Glen Thornton

TEXAS INSTRUMENTS

CONSUMER PRODUCTS GROUP

HEX-BUS(TM)

SPECIFICATIONS

TABLE OF CONTENTS

9/20/83

HEX-BUS(TM)
INTELLIGENT PERIPHERAL BUS
STRUCTURE, TIMING, AND PROTOCOL SPECIFICATION

Texas Instruments Incorporated
Consumer Products Group
Calculator Division

8/10/83

Revision 3.7

SECTION 1

Introduction

1.1 Purpose of This Document

This specification describes the timing and protocol for a communications bus to connect intelligent peripherals to calculator type products. This document is meant to be used in formulating the detailed software design and to allow an accurate review of the plans to support peripherals.

1.2 Scope of This Document

This document describes the signals, messages, and protocol on the peripheral bus. The electrical interface is not described in detail; specifically the design of hardware to implement the bus is not discussed. The primary discussion centers around the method used by devices to perform data transmission on the bus. The specification includes the following sections:

1. Overview of the bus characteristics.

2. Description of bus signals.

3. Bus message structure.

4. Access to I/O subroutines within the calculator.

5. Notes on the use of the bus

6. Bus transfer examples

An application programmer need only read the section on I/O subsystem access in order to understand how to access peripherals from assembly language. The document is structured with that section providing a complete description for software use.

1.3 Terminology

IPB - Intelligent Peripheral Bus

RAM - Random Access Memory

HSK - Handskake I/O Control Line

BAV - Bus Available I/O Control Line

LAT - Internal I/O Control Signal Derived from HSK

PAB - Peripheral Access Block

SAB - Slave Access Block

SRPAB- Service Request Peripheral Access Block

DSR - Device Service Routine

When the symbol '>' precedes a number in this document, it indicates that the number is hexadecimal.

1.4 Related Documents

Related documents may be obtained by contacting Art Hunter at TI Lubbock; phone 796-3431; MSG HEXB

SECTION 2

Specification Overview

2.1 General Features

The bus is designed to provide data transmission between a calculator and peripherals. It is organized in a master-slave arrangement with the controlling calculator as the master and the peripherals as slaves. In this manner the calculator on the bus acts to control data activity. Normal communication will be initiated by a command message from the calculator to a peripheral. The peripheral responds to the calculator with a data or status message to signal completion of the command.

Certain peripherals, when allowed by the master device, may initiate a device poll by the master. This effects a service request feature by a slave device.

Through software loaded into RAM, the calculator can also have a slave communication mode. This allows it to operate on the bus as a peripheral to another calculator (to allow two calculators to communicate) and provides the capability for a more complex communication structure for applications which require it. This capability is selectable from both assembly and high level languages. Capability has also been given for the calculator to pass the master control to another device on the bus. This does not allow a device to arbitrarily take control as the master device but only do so by command from the current master.

Transmissions on the bus are defined in the context of a "message frame" which consists of a command message from the master and a response message from the slave. The form of the messages is described in detail in a later section. This message concept requires that each peripheral be intelligent enough to decode information and bus status and be able to follow protocol. It is assumed that each peripheral will contain a microcomputer for the bus interface and control functions.

SECTION 3

Bus Signal and Timing Descriptions

3.1 Signals

The bus is connected to all devices in a parallel manner. No buffering is provided at any point along the data bus. The physical bus consists of 8 lines. A ground reference signal and a line reserved for future use make up 2 of the lines. There are 4 parallel data lines defining a 4 bit nibble as the basic unit of information carried on the bus. Data within the communication protocol is defined in 8-bit units (bytes). Each byte corresponds to two transmissions on the bus, least significant nibble first. Data is output to the bus via open collector drivers.

```
D3   --------------- most significant data bit
D2   --------------- data bit
D1   --------------- data bit
D0   --------------- least significant data bit
HSK  --------------- handshake
BAV  --------------- bus available
FUT  --------------- reserved for future use
GND  --------------- ground
```

The speed of transmission of the data bus is controlled by the handshake line HSK. The BAV (bus available) signal is used to designate the beginning of a command message from the master and for a slave device to request service as described in later sections.

3.2 Handshake Timing

The signal timing of HSK and the data lines is illustrated in Figure 3.1. The falling edge of HSK is the signal to receiving devices that a nibble of data is available on the bus. HSK triggers a signal that informs the software I/O drivers that a nibble of data is available on the bus. The rising edge of HSK is the signal to the transmitting device that all receivers have read the data. HSK is an open collector line so that any one device may keep it at a low level. When the receiving device(s) see that HSK has gone low they rapidly (through hardware) pull HSK low also. The receiving device(s) then hold HSK low until they have processed the data. The transmitting device will release HSK shortly after pulling it low. This normal interaction is illustrated in Figure 3.2. If the transmitting device is slower than the receivers then it may dictate the bus speed as shown in Figure 3.3.

When a device is not interested in the data being transmitted it may disable itself from the bus and wait for the next message frame (denoted by a BAV transition from high to low). Non-active devices need not even participate in the handshake activity.

Figure 3.1
Bus Handshake Sequence

```
          __________                            ___________
                    \                          /           \
HSK                  \________________________/             \_______________
                     |<- one handshake   ->|                |<- next --
                     |                     |

          __________________________________________________________________
          XXXXXXXXX                          XXXXXXXXXXXX
DO-D3     XXXXXXXXX__________________________XXXXXXXXXXXX___________________
                     |<-- data valid  -->|
```

Figure 3.2
Handshake Components
(Receiver Limited)

```
             ___________                          ______________
                        \                        /              \
HSK(composite)           \______________________/                \___________
                          |<-one handshake->|                    |<- next -
                          |                 |

             __________    ____________________________________
                       |\                  /                   \
HSK(xmit)              | \________________/                     \___________
                       |  |<5us>|
                       |<  Bus >|
             ______________                  __________________
                           \                /                  \
HSK(rcv)                    \______________/                    \___________
```

Figure 3.3
Handshake Components
(Transmitter Limited)

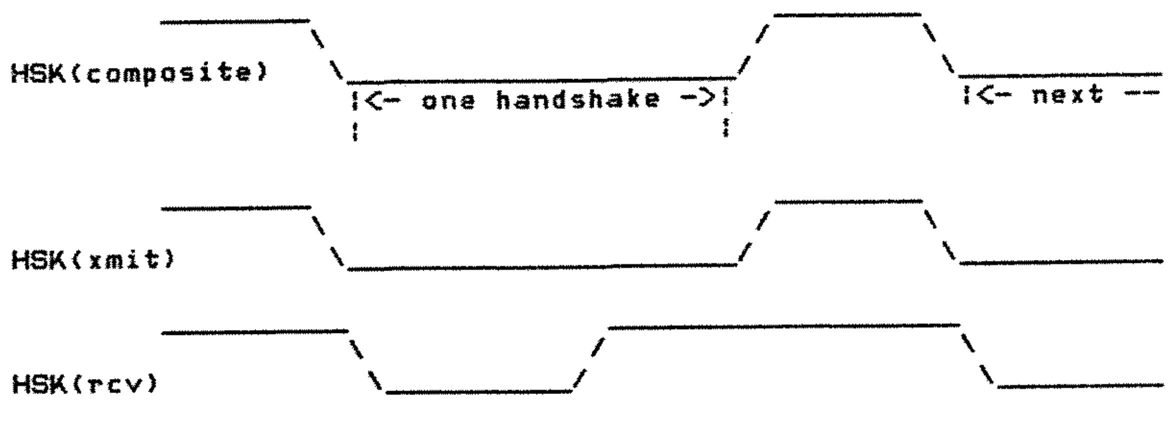


Table 3.1
Handshake Timing Parameters
(Microseconds)

Item	Min	Max
HSK low to data valid	-	0.5
HSK low(xmit) to HSK low(bus)	-	3
HSK low(bus) to HSK low(rcv)	-	5
HSK low(xmit) to HSK high(xmit)	8	-
HSK high to data invalid	0	--**
HSK high to HSK low	8	20000*

* within a message
** because data is output via open collector buffers, and HSK going high causes ones to be written into their output latch, data becomes invalid on the rising edge of HSK.

3.3 Bus Time-out

Within a message frame (when Bus Available is low), whenever HSK is high it has 20 ms to go low, or a bus time-out condition might result. Some master devices may take longer than 20 ms.

3.4 Data Transfer Order

All data and overhead information is sent in increments of one byte. As bytes are transmitted, the least significant nibble is sent on the bus first, followed by the most significant nibble. Whenever 16 bit (two byte) fields, such as the data length in the command message, are sent, the least significant byte is transmitted first.

SECTION 4

Bus Message Structure

4.1 Protocol

As mentioned earlier, the data bus transmits command and response messages within the context of a message frame. In general the transmission of one command message from the master device will cause a response message to be transmitted back from the slave device selected. Each message consists of several nibble transfers as described in the previous section.

Each transmitted message contains overhead information to indicate such things as the slave device selected, the command code to be performed, and the data length. The BAV (bus available) signal specifies the start of a message frame. When the master device starts a message frame it first pulls BAV low. The command message from the master then follows that falling edge of BAV. The falling edge of BAV alerts all slave devices to look for the two-nibble device code which is always transmitted first in the command message (again, least significant nibble first). The BAV signal does not return to the high level until the message frame is complete. This is illustrated in Figure 4.1 below.

Figure 4.1
Message Frame

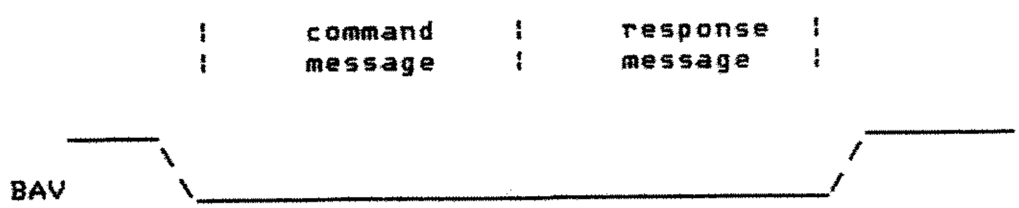


Table 4.1
Message Timing Parameters
(Microseconds)

Item	Min	Max
BAV low to HSK low	5	20000
HSK high to BAV high	1	—
End of command to start of response	10	—
HSK high to HSK low	8	20000
BAV high to BAV low (Within a message frame)	8	—

The first two nibbles of the command message always contain the device code of the slave device to be addressed. All devices on the bus will read this number and test for a match. After the device code has been sent all devices except the one selected will ignore all further data in the message. The hardware will be designed such that they will not have to participate in the handshake sequence until the next falling edge of BAV. In this way the bus will operate at the maximum data rate of the two "talking" devices once the device code has been transmitted.

Any device may extend the time to process data or wait for an operation to complete by holding HSK low until it is ready to start the next operation. Whenever HSK is high during a message, it must go low within 20ms or the receiver may time out (Time outs are not required for peripheral slave handlers).

The command and response messages are detailed in a following section.

As mentioned previously, each device has a unique device code. In addition, the device code >00 is reserved to be recognized by all devices, but only the reset and null commands are valid. All other commands for device code >00 should be ignored by the peripherals (no response message).

4.2 Poll Request

The master device may tell a slave through a normal message frame that it can request service from the master. When that has been done the slave may interrupt the master by pulling the BAV signal low. When the slave device pulls BAV low that initiates a message frame where the master polls a slave device which has been enabled to request service. The "service request poll" command code is used for this. Figure 4.2 illustrates the standard request sequence and the way BAV is driven. The master will poll one of the enabled devices during the message frame. If several devices have been enabled, then the master may not poll the requesting device in the current frame. If that happens then the polled device will say "No", and the service request handler will set up to poll the next device and return to the interrupted application. After waiting for a period, the requesting device will pull BAV low again to initiate another poll. The master device will cycle through the devices which are enabled so the requesting device will be polled eventually.

If the master receives an interrupt while in bus master mode, and the application has not yet acknowledged a previous service request from the current device, then the service request handler will transmit a null operation message down the bus. If no devices are enabled for service requests, then the IOS will ignore the interrupt.

Figure 4.2
Service Request Sequence

```
                ________        | command | response |      ______
                        \       |                         /
BAV(composite)           \______|________________________/
                                |
                                |   ______________________________
                ________        |
                        \       | /slave sees HSK
BAV(slave)               \______|/ go low and
                                |  releases BAV
                                |
                ___________     |                         ______
                           \    |                        /
BAV(master)                 \___|________________________/
                                |master pulls
                                |HSK low (1st nibble
                                |of transmission)
```

Because of timing uncertainties the service request may nearly coincide with a normal message frame. In that case, illustrated in Figure 4.3, the master device will miss the request and proceed with the normal frame. The requesting device will treat this similar to the case where another device was polled and re-assert the request after the current message is complete. A difference arises when the master is coincidentally sending a normal message when the device requests service. This might occur when the master sends a message to the requesting peripheral or to another peripheral. In that case the requesting device will either see that the device code is not its device code, or that the command code is not the "service request poll" code and must process it normally. The requesting device would then re-request service after the message is complete.

Figure 4.3
Service Request Miss

```
                                  | command | response |
                 _________                               _______
                          \                             /       \
BAV(composite)             \___________________________/         \_____
                                                        | t-wait |
                 _________        _______________________________
                          \      /                               \ re-request
BAV(slave)                 \____/                                 \_____
                           |t-low |
                 __________                               _______
                           \                             /       \
BAV(master)                 \___________________________/         \_____
```

Table 4-2
Polling Timing Paramemters

Item	Min	Max(us)
t-low	** see note **	
t-wait	2000	--

Note: The slave device will wait for the first HSK from the bus master before releasing BAV.

This specification allows multiple devices to be requesting service without any unresolvable conflicts.

4.3 Command Message

As mentioned earlier, communication between the calculator and a peripheral consists of a command message and a response message (In some cases, there will be no response to the command).

The following data is contained in a command message:

Field	Bytes
Device code	1
Command code	1
Logical Unit Number	1
Record number	2
Buffer length	2
Data length	2
Data	variable

NOTE

All data structures shown in this specification are examples of how the data can be organized, and are not meant to restrict the use of other internal data structures.

A bus master (console) should be able to transmit and receive data at a minimum rate of 3000 bytes/second. This accomodates peripherals that require a minimum I/O transfer rate.

TMS 9900 ≈ 8000 bytes/second

Each device will have a unique device code (255 codes are available). Devices with several separately addressable units will have several device codes - one for each unit. Device codes have been assigned as follows (tentative);

Device Code (in decimal):

0	- all devices
1-8	- tape mass storage
10-11	- printer/plotter
12-13	- reserved for printers
14-15	- Low cost printer
16-17	- Brother 80 column printer
18-19	- reserved for printers
20-23	- RS-232 interface
30-33	- TV interface (color)
40-43	- TV interface (black and white)
50-53	- Centronix parallel interface
60-67	- calculator in slave mode (H.C.=60-63)
70-73	- modem
80-87	- GPIB interface
90-93	- bar code reader
94-95	- scanner
100-109	- Floppy disk drive
110-117	- micro floppy
240-244	- RAM based file manager
245-254	- reserved for future console resident DSRs
255	- Console display

Device code 0 is normally used with the Null Operation or bus reset commands. The I/O subsystem will always return with a time-out error in those cases. The various fields in the command message are defined as follows:

4.3.1 Device Code.

This selects the slave device which is to respond to the command.

4.3.2 Command Code.

This field tells the slave device the nature of the operation to be performed. The following lists the standard code assignments:

Command Code (in Hexadecimal):

00 - open

01 - close
02 - delete open file
03 - read data
04 - write data
05 - restore file
06 - delete
07 - return status
08 - service request enable
09 - service request disable
0A - service request poll
0B - you are the master
0C - verify read/write operation
0D - format and certify media
0E - catalog directory
0F - set options
10 - transmit break
11 - protect/unprotect file
12 - read sectors
13 - write sectors
14 - modify file name
15 - read file descriptor
16 - write file descriptor
17 - read file sectors
18 - write file sectors
19 - load
1A - save
1B - inquire save
1C - home computer status
1D - home computer verify
FE - null operation
FF - bus reset

The specific actions of each of these commands is described in section 5. Certain peripherals may extend this list for device dependent features. Groups of peripherals with similar extensions to this standard will have similar command codes for those extensions. For example video display peripherals would use the same command codes for direct screen access and could have a command code which returns screen characteristics such as numbers of lines and characters per line. Command codes >50->EF are reserved for device dependent commands.

4.3.3 Logical Unit Number - LUNO.

This is reserved for use by devices which may contain separately addressable segments on one physical unit (e.g. files on disks). Each currently open file on a device must have a unique non zero code in this byte so that the device may continue to relate the commands to the proper files: the open command provides both the LUNO and the file name. Certain BASIC commands that include an OPEN (such as OLD, LIST, SAVE) may specify LUNO zero so each peripheral must also be able to accept LUNO zero. Note that LUNOs are only checked by peripherals that allow multiple files to be open at one time (such as a floppy disk). Peripherals that don't have files or only allow one file to be open at a time are not required to check the LUNO for validity.

4.3.4 Record Number.

The record number is reserved for future devices which support relative record (random access) files or for devices which extend the standard command code set (e.g. display address for a write screen command on the TV interface). It is ignored when a file is opened as a sequential file. This field must be maintained by the application software for compatibility with possible future random access devices. It should be zeroed before the open and restore operations for normal access to devices. It should also be incremented by the application after successful read, write, and verify operations. The first record of a file is record 0. Peripherals that do not support random access may ignore this field.

4.3.5 Buffer Length.

This field indicates the size of the data buffer for receiving data from a peripheral during the current bus operation. It is used by the master's IOS to check that the length of the returned data does not exceed the buffer size. If the master's IOS determines that more data is being received than can be put into the buffer, the operation will be aborted, and a bus time-out will result. The data returned by the peripheral in the response message must not exceed the length specified here. This length is exclusive of the data length and return status bytes which form part of the response message (The status byte and data length replace their old fields in the PAB).

4.3.6 Data Length.

This field gives the number of bytes of data which follow in the data field.

4.3.7 Data.

This field contains data to be written to the peripheral device. The use of the data depends on the command code. If the data length field is zero then the data field is not present.

4.4 Response Message

The response message contains the following data:

field	bytes
Data length	2
Data	variable
Operation status	1

4.4.1 Data Length.

This field specifies the number of bytes of data which follow in the data field.

4.4.2 Data.

This field contains the data to be returned to the master device; for example on a read data operation. If the data length field is zero then this field will be omitted. Whenever the true data length cannot be determined at the time the length is sent, the data field will be padded with trailing blanks (>20) for display type files, and with trailing zeroes for internal type files (i.e. fixed length internal disk files).

4.4.3 Operation Status.

This field contains a status of the operation. The following lists the assigned response codes in both hexadecimal and decimal:

Hex	Dec	Description
00	0	normal operation completion
01	1	device/file option error
02	2	attribute error
03	3	file/device not found error
04	4	file/device not open error
05	5	file/device already open
06	6	device error
07	7	EOF error
08	8	data/file too long error
09	9	write protect error
0A	10	not requesting service (response to poll inquiry)
0B	11	directory full error
0C	12	buffer size error
0D	13	unsupported command error
0E	14	file not opened for write
0F	15	file not opened for read
10	16	data error (checksum failure in device)
11	17	file type (relative/sequential) incorrect or not supported
12	18	improper file protection information specified
13	19	append mode not supported
14	20	output mode not supported
15	21	input mode not supported
16	22	update mode not supported
17	23	file type (internal/display) incorrect or not supported
18	24	verify error
19	25	low batteries in peripheral
1A	26	uninitialized media
1B	27	peripheral bus error (timing error)
1C	28	file is delete protected
1D	29	cartridge not installed as mass storage
1E	30	restore not allowed in this mode
1F	31	invalid file name
20	32	media full
21	33	attempted to exceed maximum number of lunos
22	34	invalid data (too short or incorrect contents)
FE	254	illegal in slave mode
FF	255	bus time out error

Error codes >50->>EF are reserved for device dependent errors.

SECTION 5

I/O Subsystem Access

This section describes the Peripheral Access Block (PAB) and the I/O subsystem calling protocol. Some portions of this section are redundant with previous sections but are included here for completeness.

5.1 Calling Sequence

To call the I/O subsystem the following steps are taken.

1. Place the command code and other information in the PAB.

2. Place the address of the PAB in the floating point accumulator (FAC - two bytes at address >75). The most significant byte of address is placed at >75 and the least significant is placed at >76.

3. Call the I/O subsystem.

4. Test the status byte in the PAB (see below) for an error condition (The first two bytes of FAC will be returned pointing to the status byte of the PAB).

5.2 Peripheral Access Block

The PAB provides information for the I/O subsystem to access a peripheral device. The PAB contains the following fields in the following order. The first field is at the address specified by the PAB pointer in FAC. The other bytes are stored at successively lower addresses. Also, the buffer pointed to by the PAB buffer address is stored from high to low addresses in memory.

field	bytes	to/from
Device code	1	to
Command Code	1	to
Logical Unit Number	1	to
Record number	2	to
Buffer length	2	to
Data length	2	both
Return status	1	from
Buffer address	2	--
* Link to next block	2	-- (only in service request PAB's)
* Service Flag	1	-- (only in service request PAB's)
* DSR Address	2	-- (only in service request PAB's)

NOTE

The last three fields (asterisked) are for use in SRPAB's. If the PAB is not going to be used with a device enabled for service requests, then those fields are not necessary.

The DSR Address is for future expansion of the bus. It is meant to be used to transfer control to a DSR immediately following the successful completion of a service request. It should be left zeroed for compatibility purposes (see section 5.6 - Polling Operation).

5.3 Standard Access

The access to a peripheral device is performed with a sequence of I/O calls. Before using a device it should be opened with an I/O call using the open command. This may be followed by other I/O calls to read or write data or perform other functions. When the application finishes using a device it should issue a close I/O call. This will ensure that any necessary device dependent actions have been performed. For instance, mass storage devices often keep file information in a local RAM memory and update the file directories on the media when the device is closed. The bus reset command code (>FF) will also close all open devices, but may have other undesirable actions as well.

5.4 Command Descriptions

This section describes the PAB setup and device response for the various standard I/O command codes. This includes codes >00(open)->11(protect/unprotect file), >FE, and >FF (bus reset).

5.4.1 Open - 00.

This command code is used to initiate the use of a device (or file on a device). A device will check access modes and ensure that the device is not already open. The PAB should be set up as follows:

field	data
Device code	as required
Command Code	00
Logical Unit Number	unique value (0 is a special case)
Record number	0000
Buffer length	as required (at least 0004)
Data length	as required (0004 returned)
Return status	(returned)
Buffer address	as required

The data buffer contains the following which is sent to the peripheral.

Input buffer length (2 bytes)
(if zero then device returns buffer size)
device attributes (1 byte)
device options (if any)

The peripheral will compare the 'Input buffer length' to its capabilities and return either the requested length, the default length if the requested is zero, or an error if the requested length is unacceptable. The input buffer length is used by the master to determine what size buffer should be allocated by the master for read operations. Thus software could open a device such as an RS-232 peripheral with a one byte input buffer, and write 80 character records out to the RS-232. Each peripheral is required to check the length of data being transferred to it in each write operation, and to generate an error if that data is longer than the maximum write acceptable to the peripheral.

The device attributes byte contains flags used to indicate the access mode of the peripheral. Several bits are unused by the I/O scheme and may be used by the application software as desired.

The bit definitions for the attributes byte are as follows (bit 0 is the least significant bit):

7-6 - access mode
- 00 - append mode (write only at end of file)
- 10 - output mode (write only)
- 01 - input mode (read only)
- 11 - update mode (read or write)

5 - relative(1)/sequential(0)
4 - fixed(1)/variable(0)
3 - internal(1)/display(0) file type
2 - device dependent use
1-0 - used by console and must be ignored by peripheral

The access mode must match the capabilities of the particular device:

1. Append mode specifies that data will only be written (or appended) to the end of a file (Note that a peripheral opened in append mode will return the number of the next record to be written to in the record number field of the returned data in the open).

2. Output mode specifies that data will only be written to the device and the "read data" command will not be used.

3. Input mode specifies that data will only be read from the device and the "write data" command will not be used.

4. Update mode means that data may be both read and written.

For instance, the following peripherals will be expected to support the given access modes:

calculator in slave mode	input, output, update
modem	input, output, update
RS-232	input, output, update
TV-interface	input, output, update
bar code reader	input
tape mass storage	input, output, update*

* update and append only work with the last file on the tape

Bit 2 may be used to specify another device attribute as an extension to this standard. Software which does not use any special device dependent feature should set this bit to zero.

Note that data transferred on the bus is variable length. If a peripheral determines that data should be formatted as fixed length records, then it is up to that device to perform the formatting.

The device options field is a variable length field which contains device-dependent information relative to setting up the device. Examples are information for the RS-232 baud rate or the file name for a file oriented device. This is normally ASCII (character) data.

Examples:

```
10 OPEN #1,"1.DATAFILE",INPUT
          opens a file on device 1 with name
          'DATAFILE' for input

10 OPEN #1,"10.C=L,S=0",OUTPUT
          opens  device 10 for  output  with
          options 'C=L,S=0'
```

If fewer than 3 bytes of data are sent to the peripheral (buffer length and attributes), then it should error off (error code 2, i.e. not enough information was provided to justify opening the peripheral).

The response buffer will contain the accepted buffer length, and the record number that the file was opened to. For some devices this record number will be meaningless, so a zero (0) will be returned. This information is always returned when a device or file has been successfully opened. Thus the response message for a successful open will be:

Data length		0004
Data	(2 bytes)	Accepted buffer length
	(2 bytes)	Record position *
Operation status		0

* The master device should place the record number in the PAB for the first I/O call that follows.

An unsuccessful open may not return data in the response message. The operation status byte may contain the following error codes:

00 - successful open
01 - device/file option error
02 - error in attributes byte **
05 - file/device already open
06 - device related error
09 - device/file write protected
0B - directory full error
0C - buffer size error
11 - file type (relative/sequential) incorrect
    or not supported
13 - append mode not supported
14 - output mode not supported
15 - input mode not supported
16 - update mode not supported
17 - file type (internal/display) incorrect
    or not supported
19 - low batteries in peripheral
1A - uninitialized media
1B - error detected in bus transfer
20 - media full

** - This error code may be used in place of error codes >11-17 when code space prohibits the use of multiple errors.

5.4.2 Close - 01.

This command terminates the use of a device. Depending on the device this command may be used to clean up internal data (e.g. write an end of file) or may be effectively ignored. In general a close command must be sent between using a device and another open command. If a close is received when the peripheral has never been opened, error 4 will be returned. The data length for the close command will be zero (no command buffer is transmitted). The PAB should be set up as follows:

field	data
Device code	as required
Command Code	01
Logical Unit Number	as in open
Record number	don't care
Buffer length	don't care
Data length	0000 (0000 returned)
Return status	(returned)
Buffer address	don't care

The response message will only contain a status byte and a zero data length (two bytes). The error status indications are;

00 - device or file closed
04 - device or file never opened
06 - device related error
08 - data/file too long error

5.4.3 Delete open file - 02.

Sometimes, it is desired to delete a file upon completion of I/O to the file (such as reading some data and deleting the old file). This command may be used in place of the close command, and causes the open file to be deleted. The PAB should be set up as follows:

field	data
Device code	as required
Command Code	02
Logical Unit Number	as in open
Record number	don't care
Buffer length	don't care
Data length	0000 (0000 returned)
Return status	(returned)
Buffer address	don't care

The response message will consist of no data (data length = 0) and the return status. The following error status indications may occur:

00 - file deleted
04 - file not open
06 - device error
09 - write protect error
0D - command not supported
1C - file is delete protected

5.4.4 Read Data - 03.

This command is used to request data from a device. The command message will contain a zero-length data field. The PAB should be set up as follows:

field	data
Device code	as required
Command Code	03
Logical Unit Number	as in open
Record number	as required
Buffer length	as determined in open
Data length	0000 (length returned)
Return status	(returned)
Buffer address	as required

The response message will contain the data requested. The record number field should be incremented by the application after each successful read data call to be compatible with random access devices. The following error status codes may occur:

00 - read successful
02 - attribute error
04 - file/device not open error
06 - device error
07 - EOF error
0C - buffer size error
0D - command not supported
0F - file/device not open for read

5.4.5 Write Data - 04.

This command is used to send data to a peripheral device. The command message will contain the data to be sent to the device. The PAB should be set up as follows:

field	data
Device code	as required
Command Code	04
Logical Unit Number	as in open
Record number	as required
Buffer length	don't care
Data length	record length (0000 returned)
Return status	(returned)
Buffer address	as required

The response message will contain zero-length data and the operation status. As in the read data operation, the application should increment the record number after each successful operation to support random access devices properly. The following error status indications may occur:

- 00 - write successfully completed
- 02 - attribute error
- 04 - file/device not open
- 06 - Device error
- 08 - file/data too long
- 09 - write protect error
- 0C - buffer size error
- 0D - command not supported
- 0E - file/device not open for write

5.4.6 Restore - 05.

This command is primarily used with mass storage devices to position a file to its first record. The PAB should be set up as follows:

field	data
Device code	as required
Command Code	05
Logical Unit Number	as in open
Record number	don't care
Buffer length	don't care
Data length	0000 (0000 returned)
Return status	(returned)
Buffer address	don't care

The following error status indications may be returned:

- 00 - restore successful
- 04 - file/device not open
- 06 - device error
- 0D - command not supported

5.4.7 Delete - 06.

This command is primarily used to remove data from mass storage devices. The PAB should be set up as follows:

field	data
Device code	as required
Command Code	06
Logical Unit Number	don't care
Record number	don't care
Buffer length	don't care
Data length	as needed (0000 returned)
Return status	(returned)
Buffer address	as required

The command data buffer will contain the name of the file to be deleted if it is appropriate to do so. The file name must be specified in the same manner as the options in the open statement. There will be no data returned. The following operation error status indications may occur:

- 00 - file deleted
- 03 - file not found
- 05 - file is open error
- 06 - device error
- 09 - write protect error
- 0D - command not supported
- 1C - file is delete protected

5.4.8 Return Status - 07.

This command is used to return device status information. The information is returned in the data buffer. Certain bit fields in the return data are assigned to standard meanings while others are reserved for device dependent extensions. Certain devices may return more bytes of status if the buffer length allows. The PAB should be set up as follows:

field	data
Device code	as required
Command Code	07
Logical Unit Number	as in open or 00
Record number	as in previous call
Buffer length	>= 0001
Data length	0000 (0001 returned)
Return status	(returned)
Buffer address	as required

When the LUNO is given as zero, then general device status will be returned. The device does not have to be open for this command (although mass storage peripherals that only support one open file will return file status when given a LUNO of 00). If the LUNO is non-zero, then status will be given for the open file referenced by the LUNO. The bit fields in the return data are as follows (bit 0 is the least significant bit).

7 - end of file has been reached (1=true, 0=false)
6 - random access supported (1=true, 0=false)
5 - file is protected (1=true, 0=false)
4 - file/device open (1=true, 0=false)
3-2 - file/device type
  0 - display type 2 - data communications
  1 - internal data type 3 - undefined
1-0 - I/O modes that the file/device can be opened in
  0 - undefined 2 - write only
  1 - read only 3 - read/write

When the buffer size permits, a data communications type device will include 2 extra bytes indicating the number of bytes that remain in its output buffer (LSB of the length first).

The following operation error status indications may occur:

04 - file/device not open (issued if incorrect LUNO used)
12 - buffer size error

5.4.9 Service Request Enable - 08.

The following SRPAB contents are used to enable a device for service requests. Most fields are not normally used (e.g. file type devices are not expected to support polling). An open call is normally required before this I/O call to initialize device parameters.

field	data
Device code	as required
Command Code	08
Logical Unit Number	don't care
Record number	don't care
Buffer length	don't care
Data length	normally 0000
Return status	(returned)
Buffer address	buffer address for device
Link to next SRPAB	as is
Service Flag	0

NOTE

The application is responsible for resetting the service flag after acknowledging a service request completion. The buffer provided by the application must be separate from that in the open to allow for normal bus operation with that device.

The following error status codes may be returned:

00 - command acknowledged
06 - device error
0D - command not supported
0F - file/device not open for read

5.4.10 Service Request Disable - 09.

To disable a device from polling the application should send an I/O call to disable the device and then remove the SRPAB from the list. The following SRPAB contents are used to disable a device for service requests. Most fields are not normally used.

field	data
Device code	as required
Command Code	09
Logical Unit Number	don't care
Record number	don't care
Buffer length	don't care
Data length	normally 0000
Return status	(returned)
Buffer address	normally don't care
Link to next block	Don't care
Service Flag	Don't care

The following error status code may be returned:

00 - service requests disabled
0D - command not supported

5.4.11 Service Request Poll - OA.

This command allows a bus master to query a peripheral as to whether it requested service from the master. The PAB should be set up as follows:

field	data
Device code	as required
Command code	OA
Logical Unit Number	don't care
Record number	don't care
Buffer length	as determined in open
Data length	0000 (data length returned)
Return status	(returned)
Buffer address	as required
Link to next SRPAB	as required
Service flag	0 (to allow the poll operation)

The response message will consist of zero or more bytes of data, and the return status. The following error status indications may occur:

OA - not requesting service (unsuccessful poll)
OD - command not supported (unsuccessful poll)

* Any other error codes indicate a successful poll operation, and reflect the reason for the service request.

The IOS directly supports service requests by maintaining the SRPAB pointer and sending the 'Service Request Poll' command whenever a bus interrupt is received. Thus, this message is never actually sent by an application.

A successful response to a service request poll will usually include data that had been received by the peripheral polled. An example is a modem that had received some data, and had interrupted the master to transmit that data to it.

5.4.12 You are the Master - 0B.

This command allows a bus master to transfer control of the bus to another device. The PAB should be set up as follows:

field	data
Device code	as required
Command code	0B
Logical Unit Number	don't care
Record number	don't care
Buffer length	don't care
Data length	0000 (0000 returned)
Return status	(returned)
Buffer address	don't care

The response message will consist of no data (data length = 0) and the return status. The following error status indications may occur:

00 - you are now a slave device
0D - command not supported

5.4.13 Verify read/write operation - OC.

This command allows a program to verify the accurracy of a read or write to or from a device. After a record is read or written, the record is sent back to the peripheral. The returned status will show whether the record was verified. The PAB should be set up as follows:

field	data
Device code	as required
Command code	OC
Logical Unit Number	as in open
Record number	as in READ/WRITE
Buffer length	don't care
Data length	record length (0000 returned)
Return status	(returned)
Buffer address	as required

The response message will consist of no data (data length = 0) and the return status. The following error status indications may occur:

00 - record verifies
04 - file not open
06 - device error
OC - program length mismatch (verify error)
OD - command not supported
10 - data/checksum error
18 - verify error

5.4.14 Format Media - OD.

This command tells a mass storage device to format the media used by the device. The return status will indicate whether the format was successful. The PAB should be set up as follows:

field	data
Device code	as required
Command code	OD
Logical Unit Number	don't care
Record number	don't care
Buffer length	as required
Data length	as required
Return status	(returned)
Buffer address	as required

If the data length in the command message is O, the media (wafertape, diskette, etc.) will be formatted with the default options, and no data will be returned. In the case of the Hexbus floppy disk peripheral, a non-zero data length means that the data following will specify the options chosen by the user. A non-zero data length to the floppy will also cause the total number of sectors formatted to be returned if the operation was successful and the buffer length allows. The following return status indications may be returned:

- 00 - operation successful
- 05 - file or device open
- 06 - device error
- 09 - media write protected
- OD - command not supported

5.4.15 Read Catalog - OE.

This command tells a mass storage peripheral to return basic information about a particular file. It can be used to catalog an entire directory through multiple uses of the command. The PAB should be set up as follows:

field	data
Device code	as required
Command code	OE
Logical Unit Number	don't care
Record number	file number
Buffer length	at least >0012
Data length	0000 (0012 returned)
Return status	(returned)
Buffer address	as required

The data buffer returned will contain the following information:

File number	1 byte
File name	12 bytes
Maximum record length	2 bytes
Number of records	2 bytes
Device dependent flags	1 bytes

The following error status codes may occur:

- 00 - operation successful
- 03 - file not found
- 05 - file already open
- 06 - device error
- 0C - buffer length error
- 0D - command not supported

By incrementing the file number with each call, a complete directory of a device may be obtained. The end of the directory is indicated by error 03 (file not found) being returned.

5.4.16 Set Options - OF.

This command is used to send device options as described in the open command without re-opening the device. The command data buffer will contain the device options which are device dependent. The PAB should be set up as follows:

field	data
Device code	as required
Command Code	OF
Logical Unit Number	as in open
Record number	don't care
Buffer length	don't care
Data length	as required (0000 returned)
Return status	(returned)
Buffer address	as required

The response message will consist of no data (data length = 0) and the return status. The following error status indications may occur.

- 00 - options changed
- 01 - error in options
- 04 - device or file not open
- 06 - device error
- 0D - command not supported

5.4.17 Transmit Break - 10.

This command is used to have a data communications device transmit a continuous break (space condition) for .25 seconds. The peripheral will send all the data in its buffer out on the communications lines, then it will set the communications line to send a continuous break, wait a minimum of .25 seconds, and then return the communications lines to their mark condition. After this point the peripheral will release the bus. The PAB should be set up as follows:

field	data
Device code	as required
Command Code	10
Logical Unit Number	don't care
Record number	don't care
Buffer length	don't care
Data length	0000 (0000 returned)
Return status	(returned)
Buffer address	as required

The response message will consist of no data (data length = 0) and the return status. The following error status indications may occur.

- 00 - break transmitted
- 04 - device or file not open
- 06 - device error
- 0D - command not supported

5.4.18 Protect/Unprotect File - 11.

This command is used to modify file protection on a mass storage device. With it, a file may be write protected or unprotected, and delete protected or unprotected. The PAB should be set up as follows:

field	data
Device code	as required
Command Code	11
Logical Unit Number	don't care
Record number	don't care
Buffer length	don't care
Data length	as required
Return status	(returned)
Buffer address	as required

The data buffer contains the following information which is sent to the peripheral:

File protection flags (1 byte)
File name (length as required)

The file protection flags byte is formatted as follows (bit 0 is the least significant bit):

- 7 - file is write protected(1)/not write protected(0)
- 6 - file is delete protected(1)/not delete protected(0)
- 5-3 - reserved for future use (must be zero)
- 2-0 - device dependent use

In the case of the Hexbus floppy disk peripheral, the flags byte should be 0 for unprotected and >80 for protected (for compatibility, >80 implies both write and delete protect).

The response message will consist of no data (data length = 0) and the return status. The following error status indications may occur:

- 00 - protection modified
- 06 - device error
- 0D - command not supported
- 12 - improper file protection information specified

5.4.19 Null Operation - FE.

When the calculator receives a BAV interrupt and either no devices are enabled for interrupts or the current service flag is set, then a null operation code is sent to all devices. There will usually be no response to this message

field	data
Device code	as required (usually 00)
Command code	FE
Logical Unit Number	don't care
Record number	don't care
Buffer length	don't care
Data length	00
(all other fields not needed)	

5.4.20 Reset Bus - FF.

It may sometimes be desired to tell a device (or devices) to close all open files (or devices). This command will usually have no response by devices, they will simply perform the action requested (If they were not open to begin with, then they will do nothing). Thus, all devices will revert to their power up status. The PAB or SRPAB should be set up as follows:

field	data
Device code	as required (usually 00)
Command code	FF
Logical Unit Number	don't care
Record number	don't care
Buffer length	don't care
Data length	00
(all other fields not needed)	

* For both the Null operation and Reset bus commands, no response will be forthcoming if device code 00 is used. However, if some other device code is used, some specific devices may initiate a response. The meaning of the response is specific to the device.

5.5 Slave Mode

Five bytes in RAM are used to control the slave mode. These are also used for the polling operation when in the master mode as described below.

Flag	Length	Address(es)
Master/Slave flag	1 byte	>0805
Slave Operation Handler	2 bytes	>0806, >0807
Slave Access Block (SAB) or Service Request Peripheral Access Block (SRPAB) address	2 bytes	>0808, >0809

When the master/slave flag is set to zero the calculator operates in the normal master mode which allows I/O commands as in the previous section and polling operations for service requests as in the next section. When the master/slave flag is set to non-zero then the calculator operates in the slave mode as described in this section. In slave mode, Master Mode Handler accesses will be inhibited. The calculator will not perform polling in slave mode, as another device on the bus is then expected to do it.

When in the slave mode a request from the master device may occur at any time and will cause an INT 1 in the calculator which will enter code in the Slave Operation Handler . This code will receive the command message from the master device placing it in the Slave Access Block (SAB), then formulate and transmit a response.

5.5.1 Recommended SAB Example.

The SAB is quite similar to the PAB except that all locations contain received data and a separate buffer size is provided to ensure that data sent with the command will fit in the buffer.

The SAB:

field	bytes	received
Command code	1	x
Logical Unit Number	1	x
Record number	2	x
Buffer length	2	x
Data length	2	x
Buffer size	2	
Buffer address	2	

All locations except the buffer size and buffer address are changed when the command is received.

The response message block exactly mimics the response message as follows.

field	bytes
Data length	2
Data	data length
Operation status	1

5.6 Polling Operation

When operating in the master mode the calculator can be set up to poll one or more devices when a service request is issued. A service request (BAV goes low) will cause an interrupt in the software being run. When this interrupt occurs the I/O subsystem performs a device poll if a device is enabled for service requests. If the device was not the one that interrupted the application, then the SRPAB pointer will be changed to point to the next Service Request Peripheral Access Block and control will be passed back to the application. If the device poll was successful, then the data, which will be in the SRPAB data buffer, will be passed back to the application, and the Service flag will be set.

If the SRPAB has a non zero DSR (Device Service Routine) address, then control will be passed to the DSR. The Device Service Routine can then process the data and return to the IOS, which will restore registers A, B, and >66 through >78, and then pass control back to the application. If the most significant byte of the DSR address is zero then control will be passed directly back to the interrupted application - that application can then test for a completed poll by periodically checking the SRPAB service flag. DSR's can use registers >66 through >78, and the A and B registers. If more registers are needed, then they will have to be saved while in the DSR, and restored upon leaving.

```
PROCEDURE SERVICE_REQUEST_HANDLER;
 BEGIN
  IF SRPAB_POINTER <> 0 THEN
   BEGIN
    IF SRPAB.SERVICE_FLAG = 0 THEN {DON'T SERVICE AN INTERRUPT IF  }
                                   {THE APPLICATION HASN'T SERVICED}
                                   {THE LAST ONE                   }
     BEGIN
      SRPAB.COMMAND_CODE:=#0A;
      SRPAB.DATA_LENGTH:=0;
      CALL MASTER_MODE_HANDLER;
      IF STATUS <> 10 THEN         {CHECK TO SEE IF CORRECT DEVICE }
       BEGIN
        SRPAB.SERVICE_FLAG:=1;
        IF SRPAB.DSR_ADDRESS <> 0 THEN
         CALL (DSR_ADDRESS);       {EXECUTE THE DSR - THEN RETURN  }
       END;
     END
```

```
    ELSE
     SEND NULL OPERATION MESSAGE;
    SRPAB_POINTER:=SRPAB.NEXT_LINK;  { WE'LL POLL THE NEXT DEVICE   }
   END;                              { NEXT TIME                    }
 END;
```

* The interrupt vector handler will save the necessary registers to assure that none are changed by the handling of the interrupt.

For a description of the SRPAB, see section 5.2. (It is a PAB with the three noted fields at the end of the PAB - DSR dispatching is not supported in the current implementation). The peripheral requesting service will return a device status, and may return a data buffer. The return status byte will give a reason code for the service request. The data will be device dependent (the amount sent will depend on the buffer size given when the device was enabled for service requests).

If a device poll is unsuccessful, a 'not requesting service' error code is returned, then the polling software will load the SRPAB pointer from the link in the current block and return to the application with a return from interrupt instruction.

To set up a device for polling the SRPAB is added to the poll list and the device is enabled to request service with a "service request enable" command in a normal I/O call using the SRPAB created for the device. When an SRPAB is added to or deleted from the list the interrupts should be disabled to prevent a service request interrupt while the links are being changed. The following sequence of operations will successfully add an SRPAB to the list.

```
DISABLE_INTERRUPTS;
IF SRPAB_POINTER=0 THEN BEGIN
  LINK(NEW):=NEW;
  SRPAB_POINTER:=NEW;
END
ELSE BEGIN
  LINK(NEW):=LINK(SRPAB_POINTER);
  LINK(SRPAB_POINTER):=NEW;
END;
ENABLE_INTERRUPTS;
```

The following SRPAB contents are used to enable a device for service requests. Most fields are not normally used (e.g. file type devices are not expected to support polling). An open call is normally required before this I/O call.

field	data
Device code	as required
Command Code	08
Logical Unit Number	don't care
Record number	don't care
Buffer length	as required
Data length	normally 0000
Return status	(returned)
Buffer address	buffer address for device
Link to next SRPAB	as is
Service Flag	0

NOTE

The application is responsible for resetting the service flag after acknowledging a service request completion. The buffer provided by the application must be separate from that in the open to allow for normal bus operation with that device.

The following error status codes may be returned:

00 - command acknowledged
06 - device error
0C - buffer size error
0D - command not supported
0F - file/device not open for read

To disable a device from polling the application should send an I/O call to disable the device and then remove the SRPAB from the list. The following SRPAB contents are used to disable a device for service requests. Most fields are not normally used.

field	data
Device code	as required
Command Code	09
Logical Unit Number	don't care
Record number	don't care
Buffer length	don't care
Data length	normally 0000
Return status	(returned)
Buffer address	normally don't care
Link to next block	Don't care
Service Flag	Don't care

The following error status codes may be returned:

00 - service requests disabled
0D - command not supported

The following sequence will successfully delete an SRPAB from the list. Note that it does not allow for error conditions such as the SRPAB not appearing in the list or the SRPAB_POINTER=0.

```
{ OLD = address of SRPAB to be deleted }
DISABLE_INTERRUPTS;
WHILE LINK(SRPAB_POINTER)<>OLD DO
  SRPAB_POINTER:=LINK(SRPAB_POINTER);
LINK(SRPAB_POINTER):=LINK(OLD);
IF SRPAB_POINTER=OLD THEN SRPAB_POINTER:=0;
ENABLE_INTERRUPTS;
```

NOTE

If a device that has been enabled for interrupts is going to be accessed as a normal slave device while interrupts are enabled, a PAB must be established for that device (separate from the SRPAB) for master mode accesses. Otherwise, the integrity of the SRPAB will be threatened by a device interrupt while preparing for the I/O operation. Interrupts should not be disabled during preparation of a PAB or SAB to prevent severe performance degradation of the system.

