer	Binary	User
4	2	6	SPTR	Sender Pointer	Binary	User
5	1	8	RQST	Service Request	Binary	User
6	1	9	FLGS	Service Options	Bit Encoded	User
7	2	10	BFFR	User Buffer	Binary	User
8	2	12	USIZ	Byte Count	Binary	Driver
9	2	14	RSIZ	Record Count	Binary	User
10	2	16	[PRMA]	Prompt Address	Binary	User
11	2	18	[PRMS]	Prompt Length	Binary	User

Fields 0 thru 4 are a MITE-80 Message Block. Fields 5 thru 9 are standard BIOS Message Block parameters. Fields 10 thru 11 are unique to the Terminal Driver and comprise the optional prompt field.

3.3.2 FIELD DESCRIPTIONS

The Message Block's field description are highlighted in the following sections. For the fields of STAT, PRIO, Link, RPTR, and SPTR, additional information can be found in the MITE-80 Operations Manual (MK79726). For the remaining fields additional information can be found in the BIOS Section of this Operations Manual.

3.3.2.1 Status (STAT)

The Status field is used by the Driver to return a coded response indicating the outcome of the service requested. The response codes are listed in Appendix B.

3.3.2.2 Priority (PRIO)

The Priority field is supplied by the user to specify the priority level of this Message Block. A priority level of 00H will indicate to the Driver that this message is an alarm Message Block.

3.3.2.3 Link (LINK)

The Link field is used by MITE-80 for linking the Message Block into a task's Message Block queue.

3.3.2.4 Receiver Pointer (RPTR)

The Receiver Pointer field is supplied by the user and specifies the receiver's TCB address to which the Message Block is to be sent.

3.3.2.5 Sender Pointer (SPTR)

The Sender Pointer field is supplied by either the user or MITE-80 and specifies the sender TCB address from which the Message Block originated. At the completion of a service request, the Driver will return the Message Block to the task whose TCB address is specified in this field.

3.3.2.6 Service Request (RQST)

The Service Request field is supplied by the user and specifies the type of I/O service operation to be performed. The requests supported by the Terminal Driver are:

Code	Service
00H	Open
01H	Close
02H	Read
03H	Write

Before the first read or write operation is requested, an Open service request code must be sent to the Driver. An error code will be returned to the calling task if either an I/O operation is requested without the Driver first being issued an Open request, or the request code is an illegal service request for this Driver.

3.3.2.7 Service Options (FLGS)

The Service Options field is supplied by the user and specifies the options to be used in performing the I/O service request of the RQST field. The options supported, along with their position in the FLGS byte, by the Terminal Driver are:

Label	Bit	Definition
SYSE	7	1 = System error recovery 0 = User error recovery
EDIT	6	1 = Edit mode 0 = No edit mode
BLCK	5	1 = Block mode 0 = No block mode
FOLD	4	1 = Fold long line 0 = Don't fold long line
NPAR	3	1 = Strip parity bit 0 = Leave parity bit as is
ONCR	2	1 = Detect Carriage Return as end of operation 0 = Don't detect carriage Return as end of operation
ECHO	1	1 = Echo input to output 0 = Don't echo
PRMT	0	1 = Prompt string in service request 0 = No prompt string

The ECHO and PRMT bits are further defined as follows; the remaining option bits are as described in section 2.3.2.7.

3.3.2.7.1 ECHO

The Echo option allows a task to echo all operator inputted characters to the terminal output device, such as CRT or printer. In most cases this ECHO option would be set as it allows an operator to see what he is typing. The option would be reset where output is not required, such as a password or other sensitive information.

3.3.2.7.2 PRMT - Prompt

The Prompt option allows a task to output a user specified string of characters before the input service request is performed by the Driver. Use of the prompt string option would include (a) line(s) of text for prompting operator input, and for general text to operator prior to operator input.

If the PRMT bit is set to a "1", the user must provide the PRMA and PRMS fields in the Message Block.

3.3.2.8 User Buffer (BFFR)

The User Buffer field is supplied by the user and contains the address of the starting location of the user's buffer to be used by the Driver in the I/O operation.

3.3.2.9 Byte Count (USIZ)

The Byte Count field is used by the Driver to return the actual number of bytes transferred during the requested I/O operation.

3.3.2.10 Record Count (RSIZ)

The Record Count field is supplied by the user and contains a binary value which specifies the maximum number of bytes allowed to be transferred to/from the user's buffer.

3.3.2.11 Prompt Address (PRMA)

The Prompt Address field is the 2 byte address of the location of the prompt string.

3.3.2.12 Prompt Size (PRMS)

The Prompt Size field is a 2 byte field that specifies in binary the number of characters in the prompt string.

3.4 CONFIGURATION REQUIREMENTS

To configure the Terminal Driver into a system the user must define the Driver's characteristics and provide for certain dependent modules. The Driver's characteristics are user specified in the Driver's TCB. The dependent modules required of the Driver are determined by the Driver characteristics specified.

3.4.1 Terminal TCB Macros

To assist in constructing a Terminal Driver TCB, two macros are provided; MTDTCB and ETDTCB. The primary difference between these two macros is that the MTDTCB macro is for RAM based systems while the ETDTCB macro is for ROM/EPROM based systems. The latter macro has executable code that will transfer the TCB from ROM to RAM, set-up the Driver's stack, allocate internal Driver required RAM area, and link the TCB into the MITE-80 system. The MTDTCB does not create this executable code and the user, once the TCB is loaded into RAM, must link the TCB into the MITE-80 system.

Both macros have identical user-specifiable parameters with the exception of the ETDTCB macro having one additional parameter.

3.4.2 Formats

The formats of both macros are as follows:

```
MTDTCB  1,2,3,4,5,6,7,8,9,10,11,12,13,14,15
ETDTCB  1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16
```

3.4.3 Parameters

The parameter positions are defined as:

MTDTCB PARM #	ETDTCB PARM #	PARM	PARAMETER DESCRIPTION	RANGE (HEX)	DEFAULT VALUE
-	1	ADDR	RAM load address	0000-FFFF	Current Address
1	2	NAME	TCB name	2-3 Characters	Error
2	3	TIME	Terminal timer count	00-FF	No timer message
3	4	VECT	Terminal I/O Vector	00-FF	(NM)VECT*
4	5	PORT	Terminal I/O Port	00-FF	(NM)PORT
5	6	PRIO	Terminal priority	00-7F	(NM)PRIO
6	7	DTYP	Terminal Device type	Bit Encoded	(NM)DTYP
7	8	DOPT	Terminal device option	Bit Encoded	(NM)DOPT
8	9	BOPT	Terminal BIOS option	Bit Encoded	(NM)BOPT
9	10	DSIZ	Terminal record size	00-FFFF	(NM)DSIZ
10	11	DLEN	Terminal form length	00-FF	(NM)DLEN
11	12	DPAD	Terminal pad count	00-FF	(NM)DPAD
12	13	TABL	Type ahead length	00-FF	(NM)TABL
13	14	TCRI	Input character rate	00-FF	(NM)TCRI
14	15	TCRO	Output character rate	00-FF	(NM)TCRO
15	16	CTCP	Terminal CTC port address	00-FF	(NM)CTCP

* NM corresponds to the first 2 characters of the NAME parameter.

The above parameters must be defined in the specified order. If no value is specified for a parameter, the default value will then be used. A comma (,) must separate all parameter positions, including those parameters not elected to be defined.

The default values for most of the parameters represent a label whose first two characters are the user defined TCB NAME, such as TDVECT where TD was the user specified TCB NAME and VECT being the default for the vector parameter.

An error message will appear if any macro parameter is incorrectly specified. Additional information on parameters 1-12 of MTDTCB and 1-13 of ETDTCB can be found in the BIOS Section of this Operations Manual.

The parameters are further defined as follows:

3.4.3.1 Address (ADDR)

The starting memory address of where this TCB and stack is to be loaded.

3.4.3.2 Name (NAME)

The 2-3 character TCB name.

3.4.3.3 Timer Value (TIME)

The I/O watchdog timer value for Terminal device.

3.4.3.4 Vector (VECT)

The terminal interrupt service routine vector address.

3.4.3.5 I/O Port (PORT)

The terminal I/O port assignment.

3.4.3.6 Priority (PRIO)

The priority level of the Terminal Driver task.

3.4.3.7 Device Type (DTYP)

Identifies the types of devices the terminal driver handles. BIOS bits 3 (Terminal), 4 (Paper Tape), and 5 (Printer) are the options available for the Terminal Driver.

