# SLR™ (TDC 4000) SERIES
## SCSI-2 Interface
## Functional Specifications

**SLR5 4.0/8.0GB**
**SLR4 2.5/5.0GB** (TDC 4222)
**SLR4 2.5GB** (TDC 4220)
**SLR3 1.2GB** (TDC 4120)
**SLR2 525MB** (TDC 3820 MK2)

**Revision 6**
May 1997

**TANDBERG DATA ASA**
P.O. Box 134 Kjelsås
N-0411 OSLO, NORWAY
Phone + 47 22 18 90 90
Telefax + 47 22 18 95 50

Part No. 42 30 42 Rev. 06.0
Publ. No. 6047-6
May 1997

## SCSI-2 Firmware Information:

This documentation supports SCSI-2 Firmware version 8.06

**Related publications available from our Sales & Marketing Department:**

| Publ. No. | Part No. | Title |
|---|---|---|
| 6046 | 42 30 40 | TDC 4000 Series SCSI-1 Interface - Functional Specifications |
| 9048 | 43 05 08 | SLR (TDC 4000) Series Reference Manual |

This publication may describe designs for which patents are granted or pending. By publishing this information, Tandberg Data ASA conveys no license under any patent or any other rights.

Every effort has been made to avoid errors in text and diagrams. However, Tandberg Data ASA assumes no responsibility for any errors which may appear in this publication.

It is the policy of Tandberg Data ASA to improve products as new techniques and components become available. Tandberg Data ASA therefore reserves the right to change specifications at any time.

We would appreciate any comments on this publication.

# Table of Contents - SCSI-2

# 1. Introduction

## 1.1. General

This manual covers the SCSI-2 Specifications for the SLR Series (TDC 4000) Series Drives. These series consist of six (6) different drive models:

SLR5 4.0/8.0GB
SLR5 4.0GB
SLR4 2.5/5.0GB (TDC 4222)
SLR4 2.5GB (TDC 4220)
SLR3 1.2GB (TDC 4120)
SLR2 525MB (TDC 3820 MK2)

The specifications described in this publication are subject to change without notice.

The main difference between these Drives is the capacity. The following table shows the different capacities and read/write formats for each Drive:

| Capacity | Format | SLR5 4.0/8.0GB | SLR5 4.0GB | SLR4 2.5/5.0GB (TDC 4222) | SLR4 2.5GB (TDC 4220) | SLR3 1.2GB (TDC 4120) | SLR2 525MB (TDC382 MK2) |
|---|---|---|---|---|---|---|---|
| 8.0 GByte *) | QIC-4GB Compressed | Write | - | - | - | - | - |
| 5.0 GByte *) | QIC-2GB Compressed | Write | - | Write | - | - | - |
| 4.0 GByte | QIC-4GB | Write | Write | - | - | - | - |
| 2.5 GByte | QIC-2GB | Write | Write | Write | Write | - | - |
| 1.2 GByte | QIC-1000 | Write | Write | Write | Write | Write | - |
| 525 MByte | QIC-525 | Write | Write | Write | Write | Write | Write |
| 155 MByte | QIC-150 | Write | Write | Write | Write | Write | Write |
| 125 MByte | QIC-120 | Write | Write | Write | Write | Write | Write |
| 60 MByte | QIC-24 | - | - | Read | Read | Read | Read |

NOTE *):
The capacity will depend on the compression ratio, typical values will be in the range from 2 to 3 times compression.

The SLR5 4.0/8.0GB is functionally fully compatible with the SLR5 4.0GB, except that the SLR5 4.0/8.0GB has the capability to perform data compression.

The SLR4 2.5/5.0GB (TDC 4222) is functionally fully compatible with the SLR4 2.5GB (TDC 4220), except that the SLR4 2.5/5.0GB (TDC 4222) has the capability to perform data compression.

SLR2 525MB (TDC 3820 MK2) is fully compatible with SLR2 525MB. The MK2 version has some added functionality compared with the standard TDC 3800.

If a certain functional specification is valid for a particular drive or a group of drives, two methods are used to separate the different specification throughout this manual:

**1)** The special functional specification is marked out with horizontal lines

**2)** The validity for a certain functional specification is written directly in the text

All other text will be common for all SLR (TDC 4000) Series Drives.

## 1.2. Purpose of this Document

This publication specifies functional requirements for the SCSI-2 Host Interface for the SLR (TDC 4000) Series Drives.
This document is meant as a complete specification of the functional behavior of the SCSI-2 Host Interface.

## 1.3. Overview

***Chapter 2*** General description of tape streamers in general and the SLR (TDC 4000) Series Drives in particular. The chapter also describes how to use Fixed and Variable Length blocks and QFA (Quick File Access).

***Chapter 3*** Gives a description of the SCSI logical characteristics as implemented by the SLR (TDC 4000) Series Drives.

***Chapter 4*** Specifies the SCSI Command Descriptors in general.

***Chapter 5*** Lists the Status Bytes implemented by the SLR (TDC 4000) Series Drives.

***Chapter 6*** Lists the Message Bytes implemented.

***Chapter 7*** Specifies the SLR (TDC 4000) Series Drives exception handling in general.

***Chapters 8- 33*** Hold detailed specifications for the individual SCSI-2 commands.

## 1.4. Glossary

**BOM**

**Beginning Of Medium.** The extreme position along the medium in the direction *from* the supply-reel which can be accessed by the use of a REWIND command.

**BOP**

**Beginning Of Partition.** The position at the beginning of the permissible recording region of a partition. If only one partition is defined, this position is equivalent to BOM (see above).

**BOT**

**Beginning Of Tape.** Physical marker on the tape marking the start of the useful area of the tape (located at BOM, see above).

**CDB**

**Command Descriptor Block.** The structure used to communicate commands from an Initiator to a Target.

**Compression Block Group**

A group of compressed data recorded as one variable block on the tape. The Compression Block Group either contains a number of Host-defined logical fixed blocks, or a complete or partial Host-defined variable logical block. The Compression Block Group also contains a Compression Header recorded at the beginning of the Compression Block Group.

**Compression Header**

A group of 10 bytes recorded as uncompressed data at the beginning of a Compression Block Group. The Header contains specific information related to the Compression Block Group recorded on the tape.

**Disconnect**

The action that occurs when a SCSI device releases control of the SCSI bus, allowing it to go to the BUS FREE phase.

**EOD**

**End Of Data.** End of data in a partition is defined in Section 1.5 [4].

**EOM**

**End Of Medium.** The extreme position along the medium in the direction *from* the take-up-reel which can be accessed by the device. This position may be accessed by the use of a LOAD/UNLOAD command with the EOT-bit set to one.

**EOP**

**End Of Partition.** The position at the end of the permissible recording area of a partition. If only one partition is defined, this position is equivalent to EOM (see above).

**EOT**

**End Of Tape.** Physical marker on the tape marking the end of the useful area of the tape (located at EOM, see above).

**EW**

**Early Warning.** Physical tape-mark near - but logically before - EOP (independent of physical direction).

**Field**

A group of one or more contiguous bits. Fields containing only one bit are usually referred to as the XX bit instead of the XX field.

**Initiator**

SCSI-bus Device issuing SCSI commands to a SCSI Target.

**LED**

**Light Emitting Diode.** An indicator on the front of the Drive.

**LSB**

**Least Significant Bit**

**LUN**

**Logical Unit Number**

**MSB** **Most Significant Bit**

**Overlength** The incorrect length condition that exists after executing a read group command, when the length of the actual block read exceeds the requested transfer length in the command descriptor block (CDB).

**Page** Several commands use regular parameter structures that are referred to as pages. These pages are identified with a value known as a page code.

**Parameter** A structure containing one or more fields.

**Partition** The entire region of recording and reading paths in a volume or in a portion of a volume.

**PSEW** **Pseudo Early Warning.** Simulated EW marker on the last track on the tape. PSEW is moved some distance in front of the actual EW.

**Reconnect** The act of re-establishing the physical Initiator/Target connection. A Target reconnects to an Initiator by issuing RESELECTION and MESSAGE IN phases after winning arbitration.

**Reserved** The term used for bits, fields and code values that are set aside for future standardization.

**SCSI** **Small Computer Systems Interface.** Industry standard computer peripheral interface. Used to connect several devices via a common data and control bus.

**SCSI address** The representation of the unique address (0-7) assigned to a SCSI device. This address would normally be assigned and set in the SCSI device during system initialization.

**SCSI ID** The bit-significant representation of the SCSI address referring to one of the SCSI bus data lines.

**Signal assertion** The act of driving a signal to the true state.

**Signal de-assertion** The act of driving a signal to the false state.

**Status** One byte of information sent from a Target to an Initiator upon completion of each command.

**Third-party** When used in reference to RESERVE or RELEASE commands, third-party means a reservation made on behalf of another device.

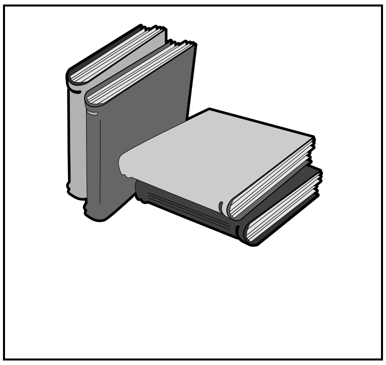
**Target** SCSI-bus Device receiving/executing SCSI commands.

**TLA** **Three Letter Acronym**

**Underlength** The incorrect length condition that exists after executing a read group command when the requested transfer length in the command descriptor block (CDB) exceeds the length of the actual block read.

**Volume** A recording medium together with its physical carrier.

## 1.5. Additional Reference Documentation



[1] *Tandberg Data AS: "TDC 3620/3640/3660 Reference Manual", Publ. No. 5671, Part No. 41 13 59,*
*"TDC 3800 Series Reference Manual", Publ. No. 5871, Part No. 42 22 38 and*
*"SLR (TDC 4000) Series Reference Manual", Publ. No. 9048, Part No. 43 05 08.*

[2] *American National Standards Institute, "SCSI-2, Enhanced Small Computer System Interface", ANSI Working Draft X3T9.2/86-109, Revision 10d, May 1. 1991.*

[3] *Development Standard For 1/4-inch Cartridge Tape Drive SCSI-2 Interface, QIC-121, Revision F, August 21. 1990.*

[4] *QIC, "Serial Recorded Magnetic Tape Cartridge For Information Interchange", QIC-525, Revision E, April 18. 1991.*

[5] *QIC, "Serial Recorded Magnetic Tape Cartridge For Information Interchange", QIC-1000, Revision C, April 17. 1991.*

[6] *QIC, "Serial Recorded Magnetic Tape Cartridge For Information Interchange", QIC-2GB, Revision B, March 10. 1994.*

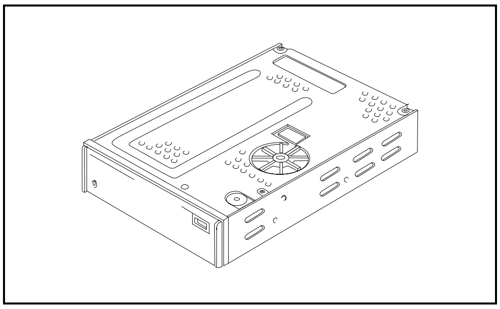
[7] *"Development Standard, Adaptive Lossless Data Compression (ALDC)", QIC-154, Revision A, March 10. 1994.*

[8] *QIC, "Serial Recorded Magnetic Tape Cartridge For Information Interchange", QIC-4GB, Revision A, November 19. 1996.*

*This Page Intentionally Left Blank*

# 2. About Tape Streamers

## 2.1. Physical Elements



Tape streamer devices optimize their use in storing or retrieving user data in a sequential manner. Since access is sequential, position changes typically take long time, when compared to direct-access devices like disks.

The recording medium used with the Drive consists of a flexible substrate coated with a semi-permanent magnetic material. The recording medium is wound onto two reels.

Both the supply reel and the take-up reel are encapsulated into a cartridge. Several standards exist, covering the construction of cartridges for data interchange.

A complete unit composed of the recording medium and its physical carrier (the cartridge) is called a volume. In tape streamers like the SLR (TDC 4000) Series Drives, volumes are removable.

When a volume is loaded, the streamer device is capable of executing commands that cause the medium to be moved (so-called media access commands). When a volume is unloaded the media access commands cannot be executed (that is when these commands report CHECK CONDITION status and a NOT READY sense key).

The write protected state determines when an Initiator may write information on a volume. This attribute is controlled by the user of the volume through the SAFE-switch on the cartridges.
The recording medium has two physical attributes called beginning-of-tape (BOT) and end-of-tape (EOT). Beginning-of-tape is at the end of the medium that is attached to the take-up reel.
End-of-tape is at the end of the medium that is attached to the supply reel.

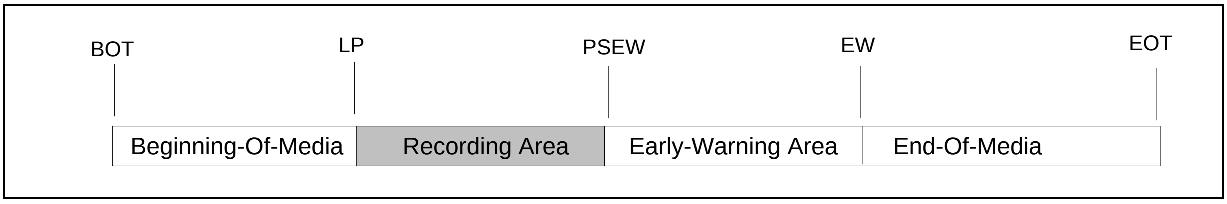
## 2.2. Data Storage Characteristics

**Serpentine Recording Method**

The position on the medium where a pattern of recorded signal may be written by one write component is called a track. The Drive is able to write only a single track at a time. On a new volume, recording of a track begins when moving the tape from Beginning-Of-Media toward End-Of-Media. When End-Of-Media is approached, the direction of recording is reversed and the Drive starts recording a track from End-Of-Media towards Beginning-Of-Media. This process repeats the number of times necessary to record all tracks. The total number of tracks depends on the tape format used. This method of recording is called serpentine.
When reading in the forward direction the same course of tracks is followed as when writing.

From the Initiators point of view the medium may be looked upon as having one large continuous logical track starting from a position called beginning-of-media and ending at a position called End-Of-Media. Beginning-of-media is always on the same side of the volume as beginning-of-tape. End-of-media may be located at the beginning-of-tape or at the end-of-tape depending on whether the total number of tracks is odd or even.

The logical track is split into several areas separated by markers. At least four parts may be identified:



**Beginning-Of-Media** This area holds no user data. It is used to record a special reference burst as defined in [4] and [5]. This area starts with the BOT (Beginning Of Tape) tape marker and ends at the LP (Load Point) tape marker.

**Recording Area** This area holds most of the user data. This area starts with the LP (Load Point) tape marker and ends at the PSEW (Pseudo Early Warning) marker.

**Early-Warning Area** When writing, the Initiator needs an indication that it is approaching the end of the Recording-Area. The position, called Pseudo Early Warning (PSEW) is reported to the Initiator at a position early enough for the Drive to write out any buffered data to the medium while still leaving enough room for additional recorded data or filemarks. It will usually be room for about 400 KBytes of data in the Early-Warning-Area (the actual amount of data is user configurable, see the PSEW Position field in Section 15.3.7.) This area ends at the EW (Early Warning) tape marker.

**End-Of-Media** This is the short area between the EW (Early Warning) and the EOT (End-Of-Tape) tape markers. It is usually possible to complete the writing of a single last frame between EW and EOT. When this has been done or when EOT is found, all further write operations are discontinued even if the there are more data to be written in the data buffer.

## 2.3. Partitions Within a Volume

A volume may be split into several mini-volumes called partitions. Each partition has its own set of beginning and ending points. Each partition within a volume has defined its own Beginning-Of-Partition, Recording-Area, Early-Warning-Area and End-Of-Partition.

On a volume with N partitions the very first partition (partition zero), the Beginning-Of-Partition is identical to Beginning-Of-Medium. For the very last partition on a volume (partition N-1), the End-Of-Partition is identical to End-Of-Media.

All volumes have a minimum of one partition called partition zero, the default data partition.

When a volume is mounted (that is inserted into the Drive and then loaded), it is logically positioned to the beginning of the default data partition (partition zero). When a REWIND command is received in any partition, the Drive positions to the beginning of the current partition.

## 2.4. Quick File Access

Partitions can be used to support the implementation of QFA (Quick File Access). QFA is a feature which provides support for two partitions on the tape cartridge, a directory partition and a data partition.

Quick File Access (QFA) is implemented in the Drive. The QFA mode can be enabled and disabled with the MODE SELECT command. When *not* in QFA mode the Drive will implement a single partition covering the whole tape. This single partition is called the Default Data Partition.

| Partition | Use |
|---|---|
| 0 | Data |
| 1 | Directory Information |

*Table: Partitions Within A Volume*

In QIC compatible tape streamers like the SLR (TDC 4000) Series Drives, all partitions start on track boundaries at the physical BOT end of the tape. A single track is allocated to the directory partition. All remaining tracks are allocated to the data partition. The following table shows the directory track allocation for all supported tape formats:

| | Tape Format | Recording Direction | Approximate Size | Directory Track Number |
|---|---|---|---|---|
| | QIC-120 | Forward | 7.0 MByte | 14 |
| | QIC-150 | Forward | 7.3 MByte | 17 |
| | QIC-525 $^{W}$/DC6320 | Forward | 10.3 MByte | 25 |
| | QIC-525 $^{W}$/DC6525 | Forward | 18.1 MByte | 25 |
| ***SLR5 4.0GB, SLR5 4.0/8.0GB, TDC 4100, TDC 4200 and SLR4 2.5/5.0GB (TDC 4222) only*** | QIC-1000 $^{W}$/DC9100 | Forward | 31.3 MByte | 29 |
| | QIC-1000 $^{W}$/DC9120 | Forward | 39.7 MByte | 29 |
| | QIC-1000 $^{W}$/DC9120XL | Forward | 50.6 Mbyte | 29 |
| | QIC-1000 $^{W}$/DC9100SL | Forward | 4.8 MByte | 29 |
| | QIC-1000 $^{W}$/DC9100FW | Forward | 4.6 MByte | 29 |
| ***SLR5 4.0GB, SLR5 4.0/8.0GB, TDC 4200 and SLR4 2.5/5.0GB (TDC 4222) only*** | QIC-2GB $^{W}$/DC9200 | Forward | 45.6 MByte | 41 |
| | QIC-2GB $^{W}$/DC9200SL | Forward | 5.8 Mbyte | 41 |
| | QIC-2GB $^{W}$/DC9250 | Forward | 58.2 MByte | 41 |
| ***SLR5 Series only*** | QIC-4GB $^{W}$/DC9400 | Forward | 88 MByte | 45 |

*Table: Directory Track Allocation*

***SLR5 4.0/8.0GB and SLR4 2.5/5.0GB (TDC 4222) only***

NOTE:
The data written in the directory partition is always uncompressed [6].

## 2.4.1. Setting the QFA-mode

The MODE SELECT command is used to enter the QFA mode.

The Drive will be in non-QFA mode when the FDP bit in the Medium Partition Parameter Page (Page Code 11h) is set to zero. The Drive will be in QFA mode when the FDP bit is set to one.

Setting or clearing this bit will only be legal when the tape is positioned at BOT or at the beginning of the Data Partition.

## 2.4.2. Changing Partitions Before Reading

The LOCATE command can be used to specify the active partition. The Change Partition (CP) must be set to one and the Partition field must be set to 0 (Data Partition) or 1 (Directory Partition). The Drive will then move to the specified partition before the actual block locate operation starts.

The LOCATE command is only useful when executed on a pre-written tape (the block to locate must exist on the tape).

The READ POSITION and MODE SENSE commands may be used to read the current partition number in use.

When the partition has been changed, the Drive will stay in the selected partition for all media access commands until one of the following actions are taken (by the Initiator):

- *A new LOCATE with CP set to one is executed.*

- *A MODE SELECT command is executed that changes the state of the Active Partition field in the Device Configuration Parameters Page (page code 10h). This is only legal at BOT.*

- *A LOAD or ERASE command is executed. These commands always bring the tape to the beginning of the Data Partition.*

- *The cartridge is removed from the Drive. A newly inserted cartridge must be loaded before it can be accessed. The load operation positions the tape at the beginning of the Data Partition. This will be true whether the load operation happens as a result of an Auto Load or an explicit LOAD command.*

### 2.4.3. Changing Partitions Before Writing

The MODE SELECT command can be used to specify the active partition. The Device Configuration Parameters Page (page code 10h) is used for this purpose. The Change Active Partition (CAP) bit must be set to one and the Active Partition field must be set to 0 (Data Partition) or 1 (Directory Partition). The Drive will then position the tape at the beginning of the specified partition. See Chapter 15. Mode Select.

The MODE SELECT command can be used to change the active partition before a WRITE or WRITE FILEMARK operation is started.

The READ POSITION and MODE SENSE commands may be used to read the current partition number in use.

When the partition has been changed, the Drive will stay in the selected partition for all media access commands until one of the following actions are taken (by the Initiator):

- *A new MODE SELECT command is executed that changes the state of the Active Partition field. This is only legal at BOT.*

- *A LOCATE with CP set to one is executed.*

- *A LOAD or ERASE command is executed. These commands always bring the tape to the beginning of the Data Partition.*

- *The cartridge is removed from the Drive. A newly inserted cartridge must be loaded before it can be accessed. The load operation positions the tape at the beginning of the Data Partition. This will be true whether the load operation happens as a result of an Auto Load or an explicit LOAD command.*

### 2.4.4. General QFA-rules on Write

The following rules applies to QFA-operation on WRITE:

- Write from BOT on DATA-partition:
  - *always OK*

- Write from BOT on DIRECTORY-partition:
  - *only allowed if data is already read/written on DATA-partition, and no reset, power-up or cartridge change has been detected since the Read/Write operation in the DATA-partition was executed. The data on DATA-partition may initially be read/written in single or dual partition mode*

    and

  - *data has not been written from BOT on DIRECTORY partition*

- Append on DATA-partition:
  - *is always OK, either if the initial WRITE operation from BOT was done in single partition or dual mode*

- Append on DIRECTORY-partition:
  - *is always OK if data has been detected on the DIRECTORY-partition*

NOTE:
***General rule:*** Always assure that more data is written on the DATA-partition than on the DIRECTORY-partition.

When writing the first track (Track 0) on DATA-partition, all other tracks are ERASED including the DIRECTORY-partition. This is not the case when writing from BOT on the DIRECTORY-partition, i.e. no ERASE of old data. Thus, if writing more on the DIRECTORY-partition than on the DATA-partition, an overwrite of old data on the DIRECTORY-partition may be the case if the tape is not fully erased prior to this operation. This may corrupt the data on the DIRECTORY-partition. Furthermore, if an APPEND on the DATA-partition is given after this scenario, the last data recorded on the DIRECTORY-partition will be ERASED.

## 2.5. Logical Elements Within a Partition

The Recording-Area on a volume may contain two types of Initiator accessible elements; data blocks and tape marks. These elements are controlled and transferred between the Initiator and the medium using READ, WRITE and WRITE FILEMARKS commands.

A unit of data supplied or requested by the Initiator is called a logical block. Logical blocks are stored according to the specifications of the tape format used [4] and may be recorded as one or more physical blocks on the medium. When the physical block and the logical block are not recorded in a one-to-one relationship, it is the responsibility of the device to perform all blocking, de-blocking or padding of the logical block(s) sent to/from the Initiator.

Filemarks are one kind of tape marks. Filemarks are special recorded elements containing no user data. Initiators traditionally use filemarks to separate user data from labels and logical groupings of data from each other.

Setmarks are another kind of tape marks. Setmarks are special recorded elements containing no user data, providing a segmentation scheme hierarchically superior to filemarks.

Each data block on the tape has a CRC-code attached. The purpose of this code is to make certain the correctness of the data in the block. The CRC-code is 4 bytes long for QIC-4GB, QIC-2GB, QIC-1000 and QIC-525 and 2 bytes long for QIC-150, QIC-120 and QIC-24. The CRC-check operation is transparent to the Initiator.

Interblock gaps, the gaps between blocks, filemarks and setmarks, are introduced on the medium at the time a block or tape mark is written without explicit action by the Initiator.
Minimum and maximum lengths for interblock gaps are defined in [4].

In addition to blocks, tape marks and inter-block gaps erase gaps can be recorded. An erase gap is automatically recorded when a write operation is properly terminated as an end-of-data marker.
In addition a single erase gap may be recorded on the medium through the use of the ERASE command. This erase gap will cover the whole medium and all pre-recorded information will be written over and lost.

The Drive is capable of supporting both fixed and variable length logical blocks. The concept of fixed or variable mode for writing and reading blocks only indicates the method by which the Initiator specifies the size of the logical block for transfer and not the method of recording physical blocks on the medium.

## 2.6. Using Fixed and Variable Length Blocks

### 2.6.1. Variable and Fixed Length Blocks

When reading or writing the Drive groups data transferred to/from the Initiator into ***blocks***. The length of a block may vary. When executing read and write commands two parameters must be specified; the block length and also the number of blocks to read or write. As the SCSI read and write commands only have room for a single parameter, only the block length or the number of block can be specified directly. For this reason there are two different versions of each command capable of transferring tape data to or from the Initiator. The fixed length type commands can specify the ***number*** of blocks to be processed. For these commands all blocks will be of equal size. The actual size is given by the current value in the Block Size field in the Block Descriptor List of the MODE SELECT command. The variable length type commands can specify the ***length*** of the individual blocks, but the number of blocks written is always only one (for each command).

When the Block Size field has a value different from zero, the Drive is said to be in ***fixed block mode.*** In this mode both the fixed length type commands and the variable length type commands are allowed. When the Block Size field is zero, the Drive is said to be in ***variable block mode***. In this mode only the variable length type commands are allowed (as writing fixed length blocks of zero (0) bytes has no meaning).

NOTE:
The illegal length handling in the READ command differs slightly depending on the current mode.

### 2.6.2. Writing

When writing, the Drive groups the data transferred from the Initiator into blocks. When using the QIC-525/1000/2GB and QIC-4GB tape format a block can be from 1 to 16777215 bytes long. When using the QIC-120 or QIC-150 tape formats it may be from 1 to 32786 bytes long.

Data blocks may be written with two different versions of the WRITE command; one with the fixed (FIX) bit set and one with the fixed (FIX) bit cleared.

When the FIX bit is cleared, a WRITE command will write a single block. The block length may be specified on a block-by-block basis. This is useful when writing blocks of varying length.

When the FIX bit is set to one, a WRITE command may write multiple blocks. The WRITE command must specify the number of blocks to write. All written blocks will be of the same length. The length used is the length reported by the MODE SENSE command (the Block Size field of the Block Descriptor List). Note that when the Block Size field has been set to zero, the Drive is said to be in Variable Block mode and commands with the FIX bit set to one is not allowed.

## 2.6.3. Reading

When reading data off a tape, the Drive is able to determine the length of each block read. When reading, the expected block length must be specified. This can be done in two different ways with the two different versions of the READ command; one with the fixed (FIX) bit set and one with the fixed (FIX) bit clear.

When the FIX bit is clear, a READ command will read a single block. The expected block length may be specified on a block-by-block basis.

When the FIX bit is set to one, a READ command may read multiple blocks. The READ command must specify the number of blocks to read. The expected block length of all blocks is the same. The expected length is specified with the MODE SELECT command(the Block Size field of the Block Descriptor List). Note that when the Block Size field has been set to zero, the Drive is said to be in Variable Block mode and commands with the FIX bit set to one is not allowed.

## 2.6.4. Illegal Length Conditions when Reading

When the specified block length does not match the actual block length, the READ command will complain (if not the Suppress Illegal Length Indicator (SILI) bit was set in the READ Command Descriptor Block).

### 2.6.4.1. Reading with the Fixed-Bit Clear

If the actual block length is smaller then the expected block length (the length specified in the Command Descriptor Block of the READ command), the READ command will transfer the actual number of bytes found in the block. The READ command is then terminated with a CHECK CONDITION Status. The Illegal Length Indicator will be set in the Sense Data List. The Information Bytes will be set to the difference between the expected number of bytes and the actual number of bytes. This will be a positive number in this case. The logical tape position will be at the beginning of the next block on the tape.

If the actual block length is larger than the expected block length, the READ command will transfer the expected number of bytes only. The READ command is then terminated with a CHECK CONDITION Status. The Illegal Length Indicator will be set in the Sense Data List. The Information bytes will be set up with the difference between the expected number of bytes and the actual number of bytes. This will be a negative number in this case. The information bytes is presented as a 32 bit 2's complement number. The logical tape position will be at the beginning of the next block on the tape. This means that the additional bytes in the block with the unexpected length is lost.

### 2.6.4.2. Reading with the Fixed-Bit Set

If the actual block length is smaller then the expected block length, the READ command will transfer the actual number of bytes found in the block. The READ command is then terminated with a CHECK CONDITION Status. The Illegal Length Indicator will be set in the Sense Data List. The Information Bytes will be set to the difference between the specified number of blocks and the actual number of blocks transferred. The block with the unexpected length is counted among the transferred blocks even if its length was wrong. Note that this means that the Information Bytes may read zero even if the READ command was terminated with CHECK CONDITION and Illegal Length Indication. The logical tape position will be at the beginning of the block following the block with the unexpected length.

If the actual block length is larger than the expected block length, the READ command will transfer the expected number of bytes only. The READ command is then terminated with a CHECK CONDITION Status. The Illegal Length Indicator will be set in the Sense Data List. The Information bytes will be set up with the difference between the specified number of blocks and the actual number of blocks transferred. The block with the unexpected length is counted among the transferred blocks even if its length was wrong. Note that this means that the Information Bytes may read zero even if the READ command was terminated with CHECK CONDITION and Illegal Length Indication. The logical tape position will be at the beginning of the block following the block with the unexpected length. This means that the additional bytes in the block with the unexpected length is lost.

## 2.7. Data Buffering

### 2.7.1. Introduction

The Drive has a temporary storage area capable of holding one or more blocks - a data buffer. The data buffer may hold any combination of data blocks, filemarks and setmarks in the process of being written to the medium, or it may contain read-ahead datablocks and filemarks transferred from the medium.

The data buffer operates as a FIFO queue, compensating for the different transfer rates on the SCSI-bus and the tape system. The Drive is usually only connected to the SCSI-bus for short amounts of time when bursts of data are transferred at a much higher speed than the normal tape transfer rate. The data buffer can be in one out of two modes; read mode or write mode. The data buffer is in write mode when executing COPY (backup), WRITE or WRITE FILEMARKS commands. The data buffer is in read mode when executing COPY (restore), LOCATE, READ, SPACE or VERIFY commands.

The physical buffer is split into three areas:

**The Scratch Pad Area**

The size of this area is 16 KBytes (16384) bytes. The Scratch Pad Area is used as a scratch pad for several of the SCSI commands. A typical example is the COPY command which uses about 3 KByte of this area to hold its Segment Descriptor List.

**The ECC Block Buffer Area**

The size of this area is 30 KBytes (30720) bytes. The ECC Block Buffer Area is used to hold the ECC blocks for every complete frame that is currently in the Data Buffer Area.

**The Data Buffer Area**

The size of this area is 210 KBytes (215040) bytes. This is the area used to buffer blocks and filemarks. Only the Data Buffer Area is used to buffer user data and filemarks.

### 2.7.2. Blocks

On both the SCSI-bus side and the tape side of the data buffer data is grouped into blocks. Blocks on the SCSI-bus side is called ***logical*** blocks. Blocks on the tape side is called ***physical*** blocks. When using the QIC-525/1000/2GB/96-34 tape format 14 physical blocks are again grouped into a ***frame***.

As long as data is moving through the Data Buffer Area it is grouped into physical blocks. The blocking and de-blocking into logical blocks is done on the SCSI-bus side of the buffer. When operating in QIC-525/1000/2GB/96-34 mode the Data Buffer Area can hold a maximum of 210 blocks (1024 bytes each). When operating in QIC-150, QIC-120 or QIC-24 mode the maximum number of blocks in the Data Buffer Area is also 210 (512 bytes each).

Note, however, that since one block is always reserved in both read and write mode and two more blocks are reserved for the tape re-write algorithm in write, the effective size of the buffer is ***209 physical blocks*** in read mode and ***207 physical blocks*** in write mode.

### 2.7.3. Blocking/De-blocking

The size of a logical data block may not be the same as the size of a physical data block. On the tape side of the data buffer the block size is given by the size of the physical tape blocks. This size depends on the tape format used. The QIC-24, QIC-120 and QIC-150 tape formats uses 512 byte physical blocks. The QIC-525/1000/2GB/96-34 uses 1024 byte physical blocks. On the bus side of the data buffer the size of a block may vary. When using READ and WRITE commands with the FIXED bit set to zero the actual size is specified in the Command Descriptor Block. When the FIXED bit is set to one the block size is controlled by the Block Size field in the MODE SELECT Block Descriptor List. Fixed block sizes of 512 and 1024 bytes are supported in QIC-525/1000/2GB/96-34 mode while only 512 bytes are supported in QIC-120/150 mode.

When operating in write mode, data transferred into the buffer from the SCSI-bus is grouped into physical blocks on the bus side of the data buffer. When operating in read mode data is de-blocked at the bus side of the data buffer before transferred to the SCSI-bus. This means that the data buffer always buffers one or more ***physical tape blocks***. Note also that every tapemark occupies one physical tape block in the data buffer. The same is true for control blocks and filler blocks.

One logical block usually occupies at least one complete physical block. When the tape format is QIC-525/1000/2GB/96-34 this means that logical blocks with a size less then or equal to 1024 bytes occupies one physical block. Logical blocks with a size in the range 1025 bytes to 2048 bytes occupies two physical blocks etc. There is one exception to this rule. When writing 512 byte blocks with the FIXED bit set to one a special option is used in the QIC-525/1000/2GB/96-34 standard allowing two logical blocks to share one physical block. When the tape format is QIC-120 or QIC-150 every logical block written with the FIXED bit set to zero occupies at least two physical blocks in the buffer. Logical blocks with a size less than or equal to 512 bytes occupies two physical blocks. Logical blocks with a size in the range 513 bytes to 1024 bytes occupies three physical blocks etc. Logical blocks written with the FIXED bit set to one occupies one single physical block.

### 2.7.4. Buffered Mode

The Drive is capable of operating in both a buffered mode and an unbuffered mode. Buffered mode is not applicable during read commands. When operating in buffered mode the Drive returns GOOD status for write operations when all data has been successfully transferred from the Initiator into the data buffer. When operating in unbuffered mode, GOOD status is not returned until all requested data or filemarks are successfully recorded on the medium.

When issuing a buffered WRITE FILEMARKS command with the immediate bit set to one, GOOD status is returned as soon as the filemark(s) has been moved into the data buffer. A WRITE FILEMARKS command with the immediate bit set to zero causes any buffered data blocks or filemarks to be written to the medium. Upon successful completion of this process, which is called a synchronize operation, no data blocks or filemarks remain in the data buffer.

Should an unrecoverable write error occur while in buffered mode, the Drive generates an error condition to the current active command. If no command is active, the error is reported on the next applicable operation as a deferred error.

Note that when operating in buffered mode with the QIC-525/1000/2GB/96-34 tape format the Drive will pack physical blocks from one WRITE or WRITE FILEMARKS command together with physical blocks from the previous WRITE or WRITE FILEMARKS command into the same frame. This is not true when operating in non-buffered mode. If a WRITE or WRITE FILEMARKS command does not transfer enough data (or filemarks) to fill a complete frame (14 KBytes or 14 filemarks) the rest of the frame is filled up with filler blocks. This may of course waste a lot of space on a tape. When reading a tape containing frames with filler blocks the Drive is able to remove the filler blocks without generating any discontinuities in the data stream on the SCSI-bus. Note, however, that even if filler blocks are inserted and removed automatically by the Drive they will occupy buffer space both in read mode and write mode.

### 2.7.5. Read-Ahead

When operating in read mode the Drive always try to fill up the buffer with read-ahead data. This means that when a read type command terminates the Drive continues to read data off the tape. The read-ahead only stops when the data buffer is full, if a non-read command is issued or if there is no more data to read. The read-ahead also stops when an uncorrectable block has been detected.

Read-ahead minimizes tape start and stops because when a read type commands (including SPACE and VERIFY) follows another read type command the wanted data might already be ready in the data buffer.

When an error has been detected by the tape system the Drive does not report the error until all data up to point where the error occurred has been read out of the data buffer. This means that if an unrecovered read error has occurred this error is not reported until the data block in error is requested by an Initiator.

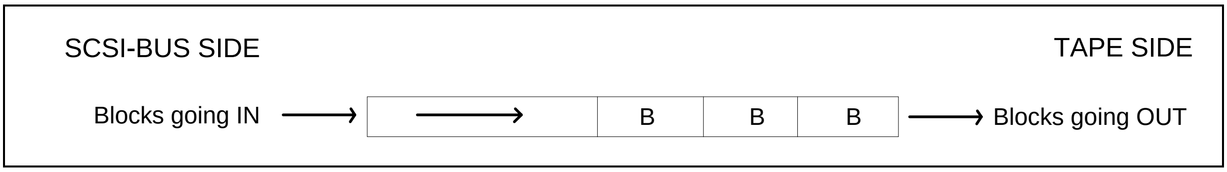
### 2.7.6. Underrun/Overrun

When the Drive is in write mode data are moved from the data buffer on to the tape. The tape write operation can only continue as long as there are any data left to write in the data buffer. If the buffer for any reason becomes empty the write operation must be stopped. This can happen if the Initiator is too slow transferring data or if the Drive is configured in non-buffered mode. When the tape is stopped in this way an ***underrun condition*** has occurred. Note, however, that the tape is not stopped immediately. The Drive is able to delay the stopping for some limited amount of time. This may be controlled by the Initiator by manipulating the Forced Streaming Count in the Miscellaneous Parameters Page (see the MODE SELECT command). When the data buffer later becomes non-empty the tape write operation can continue (see also Section 2.7.7)

When the Drive is in read mode data are moved from the tape and into the data buffer. If the data buffer becomes full the tape read operation must be stopped immediately. This can happen if the Initiator is too slow transferring data or if the Initiator has stopped issuing READ, SPACE or VERIFY commands(remember the tape read-ahead). When the tape is stopped in this way an ***overrun condition*** has occurred. When the data buffer later is emptied the tape read operation may continue(see also Section 2.7.7).

### 2.7.7. Buffer Thresholds

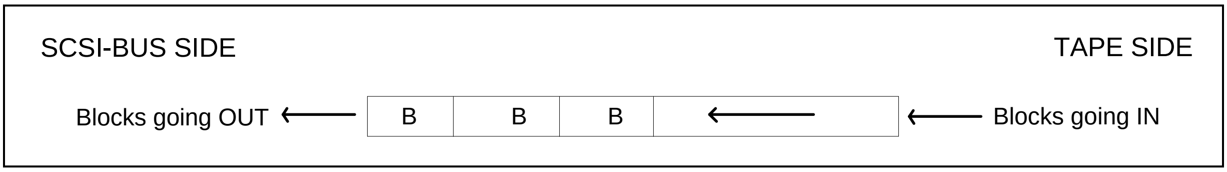
The data buffer acts like a large FIFO. When operating in write mode data blocks, control blocks, filler blocks and filemarks are inserted in one end of this FIFO and the same data and filemarks are later removed (and written to the tape) at the other end. The insertion of control blocks and filler blocks are transparent to the user (the SCSI Initiator).



When moving data into the data buffer data blocks are entered a certain number at the time. If there is no space for the desired number then no data blocks are entered at all. This number is called a buffer threshold. There must usually be at least room for buffer threshold data blocks before new data are transferred in from the SCSI-bus. This will minimize the number of disconnects/reconnects.

When transferring data blocks out of the FIFO and onto the tape a similar mechanism exists. When the FIFO for some reason is empty (an underrun condition), the tape write operation will not be started again until the number of blocks (data, control, filler, ECC or filemarks) ready in the FIFO is equal to or larger than another buffer threshold. This will have a tendency to minimize the number of tape start and stops (underrun conditions).

When the Drive is operating in read mode data blocks, control blocks, filler blocks and filemarks are moved from the tape and into one end of the data buffer. The data blocks and filemarks are later moved out (and transferred on the SCSI-bus) at the other end. Control blocks and filler blocks are just skipped.



In read mode there are also two buffer thresholds; one controlling the tape read operation and one controlling the SCSI-bus transfer. These thresholds work in much the same way as when the Drive is in write mode. Note, however, that the bus threshold is overridden if the reason for not reaching the threshold value is that a filemark or some kind of error message is waiting in the data buffer. In this case the data in front of the filemark or error message is transferred even if the total amount of data is less than the configured buffer threshold.

As can be seen there are 4 different buffer thresholds:

**Write Mode SCSI-bus Threshold**

This is the threshold used when moving data (not filemarks) from the SCSI-bus into the data buffer in write mode. This threshold is controlled by the Write Buffer Empty Ratio (see the MODE SELECT command). By manipulating the Write Buffer Empty Ratio the threshold can set to a minimum of 512 data bytes and a maximum of 65024 bytes. When operating in QIC-120 or QIC-150 mode (with a physical tape block size of 512 bytes) this corresponds to a minimum of 1 physical data block and a maximum of 127 physical datablocks. In QIC-525/1000/2GB/96-34 mode (with a physical tape block size of 1024 bytes) the corresponding numbers are 1 physical data block and 64 physical data blocks (data is taken out 1024 bytes at the time on the tape side of the data buffer).

**Write Mode Tape Threshold**

This is the threshold used when moving physical data, control, filemarks, filler or ECC blocks from the data buffer and onto the tape in write mode. This threshold is controlled by the Write Buffer Full Ratio (see the MODE SELECT command). By manipulating the Write Buffer Full Ratio the threshold can be set to a minimum of 1 physical block and a maximum of 68 physical blocks (regardless of physical block size).

**Read Mode SCSI-bus Threshold**

This is the threshold used when moving data (not filemarks) from the data buffer to the SCSI-bus in read mode. This threshold is controlled by the Read Buffer Full Ratio (see the MODE SELECT command). By manipulating the Read Buffer Full Ratio the threshold can set to a minimum of 512 bytes and a maximum of 65024 bytes. When operating in QIC-120 or QIC-150 mode (with a physical tape block size of 512 bytes) this corresponds to a minimum of 1 physical data block and a maximum of 127 physical data blocks. In QIC-525/1000/2GB/96-34 mode(with a physical tape block size of 1024 bytes) the corresponding numbers are 1 physical data block and 64 physical data blocks (to get out 512 bytes of a 1024 byte block the whole block must have been moved into the other end of the data buffer).

**Read Mode Tape Threshold**

This is the threshold used when moving data from the tape into the data buffer in read mode. This threshold is controlled by the Read Buffer Empty Ratio (see the MODE SELECT command). By manipulating the Read Buffer empty Ratio the threshold can be set to a minimum of 1 physical block and a maximum of 68 physical blocks (regardless of physical tape block size).

Note that when the Drive is writing in QIC-525/1000/2GB/96-34 mode blocks are released for writing at the tape side of the data buffer in ***complete frames***. A frame is collection of 16 QIC-525/1000/2GB/96-34 blocks (1024 bytes each). A maximum of 14 blocks can hold data or filemarks. The two last blocks in a frame is always ECC blocks generated automatically by the buffer system. The ECC blocks do, however, not take up any space in the data buffer. This means that even if there is more blocks in the buffer than specified by the tape threshold these blocks will not be visible for the tape side of the data buffer until there is enough data to build a complete frame. Frames are normally not built until there is at least 14 full QIC-525/1000/2GB/96-34 data or filemark blocks ready in the data buffer. Note, however, that when the Drive operates in non-buffered mode each WRITE command will fill up the last frame with filler blocks (and thereby forcing generation of a complete frame).

When the Drive is reading in QIC-525/1000/2GB/96-34 mode data is also released for use by the SCSI-bus handler in complete frames. This means that even if there is enough data in the data buffer to satisfy the bus threshold this data will not be visible on the bus side of the data buffer until there is enough blocks (data, control, filemark, filler or ECC blocks) to build a complete frame.

### 2.7.8. Deadlock Prevention - Read

When using large buffer thresholds on both the bus side and the tape side of the data buffer there might be a danger of getting into a deadlock situation. This could happen if the sum of the bus threshold (measured in physical tape blocks) and the tape threshold is larger than the maximum number of data blocks available in the data buffer.

When operating in QIC-120 and QIC-150 mode, the bus threshold can be set to a maximum of 127 physical data blocks. The tape threshold can be set to a maximum of 68 physical blocks (of any type). This gives a sum of 195 blocks. Since the data buffer can hold at least 207 blocks this does not seem to make it possible to get into any problems with deadlocks. When operating in QIC-525/1000/2GB/96-34 mode the bus threshold can be set to a maximum of 64 physical data blocks. The tape threshold can be set to a maximum of 68 physical blocks (of any type). When taking into account that only complete frames can pass through the data buffer this gives a sum of 81 blocks (assuming the worst case of 13 blocks in the last incomplete frame) and the chance of getting into a deadlock seems even more remote.

Problems may, however, arise in read mode because the bus threshold specifies a number of data blocks. In certain instances the data buffer may be filled up with large numbers of non-datablocks. This means that the number of blocks holding data maybe much less than 208! This can happen in certain situations in read mode when reading fixed length logical blocks.

***One extreme example:***

*A tape written with WRITE commands specifying transfer of single fixed length logical blocks in unbuffered mode when the tape format is QIC-525/1000/2GB/96-34. This tape will only have 512 bytes of data in each frame. This means that of the 14 available physical blocks in a frame only the first physical block will have data (and only 512 bytes). When reading this tape the data buffer can hold a maximum of 14 complete frames (209 div 14). This means that the data buffer is only able to hold 7168 (512*14) bytes of data. If the bus threshold is set to its maximum value of 65024 bytes it is clear that the possibility of a deadlock is very real even if the tape threshold is set to its minimum value!*

***Another example:***

*When using the QIC-120 or QIC-150 tape format the tape is written with WRITE commands specifying 512 bytes of variable length data (FIXED bit set to zero). This tape will have a control block for every data block. When reading this tape the data buffer can hold a maximum of 104 physical data blocks (209 div 2). This is again less than the maximum bus threshold of 65024 bytes (127 physical data blocks).*

To prevent deadlock the Drive has a special mechanism that will start the data transfer to the SCSI-bus even if the number of available data blocks in the data buffer is less than the configured bus threshold. This means that in certain instances the amount of data transferred in a burst on the SCSI-bus might be less than expected (that is less than the amount configured by the Read Buffer Full Ratio).

### 2.7.9. Disconnect/Re-connect

When the Drive is operating in write mode no data transfer will be initiated until the data buffer can accept the amount of data set up by the current write bus threshold. Instead the Drive disconnects from the SCSI-bus freeing the bus for use by other devices. As soon as the data buffer can accept the data, the Drive reconnects and transfer a burst of data. The size of a burst is usually equal to the bus threshold. If still more data are requested for transfer the Drive then disconnects again. A reconnect later will transfer another burst and the whole process repeats until all requested data has been transferred.

When the Drive is operating in read mode no data transfer will be initiated until the data buffer has ready the amount of data set up by the current read bus threshold. Instead the Drive disconnects waiting for enough data to become ready. When this happens (or a filemark or error message has been detected), the Drive reconnects and transfers a burst of data. The size of the burst is again controlled by the bus threshold. If still more data are requested for transfer the Drive then disconnects again. A reconnect later will transfer another burst and the whole process repeats until all requested data has been transferred. The Drive will NOT disconnect if not allowed by the initiator of the Read or Write command. See Section 6.2.

As long as the requested transfer length is larger than the bus threshold all burst except the very last one will be of equal size (as long as nothing unexpected like an error occurs). The last burst will be equal or shorter in length than the other bursts (depending on the total requested transfer length). No burst will ever be larger than the configured bus threshold. This means that the bus threshold (as set by the Write Buffer Empty Ratio and Read Buffer Full Ratio) also control the maximum burst size. Note that in some special cases there are some exceptions to these rules. See Section 2.7.7 for further details.

### 2.7.10. Data Re-transfer

As a part of its bus parity error handling the Drive is able to re-transfer the last transferred burst any number of times. Re-transfer in the Data Out phase is automatically disabled if disconnects are not allowed. (Identify message = 80h).

When receiving data in write mode the Drive may optionally check for parity errors in the data received. When the complete data burst has been transferred the Drive changes from the Data Out phase to the Message In phase. A single RESTORE POINTERS message is then transferred to the Initiator. The Initiator should in response to this message reset its data pointer to where it was when the Drive instructed it to save its pointer the last time (with the SAVE DATA POINTER message sent each time the Drive disconnects) or if no such message has been received set the pointer back to where it was when the write type command was issued. The Drive then re-enters the Data Out phase and transfers the last burst of data once more. The whole process can be repeated any number of times as long as the Drive detects parity errors in the received data.

When transferring data to the Initiator in read mode the Initiator may check for parity errors in the received data. When a parity error has been detected the Initiator may assert the ATN. After having transferred the complete data burst the Drive will honor the ATN condition by going from the Data In phase to the Message Out phase. A message is then transferred from the Initiator. If this message is an INITIATOR DETECTED ERROR message the Drive will assume that the Initiator wants to have the last burst of data transferred once more. The Drive then goes to the Message In phase and transfers a single RESTORE POINTERS message. The Initiator should then reset its data pointer to where it was when the Drive instructed it to save its pointer the last time (with the SAVE DATA POINTER message sent each time the Drive disconnects) or if no such message has been received set the pointer back to where it was when the read type command was issued. The Drive then re-enters the Data In phase and transfers the last burst of data once more. The whole process can be repeated any number of times as long as the Initiator asserts ATN during the data transfer.

***SLR5 4.0/8.0GB and SLR4 2.5/5.0GB (TDC 4222) only***

When the data is compressed, re-transfer of data is not possible.

The Drive will go from the Data-Out or Data-In phase to the CHECK CONDITION status phase if a parity error is detected. The Error Code will be set to E$STP_MEMP. The Initiator will then reset its data pointers to where they were when the last READ/WRITE type command was issued.

If a parity error is detected during a WRITE operation, the current Compression Block Group will not be written to the tape. If this situation occurs in the middle of a logical block, the logical block will be terminated before all data has been written to the tape. This situation is only valid for logical blocks larger than 32 KBytes. For smaller logical blocks, the logical block(s) will be terminated as OK, but any logical block(s) in the last transfer is (are) not written to the tape.

If a parity error is detected during a READ operation, the current Compression Block Group must be re-read. The Initiator should perform the last READ operation once more.

### 2.7.11. Buffer Parity Errors

The Drive may optionally check for parity errors in the data buffer. A buffer parity error is regarded as a fatal error and no recovery is attempted by the Drive. This means that even if the Drive signals a buffer parity error bad data may have been transferred to the Initiator or to the tape.

The Drive has three checks for buffer parity error:

- *Data moved from the data buffer to the SCSI-bus is checked while leaving the data buffer. A buffer parity error check is made at the end of a complete data burst. At this time the current command is terminated with CHECK CONDITION status.*

- *Data moved from the data buffer to the tape formatter hardware is checked while leaving the data buffer. A buffer parity error check is made at the end of physical block transfer. The current command is terminated as soon as possible. If a data transfer is taking place from the SCSI-bus to the data buffer, the current command is not terminated until this burst of data has been completed. If no command is active when the data buffer parity error is detected then the next command issued will terminate immediately with CHECK CONDITION status.*

- *Internal data transfers (like ECC generation and ECC correction) are checked for buffer parity errors at the end of an internal buffer operation. The current command is terminated as soon as possible. If a data transfer is taking place moving data to or from the SCSI-bus, the command is not terminated until this burst of data has been completed. If no command is active when the data buffer parity error is detected then the next command issued will terminate immediately with CHECK CONDITION status.*

## 2.8. Recorded Objects

The QIC recording formats specify that recorded elements (blocks, filemarks and setmarks) have identifiers included in the recorded information to help determining the write sequence and also to help detecting positioning errors. These identifiers are unique within the whole volume.

The identifiers are associated with physical blocks only. This means that a possible logical block number maintained by the host system will not be the same as the physical identifier of the same logical block because logical blocks may span over multiple physical blocks.

The physical identifiers are normally not visible to the host system. There are, however, some exceptions. In the SLR (TDC 4000) Series Drives, the READ POSITION and LOCATE commands transfer physical block identifiers to/from the host system. The host system is, however, not expected to process this data in any way. The identifiers read from the Drive is just stored and then sent unmodified back to the Drive at some future time.

## 2.9. SLR5 4.0/8.0GB and SLR4 2.5/5.0GB (TDC 4222) Only - Data Compression

Data compression and decompression are only supported by the SLR5 4.0/8.0GB / SLR4 2.5/5.0GB (TDC 4222).