SECTION 6

Notes

6.1 Peripheral Prompts

No provision is made for peripherals which must (or would like to) prompt the user. However, provision has been made to allow extensions to the language to be made to support specific peripherals. Thus, special commands to support a device could be added to make the system more 'user friendly'.

6.2 Device Options

The device options are included in the open or set options I/O calls. this data will be represented in ASCII in order to allow friendly statements such as:

```
OPEN #1,"100.MYFILE",UPDATE   for a file on a disk
OPEN #1,"3.B=300,P=O",OUTPUT  for RS232 at 300 baud odd parity
```

The open buffer would contain the text within the quotes after the decimal point and would be in ASCII.

6.3 Notes on the Record Number

The record number field is primarily used by devices that support random access files (i.e. disk drives). Such a device would use the record number field to determine which record to next perform I/O on (if the file was opened for random access). To support this type of access, the application running in the master must increment the record number for each successful read or write operation so that the correct record number is always available to the peripheral if it needs it.

SECTION 7

Bus Transfer Examples

7.1 Example of Open Command

The following example gives an open statement in BASIC, then shows how the I/O transfer would occur that corresponds to that statement:

100 OPEN #1,"20.B=4800,P=O",OUTPUT
Opens device 20 as file 1 (the file number in BASIC is used as the LUNO) for output. Options are B=4800 (4800 baud) and P=O (Odd parity).

Command:

```
  Transmitted                     Transmitted
     PAB                             Data
+-------+                        +-------+          +-------+
| >14   | Device number          |  >00  |          |   ,   |
+-------+                        +-------+          +-------+
| >00   | Command code           |  >00  |          |   P   |
+-------+                        +-------+          +-------+
| >01   | LUNO                   |  >80  |          |   =   |
+-------+                        +-------+          +-------+
| >00   | Record number          |   B   |          |   O   |
+-------+                        +-------+          +-------+
| >00   |                        |   =   |
+-------+                        +-------+
| >04   | Buffer length          |   4   |
+-------+                        +-------+
| >00   |                        |   8   |
+-------+                        +-------+
| >0D   | Data length            |   0   |
+-------+                        +-------+
| >00   |                        |   0   |
+-------+                        +-------+
```

Response:

```
+------+
| >04  | Data length
+------+
| >00  |
+------+
| >50  | Data - Accepted Buffer Length
+------+
| >00  |
+------+
| >00  | Data - Record Number
+------+
| >00  |
+------+
| >00  | Status
+------+
```

Note that the Input buffer length in the open was zero, so the peripheral returned the default buffer length that it wanted. BASIC would then use this information for allocating an 80 byte buffer (the default was >50 or decimal 80) for all subsequent I/O operations with that peripheral until it is closed (note that the peripheral would accept writes of longer than 80 bytes, but would transmit a maximum of 80 characters at a time to the master).

7.2 Example of a Read Command

The following example gives an example of an input statement in BASIC, then shows how the I/O transfer would occur that corresponds to that statement:

510 INPUT #1,A$

Inputs a string variable from file number one (for the purpose of this example assume that the device opened in the above open statement is the device being input from).

Command:

Transmitted PAB

Byte	
>14	Device number
>03	Command code
>01	LUNO
>00	Record number
>00	
>50	Buffer length
>00	
>00	Data length
>00	

Response

Byte	
>05	Data length
>00	
>32	Data - ASCII 27295
>37	
>32	
>39	
>35	
0	Status

Note that no data was transmitted in this example, and 5 bytes of data were returned to the master (the characters '27295'). This data will be stored in the 80 byte buffer allocated by BASIC, and then processed by the INPUT routine.

HEX BUS

Electrical Specifications

TI# 1056477

T. I. APPROVALS

ELECT. ENGR.	DATE
PGM. MGR.	DATE
PGM. MGR.	DATE

Consumer Products Group
Calculator Division

9/19/83

SECTION 1

OVERVIEW

1.1 Purpose of This Document

This document is a complete electrical specification for the Hex Bus (Tm) CMOS bus used to connect intelligent peripherals to calculator or small computer type products. This document is meant to be used in formulating the detailed electrical design.

1.2 Scope of this document

This document describes the electrical characteristics of the Hex Bus. Because the primary purpose is assisting in electrical design, software is not discussed in this document. For user and software assistance, see the INTELLIGENT PERIPHERAL BUS: STRUCTURE, TIMING, AND PROTOCOL SPECIFICATION.

1.3 Organization of This Document

The electrical specifications of the Hex Bus are described individually in the sections listed below.

Section 1 - Signals Description

Section 2 - Timing Characteristics

Section 3 - Electrical Characteristics

1.4 Terminology

HSK - Handshake I/O Control Line

BAV - Bus Available I/O Control Line

DO-D3 - Four Data Lines on the IPB

1.5 Related Documents

This document is a general specification for the Hex Bus hardware only. Other related documents are listed below.

INTELLIGENT PERIPHERAL BUS CONTROLLER SPECIFICATION	- TI#1052911
CABLE SPECIFICATION: 8-CONDUCTOR PERIPHERAL	- TI#1044500
INTELLIGENT PERIPHERAL BUS: STRUCTURE, TIMING, AND PROTOCOL SPECIFICATION	- TI PRELIMINARY
HEX BUS GENERIC I/O PRODUCT SPECIFICATION	- PENDING

SECTION 2

SIGNALS DESCRIPTION

2.1 General Information

the IPB contains eight lines, of which seven are currently in use. The function of each of these seven are discussed in this section. The eight lines are listed below with a brief description of each. The diagram at the bottom of the page indicates the relative location of each line in the Hex Bus cable. For more information refer to the CABLE SPECIFICATION - TI#1044500.

DO-D1 - Two LSB I/O Data Bits - - - - - - - - - - Pin 1, 2

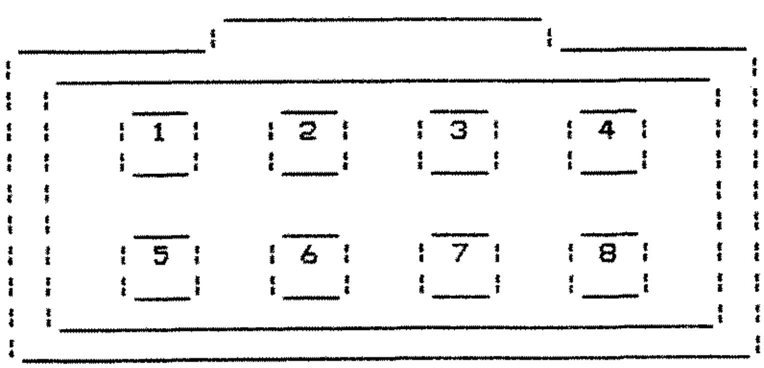
BAV - Bus Available, I/O Traffic Control Line - Pin 3

GND - Common Ground Line - - - - - - - - - - - Pin 4

HSK - Handshake, I/O Timing Control Line - - - Pin 5

FUT - Reserved for Future Use - - - - - - - - - - Pin 6

D2-D3 - Two MSB I/O Data Bits - - - - - - - - - - Pin 7, 8



RELATIVE PIN LOCATION WHEN LOOKING INTO THE CONNECTOR OF A HEX BUS CABLE.

2.2 HSK - Handshake, I/O Timing Control Line

The HSK line is used by both the master and the peripherals to signal the fact that data is on the line. HSK is an open-drain, CMOS output in order to allow it to be pulled low by any device on the bus. A device will pull HSK low to signal to the other devices on the bus that data is available from that device which may be read by the other devices on the bus. In order to prevent cross-talk between HSK and the data lines (which may cause unwanted spikes on the HSK line) the data lines should be gated out to the bus by the low level of the HSK.

HSK will be held low by the recieving devices until they have latched the four bits of data on the bus. If the transmitter is slower than the recievers, the transmitter could be the device which determines when the HSK line will go high. Refer to the diagram on the following page for a better understanding of this communication proceedure.

For more information on the timing involved in this transaction, refer to the TIMING CHARACTERISTICS section of this document.

Hex Bus Handshake Sequence

```
          __                                                                __
BAV         \                                                              /
             \____________________________________________________________/

          ____                ____                 ____                  __
 HSK          \              /    \               /    \                /
(COMPOSITE)    \____________/      \_____________/      \______________/
                |<- HSK ->|         |<- HSK ->|          |<- HSK ->|
          ____________________________________________________________________
DO-D3     XXXXXX|<-DATA ->|XXXXXXXXX|<-DATA ->|XXXXXXXXX|>-DATA ->|XXXX
          XXXXXX|_________|XXXXXXXXX|_________|XXXXXXXXX|_________|XXXX
```

===

HSK COMPONENTS - SLOW RECIEVER

```
          ____            __________           __________          ________
 HSK          \          /          \         /          \        /
(TRANS)        \________/            \_______/            \______/
                |HSK|                 |HSK|                |HSK|
                |<- data valid        |<- data valid       |<- data valid

          ______          ______           ______           __
 HSK            \        /      \         /      \         /
(SLOW RECV)      \______/        \_______/        \_______/
                  |<-HSK->|        |<-HSK->|        |<-HSK->|
          data received ->|  data received ->|  data received ->|
```

===

HSK COMPONENTS - SLOW TRANSMITTER

```
          ____              ____                 ____                __
 HSK          \            /    \               /    \              /
(SLOW TRANS)   \__________/      \_____________/      \____________/
                |<- HSK ->|       |<- HSK ->|          |<- HSK ->|
                |<- data valid    |<- data valid       |<- data valid

          ______        __________           __________          ______
 HSK            \      /          \         /          \        /
(RECV)           \____/            \_______/            \______/
                  |HSK|              |HSK|                |HSK|
      data received ->|  data received ->|  data received ->|
```

2.3 BAV - Bus Available; I/O Traffic Control Line

Each message on the I/O consists of two parts, the command message and the response message. Whenever a message transfer is in progress, BAV will be held low by the master peripheral until the message is complete. This tells the other peripherals that the bus is in use. BAV can be taken low by the master or a slave peripheral but may only be held low by the master. See the diagrams below. For more information on the timing involved in this transaction, refer to the TIMING CHARACTERISTICS section of this document.

BAV Polled by Master

```
                   !  command    !  respond    !
                   !  message    !  message    !
      ___________                               _______________
                 \                             /
BAV               \___________________________/
```

==

BAV Polled by Slave - Held by Master

```
                          !  command    !  respond    !
                          !  message    !  message    !
      __________________                               _______________
                        \                             /
BAV(composite)           \___________________________/

      _________________        _____________________________________
                       \      /
BAV(slave)              \____/ (released when HSK goes low)
                            !
                            !
      ____________________  !                        _______________
                          \ !                       /
BAV(master)                \_______________________/
                            !
                            !
      _____________________ !                        _______________
                           \!  XXXXXXXXXXXXXXX/(HSK goes high - low
HSK(composite)              \___XXXXXXXXXXXXXX/  as nibbles of data
                                                 are transmitted)
```

2.4 DO-D3 - Four I/O Data Bits

These four bits are used to send data on the Hex Bus one nibble at a time. The lower nibble of the 8 bit byte will be sent first, followed by the most significant nibble. The data lines are configured as shown below.

Data Line Configuration

D3 - Most Significant Bit
D2 - Data Bit
D1 - Data Bit
DO - Least Significant Bit

2.5 GND - Floating Reference Ground Line

A common ground line is run to all the peripherals in order to reference all voltages equally to the same point. This is a signal reference and should not be used as a power line or tied to earth-ground.

2.6 FUT - Reserved for Future Use

SECTION 3

TIMING CHARACTERISTICS

3.1 General Information

This section will deal with the timing of the various signals of the Hex Bus. The first part will deal with the timing requirements of the signals listed in Sections 1 and 2. The second part will deal with actual rise and fall timing perameters of the signals.

Handshake Timing Parameters

Item	Min. (usec.)	Max. (usec.)
HSK low to data valid	-	0.5
HSK low(xmit) to HSK low(bus)	-	3
HSK low(bus) to HSK low(rcv)	-	5
HSK low(xmit) to HSK high (xmit)	8	-
HSK high to data high	0	2.5*
HSK high to HSK low	8	20,000**

* Because data is output via open drain buffers, and HSK going high causes ones to be written into their output latch, data must be set high on the rising edge of HSK.

** Within a message (when BAV is held low) HSK has 20 ms to go low or a bus timeout (error) condition results.

BUS HANDSHAKE TIMING

```
          ______________                  _____________
                        \                /             \
HSK                      \______________/               \___________
                         |<--  one handshake  -->|           |<- next

          ____________________________________________________________
DO-D3     XXXXXXXXXXXXX|<--   data valid    -->|XXXXXXXXXXXXXX
          XXXXXXXXXXXXX________________________XXXXXXXXXXXXXX__________
```

==

HANDSHAKE COMPONENTS - SLOW RECIEVER

```
          ______________                  _____________
HSK                     \                /             \
(composite)              \______________/               \___________
                         |<--  one handshake  -->|           |<- next
                         |                       |
                         |
          ______________ |      ___________________________          _
HSK                     \|     /                            \        /
(transmitter)            \____/                 |            \______/
                         |<- data valid         |            |<- data
                         |                      |                valid
                                                |
          _________________                      ______________
HSK                        \                    /              \
(reciever)                  \__________________/                \_________
                              data received ->|
```

==

HANDSHAKE COMPONENTS - SLOW TRANSMITTER

```
          ______________                  _____________
HSK                     \                /             \
(composite)              \______________/               \___________
                         |<--  one handshake  -->|           |<- next
                         |                       |
          ______________ |                       |_____________
HSK                     \|                      /              \
(transmitter)            \_____________________/                \___________
                         |<- data valid          |              |<- data
                         |                       |                  valid
          _________________               _____________
HSK                        \             /             \
(reciever)                  \___________/               \____________
                          data received ->|
```

BAV Timing Parameters

Item	Min. (usec)	Max. (usec)
BAV low to HSK low	5	20,000
HSK high to BAV high	1	—
End of Command to start of response	10	—
HSK high to HSK low	8	20,000
BAV low(slave) to BAV high(slave)	** see note	**
BAV high to BAV low(slave)	2000**	—

** These two timing parameters occur only when a slave peripheral polls the master (by pulling BAV low) at the same time the master BAV goes low. The slave will then continue to hold BAV low until it receives the first HSK signal from the master. The slave will then release BAV and wait two milliseconds after the master has released BAV to poll again.

BAV Timing -- Polled by Master

```
                    !   command   !   respond   !
                    !   message   !   message   !
       _____________                              ________________
                    \                            /
BAV                  \__________________________/
```

==

BAV Timing -- Polled by Slave - Held by Master

```
                          !   command   !   respond   !
                          !   message   !   message   !
       ____________________                            _________________
                           \                          /
BAV(composite)              \________________________/

       ___________________         _____________________________________
                          \       /
BAV(slave)                 \_____/ (released when HSK goes low)
                                 !
       _____________________     !                    __________________
                            \    !                   /
BAV(master)                  \______________________/
                                 !
                                 !
       _______________________   !                    __________________
                              \  !  XXXXXXXXXXXXXXX/(HSK goes high - low
HSK(master)                    \___XXXXXXXXXXXXXX/   as nibbles of data
                                                     are transmitted)
```

3.2 Signal Rise/Fall Times

The chart below gives the specified AC characteristics of all bus lines. The rise and fall times for this chart are defined for all load conditions over operating temperature ranges from 0 C to 70 C.

Hex Bus AC Specification

Parameter	Max.	Units
Rise Time	3.5*	usec.
Fall Time	2.5*	usec.

* These times are calculated from the initiation of the signal transition (including propagation time through any bus interface devices) and are terminated when the level reaches 0.4 V for fall times and 4.0 V for rise times.

3.3 Nominal Device Values

The equivalent capacitance of a typical device is 200 pF (not including the Ceq of the cable and connector) and should be connected to the bus with a 8.2K ohm pull-up resistor. A device with a higher Ceq should be considered as two (or more) devices and the Rpull-up of this device should be decreased accordingly. This will affect the number of peripherals allowed on the bus. Note that when calculating the value of the pull-up resistance used, the 30pF equivalent capacitance of the cable and connector must be considered.

1. Pullups resisters are 8.2 Kohms +/- 10%.
2. Cin for each device is 20 pF.
3. Ccable = 20pF / ft.
4. Cconnector = 10 pF.
5. 180 pF Capacitors are used on all outputs for RFI and crosstalk problems.

SECTION 4

ELECTRICAL CHARACTERISTICS

Hex Bus DC Specification

Operationg Temperature = OC _ 70 C

Symbol	Conditions	Min.	Max	Units
Vdd (of device on Bus)	Ref. Voltage	4.5	5.5	V
Vil(recognizable)	Any Input	0	0.8	V
Vih(recognizable)	Any Input	4.0	5.5	V
Vol(Isink=8.0mA)	Any Output		0.4	V
Voh	Any Output	*	*	V
Iol(Vol = 0.4 V)	Any Output		8.0	mA
Iin/line	Each Device	-2	+2	uA
Rpullup/line	Each Device	8.2 -10%	8.2 +10%	K ohm
Cin/line (excluding cable capacitance)	Each Device		30	pF
Number of devices on the bus			11	Units

* Use 8.2K ohm pull-ups tied to an allowable Vdd level.

Hex Bus -- Special Electrical Considerations

4.1 Protection From External Power-up

A problem occurs when some peripherals connected to the Hex Bus are powered up when others are turned off. The high signals on the bus can interfere with the operation of the powered down peripherals.

For example, most CMOS device inputs have a protection diode to pass high voltages to the Vdd rail in order to protect the device from static electricity. This diode will pass the high voltages from the bus to the Vdd rail of an off peripheral. This voltage may cause the peripheral to turn on or float in an undefined state.

A recommended solution is to ground the Vdd rail when the device is turned off. Another method is to eliminate the protection diode; the TP0370 Hex Bus Interface chip will accomplish this.

APPLICATION		REVISIONS			
NEXT ASSY	USED ON	LTR	DESCRIPTION	DATE	APPROVED

REV	
SHEET	
REV STATUS OF SHEETS	

ART HUNTER
4/19/83

TEXAS INSTRUMENTS incorporated
Lubbock Texas

INTEGRATED CIRCUITS, INTELLIGENT
PERIPHERAL BUS CONTROLLER (IBC)

size		drawing no
A		1052911

scale	rev	sheet 1 of 28

OUTLINE

OUTLINE (continued)

OUTLINE (continued)

1.0 SCOPE:

THIS SPECIFICATION DETAILS THE REQUIREMENTS FOR AN INTELLIGENT PERIPHERALS BUS CONTROLLER (IBC). THIS IS A BUS CONTROLLER INTEGRATED CIRCUIT.

1.1 APPLICABLE DOCUMENTS

THE FOLLOWING DOCUMENTS FORM A PART OF THIS SPECIFICATION TO THE EXTENT SPECIFIED HEREIN. UNLESS OTHERWISE INDICATED, THE REVISION AND ISSUE IN EFFECT ON THE DATE OF INVITATION FOR BIDS SHALL APPLY. IN THE EVENT OF ANY CONFLICT BETWEEN THIS DOCUMENT AND THE REFERRED DOCUMENTS, THIS DOCUMENT SHALL GOVERN.

1500005 - GENERAL REQUIREMENTS FOR IC's
QRAs 10237 - PRODUCT QUALIFICATION
QRAs 10349 - IC QUALIFICATION

2.0 ABSOLUTE MAXIMUM RATINGS OVER OPERATING FREE AIR TEMPERATURE RANGE. See Note 1:

SUPPLY VOLTAGE, Vdd .... -0.5 to +7 Vdc
INPUT/OUTPUT VOLTAGE, Vin .... -0.5 to Vdd+5.5Vdc
INPUT CURRENT, Iin .... +/- 10 mA
STORAGE TEMPERATURE, Tstg .... -40 C to 125 C

NOTE: 1 STRESSES BEYOND THOSE LISTED UNDER "ABSOLUTE MAXIMUM RATINGS" MAY CAUSE PERMANENT DAMAGE TO THE DEVICE
EXPOSURE TO ABSOLUTE MAXIMUM RATED CONDITIONS FOR EXTENDED PERIODS MAY AFFECT DEVICE RELIABILITY

3.0 RECOMMENDED OPERATING CONDITIONS:

SUPPLY VOLTAGE, Vdd .... 4.5 to 5.5 Vdc
OPERATING TEMPERATURE, Top .... 0 C to 70 C

4.0 ELECTRICAL CHARACTERISTICS

4.1 D.C. CHARACTERISTICS:

Top = 0 C to 70 C, Vdd = 4.5-5.5 Vdc, Vss = 0 Vdc, see Note 2

	MIN	MAX	UNITS
INPUT LOW VOLTAGE (TTL INTERFACE), Vil1		0.8	V
INPUT LOW VOLTAGE (CMOS INTERFACE), Vil2		1.5	V
INPUT HIGH VOLTAGE (TTL INTERFACE), Vih1	2.0		V
INPUT HIGH VOLTAGE (CMOS INTERFACE), Vih2 ...	3.5		V
OUTPUT LOW VOLTAGE (TTL INTERFACE), Iol = 1 mA, Vol1		0.4	V
OUTPUT LOW VOLTAGE (IRQ OUTPUT), Iol = 2 mA, Vol2		0.4	V
OUTPUT LOW VOLTAGE (CMOS INTERFACE), Iol = 8 mA, Vol3		0.4	V
OUTPUT HIGH VOLTAGE (TTL INTERFACE), Ioh = -1 mA, Voh1	2.4		V
OUTPUT HIGH VOLTAGE (CMOS INTERFACE AND IRQ)	OPEN DRAIN		
INPUT LOW CURRENT (ALL INPUTS), Iil	-1	1	uA
INPUT HIGH CURRENT (ALL INPUTS), Iih	-1	1	uA
OUTPUT 3 STATE CURRENT (TTL IHTERFACE), Ioz ..	-10	10	uA
STANDBY CURRENT, Idd		250	uA
INPUT CAPACITANCE, Cin		10	pF

ALL INPUTS ARE SCHMITT TRIGGERS. TTL INPUTS HAVE TRIGGER POINTS AT ABOUT 0.8 V AND 2.0 V. CMOS TRIGGER POINTS ARE ABOUT 1.5 V AND 3.5 V.

4.2 A.C. CHARACTERISTICS

		MAX	UNITS
OUTPUT LOW TRANSITION TIMES (CMOS INTERFACE)	...3000pf load....	2.5	usec

NOTE: 2 TTL INTERFACE - I/O0-I/O3, CS, RS, E, R/W, O0-O2, MS, RES
CMOS INTERFACE - D0-D3, HSK, BAV

5.0 REQUIREMENTS:

5.1 PHYSICAL: 22 - 400, DUAL-IN-LINE PACKAGE, 100 MIL PIN CENTERS. SEE FIGURE 1.

5.1.1 LEADS: SOLDERABLE PER MIL-STD-883, METHOD 2003, WITHOUT AGING.

5.1.2 PIN CONFIGURATION: SEE FIGURE 2.

5.1.3 MARKING: PARTS SHALL BE MARKED WITH THE TI PART NUMBER AND TNE DATE CODE.

FIGURE 1. PACKAGE OUTLINE

```
             _____________
I/O-0----| 1          22|--- O-2
I/O-1----| 2          21|--- O-1
I/O-2----| 3          20|--- O-0
I/O-3----| 4          19|--- Vdd
  BAV ---| 5          18|--- MS
  HSK ---| 6          17|--- D-3
  IRQ ---| 7          16|--- D-2
   E  ---| 8          15|--- D-1
  Vss ---| 9          14|--- D-0
  RES ---|10          13|--- RS
R/W-O3---|11          12|--- CS
             _____________
```

FIGURE 2. PIN CONFIGURATION.

size		drawing no	
scale		rev	sheet 7 of 28

6.0 PIN ASSIGNMENT FUNCTIONS

NAME	PIN	I/O	DESCRIPTION
D-0	14	I/O	Data I/O lines that allow data transfer between IBC and ALC I/O bus.
D-1	15	I/O	
D-2	16	I/O	
D-3	17	I/O	
HSK	6	I/O	HANDSHAKE LINE: set low by source device to indicate to listeners that there is valid data on the ALC I/O bus and held low by the listeners until they accept the data.
BAV	5	I/O	BUS AVAILABLE LINE: set low by source device in the beginning of the message and held low until the end of the message. The new source can originate a new message or "Request Service" only if this line is high.
I/O-0	1	I/O	Data I/O lines that allow data transfer between IBC and the microprocessor. If MS=1 lines act as inputs only.
I/O-1	2	I/O	
I/O-2	3	I/O	
I/O-3	4	I/O	
IRQ	7	O	INTERRUPT OUTPUT (open drain): indicates to to the microprocessor the occurence of the next data nibble on the ALC I/O bus.
CS	12	I	CHIP SELECT INPUT: selects and enables the IBC for microprocessor data transfer.
RS	13	I	REGISTER SELECT INPUT: address line through which the IBC registers can be accessed by the microprocessor. When IBC operates in the latched mode this input works as address strobe.
E	8	I	ENABLE INPUT: if MS=0 there are several options in combination with a pin R/W for controlling data transfer between IBC and the microprocessor. If MS=1, E is used as active low strobe for writing data into IBC.

6.0 PIN ASSIGNMENT FUNCTIONS (continued)

NAME	PIN	I/O	DESCRIPTION
R/W (O-3)	11	I/O	READ-WRITE CONTROL: if MS=0 there are several options in combination with a pin E for controlling direction of data transfer between IBC and the microprocessor. If MS=1, R/W is used as most significant bit of data outputs.
O-0 O-1 O-2	20, 21, 22	I/O I/O I/O	OUTPUT DATA LINES: if MS=1 these lines are used as three least significant bits of data outputs. If MS=0 these lines are used for selecting the options for pins R/W, E and RS.
MS	18	I	MODE SELECT: input which selects the mode of operation.
RES	10	I	RESET INPUT: low level on this input will put IBC into reset state.
Vdd	19	I	Positive supply (3 - 7 Vdc).
Vss	9	I	0 Volt reference.

7.0 INTERNAL ORGANIZATION

7.1 BLOCK DIAGRAM: SEE FIGURE 3.

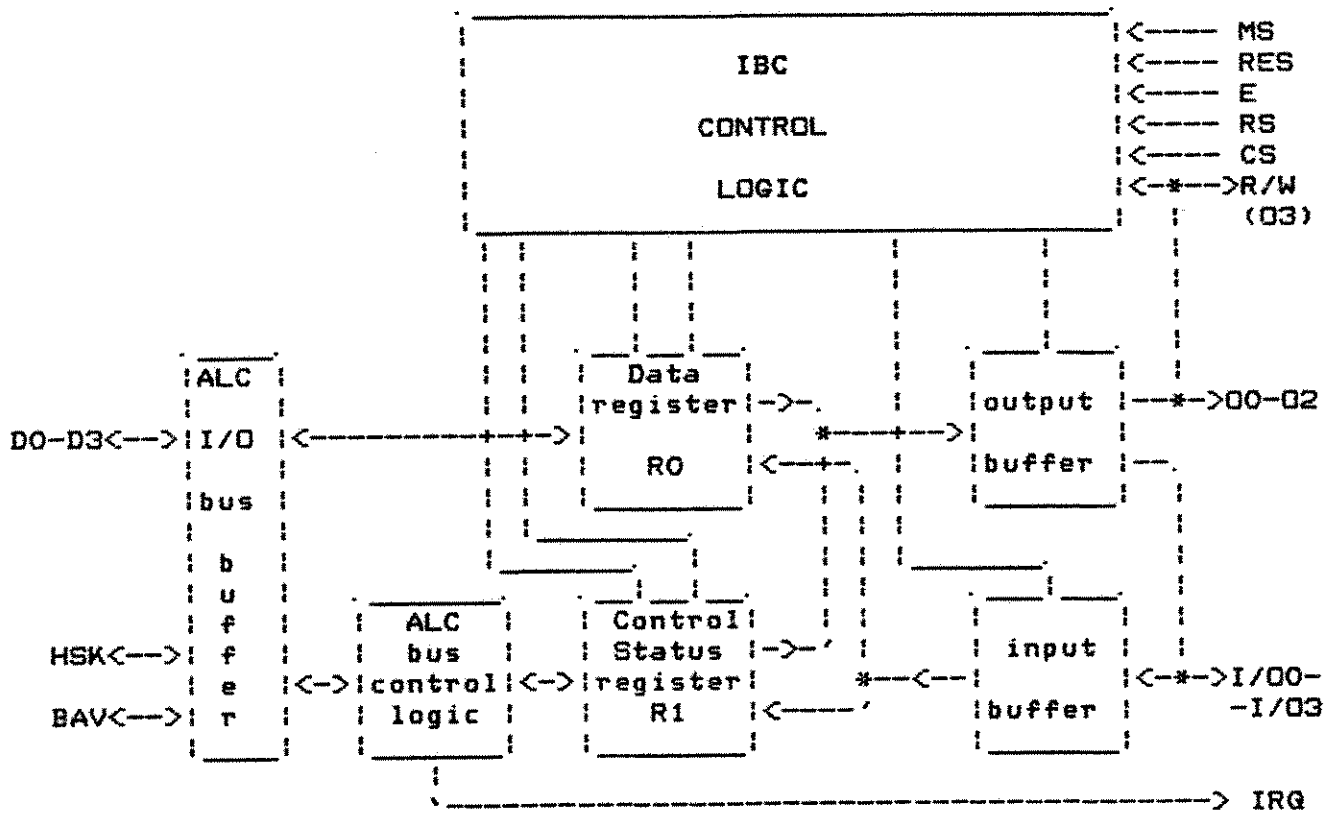


FIGURE 3. BLOCK DIAGRAM.

7.2 REGISTERS

The IBC has four registers which are accessible to the user. All registers are 4-bit wide. Table 1 shows addressing technique used to access each register. The Control Register is used to manipulate HSK and BAV lines. The mode of operation (Inhibit, Disable and Enable) is determined by the Control Register, which is depicted in Figure 4. The Status Register is used to indicate to the processor the status of HSK and BAV lines and the state of Interrupt request and the Inhibit flag. The Status Register is shown in 7.2.3. Transmit and Receive Data Registers are 4-bit buffers into and from the ALC's I/O Bus.

7.2.1 REGISTER DECODING

CS	RS	WRITE	READ
0	0	Transmit Data Register	Receiver Data Register
0	1	Control Register	Status Register

7.2.2 CONTROL REGISTER

```
 ---------------
 | 3 | 2 | 1 | 0 |
 ---------------
   |   |   |   `---- HSK/RESET: 1-Set HSK low, reset IRQ
   |   |   |                    0-Release HSK
   |   |   `---- BAV: 1-Set BAV low, 0-Release BAV when HSK low
   |   `---- INHIBIT: 1-Inhibits HSK latching and IRQ til new BAV
   `---- DISABLE: 1-Disables clearing of Inhibit by new BAV
```

```
                  3   2   1   0
                 ---------------
HARDWARE RESET  | 1 | 1 | 0 | 0 |
                 ---------------
```

7.2.3 Status Register

```
 ---------------
 | 3 | 2 | 1 | 0 |
 ---------------
   |   |   |   `----- HSK LINE  (1=line low)
   |   |   `--------- BAV LINE  (1=line low)
 -------
 | 0 | 0 |----------- ENABLE STATE
 -------
 | 0 | 1 |----------- INHIBIT/DISABLE STATE
 -------
 | 1 | 0 |----------- ACTIVE IRQ (HSK latched low)
 -------
 | 1 | 1 |----------- START OF MESSAGE  (new BAV)
 -------
```

```
                  3   2   1   0
                 ---------------
HARDWARE RESET  | 0 | 1 | X | X |
                 ---------------
```

7.3 INTERRUPT

The IBC has an interrupt which indicates when the data is available on the bus. Interrupt can be disabled completely or until the next message. The interrupt output is open drain with a low active level. The pull-up resistor value is between 3.3 K and 4.7 K Ohms.

7.3.1 INTERRUPT AC CHARACTERISTICS

```
                                                                     Vih
HSK     _______________
                       \
                        \______________________________________     Vil
                        |<---Td--->|                                 Vih
IRQ     ___________________________
                                   \
                                    \__________________________     Vil
```

Td.....IRQ active after HSK fall..................200 ns (max)

8.0 FUNCTIONAL OPERATION DESCRIPTION

8.1 POWER-UP CONSIDERATIONS

If system's power up reset is tied to the RESET pin on IBC, the controller will come up in the Disable state which means that transitions on HSK line will not be latched and no interrupt will be generated. In order to enable the controller, the user has to write "0000" into Control Register. However it might not be desirable to enable the controller in the middle of the message therefore user may perform Enable with "Inhibit Until New Message" by Writing "0100" into Control Register. This will keep the controller in disable state until a new message starts.

8.2 DISABLE MODE

In disable mode the controller will not latch HSK signal and will not produce interrupt, but user will be able to monitor state of BAV and HSK by reading Status Register. In order to put device in disable mode user should Write "1100" into the Control Register.

8.3 INHIBIT (UNTIL NEW MESSAGE)

In inhibit (until new message) mode the controller will be disabled during current message and fully enabled when next message starts. In order to put device into inhibit (until new message) user should Write "0100" into the Control Register.

8.4 LISTEN MODE

In listen mode the user has to read the first two nibbles on the bus, determine if he was selected, and if he was, the user can start communication with the master. If he was not selected the user should perform "Inhibit Until New Message" by Writing "0100" into the Control Register.

8.5 MONITOR MODE

The capability exists for user to monitor the communication even if he was not selected, however, this is not recommended since it may slow down the bus operation. This capability allows monitoring and recording of all bus communications.

8.6 TRANSMIT MODE

In transmit mode, the user has to send the data utilizing the Write Sequence, monitor the HSK line and when the HSK line goes into an inactive state the user may transmit the next data.

8.7 DATA TRANSFER OPERATION

All receive and transmit data can be handled by executing the Read and Write Sequences respectively.

8.7.1 READ SEQUENCE

The Read Sequence allows the user to obtain the data transmitted on the bus and prepares IBC to receive the next transmission. The Read Sequence has to be used whenever the user is in the listen mode and receives an interrupt from IBC.

a - Reset Interrupt - Write "0001" into Control Register
b - Obtain Data - Read Receiver Data Register
c - Reset HSK Latch - Write "0000" into Control Register

8.7.2 WRITE SEQUENCE

The Write Sequence allows the user to place the data on the bus and to signal the other devices that the data is available. The Write Sequence has to be used whenever the user is in transmit mode and ready to send the data.

a - Prepare Data - Write Data into Transmit Data Register
b - Set HSK Signal - Write "0001" into Control Register
c - Reset HSK Signal - Write "0000" into Control Register

8.8 REQUEST SERVICE SEQUENCE

The Request Service Sequence allows the user to signal the master by putting the BAV line into the low state until the first HSK transition. The Request Service Sequence may be used whenever the user has permission from the master to request the service.

a - Set BAV signal - Write "0010" into Control Register
b - Enable BAV Reset - Write "0000" into Control Register

8.9 MASTER MODE DIFFERENCES

The difference between master and slave modes of operation exists because in master mode the user has to hold the BAV line low during the entire message. Therefore the Read and Write Sequences shown above must be modified for master mode operation. The modification requires that all accesses to the Control Register must maintain the BAV bit in the high state.

9.0 MODES OF OPERATION

The IBC can operate in 7 different modes which allows simple interface with several popular microprocessor families. The desired mode can be selected by setting the Mode Select input (MS). If input MS is set low three modes can be selected by setting pins O-0 and O-1. Table 1 shows the modes selected by different bit patterns on these pins.

In addition, when MS is set low, a high on pin O-2 will enable the latching capability for CS and RS signals. This feature is useful for multiplex address/data bus similar to TMS-7000 or INTEL-8085. In this mode ALATCH is connected to the RS pin and on the falling edge of a pulse, RS will latch the level on I/O-0 pin as internal RS and the level on CS pin as internal CS which should be high to select the chip.

If the MS is set high the IBC is in Mode 7 and reconfigured as shown in Section 9.7. This mode is useful with 4-bit MPUs which don't have a bidirectional bus. Please note that pin R/W is used as MSB of the output data.

9.0 MODES OF OPERATION (continued)

Table 1. Modes of Operation.

O-2	O-1	O-0	MODE DESCRIPTION
0	0	0	Mode 0 - ENABLE active low (see Figure 3) TMS-7000 compatible
0	0	1	Mode 1 - ENABLE active high (see Figure 4) 6500 compatible
0	1	0	Mode 2 - accepts RD- and WR- signals (see Figure 5) 8085 and 8048 compatible
1	X	X	Modes 4,5,6 - control signals for R/W and E correspond to non-multiplex modes 0,1,2 but addressing arranged for multiplex busses.

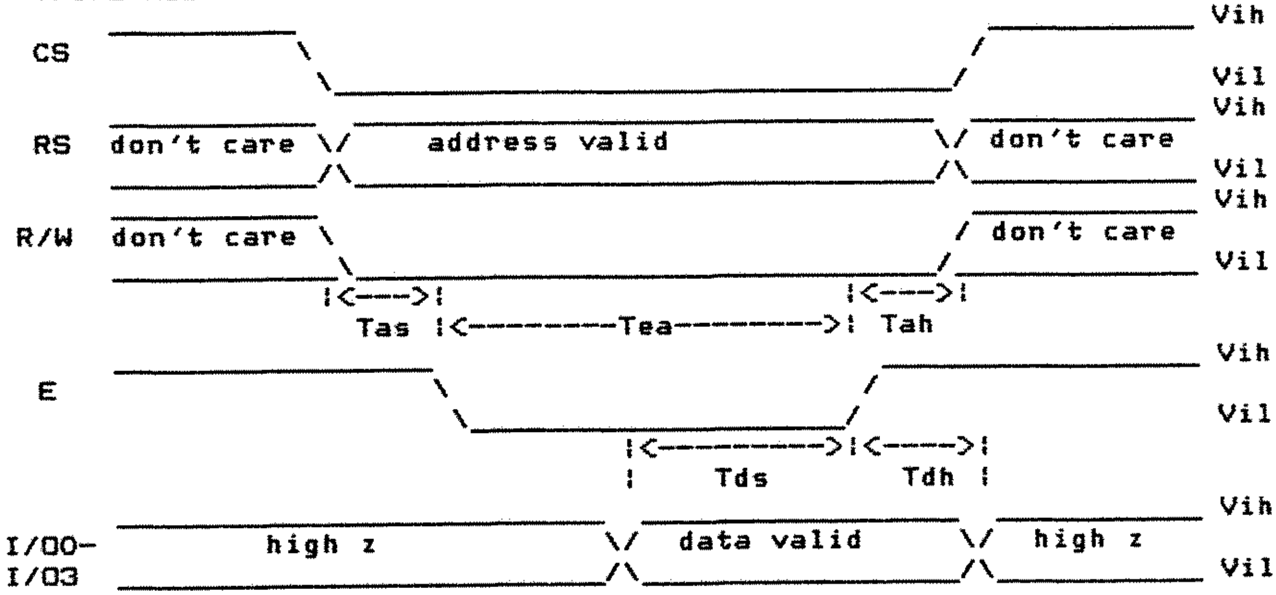
9.1 MODE 0 SPECIFICATIONS

9.1.1 MODE 0 INTERFACE CONFIGURATION

```
          TMS-7000                                      IBC
     ----------------                           ----------------
     |              |                           |              |
     |              | /---------------\         |              |
     |  PC-0 - PC-3|<  4-bit data bus >|I/O-0 - I/O-3 |
     |              | \---------------/         |              |
     |              |                           |              |
     |         R/W-|------------------->|R/W                  |
     |              |                           |              |
     |           E-|------------------->|E          O-2|---+
     |              |                           |   O-1|---|
     |         INT1|<-------------------|IRQ       O-0|---|
     |              |                           |    MS|---|
     |              |     ---------             |      |   |
     |              |---\ |address|---->|RS            |  _|_
     |  PD-0 - PD-7|    >|       |              |      |  \\\\
     |              |---/ |decoder|---->|CS            |
     |              |     ---------             |      |
     ----------------                           ----------------
```

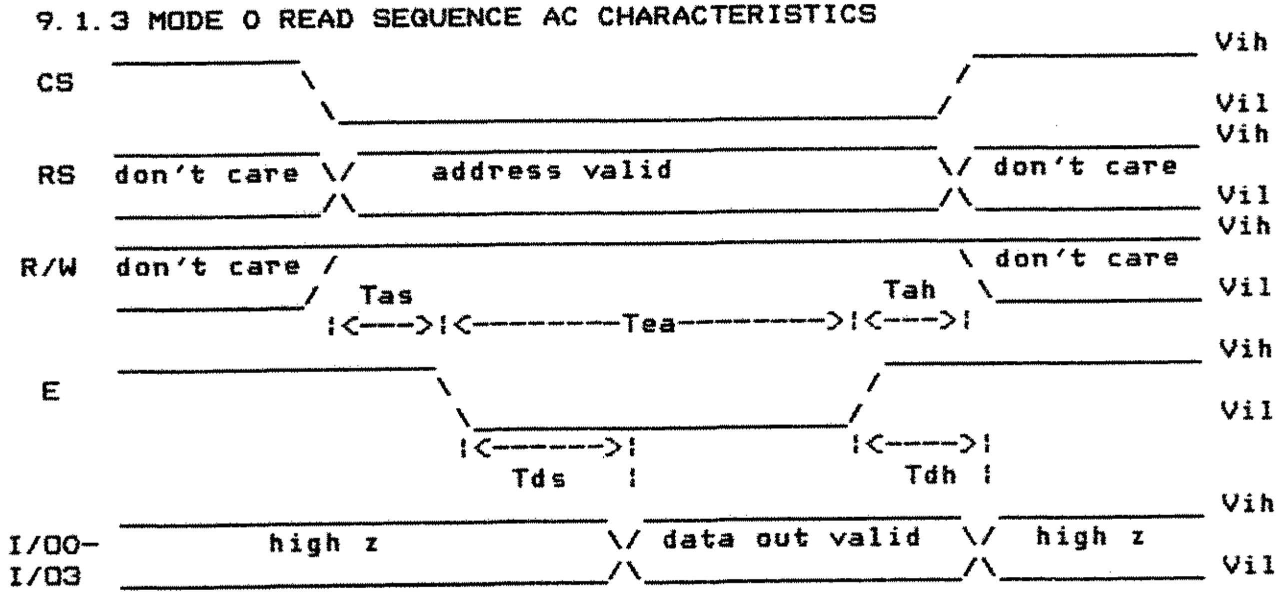
size		drawing no
scale	rev	

9.1.2 MODE 0 WRITE SEQUENCE AC CHARACTERISTICS



Tas....Address and R/W valid before E fall.......... 50 ns (min)
Tah....Address and R/W hold valid after E rise...... 30 ns (min)
Tds....Data in valid before E rise..................150 ns (min)
Tdh....Data in hold valid after E rise............. 65 ns (min)
Tea....Enable active................................300 ns (min)

9.1.3 MODE 0 READ SEQUENCE AC CHARACTERISTICS



Tas....Address and R/W valid before E fall.......... 50 ns (min)
Tah....Address and R/W hold valid after E rise...... 30 ns (min)
Tds....Data out valid after E fall..................120 ns (max)
Tdh....Data out hold valid after E rise............ 60 ns (max)
Tea....Enable active................................300 ns (min)

size		drawing no	
scale		rev	sheet 16 of 28

9.2 MODE 1 SPECIFICATIONS

9.2.1 MODE 1 INTERFACE CONFIGURATION

```
                6500                                         IBC
         ----------------                            ----------------
         |              |                            |              |
         |              | /----------------\         |              |
         |   D-0 - D-3  |<  4-bit data bus  >        |I/O-0 - I/O-3 |
         |              | \----------------/         |              |
         |              |                            |              |
         |         R/W- |------------------>         |R/W           |
         |              |                            |              |
         |        PHI-2 |------------------>         |E        O-0  |-----> +V
         |              |                            |              |
         |         IRQ- |<------------------         |IRQ      O-1  |---+
         |              |                            |         O-2  |---|
         |              |      ---------             |          MS  |---|
         |              |---\  |address|---->        |RS            |   |
         |  A-O - A-15  |    >|       |              |              | __|__
         |              |---/  |decoder|---->        |CS            |  \\\\
         |              |      ---------             |              |
         ----------------                            ----------------
```

9.2.2 MODE 1 WRITE SEQUENCE AC CHARACTERISTICS

```
        _____________                                  ______________ Vih
 CS                  \                                /
                      \______________________________/                Vil
        _____________  ______________________________  ______________ Vih
 RS     don't care \/      address valid            \/ don't care
        ____________/\______________________________/\______________ Vil
        _____________                                  ______________ Vih
 R/W    don't care \                                  / don't care
        ____________\______________________________/________________ Vil
                   |<--->|                        |<--->|
                    Tas  |<---------Tea---------->| Tah
                                                                      Vih
 E                        /________________________\
        _________________/                          \_______________ Vil
                                  |<---------->|<---->|
                                  |    Tds      Tdh   |
        ___________________________  ______________  ______________ Vih
 I/O0-          high z            \/  data valid  \/  high z
 I/O3   __________________________/\______________/\______________ Vil
```

Tas....Address and R/W valid before E rise.......... 50 ns (min)
Tah....Address and R/W hold valid after E fall...... 30 ns (min)
Tds....Data in valid before E fall.................100 ns (min)
Tdh....Data in hold valid after E fall............. 60 ns (min)
Tea....Enable active...............................140 ns (min)

9.2.3 MODE 1 READ SEQUENCE AC CHARACTERISTICS

```
                                                             ____________ Vih
 CS    ___________                                          /
                  \                                        /
                   \______________________________________/               Vil
       ___________   _____________________________________  ____________  Vih
 RS    don't care \/      address valid                   \/ don't care
       ___________/\_____________________________________/\____________  Vil
       __________________________________________________________________ Vih
R/W    don't care /                                       \  don't care
       __________/                                         \____________  Vil
                   |<--->|                           |<--->|
                     Tas |<---------Tea------------->| Tah
                                                                          Vih
                           _________________________
 E                        /                         \
       __________________/                           \__________________  Vil
                         |<------->|                 |<----->|
                              Tds  |                     Tdh |
       ____________________________  ________________  ______________    Vih
I/O0-           high z             \/ data out valid \/   high z
I/O3   ____________________________/\________________/\______________    Vil
```

Tas....Address and R/W valid before E rise.......... 50 ns (min)
Tah....Address and R/W hold valid after E fall...... 30 ns (min)
Tds....Data out valid after E rise.................120 ns (max)
Tdh....Data out hold valid after E fall............ 60 ns (max)
Tea....Enable active...............................140 ns (min)