3.4.3.8 Device Options (DOPT)

Identifies the device options supported by the Terminal Driver.

3.4.3.9 BIOS Options (BOPT)

Identifies the BIOS options supported by the Terminal Driver. This parameter's definition is identical to Service Options (FLGS), refer to Section 2.3.2.7. Each Message Block's Service Option Flag will not be executed if the corresponding BOPT flag is not set. The BOPT parameter describes what the Terminal Driver will support, regardless of the request in the Message Block's Service Options (FLGS) field. The BOPT byte is highlighted as follows:

B7 = SYSE, system error recovery
B6 = EDIT, edit mode
B5 = BLCK, block mode
B4 = FOLD, fold long lines
B3 = NPAR, parity bit dispositon
B2 = ONCR, carriage return detect

3.4.3.10 Device Size (DSIZ)

Maximum number of bytes per record. For the Terminal Device this will be the maximum number of characters per print/display line.

3.4.3.11 Device Length (DLEN)

Maximum number of lines on output device, printer or display.

3.4.3.12 Device Pad Count (DPAD)

Number of null characters to output after each carriage return or line feed character.

3.4.3.13 Type Ahead Buffer Length (TABL)

The maximum number of characters allowed to be entered into a type ahead buffer without a current read request.

3.4.3.14 Terminal Character Rate for Input (TCRI)

The character rate input timing value. Use the following table to find the value required for the specific user configuration if machine generated input is used. For normal typing input this parameter should be 255 to minimize overhead.

Baud Rate	TCRI for 2.5MHz	TCRI for 4.0MHz
110	148	237
300	163	87
600	163	131
1200	82	131
2400	41	66
4800	21	33
9600	11	17
19200	6	9

3.4.3.15 Terminal Character Rate for Output (TCRO)

The character rate output timing value. Use the table in the TCRI parameter definition to find value required for the specific configuration.

3.4.3.16 CTC Channel Port (CTCP)

The port address of the CTC channel assigned for use by the Terminal Driver.

3.5 MULTIPLE CONSOLE DEVICES

The Terminal Driver is designed to support multiple terminal devices. Each terminal device must be identified to the MITE-80 system by a unique TCB. Each TCB can specify identical terminal device characteristics but the TCB name, vector, and port addresses must be unique. The Terminal Driver TCB macros can be used to construct each terminal device's TCB along with the device's required characteristics.

3.6 DEPENDENT MODULES

The Terminal Driver executes as a task under MITE-80. In addition to MITE-80, the Driver requires the BIOS Module (BIOS) and the MITE-80 Timer Handler (M80TH). The Timer Handler must have at least one of its TCBs named 'TH' in order for the Driver to function properly.

3.7 LINKING INTO A SYSTEM

The Terminal Driver can reside anywhere within the Z80 address range and is provided in relocatable object form. The Driver can be linked directly with user programs.

To link the Terminal Driver the following procedure is used:

```
        $LINK UPROG,M80TD,BIOS,M80TH,MITE80,M8TCBQ TO USYS
```

The above command string will link a user program (UPROG) and the Terminal Driver (M80TD) into a executable load module of user system (USYS).

SECTION 4

LINE PRINTER DRIVER

4.1 INTRODUCTION

This section outlines the functional characteristics and user interface to the Line Printer Driver. The Line Printer Driver is designed to facilitate output between user task operating under MITE-80 and a printer device interfaced to a MDX-PIO Card.

4.1.1 FEATURES

The features provided by the Line Printer Driver are:

Output control for the Line Printer.

Operates as a task under MITE-80.

User service request queued by priority.

Accommodates multiple printers.

4.1.2 CONFIGURATION

The Line Printer Driver requires about 120 bytes of program memory and about 70 bytes of RAM. The RAM is for the Driver's task control block, timer message, and stack requirements. The Driver is designed to operate with the MDX-PIO Card, which is interfaced to a Centronics or equivalent type line printer. Refer to the Line Printer Driver's Configuration Requirements Section for information on user configurable printer parameters.

4.2 FUNCTIONAL DESCRIPTION

4.2.1 OPERATION

The Line Printer Driver operates as a task executing under MITE-80. User tasks communicate with the Line Printer Driver via Message Blocks. The Line Printer Driver provides three user service requests; open, close, and write. Each of the service requests are initiated by a separate Message Block issued from the user task. The Driver will return to the calling task a status response at the completion of the service request. The user task is responsible for determining the appropriate action per status response.

The OPEN request conditions the hardware to a known system state and readys the device for output. The WRITE request will output characters from the user's buffer to the line printer device. The CLOSE request will disable the hardware from any further operation until another open request is performed.

4.2.2 ERROR HANDLING

Two options exist for the handling of errors which can occur within the Driver during request operations. The options are user specified within the Message Block at service request time. The options allow for the error to be handled by the system, or by the calling task. The system error handling option will output to the terminal device the error code, and will also return the error code to the calling task. The calling task error handling option will cause the Driver to return the error code to the calling task for process disposition. In this case the Driver will not output the error code to the terminal device.

4.3 SERVICE REQUEST MESSAGE BLOCK

A Message Block is required for each line printer operation request of open, close, and write. Calling tasks must adhere to the Line Printer Driver's Message Block structure in order to assure proper service request operation.

4.3.1 DATA STRUCTURE

The Line Printer Driver's Message Block consists of a 16 byte structure as follows:

Field #	# of Bytes	Byte Offset	Name	Field	Data Type	Source
1	1	0	STAT	Status	Binary	Driver
2	1	1	PRIO	Priority	Binary	User
3	2	2	LINK	Link	Binary	MITE-80
4	2	4	RPTR	Receiver Pointer	Binary	User
5	2	6	SPTR	Sender Pointer	Binary	User
6	1	8	RQST	Service Request	Binary	User
7	1	9	FLGS	Service Options	Bit	User
8	2	10	BFFR	User Buffer	Binary	User
9	2	12	USIZ	Byte Count	Binary	Driver
10	2	14	RSIZ	Record Count	Binary	User

Fields 1 thru 5 are a MITE-80 Message Block. Fields 6 thru 10 are standard BIOS Message Block parameters used by the Line Printer Driver and comprise the Message Block's Data Field.

4.3.2 FIELD DESCRIPTIONS

The Message Block's field descriptions are highlighted in the following sections.

For the fields of STAT, PRIO, LINK, RPTR, and SPTR, additional information can be found in the MITE-80 Operations Manual (MK79726). For the remaining fields additional information can be found in the BIOS Section of this Operations Manual.

4.3.2.1 Status (STAT)

The Status field is used by the Driver to return a coded response indicating the outcome of the service requested. The response codes are listed in Appendix B.

4.3.2.2 Priority (PRIO)

The Priority field is supplied by the user to specify the priority level of this Message Block.

4.3.2.3 Link (LINK)

The Link field is used by MITE-80 for linking the Message Block into a task's Message Block queue.

4.3.2.4 Receiver Pointer (RPTR)

The Receiver Pointer field is supplied by the user and specifies the receiver's TCB address to which the Message Block is to be sent.

4.3.2.5 Sender Pointer (SPTR)

The Sender Pointer field is supplied by either the user or MITE-80 and specifies the sender TCB address from which the Message Block originated. At the completion of a service request, the Driver will return the Message Block to the task whose TCB address is specified in this field.

4.3.2.6 Service Request (RQST)

The Service Request field is supplied by the user and specifies the type of I/O service operation to be performed. The service requests supported by the Line

Printer Driver are:

Request Code	Operation
00H	Open
01H	Close
03H	Write

Before the first write operation is requested, an Open service request code must be sent to the Driver. An error code will be returned to the calling task if either an I/O operation is requested without the Driver first being issued an Open request, or the request code is an illegal service request for this Driver.

4.3.2.7 Service Option (FLGS)

The Service Options field is supplied by the user and specifies the options to be used in performing the I/O service request of the RQST field. The options supported by the Line Printer Driver are:

Label	Bit	Definition
SYSE	7	1 = System error recovery 0 = User error recovery
EDIT	6	1 = Edit mode 0 = No edit mode
BLCK	5	1 = Block mode 0 = No block mode
FOLD	4	1 = Fold long line 0 = Don't fold long line
NPAR	3	1 = Strip parity bit 0 = Leave parity bit as is
ONCR	2	1 = Detect Carriage Return as end of operation 0 = Don't detect Carriage Return as end of operation

4.3.2.8 User Buffer (BFFR)

The User Buffer field is supplied by the user and contains the address of the starting location of the user's buffer to be used by the Driver in the I/O operation.

