ASPI

Advanced SCSI Programmer's Interface

for

<u>MS-DOS</u>

Revision 2.5

October 2, 1989

PRELIMINARY

1.0 Introduction

This specification describes the Advanced SCSI Programmer's Interface for the MS-DOS environment. These specifications which were developed by Adaptec, have been proposed in conjunction with other major SCSI vendors, as the Common Access Method (or CAM) standard interface for SCSI. A similar standard interface is available for the OS/2 environment, which significantly minimizes the porting effort from MS-DOS.

The biggest advantage of ASPI is that it supports concurrent operation of multiple devices through one SCSI interface. When you want to add support for a new type of SCSI device, you don't have to worry about the new device interfering with the devices that are already working in the system.

Today, most SCSI interfaces are used for disk and tape only. Typically, the disk drivers directly manipulate the SCSI hardware. If there is no low level SCSI manager such as ASPI, it is very difficult to add support for a new SCSI device since the new driver must interact directly with the existing drivers. Each time a new device is added to the system (CD-ROM, WORM drives, printers, network interfaces ...), the process is repeated and becomes increasingly complex. ASPI eliminates this problem. It is a device independent driver for the SCSI hardware. Device drivers talk to the ASPI interface interface and ASPI manages the SCSI hardware. ASPI encourages the use of many devices with SCSI.

There is a growing list of ASPI compatible drivers for disk, tape, WORM, and CD-ROM devices available from Adaptec, third-party software companies, major OEM integrators, and peripheral manufacturers.

2.0 Accessing the SCSI Driver

After the SCSI Driver loads in memory, other device drivers can access the SCSI Driver by:

    1. Opening the SCSI Driver
    2. Getting the Entry Point of the SCSI Driver
    3. Closing the SCSI Driver, if necessary
    4. Calling the SCSI Driver

2.1 Opening the SCSI Driver

Device drivers wishing to access the SCSI Driver must open the driver by performing a DOS function call OPEN A FILE as follows:

Perform DOS Function call through INT 21h.

On Entry:

```
AH    = 3Dh
DS:DX = pointer to 'SCSIMGR$'
AL    = 0
```

On Return:

```
AX    =   File Handle if carry flag is not set,
          Error Code if carry flag is set
```

2.2 Getting the Entry Point of the SCSI Driver

Device drivers can get the entry point to the SCSI Driver by performing a DOS function call IOCTL READ as follows:

Perform DOS Function call through INT21h.

On Entry:

AH	= 44h
AL	= 02h
DS:DX	= pointer to data returned
CX	= 4
BX	= File Handle

On Return:

AX	= nothing

Data returned is the entry point of the SCSI Driver.

byte 0 : offset (lo)
byte 1 : offset (hi)
byte 2 : segment (lo)
byte 3 : segment (hi)

2.3 Closing the SCSI Driver

Device drivers wishing to close the SCSI Driver must close the driver by performing a DOS funtion call CLOSE A FILE.

Perform DOS Function call through INT 21h.
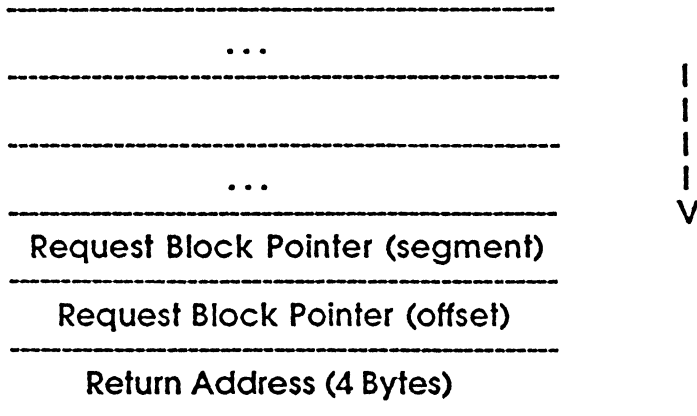
On Entry:

AH = 3Eh
BX = File Handle

On Return:

AX = Error Code if carry flag is set,
        NONE if carry flag is not set

2.4 Calling the SCSI Driver

Device drivers must perform a FAR call to the SCSI Driver with the address of a SCSI Request Block on the stack. For example:

```
PUSH       @REQUEST BLOCK          ;Push 4 byte pointer to
                                   ;request block
FAR CALL   SPI ENTRY POINT         ;Issue FAR call to SPI
                                   ;entry point
```

```
   ------------------------------------------
                    . . .
   ------------------------------------------         |
                                                      |
   ------------------------------------------         |
                    . . .                             |
   ------------------------------------------         V
      Request Block Pointer (segment)
   ------------------------------------------
       Request Block Pointer (offset)
   ------------------------------------------
         Return Address (4 Bytes)
   ------------------------------------------
```

3.1 The SCSI Request Block Header

SPI request blocks are used by both drivers and applications to send commands to the SCSI driver layer. All request blocks have an eight block header.