9.3 MODE 2 SPECIFICATIONS

9.3.1 MODE 2 INTERFACE CONFIGURATION

```
            8085                                     IBC
      ________________                      ________________
     |                |                    |                |
     |                | /----------------\ |                |
     |  AD-0 - AD-3   |<  4-bit data bus  >|I/O-0 - I/O-3   |
     |                | \----------------/ |                |
     |                |                    |                |
     |           WR-  |------------------->|R/W             |
     |                |                    |                |
     |           RD-  |------------------->|E          O-1  |-----> +V
     |                |                    |                |
     |    RST 5.5     |<-----     ---------|IRG        O-2  |---+
     |                |                    |           O-0  |---|
     |                |      __________    |            MS  |---|
     |                |---\ |address |---->|RS              |   |
     |   A-7 - A-15   |    >|        |     |                | __|__
     |                |---/ |decoder |---->|CS              |  \\\\
     |                |      __________    |                |
      ________________                      ________________
```

size		drawing no	
scale		rev	sheet 18 of 28

9.3.2 MODE 2 WRITE SEQUENCE AC CHARACTERISTICS

```
                                                                   ____________ Vih
  CS    _____________                                              /
                     \                                            /
                      \__________________________________________/              Vil
        ______________    ______________________________________    ___________ Vih
  RS    don't care \/     address valid                          \/ don't care
        ___________/\_________________________________________ /\___________    Vil
                    |<--->|                                    |<--->|
                     Tas  |<---------Tsv----------------------->| Tah
                                                                                Vih
        _______________________                                 _______________
 R/W                           \                               /
                                \_____________________________/                 Vil
        ________________________________________________________|______________ Vih
   E                                                            |
                                                Tds             |   Tdh          Vil
                                          |<------------------->|<----->|
        _____________________________________    __________________    ________ Vih
I/O0-              high z                     \/     data valid    \/  high z
I/O3    _____________________________________/\___________________/\__________ Vil
```

Tas....Address valid before R/W fall................ 50 ns (min)
Tah....Address hold valid after R/W rise............ 50 ns (min)
Tds....Data in valid before R/W rise................150 ns (min)
Tdh....Data in hold valid after R/W rise............ 60 ns (min)
Tsv....Write pulse width on R/W pin.................140 ns (min)

9.3.3 MODE 2 READ SEQUENCE AC CHARACTERISTICS

```
        ______________                                             ___________ Vih
  CS                  \                                            /
                      |\__________________________________________/|            Vil
        ______________    ______________________________________    ___________ Vih
  RS    don't care \/     address valid                          \/ don't care
        ___________/\_________________________________________ /\___________    Vil
        ____________|_________________________________________ |___________     Vih
 R/W                |                                          |
                    | Tas                                  Tah |                Vil
                    |<--->|<---------Tea------------------>|<--->|
        _________________________                          ____________________ Vih
   E                             \                        /
                                  \______________________/                      Vil
                                  |<------->|              |<----->|
                                      Tds   |                  Tdh |
        _______________________________    _________________________    _______ Vih
I/O0-              high z               \/  data out valid       \/  high z
I/O3    _______________________________/\_______________________/\____________ Vil
```

Tas....Address valid before E fall.................. 50 ns (min)
Tah....Address hold valid after E rise.............. 50 ns (min)
Tds....Data out valid after E fall..................120 ns (max)
Tdh....Data out hold valid after E rise............. 60 ns (max)
Tea....Read pulse width on E pin....................140 ns (min)

9.4 MODE 4 SPECIFICATIONS

9.4.1 MODE 4 INTERFACE CONFIGURATION

```
          TMS-7000                                      IBC
     -----------------                           ----------------
     |               | /-----------------\ |                |
     |   PC-0 - PC-3 |<  4-bit data bus   >|I/O-0 - I/O-3   |
     |               | \-----------------/ |                |
     |           R/W-|-------------------->|R/W         O-2 |----->+V
     |             E-|-------------------->|E               |
     |               |                     |            O-1 |---+
     |          INT1 |<--------------------|IRQ         O-0 |---|
     |               |                     |             MS |---|
     |        ALATCH |-------------------->|RS              |  _|_
     |           C-7 |-------------------->|CS              |  \\\\
     -----------------                      ----------------
```

9.4.2 MODE 4 WRITE SEQUENCE AC CHARACTERISTICS

```
         ______________________________________________________________ Vih
  CS     don't care/          \      don't care
         __________/Tcs | Tch \____________________________________ Vil
                   |<-->|<-->|
                                                                     Vih
                ________
  RS           /        \
 (ALATCH)_____/   Taa    \_________________________________________ Vil
              |<----->|<-->|
                 |<-->| Tah|
         ____________Tas| Tev|                     ______________ Vih
  R/W    don't care \        |                    /don't care
         ____________\_______|___________________/______________ Vil
                  |   |<---->|                 |<--->|
                  |     Trs |<--------Tea-------->| Trhe
         _________|_________                        _____________ Vih
  E               |         \                      /
                  |        | \_______Tds________/   Tdh          Vil
                  |        |     |<---------->|<---->|
         _______  ________  __________  _______________  ________ Vih
  I/O0-  high z \/ address\/ high z  \/   data valid   \/ high z
  I/O3   _______/\________/\__________/\_______________/\________ Vil
```

Taa....Alatch pulse width on RS pin.................100 ns (min)
Tas....Address valid before RS(ALATCH) fall......... 30 ns (min)
Tah....Address hold valid after RS(ALATCH) fall..... 30 ns (min)
Tcs....Chip Select valid before RS(ALATCH) fall..... 30 ns (min)
Tch....Chip Select hold valid after RS(ALATCH) fall. 30 ns (min)
Tev....E valid after RS(ALATCH) fall................ 30 ns (min)
Trs....R/W valid before E fall...................... 30 ns (min)
Trh....R/W hold valid after E rise.................. 30 ns (min)
Tds....Data in valid before E rise..................150 ns (min)
Tdh....Data in hold valid after E rise.............. 65 ns (min)
Tea....Enable active................................250 ns (min)

size		drawing no	
scale		rev	sheet 20 of 28

9.4.3 MODE 4 READ SEQUENCE AC CHARACTERISTICS

```
                                                                          Vih
   CS   don't care/        \      don't care
        __________/    |    \____________________________________________ Vil
                  |<-->|<-->|
                   Tcs | Tch
                                                                          Vih
                   ______
   RS             /      \
 (ALATCH)______/   Taa    \_______________________________________________ Vil
                |<---->|<-->|
                  |<-->| Tah|
                  | Tas| Tev|
        ____________________|__________________________________________ Vih
   R/W   don't care /       |                              \don't care
        ___________/    Trs |                          Trhe \__________ Vil
                  |  |<---->|<---------Tea--------->|<--->|
        __________|_________                          _____________ Vih
   E              |         \                        /
                  |         | \_____________________/                 Vil
                  |         |   Tds                 |   Tdh
                  |         |<------->|              |<---->|
        _________   ________   ________   _______________   _________ Vih
  I/O0-  high z \/ address\/ high z \/  data out valid \/  high z
  I/O3  _________/\________/\________/\_______________/\_________ Vil
```

Taa....Alatch pulse width on RS pin................100 ns (min)
Tas....Address valid before RS(ALATCH) fall......... 30 ns (min)
Tah....Address hold valid after RS(ALATCH) fall..... 30 ns (min)
Tcs....Chip Select valid before RS(ALATCH) fall..... 30 ns (min)
Tch....Chip Select hold valid after RS(ALATCH) fall. 30 ns (min)
Tev....E valid after RS(ALATCH) fall................ 30 ns (min)
Trs....R/W valid before E fall...................... 30 ns (min)
Trh....R/W hold valid after E rise.................. 30 ns (min)
Tds....Data out valid after E fall..................120 ns (max)
Tdh....Data out hold valid after E rise............. 60 ns (max)
Tea....Enable active................................250 ns (min)

9.5 MODE 5 SPECIFICATIONS

MODE 5 SPECIFICATIONS ARE THE SAME AS IN MODE 4 EXCEPT PULSE ON "ENABLE" LINE IS INVERTED.

9.6 MODE 6 SPECIFICATIONS

9.6.1 MODE 6 INTERFACE CINFIGURATION

```
              8085                                       IBC
         ________________                        ________________
        |                | /-----------------\ |                |
        |  AD-0 - AD-3 |<  4-bit data bus  >|I/O-0 - I/O-3   |
        |                | \-----------------/ |                |
        |           WR-|--------------------->|R/W             |
        |           RD-|--------------------->|E          O-2|--+--> +V
        |                |                     |            O-1|--'
        |      RST 5_5|<-----     ----------|IRQ              |
        |                |                     |            O-0|---+
        |           ALE|--------------------->|RS           MS|---|
        |          AD-7|--------------------->|CS              |  __|__
         ________________                        ________________   \\\\
```

9.6.2 MODE 6 WRITE SEQUENCE AC CHARACTERISTICS

```
        _____________________________________________________________ Vih
 CS     don't care/          \        don't care
        __________/           \______________________________________ Vil
                   |<-->|<-->|
                    Tcs | Tch
                                                                      Vih
                  _______
 RS              /        \
 (ALE)_______/   Taa      \_______________________________________ Vil
                |<----->|<-->|
                   |<-->| Tah|
                   | Tas| Twv|
        _________________                    ____________________ Vih
 R/W           |         \                  /
               |        | \________________/                       Vil
               |        |
               |        |<--------Twa-------->|
        _______|________|____________________|____________________ Vih
 E             |        |              Tds    |  Tdh                  Vil
               |        |          |<------->|<--->|
        ________  ________  ________  _______________  __________ Vih
 I/O0-  high z \/ address\/ high z \/  data valid     \/  high z
 I/O3   _______/\________/\________/\_______________/\__________ Vil
```

Taa....ALE pulse width on RS pin.................... 80 ns (min)
Tas....Address valid before RS(ALE) fall............ 50 ns (min)
Tah....Address hold valid after RS(ALE) fall........ 50 ns (min)
Tcs....Chip Select valid before RS(ALE) fall........ 50 ns (min)
Tch....Chip Select hold valid after RS(ALE) fall.... 50 ns (min)
Twv....R/W valid after RS(ALE) fall................. 50 ns (min)
Tds....Data in valid before R/W rise................150 ns (max)
Tdh....Data in hold valid after R/W rise............ 60 ns (max)
Twa....Write pulse width on R/W pin.................200 ns (min)

size		drawing no	
scale		rev	

9.6.3 MODE 6 READ SEQUENCE AC CHARACTERISTICS

```
         _________________________________________________________________ Vih
    CS   don't care/          \      don't care
         _________/            \__________________________________________ Vil
                  |<-->|<-->|
                   Tcs | Tch
                                                                           Vih
                     _____
   RS               /     \
 (ALATCH)_____/   Taa   \_______________________________________________ Vil
                |<----->|<-->|
                  |<-->| Tah|
                  | Tas| Tev|
         _________|_______|_______________________________________________ Vih
   R/W            |        |
                  |        |                                               Vil
                  |        |<---------Tea---------->|
         _________|_________                         _____________________ Vih
    E             |         \                       /
                  |        | \_____________________/    Tdh                Vil
                  |        |<-------->|              |<----->|
                  |            Tds    |                Tdh   |
         _______   ________   ________   _________________   _____________ Vih
  I/O0-   high z \/ address\/ high z \/  data out valid \/  high z
  I/O3   ________/\________/\________/\_________________/\_____________ Vil
```

Taa....ALE pulse width on RS pin....................150 ns (min)
Tas....Address valid before RS(ALE) fall............ 50 ns (min)
Tah....Address hold valid after RS(ALE) fall........ 50 ns (min)
Tcs....Chip Select valid before RS(ALE) fall........ 50 ns (min)
Tch....Chip Select hold valid after RS(ALE) fall.... 50 ns (min)
Tev....E valid after RS(ALE) fall................... 50 ns (min)
Tds....Data out valid after E fall..................120 ns (max)
Tdh....Data out hold valid after E rise............. 60 ns (max)
Tea....Read pulse width on E pin....................200 ns (min)

size		drawing no	
scale		rev	sheet 23 of 28

9.7 MODE 7 SPECIFICATIONS

9.7.1 MODE 7 INTERFACE CONFIGURATION

```
            TMS-1000                                IBC
      -----------------                    -----------------
      |               |-----------------\  |               |
      |   O-0 - O-3   |  4-bit data out >  |I/O-0 - I/O-3  |
      |               |-----------------/  |               |
      |               |                    |               |
      |      K-1,K-2  | /------------------|               |
      |               |<  4-bit data in    |O-0 - O-2,R/W  |
      |      K-4,K-8  | \------------------|               |
      |               |                    |               |
      |           R-0 |------------------->|E              |
      |           R-1 |------------------->|RS           MS|---> +V
      |           R-2 |------------------->|CS             |
      -----------------                    -----------------
```

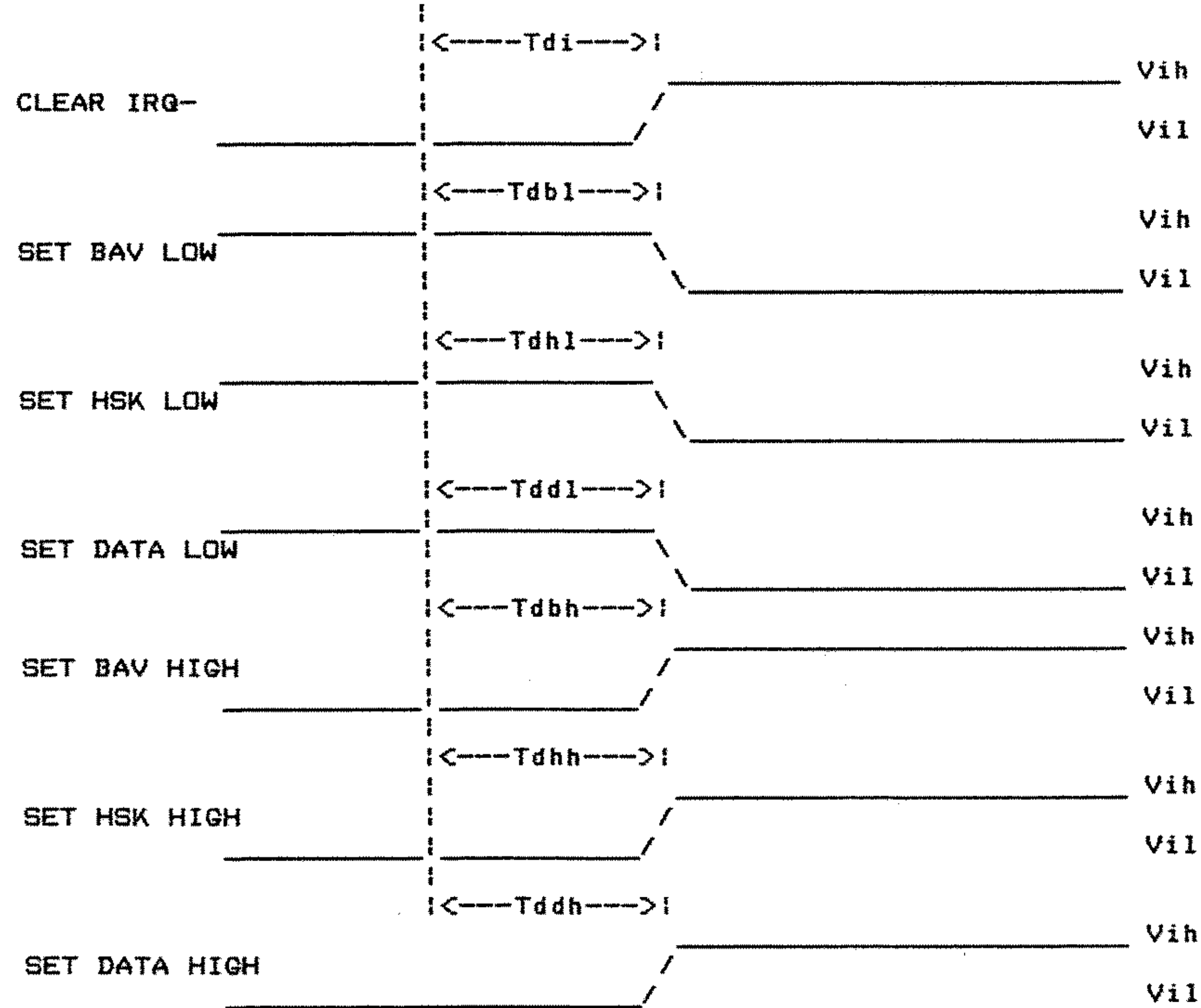
9.7.2 MODE 7 AC CHARACTERISTICS

```
                                                  ____________ Vih
 CS      ____________                            /
                     \                          /
                      \________________________/               Vil
                     |                          |
        _____________  ________________________  ____________ Vih
 RS      don't care  \/     address valid      \/ don't care
        _____________/\________________________/\____________ Vil
                     |                          |
                     |<--->|                |<--->|
                     | Tas |<-----Tea------>| Tah |
        _____________|______                ______|__________ Vih
 E                   |      \              /      |
                     |       \____________/                   Vil
                     |            |<------>|<--->||
                     |            |  Tds     Tdh  ||
        _____________|____________  ___________  ____________ Vih
 I/O0-               |high z      \/ data valid\/      high z
 I/O3   _____________|____________/\___________/\____________ Vil
                     |    Tdo                   | Tdoz
                     |<----------->|            |<---->|
        _________________________  _________________  _______ Vih
 O0-O2,          high z          \/  data out valid \/ high z
 R/W    _________________________/\_________________/\_______ Vil
```

Tas....Address valid before E fall................ 50 ns (min)
Tah....Address hold valid after E rise............. 50 ns (min)
Tds....Data in valid before E rise.................120 ns (max)
Tdh....Data in hold valid after E rise............. 60 ns (max)
Tdo....Data out valid after CS fall................ 60 ns (max)
Tdo....Data out valid after CS rise................ 60 ns (max)
Tea....Write pulse width on E pin..................200 ns (min)

10.0 PERIPHERAL BUS INTERFACE AC CHARACTERISTICS

Write Operation Complete




```
Tdi....IRQ  high after Write cycle completion.......200 ns (max)
Tdb....BAV low after Write cycle completion.........200 ns (max)
Tdh....HSK low after Write cycle completion.........200 ns (max)
Tdd....Data pin low after Write cycle completion....200 ns (max)
Tdbh...BAV high after Write cycle completion........200 ns (max) + Tbd
Tdhh...HSK high after Write cycle completion........200 ns (max) + Tbd
Tddh...Data pin high after Write cycle completion...200 ns (max) + Tbd
```

Tbd - Bus delay timing depends on Pull-up value and number of peripherals on the bus.

Write cycle completion means end of the Write pulse on E or R/W pins corresponding to the mode selected and assuming that the right command is executed.

size		drawing no	
scale		rev	sheet 25 of 28

11.0 SPECIAL FUNCTIONS SPECIFICATIONS

11.1 DATA REGISTER SELF-CLEAR

11.1.1 PROCEDURE FOR TESTING DATA REGISTER SELF-CLEAR

1 Set Data Lines D0-D3 to "0000" by writing into Data Register
2 Set HSK low by writing "0001" into Control Register
3 Hold HSK low externally
4 Release internal HSK by writing "0000" into Control Register
5 Release HSK externally and observed time delay for Data Lines to rise

11.1.2 DATA REGISTER SELF-CLEAR AC CHARACTERISTICS

```
                         __________________________________________ Vih
 HSK                    /
      _________________/                                            Vil
                       |<----Tdsf---->|
                                        ___________________________ Vih
 D0-D3                                 /
      ________________________________/                             Vil
```

Tdsf...Data lines high after HSK rise..............200 ns (max) + Tbd

Tbd - Bus delay timing depends on Pull-up value and number of peripherals on the bus.

11.2 BAV HOLD LATCH SELF-CLEAR

11.2.1 PROCEDURE FOR TESTING BAV HOLD LATCH SELF-CLEAR

1 Set BAV low by writing "0010" into Control Register
2 Release internal BAV hold by writing "0000" into Control Register
3 Set HSK low externally and observed time delay for BAV to rise

11.2.2 BAV HOLD SELF-CLEAR AC CHARACTERISTICS

```
      _______________                                               Vih
 HSK                 \
                      \____________________________________________ Vil
                     |<----Tbsf---->|
                                      _____________________________ Vih
 BAV                                 /
      ______________________________/                               Vil
```

Tbsf...Internal BAV release after HSK fall..........200 ns (max) + Tbd

Tbd - Bus delay timing depends on Pull-up value and number of peripherals on the bus.

11.2.3 BAV SETUP TIME

```
                                                                     Vih
 HSK     ______________
                       \
                        \__________________________________________  Vil
         |<-Tbsu-->|
                                                                     Vih
 BAV     ___
            \
             \_____________________________________________________  Vil
```

Tbsu...BAV setup time before HSK fall.................5 u sec (min)

size		drawing no	
scale		rev	sheet 27 of 28

12.0 QUALITY ASSURANCE PROVISIONS

12.1 RESPONSIBILITY FOR INSPECTION

UNLESS OTHERWISE SPECIFIED IN THE CONTRACT OR PURCHASE ORDER, THE SUPPLIER SHALL BE RESPONSIBLE FOR PERFORMING INSPECTIONS THAT ARE SUFFICIENT TO ASSURE THAT THE PARTS SUPPLIED MEET THE REQUIREMENTS SPECIFIED HEREIN.

12.2 LOT ACCEPTANCE

LOTS FURNISHED TO THIS SPECIFICATION SHALL BE CAPABLE OF PASSING A SAMPLING INSPECTION FOR DEFECTS TO AN ACCEPTABLE QUALITY LEVEL (AQL) OF ONE PERCENT FOR NORMAL SINGLE SAMPLING, LEVEL II, PER MIL-STD-105. FAILING LOTS SHALL BE SUBJECT TO REJECTION.

12.3 LIFE FAILURE RATE

THE MEAN LIFE FAILURE RATE FOR DEVICES SHALL BE EQUAL TO OR LESS THAN .018%/1000 HOURS AT 55 deg C DERATED AT 0.5 EV.

12.3.1 ALL PARTS SHALL BE FUNCTIONALLY TESTED AFTER BEING BURNED IN FOR 24 HOURS AT 125 deg C AT NOMINAL DC VOLTAGES AND FOR A MINIMUM OF EIGHT HOURS AT 125 deg C AT RECOMMENDED HIGH STRESS VOLTAGES.

13.0 PREPARATION FOR DELIVERY

13.1 PACKAGING

PACKING AND WRAPPING SHALL BE SUFFICIENT TO PROTECT AGAINST DAMAGE OR LOSS DURING SHIPMENT FROM THE SUPPLIER TO THE DESTINATION SPECIFIED IN THE PURCHASE ORDER

13.2 MARKING

THE SHIPPING CONTAINER SHALL BE MARKED WITH TI PART NUMBER (SEE PART NUMBER BLOCK) AND THE QUANTITY CONTAINED. ADDITIONAL MARKINGS ARE PERMITTED.

APPLICATION		REVISIONS			
NEXT ASSY	USED ON	LTR	DESCRIPTION	DATE	APPROVED

REV																															
SHEET																															
REV STATUS OF SHEETS																															

TEXAS INSTRUMENTS incorporated
Lubbock Texas

ART HUNTER
8/04/83

NOTE NEW PINOUT

INTEGRATED CIRCUITS, INTELLIGENT PERIPHERAL BUS CONTROLLER (IBC) TP0370

size		drawing no
A		PRELIMINARY
scale	rev	sheet 1 of 29

OUTLINE

OUTLINE (continued)

OUTLINE (continued)

1.0 SCOPE:

THIS SPECIFICATION DETAILS THE REQUIREMENTS FOR AN INTELLIGENT PERIPHERALS BUS CONTROLLER (IBC). THIS IS A BUS CONTROLLER INTEGRATED CIRCUIT.

1.1 APPLICABLE DOCUMENTS

THE FOLLOWING DOCUMENTS FORM A PART OF THIS SPECIFICATION TO THE EXTENT SPECIFIED HEREIN. UNLESS OTHERWISE INDICATED, THE REVISION AND ISSUE IN EFFECT ON THE DATE OF INVITATION FOR BIDS SHALL APPLY. IN THE EVENT OF ANY CONFLICT BETWEEN THIS DOCUMENT AND THE REFERRED DOCUMENTS, THIS DOCUMENT SHALL GOVERN.

1500005 - GENERAL REQUIREMENTS FOR IC's
QRAs 10237 - PRODUCT QUALIFICATION
QRAs 10349 - IC QUALIFICATION

2.0 ABSOLUTE MAXIMUM RATINGS OVER OPERATING FREE AIR TEMPERATURE RANGE. See Note 1:

SUPPLY VOLTAGE, Vcc .... -0.5 to +7 Vdc
INPUT/OUTPUT VOLTAGE, Vin .... -0.5 to +7 Vdc *
INPUT CURRENT, Iin .... +/- 10 mA
STORAGE TEMPERATURE, Tstg .... -40 C to 125 C

* I/O pins which are designated "CMOS INTERFACE" (see note 2 on next page) are designed to allow overvoltage when enabled with a mask option. When this option is used I/O pins may exceed Vcc voltage by as much as 5.5 Vdc as long as the voltage with respect to ground does not exceed 7 Vdc.

NOTE: 1 STRESSES BEYOND THOSE LISTED UNDER "ABSOLUTE MAXIMUM RATINGS" MAY CAUSE PERMANENT DAMAGE TO THE DEVICE
EXPOSURE TO ABSOLUTE MAXIMUM RATED CONDITIONS FOR EXTENDED PERIODS MAY AFFECT DEVICE RELIABILITY

3.0 RECOMMENDED OPERATING CONDITIONS:

SUPPLY VOLTAGE, Vcc .... 4.5 to 5.5 Vdc
OPERATING TEMPERATURE, Top .... 0 C to 70 C

4.0 ELECTRICAL CHARACTERISTICS

Top = 0 C to 70 C, Vcc = 4.5-5.5 Vdc
All voltages referenced to ground.

NOTE: 2 TTL INTERFACE - I/00-I/03, CS, RS, E, R/W, 00-02, MS, RES
CMOS INTERFACE - D0-D3, HSK, BAV

4.1 D.C. CHARACTERISTICS:

ALL INPUTS ARE SCHMITT TRIGGERS. TTL INPUTS HAVE TRIGGER POINTS AT ABOUT 1.3 V AND 1.7 V. CMOS TRIGGER POINTS ARE ABOUT 1.7 V AND 3.6 V.

	MIN	MAX	UNITS
INPUT LOW VOLTAGE (TTL INTERFACE), Vil1	0.9	1.6	V
INPUT LOW VOLTAGE (CMOS INTERFACE), Vil2	1.4	2.1	V
INPUT HIGH VOLTAGE (TTL INTERFACE), Vih1	1.4	2.3	V
INPUT HIGH VOLTAGE (CMOS INTERFACE), Vih2 ...	3.0	4.2	V
OUTPUT LOW VOLTAGE (TTL INTERFACE), Iol = 1 mA, Vol1		0.4	V
OUTPUT LOW VOLTAGE (IRQ OUTPUT), Iol = 2 mA, Vol2		0.4	V
OUTPUT LOW VOLTAGE (CMOS INTERFACE), Iol = 8 mA, Vol3		0.4	V
OUTPUT HIGH VOLTAGE (TTL INTERFACE), Ioh = -1 mA, Voh1	2.4		V
OUTPUT HIGH VOLTAGE (CMOS INTERFACE AND IRQ)	OPEN DRAIN		

Current specifications for all I/O lines within 0.5 volts of Vcc or GND.

	MIN	MAX	UNITS
INPUT LOW CURRENT (ALL INPUTS), Iil	-1	1	uA
INPUT HIGH CURRENT (ALL INPUTS), Iih	-1	1	uA
OUTPUT 3 STATE CURRENT (TTL IHTERFACE), Ioz ..	-10	10	uA
STANDBY CURRENT, Idd		250	uA
INPUT CAPACITANCE, Cin		10	pF

4.2 A.C. CHARACTERISTICS

Top = 25 DEGREES C

OUTPUT LOW TRANSITION TIMES..100pf/4510ohms..	250*		nsec
(CMOS INTERFACE) ......3000/670ohms.....		2.5**	usec

* MEASURED FROM 90% OF Vdd TO 10% OF Vdd

** MEASURED FROM THE INITIATION OF THE TRANSITION INCLUDING PROPAGATION DELAY AND UNTIL THE OUTPUT REACHES 0.4 VOLTS

5.0 REQUIREMENTS:

5.1 PHYSICAL: 28 - 400, DUAL-IN-LINE PACKAGE, 70 MIL PIN CENTERS.

5.1.1 LEADS: SOLDERABLE PER MIL-STD-883, METHOD 2003, WITHOUT AGING.

5.1.2 PIN CONFIGURATION: SEE FIGURE 2.

5.1.3 MARKING: PARTS SHALL BE MARKED WITH THE TI PART NUMBER AND TNE DATE CODE.

```
              ______________
I/O-0----| 1          28|---- Vcc
I/O-1----| 2          27|---- O-0 *       * changed from
I/O-2----| 3          26|---- O-1           previous spec.
I/O-3----| 4          25|---- O-2 *
BAVI ----| 5          24|---- O-3:R/W
BAV  ----| 6          23|---- D-3I
HSKI ----| 7          22|---- D-3
HSK  ----| 8          21|---- D-2I
IRQ  ----| 9          20|---- D-2
E    ----|10          19|---- D-1I
CS   ----|11          18|---- D-1
RES  ----|12          17|---- D-0I
MS   ----|13          16|---- D-0
GND  ----|14          15|---- RS
              ______________
```

FIGURE 2. PIN CONFIGURATION.

6.0 PIN ASSIGNMENT FUNCTIONS

NAME	PIN	I/O	DESCRIPTION
D-0 D-1 D-2 D-3	16/17 18/19 20/21 22/23	I/O I/O I/O I/O	Data I/O lines that allow data transfer between IBC and ALC I/O bus.
HSK	7/8	I/O	HANDSHAKE LINE: set low by source device to indicate to listeners that there is valid data on the ALC I/O bus and held low by the listeners until they accept the data.
BAV	5/6	I/O	BUS AVAILABLE LINE: set low by source device in the beginning of the message and held low until the end of the message. The new source can originate a new message or "Request Service" only if this line is high.
I/O-0 I/O-1 I/O-2 I/O-3	1 2 3 4	I/O I/O I/O I/O	Data I/O lines that allow data transfer between IBC and the microprocessor. If MS=1 lines act as inputs only.
IRQ	9	O	INTERRUPT OUTPUT (open drain): indicates to to the microprocessor the occurence of the next data nibble on the ALC I/O bus.
CS	11	I	CHIP SELECT INPUT: selects and enables the IBC for microprocessor data transfer.
RS	15	I	REGISTER SELECT INPUT: address line through which the IBC registers can be accessed by the microprocessor. When IBC operates in the latched mode this input works as address strobe.
E	10	I	ENABLE INPUT: if MS=0 there are several options in combination with a pin R/W for controlling data transfer between IBC and the microprocessor. If MS=1, E is used as active low strobe for writing data into IBC.

6.0 PIN ASSIGNMENT FUNCTIONS (continued)

NAME	PIN	I/O	DESCRIPTION
R/W (O-3)	24	I/O	READ-WRITE CONTROL: if MS=0 there are several options in combination with a pin E for controlling direction of data transfer between IBC and the microprocessor. If MS=1, R/W is used as most significant bit of data outputs.
O-0 O-1 O-2	27 26 25	I/O I/O I/O	OUTPUT DATA LINES: if MS=1 these lines are used as three least significant bits of data outputs. If MS=0 these lines are used for selecting the options for pins R/W, E and RS.
MS	13	I	MODE SELECT: input which selects the mode of operation.
RES	12	I	RESET INPUT: low level on this input will put IBC into reset state.
Vcc	28		Positive supply (4.5 - 5.5 Vdc).
GND	14		0 Volt reference.

NOTE:

D0-D3, HSK, and BAV have two pin numbers listed. These I/O lines are designed with options so that the input and output of each function may be brought out to a common pin or to separate pins. When brought out separately the even pins (6, 8, 16, 18, 20, 22) are outputs and the odd numbered pins are inputs. When combined as I/O pins, the input and output are both tied to the even numbered pins.

7.0 INTERNAL ORGANIZATION

7.1 BLOCK DIAGRAM: SEE FIGURE 3.

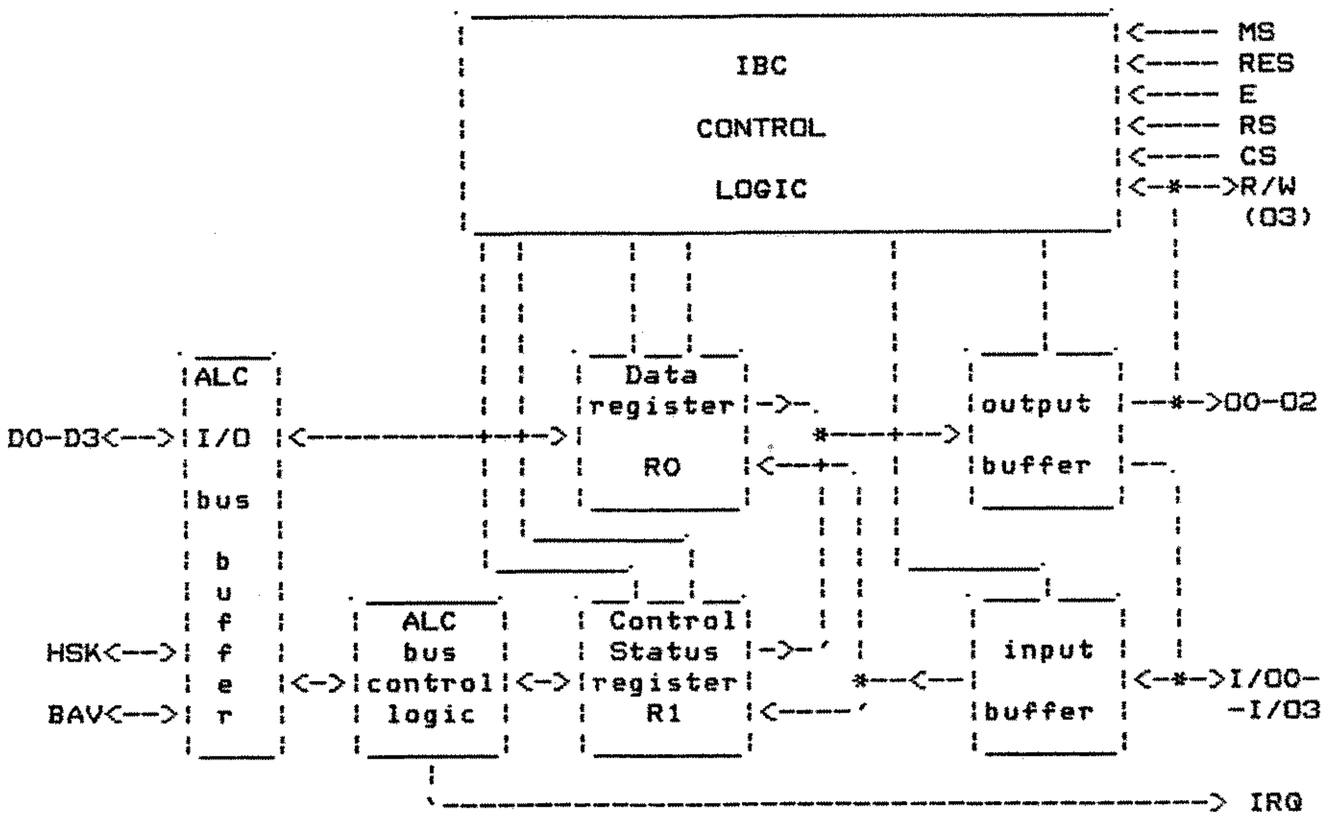


FIGURE 3. BLOCK DIAGRAM.

7.2 REGISTERS

The IBC has four registers which are accessible to the user. All registers are 4-bit wide. Table 1 shows addressing technique used to access each register. The Control Register is used to manipulate HSK and BAV lines. The mode of operation (Inhibit, Disable and Enable) is determined by the Control Register, which is depicted in Figure 4. The Status Register is used to indicate to the processor the status of HSK and BAV lines and the state of Interrupt request and the Inhibit flag. The Status Register is shown in 7.2.3. Transmit and Receive Data Registers are 4-bit buffers into and from the ALC's I/O Bus.

7.2.1 REGISTER DECODING

CS	RS	WRITE	READ
0	0	Transmit Data Register	Receiver Data Register
0	1	Control Register	Status Register

7.2.2 CONTROL REGISTER

```
| 3 | 2 | 1 | 0 |
  |   |   |   `---- HSK/RESET: 1-Set HSK low, reset IRQ
  |   |   |                    0-Release HSK
  |   |   `---- BAV: 1-Set BAV low, 0-Release BAV when HSK low
  |   `---- INHIBIT: 1-Inhibits HSK latching and IRQ til new BAV
  `---- DISABLE: 1-Disables clearing of Inhibit by new BAV
```

```
                  3   2   1   0
HARDWARE RESET  | 1 | 1 | 0 | 0 |
```

7.2.3 Status Register

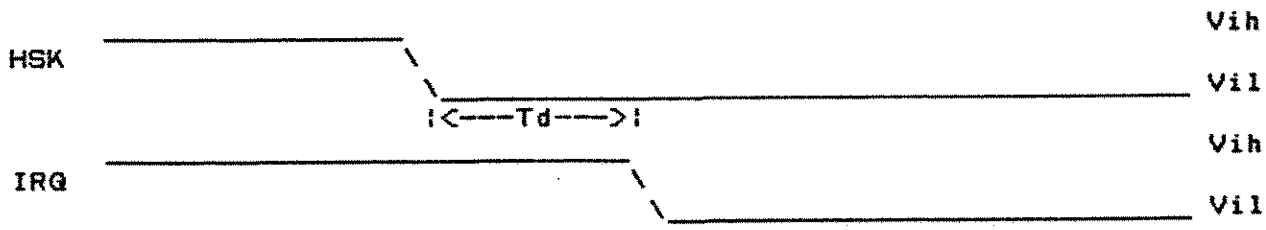
```
| 3 | 2 | 1 | 0 |
  |   |   |   `----- HSK LINE  (1=line low)
  |   |   `--------- BAV LINE  (1=line low)

| 0 | 0 |----------- ENABLE STATE
| 0 | 1 |----------- INHIBIT/DISABLE STATE
| 1 | 0 |----------- ACTIVE IRQ (HSK latched low)
| 1 | 1 |----------- START OF MESSAGE  (new BAV)
```

```
                  3   2   1   0
HARDWARE RESET  | 0 | 1 | X | X |
```

7.3 INTERRUPT

The IBC has an interrupt which indicates when the data is available on the bus. Interrupt can be disabled completely or until the next message. The interrupt output is open drain with a low active level. The pull-up resistor value is between 3.3 K and 4.7 K Ohms.

7.3.1 INTERRUPT AC CHARACTERISTICS



Td.....IRQ active after HSK fall..................200 ns (max)

8.0 FUNCTIONAL OPERATION DESCRIPTION

8.1 POWER-UP CONSIDERATIONS

If system's power up reset is tied to the RESET pin on IBC, the controller will come up in the Disable state which means that transitions on HSK line will not be latched and no interrupt will be generated. In order to enable the controller, the user has to write "0000" into Control Register. However it might not be desirable to enable the controller in the middle of the message therefore user may perform Enable with "Inhibit Until New Message" by Writing "0100" into Control Register. This will keep the controller in disable state until a new message starts.

8.2 DISABLE MODE

In disable mode the controller will not latch HSK signal and will not produce interrupt, but user will be able to monitor state of BAV and HSK by reading Status Register. In order to put device in disable mode user should Write "1100" into the Control Register.

8.3 INHIBIT (UNTIL NEW MESSAGE)

In inhibit (until new message) mode the controller will be disabled during current message and fully enabled when next message starts. In order to put device into inhibit (until new message) user should Write "0100" into the Control Register.

8.4 LISTEN MODE

In listen mode the user has to read the first two nibbles on the bus, determine if he was selected, and if he was, the user can start communication with the master. If he was not selected the user should perform "Inhibit Until New Message" by Writing "0100" into the Control Register.

8.5 MONITOR MODE

The capability exists for user to monitor the communication even if he was not selected, however, this is not recommended since it may slow down the bus operation. This capability allows monitoring and recording of all bus communications.

8.6 TRANSMIT MODE

In transmit mode, the user has to send the data utilizing the Write Sequence, monitor the HSK line and when the HSK line goes into an inactive state the user may transmit the next data.

8.7 DATA TRANSFER OPERATION

All receive and transmit data can be handled by executing the Read and Write Sequences respectively.

8.7.1 READ SEQUENCE

The Read Sequence allows the user to obtain the data transmitted on the bus and prepares IBC to receive the next transmission. The Read Sequence has to be used whenever the user is in the listen mode and receives an interrupt from IBC.

a – Reset Interrupt – Write "0001" into Control Register
b – Obtain Data – Read Receiver Data Register
c – Reset HSK Latch – Write "0000" into Control Register

8.7.2 WRITE SEQUENCE

The Write Sequence allows the user to place the data on the bus and to signal the other devices that the data is available. The Write Sequence has to be used whenever the user is in transmit mode and ready to send the data.

a – Prepare Data – Write Data into Transmit Data Register
b – Set HSK Signal – Write "0001" into Control Register
c – Reset HSK Signal – Write "0000" into Control Register

8.8 REQUEST SERVICE SEQUENCE

The Request Service Sequence allows the user to signal the master by putting the BAV line into the low state until the first HSK transition. The Request Service Sequence may be used whenever the user has permission from the master to request the service.

a - Set BAV signal - Write "0010" into Control Register
b - Enable BAV Reset - Write "0000" into Control Register

8.9 MASTER MODE DIFFERENCES

The difference between master and slave modes of operation exists because in master mode the user has to hold the BAV line low during the entire message. Therefore the Read and Write Sequences shown above must be modified for master mode operation. The modification requires that all accesses to the Control Register must maintain the BAV bit in the high state.

9.0 MODES OF OPERATION

The IBC can operate in 7 different modes which allows simple interface with several popular microprocessor families. The desired mode can be selected by setting the Mode Select input (MS). If input MS is set low three modes can be selected by setting pins O-0 and O-1. Table 1 shows the modes selected by different bit patterns on these pins.

In addition, when MS is set low, a high on pin O-2 will enable the latching capability for CS and RS signals. This feature is useful for multiplex address/data bus similar to TMS-7000 or INTEL-8085. In this mode ALATCH is connected to the RS pin and on the falling edge of a pulse, RS will latch the level on I/O-0 pin as internal RS and the level on CS pin as internal CS which should be high to select the chip.

If the MS is set high the IBC is in Mode 7 and reconfigured as shown in Section 9.7. This mode is useful with 4-bit MPUs which don't have a bidirectional bus. Please note that pin R/W is used as MSB of the output data.

9.0 MODES OF OPERATION (continued)

Table 1. Modes of Operation.

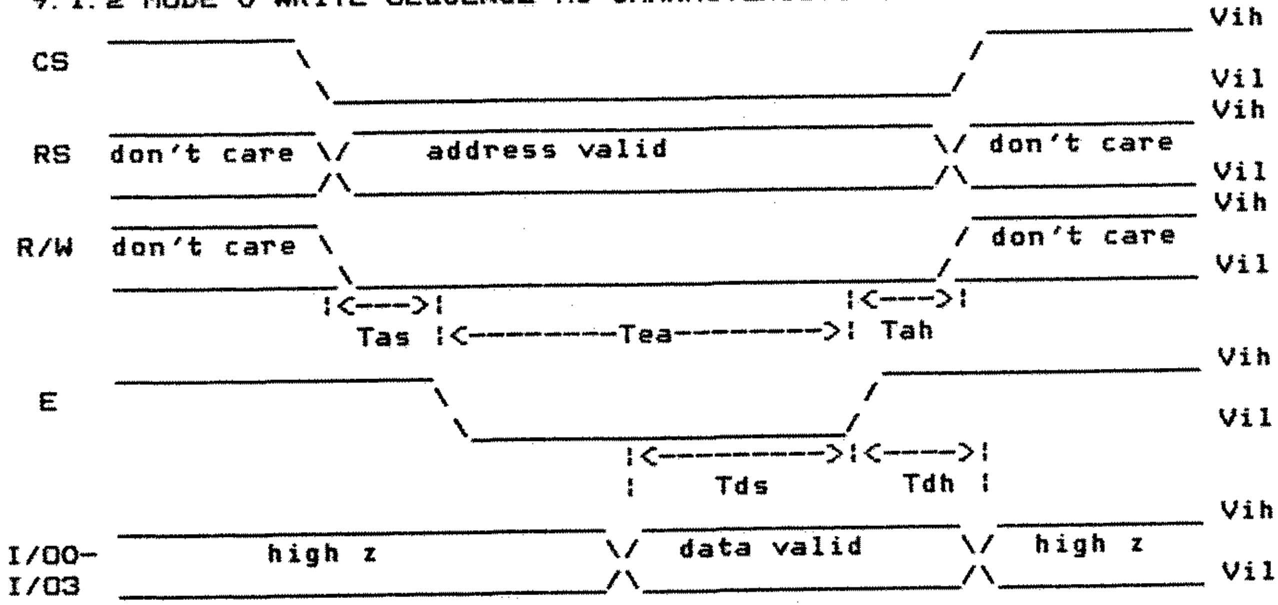
O-2	O-1	O-0	MODE DESCRIPTION
0	0	0	Mode 0 - ENABLE active low (see Figure 3) TMS-7000 compatible
0	0	1	Mode 1 - ENABLE active high (see Figure 4) 6500 compatible
0	1	0	Mode 2 - accepts RD- and WR- signals (see Figure 5) 8085 and 8048 compatible
1	X	X	Modes 4,5,6 - control signals for R/W and E correspond to non-multiplex modes 0,1,2 but addressing arranged for multiplex busses.