The SLR5 4.0/8.0GB / SLR4 2.5/5.0GB (TDC 4222) supports the Adaptive Lossless Data Compression (ALDC) algorithm. The ALDC is a variant of the LZ 1 (Lempel-Ziv 1) class of compression algorithms. For further details about the compressed algorithm please refer to the description in QIC-154 [7]. The actual compression and decompression are performed in hardware. For further details about the Drive, refer to the *SLR (TDC 4000) Series Reference Manual* [1].

The compression is enabled and disabled using the 'Data Compression Parameter Page' in MODE SELECT, see Chapter 15 for details. Data compression is only valid for the QIC-2GB and QIC-4GB formats. When data compression is enabled, all data sent to the Drive will be compressed, except the data written in the directory partition, which will remain uncompressed. When data is read, the Drive will perform an auto-decompression. The Initiator may then issue a MODE SENSE command and check the 'Data Compression Parameter Page' to investigate if the data on the tape is compressed or not. Mixing of compressed and decompressed data on the same tape is not legal.

When the compression is enabled, the data will be compressed into groups of data, called Compression Block Groups. Each of these Compression Block Groups will consist of an uncompressed 10 bytes Compression Header and compressed data. The amount of the compressed data is the result of compression of up to 32 KBytes of original data. A tape with compressed data will therefore always consist of logical blocks from 11 bytes to 32 KBytes and each of these logical blocks will start with the 10-byte Compression Header. For further details about the compression format, please refer to the QIC-2GB or QIC-4GB specifications [6 or 8].

The data compression ratio will depend on both the data pattern to compress and how the WRITE operations are issued to the Drive. The data is grouped into 'Compression Block Groups' of up to 32 KBytes of original data. The best performance is therefore achieved by issuing WRITE commands with multiples of 32 KBytes of data. Each start of a new WRITE command will force a new 'Compression Block Group' to be written, so many WRITE commands with just a few bytes of data will result in a bad compression ratio. Typical compression ratios are 2:1 to 3:1. See [1] for more information about the compression ratio.

# 3. Logical Characteristics

## 3.1. SCSI-bus Phases

The Drive has the following SCSI-bus phases:

| Bus management phases | Information transfer phases |
|---|---|
| BUS FREE<br>ARBITRATION<br>SELECTION<br>RESELECTION | COMMAND<br>DATA-IN<br>DATA-OUT<br>STATUS<br>MESSAGE-IN<br>MESSAGE-OUT |

*Table: SCSI Bus Phases*

The SCSI-bus will never be allowed to be in more than one phase at any given time. See also Section 1.5. items [1], [2] and [3].

### *3.1.1. Bus Free Phase*

The BUS FREE phase will be used to indicate that no SCSI device is actively using the SCSI-bus and that it is available for subsequent users.

The Drive will detect the BUS FREE phase after SEL and BSY are both false for at least 400 ns.

The Drive will release all SCSI-bus signals within 800 ns after BSY and SEL become continuously false for at least 400 ns.

The Drive will enter the BUS FREE phase by releasing BSY after one of the following conditions:

- *after a SCSI-bus RESET condition*
- *after an ABORT message has been received*
- *after a BUS DEVICE RESET message has been received*
- *after a DISCONNECT message has been transmitted*
- *after a COMMAND COMPLETE message has been transmitted*

The Drive will enter the BUS FREE phase by releasing SEL after one of the following conditions:

- *after an unsuccessful Selection of a Target*
- *after an unsuccessful reselection of an Initiator*

The Drive will expect that any of its Targets (during execution of a COPY command) to only enter the BUS FREE phase after one of the same conditions as specified above. If, however, the Drive detects a BUS FREE phase during execution of the COPY command at any other time, then the Drive will handle this as a "Target Sequence Error". The Drive will in this case not request any sense data from the Target. See the COPY Command Section for further details.

### 3.1.2. Arbitration Phase

The Arbitration will be handled in hardware.

### 3.1.3. Selection Phase

The Selection will be handled in hardware.
When acting as an Initiator (executing the COPY command), the Drive will implement the following Selection Time-out Procedure:

- *If the Drive has waited 250 ms and there has been no BSY response from the Target, then the Drive will continue to assert SEL but will release all DATA BUS signals.*

- *If the Drive has not detected BSY to be true after 200 microseconds, it will release SEL, allowing the SCSI-bus to go to the BUS FREE phase.*

- *The Drive will treat this condition as a "Target Selection Time-out". See COPY command section for further details.*

### 3.1.4. Reselection Phase

The Reselection will be handled in hardware.
When attempting to reselect its Initiator, the Drive will implement the following Reselection Time-out Procedure:

- *If the Drive has waited 250 ms. and there has been no BSY response from the Initiator, then the Drive will continue to assert SEL and I/O but will release all DATA BUS signals.*

- *If the Drive has not detected BSY to be true after 200 microseconds, it will release SEL and I/O, allowing the SCSI-bus to go to the BUS FREE phase.*

- *The Drive will then consider the current command for terminated (as if an ABORT message had been received). Buffered data will continue to be written to the tape if the time-out occurs during a write operation. Immediate commands will continue their execution. No sense data error information will be generated. A new command from the same or any other Initiator will execute normally (as if no time-out has occurred).*

## 3.1.5. Information Transfer Phases

### 3.1.5.1. Command Phase

The COMMAND phase will be used by the Drive to request command information from the Initiator.

The Drive will assert the C/D signal and negate the I/O and MSG signals during the REQ/ACK handshake(s) of this phase.

The Drive will always transfer either six (Group 0 commands) or ten (Group 1 commands) command bytes in one single Command Phase.

### 3.1.5.2. Data In/Out Phases

The DATA-IN phase will be used by the Drive to request that data be sent from the Drive to the Initiator.

The Drive will assert the I/O signal and negate the C/D and MSG signals during the REQ/ACK handshake(s) of the DATA-IN phase.

The DATA-OUT phase will be used by the Drive to request that data be sent from the Initiator to the Drive.

The Drive will negate the C/D, I/O and MSG signals during the REQ/-ACK handshake(s) of the DATA-OUT phase.

Both synchronous and asynchronous Data Transfer are supported. The data bus width is 8 bits.

For the:

- ***INQUIRY***
- ***LOG SENSE***
- ***MODE SENSE***
- ***READ BUFFER***
- ***READ BLOCK LIMITS***
- ***READ POSITION***

***and***

- ***REQUEST SENSE***

***commands...***

the Drive will terminate the DATA-IN phase when Allocation Length bytes have been transferred or when all available sense data have been transferred to the Initiator, whichever is less.

### 3.1.5.3. Status Phase

The STATUS phase will be used by the Drive to request that status information be sent from the Drive to the Initiator.
The Drive will assert C/D and I/O and negate MSG signals during the REQ/ACK handshake of this phase.

#### 3.1.5.4. Message-In/Out Phases

The MESSAGE-IN phase will be used by the Drive to request that message(s) be sent from the Drive to the Initiator.

The Drive will assert C/D, I/O and MSG during the REQ/ACK handshake(s) of the MESSAGE-IN phase.

The MESSAGE-OUT phase will be used by the Drive to request that message(s) be sent from the Initiator to the Drive.

The Drive will assert C/D and MSG and negate I/O during the REQ/ACK handshake(s) of the MESSAGE-OUT phase.

### 3.1.6. Signal Restrictions between Phases

When the SCSI-bus is between two information transfer phases, the Drive will obey the following restrictions:

- *the BSY, SEL, REQ and ACK will not change*
- *the C/D, I/O, MSG and DATA BUS signals may change*

When switching the DATA BUS direction from out to in, the Drive will delay driving the DATA BUS by 1 microsecond after asserting the I/O signal. When switching the DATA BUS direction from in to out, the Drive will release the DATA BUS no later than 45 ns after negating the I/O signal.

## 3.2. SCSI-bus Conditions

### 3.2.1. Attention (ATN)

The ATTENTION condition allows an Initiator to inform a Target that the Initiator has a message ready. The Drive may get this message by performing a MESSAGE-OUT phase.

The Initiator creates the ATTENTION condition by asserting ATN at any time except during the ARBITRATION or BUS FREE phases.

The Initiator will assert the ATN signal before releasing ACK for the last byte transferred in a bus phase for the ATTENTION condition to be honored before transition to a new bus phase. An ATN asserted later might not be honored until a later bus phase and then may not result in the expected action.

The Initiator will keep ATN asserted if more than one message byte is to be transferred.

The Initiator may negate the ATN signal at any time except it will not negate the ATN signal while the ACK signal is asserted during a MESSAGE-OUT phase. Normally, the Initiator negates ATN while REQ is true and ACK is false during the last REQ/ACK handshake of the MESSAGE-OUT phase.

#### 3.2.1.1. The Drive as a Target

If ATN occurs during a COMMAND phase, MESSAGE-OUT will occur after transfer of all Command Descriptor Block bytes have been completed.

If ATN occurs during a DATA phase of a:

- ***COPY***
- ***INQUIRY***
- ***LOG SENSE***
- ***MODE SELECT***
- ***MODE SENSE***
- ***READ BLOCK LIMITS***
- ***READ POSITION***
- ***REQUEST SENSE***

***or***

- ***SEND DIAGNOSTICS***

***command...***

MESSAGE-OUT will occur after transfer of the complete Parameter List.

If ATN occurs during a DATA phase of a:

- ***READ***
- ***READ BUFFER***
- ***RECOVER BUFFERED DATA***
- ***VERIFY***
- ***WRITE***

***or***

- ***WRITE BUFFER***

***command...***

MESSAGE-OUT will occur after transfer of no more than Bus Threshold data blocks (or a single block in case of variable block transfer). See the MODE SELECT Section for an explanation of the Bus Threshold parameter.

If ATN occurs during a:

- ***STATUS phase -***

MESSAGE-OUT will occur after the status byte has been acknowledged by the Initiator.

If ATN occurs during a:

- ***MESSAGE phase -***

MESSAGE-OUT will occur after the message byte has been acknowledged by the Initiator.

If ATN occurs during a:

- ***SELECTION phase***
  ***and before the Initiator releases the BSY signal***

MESSAGE-OUT will occur immediately after the SELECTION phase.

If ATN occurs during a:

- ***RESELECTION phase -***

MESSAGE-OUT will occur after the Drive has successfully sent its IDENTIFY message for that RESELECTION phase.

#### 3.2.1.2. The Drive as an Initiator

The Drive will only create an ATTENTION condition while issuing commands to the random access device during COPY command execution. The ATN line will be asserted during the device selection.

## 3.2.2. Reset

The RESET condition is used to immediately clear all SCSI devices from the bus. This condition will take precedence over all other phases and conditions.
Any SCSI device may create the RESET condition by asserting the RST signal for a minimum of 25 microseconds.

### 3.2.2.1. The Drive as a Target

The Drive will never create a RESET condition while acting as a Target. When the Drive detects a RESET condition it will do the following:

- *Release all SCSI-bus signals within 800 ns of the transition of RST becoming true. BUS FREE phase will always follow the RESET condition*

- *Clear all non-completed commands*

- *Release all SCSI-bus reservations*

- *Return all SCSI device operating modes to their saved values (MODE SELECT) or default values (PREVENT/ALLOW MEDIUM REMOVAL)*

- *UNIT ATTENTION condition will be set for all Initiators (see Section 3.3. Unit Attention)*

NOTE:
The Drive implements the SCSI "hard-reset" alternative and treats a RESET condition like a power-on reset.

### 3.2.2.2. The Drive as an Initiator

The Drive will never create a RESET condition while acting as an Initiator.

## 3.3. Unit Attention

A Unit Attention Condition will begin for each Initiator whenever a new tape cartridge has been inserted, the MODE SELECT parameters affecting this Initiator have been changed by another Initiator, the Drive has been reset by a BUS DEVICE RESET message, a SCSI-bus RESET condition or by a power-on reset.

The Unit Attention Condition will persist for each Initiator until that Initiator clears the condition as described in the following paragraphs.

If an INQUIRY command is received with a pending Unit Attention condition (before the Drive reports CHECK CONDITION status), then the Drive will perform the INQUIRY command, report GOOD status, and will not clear the Unit Attention condition. If the INQUIRY command or any other command is received after the Drive has reported CHECK CONDITION status to the Initiator for a pending Unit Condition, then the Unit Attention condition will be cleared, the Drive will perform the command and report GOOD status.

If a REQUEST SENSE command is received from an Initiator with a pending Unit Attention condition (before the Drive reports CHECK CONDITION), the Drive will discard any pending sense data, report UNIT ATTENTION Sense Key, and then clear the Unit Attention condition for that Initiator. If the Drive has already reported CHECK CONDITION status to this Initiator for Unit Attention condition, then the Drive will also report UNIT ATTENTION Sense Key, and then clear the Unit Attention condition for that Initiator.

If an Initiator issues a command other than INQUIRY or REQUEST SENSE while a Unit Attention condition exists for that Initiator (prior to reporting CHECK CONDITION for the Unit Attention condition), then the Drive will not perform the command and will report CHECK CONDITION status unless a BUSY or RESERVATION CONFLICT status (higher priority status) is also pending.

If, after reporting CHECK CONDITION status to an Initiator for a pending Unit Attention condition, the next command received from that Initiator is not REQUEST SENSE, then that command will be performed and the Unit Attention condition will be cleared for that Initiator.

## 3.4. SCSI Pointers

The SCSI architecture provides for two sets of three pointers within each Initiator.

The first set of pointers are known as the current (or active) pointers. These pointers are used to represent the state of the interface and point to the next command, data or status byte to be transferred between the Initiators memory and the Target. There is only one set of current pointers in each Initiator. The current pointers are used by the Target currently connected to the Initiator.

The second set of pointers are known as the saved pointers. There is one set of saved pointers for each command that is currently active (whether or not it is currently connected). The saved command pointer always points to the start of the Command Descriptor Block for the current command.

The saved status pointer always points to the start of the status area for the current command. At the beginning of each command, the saved data pointer points to the start of the data area. It remains at this values until the Target sends a SAVE DATA POINTER message to the Initiator.

In response to this message, the Initiator stores the value of the current data pointer into the saved data pointer. The Target may restore the current pointers to their saved values by sending a RESTORE POINTERS message to the Initiator.

The Initiator moves the saved value of each pointer into the corresponding current pointer. Whenever a SCSI device disconnects from the bus, only the saved pointer values are retained.

The current pointer values are restored from the saved values automatically upon the next reconnection.

## 3.5. SCSI-bus Phase Sequences

The order in which phases are used on the SCSI-bus follows a prescribed sequence. The RESET condition can, however, abort any phase and is always followed by BUS FREE phase. Also any other phase can be followed by the BUS FREE phase but many such instances are error conditions.

The sequences allowed by the SCSI standards is shown in the figure below. The normal progression is:

- *From the BUS FREE phase to ARBITRATION*
- *From ARBITRATION to SELECTION or RESELECTION*

  *and*

- *From SELECTION or RESELECTION to one or more of the information transfer phases (COMMAND, DATA-IN/OUT, STATUS, or MESSAGE-IN/OUT)*

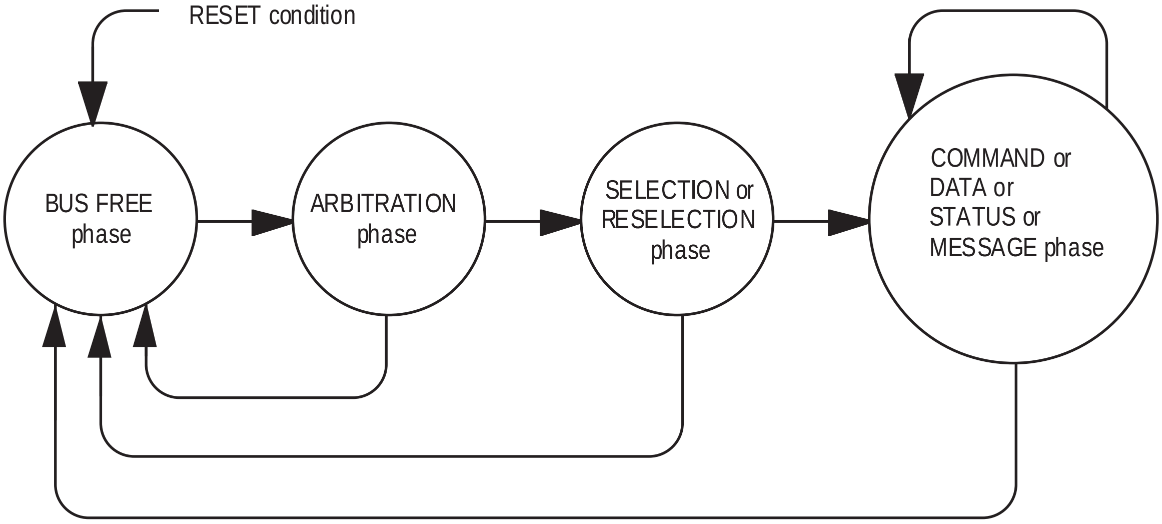

RESET condition
BUS FREE phase
ARBITRATION phase
SELECTION or RESELECTION phase
COMMAND or DATA or STATUS or MESSAGE phase

*Figure: Phase Sequencing*

The phase sequencing between the information transfer phases is always controlled by the Target.

### *3.5.1. The Drive as a Target*

#### 3.5.1.1. Notation

The notation used in the sequences is in BNF (Backus-Naur Form), also known as "context-free grammar". The elements of the notation used in this manual are as follows:

::= *Rewriting rule* or *production*. This symbol is read as *'is defined as'* or *'can be replaced by'*.

| *Or*. This symbol separates alternative definitions.

< > *Angle brackets* are used to delimit the name of a defined sequence, e.g. <sequence1>. Where it is defined, the name of a sequence is set in boldface.

{ } *Braces* are used to denote possible repetition of the enclosed symbols - one or more times.

CAPS The names of SCSI-bus phases are set in capital letters.

***Examples:*** **<initiator-part>** *::= BUS-FREE ARBITRATION SELECTION means that the sequence <initiator-part> is defined as the sequence of the three SCSI-bus phases BUS-FREE, ARBITRATION and SELECTION.*

**<status>** *::= STATUS | STATUS <message-out> mens that the sequence <status> is defined as* either *just the SCSI-bus phase STATUS* or *the SCSI-bus phase STATUS followed by the elsewhere defined sequence <message-out>.*

### 3.5.1.2. Legal Sequences

When acting as a Target, the Drive will adhere to following rules for phase sequencing:

```
<initiator-part>    ::=  BUS-FREE ARBITRATION SELECTION

<message-out>       ::=  MESSAGE-OUT |
                         MESSAGE-OUT <message-out> |
                         MESSAGE-OUT MESSAGE-IN <message-out> |
                         MESSAGE-OUT BUS-FREE

<message-in>        ::=  MESSAGE-IN | MESSAGE-IN <message-out>

<selection>         ::=  <initiator-part> | <initiator-part> <message-out>

<command>           ::=  COMMAND | COMMAND <message-out>

<data-in>           ::=  DATA-IN | DATA-IN <message-out>

<data-out>          ::=  DATA-OUT | DATA-OUT <message-out>

<status>            ::=  STATUS | STATUS <message-out>

<completed>         ::=  <status> <message-in> BUS-FREE

<completed-link>    ::=  <status> <message-in>

<disconnect>        ::=  <message-in> <message-in> BUS-FREE

<reconnect>         ::=  BUS-FREE ARBITRATION RESELECTION <message-in>

<sequence 0>        ::=  <initiator-part> STATUS MESSAGE-IN BUS-FREE

<sequence 1>        ::=  <initiator-part> <command> <completed>

<sequence 2>        ::=  <initiator-part> <command> <data-in> <completed>

<sequence 3>        ::=  <initiator-part> <command> <data-out> <completed>

<sequence 4>        ::=  <initiator-part> <message-out> <command> <completed>

<sequence 5>        ::=  <initiator-part> <message-out> <command> <data-in> <completed>

<sequence 6>        ::=  <initiator-part> <message-out> <command> <data-out>
                         <completed>

<sequence 7>        ::=  <initiator-part> <message-out> <command> <disconnect>
                         <reconnect> <completed>

<sequence 8>        ::=  <initiator-part> <message-out> <command> <disconnect>
                         <reconnect> <data-in> <completed>

<sequence 9>        ::=  <initiator-part> <message-out> <command> <disconnect>
                         <reconnect> <data-out> <completed>
```

**<sequence 10>** ::= <initiator-part> <message-out> <command> <data-out>
<disconnect> <reconnect> <completed>

**<sequence 11>** ::= <initiator-part> <message-out> <command> <disconnect>
<reconnect> <data-out> <disconnect> <reconnect> <completed>

**<sequence 12>** ::= <initiator-part> <message-out> <command>
{ <disconnect> <reconnect> <data-in> } <completed>

**<sequence 13>** ::= <initiator-part> <message-out> <command> <data-in>
{ <disconnect> <reconnect> <data-in> } <completed>

**<sequence 14>** ::= <initiator-part> <message-out> <command>
{ <data-out> <disconnect> <reconnect> } <completed>

**<sequence 15>** ::= <initiator-part> <message-out> <command>
{ <disconnect> <reconnect> <data-out> } <completed>

**<sequence 16>** ::= <initiator-part> <message-out> <command>
{ <data-out> <disconnect> <reconnect> } <data-out> <completed>

**<sequence 17>** ::= <initiator-part> <message-out> <command>
{ <disconnect> <reconnect> <data-out> } <disconnect> <reconnect>
<completed>

**<sequence 18>** ::= <initiator-part> <message-out> <command>
{ <data-in> <disconnect> <reconnect>}<completed>

## *A short description of the legal sequences:*

**<message-out>** This sequence will lead to the BUS FREE phase if the message transferred is either ABORT or RESET. When this happens, all other sequences (where <message-out> is a sub-sequence) will be terminated immediately.
The definition is recursive, it means that the sequence <message-out> is defined as a MESSAGE-OUT phase followed by zero or more MESSAGE-OUT phases and MESSAGE-IN MESSAGE-OUT subsequences, and possibly concluded by a BUS-FREE phase.

The or-part with a <message-out> sub-sequence will always be taken if the ATN line is asserted.

**<message-in>**
**<command>**
**<data-in>**
**<data-out>**
**<status>**

**<sequence 0>** This sequence will only be used when the Drive is busy executing a command for another (or the same) Initiator. The status byte will be BUSY. The message byte will be COMMAND COMPLETED.

**<sequence 1>**
**<sequence 2>**
**<sequence 3>** These are normal sequences for commands with or without a data phase if the ATN signal is NOT asserted during the SELECTION phase. <sequence 1> will also be used for commands that normally have a data phase if an error is detected in the command descriptor block.

**<sequence 4>**
**<sequence 5>**
**<sequence 6>**

These sequences will be used for commands with or without a data phase if the ATN signal is asserted during the SELECTION phase.
<sequence 4> will also be used for commands that normally have a data phase if an error is detected in the command descriptor block.

**<sequence 7> ✶**
**<sequence 8> ✶**
**<sequence 9> ✶**

These sequences will be used when the Drive needs time to complete the PREVIOUS Immediate (or possibly buffered) type command. While this command completes, the Drive will be disconnected. When the command has completed the Drive may start execution of the current command if no errors were detected during execution of the previous command.
<sequence 7> will also be used for commands without data transfer that may need time to execute (like REWIND and LOAD/UNLOAD).

**<sequence 10> ✶**
**<sequence 11> ✶**

These sequences will be used for commands with a data out phase where the Drive needs some time to process the data transferred (like WRITE and MODE SELECT with EEPROM save option). <sequence 11> will be used when it is necessary to wait for the PREVIOUS command to complete.

**<sequence 12> ✶**
**<sequence 13> ✶**

These sequences will be used for the READ command when the Transfer Length exceeds the number of available data blocks in the data buffer. The commands may disconnect in the middle of a transfer. Note that there will be no "empty" reconnect-disconnect sequences.

**<sequence 14> ✶**
**<sequence 15> ✶**
**<sequence 16> ✶**
**<sequence 17> ✶**

These commands will be used for the VERIFY and WRITE commands when the Transfer Length exceeds the number of free blocks in the data buffer. The commands may disconnect in the middle of a transfer. Note that there will be no "empty" reconnect-disconnect sequences.

**<sequence 18> ✶**

Used by the RECOVER BUFFERED DATA command.

NOTE ✶ :
<sequence 1> and <sequence 4> may come between the <disconnect> and the <reconnect> phase due to an overlapping command.

### 3.5.1.3. Disconnects/Reconnects

Disconnection is the process of going through two MESSAGE-IN phases and then to the BUS FREE phase in the middle of a command execution. When the Drive has disconnected, it will always try to reconnect at a later time. Reconnection is the process of going from the BUS FREE phase to the ARBITRATION phase, RESELECTION and MESSAGE-IN phase.

When disconnecting, the first of the two messages transferred will be a SAVE DATA POINTER message. The second message will be a DISCONNECT message. The Drive will then enter the BUS FREE phase.

When reconnecting, the message transferred will be an IDENTIFY message with the Disconnect Granted bit (Bit 6) set to zero. The three LUN bits will also be set to zero.

The following shows examples of Disconnect/Reconnect sequences for all commands. Only the normal execution is shown. It is assumed that the Initiator selects the Drive with ATN asserted so that disconnection can be granted (with the IDENTIFY message).

This section is for information only, and should not be interpreted as a rigorous specification.

As a general rule, the Drive disconnects before it starts operations that may be time consuming. The Drive also disconnects between bursts if it is necessary to break the amount of data to be transferred into several SCSI-bus bursts.

The term *'immediate type command'* means commands with the IMM bit set in the Command Descriptor Block (CDB). The WRITE FILEMARKS command is an exception here. It is not regarded as an Immediate type command even if its IMM bit is set.

Please refer to the section on the MODE SELECT command for details on the Read Buffer Full and Write Buffer Empty Ratios.

### COPY

**<sequence 10>** ::= <initiator-part> <message-out> <command> <data-out> <disconnect> <reconnect> <completed>

This sequence will be used when the COPY command does not follow an Immediate type command. As soon as the COPY Parameter List has been transferred the Drive will disconnect. The Drive will reconnect when the copy operation has terminated. Note that the COPY command will not execute if disconnection has not been granted.

**<sequence 11>** ::= <initiator-part> <message-out> <command> <disconnect> <reconnect> <data-out> <disconnect> <reconnect> <completed>

This sequence will be used when the COPY command follows an immediate type command. The Drive will disconnect the first time when the CDB has been transferred. The Drive will reconnect the first time when the previous Immediate type command has completed execution. As soon as the COPY Parameter List has been transferred the Drive will disconnect the second time. The Drive will reconnect the second time when the copy operation has terminated. Note that the COPY command will not execute if disconnection has not been granted.

### ERASE

**<sequence 7>** ::= <initiator-part> <message-out> <command> <disconnect> <reconnect> <completed>

The Drive will disconnect when the CDB has been transferred. The Drive will reconnect when the tape is positioned back at BOT (or when an error has been detected) if the IMM-bit is not set. If the IMM-bit is set, the Drive reconnects immediately.

This sequence will also be used when the ERASE command follows an Immediate type command. The Drive will disconnect when the CDB has been transferred. If the IMM bit is set, the Drive will reconnect when the previous Immediate type command has completed execution. If the IMM bit is not set the Drive will reconnect when both the previous command and the ERASE command has completed execution.

### INQUIRY

**<sequence 5>** ::= <initiator-part> <message-out> <command> <data-in> <completed>

The Drive will never disconnect when executing this command.

### LOAD/UNLOAD

**<sequence 4>** ::= <initiator-part> <message-out> <command> <completed>

This sequence will be used when the IMM bit is set and the LOAD/UNLOAD command follows a non-Immediate type command other than a COPY, READ, VERIFY, SPACE, SEEK BLOCK, WRITE or WRITE FILEMARKS command.

**<sequence 7>** ::= <initiator-part> <message-out> <command> <disconnect> <reconnect> <completed>

This sequence will be used when the IMM bit is not set. The Drive will disconnect when the CDB has been transferred. The Drive will reconnect when the tape is positioned at BOT or EOT (or when an error has been detected).

This sequence will also be used when the LOAD/UNLOAD command follows an Immediate type command. The Drive will disconnect when the CDB has been transferred. If the IMM bit is set the Drive will reconnect when the previous Immediate type command has completed execution. If the IMM bit is not set the Drive will reconnect when both the previous command and the LOAD/UNLOAD command has completed execution.

This sequence will also be used when the IMM bit is set and the LOAD/UNLOAD command follows a COPY, READ, VERIFY, SPACE, SEEK BLOCK, WRITE or WRITE FILEMARKS command. The Drive will disconnect when the CDB has been transferred. The Drive will reconnect as soon as the previous command has terminated (read/space operation stopped or data flushed from the data buffer) and the actual LOAD/UNLOAD command has started execution.

### LOCATE

**<sequence 7>** ::= <initiator-part> <message-out> <command> <disconnect> <reconnect> <completed>

The Drive will disconnect when the CDB has been transferred. The Drive will reconnect when the specified block has been located (or an error has been detected).

This sequence will also be used when the LOCATE command follows an Immediate type command. The Drive will disconnect when the CDB has been transferred. The Drive will reconnect when both the previous command and the LOCATE command has completed execution.

### LOG SELECT

**<sequence 10>** ::= <initiator-part> <message-out> <command> <data-out> <disconnect> <reconnect> <completed>

This sequence will be used when the LOG SELECT command does not follow an Immediate type command. The Drive will disconnect when the Parameter List has been transferred and then reconnect when the parameters have been saved.

**<sequence 11>** ::= <initiator-part> <message-out> <command> <disconnect> <reconnect> <data-out> <disconnect> <reconnect> <completed>

This sequence will be used when the LOG SELECT command follows an Immediate type command. The Drive will disconnect the first time when the CDB has been transferred. The Drive will reconnect the first time when the previous Immediate type command has completed execution. The Drive will disconnect the second time when the Parameter List has been transferred. The Drive will reconnect the second time when the parameters have been saved.

### LOG SENSE

**<sequence 5>** ::= <initiator-part> <message-out> <command> <data-in> <completed>

This sequence will be used when the LOG SENSE command does not follow an Immediate type command.

**<sequence 8>** ::= <initiator-part> <message-out> <command> <disconnect> <reconnect> <data-in> <completed>

This sequence will be used when the LOG SENSE command follows an Immediate type command. The Drive will disconnect when the CDB has been transferred. The Drive will reconnect when the previous Immediate command has completed execution.

### MODE SELECT

**<sequence 10>** ::= <initiator-part> <message-out> <command> <data-out> <disconnect> <reconnect> <completed>

This sequence will be used when the MODE SELECT command does not follow an Immediate type command. The Drive will disconnect when the Parameter List has been transferred and then reconnect when the parameters have been saved.

**<sequence 11>** ::= <initiator-part> <message-out> <command> <disconnect> <reconnect> <data-out> <disconnect> <reconnect> <completed>

This sequence will be used when the MODE SELECT command follows an Immediate type command. The Drive will disconnect the first time when the CDB has been transferred. The Drive will reconnect the first time when the previous Immediate type command has completed execution. The Drive will disconnect the second time when the Parameter List has been transferred. The Drive will reconnect the second time when the parameters have been saved.

### MODE SENSE

**<sequence 5>** ::= <initiator-part> <message-out> <command> <data-in> <completed>

This sequence will be used when the MODE SENSE command does not follow an Immediate type command.

**<sequence 8>** ::= <initiator-part> <message-out> <command> <disconnect>
<reconnect> <data-in> <completed>

This sequence will be used when the MODE SENSE command follows an Immediate type command. The Drive will disconnect when the CDB has been transferred. The Drive will reconnect when the previous Immediate command has completed execution.

### PREVENT/ALLOW MEDIUM REMOVAL

**<sequence 4>** ::= <initiator-part> <message-out> <command> <completed>.

This sequence will be used when the PREVENT/ALLOW MEDIUM REMOVAL command follows a non-Immediate type command.

**<sequence 7>** ::= <initiator-part> <message-out> <command> <disconnect>
<reconnect> <completed>

This sequence will be used when the PREVENT/ALLOW MEDIUM REMOVAL command follows an Immediate type command. The Drive will disconnect when the CDB has been transferred. The Drive will reconnect when both the previous command and the PREVENT/ALLOW MEDIUM REMOVAL command has completed execution.

### READ

**<sequence 4>** ::= <initiator-part> <message-out> <command> <completed>

This sequence will be used when a READ command specifying a zero transfer length follows a COPY (restore), READ, VERIFY, SEEK BLOCK (with the IMM bit set to zero) or SPACE command.

**<sequence 7>** ::= <initiator-part> <message-out> <command> <disconnect>
<reconnect> <completed>

This sequence will be used when a READ command specifying a zero transfer length follows a command other than COPY(restore), READ, VERIFY, SEEK BLOCK or SPACE.

The Drive will disconnect when the CDB has been transferred. The Drive will reconnect when an error has been detected or when the tape has been prepared for read operations (the previous command has terminated, the reference burst has been located and at least one block of data has been read from the tape.

**<sequence 5>** ::= <initiator-part> <message-out> <command> <data-in> <completed>

This sequence will be used when the previous command was not an immediate type command and the number of bytes requested for transfer is available in the data buffer and when at the same time the number of bytes requested for transfer is less than or equal to the number of bytes specified by the Read Buffer Full Ratio.

**<sequence 8>** ::= <initiator-part> <message-out> <command> <disconnect> <reconnect> <data-in> <completed>

This sequence will be used when the number of bytes requested for transfer is not available in the data buffer and when at the same time the number of bytes requested for transfer is less than or equal to the number of bytes specified by the Read Buffer Full Ratio. The Drive will disconnect when the CDB has been transferred. The Drive will reconnect when the requested amount of data is ready for transfer.

**<sequence 12>** ::= <initiator-part> <message-out> <command>{<disconnect> <reconnect> <data-in>}<completed>

This sequence will be used when the number of bytes available for transfer in the data buffer at the time the READ command was issued is less than the number of bytes specified by the Read Buffer Full Ratio and at the same time the total number of bytes transferred is larger than the number of bytes specified by the Read Buffer Full Ratio.

The Drive will disconnect when the number of data bytes ready for transfer in the data buffer is less than the number of bytes specified by the Read Buffer Full Ratio. The Drive will also disconnect to make sure that the burst size never exceeds the number of bytes specified by the Read Buffer Full Ratio. The Drive will reconnect when the number of bytes ready for transfer is equal to or larger than the Read Buffer Full Ratio.

This sequence is also used when the READ command follows an Immediate type command. The Drive will disconnect when the CDB has been transferred. The Drive will reconnect when the previous Immediate type command has completed execution and the number of bytes ready for transfer is equal to or larger than the Read Buffer Full ratio.

**<sequence 13>** ::= <initiator-part> <message-out> <command> <data-in> {<disconnect> <reconnect> <data-in>} <completed>

This sequence will be used when more data than specified by the Read Buffer Full Ratio is available in the data buffer and at the same time the total number of bytes transferred is larger than the number of bytes set up by the Read Buffer Full Ratio.

The Drive will disconnect when the number of data bytes ready for transfer in the data buffer is less than the number of bytes specified by the Read Buffer Full Ratio. The Drive will also disconnect to make sure that the burst size never exceeds the number of bytes specified by the Read Buffer Full Ratio. The Drive will reconnect when the number of bytes ready for transfer is equal to or larger than the Read Buffer Full Ratio.

### READ BLOCK LIMITS

**<sequence 5>** ::= <initiator-part> <message-out> <command> <data-in> <completed>

This sequence will be used when the READ BLOCK LIMITS command does not follow an Immediate type command.

**<sequence 8>** ::= <initiator-part> <message-out> <command> <disconnect> <reconnect> <data-in> <completed>

This sequence will be used when the READ BLOCK LIMITS command follows an Immediate type command. The Drive will disconnect when the CDB has been transferred. The Drive will reconnect when the previous Immediate command has completed execution.

### READ BUFFER

**<sequence 4>** ::= <initiator-part> <message-out> <command> <completed>

This sequence will be used when the READ BUFFER command does not follow an Immediate type command and the specified transfer length is zero.

**<sequence 7>** ::= <initiator-part> <message-out> <command> <disconnect> <reconnect> <completed>

This sequence will be used when a READ BUFFER command specifying a zero transfer length follows an Immediate type command.

**<sequence 5>** ::= <initiator-part> <message-out> <command> <data-in> <completed>

This sequence will be used when the READ BUFFER command does not follow an Immediate type command, the amount of data is less than one burst and no time consuming preparation of data is necessary.

**<sequence 8>** ::= <initiator-part> <message-out> <command> <disconnect> <reconnect> <data-in> <completed>

This sequence will be used when the READ BUFFER command follows an Immediate type command or if time consuming preparation of data is necessary and the amount of data is less than one burst.

**<sequence 13>** ::= <initiator-part> <message-out> <command> <data-in> {<disconnect> <reconnect> <data-in>} <completed>

This sequence will be used when the READ BUFFER command does not follow an Immediate type command and no time consuming data preparation is necessary and the amount of data is greater than one burst.

**<sequence 12>** ::= <initiator-part> <message-out> <command>{<disconnect> <reconnect> <data-in>}<completed>

This sequence will be used when the READ BUFFER command follows an Immediate type command or if time consuming data preparation is necessary and the amount of data is greater than one burst.

### READ POSITION

**<sequence 5>** ::= <initiator-part> <message-out> <command> <data-in> <completed>

This sequence will be used when the READ POSITION command does not follow an Immediate type command.

**<sequence 8>** ::= <initiator-part> <message-out> <command> <disconnect>
<reconnect> <data-in> <completed>

This sequence will be used when the READ POSITION command follows an Immediate type command. The Drive will disconnect when the CDB has been transferred. The Drive will reconnect when the previous Immediate command has completed execution.

### RECOVER BUFFERED DATA

**<sequence 18>** ::= <initiator-part> <message-out> <command>
{ <data-in> <disconnect> <reconnect>}<completed>

This command can only follow a failing WRITE or WRITE FILEMARKS command.

### RELEASE UNIT

**<sequence 4>** ::= <initiator-part> <message-out> <command> <completed>

This sequence will be used when the RELEASE UNIT command follows a non-Immediate type command.

**<sequence 7>** ::= <initiator-part> <message-out> <command> <disconnect>
<reconnect> <completed>

This sequence will be used when the RELEASE UNIT command follows an Immediate type command. The Drive will disconnect when the CDB has been transferred. The Drive will reconnect when both the previous command and the RELEASE UNIT command has completed execution.

### REQUEST SENSE

**<sequence 5>** ::= <initiator-part> <message-out> <command> <data-in> <completed>

The Drive will never disconnect when executing this command.

### RESERVE UNIT

**<sequence 4>** ::= <initiator-part> <message-out> <command> <completed>

This sequence will be used when the RESERVE UNIT command follows a non-Immediate type command.

**<sequence 7>** ::= <initiator-part> <message-out> <command> <disconnect> <reconnect> <completed>

This sequence will be used when the RESERVE UNIT command follows an Immediate type command. The Drive will disconnect when the CDB has been transferred. The Drive will reconnect when both the previous command and the RESERVE UNIT command has completed execution.

### REWIND

**<sequence 4>** ::= <initiator-part> <message-out> <command> <completed>

This sequence will be used when the IMM bit is set and the REWIND command follows a non-Immediate type command other than a COPY, READ, VERIFY, SPACE, SEEK BLOCK, WRITE or WRITE FILEMARKS command.

**<sequence 7>** ::= <initiator-part> <message-out> <command> <disconnect> <reconnect> <completed>

This sequence will be used when the IMM bit is not set. The Drive will disconnect when the CDB has been transferred. The Drive will reconnect when the tape is positioned at BOT (or when an error has been detected).

This sequence will also be used when the REWIND command follows an Immediate type command. The Drive will disconnect when the CDB has been transferred. If the IMM bit is set, the Drive will reconnect when the previous Immediate type command has completed execution. If the IMM bit is not set, the Drive will reconnect when both the previous command and the REWIND command has completed execution.

This sequence will also be used when the IMM bit is set and the REWIND command follows a COPY, READ, VERIFY, SPACE, SEEK BLOCK, WRITE or WRITE FILEMARKS command. The Drive will disconnect when the CDB has been transferred. The Drive will reconnect as soon as the previous command has terminated (read/space operation stopped or data flushed from the data buffer) and the actual REWIND command has started execution.

### SEND DIAGNOSTICS

**<sequence 7>** ::= <initiator-part> <message-out> <command> <disconnect> <reconnect> <completed>

This sequence will be used when the ST-bit in the command block is set to "1". Default parameters are used both for Selftest 1 and 2.

**<sequence 10>** ::= <initiator-part> <message-out> <command> <data-out> <disconnect> <reconnect> <completed>

This sequence will be used when the ST-bit in the command block is set to "0". This indicates that the Selftest 2 command must be followed by a parameter list.

### SPACE

**<sequence 7>** ::= <initiator-part> <message-out> <command> <disconnect> <reconnect> <completed>

The SPACE command will always disconnect. The Drive will disconnect when the CDB has been transferred. The Drive will reconnect when the space operation has completed (or an error has been detected).

### TEST UNIT READY

**<sequence 4>** ::= <initiator-part> <message-out> <command> <completed>

The Drive will never disconnect when executing this command.

### VERIFY

See READ command. Note, however, that all data transfers will be directed "out".

### WRITE

**<sequence 4>** ::= <initiator-part> <message-out> <command> <completed>

This sequence will be used when the WRITE command does not follow an ERASE, LOAD or REWIND type command and the Drive is in buffered mode and the WRITE command specifies a zero transfer length and when the WRITE command does not follow a READ/SPACE/VERIFY command (in an append operation).

**<sequence 7>** ::= <initiator-part> <message-out> <command> <disconnect> <reconnect> <completed>

This sequence will be used when the WRITE command follows an ERASE, LOAD or REWIND type command or if the Drive is in unbuffered mode. The WRITE command specifies a zero transfer length.
The Drive will disconnect when the CDB has been transferred. The Drive will reconnect when a possible previous command has completed execution and the tape has been positioned for write (the reference burst has been written).

**<sequence 6>** ::= <initiator-part> <message-out> <command> <data-out> <completed>

This sequence will be used when the WRITE command does not follow an ERASE, LOAD or REWIND type command and there is room in the data buffer for the amount of data requested for transfer and the Drive is configured to buffered mode and when at the same time the number of bytes requested for transfer is less than or equal to the number of bytes specified by the Write Buffer Empty Ratio and when the WRITE command does not follow a READ/SPACE/VERIFY command (in an append operation).

**<sequence 9>** ::= <initiator-part> <message-out> <command> <disconnect> <reconnect> <data-out> <completed>

This sequence will be used when the WRITE command follows an ERASE, LOAD or REWIND type command or if at the time when the WRITE command is issued there is not room in the data buffer for at least the number of bytes requested for transfer or if the tape type is unknown. The Drive must be configured to buffered mode and at the same time the number of bytes requested for transfer must be less than or equal to the number of bytes specified by the Write Buffer Empty Ratio.

The Drive will disconnect when the CDB has been transferred. The Drive will reconnect when the previous Immediate type command has completed execution and when there is room for at least the number of bytes requested for transfer and the tape type has become known.

**<sequence 10>** ::= <initiator-part> <message-out> <command> <data-out> <disconnect> <reconnect> <completed>

This sequence will be used when the WRITE command does not follow an ERASE, LOAD or REWIND type command and there is room in the data buffer for the amount of data requested for transfer and the Drive is configured to unbuffered mode and when at the same time the number of bytes requested for transfer is less than or equal to the number of bytes specified by the Write Buffer Empty Ratio and when the WRITE command does not follow a READ/SPACE/VERIFY command (in an append operation).

The Drive will disconnect when the data has been transferred. The Drive will reconnect when all buffered data (and file-marks) have been written and verified (or when an error has been detected).

**<sequence 14>** ::= <initiator-part> <message-out> <command>{<data-out> <disconnect> <reconnect>}<completed>

This sequence will be used when the WRITE command does not follow an ERASE, LOAD or REWIND type command and when there is room in the data buffer (at the time the WRITE command was issued) for at least the number of bytes specified by the Write Buffer Empty Ratio and at the same time the total number of bytes transferred is larger than the number of bytes specified by the Write Buffer Empty Ratio and when the WRITE command does not follow a READ/SPACE/VERIFY command (in an append operation). The Drive must be configured to unbuffered mode.

The Drive will disconnect when there is not room for the amount of data specified by the Write Buffer Empty Ratio. The Drive will also disconnect to make sure that the maximum burst size never exceeds the number of bytes specified by the Write Buffer Empty Ratio. The Drive will reconnect when the number of bytes ready for transfer is equal to or larger than the Write Buffer Empty Ratio.

**<sequence 15>** ::= <initiator-part> <message-out> <command>{<disconnect> <reconnect> <data-out>}<completed>

This sequence will be used when the WRITE command follows an ERASE, LOAD or REWIND type command or when the tape type is unknown or when the number of bytes available for transfer in the data buffer at the time the WRITE command was issued is less than the number of bytes specified by the Write Buffer Empty Ratio and at the same time the total number of bytes transferred is larger than the number of bytes specified by the Write Buffer Empty Ratio. The Drive must be configured to buffered mode.

The Drive will first disconnect when the CDB has been transferred. The Drive will reconnect when the tape type has become known and there is room in the data buffer for at least the number of bytes specified by the Write Buffer Empty Ratio.

The Drive will then disconnect when there is not room for the amount of data specified by the Write Buffer Empty Ratio. The Drive will also disconnect to make sure that the maximum burst size never exceeds the number of bytes specified by the Write Buffer Empty Ratio. The Drive will reconnect when the number of bytes ready for transfer is equal to or larger than the Write Buffer Empty Ratio.

**<sequence 16>** ::= <initiator-part> <message-out> <command>{<data-out> <disconnect> <reconnect>}<data-out> <completed>

This sequence will be used when the WRITE command does not follow an ERASE, LOAD or REWIND type command and there is room in the data buffer (at the time the WRITE command was issued) for at least the number of bytes specified by the Write Buffer Empty Ratio and at the same time the total number of bytes transferred is larger than the number of bytes specified by the Write Buffer Empty Ratio and when the WRITE command does not follow a READ/SPACE/VERIFY command (in an append operation). The Drive must be configured to buffered mode.
The Drive will disconnect when there is not room for the amount of data specified by the Write Buffer Empty Ratio. The Drive will also disconnect to make sure that the maximum burst size never exceeds the number of bytes specified by the Write Buffer Empty Ratio. The Drive will reconnect when the number of bytes ready for transfer is equal to or larger than the Write Buffer Empty Ratio.

**<sequence 17>** ::= <initiator-part> <message-out> <command>{<disconnect>
<reconnect> <data-out>}<disconnect> <reconnect> <completed>

This sequence will be used when the WRITE command follows an ERASE, LOAD or REWIND type command or when the tape type is unknown or when the number of bytes available for transfer in the data buffer at the time the WRITE command was issued is less than the number of bytes specified by the Write Buffer Empty Ratio and at the same time the total number of bytes transferred is larger than the number of bytes specified by the Write Buffer Empty Ratio. The Drive must be configured to unbuffered mode.

The Drive will first disconnect when the CDB has been transferred. The Drive will reconnect when a previous Immediate command has completed execution and the tape type has become known and there is room in the data buffer for at least the number of bytes specified by the Write Buffer Empty Ratio.

The Drive will then disconnect when there is not room for the amount of data specified by the Write Buffer Empty Ratio. The Drive will also disconnect to make sure that the maximum burst size never exceeds the number of bytes specified by the Write Buffer Empty Ratio. The Drive will reconnect when the number of bytes ready for transfer is equal to or larger than the Write Buffer Empty Ratio.

### WRITE BUFFER

**<sequence 4>** ::= <initiator-part> <message-out> <command> <completed>

This sequence will be used when the WRITE BUFFER command does not follow an Immediate type command and the specified transfer length is zero.

**<sequence 7>** ::= <initiator-part> <message-out> <command> <disconnect>
<reconnect> <completed>

This sequence will be used when a WRITE BUFFER command specifying a zero transfer length follows an Immediate type command (a command with the IMM bit set in the CDB).

**<sequence 16>** ::= <initiator-part> <message-out> <command>{ <data-out><disconnect>
<reconnect>} <data-out><completed>

This sequence will be used when the WRITE BUFFER command does not follow an Immediate type command.

The Drive will disconnect when the number of data bytes just transferred equals the number of bytes specified by the Write Buffer Empty Ratio. The Drive will then reconnect immediately to transfer another burst with a maximum size again controlled by the Write Buffer Empty Ratio. If less than maximum burst size data needs to be transferred, <sequence 3> will be used. WRITE BUFFER Mode 5 uses <sequence 14> instead.

**<sequence 15>** :::= <initiator-part> <message-out> <command>
{ <disconnect> <reconnect> <data-out> } <completed>

This sequence will be used when the WRITE BUFFER command follows an Immediate type command. WRITE BUFFER Mode 5 uses <sequence 17> instead. If less than maximum burst size data needs to be transferred, <sequence 9> will be used.

**WRITE FILEMARKS**

**<sequence 4>** ::= <initiator-part> <message-out> <command> <completed>.

This sequence will be used when the write filemarks operation for some reason can not be started (invalid CDB, cartridge not inserted and loaded or cartridge is write protected) and the WRITE FILEMARK command follows a non-Immediate type command.

**<sequence 7>** ::= <initiator-part> <message-out> <command> <disconnect>
<reconnect> <completed>

This sequence will be used when the write filemarks operation can be started. The Drive will disconnect when the CDB has been transferred. The Drive will reconnect when the filemarks have been transferred to the data buffer (or when the filemarks have all been written and verified, see below). This sequence is also used when the WRITE FILEMARKS command follows an Immediate type command.

This sequence will also be used if the Drive is not configured to buffered mode or if the IMM bit is not set or if the requested Number Of Filemarks is zero. The Drive will disconnect when the CDB has been transferred. The Drive will reconnect when all buffered data and the filemark(s) have been written and verified (or an error has been detected).

### 3.5.1.4. Command Linking

When the Link bit is set to one in the Control Byte of a Command Descriptor Block, the Drive will not transfer the usual GOOD status byte or the COMMAND COMPLETED message byte upon successful command completion. Instead an INTERMEDIATE status byte followed by a LINKED COMMAND COMPLETE (or LINKED COMMAND COMPLETE W/FLAG) message byte will be sent. After transferring this message byte the Drive will not go to the BUS FREE phase. Instead the Drive will go directly to a new COMMAND phase and then immediately transfer a new Command Descriptor Block.

The command link function will operate as long as commands complete successfully and the link bit is set. When some error has been detected (any other than GOOD status would have been transferred for a non linked command), the link will be broken, and the current command will be terminated with the proper status byte and then with a COMMAND COMPLETED message byte.

Command linking modifies the command phase sequencing:
The first command in a series of linked commands will follow the prescribed phase sequence except that the <completed-link> sub-sequence will be used instead of the usual <completed> sub-sequence.

The second and every other command except the last will skip the <initiator-part> sub-sequence and go directly to the <command> sub-sequence. A <completed-link> sub-sequence will be used instead of the usual <completed> sub-sequence.

The last command in a series of linked commands will skip the <initiator-part> sub-sequence and go directly to the <command> sub-sequence.

Here is a simple example with three commands. The first two have their Link bits set to one.

**<sequence-link>** ::= <initiator-part> <command-1> <completed-link> <command-2> <completed-link> <command-3> <completed>

### 3.5.2. The Drive as an Initiator

When acting as an Initiator, the Drive will follow the phase sequencing shown in Section 3.5. When the Drive's Target has taken over sequence control in the information transfer phases, the Drive will always be able to follow the Target. All possible phase sequences will be allowed with only one exception; the Drive can assume that there are never both a DATA-IN and a DATA-OUT phase for a single command sequence. The Target may enter the BUS FREE phase at any time.

*This Page Intentionally Left Blank*

# 4. Commands

## 4.1. The Command Descriptor Block

A request to the Drive is performed by sending a Command Descriptor Block. For some commands the request is accompanied by a list of parameters sent during the DATA-OUT phase.

The Drive will support Group 0 and Group 1 commands [1], [2], [3]. Group 0 commands have 6 bytes in the Command Descriptor Block. Group 1 commands have 10 bytes in the Command Descriptor Block.

Examples of Six-Byte and Ten-Byte Command Descriptor Blocks are shown in the two following tables:

| BYTE | BIT 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 00 | Operation Code | | | | | | | |
| 01 | Logical Unit Number (LUN) | | | Parameters | | | | |
| 02 | Parameters | | | | | | | |
| 03 | Parameters | | | | | | | |
| 04 | Parameters | | | | | | | |
| 05 | Control Byte | | | | | | | |

*Table: Typical Six-byte Command Descriptor Block*

| BYTE | BIT 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 00 | Operation Code | | | | | | | |
| 01 | Logical Unit Number (LUN) | | | Parameters | | | | |
| 02 | Parameters | | | | | | | |
| 03 | Parameters | | | | | | | |
| 04 | Parameters | | | | | | | |
| 05 | Parameters | | | | | | | |
| 06 | Parameters | | | | | | | |
| 07 | Parameters | | | | | | | |
| 08 | Parameters | | | | | | | |
| 09 | Control Byte | | | | | | | |

*Table: Typical Ten-byte Command Descriptor Block*

**Operation Code** The Command Descriptor Block always has an operation code as the first byte of the command. See Section 4.4. for operation codes supported by the Drive.