BYTE		DESCRIPTION
0	I	Command Code
1	I	Status
2	I	Host Adapter Number
3	I	Flags
4	I	Reserved (00)
5	I	Reserved (00)
6	I	Reserved (00)
7	I	Reserved (00)

3.1 SRB Command Codes

COMMAND CODE	DESCRIPTION	SECTION
00H	Host Adapter Inquiry	3.2.1
01H	Get Device Type	3.2.2
02H	Execute SCSI I/O command	3.2.3
03H	Abort SCSI I/O command	3.2.4
04H	Reset SCSI Device	3.2.5
05H	Set Host Adapter Parameters	3.2.6

© 1989 Adaptec, Inc. 6

3.1.2 SRB Status Codes

The status byte is used to post the status of the command. A zero status indicates that the request is still in progress or queued.

STATUS CODE	DESCRIPTION
00H	SCSI request in progress
01H	SCSI request completed without error
02H	SCSI request aborted by host
03H	Unable to Abort SCSI request
04H	SCSI request completed with error
80H	Invalid SCSI request
81H	Invalid Host Adapter Number
82H	SCSI device not installed

3.2.1 Host Adapter Inquiry

BYTE	DESCRIPTION
0	Command Code = 0
1	Status
2	Host Adapter Number
3	Flags
4-7	Reserved (00)
8	# of Host Adapters
9	Target Id of Host Adapter
10-25	SCSI Manager ID
26-41	Host Adapter ID
42-57	Host Adapter Unique Parameters

This command will always return non-zero status. A SCSI request complete without error status indicates that the remaining fields are valid. *An invalid host adapter number status indicates that the specified host adapter is not installed.*

This function is also used to get information on the installed host adapter hardware including the number of host adapters installed. It can be issued once with host adapter zero specified, in order to get the number of host adapters. If further information is desired, the command can be issued for each individual host adapter.

3.2.2 Get Device Type

BYTE	DESCRIPTION
0	Command Code = 1
1	Status
2	Host Adapter Number
3	Flags
4-7	Reserved (00)
8	Target Id
9	LUN
10	Peripheral Device Type of specified Target/LUN
11-16	Reserved for Expansion

This command will always return with non-zero status. A command complete without error status indicates that the specified device is installed and the peripheral device type field is valid. A SCSI device not installed error indicates that the peripheral device type field is not valid.

8

3.2.3 Execute SCSI I/O Request

BYTE	DESCRIPTION
0	Command Code = 2
1	Status
2	Host Adapter Number
3	Flags (Direction,SCSI Linking and Posting)
4-7	Reserved (00)
8	Target Id
9	LUN
10-13	Data Allocation Length
14	Sense Allocation Length (N)
15	Data Buffer Pointer (offset lo)
16	Data Buffer Pointer (offset hi)
17	Data Buffer Pointer (segment lo)
18	Data Buffer Pointer (segment hi)
19	Link Pointer (offset lo)
20	Link Pointer (offset hi)
21	Link Pointer (segment lo)
22	Link Pointer (segment hi)
23	SCSI CDB Length (M)
24	Host Adapter Status
25	Target Status
26	POST Routine Address (offset lo)
27	POST Routine Address (offset hi)
28	POST Routine Address (segment lo)
29	POST Routine Address (segment hi)
30-41	Reserved for DOS Specific definition
42-63	Reserved for SCSI Manager Workspace
63-(63+M)	SCSI Command Descriptor Block
(63+M)-(63+M+N)	Sense Allocation Area

This command will usually return with zero status indicating that the request was queued successfully. Command completion can be determined by polling for non-zero status or through the use of the POST routine field.

The **FLAG** Byte (byte 3) has four valid bits:

Bit 4 - Outbound Data transfer (WRITE commands).
Bit 3 - Inbound Data transfer (READ commands).
Bit 1 - Link Pointer Address is valid, SCSI Linking is enabled.
Bit 0 - POST Routine Addressis valid,POSTing is enabled.

Note 1: Bit 1 must be set to enable SCSI Linking, and
Bit 0 must be set to enable POSTing.

Note 2: If bits 3 and 4 are zero, then the SCSI command will be executed, but data overrun and underrun errors may not be detected.

Target ID and LUN fields are used to specify the peripheral device involved in the I/O.

The **data allocation length** indicates the number of bytes to be transferred. The maximum data allocation length is 64k bytes.

The **request sense allocation length** indicates (in bytes) the number of bytes allocated at the end of the SRB for sense data. Request sense is automatically generated if a check condition status is presented at the end of a SCSI command.

The **data buffer pointer** field points to the data area for SCSI transfer. The data buffer pointer must be an even address.

The **link pointer field** points to the next SRB, which contains the next SCSI link command. If Bit 1 of the flag byte is set, the SCSI Driver executes the next SRB in the link pointer field. Note that the linking feature is only for SCSI devices that support command linking. It is not intended for SRB linking. It is intended for SCSI command linking.

The SCSI command length is the length of the SCSI command descriptor block.