9.1 MODE 0 SPECIFICATIONS

9.1.1 MODE 0 INTERFACE CONFIGURATION

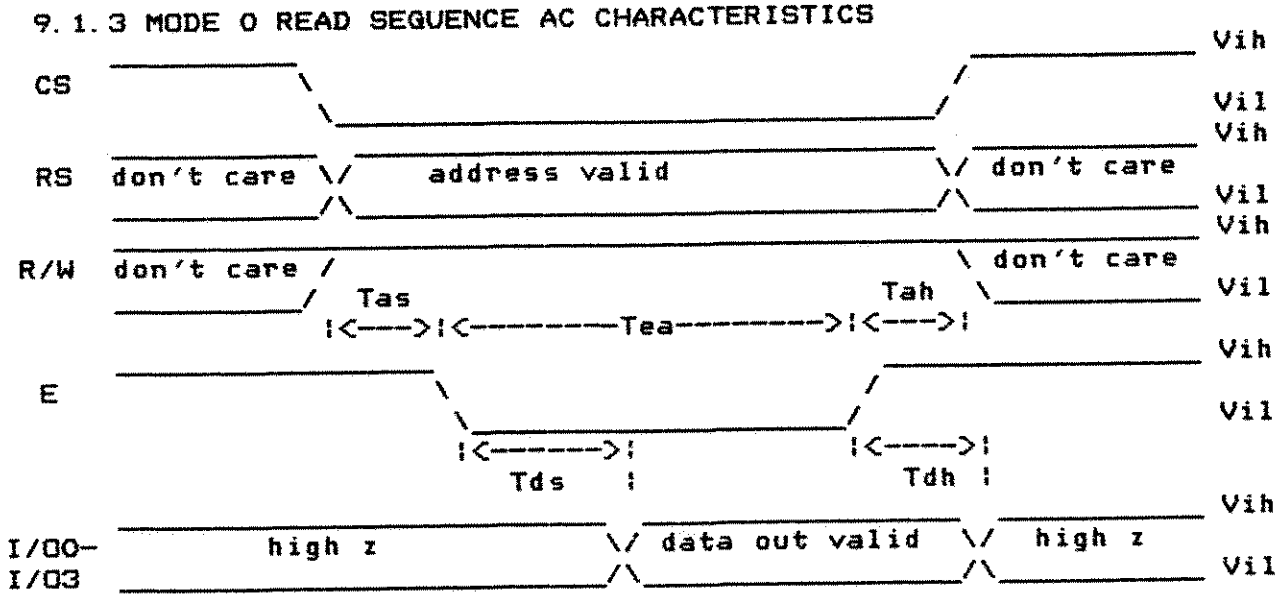
```
          TMS-7000                                         IBC
     -------------------                        -------------------
     |                 |                        |                 |
     |                 | /-------------------\  |                 |
     |   PC-0 - PC-3   |<  4-bit data bus     >|I/O-0 - I/O-3    |
     |                 | \-------------------/  |                 |
     |                 |                        |                 |
     |            R/W- |----------------------->|R/W              |
     |                 |                        |                 |
     |              E- |----------------------->|E           O-2  |----+
     |                 |                        |            O-1  |----|
     |            INT1 |<-----------------------|IRQ         O-0  |----|
     |                 |                        |             MS  |----|
     |                 |       ----------       |                 |    |
     |                 |---\  |address |------>|RS                |  __|__
     |   PD-0 - PD-7   |    >|        |        |                 |   \\\\
     |                 |---/  |decoder |------>|CS                |
     |                 |       ----------       |                 |
     -------------------                        -------------------
```

9.1.2 MODE 0 WRITE SEQUENCE AC CHARACTERISTICS



Tas....Address and R/W valid before E fall.......... 50 ns (min)
Tah....Address and R/W hold valid after E rise...... 30 ns (min)
Tds....Data in valid before E rise.................150 ns (min)
Tdh....Data in hold valid after E rise............. 65 ns (min)
Tea....Enable active...............................300 ns (min)

9.1.3 MODE 0 READ SEQUENCE AC CHARACTERISTICS



Tas....Address and R/W valid before E fall.......... 50 ns (min)
Tah....Address and R/W hold valid after E rise...... 30 ns (min)
Tds....Data out valid after E fall.................120 ns (max)
Tdh....Data out hold valid after E rise............ 60 ns (max)
Tea....Enable active...............................300 ns (min)

9.2 MODE 1 SPECIFICATIONS

9.2.1 MODE 1 INTERFACE CONFIGURATION

```
              6500                                        IBC
      ------------------                        ------------------
      |                |                        |                |
      |                | /------------------\   |                |
      |    D-0 - D-3   |<  4-bit data bus   >|I/O-0 - I/O-3      |
      |                | \------------------/   |                |
      |                |                        |                |
      |          R/W-  |----------------------->|R/W             |
      |                |                        |                |
      |         PHI-2  |----------------------->|E          O-0  |------> +V
      |                |                        |                |
      |          IRQ-  |<-----------------------|IRQ        O-1  |----+
      |                |                        |           O-2  |----|
      |                |      -----------       |            MS  |----|
      |                |---\  |address|-------->|RS              |    |
      |   A-0 - A-15   |    >|       |          |                |  __|__
      |                |---/  |decoder|-------->|CS              |  \\\\
      |                |      -----------       |                |
      ------------------                        ------------------
```

9.2.2 MODE 1 WRITE SEQUENCE AC CHARACTERISTICS

```
                                                                   _____________ Vih
 CS       ____________\                                          /
                       \________________________________________/                Vil
          ____________   _______________________________________   _____________ Vih
 RS       don't care \/      address valid                      \/ don't care
          ____________/\_______________________________________/\_____________ Vil
          ____________                                              _____________ Vih
 R/W      don't care \                                           / don't care
          ____________\_________________________________________/_____________ Vil
                      |<--->|                              |<--->|
                        Tas |<---------Tea--------->|  Tah
                                 ________________________                         Vih
 E                              /                        \
          _____________________/                          \_____________________ Vil
                                          |<----------->|<---->|
                                          |     Tds        Tdh |
          ________________________________   ___________________   ___________ Vih
 I/O0-             high z                 \/  data valid        \/  high z
 I/O3     ________________________________/\___________________/\___________ Vil
```

Tas....Address and R/W valid before E rise.......... 50 ns (min)
Tah....Address and R/W hold valid after E fall...... 30 ns (min)
Tds....Data in valid before E fall..................100 ns (min)
Tdh....Data in hold valid after E fall.............. 60 ns (min)
Tea....Enable active................................140 ns (min)

9.2.3 MODE 1 READ SEQUENCE AC CHARACTERISTICS

```
                                                                         Vih
 CS      ____________                                    ______________
                     \                                  /
                      \________________________________/                 Vil
         ____________  ______________________________  ______________  Vih
 RS      don't care \/       address valid           \/ don't care
         ___________/\______________________________/\______________   Vil
         ____________________________________________________________  Vih
 R/W     don't care /                                 \  don't care
         __________/                                   \____________   Vil
                   |<--->|                          |<--->|
                     Tas |<--------Tea--------->|  Tah
                                                                        Vih
 E                        ________________________
                         /                        \
         _______________/                          \_______________    Vil
                         |<------->|                |<---->|
                              Tds  |                   Tdh |
                                                                        Vih
 I/O0-   ______________________  ________________  ______________
              high z           \/ data out valid \/   high z
 I/O3    ______________________/\________________/\______________     Vil
```

Tas....Address and R/W valid before E rise.......... 50 ns (min)
Tah....Address and R/W hold valid after E fall...... 30 ns (min)
Tds....Data out valid after E rise.................120 ns (max)
Tdh....Data out hold valid after E fall............ 60 ns (max)
Tea....Enable active...............................140 ns (min)

9.3 MODE 2 SPECIFICATIONS

9.3.1 MODE 2 INTERFACE CONFIGURATION

```
              8085                                        IBC
       _______________                              _______________
       |             |                              |             |
       |             |  /----------------\          |             |
       | AD-0 - AD-3 |<  4-bit data bus   >| I/O-0 - I/O-3        |
       |             |  \----------------/          |             |
       |             |                              |             |
       |         WR- |----------------------------->| R/W         |
       |             |                              |             |
       |         RD- |----------------------------->| E       O-1 |------> +V
       |             |                              |             |
       |    RST 5.5  |<-----        ----------------| IRQ     O-2 |---+
       |             |                              |         O-0 |---|
       |             |       __________             |          MS |---|
       |             |---\  |address  |------------>| RS          |   |
       |  A-7 - A-15 |    > |         |             |             | __|__
       |             |---/  |decoder  |------------>| CS          |  \\\\
       |_____________|       ----------             |_____________|
```

size		drawing no
scale	rev	sheet 18 of 29

9.3.2 MODE 2 WRITE SEQUENCE AC CHARACTERISTICS

```
         ____________                                      _______________ Vih
  CS                 \                                    /
                      \__________________________________/                 Vil
         _____________   _________________________________   _____________ Vih
  RS     don't care  \/     address valid                 \/ don't care
         _____________/\_________________________________/\_______________ Vil
                      |<--->|                        |<--->|
                        Tas |<---------Tsv---------->| Tah
         ____________________                         ____________________ Vih
 R/W                         \                       /
                              \_____________________/                      Vil
         ____________________________________________|____________________ Vih
  E                                                  |
                                            Tds      |  Tdh                Vil
                                    |<-------------->|<----->|
         ___________________________   _____________________   ___________ Vih
 I/O0-           high z             \/  data valid          \/  high z
 I/O3    ___________________________/\_____________________/\_____________ Vil
```

Tas....Address valid before R/W fall................ 50 ns (min)
Tah....Address hold valid after R/W rise............ 50 ns (min)
Tds....Data in valid before R/W rise................150 ns (min)
Tdh....Data in hold valid after R/W rise............ 60 ns (min)
Tsv....Write pulse width on R/W pin.................140 ns (min)

9.3.3 MODE 2 READ SEQUENCE AC CHARACTERISTICS

```
         ____________                                      _______________ Vih
  CS                 \                                    /
                     |\__________________________________/|                Vil
         _____________   _________________________________   _____________ Vih
  RS     don't care  \/     address valid                 \/ don't care
         _____________/\_________________________________/\_______________ Vil
         ____________|_____________________________________|______________ Vih
 R/W                 |                                     |
                     | Tas                             Tah |               Vil
                     |<--->|<---------Tea----------->|<--->|
         ____________________                         ____________________ Vih
  E                          \                       /
                              \_____________________/                      Vil
                             |<------->|             |<----->|
                                 Tds   |                Tdh  |
         ___________________________   _____________________   ___________ Vih
 I/O0-           high z             \/  data out valid      \/  high z
 I/O3    ___________________________/\_____________________/\_____________ Vil
```

Tas....Address valid before E fall.................. 50 ns (min)
Tah....Address hold valid after E rise.............. 50 ns (min)
Tds....Data out valid after E fall..................120 ns (max)
Tdh....Data out hold valid after E rise............. 60 ns (max)
Tea....Read pulse width on E pin....................140 ns (min)

9.4 MODE 4 SPECIFICATIONS

9.4.1 MODE 4 INTERFACE CONFIGURATION

```
           TMS-7000                                        IBC
      ----------------------                    ---------------------
      |                    | /------------------\ |                   |
      |    PC-0 - PC-3|<  4-bit data bus  >|I/O-0 - I/O-3  |
      |                    | \------------------/ |                   |
      |              R/W-|-------------------->|R/W         O-2|---->+V
      |                E-|-------------------->|E                   |
      |                    |                      |             O-1|---+
      |             INT1|<--------------------|IRQ         O-0|---|
      |                    |                      |              MS|---|
      |           ALATCH|-------------------->|RS                |  __|__
      |              C-7|-------------------->|CS                |   \\\\
      ----------------------                    ---------------------
```

9.4.2 MODE 4 WRITE SEQUENCE AC CHARACTERISTICS

```
                                                                      Vih
          ______________________________________________________________
  CS      don't care/          \      don't care
          _________/Tcs | Tch \___________________________________   Vil
                   |<-->|<-->|
                                                                      Vih
                      _______
  RS                 /       \
 (ALATCH)______/    Taa     \_________________________________________ Vil
                |<------>|<-->|
                   |<-->| Tah|
          _________Tas| Tev|                                _________ Vih
  R/W     don't care \     |                           /don't care
                       \_____|_________________________/___________    Vil
          ___________________
                   |   |<---->|                     |<--->|
                   |   Trs |<---------Tea--------->| Trhe
          _________|_________                                  _________ Vih
  E                |         \                              /
                   |         | \______________Tds______/  Tdh          Vil
                   |         |           |<---------->|<---->|
                                                                      Vih
          ________  __________  ________  ___________  ________
 I/O0-    high z \/ address\/ high z \/  data valid   \/  high z
 I/O3     ________/\__________/\________/\___________/\________       Vil
```

Taa....Alatch pulse width on RS pin................100 ns (min)
Tas....Address valid before RS(ALATCH) fall......... 30 ns (min)
Tah....Address hold valid after RS(ALATCH) fall..... 30 ns (min)
Tcs....Chip Select valid before RS(ALATCH) fall..... 30 ns (min)
Tch....Chip Select hold valid after RS(ALATCH) fall. 30 ns (min)
Tev....E valid after RS(ALATCH) fall................ 30 ns (min)
Trs....R/W valid before E fall...................... 30 ns (min)
Trh....R/W hold valid after E rise.................. 30 ns (min)
Tds....Data in valid before E rise..................150 ns (min)
Tdh....Data in hold valid after E rise.............. 65 ns (min)
Tea....Enable active................................250 ns (min)

9.4.3 MODE 4 READ SEQUENCE AC CHARACTERISTICS

```
          __________________________________________________________  Vih
  CS      don't care/        \       don't care
          __________/     :   \_____________________________________  Vil
                    :<-->:<-->:
                     Tcs : Tch
                                                                      Vih
                  _______
   RS            /       \
 (ALATCH)_______/   Taa   \_________________________________________  Vil
                :<------>:<-->:
                  :<-->: Tah:
                  : Tas: Tev:
          ___________________:________________________________________  Vih
  R/W      don't care /      :                        \don't care
          ___________/  Trs  :                    Trhe \____________  Vil
                   :  :<---->:<---------Tea--------->:<--->:
          __________:________                       ________________  Vih
   E                :        \                      /
                    :        : \___________________/                  Vil
                    :        :   Tds                 :  Tdh
                    :        :<------->:             :<---->:
          ________  _________  ________  _________________  ________  Vih
  I/O0-    high z \/ address\/ high z \/ data out valid \/ high z
  I/O3    ________/\_________/\________/\_________________/\________  Vil
```

Taa....Alatch pulse width on RS pin................100 ns (min)
Tas....Address valid before RS(ALATCH) fall......... 30 ns (min)
Tah....Address hold valid after RS(ALATCH) fall..... 30 ns (min)
Tcs....Chip Select valid before RS(ALATCH) fall..... 30 ns (min)
Tch....Chip Select hold valid after RS(ALATCH) fall. 30 ns (min)
Tev....E valid after RS(ALATCH) fall................ 30 ns (min)
Trs....R/W valid before E fall...................... 30 ns (min)
Trh....R/W hold valid after E rise.................. 30 ns (min)
Tds....Data out valid after E fall..................120 ns (max)
Tdh....Data out hold valid after E rise............. 60 ns (max)
Tea....Enable active................................250 ns (min)

9.5 MODE 5 SPECIFICATIONS

MODE 5 SPECIFICATIONS ARE THE SAME AS IN MODE 4 EXCEPT PULSE ON "ENABLE" LINE IS INVERTED.

9.6 MODE 6 SPECIFICATIONS

9.6.1 MODE 6 INTERFACE CINFIGURATION

```
              8085                                          IBC
        ----------------                         ----------------
        |              | /------------------\ |               |
        |  AD-0 - AD-3|<  4-bit data bus  >|I/O-0 - I/O-3  |
        |              | \------------------/ |               |
        |          WR-|--------------------->|R/W             |
        |          RD-|--------------------->|E          O-2|---+--> +V
        |              |                      |           O-1|--'
        |      RST 5_5|<-----      ----------|IRQ             |
        |              |                      |           O-0|----+
        |          ALE|--------------------->|RS           MS|----|
        |         AD-7|--------------------->|CS             | __|__
        ----------------                     ----------------   \\\\
```

9.6.2 MODE 6 WRITE SEQUENCE AC CHARACTERISTICS

```
        _____________________________________________________________ Vih
  CS    don't care/          \      don't care
        __________/           \______________________________________ Vil
                  |<-->|<-->|
                   Tcs | Tch
                                                                      Vih
                   _______
  RS              /       \
 (ALE)______/    Taa      \___________________________________________ Vil
            |<------>|<-->|
                 |<-->| Tah|
                 | Tas| Twv|
        ______________________                      _________________ Vih
  R/W               |      \                       /
                    |      | \____________________/                   Vil
                    |      |
                    |      |<--------Twa--------->|
        ____________|______|______________________|__________________ Vih
  E                 |      |              Tds     |  Tdh              Vil
                    |      |         |<---------->|<---->|
        __________  ________  _______  _________________   _________ Vih
 I/OO-   high z  \/ address\/ high z \/   data valid    \/  high z
 I/O3   _________/\________/\_______/\__________________/\__________ Vil
```

```
Taa....ALE pulse width on RS pin.................... 80 ns (min)
Tas....Address valid before RS(ALE) fall............ 50 ns (min)
Tah....Address hold valid after RS(ALE) fall........ 50 ns (min)
Tcs....Chip Select valid before RS(ALE) fall........ 50 ns (min)
Tch....Chip Select hold valid after RS(ALE) fall.... 50 ns (min)
Twv....R/W valid after RS(ALE) fall................. 50 ns (min)
Tds....Data in valid before R/W rise................150 ns (max)
Tdh....Data in hold valid after R/W rise............ 60 ns (max)
Twa....Write pulse width on R/W pin.................200 ns (min)
```

size		drawing no	
scale		rev	sheet 22 of 29

9.6.3 MODE 6 READ SEQUENCE AC CHARACTERISTICS

```
             _______________________________________________________________ Vih
    CS      don't care/           \        don't care
            ___________/            \_______________________________________ Vil
                        |<-->|<-->|
                         Tcs | Tch
                                                                            Vih
                    _______
    RS            /         \
 (ALATCH)______/    Taa     \_____________________________________________ Vil
                  |<----->|<-->|
                     |<-->| Tah|
                     | Tas| Tev|
            _________|_________|____________________________________________ Vih
   R/W               |         |
                     |         |                                            Vil
                     |         |<--------Tea--------->|
            _________|_________                         _____________________ Vih
    E                |          \                     /
                     |         | \___________________/    Tdh              Vil
                     |         |<-------->|             |<---->|
                     |              Tds   |                Tdh |
            ________   _________   _________   _________________   __________ Vih
  I/O0-     high z \/ address\/  high z \/   data out valid \/   high z
  I/O3      ________/\_________/\_________/\_________________/\__________ Vil
```

Taa....ALE pulse width on RS pin....................150 ns (min)
Tas....Address valid before RS(ALE) fall............ 50 ns (min)
Tah....Address hold valid after RS(ALE) fall........ 50 ns (min)
Tcs....Chip Select valid before RS(ALE) fall........ 50 ns (min)
Tch....Chip Select hold valid after RS(ALE) fall.... 50 ns (min)
Tev....E valid after RS(ALE) fall................... 50 ns (min)
Tds....Data out valid after E fall..................120 ns (max)
Tdh....Data out hold valid after E rise............. 60 ns (max)
Tea....Read pulse width on E pin....................200 ns (min)

size		drawing no	
scale		rev	sheet 23 of 29

9.7 MODE 7 SPECIFICATIONS

9.7.1 MODE 7 INTERFACE CONFIGURATION

```
           TMS-1000                                    IBC
    ------------------                        ------------------
    |               |-------------------\ |                    |
    |     O-O - O-3|   4-bit data out  >|I/O-0 - I/O-3       |
    |               |-------------------/ |                    |
    |               |                      |                    |
    |       K-1,K-2|  /-------------------|                    |
    |               |<  4-bit data in     |O-O - O-2,R/W       |
    |       K-4,K-8|  \-------------------|                    |
    |               |                      |                    |
    |           R-0|--------------------->|E                   |
    |           R-1|--------------------->|RS               MS|---> +V
    |           R-2|--------------------->|CS                  |
    ------------------                        ------------------
```

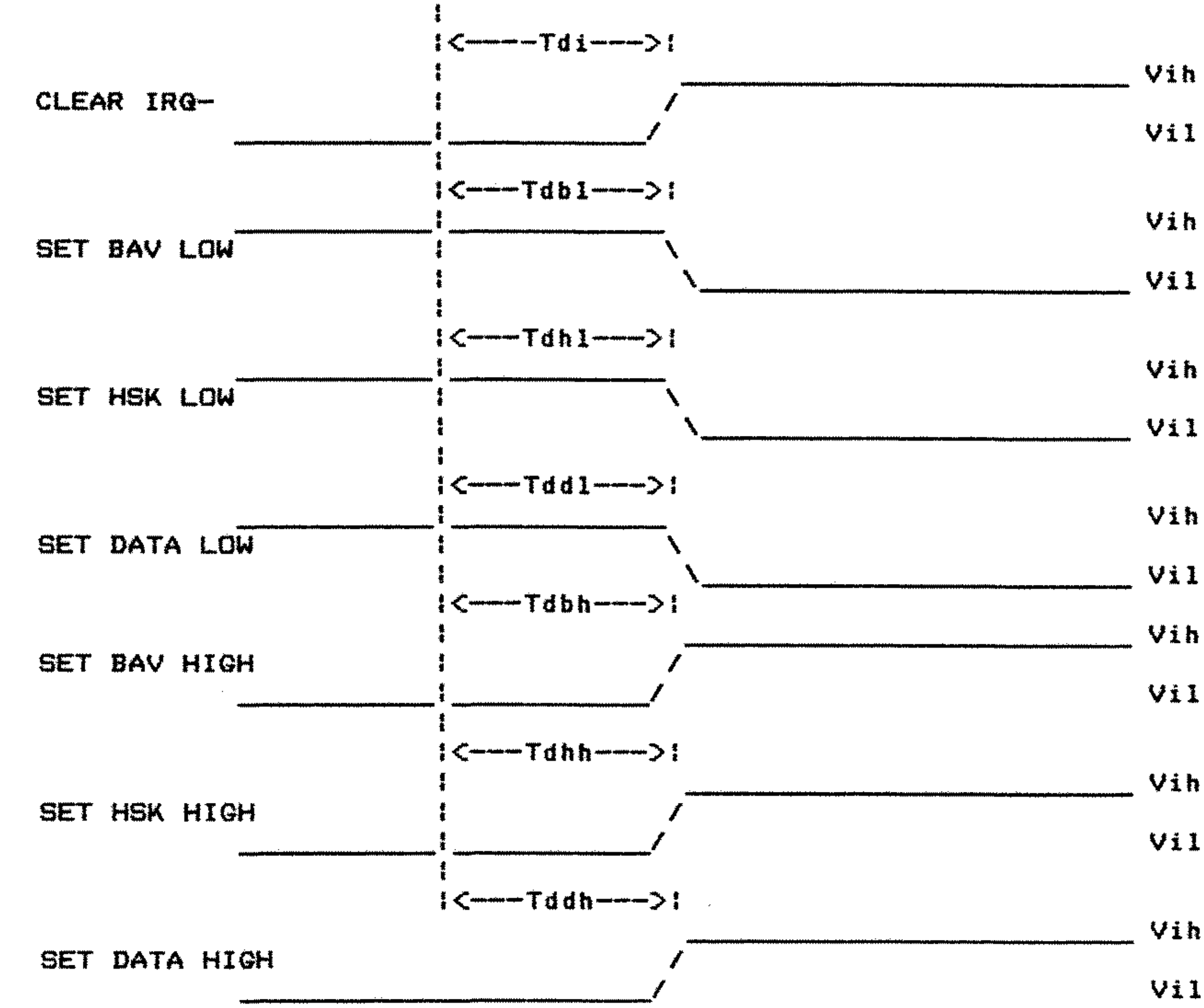
9.7.2 MODE 7 AC CHARACTERISTICS

```
          _____________                                     _________________ Vih
 CS                    \                                   /
                        \_________________________________/                   Vil
                        |                                  |
          _____________   _________________________________   _______________ Vih
 RS       don't care   \/     address valid                \/ don't care
          _____________/\_________________________________/\_______________ Vil
                        |                                  |
                        |<--->|                     |<--->|
                        | Tas |<--------Tea-------->| Tah |
          _____________ |______                      _____|_______________ Vih
 E                      |      \                    /     |
                        |       \__________________/                       Vil
                        |              |<-------->|<--->||
                        |              |    Tds      Tdh ||
          _____________ |____________   ______________   ______________ Vih
 I/OO-                  |high z      \/ data valid    \/       high z
 I/O3     _____________ |____________/\______________/\______________ Vil
                        |     Tdo                        | Tdoz
                        |<-------------->|               |<---->|
          _________________________________   __________________   __________ Vih
 OO-O2,              high z               \/   data out valid   \/  high z
  R/W     _________________________________/\__________________/\__________ Vil
```

Tas....Address valid before E fall.................. 50 ns (min)
Tah....Address hold valid after E rise............. 50 ns (min)
Tds....Data in valid before E rise..................120 ns (max)
Tdh....Data in hold valid after E rise............. 60 ns (max)
Tdo....Data out valid after CS fall................ 60 ns (max)
Tdo....Data out valid after CS rise................ 60 ns (max)
Tea....Write pulse width on E pin..................200 ns (min)

10.0 PERIPHERAL BUS INTERFACE AC CHARACTERISTICS

Write Operation Complete




```
Tdi....IRQ  high after Write cycle completion.......200 ns (max)
Tdb....BAV low after Write cycle completion.........200 ns (max)
Tdh....HSK low after Write cycle completion.........200 ns (max)
Tdd....Data pin low after Write cycle completion....200 ns (max)
Tdbh...BAV high after Write cycle completion........200 ns (max) + Tbd
Tdhh...HSK high after Write cycle completion........200 ns (max) + Tbd
Tddh...Data pin high after Write cycle completion...200 ns (max) + Tbd
```

Tbd - Bus delay timing depends on Pull-up value and number of peripherals on the bus.

Write cycle completion means end of the Write pulse on E or R/W pins corresponding to the mode selected and assuming that the right command is executed.

size		drawing no	
scale		rev	

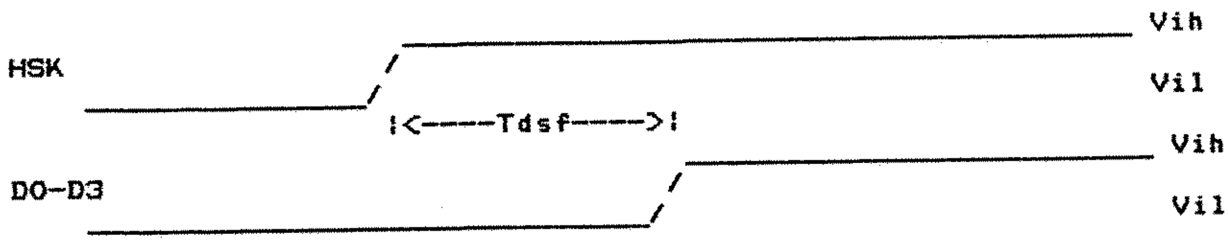
11.0 SPECIAL FUNCTIONS SPECIFICATIONS

11.1 DATA REGISTER SELF-CLEAR

11.1.1 PROCEDURE FOR TESTING DATA REGISTER SELF-CLEAR

1 Set Data Lines D0-D3 to "0000" by writing into Data Register
2 Set HSK low by writing "0001" into Control Register
3 Hold HSK low externally
4 Release internal HSK by writing "0000" into Control Register
5 Release HSK externally and observed time delay for Data Lines to rise

11.1.2 DATA REGISTER SELF-CLEAR AC CHARACTERISTICS



Tdsf...Data lines high after HSK rise...............200 ns (max) + Tbd

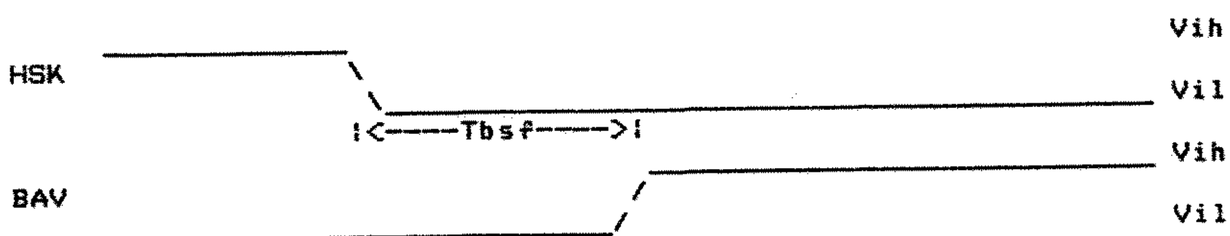
Tbd - Bus delay timing depends on Pull-up value and number of peripherals on the bus.

11.2 BAV HOLD LATCH SELF-CLEAR

11.2.1 PROCEDURE FOR TESTING BAV HOLD LATCH SELF-CLEAR

1 Set BAV low by writing "0010" into Control Register
2 Release internal BAV hold by writing "0000" into Control Register
3 Set HSK low externally and observed time delay for BAV to rise

11.2.2 BAV HOLD SELF-CLEAR AC CHARACTERISTICS



Tbsf...Internal BAV release after HSK fall..........200 ns (max) + Tbd

Tbd - Bus delay timing depends on Pull-up value and number of peripherals on the bus.

size		drawing no
scale	rev	sheet 26 of 29

11.2.3 BAV SETUP TIME

```
                                                                      Vih
 HSK     ____________
                     \
                      \_____________________________________________  Vil
       |<-Tbsu-->|
                                                                      Vih
 BAV     ___
            \
             \______________________________________________________  Vil
```

Tbsu...BAV setup time before HSK fall................5 u sec (min)

12.0 QUALITY ASSURANCE PROVISIONS

12.1 RESPONSIBILITY FOR INSPECTION

UNLESS OTHERWISE SPECIFIED IN THE CONTRACT OR PURCHASE ORDER, THE SUPPLIER SHALL BE RESPONSIBLE FOR PERFORMING INSPECTIONS THAT ARE SUFFICIENT TO ASSURE THAT THE PARTS SUPPLIED MEET THE REQUIREMENTS SPECIFIED HEREIN.

12.2 LOT ACCEPTANCE

LOTS FURNISHED TO THIS SPECIFICATION SHALL BE CAPABLE OF PASSING A SAMPLING INSPECTION FOR DEFECTS TO AN ACCEPTABLE QUALITY LEVEL (AQL) OF ONE PERCENT FOR NORMAL SINGLE SAMPLING, LEVEL II, PER MIL-STD-105. FAILING LOTS SHALL BE SUBJECT TO REJECTION.

12.3 LIFE FAILURE RATE

THE MEAN LIFE FAILURE RATE FOR DEVICES SHALL BE EQUAL TO OR LESS THAN .018%/1000 HOURS AT 55 deg C DERATED AT 0.5 EV.

13.0 BUS LOADING CALCULATIONS

# OF DEVICES	Req(Kohms)	Ceq(pF) 6"CABLES	RISE TIME (usec.)	Ceq(pF) 3'CABLES	RISE TIME (usec.)
CC-40 PLUS 1	4.51	430	2.52	480	2.81
2	3.01	660	2.58	760	2.97
3	2.25	890	2.60	1040	3.04
4	1.80	1120	2.62	1320	3.09
5	1.50	1350	2.63	1600	3.12
6	1.29	1580	2.65	1880	3.15
7	1.13	1810	2.66	2160	3.17
8	1.00	2040	2.65	2440	3.17
9	0.90	2270	2.66	2720	3.18
10	0.82	2500	2.67	3000	3.20

ASSUMPTIONS

1. RISE TIMES ARE CALCULATED FOR Vdd=4.5 VOLTS, MEASURING RISE FROM 0 TO 3.5 VOLTS.
2. PULLUPS ARE 8.2Kohms +/- 10%
3. Cin FOR EACH DEVICE IS 20 pF.
4. Ccable = 20 pF / ft.
5. Cconnector = 10 pF.
6. 180 pF CAPACITORS ARE USED ON ALL OUTPUTS FOR RFI.

MAXIMUM DC CURRENT CALCULATION

Req MINIMUM = 670 ohms

Vdd MAXIMUM = 5.5 volts

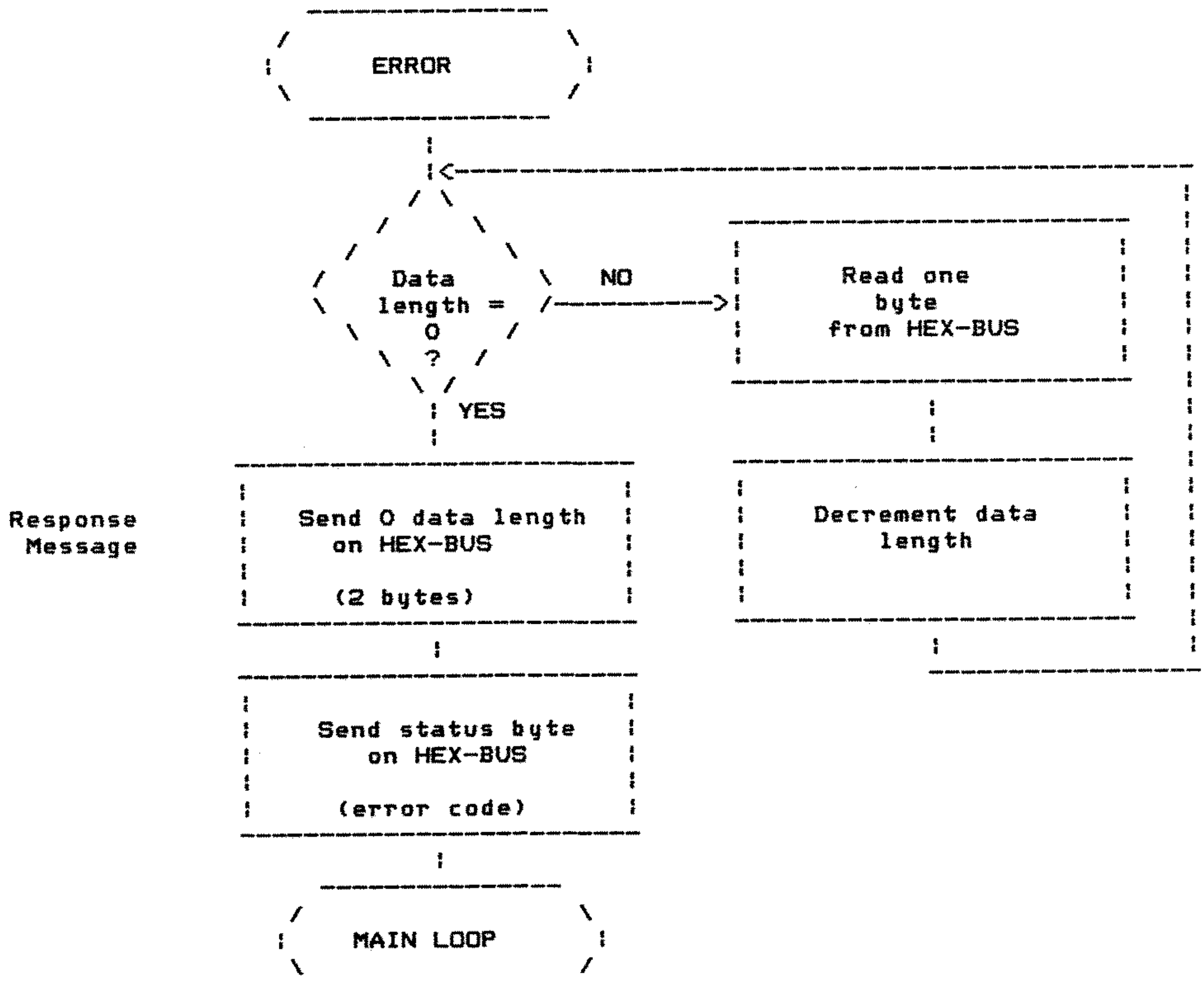
Isink MAXIMUM = 8.2 milliamps.

```
                 ---------------
               /      READ       \
              !    BYTE FROM      !
               \    HEX-BUS      /
                 ---------------
                        !
            -------------------------
            !                       !
            !      Release HSK      !
            !                       !
            ! ( 0 -> IBC control    !
            !        register )     !
            -------------------------
                        !
     ---------------    !
  /   READ INITIAL  \   !
 ! BYTE (DEVICE CODE)!--->!
  \     FROM BUS    /   !
     ---------------    !
                        !
                       / \
                     /     \
                   /   IBC   \                 ---------------
                 /   status    \   NO      /      RESET       \
        -------->\  register = /--------->!      command       !
        !          \  XX1X   / (BAV bit=0) \                  /
        !            \  ?  /                 ---------------
        !              \ /
        !               ! YES (BAV line still low)
        !               !
        !              / \
        !            /     \
        !          /   IBC   \
        !   NO   /   status    \
        ---------\  register = /
  (IRQ bit=0)  \   1XXX    /
                 \  ?  /
                   \ /
                    ! YES (digit ready to read)
                    !
            -------------------------
            !                       !
            !       Reset IRQ       !
            !                       !
            ! ( 1 -> IBC control    !
            !        register )     !
            -------------------------
                        !
            -------------------------
            !                       !
            !     Read least        !
            !   significant digit   !
            !   of byte from IBC    !
            !    data register      !
            -------------------------
                        !
            -------------------------
            !                       !
            !      Release HSK      !
            !                       !
            ! ( 0 -> IBC control    !
            !        register )     !
            -------------------------
```

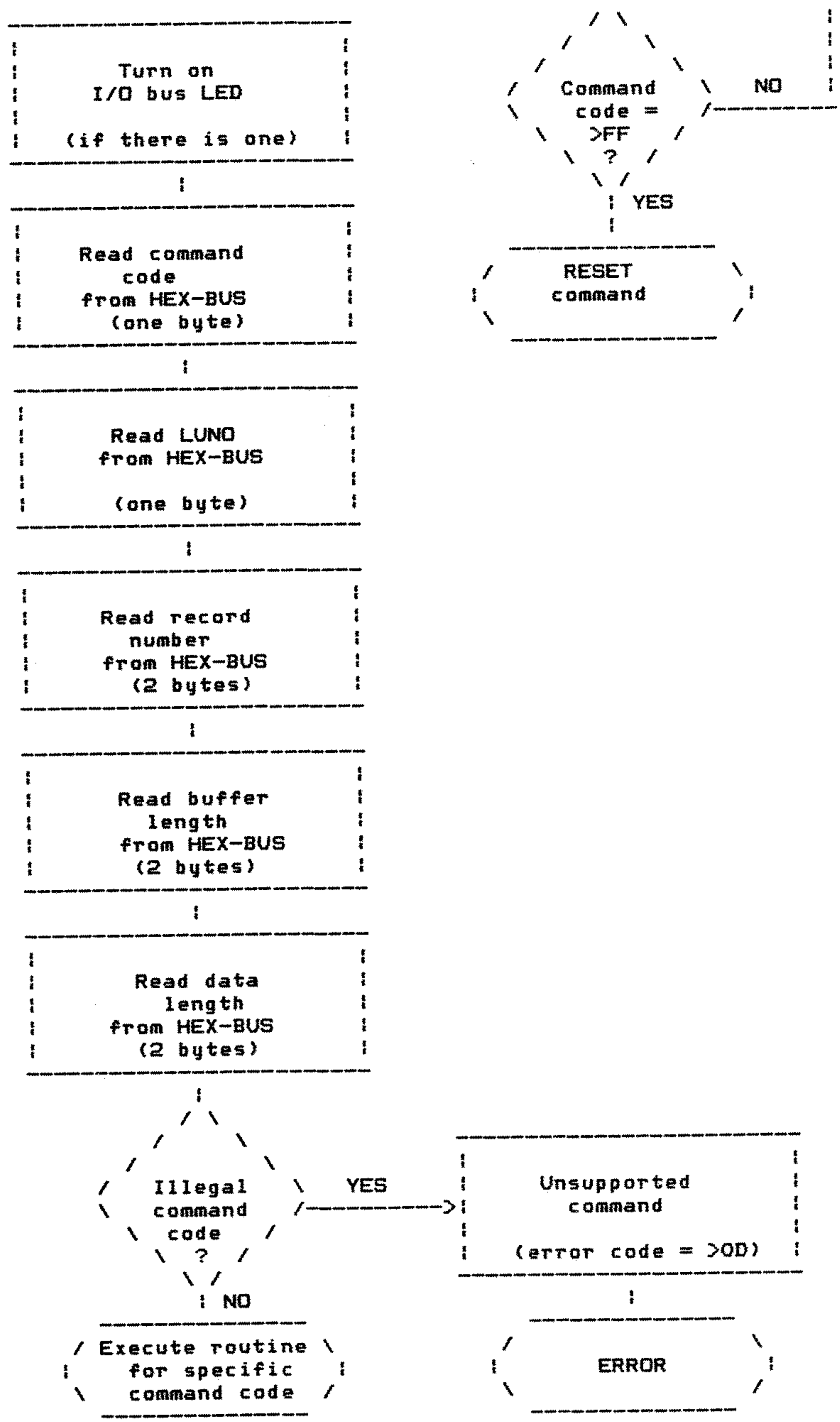
```
                         |
                        / \
                      /     \
                    /   IBC   \                    ---------------
                  /   status    \   NO          /       RESET        \
  ----------->\   register =  /----------->|       command         |
  |             \   XX1X    /  (BAV bit=0)  \                      /
  |               \   ?   /                    ---------------
  |                 \ /
  |                  |  YES (BAV bit=1)
  |                  |
  |                 / \
  |               /     \
  |             /   IBC   \
  |   NO      /   status    \
  ------------\   register = /
  (IRQ bit=0) \   1XXX    /
                \   ?   /
                  \ /
                   |  YES (IRQ bit=1)
                   |
        ------------------------------
        |                            |
        |         Reset IRQ          |
        |                            |
        |  ( 1 -> IBC control        |
        |         register )         |
        ------------------------------
                   |
        ------------------------------
        |       Read most            |
        |   significant digit        |
        |   of byte from IBC         |
        |     data register          |
        |      (hold HSK)            |
        ------------------------------
                   |
              ----------------
            /                  \
           |       return        |
            \                  /
              ----------------
```

```
                 ____________________
               /       WRITE          \
              |       BYTE TO          |
               \      HEX-BUS         /
                 ____________________
                        |
          ________________________________
          |                              |
          |      Release HSK             |
          |                              |
          | ( 0 -> IBC control           |
          |        register )            |
          ________________________________
                        |
                       / \
                      /   \
                     /  IBC  \
                    /  status  \   NO              /      RESET        \
    --------------->\ register = /--------------->|       command       |
    |                \  XX1X   / (BAV bit=0)      \                    /
    |                 \  ?   /                      ____________________
    |                  \   /
    |                    | YES (BAV bit=1)
    |                    |
    |                   / \
    |                  /   \
    |                 /  IBC \
    |   YES         /  status  \
    ----------------\ register = /
    (HSK bit=1)      \  XXX1   /
                      \  ?   /
                       \   /
                         | NO (HSK bit=0)
                         |
          ________________________________
          |                              |
          |     Write least              |
          |  significant digit           |
          |   of byte to IBC             |
          |    data register             |
          ________________________________
                         |
          ________________________________
          |                              |
          |   Set HSK line low           |
          |                              |
          |  (1 -> IBC control           |
          |         register)            |
          ________________________________
                         |
          ________________________________
          |                              |
          |     Release HSK              |
          |                              |
          | ( 0 -> IBC control           |
          |        register )            |
          ________________________________
                         |
```

```
                 / \
               /     \
             /   IBC   \                  -------------
           /   status    \   NO         /     RESET      \
--------->\  register =  /---------->:      command       :
 :         \   XX1X    / (BAV bit=0)  \                  /
 :           \   ?   /                  -------------
 :             \ /
 :              : YES (BAV bit=1)
 :              :
 :             / \
 :           /     \
 :         /   IBC   \
 :   YES  /   status   \
---------\  register = /
(HSK bit=1) \  XXX1   /
             \   ?   /
               \ /
                : NO (HSK bit=0)
                :
     -------------------------
     :                       :
     :       Write most      :
     :   significant digit   :
     :    of byte to IBC     :
     :     data register     :
     -------------------------
                :
     -------------------------
     :                       :
     :     Set HSK line low  :
     :                       :
     :  (1 -> IBC control    :
     :        register)      :
     -------------------------
                :
          -------------
       /                 \
      :       return       :
       \                 /
          -------------
```


ERROR
Data length = 0 ?
NO
YES
Read one byte from HEX-BUS
Decrement data length
Response Message
Send 0 data length on HEX-BUS
(2 bytes)
Send status byte on HEX-BUS
(error code)
MAIN LOOP

```
                      ------------------
                    /                    \
                   !     MAIN LOOP        !
                    \                    /
                      ------------------
                              !
-------------------------------->!<------------------------------------------------------
!                 ------------------------------                                        !
!                 !                            !                                        !
!                 !        Inhibit IBC         !                                        !
!                 !                            !                                        !
!                 ! ( 4 -> IBC control         !                                        !
!                 !      register )            !                                        !
!                 ------------------------------                                        !
!                               !                                                       !
!                 ------------------------------                                        !
!                 !                            !                                        !
!                 !         Turn off           !                                        !
!                 !       I/O bus LED          !                                        !
!                 !                            !                                        !
!                 !    (if there is one)       !                                        !
!                 ------------------------------                                        !
!                               !                                                       !
!            ------------------>!                                                       !
!            !                 / \                                                      !
!            !               /     \                                                    !
!            !             /   IBC   \                                                  !
!            !   NO      /   status    \                                                !
!            -----------\  register =  /                                                !
!                         \   11XX    /                                                 !
!                           \   ?   /                                                   !
!                             \ /                                                       !
!                               ! YES (start of new message)                            !
!                               !                                                       !
!                 ------------------------------                                        !
!                 !                            !                                        !
!                 !    Read initial byte       !                                        !
!                 !     (device code)          !                                        !
!                 !      from HEX-BUS          !                                        !
!                 !                            !                                        !
!                 ------------------------------                                        !
!                               !                                                       !
!                              / \                                                      !
!                            /     \                   ------------------------------   !
!                          /         \                 !                            !   !
!                        /   Device    \   YES         !         Turn on            !   !
!                        \   code =    /-------------->!       I/O bus LED          !   !
!                          \   00    /                 !                            !   !
!                            \  ?  /                   !    (if there is one)       !   !
!                              \ /                     ------------------------------   !
!                               ! NO                                 !                  !
!                               !                                    !                  !
!                              / \                                   !                  !
!                            /     \                   ------------------------------   !
!                          /   Is    \                 !                            !   !
!              NO        /   it my     \               !     Read commmand          !   !
-------------------------\   device    /               !         code               !   !
                           \  code   /                 !      from HEX-BUS          !   !
                             \  ?  /                   !      (one byte)            !   !
                               \ /                     ------------------------------   !
                                ! YES                                !                  !
                                !                                    !                  !
```