**LUN** The Drive supports only one Logical Unit Number (LUN). This field must be set to zero in all Command Descriptor Blocks for the Drive.

**Parameters** See the specific commands for detailed information on the various parameter bytes.

**Control Byte** See Section 4.2. for details on the Command Control Byte.

## 4.2. Command Control Byte

The Control Byte is the last byte of every Command Descriptor Block. A typical Control Byte is shown below.

| BYTE | BIT 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| *Last Byte* | X | X | RESERVED | | | | Flag | Link |

*Table: The Command Control Byte*

**X** Bit 7 and bit 6 will be ignored by the Drive.

**RESERVED** These bits MUST always be set to zero.

**Flag** If the Link bit is zero, then the Flag bit will be set to zero. If the Link bit is one, and if the command terminates successfully, then the Drive will send LINKED COMMAND COMPLETE message if the Flag bit is zero and LINKED COMMAND COMPLETE W/FLAG message if the Flag bit is one.

**Link** This bit is set to one to indicate that the Initiator desires an automatic link to the next command upon successful completion of the current command. If the Link bit is one, then upon successful termination of the command, the Drive will return INTERMEDIATE status (instead of GOOD status) and will then send one of the two messages defined by the Flag bit above.

When the Link bit and the IMMediate bit of a command (where applicable) are both set one, the Drive will return CHECK CONDITION status with the Error Code set to E$STE_IFIC.

## 4.3. Reserved Fields

Reserved bits, fields, bytes and code values are set aside for future standardization. These bits, fields or bytes will be set to zero. They are marked with the word RESERVED or the letter R in the Command Descriptor Blocks and Parameter Lists. If the Drive receives a reserved bit, field, byte that is not zero or receives a reserved code value, it will terminate the command with a CHECK CONDITION status.

If the offending bit, field, byte or code is located in a Command Descriptor Block, then the whole Command Descriptor Block (6 or 10 bytes) will be transferred before the command is terminated with CHECK CONDITION and the Error Code will be set to E$STE_IFIC.

If the offending bit, field, byte or code is located in a Parameter List, then the whole Parameter List will be transferred before the command is terminated with CHECK CONDITION and the Error Code will be set to E$STE_IFIP.

## 4.4. Command Set Summary

The SCSI-2 version must support the commands listed in the following table.

| Description | Group | Media | Type | Hex Code |
|---|---|---|---|---|
| COPY, Copy Function = 0 | W | Yes | O | 18 |
| COPY, Copy Function = 1 | R | Yes | O | 18 |
| ERASE | M | Yes | M | 19 |
| INQUIRY | | | M | 12 |
| LOAD/UNLOAD | M | Yes | O | 1B |
| LOCATE | S | Yes | O | 2B |
| LOG SELECT | | | O | 4C |
| LOG SENSE | | | O | 4D |
| MODE SELECT | | | M | 15 |
| MODE SENSE | | | M | 1A |
| PREVENT/ALLOW MEDIA REMOVAL | | Yes | O | 1E |
| READ | R | Yes | M | 08 |
| READ BLOCK LIMITS | | | M | 05 |
| READ BUFFER | | | O | 3C |
| READ POSITION | R | | O | 34 |
| RECOVER BUFFERED DATA | | | O | 14 |
| RELEASE UNIT | | | M | 17 |
| REQUEST SENSE | | | M | 03 |
| RESERVE UNIT | | | M | 16 |
| REWIND | M | Yes | M | 01 |
| SEND DIAGNOSTICS | | ✶) | M | 1D |
| SPACE | S | Yes | M | 11 |
| TEST UNIT READY | | | M | 00 |
| VERIFY | R | Yes | O | 13 |
| WRITE | W | Yes | M | 0A |
| WRITE BUFFER | | | O | 3B |
| WRITE FILEMARKS | W | Yes | M | 10 |

*Table: SCSI-2 Command Set*

NOTE ✶)
When executing a Selftest 1, the SEND DIAGNOSTICS is not a media access command. When executing Selftest 2 the SEND DIAGNOSTICS is a media access command.

The ***Command Group*** is used to specify legal (and illegal) command sequences (see also Section 4.5).

**Group M** These are Move Type commands.

**Group R** These are Read Type commands. Data is read off the tape during command execution.

**Group S** These are Space Type commands. Data is read off the tape during command execution, but no data transfer takes place on the SCSI-bus.

**Group W**

These are Write Type commands. Data is written to the tape during command execution.

**All other commands...**

These are Neutral commands.

Commands marked with "**Yes**" in the "**Media**" column are called "Media Access Commands". The Drive will terminate the command with CHECK CONDITION status if any media access command is issued with no cartridge loaded. A cartridge is loaded when it is inserted and the Auto Load option is enabled or a LOAD/UNLOAD command has been executed with the Load bit set to one.

The ***Command Type*** is defined by the SCSI-standards [1] [2] [3] like this:

**Type M**

These commands must be implemented by a sequential access device in order to meet the minimum requirements.

**Type O**

These commands are optional for sequential devices.

## 4.5. Command Sequencing

Usually the Initiator must issue a sequence of SCSI commands to be able to have the Drive perform a certain operation. As a general rule any sequence of SCSI commands are legal. There are, however, a few exceptions.

The tape and buffer system in the Drive can be in one of 4 different normal modes. In addition there are 7 exception modes. The Drive changes normal modes before starting to execute certain commands. the Drive may enter one of the exception modes when a command has failed. The action taken by the Drive when a command is received depends on the current mode. For the discussion of modes the command set is grouped into 5 different command groups according to the command mode specified in Section 4.4.

### *4.5.1. Normal Modes*

There are 4 different normal modes. The default mode after power-up or reset is always MOVE. The Drive also enters MOVE mode when a new cartridge is inserted.

**MOVE** The Drive attempts to enter the MOVE mode when a command from the move-group has been received. In this mode the data buffer is not used.

**READ** The Drive attempts to enter the READ mode when a command from the read-group has been received. The data buffer system is set up to transfer data and filemarks from the tape to the SCSI bus.

**SPACE** The Drive attempts to enter the SPACE mode when a command from the space-group has been received. The data buffer system is set up to transfer data and filemarks from the tape.

**WRITE** The Drive attempts to enter the WRITE mode when a command from the write-group has been received. The data buffer system is set up to transfer data (WRITE commands) or filemarks (WRITE FILEMARKS commands) from the SCSI bus to the tape.

The action taken by the Drive when a command is received depends on the current mode. Commands from one group can always follow a command from the same group with no special action taken. Note also that neutral-group commands can be inserted into any sequence of commands as they do not change the Drive's mode. When a command from one group follows a command from another group the Drive usually takes special action. In a few cases going from one group to another is not allowed. The command from the new group is then not executed at all. Instead it is just terminated with CHECK CONDITION.

The following table shows the actions taken by the Drive when a command from a certain group is received in the different normal modes:

| Current Mode | Next Command | Actions |
|---|---|---|
| **Move** | move-group | No action. The move-group command is executed. |
| | read-group | The buffer system is re-initialized (all buffered data is lost). The Drive then seeks the tape reference burst. The Density Code is updated. The read-group command is executed. |
| | space-group | The buffer system is re-initialized (all buffered data is lost). The Drive then seeks the tape reference burst. The Density Code is updated. The space-group command is executed. |
| | write-group | The Drive waits until the cartridge type is known (in case an autoload is in progress). If the cartridge type is suited for the selected tape format (Density Code) the buffer system is re-initialized (all buffered data is lost). The Drive then seeks the tape edge and write the tape reference burst. The Density Code is updated. The write-group command is executed. |
| **Read** | move-group | The Drive stops any read-ahead operation. The buffer system is re-initialized (all re-ahead data is lost). The move-group command is then executed. See Section 4.5.2 for a discussion of exceptions. |
| | read-group | No action. The read-group command is executed. |
| | space-group | The Drive just enters SPACE mode. The space-group command is executed. See Section 4.5.2 for a discussion of exceptions. |
| | write-group | The Drive will check if the tape is logically positioned at end-of-data (Logical End Of Partition). If the tape is at LEOP then the write-group command is executed. This will append new data after the last written block on the tape. If the tape is not at LEOP then the write-group command is terminated immediately with CHECK CONDITION status. The Error Code is set to E$BTD_WRRD (Write After Read). The Drive continues to be in READ mode. See Section 4.5.2 for a discussion of exceptions. |
| **Space** | move-group | The Drive stops any read-ahead operation. The buffer system is re-initialized (all read-ahead data is lost). The move-group command is then executed. See Section 4.5.2 for a discussion of exceptions. |
| | read-group | The Drive just enters READ mode. The read-group command is executed. See Section 4.5.2 for a discussion of exceptions. |
| | space-group | No action. The space-group command is executed. |
| | write-group | The Drive will check if the tape is logically positioned at end-of-data (Logical End Of Partition). If the tape is at LEOP then the write-group command is executed. This will append new data after the last written block on the tape. If the tape is not at LEOP then the write-group command is terminated immediately with CHECK CONDITION status. The Error Code is set to E$BTD_WRRD (Write After Read). The Drive continues to be in SPACE mode. See Section 4.5.2 for a discussion of exceptions. |

*Table: Normal Mode Actions (table to be continued...)*

| Current Mode | Next Command | Actions |
|---|---|---|
| **Write** | move-group | Data and filemarks remaining in the data buffer are written to the tape. If this operation is successful then The Drive enters MOVE mode. The move- group command is executed. See Section 4.5.2 for a discussion of exceptions. |
|  | read-group | The Drive terminates the read-group command immediately with CHECK CONDITION status. The Error Code is set to E$BTD_RDWR (Read After Write). The Drive continues to be in WRITE mode. |
|  | space-group | If the space-group command is a SPACE forward command then The Drive terminates the command immediately with CHECK CONDITION status. The Error Code is set to either E$TEM_EOR or E$TEM_EOREW depending on whether the tape is positioned before or after PSEW. The Drive continues to be in WRITE mode. In other cases The Drive enters SPACE mode and the space-group command is executed. See Section 4.5.2 for a discussion of exceptions. |
|  | write-group | A test is made to see if the current Density Code (tape format) has changed. The write-group command is executed if the tape format has not changed. |

*Table: Normal Mode Actions*

### 4.5.2. Exception Modes

When an exception has occurred in the Drive's Tape Handler, a transition is often made from one of the normal modes (move, read/verify, space and write) to an ***exception mode.*** The Drive supports 8 exception modes:

**MOVE ERROR**

The Drive enters this mode when a fatal error has been detected during execution of a Move type command. The following exceptions brings the Drive into this mode:

| | |
|---|---|
| E$TCM_CFST | Fast cartridge |
| E$TCM_CRMD | Cartridge removed |
| E$TCM_CSLW | Slow cartridge |
| E$TCM_CSTK | Stuck cartridge |
| E$TCM_ERN1 | EREN_IN was low when it should have been high |
| E$TCM_SENS | Illegal sensor condition |
| E$TCM_TRUN | Tape runout |
| E$TCM_VLT0 | WRVOLT was high when it should have been low |

**READ ERROR**

The Drive enters this mode when a fatal error has been detected during execution of a read or space-group command. The following exceptions brings the Drive into this mode:

| | |
|---|---|
| E$BHI_CPHD | Illegal Compression Header |
| E$BTD_LALG | Decompression Exception Long ID |
| E$BTD_SALG | Decompression Exception Short ID |
| E$DCM_MISC | Compression Check Miscompare Error |
| E$TCM_CFST | Fast cartridge |
| E$TCM_CRMD | Cartridge removed |
| E$TCM_CSLW | Slow cartridge |
| E$TCM_CSTK | Stuck cartridge |
| E$TCM_ERN0 | EREN_IN was high when it should have been low |
| E$TCM_ERN1 | EREN_IN was low when it should have been high |
| E$TCM_NODATA | No data found (blank cartridge) |
| E$TCM_SAF0 | SAFE* was high when it should have been low |
| E$TCM_SAF1 | SAFE* was low when it should have been high |
| E$TCM_SENS | Illegal sensor condition |
| E$TCM_TIME | Operation takes too long time |
| E$TCM_TRUN | Tape runout |
| E$TCM_VLT0 | WRVOLT was high when it should have been low |
| E$TCM_VLT1 | WRVOLT was low when it should have been high |
| E$TEM_ILTERM | Illegal termination of read data |

**WRITE ERROR**

The Drive enters this mode when a fatal error has been detected during execution of a write-group command. The following exceptions brings the Drive into this mode:

| | |
|---|---|
| E$DCM_MISC | Compression Check Miscompare Error |
| E$TCM_CFST | Fast cartridge |
| E$TCM_CRMD | Cartridge removed |
| E$TCM_CSLW | Slow cartridge |
| E$TCM_CSTK | Cartridge stuck |
| E$TCM_ERN0 | EREN_IN was high when it should have been low |
| E$TCM_ERN1 | EREN_IN was low when it should have been high |
| E$TCM_NODATA | No data found (blank cartridge) |
| E$TCM_NSIG | No data was received from tape during writr |
| E$TCM_NTEF | No tape edge found |
| E$TCM_SAF0 | SAFE* was high when it should have been low |
| E$TCM_SAF1 | SAFE* was low when it should have been high |

| | |
|---|---|
| E$TCM_SENS | Illegal sensor condition |
| E$TCM_TIME | Operation takes too long time |
| E$TCM_TRUN | Tape runout |
| E$TCM_VLT0 | WRVOLT was high when it should have been low |
| E$TCM_VLT1 | WRVOLT was low when it should have been high |
| E$TEM_ILTERM | Illegal termination of read data |
| E$WRT_APFAIL | Maximum number of Rewrites |
| E$WRT_REWRITE | Failed during enhanced Rewrite |

### UNCORRECTABLE BLOCK

The Drive enters this mode when an uncorrectable block has been detected during execution of a read or space-group command. The following exceptions brings the Drive into this mode:

E$BTD_RTRY : Read Retries Exhausted

### READ END OF PARTITION

The Drive enters this mode when Logical End Of Partition has been detected during execution of a read or space-group command. The following exceptions brings the Drive into this mode:

E$TEM_EOR : End Of Data Detected On The Current Partition (LEOP)

E$TEM_EOREW : End Of Data Detected After PSEW

### WRITE AFTER PSEW

The Drive enters this mode when Pseudo Early Warning (PSEW) has been detected during execution of a write-group command. The following exceptions brings the Drive into this mode:

E$BTD_PSEW : PSEW Detected During Write

### READ PHYSICAL END OF PARTITION

The Drive enters this mode when Physical End Of Partition has been detected during execution of a read or space-group command. The following exceptions brings the Drive into this mode:

E$TEM_PEOP : Physical End Of Partition Detected During Read

### WRITE PHYSICAL END OF PARTITION

The Drive enters this mode when Physical End Of Partition has been detected during execution of a write-group command. The following exceptions brings the Drive into this mode:

E$WRT_EOM : Physical End Of Partition Detected During Write

When the Drive has gone into one of the exception modes the response to various commands may be a little different then when the Drive is in one of the normal modes. The following table is a summary of exception mode behavior:

| Current Mode | Next Command | Actions |
|---|---|---|
| **Read Error** | move-group | The Drive enters MOVE mode. The buffer system is re-initialized (all read-ahead data is lost). The move-group command is executed. |
| | read-group | The Drive terminates the command immediately with CHECK CONDITION status. The Error Code reported will be the same as the original error code (the one that set the Drive into READ ERROR mode). The Drive remains in READ ERROR mode. A move-group command (like REWIND) brings the Drive out of this mode. |
| | space-group | Same as for read-group commands. |
| | write-group | The Drive terminates the command immediately with CHECK CONDITION status. The Error Code is set to E$BTD_WRRD (Write After Read, can't append when not at end-of-data). The Drive remains in READ ERROR mode. A move type command (like REWIND) brings the Drive out of this mode. |
| **Write Error** | move-group | The Drive enters MOVE mode. The buffer ERROR system is reinitialized (all buffered data is lost). The move-group command is executed. |
| | read-group | The Drive terminates the command immediately with CHECK CONDITION status. The Error Code reported will be the same as the original error code (the one that set the Drive into WRITE ERROR mode). The Drive continue to be in WRITE ERROR mode. A move type command (like REWIND) brings the Drive out of this mode. |
| | space-group | Same as for read-group commands. |
| | write-group | Same as for read-group commands. |
| **Uncorrectable Block** | move-group | The Drive enters MOVE mode. The buffer system is re-initialized (all read- ahead data is lost). The move-group command is executed. |
| | read-group | If the Drive was set into uncorrectable block exception mode due to an uncorrectable block detected when spacing in the reverse direction, then the current block position is unknown and a new read or space-group command will most likely fail. In other cases the Drive will be able to recover when a new read or space-group command is issued. |
| | space-group | Same as for read-group commands. |
| | write-group | The Drive terminates the command immediately with CHECK CONDITION status. The Error Code is set to E$BTD_WRRD (Write After Read, cannot append when not at end-of-data). The Drive remains in <uncorrectable block> mode. |
| **Read End-of-Partition** | move-group | The Drive enters MOVE mode. The buffer system is re-initialized (all buffered data is lost). The move-group command is executed. |
| | read-group | The Drive terminates the command immediately with CHECK CONDITION status. The Error Code reported will be the same as the original error code (the one that set the Drive into READ END OF PARTITION mode). The Drive continues to be in READ END OF PARTITION mode. |
| | space-group | If it is a SPACE forward command the actions taken is the same as for read-group commands. Other space commands will execute normally and the Drive enters SPACE mode. |
| | write-group | A write append operation is started. The Drive enters WRITE mode. The write- group command is executed normally. |

*Table: Exception Mode Actions (table to be continued...)*

| Current Mode | Next Command | Actions |
|---|---|---|
| **Write After PSEW** | move-group | The Drive enters MOVE mode. The buffer AFTER PSEW system is re-initialized (all buffered data is lost). The move-group command is executed. |
| | read-group | The Drive terminates the command immediately with CHECK CONDITION status. The Error Code is set to E$BTD_RDWR (Read After Write). The Drive continues to be in WRITE AFTER PSEW mode. |
| | space-group | If it is a SPACE forward command it will be terminated immediately with CHECK CONDITION status. The Error Code is set to E$TEM_EOREW. The Drive remains in WRITE AFTER PSEW mode. Other space-group commands will execute normally and the Drive enters SPACE mode. |
| | write-group | The write-group command is executed. It will then terminate with CHECK CONDITION. The Error Code is set to E$BTD_PSEW (Write After Pseudo Early Warning). The Drive remains in WRITE AFTER PSEW mode. |
| **Read Physical End-of-Partition** | move-group | The Drive enters MOVE mode. The buffer system is re-initialized (all buffered data is lost). The move-group PARTITION command is executed. |
| | read-group | The Drive terminates the command immediately with CHECK CONDITION status. The Error Code reported will be the same as the original error code (the one that set the Drive into READ PHYSICAL END OF PARTITION mode). The Drive remains in READ PHYSICAL END OF PARTITION mode. |
| | space-group | If it is a SPACE forward command the actions taken will be the same as for read-group commands. Other space-group commands will execute normally and the Drive enters SPACE mode. |
| | write-group | The Drive terminates the command immediately with CHECK CONDITION status. The Error Code is set to E$WRT_EOM (Write Physical End Of Partition). The Drive remains in READ PHYSICAL END OF PARTITION mode. |
| **Write Physical End-of-Partition** | move-group | The Drive enters MOVE mode. The buffer system is re-initialized (all buffered data is lost). The move-group command is executed. |
| | read-group | The Drive terminates the command immediately with CHECK CONDITION status. The Error Code reported will be the same as the original error code (the one that set the Drive into WRITE PHYSICAL END OF PARTITION mode). The Drive remains in WRITE PHYSICAL END OF PARTITION mode. |
| | space-group | Same as for read-group commands. |
| | write-group | The Drive terminates the command immediately with CHECK CONDITION status. The Error Code reported will be the same as the original error code (the one that set the Drive into WRITE PHYSICAL END OF PARTITION mode). The Drive remains in WRITE PHYSICAL END OF PARTITION mode. |

*Table: Exception Mode Actions*

# 4.6. Overlapped Command Handling

## 4.6.1. Background

While the SCSI-bus is in the BUS FREE state, any Initiator may attempt a connection to the Drive. In most cases this will happen when the Drive has completed execution of some previous command (a STATUS byte and a COMMAND COMPLETED message has been sent). As soon as the SCSI-bus becomes free, an Initiator may again select the Drive. The Drive will respond to the selection and a new command may be transferred and later executed by the Drive. This is the usual scenario where commands are executed in a serial manner. The connecting Initiators may be the same or a different Initiator.

If allowed, the Drive may disconnect from the current Initiator in the middle of a command execution. The Drive reconnects automatically at a later stage, but this will temporarily leave the SCSI-bus in the BUS FREE state, even if the current command has not completed its execution. During these periods of BUS FREE phases, any Initiator (including the original), may seize the opportunity to select the Drive. In a multi-initiator system, one Initiator may not even know that another already has established a connection to the Drive. When the Drive is connected to *two Initiators* at the same time (or the same Initiator *twice),* an *Overlapped Command situation* exists.

## 4.6.2. The Drive is Selected by the Same Initiator

When the same Initiator (that already is executing a command) selects the Drive once again, the Drive will behave as described in Section 6.5.2 (Incorrect Initiator Connection) of the *ANSI SCSI-2 Standard:*

**1)** *The Drive will respond normally during the selection phase.*

**2)** *If the ATN-line is asserted during the selection, the Drive will respond by going to the MESSAGE-OUT phase. Messages will be transferred as long as the ATN-line is asserted. The following messages will be allowed at this stage:*

*ABORT*
*BUS DEVICE RESET*
*EXTENDED MESSAGES*
*IDENTIFY*
*NOP*

*Other messages will be rejected by the Drive by going to the MESSAGE-IN phase and transferring a MESSAGE REJECT message.*

**3)** *If the Drive receives a BUS DEVICE RESET message, it will reset itself and go to the BUS FREE state (normal BUS DEVICE RESET handling). Note that this also affects (resets) the first command already under execution from the same Initiator.*

**4)** *If the Drive receives an ABORT message, it will clear the active I/O process for the selecting Initiator. Both the executing command and the new, overlapped command will be aborted. The Drive will then go to the BUS FREE state. Previously established conditions, including MODE SELECT parameters and reservations, will not be changed by the ABORT message. See Section 6.5. Abort Message Handling for further details.*

**5)** *If the ATN-line is not asserted during the selection, or if no ABORT or BUS DEVICE RESET message has been received immediately after the selection, the Drive will go to the COMMAND phase. A Command Descriptor Block will be transferred as normal. The command will not be executed. It is not required that the Drive performs the normal checks on the CDB for invalid/reserved fields. This means that the command is effectively ignored. The Drive will then go to the STATUS phase. A CHECK CONDITION status will be transferred. The Drive will then enter the MESSAGE-IN phase and transfer a COMMAND COMPLETED message. Then the Drive will enter the BUS FREE state. The generated sense data will have a Sense Key set to ABORTED COMMAND and the additional sense code/qualifier will be set to OVERLAPPED COMMANDS ATTEMPTED. The Error Code will be set to E$STE_OLAP. The first command already under execution for the same Initiator will be aborted. If the executing command, aborted by an OVERLAPPED command, is a LOCATE or SPACE, the tape will be positioned at BOT on the current partition.*

### 4.6.3. The Drive is Selected by Another Initiator

When an Initiator selects the Drive while it is executing a command for another Initiator, this new selection will be treated as normally as possible. The new selecting Initiator may not know that the Drive is already busy executing a command. The sequence of events will be as described below:

**1)** *The Drive will respond normally during the selection phase.*

**2)** *If the ATN-line is asserted during the selection, the Drive will respond by going to the MESSAGE-OUT phase. Messages will be transferred as long as the ATN-line is asserted. The following messages will be allowed at this stage:*

*ABORT*
*BUS DEVICE RESET*
*EXTENDED MESSAGES*
*IDENTIFY*
*NOP*

*Other messages will be rejected by the Drive by going to the MESSAGE-IN phase and transferring a MESSAGE REJECT message.*

**3)** *If the Drive receives a BUS DEVICE RESET message, it will reset itself and go to the BUS FREE state (normal BUS DEVICE RESET handling). Note that this also affects (resets) the command under execution for the original Initiator.*

**4)** *If the Drive receives an ABORT message, it will clear the active I/O process for the selecting Initiator. The pending data and status for the original Initiator will not be cleared. The Drive will then go to the BUS FREE state. Previously established conditions, including MODE SELECT parameters and reservations will not be changed by the ABORT message.*

**5)** *If the ATN-line is not asserted during the selection, or if no ABORT or BUS DEVICE RESET message has been received immediately after the selection, the Drive will go to the COMMAND phase. A Command Descriptor Block will be transferred as normal.*

**6)** *The Drive will go to the STATUS phase after having received the Command Descriptor Block. The command will not be executed. It is not required that the Drive performs the normal checks on the CDB for invalid/reserved fields. The transferred STATUS byte will be a* ***BUSY*** *status. The Drive will then enter the MESSAGE phase and transfer a COMMAND COMPLETED message. The Drive will then enter the BUS FREE state.*

# 5. Status Bytes

A status byte will be sent from the Drive to the Initiator during the STATUS phase at the termination of each command unless the command is cleared by an ABORT message, by a BUS DEVICE RESET message or a SCSI-bus reset condition.
The Drive must support the Status Bytes shown in the table below:

| Status Byte Name | Hex Code |
|---|---|
| BUSY STATUS | 08 |
| CHECK CONDITION STATUS | 02 |
| GOOD STATUS | 00 |
| INTERMEDIATE STATUS | 10 |
| RESERVATION CONFLICT STATUS | 18 |

*Table: The Status Set*

**BUSY Status**

The Drive is busy executing a command.

Two different BUSY-situations may occur:

1) The Drive is already executing a command from another Initiator.

2) The Drive is executing an Immediate-type command and the BSY-option is turned on (see the Miscellaneous Parameter Page in the MODE SELECT command). This condition will prevail until the Immediate-type command has completed its execution (or the BSY-option is turned off).

The Initiator may use the TEST UNIT READY command to determine when the Drive is again ready for new commands.

**CHECK CONDITION Status**

An abnormal condition has occurred. The Initiator should issue a REQUEST SENSE command to get further information.

**GOOD Status**

The requested operation (the last command) was completed successfully.

**INTERMEDIATE Status**

This status is sent after command in series of linked commands, as long as the command completed successfully.

**RESERVATION CONFLICT Status**

The status is sent to an Initiator that attempts to access the Drive when it is reserved for another Initiator.

*This Page Intentionally Left Blank*

# 6. Message System

The message system allows communication between an Initiator and the Drive for the purpose of physical path management.

The physical path may be broken and re-established several times during the execution of a SCSI command if the Initiator has granted the Drive the privilege of disconnection.

The first message sent by the Initiator after the SELECTION phase will be either the IDENTIFY, ABORT or BUS DEVICE RESET message. The IDENTIFY message may be immediately followed by other messages. The IDENTIFY message establishes the physical path for a particular logical unit specified by the Initiator. Since the Drive has only one logical unit, the specified logical unit number must always be zero. The IDENTIFY message may also grant the Drive disconnection privilege. If the Drive is allowed to disconnect, it may do so by transferring a DISCONNECT message and release the SCSI-bus (by entering the BUS FREE phase).

The Drive will always at some later stage re-establish the physical path by reselecting the Initiator. After the RESELECTION phase, the Drive's first message will be IDENTIFY. This allows the physical path to be re-established for the Drive's specified logical unit number. The Drive will always identify itself with a logical unit number of zero.

Whenever a physical path is established in an Initiator that is utilizing disconnection and reconnection, the Initiator will ensure that the active pointers of the physical path are equal to the saved pointers for that particular logical unit number (an implied restore operation will occur as a result of a reselection).

When the Drive has completed the execution of a SCSI command (successfully or not) it will signal the Initiator that it is about to break the physical path for good (for this selection sequence) by transferring a COMMAND COMPLETE message. The Drive will then enter the BUS FREE phase.

## 6.1. Message-In

The Drive supports the Message-In Bytes shown in the table below:

| Message Name | Input/ Output | Hex Code |
|---|---|---|
| COMMAND COMPLETE | I | 00 |
| DISCONNECT | I | 04 |
| EXTENDED MESSAGE | I/O | 01 |
| IDENTIFY | I/O | 80 |
| LINKED COMMAND COMPLETE | I | 0A |
| LINKED COMMAND COMPLETE W/FLAG | I | 0B |
| MESSAGE REJECT | I/O | 07 |
| RESTORE POINTERS | I | 03 |
| SAVE DATA POINTER | I | 02 |

*Table: The Message-In Set*

**COMMAND COMPLETE**

This message will be sent from the Drive to the Initiator to indicate that the execution of a command has terminated. Valid Status will have been sent. This message will always be sent next to a CHECK CONDITION, GOOD or RESERVATION CONFLICT status byte. After successfully sending this message, the Drive will enter the BUS FREE phase.

**DISCONNECT**

This message will be sent from the Drive to the Initiator to inform the Initiator that the Drive is about to disconnect. This message is always sent second to the SAVE DATA POINTER message. After successfully sending this message, the Drive will enter the BUS FREE phase.

**EXTENDED MESSAGE**

This message is sent by the Drive to the Initiator as the first byte of a multibyte message. See Section 6.3 for a description of the extended messages.

**IDENTIFY**

IDENTIFY will be sent from the Drive to the Initiator immediately after a reconnect (this will be the IDENTIFY LUN zero form of IDENTIFY with no disconnect bit set, code 80h).

**LINKED COMMAND COMPLETE**

This message will be sent from the Drive to the Initiator to indicate that the execution of a linked command (without the FLAG bit set) has completed. This message will always be sent next to INTERMEDIATE status.

**LINKED COMMAND COMPLETE W/FLAG**

This message will be sent from the Drive to the Initiator to indicate that the execution of a linked command (with the FLAG bit set) has completed. This message will always be sent next to INTERMEDIATE status.

**MESSAGE REJECT**

The Drive will send MESSAGE REJECT to the Initiator if it receives a message other than the messages listed in the Message-Out table (see Section 6.2). The Drive will also send MESSAGE REJECT if a INITIATOR DETECTED ERROR message is received following a DATA-OUT phase. The Drive will send MESSAGE REJECT if a MESSAGE PARITY ERROR message is received when no message has been sent. Finally the Drive will send MESSAGE REJECT if an IDENTIFY message is received following phases other than Selection phase.

**RESTORE POINTERS**

This message will be sent by the Drive to the Initiator before the Drive attempts a retransfer of Data or a Status Byte.

**SAVE DATA POINTER**

This message will be sent by the Drive to the Initiator before sending the DISCONNECT message.

## 6.2. Message-Out

The Drive supports the Message-Out Bytes shown in the table below:

| Message Name | Input/ Output | Hex Code |
|---|---|---|
| ABORT | O | 06 |
| BUS DEVICE RESET | O | 0C |
| EXTENDED MESSAGE | I/O | 01 |
| IDENTIFY | I/O | 80 |
| INITIATOR DETECTED ERROR | O | 05 |
| MESSAGE PARITY ERROR | O | 09 |
| MESSAGE REJECT | I/O | 07 |
| NO OPERATION | O | 08 |

*Table: The Message-Out Set*

**ABORT** Abort the current operation. The Drive must go to the BUS FREE phase. No Status or ending Message will be sent (See Section 6.4).

**BUS DEVICE RESET** Abort the current operation. The Drive must go to the BUS FREE phase. No Status or ending Message will be sent. Possible data in the data buffer will not be transferred to the Initiator or written to the tape. Possible pending error conditions will be cleared.
This message will in other respects have the same effect as a SCSI-bus RESET (if possible the power-up initialization procedure will be shortened down to an absolute minimum).

**EXTENDED MESSAGE** This message is sent by the Initiator to the Drive as the first byte of a multibyte message. See Section 6.3 for a description of the extended messages.

**IDENTIFY** When transferred from the Initiator to the Drive, it will accept two forms of IDENTIFY; disconnect allowed on LUN zero, and disconnect not allowed on LUN zero.

| BIT 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Idfy | DscP | LUNTAR | R | R | LUNTRN | | |

*Table: The IDENTIFY Message*

**Idfy** The Identify bit is always set in the IDENTIFY message.

**DscP** A Disconnect Privilege bit of one specifies that the Initiator has granted the Drive the privilege of disconnecting. A DscP-bit of zero means that the Drive will not disconnect.

**LUNTAR** The Logical Unit Target bit MUST be set to zero to specify that the IDENTIFY message is directed to a Logical Unit (LUN) (The Drive does not support Target Routines).

**LUNTRN** The Logical Unit Number bit MUST be set to zero because the Drive supports only Logical Unit 0.

**INITIATOR DETECTED ERROR** The Initiator may send this message to the Drive to inform that an error has been detected in the Initiator. A re-transfer will then be performed. See the section on General Exception Handling for further details.

**MESSAGE PARITY ERROR** The Initiator may send this message to the Drive to inform that the last transferred message had a parity error. A re-transfer will then be performed. See the section on General Exception Handling for further details.

**MESSAGE REJECT** The Initiator may send this message to the Drive to indicate that the last message transferred from the Drive was inappropriate or not implemented (See Section 6.3).

**NO OPERATION** This message will simply be ignored

## 6.3. Extended Message

A value of one (01h) of the first byte of a message indicates the beginning of a multiple-byte extended message. The minimum number of bytes sent for an extended message is three. The extended message format is shown in the table below:

| Byte | Value | Description |
|---|---|---|
| 0 | 01h | Extended Message |
| 1 | n | Extended Message Length |
| 2 | y | Extended Message Code |
| 3 - n+1 | x | Extended Message Arguments |

*Table: Extended Message Format*

The extended message length specifies the length in bytes of the extended message code plus the extended message arguments to follow. Therefore, the total length of the message is equal to the extended message length plus two. A value of zero for the extended message length indicates 256 bytes to follow.

The extended messages supported by the Drive is shown in the table below. The extended messages are described in detail in Section 6.3.

| Extended Message Code | Description |
|---|---|
| 01h | Synchronous Data Transfer Request |

*Table: Extended Message Codes*

If a parity error is detected during a MESSAGE-OUT phase, the Target will consume all the remaining bytes in the message and ask for a retransfer of the whole message.

If ATN is de-asserted before the expected number of bytes is transferred, the Target will send a MESSAGE REJECT message back to the Initiator. If ATN is still asserted after the expected number of bytes is transferred, the Target goes to the BUS FREE state.

If a parity error is detected during a MESSAGE-IN phase, the Initiator will signal a MESSAGE PARITY error. The Target responds by transferring the whole message once more.

### *6.3.1. Synchronous Data Transfer Request Message*

Synchronous Data Transfer Request (SDTR) message exchange can be initiated by the Initiator to change the data transfer agreement. The default data transfer mode is asynchronous data transfer mode.

The default transfer mode is entered at power on, after a BUS DEVICE RESET message or a hard reset condition.

A SCSI Initiator may initiate an SDTR message whenever it is appropriate to negotiate a new data transfer agreement (either synchronous or asynchronous). The Drive will never respond to an SDTR message with a MESSAGE REJECT message. Re-negotiation at every selection is not recommended, since a significant performance impact is likely.

When the Drive receives a SDTR message it will evaluate the two arguments, the transfer period and the REQ/ACK offset, to determine if it is able to receive data successfully with these values. If the Drive accepts the two arguments from the Initiator, it will return the two accepted parameters in a new SDTR message back to the Initiator. This ends the exchange of the SDTR messages and synchronous data transfer is established between Initiator and the Drive.

If one or both parameters are not accepted by the Drive, a new SDTR message with arguments adjusted to meet the Drive's requirements is returned back to the Initiator. It is then up to the Initiator to evaluate the new proposal. If these parameters are accepted, no more SDTR messages are sent and synchronous data transfer is established. If the Initiator does not accept the values from the Drive, it can either send a new SDTR message to start a new negotiation process, or send a MESSAGE REJECT message to signal that the data transfer mode to be used is asynchronous data transfer mode.

A Synchronous Data Transfer Request message exchange process will not be initiated by the Drive. The only way to establish a synchronous data transfer mode between an Initiator and the Drive is, since the Drive take no initiative for such a request, totally controlled by the Initiator.

The Synchronous Data Transfer Request Message has the following format:

| Byte | Value | Description |
|---|---|---|
| 0 | 01h | Extended Message |
| 1 | 03h | Extended Message Length |
| 2 | 01h | Synchronous Data Transfer Request Code |
| 3 | m | Transfer Period (m x 4 ns) |
| 4 | x | REQ/ACK Offset |

*Table: Synchronous Data Transfer Request*

**Transfer Period**

The Transfer Period is the minimum time allowed between leading edges of successive REQ pulses and of successive ACK pulses, set by the device specifications for successful reception of data. The minimum Transfer Period is hardware dependent and is either 208 nanoseconds, m = 34h (52) or 260 nanoseconds, m = 41h (65).

**REQ/ACK Offset**

The REQ/ACK Offset is the maximum number of REQ pulses allowed to be outstanding before the corresponding ACK pulse is received at the Target. The value is limited by the size of the device's reception buffer. A REQ/ACK off- set of zero indicates asynchronous data transfer mode. The maximum REQ/ACK offset of the Drive is 08h (8).

## 6.4. Message Reject Message Handling

If after a MESSAGE-IN phase the Initiator asserts ATN and transfers a REJECT message, the Drive will react as shown in the table below:

| Last Message-In | Action Taken |
|---|---|
| COMMAND COMPLETE | Ignore the REJECT message |
| DISCONNECT | The Drive will not disconnect any more while executing the current command. The Drive may again attempt disconnection for the next command (if the Initiator has signalled that disconnection is allowed) |
| IDENTIFY | Ignore the REJECT message |
| LINKED COMMAND COMPLETE | Ignore the REJECT message |
| LINKED COMMAND COMPLETE W/FLAG | Ignore the REJECT message |
| MESSAGE REJECT | Ignore the REJECT message |
| SAVE DATA POINTER | The Drive will not disconnect any more while executing the current command. The Drive may again attempt disconnection for the next command (if the Initiator has signalled that disconnection is allowed) |

*Table: Response To MESSAGE REJECT*

## 6.5. Abort Message Handling

If after a MESSAGE-IN phase the Initiator asserts ATN and transfers an ABORT message, then the Drive will immediately go to the BUS FREE phase. No Status or ending Message will be sent. The Drive will remain in its current state (MODE settings and current tape position must be kept).
When receiving a new command, the Drive will be able to continue where that last aborted command left off.
See Section 4.6. (Overlapped Command Handling) for a description of ABORT in overlapped command situations.
See also the Section on LOGICAL CHARACTERISTICS for details on ATN signal handling. The following is true for all commands:

- *If the ABORT message is transferred before the Drive enters the Command phase, then the Drive will just go to the BUS FREE phase, effectively ignoring the whole selection.*

- *If the ABORT message is transferred immediately after the Command phase (ATN must be asserted during command transfer), then the Drive will just go to the BUS FREE phase, effectively ignoring the whole command. This is true even if the Drive detects errors in the command block (like Bus Parity Error).*

- *If the Drive is disconnecting immediately after the command transfer, and an ABORT is issued in connection with the disconnected phase sequence, the ABORT is treated as if Attention was set in the command phase.*

- *If the Drive detects Attention while sending Status or Command Complete message and receives an ABORT-message, the command has already completed. The Drive will just go to the BUS FREE phase after the ABORT-message has been transferred.*

The following table describes ABORT-handling in *non-overlapped command situations* for all commands when the ABORT-message is transferred at any later stage in the phase sequencing.

| **Command** | **Action** |
|---|---|
| COPY | Once the Drive has disconnected, the original Initiator will not be able to issue non-overlapped ABORT before the COPY command has completed and the Drive reconnects to send STATUS and COMMAND COMPLETE message. |
| ERASE<br>LOAD/UNLOAD<br>REWIND<br>WRITE FILEMARK | **Immediate Mode:**<br>The command does not disconnect, but starts execution immediately after the command has been transferred without an ABORT and before STATUS and COMMAND COMPLETE are sent. If an ABORT is received during the last two phases, the command will complete in Immediate mode.<br><br>**Non-immediate Mode:**<br>The Initiator cannot issue any non-overlapped ABORT to these commands after they have disconnected and before they reconnect to send STATUS and COMMAND COMPLETE. Once the reconnection starts, the command has already completed. |
| INQUIRY<br>LOG SENSE<br>MODE SENSE<br>READ BLOCK LIMITS<br>READ POSITION<br>REQUEST SENSE | If the Initiator sets Attention during the DATA-IN phase of these commands, the requested parameters have already been transferred and the command has completed execution. The Drive will go to the BUS FREE phase after the MESSAGE PHASE and no STATUS or COMPLETE message is sent. |
| MODE SELECT | If the Initiator sets Attention during the DATA-OUT phase of this command, or while the Drive is disconnecting after the parameter transfer, the received Mode Parameters will not be set. The command is aborted and no STATUS or COMPLETE message is sent. If the Attention is set in any later phase, the Mode Parameters are already set. It is, however, recommended that the Host reissues the MODE SELECT to be certain the parameters are properly set. |
| PREVENT/ALLOW MEDIA REMOVAL<br>RELEASE UNIT<br>RESERVE UNIT<br>TEST UNIT READY | These commands do not disconnect. After the COMMAND PHASE the Attention can first be set during STATUS or COMPLETED MESSAGE transfer. |

*Table: Non-overlapped ABORT Message Handling (to be continued...)*

| Command | Action |
|---|---|
| READ | If Attention is set during the data transfer or during the subsequent Save Pointer or the Disconnect message transfer, and the Drive receives an ABORT message, the data transferred in the last transfer will **not** be considered transferred correctly and will **not** be removed from the buffer. The Drive will go BUS FREE and the READ operation is aborted. A new READ command will resend the last data from the aborted READ. |
| READ BUFFER | If Attention is set in any phase after the COMMAND PHASE and an ABORT message is issued, the command is immediately aborted. None of the transferred data is removed from the buffer. The command can be issued again to retransfer the data including the data not transferred when the command was aborted. |
| WRITE BUFFER | If Attention is set in any phase after the COMMAND PHASE and an ABORT message is issued, the command will be aborted immediately. All data transferred up to the ABORT message will be written to the buffer. If the same command is reissued, the new data will overwrite the old data. |
| RECOVER BUFFERED DATA | This command will behave like a READ command. |
| LOCATE<br>SPACE<br>SEND DIAGNOSTICS | These commands will always disconnect before command execution is started, and stay disconnected until the command execution has completed. If Attention is set during command transfer or during the DISCONNECT PHASE sequence, and an ABORT is issued, the command will be ignored. If Attention is set in any later phase, the command has already completed execution. |
| VERIFY | See description for the READ command. |
| WRITE | If Attention is set during the data transfer or during the subsequent Save Pointer or the DISCONNECT message transfer, and the Drive receives an ABORT message, the last data burst will **not** be considered transferred correctly. The data will **not** be written to the tape. The Drive will go BUS FREE and the WRITE operation is aborted. A new WRITE command must resend the last data from the aborted WRITE. |

*Table: Non-overlapped ABORT Message Handling*

The following table describes ABORT-handling in *overlapped command situations*.

| Command | Action |
|---|---|
| ERASE | In most cases the abortion will immediately bring the Drive to the BUS FREE phase. The tape movement will, however, continue until its normal completion. |
| LOAD/UNLOAD | In most cases the abortion will immediately bring the Drive to the BUS FREE phase. The tape movement will, however, continue until its normal completion. |
| READ | In most cases this command will be aborted immediately. Note, however, that while seeking the reference burst (when starting a READ operation on a new cartridge), the command will not abort until the reference burst has been found and the tape format has been determined. The Initiator may resume the READ operation by issuing a new READ command. |
| READ BUFFER | This command will be aborted immediately. |
| RECOVER BUFFERED DATA | This command will be aborted immediately. |
| REWIND | In most cases the abortion will immediately bring the Drive to the BUS FREE phase. The tape movement will, however, continue until its normal completion. |
| VERIFY | In most cases this command will be aborted immediately. Note, however, that while seeking the reference burst (when starting a READ operation on a new cartridge), the command will not abort until the reference burst has been found and the tape format has been determined. The Initiator may resume the VERIFY operation by issuing a new VERIFY command. |
| WRITE | In most cases this command will be aborted immediately. The Initiator may resume the WRITE operation by issuing a new WRITE command. |
| WRITE BUFFER | This command will be aborted immediately |
| WRITE FILEMARKS | In most cases this command will be aborted immediately. The Initiator may resume the WRITE FILEMARK operation by issuing a new WRITE  FILEMARKS command. |

*Table: Overlapped ABORT Message Handling*

*This Page Intentionally Left Blank*

# 7. General Exception Handling

## 7.1. Error Codes

When an error condition is detected in the Drive this error condition will result in generation of an internal error code. Every detectable error condition has assigned a unique error code.

Usually an Error Code is used to generate Sense Data. Some Error Codes are, however, used only for internal error processing (like E$STP_SRFL, Reselection Time-out). When generating Sense Data the Error Code will be used to generate all other error codes, keys and bits (the Sense Key, FMK, EOM and ILI bits).

The table on the following pages maps Error Codes to various error indicators found in the REQUEST SENSE Data List and is used in the following way :

### How to interpret the Error Code Table

*If you, for example, seek information about an "**E$BTD_CFMT**"-error, look it up in the alphabetical "**Name**"-column. The "**Description**"-column indicates an "**Incompatible media type**"-error. For more details about the "**SK**" = "**Sense Key**" **and** "**AS/AQ**" = "**Additional Sense Code and Qualifier**"-columns, see Chapter 24. Request Sense, Sections 24.4. and 24.5.*

| Name | FMK | EOM | ILI | SK | AS | AQ | Description |
|---|---|---|---|---|---|---|---|
| | | | | | | | |
| | | | | | | | |
| E$BTD_CFMT | | | | 5h | 30h | 00h | Incompatible media type |
| | | | | | | | |

| Name | FMK | EOM | ILI | SK | AS | AQ | Description |
|---|---|---|---|---|---|---|---|
| E$BHI_CPHD | | | | 3h | 71h | 00h | Illegal Compression Header |
| E$BTD_APUF | | | | 3h | 2Dh | 00h | WRITE append incomplete frame |
| E$BTD_CFMT | | | | 5h | 30h | 00h | Incompatible media type |
| E$BTD_FIMK | YES | | | 0h | 00h | 01h | Filemark detected |
| E$BTD_IWSD | | | | 5h | 50h | 00h | Illegal Write sequence in dual partition |
| E$BTD_LALG | | | | 3h | 71h | 00h | Decompression Exception Long Algorithm |
| E$BTD_PBOP | | YES | | 0h | 00h | 04h | Physical beginning of partition detected during SPACE |
| E$BTD_PSEW | | YES | | 0h | 00h | 02h | Pseudo early warning (PSEW) detected during WRITE |
| E$BTD_RDWR | | | | 5h | 2Ch | 00h | READ command after WRITE command |
| E$BTD_RTRY | | | | 3h | 11h | 01h | READ retries exhausted |
| E$BTD_SALG | | | | 3h | 70h | FFh | Decompression Exception Short Algorithm |
| E$BTD_SEMK | YES | | | 0h | 00h | 03h | Setmark detected during READ/SPACE |
| E$BTD_SPEW | | YES | | 0h | 00h | 00h | Pseudo Early Warning (PSEW) detected during SPACE |
| E$BTD_TFMT | | | | 5h | 30h | 00h | Incompatible tape format, cannot append |
| E$BTD_VRFY | | | | Eh | 1Dh | 00h | Compare error on VERIFY |
| E$BTD_WPRO | | | | 7h | 27h | 00h | Write protected cartridge |
| E$BTD_WRRD | | | | 5h | 50h | 00h | WRITE command after READ command |
| E$DCM_MISC | | | | 3h | 0Ch | 04h | Compression Check Miscompare (SLR5 4.0/8.0GB and SLR4 2.5/5.0GB (TDC 4222) only) |
| E$SIE_CHDF | | | | 5h | 26h | 00h | Copy, illegal copy function |
| E$SIE_CHDI | | | | 5h | 26h | 00h | Copy, bad header |
| E$SIE_CHDN | | | | 5h | 1Ah | 00h | Copy, truncated header |
| E$SIE_CILC | | | | 5h | 2Bh | 00h | Copy, cannot execute since host cannot disconnect |
| E$SIE_CODD | | | | 5h | 26h | 00h | Copy, inexact segment; odd number of blocks |
| E$SIE_CPDT | | | | Ah | 44h | 00h | Copy, internal CHECK CONDITION |
| E$SIE_CRES | | | | 5h | 26h | 00h | Copy, inexact segment; tape residual |
| E$SIE_CSGA | | | | 5h | 21h | 00h | Copy, address out of range |
| E$SIE_CSGI | | | | 5h | 26h | 00h | Copy, bad ID or LUN |
| E$SIE_CSGP | | | | 5h | 1Ah | 00h | Copy, truncated descriptor |
| E$SIP_CIBS | | | | Ah | 00h | 00h | Copy, Target status not GOOD or CHECK CONDITION |
| E$SIP_CICH | | | | Ah | 00h | 00h | Copy, Target status is CHECK CONDITION |

*Table: Error Codes (to be continued...)*

| Name | FMK | EOM | ILI | SK | AS | AQ | Description |
|---|---|---|---|---|---|---|---|
| E$SIP_CIDP | | | | 4h | 47h | 00h | Copy, parity error in parameter |
| E$SIP_CIDT | | | | 4h | 47h | 00h | Copy, parity error in data |
| E$SIP_CILB | | | | 5h | 00h | 00h | Copy, Target illegal block size |
| E$SIP_CISE | | | | Bh | 45h | 00h | Copy, Target selection timeout |
| E$SIP_CISQ | | | | Bh | 4Ah | 00h | Copy, Target phase sequence error |
| E$STE_BUSY | | | | 2h | 00h | 00h | Drive busy |
| E$STE_CSEQ | | | | 5h | 2Ch | 00h | Command sequence error |
| E$STE_ICOP | | | | 5h | 20h | 00h | Invalid command operation code |
| E$STE_IFIC | | | | 5h | 24h | 00h | Invalid field in CDB |
| E$STE_IFIP | | | | 5h | 26h | 00h | Invalid field in parameter list |
| E$STE_ILLN | | | YES | 0h | 00h | 00h | Illegal length indication |
| E$STE_ILOD | | | | 2h | 04h | 01h | Microcode downld. illeg. during load/retension |
| E$STE_MCHN | | | | 6h | 3Fh | 0ih | Unit attention, microcode changed |
| E$STE_MPCH | | | | 6h | 2Ah | 01h | Unit attention, mode parameters changed |
| E$STE_NCAR | | | | 2h | 3Ah | 00h | No cartridge present |
| E$STE_NLOD | | | | 2h | 3Ah | 00h | Cartridge not loaded |
| E$STE_NRRT | | | | 6h | 28h | 00h | Unit attention, media changed |
| E$STE_OLAP | | | | Bh | 4Eh | 00h | Overlapped commands attempted |
| E$STE_PLEN | | | | 5h | 1Ah | 00h | Parameter list length error |
| E$STE_PWRN | | | | 6h | 29h | 00h | Unit attention, power-up |
| E$STE_RECV | | | | 1h | 17h | 01h | Recovered error in last command |
| E$STE_REOB | | YES | | 0h | 00h | 02h | Recover end of buffer |
| E$STE_SREV | | | | 3h | 11h | 00h | Read retries exhausted during reverse |
| E$STE_SRST | | | | 6h | 29h | 00h | Unit attention, SCSI reset |
| E$STE_ULUN | | | | 5h | 25h | 00h | Unsupported LUN |
| E$STM_BUFFER | | | | 4h | 40h | 80h | Selftest buffer error |
| E$STM_CPU | | | | 4h | 40h | D3h | Selftest CPU error |
| E$STM_DRVCON | | | | 4h | 40h | 90h | Selftest drive controller error |
| E$STM_EDC | | | | 4h | 40h | B0h | Selftest EDC controller error |
| E$STM_EEPROM | | | | 4h | 40h | A0h | Selftest EEPROM error |
| E$STM_EPROM | | | | 4h | 40h | C0h | Selftest EPROM error |
| E$STM_EXTRAM | | | | 4h | 40h | D0h | Selftest external RAM error |
| E$STM_INTRAM | | | | 4h | 40h | E0h | Selftest internal RAM error |
| E$STM_READ | | | | 4h | 40h | D1h | Selftest error |
| E$STM_SCSI | | | | 4h | 40h | F0h | Selftest SCSI controller error |

*Table: Error Codes (to be continued...)*

| Name | FMK | EOM | ILI | SK | AS | AQ | Description |
|---|---|---|---|---|---|---|---|
| E$STM_SDCP | | | | 4h | 40h | F8h | Selftest compression controller error (SLR5 4.0/8.0GB and SLR4 2.5/5.0GB (TDC 4222) only) |
| E$STM_WRITE | | | | 4h | 40h | D2h | Selftest error |
| E$STP_COMP | | | | 4h | 47h | 00h | Parity error in CDB |
| E$STP_DTAP | | | | 4h | 47h | 00h | Parity error in data |
| E$STP_IDMR | | | | Bh | 48h | 00h | INITIATOR DETECTED ERROR message received |
| E$STP_MSGP | | | | 4h | 47h | 00h | Parity error in message |
| E$STP_PARP | | | | 4h | 47h | 00h | Parity error in parameter data |
| E$TCM_CFMT | | | | 5h | 30h | 00h | Incompatible media type reported when Read, Verify or Erase is attempted on an incompatible tape |
| E$TCM_CFST | | | | 3h | 52h | 00h | Fast cartridge |
| E$TCM_CRMD | | | | 3h | 3Ah | 00h | Cartridge removed |
| E$TCM_CSLW | | | | 3h | 52h | 00h | Slow cartridge |
| E$TCM_CSTK | | | | 3h | 52h | 00h | Stuck cartridge |
| E$TCM_MEDERR | | | | 3h | 30h | 01h | Cannot read, unknown tape format |
| E$TCM_NODATA | | | | 8h | 14h | 00h | No data found during READ/SPACE |
| E$TCM_NSIG | | | | 3h | 03h | 02h | No signal during Write |
| E$TCM_NTEF | | | | 3h | 52h | 00h | No tape edge found |
| E$TCM_SAF1 | | | | 4h | 40h | 9Ah | SAFE* was low while ERAEN2* and SAFE were both low |
| E$TCM_SAF0 | | | | 4h | 40h | 9Bh | SAFE* was high while ERAEN2* or SAFE was high |
| E$TCM_VLT1 | | | | 4h | 40h | 9Ch | WRVOLT was above the safe maximum low voltage with WREN not enabled |
| E$TCM_VLT0 | | | | 4h | 40h | 9Dh | WRVOLT was below the minimum operating voltage with WREN enabled |
| E$TCM_ERN1 | | | | 4h | 40h | 9Eh | EREN_IN was high while EREN was low |
| E$TCM_ERN0 | | | | 4h | 40h | 9Fh | EREN_IN was low while EREN was high |
| E$TCM_SENS | | | | 3h | 52h | 00h | Illegal sensor condition |
| E$TCM_TIME | | | | 3h | 52h | 00h | Operation time-out |
| E$TCM_TRUN | | | | 3h | 52h | 00h | Tape run-out |
| E$TEM_EOR | | | | 8h | 00h | 05h | Logical end of partition detected during READ/SPACE |
| E$TEM_EOREW | | YES | | 8h | 00h | 05h | Logical end of partition detected after Pseudo Early Warning (PSEW) marker during READ/SPACE |
| E$TEM_ILTERM | | | | 3h | 11h | 01h | Illegal termination of last block on tape |
| E$TEM_PEOP | | YES | | 3h | 00h | 02h | Physical end of partition detected during READ/SPACE |

*Table: Error Codes (to be continued...)*

| Name | FMK | EOM | ILI | SK | AS | AQ | Description |
|---|---|---|---|---|---|---|---|
| E$THI_PARITY | | | | 4h | 40h | 80h | Buffer parity error |
| E$WRT_APFAIL | | | | 3h | 50h | 01h | WRITE append failure |
| E$WRT_EOM | | YES | | Dh | 00h | 02h | WRITE to physical end of partition |
| E$WRT_REWRITE | | | | 3h | 03h | 02h | WRITE retries exhausted |

*Table: Error Codes*

**NAME** Mnemonic for the error condition

**FMK** Set to "YES" if the Filemark bit (FMK) is set to one

**EOM** Set to "YES" if the End of Media bit (EOM) is set to one

**ILI** Set to "YES" if the Illegal Length bit (ILI) is set to one

**SK** Hexadecimal code value presented as Sense Key in the REQUEST SENSE data list

**AS** Additional Sense Code, hexadecimal

**AQ** Additional Sense Code Qualifier, hexadecimal

## 7.2. Error Conditions for All Commands

When an Initiator accesses the Drive, there are a number of error conditions that may occur regardless of the command the Initiator attempts to issue. This section summarizes all such error conditions. See the separate sections for details. For command specific error conditions, please see the Exception Handling Section for the actual command.