4.3.2.9 Byte Count (USIZ)

The Byte Count field is used by the Driver to return the actual number of bytes transferred during the requested I/O operation.

4.3.2.10 Record Count (RSIZ)

The Record Count field is supplied by the user and contains a binary value which the maximum number of bytes allowed to be transferred to/from the user's buffer.

4.4 CONFIGURATION REQUIREMENTS

To configure the Line Printer Driver into a system the user must define the Driver's characteristics and provide for certain dependent modules. The Driver's characteristics are user specified in the Driver's TCB. The dependent modules required of the Driver are determined by the Driver charateristics specified.

4.4.1 LINE PRINTER TCB

To assist in constructing a Line Printer Driver TCB, two macros are provided; MIOTCB and EIOTCB. The primary difference between these two macros is that the MIOTCB macro is for RAM based systems while the EIOTCB macro is for ROM/EPROM based systems. The latter macro creates executable code that will transfer the TCB from ROM to RAM, set-up the Driver's stack, allocate internal Driver required RAM area, and link the TCB into the MITE-80 system. The MIOTCB does not create this executable code and the user, once the TCB is loaded into RAM, must link the TCB into the MITE-80 system. These two macros are general purpose macros, and can be used to configure any standard I/O device. Refer to the BIOS Section of this Operations Manual for additional information on the MIOTCB and EIOTCB macros.

Both macros have identical user specifiable parameters with the exception of the EIOTCB macro having one additional parameter for RAM load address (ADDR).

4.4.2 Formats

The formats of both macros oriented to the Line Printer Driver are as follows:

```
MIOTCB  1,2,3,4,5,6,7,8,9,10,11,12,13,14,15
EIOTCB  1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16
```

The parameter positions are defined as:

MIOTCB PARM #	EIOTCB PARM #	PARM	PARAMETER DESCRIPTION	RANGE (HEX)	DEFAULT VALUE
-	1	ADDR	RAM Load Address	0000-FFFF	Current
1	2	NAME	TCB Name	2 CHAR	ERROR
2	3	TIME	Timer Count	00-FF	No timer
3	4	VECT	I/O VECTor	00-FF	(NM)VECT*
4	5	PORT	I/O PORT	00-FF	(NM)PORT
5	6	STAT	TCB Status	M or A	M
6	7	PRIO	TCB Priority	00-7F	(NM)PRIO
7	8	STKL	Stack Length	0000-FFFF	(NM)STKL
8	9	DTYP	Device Type	Bit encoded	(NM)DTYP
9	10	DSTA	Device Status	Bit encoded	(NM)DSTA
10	11	DOPT	Device Options	Bit encoded	(NM)DOPT
11	12	BOPT	BIOS Options	Bit encoded	(NM)BOPT
12	13	DSIZ	Record Size	0000-FFFF	(NM)DSIZ
13	14	DLEN	Form Length	00-FF	(NM)DLEN
14	15	DPAD	Pad Count	00-FF	(NM)DPAD

* NM corresponds to the first 2 characters of the NAME parameter.

The above parameters must be defined in the specified order. If no value is specified for a parameter,the default value will then be used. A comma (,) must separate all parameter positions, including those parameters not elected to be defined.

The default values for most of the parameters represent a label whose fist two characters are the user defined TCB NAME, such as LPVECT where LP was the user specified TCB NAME and VECT being the default for the vector parameter.

The parameters are further defined as follows, and additional information on parameters 1-12 of MIOTCB and 1-13 of EIOTCB can be found in the BIOS Section of this Operations Manual.

An error message will occur for any macro parameter that is incorrectly specified.

4.4.2.1 Address (ADDR)

The starting memory address of where this TCB and stack is to be loaded.

4.4.2.2 Name (NAME)

The 2-3 character TCB name.

4.4.2.3 Timer Value (TIME)

The I/O watchdog timer value for line printer device.

4.4.2.4 Vector (VECT)

The line printer interrupt service routine vector address.

4.4.2.5 I/O Port (PORT)

The line printer I/O port assignment.

4.4.2.6 Status (STAT)

The priority level of the Line Printer Driver task.

4.4.2.7 Stack Length (STKL)

Defines the stack length for the Line Printer Driver, a stack length of 20 is required.

4.4.2.8 Device Type (DTYP)

Identifies the type of device this Driver handles. For the line printer, BIOS bit 5 (Printer) is set (20H).

4.4.2.9 Device Status (DSTA)

Line printer status of current I/O state.

4.4.2.10 Device Options (DOPT)

Identifies the device options supported by the Line Printer Driver.

4.4.2.11 BIOS Options (BOPT)

Identifies the BIOS options supported by the Line Printer Driver. This parameter's definition is identical to the Service Options (FLGS), refer to 2.3.2.7. Each Message Block's Service Option Flag will not be executed if the corresponding BOPT flag is not set. The BOPT parameter describes what the Line Printer Driver will support regardless of the request in the Message Block's Service Options (FLGS) field. The BOPT byte is highlighted as follows:

B7	=	SYSE, system error recovery
B6	=	EDIT, edit mode
B5	=	BLCK, block mode
B4	=	FOLD, fold long lines
B3	=	NPAR, parity bit disposition
B2	=	ONCR, carriage return detect
B1	=	not used
B0	=	not used

4.4.2.12 Device Size (DSIZ)

Maximum number of bytes per record. For the line printer this will be the maximum number of characters per print line.

4.4.2.13 Device Length (DLEN)

Maximum number of lines on a page.

4.4.2.14 Device Pad Count (DPAD)

Number of null chracters to output after each carriage return or line feed character.

4.4.3 MULTIPLE LINE PRINTER DEVICES

The Line Printer Driver is designed to support multiple line printer devices. Each line printer device must be identified to the MITE-80 System by a unique TCB. Each TCB can specify identical line printer device characteristics but the TCB NAME, PORT address, and VECT must be unique. The Line Printer Driver TCB macros can be used to construct each line printer device's TCB along with it's required characteristics.

4.4.4 DEPENDENT MODULES

The Line Printer Driver executes as a task under MITE-80. In addition to MITE-80, the Driver requires the BIOS Module (BIOS) and optionally requires the MITE-80 Timer Handler (M80TH). If the Driver's TCB "TIME" parameter is specified, then this timer module must be included in the user's system in order to provide the device time-out feature. The Timer Handler must have at least one of its TCBs named 'TH' in order for the Driver's watchdog timer to function properly.

4.4.5 LINKING INTO A SYSTEM

The Line Printer Driver can reside anywhere within the Z80 address range and is provided in relocatable object form. The Driver can be linked directly with the user programs.

To link the Line Printer Driver, the following procedure is used:

```
        $LINK UPROG,M80LP,BIOS,M80TH,MITE80,M8TCBQ TO USYS
```

The above command string will link a user program (UPROG) and the Line Printer Driver (M80LP) into an executable load module of user system (USYS).

SECTION 5

FLOPPY DISK DRIVER

5.1 INTRODUCTION

This section outlines the functional characteristics and user interface to the Floppy Disk Driver. The Floppy Disk Driver is designed to facilitate input and output between user tasks operating under MITE-80 and a floppy disk drive interfaced to an MDX-FLP Card.

5.1.1 FEATURES

The features provided by the Floppy Disk Driver are:

Input/output control of floppy disk drive.
Operates as a task under MITE-80.
User service request queued by Message Block Priority.
User configurable parameters for specifying unique disk drive characteristics.
Accommodates one or multiple disk drives.
Handles mini or standard size disk drives, single density, single or dual sided.

5.1.2 CONFIGURATION

The Floppy Disk Driver requires about 900 bytes of program memory and about 80 bytes of RAM. The RAM is for the Driver's task control block and stack requirements. The Driver is designed to operate with the MDX-FLP Card which is interfaced to a Shugart or Siemens, or equivalent type disk drive. Refer to the Floppy Disk Driver's Configuration Requirements Section for information on user configurable disk unit parameters.

5.2 FUNCTIONAL DESCRIPTION

5.2.1 OPERATION

The Floppy Disk Driver operates as a task executing under MITE-80. User tasks communicate with the Floppy Disk Driver by Message Blocks. The Floppy Disk Driver provides six user service requests; open, read, write, close, status, and write track. Each of the service requests are initiated by a separate Message Block issued from the user task. The driver will return to the calling task a status response at the completion of the service request. With each service request the calling task has several options with which to direct the Driver's operation, such as number of blocks to transfer, type of error recovery, and starting track/sector or physical block number. All of these options are described in the Message Block Section.