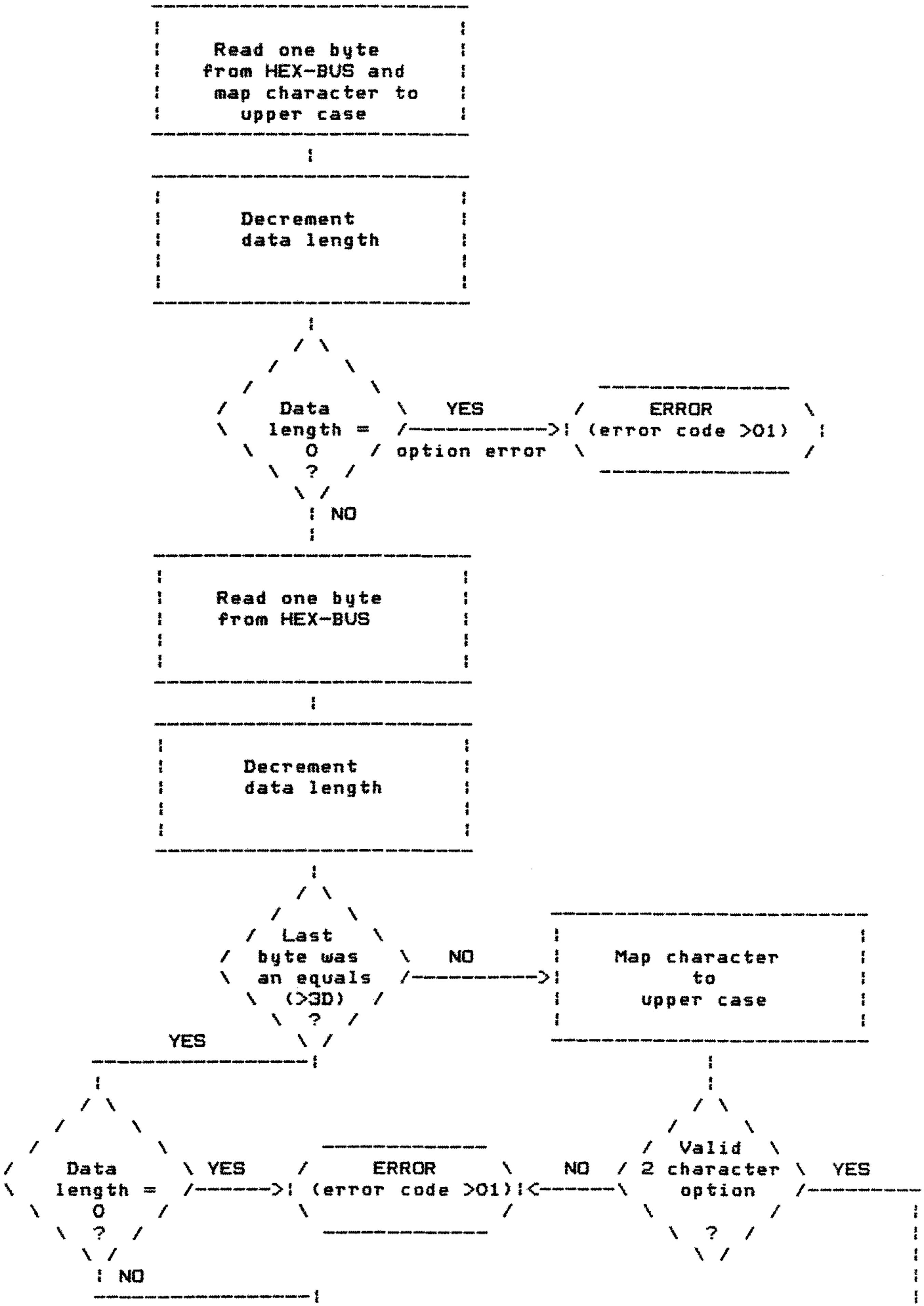
Turn on I/O bus LED (if there is one)
Read command code from HEX-BUS (one byte)
Read LUNO from HEX-BUS (one byte)
Read record number from HEX-BUS (2 bytes)
Read buffer length from HEX-BUS (2 bytes)
Read data length from HEX-BUS (2 bytes)
Illegal command code ?
YES
Unsupported command (error code = >0D)
ERROR
NO
Execute routine for specific command code
Command code = >FF ?
NO
YES
RESET command

```
          ---------------
        /                 \
       !     POWER UP      !
        \                 /
          ---------------
                 !
 -------------------------
 !                       !
 !  Turn on I/O bus LED  !
 !  for 0.5 sec if there !
 !  is one, otherwise    !
 !  0.5 sec delay        !
 -------------------------
                 !
          ---------------
        /                 \
       !     MAIN LOOP     !
        \                 /
          ---------------
```

```
           ________________
         /                  \
        !   RESET COMMAND    !
         \   (code >FF)     /
           ________________
                  !
   ____________________________
   !                          !
   !    Close peripheral      !
   !         and              !
   !     reinitialize         !
   !                          !
   ____________________________
                  !
           ________________
         /                  \
        !     MAIN LOOP      !
         \                  /
           ________________



           ________________
         /                  \
        !    NULL COMMAND    !
         \    (code >FE)    /
           ________________
                  !
           ________________
         /                  \
        !     MAIN LOOP      !
         \                  /
           ________________
```

```
                 --------------
               /                \
              !    OPEN COMMAND   !
               \    (code >00)   /
                 --------------
                       !
                      / \
                    /     \
                  /         \
                / Peripheral  \    YES            ---------------
                \   already   /---------->! /       ERROR          \
                  \  open   /               ! (error code >05)     !
                    \  ?  /                   \                    /
                      \ /                       ---------------
                       ! NO
                       !
                      / \
                    /     \
                  /         \                      ---------------
                /     Data    \    YES          /       ERROR        \
                \    length   /------------>! (error code >02)     !
                  \    <3    /attribute error\                     /
                    \  ?  /                        ---------------
                      \ /
                       ! NO
                       !
                      / \
                    /     \
                  /         \                    ---------------
                /    Buffer   \    YES        /       ERROR        \
                \    length   /---------->! (error code >0C)     !
                  \    <4   / buffer error \                     /
                    \  ?  /                      ---------------
                      \ /
                       ! NO
                       !
           --------------------------
           !                        !
           !      Read input        !
           !     buffer length      !
           !      from HEX-BUS      !
           !       (2 bytes)        !
           --------------------------
                       !
           --------------------------
           !                        !
           !    Read attribute      !
           !         byte           !
           !      from HEX-BUS      !
           !      (one byte)        !
           --------------------------
                       !
           --------------------------
           !                        !
           !      Subtract 3        !
           !         from           !
           !      data length       !
           !                        !
           --------------------------
                       !
```

```
                   / \
                  /   \
                 /     \
                /  attribute  \   NO      /       ERROR        \
                \    byte     /---------->| (error codes >02,  |
                 \ correct /               \    >11->17)       /
                  \   ?   /
                   \ /
                    | YES
                    |
                   / \
                  /   \
                 /  Input  \
                /   buffer    \  NO
                \   length =  /-----------
                 \    0     /             |
                  \   ?   /               |
                   \ /                    |
                    | YES                 |
                    |                     |
          ------------------------        |
          |                      |        |
          |   Set input buffer   |        |
          |  length to default   |        |
          |    buffer length     |        |
          |                      |        |
          ------------------------        |
                    |<--------------------
                    |
                   / \
                  /   \
                 /  Input  \
                /buffer length\  NO       /       ERROR        \
                \    large    /---------->| (error code >0C)   |
                 \ enough   / buffer error \                   /
                  \   ?   /
                   \ /
                    | YES
                    |
          ------------------------
          |                      |
          |   Set peripheral     |
          |      defaults        |
          |                      |
          |                      |
          ------------------------
                    |
   --------------   |
  /              \  |
  |      A       |---->|
  \              /  |
   --------------   |
                    |
                   / \
                  /   \
                 /     \          --------------
                /  Data   \  YES  /              \
                \  length =  /---------->|      B       |
                 \    0    /no more options\          /
                  \   ?   /               --------------
                   \ /
                    | NO
                    |
```


Read one byte from HEX-BUS and map character to upper case
Decrement data length
Data length = 0 ?
YES
option error
ERROR (error code >01)
NO
Read one byte from HEX-BUS
Decrement data length
Last byte was an equals (>3D) ?
NO
Map character to upper case
YES
Data length = 0 ?
YES
ERROR (error code >01)
NO
Valid 2 character option ?
YES
NO

```
 ---------------------------                                             |
 |    Read option data     |                                             |
 |       (one byte)        |                                             |
 |     from HEX-BUS and    |                                             |
 |    map to upper case    |                                             |
 |     if not a digit      |                                             |
 ---------------------------                                             |
              |                                                          |
 ---------------------------                                             |
 |                         |                                             |
 |        Decrement        |                                             |
 |       data length       |                                             |
 |                         |                                             |
 |                         |                                             |
 ---------------------------                                             |
              |                                                          |
             / \                                                         |
           /     \                                                       |
         /         \                  ----------------                   |
       /    Valid    \    NO        /       ERROR        \               |
       \    option   /------------>| (error code >01)     |              |
         \         / option error   \                    /               |
           \  ?  /                    ----------------                   |
             \ /                                                         |
              | YES                                                      |
              |                                                          |
             / \                                                         |
           /     \                                                       |
         /  Valid  \                  ----------------                   |
       /   data for  \  NO          /       ERROR        \               |
       \     that    /------------>| (error code >01)     |              |
         \  option / option error   \                    /               |
           \  ?  /                    ----------------                   |
             \ /                                                         |
              | YES                                                      |
              |<----------------------------------------------------------
 ---------------------------
 |                         |
 |          Save           |
 |       option data       |
 |                         |
 |                         |
 ---------------------------
              |
             / \
           /     \                        ---------------------------
         /         \                      |                         |
       /     Data    \   NO               |      Read one byte      |
       \   length =  /------------------->|       from HEX-BUS      |
         \     0   /                      |                         |
           \  ?  /                        |                         |
             \ /                          ---------------------------
              | YES                                    |
              |                           ---------------------------
  ---------------                         |                         |
 /               \                        |        Decrement        |
|       B         |----->|                |       data length       |
 \               /       |                |                         |
  ---------------        |                |                         |
                         |                ---------------------------
                         |                             |
                         |                             |
```

```
                                   !
                     ------------------------------
                     !                            !
                     !    Send data length        !
   Response          !   of 0004 on HEX-BUS       !
    message          !                            !
                     !        (2 bytes)           !
                     ------------------------------
                                   !
                                   !
                     ------------------------------
                     !                            !
                     !   Send input buffer        !
                     !   length on HEX-BUS        !
                     !                            !
                     !        (2 bytes)           !
                     ------------------------------
                                   !
                     ------------------------------
                     !                            !
                     !  Send record number        !
                     !  if applicable or          !
                     !  0000 on HEX-BUS           !
                     !       (2 bytes)            !
                     ------------------------------
                                   !
                     ------------------------------
                     !                            !
                     !      Send status           !
                     !     0 on HEX-BUS           !
                     !                            !
                     !       (one byte)           !
                     ------------------------------
                                   !
                     ------------------------------
                     !                            !
                     !     Set peripheral         !
                     !      open flag             !
                     !                            !
                     !                            !
                     ------------------------------
                                   !
                          ------------------
                        /                    \
                       !      MAIN LOOP       !
                        \                    /
                          ------------------
```

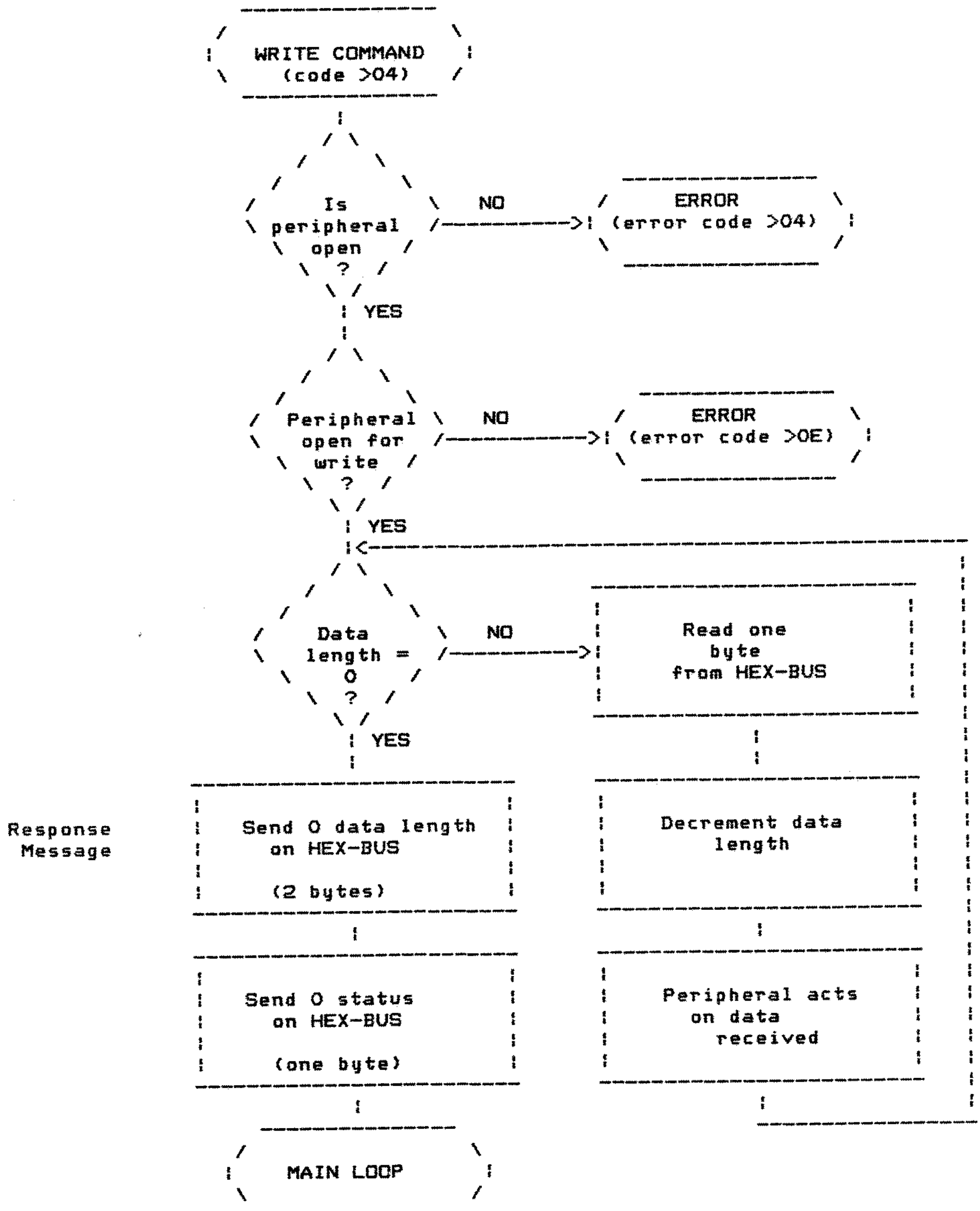
```
            / \
           /   \
          / Last  \                       --
         / byte was \ YES               /    \
         \  a comma  /----->!            !  A   !
          \  (>2C)  /                   \    /
           \  ?   /                       --
            \ /
             ! NO
            / \
           /   \
          / Last  \                       --
         / byte was \ YES               /    \
         \ a period  /----->!            !  A   !
          \  (>2E)  /                   \    /
           \  ?   /                       --
            \ /
             ! NO
      -----------------
    /      ERROR        \
   ! (error code >01)    !
    \                   /
      -----------------
```

```
                 ________________
                /                \
               |  CLOSE COMMAND   |
                \   (code >01)   /
                 ________________
                        |
                       / \
                      /   \
                     /     \
                    /  Is   \   NO         ________________
                    \ peripheral /------------>| (error code >04) |
                     \  open  /            \ ________________ /
                      \  ?  /
                       \ /
                        | YES
                       / \
                      /   \
                     /     \
                    /  Data  \   NO         ________________
                    \ length  /------------>| (error code >08) |
                     \  = 0  / data too long \ ________________ /
                      \  ?  /
                       \ /
                        | YES
                        |
               ______________________
              |                      |
              |   Complete any       |
              |  actions required    |
              |     by the           |
              |     peripheral       |
               ______________________
                        |
               ______________________
              |                      |
Response      |  Send data length    |
message       |    of 0000 on        |
              |      HEX-BUS         |
              |     (2 bytes)        |
               ______________________
                        |
               ______________________
              |                      |
              |     Send status      |
              |    0 on HEX-BUS      |
              |                      |
              |     (one byte)       |
               ______________________
                        |
               ______________________
              |                      |
              |  Reset peripheral    |
              |     open flag        |
              |                      |
              |                      |
               ______________________
                        |
                 ________________
                /                \
               |    MAIN LOOP     |
                \                /
                 ________________
```

Note: the two error boxes are labelled "ERROR" above the error code lines:

```
ERROR
(error code >04)

ERROR
(error code >08)
```

```
                    ---------------
                   /               \
                  !   READ COMMAND  !
                   \   (code >03)  /
                    ---------------
                          !
                         / \
                       /     \
                     /         \
                   /     Is      \    NO        /       ERROR        \
                   \ peripheral  /------------>! (error code >04)   !
                     \  open   /                \                    /
                       \  ?  /                   ------------------
                         \ /
                          ! YES
                         / \
                       /     \
                     /         \
                   /    Data     \    NO         /       ERROR        \
                   \   length    /------------->! (error code >08)   !
                     \   = 0   /  data too long  \                    /
                       \  ?  /                    ------------------
                         \ /
                          ! YES
                         / \
                       /     \
                     /         \
                   /  Peripheral \    NO         /       ERROR        \
                   \   open for  /------------->! (error code >0F)   !
                     \   read  /                 \                    /
                       \  ?  /                    ------------------
                         \ /
                          ! YES
                ------------------------
                !  Send data length    !
 Response       !  less than or equal  !
 message        !  to the buffer length!
                !    on the HEX-BUS    !
                !      (2 bytes)       !
                ------------------------
                          !
                ------------------------
                !                      !
                !  Send data received  !
                !    by peripheral     !
                !      on HEX-BUS      !
                ! (# bytes=data length)!
                ------------------------
                          !
                ------------------------
                !                      !
                !      Send status     !
                !    0 on HEX-BUS      !
                !                      !
                !      (one byte)      !
                ------------------------
                          !
                    ---------------
                   /               \
                  !    MAIN LOOP    !
                   \               /
                    ---------------
```


WRITE COMMAND (code >04)
Is peripheral open ?
NO
ERROR (error code >04)
YES
Peripheral open for write ?
NO
ERROR (error code >0E)
YES
Data length = 0 ?
NO
Read one byte from HEX-BUS
Decrement data length
Peripheral acts on data received
YES
Response Message
Send 0 data length on HEX-BUS (2 bytes)
Send 0 status on HEX-BUS (one byte)
MAIN LOOP

```
                  ______________
                 /              \
                |  RETURN STATUS |
                 \  (code >07)  /
                  ______________
                        |
                       / \
                      /   \
                     /     \                        ______________
                    /  Data \    NO               /     ERROR      \
                    \ length /---------------->| (error code >08) |
                     \ = 0  / data too long    \                  /
                      \ ?  /                     ______________
                       \ /
                        | YES
                       / \
                      /   \
                     /     \                        ______________
                    / Buffer \   YES              /     ERROR      \
                    \ length = /---------------->| (error code >0C)
                     \  0   /                    \                  /
                      \ ?  /                     ______________
                       \ /
                        | NO
                        |
              ________________________
             | Determine device       |
             | status based on end    |
             | of file,random access, |
             | protect,open,file      |
             | type,I/O modes         |
              ________________________
                        |
              ________________________
             |                        |
Response     |   Send data length     |
message      |      of 0001           |
             |   on the HEX-BUS       |
             |     (2 bytes)          |
              ________________________
                        |
              ________________________
             |                        |
             |  Send device status    |
             |   just determined      |
             |     on HEX-BUS         |
             |    (one byte)          |
              ________________________
                        |
              ________________________
             |                        |
             |     Send status        |
             |    0 on HEX-BUS        |
             |                        |
             |     (one byte)         |
              ________________________
                        |
                  ______________
                 /              \
                |   MAIN LOOP    |
                 \              /
                  ______________
```

```
                 ---------------
               / SERVICE REQUEST \
              !       POLL        !
               \    (code >0A)   /
                 ---------------
                        !
                       / \
                     / Is  \
                   / there a \                          ---------------
                 /   service   \   NO            /       ERROR         \
                 \   request   /----------->! (error code >0A)    !
                   \ pending /not requesting\                        /
                     \  ?  /                            ---------------
                       \ /
                        ! YES
                       / \
                     /     \
                   /         \                           ---------------
                 /    Data     \   NO             /       ERROR         \
                 \   length    /------------>! (error code >08)   !
                   \   = 0   / data too long \                      /
                     \  ?  /                             ---------------
                       \ /
                        ! YES
                        !
           ------------------------
           !                      !
           !    Action taken      !
           !  depends upon reason !
           !    for requesting    !
           !       service        !
           ------------------------
                        !
           ------------------------
Response   !  Send data length    !
 message   !  less than or equal  !
           !  to buffer length    !
           !    on the HEX-BUS    !
           !      (2 bytes)       !
           ------------------------
                        !
           ------------------------
           !                      !
           !      Send data       !
           !     on HEX-BUS       !
           !                      !
           !(# bytes = data length)!
           ------------------------
                        !
           ------------------------
           !                      !
           !     Send status      !
           !    0 on HEX-BUS      !
           !                      !
           !      (one byte)      !
           ------------------------
                        !
                 ---------------
               /                 \
              !     MAIN LOOP     !
               \                 /
                 ---------------
```

HEX BUS
DISPLAY INTERFACE FUNCTIONAL SPECIFICATION

Texas Instruments Incorporated
Consumer Products Group
7/22/83
Revision 2.2

SECTION 1

Introduction

1.1 Purpose

This document describes a base set of functional aspects for all future displays that are either resident in a compact computer system, or interface to the TI HEX-BUS(TM). In addition, a section is included that overviews the peripheral's interface with the HEX-BUS. A final section overviews some optional enhancements to the base functional specification.

SECTION 2

Product Overview

2.1 Concept

All display devices incorporated in any way in a compact computer system should be interchangeable among themselves. That is, software that uses the display should be able to interchange displays without great difficulty. To allow this, all display devices will be accessed as peripheral devices (even those built in to consoles). Software to interface to a 'general' display device would be responsible for allowing for varying display sizes (the display size can be determined by the return status command).

SECTION 3

Operation

3.1 Introduction

The display is cleared and initialized to scrolling mode at power up, reset, and open. The blinking block (reverse video) cursor will appear in the upper left corner of the display. Power up, reset, and open also enable line wrap, unprotect the top and bottom lines, and initialize the cursor to position (0,0) on page 0. While the display remains in scrolling mode, line feeds received while the cursor is at the bottom of the display will result in the contents of the display being shifted up by one line, with the old topmost unprotected line being lost. The bottom line will then be cleared. (Note that if the bottom line is protected, the same thing results when the cursor is on the display line second from the bottom). The reverse occurs when a line up is received when the cursor is at the top of the display and the display is in scroll mode.

It displays characters corresponding to the ASCII formatted input data. It also accepts nine control codes. The character set and control codes are shown in Figure 1. Control codes not defined are ignored.

lsb\msb	0	1	2	3	4	5	6	7
0			sp	0	@	P	`	p
1			!	1	A	Q	a	q
2			"	2	B	R	b	r
3			#	3	C	S	c	s
4			$	4	D	T	d	t
5		TEST	%	5	E	U	e	u
6			&	6	F	V	f	v
7	BELL		'	7	G	W	g	w
8	BS		(	8	H	X	h	x
9			)	9	I	Y	i	y
A	LF		*	:	J	Z	j	z
B	LU	ESC	+	;	K	[	k	{
C	FS		,	<	L	\	l	¦
D	CR		-	=	M	]	m	}
E			.	>	N	^	n	~
F		EP	/	?	O	_	o	

Figure 1:CHARACTER SET

Control codes:

Bell (BELL): Sounds an audible tone.
If the device is capable of producing an audible tone, it produces such a tone upon the receipt of the BELL code within a write command.

Backspace (BS): Moves cursor backward one character space.
If the cursor is at the first unprotected display location on the page and line wrapping is enabled, it will always wrap to the last unprotected display location on the page. If the cursor is in column 0 on any other line and line wrapping is enabled, the cursor is positioned to the last column of the previous line when a backspace is received.

Line Feed (LF): Moves the cursor down one line.
If the cursor is on the last unprotected line of the page, the display will scroll if in scroll mode, inserting a blank line at the bottom of the page and losing the topmost unprotected line. The cursor will wrap to the topmost unprotected line if in page mode. In both cases the column position of the cursor will remain unchanged.

Line Up (LU): Moves the cursor up one line.
If the cursor is on the topmost unprotected line of the page, it will wrap to the bottommost unprotected line if in page mode, or will cause a reverse scroll to occur if in scroll mode, losing the contents of the bottommost unprotected line and inserting a blank line at the top. In both cases the column position of the cursor will remain unchanged.

Forward space (FS): Moves the cursor one position to the right.
If the cursor is at the last unprotected display location on the page and line wrapping is enabled, it will always wrap to the first unprotected display location on the page. If the cursor is in the last column of any other line and line wrapping is enabled, the cursor is positioned to column 0 of the next line when a forward space is received.

Carriage Return (CR): Returns cursor to column zero of the current line.

TEST: Runs through display self test.

Escape (ESC): Prefaces a display control sequence.

Erase Page (EP): Homes the cursor and erases to end of page.
If the top line is protected, the cursor homes to the start of the second line. If the bottom line is protected, the

display is cleared only to the end of the line second from the bottom.

Note that the cursor is not allowed to be on protected lines. Thus, when the top line is protected, a command that would normally put the cursor on that line will put the cursor in the same column position on the second line. In addition, if the bottom line is protected, commands (such as erase to end of page) that would normally affect it will not.

The display will automatically generate its own Carriage Return/Linefeed to segment a data record of displayable characters that exceeds the display line length (except when line wrap is disabled - see Disable/Enable Line Wrap commands).

3.2 Display Control Sequences

The display has several escape sequences that can be sent that allow it to perform various operations, including line editing functions and mode control.

Note that these sequences may be split across record transmissions. In other words, part of a command may be sent in one record, and the rest of the command in the next record. Unrecognized escape sequences are ignored - i.e. <ESC>* would be ignored.

3.2.1 <ESC>M - Scroll Mode Select (default).

The <ESC>M character sequence sets the display to scroll mode. In scroll mode the peripheral will scroll all unprotected text on the page up one line whenever a Line Feed is received when the cursor is at the bottom of the page, inserting a blank line and losing the top unprotected line. All unprotected text on the page will be scrolled down one line whenever a Line Up is received and the cursor is at the top of the page, inserting a blank line and losing the bottom unprotected line.

3.2.2 <ESC>m - Page Mode Select.

The <ESC>m character sequence sets the display to page mode. When the cursor is on the bottom unprotected line of the page, a line feed control code moves the cursor to the same column position on the top unprotected line. When the cursor is on the top unprotected line of the page, a line up moves the cursor to the bottom unprotected line.

3.2.2.1 <ESC>Q - Character Insert.

<ESC>Q moves the character at the cursor position and all following characters to the right by one position. The character at the end of the line will be lost.

3.2.3 <ESC>W - Character Delete.

This command deletes the character at the current cursor position, and moves all following characters on that line to the left by one position. A blank will be moved into the rightmost character position on the line.

3.2.4 <ESC>E - Line Insert.

The line insert command inserts a line of blanks at the line where the cursor is positioned. The line where the cursor was located and all following unprotected lines will move one line downwards. The line at the bottom of the page will be lost (Note that the cursor will remain where it is relative to its screen location).

3.2.5 <ESC>R - Line Delete.

<ESC>R deletes the line where the cursor is located. All of the following unprotected lines will be moved one line upwards, and a line of blanks will be placed at the bottom of the page. The cursor will move to the start of the line.

3.2.6 <ESC>T - Erase Through End of Line.

This command erases all character positions starting with the cursor to the end of the line.

3.2.7 <ESC>Y - Erase Through End of Page.

This command erases all character positions starting with the cursor through the end of the page (if the last line is protected, it is left as is).

3.2.8 <ESC>=<row><col> - Set Cursor Position.

The <ESC>=<row><col> sequence sets the cursor position. The <row> character specifies the line (starting with line 0) and the <col> specifies the column (starting with column 0) for the new cursor position. (coordinates are in binary).

3.2.9 <ESC>0 - Protect Top Line.

<ESC>0 sets the status of the first line of the page to protected. When protected, no commands except the unprotect top line command have any effect on that line. If the display page is changed, the top line of the new page will be the new protected top line.

3.2.10 <ESC>1 - Unprotect Top Line (default).

<ESC>1 sets the status of the top line of the page to unprotected.

3.2.11 <ESC>2 - Protect Bottom Line.

The protect bottom line command sets the status of the bottom line of the page to protected. When protected, no commands except unprotect bottom line will have any effect on that line. If the display page is changed, the bottom line of the new page will be the new protected bottom line.

3.2.12 <ESC>3 - Unprotect Bottom Line (default).

The unprotect bottom line command sets the status of the bottom line of the page to unprotected.

3.2.13 <ESC>4 - Disable Line Wrap.

If the cursor is in column zero, a back space will be ignored. If the cursor is in the last column (end of the line), a forward space will be ignored; or, if a printing character is received, it will replace the character at the cursor position.

3.2.14 <ESC>5 - Enable Line Wrap (default).

The <ESC>5 sequence sets the display to allow line wrapping. If the cursor is in column zero, the back space moves the cursor to the end of the previous line unless the cursor is in column zero on the first unprotected line of the page. In this case, back space moves the cursor to the end of the bottom unprotected line of the page. If the cursor is at the end of a line, a forward space control code or any character moves the cursor to column zero on the next line unless the cursor is at the end of the bottom unprotected line of the page. In this case the cursor will move to column zero in the top unprotected line of the page.

3.2.15 <ESC>6<chr1>...<chrN> - Shift Page Right and Load Column.

This command shifts all unprotected text on the page right one column and loads the next 'N' characters into column zero. 'N' is the number of unprotected lines on the current page. The rightmost column will be lost. After the completion of this command, the cursor is left at its original position.

3.2.16 <ESC>7<chr1>...<chrN> - Shift Page Left and Load Column.

This command shifts all unprotected text on the page left one column and loads the next 'N' characters into the last column on the page. 'N' is the number of unprotected lines on the current page. Column zero will be lost. After the completion of this command, the cursor is left at its original position.

3.2.17 <ESC>9 - Read Cursor Address.

The <ESC>9 command causes the next read data command to read back 2 bytes of binary data - the cursor's row coordinate (starting with line 0), and the cursor's column coordinate (starting with column 0). When multiple pages are supported by the display peripheral, and the buffer size allows, the read data command will send back one additional byte of binary data - the number of the current display page (starting with page 0).

3.2.18 <ESC>.<cmd> - Changing Cursor Attributes.

The cursor attributes commands change how the cursor is displayed on the screen. If a particular cursor is not supported, the closest cursor that is supported should be used.

* <ESC>.0 - Don't Display Cursor
* <ESC>.1 - Blinking Reverse Video Cursor (default)
* <ESC>.2 - Solid Reverse Video Cursor
* <ESC>.3 - Blinking Underline Cursor
* <ESC>.4 - Solid Underline Cursor

3.2.19 <ESC>G<cmd> - Text Attributes.

3.2.19.1 <ESC>G0 - Forward Video (default).

This command causes subsequent characters sent to the display to be displayed as foreground colored text (white if B&W) on a background colored screen (black if B&W).

3.2.19.2 <ESC>G4 - Reverse Video.

This command causes all subsequent characters sent to the display to be displayed as background colored text (black if B&W) on a foreground colored screen (white if B&W). This also includes any blanks generated by insert, delete, shift page, erase page, erase line, etc.

SECTION 4

Bus Interface

4.1 Standard Access

The access to a display is performed with a sequence of I/O calls. Before using the display it should be opened using the open command. This may be followed by other I/O calls to read or write data or perform other functions. When the computer finishes using the peripheral it will issue a close command code (>01). This will ensure that any necessary device dependent actions have been performed. The bus reset command code (>FF) will also close the display if it is open.

4.2 Command Descriptions

This section describes the Command message setup and device response for the various standard I/O command codes.

4.2.1 Open - 00.

This command code is used to initiate the use of a peripheral. The display will check access modes and ensure that it is not already open. If it is open or any other error occurs while the open is being attempted, then the appropriate error will be issued. Otherwise, the peripheral will be opened and reset to its initialized (power up) state. The command message will be set up as follows:

field	data
Device code	as required
Command Code	00
Logical Unit Number	don't care
Record number	don't care
Buffer length	as required (at least 0004)
Data length	as required (0004 returned)

The data buffer contains the following which is sent to the peripheral.

Input buffer length (2 bytes)
(if zero then device returns buffer size)
device attributes (1 byte)
device options (if any)

The peripheral will compare the 'Input buffer length' to its capabilities and return either the requested length, the default length if the requested length is zero, or an error if the requested length is unacceptable. The input buffer length is used by the bus master to determine what size buffer should be allocated by the master for read operations. Thus software could open a device such as an RS-232 peripheral with a one byte input buffer, and write 80 character records out to the RS-232. Each peripheral is required to check the length of data being transferred to it in each write operation, and to generate an error if that data is longer than the maximum write acceptable to the peripheral. The default buffer size for the display is 80 bytes.

The device attributes byte contains flags used to indicate the access mode of the peripheral. Several bits are unused by the I/O scheme and may be used by the application software as desired.

The bit definitions for the attributes byte are as follows (bit 0 is the least significant bit):

```
7-6 - access mode
      00 - append mode (not allowed)
      10 - output mode (write only)
      01 - input mode  (read only)
      11 - update mode (read or write)
  5 - relative(1)/sequential(0) (always zero for display)
  4 - fixed(1)/variable(0) (display should ignore)
  3 - internal(1)/display(0) file type  (always 0 for display)
  2 - device dependent use
1-0 - unused (may be used as desired by application)
```

The access mode must match the capabilities of the particular device:

1. Append mode specifies that data will only be written (or appended) to the end of a file (Note that the display does not allow append mode).

2. Output mode specifies that data will only be written to the device and the "read data" command will not be used.

3. Input mode specifies that data will only be read from the device and the "write data" command will not be used.

4. Update mode means that data may be both read and written.

The device options field is a variable length field which contains device-dependent information relative to setting up the device. For the display, this field could either not be included, or consist of the following options:

Carriage Return Options:

R=C (and .LF)	This option indicates that the display should append a carriage return to the end of each record written.
R=L	(Default if option not included). This option indicates that the display should append a carriage return and line feed to the end of each record written.
R=N (and .CR)	This option indicates that no carriage return or linefeed is to be appended to the end of each record written.

The response buffer will contain the accepted buffer length, and the record number that the file was opened to. For the display, this record number is meaningless, so a zero (0) will be returned. This information is always returned. Thus the response message for a successful open will be:

Data length		4
Data	(2 bytes)	Accepted buffer length (LSB first)
	(2 bytes)	always >0000
Operation status		0

An unsuccessful open may not return data in the response message. The operation status byte may contain the following error codes (decimal):

- 00 - successful open
- 01 - device/file options error
  - * Open options not correct
- 02 - Invalid attributes
  - * Open mode not sequential
  - * Attributes byte (possibly including I/O buffer length) not included in data
- 05 - file/device already open
- 12 - buffer size error (Given if the buffer size in the PAB < 0004)
- 19 - append mode not supported
- 23 - internal file type not supported

4.2.2 Close - 01.

This command terminates the use of a device. Depending on the device this command may be used to clean up internal data (e.g. write an end of file) or may be effectively ignored. In general a close command must be sent between using a device and another open command. When the display is closed, the current data will remain on the screen. The data length for the close command will be zero (no data buffer is transmitted). The PAB should be set up as follows:

field	data
Device code	as required
Command Code	01
Logical Unit Number	don't care
Record number	don't care
Buffer length	don't care
Data length	0000 (0000 returned)

The response message will only contain a status byte and a zero data length (two bytes). The error status indications are (decimal):

00 - device or file closed
04 - device or file never opened
08 - data too long error
  * data sent with close command

4.2.3 Read Data - 03.

This command is used to read the current line. It transmits all characters starting with the current cursor position through the end of the line. The PAB should be set up as follows:

field	data
Device code	as required
Command Code	03
Logical Unit Number	don't care
Record number	don't care
Buffer length	as required
Data length	0000 (record length returned)

The response message will contain the requested data and the operation status. The following error status indications may occur (decimal):

00 - read successfully completed
04 - file/device not open
06 - Device error
15 - not open for read

1. If an <ESC>9 was sent prior to issuing the read data command, the read data command will send back the cursor coordinates as described in the Read Cursor Address command description.

   In addition, if the display peripheral supports multiple screen pages, and the buffer size allows, the read data command will also send back the current screen page.

2. When reading to the end of a line, all character codes <u>must</u> be returned the same as they are written - i.e. character graphic codes and user defined characters must be returned as the same code as when they were written out, even if they were mapped to a different range of the character set.

3. When in graphics mode the read data command always returns the coordinates of the cursor (relative to the origin). This will consist of 4 bytes of data - 2 for the line coordinate and 2 for the column coordinate.

4.2.4 Write Data - 04.

This command is used to send data to a peripheral device. The command message will contain the data to be sent to the device. The PAB should be set up as follows:

field	data
Device code	as required
Command Code	04
Logical Unit Number	don't care
Record number	don't care
Buffer length	don't care
Data length	record length (0000 returned)

The response message will contain zero-length data and the operation status. The following error status indications may occur (decimal):

00 - write successfully completed
04 - file/device not open
06 - Device error
14 - not open for write
80 - Syntax error in data
81 - Coordinates out of range

4.2.5 Return Status - 07.

This command is used to return device status information. The information is returned in the data buffer. Certain bit fields in the return data are assigned to standard meanings while others are reserved for device dependent extensions. Certain devices may return more bytes of status if the buffer length allows - if the buffer length is five bytes or greater, the display will return the display attributes. The PAB should be set up as follows:

field	data
Device code	as required
Command Code	07
Logical Unit Number	don't care
Record number	don't care
Buffer length	>= 0001
Data length	0000 (at least 0001 returned)

The bit fields in the returned data are as follows (bit 0 is the least significant bit).

```
 First byte:  >13 if the peripheral is open
              >03 if the peripheral is closed
Second byte:  number of lines in the display
 Third byte:  number of columns in the display *
Fourth byte:  flag byte one
     bits:   7 - bit mapped graphics supported
             6 - character graphics supported
             5 - extended text attributes supported
             4 - multiple grey levels/colors supported
             3 - user defined characters supported
             2 - window on an 80 character line supported
           1:0 - undefined (should be zero)
```

* - the display peripheral will either return the actual screen width, or, if it has been configured as a window on a larger line, the actual line length will be returned for the number of columns.

Return Status (continued):

```
     Fifth byte:  flag byte two
        bits: 7:6 - 00: page command not supported
                    01: 2 screen pages supported
                    10: 3 screen pages supported
                    11: 4 screen pages supported
              5:4 - undefined (should be zero)
              3:0 - reserved for device dependent use
                    (ex. total # of pages if more than 4)

     Sixth byte:  LSB of number of horizontal pixels *
   Seventh byte:  MSB of number of horizontal pixels *
    Eighth byte:  LSB of number of vertical pixels *
     Ninth byte:  MSB of number of vertical pixels *

  * - only when bit mapped graphics are supported and the
      buffer size allows the transmission of the extra
      data.
```

A bit being set to one in the flag bytes indicates that the corresponding option is supported. The following error status indications may occur (decimal):

00 - status returned
12 - buffer size error

4.2.6 Service Request Poll - OA.

This command allows a bus master to query a peripheral as to whether it requested service from the master. The PAB should be set up as follows:

field	data
Device code	as required
Command code	OA
Logical Unit Number	don't care
Record number	don't care
Buffer length	as determined in open
Data length	0000 (0000 returned)
Return status	OA

The response message will consist of zero bytes of data, and the following return status for a display peripheral.

OA - it wasn't me (unsuccessful poll)

* Any other error code indicates a successful poll operation, and reflects the reason for the service request. Thus even devices that don't support service requests need to return the "it wasn't me" status code.

4.2.7 Null Operation - FE.

When the master computer receives a BAV interrupt and either no devices are enabled for interrupts or the current service flag is set, then a null operation code is sent to all devices. There will be no response to this message.

field	data
Device code	as required
Command code	FE
Logical Unit Number	don't care
Record number	don't care
Buffer length	don't care
Data length	00

4.2.8 Reset Bus - FF.

It may sometimes be desired to tell a device (or devices) to close all open files (or devices). This command will have no response by devices, they will simply perform the action requested (If they were not open to begin with, then they will do nothing). The display reverts to a 'power up' status. The PAB should be set up as follows:

field	data
Device code	as required
Command code	FF
Logical Unit Number	don't care
Record number	don't care
Buffer length	don't care
Data length	00

* For both the Null operation and Reset bus commands, no response will be forthcoming.

SECTION 5

Enhancements to the Base Definition

5.1 Introduction

This section overviews a few pre-defined enhancements to the base functional specification of the video peripheral. It is intended to provide a few optional features that may be implemented in a standard format, and is not intended to limit the features provided in a video peripheral (i.e. some video peripherals may provide more (or other) features than those documented herein).

5.2 Vector Graphics

In this instance, the display would have two modes - text mode and graphic mode. In text mode, it would mimic a video terminal. When in graphic mode, the peripheral would accept certain graphic commands to enable 'drawing' on the display.

5.3 Text Mode

The display is initialized to both text and scrolling modes at power up and reset.

It displays characters corresponding to the ASCII formatted input data. It also accepts extra control codes TEXT (Hexadecimal 11), and GRAPH (Hexadecimal 13).

When in text mode the following restriction applies - All control codes from >00 to >1F not defined will be ignored.

Control codes:

Text:	Selects Text Mode
Graph:	Selects Graphic Mode

The Text control code is the only control code valid in the Graphic Mode (The graph control code will be ignored). User defined characters are not supported while in graphic mode (although the definitions will be maintained).

5.4 Graphic Mode

In the Graphic Mode input data is processed as graphic commands. When graphic mode is entered, the screen will be cleared, and the cursor homed to the upper left corner of the screen. Graphic commands are:

Define Origin : O
Set current location as origin.

Draw Line : L(x1,y1)[:[i][:p]],(x2,y2)[:[i][:p]],
...(xn,yn)[:[i][:p]]
Draws lines between coordinates (x1,y1), (x2,y2), through (xn,yn). Each point coordinate after the first represents the endpoint of the next line segment to be drawn, just as present position is assumed to be the beginning point for that line segment.

p specifies the line type; p=0 specifies a solid line with p=1-9 specifying a dotted line with increasingly greater pitch.

i specifies the intensity level (i=-1 turns subsequent pixels off, i=0 indicates low intensity, and i=1 indicates high intensity). If multiple intensities are not supported, this field is ignored. For color displays, the intensity field indicates the color desired (as per the home computer extended basic COLOR subprogram).

```
                y
                +
        ------- ¦         +
        ¦       ¦
        ¦       ¦
        ¦  ---  ¦                   +
        ¦   ¦   ¦
        y1  ¦   ¦
        ¦   y2  ¦
        ¦   ¦   ¦
        ¦   ¦   ¦
      ----------*-----------------------------x
 * - origin     ¦------------x2--------¦
                ¦----x1----¦
```

L(x1,y1),(x2,y2)

Note: Point coordinates may be specified to +/- 999. Internally, they are maintained to +/- 9999.

Draw Relative Line : R(x1,y1)[:[i][:p]],(x2,y2)[:[i][:p]],
...(xn,yn)[:[i][:p]]

Same as Line Command except all point coordinates are taken relative to the present cursor position rather than the origin.

```
                    y
                    :
          ------    :            +
          :    :    :
          :   y2    :
          :    :    :
          y1   :    :
          :   ---   :                                  +
          :         :
          --------*-------------------------------------x
* - current         :
    position        :------x1----:---------x2----------:
```

R(x1,y1),(x2,y2)

Move Cursor : M(x,y)

The cursor is positioned to coordinate x,y.

Relative Move Cursor : J(x,y)

The cursor is positioned to coordinate x,y relative to the current cursor position.

Scale Character : Sn

Specifies character size n=0-9. The scale is initialized to 1 by power up or reset. The value set by Sn is retained when the mode is shifted from graphic mode to text mode and then back to graphic mode.

Home : H

Homes the cursor to the upper left corner of the screen.

Angle : An

Specifies the angle at which to display characters. n=0 is an upright character, n=1 rotates the character 90 deg clockwise, n=2 rotates the character 180 deg clockwise and n=3 rotates the character 270 deg clockwise.

Text Display : T(c1 c2 c3 .... cn)

Displays the text enclosed in parenthesis starting at the current cursor position. If the end of the line is reached while printing the text, remaining text will be lost. Note that parenthesis may be embedded within the text.

PROGRAM EXAMPLES

Program

```
10 PRINT #1, CHR$(19)   ! Enter graphics
20 PRINT #1,"M(16,-4)"  ! Position cursor
30 PRINT #1,"O"         ! Set origin
40 PRINT #1,"L(6,12):1:3,(30,12)"
```

Characters Printed

```
y       y'
:       :
:       :   - - - - - - -
:       :
:       :
:       :
+-------:---------------x
O       :
        +--------------------x'
        O'
```

In this example statement 10 selects Graphics Mode. Statement 20 moves the cursor to coordinates (16,-4) without drawing a line. Statement 30 redefines the origin as (16,-4). Statement 40 draws a dotted line from new coordinates (6,12) to (30,12).

Delete statements 20 and 30. Change statement 40 to read

```
40 PRINT #1,"L(22,8):0:3,(46,8)"
```

This will draw the same line in low intensity.

Re-insert statement 20. Change statement 40 to read

```
40 PRINT #1,"R(6,12):0:3,(24,0)"
```

This will again draw the same line in low intensity.

Add lines 50 and 60 -

```
50 PRINT #1,"A2"
60 PRINT #1,"T(ABC)"
```

This will print the characters ABC upside down to the left of where the line had terminated.

5.5 Displaying Characters in Graphic Mode

Each character is displayed within an area on the screen called a character cell. Within this cell is the actual character as well as spacing to separate the character from the next. All character sizes are obtained by getting the integer multiple of the size of the smallest character as described below:

Scale Size	0	1	2	3	4	5	6	7	8	9
# Chrs/line *	80	40	20	10	8	6	5	4	4	3
Size ratio	1	2	3	4	6	7	8	9	10	11

* - If displayed horizontally across the screen

Note that the above chart only provides an approximation of the relative sizes of characters and number of characters obtainable per line when using an 80 column (standard) display.