**Bus Parity Error**

*Every command will be terminated with CHECK CONDITION status if a bus parity error is detected in the Command Descriptor Block, data out or message out. See Section 7.6. for details.*

**BUSY Status**

*The Drive is busy executing a command.*

*Two different BUSY-situations may occur:*

*1) The Drive is already executing a command.*

*2) The Drive is executing an Immediate-type command and the BSY-option is turned on (see the Miscellaneous Parameters List in the MODE SELECT command). This condition will prevail until the Immediate-type command has completed its execution (or the BSY-option is turned off).*

*The Initiator may use the TEST UNIT READY command to determine when the Drive is again ready for new commands.*

**Initiator Detected Error Message**

*Every command may be terminated if an INITIATOR DETECTED ERROR message is received. The Error Code will be set to E$STP_IDMR. See Section 7.6. for details.*

**Invalid Command Operation Code**

*If the Command Operation Code (byte 0 of the CDB) is not in the range of supported Command Operation Codes, the command will be terminated with CHECK CONDITION Status. The Error Code will be set to E$STE_ICOP.*

**Message Parity Error Message**

*Every command may be terminated if an MESSAGE PARITY ERROR message is received. The Error Code will be set to E$STP_MSGP. See Section 7.6. for details.*

**Overlapped Commands**

*If an Initiator issues a new command while it still has a command under execution by the Drive, the first command is aborted and the new command is terminated with CHECK CONDITION. The Error Code will be set to E$STE_OLAP.*

**Reservation Conflict**

*Every command except INQUIRY, REQUEST SENSE and RELEASE UNIT will be terminated with RESERVATION CONFLICT status if the Drive has been reserved for another Initiator.*

**Reserved Field**

*Every command will be terminated with CHECK CONDITION if one or more reserved bit, field, or byte is not zero. The Error Code will be set to E$STE_IFIC or E$STE_IFIP.*

**Unit Attention** *Each command except INQUIRY and REQUEST SENSE will be terminated with CHECK CONDITION status due to a Unit Attention Condition.*

*The Error Code will be set to E$STE_PWRN after power-up, and to E$STE_SRST after a SCSI-reset. The Error Code will be set to E$STE_NRRT when a cartridge has been changed and to E$STE_MPCH when the MODE SELECT parameters have been changed. The Error Code generated after a microcode change is E$STE_MCHN.*

**Unsupported LUN** *The Drive supports only Logical Unit 0. The LUN field in the Command Descriptor Block and the IDENTIFY message must always be set to zero. All commands except INQUIRY and REQUEST SENSE will terminate with CHECK CONDITION if the LUN field is not set to zero. The Error Code will be set to E$STE_ULUN.*

## 7.3. Deferred Errors

A deferred error is an error that occurs on an Immediate or Buffered type command after that command has terminated with GOOD status and before the next command has started execution. If the error occurs after the next command has started execution, the error will be reported as a normal non-deferred error for that command.

All commands except INQUIRY and REQUEST SENSE may be terminated due to deferred errors.

When a Deferred Error has been detected in the Drive, the first Initiator to access the Drive will have its command terminated with CHECK CONDITION even if this Initiator is different from the Initiator that issued the command that failed. To avoid this situation an Initiator may use the RESERVE UNIT command to have exclusive access and then the WRITE FILEMARKS (with filemark count equal to zero if necessary) to synchronize with the Drive before letting other Initiators access the Drive again.

NOTE:
The VADD bit will not be set and the Information Bytes will not be valid when a Deferred Error has been detected.

**Append Failure** *The Drive was not able to append new data to data already existing on the tape. This is a fatal error. Unwritten data and filemarks may be left in the data buffer. See also the Exception Handling Section for the WRITE and WRITE FILEMARKS commands. The Error Code will be set to E$WRT_APFAIL*

**Buffer Parity Error** *All commands transferring data to or from the data buffer will be terminated with CHECK CONDITION if a buffer parity error is detected. See Section 7.6. for details*

**Cartridge Error** *See section on Error Conditions For Media Access Commands*

**Head Servo Error** *See section on Error Conditions For Media Access Commands*

**No Cartridge** *See section on Error Conditions For Media Access Commands*

**Non-Recoverable Write Error**

*A non-recoverable write error has occurred while writing data or filemarks. This is a fatal error. Unwritten data and filemarks may be left in the data buffer. See also the Exception Handling Section for the WRITE and WRITE FILEMARKS commands. The Error Code will be set to E$WRT_REWRITE*

**Sensor Error**

*See section on Error Conditions For Media Access Commands*

**Tape Runout**

*See section on Error Conditions For Media Access Commands*

**Write PSEW**

*The Pseudo Early Warning (PSEW) tape marker has been encountered while writing data or filemarks. This indicates that the tape cartridge is full. The Error Code will be set to E$BTD_PSEW*

**Write EOM**

*The physical end of partition has been encountered while writing data or filemarks. This is a fatal error. Unwritten data and filemarks may be left in the data buffer. See also the Exception Handling Section for the WRITE and WRITE FILEMARKS commands. The Error Code will be set to E$WRT_EOM*

## 7.4. Error Conditions for Media Access Commands

When the Initiator issues one of the Media Access Commands, there are a number of error conditions (in addition to the general error conditions) that may occur. This section summarizes all such error conditions. For command specific error conditions, please see the Exception Handling Section for the actual command.

**Cartridge Error**

*An error has been detected in the Capstan Motor System. Media Access commands will be terminated with CHECK CONDITION status. The Error Code will be set to E$TCM_CSTK.*

**No Tape Edge Found**

*An error has been detected in the Tape Head Servo System. Media Access commands will be terminated with CHECK CONDITION status. The Error Code will be set to E$TCM_NTEF.*

**No Cartridge**

*No cartridge is inserted or the cartridge was removed during command execution. Media Access commands will be terminated with CHECK CONDITION status. The Error Code will be set to E$STE_NCAR (no cartridge) or E$TCM_CRMD (cartridge removed).*

**Not Loaded**

*The cartridge has not been loaded by a LOAD/UNLOAD command or by the AutoLoad function. The Error Code will be set to E$STE_NLOD.*

**Sensor Error**

*An error has been detected in the Tape Hole Sensor System. Media Access commands will be terminated with CHECK CONDITION status. The Error Code will be set to E$TCM_SENS.*

**Tape Runout**

*The tape has run out on one of the cartridge reels. Media Access commands will be terminated with CHECK CONDITION status. The Error Code will be set to E$TCM_TRUN.*

## 7.5. Recovered Errors

Recovered data errors are normally not reported to the Initiator. While writing data on a tape it is quite normal that some data blocks have to be written more than once. This may be due to minor defects that are present in most tape cartridges. Even when reading it may happen once in a while that a block must be re-read. The PER (Post Error Recovery) bit in the Error Recovery Page of the MODE SELECT can be used to turn Recovered Error Reporting on or off.

When the PER bit is set to one the Drive will terminate any command (except REQUEST SENSE) with a CHECK CONDITION status if there has been any re-reads or re-writes since the last command and no other errors has occurred. The Error Code will be set to E$STE_RECV.

NOTE:
The terminated command will be fully executed before the CHECK CONDITION is returned.

As long as the Drive is in unbuffered mode the CHECK CONDITION will only occur on the command that transferred the data that had to be re-read or re-written. The Drive does not report re-reads on read-ahead data.

When the Drive is in buffered mode a re-write might occur when no write command is active (the re-write might occur on data that was written after the last write command terminated with GOOD status). In this case the next command issued to the Drive will be terminated with CHECK CONDITION status after execution.

NOTE:
A REQUEST SENSE command will execute normally and not report CHECK CONDITION even if re-writes has occurred.

## 7.6. Bus Parity Error Handling

The Drive checks the state of the Bus Parity Error Jumper during Power- Up/Reset initialization. Depending on the state of this jumper, bus parity error checking is either globally enabled or disabled.

The Drive supports re-transfer of commands, status, data and messages both as a Target and as an Initiator.

***SLR5 4.0/8.0GB and SLR4 2.5/5.0GB (TDC 4222) only***

Retransfer of data is NOT supported if the data is compressed. See Section 2.7.10.

### *7.6.1. The Drive as a Target*

When bus parity error checking is enabled, the Drive will check for bus parity errors during transfer of Command Descriptor Blocks, Data Out and Message Out. In addition the Drive takes appropriate action when a INITIATOR DETECTED ERROR or MESSAGE PARITY error message is received.

When transferring Command Descriptor Blocks, the Drive will transfer the complete block (or at least 6 bytes) before taking any action on bus parity errors.

When transferring Data Out, the Drive will transfer complete blocks before taking any action on bus parity errors. When a bus parity error has been detected, then the Drive will ensure that the erroneous block is not written to the tape.

#### 7.6.1.1. Errors Detected by the Drive

When a bus parity error has been detected in a SELECTION phase, the selection will be ignored by the Drive.

When a bus parity error has been detected in COMMAND phase, the Drive will go to the to MESSAGE IN phase and transfer a RESTORE POINTERS message. The Drive will then go back to the COMMAND phase and the Command Descriptor Block will be transferred once more from the Initiator.

When a bus parity error has been detected in a DATA OUT phase, the Drive will go to the MESSAGE IN phase and transfer a RESTORE POINTERS message. The Drive will then go back to the DATA OUT phase and the data transferred since the last reconnect (or COMMAND phase) will be transferred once more from the Initiator.

If a parity error is detected during a MESSAGE OUT phase, the Target will consume all the remaining bytes in the message and ask for retransfer of the whole message.

If ATN is deasserted before the expected number of bytes is transferred, the Target will send a REJECT MESSAGE back to the Initiator.
If ATN is still asserted after the expected number of bytes is transferred, the Target goes to the BUS FREE state.

If a parity error is detected during a MESSAGE IN phase, the Initiator signals message parity error. The Target responds by transferring the whole message once more.

#### 7.6.1.2. Errors Detected by the Initiator

The Initiator signals bus parity errors by asserting ATN. The Drive must acknowledge the ATN by going to the MESSAGE OUT Phase. The MESSAGE OUT transferred will be either an INITIATOR DETECTED ERROR message or a MESSAGE PARITY ERROR message.

If the previous phase was a DATA IN phase and the last transferred message was INITIATOR DETECTED ERROR, then the Drive will go to the MESSAGE IN phase and transfer a RESTORE POINTERS message. The Drive will then go back to the DATA IN phase and the data transferred since the last reconnect (or COMMAND phase) will be transferred once more to the Initiator.

If the previous phase was a STATUS phase and the last transferred message was INITIATOR DETECTED ERROR, the Drive will simply retransfer the Status byte once more to the Initiator.

If the previous phase was a MESSAGE IN phase and the last transferred message was MESSAGE PARITY ERROR, then the Drive will go back to the MESSAGE IN phase and the previous Message byte will be transferred once more.

### 7.6.2. The Drive as an Initiator

When bus parity error checking is enabled, the Drive will check for bus parity errors during transfer of Data In, Status and Message In. In addition the Drive takes appropriate action when the Target issues RESTORE DATA POINTER messages and when message retransfer is necessary.

#### 7.6.2.1. Errors Detected by the Drive

When a bus parity error has been detected in a RESELECTION phase, the reselection will be ignored by the Drive.

When a bus parity error has been detected in a DATA IN phase, the Drive will assert the ATN line on the SCSI bus. If the Target goes to the Message Out phase the Drive will transfer an INITIATOR DETECTED ERROR message. The received data is not written to the tape. When the Target goes to the Message In phase and transfers a RESTORE POINTERS message, the Drive will reset its data pointer to the value saved last time the Target issued a SAVE DATA POINTER message (or since the last non-data in phase). In a subsequent new DATA IN phase the Target may then re-transfer the last transferred data once more. Note, however, that the Drive is not able to move its data pointer further back than 65024 bytes from where the RESTORE POINTERS message is received. If the Target lets more than this amount of data be transferred between each SAVE DATA POINTER message, then the Drive can not guarantee that the corrupted data is not written out to the tape.

When a bus parity error has been detected in a STATUS phase, the Drive will assert the ATN line before negating the status phase ACK line. If the Target goes to the Message Out phase the Drive will transfer an INITIATOR DETECTED ERROR message. The Drive will then accept both SAVE DATA POINTER messages and RESTORE POINTERS messages. The Target may the re-enter the STATUS phase and transfer the status byte once more.

When a bus parity error has been detected in a MESSAGE IN phase, the Drive will assert the ATN line before negating the message phase ACK line. If the Target goes to the Message Out phase the Drive will transfer a MESSAGE PARITY ERROR message. The Target may then re-enter the MESSAGE IN phase and transfer the message once more.

#### 7.6.2.2. Errors Detected by the Target

When the Target has detected a parity error in the COMMAND phase it may go to the Message In phase and transfer a RESTORE POINTERS message. The Drive will then reset its command pointer back to the beginning of the last transferred Command Descriptor Block (CDB). When the Target re-enters the COMMAND phase the Drive will let the Target transfer the same CDB once more.

When the Target has detected a parity error in the DATA OUT phase it may go to the Message In phase and transfer a RESTORE POINTERS message. The Drive will then reset its data pointer to the value saved last time the Target issued a SAVE DATA POINTER message (or since the last non-data in phase). When the Target re-enters the DATA OUT phase the Drive will let the Target transfer the same data once more. Note, however, that the Drive is not able to move its data pointer further back than 65024 bytes from where the RESTORE POINTERS message is received. If the Target lets more than this amount of data be transferred between each SAVE DATA POINTER message, then the Drive can not guarantee that the data actually transferred is that same data transferred before the RESTORE POINTERS message was issued.

When the Target has detected a parity error in the MESSAGE OUT phase it may signal re-transfer of the message bytes by continue the MESSAGE OUT phase even if the Drive de-asserted the ATN line. The Drive will then let the Target re-transfer all message bytes transferred when the ATN line was asserted.

## 7.7. Buffer Parity Error Handling

The Drive will check for Buffer Parity Errors whenever data is transferred in or out of the data buffer.

The Drive will check for possible buffer parity errors after every SCSI command. The commands first execute as normal. The buffer parity error check is then performed immediately before status is returned as long as no other errors has occurred during the normal command execution.

Note that when operating in buffered mode or when the Drive processes read-ahead data, a buffer parity error may occur even if no commands are active in the Drive. This means that any later SCSI command may flag a buffer parity error.

When a buffer parity error has been detected, a command is terminated with a CHECK CONDITION status. The Drive error code will be set to E$THI_PARITY. Note that this is true for all commands including the REQUEST SENSE command. This means that when a buffer parity error has occurred, a REQUEST SENSE command may first return CHECK CONDITION status (after having returned the requested sense data), and then when the REQUEST SENSE command is issued again it will return sense data indicating a buffer parity error. When a buffer parity error has been reported the buffer parity error detection logic is reset so that unless another errors occurs, new commands will be processed normally.

## 7.8. Error Priority

The Drive will implement the following error reporting priority:

| | |
|---|---|
| **High Priority:** | Parity Errors<br>Drive Busy<br>Reservation Conflict<br>Unit Attention<br>Deferred Errors<br>Invalid Command Code<br>Reserved Bits/Fields In CDB<br>Unsupported LUN |
| **Low Priority:** | Other Errors |

When reading or verifying variable blocks, the Drive may signal ‘Illegal Length Indication’ if the actual length of a block does not match the requested length. When the actual number of data bytes found is less than the number of bytes requested, the Drive usually signals ‘Illegal Length Indication’ (the Drive error code is set to E$STE_ILLN and the ILI-bit is set in the Sense Data List). The following error conditions have higher priority than ‘Illegal Length Indication’, and the actual error is reported instead of ‘Illegal Length Indication’.

| Name | Drive Error Code |
|---|---|
| Blank Tape (No reference track) | E$TCM_NODATA |
| Cartridge Removed | E$TCM_CRMB |
| Fast Cartridge | E$TCM_CFST |
| Illegal Termination | E$TEM_ILTERM |
| LEOP After PSEW *) | E$TEM_EOREW |
| Logical End Of Partition *) | E$TEM_EOR |
| Operation Timeout | E$TCM_TIME |
| Physical End Of Partition | E$TEM_PEOP |
| Read Retries Exhausted | E$BTD_RTRY |
| Sensor Error | E$TCM_SENS |
| Tape Runout | E$TCM_TRUN |
| Cannot Read, Unknown Format | E$TCM_MEDERR |
| Write Integrity | E$TCM_SAF0 |
| Write Integrity | E$TCM_SAF1 |
| Write Integrity | E$TCM_VLT0 |
| Write Integrity | E$TCM_VLT1 |
| Write Integrity | E$TCM_ERN0 |
| Write Integrity | E$TCM_ERN1 |

NOTE *):
This error will only have priority over ‘Illegal Length Indication’ when it is a real error. This means that if the block preceding the erased tape area is terminated in a normal way, the ‘Illegal Length Indication’ is reported when a READ/VERIFY command has specified a transfer length larger than the block length. Only when a variable length block has been truncated by some kind of media error (as when the previously written data has been erased) will the ‘Logical End Of Partition’ error take priority over ‘Illegal Length Indication’.

# 8. Copy

## 8.1. Command Description

***SLR5 4.0/8.0GB and SLR4 2.5/5.0GB (TDC 4222) only***

The COPY command is not supported by the SLR5 4.0/8.0GB and the SLR4 2.5/5.0GB (TDC 4222). Hence, the Drive will terminate the COPY command with CHECK CONDITION status. The Error Code will be set to E$STE_ICOP.

The COPY command copies data directly between the Drive and another SCSI device. The original Initiator initiates the COPY operation, but the actual data transfer will occur without further Initiator intervention.

When executing the COPY command the Drive takes two roles; managing the copy operation and doing the actual data transfer. The management part reads and checks the Copy Parameter List, takes care of error handling etc. The management part operates as an Initiator for both the other SCSI device and also for the internal data transfer part. The internal data transfer part operates much like a "normal" Drive when executing the READ and WRITE commands.

Typically the COPY command is used to move data to or from some Hard Disk. The Drive can then manage a full Backup or Restore operation.

The Copy Operation starts when the COPY Command and the COPY Parameter List has been transferred to the Drive. Note that the whole Parameter List is transferred in one single transfer. The Drive will then disconnect from the Initiator. This means that the Initiator must support disconnection for the COPY command to work. If this is not so the Drive will abort the COPY command with CHECK CONDITION status (See also Section 8.7.1).

The Drive will process all Segment Descriptors (or until an error has been detected) before re-connecting to the original Initiator.

When a COPY operation leads to reading or writing from beginning of media, the Drive will position the tape and the head as if a normal READ or WRITE operation has started.

If the tape operation is to start from another position than BOM, it is the Initiator's responsibility to position the tape before issuing the COPY command. Any necessary error recovery procedures for any device are also the responsibility of the Initiator (See also Section 8.7.2)

To allow the Drive to work with all kinds of hard disk controllers, both 6 and 10 byte READ and WRITE commands are offered. The decision of which type of READ /WRITE command to be used is made by analyzing the direct device start block address, specified in the Segment Descriptor List. When a start address greater than (2**21-1) is specified, the 10 byte command is used, else the standard 6 byte command it used. It is the responsibility of the Initiator to be sure of that the direct device involved in the COPY command supports the 10 byte READ/WRITE command when the start address does not fit into the specified 21 bit Logical Block Address field in the standard 6 byte READ/WRITE command.

When processing a Segment Descriptor the Drive will issue a sequence of READ/WRITE commands to the direct access device.
Even if the direct access device does support disconnection or not, it is recommended that the Drive should be configured in such a way that it will request only small amounts of data at a time, data which are already in the Drive data buffer or which there is reserved space for in the buffer. It also means that the Drive will have control over the utilization of the SCSI bus (the direct access device is only selected when the Drive is ready for a data transfer). This makes sense because a streaming tape device like the Drive usually has a much lower average data transfer rate than most direct access devices (typically the direct access device is a hard disk). However, since direct device disconnects during the COPY command are supported by the Drive, it is up to the Initiator to configure the SCSI devices involved to obtain the best overall system performance. (See also Section 8.4).

For every Segment Descriptor in the Parameter List the sequence will be like this:

**1)** *The Drive issues a READ CAPACITY command to establish the current direct access device block size (See also Section 8.5).*

**2)** *The Drive issues a 6 or 10 byte READ or WRITE command following the rules described above.*
*The number of blocks requested for transfer is controlled by the COPY THRESHOLD parameter in the MODE SELECT block (the default is normally 8 blocks, see also Section 8.4.2). The Logical Block Address in the first Command Descriptor Block issued to the Direct Access Drive, is copied from the Segment Descriptor. The Logical Block Address for the next command blocks are incremented by the number of blocks transferred.*

**3)** *The data is transferred to/from the direct access device from/to the data buffer in the Drive. From here it goes from/to the tape.*

**4)** *Point 2 and 3 are repeated until the total number of blocks specified in the current Segment Descriptor List has been transferred (this will sometimes mean that the last READ/WRITE command will request less than COPY THRESHOLD blocks).*

**5)** *Points 1, 2, 3 and 4 are repeated for each Segment Descriptor in the Segment Descriptor List.*

## 8.2. Copy Command Descriptor Block

| BYTE | BIT 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 00 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| 01 | Logical Unit Number (LUN) | | | RESERVED | | | | PAD |
| 02 | Length of Parameter List | | | | | | | |
| 03 | | | | | | | | |
| 04 | | | | | | | | |
| 05 | Control Byte | | | | | | | |

*Table: COPY Command Block*

**LUN** Logical Unit. Must be set to zero (only one LUN in the Drive).

**PAD** The pad option (PAD) MUST be set to zero.

**Length of Parameter List** Total length of the parameter block including the Copy Header List and all Segment Descriptors. A zero value means copy of no data. For an active copy the Length must be equal to or larger than 4 (at least the Copy Header List must be transferred). The maximum value is 3076. This corresponds to 256 Segment Descriptors (12 bytes long) and a single Header.

## 8.3. Parameter List

The Parameter List has two parts. First there is a four-byte header block which contains the Copy Function Code. One or more Segment Descriptors are provided after the Header. Up to 256 Segment Descriptors are supported. The Segment Descriptors are identified with ascending numbers, beginning with zero.

### 8.3.1. Header List

The Copy Header List controls the direction of the copy (BACKUP or RESTORE).

| BYTE | BIT 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 00 | Copy Function | | | | | RESERVED | | |
| 01 | RESERVED | | | | | | | |
| 02 | RESERVED | | | | | | | |
| 03 | RESERVED | | | | | | | |

*Table: COPY Header List*

The following **Copy Functions** are supported by the Drive:

- 00: Direct To Sequential Access Device (BACKUP)
- 01: Sequential To Direct Access Device (RESTORE)

### 8.3.2. Segment Descriptor List

Each Segment Descriptor describes a single copy operation. Both the Direct Access Block Address and the Number Of Blocks are specified. The Source and Destination ID (and LUN) are also specified in each Segment Descriptor. This means that a single COPY command can transfer data to/from several Direct Access Devices. Note, however, that either the Source (RESTORE) or Destination (BACKUP) device always must be the Drive executing the COPY command.

| BYTE | BIT 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 00 | Source ID | | | R | CAT | Source LUN | | |
| 01 | Destination ID | | | RESERVED | | Destination LUN | | |
| 02 | Sequential Access Device Block Size = 0200 Blocks | | | | | | | |
| 03 | | | | | | | | |
| 04 | Direct Access Device Number of Blocks | | | | | | | |
| 05 | | | | | | | | |
| 06 | | | | | | | | |
| 07 | | | | | | | | |
| 08 | Direct Access Device Logical Block Address | | | | | | | |
| 09 | | | | | | | | |
| 0A | | | | | | | | |
| 0B | | | | | | | | |

*Table: COPY Segment Descriptor*

**Source ID** The ID number (0..7) of the Source Device

**Destination ID** The ID number (0..7) of the Destination Device

**Source LUN** The Logical Unit Number (0..7) of the Source Device

**Destination LUN** The Logical Unit Number (0..7) of the Destination Device

**CAT** A catenate (CAT) bit of one indicates that the Drive will catenate the last source block of a segment with the first source block of the next segment if the last source block does not end exactly at the end of the destination block. The CAT bit can not be set in the very last segment descriptor.

**Sequential Access Device Block Size** The block size of the Sequential Access Device. The only supported block size is 512 bytes (200h). See also Section 8.5.

**Direct Access Device Number Of Blocks** This is the number of data blocks to be transferred for this Segment. Depending on the current Direct Access Device Block Size this will correspond to various number of the Drive's data blocks. A Block Size of 1024 bytes will lead to a 1:2 relationship between number of Direct Access Device block counts and the Drive's block counts etc. A zero value means copy of no data for this Segment.

**Random Access Device Logical Block Address**

This is the address of the first block on the Random Access Device to be copied.

NOTE:
The device must support the 10 byte READ/WRITE command if the Logical Block Address is greater than (2**21-1).

## 8.4. Selectable Options

The MODE SELECT command can be used to control some parameters for the COPY command. On power up or reset all parameters will have default values (See the MODE SELECT Section for details). Use the MODE SELECT command to change these values. The Save Mode Parameters (SMP) bit in the MODE SELECT command block can be used to make the mode settings permanent. This means that the current settings (after MODE SELECT) will be used as defaults on the next power up or reset.

### *8.4.1. Buffered Mode*

The COPY command supports Buffered Mode when performing a copy from a direct access device to the Drive (BACKUP).

The Buffered Mode is always active when moving from one Segment Descriptor to the next. This means that the Drive will start processing a new Segment Descriptor as soon as the data from the previous Segment Descriptor has been transferred into the data buffer.

When the entire COPY operation has been completed (data from the last Segment Descriptor has been transferred into the buffer), the Buffered Mode operation is controlled by the Buffered Mode (BM) bit in byte 02 of the MODE SELECT parameter block - exactly in the same way as for WRITE operations. When the BM bit is not set, the Drive will not reconnect and send status before all data in the buffer has been written to the tape. If the BM bit is set, the Drive re-connects and reports GOOD STATUS as soon as the last data byte of the last Segment Descriptor is transferred into the buffer.

### *8.4.2. Copy Threshold*

The COPY THRESHOLD controls the number of blocks requested for transfer to/from the direct access device in each READ/WRITE command issued from the Drive during Copy. The minimum value is 1. The maximum is dependent on the block size of the direct access device. No more than (64K-1) bytes should be transferred for a single READ/WRITE command. This corresponds to a maximum COPY THRESHOLD of 254 for 256 byte blocks, 127 for 512 byte block, 63 for 1024 byte blocks, 31 for 2048 byte blocks and 15 for 4096 byte blocks. The COPY THRESHOLD is found in byte 05 in the Miscellaneous Parameter Page List of the MODE SELECT command.

### 8.4.3. Copy Sense Allocation

The COPY SENSE ALLOCATION controls the Sense Data handling of both the Target direct access device and the data transfer part of the Drive in case of an error. The COPY SENSE ALLOCATION is located in byte 04 in the Miscellaneous Parameter Page List of the MODE SELECT command.

## 8.5. Block Sizes

The normal block size used in the SCSI-bus interface of the Drive is 512 bytes. With some restrictions the Drive is, however, able to copy data to/from direct access devices with block sizes different from 512.
The block size of the Drive will, however, always be 512 bytes.
The following five Direct Access Device block sizes are supported:

| | | | | | |
|---|---|---|---|---|---|
| 1: | **256** | **bytes** | 4: | **2048** | **bytes** |
| 2: | **512** | **bytes** | 5: | **4096** | **bytes** |
| 3: | **1024** | **bytes** | | | |

The fact that the block size on the tape is 512 bytes will not in any way limit the use of the COPY command. Possible differences between the Sequential Access and the Direct Access Device block sizes are handled internally in the Drive. This automatic mapping between the tape and the Direct Access block size is invisible to the user - both in the back-up and the restore modes.

***Example:*** *Direct Access Device block size: 1024 bytes*
*No. of blocks to transfer: 7 blocks*
*This gives 7 KBytes and 14 logical blocks of 512 bytes when stored on the tape. Restoring will follow the reverse sequence.*

### 8.5.1. Checking the Block Size

The Drive uses a double method to determine the actual block size of the direct access device. Before processing a new Segment Descriptor, the Drive first issues a READ CAPACITY (25 Hex) command. If the direct access device supports this command it will respond by sending 8 bytes of parameter data back to the Drive. The last 4 bytes holds the block size in bytes. This block size is used for all data transfers for the current segment.

If, however, the direct access device does not support the READ CAPACITY command, it will terminate this command with a CHECK CONDITION status. The Drive will then issue the READ CAPACITY command once more in case the first CHECK CONDITION was due to a pending UNIT ATTENTION condition in the direct access device.

If this second attempt also fails, the Drive will use its other method for finding the block size. First any pending sense information are cleared by issuing a REQUEST SENSE COMMAND. 4 bytes of sense data is requested. Then a READ command with the Transfer Length set to 1 (read 1 block) is sent to the direct access device (the logical address used is 0). The Drive will transfer one data block and count the number of bytes transferred.

Regardless of the method used, the Drive will check the block size found to see if it is legal (256, 512, 1024, 2048 or 4096).

### 8.5.2. Inexact Segments

It should be noted that even if the normal block size used on the SCSI-bus interface of the Drive is 512 bytes, the real (internal) block size is 1024 bytes when the current tape format is QIC-525/1000/2GB/96-34. When reading old tapes with QIC-24, QIC-120 or QIC-150 format, the internal block size is, however, 512 bytes. The Drive will always automatically take care of the mapping between the 512 byte blocks used on the SCSI-bus and the 1024 bytes blocks used for QIC-525/1000/2GB/96-34 on the tape.

When copying to or from direct access devices with a block size different from the Drive simulated block size of 512 bytes, it is, however, possible for the total amount of data requested for copy in a segment does not fit into an integral number of 512 bytes blocks. This is called an Inexact Segment situation. According to the ANSI SCSI-2 specification there are two ways of handling this; padding or catenating.

The Drive does not support the Pad option for the COPY command. The Drive does, however, support the catenate (CAT) option. The CAT option can be enabled or disabled on a segment to segment basis.

## 8.6. Commands Used as an Initiator

### *8.6.1. Read (6-byte Command)*

| BYTE | BIT 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 00 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 01 | Logical Unit Number (LUN) | | | Logical Block Address | | | | |
| 02<br>03 | Logical Block Address | | | | | | | |
| 04 | Number of Blocks | | | | | | | |
| 05 | 0 | 0 | 0 | | | | 0 | 0 |

*Table: READ Command Block (Group 0) used during COPY*

This command is used to transfer data from the Direct access device to the Drive (Backup).

**LUN**

The LUN field is copied from the corresponding LUN field in the Segment Descriptor List.

**Logical Block Address**

For the very first READ command the value in the Logical Block Address field is also copied from the Segment Descriptor List. For the next READ commands the value of the Logical Block Address field is incremented by the value of the Number Of Blocks field in the previous READ command.

**Number Of Blocks**

The Number Of Blocks field is copied from the COPY THRESHOLD byte (see Section 8.4.2).

### 8.6.2. Read (10-byte Command)

| BYTE | BIT 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 00 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| 01 | Logical Unit Number (LUN) | | | 0 | | | | 0 |
| 02 | Logical Block Address | | | | | | | |
| 03 | | | | | | | | |
| 04 | | | | | | | | |
| 05 | | | | | | | | |
| 06 | 0 | | | | | | | |
| 07 | Number of Blocks | | | | | | | |
| 08 | | | | | | | | |
| 09 | 0 | 0 | 0 | | | | 0 | 0 |

*Table: READ Command Block (Group 1) used during COPY*

This command is used to transfer data from the Direct access device to the Drive (Backup).

**LUN** The LUN field is copied from the corresponding LUN field in the Segment Descriptor List.

**Logical Block Address** For the very first READ command the value in the Logical Block Address field is also copied from the Segment Descriptor List. For the next READ commands the value of the Logical Block Address field is incremented by the value of the Number Of Blocks field in the previous READ command.

**Number Of Blocks** The Number Of Blocks field is copied from the COPY THRESHOLD byte (see Section 8.4.2).

### 8.6.3. Read Capacity

| BYTE | BIT 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 00 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 |
| 01 | Logical Unit Number (LUN) | | | 0 | | | | |
| 02 | 0 | | | | | | | |
| 03 | 0 | | | | | | | |
| 04 | 0 | | | | | | | |
| 05 | 0 | | | | | | | |
| 06 | 0 | | | | | | | |
| 07 | 0 | | | | | | | |
| 08 | 0 | | | | | | | |
| 09 | 0 | 0 | 0 | | | | 0 | 0 |

*Table: READ CAPACITY Command Block used during COPY*

This command is used to get the block size of the Direct access device.

**LUN** The LUN field is copied from the corresponding LUN field in the Segment Descriptor List.

### 8.6.4. Request Sense

| BYTE | BIT 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 00 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 01 | Logical Unit Number (LUN) | | | 0 | | | | |
| 02 | 0 | | | | | | | |
| 03 | 0 | | | | | | | |
| 04 | Target Sense Length | | | | | | | |
| 05 | 0 | 0 | 0 | | | | 0 | 0 |

*Table: REQUEST SENSE Command Block used during COPY*

This command is used when the Direct access device has terminated a command with a CHECK CONDITION status.

**LUN** The LUN field is copied from the corresponding LUN field in the Segment Descriptor List.

**Target Sense Length** The Target Sense Length field is controlled by the SENSE ALLOCATION byte (see Section 8.4.3).

### 8.6.5. Write (6-byte Command)

| BYTE | BIT 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 00 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| 01 | Logical Unit Number (LUN) | | | Logical Block Address | | | | |
| 02<br>03 | Logical Block Address | | | | | | | |
| 04 | Number of Blocks | | | | | | | |
| 05 | 0 | 0 | 0 | | | | 0 | 0 |

*Table: WRITE Command Block (Group 0) used during COPY*

This command is used to transfer data from the Drive to the Direct access device (Restore).

**LUN**

The LUN field is copied from the corresponding LUN field in the Segment Descriptor List.

**Logical Block Address**

For the very first WRITE command the value in the Logical Block Address field is also copied from the Segment Descriptor List. For the next WRITE commands the value the Logical Block Address field is incremented by the value of the Number Of Blocks field in the previous WRITE command.

**Number Of Blocks**

The Number Of Blocks field is copied from the COPY THRESHOLD byte (see Section 8.4.2).

### 8.6.6. Write (10-byte Command)

| BYTE | BIT 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 00 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| 01 | Logical Unit Number (LUN) | | | 0 | | | | 0 |
| 02 | Logical Block Address | | | | | | | |
| 03 | | | | | | | | |
| 04 | | | | | | | | |
| 05 | | | | | | | | |
| 06 | 0 | | | | | | | |
| 07 | Number of Blocks | | | | | | | |
| 08 | | | | | | | | |
| 09 | 0 | 0 | 0 | | | | 0 | 0 |

Table: WRITE Command Block (Group 1) used during COPY

This command is used to transfer data from the Drive to the Direct access device (Restore).

**LUN**

The LUN field is copied from the corresponding LUN field in the Segment Descriptor List.

**Logical Block Address**

For the very first WRITE command the value in the Logical Block Address field is also copied from the Segment Descriptor List. For the next WRITE commands the value of the Logical Block Address field is incremented by the value of the Number Of Blocks field in the previous WRITE command.

**Number Of Blocks**

The Number Of Blocks field is copied from the COPY THRESHOLD byte (see Section 8.4.2).

## 8.7. Exception Handling

See sections on Error Conditions For All Commands, Deferred Errors, Error Conditions For Media Access Commands and Buffer Parity Errors.

There are two classes of error conditions which may occur during a COPY operation.

The first class is the Management Errors. These are detected by the management part of the Drive. The management part are responsible for checking the COPY command and parameter blocks. It is the management part that interprets and splits the Segment Descriptors into multiple READ/WRITE commands.

The second class is the Data Transfer (or Target) errors. These are detected by the data transfer part of the Drive. The data transfer part takes care of the actual tape read and write process.
This class includes error conditions like Filemark Detected, Logical End Of Partition, Illegal Length etc.

### 8.7.1. Management Errors

When a Management Error has been detected, the Drive will react as follows:

**1)** *Terminate the COPY command with CHECK CONDITION status.*

**2)** *Sets VADD bit. Sets the Sense Key in the Sense Data List to the code that describes the error condition.*

**3)** *The Segment Number in the Sense Data List (byte 01) is set to the number of the segment where the error occurred.*

**4)** *The Information Bytes are updated with the difference between the requested and the actually processed blocks/bytes for the current segment.*

Possible Error Codes are shown in the table below:

| Description | Drive Error Code |
|---|---|
| Illegal Copy Function | E$SIE_CHDF |
| Bad Header | E$SIE_CHDI |
| No Disconnect Allowed | E$SIE_CILC |
| Inexact Segment: Odd Number | E$SIE_CODD |
| Inexact Segment: Tape Residual | E$SIE_CRES |
| Too Large Logical Address | E$SIE_CSGA |
| Bad ID or LUN | E$SIE_CSGI |
| Partial Descriptor Received | E$SIE_CSGP |
| Parity Error in Parameter or Data | E$SIP_CIDT |
| Target Illegal Block Size | E$SIP_CILB |
| Target Selection Time-out | E$SIP_CISE |
| Target Phase Sequence Error | E$SIP_CISQ |
| Bad Command Block | E$STE_IFIC |
| Too Short/Long Header | E$STE_PLEN |

*Table: COPY Management Errors*

**Bad Command Block**

*The COPY command descriptor had illegal values in some fields. (See also the Section on GENERAL EXCEPTION HANDLING).*

**No Disconnect Allowed**

*The original Initiator transferred the COPY command to the Drive without granting disconnection (ATN not set or Disconnect Grant bit in IDENTIFY message not set).*

**Illegal Copy Function**

*The Copy Function specified in the Header was not 00 (backup) or 01 (restore).*

**Bad Header**

*One or more of the reserved fields in the Header was not set to zero.*

**Too Short /Long Header**

*The received Header has less or more than the normal 4 bytes.*

**Too Large Logical Address**

*The Direct Access Device Logical Block Address field specified an address larger than (2**21 - 1) and the Direct Access Device did not support the 10 byte READ/WRITE command. The largest address that will fit into a 6 byte READ or WRITE command descriptor block is (2**21 - 1).*

**Bad ID or LUN**

*The LUN was not zero for the Drive or the Source ID was not the ID of the Drive (Copy Function = Restore) or the Destination ID was not the ID of the Drive (Copy Function = Backup).*

**Partial Descriptor Received**

*The last Segment Descriptor in the Segment Descriptor List had less than 12 bytes.*

**Parity Error In Parameter or Data**

*The management part of the Drive has detected a bus parity error on either data or parameters transferred from the random access device.*

**Target Selection Time-out**

*The random access device did not respond to a selection within a 250 ms time-out period.*

**Target Phase Sequence Error**

*The random access device did not transfer any Status byte or any Message byte or entered one of the two illegal bus phases.*

**Target Illegal Block Size**

*The block size found on the random access device was not in the set of legal sizes (256, 512, 1024, 2048 or 4096 bytes).*

**Inexact Segment Odd Number**

*See Section 8.7.1.1.*

**Inexact Segment Tape Residual**

*See Section 8.7.1.1. "**Insufficient Number of Tape Blocks**".*

#### 8.7.1.1. Inexact Segment Errors

**CAT Option Enabled**

As long as the CAT option is active there will never be any inexact segment errors when data is transferred to the tape (copy function code = 0). The Drive will catenate the last source block of a segment with the first source block of the next segment if the last source block does not end exactly at the end of the destination block. Note that the CAT option is not allowed on the last segment in the COPY command. It is required that the sum total of all segments in a COPY command fits into an integral number of 512 byte blocks. If this is not the case the Drive will flag an Inexact Segment Error for the last segment by terminating the COPY operation with CHECK CONDITION status. This can happen during both BACKUP and RESTORE. Enough data only to complete the last source block will be transferred from the segment in error. The Error Code will be set to E$SIE_CODD.

**CAT Option Disabled**

When the CAT option is not enabled the total amount of data in a segment must always fit into an integral number of 512 bytes blocks. If this is not the case, the Drive will flag an inexact segment error by terminating the COPY operation immediately with CHECK CONDITION status. This can happen during both BACKUP and RESTORE. No data will be transferred for the segment in error. The Error Code will be set to E$SIE_CODD.

**Insufficient Number of Tape Blocks**

When the block size is larger than 512 an inexact segment error will be flagged if the residual number of blocks on the tape is to few to fill a complete direct access block. This can happen during RESTORE only if the data transfer part of the Drive encounters an unexpected Filemark, End Of Recorded Area (BLANK CHECK) or a Bad Block. All data up to the last residual block(s) will be transferred to/from the direct access device. The last incomplete block is not transferred. CHECK CONDITION status is sent to the original Initiator. The Error Code will be set to E$SIE_CRES. The tape position will be immediately after the last transferred block. This means that there will be one or more (maximum 7) blocks left on the tape. It is up to the Initiator to recover these blocks. The real cause of error can then be discovered. Note that it is up to the original Initiator to generate tapes (to end a tape write operation or write Filemarks) so that an inexact segment error like this does not occur. Normally this is no problem as long as the direct access device used during RESTORE has the same block size as the one used during BACKUP for a particular tape.

### *8.7.2. Data Transfer Errors*

When a Data Transfer Error has been detected, the Drive reacts as follows:

**1)** *Request/generate Target Sense Data.*

**2)** *Terminate the COPY command with CHECK CONDITION status.*

**3)** *Set the VADD bit. The Sense Key is set to COPY ABORTED. The Error Code is set to a code describing the error condition.*

**4)** *The Segment Number is updated.*

**5)** *The Information Bytes are updated with the difference between the requested number of block and the actual number of blocks transferred for the current segment.*

**6)** *The Additional Sense Length is updated to indicate the number of additional sense data available. As a minimum a Target status Byte will be available. The Additional Sense Length is then set to 18 (17 normal additional sense byte plus the Target Status Byte). If Target Sense Data is available, the Additional Sense Length indicates the number of bytes available like this:*

*Target Sense Bytes = (Additional Sense Length) - 18.*

**7)** *The Source Sense Pointer is updated to point to the source Target Status Byte. This pointer is relative to the first byte of the Sense Data List. Possible source Target Sense Data follows immediately after the Target Status Byte. A zero value Source Sense Pointer indicates that no source Target Status or Sense are available for the source logical unit.*

**8)** *The Destination Sense Pointer is updated to point to the destination Target Status Byte. This pointer is relative to the first byte of the Sense Data List. Possible destination Target Sense Data follows immediately after the Target Status Byte. A zero value Destination Sense Pointer indicates that no destination Target Status or Sense are available for the destination logical unit.*

Possible Error Codes are shown in the table below:

| Description | Drive Error Code |
|---|---|
| Internal CHECK CONDITION | E$SIE_CPDT |
| Target Status Not GOOD STATUS | E$SIP_CIBS |
| Target Status = CHECK CONDITION | E$SIP_CICH |

*Table: COPY Data Transfer Errors*

**Internal CHECK CONDITION**

*The Data Transfer part of the Drive has signalled CHECK CONDITION. Some error like Filemark, End Of Partition etc. has been detected. The original Initiator should see the Source/Destination Sense Data for further information.*

**Target Status Not GOOD STATUS**

*The random access device terminated a command with a Status byte different from GOOD STATUS and CHECK CONDITION. The original Initiator should see the Source/Destination Sense Data for further information.*
*If the random access device terminates a command with BUSY or RESERVATION CONFLICT status, the Drive will try to retransfer the command up to 255 times as long as BUSY or RESERVATION CONFLICT status is received. This is done to prevent a termination of the COPY command if the random access device is accessed by an other initiator during the COPY. The BUSY or RESERVATION CONFLICT status is reported only after 255 non successful (BUSY or RESERVATION CONFLICT status) attempts to perform a command on the random* access device.

**Target Status = CHECK CONDITION**

*The random access device terminated a command with CHECK CONDITION status. The original Initiator should see the Source/Destination Sense Data for further information. CHECK CONDITION will not be reported on a Unit Attention Condition. When this condition occurs, target sense is read by the Drive, and the command that caused this condition is retransferred.*

The random access device notifies the Drive of an error by terminating the current READ or WRITE command with a Status Byte different from GOOD STATUS. If the Status Byte received is a CHECK CONDITION status, the action taken by the Drive depends on the current AUTOSENSE MODE (See also Section 8.4). Normally the AUTOSENSE MODE is on. The Drive will then issue a standard 6 byte REQUEST SENSE command to the direct access device. The number of bytes requested for transfer is controlled by the COPY SENSE ALLOCATION parameter in the MODE SELECT parameter block (See Section 8.4). The default is normally 4 bytes. The Sense Data is then transferred into a buffer in the Drive.

If the last Status Byte from the direct access device is not CHECK CONDITION, the Drive will not request any sense data.

The whole COPY operation is then aborted (even if there are more Segment Descriptors). The management part of the Drive sends CHECK CONDITION status to the original Initiator. The Initiator should then Request Sense Data from the Drive. The Sense Key will read COPY ABORTED. The Sense Data List will have an non-modified copy of the direct access device Status Byte and Sense Data (if available).

The action taken by the Drive in case of an error detected in the Drive itself is very similar. The data transfer part (target part) of the Drive generates a Status Byte and Sense Data internally, and the COPY operation is aborted. The management part of the Drive sends CHECK CONDITION status to the original Initiator. The Initiator should then Request Sense Data from the Drive. The Sense Key will read COPY ABORTED. The Sense Data List will have a copy of the internally generated Status Byte and Sense Data. The Status Byte will always be CHECK CONDITION.

## 8.8. Internally Generated Sense Bytes

When an error is detected internally in the Drive (Target part) during a Copy command, the Source/Destination Sense data of the Request Sense Parameter List will contain information reflecting this error.

The layout of the Source/Destination Sense Bytes will be as follows (see also Chapter 24. Request Sense):

| BYTE | BIT 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 00 | VADD | Error Code | | | | | | |
| 01 | Segment Number | | | | | | | |
| 02 | FMK | EOM | ILI | R | Sense Key | | | |
| 03 | Information Bytes | | | | | | | |
| 04 | | | | | | | | |
| 05 | | | | | | | | |
| 06 | | | | | | | | |
| 07 | Additional Sense Length | | | | | | | |
| 08 | Source Sense Pointer | | | | | | | |
| 09 | Destination Sense Pointer | | | | | | | |
| 10 | RESERVED | | | | | | | |
| 11 | | | | | | | | |
| 12 | Additional Sense Code | | | | | | | |
| 13 | Additional Sense Code Qualifier | | | | | | | |
| 14 | FOR INTERNAL USE | | | | | | | |

*Table: Source/Destination Sense Bytes*

The Source/Destination Status Byte will, if the error is detected in the Target part of the Drive, always have the value 02 (CHECK CONDIT-ION).

The Source/Destination Status Byte is byte 30 in the Request Sense Parameter List.

The first Source/Destination Sense Byte (byte 0 in the described list) is location 31 in the Request Sense Parameter List.

Some of the bytes in the list will have fixed values:

| | | |
|---|---|---|
| **Byte 00** | **:** | **70 (hex)** |
| **Byte 01** | **:** | **00** |
| **Byte 03** | **:** | **00** |
| **Byte 04** | **:** | **00** |
| **Byte 05** | **:** | **00** |
| **Byte 06** | **:** | **00** |
| **Byte 07** | **:** | **07** |
| **Byte 08** | **:** | **00** |
| **Byte 09** | **:** | **00** |
| **Byte 10** | **:** | **00** |
| **Byte 11** | **:** | **00** |

*This Page Intentionally Left Blank*

# 9. Erase

## 9.1. Command Description

The ERASE command causes the Drive to erase the entire tape. When finished the tape will be positioned at BOT.
The Drive will disconnect during execution of the ERASE command if disconnection is allowed.

## 9.2. Command Descriptor Block

| BYTE | BIT 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 00 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 |
| 01 | Logical Unit Number (LUN) | | | RESERVED | | | IMM | LONG |
| 02 | RESERVED | | | | | | | |
| 03 | RESERVED | | | | | | | |
| 04 | RESERVED | | | | | | | |
| 05 | Control Byte | | | | | | | |

*Table: ERASE Command Block*

**IMM** An Immediate bit (IMM) of zero indicates that the Drive will not return status until the erase operation has completed. An IMM bit of one indicates that the Drive will return status as soon as the execution of all previous commands have been completed and the Command Descriptor Block of the ERASE command has been validated. If CHECK CONDITION status is returned for the ERASE command with an IMM bit of one, the erase operation is not performed.