The open request conditions the hardware for I/O operations. The read request allows data to be inputted from the disk device and placed into the user's buffer. The write request will output data from the user's buffer to the disk device. The close request will disable the disk unit from any further operation until a open request is performed. The status request will return in the user's buffer the disk unit table which identifies the characteristics of the unit, such as sector size, sector per track, and tracks per cylinder. The write track request will write a track of data, and is primarily used for formatting a diskette.

Each disk unit must be individually opened by a separate open service request before any read or write operation can be performed to that unit.

5.2.2 ERROR HANDLING

Two options exist for the handling of errors which can occur within the Driver during request operations. The options are user specified within each Message Block at service request time. The options allow for the error to be handled by the system, or by the calling task. The system error handling option will output to the terminal device the error code, and will also return the error code to the calling task. The calling task error handling option will cause the Driver to return the error code to the calling task for process disposition. In this case the Driver will not output the error code to the terminal device.

5.3 SERVICE REQUEST MESSAGE BLOCK

A Message Block is required for each disk operation request of open, read, write, status, write track, or close, and calling tasks must adhere to the Floppy Disk Driver's Message Block structure in order to assure proper service request operation.

5.3.1 DATA STRUCTURE

The Floppy Disk Driver Message Block is 17 bytes in length and its structure is as follows:

Field	#of	Byte				
#	Bytes	Offset	Name	Field	Data Type	Source
0	1	0	STAT	Status	Binary	Driver
1	1	1	PRIO	Priority	Binary	User
2	2	2	LINK	Link	Binary	MITE-80
3	2	4	RPTR	Receiver Pointer	Binary	User
4	2	6	SPTR	Sender Pointer	Binary	User
5	1	8	RQST	Service Request	Binary & Bit Encoded	User
6	1	9	FLGS	Service Options	Bit Encoded	User
7	2	10	BFFR	User Buffer	Binary	User
8	1	12	NBKS	# of Blocks	Binary	User
9	3	13	PBLK	Physical Block #	Binary	User
10	1	16	[SCSZ]	Sector Size	Binary	User
11	1	17	[SCTK]	Sector/ Track	Binary	User

The Fields 0 through 4 are a MITE-80 Message Block. Fields 5 through 11 are unique to the Floppy Disk Driver and comprise the Message Block's Data field.

5.3.2 FIELD DESCRIPTIONS

The Message Block's field descriptions are highlighted in the following sections. For the fields of STAT, PRIO, LINK, RPTR, and SPTR, additional information can be found in the MITE-80 Operations Manual (MK79726). For the remaining fields additional information can be found in the BIOS Section of this Operations Manual.

5.3.2.1 Status (STAT)

The Status field is used by the Driver to return a coded response indicating the outcome of the service requested. The response codes are listed in Appendix B.

5.3.2.2 Priority (PRIO)

The Priority field is supplied by the user to specify the priority level of this Message Block.

5.3.2.3 Link (LINK)

The Link field is used by MITE-80 for linking the message into a task's Message Block queue.

5.3.2.4 Receiver Pointer (RPTR)

The Receiver Pointer field is supplied by the user and specifies the receiver's TCB address to which the Message Block is to be sent.

5.3.2.5 Sender Pointer (SPTR)

The Sender Pointer field is supplied by either the user or MITE-80 and specifies the sender TCB address from which the Message Block originated. At the completion of a service request, the Driver will return the Message Block to the task whose TCB address is specified in this field.

5.3.2.6 Service Request (RQST)

The Service Request field is supplied by the user and specifies the type of I/O service operation to be performed. The user requests supported by the Terminal Driver are:

Code	Service
00H	Open
01H	Close
02H	Read
03H	Write
04H	Status
05H	Write Track

Before the first read or write operation is requested, an Open service request code must be sent to the Driver. An error code will be returned to the calling task if either an I/O operation is requested without the Driver first being issued an Open request, or the request code is an illegal service request for this Driver.

5.3.2.7 Service Option (FLGS)

The Service Option field is supplied by the user and specifies the options to be used in performing the I/O service request of the RQST field. The options supported by the Floppy Disk Driver are:

Label	Bit	Definition
SYSE	7	1 = System error recovery 0 = User error recovery
FORM	6	1 = Formatting process(Write track request) 0 = Formatting finished and re-open unit
EXMB	6	1 = Extended Message Block--fields 10 & 11 (Read and write request) 0 = No extended Message Block
DMTY	5-4	00 = Data mark 01 = User defined 10 = User defined 11 = Deleted data mark
TFOR	3	1 = 128 x 2N (IBM format) 0 = 16 x N
CNVT	2	1 = Physical block number being sent 0 = No physical block number being sent
UNUM	1-0	Unit number 00 = Unit 0 01 = Unit 1 10 = Unit 2 11 = Unit 3

5.3.2.8 User Buffer (BFFR)

The User Buffer field is supplied by the user and contains the address of the starting location of the user's buffer to be used by the Driver in the I/O operation.

5.3.2.9 Number of Blocks (NBKS)

The Number of Blocks field is supplied by the user and contains the number of records the Driver is to transfer.

5.3.2.10 Physical Block Number (PBLK)

The Physical Block Number Field is supplied by the user and contains the block number of the disk record of where the read or write driver operation is to start. The three byte field is allocatted as follows:

Byte #13	PBLK	= least significant byte (Bits 7-0)
Byte #14	PBLK + 1	= middle byte (Bits 15-8)
Byte #15	PBLK + 2	= most significant byte (Bits 23-16)

The Physical Block Number is computed by the following formula:

PBLK=SECTOR-1+(TRACK+TRACKS PER CYLINDER*CYLINDER)*SECTORS PER TRACK

Where:

	5.25 inch disk drive:	8 inch disk drive:
Sector	= 1-16	= 1-26
Cylinder	= 0-34	= 0-76
Track Cylinder	= 1 for single side	= 1 for single side
	= 2 for double side	= 2 for double side
Track	= 0 for side	= 0 for single side
	= 0-1 for double side	= 0-1 for double side
Sector/Track	= 16	= 26

When specifying the Physical Block Number as above, the Service Options (FLGS) field's CNVT bit (BIT 2) must be set to a '1'.

The PBLK value can be optionally specified in the form of sector, track, and cylinder numbers. The three byte field is then formatted as:

Byte #13	PBLK	= Sector
Byte #14	PBLK + 1	= Track
Byte #15	PBLK + 2	= Cylinder

When specifying the Physical Block Number in the optional way, the Service Options (FLGS) field's CNVT bit (BIT 2) must be reset to a '0'.

5.3.2.11 Sector Size (SCSZ)

The Sector Size field is supplied by the user and its contents is dependent on the Service Options (FLGS) field specified values.

For FLGS field with EXMB (BIT 6 = 1) and write or read request:

$$SCSZ = \frac{\text{Number of bytes per sector}}{16}$$

For FLGS field wth FORM (BIT 6 = 1) and write track request:

SCSZ = least significant byte of buffer length

5.3.2.12 Sector/Track (SCTR)

The Sector Per Track field is supplied by the user and its contents is dependent on the Service Option (FLGS) field specified values.

For FLGS field with EXMB (BIT 6 = 1) and write or read request:

SCSZ = Number of sectors per track, in binary

For FLGS field with FORM (BIT 6 = 1) and write track request:

SCSZ = Most significant byte of buffer length

5.4 CONFIGURATION REQUIREMENTS

To configure the Floppy Disk Driver into a system the user must define the Driver's characteristics and provide for certain dependent modules. The Driver's characteristics are user specified in the Driver TCB. The dependent modules required of the Driver are determined by the Driver characteristics specified.

5.4.1 Floppy Disk TCB Macros

To assist in constructing a Floppy Disk Driver TCB, two macros are provided; MFDTCB and EFDTCB. The primary difference between these two macros is that the MFDTCB macro is for RAM based systems while the EFDTCB macro is for ROM/EPROM based systems. The latter macro will create executable code that will transfer the TCB from ROM to RAM, set-up the Driver's stack, allocate internal Driver required RAM area, and link the TCB into the MITE-80 system. The MFDTCB does not create this executable code and the user, once the TCB is loaded into RAM, must link the TCB into the MITE-80 system.

Both macros have identical user-specifiable parameters with the exception of the EFDTCB macro having one additional parameter.