The Host Adapter Status format is as follows:

HA STATUS	DESCRIPTION
00H	No host adapter detected error
11H	Selection Timeout
12H	Data OverRun/UnderRun
13H	Unexpected Bus Free
14H	Target bus phase sequence failure
15H - FFH	Reserved

The Target Status format is as follows:

TARGET STATUS	DESCRIPTION
00H	No target Status
02H	Check Status (Sense Data is in Sense Area)
08H	Specified LUN Busy
18H	Reservation Conflict

The **POST routine address**, if specified, is called when the I/O is complete. This occurs much like an interrupt to the calling driver/program, signaling that the I/O is complete. A POST routine address of zero indicates that POSTing is not necessary.

On entry into the POST routine, *the POST routine assumes that interrupts are enabled and the pointer to the SRB POSTed is on the stack.* On return from the POST routine the SCSI Driver assumes interrupts are enabled.

The SCSI Driver performs a FAR CALL to the POST routine address and the POST routine performs a FAR RETURN when done.
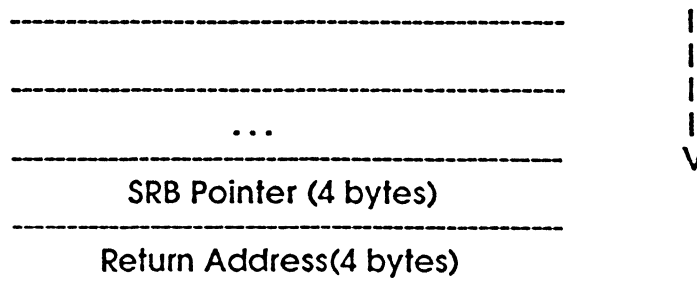
The SCSI command byte fields contain the **SCSI Command Descriptor Block (CDB)**. The length of the SCSI CDB is specified in the SCSI command length field.

The **sense data area** is filled with sense data on a check condition. The length of the sense data area is specified in the sense allocation length field.

SCSI Manager workspace, bytes 30-63, are reserved by the SCSI Driver as long as the SRB status byte is zero. When I/O is complete and the SRB status byte is non-zero the workspace is no longer reserved and are don't care.

ASPI Command Posting

POSTING refers to the SCSI manager making a FAR call to a POST routine as specified in the SRB. The POST routine is called to indicate that the request I/O is complete. The specific SRB completed is indicated by a 4 byte SRB pointer on the stack. The DS of the POST routine as specified in the SRB is also passed on the stack. The POST routine will be called with interrupts enabled. It is assumed that registers are preserved by the POST routine. When the POST routine is called the stack will be set up as follows:

```
--------------------------------------------       |
                                                    |
--------------------------------------------       |
                                                    |
                    . . .                           |
--------------------------------------------       V
             SRB Pointer (4 bytes)
--------------------------------------------
             Return Address(4 bytes)
--------------------------------------------
```

Note: The stack grows downward.

ASPI SCSI Command Linking

ASPI provides the ability to use SCSI linking to guarantee the sequential execution of several commands. Note that the use of this feature requires the involved target(s) to support SCSI linking.

To use SCSI linking, a chain of SRBs is built, with the link pointer in the SRB used to link the elements together. The SCSI linking bit should be set in the SRB flags byte of all SRBs but the last in the chain. When a SCSI target returns LINKED COMMAND COMPLETE, the next SRB is immediately processed, and the appropriate CDB is dispatched. When using SCSI linking, check that the linking flags in the SCSI CDB agree with the SCSI LINKING flag in the flags byte. Inconsistencies can cause unpredictable results. For example, setting the SDB up for linking, but failing to set the SCSI LINKING flag may result in a random address being used for the next SRB pointer. Any error returned from the target on a linked command will break the chain. Note that if linking without tags is used as defined in SCSI, POSTING may not occur on any elements in the chain until the chain is complete.

3.2.4 Abort SCSI I/O Request

This command is used to remove a waiting SCSI request from the queue. It should be issued on any I/O request that has not completed if the driver wishes to timeout on that request. If the specified request is not found in the queue, the abort will fail. This command is not currently implemented.

BYTE		DESCRIPTION
0	I	Command Code = 3
1	I	Status
2	I	Host Adapter Number
3	I	Flags
4-7	I	Reserved (00)
8-9	I	SRB Pointer to Abort (offset)
10-11	I	SRB Pointer to Abort (segment)

Note: Status in the SRB header will be as follows:

02H - SCSI request aborted by host
03H - Unable to abort SCSI request

3.2.5 Reset SCSI Device

This command is used to reset a specific SCSI target. This should not be of use in normal operation, but if the I/O to a particular device hangs up for some reason, drivers can abort the I/O and reset the device before trying again.

BYTE		DESCRIPTION
0	I	Command Code = 4
1	I	Status
2	I	Host Adapter Number
3	I	Flags
4-7	I	Reserved (00)
8	I	Target Id
9	I	LUN
10-23	I	Reserved for SCSI Manager Workspace
24	I	Host Status
25	I	Target Status
26-63	I	Reserved for SCSI Manager Workspace

3.2.6 Set Host Adapter Parameters

This command is set aside for programmers to setup vendor unique features of a specific host adapter.

BYTE		DESCRIPTION
0	I	Command Code = 5
1	I	Status
2	I	Host Adapter Number
3	I	Flags
4-7	I	Reserved (00)
8-?	I	Vendor Unique