**LONG** This bit MUST be set to one. It indicates that the whole tape will be erased.

## 9.3. Exception Handling

See sections on Error Conditions For All Commands, Deferred Errors and Error Conditions For Media Access Commands.

If the IMM and Link bits are both set to one, the Drive will terminate the ERASE command with CHECK CONDITION status. The Error Code will be set to E$STE_IFIC.

The Drive returns CHECK CONDITION status if the Long bit is not set. The Error Code will be set to E$STE_IFIC.

The ERASE command will not allow erasure of DC300XLP type cartridges. If a DC300XLP type cartridge is present when an ERASE command is received, the ERASE command is immediately terminated with CHECK CONDITION status.
However, if the Immediate-bit (IMM) is set, the CHECK CONDITION status is flagged with the following command.
The Error Code will be set to E$BTD_CFMT.

# 10. Inquiry

## 10.1. Command Description

The INQUIRY command requests that information regarding parameters of the Drive to be sent to the Initiator.

The Parameter List will be transferred during the DATA-IN phase of the command.

The INQUIRY command will execute even if the Initiator specifies an unsupported LUN (LUN field in the Command Descriptor Block or IDENTIFY message is not set to zero).

The INQUIRY command will execute normally even if a reservation conflict exists.

The INQUIRY command will execute even if a Unit Attention condition is pending.

The INQUIRY command will not check for Deferred Errors.

The Drive will never disconnect for this command.

## 10.2. Command Descriptor Block

| BYTE | BIT 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 00 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| 01 | Logical Unit Number (LUN) | | | RESERVED | | | | EVPD |
| 02 | Page Code | | | | | | | |
| 03 | RESERVED | | | | | | | |
| 04 | Allocation Length | | | | | | | |
| 05 | Control Byte | | | | | | | |

*Table: INQUIRY Command Block*

**EVPD**

An Enable Vital Product Data bit of one specifies that the Drive will return the Vital Product Data (VPD) specified by the Page Code field. An EVPD bit of zero specifies that the Drive will return the standard INQUIRY data.

**Page Code**

This field specifies which page of VPD information the Drive will return. This field MUST be set to zero if the EVPD bit is not set to one. Legal values are:

| | | |
|---|---|---|
| **00h** | : | Summary of supported pages |
| **80h** | : | Unit Serial Number Page |
| **81h** | : | Implemented Operating Definition Page |
| **82h** | : | ASCII Implemented Operating Definition Page |
| **C0h** | : | Hardware Revision Levels |
| **C1h** | : | ROM Software Code Revision Level |
| **C2h** | : | Drive Manufacturing Date |
| **C3h** | : | ROM Software Code Creation Date |
| **C4h** | : | Drive Adjustment Date |

**Allocation Length**

This field specifies the maximum number of bytes that the Initiator has allocated for returned INQUIRY data. An Allocation Length of zero indicates that no INQUIRY data will be sent. The Drive terminates the DATA-IN phase when Allocation Length bytes have been transferred or when all available INQUIRY data have been transferred, whichever is less.

## 10.3. Parameter Lists

### 10.3.1. Standard Inquiry Data

| BYTE | BIT 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 00 | Qualifier | | | Peripheral Device Type | | | | |
| 01 | RMB | Device Type Qualifier | | | | | | |
| 02 | ISO | | ECMA | | | | ANSI | |
| 03 | AENC | TIOP | RESERVED | | Response Data Format | | | |
| 04 | Additional Length | | | | | | | |
| 05 | RESERVED | | | | | | | |
| 06 | RESERVED | | | | | | | |
| 07 | RelA | WB32 | WB16 | Sync | Link | Cach | CQue | SftR |
| 08 | Vendor ID | | | "T" | | | | |
| 09 | | | | "A" | | | | |
| 10 | | | | "N" | | | | |
| 11 | | | | "D" | | | | |
| 12 | | | | "B" | | | | |
| 13 | | | | "E" | | | | |
| 14 | | | | "R" | | | | |
| 15 | | | | "G" | | | | |
| 16 | Product ID | " " | " " | " " | " " | " " | " " | |
| 17 | | "T" | "T" | "T" | "T" | "S" | "S" | |
| 18 | | "D" | "D" | "D" | "D" | "L" | "L" | |
| 19 | | "C" | "C" | "C" | "C" | "R" | "R" | |
| 20 | | " " | " " | " " | " " | "5" | "5" | |
| 21 | | "3" | "4" | "4" | "4" | " " | " " | |
| 22 | | "8" | "1" | "2" | "2" | "4" | "4" | |
| 23 | | "2" (or) | "0" (or) | "0" (or) | "2" (or) | "/" (or) | "G" | |
| 24 | | "0" | "0" | "0" | "2" | "8" | "B" | |
| 25 | | " " | " " | " " | " " | "G" | " " | |
| 26 | | " " | " " | " " | " " | "B" | " " | |
| 27 | | " " | " " | " " | " " | " " | " " | |
| 28 | | " " | " " | " " | " " | " " | " " | |
| 29 | | " " | " " | " " | " " | " " | " " | |
| 30 | | " " | " " | " " | " " | " " | " " | |
| 31 | | " " | " " | " " | " " | " " | " " | |
| 32 | PROM ID-character | | | | | | | |
| 33<br>34 | PROM Revision Code | | | | | | | |
| 35 | PROM Type | | | | | | | |
| 36<br>37 | Option Level | | | | | | | |

Table: INQUIRY Parameter List (to be continued...)

| BYTE | BIT 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 38 | Software ID | | | "C" | | | | |
| 39 | | | | "R" | | | | |
| 40 | | | | "E" | | | | |
| 41 | | | | "A" | | | | |
| 42 | | | | "T" | | | | |
| 43 | | | | "E" | | | | |
| 44 | | | | "D" | | | | |
| 45 | Month | | | | | | | |
| 46 | | | | | | | | |
| 47 | Day | | | | | | | |
| 48 | | | | | | | | |
| 49 | Year | | | | | | | |
| 50 | | | | | | | | |

*Table: INQUIRY Parameter List*

**Qualifier**
This field will normally be set to zero. However, when an Unsupported LUN condition exists (LUN field in the Command Descriptor Block or IDENTIFY message is not set to zero), this field will be set to 3.

**Peripheral Device Type**
This field is normally set to 1. When an Unsupported LUN condition exists (LUN field in the Command Descriptor Block or IDENTIFY message is not set to zero), this field will , however, be set to 1Fh.

**RMB**
Removable Medium. This field is always set to 1.

**Device Type Qualifier**
This field is always set to zero.

**ISO**
ISO version is always set to zero.

**ECMA**
ECMA version is always set to zero.

**ANSI**
ANSI version is always set to 2.

**AENC**
This bit will be set to zero.

**TIOP**
This bit will be set to zero to indicate that the Drive does not support the TERMINATE I/O PROCESS message.

**Response Data Format**
This field will be set to 2 (SCSI-2 Standard Format).

**Additional Length**
This field specifies the number of additional INQUIRY parameter bytes. This field is always set to 46 (2Eh).

**RelA**
This bit will be set to zero to indicate that the Drive does not support relative addressing.

**WB32**
This bit will be set to zero to indicate that the Drive does not support 32 bit data transfer.

**WB16**
This bit will be set to zero to indicate that the Drive does not support 16 bit data transfer.

**Sync**
This bit will be set to one to indicate that the Drive supports synchronous data transfer.

**Link**
This bit will be set to one to indicate that the Drive supports linked commands.

**Cach**
This bit will be set to zero to indicate that the Drive does not support caching.

**CQue**
This bit will be set to zero to indicate that the Drive does not support command queuing.

**SftR**
This bit will be set to zero to indicate that the Drive supports the "hard" reset option.

**Vendor ID**
A string of ASCII-characters will be returned.

**Product ID**

These bytes hold one of the ASCII strings: "TDC 3820", "TDC 4100", "TDC 4200", "TDC 4222", "SLR5 4GB" or "SLR5 4/8GB"

**PROM ID-Character**

A single ASCII character designating the SCSI bus protocol version:

"=" : Tandberg Data Standard SCSI-2.

Other characters in this position mean special versions deviating from the SCSI-2 standard.

**PROM Revision Code**

Two ASCII characters showing the PROM revision number (in decimal) starting from "00". This number is incremented by "1" for every major new PROM revision.

**PROM Type**

A single ASCII character showing whether this PROM is a released or experimental version:

"Y" : Experimental Version.

":" : Released Version.

**Option Level**

Two ASCII characters showing the PROM option level (in decimal) starting from "00". It is incremented by one for every new option level.

**Software ID**

These bytes hold this ASCII string: "CREATED".

**Month, Day, Year**

These bytes hold an ASCII string representing the creation date for the current drive software on the form "MM.DD.YY".

## 10.3.2. Vital Product Data

### 10.3.2.1. Summary of Supported Pages

| BYTE | BIT 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 00 | Qualifier | | | Peripheral Device Type | | | | |
| 01 | Page Code = 00h | | | | | | | |
| 02 | RESERVED | | | | | | | |
| 03 | Page Length = 09h | | | | | | | |
| 04 | Supported Page = 00h | | | | | | | |
| 05 | Supported Page = 80h | | | | | | | |
| 06 | Supported Page = 81h | | | | | | | |
| 07 | Supported Page = 82h | | | | | | | |
| 08 | Supported Page = C0h | | | | | | | |
| 09 | Supported Page = C1h | | | | | | | |
| 10 | Supported Page = C2h | | | | | | | |
| 11 | Supported Page = C3h | | | | | | | |
| 12 | Supported Page = C4h | | | | | | | |

*Table: Summary Of Supported VPD Pages*

**Qualifier** This field will normally be set to zero. When an Unsupported LUN condition exists (LUN field in the Command Descriptor Block or IDENTIFY message is not set to zero), this field will, however, be set to 3.

**Peripheral Device Type** This field is normally set to 1. When an Unsupported LUN condition exists (LUN field in the Command Descriptor Block or IDENTIFY message is not set to zero), this field will, however, be set to 1Fh.

**Page Code** This field will be set to the value of the Page Code field in the Command Descriptor Block.

**Page Length** This field will specify the length in bytes of the parameters that follow the Page Length field.

**Supported Page** This is a list of the pages supported by the Drive.

### 10.3.2.2. Unit Serial Number

| BYTE | BIT 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 00 | Qualifier | | | Peripheral Device Type | | | | |
| 01 | Page Code = 80h | | | | | | | |
| 02 | RESERVED | | | | | | | |
| 03 | Page Length = 0Bh | | | | | | | |
| 04 | Unit Serial Number | | | | | | | |
| 05 | | | | | | | | |
| 06 | | | | | | | | |
| 07 | | | | | | | | |
| 08 | | | | | | | | |
| 09 | | | | | | | | |
| 10 | | | | | | | | |
| 11 | | | | | | | | |
| 12 | | | | | | | | |
| 13 | | | | | | | | |
| 14 | End of String = 00h | | | | | | | |

*Table: Unit Serial Number Page*

**Qualifier** This field will normally be set to zero. When an Unsupported LUN condition exists (LUN field in the Command Descriptor Block or IDENTIFY message is not set to zero), this field will, however, be set to 3.

**Peripheral Device Type** This field is normally set to 1. When an Unsupported LUN condition exists (LUN field in the Command Descriptor Block or IDENTIFY message is not set to zero), this field will, however, be set to 1Fh.

**Page Code** This field will be set to the value of the Page Code field in the Command Descriptor Block.

**Page Length** This field will specify the length in bytes of the parameters that follow the Page Length field.

**Unit Serial Number** This is the unit serial number represented with 10 ASCII characters. The string is terminated with a zero (00h) character.

### 10.3.2.3. Implemented Operating Definitions

| BYTE | BIT 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 00 | Qualifier | | | Peripheral Device Type | | | | |
| 01 | Page Code = 81h | | | | | | | |
| 02 | RESERVED | | | | | | | |
| 03 | Page Length = 02h | | | | | | | |
| 04 | R | Current Operating Definition = 03h | | | | | | |
| 05 | SImp | Default Operating Definition = 03h | | | | | | |

*Table: Implemented Operating Definitions Page*

**Qualifier** This field will normally be set to zero. When an Unsupported LUN condition exists (LUN field in the Command Descriptor Block or IDENTIFY message is not set to zero), this field will, however, be set to 3.

**Peripheral Device Type** This field is normally set to 1. When an Unsupported LUN condition exists (LUN field in the Command Descriptor Block or IDENTIFY message is not set to zero), this field will, however, be set to 1Fh.

**Page Code** This field will be set to the value of the Page Code field in the Command Descriptor Block.

**Page Length** This field will specify the length in bytes of the parameters that follow the Page Length field.

**Current Operating Definition** This field will be set to 03h to indicate that the Drive implements the SCSI-2 X3.131-199X Operating Definition.

**SImp** This bit will be set to zero.

**Default Operating Definition** This field will be set to 03h to indicate that the Drive implements the SCSI-2 X3.131-199X Operating Definition.

### 10.3.2.4. ASCII Implemented Operating Definition

| BYTE | BIT 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 00 | Qualifier | | | Peripheral Device Type | | | | |
| 01 | Page Code = 82h | | | | | | | |
| 02 | RESERVED | | | | | | | |
| 03 | Page Length = 13h | | | | | | | |
| 04 | ASCII Operating Definition | | | | | "S" | | |
| 05 | | | | | | "C" | | |
| 06 | | | | | | "S" | | |
| 07 | | | | | | "I" | | |
| 08 | | | | | | "-" | | |
| 09 | | | | | | "2" | | |
| 10 | | | | | | " " | | |
| 11 | | | | | | "X" | | |
| 12 | | | | | | "3" | | |
| 13 | | | | | | "." | | |
| 14 | | | | | | "1" | | |
| 15 | | | | | | "3" | | |
| 16 | | | | | | "1" | | |
| 17 | | | | | | "-" | | |
| 18 | | | | | | "1" | | |
| 19 | | | | | | "9" | | |
| 20 | | | | | | "9" | | |
| 21 | | | | | | "X" | | |
| 22 | End Of String = 00h | | | | | | | |

*Table: ASCII Implemented Operating Definition Page*

**Qualifier** This field will normally be set to zero. When an Unsupported LUN condition exists (LUN field in the Command Descriptor Block or IDENTIFY message is not set to zero), this field will, however, be set to 3.

**Peripheral Device Type** This field is normally set to 1. When an Unsupported LUN condition exists (LUN field in the Command Descriptor Block or IDENTIFY message is not set to zero), this field will, however, be set to 1Fh.

**Page Code**

This field will be set to the value of the Page Code field in the Command Descriptor Block.

**Page Length**

This field will specify the length in bytes of the parameters that follow the Page Length field.

**ASCII Operating Definition**

This field will hold the string "SCSI-2 X3.131-199X" terminated with a zero (00h) character.

### 10.3.2.5. Hardware Revision Levels

| BYTE | BIT 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 00 | Qualifier | | | Peripheral Device Type | | | | |
| 01 | Page Code = C0h | | | | | | | |
| 02 | RESERVED | | | | | | | |
| 03 | Page Length = 0Dh | | | | | | | |
| 04<br>05 | Capstan Motor Assembly Revision Level | | | | | | | |
| 06<br>07 | Step Motor Assembly Revision Level | | | | | | | |
| 08<br>09 | Sensor Assembly Revision Level | | | | | | | |
| 10<br>11 | Mainboard Assembly Revision Level | | | | | | | |
| 12<br>13 | Basic Mechanics Assembly Revision Level | | | | | | | |
| 14<br>15 | Head Assembly Revision Level | | | | | | | |
| 16 | End Of String = 00h | | | | | | | |

*Table: Hardware Revision Levels Page*

**Qualifier** This field will normally be set to zero. When an Unsupported LUN condition exists (LUN field in the Command Descriptor Block or IDENTIFY message is not set to zero), this field will, however, be set to 3.

**Peripheral Device Type** This field is normally set to 1. When an Unsupported LUN condition exists (LUN field in the Command Descriptor Block or IDENTIFY message is not set to zero), this field will, however, be set to 1Fh.

**Page Code** This field will be set to the value of the Page Code field in the Command Descriptor Block.

**Page Length** This field will specify the length in bytes of the parameters that follow the Page Length field.

**Revision Levels** These are all two ASCII characters representing the revision level of different parts of the Drive's hardware. The string is terminated with a zero (00h) character.

### 10.3.2.6. ROM Software Code Revision Level

| BYTE | BIT 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 00 | Qualifier | | | Peripheral Device Type | | | | |
| 01 | Page Code = C1h | | | | | | | |
| 02 | RESERVED | | | | | | | |
| 03 | Page Length = 07h | | | | | | | |
| 04<br>05<br>06<br>07<br>08<br>09 | ROM Software Code Revision Level | | | | | | | |
| 10 | End Of String = 00h | | | | | | | |

*Table: ROM Software Code Revision Level*

**Qualifier**

This field will normally be set to zero. When an Unsupported LUN condition exists (LUN field in the Command Descriptor Block or IDENTIFY message is not set to zero), this field will, however, be set to 3.

**Peripheral Device Type**

This field is normally set to 1. When an Unsupported LUN condition exists (LUN field in the Command Descriptor Block or IDENTIFY message is not set to zero), this field will, however, be set to 1Fh.

**Page Code**

This field will be set to the value of the Page Code field in the Command Descriptor Block.

**Page Length**

This field will specify the length in bytes of the parameters that follow the Page Length field.

**ROM Software Code Revision Level**

This is 6 ASCII characters representing the software revision and option levels. The string is terminated with a zero (00h) character.

### 10.3.2.7. Drive Manufacturing Date

| BYTE | BIT 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 00 | Qualifier | | | Peripheral Device Type | | | | |
| 01 | Page Code = C2h | | | | | | | |
| 02 | RESERVED | | | | | | | |
| 03 | Page Length = 09h | | | | | | | |
| 04<br>05<br>06<br>07<br>08<br>09<br>10<br>11 | Drive Manufacturing Date | | | | | | | |
| 12 | End Of String = 00h | | | | | | | |

*Table: Drive Manufacturing Date Page*

**Qualifier** This field will normally be set to zero. When an Unsupported LUN condition exists (LUN field in the Command Descriptor Block or IDENTIFY message is not set to zero), this field will, however, be set to 3.

**Peripheral Device Type** This field is normally set to 1. When an Unsupported LUN condition exists (LUN field in the Command Descriptor Block or IDENTIFY message is not set to zero), this field will, however, be set to 1Fh.

**Page Code** This field will be set to the value of the Page Code field in the Command Descriptor Block.

**Page Length** This field will specify the length in bytes of the parameters that follow the Page Length field.

**Drive Manufacturing Date** This is 8 ASCII characters representing the manufacturing date on the format 'MM.DD.YY'. The string is terminated with a zero (00h) character.

### 10.3.2.8. ROM Software Creation Date

| BYTE | BIT 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 00 | Qualifier | | | Peripheral Device Type | | | | |
| 01 | Page Code = C3h | | | | | | | |
| 02 | RESERVED | | | | | | | |
| 03 | Page Length = 09h | | | | | | | |
| 04<br>05<br>06<br>07<br>08<br>09<br>10<br>11 | ROM Software Creation Date | | | | | | | |
| 12 | End Of String = 00h | | | | | | | |

*Table: ROM Software Creation Date Page*

**Qualifier** This field will normally be set to zero. When an Unsupported LUN condition exists (LUN field in the Command Descriptor Block or IDENTIFY message is not set to zero), this field will, however, be set to 3.

**Peripheral Device Type** This field is normally set to 1. When an Unsupported LUN condition exists (LUN field in the Command Descriptor Block or IDENTIFY message is not set to zero), this field will, however, be set to 1Fh.

**Page Code** This field will be set to the value of the Page Code field in the Command Descriptor Block.

**Page Length** This field will specify the length in bytes of the parameters that follow the Page Length field.

**ROM Software Creation Date** This is 8 ASCII characters representing the software creation date on the format 'MM.DD.YY'. The string is terminated with a zero (00h) character.

### 10.3.2.9. Drive Adjustment Date

| BYTE | BIT 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 00 | Qualifier | | | Peripheral Device Type | | | | |
| 01 | Page Code = C4h | | | | | | | |
| 02 | RESERVED | | | | | | | |
| 03 | Page Length = 09h | | | | | | | |
| 04<br>05<br>06<br>07<br>08<br>09<br>10<br>11 | Drive Adjustment Date | | | | | | | |
| 12 | End Of String = 00h | | | | | | | |

*Table: Drive Adjustment Date Page*

**Qualifier** This field will normally be set to zero. When an Unsupported LUN condition exists (LUN field in the Command Descriptor Block or IDENTIFY message is not set to zero), this field will, however, be set to 3.

**Peripheral Device Type** This field is normally set to 1. When an Unsupported LUN condition exists (LUN field in the Command Descriptor Block or IDENTIFY message is not set to zero), this field will, however, be set to 1Fh.

**Page Code** This field will be set to the value of the Page Code field in the Command Descriptor Block.

**Page Length** This field will specify the length in bytes of the parameters that follow the Page Length field.

**Drive Adjustment Date** This is 8 ASCII characters representing the date of the last drive adjustment on the format 'MM.DD.YY'. The string is terminated with a zero (00h) character.

## 10.4. Exception Handling

See section on Error Conditions For All Commands.

If the EVPD bit is not set and the Page Code is not set to zero, the Drive will return CHECK CONDITION status. No parameter data will be sent. The Error Code will be set to E$STE_IFIC.

If an INQUIRY command is received with a pending Unit Attention Condition, the Drive will perform the INQUIRY command and will not clear the Unit Attention Condition.

*This Page Intentionally Left Blank*

# 11. Load/Unload

## 11.1. Command Description

The LOAD/UNLOAD command requests the Drive to position a newly inserted tape cartridge at BOT. This command may also be used to request a retension function. The UNLOAD command requests the Drive to prepare a loaded tape cartridge for removal. The UNLOAD command can position the cartridge at either BOT or EOT.

NOTE:
It is possible to configure the Drive for both Auto Load and Auto Retension operation (see MODE SELECT Section for details).

After a successful execution of a LOAD command (Load bit set to one), all media access commands will be legal. After a successful execution of an UNLOAD command (Load bit set to zero) all media access commands will be terminated with CHECK CONDITION (See also the sections on COMMAND DESCRIPTORS and GENERAL EXCEPTION HANDLING).

If immediate operation is requested (IMM bit set to one), the Drive will be logically loaded (Load bit set to one) or unloaded (Load bit set to zero) even if the tape has not reach its final destination.

If immediate operation is not requested (IMM bit set to zero), then prior to performing an LOAD or UNLOAD operation, the Drive will write any buffered data that is to be written to the tape.

When the Drive is in buffered mode it will discard any buffered data when a LOAD/UNLOAD command is validated if the previous command was terminated with CHECK CONDITION.

It will not be considered an error when multiple LOAD/UNLOAD commands are received in a sequence (as long as the first LOAD/UNLOAD command executed with no error).

The Drive will disconnect during execution of this command if disconnection is allowed.

## 11.2. Command Descriptor Block

| BYTE | BIT 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 00 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 |
| 01 | Logical Unit Number (LUN) | | | RESERVED | | | | IMM |
| 02 | RESERVED | | | | | | | |
| 03 | RESERVED | | | | | | | |
| 04 | RESERVED | | | | | EOT | RET | Load |
| 05 | Control Byte | | | | | | | |

*Table: LOAD/UNLOAD Command Block*

**IMM**

An Immediate (IMM) bit of zero indicates that the Drive will not return status until the LOAD/UNLOAD operation has completed. An IMM bit of one indicates that the Drive will return status as soon as the execution of all previous immediate commands have been completed and the Command Descriptor Block of the LOAD/UNLOAD command has been validated. Note that the LINK bit MUST be zero if the IMM bit is set.

**RET**

A Retension (RET) bit of one indicates that the Drive will perform a retension pass before the load or unload operation is performed. Retension means moving the tape one complete pass between EOT and BOT.

**Load**

A Load bit of one indicates that the tape will be moved to BOT. The tape is logically loaded which means that the Drive is able to accept medium access commands. A Load bit of zero indicates that the tape is logically unloaded. The Drive will no longer accept media access commands.

**EOT**

An End Of Tape (EOT) of zero indicates that the tape will be positioned at BOT after the load/unload operation has been performed. A EOT bit of one indicates that the tape will be positioned at EOT after the load/-unload operation has been performed. This allows fast retensioning of the cartridge next time it is used.

Combinations of the EOT, RET and Load bits are shown in the table below:

| EOT | RET | Load | Operation Performed |
|---|---|---|---|
| 0 | 0 | 0 | Unload, move to BOT |
| 0 | 0 | 1 | Load, move to BOT |
| 0 | 1 | 0 | Retension, Unload, move to BOT |
| 0 | 1 | 1 | Retension, Load, move to BOT |
| 1 | 0 | 0 | Unload, move to EOT |
| 1 | 0 | 1 | Illegal, CHECK CONDITION |
| 1 | 1 | 0 | Retension, Unload, move to EOT |
| 1 | 1 | 1 | Illegal, CHECK CONDITION |

*Table: LOAD/UNLOAD Operations*

## 11.3. Exception Handling

If CHECK CONDITION status is returned for a LOAD/UNLOAD command with an IMM bit of one, the load or unload operation has not been performed.

See sections on Error Conditions For All Commands, Deferred Errors and Error Conditions For Media Access Commands.

If the LINK and IMM bits both are set, then the Drive will return CHECK CONDITION status. The Error Code will be set to E$STE_IFIC.

If an illegal combination of Load, EOT and RET bits is detected, the Drive returns CHECK CONDITION status. The Error Code will be set to E$STE_IFIC.

*This Page Intentionally Left Blank*

# 12. Locate

## 12.1. Command Description

The LOCATE command causes the Drive to position the tape to a specified position in the data stream. Both physical and logical positions are supported.

**LOCATE Physical Command**

The LOCATE Physical command will interpret the Block Address in the CDB as a special key or “bookmark”. All blocks, filemarks and setmarks can be identified with such a key. This position key can be obtained with the use of the READ POSITION command. While writing (or reading), the READ POSITION command is typically executed every time the tape is at a position that the Host system might want to go back to at a later stage. The returned position key can then be stored and used later as an input to the LOCATE command. These values can be regarded as keys that are unique to any given physical position on the tape. It might, however, be several keys that will result in the same logical position, as some physical tape blocks do not have any logical contents (i.e. ECC blocks or Filler blocks). The physical position numbers must be obtained with the READ POSITION command, and must not be manipulated in any way by the Host system. The LOCATE command will then bring the tape back to the same position as it was when the READ POSITION command was executed.

**LOCATE Logical Command**

The LOCATE Logical command does not need special keys to get to a certain logical position. The Block Address in the CDB will for this command be interpreted as the logical block position, meaning the number of blocks seen on the SCSI-bus from BOT. If a tape contains filemarks or setmarks, they will not count as logical blocks. It is therefore not possible to issue a LOCATE Logical command to such a tapemark.

## 12.2. Command Descriptor Block

| BYTE | BIT 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 00 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 |
| 01 | Logical Unit Number (LUN) | | | RESERVED | | BT | CP | R |
| 02 | RESERVED | | | | | | | |
| 03<br>04<br>05<br>06 | Block Location | | | | | | | |
| 07 | RESERVED | | | | | | | |
| 08 | Partition | | | | | | | |
| 09 | Control Byte | | | | | | | |

*Table: LOCATE Command Block*

**BT** When the Block address Type (BT) bit is cleared, the Block Location field in the CDB will be the logical block identifier for the LOCATE operation. If the BT bit is set, the Block Location field in the CDB will be the physical block identifier.

**CP** A change partition (CP) bit of one indicates that a change to the partition specified in the Partition field is to occur prior to positioning to the physical block specified in the Block Location address field (a CP of one is only legal when the Drive is in QFA mode). A CP bit of zero indicates that no partition change is to be made and the Partition field is to be ignored.

**Block Location** The input to the LOCATE command can be either *physical* or *logical* block identifiers. After a successful command execution, the logical tape position will be located before the specified block (beginning.-of-media side).

**Partition**

If the CP bit is one, the Partition field specifies the partition number into which to locate.

The Drive has two valid partitions:

**0 - data partition**
**1 - directory partition**

All other partition numbers are illegal. See Sections 2.3. and 2.4. for how to use partitions.

## 12.3. Exception Handling

See sections on Error Conditions For All Commands, Deferred Errors and Error Conditions For Media Access Commands.

If the CP bit is not set to zero and the Drive is not in QFA mode, the LOCATE command will be terminated with CHECK CONDITION status. The Error Code will be set to E$STE_IFIC.

If a LOCATE PHYSICAL is issued to a ECC block, the Drive will position the tape to the next physical data block.

A LOCATE PHYSICAL BLOCK 0 for QIC-120 and QIC-150 tape formats will always position the tape at the Beginning Of Media (BOM).

If the Drive is not able to find a reference burst on the inserted cartridge, the cartridge is assumed to be blank and the READ command will be terminated with CHECK CONDITION. The Error code will be set to E$TCM_NODATA.

If the specified location can not be found on the tape, the Drive will terminate the LOCATE command with CHECK CONDITION status. The Error Code will be set to E$TEM_EOR or E$TEM_EOREW.

If the number found in the Partition field is not in the range 0..1, then the LOCATE command will return CHECK CONDITION status. The Error Code will be set to E$STE_IFIC. The logical tape position will not change.

If a LOCATE is aborted by an Overlapped command, the command will stop execution and the tape will be positioned at BOT on the current partition.

***SLR5 4.0/8.0GB and SLR4 2.5/5.0GB (TDC 4222) only***

For the SLR5 4.0/8.0GB and SLR4 2.5/5.0GB (TDC 4222), see the READ command Sections 18.3.9. Data Compression Exception Handling and 18.3.10. Reading From the Beginning of the Directory Partition.

*This Page Intentionally Left Blank*

# 13. Log Select

## 13.1. Command Description

The LOG SELECT command can be used to clear the statistical information maintained by the Drive. It is also possible to write any value into the different statistical counters.

The LOG-information includes the number of blocks and filemarks read and written, the number of buffer over and underruns, the number of recoverable read and write errors and the number of corrections performed by the ECC.

If a parameter list is needed, it will be transferred during the DATA-OUT phase of the command.

If disconnect is allowed, the Drive will disconnect while the transferred parameters are processed.

## 13.2. Command Descriptor Block

| BYTE | BIT 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 00 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 |
| 01 | Logical Unit Number (LUN) | | | RESERVED | | | PCR | SP |
| 02 | PC | | RESERVED | | | | | |
| 03 | RESERVED | | | | | | | |
| 04 | RESERVED | | | | | | | |
| 05 | RESERVED | | | | | | | |
| 06 | RESERVED | | | | | | | |
| 07 | Parameter List Length (MSB) | | | | | | | |
| 08 | Parameter List Length (LSB) | | | | | | | |
| 09 | Control Byte | | | | | | | |

*Table: LOG SELECT Command Block*

**PCR** If the Parameter Code Reset (PCR) bit is be set to zero, all counters are cleared. In this case the Parameter List Length must be zero and no parameters are transferred. Any legal value in the PC-bits is ignored.

**SP** The Drive do not support Saveable Parameters. This bit must always be set to zero.

**PC**

Only “01b” and “11b” are legal values for the two Page Control (PC) bits. If these bits are set to “01b”, the Host can update any or all the LOG-counters with zeroes or any other legal value. New counter values are transferred as LOG-pages in the DATA-OUT phase of the command.
If PC is set to "11b”, the pages are updated with the Drive default values. In this Drive the default values are all “00h”. No parameters are transferred and the Parameter List Length must be “00h”.

**Parameter List Length**

These two bytes specifies the length of the parameter list that will be transferred during the DATA-OUT phase of the command. A Parameter List Length of zero indicates that no parameters will be transferred.

## 13.3. Parameter List

### *13.3.1. General Parameter Description*

See the LOG SENSE command for a description of the different LOG-pages supported by the Drive.

Since Page 00 only contains unchangeable parameters (list of supported LOG-pages), this page is not supported by LOG SELECT.

A LOG-page consists of a 4-byte header and one or more parameter blocks. The parameter blocks are either 6 or 8 blocks long for this Drive. In the header the 2-byte page length is set to the sum of the length of all transferred parameter blocks.

The parameter list length in the command block, or the page length in the parameter header must not result in truncation of any LOG-parameters.

In LOG-pages with more than one parameter block, it is not required to transfer all parameters. If more than one parameter block is transferred in a page, the parameters must be transferred in an ascending order by parameter code.

If more than one LOG-page is transferred in a LOG SELECT command, the pages must be transferred in an ascending order by page number.

### *13.3.2. Buffer Over/Underrun Counter Page*

The Page Code is 01h for this page, and it supports two 6 byte parameter blocks. If only one block is transferred, the Page Length in the header must be 0006h. If both blocks are transferred, the Page Length must be 000Ch.

For both parameter blocks the Count Base and the Cause-fields must be set to 0h. In the underrun parameter block, bit 0 in byte 1 must be 0 while in the overrun parameter block the same bit must be set to 1.

In the Parameter Control Byte the DS-bit must be 1h. The remaining bits (DU, TDS, ET, TMC and LP) must be set to 0h. This is valid for both parameter blocks, even if the counters are set to FFFFh.

The Parameter Length byte must be set to 02h.

The two-byte counters can be set to any value from 0000h to FFFFh.

### 13.3.3. Recoverable Write Error Counter Page

The Page Code is 02h for this page, and it supports only one 6-byte parameter block. The Page Length in the header must be set to 0006h.

The parameter code must be set to 0002h.

In the Parameter Control Byte the DS-bit must be 1h. The remaining bits (DU, TDS, ET, TMC and LP) must be set to 0h. This is valid even if the counter is set to FFFFh.

The Parameter Length byte must be set to 02h.

The two-byte counter can be set to any value from 0000h to FFFFh.

### 13.3.4. Recoverable Read Error Counter Page

The Page Code is 03h for this page, and it supports two 6-byte parameter blocks. If only one block is transferred, the Page Length in the header must be 0006h. If both blocks are transferred, the Page Length must be 000Ch.

In the Reread Error Counter parameter block the parameter code must be set to 0001h, while in the ECC correction counter block the parameter code must be set to 0004h.

In the Parameter Control Byte the DS-bit must be 1h. The remaining bits (DU, TDS, ET, TMC and LP) must be set to 0h. This is valid even if the counters are set to FFFFh.

The Parameter Length byte must be set to 02h for both blocks.

The two-byte counter can be set to any value from 0000h to FFFFh.

### 13.3.5. Block Counter Page

The Page Code is 30h for this page, and it supports only one 8-byte parameter block. The Page Length in the header must be set to 0008h.

The parameter code must be set to 0000h.

In the Parameter Control Byte the DS-bit must be 1h. The remaining bits (DU, TDS, ET, TMC and LP) must be set to 0h. This is valid even if the counter is set to FFFFh.

The Parameter Length byte must be set to 04h.

The two-byte counter can be set to any value from 00000000h to FFFFFFFFh.

### 13.3.6. Filemark Counter Page

The Page Code is 31h for this page, and it supports only one 8-byte parameter block. The Page Length in the header must be set to 0008h.

The parameter code must be set to 0000h.

In the Parameter Control Byte the DS-bit must be 1h. The remaining bits (DU, TDS, ET, TMC and LP) must be set to 0h. This is valid even if the counter is set to FFFFh.

The Parameter Length byte must be set to 04h.

The two-byte counter can be set to any value from 0000h to FFFFh.

## 13.4. Exception Handling

See the section on Error Conditions For All Commands.

If any of the bits PCR, SP or PC have invalid values, the Drive will return CHECK CONDITION status. No parameters will be transferred, and the Error Code will be E$STE_IFIP.

If any bits or bytes in the transferred parameters have invalid values, the Drive will report CHECK CONDITION status and the Error Code will be E$STE_IFIP.

If the combination of the Parameter List Length and the real length of the requested LOG-pages results in truncation of any LOG-parameter, the Drive will report CHECK CONDITION status and the Error Code will be E$STE_IFIC.

# 14. Log Sense

## 14.1. Command Description

The LOG SENSE command is used to retrieve statistical information maintained by the Drive about the current cartridge.

The log information includes the number of blocks and filemarks written or read, the number of buffer underruns and overruns, the number of recovered errors and the number of blocks corrected by ECC.

The parameter list will be transferred during the DATA IN phase of the command.

If disconnection is allowed, the Drive will disconnect when executing this command if the previous command was an immediate type command.

## 14.2. Command Descriptor Block

| BYTE | BIT 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 00 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 |
| 01 | Logical Unit Number (LUN) | | | RESERVED | | | PPC | SP |
| 02 | PC | | Page Code | | | | | |
| 03 | RESERVED | | | | | | | |
| 04 | RESERVED | | | | | | | |
| 05<br>06 | Parameter Pointer | | | | | | | |
| 07<br>08 | Allocation Length | | | | | | | |
| 09 | Control Byte | | | | | | | |

*Table: LOG SENSE Command Block*

**PPC** The parameter Pointer Control (PPC) bit MUST be set to zero.

**SP** The Save Parameter (SP) bit MUST be set to zero.

**PC** The Page Control (PC) field MUST be set to 01h. This indicates that the Drive will transfer the Current Cumulative log parameter values.

**Page Code**

The Page Code field specifies which page to return. Legal Page Codes are:

| | | |
|---|---|---|
| **00h** | : | Supported Log Pages Page |
| **01h** | : | Buffer Overrun/Underrun Counters Page |
| **02h** | : | Recoverable Write Error Counter Page |
| **03h** | : | Recoverable Read Error Counters Page |
| **30h** | : | Block Counter Page |
| **31h** | : | Tape Capacity Page |
| **32h** | : | Filemark Counter Page |
| **36h** | : | Physical Position Page |
| **3Eh** | : | All Log Pages *) |

NOTE *):
If Log Sense is issued with a Page Code = 3Eh, the Drive will return all Log Pages in the same order as listed above.

**Parameter Pointer**

The Parameter Pointer field MUST be set to zero to indicate that the Drive always transfers all supported parameter codes for each page.

**Allocation Length**

This field specifies the maximum number of bytes that the Initiator has allocated for returned LOG SENSE data. An Allocation Length of zero indicates that no LOG SENSE data will be sent. The Drive terminates the DATA IN phase when Allocation Length bytes have been transferred or when all available LOG SENSE data have been transferred, whichever is less.

## 14.3. Parameter List

### 14.3.1. Supported Log Pages

| BYTE | BIT 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 00 | RESERVED | | Page Code = 00h | | | | | |
| 01 | RESERVED | | | | | | | |
| 02<br>03 | Page Length = 0008h | | | | | | | |
| 04 | Supported Page = 00h | | | | | | | |
| 05 | Supported Page = 01h | | | | | | | |
| 06 | Supported Page = 02h | | | | | | | |
| 07 | Supported Page = 03h | | | | | | | |
| 08 | Supported Page = 30h | | | | | | | |
| 09 | Supported Page = 31h | | | | | | | |
| 10 | Supported Page = 32h | | | | | | | |
| 11 | Supported Page = 36h | | | | | | | |

*Table: Supported Log Pages Page*

**Page Code** The Page Code for this page is always set to 00h (zero).

**Page Length** This field is always set to 8 indicating that the Supported Page List holds 8 bytes (8 pages supported).

### 14.3.2. Buffer Overrun/Underrun Counters Page

| BYTE | BIT 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 00 | RESERVED | | | Page Code = 01h | | | | |
| 01 | RESERVED | | | | | | | |
| 02<br>03 | Page Length = 000Ch | | | | | | | |
| 04<br>05<br>06<br>07<br>08<br>09 | Underrun Log Parameters | | | | | | | |
| 10<br>11<br>12<br>13<br>14<br>15 | Overrun Log Parameters | | | | | | | |

*Table: Buffer Overrun/Underrun Counter Page*

**Page Code** The Page Code for this page is always set to 01h.

**Page Length** This field is always set to 12 indicating that two counter lists with 6 bytes in each are supported.

### Underrun Log Parameters

The Underrun Log Parameters have the following format:

| BYTE | BIT 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| n+0 | RESERVED | | | | | | | |
| n+1 | Count Basis | | | Cause | | | | 0 |
| n+2 | DU | DS | TSD | ET | TMC | | R | LP |
| n+3 | Parameter Length = 02h | | | | | | | |
| n+4<br>n+5 | Underrun Counter | | | | | | | |

*Table: Buffer Underrun Log Parameter*

**Count Basis** This field will be set to 000b to indicate that the criteria for incrementing the counter is undefined.

**Cause** This field will be set to 0h to indicate that the reason for underrun is undefined.

**DU** The Disable Update (DU) bit will normally be set to zero to indicate that the Drive is always free to update the counter. When the counter reaches its maximum values (FFFFh) the DU bit will be set to one.

**DS** The Disable Save (DS) bit will be set to one to indicate that the Drive does not support saving of this log parameter.

**TSD** The Target Save Disable (TSD) bit will be set to zero.

**ET** The Enable Threshold (ET) bit will be set to zero to indicate that no threshold comparison will take place.

**TMC** The Threshold Met Criteria (TMC) field will be set to zero.

**LP** The List Parameters (LP) bit will be set to zero to indicate that the parameter is a data counter.

**Parameter Length** This field will be set to 2 to indicate that the counter is 16 bits wide.

**Underrun Counter** This counter counts Underruns (during tape write operations). After the Drive has been reset the counter contains a value of zero. Upon reaching the maximum value the counter does not wrap back to zero, but retains its maximum value (FFFFh).

### Overrun Log Parameters

The Overrun Log Parameters has the following format:

| BYTE | BIT 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| n+0 | RESERVED | | | | | | | |
| n+1 | Count Basis | | | Cause | | | | 1 |
| n+2 | DU | DS | TSD | ET | TMC | | R | LP |
| n+3 | Parameter Length = 02h | | | | | | | |
| n+4 | Overrun Counter | | | | | | | |
| n+5 | | | | | | | | |

*Table: Buffer Overrun Log Parameter*

**Count Basis** This field will be set to 000b to indicate that the criteria for incrementing the counter is undefined.

**Cause** This field will be set to 0h to indicate that the reason for underrun is undefined.

**DU** The Disable Update (DU) bit will normally be set to zero to indicate that the Drive is always free to update the counter. When the counter reaches its maximum values (FFFFh) the DU bit will be set to one.

**DS** The Disable Save (DS) bit will be set to one to indicate that the Drive does not support saving of this log parameter.

**TSD** The Target Save Disable (TSD) bit will be set to zero.

**ET** The Enable Threshold (ET) bit will be set to zero to indicate that no threshold comparison will take place.

**TMC** The Threshold Met Criteria (TMC) field will be set to zero.

**LP** The List Parameters (LP) bit will be set to zero to indicate that the parameter is a data counter.

**Parameter Length** This field will be set to 2 to indicate that the counter is 16 bits wide.

**Overrun Counter** This counter counts Overruns (during tape read operations). After the Drive have been reset the counter contains a value of zero. Upon reaching the maximum value the counter does not wrap back to zero, but retains its maximum value (FFFFh).

### 14.3.3. Recoverable Write Error Counter Page

| BYTE | BIT 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 00 | RESERVED | | Page Code = 02h | | | | | |
| 01 | RESERVED | | | | | | | |
| 02 | Page Length = 06h | | | | | | | |
| 03 | | | | | | | | |
| 04 | Parameter Code = 0002h | | | | | | | |
| 05 | | | | | | | | |
| 06 | DU | DS | TSD | ET | TMC | | R | LP |
| 07 | Parameter Length = 02h | | | | | | | |
| 08 | Rewrite Error Counter | | | | | | | |
| 09 | | | | | | | | |

*Table: Recoverable Write Error Counter Page*

**Page Code** The Page Code for this page is always set to 02h.

**Page Length** This field is always set to 6 indicating that one lists with 6 bytes follows.

**Parameter Code** The Parameter Code will be set to 0002h to indicate that the counter counts the total number of re-writes.

**DU** The Disable Update (DU) bit will normally be set to zero to indicate that the Drive is always free to update the counter. When the counter reaches its maximum values (FFFFh) the DU bit will be set to one.

**DS** The Disable Save (DS) bit will be set to one to indicate that the Drive does not support saving of this log parameter.

**TSD** The Target Save Disable (TSD) bit will be set to zero.

**ET** The Enable Threshold (ET) bit will be set to zero to indicate that no threshold comparison will take place.

**TMC** The Threshold Met Criteria (TMC) field will be set to zero.

**LP** The List Parameters (LP) bit will be set to zero to indicate that the parameter is a data counter.

**Parameter Length** This field will be set to 2 to indicate that the counter is 16 bits wide.

**Rewrite Error Counter** This counter counts re-writes during write operations. When a block is re-written (one or more times), the counter is incremented by one. After the Drive have been reset the counter contains a value of zero. Upon reaching the maximum value the counter does not wrap back to zero, but retains its maximum value (FFFFh).

## 14.3.4. Recoverable Read Error Counters Page

| BYTE | BIT 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 00 | RESERVED | | | Page Code = 03h | | | | |
| 01 | RESERVED | | | | | | | |
| 02<br>03 | Page Length = 000Ch | | | | | | | |
| 04<br>05<br>06<br>07<br>08<br>09 | Reread Error Counter Parameters | | | | | | | |
| 10<br>11<br>12<br>13<br>14<br>15 | ECC Correction Counter Parameters | | | | | | | |

*Table: Recoverable Read Error Counters Page*

**Page Code** The Page Code for this page is always set to 03h.

**Page Length** This field is always set to 12 indicating that two counter lists with 6 bytes in each are supported.

### Reread Error Counter Parameters

The Reread Counters Parameters has the following format:

| BYTE | BIT 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| n+0<br>n+1 | Parameter Code = 0001h | | | | | | | |
| n+2 | DU | DS | TSD | ET | TMC | | R | LP |
| n+3 | Parameter Length = 02h | | | | | | | |
| n+4<br>n+5 | Reread Error Counter | | | | | | | |

*Table: Reread Error Counter Parameters*

**Parameter Code** The Parameter Code will be set to 0001h to indicate that the counter counts the total number of rereads.

**DU** The Disable Update (DU) bit will normally be set to zero to indicate that the Drive is always free to update the counter. When the counter reaches its maximum values (FFFFh) the DU bit will be set to one.

**DS** The Disable Save (DS) bit will be set to one to indicate that the Drive does not support saving of this log parameter.

**TSD** The Target Save Disable (TSD) bit will be set to zero.

**ET** The Enable Threshold (ET) bit will be set to zero to indicate that no threshold comparison will take place.

**TMC** The Threshold Met Criteria (TMC) field will be set to zero.

**LP** The List Parameters (LP) bit will be set to zero to indicate that the parameter is a data counter.

**Parameter Length** This field will be set to 2 to indicate that the counter is 16 bits wide.

**Reread Error Counter** This counter counts re-reads during read operations. When a block must be reread (one or more times), this counter is incremented by one. After the Drive have been reset the counter contains a value of zero. Upon reaching the maximum value the counter does not wrap back to zero, but retains its maximum value (FFFFh).

**ECC Correction Counter Parameters**

The ECC Correction Counter Parameters has the following format:

| BYTE | BIT 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| n+0<br>n+1 | Parameter Code = 0004h | | | | | | | |
| n+2 | DU | DS | TSD | ET | TMC | | R | LP |
| n+3 | Parameter Length = 02h | | | | | | | |
| n+4<br>n+5 | ECC Error Counter | | | | | | | |

*Table: ECC Correction Counter Parameters*

**Parameter Code**

The Parameter Code will be set to 0004h to indicate that the counter counts the total times correction algorithm has been processed.

**DU**

The Disable Update (DU) bit will normally be set to zero to indicate that the Drive is always free to update the counter. When the counter reaches its maximum values (FFFFh) the DU bit will be set to one.

**DS**

The Disable Save (DS) bit will be set to one to indicate that the Drive does not support saving of this log parameter.

**TSD**

The Target Save Disable (TSD) bit will be set to zero.

**ET**

The Enable Threshold (ET) bit will be set to zero to indicate that no threshold comparison will take place.

**TMC**

The Threshold Met Criteria (TMC) field will be set to zero.

**LP**

The List Parameters (LP) bit will be set to zero to indicate that the parameter is a data counter.

**Parameter Length**

This field will be set to 2 to indicate that the counter is 16 bits wide.

**ECC Correction Counter**

This counter counts the number of blocks corrected by ECC during tape read operations. After the Drive have been reset the counter contains a value of zero. Upon reaching the maximum value the counter does not wrap back to zero, but retains its maximum value (FFFFh).

### 14.3.5. Block Counter Page

| BYTE | BIT 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 00 | RESERVED | | Page Code = 30h | | | | | |
| 01 | RESERVED | | | | | | | |
| 02 | Page Length = 0008h | | | | | | | |
| 03 | | | | | | | | |
| 04 | Parameter Code = 0000h | | | | | | | |
| 05 | | | | | | | | |
| 06 | DU | DS | TSD | ET | TMC | | R | LP |
| 07 | Parameter Length = 04h | | | | | | | |
| 08 | Block Counter | | | | | | | |
| 09 | | | | | | | | |
| 10 | | | | | | | | |
| 11 | | | | | | | | |

*Table: Block Counter Page*

**Page Code** The Page Code for this page is always set to 30h.

**Page Length** This field is always set to 8 indicating that one lists with 8 bytes follows.

**Parameter Code** The Parameter Code will be set to 0000h.

**DU** The Disable Update (DU) bit will normally be set to zero to indicate that the Drive is always free to update the counter. When the counter reaches its maximum value (FFFFFFh) the DU bit will be set to one.

**DS** The Disable Save (DS) bit will be set to one to indicate that the Drive does not support saving of this log parameter.

**TSD** The Target Save Disable (TSD) bit will be set to zero.

**ET** The Enable Threshold (ET) bit will be set to zero to indicate that no threshold comparison will take place.

**TMC** The Threshold Met Criteria (TMC) field will be set to zero.

**LP** The List Parameters (LP) bit will be set to zero to indicate that the parameter is a data counter.

**Parameter Length** This field will be set to 4 to indicate that the counter is 32 bits wide.

**Data Block Counter** This counter counts the number of data blocks transferred to (during read operations) or from (during write operations) the Initiator. A variable block (of any length) counts as one block. After the Drive have been reset the counter contains a value of zero. Upon reaching the maximum value the counter does not wrap back to zero, but retains its maximum value (FFFFFFFFh).

### 14.3.6. Tape Capacity Page

| BYTE | BIT 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 00 | Page Code = 31h | | | | | | | |
| 01 | RESERVED | | | | | | | |
| 02<br>03 | Page Length = 20h | | | | | | | |
| 04<br>05 | Parameter Code (1) | | | | | | | |
| 06 | DU | DS | TSD | ET | TMC | | R | LP |
| 07 | Parameter Length (4) | | | | | | | |
| 08-09<br>10-11 | Remaining Capacity, Partition 0 | | | | | | | |
| 12<br>13 | Parameter Code (2) | | | | | | | |
| 14 | DU | DS | TSD | ET | TMC | | R | LP |
| 15 | Parameter Length (4) | | | | | | | |
| 16-17<br>18-19 | Remaining Capacity, Partition 1 | | | | | | | |
| 20<br>21 | Parameter Code (3) | | | | | | | |
| 22 | DU | DS | TSD | ET | TMC | | R | LP |
| 23 | Parameter Length (4) | | | | | | | |
| 24-25<br>26-27 | Maximum Capacity, Partition 0 | | | | | | | |
| 28<br>29 | Parameter Code (4h) | | | | | | | |
| 30 | DU | DS | TSD | ET | TMC | | R | LP |
| 31 | Parameter Length (4) | | | | | | | |
| 32-33<br>34-35 | Maximum Capacity, Partition 1 | | | | | | | |

*Table: Tape Capacity Page*

**Page Code** The Page Code for this page is always set to 31h.

**Page Length** This field is always set to 20h.

**Parameter Code** Identifies the parameters within the page. Values used for the four parameters are 1, 2, 3 and 4.

**DU** The Disable Update (DU) bits are set to zero for all the parameters to indicate that the parameters are updated by the Drive.

**DS** The Disable Save (DS) bits are set to one for all parameters to indicate that the Drive does not support saving of parameters.

**TSD** The Target Save Disable (TSD) bits are set to zero for all parameters. This field is only significant if the DS-bit is cleared.

**ET**

The Enable Threshold (ET) bits are set to zero for all parameters to indicate that no threshold comparison will take place.

**TMC**

The Threshold Met Criteria (TMC) bits are set to zero for all parameters. This field is only valid if the ETC-bit is set.

**LP**

The List Parameters (LP) bits are set to zero for all parameters to indicate that the parameter is a counter and not a list parameter (ASCII string).