5.4.2 Formats

The formats of both macros are as follows:

```
        MFDTCB 1,2,3,4,5,6,7,'8','9','10','11'
        EFDTCB 1,2,3,4,5,6,7,8,'9','10','11','12'
```

The Parameter positions are defined as:

MFDTCB	EFDTCB			RANGE	DEFAULT
PARM #	PARM #	PARM	PARAMETER DESCRIPTION	(HEX)	VALUE
-	1	ADDR	RAM load address	0000-FFFF	Current Address
1	2	NAME	TCB name	2-3 character	ERROR
2	3	TIME	Disk timer count	00-FF	No timer
3	4	VECT	Disk I/O Vector	00-FF	(NM)VECT*
4	5	PORT	Disk I/O Port	00-FF	(NM)PORT
5	6	PRIO	Disk priority	01-7F	(NM)PRIO
6	7	DOPT	Disk device options	Bit Encoded	(NM)DOPT
7	8	NOUN	Number disk units	1-4	(NM)NOUN
'8'-'11'	'9'-'12'	DEVICE UNIT TABLE			
A	A	DCTR	Disk Drive Control	00-FF	(NM)DCTR
B	B	STSZ	Sector Size	00-FF	(NM)STSZ
C	C	STTK	Sectors Per Track	00-FF	(NM)STTK
D	D	TKCL	Tracks Per Cylinder	00-FF	(NM)TKCL
E	E	CLUT	Cylinders Per Unit	00-FF	(NM)CLUT

* NM corresponds to the first 2 characters of the NAME parameter.

The above parameters must be defined in the specified order. If no value is specified for a parameter, the default value will then be used. A comma (,) must separate all parameter positions, including those parameters not elected to be defined.

Parameters '8' - '11' and '9'-'12' represent the Device Unit Table must be specified for each disk unit specified in the "NOUN" parameter (e.g. 2 units require 2 Device Unit Tables). Each Device Unit Table is defined by the parameters A-E with each parameter separated by a comma (,) . Each Table's parameters must also be grouped within quotes (' '). For example, 'A,B,C,D,E','A,B,C,D,E'.

The default values for most of the paramters represent a label whose first two characters are the user defined TCB NAME, such as FDVECT where FD was the user specified TCB NAME and VECT being the default for the vector parameter.

An error message will occur if any macro parameter is incorrectly specified.

The parameters are further defined as follows, and additional information on parameters 1-6 of MFDTCB and 1-7 of EFDTCB can be found in the BIOS Section of this Operations Manual:

5.4.2.1 Address (ADDR)

The starting memory address of where this TCB and stack is to be loaded.

5.4.2.3 Timer Value (TIME)

I/O watchdog timer value for floppy disk device

5.4.2.4 Vector (VECT)

Floppy disk interrupt service routine vector address

5.4.2.5 Port (PORT)

Floppy disk I/O port assignment

5.4.2.6 Priority (PRIO)

Priority level of the Floppy Disk Driver task.

5.4.2.7 Device Options (DOPT)

Identifies the device options supported by the Floppy Disk Driver.

5.4.2.8 Number of Units (NOUN)

Number of floppy disk units that are interfaced to the MDX-FLP Card. For each unit a Device Unit Table must be specified; parameters A-E.

5.4.2.9 Disk Unit Control (DCTR)

Floppy disk unit control, drive stepping rate as follows:

DCTR		8 inch drive	5.25 inch drive
00H	=	6 milliseconds	12 milliseconds
01H	=	6 milliseconds	12 milliseconds
02H	=	10 milliseconds	20 milliseconds
03H	=	20 milliseconds	40 milliseconds

5.4.2.10 Sector Size (STSZ)

Size of the sectors on this unit, calculate as:

$$\text{STSZ} = \frac{\text{number of bytes per sector}}{16}$$

5.4.2.11 Sectors/Track (STTK)

Number of sectors per track on this unit. For example with a 128 byte sector size, 8 inch drive = 26 (1AH) sectors and 5.25 inch drive = 16 (10H) sectors.

5.4.2.12 Tracks/Cylinder (TKCL)

Number of tracks per cylinder on this unit:

01H = single sided; 1 track per cylinder
02H = double sided; 2 track per cylinder

5.4.2.13 Cylinders/Unit (CLUT)

Number of cylinders per unit, as:

4DH = 77 cylinders (8 inch drive)
23H = 35 cylinders (5.25 inch drive)

5.4.3 MULTIPLE FLOPPY DISK DEVICES

The Floppy Disk Driver is designed to support multiple floppy disk devices. Up to four disk devices can be interfaced to the MDX-FLP Card, and each device must be identified by a Device Unit Table in the Driver's TCB. The Floppy Disk Driver also supports multiple MDX-FLP Cards. Each Card must be identified to the MITE-80 system by a unique TCB along with the appropriate Device Unit Table. Each TCB and Device Unit table can identify identical disk device characteristics, but the TCB NAME, port address and vector address must be unique. The Floppy Disk Driver TCB macros can be used to construct each disk devices TCB along with the devices required characteristics.

5.4.4 DEPENDENT MODULES

The Floppy Disk Driver executes as a task under MITE-80. In addition to MITE-80, the Driver requires the BIOS Module (BIOS) and optionally requires the MITE-80 Timer Handler (M80TH). If the Driver's TCB "TIME" parameter is specified, then this timer module must be included in the user's system in order to provide the desired device time-out feature. The Timer Handler must have its, or one of its, TCBs named as 'TH' in order for the Driver's watchdog timer to function properly.

5.4.5 LINKING INTO A SYSTEM

The Floppy Disk Driver can reside anywhere within the Z80 address range and is provided in relocatable object form. The Driver can be linked directly with user programs. The Driver's TCB macro(s) should be grouped, assembled, and linked with all of the other tasks TCBs.

To link the Floppy Disk Driver the following procedure is used

```
$LINK UPROG,M80FD,MITE80,M80TH,BIOS,M8TCBQ TO USYS
```

The above command string will link a user program (UPROG) and the Floppy Disk Driver (M80FD) into an executable load module of user systems (USYS).

SECTION 6

ERROR HANDLER

6.1 INTRODUCTION

This section describes the MITE-80/BIOS Error Handler Task. It provides standard system error recovery for each of the drivers running under MITE-80/BIOS.

6.1.1 FEATURES

The features provided by the Error Handler are:

A standard error recovery interface.
Operates as a task under MITE-80.
Provides a display of error status to operator.

6.1.2 CONFIGURATION

The Error Handler comes in two different versions: (1) M80EH which provides a simplified error display and (2) M80EHM which decodes error status for the operator. The first configuration, M80EH, requires about 320 bytes of program memory and about 170 bytes of RAM. The second configuration, M80EHM, requires about 560 bytes of program memory and about 190 bytes of RAM. In both cases, the RAM is used for TCB, stack, and buffer space. One of the two versions of the Error Handler is required when using any BIOS I/O driver.

6.2 FUNCTIONAL DESCRIPTION

6.2.1 OVERVIEW

The Error Handler (EH) operates as a task executing under MITE-80. The BIOS routine IO?EOB Forwards any I/O Message Blocks which specify System Error recovery and which do not complete successfully to the Error Handler. EH then builds a message describing the error and sends it to the Terminal Driver (TD). TD then displays the message on to the terminal device and asks for a response. EH takes the operator's response and performs the requested action.

6.2.2 BIOS INTERFACE

The BIOS subroutine IO?EOB examines the status of each I/O operation when it is entered. If the most significant bit is set this indicates that an I/O error of some type has occurred. BIOS then checks the SYSE bit of FLGS in the I/O message block (see section 2.3.2.7 for further information). If it is also set IO?EOB forwards the message block to the Error Handler task.

EH takes the message block (MB), formats an error message, and sends the error message to the Terminal Driver (TD) for printing. EH waits for a response from the system operator to determine what further processing is to be taken for the message. The Error Handler will, based on the operator's response, either retry the original request, cancel it, dump the contents of the MB on the terminal printer, or kill the task which sent the message.

6.2.3 ERROR FORMATTING

The Error Handler takes any I/O block forwarded to it and formats a message to be sent to the system terminal task (task 'TD').