In graphic mode, the starting cursor position for that character defines the lower left corner of the character cell. From that point (for all four display angles) the character is drawn. Each successive scale size is an integer multiple of the number of rows and columns used in the above cell

When the display angle is rotated, what actually happens is that the character cell is rotated to the desired angle (0, 90, 180, 270 degrees). Characters are then displayed in the direction indicated by that angle.

```
      A3  /|\
           | +-+
        +---| | ---> A0  (Characters are displayed in the
        +-+-*-+-+          direction indicated by the arrows)
   A2 <--- | |---+
          +-+ |   A1
             \|/        * - Starting cursor position
```

5.6 Sprites

5.7 Character Graphics

The 32 characters (character codes hexadecimal 80 - hexadecimal 9F) shown in appendix A are the characters defined for use when character graphics are supported. These characters are useful for box drawing and may also be used to create simple bar charts. Note that the character cell size used in the appendix is 6 X 8. This cell size will change to correspond to the cell size of the particular display. If a 7 X 10 cell size is desired, the Model 911 VDT Graphics Character Set should be used.

5.8 Extended Text Attributes

The below commands allow for various text highlighting abilities often desired on displays for word processing. Any of the below commands that are not supported will be ignored. The <ESC>GO command will return the display to normal (white on black).

5.8.1 <ESC>G) - Half Intensity On.

Half intensity on causes subsequent characters sent to the display to be displayed in half intensity (or different color in color display).

5.8.2 <ESC>G(- Half Intensity Off.

Half intensity off causes subsequent characters sent to the display to be displayed in normal intensity (or normal color in color display).

5.8.3 <ESC>G8 - Underline.

This sequence causes subsequent characters to be underlined.

5.8.4 <ESC>G< - Underline Reverse.

Underline reverse displays underlined characters using reverse video (black characters on a white background - or per color display).

5.8.5 <ESC>G2 - Blinking.

The blinking command causes subsequent characters to blink.

5.8.6 <ESC>G: - Blinking Underlined.

<ESC>G: causes subsequent characters to be underlined and to blink.

5.8.7 <ESC>G>- Blinking Underlined Reverse.

This command causes subsequent characters to be underlined, in reverse video, and to blink.

5.9 Special optional commands

5.9.1 <ESC>8<chr><row1><row2>...<rowN> - Define Character.

The define character command takes the next N (usually 8) bytes and loads the character definition corresponding to <chr>. <chr> is the 8-bit code of the character to be defined. If that character is not definable, the command is ignored.

5.9.2 <ESC>C<color> - Define Foreground and Background Colors.

Different foreground and background colors may be desired when using a color monitor. <ESC>C<color> tells the peripheral to use the <color> byte it receives to define the foreground (first nibble) and background (second nibble) colors.

Nibble (Hex)	Color
0	Transparent
1	Black
2	Medium Green
3	Light Green
4	Dark Blue
5	Light Blue
6	Dark Red
7	Cyan
8	Medium Red
9	Light Red
A	Dark Yellow
B	Light Yellow
C	Dark Green
D	Magenta
E	Gray
F	White

HX-1100 Video Interface Color Table

5.9.3 <ESC>K<chr1>...<chrN> - Shift Partial Page Right and Load Column.

This command shifts all unprotected text on the page right one column beginning with the column the cursor is in and loads the next 'N' characters into the newly created column. 'N' is the number of unprotected lines on the current page. The rightmost column will be lost and the cursor will return to the same relative position after the new column is loaded.

5.9.4 <ESC>L<chr1>...<chrN> - Shift Partial Page Left and Load Column.

This command shifts all unprotected text on the page left one column beginning with the column the cursor is in and loads the next 'N' characters into the newly created column. 'N' is the number of unprotected lines on the current page. Column zero will be lost and the cursor will return to the same relative position after the new column is loaded.

5.10 Page Control Commands

5.10.1 <ESC>P<page> - Change Display Page.

This command switches to the display page number indicated by <page> (in binary). This command changes nothing but the page - all previous modes and options and the cursor address remain unaffected. Bits 7 and 6 in the fifth byte of the returned status indicate whether this command is supported and how many pages of display memory are available if it is supported. Note that if there are protected lines on the display, and the display page is changed, those lines will still be protected on the new page. Pages are numbered beginning with 0.

5.11 Window on a Line Longer Than the Screen Width

Some video peripherals will support a mode that allows the screen to be a window on a longer line (such as 80 characters). These peripherals will automatically move the window (if necessary) at the end of each write command in order that the window position indicated by the cursor position be displayed on the screen.

```
0       5               44             79
+-------+---------------+---------------+
|       |               |               |0
| off   |    window     | off           |
| screen|               | screen        |
|       |       *       |               |   24 lines
|       |    (cursor)   |               |
|       |               |               |
|       |               |               |
|       |<-- 40 chrs -->|               |23
+-------+---------------+---------------+
```

Example of Display Window on an 80 Character Line

Figure 5-0

5.11.1 Moving the Window.

1. Whenever the cursor is at the end of a line as displayed on the screen, and text exists off the screen to the right of the cursor, and either a character is printed to the display or a forward space code is received, the entire window will move to the right one column. The cursor will remain at the end of the displayed line (i.e. the cursor position was actually incremented, but as the window moved to the right one column, the apparent cursor position relative to the screen is unchanged).

2. Whenever the cursor is at the beginning of a line as displayed on the screen, and text exists off the screen to the left of the cursor, and a backspace code is received, the entire window will move to the left one column. The cursor will remain at the beginning of the displayed line (i.e. the cursor position was actually decremented, but as the window moved to the left one column, the apparent cursor position relative to the

screen is unchanged).

3. Whenever the cursor is at the beginning or end of the actual line, then the display will perform whatever wrap or scrolling is called for.

4. Whenever it has been determined (at the end of a write command) that the new cursor position is not in the current window, the window will be repositioned so that, if possible, the cursor position will not change relative to the screen (i.e. If the cursor is at the 3rd row, 15th column of the display, and a move cursor position command is issued that causes the cursor to move off the screen, if it is possible to move the window and maintain the cursor at the 3rd row, 15th column, it will be done. If not, if the new cursor position is to the left, the cursor will be placed in column zero of the display, and if the new position is to the right, the cursor will be placed in the rightmost column of the display).

5.11.2 Commands for Changing the Window Configuration.

5.11.2.1 <ESC>D - Use Physical Display Size.

The <ESC>D command causes the display to use its physical display size for displaying data.

5.11.2.2 <ESC>d - Use Window on 80 Character Line.

The <ESC>d command causes the display to have its screen act as a window on an 80 character line.

5.11.2.3 <ESC>S<rows><cols> - Define Page Dimensions.

<ESC>S tells the peripheral to use the next 2 bytes it receives to define the page dimensions. <rows> (in binary) sets the number of lines per page and the second byte <cols> sets the number of columns per page.

Note: The internal display memory is treated as one continuous block of data with tags where each new page begins. Changing page dimensions changes the location of the tags, without inserting or deleting any characters, possibly causing the contents to appear scrambled.

SECTION 6

Escape Sequence Summary

Standard escape sequences:

<ESC>M - Scroll mode select (default)
<ESC>m - Page mode select
<ESC>Q - Character insert
<ESC>W - Character delete
<ESC>E - Line insert
<ESC>R - Line delete
<ESC>T - Erase through end of line
<ESC>Y - Erase through end of page
<ESC>= - Set cursor position
<ESC>0 - Protect top line of page
<ESC>1 - Unprotect top line of page (default)
<ESC>2 - Protect bottom line of page
<ESC>3 - Unprotect bottom line of page (default)
<ESC>4 - Disable line wrap
<ESC>5 - Enable line wrap (default)
<ESC>6 - Shift page right and load column
<ESC>7 - Shift page left and load column
<ESC>9 - Read cursor address
<ESC>. - Change cursor attributes
<ESC>G - Text attributes

Optional escape sequences:

<ESC>G - Extended text attributes
<ESC>8 - Define character
<ESC>C - Define foreground and background colors
<ESC>K - Shift partial page right and load column
<ESC>L - Shift partial page left and load column
<ESC>P - Change display page
<ESC>D - Use physical display size
<ESC>d - Use window on 80 character line
<ESC>S - Define page dimensions

SECTION 7

Extended Open Options

7.1 Introduction

This section overviews a few pre-defined enhancements to the options in the open statement. It is intended to provide a few optional features that may be implemented in a standard format, and is not intended to limit the features provided in a video peripheral (i.e. some video peripherals may provide more (or other) features than those documented herein).

7.2 .NI - No Initialization

The option .NI in the open statement causes the peripheral to save all previously defined modes, settings, and data instead of initializing to the power up state as is usually done during an open command.

7.3 .80 - Window on an 80 Character Line

The option .80 when included in the open statement causes the peripheral to format to a 24 X 80 page dimension so that the actual screen acts as a window on the page.

APPENDIX A

Character codes >80 - >9F in a 6 X 8 Cell

```
 ------     ------     ------     ------     ------
:      :   :      :   :      :   :      :   :      :
:      :   :      :   :      :   :      :   :      :
:      :   :      :   :      :   :      :   :      :
:      :   :      :   :      :   :      :   :  **  :
:      :   :      :   :      :   :  **  :   :  **  :
:      :   :      :   :  **  :   :  **  :   :  **  :
:      :   :      :   :  **  :   :  **  :   :  **  :
:  **  :   :  **  :   :  **  :   :  **  :   :  **  :
 ------     ------     ------     ------     ------
  >80        >81        >82        >83        >84
```

```
 ------     ------     ------     ------     ------
:      :   :      :   :      :   :  **  :   :  **  :
:      :   :      :   :  **  :   :  **  :   :  **  :
:      :   :  **  :   :  **  :   :  **  :   :  **  :
:  **  :   :  **  :   :  **  :   :  **  :   :  **  :
:  **  :   :  **  :   :  **  :   :  **  :   :  **  :
:  **  :   :  **  :   :  **  :   :  **  :   :  **  :
:  **  :   :  **  :   :  **  :   :  **  :   :  **  :
:  **  :   :  **  :   :  **  :   :  **  :   :  **  :
 ------     ------     ------     ------     ------
  >85        >86        >87        >88        >89
```

```
 ------     ------     ------     ------     ------
:      :   :  **  :   :      :   :  **  :   :     *:
:      :   :  **  :   :      :   :  **  :   :    * :
:      :   :  **  :   :      :   :  **  :   :    * :
:  ****:   :****  :   :******:   :  ****:   :   *  :
:  ****:   :****  :   :******:   :  ****:   :  *   :
:  **  :   :  **  :   :  **  :   :      :   : *    :
:  **  :   :  **  :   :  **  :   :      :   : *    :
:  **  :   :  **  :   :  **  :   :      :   :*     :
 ------     ------     ------     ------     ------
  >8A        >8B        >8C        >8D        >8E
```

```
 ------    ------    ------    ------    ------
|      |  |      |  |      |  |      |  |      |
|******|  |      |  |      |  |      |  |      |
|      |  |      |  |      |  |      |  |      |
|      |  |*     |  |**    |  |***   |  |****  |
|      |  |*     |  |**    |  |***   |  |****  |
|******|  |      |  |      |  |      |  |      |
|      |  |      |  |      |  |      |  |      |
|      |  |      |  |      |  |      |  |      |
 ------    ------    ------    ------    ------
  >8F       >90       >91       >92       >93

 ------    ------    ------    ------    ------
|      |  |      |  |******|  |      |  |  **  |
|      |  |      |  |      |  |      |  |  **  |
|      |  |      |  |      |  |      |  |  **  |
|***** |  |******|  |      |  |  **  |  |******|
|***** |  |******|  |      |  |  **  |  |******|
|      |  |      |  |      |  |      |  |  **  |
|      |  |      |  |      |  |      |  |  **  |
|      |  |      |  |      |  |      |  |  **  |
 ------    ------    ------    ------    ------
  >94       >95       >96       >97       >98

 ------    ------    ------    ------    ------
|******|  |      |  |  **  |  |  **  |  |  **  |
|******|  |      |  |  **  |  |  **  |  |  **  |
|******|  |      |  |  **  |  |  **  |  |  **  |
|******|  |****  |  |  ****|  |******|  |****  |
|******|  |****  |  |  ****|  |******|  |****  |
|******|  |  **  |  |  **  |  |      |  |      |
|******|  |  **  |  |  **  |  |      |  |      |
|******|  |  **  |  |  **  |  |      |  |      |
 ------    ------    ------    ------    ------
  >99       >9A       >9B       >9C       >9D

 ------    ------
|*     |  | *  * |
| *    |  | *  * |
| *    |  | *  * |
|  *   |  | *  * |
|   *  |  | *  * |
|    * |  | *  * |
|    * |  | *  * |
|     *|  | *  * |
 ------    ------
  >9E       >9F
```

ALC
PRINTER/PLOTTER
PRODUCT SPECIFICATION

Texas Instruments Incorporated
Consumer Products Group
Calculator Division
2/22/83

SECTION 1

Introduction

1.1 Purpose

This document describes the features and operational aspects of a printer peripheral to a calculator. The interface of this device to the calculator is described.

SECTION 2

Product Overview

2.1 Concept

The printer is an portable accessory to a calculator. It is a low power, low cost, plain paper printing device. It has a graphics mode in addition to a text mode.

2.2 Features

*Print mechanism : Alps Micrographic Printer DPG 13
... Print method : Ball point Pen
... Printing speed : 12 characters/second
... Characters/line : 18 std, 36 compressed
... Printing area : 43,2mm, 216 steps
... Line drawing speed :
X axis - 52mm/sec
Y axis - 52mm/sec
45 deg vector - 73mm/sec
... Line drawing resolution :
X axis - 0.2mm
Y axis - 0.2mm

*Paper
... Width : 58mm
... Diameter : 30mm internal, 63.5 external
... Length : 8.9M internal, ____ external

*Character set
... Characters : 96 ASCII
... Character size: 2 (text mode), 10 (graphics mode)

*Supply voltage
... 5 Ni-Cd batteries
... AC adapter 120 VAC input, 6 VDC regulated and unregulated output

*Current consumption
... 500 ma max when printing character set
... 80 microamps standby

*Physical
... Size : 118mm(W) X 148mm(L) X 44mm(H)
... Case : Injection molded painted thermoplastics

*Connectors
... Interface to calculator : 2 ea Berg 65945-208
... AC adapter :

SECTION 3

Operation

3.1 Text Mode

The printer is initialized to the text mode on power up and on reset. It prints characters corresponding to the ASCII formatted input data. It also accepts thirteen control codes. The character set and control codes are shown in Figure 1.

When in text mode the following restrictions apply:

1. All control codes from >00 to >0F not defined in Figure 1 will be ignored
2. All control codes from >10 to >1F must be the only character in the transmitted record

lsb\msb	0	1	2	3	4	5	6	7
0			sp	0	@	P	`	p
1	PEN 1	TEXT	!	1	A	Q	a	q
2	PEN 2	COMP.	"	2	B	R	b	r
3	PEN 3	GRAPH	#	3	C	S	c	s
4	PEN 4	NORM	$	4	D	T	d	t
5		TEST	%	5	E	U	e	u
6			&	6	F	V	f	v
7			'	7	G	W	g	w
8	BS		(	8	H	X	h	x
9			)	9	I	Y	i	y
A	LF		*	:	J	Z	j	z
B	LU		+	;	K	[	k	{
C			,	<	L	\	l	¦
D	CR		-	=	M	]	m	}
E			.	>	N		n	-->
F			/	?	O	_	o	<--

FIGURE 1 CHARACTER SET

Control codes:

Pen 1 - Pen 4 - Black, Blue, Green, and Red respectively

Backspace - Moves pen carriage backward one character space

Line Feed - Advances paper one line

Line Up - Advances paper one line in direction opposite LF

Carriage Return - Returns pen to left edge of paper

Text - Selects Text Mode

Comp - 36 char/line

Graph - Selects Graphic Mode

Norm - 18 char/line

Test - Draw a box in each of four colors

The Text control code is the only control code valid in the Graphic Mode (The graph control code will be ignored).

When in text mode the printer will automatically generate its own Carriage Return/Line Feed to segment a data record of printable characters that exceeds character line length.

As an option the printer will also generate a Carriage Return/Line Feed after each data record transmission. This option will be selected in the open statement from the calculator. Additionally, the user may select one of two character sizes in the open statement. See BUS MESSAGE STRUCTURE.

PROGRAM EXAMPLE

Program	Characters Printed
PRINT #1,"LOW COST PRINTER"	LOW COST PRINTER

Program

```
10  FOR I=33 TO 96
20  PRINT #1, CHR$(I);
30  NEXT I
40  END
```

Characters Printed

```
!"#$%&'()*+,-./01234
56789:;<=>?@ABCDEFGH
IJKLMNOPQRSTUVWXYZ[\
] _`
```

3.2 Graphic Mode

In the Graphic Mode input data is processed as graphic commands. Graphic commands are:

Define Origin : O
Set current location as origin.

Draw Line : L(x1,y1)[:[c][:p]],(x2,y2)[:[c][:p]],
...(xn,yn)[:[c][:p]]
Draws lines between coordinates (x1,y1), (x2,y2), through (xn,yn). Each point coordinate after the first represents the endpoint of the next line segment to be drawn, just as present position is assumed to be the beginning point for for that line segment.

p specifies the line type; p=0 specifies a solid line with p=1-9 specifying a dotted line with pitch .2mm+p*.2mm.

c specifies the pen color (c=1 specifies black, c=2 selects blue, c=3 gives green, and c=4 specifies red).

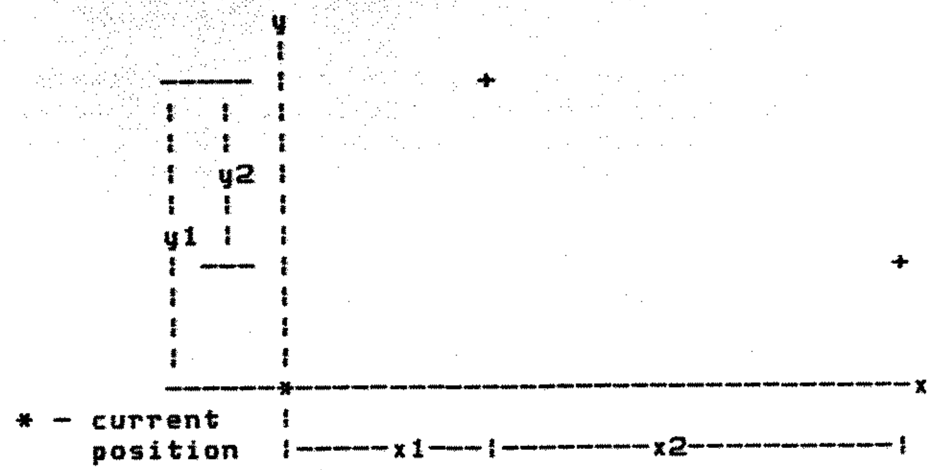
```
                  y
                  +
          -------:          +
          :      :
          :      :
          :      :
          :  --- :                    +
          :   :  :
          y1  :  :
          :   y2 :
          :   :  :
          :   :  :
        ---------*-----------------------------x
   * - origin    :-------------x2--------:
                 :-----x1-----:
```

L(x1,y1),(x2,y2)

Note: Point coordinates may be specified to $+/- 999.
Internally, they are maintained to $+/- 9999.

Draw Relative Line : R(x1,y1)[:[c][:p]],(x2,y2)[:[c][:p]], ...(xn,yn)[:[c][:p]]

Same as Line Command except all point cooridinates are taken relative to the present pen position rather than the origin.



R(x1,y1),(x2,y2)

Move Pen : M(x,y)

The pen is raised and positioned to coordinate x,y. The next command which causes the pen to draw will put the pen down.

Relative Move Pen : J(x,y)

The pen is raised and positioned to coordinate x,y relative to the current origin. The next command which causes the pen to draw will put the pen down.

Scale Character : Sn

Specifies character size n=0-9. The scale is initialized to 1 by power up or reset. The value set by Sn is retained when the mode is shifted from graphic mode to text mode and then back to graphic mode.

Home : H

Picks the pen up and positions it to the beginning of the line. The origin will be reset to that location. The pen will be set down by the next command that causes it to draw.

Angle : An

Specifies the angle at which to print characters. n=0 is an upright character, n=1 rotates the character 90 deg clockwise, n=2 rotates the character 180 deg clockwise and n=3 rotates the character 270 deg clockwise.

Text Printing : T(c1 c2 c3 .... cn)

Prints the text enclosed in parenthesis starting at the current cursor position. If the end of the line is reached while printing the text, scissoring will occur. Note that parenthesis may be embedded within the text stream.

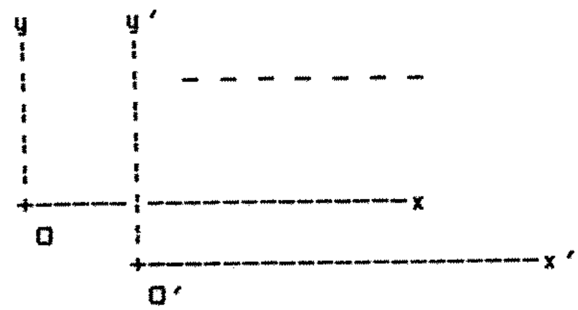
Color : Cn

Specifies pen color. n=1 selects black, n=2 selects blue, n=3 selects green, and n=4 selects red.

PROGRAM EXAMPLES

Program / **Characters Printed**

```
10 PRINT #1, CHR$(19)   ! Enter graphics
20 PRINT #1,"M(16,-4)" ! Position pen
30 PRINT #1,"O"         ! Set origin
40 PRINT #1,"L(6,12):4:3,(30,12)"
```



In this example statement 10 selects Graphics Mode. Statement 20 moves the pen to coordinates (16,-4) without drawing a line. Statement 30 redefines the origin as (16,-4). Statement 40 draws a dotted line from new coordinates (6,12) to (30,12).

Delete statements 20 and 30. Change statement 40 to read

```
40 PRINT #1,"L(22,8):2:3,(46,8)"
```

This will draw the same line with the blue pen.

Re-insert statement 20. Change statement 40 to read

```
40 PRINT #1,"R(6,12):2:3,(24,0)"
```

This will again draw the same line again in blue.

Add lines 50 and 60 -

```
50 PRINT #1,"A2"
60 PRINT #1,"T(ABC)"
```

This will print the characters ABC upside down to the left of where the line had terminated.

3.3 Initialization

The printer powers up and initializes to the Text Mode, Assuming a 18 character/line character scale. The x and y axis will be initialized as recommended in the specification for the printer mechanism. As recommended, the trial print for pen acclimatization will also be executed on power up. Upon completion of the initialization, the pen will reside at column position zero, the color used will be black, and pitch zero will be selected.

SECTION 4

Printing Characters in Graphic Mode

Each character is printed within an area on the paper called a character cell. Within this cell is the actual character as well as spacing to separate the character from the next. All character sizes are obtained by getting the integer multiple of the size of the smallest character as described below:

Scale Size	0	1	2	3	4	5	6	7	8	9
# Chrs/line *	36	18	12	9	7	6	5	4	4	3
Height ratio	1	2	3	4	5	6	7	8	9	10
Width ratio	1	2	3	4	5	6	7	8	9	10

* – If displayed horizontally across the page

Figure 4-1 Relative Character Scale Size

In graphic mode, the starting pen position for that character defines the lower left corner of the character cell. From that point (for all four print angles) the character is drawn. The below figure describes the character cell for compressed print (scale S0).

```
1 2 3 4 5 6
+-+-+-+-+-+-+ A
| | | | | | |
+-+-+-+-+-+-+ B
| | | | | | |
+-+-+-+-+-+-+ C
| | | | | | |
+-+-+-+-+-+-+ D
| | | | | | |
+-+-+-+-+-+-+ E
| | | | | | |
+-+-+-+-+-+-+ F ---
| | | | | | |       0.2 mm
+-+-+-+-+-+-+ G ---
| | | | | | |
+-+-+-+-+-+-+ H
| |
0.2 mm
```

Columns 1 through 5 and rows A through G are normally used for character drawing. Column 6 is always left blank to provide spacing between characters, and row H is used for lower case letter descenders.

Figure 4-2 Character Cell Example

Each successive scale size is an integer multiple of the number of rows and columns used in the above cell (i.e. S1 would have 12 columns and 16 rows, S2 would have 18 columns and 24 rows, etc.). Note that unlike dot matrix printers, characters are formed from lines drawn between the vortices of the cell.

When the print angle is rotated, what actually happens is that the character cell is rotated to the desired angle (0, 90, 180, 270 degrees). Characters are then printed in the direction indicated by that angle

```
       A3  /|\
             | +-+
          +---| | ---> A0  (Characters are printed in the
          +-+-*-+-+          direction indicated by the arrows)
    A2 <--- | |---+
             +-+ |    A1
                \|/        * - Starting pen position
```

Figure 4-3 Character Rotations

SECTION 5

Bus Interface

5.1 General

The printer communicates with the calculator via a peripheral bus. The bus supports communications between the calculator and other peripherals in addition to the printer.

Transmissions on the bus are defined in the context of a "message frame" which consists of a command message from the calculator and one or more response messages from the peripheral.

SECTION 6

Bus Signal and Timing description

6.1 Signals

The physical bus consists of 8 lines. There are 4 parallel data lines defining a 4 bit nibble as the basic unit of information carried on the bus. Data within the communication protocol is defined in 8-bit units (bytes). Each byte corresponds to two transmissions on the bus, least significant nibble first. The last two lines include a ground referance signal and a line reserved for future use. Data is output to the bus via open drain drivers.

D3	most significant data bit
D2	data bit
D1	data bit
D0	least significant data bit
HSK	handshake
BAV	bus available
FUT	reserved for future use
GND	ground

The speed of transmission of the data bus is controlled by the handshake line HSK. The BAV (bus available) signal is used to designate the beginning of a command message from the calculator.

6.2 Handshake Timing

The signal timing of HSK and the data lines is illustrated in Figure 2. The falling edge of HSK is the signal to receiving devices that a nibble of data is available on the bus. The rising edge of HSK is the signal to the transmitting device that all receivers have read the data. HSK is an open drain line so that any one device may hold it at a low level.

When the receiving device(s) see that HSK has gone low they rapidly (through hardware) pull HSK low also. The receiving device(s) then hold HSK low until they have processed the data. The transmitting device will release HSK shortly after pulling it low. This normal interaction is illustrated in Figure 3. If the transmitting device is slower than the receivers then it may dictate the bus speed as shown in Figure 4.

When a device is not interested in the data being transmitted it may disable itself from the bus and wait for the next message frame (denoted by BAV going low). Once a device has disabled itself from the bus, it need not participate in the handshake activity.

Figure 2
Bus Handshake Sequence

```
          __________                          __________
                    \                        /          \
   HSK               \______________________/            \__________
                     |                      |             |
                     |<- one handshake -->|             |<- next --

          ___________________________________________________________
          XXXXXXXXXX                          XXXXXXXXXXX
   DO-D3  XXXXXXXXXX__________________________XXXXXXXXXXX___________
                      |<--- data valid -->|
```

Figure 3
Handshake Components
(Receiver Limited)

```
                     __________                         ___________
                               \                       /           \
 HSK(composite)                 \_____________________/             \__________
                                |                     |             |
                                |<- one handshake -->|             |<- next --

                     __________           ____________________________
                               \         /                            \
 HSK(xmit)                      \_______/                              \__________

                     __________                         ___________
                               \                       /           \
 HSK(rcv)                       \_____________________/             \__________
```

Figure 4
Handshake Components
(Transmitter Limited)

```
                 ________                     ________
                         \                   /        \
HSK(composite)            \_________________/          \__________
                          |                 |           |
                          |<- one handshake -->|        |<- next --

                 ________                     ________
                         \                   /        \
HSK(xmit)                 \_________________/          \__________

                 ________             _________________
                         \           /                 \
HSK(rcv)                  \_________/                   \__________
```

Table 1
Handshake Timing Parameters
(Microseconds)

Item	Min	Max
HSK low to data valid	-	0
HSK low(xmit) to HSK low(bus)	-	3
HSK low(bus) to HSK low(rcv)	-	5
HSK low(xmit) to HSK high(xmit)	8	-
HSK high to data invalid	0	- **
HSK high to HSK low	8	20000 *

* within a message

** because data is output via open drain buffers, and HSK going high causes ones to be written into their output latch, data becomes invalid on the rising edge of HSK.

SECTION 7

Bus Message Structure

7.1 Protocol

As mentioned earlier, the data bus transmits command and response messages within the context of a message frame. In general the transmission of one command message from the calculator will cause one or more response messages to be transmitted back from the peripheral device selected. Each message consists of several nibble transfers as described in the previous section.

Each transmitted message contains overhead information to indicate such things as the selected device, the command code to be performed, and data length. The BAV (bus available) signal specifies the start of a message frame. When the calculator starts a message frame it first pulls BAV low. The command message from the master then follows that falling edge of BAV. The falling edge of BAV alerts all peripherals to look for the two-nibble device code which is always transmitted first in the command message (least significant nibble then most significant nibble). The BAV signal does not return to the high level until the message frame is complete. This is illustrated in Figure 5 below.

Figure 5
Message Frame

```
other   |   command   |   response   |
 lines  |   message   |   message    |

     _____                                               __________
          \                                             /
BAV        \________//_________________//_________/
                   //                  //
```

Table 2
Message Timing Parameters

Item	Min	Max
BAV low to HSK low	5	20000
HSK high to BAV high	1	--
End of command to start of response	10	--
HSK high to HSK low	8	20000
BAV high to BAV low (Within a message frame)	8	--

The first two nibbles of the command message always contain the device code of the peripheral to be addressed. All devices on the bus will read this number and test for a match. After the device code has been sent all devices except the one selected will ignore all further data in the message. The hardware will be designed such that they will not have to participate in the handshake sequence until the next falling edge of BAV.

Any device may extend the time to process data or wait for an operation to complete by holding HSK low until it is ready to start the next operation. Whenever HSK is high during a message, it must go low within 20ms or the receiver will time out.

The command and response messages are detailed in a following section.

As mentioned previously each device has a unique device code. In addition, the device code >00 is reserved to be recognized by all devices.

7.2 Command Message

The standard command message is divided into two distinct transmission phases. The first phase is the command transmission. The information necessary to the peripheral (including the data) is transmitted to the peripheral. The second phase consists of the peripheral transmitting its response to the command (In some cases, there will be no response to the command).

The following data is contained in a standard command message:

Field	Bytes
Device code	1
Command code	1
Logical Unit Number	1
Record number	2
Buffer length	2
Data length	2
Data	variable

The fields in a command message are described below.

7.2.1 Device Code.

This selects the peripheral which is to respond to the command. The printer peripherals have been assigned device codes 10 - 13 (decimal). Each printer on the bus must have a unique code within this range.

All devices on the bus will respond to device code 00, but will not return a response message.

7.2.2 Logical Unit Number - LUNO.

This is reserved. This field should be zero.

7.2.3 Command Code.

This field tells the peripheral the nature of the operation to be performed. The following lists the standard code assignments to be supported by the printer (hexadecimal).

- 00 - open
- 01 - close
- 04 - write data
- 07 - return status
- 0A - was it you?
- FE - null operation
- FF - bus reset

The specific actions of each of these commands is described later.

7.2.4 Record Number.

This is reserved. This field should be zero.

7.2.5 Buffer Length.

This field indicates the size of the data buffer for receiving data from a peripheral during the current bus operation. It is used by the master's IOS to check that the length of returned data does not exceed the buffer size, and may be ignored by the peripheral. This length is exclusive of the data length and return status bytes which form part of the response message (The status byte and data length replace their old fields in the PAB).

7.2.6 Data Length.

This field gives the number of bytes of data which follow in the data field.

7.2.7 Data.

This field contains data to be written to the peripheral device. The use of the data depends on the command code. If the data length field is zero then the data field is not present.

7.3 Response Message

The response message contains the following data

field	bytes
Data length	2
Data	variable
Operation status	1

7.3.1 Data Length.

This field specifies the number of bytes of data which follow in the data field.

7.3.2 Data.

This field contains the data to be returned to the master device; for example on an open operation. If the data length field is zero then this field will be omitted. Whenever the true data length cannot be determined at the time the length is sent, the data field will be padded with trailing zeroes for internal type files or devices, and with trailing blanks (>20) for display type files or devices.

7.3.3 Operation Status.

This field contains a status of the operation. The following lists the assigned response codes to be supported by the printer (decimal).

0 - normal operation completion
1 - Incorrect device option
2 - Attempt to open in incorrect mode (i.e. input)
4 - file/device not open error
5 - file/device already open
6 - device error
10 - it wasn't me
12 - buffer size error
13 - unsupported command error
25 - low batteries in peripheral (optional)
80 - Syntax error in data
81 - Coordinates out of bounds (> $+/- 9999)

7.4 Standard Access

The access to a printer is performed with a sequence of I/O calls. Before using the printer it should be opened with using the open command. This may be followed by other I/O calls to read or write data or perform other functions. When the calculator finishes using the printer it will issue a close I/O call. This will ensure that any necessary device dependent actions have been performed. The bus reset command code (>FF) will also close the printer if it is open.

7.5 Command Descriptions

This section describes the Command message setup and device response for the various standard I/O command codes.

7.5.1 Open - 00.

This command code is used to initiate the use of a peripheral. The printer will check access modes and ensure that it is not already open. If it is open or any other error occurs while the open is being attempted, then the appropriate error will be issued. Otherwise, the printer will be opened and reset to its initialized state. The command message will be set up as follows:

field	data
Device code	as required
Command Code	00
Logical Unit Number	don't care
Record number	don't care
Buffer length	as required (at least 0004)
Data length	as required (0004 returned)

The data buffer contains the following which is sent to the peripheral.

I/O buffer length (2 bytes)
(if zero then device returns buffer size)
device attributes (1 byte)
device characteristics (if any)

The printer will compare the 'I/O buffer length' to its capabilities and return either the requested length or the default length if the requested is zero (Default for the printer is >0050). The I/O buffer length is used by the master to determine what size buffer should be allocated by the master for read operations. It differs from the buffer length field in the PAB in that the buffer length field in the PAB only indicates the size of the buffer for the current bus operation.

The device attributes byte contains flags used to indicate the access mode of the peripheral. Several bits are unused by the I/O scheme and may be used by the application software as desired. Other bits must be set to zero to allow compatibility with future peripheral protocol enhancements.

The bit definitions are as follows (bit 0 is the least significant bit):

```
7-6 - access mode
      00 - invalid mode
      10 - output mode (write only - MUST be in this mode)
      01 - invalid mode
      11 - invalid mode
  5 - MUST be zero(0) - otherwise error
                (relative/sequential file type)
  4 - don't care (fixed/variable length records)
  3 - don't care (internal/display type)
  2 - don't care (reserved for device dependent use)
1-0 - don't care (these two bits may be used as desired)
```

The device characteristics field is a variable length field which contains device-dependent information relative to setting up the device.

Printer device dependant features:

C=[N or L]
- C=N causes the printer to NOT issue a carraige return and line feed at the end of each write command.
- C=L (default if omitted) causes the printer to issue a carraige return and line feed following each write command while in text mode.

S=[0 or 1]
- S=0 sets the text mode print size to compressed, and the graphic mode scale to S0.
- S=1 (default if omitted) sets the text mode print size to normal, and the graphic mode scale to S1.

Examples:

```
10 OPEN #1,"10.S=1",OUTPUT
```

causes the printer to set the text mode print size to normal, and the graphic mode scale to S1.

```
10 OPEN #1,"10.C=N",OUTPUT
```

causes the printer to never issue carraige return or linefeed unless the end of the line has been passed.

```
10 OPEN #1,"10.C=N,S=0",OUTPUT
```

causes the printer to never issue carraige return or linefeed unless the end of the line has been passed, and sets the text mode print size to compressed and the graphic mode scale to S0.

```
10 OPEN #1,"10",OUTPUT
```

uses both of the defaults described above.

NOTE

If both options are included in the open at the same time, the C option must precede the S option.

The response buffer will contain the accepted buffer length, and the record number that the file was opened to. For the printer, this record number is be meaningless, so a zero (0) will be returned. This information is always returned. Thus the response message for a successful open will be:

Data length		4
Data	(2 bytes)	Accepted buffer length
	(2 bytes)	always >0000
Operation status		0

An unsuccessful open may not return data in the response message. The operation status byte may contain the following error codes (decimal):

00 - successful open
01 - device/file characteristics error
* Open options not correct
* Attributes byte (possibly including I/O buffer length) not included in data
02 - Invalid attributes
* Open mode not output
* Open mode not sequential
05 - file/device already open
06 - device related error
* Pen detection mechanism broken
12 - buffer size error (Given if the buffer size in the PAB < 0004)
25 - batteries low in peripheral

7.5.2 Close - 01.

This command terminates the use of a device. Depending on the device this command may be used to clean up internal data (e.g. write an end of file) or may be effectively ignored. In general a close command must be sent between using a device and another open command. The data length for the close command will be zero (no data buffer is transmitted). The PAB should be set up as follows:

field	data
Device code	as required
Command Code	01
Logical Unit Number	don't care
Record number	don't care
Buffer length	don't care
Data length	0000 (0000 returned)

The response message will only contain a status byte and a zero data length (two bytes). The error status indications are (decimal):

00 - device or file closed
04 - device or file never opened
06 - device related error
  * - The ALC printer gives this error if data is transmitted to it in the close command
25 - batteries low in peripheral

7.5.3 Write Data - 04.

This command is used to send data to a peripheral device. The command message will contain the data to be sent to the device. The PAB should be set up as follows:

field	data
Device code	as required
Command Code	04
Logical Unit Number	don't care
Record number	don't care
Buffer length	don't care
Data length	record length (0000 returned)

The response message will contain zero-length data and the operation status. The following error status indications may occur (decimal):

00 - write successfully completed
04 - file/device not open
06 - Device error
25 - batteries low in peripheral (optional)
80 - Syntax error in data
81 - Coordinates out of range

7.5.4 Return Status - 07.

This command is used to return device status information. The information is returned in the data buffer. Certain bit fields in the return data are assigned to standard meanings while others are reserved for device dependent extensions. Certain devices may return more bytes of status if the buffer length allows. The PAB should be set up as follows:

field	data
Device code	as required
Command Code	07
Logical Unit Number	don't care
Record number	don't care
Buffer length	>= 0001
Data length	0000 (0001 returned)

The bit fields in the return data are as follows (bit 0 is the least significant bit).

- 7 - always one(1)
- 6 - always zero(0)
- 5 - always zero(0)
- 4 - 1 if open 0 if closed
- 3-2 - always zero(00)
- 1-0 - always two(10)

No errors can occur

7.5.5 Service Request Poll - OA.

This command allows a bus master to query a peripheral as to whether it requested service from the master. The PAB should be set up as follows:

field	data
Device code	as required
Command code	OA
Logical Unit Number	don't care
Record number	don't care
Buffer length	as determined in open
Data length	0000 (data length returned)
Return status	OA

The response message will consist of zero or more bytes of data, and the return status. The following error status indications may occur:

OA - it wasn't me (unsuccessful poll)

* Any other error code indicates a successful poll operation, and reflects the reason for the service request. Thus even devices that don't support service requests need to return the "it wasn't me" status code.

7.5.6 Null Operation - FE.

When the calculator receives a BAV interrupt and either no devices are enabled for interrupts or the current service flag is set, then a null operation code is sent to all devices. There will be no response to this message

field	data
Device code	00
Command code	FE
Logical Unit Number	don't care
Record number	don't care
Buffer length	don't care
Data length	00

7.5.7 Reset Bus - FF.

It may sometimes be desired to tell a device (or devices) to close all open files (or devices). This command will have no response by devices, they will simply perform the action requested (If they were not open to begin with, then they will do nothing). The ALC printer will initialize the print head and revert to a 'closed' status. The PAB should be set up as follows:

field	data
Device code	00
Command code	FF
Logical Unit Number	don't care
Record number	don't care
Buffer length	don't care
Data length	00

* For both the Null operation and Reset bus commands, no response will be forthcoming.

SECTION 8

Specifications

8.1 Mechanical

8.1.1 Top Case.

Injection molded thermoplastic part to provide upper portion of printer enclosure. Painted.

8.1.2 Bottom Case.

Injection molded thermoplastic part to provide mounting for PCB, mechanism, battery clips, and lower portion of printer enclosure. Painted.

8.1.3 Window.

Transparent injection molded thermoplastic part to permit viewing of the most recent line of printed information. Upper edge to be serrated to facilite tearing of paper.

8.1.4 Paper Storage Cover.

Injection molded thermoplastic part to enclose paper storage area. Lifts off to provide easy access to paper. Painted.

8.1.5 Retractable External Paper Spindle.

Detented in retracted and extended position to provide mointing for 63.5mm diameter paper roll external to the printer.

8.1.6 Feet.

Elastomer strips which attach to the bottom case and provide cushioning to reduce vibrations transmitted from the print mechanism and to prevent unwarranted movement on a working surface.

8.1.7 Pen Storage Area.

An area will be provided for the user to store four extra pens vertically oriented and capped. This storage area will be located under the paper access door.

8.2 Connectors

The printer includes a connector for the power adapter.

The printer inclydes two bus connectors wired in parallel. The connectors are Berg type 65945-208. Connector pin layout is shown in Figure 6 (looking into connector).

```
          __
         |  |
   ------    ------
  | 4   3   2   1 |
  |               |
  | 8   7   6   5 |
   ---------------
```

Terminal	Signal
1	DO
2	D1
3	BAV
4	GND
5	HSK
6	FUT
7	D2
8	D3

Figure 6
Bus Connector Layout

8.3 Switches

The printer has a on/off slide switch located on the front of the enclosure. It will also serve to reset the ALC Printer.

The printer has an internal 2 position slide switch to allow user programming of the least significant bit in the device code (decimal addresses 10 and 11). Additionally, there will be a jumper option to select the other two addresses in conjunction with the switch (decimal addresses 12 and 13).

The printer will have a momentary contact switch located on the top case to the right of the paper advance door to allow the user to advance the paper roll. This switch will only be active when the printer is not busy printing.

8.4 Packaging

The printer, AC adaptor, batteries, manual, paper roll, and pens will be packaged in a cardboard box and appropriate insert to meet Texas Instruments Consumer Products Group document QA 10237, Group A.

LONESTAR
RS-232/PARALLEL PORT
SOFTWARE FUNCTIONAL SPECIFICATION

CONSUMER PRODUCTS GROUP
CALCULATOR DIVISION

2/1/83

REVISION 1.7

SECTION 1

INTRODUCTION

1.1 PURPOSE

The RS-232/Parallel port interface provides the ALC with the ability to communicate with an RS-232 compatable device using the RS-232 port. It is also can communicate with a printer attached to the Centronix parallel port.

1.2 SCOPE

This document covers the functions of the RS-232/Parallel port software. Discussed are the software switch options, how BASIC and assembly language are to access the devices hooked up to the ports, how each port will respond to the IOS commands.

1.3 TERMINOLOGY

IOS - INPUT/OUTPUT SUBSYSTEM

1.4 RELATED DOCUMENTS

INTELLIGENT PERIPHERAL BUS STRUCTURE, TIMING, AND PROTOCOL SPECIFICATION

RS-232/PARALLEL PORT SOFTWARE DESIGN SPECIFICATION

SECTION 2

OVERVIEW OF THE RS-232/PARALLEL PORT

2.1 OPTION SUMMARY FOR RS-232

The RS-232 has several software switch options that are used to tell the peripheral and the computer how they are to communicate. These switches are specified when the device is opened.

1. BAUD RATE - The rate (in bits per second) at which data will be transferred.

2. DATA BITS - The number of bits of data in each character transferred.

3. PARITY - The method of checking the data transmission for errors.

4. NULLS - The number of null characters sent after a carriage return to allow for the printer carriage mechanism timing.

5. STOP BITS - The number of bits sent at the end of each character to indicate the end of the transmission of the character.

6. ECHO - Whether to retransmit each character received.

7. CARRIAGE RETURN - The adding of a linefeed and a carriage return or a carriage return to each variable length display type record.

8. TRANSFER TYPE - The method used to transfer data to the ALC (record or character).

9. DATA OVERRUN - Whether to report a data overrun error from the RS-232.

2.1.1 SPECIAL COMMAND CODES FOR THE RS232. There is a special command code in the RS-232 which causes the RS-232 to issue a continuous break (space condition) for .25 seconds on the communication lines. The RS-232 upon recieving this command holds up the bus until all of the characters in the output buffer are sent out and then until the break is complete. During this time it is still enabled to recieve characters if it is in update mode. The RS-232 is required to be in output or update mode to accept the command. The command code is >10(decimal 16) for this break command. This command is usefull in Telecommunications applications and is commonly used to get the user up to the top command level of a system that you are communicating with.

2.2 OPTION SUMMARY FOR PARALLEL PORT

The parallel port has the following software switches which controls how it communcates with the device on the parallel port. These switches are:

1. CARRIAGE RETURN - The adding of a linefeed and a carriage return or a carriage return to each variable length display type record.

2. STROBE LEVEL - The level of the strobe pulse sent out on the Parallel port to tell the device data is ready.

SECTION 3

RS-232 OPTIONS IN DEPTH

This section contains a list of the software options and their defaults. Then it goes into a good description of the software options which may be specified as part of the OPEN operation. The defaults will be set up at every open statement so

3.1 DEFAULT VALUES FOR THE SOFTWARE SWITCH OPTIONS

	DEFAULT	FORMAT	OPTIONS
BAUD RATE	(300) *	B=	50, 75, 110, 135, 150, 300, 600, 1200, 1800, 2400, 3600, 4800, 7200, 9600, 19200
DATA BITS	(7) *	D=	5, 6, 7, 8
PARITY	(O) *	P=	O, E, N, S, M
CHECK PARITY	(N)	C=	N, Y
NULLS	(0)	N=	0, 1...99
STOP BITS	(1)	S=	1, 2 -1 1/2 IF 5 DATA BITS
ECHO	(Y)	E=	N, Y
CARRIAGE RETURN	(L)	R=	N, C, L
TRANSFER TYPE	(R)	T=	R, C, W
DATA OVERRUN	(Y)	O=	N, Y

* indicates the default is set at powerup and then remains the same as its last value if it is not specified in the open.

NOTE

The RS-232 is also compatable with the TI home computer RS-232 software switches. All of the switches are supported and they are:

.BA= IS TREATED THE SAME AS SAYING B=
.PA= IS TREATED THE SAME AS SAYING P=
.DA= IS TREATED THE SAME AS SAYING D=
.TW IS TREATED THE SAME AS SAYING S=2
.NU IS TREATED THE SAME AS SAYING N=6
.CH IS TREATED THE SAME AS SAYING C=Y
.EC IS TREATED THE SAME AS SAYING E=N
.CR IS TREATED THE SAME AS SAYING R=N
.LF IS TREATED THE SAME AS SAYING R=C

The defaults are also completly TI home computer compatable.