**Parameter length**

These fields are always set to 4 to indicate that the parameters are 32 bits wide.

**Remaining Capacity Partition 0**

For dual partition tapes, when Partition 1 is the current partition, the parameter is set to the Maximum Capacity Partition 0 parameter.

When Partition 0 is the current partition and for single partition tapes, the parameter will return the remaining capacity of this partition in KBytes.

Remaining Capacity is updated and valid after a successful SPACE TO EOD, WRITE FILEMARK non-immediate mode and a WRITE in unbuffered mode. After a successful completion of the following commands in non-immediate mode; LOAD, REWIND, LOCATE, SPACE blocks and filemarks, VERIFY, READ and MODE SELECT, the parameter will reflect the capacity from the current tape position to the end of the partition and not the actual remaining capacity of the partition. For capacity calculation, see NOTE.

**Remaining Capacity Partition 1**

For single partition tapes this parameter is set to zero (0).

For dual partiton tapes the parameter is set equal to the Maximum Capacity Partition 1 parameter when Partition 0 is the current partition.

When Partition 1 is the current partition, the parameter will reflect the remaining capacity of this partition in KBytes in the same way as described for the Remaining capacity Partition 0 parameter. For capacity calculations, see NOTE.

**Maximum Capacity Partition 0**

The Maximum Capacity Partition 0 parameter specifies the maximum capacity of this partition in KBytes. The parameter is only valid once the tape has successfully completed a LOAD sequence and the tape movement has ceased. For capacity claculations, see NOTE.

**Maximum Capacity Partition 1**

For single partiton tapes the Maximum Capacity parameter is set to zero (0).

For dual partition tapes the parameter specifies the maximum capacity of this partition in KBytes. The parameter is only valid once the tape has successfully completed a LOAD sequence and the tape movement has ceased. For capacity calculations, see NOTE.

> NOTE:
> The capacities are specified in KBytes (no. of 1024-byte blocks) of user data based on a conservative estimate.
>
> However, if a DC6320 (620 ft.) or a DC6525 (1020 ft.) is inserted, the default remaining capacity will be according to a DC6525 (1020 ft.). If a DC6150 (620 ft.) or a DC 6250 (1020 ft.) is inserted, the default remaining capacity will be according to a DC6250 (1020 ft.)
>
> System Log areas, Vendor Group and Early Warning areas are not included to ensure a conservative value.
>
> ECC-block writing is allowed, i.e. the capacities are the actual user data capacities available.

### *14.3.7. Filemark Counter Page*

| BYTE | BIT 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 00 | RESERVED | | | Page Code = 32h | | | | |
| 01 | RESERVED | | | | | | | |
| 02<br>03 | Page Length = 0008h | | | | | | | |
| 04<br>05 | Parameter Code = 0000h | | | | | | | |
| 06 | DU | DS | TSD | ET | TMC | | R | LP |
| 07 | Parameter Length = 04h | | | | | | | |
| 08<br>09<br>10<br>11 | Filemark Counter | | | | | | | |

*Table: Filemark Counter Page*

**Page Code** This field is always set to 32h.

**Page Length** This field is always set to 8 indicating that one list with 8 bytes follows.

**Parameter Code** The Parameter Code will be set to 0000h.

**DU** The Disable Update (DU) bit will normally be set to zero to indicate that the Drive is always free to update the counter. When the counter reaches its maximum value (FFFFh) the DU bit will be set to one.

**DS** The Disable Save (DS) bit will be set to one to indicate that the Drive does not support saving of this log parameter.

**TSD** The Target Save Disable (TSD) bit will be set to zero.

**ET** The Enable Threshold (ET) bit will be set to zero to indicate that no threshold comparison will take place.

**TMC** The Threshold Met Criteria (TMC) field will be set to zero.

**LP** The List Parameters (LP) bit will be set to zero to indicate that the parameter is a data counter.

**Parameter Length** This field will be set to 4 to indicate that the counter is 32 bits wide.

**Filemark Counter** This counter counts filemarks received from the Initiator (during write operations) or sent to the Initiator (during read operations). After the Drive has been reset the counter contains a value of zero. Upon reaching the maximum value the counter does not wrap back to zero, but retains its maximum value (FFFFFFFFh).

### 14.3.8. Physical Position Page

| BYTE | BIT 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 00 | Page Code = 36h | | | | | | | |
| 01 | RESERVED | | | | | | | |
| 02<br>03 | Page Length=16h | | | | | | | |
| 04<br>05 | Parameter Code (1) | | | | | | | |
| 06 | DU | DS | TSD | ET | TMC | | R | LP |
| 07 | Parameter Length (2) | | | | | | | |
| 08<br>09 | Track Number | | | | | | | |
| 10<br>11 | Parameter Code (2) | | | | | | | |
| 12 | DU | DS | TSD | ET | TMC | | R | LP |
| 13 | Parameter Length (4) | | | | | | | |
| 14-15<br>16-17 | Tape Position Count | | | | | | | |
| 18<br>19 | Parameter Code (3) | | | | | | | |
| 20 | DU | DS | TSD | ET | TMC | | R | LP |
| 21 | Parameter Length (4) | | | | | | | |
| 22-23<br>24-25 | Track Length | | | | | | | |

*Table: Physical Position Page*

**Page Code** The Page Code for this page is always set to 36h.

**Page Length** This field is always set to 16h.

**Parameter Code** Identifies the parameters within the page. Values used for the four parameters are 1, 2 and 3.

**DU** The Disable Update (DU) bits are set to zero for all the parameters to indicate that the parameters are updated by the Drive.

**DS** The Disable Save (DS) bits are set to one for all parameters to indicate that the Drive does not support saving of parameters.

**TSD** The Target Save Disable (TSD) bits are set to zero for all parameters. This field is only significant if the DS-bit is cleared.

**ET** The Enable Threshold (ET) bits are set to zero for all parameters to indicate that no threshold comparison will take place.

**TMC** The Threshold Met Criteria (TMC) bits are set to zero for all parameters. This field is only valid if the ETC-bit is set.

**LP**

The List Parameters (LP) bits are set to zero for all parameters to indicate that the parameter is a counter and not a list parameter (ASCII string).

**Parameter length**

These fields are always set to 2 for the track number and to 4 for the position values to indicate that the parameters are 16 and 32 bits wide respectively.

**Track Number**

This parameter echoes the current track number of the Drive.

**Position Count**

This list parameter echoes a drive internal, 4-byte signed counter. This counter starts at 0000 at the Beginning Of Track (BOT) and is incremented for every 4.22 mm the tape travels in the forward direction. It is decremented for every 4.22 mm the tape travels in the reverse direction. As several mechanical tolerances are involved, like slippage between tape and cartridge drive system, slippage between drive motor wheel and cartridge drive wheel, compression of drive capstan wheel etc., this is not an exact measure. However, it gives an indication of approximate, physical position relative to BOT. Please note that after a REWIND operation, this number is negative, as the tape is positioned in front of BOT.

**Track Length**

This 4-byte list parameter value is the number of 4.22 mm increments counted by the Drive between BOT and EOT. It is updated each time the Drive is at EOT. Before the first time a new cartridge has been at EOT, its value is set to an approximate value according to the inserted type of tape.

NOTE:
The Physical Position parameters are not meant to be used as a means of monitoring the physical position so as to check position data on the tape.

## 14.4. Exception Handling

See section on Error Conditions For All Commands.

If the PPC, SP, PC, Page Code or Parameter Pointers fields do not have legal values, the Drive will return CHECK CONDITION status. No parameter data will be sent. The Error Code will be set to E$STE_IFIC.

*This Page Intentionally Left Blank*

# 15. Mode Select

## 15.1. Command Description

The MODE SELECT command provides a means for the Initiator to specify a number of device parameters in the Drive. New parameter values are included in the MODE SELECT Parameter List. As an option the MODE SELECT command may save the selected parameter values. Saved values will be used as default values at the next power-up or reset.

The Drive will implement only one common set of parameters for all Initiators. If any parameters that affect another Initiator are changed, the Drive will generate a Unit Attention condition (with an E$STE_-MPCH Error Code) for all Initiators except the one that issued the MODE SELECT command.

The MODE SELECT Parameter List will be transferred during the DATA-OUT phase of the command.

If disconnection is allowed, the Drive may disconnect when executing this command.

***SLR5 4.0/8.0GB and SLR4 2.5/5.0GB (TDC 4222) only***

Unlike for other Mode parameters, the Drive may change the current data compression parameters dynamically during normal operation. Both the compression and the decompression settings will reflect the present operating settings as described below:

- The Drive will allow compression to be turned ON when writing and reading on the QIC-2GB or QIC-4GB tape formats only.
- The Drive will not allow any change to the Data Compression Page when the tape is positioned away from BOT. When appending or reading data it is NOT legal to select data compression settings different from the settings currently used.
- A MODE SELECT command will only affect the data compression parameters used when issuing a WRITE-type command from BOT.

## 15.2. Command Descriptor Block

| BYTE | BIT 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 00 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 |
| 01 | Logical Unit Number (LUN) | | | PF | RESERVED | | | SP |
| 02 | RESERVED | | | | | | | |
| 03 | RESERVED | | | | | | | |
| 04 | Parameter List Length | | | | | | | |
| 05 | Control Byte | | | | | | | |

*Table: MODE SELECT Command Block*

**PF**

The Page Format (PF) bit determines whether the Drive will accept any Mode Pages in the parameter list sent to the Drive in the DATA-OUT phase of the MODE SELECT command. If the PF-bit is set to zero, the Drive will not accept any Mode Pages in the parameter list; only the Header List and a Block Descriptor List will be accepted. If the Drive receives a parameter list containing bytes beyond the Header List and a Block Descriptor List, it will terminate the MODE SELECT command with CHECK CONDITION and set the Sense Key to ILLEGAL REQUEST and the AS/AQ sense bytes to PARAMETER LIST LENGTH ERROR. The Error Code will be set to E$STE_PLEN. If the PF-bit is set to one, the Drive will accept the Mode Pages as defined in this specification.

**SP**

A Save Page (SP) bit of zero indicates that the Drive will perform the specified MODE SELECT operation, but not save any mode parameters. A SP bit of one indicates that the Drive will perform the specified MODE SELECT operation and also save all saveable MODE SELECT parameters received during the DATA OUT phase.

**Parameter List Length**

This field specifies the length in bytes of the MODE SELECT parameter list that will be transferred from the Initiator to the Drive during the DATA OUT phase. A Parameter List Length of zero indicates that no data will be transferred. No mode selection parameters are then changed. A parameter list length must never result in the truncation of any header, descriptor or page of parameters.

## 15.3. Parameter List

The MODE SELECT parameter list consists of three sub-lists. The first list is a 4 byte Header List. This may be followed by a 8 byte Block Descriptor List. At last there may be from one to several Page Descriptor Lists. The Page Descriptor Lists may be transferred in any order.

The Drive supports the following Page Descriptor Lists:

***SLR5 4.0/8.0GB and SLR4 2.5/5.0GB (TDC 4222) only***

| | | |
|---|---|---|
| **01h** | : | Error Recovery Page |
| **02h** | : | Disconnect/Reconnect Page |
| **0Fh** | : | Data Compression Parameter Page |
| **10h** | : | Device Configuration Parameter Page |
| **11h** | : | Medium Partition Parameter Page |
| **20h** | : | Miscellaneous Parameter Page |
| **21h** | : | User Page 0 |
| **22h** | : | User Page 1 |

### *15.3.1. Header List*

| BYTE | BIT 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 00 | RESERVED | | | | | | | |
| 01 | RESERVED | | | | | | | |
| 02 | R | Buffered Mode | | | Tape Speed/Reserved for TDC 3820 | | | |
| 03 | Block Descriptor Length | | | | | | | |

*Table: MODE SELECT Header List*

**Buffered Mode**

The Drive supports the following mode: 0 and 1. Mode 0 indicates that the WRITE and COPY (copy function = backup) and WRITE FILEMARKS command will report GOOD status when the requested data blocks have been actually written and verified. Mode 1 indicates that the WRITE (and COPY, WRITE FILEMARKS) command will report GOOD status as soon as the requested data has been transferred into the Drive's data buffer.

- Legal values are 0 and 1

- The default (factory programmed) value is 1

**Tape Speed**

Specifies the current tape speed. The following values are legal:

**0h:** No change.

**1h:** Low Speed (53 ips when the Drive is in QIC-1000 mode).

**2h:** High Speed (80 ips when the Drive is in QIC-1000 mode). For SLR5 Series Drives, High Speed is 107 ips when the Drive is in QIC-1000 mode.

**Fh:** Set default value (Mode Sense value 0).

See Chapter 16. Mode Sense for further description.

- Legal values are numbers in the range 0..2 and Fh
- The default (factory programmed) value is 0. (Default values and 53 ips for QIC-1000)

***SLR5 Series only*** Default speed field for QIC-1000 is 107 ips.

The configured tape speed for QIC-1000 can be changed at any time. The tape speed change will, however, first take place when starting to Read or Write from BOT in the data partition.

***SLR2 525MB (TDC 3820 MK2)***

The Tape Speed field is reserved and must be set to 0.

**Block Descriptor Length**

This field specifies the length in bytes of the block descriptor list. Legal values are either 0 or 8. A value of zero means that no block descriptor list is included in the Parameter List.

### 15.3.2. Block Descriptor List

| BYTE | BIT 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 00 | Density Code | | | | | | | |
| 01<br>02<br>03 | Number of Blocks | | | | | | | |
| 04 | RESERVED | | | | | | | |
| 05<br>06<br>07 | Block Size | | | | | | | |

*Table: MODE SELECT Block Descriptor List*

**Density Code**

This field indicates the tape format to use when a write operation is started. Legal values are:

| SLR5 Series | TDC 4200 and SLR4 2.5/5.0GB (TDC 4222) | TDC 4100 | SLR2 525MB (TDC 3820 MK2) | Description |
|---|---|---|---|---|
| 00h | 00h | 00h | 00h | Default Density |
| 0Fh | 0Fh | 0Fh | 0Fh | QIC-120 |
| 10h | 10h | 10h | 10h | QIC-150 |
| 11h | 11h | 11h | 11h | QIC-525 |
| 15h | 15h | 15h | | QIC-1000 |
| 22h | 22h | | | QIC-2GB |
| 26h | | | | QIC-4GB |

- The default (factory programmed) value is 00h (Default Density)

When the Density Code field is set to zero this means that the Drive will use the ***last*** saved density code (the density code stored in the Drive's EEPROM as a result of a MODE SELECT command with the Save option enabled). Note that when the EEPROM value is 00h (default) and the MODE SELECT value is also 00h (default) ...

***TDC 4100 Series*** ... the Drive will select QIC-1000 tape format.

***TDC 4200 Series*** ... the Drive will select QIC-2GB tape format.

***SLR2 525MB (TDC 3820 MK2) only*** ... the Drive will select QIC-525 tape format.

***SLR5 Series*** ... the Drive will select QIC-4GB tape format.

Default density code will be interpreted as 'no change' if issued away from BOT. The density code that already existed on the tape (on-tape density) will be used. If the ***Save*** option is ON and the density code is 00h, the saved density code will remain unchanged.

When the Enable Automatic Density Code Selection EADS-field in the Miscellaneous Parameter Page is set to one, this means that the Drive will automatically select a suitable density code when a WRITE operation is started.

When automatic density code selection is enabled, the Drive will always try to use the tape format giving the largest capacity on a given cartridge. The chosen density code is reported back to the Density Code field of the Block Descriptor List of the MODE SENSE command after any READ or WRITE operation. When writing from BOM the following table shows the selected density (tape format) for the various types of media (cartridge types):

| Medium | Tape Format |
|---|---|
| DC300<br>DC300XLP | The cartridge requires QIC-24 tape format.<br>The Drive is not able to write this format and issues an E$BTD_CFMT error message when an attempt is made to write on this cartridge. |
| DC615<br>DC600A | The Drive selects QIC-120 tape format (density code = 0Fh) |
| DC6037<br>DC6150<br>DC6250 | The Drive selects QIC-150 tape format (density code = 10h) |
| DC6320<br>DC6525 | The Drive selects QIC-525 tape format (density code = 11h) |
| DC9100<br>DC9120<br>DC9120XL<br>DC9100SL<br>DC9100FW | The Drive selects QIC-1000 tape format (density code = 15h)<br>SLR2 (TDC 3820) is NOT able to write or read these formats and issues an E$BTD_CFMT error message when an attempt is made to write on these cartridges. |
| DC9200<br>DC9200SL<br>DC9250 | The TDC 4200 Series Drives select QIC-2GB tape format (density code = 22h)<br>SLR2 (TDC 3820) and TDC 4100 Drives are NOT able to write or read this format and issues an E$BTD_CFMT error message when an attempt is made to write or read on these cartridges. |
| DC9400 | The SLR5 Series Drives select QIC-4GB tape format.<br>SLR2 (TDC 3820), TDC 4100 and SLR4 2.5/5.0GB (TDC 4222) Drives are NOT able to write or read this format and issues an E$BTD_CFMT error message when an attempt is made to write or read on these cartridges. |

*Table: Type of Media Related to Tape Format*

***SLR5 Series only*** These drives can neither read nor write the QIC-24 tape format.

NOTE:
It is not possible to select QIC-120 and QIC-150 with ECC. The ECC on the QIC-525/1000/2GB or QIC-4GB tape formats are controlled by the Disable Correction (DCR) bit.
When appending data in the automatic density select mode, the Drive will append data with the tape format found on the inserted tape cartridge.

The Drive will not allow any change of Density Code when the tape is positioned away from beginning-of-tape (BOT). When writing it is not legal to select a tape format different from the tape format that is currently used. When reading it is not legal to select a tape format different from the one that the Drive has detected on the currently inserted tape. A MODE SELECT command changing the Density Code is legal when executed after an ERASE, LOAD or REWIND command (with no intervening read, space, verify or write operations). A MODE SELECT command changing the Density Code is also legal when no read, space, verify or write operations has been executed on a newly inserted cartridge.

Please note that not all tape formats are legal in all situations. When appending data to a tape, the only format allowed is the format that where used to write the data already present on the cartridge.

Note that some tape formats are not allowed on some media (cartridge) types. A WRITE (or WRITE FILEMARK) command will terminate with CHECK CONDITION status if the configured tape format has been set to some inappropriate value. See the WRITE or WRITE FILEMARKS sections for details.

Note also that not all combinations of tape formats and Block Sizes are allowed (see the description of the Block Size field).

**Number Of Blocks**

This field MUST be set to zero to indicate that the whole tape has the same density code and block length.

**Block Size**

This field selects the block size used (on the SCSI-bus) when reading, recovering, verifying and writing fixed length blocks. There are three legal values:

NOTE:
When the Auto Read (ARD) bit is set to one, the Block Size can only be changed when the tape is (logically) positioned at BOT.

| Block Size | Description |
|---|---|
| 000000h | The Drive is set into Variable Block mode. In this mode it is illegal to issue commands that have the FIX bit set to one. Only variable block READ, RECOVER BUFFERED DATA, VERIFY or WRITE commands are allowed. When writing variable blocks in the QIC-525/1000/2GB/96-34 tape format the Drive will write all blocks as Variable Data Blocks according to the QIC-525/1000/2GB/96-34 - QIC Development Standards [4] and [5]. When the tape format is QIC-120 or QIC-150, the Drive will add special Control Blocks to record the length of each individual logical block. See also the section on the READ command for further details on the difference between Variable Block mode and Fixed Block mode. |
| 000200h | The Drive is set into Fixed Block mode. The block size used when transferring fixed length blocks on the SCSI-bus is 512 bytes. When writing 512 byte fixed length blocks in the QIC-525/1000/2GB/96-34 tape format the Drive will pack two 512 byte logical blocks into one physical tape block according to the QIC-02 Compatible Full Data Blocks of the QIC-525/1000/2GB/96-34 - QIC Development Standards [4] and [5]. When the tape format is QIC-120 or QIC-150, the Drive will write normal 512 byte physical tape blocks. |
| 000400h | The Drive is set into Fixed Block mode. The block size used when transferring fixed length blocks on the SCSI bus is 1024 bytes. When writing 1024 byte fixed length blocks in the QIC-525/1000/2GB/96-34 tape format, the Drive will write them as full Data Blocks according to the QIC-525/1000/2GB/96-34 QIC - Development Standards [4] and [5]. The block size is not legal for QIC-120/150. |

*Table: Fixed Block Sizes*

### 15.3.3. Error Recovery Page

This page is used to specify error recovery and reporting parameters.

The page can be saved.

| BYTE | BIT 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 00 | PS | R | Page Code = 01h | | | | | |
| 01 | Parameter Length = 0Ah | | | | | | | |
| 02 | RESERVED | | TB | R | EER | PER | DTE | DCR |
| 03 | Read Retry Count | | | | | | | |
| 04 | RESERVED | | | | | | | |
| 05 | RESERVED | | | | | | | |
| 06 | RESERVED | | | | | | | |
| 07 | RESERVED | | | | | | | |
| 08 | Write Retry Count | | | | | | | |
| 09 | RESERVED | | | | | | | |
| 10 | RESERVED | | | | | | | |
| 11 | RESERVED | | | | | | | |

*Table: Error Recovery Page Descriptor*

**PS** The Parameter Saveable (PS) bit MUST be set to zero.

**Page Code** The Page Code for this page will be set to 01h.

**Parameter Length** The Parameter Length field MUST always be set to 0Ah.

**Transfer Block** The Transfer Block (TB) bit MUST be set to 1 to indicate that the failing data block (unrecoverable) will always be transferred to the Initiator (it is not changeable).

**EER** The enable early recovery (EER) bit MUST be set to 1 to indicate that the Drive will always enable the use of ECC **before** applying retries (it is not changeable).

**PER**

When the Post Error Recovery (PER) bit is set to one the Drive will report CHECK CONDITION status and an Error Code set to E$STE_RECV on any command (except REQUEST SENSE) if there has been any re-reads or re-writes since the last command and no other errors has occurred. A PER bit if zero indicates that the Drive will not flag any re- reads or re-writes. Note that all commands will be fully executed before a CHECK CONDITION is reported due to re-reads or re-writes. Note also that when operating in buffered mode a re-write might occur on data that was written to the tape after the last write command terminated with GOOD status. In this case the next command issued will report CHECK CONDITION status after execution.

- Legal values are 0 and 1.
- The default (factory programmed) value is 0.

**DTE**

The disable transfer on error (DTE) bit MUST be set to zero (it is not changeable).

**DCR**

A Disable Correction (DCR) bit of one indicates that the Drive will disable the ECC mechanism. Even if a block is correctable, the Drive will not correct it, but instead use the rereads to recover the block. A DCR bit of zero indicates that the Drive will use ECC correction whenever possible.

- Legal values are 0 and 1.
- The default (factory programmed) value is 0.

**Read Retry Count**

The read retry count specifies the number of times that the Drive should attempt to re-read a block. This field is changeable.

- Legal values are numbers in the range 1..24
- The default (factory programmed) value is 24.

**Write Retry Count**

The write retry count specifies the number of times the Drive should attempt to re-write a block. This field is changeable.

- Legal values are numbers in the range 1..16
- The default (factory programmed) value is 16.

### *15.3.4. Disconnect/Reconnect Page*

This page is used to specify the Drive's disconnect and reconnect parameters.

The page can be saved.

| BYTE | BIT 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 00 | PS | R | Page Code = 02h | | | | | |
| 01 | Parameter Length = 0Eh | | | | | | | |
| 02 | Read Buffer Full Ratio | | | | | | | |
| 03 | Write Buffer Empty Ratio | | | | | | | |
| 04 | Bus Inactivity Time | | | | | | | |
| 05 | | | | | | | | |
| 06 | Disconnect Time Limit | | | | | | | |
| 07 | | | | | | | | |
| 08 | Connect Time Limit | | | | | | | |
| 09 | | | | | | | | |
| 10 | Maximum Burst Size | | | | | | | |
| 11 | | | | | | | | |
| 12 | RESERVED | | | | | | | DTDC |
| 13 | RESERVED | | | | | | | |
| 14 | RESERVED | | | | | | | |
| 15 | RESERVED | | | | | | | |

*Table: Disconnect/Reconnect Page Descriptor*

**PS** The Parameter Saveable (PS) bit MUST be set to zero.

**Page Code** The Page Code for this page will be set to 02h.

**Parameter Length** The Parameter Length field MUST always be set to 0Eh.

**Read Buffer Full Ratio**

The read buffer full ratio indicates to the Drive, on READ commands, how full the buffer will be prior to reconnecting. The read buffer full ratio is a number in the range 0..63. A value of 0 indicates that only one block ready in the buffer is enough to force a reconnect. A value of 63 indicates that the buffer must be one quarter full before a reconnect occurs. This field is changeable.

The actual amount of data (in number of bytes) needed for a reconnect can be calculated using the following formula:

| | |
|---|---|
| **Nb = 512** | **for Rbfr equal to 0** |
| **Nb = (Rbfr * 1024)** | **for Rbfr *not* equal to 0** |

(Where Nb = Number of bytes and Rbfr = Read Buffer Full Ratio)

Note that the Read Buffer Full Ratio also controls the maximum burst size. When transferring data during read type operations, the Drive will never transfer more data than given by the Read Buffer Full Ratio before disconnecting. This is done to ease parity error recovery. The last disconnect (with its SAVE DATA POINTERS message) will always serve as a reference point from where to start a retransfer in case of an Initiator detected parity error. See also the section on Data Buffering for further information on this parameter.

- Legal values are numbers in the range 0..63.

- The default (factory programmed) value is 16.

***SLR5 4.0/8.0GB and SLR4 2.5/5.0GB (TDC 4222) only***

For the SLR5 4.0/8.0GB and SLR4 2.5/5.0GB (TDC 4222), the amount of data will always be limited to a maximum of 32 KBytes.

**Write Buffer Empty Ratio**

The write buffer empty ratio indicates to the Drive, on VERIFY and WRITE commands, how empty the buffer will be prior to reconnecting to fetch more data. The write buffer empty ratio is a number in the range 0..63. A value of 0 indicates that if there is room for just one block in the buffer then the Drive will reconnect. A value of 63 indicates that the buffer must be one quarter empty before a reconnect occurs. This field is changeable.

The actual amount of data (in number of bytes) needed for a reconnect can be calculated using the following formula:

| | |
|---|---|
| **Nb = 512** | **for Wber equal to 0** |
| **Nb = (Wber * 1024)** | **for Wber *not* equal to 0** |

(Where Nb = Number of bytes and Wber = Write Buffer Empty Ratio)

Note that the Write Buffer Empty Ratio also controls the maximum burst size. When transferring data during write type operations, the Drive will never transfer more data than given by the Write Buffer Empty Ratio before disconnecting. This is done to ease parity error recovery. The last disconnect (with its SAVE DATA POINTERS message) will always serve as a reference point from where to start a retransfer in case of parity error detected by the Drive. See also the section on Data Buffering for further information on this parameter.

- Legal values are numbers in the range 0..63.
- The default (factory programmed) value is 16.

***SLR5 4.0/8.0GB and SLR4 2.5/5.0GB (TDC 4222) only***

For the SLR5 4.0/8.0GB and SLR4 2.5/5.0GB (TDC 4222), the amount of data will always be limited to a maximum of 32 KBytes.

**Bus Inactivity Time**

This field MUST be set to zero (it is not changeable).

**Disconnect Time Limit**

This field MUST be set to zero to indicate that the Drive is allowed to reconnect immediately (it is not changeable).

**Connect Time Limit**

This field MUST be set to zero (it is not changeable).

**Maximum Burst Size**

The maximum burst size field MUST be set to zero to indicate no limit on the amount of data transferred. This field is not changeable.

**DTDC**

The Data Transfer Disconnect Control field MUST be set to 00h to indicate that data transfer disconnect control is not used.

### 15.3.5. SLR5 4.0/8.0GB and SLR4 2.5/5.0GB (TDC 4222) Only - Data Compression Parameters Page

This page is only supported for the SLR5 4.0/8.0GB and SLR4 2.5/5.0GB (TDC 4222) Drives and is used to specify the data compression configuration. The data compression can only be changed when the tape is positioned at BOT.
See Section 15.3.8. regarding the IDSOT-bit (Ignore Data compression Settings when On Tape).

The page can be saved.

| BYTE | BIT 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 00 | PS | R | Page Code = 0Fh | | | | | |
| 01 | Parameter Length = 0Eh | | | | | | | |
| 02 | DCE | DCC | RESERVED | | | | | |
| 03 | DDE | RED | | RESERVED | | | | |
| 04<br>05<br>06<br>07 | (MSB)<br>Compression Algorithm<br>(LSB) | | | | | | | |
| 08<br>09<br>10<br>11 | (MSB)<br>Decompression Algorithm<br>(LSB) | | | | | | | |
| 12 | RESERVED | | | | | | | |
| 13 | RESERVED | | | | | | | |
| 14 | RESERVED | | | | | | | |
| 15 | RESERVED | | | | | | | |

*Table: Data Compression Parameters Page Descriptor*

**PS**

The Parameter Saveable (PS) bit MUST be set to zero.

**Page Code**

This field is always set to 0Fh.

**Parameter Length**

This field will be set to 0Eh.

**DCE**

When the Data Compression Enable (DCE) bit is set, data sent to the Drive by the Initiator will be processed using the selected compression algorithm before being written to the tape (refer to the Compression Algorithm field). A DCE-bit of zero indicates that data compression is disabled. The DCE-bit can only be set to one if the DCC-bit indicates that the Drive supports data compression.

- Legal values are 0 and 1

- The default (factory programmed) value is 1

**DCC**

A Data Compression Capable (DCC) bit of one indicates that the Drive supports data compression. A DCC-bit of zero indicates that the Drive does not support data compression. This bit is not changeable.

**DDE**

When the Data Decompression Enable (DDE) bit is set to one, data on the tape will be processed using the selected decompression algorithm before being transferred to the Initiator (refer to the Decompression Algorithm field). A DDE-bit of zero indicates that data decompression is disabled. The DDE-bit can only be set to one if the DCC-bit indicates that the Drive supports data decompression.

MODE SENSE: The DDE-bit will always be equal to the DCE-bit.
MODE SELECT: The DDE-bit will be ignored.

- Legal values are 0 and 1
- The default (factory programmed) value is 1

**RED**

The Report Exception on Decompression (RED) field indicates the Drive's response to certain boundaries detected in the data on the tape. Possible boundaries are:

1) *transitions between compressed and uncompressed data*
2) *transitions between uncompressed and compressed data*

or

3) *changes in compression algorithm*

Since none of these boundaries are legal, the RED-field will be set to 2 (two) indicating that the Drive will return with CHECK CONDITION status when a boundary is found.

- This field must be set to 2

**Compression Algorithm**

This field specifies the data compression algorithm used by the Drive to process data sent to it by the Initiator when the DCE-bit is set to one. The following values are legal:

| | |
|---|---|
| **00 00 00 00h** | : No algorithm selected (identifies uncompressed data despite that the DCE-bit is set to one) |
| **00 00 00 03h** | : ALDC (Adaptive Lossless Data Compression) algorithm |

- The default (factory programmed) value is 00 00 00 03h

**Decompression Algorithm**

This field specifies the decompression algorithm used by the Drive when reading data from the tape when the DDE-bit is set to one.

MODE SENSE: This field will always be equal to the Compression Algorithm field.
MODE SELECT: This field will be ignored.

The following values are legal:

| | |
|---|---|
| **00 00 00 00h** | : No algorithm selected (identifies uncompressed data despite that the DDE-bit is set to one) |
| **00 00 00 03h** | : ALDC (Adaptive Lossless Data Compression) algorithm |

- The default (factory programmed) value is 00 00 00 03h

### 15.3.6. Device Configuration Parameters Page

This page is used to specify various Drive configurations.

The page can be saved.

| BYTE | BIT 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 00 | PS | R | Page Code = 10h | | | | | |
| 01 | Parameter Length = 0Eh | | | | | | | |
| 02 | R | CAP | CAF | Active Format | | | | |
| 03 | Active Partition | | | | | | | |
| 04 | Write Buffer Full Ratio | | | | | | | |
| 05 | Read Buffer Empty Ratio | | | | | | | |
| 06<br>07 | Write Delay Time | | | | | | | |
| 08 | DBR | BIS | RSMK | AVC | SOCF | | RBO | REW |
| 09 | Gap Size | | | | | | | |
| 10 | EOD Defined | | | EEG | SEW | RESERVED | | |
| 11<br>12<br>13 | Buffer Size at Early Warning | | | | | | | |
| 14 | RESERVED | | | | | | | |
| 15 | RESERVED | | | | | | | |

*Table: Device Configuration Page Descriptor*

**PS** The Parameter Saveable (PS) bit MUST be set to zero.

**Page Code** This field is always set to 10h.

**Parameter Length** This field will be set to 0Eh.

**CAP** The Change Active partition (CAP) bit set to one indicates that the logical partition is to be changed to the one specified in the Active Partition field. This is only allowed if the FDP bit in the Medium Partition Parameter Page has been set to one. A CAP bit of zero indicates that no partition change is specified. The CAP bit is not saveable.

**CAF** The change active format bit MUST be set to zero.

**Active Format** This field MUST be set to zero.

**Active Partition**

This field indicates the current partition number in use on the medium when the CAP bit is set to one. This field is ignored when the CAP bit is zero. The Active Partition is not saveable. The Drive will always be set to *Data Partition* (Partition 0) after reset.

- Legal values are 0 and 1.

- The default (factory programmed) value is 0.

**Write Buffer Full Ratio**

The Write Buffer Full Ratio field, on WRITE commands, indicates to the Drive how full the buffer will be before writing data to the medium. The Write Buffer Full Ratio is a number in the range 0..255 where 0 indicates that the Drive will start writing as soon as there are any data in the buffer. A value of 255 indicates that the buffer must be one quarter full before the Drive starts writing.
(Max. 68 physical data blocks regardless of the physical block size).

The actual number of physical tape blocks (1024 or 512 bytes long) needed for a write operation to start can be calculated with the following formula:

**Nb = (Wbfr * 67)/ 255 + 1**

(Where Nb = Number of blocks and Wbfr = Write Buffer Full Ratio)

- Legal values are numbers in the range 0..255.

- The default (factory programmed) value is 20.

**Read Buffer Empty Ratio**

The Read Buffer Empty Ratio field, on READ commands, indicates to the Drive how empty the buffer will be before additional data is read from the medium. The Read Buffer Empty Ratio is a number in the range 0..255 where 0 indicates that the Drive will start reading as soon as there are any free space in the buffer. A value of 255 indicates that the buffer must completely empty before the Drive starts reading.

The actual number of physical tape blocks (1024 or 512 bytes long) needed for a read operation to start can be calculated with the following formula:

**Nb = (Rber * 67)/ 255 + 1**

(Where Nb = Number of blocks and Rber = Read Buffer Empty Ratio)

- Legal values are numbers in the range 0..255.

- The default (factory programmed) value is 20.

**Write Delay Time**

The Write Delay Time field indicates the maximum time, in 100 ms increments, that the Drive must wait before any buffered data to be written is forced out to the tape after the last buffered WRITE (or WRITE FILEMARKS) command that did not make up a complete set of ECC-frames or caused the buffer to exceed the Write Buffer Full Ratio. A value of zero indicates that the Drive will never force buffered data to the tape under these conditions.

- Legal values are any number in the range 0..65535.
- The default (factory programmed) value is 0, causing the buffer time-out feature to be turned off.

**DBR**

This bit is set to one to indicate that the Drive supports the RECOVER BUFFERED DATA command (it is not changeable).

**BIS**

This bit MUST be set to one to indicate that the Drive will support Block Identifiers.

**RSMK**

This bit must be set to one to indicate that the Drive reports setmarks.

**AVC**

This bit will always be set to zero for QIC devices (including the Drive).

**SOCF**

This field will always be set to 00h for QIC devices (including the Drive).

**RBO**

This bit will always be set to zero for QIC devices (including the Drive).

**REW**

The Report Early Warning (REW) bit MUST be set to zero. This indicates that the Drive will report Early Warning (PSEW) at a vendor-specific point on WRITE operations only.

**Gap Size**

This field will always be set to 00h for QIC devices (including the Drive).

**EOD Defined**

This field will always be set to 01h for QIC devices (including the Drive). This indicates that End-Of-Data (Logical End Of Partition) is defined by a tape format defined erased area of the tape.

**EEG**

This field will always be set to 1 for QIC devices (including the Drive).

**SEW**

This field will always be set to 1 for QIC devices (including the Drive).

**Buffer Size At Early Warning**

This field will always be set to 00h for QIC devices (including the Drive).

### 15.3.7. Medium Partition Parameters Page

This page is used to specify medium partitions needed when using QFA.

The page can be saved.

| BYTE | BIT 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 00 | PS | R | Page Code = 11h | | | | | |
| 01 | Parameter Length = 06h | | | | | | | |
| 02 | Maximum Additional Partitions | | | | | | | |
| 03 | Additional Partitions Defined | | | | | | | |
| 04 | FDP | SDP | IDP | PSUM | | | RESERVED | |
| 05 | Medium Format Recognition | | | | | | | |
| 06 | RESERVED | | | | | | | |
| 07 | RESERVED | | | | | | | |

*Table: Medium Partitions Parameter Page*

**PS**
The Parameter Saveable (PS) bit MUST be set to zero.

**Page Code**
This field will be set to 11h.

**Parameter Length**
This field will be set to 6.

**Maximum Additional Partitions**
This field is not changeable and is returned on a MODE SENSE. It indicates the maximum number of additional partitions supported by the Drive. It will always be set to 1 (indicating a total of two partitions).

**Additional Partitions Defined**
This field MUST be set to zero.

**FDP**
The Fixed Data Partitions (FDP) bit, when set to one indicates that the Drive will assign one additional fixed partition. When used for QFA, the first partition (partition 0) is the data partition, the second partition (1) is the directory partition. A FDB bit of zero indicates that the Drive will not operate with any additional partitions. Changing the value of this bit is only valid when the tape is positioned at BOT.

- Legal values are 0 and 1.
- The default (factory programmed) value is 0 (only one partition defined).

**SDP**
This bit MUST be set to zero.

**IDP**
This bit MUST be set to zero.

**PSUM**
This field MUST be set to zero.

**Medium Format Recognition**
This field is not changeable and is returned on MODE SENSE. The Drive will always return 00h. This indicates that the Drive is not able to independently recognize the format on the medium.

### 15.3.8. Miscellaneous Parameters Page

This page is used to change various drive configurations and to specify special drive functionality.

The page can be saved.

| BYTE | BIT 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 00 | PS | R | Page Code = 20h | | | | | |
| 01 | Parameter Length = 0Ah | | | | | | | |
| 02 | Forced Streaming Count | | | | | | | |
| 03 | | | | | | | | |
| 04 | ASI | Target Sense Length | | | | | | |
| 05 | Copy Threshold | | | | | | | |
| 06 | Load Function | | | | | | | |
| 07 | Power-Up Auto Load/Retension Delay | | | | | | | |
| 08 | DTM1 | DTM2 | SPEW | EOWR | EADS | BSY | RD | FAST |
| 09 | LED Function | | | | RESERVED | | | |
| 10 | PSEW Position | | | | | | | |
| 11 | ARD | | SLR4 2.5/5.0GB (TDC 4222): IDSOT<br>Others: RESERVED | RESERVED | | | | |

*Table: Miscellaneous Page Descriptor*

**PS**

The Parameter Saveable (PS) bit MUST be set to zero.

**Page Code**

The Page Code for this page will be set to 20h.

**Parameter Length**

The Parameter Length field MUST always be set to 0Ah.

**Forced Streaming Count**

Normally, when the data buffer becomes empty during write operation, the Drive will enter the underrun mode and stop the tape motion. When new data is entered (from host), the tape is restarted and streaming continues for as long as data is available. This applies when the Forced Streaming count is set to zero.
By setting the Forced Streaming Count to a value different from zero, the Drive will rewrite the last block before entering the underrun mode (and stopping the tape). The number of rewrites is specified by the Forced Streaming Count field.

- Legal values are numbers in the range 0..65535. However, a higher value than 767 will internally be replaced by 767.

- The default (factory programmed) value is 0.

**ASI**

The Auto Sense Inhibit controls the action taken by the Drive during Copy when CHECK CONDITION has been signalled by the direct access device. If the ASI-bit is set to zero, the Drive will automatically issue a REQUEST SENSE command when the direct access device terminates a command with a CHECK CONDITION status. If the ASI-bit is set to one, the Drive will not issue any REQUEST SENSE command.

- Legal values are 0 and 1

- The default (factory programmed) value is 0.

**Target Sense Length**

This field controls the Transfer Length byte in the REQUEST SENSE command sent to the direct access device when Auto Sense is enabled (see also the ASI-bit). The minimum value is 0. The maximum value is 24 (there is only a 24-byte buffer internally in the Drive for Target sense data). If the direct access device transfers more than 24 bytes, the additional bytes will wrap around and start writing over the first bytes received.

- Legal values are numbers in the range 0..24.

- The default (factory programmed) value is 24.

**Copy Threshold**

During execution of a COPY command, the Drive issues a sequence of READ or WRITE commands to the assigned Target device. The Copy Threshold controls the number of (Target) blocks requested for transfer in each of these READ/WRITE commands. If the total number of blocks to transfer (given by the current COPY Segment Descriptor) exceeds the current Copy Threshold, the Drive will issue as many READ/WRITE commands as necessary to transfer the total number of blocks. See also the COPY Section.

- Legal values are numbers in the range 1..127.

- The default (factory programmed) value is 8.

**Load Function**

This field controls Auto Load and Auto Retension.

| | | |
|---|---|---|
| **00** | **:** | **Auto Load** |
| **01** | **:** | **Auto Retension** |
| **02** | **:** | **no Auto Load, no Auto Retension** |

If Auto Load is enabled the Drive will automatically perform an operation equivalent to a LOAD/UNLOAD command with the Load bit set to one every time a new cartridge is inserted into the Drive.
Note that the Drive will always perform a physical load operation (moving the tape to BOT). When the Load Function is set to 02 (no Auto Load), the physical load is still performed. The media access commands will, however, terminate with CHECK CONDITION until a LOAD/-UNLOAD command (with the Load bit set) has been executed.

If Auto Retension is enabled the Drive will automatically perform an operation equivalent to a LOAD/UNLOAD command with the Load and RET bits set to one every time a new cartridge is inserted into the Drive.

Note that while the Auto Load or Auto Retension executes, the Drive will be ready for any commands.

- Legal values are numbers in the range 0..2.

- The default (factory programmed) value is 0.

**Power-Up Auto Load/Retension Delay**

This field specifies the delay that will be applied before an Auto Load or an Auto Retension is started after Power-Up; given in increments of 100 ms.

- Legal values are numbers in the range 0..255

- The default (factory programmed) value is 0.

**DTM1**

When set to one, this bit (Disable Tape Map) disables the writing of control blocks containing information used by the fast space algorithm. This bit is valid for QIC-120/150 mode only.

Setting this bit will also disable search for EOR tape map.
FAST operations are, however, still possible as long as the Drive has sufficient information about the tape.

- Legal values are numbers in the range 0..1

- The default (factory programmed) value is 1.

**DTM2**

When set to one, this bit (Disable Tape Map) disables the writing of filler block frames containing information used by the fast space algorithm. This bit is valid for QIC-525/1000/2GB/96-34 mode.

Setting this bit will also disable search for EOR tape map.
FAST operations are, however, still possible as long as the Drive has sufficient information about the tape.

- Legal values are numbers in the range 0..1

- The default (factory programmed) value is 0.

**SPEW**

When set to *one* the Space Pseudo Early Warning (SPEW) bit indicates that all SPACE commands must be terminated with CHECK CONDITION status if the SPACE operation ended up *behind* the PSEW tape-marker, even if the SPACE operation was successful (no errors signalled).
If the SPACE operation, in addition, ended up at End Of Data (EOD), the sense data will have a Sense Key of 8h and the EOM bit will be set to one. The Additional Sense Code will be set to 00h. The Additional Sense Code Qualifier will be set to 05h.
If the SPACE operation did not reach EOD, the sense data will have a Sense Key of 0h and the EOM bit will be set to one. The Additional Sense Code will be set to 00h. The Additional Sense Code Qualifier will also be set to 00h. This will be true for all SPACE codes (blocks, filemarks, sequential filemarks, setmarks and EOD).
Note, however, that only SPACE commands with EOD code is able to bring the tape position up to EOD with no error signalled.

Further SPACE operations behind PSEW will keep terminating with CHECK CONDITION and the same sense data as long as the SPACE operation is successful. When the SPACE operation is *not* successful, the SPACE command will in any case terminate with CHECK CONDITION and the usual sense data.

When set to *zero* the SPEW bit indicates that SPACE commands must take no special notice of the PSEW tape-marker if the SPACE operation is successful. As long as a SPACE operation is successful the SPACE command will be terminated with GOOD status. When the SPACE operation is *not* successful, the SPACE command will terminate with CHECK CONDITION and the usual sense data.

- Legal values are 0 and 1.

- The default (factory programmed) value is 0.

**EOWR**

When this bit is set the Drive will simulate the TAR (1/2' reel-to-reel) overwrite feature.

The overwrite function can be used to overwrite data after the first data block on the tape or to overwrite the last of two sequential filemarks before EOD.

To overwrite data block(s) on the tape, the following cases must be satisfied:

> *The tape must be positioned after the 1st logical block on the tape. If variable block, the logical block must not be more than 65534 (FFFEh) bytes.*
>
> *There are no filemarks so far on the tape and the next block from the tape is a data block.*

To overwrite a filemark, the following cases must be satisfied:

> *The tape must be positioned at the 2nd of two sequential filemarks right in front of EOD. That means there are no data blocks following the filemarks. In this case overwrite from EOD will be allowed and the filemark cancel block will be written as the first block.*

If none of these conditions are true, the append attempt will be terminated with CHECK CONDITION status and an E$BTD_WRRD (Illegal Append) error message.

Append at EOD is allowed as usual.

- Legal values are 0 and 1.

- The default (factory programmed) value is 0.

**EADS**

When set to one, the EADS (Enable Automatic Density Select) bit indicates that the Drive may use automatic density selection. See the description of the Density Code field (Section 15.3.2.) in the Block Descriptor List of the MODE SELECT command for further details.
When the EADS bit is set to zero, automatic density selection is turned off.

- The default (factory programmed) value is 1.

**BSY**

When set to one, the BSY (Busy) bit indicates that the Drive will respond with BUSY status as long as an Immediate type command is under execution. The BUSY status is returned on every new command until the executing Immediate type command has completed its execution. When the BSY bit is set to zero, new commands will be accepted even if an Immediate type command is under execution. The new command may disconnect while waiting for the Immediate type command to finish.

- Legal values are 0 and 1.
- The default (factory programmed) value is 0.

**RD**

When set to one, the RD (Ready Delayed) bit delays the Unit Attention condition for a newly inserted cartridge until the tape has been physically positioned at BOM. While the load is in progress, ***all*** commands (except REQUEST SENSE and INQUIRY) will be terminated with CHECK CONDITION and a Sense Key set to 02h (Not Ready). The other sense data will be the same as when a media access command is executed after an UNLOAD command. A REQUEST SENSE command will return sense data that shows a Not Ready status. An INQUIRY command is executed as it normally would. The Not Ready condition will prevail until the tape has been positioned at BOT. The next command will then see the Unit Attention condition for the inserted cartridge as normal.

When the RD bit is set to zero, the Unit Attention condition will exist as soon as the new cartridge has been inserted, even if the Drive must take time to position it to BOM.

- Legal values are 0 and 1.
- The default (factory programmed) value is 0.

**FAST**

When set to one, this bit enables the special FAST space mode. When set to zero this bit disables the FAST space mode. See the SPACE commands for details.

- Legal values are 0 and 1.
- The default (factory programmed) value is 0.

**LED Function**

This field controls the LED operation. The following 6 modes are possible:

**Mode 0:** The LED is GREEN except when the Drive is logically unloaded and no Prevent Media Removal command has been issued. The LED is OFF when no cartridge is in place.

**Mode 1:** The LED is GREEN when the Drive is busy operating the capstan or the stepper motor. The Prevent Media Removal will turn the LED to STEADY ON if a cartridge is inserted and the Drive is logically loaded. The LED is OFF when no cartridge is in place.

**Mode 2:** Same as for Mode 0, but the LED is turned OFF when the tape is in position at BOM.

**Mode 3:** Power-On indicator. The LED is always GREEN.

**Mode 4:** The LED is STEADY GREEN when a cartridge is loaded and BLINKING GREEN when the Drive is busy operating the capstan or the stepper motor.

AMBER light indicates an error situation:

| | |
|---|---|
| *Sense Key = 04h:* | Only SCSI-bus reset or Bus Device reset causes the LED function to change. |
| *Sense Key = 03h:* | (AS=52h, AQ=00h) SCSI-bus reset, Bus Device reset or cartridge removal causes the LED function to change. |

Other errors are ignored.

The LED is OFF when no cartridge is inserted.

**Mode 5:** In this mode the LED will be STEADY GREEN when a cartridge is loaded and the Drive is busy, or away from BOM. The LED is *off* when the Drive is idle at BOM or a cartridge is not present or loaded.
When a cartridge is *not* present or *not* loaded, the Prevent/-Allow Media Removal command will turn the LED *on* and *off*. The same will happen if the Drive is idle at BOM.

| | |
|---|---|
| **Mode 6:** | LED is OFF |
| **Mode 7:** | LED is STEADY GREEN |
| **Mode 8:** | LED is STEADY RED |
| **Mode 9:** | LED is STEADY AMBER |
| **Mode 10:** | LED is BLINKING GREEN |
| **Mode 11:** | LED is BLINKING RED |
| **Mode 12:** | LED is BLINKING AMBER |

- Legal values are numbers in the range 0..12

- The default (factory programmed) value is 0.

**PSEW Position**

Position of Pseudo Early Warning tape marker specifies the distance between PSEW (Pseudo Early Warning) and EW (Early Warning). The amount of data can be calculated with the following formula:

**(PSEW Position + 1)/2**

Where PSEW Position is the numeric input and the answer is the number of bytes from PSEW to EW in MBytes. The total range of the position of Pseudo Early Warning is from 0.5 MByte and up to 25 MByte or last track.
Beginning of the last track will be used as position of PSEW if more data than possible on one single track is specified. A value of 255 will always give last track as PSEW Position.

NOTE:
These values are nominal and will vary with mode of operation.

- Legal values are numbers in the range 00..49 and 255.

- The default (factory programmed) value is 0.

**ARD**

When the Auto Read (ARD) bit is set to one the Auto Read function is enabled. The Auto Read function will automatically determine the density code (tape format) and the Block Size when a cartridge is inserted. The found density code and the block size are reported in the parameter list of the first MODE SENSE command issued after cartridge insertion. When the ARD bit is set to zero the Auto Read function is disabled. If the Initiator then needs to know the density code and block size when a new cartridge is inserted, an explicit READ command must be issued before the MODE SENSE command.

Note that the auto read processing is split into two phases. When the new cartridge is inserted a search for the reference track is started immediately. The actual reading of the first block on the tape is, however, not done until a MODE SENSE command has been received. When the auto read has completed (and the MODE SENSE command has returned its parameter list), the logical tape position will be at BOT. The MODE SENSE command will automatically update the Density Code and the Block Size of both the MODE SENSE data list and the MODE SELECT data list. The MODE SELECT Density Code and the Block Size (controlling the tape format and the Block Size used when a write operation is started) is set equal to the MODE SENSE Density Code and the Block Size (the density and Block Size found on the tape). No additional MODE SELECT command is necessary to make the Drive write with the tape format found on the inserted cartridge. However, if the EADS (Enable Automatic Density Selection) field is set to one and a WRITE from BOT is issued, the Drive will use the tape format giving the largest capacity on the inserted cartridge. Note, however, that the QIC-24 tape format is not a valid write tape format.
If the inserted cartridge does not contain data (a blank cartridge), the MODE SENSE and MODE SELECT Density Codes will automatically be set to the "highest" possible format for the current cartridge. See the Type of Media Related to Tape Format table under the description of the Density Code field for the actual tape format selected for various cartridge types. The Block Size will be set to 512 bytes when the selected tape format is QIC-24/120/150.

When the tape format is set to QIC-525/1000/2GB/96-34 the Block Size will be set to the saved Block Size found in the Drive's EEPROM. Note that when no tape is inserted, the MODE SENSE and MODE SELECT Density Codes will be set to UNKNOWN.

- Legal values are numbers in the range 0..1.
- The default (factory programmed) value is 0.

The Drive will skip up to 8 sequential filemarks at the Beginning Of Tape to get to the first data block. If only filemarks on the tape, or more than 8 sequential filemarks on the Beginning Of Tape, the Block Size will be set to zero.