6.2.3.1 ERROR MESSAGE

Two formats of the Error Handler error message are available and is selected at system creation time. The first, provided by M80EH, is:

```
*** I/O ERROR xx, DRIVER=aa, TASK=bb (R,C,K,D)?
```

The second, provided by M80EHM, is:

```
*** I/O ERROR xx message, DRIVER=aa, TASK=bb (R,C,K,D)?
```

6.2.3.1.1 DESCRIPTION

xx The error status (STAT) from the user's Message Block is displayed in hexadecimal.

message A short explanation of the error provided (optionally) in ASCII (see Appendix B)

aa The driver task's name is displayed in ASCII

bb The calling task's name is displayed in ASCII

6.2.3.2 OPERATOR RESPONSES

Four options are provided for operator response:

R Retry operation after correcting problem.

C Cancel unrecoverable operation (returns message to caller).

K Kill calling task because further operation of the caller depends upon this operation's successful completion (use with CAUTION).

D Dump message block which caused error.

6.2.3.3 DUMP FORMAT

The Dump command causes the EH task to dump 20 bytes of memory starting at the first byte of the caller's Message Block. The format of the dump is as follows:

```
Addr  PrSt Link Rptr Sptr OpRq Bffr Usiz Rsiz Wd08 Wd09
```

The definition of each of the words displayed follows:

Addr Address of the errant Message Block (MB).

Pr Priority of the MB.

St	Status of the MB.
Link	(Contains no useful information).
Rptr	Reciever (Driver) TCB address.
Sptr	Sender TCB address.
Op	Option flags.
Rq	Request code.
Bffr	Buffer address.
Usiz	Used size (number of bytes transferred before error occurred).
Rsiz	Record size.
Wd08	Word 8 of the Message Block (Prompt address for Terminal Driver).
Wd09	Word 9 of the Message Block (Prompt message length for TD).

6.3 CONFIGURATION REQUIREMENTS

To configure the Error Handler task into a system the user must define the Task Control Block for the Error Handler. The user must also specify which version of EH is required. Both of these requirements are satisfied by the following macros.

6.3.1 Error Handler TCB Macros

To assist in constructing an Error Handler TCB, two macros are provided; MEHTCB and EEHTCB. The primary difference between these two macros is that the MEHTCB macro is for RAM based systems while the EEHTCB macro is for ROM/EPROM based systems. The latter macro will create executable code that will transfer the TCB from ROM to RAM, set-up the Handler's stack, allocate internal Handler required RAM area, and link the TCB into the MITE-80 system. The MEHTCB does not create this executable code and the user, once the TCB is loaded into RAM, must link the TCB into the MITE-80 system.

Both macros have identical user specifiable parameters with the exception of the EEHTCB macro having one additional parameter.

6.3.2 Formats

The formats of both macros are as follows:

```
MEHTCB  1
EEHTCB  1,2
```

The parameters are defined as follows:

MEHTCB PARM #	EEHTCB PARM #	PARM	PARAMETER DESCRIPTION	RANGE (HEX)	DEFAULT VALUE
-	1	ADDR	RAM load address	0000-FFFF	Current Address
1	2	MSG	Expanded Message Request	'MSG' or missing	No Expand

The above parameters must be defined in the specified order. If no value is specified for a parameter, the default value will then be used. A comma (,) must separate all parameter positions, including those parameters not elected to be defined.

The parameters are further defined as follows:

6.3.2.1 Address (ADDR)

The starting memory address of where this TCB and stack is to be loaded.

6.3.2.2 Message (MSG)

This parameter must be either the three characters 'MSG' or omitted. If 'MSG' is specified, the extended version of EH is selected (M80EHM). If the parameter is omitted or not equal to 'MSG' then the shorter version (M80EH) is selected.

6.4 MULTIPLE ERROR HANDLERS

The Error Handler macros are designed to support a single Error Handler. Any attempt at installing more than one in a system will cause several assembly errors.

6.5 DEPENDENT MODULES

The Error Handler executes as a task under MITE-80. In addition to MITE-80, the Handler requires the BIOS Module (BIOS) and a terminal driver named 'TD' to function properly. BIOS conversely requires the Error Handler to function properly if system error recovery is desired.

6.6 LINKING INTO A SYSTEM

The Error Handler can reside anywhere within the Z80 address range and is provided in relocatable object form. The Handler can be linked directly with user programs.

Section 7 of this document provides an example of how to link this Handler into a system.

SECTION 7

BIOS-80 SYSTEM FILES

7.1 INTRODUCTION

All of the system files provided on the BIOS-80 diskette are outlined in this section. A majority of the files are provided in relocatable object format, while others are in Z80 Assembler source format.

7.2 FILE LIST

The file names are as follows:

File Name	Software Module
BIOS.OBJ	BIOS-80 nucleus, relocatable object.
M80TD.OBJ	Terminal Driver, relocatable object.
M80LP.OBJ	Printer Driver, relocatable object.
M80FD.OBJ	Floppy Disk Driver, relocatable object.
M80EH.OBJ	Error Handler, relocatable object
M80EHM.OBJ	Error Handler with Messages, relocatable object.
BIOS.EQU	BIOS equates, source.
IOTASK.EQU	I/O Task Default equates, source.
BIOSYS.MAC	BIOS system macros, source.
BIOESY.MAC	BIOS system executable macros, source.

The system routines of BIOS, Terminal Driver, Printer Driver, and Floppy Disk Driver have all been discussed in previous sections of this manual. The other files are outlined as follows:

7.2.1 BIOS EQUATES - BIOS.EQU

All BIOS system equates that a user program would require are contained in the file, BIOS.EQU. This file can be included in every user program that has references to BIOS-80 services and labels. The labels defined in the BIOS.EQU file are reserved as BIOS-80 labels. When developing programs, the user must ONLY use these labels for BIOS-80 references in order to prevent multiple defined label errors from occurring.

7.2.1.1 USING BIOS.EQU

The BIOS.EQU file can be used as an INCLUDE file in a user program. A listing of BIOS.EQU is provided in Appendix A. Example:

```
INCLUDE        BIOS.EQU        ;BIOS-80 GLOBALS & EQUATES
GLOBAL         ULAB1           ;USER REQUIRED
GLOBAL         ULAB2           ;GLOBAL & EQUATES
```

7.2.2 I/O TASK EQUATES - IOTASK.EQU

All BIOS I/O Task default equates that a user program would require are contained in one file, IOTASK.EQU. This file is provided as a development aid and has all default equates for BIOS-80 I/O Tasks references defined. The labels defined in the IOTASK.EQU file are reserved as BIOS-80 labels.

7.2.2.1 USING IOTASK.EQU

The IOTASK.EQU file is used as an INCLUDE file in a user program. A listing of IOTASK.EQU is provided in Appendix A.

7.2.2.2 EXAMPLE

```
INCLUDE        IOTASK.EQU      ;BIOS-80 I/O TASK DEFAULT EQUATES
MIOTCB         LP              ;ALL L.P. DEFAULT EQUATES
                               ;COME FROM IOTASK.EQU
```

7.2.3 BIOS SYSTEM MACROS - BIOSYS.MAC

The BIOS system macro file, BIOSYS.MAC, contains all of the BIOS-80 system macros in Z80 assembler source format. The purpose and use of these macros has been outlined in previous sections. All of the macros provided require the use of the MOSTEK MACRO-80 Assembler. The macros provided in the BIOSYS.MAC file are:

MACRO	PURPOSE	REFER TO
MTDTCB	Builds a Terminal Driver TCB and stack.	Section 3
MFDTCB	Builds a Floppy Disk Driver TCB and stack.	Section 5
MIOTCB	Builds a General I/O Driver TCB and stack.	Section 2

7.2.3.1 USING BIOSYS.MAC

The BIOSYS.MAC file is used as an INCLUDE file in a user program. An example follows:

```
        INCLUDE         BIOSYS.MAC              ;BIOS-80 SYSTEM MACROS
        MTDTCB          ..........              ;TD TCB DEFINED
        MFDTCB          ..........              ;FD TCB DEFINED
```

7.2.4 BIOS SYSTEM EXECUTABLE MACROS - BIOESY.MAC

The BIOS system executable macro file, BIOESY.MAC, contains all of the BIOS-80 system macros that create executable code. The file is provided in source form. The macros provided in this file are:

MACRO	PURPOSE	REFER TO
ETDTCB	Builds a terminal Driver	Section 3
EFDTCB	Builds a Floppy Disk Driver TCB, stack, and installation code.	Section 5
EIOTCB	Builds a General I/O Driver TCB, stack, and installation code.	Section 2