3.2 BAUD RATE

The BAUD RATE is measured in bits per second. The default powerup BAUD RATE for the RS-232 is 300 BAUD. Once it has been set in an OPEN statement it will remain that way until it is specified in another OPEN statement. Any of the following baud rates may be selected by the option: 50, 75, 110, 135, 150, 300, 600, 1200, 1800, 2400, 3600, 4800, 7200, 9600, 19200. In addition to being one of the above the Baud Rate selected must be compatible with the device being attached to the RS-232 port.

For example if the option B=300 were specified in an open the device would be set up to transmit at a rate of up to 300 bits per second.

3.3 DATA BITS

The Data Bits switch specifies how many bits are in the characters being sent. The default value for data bits is 7. The values which can be used are 5, 6, 7, and 8.

For example the option D=7 will indicate that 7 bits are to be taken as a character, this is the ASCII standard.

3.4 PARITY

The PARITY switch specifies which type of parity is being used in the data transmission scheme. Some of the parity types will be checked for by the RS-232 interface and if they were not received correctly it will generate an error. The powerup default is P=O this implies an odd parity bit will be sent. Once the parity is set by an OPEN command it will stay at that value until a new OPEN command which specifies a different parity has been issued. the options are:

E for even parity, that is if there are an odd number of 1's in the transmission a the parity bit will be one, if not it will be a zero. If specified will be checked if the parity check flag is set.

O for odd parity, that is if there are an even number of 1's the parity bit will be a one, if not then it will be a zero. If specified will be checked for by the RS-232 interface if parity check flag is set.

S for space parity, that is the parity bit will allways be a zero. Will never be checked for by the RS-232 interface.

M for mark parity, that is the parity bit will allways be a one. Will never be checked for by the RS-232 interface.

N for don't send a parity bit at all.

For example the option P=O means that the parity bit is one if the transmission contains an even number of 1's and it is zero if the transmission contains an odd number of 1's.

3.5 CHECK PARITY

The CHECK PARITY switch tells the RS-232 weather or not to check parity. The default is N so the RS-232 will not check to make sure the parity returned by an attached device is correct. The Y option allows the RS-232 to check for any parity errors and if one occurs it issues an i/o error 81 parity error.

For example C=Y will cause the RS-232 to report any parity errors when they occur.

3.6 NULLS

The NULLS switch tells the software how many null characters to put after a carriage return. The default is zero nulls. You may specify any number between 0 and 99 for the number of nulls you wish to issue after a carriage return so that a printer mechanism will have time to complete it's return before it has to start printing characters.

For example the option N=27 will send 27 null characters after a carriage return to allow the printer mechanism to return to the beginning of a line before it starts to print.

3.7 STOP BITS

The STOP BITS switch tells the RS-232 how many stop bits to transmit with the data in a transmission. The default is 1 stop bit this means that 1 bit will be transmitted beyond the data to indicate the transmission is terminated. The options are 1 or 2.

For example the option S=2 would cause the RS-232 to generate 2 stop bits to terminate its transmission if data bits <> 5. If data bits = 5 then it would generate 1-1/2 stop bits to terminate its transmission.

3.8 ECHO

The ECHO switch tells the RS-232 whether or not to send a character it has just received back to the sending device. The default for this switch is Y. The two options are N which does not echo the characters back, and Y which echos the character back to the sending device.

For example E=Y would send back any character to the device it came from.

3.9 CARRIAGE RETURN

The CARRIAGE RETURN switch tells the RS-232 to send an automatic linefeed and carriage return or carriage return at the end of each display type record. The default is L for carriage return and linefeed. The options are:

N for no carriage return or linefeed.

C for carriage return only.

L for carriage return and linefeed.

For example R=N means the RS-232 will not send a carriage return and a linefeed at the end of every display type record.

3.10 TRANSFER TYPE

The TRANSFER TYPE switch tells the RS-232 whether to transfer a record, a character if it has one or wait for a character to come in and then transfer it. The default option is R for record. The options are:

R Transfers input to the ALC in records. These records are of the same length as the buffer length in the open. If a <CR> is encountered before a record as large as the buffer length has been received, the transmission is blank filled up to the buffer length. The RS-232 interface will hold up the IPB till enough characters or a <CR> has been received.

C Allows character by character transfers to the ALC. If any characters have been received, they will be sent to the ALC. If more characters have been received than the buffer allows for only the number of characters that fit in to the buffer will be sent. The rest will be kept in the buffer for future transmission. If no characters have been received then the RS-232 will return with a data length of 0.

W Also allows character by character transfers to the ALC. If any characters have been received it will behave like option C. If no characters have been received it will wait until a character is sent on the RS-232 and then send that character out on the IPB.

For example T=C will transfer one or more characters if the buffer is not empty. If the buffer is empty it will return a data length of zero and a status of 0.

3.11 DATA OVERRUN

The DATA OVERRUN option tells the RS-232 whether to report a data overrun error or not. When a data overrun error occurs data has been lost. The reason for a data overrun error is that the buffers filled up and no more characters could be received. Data received after this point is ignored till there is room for more characters. The default option is Y for report a data overrun as an error. The other option is N for do not report a data overrun condition to the ALC.

For example O=N will ignore data overruns when they happen. If the device is in echo mode the overrun character will not be echoed. The data will be lost which is received from the RS-232 during this time. Since there will be no data overrun error generated, a basic program will not stop running with an I/O error. This allows a basic program to do input from an RS-232 terminal and have a type ahead capability without erroring off if the user has typed too many characters ahead.

O=Y will issue an error when a data overrun error has occured. This will terminate execution of a basic program with an I/O error. This is important if you are using a modem or other data transfering device because if you have lost data the rest of the communication may be useless.

SECTION 4

PARALLEL PORT OPTIONS IN DEPTH

This section contains a list of the Parallel port software switch options and their defaults. Then it contains a good description of each option which may be specified as part of the open command.

4.1 DEFAULT VALUES FOR THE SOFTWARE SWITCH OPTION

	DEFAULT	FORMAT	OPTIONS
CARRIAGE RETURN	(L)	R=	N, C, L
STROBE LEVEL	(N)	S=	P, N

4.2 CARRIAGE RETURN

The CARRIAGE RETURN switch tells the RS-232 to send an automatic linefeed and carriage return or carriage return at the end of each display type record. The default is L for carriage return and linefeed. The options are:

N for no carriage return or linefeed.

C for carriage return only.

L for carriage return and linefeed.

For example R=N means that the Parallel port will not send out a carriage return and a linefeed at the end of every display type record.

4.3 STROBE LEVEL

The STROBE LEVEL switch tells the Parallel port to which level to use for the data strobe. The default is N for negative strobe. The option P causes the strobe to be positive.

For example S=P sets the Parallel port strobe to a positive pulse.

SECTION 5

GENERAL OPERATION

The programmer can communicate with the RS-232 port and the Parallel port in two ways, in assembly language, and thru BASIC commands and statements.

5.3.1 USE FROM ASSEMBLY LANGUAGE.

Communicating with devices using the Intelligent Peripheral Bus is explained in depth in the Lonestar Intelligent Peripheral Bus Structure, Timing, and Protocol Specification. The device numbers for the RS-232 port will range between 20 and 27. The device numbers for the Parallel port will range between 50 and 57.

5.3.2 USE FROM BASIC.

COMMANDS WHICH CAN BE USED WITH THE RS-232/PARALLEL PORT

COMMAND	RS-232	PARALLEL PORT
LIST	YES	YES
OLD	NO	NO
SAVE	NO	NO
OPEN	YES	YES
PRINT	YES *	YES *
INPUT	YES *	NO
LINPUT	YES *	NO
CLOSE	YES *	YES *
CALL I/O	YES	YES
EOF	YES *&	YES *&
FORMAT	NO	NO
DIR	NO	NO
VERIFY	NO	NO

* AFTER AN OPEN
& ALWAYS RETURNS NO END OF FILE

AN RS-232 EXAMPLE IN BASIC

```
10 DIM A$(80)
20 OPEN #20;"25.B=9600,P=E,R=N,T=W",DISPLAY,UPDATE
30 FOR I = 1 TO 80 :  INPUT #20;A$(I)
40 NEXT I
50 FOR I = 1 TO 80 :  PRINT #20;A$(I)
60 NEXT I
70 CLOSE #20
```

This program segment opens an RS-232 port, reads in at least 80 characters, writes them back to the RS-232 device, and then closes it. File number 20 is assigned to device 25 which is an RS-232 port by the open. The RS-232 is set up with a Baud rate of 9600, even Parity, and the transfer mode is to wait until there is at least one character received. The records will be of display type and you may read from and write to the device since it is in update mode. You will first input all 80 characters strings of 1 or more characters from using the file number 20 to indicate which file you are to use. Next the 80 character strings will be sent out to the RS-232 port. At the end the file number 20 must be closed. This close statement will wait for the output buffer of the RS-232 port to empty before allowing the program to complete.

ANOTHER RS-232 EXAMPLE IN BASIC

```
10 OPEN #7;"20.P=O,T=R",DISPLAY,INPUT
20 INPUT #7;A$
30 PRINT A$
40 CLOSE #7
```

This program segment opens an RS-232 port, reads in 80 characters and prints them to the LCD display of the ALC. File number 7 is assigned to device number 7 which is an RS-232 port by the open. The RS-232 is set up with a baud rate of 300 (which is the default), odd parity, now carriage return or linefeed at the end of the record (which is the default), transfer will be by records. These records will be 80 characters which is the default assigned by the RS-232 interface. The characters will be in ASCII form as specified by the display file type. The open mode will be Input so characters will be received from the RS-232 port. The 80 characters will be put in to the string A. The RS-232 interface will wait until either 80 characters or a carriage return have been typed. If a carriage return is typed the rest of the transmission will be padded with blanks up to 80 characters. The string a will then be printed out. The device will be closed imediately since there will be nothing in the output buffer to print out.

AN PARALLEL PORT EXAMPLE IN BASIC

```
10 DIM A$(40)
20 FOR I = 1 TO 40 : READ A$(I)
30 NEXT I
40 OPEN #23;"56.R=L",DISPLAY,OUTPUT
50 FOR I = 1 TO 40 : PRINT #23;A$(I)
60 NEXT I
70 PRINT #23;CHR$(12)
80 CLOSE #23
```

In this example 40 strings are read from data statements. File number 23 is assigned to device 56 which is a parallel port. The Parallel port is opened for output in display mode so it is to receive ASCII characters. The L option has been selected for the carriage return this means that at the end of the record a Carriage Return and a Linefeed are appended. The default for the strobe has been selected which is a negative strobe. The FOR loop will be executed 40 times and each of the strings read in will be printed out on a new line. After all the strings have been printed a form feed (ASCII 12) is sent to tell the printer to go to top of the form. Then the device must be closed using the close command.

SECTION 6

RESPONSE TO IOS COMMANDS

Refer to the I/O bus specification for a better clarification of each of the command codes. This section will deal with what commands are supported and which errors they generate. Both the LUNO and the RECORD NUMBER are ignored by the RS-232 and the Parallel port.

00(OPEN)

ACTION: AS IN I/O BUS SPECIFICATION

UNSUPPORTED: APPEND mode for both the Parallel port and the RS-232
UPDATE MODE for the Parallel port
INPUT MODE for the Parallel port
RELATIVE files for both the parallel port and the RS-232

Error Code	Possible Causes	Possible Fixes
1	-Bad option	-Check options and fix incorrect one.
	-No separator	-Check for comma, period or equals in the correct place.
2	-Bus failure	-Try again.
	-No attributes in the IO call	-Fix PAB.
	-No buffer length in the IO call	-Fix PAB.
5	-Device was opened before and never closed	-Close the device and try again.
17	-Device opened for relative files	-Select a different file type.
19	-Device opened in append mode	-Change to another open mode.
21	-Parallel port opened in input mode	-Select output mode.

Error Code	Possible Causes	Possible Fixes
22	-Parallel port opened update mode	-Select output mode.

01(CLOSE)

ACTION: As in I/O BUS SPECIFICATION The RS-232 will not close until its output buffer is empty.

Error Code	Possible Causes	Possible Fixes
4	-Device not open	-Open device first
8	-Bus failure -Data too long in IO call	-Try again -Fix PAB

02(DELETE OPEN FILE)

ACTION: UNSUPPORTED

Error Code	Possible Causes	Possible Fixes
13	-Peripherals won't support this command	-Use regular close command.

03(READ)

ACTION: As in I/O BUS SPECIFICATION for RS-232 Unsupported for the Parallel port

Error Code	Possible Causes	Possible Fixes
4	-Device not open	-Open the device before read.
8	-Bus failure -Data too long in IO call	-Try again -Fix PAB
13	-Parallel Port read not supported	-Don't try to read Parallel Port.
16	-Device not opened in input mode	-Change the open statement to open the device in input mode.
80	-Data was lost due to a full input buffer or output buffer if	-Do input statements from the RS-232 more frequently. -If it is not important that

	you are in echo mode	this data was lost you may set the Data overrun option to O=N so the error is not reported.
81	-A parity error has occured.	-Check parity setting in the open with that of the RS-232 device and Set them to the same parity.
82	-A Framing error has occured.	-Check the number of stop bits in the open against the setting of the device connected make sure they are the same. -Check the Parity and Data bits to make sure they are set the same as the device.
83	-Framing and parity errors have occured	-Refer to errors 81 and 82 fixes.

04(WRITE)

ACTION: As in I/O BUS SPECIFICATION
The Parallel port will hold up the bus until the last character is sent by the alc. It will then issue a return message to the alc. Then it will print the last character followed by a cr and lf depending upon the carriage return option specified in the open. If for some reason writing these last characters fails it could hold up the bus the next time bus communications are attempted. The only way to clear this condition would be to turn off the RS-232/Parallel port interface to reset it.

The RS-232 will hold bus up until it has buffered up the data.

Error Code	Possible Causes	Possible Fixes
4	-Device not open	-Open the device.
14	-Device not open for output	-Open the device in output mode.

05(POSITION FILE)

ACTION: UNSUPPORTED

Error Code	Possible Causes	Possible Fixes

13	-Peripherals won't support this command	-Don't use this command

06(DELETE)
ACTION: UNSUPPORTED

Error Code	Possible Causes	Possible Fixes
13	-Peripherals won't support this command	-Don't use this command

07(RETURN STATUS)
ACTION: As in I/O BUS SPECIFICATION under data comunications dev
If the buffer allows (ie buffer size of 3 or greater)
then two additional bytes of data are sent back they a
amount of data in the devices input buffer. The LSB of t
length is sent first then the MSB followed by the device
us.

Error Code	Possible Causes	Possible Fixes
8	-Bus error -misuse of IO call by sending data	-Try it again. -Fix PAB.

08(YOU MAY REQUEST SERVICE)
ACTION: As in I/O BUS SPECIFICATION for RS-232
UNSUPPORTED for Parallel port

Error Code	Possible Causes	Possible Fixes
4	-Device not open	-Open device before issuing this command.
13	-Peripherals won't support this command	-Don't use this command

09(YOU MAY NOT REQUEST SERVICE)
ACTION: As in I/O BUS SPECIFICATION for RS-232
UNSUPPORTED for Parallel port

Error code	Possible causes	Possible fixes

4	-Device not open	-Open device before issuing this command.
13	-Peripherals won't support this command	-Don't use this command

OA(WAS IT YOU?)

ACTION: As in I/O BUS SPECIFICATION
Parallel port will allways return OA

Error code	Possible causes	Possible fixes
8	-Bus error -misuse of IO call	-Try it again. -Fix PAB.
10	-Misuse of the status from an IO call	-Use response for correct purpose.
16	-Device not opened in input mode	-Change the open statement to open the device in input mode.
80	-Data was lost due to a full input buffer or output buffer if you are in echo mode	-Do input statements from the RS-232 more frequently. -If it is not important that this data was lost you may set the Data overrun option to O=N so the error is not reported.
81	-A parity error has occured.	-Check parity setting in the open with that of the RS-232 device and Set them to the same parity.
82	-A Framing error has occured.	-Check the number of stop bits in the open against the setting of the device connected make sure they are the same. -Check the Parity and Data bits to make sure they are set the same as the device.
83	-Framing and parity errors have occured	-Refer to errors 81 and 82 fixes.

OB(YOU ARE THE MASTER)
ACTION: UNSUPPORTED

Error code	Possible causes	Possible fixes
13	-Peripherals won't support this command	-Don't use this command

OC(VERIFY READ/WRITE OPERATION)
ACTION: UNSUPPORTED

Error code	Possible causes	Possible fixes
13	-Peripherals won't support this command	-Don't use this command

OD(FORMAT MEDIA)
ACTION: UNSUPPORTED

Error code	Possible causes	Possible fixes
13	-Peripherals won't support this command	-Don't use this command

OE(READ CATALOG)
ACTION: UNSUPPORTED

Error code	Possible causes	Possible fixes
13	-Peripherals won't support this command	-Don't use this command

OF(SET CHARACTERISTICS)
ACTION: AS IN I/O BUS SPECIFICATION

Error code	Possible causes	Possible fixes
1	-Bad option	-Check options and fix incorrect one.
	-No separator	-Check for comma, period or equals in the correct place
4	-Device not open	-Open device before issuing

this command.

10(SEND BREAK)

ACTION: SENDS A BREAK SIGNAL OUT OF THE RS-232 FOR .25 SECONDS

Error code	Possible causes	Possible fixes
4	-Device not open	-Open device before issuing
8	-Bus error -misuse of IO call	-Try it again. -Fix PAB.
14	-Device not open for output	-Open the device in output mode.

FE(NULL OPERATION)

ACTION: AS IN I/O BUS SPECIFICATION will be ignored by both devices

ERRORS: NONE

FF(RESET BUS)

ACTION: AS IN I/O BUS SPECIFICATION If the device is open it will be closed and the peripheral will reset thru its initialization routine

ERRORS: NONE

SECTION 7

I/O ERROR CODES

CODE	DECIMAL	MEANING
00	0	Everything is ok
01	1	Device/file characteristics error
02	2	Error in attributes byte
03 *	3	File not found
04	4	File/device not open
05	5	File/device already open
06 *	6	Device error
07 *	7	EOF error
08	8	Data/file too long error
09 *	9	Write protect error
0A	10	It wasn't me
0B *	11	Directory full error
0C	12	Buffer size error
0D	13	Unsupported command error
0E	14	File not open for write
0F	15	File not open for read
10 *	16	Low batteries in peripheral
11	17	Relative file not supported
12 *	18	Sequential file not supported
13	19	Append mode not supported
14 *	20	Output mode not supported
15	21	Input mode not supported
16	22	Update mode not supported
17 *	23	Internal or display type file not supported
FE *	254	Call to bus master from slave application error
FF	255	Bus time out error

A * indicates that this error is not used in the RS-232 interface

SECTION 8

SPECIAL SOFTWARE MODES

There are several software modes available in the RS-232 Parallel port interface Version V821006. These modes are selected by the state of certain lines on the A Port of the TMS 7040 containing the code. The different modes select what device range the parallel port is selected whether or not the RS-232 is enabled. Below is a table of the way to select the modes.

A PORT VALUE	Parallel port address range	RS-232 state
XXX00XXX	50 TO 54	ON
XXX01XXX	15 TO 19	ON
XXX10XXX	50 TO 54	OFF
XXX11XXX	15 TO 19	OFF

NOTE

X Means don't care
1 Means a high value on the port bit
0 Means a low value on the port bit

Bit 3 on the A port selects device range 15-19 if it is set for parallel port.
If it is not set then it selects device range 20-24 for the parallel port.
Bit 4 on the A port if it is set turns off the RS-232.
If it is not set then it turns on the RS-232.

SECTION 9

ISSUES

HEX-BUS MODEM
FUNCTIONAL SPECIFICATION

CONSUMER PRODUCTS GROUP
CALCULATOR AND COMPACT COMPUTER DIVISION

5/17/83

REVISION 2.5

SECTION 1

INTRODUCTION

1.1 PURPOSE

This document describes the functional attributes of a modem designed for use with the CC-40, TI-99/2, and other TI computers with Hex-bus interface capability.

1.2 SCOPE

This document covers the features and basic operation of the Modem including software switch options and the basic electrical and mechanical features.

1.3 RELATED DOCUMENTS

INTELLIGENT PERIPHERAL BUS STRUCTURE, TIMING, AND PROTOCOL SPECIFICATION

INTEGRATED CIRCUITS, INTELLIGENT PERIPHERAL BUS CONTROLLER (IBC) drawing no. 1052911

SECTION 2

OVERVIEW

2.1 BASIC FEATURES

1. BELL 103 COMPATIBILITY - 300 baud FSK originate or answer mode, full or half duplex.

2. DIRECT CONNECT - Connects to a standard RJ11C Type of modular phone jack.

3. HEX-BUS - Operates directly with all computers equipped with Hex-bus interface.

4. INTELLIGENCE - A microprocessor is used to implement the Hex-bus interface as well as provide many of the features which would be found in an RS-232 interface. Some of the features include options for parity, number of bits, echo, etc. Data buffers are also provided: 80 characters for incoming data and 1 character for output.

5. INDICATOR - An LED is provided to signal Hex-bus activity.

6. BATTERY OPERATION - The Modem uses 4 AA alkaline cells for portable operation and has a battery life of at least 30 hours. An AC adaptor is an optional accessory.

2.2 OPTION SUMMARY

The Modem has several software switch options that are used to tell the peripheral and the computer how they are to communicate. These switches are specified when the device is opened.

1. ANSWER/ORIGINATE MODE - Selects which set of tones are

used for transmitting data.

2. DATA BITS - The number of bits of data in each character transferred.

3. PARITY - The method of checking the data transmission for errors.

4. STOP BITS - The number of bits sent at the end of each character to indicate the end of the transmission of the character.

5. ECHO - Whether to retransmit each character received.

6. CARRIAGE RETURN - The adding of a linefeed and a carriage return or a carriage return to each record.

7. TRANSFER TYPE - The method used to transfer data to the computer (record or character).

8. DATA OVERRUN - Whether to report a data overrun error from the Modem.

9. BAUD RATE - Only 300 baud may be specified.

10. MESSAGE - Whether to transmit the "CARRIER DETECT" message.

2.3 SPECIAL COMMAND CODES FOR THE MODEM

There is a special command code in the Modem which causes the Modem to issue a continuous break (space condition) for 0.25 seconds on the communication lines. The Modem upon receiving this command holds up the bus until all of the characters in the output buffer are sent out and then until the break is complete. During this time it is still enabled to receive characters if it is in update mode. The Modem is required to be in output or update mode to accept the command. The command code is >10 (decimal 16) for this break command. This command is commonly used to get the user up to the top command level of a system that you are communicating with.

SECTION 3

MODEM OPTIONS IN DEPTH

This section contains a list of the software options and their defaults. Then it goes into a description of the software options which may be specified as part of the OPEN operation. The defaults will be set up at every OPEN statement unless otherwise specified.

3.1 DEFAULT VALUES FOR THE SOFTWARE SWITCH OPTIONS

	DEFAULT	FORMAT	OPTIONS
ANSWER/ORIGINATE MODE	(O)	M=	A, O
DATA BITS	(7) *	D=	7, 8
PARITY	(O) *	P=	O, E, N, S, M
CHECK PARITY	(N)	C=	N, Y
STOP BITS	(1)	S=	1, 2
ECHO	(Y)	E=	N, Y
CARRIAGE RETURN	(L)	R=	N, C, L
TRANSFER TYPE	(R)	T=	R, C, W
DATA OVERRUN	(Y)	O=	N, Y
BAUD RATE	(300)	B=	300 **
NO MESSAGE		NM	

* Indicates the default is set at powerup and then remains the same as its last value if it is not specified in the OPEN.

** Only one value may be used. The command is used to provide commonality with software written for RS-232 driven modems.

NOTE

The Modem is also compatible with the TI home computer RS-232 software switches. These options are supported:

BA= IS TREATED THE SAME AS SAYING B=
PA= IS TREATED THE SAME AS SAYING P=
DA= IS TREATED THE SAME AS SAYING D=
TW IS TREATED THE SAME AS SAYING S=2
CH IS TREATED THE SAME AS SAYING C=Y
EC IS TREATED THE SAME AS SAYING E=N
CR IS TREATED THE SAME AS SAYING R=N
LF IS TREATED THE SAME AS SAYING R=C

The defaults are also completely TI home computer compatible.

3.2 BAUD RATE

The BAUD RATE is measured in bits per second. The default powerup BAUD RATE for the Modem is 300 BAUD. Only B=300 or BA=300 are allowed. Any other value will produce an error (code 1). This option is only useful to maintain software compatibility with RS-232 driven modems.

3.3 DATA BITS

The Data Bits switch specifies how many bits are in the characters being sent. The default value for data bits is 7. The values which can be used are 7 and 8. Once the number of bits is set by an OPEN command, it will stay at that value until a new OPEN command specifies a different value.

For example the option D=7 will indicate that 7 bits are to be taken as a character. This is the ASCII standard.

3.4 PARITY

The PARITY switch specifies which type of parity is being used in the data transmission scheme. Some of the parity types will be checked for by the Modem and if they were not received correctly it will generate an error. The powerup default is P=O. This implies an odd parity bit will be sent. Once the parity is set by an OPEN command it will stay at that value until a new OPEN command which specifies a different parity has been issued. The options are:

E for even parity, that is if there are an odd number of 1's in the transmission the parity bit will be one, if not it will be a zero. If specified will be checked if the parity check flag is set.

O for odd parity, that is if there are an even number of 1's the parity bit will be a one, if not then it will be a zero. If specified will be checked for by the Modem if parity check flag is set.

S for space parity, that is the parity bit will always be a zero. Will never be checked for by the Modem.

M for mark parity, that is the parity bit will always be a one. Will never be checked for by the Modem.

N for don't send a parity bit at all.

For example the option P=O means that the parity bit is one if the transmission contains an even number of 1's and it is zero if the transmission contains an odd number of 1's.

3.5 CHECK PARITY

The CHECK PARITY switch tells the Modem whether or not to check parity. The default is N so the Modem will not check to make sure the parity returned by an attached device is correct. The Y option allows the Modem to check for any parity errors and if one occurs it issues an I/O error 81 parity error.

For example C=Y will cause the Modem to report any parity errors when they occur.

3.6 STOP BITS

The STOP BITS switch tells the Modem how many stop bits to transmit with the data in a transmission. The default is 1 stop bit. This means that 1 bit will be transmitted beyond the data to indicate the transmission is terminated. The options are 1 or 2.

For example the option S=2 would cause the Modem to generate 2 stop bits to terminate its transmission.

3.7 ECHO

The ECHO switch tells the Modem whether or not to send a character it has just received back to the sending device. The default for this switch is Y. The two options are N which does not echo the characters back, and Y which echos the character back to the sending device.

For example E=Y would send back any character to the device it came from.

3.8 CARRIAGE RETURN

The CARRIAGE RETURN switch tells the Modem to send an automatic linefeed and carriage return or carriage return at the end of each record. The default is L for carriage return and linefeed. The options are:

N for no carriage return or linefeed.

C for carriage return only.

L for carriage return and linefeed.

For example R=N means the Modem will not send a carriage return and a linefeed at the end of every record.

3.9 TRANSFER TYPE

The TRANSFER TYPE switch tells the Modem whether to transfer a record, a character if it has one or wait for a character to come in and then transfer it. The default option is R for record. The options are:

R Transfers input to the computer in records. These records are of the same length as the buffer length in the OPEN. If a <CR> is encountered before a record as large as the buffer length has been received, the transmission is blank filled up to the buffer length. The Modem will hold up the Hex-bus until enough characters or a <CR> has been received. Any <LF> and <CR> characters are removed before the record is sent to the computer.

C Allows character by character transfers to the computer. If any characters have been received, they will be sent to the computer. If more characters have been received than the buffer allows for, only the number of characters that fit into the buffer will be sent. The rest will be kept in the buffer for future transmission. If no characters have been received then the Modem will return with a data length of 0.

W Also allows character by character transfers to the computer. If any characters have been received it will behave like option C. If no characters have been received it will wait until 1 character is received by the Modem and then send that character out on the Hex-bus.

For example T=C will transfer one or more characters if the buffer is not empty. If the buffer is empty it will return a data length of zero and a status of 0.

3.10 DATA OVERRUN

The DATA OVERRUN option tells the Modem whether to report a data overrun error or not. When a data overrun error occurs, data has been lost. The reason for a data overrun error is that the buffer filled up and no more characters could be received. Data received after this point is ignored till there is room for more characters. The default option is Y for report a data overrun as an error. The other option is N for do not report a data overrun condition to the computer.

For example O=N will ignore data overruns when they happen. If the device is in echo mode the overrun character will not be echoed. Data received by the Modem during this time will be lost. Since there will be no data overrun error generated, a BASIC program will not stop running with an I/O error. This allows a BASIC program to do input from a Modem connected terminal and have a type ahead capability without erroring off if the user has typed too many characters ahead.

O=Y will issue an error when a data overrun error has occured. This will terminate execution of a BASIC program with an I/O error. This is important in some applications where losing part of the data will mean that the rest of the communication may be useless.

3.11 NO MESSAGE

Unless the NM (no message) option is chosen in the OPEN, the message "CARRIER DETECT" will be placed in the input buffer each time the Modem is opened as soon as the carrier from the other modem is detected. Therefore the first read command issued after the carrier is found will return the data "CARRIER DETECT". If NM is chosen as an option, "CARRIER DETECT" will not be placed in the input buffer.

SECTION 4

GENERAL OPERATION

The programmer can communicate with the Modem in two ways, in assembly language, and thru BASIC commands and statements.

4.11.1 USE FROM ASSEMBLY LANGUAGE.

Communicating with devices using the Intelligent Peripheral Bus is explained in depth in the Intelligent Peripheral Bus Structure, Timing, and Protocol Specification. The device numbers for the Modem range between 70 and 73.

4.11.2 USE FROM BASIC.

COMMANDS WHICH CAN BE USED WITH THE MODEM

COMMAND	ALLOWED
LIST	NO
OLD	NO
SAVE	NO
OPEN	YES
PRINT	YES *
INPUT	YES *
LINPUT	YES *
CLOSE	YES *
CALL I/O	YES
EOF	YES *&
FORMAT	NO
DIR	NO
VERIFY	NO

* AFTER AN OPEN
& ALWAYS RETURNS NO END OF FILE

A MODEM EXAMPLE IN BASIC

```
10 DIM A$(80)
20 OPEN #20,"70.B=300,P=E,R=N,T=W,E=N",DISPLAY,UPDATE
```

```
30 FOR I = 1 TO 80 : INPUT #20,A$(I)
40 NEXT I
50 FOR I = 1 TO 80 : PRINT #20,A$(I)
60 NEXT I
70 PAUSE
80 CLOSE #20
```

This program segment opens the Modem, reads in at least 80 characters, writes them back to the Modem, and then closes it. File number 20 is assigned to device 70 which is a Modem by the open. The Modem is set up with a Baud rate of 300, even Parity, no echo, and the transfer mode is to wait until there is at least one character received. The records will be of display type and you may read from and write to the device since it is in update mode. You will first input all 80 characters strings of 1 or more characters from the file number 20. Next the 80 character strings will be sent out to the Modem. At the end, a pause is used before the close since a close will cause the receiving modem to pick up at least one illegal character in this case due to the noise created when the tone is removed from the phone line by the close statement. When writing routines for the Modem, the receiving modem should close first whenever possible to avoid this problem.

ANOTHER EXAMPLE IN BASIC

```
10 OPEN #7,"70.P=O,T=R",DISPLAY,INPUT
20 INPUT #7,A$
30 PRINT A$;PAUSE 2
40 CLOSE #7
```

This program segment opens a Modem, reads in 80 characters and prints them to the LCD display of the CC-40 for 2 seconds. File number 7 is assigned to device number 70 which is a Modem by the OPEN. The Modem is set up with a baud rate of 300 (which is the default), odd parity, carriage return and linefeed at the end of the record (which is the default), transfer will be by records. These records will be 80 characters which is the default assigned by the Modem interface. The characters will be in ASCII form as specified by the display file type. The open mode will be Input so characters will be received from the Modem. The 80 characters will be put in to the string A. The Modem interface will wait until either 80 characters or a carriage return have been typed. If a carriage return is typed the rest of the transmission will be padded with blanks up to 80 characters. In this case, the first record received after the Modem is opened is always "CARRIER DETECT". The option, NM, can be specified to eliminate this message. The string will then be printed out. The device will be closed immediately since there will be nothing in the output buffer to send out.

SECTION 5

RESPONSE TO IOS COMMANDS

Refer to the I/O bus specification (INTELLIGENT PERIPHERAL BUS STRUCTURE, TIMING, AND PROTOCOL SPECIFICATION) for a better clarification of each of the command codes. This section will deal with what commands are supported and which errors they generate. Both the LOGICAL UNIT NUMBER and the RECORD NUMBER are ignored by the Modem.

OO(OPEN)

ACTION: AS IN I/O BUS SPECIFICATION

UNSUPPORTED: APPEND mode
RELATIVE files

Error Code	Possible Causes	Possible Fixes
>01	-Bad option	-Check options and fix incorrect one.
	-No separator	-Check for comma, period or equals in the correct place.
>02	-No attributes in the IO call	-Fix PAB.
	-No buffer length in the IO call	-Fix PAB.
>05	-Device was opened before and never closed	-Close the device and try again.
>0C	-Buffer length small	-Must be 4 or more.
>11	-Device opened for relative files	-Select a different file type.
>13	-Device opened in append mode	-Change to another open mode.
>19	-Low battery	-Change batteries or plug in adaptor.

01(CLOSE)

ACTION: As in I/O BUS SPECIFICATION
The Modem will not close until its output buffer is empty.

Error Code	Possible Causes	Possible Fixes
>04	-Device not open	-Open device first.
>08	-Data too long in IO call	-Fix PAB.

02(DELETE OPEN FILE)

ACTION: UNSUPPORTED

Error Code	Possible Causes	Possible Fixes
>0D	-Peripheral won't support this command	-Use regular close command.

03(READ)

ACTION: As in I/O BUS SPECIFICATION

Error Code	Possible Causes	Possible Fixes
>04	-Device not open	-Open the device before read.
>08	-Data too long in IO call	-Fix PAB.
>0F	-Device not opened in input mode	-Change the OPEN statement to open the device in input mode.
>50	-Data was lost due to a full input buffer or output buffer if you are in echo mode	-Do input statements from the Modem more frequently. -If it is not important that this data was lost you may set the Data overrun option to O=N so the error is not reported.
>51	-A parity error has occurred.	-Check parity setting in the OPEN with that of the Modem device and set them to the same parity.

>52	-A Framing error has occurred.	-Check the number of stop bits in the OPEN against the setting of the device connected. Make sure they are the same. -Check the Parity and Data bits to make sure they are set the same as the device.

04(WRITE)

ACTION: As in I/O BUS SPECIFICATION

The Modem will hold bus up until it has buffered up the data.

Error Code	Possible Causes	Possible Fixes
>04	-Device not open	-Open the device.
>0E	-Device not open for output	-Open the device in output mode.

05(POSITION FILE)

ACTION: UNSUPPORTED

Error Code	Possible Causes	Possible Fixes
>0D	-Peripheral won't support this command	-Don't use this command.

06(DELETE)

ACTION: UNSUPPORTED

Error Code	Possible Causes	Possible Fixes
>0D	-Peripheral won't support this command	-Don't use this command.

07(RETURN STATUS)

ACTION: As in I/O BUS SPECIFICATION under data comunications devices. If the buffer allows (ie. buffer size of 3 or greater) then two additional bytes of data are sent back. They are the amount of data in the devices input buffer. The LSB of the length is sent first, then the MSB followed by the device status.

Error Code	Possible Causes	Possible Fixes
>08	-misuse of IO call by sending data	-Fix PAB.

08(YOU MAY REQUEST SERVICE)

ACTION: As in I/O BUS SPECIFICATION

Error Code	Possible Causes	Possible Fixes
>04	-Device not open	-Open device before issuing this command.
>0F	-Device not opened in input mode	-Change the OPEN statement to open the device in input mode.

09(YOU MAY NOT REQUEST SERVICE)

ACTION: As in I/O BUS SPECIFICATION

Error code	Possible causes	Possible fixes
>04	-Device not open	-Open device before issuing this command.

0A(WAS IT YOU?)

ACTION: As in I/O BUS SPECIFICATION

Error code	Possible causes	Possible fixes
>08	-misuse of IO call	-Fix PAB.
>0A	-It was not me	-Correct response for a peripheral that did not request service.
>0F	-Device not opened in input mode	-Change the OPEN statement to open the device in input mode.
>50	-Data was lost due to a full input buffer or output buffer if	-Do input statements from the Modem more frequently. -If it is not important that

Error code	Possible causes	Possible fixes
	you are in echo mode	this data was lost you may set the Data overrun option to O=N so the error is not reported.
>51	-A parity error has occurred.	-Check parity setting in the OPEN with that of the Modem device and set them to the same parity.
>52	-A Framing error has occurred.	-Check the number of stop bits in the OPEN against the setting of the device connected. Make sure they are the same. -Check the Parity and Data bits to make sure they are set the same as the device.

OB(YOU ARE THE MASTER)
ACTION: UNSUPPORTED

Error code	Possible causes	Possible fixes
>OD	-Peripheral won't support this command	-Don't use this command.

OC(VERIFY READ/WRITE OPERATION)
ACTION: UNSUPPORTED

Error code	Possible causes	Possible fixes
>OD	-Peripheral won't support this command	-Don't use this command.

OD(FORMAT MEDIA)
ACTION: UNSUPPORTED

Error code	Possible causes	Possible fixes
>OD	-Peripheral won't support this command	-Don't use this command.

OE(READ CATALOG)
ACTION: UNSUPPORTED

Error code	Possible causes	Possible fixes
>0D	-Peripheral won't support this command	-Don't use this command.

OF(SET CHARACTERISTICS)
ACTION: AS IN I/O BUS SPECIFICATION

Error code	Possible causes	Possible fixes
>01	-Bad option	-Check options and fix incorrect one.
	-No separator	-Check for comma, period or equals in the correct place.
>04	-Device not open	-Open device before issuing this command.

10(SEND BREAK)
ACTION: SENDS A BREAK SIGNAL OUT OF THE MODEM FOR 0.25 SECONDS

Error code	Possible causes	Possible fixes
>04	-Device not open	-Open device before issuing
>08	-misuse of IO call	-Fix PAB.
>0E	-Device not open for output	-Open the device in output mode.

50(SELF TEST)
ACTION: PLACES THE MODEM CHIPS IN THE SELF TEST MODE. This creates a closed loop so any data transmitted by the Modem is also received by the Modem.
ERRORS: NONE

FE(NULL OPERATION)
ACTION: AS IN I/O BUS SPECIFICATION
ERRORS: NONE

FF(RESET BUS)
ACTION: AS IN I/O BUS SPECIFICATION
If the device is open it will be closed and the peripheral will reset thru its initialization routine, saving only the

parity and data bits options.

ERRORS: NONE

SECTION 6

I/O ERROR CODES

CODE	DECIMAL	MEANING
00	0	Everything is ok
01	1	Device/file option error
02	2	Error in attributes byte
03 *	3	File not found
04	4	File/device not open
05	5	File/device already open
06 *	6	Device error
07 *	7	EOF error
08	8	Data/file too long error
09 *	9	Write protect error
0A	10	It wasn't me
0B *	11	Directory full error
0C	12	Buffer size error
0D	13	Unsupported command error
0E	14	File not open for write
0F	15	File not open for read
10 *	16	Data error (checksum failure in device)
11	17	File type (relative/sequential) not supported
13	19	Append mode not supported
14 *	20	Output mode not supported
15 *	21	Input mode not supported
16 *	22	Update mode not supported
17 *	23	Internal or display type file not supported
18 *	24	Verify error
19	25	Low batteries in peripheral
1A *	26	Uninitialized media
1B *	27	Peripheral bus error
20 *	32	Media full
50 **	80	Data lost due to buffer overrun
51 **	81	Parity error
52 **	82	Framing error
FE *	254	Call to bus master from slave application error
FF	255	Bus time out error

A * indicates that this error is not used in the Modem.
A ** indicates a special Modem error.

SECTION 7

Specifications

7.1 Mechanical

7.1.1 Top Case.

Injection molded thermoplastic part to provide upper portion of enclosure and battery well for 4 AA batteries.

7.1.2 Bottom Case.

Injection molded thermoplastic part to provide mounting for PCB and lower portion of enclosure.

7.1.3 Front Case.

Injection molded thermoplastic part to provide mounting for two switches and an LED and provide the front portion of enclosure.

7.1.4 Backplate.

A metal plate with openings for an AC power connector, two Hex-bus connectors, and two telephone modular connectors.

7.1.5 Feet.

Elastomer strips which attach to the bottom case and provide cushioning to reduce vibrations and to prevent unwarranted movement on a working surface.

7.2 Connectors

7.2.1 AC Power Connector.

The Modem includes a connector for the AC power adapter.

7.2.2 Hex-bus Connectors.

The Modem includes two bus connectors wired in parallel. The connectors are Berg type 65945-208. Connector pin layout is shown below (looking into connector).

```
               __
              |  |
     ______        ______
    |  4    3    2    1  |
    |                    |
    |  8    7    6    5  |
     ____________________
```

Terminal		Signal
1	GRY	DO
2	YEL	D1
3	RED	BAV
4	ORN	GND
5	BRN	HSK
6	GRN	FUT
7	BLK	D2
8	BLU	D3

Pin 6 is not used, but they should be tied together from one connector to the second connector.

7.2.3 Telephone Modular Connectors.

Two telephone modular connectors (RJ11C) are included on the Modem. One jack is used to connect the Modem to the phone line using the modular phone cord provided and the other is a spare jack which may be used to connect a phone for dialing up the

distant modem.

7.3 Switches

7.3.1 ON/OFF Switch.

The Modem has an on/off slide switch located on the left front of the enclosure. It will also serve to reset the Modem.

7.3.2 Telephone Connect Switch.

The Modem has a slide switch located on the back of the case. This switch is used to connect the Modem to the phone line. This connection is made after a call has been placed with a telephone and the telephone may then be hung up. In Answer mode the switch is used to connect the Modem when the phone rings (without picking up the telephone).

7.4 Bus Activity Indicator

A yellow LED (LITE-ON LT-3258A) located below the ON/OFF switch on the front panel is used to indicate Hex-bus communication.

7.5 Device Code Jumper Options

The Modem is normally configured to respond to device code 70 (hex 46). However, two jumpers on the PCB are provided to change the device code in order to use more than one Modem on the bus. Cutting one or both of the jumpers changes the device code to one of the following codes:

	Device Code Decimal	Device Code Hex
Normal	70	46
Options with Jumpers Cut	71	47
	72	48
	73	49

7.6 Battery Operation

The Modem operates from 4 AA alkaline batteries when used as a portable device. Under conditions of continuous communication it will operate for at least 30 hours on a set of alkaline batteries.

The Modem provides a low battery indication to the computer to warn of poor battery condition. When the OPEN command is sent to the Modem, a test of the battery voltage is made and if a low reading is discovered, error code 25 (>19) is sent to the computer.

Power to the Hex-bus pullups and IBC bus control circuit must be maintained between 4.5 and 5.5 volts.

7.7 AC Adaptor Operation

The Modem may be operated from the AC adaptor which is sold separately. The Modem is designed so that when the AC adaptor is used batteries are not necessary. However, if batteries are in the Modem, they will not be discharged while the Modem is used with the AC adaptor.

Power to the Hex-bus pullups and IBC bus control circuit must be maintained at 5.5 volts when the AC adaptor is used.

The AC adaptor provides 6 volts DC regulated power to the Modem.

7.8 Packaging

The Modem, telephone cable, Hex-bus cable, and manual will be packaged in a cardboard box and appropriate insert to meet Texas Instruments Consumer Products Group document QA 10237, Group A.

HEX-BUS 80 COLUMN PRINTER
PRODUCT SPECIFICATION

Texas Instruments Incorporated
Consumer Products Group
Calculator and Compact Computer Products
5/20/83
Revision 1.3

SECTION 1

Introduction

1.1 Purpose

This document describes the features and operational aspects of a printer peripheral. The interface of this device to the computer is described.

1.2 Applicable Documents

1. Intelligent Bus Controller Spec #1052911
2. Intelligent Peripheral Bus
3. QRAS 10237
4. MIL STD 105
5. QRAS 10348
6. QRAS 10332
7. I/O Cable Specification

SECTION 2

Product Overview

2.1 Concept

The printer is a portable accessory to a computer. It is a low power, low cost printing device.

2.2 Features

*Print mechanism : Brother EP-20
... Print method : Thermal Ink Transfer
... Printing speed : 20 characters/second maximum
... Characters/line : 80
... Printing area : 190.4 mm

*Paper
... Width : 8.5 inches

*Character set
... Characters : 96 ASCII

*Supply voltage
... 4 D Cell Batteries (Alkaline)
... AC adapter 120 VAC input

*Physical
... Size : 318mm(W) X 163mm(L) X 48mm(H)
... Case : Injection molded painted thermoplastics

*Connectors
... Interface to calculator : 2 ea Berg 65945-208 or equivalent
... AC adapter :

SECTION 3

Operation

The printer is initialized on power up and on reset. A line feed and carriage return are performed when the printer is turned on. It prints characters corresponding to the ASCII formatted input data. It also accepts control codes. The character set and control codes are shown in Figure 1. All control codes from >00 to >1F not defined in Figure 1 will be ignored. Character codes >7F to >FF will print a space.

If a specified key sequence is used when the printer is first turned on, a self test consisting of printing all 96 ASCII characters will be executed.

lsb\msb	0	1	2	3	4	5	6	7
0			sp	0	@	P	`	p
1			!	1	A	Q	a	q
2			"	2	B	R	b	r
3			#	3	C	S	c	s
4			$	4	D	T	d	t
5	2 L		%	5	E	U	e	u
6			&	6	F	V	f	v
7	1 L		'	7	G	W	g	w
8			(	8	H	X	h	x
9			)	9	I	Y	i	y
A	LF		*	:	J	Z	j	z
B	LU		+	;	K	[	k	{
C			,	<	L	\	l	¦
D	CR		-	=	M	]	m	}
E	1/2LF		.	>	N		n	~
F	1/2LU		/	?	O	_	o	

FIGURE 1 CHARACTER SET

Control codes:

2L, 1L - These are mode select codes which set the printer to advance 1 or 2 lines when a LF code is executed. For example, if L=D is encountered in the open statement or control code 2L is received, all LF commands will advance the paper 2 lines (including automatically added CR/LF at the end of data records). It will remain in effect until a 1L code is encountered or until a close command. The option L=[D,S] in the open statement and these control codes have the same effect. One line advance is the default in the open command.

LF Line Feed - Advances paper one line

LU Line Up - Moves paper one line in opposite direction of LF.