***SLR5 4.0/8.0GB and SLR4 2.5/5.0GB (TDC 4222) only***
**IDSOT**

Ignore Data compression Settings when On Tape. When set, this bit makes the Drive ignore MODE SELECT commands with data compression ON or OFF and compression algorithm settings when the tape is positioned away from BOT. See Section 15.3.5. Data Compression Page (Page Code 20H). GOOD STATUS is reported, but the settings are ignored. However, Normal Field Checking is still performed, thus settings that are RESERVED, or illegal parameters, will still result in a CHECK CONDITION status as described in Section 15.3.5.

- Legal values are 0 and 1.
- The default (factory programmed) value is 0.

### 15.3.9. User Page 0

This page can be used to save any information (up to ten bytes).

The page can be saved.

| BYTE | BIT 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 00 | PS | R | Page Code = 21h | | | | | |
| 01 | Parameter Length = 0Ah | | | | | | | |
| 02 | User Defined Field | | | | | | | |
| 03 | User Defined Field | | | | | | | |
| 04 | User Defined Field | | | | | | | |
| 05 | User Defined Field | | | | | | | |
| 06 | User Defined Field | | | | | | | |
| 07 | User Defined Field | | | | | | | |
| 08 | User Defined Field | | | | | | | |
| 09 | User Defined Field | | | | | | | |
| 10 | User Defined Field | | | | | | | |
| 11 | User Defined Field | | | | | | | |

*Table: User Page 0 Page Descriptor*

**PS** The Parameter Saveable (PS) bit MUST be set to zero.

**Page Code** The Page Code for this page will be set to 21h.

**Parameter Length** The Parameter Length field MUST always be set to 0Ah.

**User Defined Field** These fields can be used to store up to ten bytes of user defined information.

- Any value is legal (no check).
- The default (factory programmed) value is FFh.

### 15.3.10. User Page 1

This page can be used to save any information (up to ten bytes).

The page can be saved.

| BYTE | BIT 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 00 | PS | R | Page Code = 22h | | | | | |
| 01 | Parameter Length = 0Ah | | | | | | | |
| 02 | User Defined Field | | | | | | | |
| 03 | User Defined Field | | | | | | | |
| 04 | User Defined Field | | | | | | | |
| 05 | User Defined Field | | | | | | | |
| 06 | User Defined Field | | | | | | | |
| 07 | User Defined Field | | | | | | | |
| 08 | User Defined Field | | | | | | | |
| 09 | User Defined Field | | | | | | | |
| 10 | User Defined Field | | | | | | | |
| 11 | User Defined Field | | | | | | | |

*Table: User Page 1 Page Descriptor*

**PS** The Parameter Saveable (PS) bit MUST be set to zero.

**Page Code** The Page Code for this page will be set to 22h.

**Parameter Length** The Parameter Length field MUST always be set to 0Ah.

**User Defined Field** These fields can be used to store up to ten bytes of user defined information.

- Any value is legal (no check).

- The default (factory programmed) value is FFh.

## 15.4. Exception Handling

See sections on Error Conditions For All Commands and Deferred Errors.

If the MODE SELECT command, for any reason, returns with CHECK CONDITION status, no parameters will have been changed.

If the PF bit is not set to one, the MODE SELECT command will return CHECK CONDITION status. No parameter data will be transferred. The Error Code will be set to E$STE_IFIC.

A parameter length that results in the truncation of any descriptor, header or page of parameters will cause the Drive to terminate the command with CHECK CONDITION status. The Error Code will be set to E$STE_PLEN.

If the Buffered Mode, Speed or Block Descriptor Length fields in the parameter list header are set to illegal values, the Drive will terminate the MODE SELECT command with CHECK CONDITION status. The Error Code will be set to E$STE_IFIP.

If the Density Code, Number Of Blocks or Block Size fields in the Block Descriptor List are set to illegal values, the Drive will terminate the MODE SELECT command with CHECK CONDITION status. The Error Code will be set to E$STE_IFIP.

If the Page Code in a Page Descriptor List is not in the legal set of values, the Drive will terminate the MODE SELECT command with CHECK CONDITION status. The Error Code will be set to E$STE_IFIP.

If any of the fields in any of the Mode Pages are set to illegal values, the Drive will terminate the MODE SELECT command with CHECK CONDITION status. The Error Code will be set to E$STE_IFIP.

If the Parameter Length in any page is wrong, the Drive will terminate the MODE SELECT command with CHECK CONDITION status. The Error Code will be set to E$STE_IFIP.

If any non-changeable pages or any non-changeable fields in any page are specified for change, the Drive will terminate the MODE SELECT command with CHECK CONDITION status. The Error Code will be set to E$STE_IFIP.

If any non-saveable pages are specified when the SP bit is set, the Drive will terminate the MODE SELECT command with CHECK CONDITION status. The Error Code will be set to E$STE_IFIP.

If an illegal change of Density Code is attempted, the Drive will terminate the MODE SELECT command with CHECK CONDITION status. The Error Code will be set to E$BTD_TFMT (Incompatible Tape Format).

***SLR5 4.0/8.0GB and SLR4 2.5/5.0GB (TDC 4222) only***

If an unsupported Compression or Decompression Algorithm is specified, the Drive will terminate the MODE SELECT command with CHECK CONDITION status.

The Error Code will be set to E$STE_IFIP.

If the Data Compression Page settings are different from the current page settings and the tape is positioned away from BOT, the Drive will terminate the MODE SELECT command with CHECK CONDITION status.

The Error Code will be set to E$STE_IFIP.

# 16. Mode Sense

## 16.1. Command Description

The MODE SENSE command provides a means for the Drive to report parameters to the Initiator. It is a complementary command to the MODE SELECT command.

The Drive will implement only one common set of parameters for all Initiators.

The MODE SENSE parameter list will be returned during the DATA IN phase of the command.

If disconnection is allowed, the Drive will disconnect when executing this command if the previous command was an Immediate type command.

When page format is used the MODE SENSE command may return 4 different types of parameters; current values, changeable values, default values or saved values.

**Current Values**

The current values are the values under which the Drive is presently configured for the page specified. The current values returned are:

1) The parameters set in the last successful MODE SELECT command.

2) The saved values if a MODE SELECT command has not been executed since the last power-up, RESET condition or BUS DEVICE RESET message.

3) The default values if a MODE SELECT command has never been executed with the save parameter (SP) bit set.

**Changeable Values**

The page requested will be returned containing information that indicates which fields are changeable. Parameters that are changeable will be set to one. Parameters that are not changeable will be set to zero. If any part of a field is changeable all bits in that field will be set to one. If none of the parameters are changeable within a page, the Page Length value will be set to zero.

**Default Values**

The default values are set once and for all in the Drive's production line. Parameters not supported by the Drive will be set to zero.

**Saved Values**

The saved values are the values saved by the last successful MODE SELECT command with the save parameter (SP) set or the default values if no MODE SELECT with the SP bit has been executed. Saved values are located in the EEPROM. Parameters not supported by the Drive will be set to zero.

***SLR5 4.0/8.0GB and SLR4 2.5/5.0GB (TDC 4222) only***

Unlike for other Mode parameters, the Drive may change the current data compression parameters dynamically during normal operation. The current values for the Data Compression Page reflect the present operating settings. The current values returned in response to a MODE SENSE command will be as described below:

- The Drive may turn compression ON when writing and reading on the QIC-2GB or QIC-4GB tape formats only.
- The data compression parameters will be set equal to the saved EEPROM values when the Drive has reported Unit Attention due to power-up/reset. The settings will remain the same until a MODE SELECT command is executed, or until a READ/SPACE operation is performed.
- When a READ/SPACE-type command has been performed, the data compression parameters will automatically be updated to reflect the values on the tape just read. Both decompression and compression settings will be set equal to the compression parameters from the tape. If NO data compression parameters are found on the tape, compression will be turned OFF.
- When a WRITE-type command has been performed from BOT, the data compression settings will be equal to the values set by the last MODE SELECT command. If no such command has been executed, the values will be equal to the saved EEPROM values.

Note that after having issued a READ-type command succeeded by a REWIND command, the data compression parameters returned on a MODE SENSE command will reflect the settings detected on the tape. A following WRITE command from BOT will, however, use the parameters last set by MODE SELECT.

## 16.2. Command Descriptor Block

| BYTE | BIT 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 00 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 |
| 01 | Logical Unit Number (LUN) | | | R | DBD | RESERVED | | |
| 02 | PC | | Page Code | | | | | |
| 03 | RESERVED | | | | | | | |
| 04 | Allocation Length | | | | | | | |
| 05 | Control Byte | | | | | | | |

*Table: MODE SENSE Command Block*

**DBD**

A Disable Block Descriptor (DBD) bit of one indicates that the Block Descriptor List will not be transferred to the Initiator. A DBD bit of zero indicates that the Block Descriptor List will be transferred following the Header List.

**PC**

The page control (PC) field defines the type of parameter values to be returned.

**0 : Return current values**
**1 : Return changeable values**
**2 : Return default values**
**3 : Return saved values**

**Page Code**

The page code specifies which page(s) to return.

NOTE:
The special page code 3Fh can be used to have the Drive return all its page descriptors.

- Legal values are 00h, 01h, 02h, 10h, 11h, 20h, 21h, 22h and 3Fh.

***SLR5 4.0/8.0GB and SLR4 2.5/5.0GB (TDC 4222) only***

- Additional legal value is 0Fh.

Page Code 00h is returning the Header List followed by the Block Descriptor List - a total of 12 bytes. When selecting Page Code 00h the DBD bit is ignored.

See the MODE SELECT command for a description of the various pages.

**Allocation Length**

This field specifies the maximum number of bytes that the Initiator has allocated for the returned MODE SENSE data. An Allocation Length of zero indicates that no MODE SENSE data will be sent. The Drive terminates the DATA-IN phase when Allocation Length bytes have been transferred or when all available MODE SENSE data have been transferred, whichever is less.

## 16.3. Parameter List

### 16.3.1. Header List

The MODE SENSE parameter list consists of three sub-lists. The first list is a 4-byte Header List. This may be followed by a 8-byte Block Descriptor List. At last there may be one or up to seven Page Descriptor Lists.

***SLR5 4.0/8.0GB and SLR4 2.5/5.0GB (TDC 4222)***

Up to eight Page Descriptor Lists are possible for the SLR5 4.0/8.0GB and SLR4 2.5/5.0GB (TDC 4222) Drives.

| BYTE | BIT 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 00 | Sense Data Length | | | | | | | |
| 01 | Medium Type | | | | | | | |
| 02 | WP | Buffered Mode | | | Tape Speed | | | |
| 03 | Block Descriptor Length | | | | | | | |

*Table: MODE SENSE Header List*

**Sense Data Length**

This field specifies the length in bytes of the following MODE SENSE data (the Sense Data Length byte itself is not included) that is available to be transferred during the DATA IN phase

**Medium Type**

This field will indicate the type of cartridge inserted into the Drive.

| | | |
|---|---|---|
| **00h** | **:** | **UNKNOWN** |
| **02h** | **:** | **DC300 or DC300XLP** |
| **04h** | **:** | **DC615 or DC600A** |
| **06h** | **:** | **DC6037, DC6150 or DC6250** |
| **08h** | **:** | **DC6320 or DC6525** |
| **21h** | **:** | **DC9135SL** |
| **22h** | **:** | **DC9210** |
| **23h** | **:** | **DC9135** |
| **24h** | **:** | **DC9100** |
| **25h** | **:** | **DC9120** |
| **26h** | **:** | **DC9100S** |
| **27h** | **:** | **DC9164** |
| **30h** | **:** | **DC9100FW** |
| **31h** | **:** | **DC9200SL** |
| **32h** | **:** | **DC9210XL** |
| **33h** | **:** | **DC10GB** |
| **34h** | **:** | **DC9200** |
| **35h** | **:** | **DC9120XL** |
| **36h** | **:** | **DC9210SL** |
| **37h** | **:** | **DC9164XL** |
| **40h** | **:** | **DC9200XL** |
| **42h** | **:** | **DC9500** |
| **55h** | **:** | **DC9400** |
| **56h** | **:** | **DC9400SL** |

**WP**

The Write Protect (WP) bit of zero indicates that the cartridge is write enabled. A WP bit of one indicates that the cartridge is write protected.

**Buffered Mode**

The Drive supports the following mode: 0 and 1. Mode 0 indicates that the WRITE and COPY (copy function = backup) and WRITE FILEMARKS command reports GOOD status when the requested data blocks have been actually written and verified. Mode 1 indicates that the WRITE (and COPY, WRITE FILEMARKS) command reports GOOD status as soon as the requested data has been transferred into the Drive's data buffer.

**Tape Speed**

This field specifies the current tape speed. The following values are legal:

**0h:** Default
The actual speed depends on the current tape format:

| Tape Format | Tape Speed SLR5 Series | Tape Speed SLR (TDC 4000) Series |
|---|---|---|
| QIC-24/120/150 | 96 ips | 96 ips |
| QIC-525 | 120 ips | 120 ips |
| QIC-1000 | 107 ips | 53 ips |
| QIC-2GB | 95 ips | 70.9 ips |
| QIC-4GB | 76 ips | - |

**2h:** 80 ips when the Drive is in QIC-1000 mode. For the SLR5 Series Drive, QIC-1000 format is written/read in 107 ips (400 KByte/s).

***SLR2 (TDC 3820) only*** The Tape Speed-bit is reserved for SLR2 (TDC 3820) and will always be set to 0.

***SLR5 Series*** The Default tape speed (0h) is 107 ips for QIC-1000.

**Block Descriptor Length**

This field specifies the length in bytes of the block descriptor list. If the block descriptor list is transferred (DBD-bit = 0 in the CDB), the Block Descriptor Length is 8. If the block descriptor list is not transferred (DBD-bit = 1), the Block Descriptor Length is 0.

### 16.3.2. Block Descriptor List

| BYTE | BIT 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 00 | Density Code | | | | | | | |
| 01 | Number of Blocks | | | | | | | |
| 02 | | | | | | | | |
| 03 | | | | | | | | |
| 04 | RESERVED | | | | | | | |
| 05 | Block Size | | | | | | | |
| 06 | | | | | | | | |
| 07 | | | | | | | | |

*Table: MODE SENSE Block Descriptor List*

**Density Code**

This field indicates the current operating tape format. The Density Code value returned in response to a MODE SENSE command will be as described below:

- *The tape format will be set Unknown when the Drive has reported Unit Attention due to power-up/reset or when a new cartridge has been inserted. The tape format will remain Unknown until a read/space or write type operation has been performed.*
- *When a successful read/space type command has been performed the Density Code will be automatically updated to reflect the format on the tape just read.*
- *When a write type command has been performed the Density Code will be set equal to the density code used by the last write type command. See the corresponding field in the MODE SELECT command for details.*

Possible Density Codes are:

| | | |
|---|---|---|
| **00h** | **:** | **Unknown** |
| **05h** | **:** | **QIC-24** |
| **0Fh** | **:** | **QIC-120** |
| **10h** | **:** | **QIC-150** |
| **11h** | **:** | **QIC-525** |
| **15h** | **:** | **QIC-1000** |
| **22h** | **:** | **QIC-2GB** |
| **26h** | **:** | **QIC-4GB** |
| **90h** | **:** | **QICVault720** |

| | |
|---|---|
| ***SLR2 (TDC 3820) only*** | Density codes 00h - 11h and 90h may be returned. |
| ***TDC 4100 only*** | Density codes 00h - 15h and 90h may be returned. |
| ***TDC 4200 & SLR4 2.5/5.0GB (TDC 4222)*** | Density codes 00h - 22h and 90h may be returned. |
| ***SLR5 Series*** | Density codes 00h, 0Fh -26h may be returned. |

**Number Of Blocks**

This field will always be set to zero to indicate that the whole tape has the same density code and block length.

**Block Size**

This field reports the currently configured fixed block size. See the corresponding field in the MODE SELECT command for details.

### 16.3.3. Error Recovery Page Descriptor

See the MODE SELECT command for details on the current, saved and default values of this page (Page Code 01h)

The PS bit will be set to one for this page.

The Changeable Values page will return the following values:

| | | | |
|---|---|---|---|
| Byte 00 | : | 81h | |
| Byte 01 | : | 0Ah | |
| Byte 02 | : | 05h | The DCR and PER bits are changeable |
| Byte 03 | : | FFh | Read Retry Count is changeable |
| Byte 04 | : | 00h | |
| Byte 05 | : | 00h | |
| Byte 06 | : | 00h | |
| Byte 07 | : | 00h | |
| Byte 08 | : | FFh | Write Retry Count is changeable |
| Byte 09 | : | 00h | |
| Byte 10 | : | 00h | |
| Byte 11 | : | 00h | |

### 16.3.4. Disconnect/Reconnect Page Descriptor

See the MODE SELECT command for details on the current, saved and default values of this page (Page Code 02h)

The PS bit will be set to one for this page.

The Changeable Values page will return the following values:

| | | | |
|---|---|---|---|
| Byte 00 | : | 82h | |
| Byte 01 | : | 0Eh | |
| Byte 02 | : | FFh | Read Buffer Full Ratio is changeable |
| Byte 03 | : | FFh | Write Buffer Empty Ratio is changeable |
| Byte 04 | : | 00h | |
| Byte 05 | : | 00h | |
| Byte 06 | : | 00h | |
| Byte 07 | : | 00h | |
| Byte 08 | : | 00h | |
| Byte 09 | : | 00h | |
| Byte 10 | : | 00h | |
| Byte 11 | : | 00h | |
| Byte 12 | : | 00h | |
| Byte 13 | : | 00h | |
| Byte 14 | : | 00h | |
| Byte 15 | : | 00h | |

### 16.3.5. SLR5 4.0/8.0GB and SLR4 2.5/5.0GB (TDC 4222) Only - Data Compression Page Descriptor

See the MODE SELECT command for details on the current, saved and default values of this page (Page Code 0Fh)

The PS bit will be set to one for this page.

The Changeable Values page will return the following values:

| | | | |
|---|---|---|---|
| **Byte 00** | **:** | **8Fh** | |
| **Byte 01** | **:** | **0Eh** | |
| **Byte 02** | **:** | **80h** | **DCE-bit is changeable** |
| **Byte 03** | **:** | **80h** | **DDE-bit is changeable** |
| **Byte 04** | **:** | **FFh** | |
| **Byte 05** | **:** | **FFh** | |
| **Byte 06** | **:** | **FFh** | |
| **Byte 07** | **:** | **FFh** | **Comp. Algorithm field is changeable** |
| **Byte 08** | **:** | **FFh** | |
| **Byte 09** | **:** | **FFh** | |
| **Byte 10** | **:** | **FFh** | |
| **Byte 11** | **:** | **FFh** | **Decomp. Algorithm field is changeable** |
| **Byte 12** | **:** | **00h** | |
| **Byte 13** | **:** | **00h** | |
| **Byte 14** | **:** | **00h** | |
| **Byte 15** | **:** | **00h** | |

### 16.3.6. Device Configuration Parameters Page Descriptor

See the MODE SELECT command for details on the current, saved and default values of this page (Page Code 10h)

The PS bit will be set to one for this page.

The Changeable Values page will return the following values:

| | | | |
|---|---|---|---|
| **Byte 00** | **:** | **90h** | **Page Code** |
| **Byte 01** | **:** | **0Eh** | **Page Length** |
| **Byte 02** | **:** | **40h** | **CAP bit is changeable** |
| **Byte 03** | **:** | **FFh** | **Active Partition is changeable** |
| **Byte 04** | **:** | **FFh** | **Write Buffer Full Ratio is changeable** |
| **Byte 05** | **:** | **FFh** | **Read Buffer Empty Ratio is changeable** |
| **Byte 06** | **:** | **FFh** | **Write Delay Time is changeable** |
| **Byte 07** | **:** | **FFh** | **Write Delay Time is changeable** |
| **Byte 08** | **:** | **00h** | |
| **Byte 09** | **:** | **00h** | |
| **Byte 10** | **:** | **00h** | |
| **Byte 11** | **:** | **00h** | |
| **Byte 12** | **:** | **00h** | |
| **Byte 13** | **:** | **00h** | |
| **Byte 14** | **:** | **00h** | |
| **Byte 15** | **:** | **00h** | |

### 16.3.7. Medium Partition Parameters Page Descriptor

See the MODE SELECT command for details on the current, saved and default values of this page (Page Code 11h)

The PS bit will be set to one for this page.

The Changeable Values page will return the following values:

| | | | |
|---|---|---|---|
| **Byte 00** | **:** | **91h** | |
| **Byte 01** | **:** | **06h** | |
| **Byte 02** | **:** | **00h** | |
| **Byte 03** | **:** | **00h** | |
| **Byte 04** | **:** | **80h** | **FDP bit is changeable** |
| **Byte 05** | **:** | **00h** | |
| **Byte 06** | **:** | **00h** | |
| **Byte 07** | **:** | **00h** | |

### 16.3.8. Miscellaneous Parameters Page Descriptor

See the MODE SELECT command for details on the current, saved and default values of this page (Page Code 20h)

The PS bit will be set to one for this page.

The Changeable Values page will return the following values:

| | | | |
|---|---|---|---|
| **Byte 00** | **:** | **A0h** | |
| **Byte 01** | **:** | **0Ah** | |
| **Byte 02** | **:** | **03h** | **Forced Streaming Count is changeable** |
| **Byte 03** | **:** | **FFh** | **Forced Streaming Count is changeable** |
| **Byte 04** | **:** | **FFh** | **ASI and Target Sense Length are changeable** |
| **Byte 05** | **:** | **FFh** | **Copy Threshold is changeable** |
| **Byte 06** | **:** | **FFh** | **Load Function is changeable** |
| **Byte 07** | **:** | **FFh** | **Power-Up/Reset Load-Delay is changeable** |
| **Byte 08** | **:** | **FFh** | **FAST, RD, BSY, EADS, EOWR, SPEW, DTM2 and DTM1 bits are changeable** |
| **Byte 09** | **:** | **F0h** | **LED Function is changeable** |
| **Byte 10** | **:** | **FFh** | **PSEW Position is changeable** |
| **Byte 11** | **:** | **00h** | |

### 16.3.9. User Page 0 Page Descriptor

See the MODE SELECT command for details on the current, saved and default values of this page (Page Code 21h)

The PS bit will be set to one for this page.

The Changeable Values page will return the following values:

| | | | |
|---|---|---|---|
| **Byte 00** | **:** | **A1h** | |
| **Byte 01** | **:** | **0Ah** | |
| **Byte 02** | **:** | **FFh** | **All Bytes are changeable** |
| **Byte 03** | **:** | **FFh** | |
| **Byte 04** | **:** | **FFh** | |
| **Byte 05** | **:** | **FFh** | |
| **Byte 06** | **:** | **FFh** | |
| **Byte 07** | **:** | **FFh** | |
| **Byte 08** | **:** | **FFh** | |
| **Byte 09** | **:** | **FFh** | |
| **Byte 10** | **:** | **FFh** | |
| **Byte 11** | **:** | **FFh** | |

### 16.3.10. User Page 1 Page Descriptor

See the MODE SELECT command for details on the current, saved and default values of this page (Page Code 22h)

The PS bit will be set to one for this page.

The Changeable Values page will return the following values:

| | | | |
|---|---|---|---|
| **Byte 00** | **:** | **A2h** | |
| **Byte 01** | **:** | **0Ah** | |
| **Byte 02** | **:** | **FFh** | **All Bytes are changeable** |
| **Byte 03** | **:** | **FFh** | |
| **Byte 04** | **:** | **FFh** | |
| **Byte 05** | **:** | **FFh** | |
| **Byte 06** | **:** | **FFh** | |
| **Byte 07** | **:** | **FFh** | |
| **Byte 08** | **:** | **FFh** | |
| **Byte 09** | **:** | **FFh** | |
| **Byte 10** | **:** | **FFh** | |
| **Byte 11** | **:** | **FFh** | |

## 16.4. Exception Handling

See sections on Error Conditions For All Commands and Deferred Errors.

If the Page Code is not in the range of legal values, the MODE SENSE command will be terminated with CHECK CONDITION status. The Error Code will be set to E$STE_IFIC.

# 17. Prevent/Allow Medium Removal

## 17.1. Command Description

The PREVENT/ALLOW MEDIUM REMOVAL command does nothing but operate the front LED. PREVENT may turn the Green LED *on*. ALLOW may turn the Green LED *off*. The actual LED operation depends on the current LED function. (See the MODE SELECT *LED function* in the Miscellaneous Parameters page for details). This is the only command that operates the front LED.

If disconnection is allowed, the Drive will disconnect when executing this command if the previous command was an immediate type command.

## 17.2. Command Descriptor Block

| BYTE | BIT 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 00 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 |
| 01 | Logical Unit Number (LUN) | | | RESERVED | | | | |
| 02 | RESERVED | | | | | | | |
| 03 | RESERVED | | | | | | | |
| 04 | RESERVED | | | | | | | PREV |
| 05 | Control Byte | | | | | | | |

*Table: PREVENT/ALLOW MEDIUM REMOVAL Command Block*

**PREV**

A Prevent (PREV) bit of one may turn the Green LED *on*. A PREV bit of zero may turn the Green LED *off*. See also Chapter 15.

## 17.3. Exception Handling

See sections on Error Conditions For All Commands, Deferred Errors and Error Conditions For Media Access Commands.

PREVENT will terminate with CHECK CONDITION status if the LED mode is 0 to 4, and if no cartridge is inserted into the Drive.
The Error Code will be set to E$STE_NCAR.

If the LED mode is 5, the command will be accepted and executed even without a cartridge.

*This Page Intentionally Left Blank*

# 18. Read

## 18.1. Command Description

The READ command transfers one or more blocks to the Initiator beginning with the next block on the tape. The Fixed (FIX) bit specifies both the meaning of the Transfer Length field and whether fixed-length or variable length block(s) are to be transferred. The data read will be returned during the DATA-IN phase of the command.

When the FIX bit is set to zero, the Drive is requested to transfer a single variable length data block. The Transfer Length specifies the block length in number bytes. The block length found on the tape is expected to be equal to the specified block length.

When the FIX bit is set to one, the Drive is requested to transfer a number of fixed length blocks. The Transfer Length specifies the number of blocks to transfer. All the blocks is expected to be of the same length. The length expected is the length reported by the MODE SENSE command (the Block Size field of the Block Descriptor List). Note that a FIX bit of one is not legal when the Drive has been set into Variable Block mode. Variable Block mode is in effect when the Block Size field in the Block Descriptor List of the MODE SELECT command is set to zero (000000h). See the MODE SELECT command for further details.

If the requested transfer length is zero, then the Drive will transfer no data and the logical tape position will not be changed. This will not be considered an error.

If the READ command is the first media access command executed on a newly inserted cartridge, the read operation will start from BOM. If the READ command follows a ERASE, LOAD/UNLOAD (with Load bit set to one) or REWIND command, the read operation will also start from BOM. If the READ command follows a COPY (with Copy Function = restore), LOCATE, SPACE, VERIFY or another READ command, the read operation will start with the next block on the tape.

Upon termination of a successful READ command, the logical tape position will be after the last block (fixed or variable) read (end-of-media side).

If disconnection is allowed, the Drive will disconnect when executing this command if the number of blocks requested for transfer exceeds the number of blocks available in the data buffer when the command has been received.

## 18.2. Command Descriptor Block

| BYTE | BIT 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 00 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 01 | Logical Unit Number (LUN) | | | RESERVED | | | SILI | FIX |
| 02 | Transfer Length | | | | | | | |
| 03 | | | | | | | | |
| 04 | | | | | | | | |
| 05 | Control Byte | | | | | | | |

*Table: READ Command Block*

**SILI**

If the Suppress Incorrect Length Indicator (SILI) bit is one and the fixed bit is zero, the Drive will not report CHECK CONDITION when an incorrect length block is read and the only error is that the requested transfer length exceeds the actual block length. If, however, the requested transfer length is less than the actual block length, the Drive will report CHECK CONDITION even if the SILI bit is set.

**FIX**

A Fixed (FIX) bit of zero indicates that a single block will be transferred with the Transfer Length specifying the maximum number of bytes the Initiator has allocated for the returned data. A FIX bit of one indicates that the Transfer Length specifies the number of blocks to be transferred to the Initiator.

NOTE:
A FIX bit of one is not allowed when the Drive is in Variable Block mode (see the Block Size field in the Block Descriptor List of the MODE SELECT command for further details).

**Transfer Length**

This field specifies the number of bytes or blocks requested for transfer. Any value in the range 0..16777215 is legal.

NOTE:
For the QIC-120 and QIC-150 tape formats the maximum block size that can be written by the Drive is 32768 bytes and for the QIC-24 tape format the block length is always 512 bytes.

## 18.3. Exception Handling

### 18.3.1. General

See sections on Error Conditions For All Commands, Deferred Errors, Error Conditions For Media Access Commands and Buffer Parity Errors.

When the READ command has started to execute, all detected errors will set the VADD bit and the Information Bytes will hold the difference between the requested and the actual transfer length. See the following sections for details.

If the FIX bit is one and the Drive is in Variable Block mode, the READ command will be terminated with CHECK CONDITION. The Error Code will be set to E$STE_IFIC. No data will be transferred.

If the FIX bit is one and the configured Block Size is 1024, the READ command will be terminated with CHECK CONDITION if the tape format is different from QIC-525/1000/2GB/96-34.
The Error Code will be set to E$BTD_TFMT. No data will be transferred and the tape position will be at BOT

If both the SILI and the FIX bits are one, the Drive will terminate the READ command with CHECK CONDITION status. The Error Code will be set to E$STE_IFIC.

### 18.3.2. No Data

If the Drive is not able to find a reference burst on the inserted cartridge, the cartridge is assumed to be blank and the READ command will be terminated with CHECK CONDITION. The Error Code will be set to E$TCM_NODATA. The Valid (VADD) bit in the sense data list will be set to one. The Information Bytes will be set equal to the Requested Transfer Length.

### 18.3.3. Filemark Detected

If a filemark is encountered during execution of a READ command, the command will transfer all data up to the filemark. The READ command will then be terminated with CHECK CONDITION status. The filemark (FMK) bit will be set to one and the Error Code will be set to E$BTD_FIMK. The Valid (VADD) bit in the sense data list will be set to one. The Information Bytes will be set to the difference (residue) between the requested transfer length and the actual transfer length (bytes or blocks). For variable length reads the Information Bytes will be set equal to the Transfer Length (because a filemark block was found instead of a data block and no data was transferred). When the command has terminated, the logical tape positions will be located after the filemark (end-of-media side).

## 18.3.4. Illegal Length

### 18.3.4.1. FIX Bit Set to ZERO

If the actual block length (length of block found on the tape) is different from the specified transfer length, the illegal length block will first be transferred to the Initiator. The Drive will, however, not transfer more data than specified in the Transfer Length of the READ Command Descriptor Block. If the SILI bit is zero, the READ command will then be terminated with CHECK CONDITION status. The Error Code will be set to E$STE_ILLN. The illegal length indicator (ILI) and Valid (VADD) bits in the sense data list will be set to one. The Information Bytes will be set to the difference (residue) of the requested transfer length minus the actual block length. If the actual block length was smaller than the specified length, the residual will be a positive number. If the actual block length was larger than the specified length, the residual will be a negative number. Negative residues will be presented on 2's complement form.

If the SILI bit is one, the behavior depends on the current Fixed or Variable Block mode:

1) **Fixed Block Mode**
   In Fixed Block mode (configured Block Size is different from zero) a SILI bit of one indicates that the Drive will not return CHECK CONDITION status if the only error is that the transfer length exceeds the actual block length recorded on the tape. When the command has terminated, the logical tape position will be located after the incorrect length block (end of partition side).

2) **Variable Block Mode**
   In Variable Block mode (configured Block Size is set to zero) a SILI bit of one indicates that the Drive will not return CHECK CONDITION status if the only error is that the transfer length does not match the actual block length recorded on the tape. When the command has terminated, the logical tape position will be located after the incorrect length block (end of partition side).

### 18.3.4.2. FIX Bit Set to ONE

If the actual block length (length of block found on the tape) is different from the configured block length, the illegal length block will first be transferred to the Initiator. The Drive will, however, not transfer more data than configured as the block length (see MODE SELECT command). The READ command will then be terminated with CHECK CONDITION status. The Error Code will be set to E$STE_ILLN. The illegal length indicator (ILI) and Valid (VADD) bits in the sense data list will be set to one. The Information Bytes will be set to the difference (residue) of the requested transfer length minus the actual number of blocks transferred (not including the incorrect length block). The block with the unexpected length is not counted among the transferred blocks. When the command has terminated, the logical tape position will be located after the incorrect length block (end of partition side).

### 18.3.5. Logical End of Partition

If logical end of partition (end-of-recorder-area) is encountered during execution of the READ command, the command will transfer all data block(s) up to the logical end of partition. The READ command will then be terminated with CHECK CONDITION status. The Sense Key will be set to BLANK CHECK. Additionally the Error Code and the End Of Media (EOM) bit will be set as follows; if the logical end of partition is encountered before the early warning (EW) tape marker on the last track, the Drive will set the Error Code to E$TEM_EOR and the EOM bit to zero. If logical end of data is encountered at or after the physical early warning (EW) tape marker on the last track, the Drive will set the Error Code to E$TEM_EOREW and the EOM bit to one. The Valid (VADD) bit in the sense data list will be set to one. The Information Bytes will be set to the difference (residue) between the requested transfer length and the actual transfer length (bytes or blocks). When the command has terminated, the logical tape position will be located after the last block transferred to the Initiator.

When a variable length block has been abnormally truncated, due to a Logical End Of Partition detection, Illegal Length is not signalled. Instead the Logical End Of Partition Error takes priority and the Error Code is set to E$TEM_EOR or E$TEM_EOREW.

### 18.3.6. Physical End of Partition

If physical end of partition is encountered during execution of the READ command, the command will transfer all data block(s) up to the physical end of partition. The READ command will then be terminated with CHECK CONDITION status. The Error Code will then be set to E$TEM_PEOP and the Sense Key will be set to MEDIUM ERROR. The End Of Media (EOM) and Valid (VADD) bits in the sense data list will be set to one. The Information Bytes will be set to the difference (residue) between the requested transfer length and the actual transfer length (bytes or blocks). When the command has terminated the logical tape position is undefined and all new COPY (copy function = restore), READ, SEEK BLOCK/LOCATE or SPACE commands will be terminated immediately with CHECK CONDITION as if they just ran into physical end of partition. A possible COPY (copy function = backup), WRITE or WRITE FILEMARKS command will also be terminated immediately with CHECK CONDITION status. The Error Code will then be set to E$BTD_WRRD. This situation will maintained until a position type command has been executed (ERASE, LOAD/UNLOAD or REWIND).

When a variable length block has been truncated, due to a Physical End Of Partition detection, Illegal Length is not signalled. Instead the Physical End Of Partition Error takes priority and the Error Code is set to E$TEM_PEOP.

## 18.3.7. Non-Recoverable Read Error

### 18.3.7.1. Fixed Blocks

**1) QIC-525/1000/2GB/96-34 Tape Format**

If a non-recoverable read error occurs during the execution of a READ fixed blocks command, the Drive will transfer all good data up to the non-recoverable physical tape block. The Drive will then transfer one dummy SCSI-block (block with random data). This dummy block is transferred instead of the first half of the non-recoverable physical tape block (the size of the logical blocks on the SCSI-bus is 512 bytes or half the size of the QIC-525/1000/2GB/96-34 physical tape blocks). The Drive will then terminate the READ command with CHECK CONDITION status. The Error Code will be set to E$BTD_RTRY. The Valid (VADD) bit in the sense data list will be set to one. The Information Bytes will be set to the difference (residue) between the requested transfer length and the actual transfer length. When the command has terminated the logical tape position will be located in the middle of the bad physical tape block. Note that if the residual length is zero, the Initiator has all the requested data with the last SCSI-block having random data.

If a new READ (or COPY (backup), SPACE or VERIFY) command is issued after a non-recoverable read error has occurred, the read operation will continue with the second half of the non-recoverable physical tape block. The new READ command will transfer a second dummy SCSI-block as the first and only block. The READ command will then again terminate with CHECK CONDITION and the Error Code will be set to E$BTD_RTRY. The Valid (VADD) bit and Information Bytes will be set as described above. The logical tape position is now located after the complete bad physical tape block (end-of-partition side).

**2) Other Tape Formats**

If a non-recoverable read error occurs during the execution of a READ fixed blocks command, the Drive will transfer all good data up to the non-recoverable physical tape block. The Drive will then transfer one dummy block (block with random data). This dummy block is transferred instead of the non-recoverable physical tape block (the size of the logical blocks on the SCSI-bus is equal to the size of the physical blocks on the tape). The Drive will then terminate the READ command with CHECK CONDITION status. The Error Code will be set to E$BTD_RTRY. The Valid (VADD) bit in the sense data list will be set to one. The Information Bytes will be set to the difference (residue) between the requested transfer length and the actual transfer length. When the command has terminated the logical tape position will be located after the complete bad physical tape block (end-of-partition side). Note that if the residual length is zero, the Initiator has all the requested data with the last SCSI-block having random data.

If a new READ (or COPY (backup), SPACE or VERIFY) command is issued after a non-recoverable read error has occurred, the read operation will continue with the block following the bad block.

### 18.3.7.2. Variable Blocks

If a non-recoverable read error occurs during the execution of a READ variable block command, the Drive will transfer all god data up to the non-recoverable physical tape block.
There are now two possibilities; the bad block holds data that the Initiator has requested for transfer or the bad block does not hold data that is requested for transfer.

When a variable length block has been truncated due to a Non-recoverable Read Error, Illegal Length is not signalled. Instead the Non-recoverable Read Error takes priority and the Error Code is set as described in the following sections.

1) **Bad Block Holds Requested Data**

The Drive will then transfer up 1024 bytes of the remaining data from a dummy block (block with random data) in place of the non-recoverable physical tape block. The Drive will then terminate the READ command with CHECK CONDITION status. The Error Code will be set to E$BTD_RTRY. The Valid (VADD) bit in the sense data list will be set to one. The Information Bytes will be set to the difference (residue) between the requested transfer length and the actual transfer length. When the command has terminated, the logical tape positions will be located after the bad physical tape block (end of partition side) even if this is in the middle of a large variable block. Note that if the residual length is zero, then the Initiator has all the requested data with the last bytes of the variable block having random data. The number of bytes with random data can be calculated with the following formula:

**Nr = ( Nt - 1) mod 1024**

**Nr : number of bytes with random data**
**Nt : total number of bytes actually transferred**

If a new READ (or SPACE or VERIFY) command is issued after a non-recoverable read error has occurred, the read operation will begin with the physical tape block following the erroneous physical tape block. If this block was located in the middle of a variable block and the residual count after the last READ command was not zero, this new READ command will read the remaining bytes of the previous (truncated) variable block. This means that the length of this rest-block will be equal to the original length minus the length actually transferred in the previous READ command (including any dummy bytes). This mechanism has the effect of splitting a variable length block with a non-recoverable physical tape block in the middle, into two variable blocks with total length equal to the total length of the original variable block. The first part of the block will be transferred with 1024 dummy (random) bytes as the last bytes.

2) **Bad Block Does NOT Hold Requested Data**
This situation may arise when the length of the requested block is less than the length of the actual block found on the tape. Normally, the Drive will just skip the data that is not requested for transfer. If, however, there is more bad blocks located in this data, the Drive will stop in front of the first bad block. The next read command will then start with this bad block and it will be terminated with CHECK CONDITION as described in under 1).

### 18.3.8. Illegal Termination

In addition to the error conditions described in Section 18.3.7. there is a condition that may indicate some kind of media error. When the Drive can see no more data on a tape, it always checks that the last block is followed by a correct postamble and a correct erase gap before reporting End Of Data (Blank Check). If the postamble or erase gap is bad, it may be because some data blocks have been lost after the last Read Good Blocks. This situation will lead to a series of re-reads to recover possible marginal data. If the re-read operation succeeds, the Read operation will continue in a normal manner. If the re-read operation fails, the READ command is terminated with CHECK CONDITION status. The Error Code will be set to E$TEM_ILTERM.
The Valid (VADD) bit in the sense data list will be set to one. The Information Bytes will be set to the difference (residue) between the requested transfer length and the actual transfer length (with good data). It will not be possible to continue the Read operation.

When a variable length block has been truncated due to an Illegal Termination error, Illegal Length is not signalled. Instead the Illegal Termination error takes priority and the Error Code is set to E$TEM_ILTERM.

### 18.3.9. SLR5 4.0/8.0GB and SLR4 2.5/5.0GB (TDC 4222) Only - Data Compression Exception Handling

If an illegal Compression Header is found during decompression, the Drive will terminate the READ command with CHECK CONDITION status. The Error Code will be set to E$BHI_CPHD.

If an unsupported compression algorithm is detected, the Drive will terminate the command with CHECK CONDITION status. The Error Code will be set to E$BTD_SALG or E$BTD_LALG depending on whether the algoritm code is higher or lower than 00 00 00 FFh.
In the case of E$BTD_SALG, the illegal compression algorithm may be obtained through the Compression and Decompression Algorithm Fields returned in response to a MODE SENSE current Data Compression Parameter Page command.

If an unrecoverable decompression error is encountered, the Drive will terminate the READ command with CHECK CONDITION status. The Error Code will be set to E$DCM_MISC.

### 18.3.10. SLR5 4.0/8.0GB and SLR4 2.5/5.0GB (TDC 4222) Only - Reading From the Beginning of the Directory Partition

Reading from the start of the directory partition will make the Drive do a pre-read on the data partition to evaluate the current data compression settings with respect to those found on the tape.

*This Page Intentionally Left Blank*

# 19. Read Block Limits

## 19.1. Command Description

The READ BLOCK LIMITS command requests that the Drive's capability for block length limits be returned. The READ BLOCK LIMITS Parameter List will be returned during the DATA-IN phase of the command.

If disconnection is allowed, the Drive will disconnect when executing this command if the previous command was an immediate type command.

## 19.2. Command Descriptor Block

| BYTE | BIT 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 00 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| 01 | Logical Unit Number (LUN) | | | RESERVED | | | | |
| 02 | RESERVED | | | | | | | |
| 03 | RESERVED | | | | | | | |
| 04 | RESERVED | | | | | | | |
| 05 | Control Byte | | | | | | | |

*Table: READ BLOCK LIMITS Command Block*

## 19.3. Parameter List

| BYTE | BIT 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 00 | RESERVED | | | | | | | |
| 01 | Maximum Block Length | | | | | | | |
| 02 | | | | | | | | |
| 03 | | | | | | | | |
| 04 | Minimum Block Length | | | | | | | |
| 05 | | | | | | | | |

*Table: READ BLOCK LIMITS Data*

**Maximum Block Length**

This field shows the maximum length of a variable length block. The actual value depends on the current tape format:

| | Tape Format | Maximum Block Length |
|---|---|---|
| ***NOT for SLR5 Series*** | QIC-24 | 000200h (512) |
| | QIC-120 | 008000h (32768) |
| | QIC-150 | 008000h (32768) |
| | QIC-525 | FFFFFFh (16777215) |
| | QIC-1000 | FFFFFFh (16777215) |
| | QIC-2GB | FFFFFFh (16777215) |
| ***SLR5 Series only*** | QIC-4GB | FFFFFFh (16777215) |

*Table: Maximum Block Size*

After a Reset or a cartridge removal/change, the current tape format is the default format or the format configured by the last Mode Select. After a Write or Read/Space type command has been executed on the tape, the current tape format is the format on the currently written/read tape.

**Minimum Block Length**

This field shows the minimum length of a variable block. The actual value depends on the current tape format:

| | Tape Format | Minimum Block Length |
|---|---|---|
| ***NOT for SLR5 Series*** | QIC-24 | 0200h (512) |
| | QIC-120 | 0001h (1) |
| | QIC-150 | 0001h (1) |
| | QIC-525 | 0001h (1) |
| | QIC-1000 | 0001h (1) |
| | QIC-2GB | 0001h (1) |
| ***SLR5 Series only*** | QIC-4GB | 0001h (1) |

*Table: Minimum Block Size*

After a Reset or a cartridge removal/change, the current tape format is the default format or the format configured by the last Mode Select. After a Write or Read/Space type command has been executed on the tape, the current tape format is the format on the currently written/read tape.

## 19.4. Exception Handling

See sections on Error Conditions For All Commands and Deferred Errors.

# 20. Read Buffer

## 20.1. Command Description

The READ BUFFER command is used in conjunction with the WRITE BUFFER command as a diagnostic function for testing the Drive's data buffer and the SCSI-bus integrity.

The READ BUFFER command may also be used to transfer the Drive's microcode and some data from the CPU memory map.

This command will not alter the status of a possible inserted tape cartridge in any way. However, a READ BUFFER - Microcode store command may write over tape related data already present in the data buffer (read ahead data after a READ command or data not written after a buffered Write command).

The READ BUFFER parameter list will be returned during the DATA-IN phase of the command.

If disconnection is allowed, the Drive may disconnect when executing this command. When transferring data, the total data transfer will be split into smaller bursts with a maximum size. The maximum burst size (the amount of data transferred between reconnects/disconnects) is controlled by the bus ratio/threshold parameters set up by the MODE SELECT command (just as for the READ command).

## 20.2. Command Descriptor Block

| BYTE | BIT 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 00 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 |
| 01 | Logical Unit Number (LUN) | | | RESERVED | | Mode | | |
| 02 | Buffer ID | | | | | | | |
| 03 | Buffer Offset | | | | | | | |
| 04 | | | | | | | | |
| 05 | | | | | | | | |
| 06 | Allocation Length | | | | | | | |
| 07 | | | | | | | | |
| 08 | | | | | | | | |
| 09 | Control Byte | | | | | | | |

*Table: READ BUFFER Command Block*

**Mode**

This field controls the function of the READ BUFFER command. It also controls the meaning of the other fields within this command descriptor block. The following modes are supported:

| Mode | Description |
|---|---|
| 000b | Read Combined Header and Data |
| 010b | Read Data |
| 011b | Read Descriptor |

***SLR5 4.0/8.0GB and SLR4 2.5/5.0GB (TDC 4222)***

The data returned is always uncompressed, independent of the data compression configuration.

**Buffer ID**

This field is used to select one out of several buffers in the Drive. The Buffer ID field can be used in mode 010b (Read Data) and mode 011b (Read Descriptor). The Buffer ID field is reserved in mode 000b (Read Combined Header and Data). The following Buffer IDs are supported:

| Buffer ID | Description |
|---|---|
| 0 | Data Buffer |
| 1 | CPU Memory |
| 2 | Hardware Registers |
| 3 | Internal RAM |
| 4 | EEPROM |
| 5 | External RAM |
| 6 | Microcode Store |

**Buffer Offset**

The Buffer Offset field specifies an offset into the buffer given by the Buffer ID field. The Buffer Offset is always a *byte* offset into the buffer. If the Buffer Offset is set to *N*, then the first data byte transferred by the READ BUFFER command will be byte *N* relative to the first available byte of the specified buffer. If the offset reaches beyond the end of the buffer specified by the Buffer ID field (the value in the Buffer Offset field is equal to or larger than the size of the specified buffer), the READ BUFFER command will transfer zero (no) bytes. The Buffer Offset field can only be used in mode 010b (Read Data). The Buffer Offset field is reserved in other modes.

**Allocation Length**

This field specifies the maximum number of bytes that the Drive is requested to return during the DATA IN phase of the command. Note that in mode 000b (Read Combined Header and Data) Allocation Length specifies the sum of Header and Data bytes. If Allocation Length is zero, the Drive will return no data. The Drive terminates the DATA IN phase when Allocation Length bytes have been transferred or when all the available data from the buffer has been transferred to the Initiator, whichever is less.

## 20.3. Read Combined Header and Data Mode (000b)

In this mode a four byte header followed by data are returned to the Initiator during the DATA IN phase. The Buffer ID and Buffer Offset fields are reserved. The READ BUFFER command will transfer data from the Drive's data buffer. Note that this is the buffer that is normally used to store data going to or coming from the media.

### 20.3.1. Header List

| BYTE | BIT 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 00 | RESERVED | | | | | | | |
| 01 | Buffer Capacity | | | | | | | |
| 02 | | | | | | | | |
| 03 | | | | | | | | |

*Table: READ BUFFER Header List*

**Buffer Capacity** The Buffer Capacity field specifies the total number of data bytes that are available in the Drive's data buffer - not inluding the header. This number is not reduced to reflect the Allocation Length nor is it reduced to reflect the actual number of bytes written using the WRITE BUFFER command. Note that the four header bytes are transferred before transferring the buffer bytes. The buffer will have a total of Buffer Capacity (245760) bytes available. This means that the READ BUFFER command in mode 0 is able to transfer up to Buffer Capacity + 4 (245764) bytes including the Header List.

### 20.3.2. Data List

Following the READ BUFFER Header, the Drive will transfer data from its data buffer. The first byte transferred will be the byte found at buffer address 0.

## 20.4. Read Data Mode (010b)

In this mode, the DATA-IN phase contains data only (no header). The Buffer ID field identifies a specific buffer within the Drive from which data will be transferred.

### 20.4.1. Data Buffer (Buffer ID = 0)

The READ BUFFER command will transfer data from the Drive's data buffer. Note that this is the buffer that is normally used to store data going to or coming from the media. The data transferred will be the same as in mode 000b (Read Combined Header and Data). Note, however, that the first four data bytes of the data buffer are also transferred and that the upper 16K parameter buffer is also available. The Drive will have a total of 262144 data bytes available.

### 20.4.2. CPU Memory (Buffer ID = 1)

The READ BUFFER command will transfer the contents of its CPU Memory. This data is a combination of Hardware Registers, Internal RAM, EEPROM and External RAM (in this order):

| Data Transferred From | Number of Bytes |
|---|---|
| Hardware Registers | 152 |
| Internal RAM | 928 |
| EEPROM | 512 |
| External RAM | 8192 |

Using Buffer ID 1 makes it possible to transfer the data associated with Buffer IDs 2 to 5 in a single READ BUFFER command. See the following sections for a description of the sub-components. The Drive will have a total of 9784 data bytes available.

### 20.4.3. Hardware Registers (Buffer ID = 2)

The READ BUFFER command will transfer the contents of its hardware registers (at the time the READ BUFFER command was received). The register data will be transferred in the following order:

| Data Transferred From | Number of Bytes |
|---|---|
| CPU | 96 |
| SCSI Controller | 16 |
| Drive Controller | 8 |
| EDC Controller | 32 |

This data is mainly intended for diagnostics/debugging purposes. The Drive will have a total of 152 data bytes available.

### 20.4.4. Internal RAM (Buffer ID = 3)

The READ BUFFER command will transfer the contents of its Internal RAM. This data is mainly intended for diagnostics/debugging purposes. The Drive will have a total of 928 data bytes available.

### 20.4.5. EEPROM (Buffer ID = 4)

The READ BUFFER command will transfer the contents of its EEPROM. This data is mainly intended for diagnostics/debugging purposes. The Drive will have a total of 512 data bytes available.

### 20.4.6. External RAM (Buffer ID = 5)

The READ BUFFER command will transfer the contents of its External RAM. This data is mainly intended for diagnostics/debugging purposes. The Drive will have a total of 8192 data bytes available.

### 20.4.7. Microcode Store (Buffer ID = 6)

The READ BUFFER command will transfer data from its microcode store. The Drive will have a total of 131076 data bytes available. These bytes will be copied into the data buffer before the data is transferred to the initiator. Tape related data present in the data buffer may therefore be overwritten. The data returned is in a format that makes it suitable for use with the WRITE BUFFER in Download Microcode and Save mode. See the WRITE BUFFER command for details.

## 20.5. Read Descriptor Mode (011b)

In this mode, a maximum of four bytes of READ BUFFER descriptor information are returned. The Drive will return the descriptor information for the buffer specified by the Buffer ID (see the description of the Buffer ID in Section 20.4). The Drive will return all zeros in the READ BUFFER descriptor if there is no buffer associated with the specified Buffer ID. The Buffer Offset field is reserved in this mode. The Allocation Length should be set to four or greater. The Drive will transfer the lesser of the Allocation Length or four bytes of READ BUFFER descriptor.

| BYTE | BIT 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 00 | Offset Boundary | | | | | | | |
| 01<br>02<br>03 | Buffer Capacity | | | | | | | |

*Table: READ BUFFER Descriptor List*

**Offset Boundary** The Offset Boundary field will always be zero. This means that the buffer offset in the READ BUFFER command descriptor block can be on any byte boundary.

**Buffer Capacity**

The value in the Buffer Capacity field depends on the Buffer ID:

| Buffer ID | Associated Buffer | Buffer Capacity |
|---|---|---|
| 0 | Data Buffer | 262144 |
| 1 | CPU Memory | 9784 |
| 2 | Hardware Registers | 152 |
| 3 | Internal RAM | 928 |
| 4 | EEPROM | 512 |
| 5 | External RAM | 8192 |
| 6 | Microcode Store | 131076 |

## 20.6. Exception Handling

See sections on Error Conditions For All Commands, Deferred Errors and Buffer Parity Errors.