7.2.4.1 USING BIOESY.MAC

The BIOESY.MAC is designed to be used in an absolute program segment, and generates executable code to transfer the TCB and stack structures from ROM/EPROM to RAM. A usage example follows:

```
PSECT   ABS             ;ABSOLUTE PROGRAM SEGMENT
ORG     XXXXH           ;XXXX LOCATION OF EXECUTABLE TCBS
.
:
INCLUDE BIOESY.MAC      ;BIOS-80 SYSTEM EXECUTABLE MACROS
.
:
ETDTCB  ...             ;TCB TD DEFINED
EFDTCB  ...             ;TCB FD DEFINED
```

7.3 TYPICAL SYSTEM GENERATION FILE

The following is a typical system generation showing an example of each of the 'M' type TCB macros and the start up code for the system:

```
        PSECT   ABS
        NAME    USYS            ;USER SYSTEM
        INCLUDE M80SYS.MAC
        INCLUDE BIOSYS.MAC
        INCLUDE M80SYS.EQU
        INCLUDE BIOS.EQU
        INCLUDE IOTASK.EQU
        CLIST   0               ;SUPRESS MACRO PSEUDO OPS
        GLOBAL  USEREP          ;USER TASK ENTRY POINT
 USYS:  LD      SP,0FFAAH       ;INITIALIZE STACK
        OR      A
        LD      I,A             ;USE PAGE ZERO FOR INTERRUPTS
        IM      2               ;USE MODE 2 INT'S
        LD      DE,M8TCBQ       ;GET TCB QUEUE ADDRESS
        LD      BC,TCBTH        ;ALWAYS START TIMER FIRST
        CALL    M8SN            ;ACTIVATE TIMER TASK
        LD      BC,TCBEH        ;NOTE DE STILL=M8TCBQ
        CALL    M8SN            ;ACTIVATE ERROR HANDLER
        LD      BC,TCBTD
        CALL    M8SN            ;ACTIVATE TERMINAL DRIVER
        LD      BC,TCBFD
        CALL    M8SN            ;ACTIVATE FLOPPY DISK DRIVER
        LD      BC,TCBLP
        CALL    M8SN            ;ACTIVATE LINE PRINTER DRIVER
        LD      BC,TCBUT
        CALL    M8SN            ;ACTIVATE USER TASK
HLOOP:  HALT                    ;NOTHING MORE TO DO
        JR      HLOOP-$
        ORG     USYS+256        ;RESERVE REMAINDER OF PAGE
                                ;FOR INT VECTORS
        EJECT
        MEHTCB  MSG             ;ERROR HANDLER WITH MESSAGE
        MTHTCB  TH              ;TIMER HANDLER
        EJECT
        MTDTCB  TD              ;TERMINAL DRIVER
        EJECT
        MFDTCB  FD              ;FLOPPY DISK DRIVER
        EJECT
        MIOTCB  LP              ;LINE PRINTER DRIVER
 ;
 ; USER TCB:    NAME='UT', USES ALL REGISTERS, PRIORITY=20,
 ;              STACK=100 BYTES, ENTRY POINT=USEREP
         MTCB   UT,ALL,20,100,USEREP
```

7.4 SYSGEN PROCEDURE

For the above Sysgen file the following procedure could be used:

```
$MACRO USYS TO CP:

  MOSTEK MACRO-80 Assembler Vn.n   Options?

$LINK USYS,MITE80,BIOS TO USYS,CP:

  MOSTEK FLP-80/DOS Vn.n LINKER
OPTIONS ? ULC

 .
 .
 .

UNDEFINED SYMBOLS xx
SEARCH DISK UNIT 1/0? 0 <ASSUMING MITE-80/BIOS ON UNIT 0>
.
.
.

$G USYS

$DDT

.E 0
```