CR Carriage Return - Returns printer to left-most position.

1/2LF 1/2 Line Feed - Advances paper 1/2 line.

1/2LU 1/2 Line Up - Moves paper 1/2 line in opposite directi… of LF.

The printer will automatically generate its own Carriage Return/Line Feed to segment a data record of printable characters that exceeds character line length or when its 80 byte buffer is full. In addition, all control codes except 2L and 1L will cause the buffer to print.

As an option the printer will also generate a Carriage Return/Line Feed after each data record transmission. This option will be selected in the open statement from the computer. See BUS MESSAGE STRUCTURE.

PROGRAM EXAMPLES

Program

```
OPEN #1,"16",OUTPUT
PRINT #1,"LOW COST PRINTER"
PRINT #1,"80 COLUMNS"
CLOSE #1
```

Characters Printed

```
LOW COST PRINTER
80 COLUMNS
```

Program

```
10  OPEN #1,"16,R=N",OUTPUT
20  FOR I=33 TO 127
30  PRINT #1, CHR$(I);
40  NEXT I
50  CLOSE #1
60  END
```

Characters Printed

```
!"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\] _`abcdef
qrstuvwxyz{|}~
```

SECTION 4

Bus Interface

4.1 General

The printer communicates with the computer via a peripheral bus. The bus supports communications between the computer and other peripherals in addition to the printer.

Transmissions on the bus are defined in the context of a "message frame" which consists of a command message from the computer and one or more response messages from the peripheral.

SECTION 5

Bus Signal and Timing Description

5.1 Signals

The physical bus consists of 8 lines. There are 4 parallel data lines defining a 4 bit nibble as the basic unit of information carried on the bus. Data within the communication protocol is defined in 8-bit units (bytes). Each byte corresponds to two transmissions on the bus, least significant nibble first. Two more lines used in the communication are HSK and BAV, both explained later. The last two lines include a ground referance signal and a line reserved for future use. Data is output to the bus via open drain drivers.

D3	————————— most significant data bit
D2	————————— data bit
D1	————————— data bit
D0	————————— least significant data bit
HSK	————————— handshake
BAV	————————— bus available
FUT	————————— reserved for future use
GND	————————— ground

The speed of transmission of the data bus is controlled by the handshake line HSK. The BAV (bus available) signal is used to designate the beginning of a command message from the computer.

5.2 Handshake Timing

The signal timing of HSK and the data lines is illustrated in Figure 2. The falling edge of HSK is the signal to receiving devices that a nibble of data is available on the bus. The rising edge of HSK is the signal to the transmitting device that all receivers have read the data. HSK is an open drain line so that any one device may hold it at a low level.

When the receiving device(s) see that HSK has gone low they rapidly (through hardware) pull HSK low also. The receiving device(s) then hold HSK low until they have processed the data.

The transmitting device will release HSK shortly after pulling it low. This normal interaction is illustrated in Figure 3. If the transmitting device is slower than the receivers then it may dictate the bus speed as shown in Figure 4.

When a device is not interested in the data being transmitted it may disable itself from the bus and wait for the next message frame (denoted by BAV going low). Once a device has disabled itself from the bus, it need not participate in the handshake activity.

Figure 2
Bus Handshake Sequence

```
          __________                               __________
                    \                             /          \
  HSK                \___________________________/            \____________
                     |                           |             |
                     |<- one handshake -->|                    |<- next --
          ___________________________________________________________________
          XXXXXXXXXX                            XXXXXXXXXXX
  D0-D3   XXXXXXXXXX____________________________XXXXXXXXXXX__________
                     |<--- data valid -->|
```

Figure 3
Handshake Components
(Receiver Limited)

```
                  __________                               __________
                            \                             /          \
  HSK(composite)             \___________________________/            \____________
                             |                           |             |
                             |<- one handshake -->|                    |<- next --

                  __________                ____________________________
                            \              /                            \
  HSK(xmit)                  \____________/                              \__________

                  __________                               ____________
                            \                             /            \
  HSK(rcv)                   \___________________________/              \__________
```

Figure 4
Handshake Components
(Transmitter Limited)

```
                 ________                      ________
                         \                    /        \
HSK(composite)            \__________________/          \__________
                          !                  !           !
                          !<- one handshake ->!          !<- next --

                 ________                      ________
                         \                    /        \
HSK(xmit)                 \__________________/          \__________

                 ________             _________________
                         \           /                  \
HSK(rcv)                  \_________/                    \__________
```

Table 1
Handshake Timing Parameters
(Microseconds)

Item	Min	Max
HSK low to data valid	-	.5
HSK low(xmit) to HSK low(bus)	-	3
HSK low(bus) to HSK low(rcv)	-	5
HSK low(xmit) to HSK high(xmit)	8	-
HSK high to data invalid	0	- **
HSK high to HSK low	8	20000 *

* within a message

** because data is output via open drain buffers, and HSK going high causes ones to be written into their output latch, data becomes invalid on the rising edge of HSK.

SECTION 6

Bus Message Structure

6.1 Protocol

As mentioned earlier, the data bus transmits command and response messages within the context of a message frame. In general the transmission of one command message from the computer will cause one or more response messages to be transmitted back from the peripheral device selected. Each message consists of several nibble transfers as described in the previous section.

Each transmitted message contains overhead information to indicate such things as the selected device, the command code to be performed, and data length. The BAV (bus available) signal specifies the start of a message frame. When the computer starts a message frame it first pulls BAV low. The command message from the master then follows that falling edge of BAV. The falling edge of BAV alerts all peripherals to look for the two-nibble device code which is always transmitted first in the command message (least significant nibble then most significant nibble). The BAV signal does not return to the high level until the message frame is complete. This is illustrated in Figure 5 below.

Figure 5
Message Frame

```
other   |    command     |    response    |
 lines  |     message    |     message    |

 _____                                       ________
      \                                    /
BAV    \________//_________________//________/
               //                  //
```

Table 2
Message Timing Parameters

Item	Min	Max
BAV low to HSK low	5	20000
HSK high to BAV high	1	—
End of command to start of response	10	—
HSK high to HSK low	8	20000
BAV high to BAV low	8	—
(Within a message frame)		

The first two nibbles of the command message always contain the device code of the peripheral to be addressed. All devices on the bus will read this number and test for a match. After the device code has been sent all devices except the one selected will ignore all further data in the message. The hardware will be designed such that they will not have to participate in the handshake sequence until the next falling edge of BAV.

Any device may extend the time to process data or wait for an operation to complete by holding HSK low until it is ready to start the next operation. Whenever HSK is high during a message, it must go low within 20ms or the receiver will time out.

The command and response messages are detailed in a following section.

As mentioned previously each device has a unique device code. In addition, the device code >00 is reserved to be recognized by all devices, but only the null and reset commands are valid. All others should be ignored by the peripheral and no response message should be sent.

6.2 Command Message

The standard command message is divided into two distinct transmission phases. The first phase is the command transmission. The information necessary to the peripheral (including the data) is transmitted to the peripheral. The second phase consists of the peripheral transmitting its response to the command (In some cases, there will be no response to the command).

The following data is contained in a standard command message:

Field	Bytes
Device code	1
Command code	1
Logical Unit Number	1
Record number	2
Buffer length	2
Data length	2
Data	variable

The fields in a command message are described below.

6.2.1 Device Code.

This selects the peripheral which is to respond to the command. The printer peripheral has been assigned device codes 16 and 17 (decimal). Each printer on the bus must have a unique code. Normally the code is 16. A jumper wire on the PCB may be cut to change the device code to 17. Two printers may be used if one is set to 16 and the other to 17. Device code 17 may be removed if not easily available on the microprocessor.

All devices on the bus will respond to device code 00, but will not return a response message.

6.2.2 Command Code.

This field tells the peripheral the nature of the operation to be performed. The following lists the standard code assignments to be supported by the printer (hexadecimal).

>00 - open
>01 - close
>04 - write data
>07 - return status
>0A - service request poll
>FE - null operation
>FF - bus reset

The specific actions of each of these commands is described later.

6.2.3 Logical Unit Number - LUNO.

This is reserved. This field should be ignored.

6.2.4 Record Number.

This is reserved. This field should be zero.

6.2.5 Buffer Length.

This field indicates the size of the data buffer for receiving data from a peripheral during the current bus operation. If it is not large enough to receive all the data to be returned by the printer, a buffer length error (>0C) should be returned instead. This length is exclusive of the data length and return status bytes which form part of the response message.

6.2.6 Data Length.

This field gives the number of bytes of data which follow in the data field.

6.2.7 Data.

This field contains data to be written to the peripheral device. The use of the data depends on the command code. If the data length field is zero then the data field is not present.

6.3 Response Message

The response message contains the following data

field	bytes
Data length	2
Data	variable
Operation status	1

6.3.1 Data Length.

This field specifies the number of bytes of data which follow in the data field and must be less than or equal to the buffer length in the command transmission.

6.3.2 Data.

This field contains the data to be returned to the master device; for example on an open operation. If the data length field is zero then this field will be omitted. Whenever the true data length cannot be determined at the time the length is sent, the data field will be padded with trailing zeroes for internal type files or devices, and with trailing blanks (>20) for display type files or devices.

6.3.3 Operation Status.

This field contains a status of the operation. The following lists the assigned response codes to be supported by the printer (hex).

>00 - normal operation completion
>01 - incorrect device option
>02 - error in attributes byte
>04 - file/device not open error
>05 - file/device already open
>08 - data length too large
>0A - did not request service
>0C - buffer size error
>0D - unsupported command error

6.4 Standard Access

The access to a printer is performed with a sequence of I/O calls. Before using the printer it should be opened using the open command. This may be followed by other I/O calls to write data or perform other functions. When the computer finishes using the printer it will issue a close I/O call. This will ensure that any necessary device dependent actions have been performed. The bus reset command code (>FF) will also close the printer if it is open.

6.5 Command Descriptions

This section describes the Command message setup and device response for the various standard I/O command codes.

6.5.1 Open - 00.

This command code is used to initiate the use of a peripheral. The printer will check access modes and ensure that it is not already open. If it is open or any other error occurs while the open is being attempted, then the appropriate error will be issued and the open will not occur. Otherwise, the printer will be opened and reset to its initialized state. The command message will be set up as follows:

field	data
Device code	as required
Command Code	00
Logical Unit Number	don't care
Record number	don't care
Buffer length	as required (at least 0004)
Data length	as required (0004 returned)

The data buffer contains the following which is sent to the peripheral.

I/O buffer length (2 bytes)
(if zero then device returns buffer size)
device attributes (1 byte)
device characteristics (if any)

The printer will compare the 'I/O buffer length' to its capabilities and return either the requested length or the default length if the requested is zero (Default for the printer is >0050). The I/O buffer length is used by the master to determine what size buffer should be allocated by the master for read operations. It differs from the buffer length field in the PAB in that the buffer length field in the PAB only indicates the size of the buffer for the current bus operation.

The device attributes byte contains flags used to indicate the access mode of the peripheral. Several bits are unused by the I/O scheme and may be used by the application software as desired. Other bits must be set to zero to allow compatibility with future peripheral protocol enhancements.

The bit definitions are as follows (bit 0 is the least significant bit):

```
    7-6 - access mode
          00 - invalid mode
          10 - output mode (write only - MUST be in this mode)
          01 - invalid mode
          11 - invalid mode
      5 - MUST be zero(0) - otherwise error
                   (relative/sequential file type)
      4 - don't care (fixed/variable)
      3 - don't care (internal/display type)
      2 - don't care (reserved for device dependent use)
    1-0 - don't care (these two bits may be used as desired)
```

The device characteristics field is a variable length field which contains device-dependent information relative to setting up the device.

Printer device dependant features:

R=[N or L]

R=N causes the printer to NOT issue a carriage return and line feed at the end of each write command.

R=L (default if omitted) causes the printer to issue a carriage return and line feed following each write command.

CR Is the same as R=N and caused the printer to NOT issue a carriage return and line feed after each write command.

L=[D or S]

L=D (double spacing) causes the printer to advance the paper 2 lines each time the LF command is encountered. It is the same as receiving a 2L control code to the printer.

L=S (single spacing) caused the printer to advance the paper 1 line each time the LF command is encountered. It is the same as receiving a 1L control code to the printer. This is the default for the open if the L option is omitted.

Examples:

```
10 OPEN #1,"16",OUTPUT
    causes the printer to issue a carriage return and
    one line feed after each write command.

10 OPEN #1,"16.R=N",OUTPUT or 10 OPEN #1,"16.CR",OUTPUT
    causes the printer to never issue carriage return
    or line feed unless the end of the  line has  been
    passed.  Each line feed performed will advance the
    paper 1 line.

10 OPEN #1,"10.R=L,L=D",OUTPUT
    causes the printer to issue a carriage return and
    line feed after each write command.  Each line
    feed performed will advance the paper 2 lines.
```

The response buffer will contain the accepted buffer length, and the record number that the file was opened to. For the printer, this record number is be meaningless, so a zero (00) will be returned. This information is always returned. Thus the response message for a successful open will be:

```
Data length          0004
Data           (2 bytes) Accepted buffer length
               (2 bytes) always >0000
Operation status       00
```

An unsuccessful open may not return data in the response message. The operation status byte may contain the following error codes (hex):

```
>00 - successful open
>01 - incorrect device option
        * Open options not correct
>02 - Invalid attributes
        * Attributes byte (possibly including I/O
          buffer length) not included in data
        * Open mode not output
        * Open mode not sequential
>05 - file/device already open
>0C - buffer size error (Given if the buffer size in
      the PAB < 0004)
```

6.5.2 Close - 01.

This command terminates the use of a device. Depending on the device this command may be used to clean up internal data (e.g. write an end of file) or may be effectively ignored. In general a close command must be sent between using a device and another open command. The data length for the close command will be zero (no data buffer is transmitted). The PAB should be set up as follows:

field	data
Device code	as required
Command Code	01
Logical Unit Number	don't care
Record number	don't care
Buffer length	don't care
Data length	0000 (0000 returned)

The response message will only contain a status byte and a zero data length (two bytes). The error status indications are (hex):

>00 - device or file closed
>04 - device or file never opened
>08 - data too long
    * - The printer gives this error if data is transmitted to it in the close command

6.5.3 Write Data - 04.

This command is used to send data to a peripheral device. The command message will contain the data to be sent to the device. The PAB should be set up as follows:

field	data
Device code	as required
Command Code	04
Logical Unit Number	don't care
Record number	don't care
Buffer length	don't care
Data length	record length (0000 returned)

The response message will contain zero-length data and the operation status. The following error status indications may occur (hex):

>00 - write successfully completed
>04 - file/device not open

6.5.4 Return Status - 07.

This command is used to return device status information. The information is returned in the data buffer. Certain bit fields in the return data are assigned to standard meanings while others are reserved for device dependent extensions. Certain devices may return more bytes of status if the buffer length allows. The PAB should be set up as follows:

field	data
Device code	as required
Command Code	07
Logical Unit Number	don't care
Record number	don't care
Buffer length	>= 0001
Data length	0000 (0001 returned)

The bit fields in the return data are as follows (bit 0 is the least significant bit).

- 7 - always zero(0)
- 6 - always zero(0)
- 5 - always zero(0)
- 4 - 1 if open 0 if closed
- 3-2 - always zero(00)
- 1-0 - always two(10)

The following error status indications may occur (hex):

- >00 - status returned
- >08 - data too long (data length was not 0)
- >0C - buffer size error

6.5.5 Service Request Poll - 0A.

This command allows a bus master to query a peripheral as to whether it requested service from the master. The PAB should be set up as follows:

field	data
Device code	as required
Command code	0A
Logical Unit Number	don't care
Record number	don't care
Buffer length	as determined in open
Data length	0000 (data length returned)
Return status	0A

The response message will consist of zero or more bytes of data, and the return status. The following error status indications may occur:

0A - it wasn't me (unsuccessful poll)

* Any other error code indicates a successful poll operation, and reflects the reason for the service request. Thus even devices that don't support service requests need to return the "it wasn't me" status code.

6.5.6 Null Operation - FE.

When the calculator receives a BAV interrupt and either no devices are enabled for interrupts or the current service flag is set, then a null operation code is sent to all devices. There will be no response to this message

field	data
Device code	00 or printer device code
Command code	FE
Logical Unit Number	don't care
Record number	don't care
Buffer length	don't care
Data length	00

6.5.7 Reset Bus - FF.

It may sometimes be desired to tell a device (or devices) to close all open files (or devices). This command will have no response by devices, they will simply perform the action requested (If they were not open to begin with, then they will do nothing). The printer will initialize the print head and revert to a 'closed' status. The PAB should be set up as follows:

field	data
Device code	00 or printer device code
Command code	FF
Logical Unit Number	don't care
Record number	don't care
Buffer length	don't care
Data length	00

* For both the Null operation and Reset bus commands, no response will be forthcoming.

SECTION 7

Specifications

7.1 Connectors

The printer includes a connector for the power adapter.

The printer includes two bus connectors wired in parallel. The connectors are Berg type 65945-208 or equivalent. Connector pin layout is shown in Figure 6 (looking into connector).

```
              __
             |  |
     ______        ______
    | 4    3    2    1 |
    |                  |
    | 8    7    6    5 |
     ___________________
```

Terminal	Signal
1	D0
2	D1
3	BAV
4	GND
5	HSK
6	FUT
7	D2
8	D3

Figure 6
Bus Connector Layout

7.2 Switches

The printer has an on/off switch. It will also serve to reset the printer.

There will be 2 push button switches, one for line feed and one for reverse line feed (line up). They cause continuous paper feed until released.

The printer has an internal jumper to allow user programming of the least significant bit in the device code (decimal addresses 16 and 17).

7.3 Packaging

The printer, AC adaptor, hex-bus cable, and manual will be packaged in a cardboard box and appropriate insert.

Hex-bus Microtape Peripheral
Design Specification

Consumer Products Group
Calculator Division

September 20, 1983

Revision 2.1

SECTION 1

Introduction

1.1 Purpose of This Document

This specification describes the features and operational aspects of a microtape peripheral for the hex-bus peripheral system. This document is meant to be used in formulating the detailed software design and to afford an accurate review of the product features.

1.2 Scope of This Document

This document describes the features of the product which are apparent to the user of a computer using the hex-bus. The interface from the BASIC language is used as an example throughout this specification. The electrical design is not discussed in detail although proposed gross concepts are presented. The interface of this (and other) peripherals to the peripheral bus is discussed in other documents.

1.3 Terminology

wafer, drive, drive controller, peripheral bus, file, record, directory, header, BOT/EOT, MRL, LUNO, DL, access mode, file type,

1.4 Related Documents

The following documents are referenced in the text or may contain other material of interest to the reader of this document. Copies of any of these may be obtained by contacting Art Hunter at TI Lubbock; phone 796-3431; MSG HEXB.

Advanced Language Calculator Product A Functional Specification.

Advanced Language Calculator Microtape Peripheral Functional Specification.

Hex-bus Intelligent Peripheral Bus Structure, Timing, and Protocol Specification.

Lonestar Console Intelligent Peripheral Bus Software Design Specification.

SECTION 2

Product Overview

2.1 The Microtape Concept

The microtape peripheral is a magnetic tape drive which uses a small (1/8 inch), endless loop, magnetic tape called a wafer. The wafers vary in length from 5 to 75 feet. Tape speed is about 8 inches per second and the motor only runs during wafer access. Data density amounts to about 1.5K bytes per foot. However, the capacity of a wafer is reduced significantly by a file diectory, synchronization patterns preceeding data records and motor start/stop time considerations. Thus the maximum access time and the amount of data storage can be traded off by choosing tapes of a certain length. Each tape has a reflective Beginning Of Tape marker to form a reference point on the tape. The bit-level recording scheme uses a single track with ones and zeros recorded as shown below. The transitions key the timing and the levels determine the data content. Every bit has a transition in the middle. The scheme for reading data requires that this transition be found and the data level following the transition is read as a bit of data. The data level read from the wafer will be the inverse of what was written so that the logic levels written must be the inverse of the desired read waveform. It should be noted that transitions do not have to be present at a bit edge and must be ignored when they do occur. For this reason, once the bit time is known and a mid-bit edge found, a 2/3 bit-time wait must separate tests for the next mid-bit edge.

Read Data Patterns: high to low mid-bit transition is a 0
low to high mid-bit transition is a 1

```
    |< Zero  >|                    |<  One  >|

         ____                                ____
////    |    |    ////        ////      |    |    ////
////    |    |____////        ////______|    |    ////

    |<  0  >|<  0  >|<  1  >|<  1  >|<  0  >|
         ____    ____          ____     _________
////    |    |    |    |            |    |    |         |    ////
////    |    |____|    |____________|    |____|         |____////
```

```
         |------*  |------*  |------*  |------*
  * timer interrupts every 2/3 of a bit-time starting from a mid-bit
      (the interrupt occurs 1/6 of the way into every bit)
```

2.2 Product Configuration

The product will consist of one microtape controller/drive which connects to the Hex-bus intelligent periperal bus. Up to eight drives may be connected to the bus. Each drive on the bus must have a unique device code selected by two switches and a jumper inside the microtape cover. Device codes 1 through 8 in the Intelligent Peripheral Bus specification have been allotted for microtapes.

A microtape drive will consist of the following major parts:

* 70C20 microcomputer
* Drive electronics
* Drive mechanics
* Intelligent bus controller
* Power supply

```
  peripheral bus ______________________
                   |
                   |
        ,_____________,    ,_________,     ,_______,
        | intelligent |----|micro p  |----| drive |
        |    bus      |    |         |    |       |
        | controller  |    |         |    |       |
        |_____________|    |_________|    |_______|
                      \         |        /
                       ,_____________,
                       |power supply|
                       |____________|
```

2.3 Data Format

Each microtape wafer will contain a file directory which will serve as an index to all files on the wafer. Each wafer will be able to support up to 16 active or inactive files. Each file will be represented in the directory by the the following information:

An active/inactive flag to indicate whether the file is active, deleted or superceeded.

A last/not last flag to indicate whether or not the file is the last active file on the wafer. This also indicates the capacity to overwrite the file and/or open the file in certain modes. For an existing active file to be opened in the update mode, it must be the last file on the wafer.

A data type flag for internal or display format

The number of records contained in the file.

The length of the longest record in the file.

Each file has a filename associated with it. This filename occurs only in the file header record at the beginning of the actual file. The information in the directory is used to locate the beginning of each active file at which time the filename contained in the file header can be compared to the filename being searched for to determine if there is a match. This can be be repeated until the file is found or a last file flag is found in the directory.

The first time a file is accesssed, the directory is read into controller RAM. As various operations are performed on the file, the directory will be updated at certain points of access if the contents of the wafer are modified.

Each record in a file has a variable length format, and carries a 16 bit data length field. Each record in a file, including the file header record, contains the ordered number of the file as it occurs in the directory and the ordered number of the record as it occurs in the file.

Every record on the wafer is separated from each adjacent record by a record gap which will provide for motor stop and start times between records. A synchronization data field will occur at the start of each record to indicate the beginning of valid data and provide tape speed calibration and acceptance.

At least 2 inches of tape on either side of the BOT/EOT mark must be regarded as unusable due to vendor assembly techniques.

Wafers may be write protected by removing a reflective sticker on the cover of the wafer. The presence of this sticker is detected by software to control whether or not the wafer may be written to.

2.4 General File Handling

Any currently non-existent file opened for output will become the last file on the wafer.

Any currently existing file opened for output will become inactive and will be recreated as the last file on the wafer.

If at any time the last file is deleted, the last file status will revert to the last active file. Any inactivated files following the last active file will no longer be recognized and may be overwritten.

Any records contained in a file which is subsequently opened for output or deleted will become inaccessible to normal I/O operations. They must be transferred before the file becomes inactive if the application wants to keep them.

Inactivated files may be opened by file number and the data records may be read if the file occurs before the last active file. Inactive files may never be opened by file name since the same name may occur later in an active file.

Example Of Data Format On Wafer

```
                 ----------------
                !BOT MARK         ! BEGINNING OF TAPE MARKER
                !-----------------!
        :       !DEAD SPACE       ! LEAVE 2" OF DEAD SPACE BY BOT
        :       !-----------------!
      Tape      !SYNC             !
     Travel     !-----------------! REV. 0 TO ALLOW DRIVES TO BE
        :       !DIRECTORY  REV.0! BACKWARD COMPATIBLE
        :       !-----------------!
        :       !ACT/DISP/3/100   ! 1ST: ACTIVE, DISPLAY,
                !-----------------!      3 REC.S, 256 MAX REC LEN
                !INACT/DISP/3/F   ! 2ND: INACTIVE, DISPLAY,
                !-----------------!      3 REC.S, 15 MAX
                !ACT/INT/4/9      ! 3RD: ACTIVE, INTERNAL,
                !-----------------!      4 REC.S, 9 MAX
                !ACT/LST/DISP/4/F! 4TH: ACTIVE, LAST FILE, DISPLAY,
                !-----------------!      4 REC.S, 15 BYTES MAX
                !                 !
```

```
!----------------! 4TH FILE WAS CREATED WHEN 2ND
!                ! FILE WAS OPENED FOR OUTPUT
!-------           AFTER 3RD FILE HAD BEEN CREATED.
                   THE 3 RECORDS IN THE 2ND FILE
                   MAY HAVE BEEN SAVED BEFORE THE
                   FILE WAS OPENED SO THEY COULD BE
          -----!   TRANSFERRED
!                !
!----------------!
!                ! END OF DIRECTORY SPACE
!----------------!
!LONG SYNC       ! LONG SYNC FOR REWRITING DIRECTORY
!----------------!
!FILE1           ! FILE NAME OF 1ST FILE
!----------------!
!SYNC            !
!----------------!
!RECORD #1       ! 1ST RECORD OF 1ST FILE
!----------------!
!SYNC            !
!----------------!
!RECORD #2       ! 2ND RECORD OF 1ST FILE
!----------------!
!SYNC            !
!----------------!
!RECORD #3       !
!----------------!
!SYNC            !
!----------------!
!FILE2           ! FILE NAME OF 2ND FILE
!----------------!
!SYNC            !
!----------------!
!RECORD #1       ! 1ST RECORD OF 2ND FILE
!----------------!
!SYNC            !
!----------------! EVEN THOUGH THIS FILE IS
!RECORD #2       ! INACTIVE IT'S RECORDS AND
!----------------! FILE NAME REMAIN INTACT.
!SYNC            ! THE DIRECTORY WILL NOT ALLOW
!----------------! THESE RECORDS TO BE ACCESSED
!RECORD #3       ! BY NORMAL MEANS SINCE THE FILE
!----------------! MAY HAVE BEEN RECREATED
!SYNC            ! ELSEWHERE
!----------------!
!FILE3           ! FILE NAME OF 3RD FILE
!----------------!
!SYNC            !
!----------------!
!RECORD #1       !
```

```
!-----------------!
!SYNC             !
!-----------------!
!RECORD #2        !
!-----------------!
!SYNC             !
!-----------------!
!RECORD #3        !
!-----------------!
!SYNC             !
!-----------------!
!RECORD #4        !
!-----------------!
!SYNC             !
!-----------------!
!FILE2            !  FILE NAME OF 4TH FILE
!-----------------!
!SYNC             !
!-----------------!  THIS IS AN ACTIVE FILE HAVING
!RECORD #1        !  THE SAME FILE NAME AS 2ND FILE.
!-----------------!  IT WAS CREATED WHEN THE ORIGINAL
!SYNC             !  FILE2 WAS OPENED FOR OUTPUT
!-----------------!  AFTER FILE3 HAD BEEN
!RECORD #2        !  CREATED
!-----------------!
!SYNC             !
!-----------------!
!RECORD #3        !
!-----------------!
!SYNC             !
!-----------------!
!RECORD #4        !
!-----------------!
!                 !
!-----------------!
!                 !
!------


            -----!
!                 !
!-----------------!
!                 !
!-----------------!
!DEAD SPACE       !  LEAVE 2" OF DEAD SPACE BY EOT
!-----------------!
!EOT              !  END OF TAPE MARKER
!-----------------!
```

Individual Record Structure

```
 ------------------
!INTER-RECORD GAP!
!----------------!
!SYNC            !
!----------------!
!START NIBBLE    !
!----------------!
!FILE NUMBER     !
!----------------!
!RECORD NUMBER   !
!----------------!
!DATA LENGTH     ! NOT PART OF DIRECTORY
!----------------! OR FILE HEADER RECORDS
!DATA            !
!----------------!
!CHECKSUM        !
!----------------!
!INTER-RECORD GAP!
!----------------!
```

SECTION 3

Input/Output Interface

This section describes the software interface to the microtape functions. Specifically the actions taken for each valid I/O command code are described as well as such ancillary subjects as error codes.

3.1 Capabilities

Each wafer contains up to 16 files composed of one or more data records of varying lengths. Any record length up to 32767 bytes may be supported by the microtape as the data rate of the bus will exceed the data rate of the drive and the record need not be buffered in memory.

The group of records following a file header define a sequential file. Random record access is not supported. The last record written will define the end of file. Up to 4095 records may be written to a file. In the update mode, write operations must be followed by a restore (or a close and open)before a read can be done. In update mode, a read may be followed by a write operation which will maintain the preceding part of the file read and set a new End Of File.

The microtape controller recognizes device codes 01 through 08 as indicated by the device select switches

3.2 Standard Command Actions

This section discusses the controller response to various command messages. Both standard and special command codes are discussed. The PAB construction examples show the use for the microtape subset of the I/O standards. The examples <u>do not</u> necessarily conform to the standards presented in the Peripheral Bus Specification. Differences for the microtape primarily consist of don't care fields in certain bytes of the PAB which have defined values for standard access. Fields which subset the standard are noted with an asterisk (*) in the examples.

Between commands a microtape drive will be idling, waitng for the next command. The motor will be stopped, the front panel LED will be off, and the bus interface will be enabled.

The following general actions take place in the microtape during every command:

*Initialize interrupts and stack, reset drive controls
*Enable bus communication at start of next message
*Wait for Start Of Message (SOM)
*Turn on sensors and LED
*Read device code and compare to microtape device number
    if microtape selected, then continue
    if not selected, then restart sequence
*Read remainder of command PAB
*If NULL command, then restart sequence
*Test for wafer write protect and set flag
*Decode command and execute
*If unsupported command, then ERROR 13

*If, while reading, the drive loses its position on the wafer, then ERROR 6
*If, while writing, the end of tape mark is detected, then ERROR 32

3.2.1 Open - 00. This command code is used to initiate access to a particular file on a drive. The PAB should be set up as follows:

field	data
Device code	as required
Command Code	00
Logical Unit Number	don't care* (LUNO)
Record number	don't care*(next returned)
Buffer length	>= 0004
Data length	>= 0004 (0004 returned)
Return status	(returned)
Buffer address	as required

The data buffer will contain the following which will be sent to the peripheral:

Buffer length (2 bytes) - the maximum requested record length for operations to the device.

Device attributes (1 byte)

Device characteristics (variable length) - file name, may contain string for name of file or a 1 digit hexadecimal file number.

The following actions take place in the microtape during an OPEN command:

```
*Read command data; buffer length,attributes
    if any data is missing, then ERROR 1
*Read filename or number
    if missing, then ERROR 1
    if filename data length gt 12, then ERROR 1
    set flag for file by name or number
*If a file is already open on drive, then ERROR 5
*Load directory from wafer
*If no directory is found on the wafer, then ERROR 26
*If not sequential file, then ERROR 2
*If opening for append, then ERROR 2
*If not opening for input, then check for write protect
    if write protected, then ERROR 9
*Search for file or end of data
*If the file is not found, then check access mode
    if output/update, then check number of files
       if room, then create new file, write file header
       if full, then ERROR 11
    if input, then ERROR 3
*If the file is found then, check access mode
    if update or input, then check file type
       if not same file type, then ERROR 23
    if update, then check file position
       if last file, then continue
       if not last file, then ERROR 80
    if input or update, then check buffer length
       if bl lt max record length, then ERROR 12
       if bl ge max record length, then ok
       if bl eq zero, then check max record length
            if mrl eq zero, then return 256
            if mrl not zero, then return mrl
    if output, then check file position
       if last file, then continue
       if not last file, then check number of files
            if room, then create new file, write header,
                     inactivate old file, init mrl
            if full, then ERROR 11
    if output, then check buffer length
       if bl eq zero, then return 256
       if bl ne zero, then maintain existing mrl
*Test for EOT while writing file header
    if EOT found, then set EOT flag, reject record,
                 dont open file, ERROR 32
*If new file was created, then set new last flag,
```

set file type flag
*If bl gt 32767, then ERROR 8
*Set open flag
*Save attributes
*Return buffer length, record number, and status

The peripheral will accept any maximum record length up to 32767 since records need not be buffered in memory when they are written or read. If the requested record length is zero; asking for the default; then 256 is returned except in input mode. In input mode the requested record length is compared to the maximum record length of the file. If the the maximum record length on the file exceeds that requested, then an error occurs. If the the length requested exceeds the length on the file then length requested is accepted. If a zero length is requested in input mode then the maximum length in the file header is returned. The device attributes byte contains flags used to indicate the access mode of the peripheral. Some bits are unused but should be set to zero to allow compatibility with future peripheral protocol enhancements. The bit definitions are as follows (bit 0 is the least significant bit).

```
7-6 - access mode
      00 - append mode (write only, at end of file)
      10 - output mode (write only)
      01 - input mode  (read only)
      11 - update mode (read or write)
  5 - relative(1)/sequential(0) (must be 0)
  4 - fixed(1)/variable(0)   (dont care)
  3 - internal(1)/display(0)
2-0 - don't care
```

Output mode specifies that data will only be written to the device and the "read data" command will not be used. Input mode specifies that data will only be read from the device and the "write data" command will not be used. Update mode means that data may be both read and written. A read operation cannot immediately follow a write operation in update mode, but must be separated by a restore. This product and the associated peripherals will only support sequential files with variable length records. Only input and update modes check the file type bit of an existing file. Opening an existing file for output allows the file to be recreated for any data type.

The response buffer will contain the accepted/corrected buffer length and the next record number if there are no errors. Thus the response message for a successful open will be;

Data length 4

Data	Max. record length & record number
Operation status	0

An unsuccessful open will return a maximum record length and record number if the failure was related to the buffer length. If there was another type of error no data will be returned. The operation status byte may contain the following error/status codes:

00 - ok
01 - device characteristics error
02 - attributes error
03 - file not found
05 - file already open
06 - device error
08 - data/file too long
09 - write protect error
0B - too many files error (11)
0C - buffer size error (12)
17 - wrong file type (23)
1A - no directory found (26)
20 - wafer full (32)
50 - not last file for append or update (80)
FF - time-out error (255)

3.2.2 Close - 01. This command terminates access to a file and rewrites the directory if necessary (after output or update mode). The PAB should be set up as follows:

field	data
Device code	as required
Command Code	01
LUNO	don't care*
Record number	don't care*
Buffer length	don't care*
Data length	0000 (0000 returned)
Return status	(returned)
Buffer address	don't care*

The following actions take place in the microtape during a CLOSE command:

*If no file is open on drive, then ERROR 4
*Reset open flag
*If update, or output mode, then check write protect flag
  if not write protected, then rewrite directory
  if write protected, then ERROR 9

*Return status

The response message will only contain a status byte and a zero data length (two bytes). The error status indications are:

00 - ok
04 - no file open
06 - device error
09 - write protect
FF - time-out error (255)

3.2.3 Delete Open - 02. This command terminates access to a file, attempts to delete the file and rewrite the directory (if the wafer is not write protected). The PAB should be set up as follows:

field	data
Device code	as required
Command Code	02
LUNO	don't care*
Record number	don't care*
Buffer length	don't care*
Data length	0000 (0000 returned)
Return status	(returned)
Buffer address	don't care*

The following actions take place in the microtape during a DELETE OPEN command:

*If no file is open on drive, then ERROR 4
*Reset open flag
*If write protected, then ERROR 9
*Inactivate file
*If file was last, then find new last file
*Rewrite directory
*Return status

The response message will only contain a status byte and a zero data length (two bytes). The error status indications are;

00 - ok
04 - no file open
06 - device error
09 - write protect
FF - time-out error (255)

3.2.4 Read Data - 03. This command is used to request a data record from the wafer. The PAB should be set up as follows:

field	data
Device code	as required
Command Code	03
LUNO	don't care*
Record number	don't care*
Buffer length	max record length
Data length	0000 (length returned)
Return status	(returned)
Buffer address	as required

No data will be sent with the command message.

The following actions take place in the microtape during a READ DATA command:

*If no file is open on drive, then ERROR 4
*If output mode, then ERROR 15
*If EOF, then ERROR 7
*Read data length and data and transmit over bus
*If there is a checksum mismatch, then ERROR 16
*Return status

The response message will contain the data requested. The record number field is ignored for this peripheral although others may use it. The following error status codes may occur,

00 - ok
04 - no file open
06 - device error
07 - EOF error
0F - read in write only mode (15)
10 - data error (16)
FF - time-out error (255)

The device error will occur if the record sync data is not found before the BOT mark is sensed. Valid records will not coincide with the EOT/BOT mark being sensed, or occur between the EOT/BOT mark and the directory as this is detected during WRITE DATA and would not be allowed.

3.2.5 Write Data - 04. This command is used to send a data record to a wafer. The PAB should be set up as follows:

field	data
Device code	as required
Command Code	04
LUNO	don't care*
Record number	don't care*
Buffer length	don't care*
Data length	record length (0000 returned)
Return status	(returned)
Buffer address	as required

Data will be sent after the command PAB assuming the data length was non zero.

The following actions take place in the microtape during a WRITE DATA command:

*If no file is open on drive, then ERROR 4
*If write protected, then ERROR 9
*If input mode, then ERROR 14
*If data length is too big, then ERROR 8
*Receive and write record
*Test for EOT during WRITE DATA
    if EOT found, then set EOT flag, reject record,
        close file, ERROR 32
*Replace number of records count with current record number
*Compare DL of record to maximum record length (MRL)
    if DL is greater than MRL, then replace MRL with DL
*Return status

The response message will contain zero-length data and the operation status. The following error status codes may occur:

00 - ok
04 - no file open
06 - device error
08 - data too long
09 - write protected
0E - write in read only mode (14)
20 - wafer full (32)
FF - time out error (255)

The "Wafer Full" error will occur if the EOT marker is sensed any time the motor is running during a write operation.

3.2.6 Restore - 05. This command is used to position the wafer to the first record of the currently open file. The PAB should be set up as follows:

field	data
Device code	as required
Command Code	05
LUNO	don't care*
Record number	don't care*
Buffer length	don't care*
Data length	0000 (0000 returned)
Return status	(returned)
Buffer address	don't care*

The wafer is positioned to access the first record of an open file.

The following actions take place in the microtape during a RESTORE command:

*If no file is open on drive, then ERROR 4
*Save file number
*If not write protected, then rewrite directory
*If write protected, then poition to beginning of wafer
*Search for file by number
*Return status

The response message will contain zero-length data and the operation status. The following error status codes may occur:

00 - ok
04 - no file open
06 - device error
FF - time out (255)

A file not found error may occur if the wafer needs to be cycled and the file can not be found again

3.2.7 Delete - 06. This command deletes a file from the wafer. The PAB should be set up as follows:

field	data
Device code	as required
Command Code	06
LUNO	don't care*
Record number	don't care*
Buffer length	don't care*
Data length	>=0001 (0000 returned)
Return status	(returned)
Buffer address	as required

The following data is sent in a DELETE command:

Device characteristics (variable length) - file name, may contain string for name of file or a 1 digit hexadecimal file number.

The following actions take place in the microtape during a DELETE command:

```
*Read filename or number
    if missing, then ERROR 1
    if filename data length gt 12, then ERROR 1
    set flag for file by name or number
*If a file is open, then ERROR 5
*If write protected, then ERROR 9
*Search for file
*If file not found, then ERROR 3
*If file found, then inactivate file
*If file was last, then find new last file
*Rewrite directory
*Return status
```

The response message will contain zero-length data and the operation status. The following error/status codes may occur:

- 00 - ok
- 01 - characteristics error
- 03 - file not found
- 05 - file already open
- 06 - device error
- 09 - write protect
- FF - time out error (255)

3.2.8 Return Status - 07. This command is used to return device status information. to the data buffer. The PAB should be set up as follows:

field	data
Device code	as required
Command Code	07
LUNO	don't care*
Record number	don't care*
Buffer length	>= 0001
Data length	0000 (>=0001 returned)
Return status	(returned)
Buffer address	as required

The bit fields in the return data are as follows (bit 0 is the least significant bit).

```
      7 - end of file has been reached (1=true,0=false)
      6 - random access supported (zero returned)
      5 - file is protected (zero returned)
      4 - file/device open (1=true,0=false)
    3-2 - device type (01 returned)
            0 - display type
            1 - internal data type
            2 - data communications
            3 - undefined
    1-0 - I/O modes (11 returned)
            1 - read only
            2 - write only
            3 - read/write
```

The return status call must normally be preceeded by an open call. If it is not preceeded by an open call than the returned value in bit 4 will be zero. The following error status codes may be returned

```
    00 - ok
    FF - time out
```

3.2.9 Verify - OC. This command is used to verify a data record written to or read from the microtape peripheral. The PAB should be set up as follows:

field	data
Device code	as required
Command Code	OC
LUNO	don't care*
Record number	don't care*
Buffer length	don't care*
Data length	record length (0000 returned)
Return status	(returned)
Buffer address	as required

Data will be sent after the command pab assuming the data length was non zero.

The following actions take place in the microtape during a VERIFY command:

*If no file is open on drive, then ERROR 4
*If EOF, then ERROR 7
*Read wafer and compare to bus data received
*If data length not same, then ERROR 24

*If data mismatch, then ERROR 24
*If checksum mismatch, then ERROR 16
*Return status

The response message will contain zero-length data and the operation status. The following error status codes may occur:

00 - ok
04 - no file open
06 - device error
07 - EOF error
10 - data error (16)
18 - verify error (24)
FF - time out error (255)

3.2.10 Format Wafer - 0D. This command code is used to initialize a wafer before data is written. The PAB should be set up as follows:

field	data
Device code	as required
Command Code	0D
LUNO	don't care*
Record number	don't care*
Buffer length	don't care*
Data length	0000 (0000 returned)
Return status	(returned)
Buffer address	as required

This command will write a tape header showing zero records on the tape.

The following actions take place in the microtape during a FORMAT command:

*If a file is open, then ERROR 5
*If write protected, then ERROR 9
*Initialize directory in the controller RAM, no active files
*Write the directory to the wafer
*Return status

The response message will contain zero-length data and the operation status. The following error status codes may occur:

00 - ok
05 - file already open
06 - device error
09 - write protect

FF - time out error (255)

3.2.11 Read Catalog - OE. This command will read the directory and the file headers and return the file information therein. The PAB should be set up as follows:

field	data
Device code	as required
Command Code	OE
LUNO	don't care*
Record number	file number
Buffer length	>= 0012
Data length	0000 (0012 returned)
Return status	(returned)
Buffer address	as required

The following actions take place in the microtape during a READ CATALOG command:

*Set file name number flag to number
*If a file is already open on drive, then ERROR 5
*If file number in PAB is O, then read directory
*Find file by number, read file name from header
*If file not found, then ERROR 3
*Position to end of file
*If buffer length is lt 18, then ERROR 12
*Return data length of 18
*Return file number
*Return file name
*Return maximum record length
*Return number of records
*Return flags (active, last, file type)
*Return status

The data buffer returned will contain the following information.

File number	1 byte
File name	12 bytes
Number of records	2 bytes
Maximum record length	2 bytes
Flags	1 bytes

The following error status codes may occur.

00 - ok
03 - file not found
05 - file already open

06 - device error
0C - buffer length error (12)
FF - time out error (255)

3.2.12 Null - FE. This is a dummy command and should be ignored by all devices. No data will be sent and no response will be returned. The PAB should be set up as follows:

field	data
Device code	as required (usually 00)
Command Code	FE
LUNO	don't care*
Record number	don't care*
Buffer length	don't care*
Data length	0000
Return status	(none returned)
Buffer address	none

The only response to a NULL command is to inhibit bus communication until the next message. The transmitter will time out

3.2.13 Reset - FF. This command causes all devices on the bus to close any and all open files. No response will be returned since all devices execute this command. The PAB should be set up as follows:

field	data
Device code	as required (usually 00)
Command Code	FF
LUNO	don't care*
Record number	don't care*
Buffer length	don't care*
Data length	0000
Return status	(none returned)
Buffer address	none

The following actions take place in the microtape during a RESET command:

*If no file is open on drive, then end command
*Reset open flag
*If update, or output mode, then check write protect flag
    if not write protected, then rewrite directory
    if write protected, then ERROR 9

*Inhibit bus

There will be no response to a RESET command. The transmitter will time out.

3.2.14 Unsupported Commands.

After the PAB has been read, if the command is determined to be unsupported, the microtape will wait until the bus becomes idle and then return a data length 0000 and status 0D (13).

SECTION 4

Stringy Floppy Utilities

Utilities will be provided for backing-up wafers, testing drives and wafers, and other functions as needed. The utilities will be supplied on a microtape wafer provided with the controller/drive. The utilities are written in assembly language and execute in the calculator RAM memory. The special I/O calls described above will be used by the utilities.

4.1 Operation

The utilities are invoked from the BASIC interpreter with the following commands:

```
OLD (n.utilities)
RUN
```

When the utilities begin the display will show;

```
MICROTAPE UTILITIES V:1.0
```

This will be displayed for 4 seconds or until a key is pressed. Then the main prompt is displayed. The user will enter the desired command and the utilities will prompt for further information.

4.1.1 Back-up.

The software will prompt for the device number and ask the user for verification.

```
enter source device number:
enter destination device number:
verify back-up ss to dd (enter)
back-up begun
complete;  n files
```

4.1.2 Quit.

SECTION 5

Implementation Overview

File management requires certain information to be maintained in the local processor memory. The memory information block consists of the following:

item		
Current file flags		1 byte
EOF flag	bit 7	
Error flag	bit 6	
Write protect flag	bit 5	
File open flag	bit 4	
File/record found flag	bit 3	
EOT flag	bit 2	
File reference flag (name/#)	bit 1	
Unused	bit 0	
Device attributes	1 byte	(described in OPEN)
Permanent file flags	.5 bytes	
Active flag	bit 15	
Last file flag	bit 14	
Unused	bit 13	
File type flag	bit 12	
Number of records in file	1.5 bytes (bit 11 - 0)	
Current record number	2 bytes	
Maximum record length in file	2 bytes	
Current file number	1 byte	

The current file flags are changed and tested on a command to command basis and, for the most part, apply to handling an open file. The permanent file flags refer strictly to one file and are kept in the directory when that file is not open and will be written onto the wafer as a part of the directory.