APPENDIX A

BIOS EQUATE FILE

```
                                MOSTEK MACRO-80 ASSEMBLER  V2.2 PAGE   1
LOC  OBJ.CODE  STMT-NR SOURCE-STMT PASS2 BIOS   BIOS   BIOS   REL
                     1         GLOBAL IO?WHY
                     2         GLOBAL IO?ILL
                     3         GLOBAL IO?EOB
                     4         GLOBAL IO?ISR
                     5         GLOBAL IO?OPN
                     6         GLOBAL IO?GNC
                     7         GLOBAL IO?PNC
                     8         GLOBAL TCBEH      ;ERROR HANDLER TASK
                          ;    IOTCB EQUATES
                    55         LIST LIST
          =FFE6     56 TMSG    DEFL TMSG-IOTCB
          =FFEF     57 TIME    DEFL TIME-IOTCB
          =FFF0     58 DCUR    DEFL DCUR-IOTCB
          =FFF2     59 NODE    DEFL NODE-IOTCB
          =FFF4     60 DFLG    DEFL DFLG-IOTCB
          =0007     61  SYSE:  DEFL 7            ;SYSTEM ERROR RECOVERY
          =0006     62  EDIT:  DEFL 6            ;EDIT MODE
          =0005     63  BLCK:  DEFL 5            ;BLOCK MODE I/O
          =0004     64  FOLD:  DEFL 4            ;FOLD LONG LINES
          =0003     65  NPAR:  DEFL 3            ;STRIP PARITY BIT
          =0002     66  ONCR:  DEFL 2            ;TERMINATE I/O ON <CR>
          =0001     67  ECHO:  DEFL 1            ;ECHO INPUT TO OUTPUT
          =0000     68  PRMT:  DEFL 0            ;PROMPT STRING IN RQST
          =FFF5     69 DPOS    DEFL DPOS-IOTCB
          =FFF7     70 DLIN    DEFL DLIN-IOTCB
          =FFF7     71 NBLK    DEFL DLIN         ;DISK ONLY
          =FFF8     72 PBKL    DEFL DLIN+1       ;DISK ONLY
          =FFF8     73 DTAB    DEFL DTAB-IOTCB
          =0007     74  SPEC:  DEFL 7             ;SPECIAL CHARACTER PROCESSING IN
                                                  EFFECT
          =0006     75  CTLU:  DEFL 6            ;CONTROL "U" PROCESSING IN EFFECT
```

```
                                   MOSTEK MACRO-80 ASSEMBLER  V2.2 PAGE   2
LOC  OBJ.CODE  STMT-NR SOURCE-STMT PASS2 BIOS   BIOS   BIOS   REL

      =0005      76  DOLF:  DEFL 5          ;"OUTPUT LINE FEED NEXT TIME" FLAG
      =0004      77  STAB:  DEFL 4          ;"SIMULATING TABS" IN EFFECT
      =0003      78  DONE:  DEFL 3          ;NEXT CALL IS END-OF-RECORD
      =0002      79  PADS:  DEFL 2          ;PADDING IN EFFECT
      =0001      80  DOFF:  DEFL 1          ;"SIMULATING FORM FEED" IN EFFECT
      =FFF9      81  DCNT   DEFL DCNT-IOTCB
      =FFF9      82  PBKM   DEFL DCNT       ;DISK ONLY
      =FFFA      83  PBKU   DEFL DCNT+1     ;DISK ONLY
      =FFFB      84  VECT   DEFL VECT-IOTCB
      =FFFC      85  PORT   DEFL PORT-IOTCB
      =FFFD      86  CISR   DEFL CISR-IOTCB
      =000A      87  DTYP   DEFL DTYP-IOTCB
      =0007      88  DISK:  DEFL 7
      =0006      89  CARD:  DEFL 6
      =0005      90  PRTR:  DEFL 5
      =0004      91  PTAP:  DEFL 4
      =0003      92  TERM:  DEFL 3
      =000B      93  DSTA   DEFL DSTA-IOTCB
      =0007      94  DOPN:  DEFL 7          ;DEVICE IS OPEN
      =000C      95  DOPT   DEFL DOPT-IOTCB
      =0002      96  TIMO:  DEFL 2          ;DEVICE IS TIMED
      =0001      97  NTAB:  DEFL 1          ;SIMULATE TAB STOPS
      =0000      98  NFFD:  DEFL 0          ;SIMULATE FORM FEED
      =000D      99  BOPT   DEFL BOPT-IOTCB
      =0007     100  SYSE:  DEFL 7          ;SYSTEM ERROR RECOVERY
      =0006     101  EDIT:  DEFL 6          ;EDIT MODE
      =0005     102  BLCK:  DEFL 5          ;BLOCK MODE I/O
      =0004     103  FOLD:  DEFL 4          ;FOLD LONG LINES
      =0003     104  NPAR:  DEFL 3          ;STRIP PARITY BIT
      =0002     105  ONCR:  DEFL 2          ;TERMINATE I/O ON <CR>
      =0001     106  ECHO:  DEFL 1          ;ECHO INPUT TO OUTPUT
      =0000     107  PRMT:  DEFL 0          ;PROMPT STRING IN RQST
      =000E     108  DSIZ   DEFL DSIZ-IOTCB
```

```
                                        MOSTEK MACRO-80 ASSEMBLER  V2.2 PAGE   3
LOC  OBJ.CODE  STMT-NR SOURCE-STMT PASS2 BIOS   BIOS   BIOS   REL

       =0010          109 DLEN    DEFL DLEN-IOTCB
       =0011          110 DPAD    DEFL DPAD-IOTCB
0000                  111         ORG  IO?BEG
                          ;
                          ;     I/O MESSAGE EQUATES
                      127         LIST LIST
       =0008          128 RQST    DEFL RQST-IOMSG
       =0009          129 FLGS    DEFL FLGS-IOMSG
       =000A          130 BFFR    DEFL BFFR-IOMSG
       =000C          131 USIZ    DEFL USIZ-IOMSG
       =000E          132 RSIZ    DEFL RSIZ-IOMSG
0000                  133         ORG  IO?BEG
                          ;
                          ; ERROR CODES
       =0080          136 E?RQST  EQU  80H
       =0081          137 E?TIME  EQU  81H
       =0082          138 E?OPEN  EQU  82H
       =0001          139 S?DONE  EQU  01H
```

```
     BIOS EQUATES MOSTEK MACRO-80 ASSEMBLER  V2.2 PAGE   1
LOC  STMT-NR SOURCE-STMT PASS2 BIOS   BIOS   BIOS   REL

        1          GLOBAL IO?WHY
        2          GLOBAL IO?ILL
        3          GLOBAL IO?EOB
        4          GLOBAL IO?ISR
        5          GLOBAL IO?OPN
        6          GLOBAL IO?GNC
        7          GLOBAL IO?PNC
        8          GLOBAL TCBEH      ;ERROR HANDLER TASK
          ;     IOTCB EQUATES
       55          LIST LIST
=FFE6  56 TMSG     DEFL TMSG-IOTCB
=FFEF  57 TIME     DEFL TIME-IOTCB
=FFF0  58 DCUR     DEFL DCUR-IOTCB
=FFF2  59 NODE     DEFL NODE-IOTCB
=FFF4  60 DFLG     DEFL DFLG-IOTCB
=0007  61  SYSE:   DEFL 7            ;SYSTEM ERROR RECOVERY
=0006  62  EDIT:   DEFL 6            ;EDIT MODE
=0005  63  BLCK:   DEFL 5            ;BLOCK MODE I/O
=0004  64  FOLD:   DEFL 4            ;FOLD LONG LINES
=0003  65  NPAR:   DEFL 3            ;STRIP PARITY BIT
=0002  66  ONCR:   DEFL 2            ;TERMINATE I/O ON <CR>
=0001  67  ECHO:   DEFL 1            ;ECHO INPUT TO OUTPUT
=0000  68  PRMT:   DEFL 0            ;PROMPT STRING IN RQST
=FFF5  69 DPOS     DEFL DPOS-IOTCB
=FFF7  70 DLIN     DEFL DLIN-IOTCB
=FFF7  71 NBLK     DEFL DLIN         ;DISK ONLY
=FFF8  72 PBKL     DEFL DLIN+1       ;DISK ONLY
=FFF8  73 DTAB     DEFL DTAB-IOTCB
=0007  74  SPEC:   DEFL 7            ;SPECIAL CHARACTER PROCESSING IN EFFECT
=0006  75  CTLU:   DEFL 6            ;CONTROL "U" PROCESSING IN EFFECT
=0005  76  DOLF:   DEFL 5            ;"OUTPUT LINE FEED NEXT TIME" FLAG
=0004  77  STAB:   DEFL 4            ;"SIMULATING TABS" IN EFFECT
=0003  78  DONE:   DEFL 3            ;NEXT CALL IS END-OF-RECORD
=0002  79  PADS:   DEFL 2            ;PADDING IN EFFECT
=0001  80  DOFF:   DEFL 1            ;"SIMULATING FORM FEED" IN EFFECT
```

```
     BIOS EQUATES MOSTEK MACRO-80 ASSEMBLER  V2.2 PAGE   2
LOC  STMT-NR SOURCE-STMT PASS2 BIOS   BIOS   BIOS   REL

=FFF9  81 DCNT    DEFL DCNT-IOTCB
=FFF9  82 PBKM    DEFL DCNT         ;DISK ONLY
=FFFA  83 PBKU    DEFL DCNT+1       ;DISK ONLY
=FFFB  84 VECT    DEFL VECT-IOTCB
=FFFC  85 PORT    DEFL PORT-IOTCB
=FFFD  86 CISR    DEFL CISR-IOTCB
=000A  87 DTYP    DEFL DTYP-IOTCB
=0007  88  DISK:  DEFL 7
=0006  89  CARD:  DEFL 6
=0005  90  PRTR:  DEFL 5
=0004  91  PTAP:  DEFL 4
=0003  92  TERM:  DEFL 3
=000B  93 DSTA    DEFL DSTA-IOTCB
=0007  94  DOPN:  DEFL 7            ;DEVICE IS OPEN
=000C  95 DOPT    DEFL DOPT-IOTCB
=0002  96  TIMO:  DEFL 2            ;DEVICE IS TIMED
=0001  97  NTAB:  DEFL 1            ;SIMULATE TAB STOPS
=0000  98  NFFD:  DEFL 0            ;SIMULATE FORM FEED
=000D  99 BOPT    DEFL BOPT-IOTCB
=0007 100  SYSE:  DEFL 7            ;SYSTEM ERROR RECOVERY
=0006 101  EDIT:  DEFL 6            ;EDIT MODE
=0005 102  BLCK:  DEFL 5            ;BLOCK MODE I/O
=0004 103  FOLD:  DEFL 4            ;FOLD LONG LINES
=0003 104  NPAR:  DEFL 3            ;STRIP PARITY BIT
=0002 105  ONCR:  DEFL 2            ;TERMINATE I/O ON <CR>
=0001 106  ECHO:  DEFL 1            ;ECHO INPUT TO OUTPUT
=0000 107  PRMT:  DEFL 0            ;PROMPT STRING IN RQST
=000E 108 DSIZ    DEFL DSIZ-IOTCB
=0010 109 DLEN    DEFL DLEN-IOTCB
=0011 110 DPAD    DEFL DPAD-IOTCB
0000  111         ORG  IO?BEG
          ;
          ;     I/O MESSAGE EQUATES
     BIOS EQUATES MOSTEK MACRO-80 ASSEMBLER  V2.2 PAGE   3
LOC  STMT-NR SOURCE-STMT PASS2 BIOS    BIOS    BIOS    REL

      127          LIST LIST
=0008 128 RQST     DEFL RQST-IOMSG
=0009 129 FLGS     DEFL FLGS-IOMSG
=000A 130 BFFR     DEFL BFFR-IOMSG
=000C 131 USIZ     DEFL USIZ-IOMSG
=000E 132 RSIZ     DEFL RSIZ-IOMSG
0000  133          ORG  IO?BEG
          ;
          ; ERROR CODES
=0080 136 E?RQST   EQU  80H
=0081 137 E?TIME   EQU  81H
=0082 138 E?OPEN   EQU  82H
=0001 139 S?DONE   EQU  01H
```

APPENDIX B

ERROR MESSAGES:

Error #	Message
81	TIMEOUT
82	NOT OPEN
90	NOT READY
91	WRITE PROTECT
92	WRITE FAULT
93	RECORD NOT FOUND
94	CRC ERROR
95	LOST DATA
97	BUSY
98	SEEK
99	READ ID
9A	ILLEGAL BLOCK #
All Other #'s	ILLEGAL STATUS

MOSTEK®

Z80•F8 Covering the full spectrum of
3870 microcomputer applications.

1215 W. Crosby Rd. • Carrollton, Texas 75006 • 214/323-6000
In Europe, Contact: MOSTEK Brussels
150 Chaussee de la Hulpe, B1170, Belgium;
Telephone: 660.69.24

Mostek reserves the right to make changes in specifications at any time and without notice. The information furnished by Mostek in this publication is believed to be accurate and reliable. However, no responsibility is assumed by Mostek for its use; nor for any infringements of patents or other rights of third parties resulting from its use. No license is granted under any patents or patent rights of Mostek.

PRINTED IN USA May 1980
Publication No. MK79784