If the Mode is not in the set of legal modes, the Drive will terminate the command with CHECK CONDITION. No data will be transferred. The Drive Error Code will be set to E$STE_IFIC.

In mode 000b (Read Combined Header and Data), if the Buffer ID or Buffer Offset fields are not set to zero, the Drive will terminate the command with CHECK CONDITION. No data will be transferred. The Drive Error Code will be set to E$STE_IFIC.

In mode 010b (Read Data), if the value in the Buffer ID field is not in the of legal values, the Drive will terminate the command with CHECK CONDITION. No data will be transferred. The Drive Error Code will be set to E$STE_IFIC.

In mode 011b (Read Descriptor), if the Buffer Offset field is not set to zero, the Drive will terminate the command with CHECK CONDITION. No data will be transferred. The Drive Error Code will be set to E$STE_IFIC.

# 21. Read Position

## 21.1. Command Description

The READ POSITION command requests the Drive to return a special position key that identifies the current logical position in the data stream on the tape. While writing (or reading) the READ POSITION command is typically executed every time the tape is at a position that the host system might want to go back to at a later time. The returned position key can then be stored and used as an input to the LOCATE command later. The LOCATE command will then bring the tape back to the same (logical) position as it was when the READ POSITION command was executed.

Note that reading, spacing or writing to the same logical position on the tape, and then issuing a READ POSITION command, may give different position keys.
The logical positions can either be given as physical tape block addresses (device-specific values) or as SCSI logical block addresses relative to the beginning of a partition. This is controlled by the Block address Type bit (BT) in the CDB (refer to the description of this bit).
The SCSI logical block address is found by counting the number of data blocks (*not* Filemarks and Setmarks) from the beginning of a partition.

The READ POSITION command requests the Drive to return the current position of data blocks on both the SCSI-bus side of the data buffer and on the tape side of the data buffer. When the buffer does not contain a whole block of data, or is empty, the two values are equal. (Refer to the description of the Last Block Location parameter).

Note that when the READ POSITION command returns block positions as physical tape block addresses, two SCSI blocks might have the same physical address when the tape format is QIC-525/1000/2GB/96-34 and the current block size is 512 bytes. To be able to assign unique addresses to all SCSI blocks, the READ POSITION command will during Write, if necessary, pad the last half of a QIC-525/1000/2GB/96-34 (1024 bytes long) physical tape block and move the tape position to the start of the next physical tape block before returning any block positions.

The READ POSITION parameter list is always 20 bytes long and it is returned during the DATA IN phase of the command.

The Drive will also disconnect if no Read or Write operations have been performed and the cartridge is not physically loaded. This is done for setting up correct Block Location according to cartridge type.

## 21.2. Command Descriptor Block

| BYTE | BIT 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 00 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 |
| 01 | Logical Unit Number (LUN) | | | RESERVED | | | | BT |
| 02 | RESERVED | | | | | | | |
| 03 | RESERVED | | | | | | | |
| 04 | RESERVED | | | | | | | |
| 05 | RESERVED | | | | | | | |
| 06 | RESERVED | | | | | | | |
| 07 | RESERVED | | | | | | | |
| 08 | RESERVED | | | | | | | |
| 09 | Control Byte | | | | | | | |

*Table: READ POSITION Command Block*

**BT**

The Block address Type (BT) bit can be set to either 1 or 0. When set to 1 the logical position will be given as a physical tape block address (device-specific value). When set to 0 the logical position will be given as a SCSI logical block address

## 21.3. Parameter List

| BYTE | BIT 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 00 | BOP | EOP | RESERVED | | | BPU | RESERVED | |
| 01 | Partition Number | | | | | | | |
| 02 | RESERVED | | | | | | | |
| 03 | RESERVED | | | | | | | |
| 04 | First Block Location | | | | | | | |
| 05 | | | | | | | | |
| 06 | | | | | | | | |
| 07 | | | | | | | | |
| 08 | Last Block Location | | | | | | | |
| 09 | | | | | | | | |
| 10 | | | | | | | | |
| 11 | | | | | | | | |
| 12 | RESERVED | | | | | | | |
| 13 | RESERVED | | | | | | | |
| 14 | | | | | | | | |
| 15 | | | | | | | | |
| 16 | RESERVED | | | | | | | |
| 17 | | | | | | | | |
| 18 | | | | | | | | |
| 19 | | | | | | | | |

*Table: READ POSITION Header List*

**BOP**

A beginning of partition (BOP) bit indicates that the current logical tape position is at the beginning-of-partition. A possible read command will read the very first block on the current partition. A possible write command will start writing the very first block on the current partition.

**EOP**

An end of partition (EOP) bit of one indicates that the current logical tape position is located between the Early Warning (EW) tape marker and the end-of-partition. A possible read or write command will terminate immediately with CHECK CONDITION due to the end-of-partition condition.

**BPU**

The Block Position Unknown (BPU) bit is used to indicate whether or not the position is known. When Physical Block Addressing is selected (BT=1), BPU will always be set to 0 to indicate that the position is always known.

When SCSI logical block addressing is selected (BT=0), BPU may be set to 1 to indicate that the position is unknown. This will happen in the following situations:

1: If no Tape Map (filler blocks holding information used by the Fast Space algorithm) is found on the tape during a SPACE to EOR
2: If the Drive has spaced over bad blocks
3: If bad blocks are detected
4: If an error message other than E$TEM_EOR is reported during a SPACE to EOR

If none of these situations occur, the position will be known and BPU will be set to 0 to indicate this. (Refer to the DTM1 and DTM2 bits of the MODE SELECT command for writing of Tape Maps).

NOTE:
If Tape Maps do not exist on a tape, the tape must be moved in serpentine mode from the beginning of a partition if the SCSI logical block address should be known. If, however, a command is issued (e.g. SPACE to EOR) that moves the tape to an area where a Tape Map is needed, the position will be lost and BPU will be set to 1.

**Partition Number**

The partition number indicates the current partition number. If the Drive is in QFA mode, the partition number returned may be 0 (data partition) or 1 (directory partition). If the Drive is not in QFA mode the returned value will always be 0.

**First Block Location**

The first block location field indicates the position of the next data block to be transferred between the Initiator and the Drive's data buffer on the next COPY, READ, VERIFY or WRITE command. When BT=1 the values returned are physical block identifiers (the actual tape block addresses as specified in [4]). These values can be looked upon as keys that are unique for any given logical position on the tape (the position in the data stream). These values must not be manipulated in any way by the host system. The position numbers should only be used as inputs to the LOCATE command. When the tape is positioned at BOT the value returned will be 0h when the current tape format is QIC-525/1000/2GB/96-34 and 1h when the current tape format is QIC-120 or QIC-150. When BT=0 the value returned will be the SCSI logical block number of the next block to be transferred. The SCSI logical block number is the number of blocks seen on the SCSI-bus counted from BOP. If at BOP, the value returned will always be 0 when SCSI logical block addressing is selected.

**Last Block Location**

The last block location field indicates the position of the next data block to be transferred between the Drive's data buffer and the tape on the next COPY, READ, VERIFY or WRITE command. When BT=1 the values returned are physical block identifiers (the actual tape block addresses as specified in [4]). These values can be looked upon as keys that are unique for any given logical position on the tape (the position in the data stream). These values should not be manipulated in any way by the host system. The position numbers should only be used as inputs to the LOCATE command. When the tape is positioned at BOT the value returned will be 0h when the current tape format is QIC-525/1000/2GB/96-34 and 1h when the current tape format is QIC-120 or QIC-150. When BT=0 the value returned will always be 0 because the Drive does not update this parameter when SCSI logical block positioning is selected.

## 21.4. Exception Handling

See sections on Error Conditions For All Commands and Deferred Errors.

If no cartridge is inserted, the READ POSITION will be terminated with CHECK CONDITION status. No parameter list will be transferred. The Error Code will be set to E$STE_NCAR.

If BT=1 and the currently inserted tape cartridge is blank, or the tape format is unknown, the Read Position command will return First Block Location = 0h and Last Block Location = 0h if DC6320/DC6525-type tapes or better. For lower capacity tape types the Read Position command will return First Block Location = 1h and Last Block Location = 1h. If BT=0 and the currently inserted tape cartridge is blank, the Read Position command will always return both First and Last Block Locations = 0h.

*This Page Intentionally Left Blank*

# 22. Recover Buffered Data

## 22.1. Command Description

The RECOVER BUFFERED DATA command is used to read back data that has been transferred to the Drive's data buffer but has not been written to the tape. It is normally used to recover from error or exception conditions that make it impossible to write the buffered data to the tape.

The Drive will only accept the Recover Buffered Data command in the following situations:

- Directly after a Write or Write Filemark command that terminated with a fatal error.
- Directly after another Recover Buffered Data command. (Sequential Recover Buffered Data commands are allowed).

The recovered data will be transferred during the DATA-IN phase of the command.

This command functions similarly to the READ command (see READ Section) except that the data is transferred from the Drive's data buffer instead of from the tape. The order in which block(s) are transferred is the same as if they had been transferred from the tape. One or more RECOVER BUFFERED DATA commands may be used to read the unwritten buffered data.

Upon termination of a successful RECOVER BUFFERED DATA command, the logical buffer position will be after the last block (fixed or variable) transferred (end-of-buffer side).

If disconnection is allowed, the Drive will disconnect when executing this command if the previous command was an immediate type command.

## 22.2. Command Descriptor Block

| BYTE | BIT 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 00 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| 01 | Logical Unit Number (LUN) | | | RESERVED | | | SILI | FIX |
| 02 | Transfer Length | | | | | | | |
| 03 | | | | | | | | |
| 04 | | | | | | | | |
| 05 | Control Byte | | | | | | | |

*Table: RECOVER BUFFERED DATA Command Block*

**FIX**

A Fixed (FIX) bit of zero indicates that a single block will be transferred with the Transfer Length specifying the maximum number of bytes the Initiator has allocated for the returned data. A FIX bit of one indicates that the Transfer Length specifies the number of blocks to be transferred to the Initiator.

**SILI**

If the Suppress Incorrect Length Indicator (SILI) bit is one and the FIX bit is zero, the Drive will not report CHECK CONDITION when an incorrect length block is found and the only error is that the requested transfer length exceeds the actual block length. If, however, the requested transfer length is less than the actual block length, the Drive will report CHECK CONDITION *even* if the SILI bit is set.

**Transfer Length**

This field specifies the number of bytes or blocks requested for transfer. Any value in the range 0..16777215 is legal both in fixed and variable block mode.

## 22.3. Exception Handling

### *22.3.1. General*

See sections on Error Conditions For All Commands, Deferred Errors and Buffer Parity Errors.

When the RECOVER BUFFERED DATA command has started execution, all errors will set the VADD bit and the Information Bytes will hold the difference between the requested and the actual transfer length. See the following sections for more details.

If both the SILI and the FIX bit is one, the Drive will terminate the RECOVER BUFFERED DATA command with CHECK CONDITION status. The Error Code will be set to E$STE_IFIC.

### *22.3.2. Command Sequencing*

The Drive will only accept the RECOVER BUFFERED DATA command in the following situations:

- Directly after a Write or Write Filemark command that terminated with a fatal error.
- Directly after another RECOVER BUFFERED DATA command. (Sequential Recover Buffered Data commands are allowed).

If the RECOVER BUFFERED DATA command is issued in one of the situations listed above, the Drive will accept the command and start to execute it.
If the data buffer contains the requested data, the Drive will transfer data back to the Host and respond with OK (no CHECK CONDITION).
If the data buffer contains less data than requested, the Drive will transfer the available data and report a CHECK CONDITION and Error Code E$STE_REOB. The Information Bytes will contain the residue.
Any subsequent Write commands will result in CHECK CONDITION with Error Code E$BTD_WRRD (Write After Read).
Subsequent Read or Space commands will result in Error Code E$BTD_-RDWR (Read After Write).

If the RECOVER BUFFERED DATA command is issued in any other situations than after a failing Write/Write Filemark command or a RECOVER BUFFERED DATA command, the Drive will respond with CHECK CONDITION and Error Code E$STE_CSEQ (Command Sequence Error) and ignore the command.

### 22.3.3. Filemark Detected

If a buffered filemark is encountered during execution of the RECOVER BUFFERED DATA command, the command will transfer all data blocks up to the filemark. The RECOVER BUFFERED DATA command will then be terminated with CHECK CONDITION status. The filemark (FMK) bit will be set to one and the Error Code will be set to E$BTD_FIMK. The Valid (VADD) bit in the sense data list will be set to one. The Information Bytes will be set to the difference (residue) between the requested transfer length and the actual transfer length (bytes or blocks). When the command has terminated, the buffer positions will be located after the filemark.

### 22.3.4. Illegal Length

#### 22.3.4.1. FIX Bit Set to ZERO

If the actual block length (length of block found in the buffer) is different from the specified transfer length, the illegal length block will first be transferred to the Initiator. The Drive will, however, not transfer more data than specified in the Transfer Length of the Command Descriptor Block. If the SILI bit is zero, the RECOVER BUFFERED DATA command will then be terminated with CHECK CONDITION status. The Error Code will be set to E$STE_ILLN. The illegal length indicator (ILI) and Valid (VADD) bits in the sense data list will be set to one. The Information Bytes will be set to the difference (residue) of the requested transfer length minus the actual block length. If the actual block length was smaller than the specified length, the residual will be a positive number. If the actual block length was larger than the specified length, the residual will be a negative number. Negative residues will be presented on 2's complement form. A SILI bit of one indicates that the Drive will not return CHECK CONDITION status if the only error is that the transfer length exceeds the actual block length found in the data buffer. When the command has terminated, the logical buffer position will be located after the incorrect length block (end of buffer side).

#### 22.3.4.2. FIX Bit Set to ONE

If the actual block length (length of block found in the data buffer) is different from the configured block length, the illegal length block will first be transferred to the Initiator. The Drive will, however, not transfer more data than configured as the block length (see MODE SELECT command). The RECOVER BUFFERED DATA command will then be terminated with CHECK CONDITION status. The Error Code will be set to E$STE_ILLN. The illegal length indicator (ILI) and Valid (VADD) bits in the sense data list will be set to one. The Information Bytes will be set to the difference (residue) of the requested transfer length minus the actual number of blocks transferred (not counting the illegal length block). The block with the unexpected length is not counted among the transferred blocks. When the command has terminated, the logical buffer position will be located after the incorrect length block (end of buffer side).

### 22.3.5. End of Buffer

If an attempt is made to recover more data than are contained in the Drive's data buffer, the command will transfer all data block(s) up to the end of the buffer. The RECOVER BUFFERED DATA command will then be terminated with CHECK CONDITION status. The Error Code will then be set to E$STE_REOB. The End Of Media (EOM) and Valid (VADD) bits in the sense data list will be set to one. The Information Bytes will be set to the difference (residue) between the requested transfer length and the actual transfer length (bytes or blocks).

# 23. Release Unit

## 23.1. Command Description

The RELEASE UNIT command will release the Drive if it is currently reserved by the requesting Initiator.

It is not an error to attempt to release the Drive if it is not currently reserved to the requesting Initiator. However, the Drive will not be released if it is reserved by another Initiator (the RELEASE command will just be ignored).

The third-part release option allows an Initiator to release the Drive if it was previously reserved using the third-party reservation option (see RESERVE UNIT Section). This option is intended for use in multiple-initiator systems that use the COPY command.

If disconnection is allowed, the Drive will disconnect when executing this command if the previous command was an immediate type command.

## 23.2. Command Descriptor Block

| BYTE | BIT 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 00 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 |
| 01 | Logical Unit Number (LUN) | | | 3RD | 3RD Party ID | | | R |
| 02 | RESERVED | | | | | | | |
| 03 | RESERVED | | | | | | | |
| 04 | RESERVED | | | | | | | |
| 05 | Control Byte | | | | | | | |

*Table: RELEASE UNIT Command Block*

**3RD** If the third-party (3RD) bit is zero, then the third-party release option is not requested. If the 3RD bit is one, then the Drive will release itself, but only if the reservation was made using the third-party reservation option by the Initiator that is requesting the release.

**3RD Party ID** This field specifies the ID of the third-party device. This field will be ignored if the 3RD bit is not set to one.

## 23.3. Exception Handling

See sections on Error Conditions For All Commands and Deferred Errors

If the third party (3RD) bit is one, the third part ID (3RD Party ID) is equal to the Drive's ID and the Drive has been reserved by the requesting Initiator, the RELEASE UNIT command will be terminated with CHECK CONDITION status. The Error Code will be set to E$STE_IFIC.

If the third party (3RD) bit is one, the third part ID (3RD Party ID) is equal to the Drive's ID and the Drive has been reserved by another Initiator, the RELEASE UNIT command will be ignored.

# 24. Request Sense

## 24.1. Command Description

The REQUEST SENSE command requests that the Drive transfer sense data to the Initiator.

The sense data will be valid for a CHECK CONDITION returned on the prior command. This sense data will be preserved by the Drive for the Initiator until retrieved by the REQUEST SENSE command or until the receipt of any other command from the Initiator that issued the command resulting in the CHECK CONDITION status. Sense data will be cleared upon receipt of any subsequent command to the Drive from the Initiator receiving the CHECK CONDITION.

The REQUEST SENSE command will execute even if the Initiator specifies an unsupported LUN (LUN field in the Command Descriptor Block or IDENTIFY message is not set to zero). In this situation other pending sense data will be cleared and the transferred Parameter List will reflect the Unsupported LUN condition.

The REQUEST SENSE command will execute normally even if a reservation conflict exists.

The Drive may disconnect for this command.

## 24.2. Command Descriptor Block

| BYTE | BIT 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 00 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 01 | Logical Unit Number (LUN) | | | RESERVED | | | | |
| 02 | RESERVED | | | | | | | |
| 03 | RESERVED | | | | | | | |
| 04 | Allocation Length | | | | | | | |
| 05 | Control Byte | | | | | | | |

*Table: REQUEST SENSE Command Block*

**Allocation Length**

This field specifies the maximum number of bytes allocated by the Initiator for sense data. If the Allocation Length is zero, the Drive will not return any sense data. For any other Allocation Length value, the Drive terminates the DATA-IN phase when Allocation Length bytes have been transferred or when all available sense data have been transferred to the Initiator, whichever is less.

## 24.3. Parameter List

| BYTE | BIT 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 00 | VADD | Error Code | | | | | | |
| 01 | Segment Number | | | | | | | |
| 02 | FMK | EOM | ILI | R | Sense Key | | | |
| 03 | Information Bytes | | | | | | | |
| 04 | | | | | | | | |
| 05 | | | | | | | | |
| 06 | | | | | | | | |
| 07 | Additional Sense Length | | | | | | | |
| 08 | Source Sense Pointer | | | | | | | |
| 09 | Destination Sense Pointer | | | | | | | |
| 10 | RESERVED | | | | | | | |
| 11 | | | | | | | | |
| 12 | Additional Sense Code | | | | | | | |
| 13 | Additional Sense Code Qualifier | | | | | | | |
| 14 | RESERVED | | | | | | | |
| 15 | SKSV | Sense Key Specific | | | | | | |
| 16 | Sense Key Specific | | | | | | | |
| 17 | | | | | | | | |
| 18 | Block Counter | | | | | | | |
| 19 | | | | | | | | |
| 20 | | | | | | | | |
| 21 | Filemark Counter | | | | | | | |
| 22 | | | | | | | | |
| 23 | Underrun Counter | | | | | | | |
| 24 | | | | | | | | |
| 25 | Number of Recoverable Errors | | | | | | | |
| 26 | | | | | | | | |
| 27 | ECC Correction Counter | | | | | | | |
| 28 | | | | | | | | |
| 29 | FOR INTERNAL USE | | | | | | | |
| 30 | Source/Destination Status Byte | | | | | | | |
| 31 | Source/Destination Status Byte 0 | | | | | | | |
| 32 | Source/Destination Status Byte 1 | | | | | | | |
| .. | ............................. | | | | | | | |
| .. | ............................. | | | | | | | |
| xx *) | Source/Destination Status Byte n | | | | | | | |

*) NOTE:
xx = 30+n, where n is the number of sense bytes transferred from a source/destination device during COPY.

*Table: REQUEST SENSE Parameter List*

**VADD**

A Valid Address (VADD) bit of zero indicates that the Information Bytes are undefined. A VADD bit of one indicates that the Information Bytes contain valid information.

**Error Code**

Error Code hex 70 will be used for normal errors. Error Code hex 71 will be used for deferred errors.

**Segment Number**

This field contains the number of the current segment descriptor if the REQUEST SENSE command is in response to a COPY command error. Up to 256 segments are supported beginning with segment zero.

**FMK**

The Filemark (FMK) bit indicates that the current command has read a filemark.

**EOM**

The End Of Media (EOM) bit indicates that the last command encountered end-of-partition or beginning-of-partition.

**ILI**

The Incorrect Length Indicator (ILI) bit indicates that the requested logical block length did not match the logical block length found on the tape.

**Sense Key**

This field holds information about the cause of error. See Section 24.4 for a description of the Sense Key codes.

**Information Bytes**

The contents of the Information Bytes is command specific and is defined within the appropriate for the command of interest. Unless otherwise specified, this field contains:

- *The difference (residue) of the requested length minus the actual length in either bytes or blocks, as determined by the command. Negative values are indicated by two's complement notation.*
- *The difference (residue) of the requested number of blocks minus the actual number of blocks copied for the current segment descriptor of a COPY command.*

**Additional Sense Length**

This field specifies the number of additional sense bytes to follow. When the previous command was not a COPY command, Additional Sense Length will be set to 22. If the previous command was a COPY command, Additional Sense Length may be set to 22 (no source or destination device status or sense data), 23 (source or destination device status byte valid) or 23+n (both status and sense data valid for source or destination device). The number n is the number of sense data bytes actually transferred from the source/destination device (see also COPY Section).

If the Allocation Length of the Command Descriptor Block is too small to transfer all the additional sense bytes, the Additional Sense Length will not be adjusted to reflect the truncation.

**Source Sense Pointer**

This field is only valid after a data transfer error during a COPY operation. See COPY Section for details.

**Destination Sense Pointer**

This field is only valid after a data transfer error during a COPY operation. See COPY Section for details.

**Additional Sense Code**

This field holds additional error information. See Section 24.5 and the section on General Exception Handling.

**Additional Sense Code Qualifier**

This field holds additional error information. See also Section 24.5 and the section on General Exception Handling.

**SKSV**

A Sense Key Specific Valid bit of one indicates that the Sense Key Specific field is valid. A SKSV bit of zero indicates that the Sense Key Specific field is not valid.

**Sense Key Specific**

When the Error Code is E$STE_IFIC or E$STE_IFIP, this field holds pointers to the invalid bit in the Command Descriptor Block where CHECK CONDITION was signalled. The format is shown in the table below:

| BYTE | BIT 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 00 | SKSV | C/D | RESERVED | | BPV | Bit Pointer | | |
| 01<br>02 | Field Pointer | | | | | | | |

**C/D**

A Command/Data bit of one indicates that the illegal parameter is in the Command Descriptor Block. A C/D bit of zero indicates that the illegal parameter is in the data parameters sent by the Initiator during the DATA OUT phase.

**BPV**

A Bit Pointer Valid bit of zero indicates that the Bit Pointer field is not valid. A BPV bit of one indicates that the Bit Pointer field is valid.

**Bit Pointer**

The Bit Pointer field specifies the erroneous bit in the byte designated by the Field Pointer. When a multiple-bit field is faulty, the Bit Pointer field will point to the most significant (leftmost) bit of the field.

**Field Pointer**

The Field Pointer field indicates the erroneous byte of the Command Descriptor Block or of the Parameter Block. Bytes are numbered starting from zero.

### Block Counter

This field is a 24 bit counter that counts the number of data blocks transferred to/from the Initiator. A variable block (of any length) counts as one block. The counter is cleared when a transition is made from one mode (read, write or position) to read or write mode.

**Fixed/Variable Blocks:**

| | Write | Read | Space EOR (Always Fast) | Space Filemark Forward | Space Filemark Reverse | Space Block Forward | Space Block Reverse | Any Fast Space |
|---|---|---|---|---|---|---|---|---|
| **QIC-2GB/ 1000/525/ 120/150** | B F | B F | NID | B F | NID | B F | NID | NID |

**B** The Block Counter is incremented by one on the detection of each new block/start writing of a new block.
The counter is incremented even if the reading/spacing/writing operation of a block is not completed successfully, e.g. a bad block is detected.

**F** The Filemark Counter is incremented by one when a new filemark is detected or at the start of writing a filemark.

**NID** The Block Counter or Filemark Counter is not incremented or decremented.

**Special Conditions:**

✶ In the QIC-525/1000/2GB/96-34 format the Block Counter will not be incremented when reading the rest of a variable block after a bad block is detected within this block.

✶ The Drive will in the QIC-525/1000/2GB/96-34 format look upon a variable block containing a bad block as two variable blocks.

✶ If a Control Block is bad on a QIC-120/150 tape read by the Drive, the following would happen since a mix of fixed and variable blocks is legal with this format:

- All the physical blocks (512 bytes) in the variable block with the bad Control Block will be treated as fixed blocks. The Block Counter will increment by one, and if the Read command was a Read Variable Block command, Illegal Length will be reported.

**Filemark Counter**

This field counts filemarks received from the Initiator (during WRITE FILEMARKS operations) or sent to the Initiator (during read/verify operations). The counter is cleared when a transition is made from one mode (read, write or position) to read or write mode.

**Underrun Counter**

This counter counts Underruns (during write operations) or Overruns (during read/verify operations). The counter is cleared when a transition is made from one mode (read, write or position) to read or write mode.

**Number Of Recoverable Errors**

This field counts re-writes during WRITE operations and re-reads during READ/VERIFY operations. The counter is incremented by 1 for every block that is re-written (even if the same block is re-written more than once) or re-read (even if the same block is re-read more than once). The counter is cleared when a transition is made from one mode (read, write or position) to read or write mode.

**ECC Correction Counter**

This field counts the number of blocks corrected with ECC (during read or verify operations). The counter is cleared when a transition is made from one mode (read, write or position) to read or write mode.

**FOR INTERNAL USE**

This field is for Tandberg Data internal use only.

**Source/Destination Status Byte**

This is the status byte returned from the source/destination device when a data transfer error has occurred during execution of the COPY command (see COPY section).

**Source/Destination Sense Bytes**

These are the sense data bytes returned from the source/destination device when the COPY manager has requested sense data in response to a data transfer error during execution of a COPY command. Note that the number of valid bytes is a function of the Copy Sense Allocation field in the MODE SELECT parameter list (see COPY section).

## 24.4. Sense Keys

| Code | Name | Description |
|---|---|---|
| 0h | NO SENSE | Indicates that there is no specific sense key information to be reported. This would be the case for a successful command or a command that received a CHECK CONDITION status because of one if the FMK, EOM or ILI bits is set to one. |
| 1h | RECOVERED ERROR | When the PER (Post Error Recovery) bit in the Error Recovery Page of the MODE SELECT command is set to one, the Drive will terminate any command (except REQUEST SENSE) with a CHECK CONDITION status and a RECOVERED ERROR sense key if there has been any rereads or re-writes since the last command and no other errors has occurred. See the section on General Exception Handling for further details. |
| 2h | NOT READY | Indicates that the Drive medium cannot be accessed. This will be the case if there is no cartridge inserted or if it is unloaded, and a media access command is issued. |
| 3h | MEDIUM ERROR | Indicates that the command terminated with a unrecoverable error |
| 4h | HARDWARE ERROR | The Drive has detected a parity error or some fatal error in the Drive hardware |
| 5h | ILLEGAL REQUEST | Indicates that there was an illegal parameter in Command Descriptor Block or in the additional parameters supplied as data for some commands. |
| 6h | UNIT ATTENTION | Indicates that a cartridge has been inserted or that the Drive has been reset since the last command. The condition is cleared for the next command from the same Initiator (see UNIT ATTENTION Section for details) |
| 7h | DATA PROTECT | Indicates that a write operation has been attempted on write protected cartridge |
| 8h | BLANK CHECK | Indicates that a SEEK BLOCK, SPACE, READ or VERIFY operation encountered erased tape (end of the recorded area) |
| Ah | COPY ABORTED | Indicates that a COPY command was aborted due to an error condition in either the source or destination device (a data transfer error). See also COPY Section |
| Bh | ABORTED COMMAND | Indicates that the Drive aborted the command. The Initiator may be able to recover by trying the command again |
| Dh | VOLUME OVERFLOW | This condition occurs if additional data blocks are appended after the Drive has reported EOM, and there is not sufficient space left on the tape. The buffered nonwritten blocks can be read back by issuing a RECOVER BUFFERED DATA command (see also the WRITE and WRITE FILEMARK Sections) |
| Eh | MISCOMPARE | Indicates that the source data did not match the data read from the tape during execution of the VERIFY command |

## 24.5. Additional Sense Code and Qualifier

| AS | AQ | Description |
|---|---|---|
| 00h | 00h | No Additional Sense Information |
| 00h | 01h | Filemark Detected During READ or SPACE |
| 00h | 02h | End Of Partition Detected:<br>Pseudo Early Warning detected during WRITE or WRITE FILEMARKS.<br>*The physical end of the current partition has been detected during a execution of a READ, SPACE or LOCATE command*<br>Physical End Of Partition Encountered:<br>*The physical end of the active partition has been detected during execution of a WRITE or WRITE FILEMARK command* |
| 00h | 03h | Setmark Detected During READ or SPACE |
| 00h | 04h | Beginning Of Medium Detected:<br>*The physical beginning of the current partition has been detected during execution of a SPACE or LOCATE command* |
| 00h | 05h | End Of Data Detected:<br>*End of data has been detected during execution of a READ, SPACE or LOCATE command* |
| 03h | 02h | Excessive Write Errors:<br>*WRITE retries exhausted* |
| 04h | 01h | Logical unit is in process of becoming ready:<br>*A Write Buffer Mode 5 command has been attempted during a load/retension sequence* |
| 0Ch | 04h | Compression Check Miscompare:<br>*An unrecoverable compression or decompression error is encountered during a READ, SPACE or WRITE command (SLR5 4.0/8.0GB and SLR4 2.5/5.0GB (TDC 4222) only)* |
| 11h | 00h | Unrecovered Read Error:<br>*Read retries exhausted during execution of a SPACE command when spacing in the reverse direction* |
| 11h | 01h | Read Retries Exhausted:<br>*Uncorrectable data block found during READ, SPACE, LOCATE or VERIFY* |
| 14h | 00h | Recorded Entity Not Found:<br>*A READ, SPACE, VERIFY or LOCATE operation was attempted on a blank (erased) cartridge* |
| 17h | 01h | Recovered Error With Retries:<br>*There has been one or more re-read or re-write since the last command. The Drive only checks for this error when the PER (Post Error Recovery) bit in the Error Recovery Page of the MODE SELECT command is set to one.* |
| 1Ah | 00h | Parameter List Length Error:<br>*The supplied parameter list is too small or to large* |
| 1Dh | 00h | Miscompare During Verify Operation |
| 20h | 00h | Invalid Command Operation Code |
| 21h | 00h | Logical Block Address Out Of Range:<br>*The address specified for the direct access device in the Segment Descriptor List of a COPY command is too large* |
| 24h | 00h | Invalid Field In CDB:<br>*Sense Key Specific field is set up to point to the offending byte and bit* |
| 25h | 00h | Unsupported Logical Unit:<br>*The LUN field in the last IDENTIFY Message or last CDB is not set to zero* |
| 26h | 00h | Invalid Field In Parameter List:<br>*Sense Key Specific field is set up to point to the offending byte and bit* |
| 27h | 00h | Write Protected |
| 28h | 00h | Not Ready to Ready Transition:<br>*Unit Attention, a new cartridge has been inserted into the Drive* |
| 29h | 00h | Unit Attention, Power-Up, Reset or Bus Device Reset occurred |
| 2Ah | 01h | Mode Parameters Change:.<br>*Unit Attention, another Initiator has changed the Mode Parameters* |
| 2Bh | 00h | COPY Cannot Execute Since Host Cannot Disconnect |

*Table: Additional Sense Code And Qualifier (to be continued...)*

| AS | AQ | Description |
|---|---|---|
| 2Ch | 00h | Command Sequence Error:<br>*A READ, SPACE or LOCATE command cannot follow a WRITE or WRITE FILEMARKS command* |
| 2Dh | 00h | Overwrite Error On Update In Place:<br>*Cannot append data to an incomplete QIC-525/1000/2GB/96-34 frame* |
| 30h | 00h | Incompatible Medium Installed:<br>*Cannot write with the selected tape format on this cartridge.*<br>*Cannot append data on this cartridge with the selected tape format* |
| 30h | 01h | Cannot read, unknown tape format |
| 3Ah | 00h | Medium Not Present |
| 40h | NNh | Diagnostic Failure on Component NN. NN:<br>*80h : Data Buffer error*<br>*90h : Drive Controller error*<br>*9Ah :*<br>*9Bh :*<br>*9Ch : – Write or Erase Circuit HW-errors*<br>*9Dh :*<br>*9Eh :*<br>*9Fh :*<br>*A0h : EEPROM error*<br>*B0h : EDC Controller error*<br>*C0h : EPROM error*<br>*D0h : External RAM error*<br>*D1h : Selftest READ error*<br>*D2h : Selftest WRITE error*<br>*D3h : Selftest CPU error*<br>*E0h : Internal RAM error*<br>*F0h : SCSI Controller error*<br>*F8h : Data Compression Controller error (SLR5 4.0/8.0GB and SLR4 2.5/5.0GB (TDC 4222) only* |
| 44h | 00h | Internal Target Failure:<br>*The data transfer part of the Drive has signalled an error (a CHECK CONDITION status) during execution of the COPY command.* |
| 45h | 00h | Select/Reselect Failure:<br>*The selection of the COPY Target failed* |
| 47h | 00h | SCSI Parity Error |
| 48h | 00h | INITIATOR DETECTED ERROR Message Received |
| 4Ah | 00h | Command Phase Error:<br>*The COPY target did not have a STATUS or MESSAGE phase before going to the BUS FREE phase* |
| 4Eh | 00h | Overlapped Commands Attempted |
| 50h | 00h | Write Append Error:<br>*Cannot append data until Logical End Of Partition has been reached* |
| 50h | 01h | Write Append Position Error:<br>*The append failed because the last written block could not be found* |
| 52h | 00h | Cartridge Failure:<br>*No Tape Edge Found*<br>*Incorrect Tape Speed*<br>*Tape Runout*<br>*Cartridge Stuck*<br>*Cartridge Not Up To Speed* |
| 70h | FFh | Unsupported Short Compression Algorithm: *An unsupported Compression Short Algorithm is detected on the tape during a READ or SPACE command - the algorithm code is less or equal to FFh (SLR5 4.0/8.0GB and SLR4 2.5/5.0GB (TDC 4222) only* |
| 71h | 00h | Unsupported Long Compression Algorithm: *An unsupported Compression Long Algorithm is detected on the tape during a READ or SPACE command - the algorithm code is higher than FFh*<br>*(SLR5 4.0/8.0GB and SLR4 2.5/5.0GB (TDC 4222) only* |
| 80h<br>(Prelim.) | 00h | Illegal Compression Header: *Illegal Comp. Header is detected on the tape during a READ or SPACE command (SLR5 4.0/8.0GB and SLR4 2.5/5.0GB (TDC 4222) only)* |

*Table: Additional Sense Code And Qualifier*

## 24.6. Exception Handling

The REQUEST SENSE command will return the CHECK CONDITION status only to report fatal errors for the REQUEST SENSE command. Fatal errors are; non-zero bit in command descriptor or parity error on the data bus.

# 25. Reserve Unit

## 25.1. Command Description

The RESERVE UNIT command will reserve the Drive for exclusive use by the requesting Initiator or to another specified SCSI device.

The reservation will remain in effect until superseded by another RESERVE UNIT command from the Initiator that made the reservation or until released by a RELEASE UNIT command from the same Initiator, or a BUS DEVICE RESET message from any Initiator, or a SCSI-bus reset condition. It will not be an error to issue this command to the Drive if it is currently reserved to the requesting Initiator.

If the Drive is previously reserved by another Initiator, then the Drive will return RESERVATION CONFLICT status.

If, after honoring the reservation, any other Initiator then subsequently attempts to perform any command except INQUIRY, REQUEST SENSE or RELEASE UNIT, then the command will be rejected with RESERVATION CONFLICT status. A RELEASE UNIT command issued by another Initiator will be ignored by the reserved Drive.

The third-party reservation option allows an Initiator to reserve the Drive for another SCSI device. This option is intended for use in multiple-initiator systems that use the COPY command.

If the third-party reservation option is used (by setting the 3RD bit), then the RESERVE UNIT command will reserve the Drive for the SCSI device specified in the third-part device ID field (3RD Party ID). The Drive will preserve the reservation until superseded by another RESERVE UNIT command from the Initiator that made the reservation or until released by the same Initiator, by a BUS DEVICE RESET message from any Initiator, or by a SCSI-bus reset condition. The Drive will ignore (i.e., return GOOD status) any attempt made by any other Initiator to release the reservation.

An Initiator that holds a current reservation may modify that reservation (e.g., switch third-parties) by issuing another RESERVE UNIT command to the Drive. The superseding RESERVE UNIT command will release the previous reservation state only when the new reservation is granted.

If disconnection is allowed, the Drive will disconnect when executing this command if the previous command was an immediate type command.

## 25.2. Command Descriptor Block

| BYTE | BIT 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 00 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 |
| 01 | Logical Unit Number (LUN) | | | 3RD | 3RD Party ID | | | R |
| 02 | RESERVED | | | | | | | |
| 03 | RESERVED | | | | | | | |
| 04 | RESERVED | | | | | | | |
| 05 | Control Byte | | | | | | | |

*Table: RESERVE UNIT Command Block*

**3RD** If the third-party (3RD) bit is zero, then the third-party reservation option is not requested. If the 3RD bit is one, then the Drive will reserve itself for the SCSI device specified in the third-party device ID field (3RD Party ID).

**3RD Party ID** This field specifies the ID of the third-party device.

## 25.3. Exception Handling

See sections on Error Conditions For All Commands and Deferred Errors.

If the third party (3RD) bit is one and the third part ID (3RD Party ID) is equal to the Drive's ID, the RESERVE UNIT command will be terminated with CHECK CONDITION status. The Error Code will be set to E$STE_IFIC.

# 26. Rewind

## 26.1. Command Description

The REWIND command requests the Drive to rewind the tape to the beginning of the current partition.

Prior to the execution of the rewind operation, the Drive will write any buffered data that is to be written to the tape. If however, the previous command was terminated with CHECK CONDITION and the Drive is in buffered mode, then the Drive will discard any buffered data when a REWIND command has been validated

If disconnection is allowed, the Drive will disconnect when executing this command.

## 26.2. Command Descriptor Block

| BYTE | BIT 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 00 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 01 | Logical Unit Number (LUN) | | | RESERVED | | | | IMM |
| 02 | RESERVED | | | | | | | |
| 03 | RESERVED | | | | | | | |
| 04 | RESERVED | | | | | | | |
| 05 | Control Byte | | | | | | | |

*Table: REWIND Command Block*

**IMM**

An Immediate (IMM) bit of zero indicates that the Drive will not return status until the rewind operation has completed. An IMM bit of one indicates that the Drive will return status as soon as the execution of all previous commands have been completed and the Command Descriptor Block of the REWIND command has been validated. If CHECK CONDITION status is returned for the REWIND command with an IMM bit of one, the rewind operation will not be performed.

## 26.3. Exception Handling

See sections on Error Conditions For All Commands, Deferred Errors and Error Conditions For Media Access Commands.

If the IMM and Link bits are both set to one, the Drive will terminate the REWIND command with CHECK CONDITION status. The Error Code will be set to E$STE_IFIC.

*This Page Intentionally Left Blank*

# 27. Send Diagnostics

## 27.1. Command Description

The SEND DIAGNOSTICS command requests the Drive to perform diagnostic tests on itself.

The SEND DIAGNOSTICS parameter list is transferred during the DATA OUT phase of the command.

Note that the SEND DIAGNOSTICS command will destroy possible buffered data in the Drive's data buffer. Make sure that all data is written to the tape after a WRITE operation before calling the SEND DIAGNOSTICS command (use a WRITE FILEMARKS command specifying zero filemarks if the Drive is in buffered mode).

When a diagnostic test has executed successfully, the SEND DIAGNOSTICS command will return GOOD status. When a diagnostic test has failed, the SEND DIAGNOSTICS command will return CHECK CONDITION status. The REQUEST SENSE command can then be used to get further information on the error.

| | |
|---|---|
| ***SLR2 (TDC 3820) only*** | The tape format will always be set to QIC-525 for the TDC 3820 Drive, and a suitable tape type (e.g. DC 6525) should be used for this format. |
| ***SLR3 (TDC 4100) only*** | The tape format will always be set to QIC-1000 for the TDC 4100 Drive, and a suitable tape type (e.g. DC 9100) should be used for this format. |
| ***SLR4 (TDC 4200) Series*** | The tape format will always be set to QIC-2GB for the TDC 4200 Series Drives, and a suitable tape type (e.g. DC 9200) should be used for this format. |
| ***SLR5 Series only*** | The tape format will always be set to QIC 94-36 for the SLR5 Series Drives, and a suitable tape type (e.g. DC 9400) should be used for this format. |

See Sections 27.5. and 27.6. for a description of the actual tests performed by the SEND DIAGNOSTICS command.

If disconnection is allowed, the Drive will always disconnect when executing this command.

## 27.2. Command Descriptor Block

| BYTE | BIT 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 00 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 |
| 01 | Logical Unit Number (LUN) | | | PF | R | ST | DOF | UOF |
| 02 | RESERVED | | | | | | | |
| 03<br>04 | Parameter List Length | | | | | | | |
| 05 | Control Byte | | | | | | | |

*Table: SEND DIAGNOSTIC Command Block*

**PF**

The Page Format (PF) bit MUST be set zero since all parameters are vendor unique.

**ST**

See table: *Send Diagnostics Functions.*

**DOF**

See table: *Send Diagnostics Functions.*

**UOF**

See table: *Send Diagnostics Functions.*

**Parameter List Length**

This field specifies the length in bytes of the parameter list that will be transferred from the Initiator to the Drive. Legal values are 0 and 8. A Parameter List Length of zero indicates that no data will be transferred.

The following table shows the legal settings and the corresponding actions taken for the ST (SelfTest), DOF (DevOfl) and UOF (UnitOfl) bits:

| ST | DOF | UOF | Selftest Action |
|---|---|---|---|
| 0 | 0 | 0 | Illegal combination |
| 0 | 0 | 1 | Illegal combination |
| 0 | 1 | 0 | Illegal combination |
| 0 | 1 | 1 | Selftest 2 with parameters. The SEND DIAGNOSTICS command must be followed by a parameter list |
| 1 | 0 | 0 | Selftest 1 |
| 1 | 0 | 1 | Selftest 1 |
| 1 | 1 | 0 | Selftest 1 |
| 1 | 1 | 1 | Selftest 2 with default settings |

*Table: SEND DIAGNOSTICS Functions*

## 27.3. Parameter List

| BYTE | BIT 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 00 | LLD | Operation Type = 00h | | | | | | |
| 01 | Number of Tracks to Test | | | | | | | |
| 02 | Number of 512 Byte Blocks per File (logical) | | | | | | | |
| 03 | | | | | | | | |
| 04 | File Rewrite Limit | | | | | | | |
| 05 | Total Rewrite Limit | | | | | | | |
| 06 | File Reread Limit | | | | | | | |
| 07 | Total Reread Limit | | | | | | | |

*Table: SEND DIAGNOSTICS Command Block*

**LLD** The Low Level Debugging (LLD) bit MUST be set to zero.

**Operation Type** The Operation Type MUST be set to zero.

**No. of Tracks to Test**

| | |
|---|---|
| ***SLR2 (TDC 3820) only*** | This field must have a value in the range 0..26 for the TDC 3820 Drive. |
| ***SLR3 (TDC 4100) only*** | This field must have a value in the range 0..30 for the TDC 4100 Drive. |
| ***SLR4 (TDC 4200) Series*** | This field must have a value in the range 0..42 for the TDC 4200 Series Drives. |
| ***SLR5 Series*** | This field must have a value in the range 0..46 for the SLR5 Series Drives. |

If the value 0 is selected then the read/write part of the selftest is skipped. This means that only the Selftest 1 part of a Selftest 2 is actually performed.

**Number of 512 Byte Blocks per File (logical)** This field may have a value in the range 0..65525. If the value 0 is selected then the read/write part of the self test is skipped. This means that only the Selftest 1 part of a Selftest 2 is actually performed. If the number of blocks selected takes up more tracks than the Number Of Tracks To Test parameters specifies (may happen on short tapes), then the Number Of Tracks To Test takes priority.
One block contains 512 bytes of user data.

**File Rewrite Limit** This field may contain a value in the range 0...100 where 0 represents 0 % or no rewrites allowed, and 100 represents 100 % or that a rewrite on every block in each file is allowed.

**Total Rewrite Limit** This field may contain a value in the range 0...100 where 0 represents 0 % or no rewrites allowed, and 100 represents 100 % or that a rewrite on every block in all files is allowed.

**File Reread Limit** This field may contain a value in the range 0...100 where 0 represents 0 % or no rereads allowed, and 100 represents 100 % or that a reread on every block in each file is allowed.

**Total Reread Limit** This field may contain a value in the range 0...100 where 0 represents 0 % or no rereads allowed, and 100 represents 100 % or that a reread on every block in all files is allowed.

## 27.4. Default Test Parameters

The following table lists the parameter values used when a selftest with default parameters is performed.

| Parameter | Default Value |
|---|---|
| Number Of Tracks To Test | 2 |
| Number Of Blocks pr. File (logical) | 1400 |
| File Rewrite Limit | 10 % |
| Total Rewrite Limit | 2 % |
| File Reread Limit | 0 % |
| Total Reread Limit | 0 % |

*Table: SEND DIAGNOSTICS Default Parameters*

## 27.5. Selftest 1

This test will check most of the digital hardware:

**CPU Test**

Most of the instruction set for the micro-processor is tested. The test is divided into an arithmetic, a logical and a data move test. A fixed sequence of instructions is executed, then the result is checked against a pre-calculated answer.

**Scratch Pad RAM Test**

All RAM cells are read from and written to. This test is using two fixed patterns and a count pattern that will be reset when the prime number 251 is reached.

**Drive Controller Test**

The hardware for write and read formatting are tested. The Drive Controller chip is placed in digital loopback mode and one data block is fed into the write sequencer by the DMA0 channel. The last 8 bytes in the block + CRC are read back from the read sequencer and compared. CRC is also checked.

**SCSI Controller Test**

No advanced test of the SCSI controller can be performed due to the need of maintaining the communication with the Initiator during the test.
The SCSI Controller Test will only check that the SCSI controller is present.

**EDC Controller Test**

The DMA0 channel is tested by sending a data block to the Drive controller (in Test Mode) and then read back and checked. The DMA1 channel is tested by copying a block in the data buffer and then by reading it back and checking it. The ECC channel is tested by writing 14 data blocks (one frame) to the data buffer with ECC generation turned on. The generated ECC pattern is checked. One byte is then "bombed" in this frame. The ECC channel is then used to regenerate the bad byte. The frame is then read back and checked. The MPU transfer is checked by writing data to the data buffer. The data is read back and checked.

**Data Buffer Test**

The Data Buffer is tested with read and write using DMA1 in the EDC controller. The data patterns are 55h and AAh. The time consumption is approximately 500 ms. The entire 256 KByte is filled with the test pattern, then read back and compared. Both parity and compare errors are checked.

**Data Compression/-EDC-controller Test**

The Data Compression Controller Test will check that the Data Compression controller is NOT present for the standard Drive.

***SLR5 4.0/8.0GB and SLR4 2.5/5.0GB (TDC 4222) only***

No advanced test of the Data Compression chip is performed. The Data Compression Controller Test will check that the Data Compression Controller IS present.

## 27.6. Selftest 2

When Selftest 2 is selected then a Selftest 1 is performed first. Then a further test is performed. This test involves actual reading and writing on the tape. Note that the tape is always rewound back to BOT before Selftest 2 is started.

***SLR2 (TDC 3820) only***

The tape format used is always QIC-525 for the TDC 3820 Drive.

***SLR3 (TDC 4100) only***

The tape format used is always QIC-1000 for the TDC 4100 Drive.

***SLR4 (TDC 4200) Series***

The tape format used is always QIC-2GB for the TDC 4200 Series Drives.

***SLR5 Series***

The tape format used is always QIC-4GB for the SLR5 Series Drives.

This is regardless of the Density Code set up by the MODE SELECT command. With default parameters two tracks are written in files of 1400 logical data blocks. Between each file an underrun is forced to make the tape stop and reposition before a new file is appended (data append activates the erase circuitry).

The Number of Rewrites for a single block is 16 and the Number of Rereads for a single block is 24, independent of the Drive set-up.

The data pattern in each block alternates between three patterns. One block is written with a block count pattern, the next with a 29h pattern and the last with a 60h pattern. This sequence is then repeated.

After each file and when all files have been written, the number of rewrites is compared to the corresponding rewrite limits (see default parameters). If the number of rewrites is above the limit, the test is aborted and the SEND DIAGNOSTICS command returns with CHECK CONDITION status.

After the write test the tape is rewound to BOT.

If the write test detected no errors, a read test is performed. Here the data is read file-by-file from the tape in streaming mode. Note that ECC is turned off when the read test executes.

During the read test no "hard" read errors are normally allowed. If a "hard" read error or more rereads than specified should occur, the read test will be aborted and then performed again. If there is still a problem, the whole test is aborted. The SEND DIAGNOSTICS command returns with CHECK CONDITION status. If no "hard" read error has occurred and the rereads are within the specified limit, no error is reported when the test completes.

## 27.7. Exception Handling

See sections on Error Conditions For All Commands, Deferred Errors and Buffer Parity Errors.

If the PF bit is not set to zero, the Drive will terminate the SEND DIAGNOSTICS command with CHECK CONDITION status. No diagnostic tests will be performed and parameter data will be transferred. The Error Code will be set to E$STE_IFIC.

If the ST, DOF and UOF bits are not within the legal values, the Drive will terminate the SEND DIAGNOSTICS command with CHECK CONDITION status. No diagnostic tests will be performed and no parameter data will be transferred. The Error Code will be set to E$STE_IFIC.

If the Parameter List Length is not within the legal values, the SEND DIAGNOSTICS command will be terminated with CHECK CONDITION status. No diagnostic tests will be performed and no parameter data will be transferred. The Error Code will be set to E$STE_IFIC.

If the Parameter List has any illegal or out of range values, the SEND DIAGNOSTICS command will be terminated with CHECK CONDITION status. No diagnostic tests will be performed and no parameter data will be transferred. The Error Code will be set to E$STE_IFIC.

# 28. Space

## 28.1. Command Description

The SPACE command provides a variety of positioning functions determined by a space code and a space count. Both forward (toward end-of-media) and reverse (toward beginning-of-media) positioning are provided.

The space code allows the Initiator to space over blocks, tapemarks (= filemarks or setmarks), sequential filemarks or to End Of Recorded Area.

When spacing over blocks or tapemarks, the count specifies the number of blocks or tapemarks to be spaced over. When spacing N blocks or tapemarks in the forward direction, the space will end on the end-of-media side of the last block or tapemark spaced. When spacing N blocks or tapemarks in the reverse direction the space will end on the beginning-of-media side of the last block or tapemark spaced.

When spacing over N sequential filemarks, the count specifies that the space will end at the first occurrence of N or more consecutive filemarks. The tape will be logically positioned at the end-of-media side (forward space) or at the beginning-of-media side (reverse space) of the n'th filemark.

When spacing to End Of Recorded Area the count field will be ignored. This space function is always using the "fast seek" algorithm. After a successful space to End Of Recorded Area, a subsequent WRITE or COPY (copy function = backup) or WRITE FILEMARKS command will append data to the last recorded block.

The SPACE command is able to space over blocks written in both fixed and variable length mode. The SPACE command is able to automatically determine the block type while spacing. Note, however, that when the current tape format is QIC-120 or QIC-150 and there is a mixture of fixed and variable length blocks on the tape the SPACE REVERSE command may loose track of the actual tape position. If the use of the SPACE REVERSE command is required then the user should make sure that a QIC-120 or QIC-150 tape has been written with fixed length blocks only or variable length blocks only.

If disconnection is allowed, the Drive will disconnect when executing this command.

## 28.2. Command Descriptor Block

| BYTE | BIT 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 00 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| 01 | Logical Unit Number (LUN) | | | RESERVED | | Code | | |
| 02 | Count | | | | | | | |
| 03 | | | | | | | | |
| 04 | | | | | | | | |
| 05 | FAST | X | RESERVED | | | | Flag | Link |

*Table: SPACE Command Block*

**Code**

The Space Code is defined as follows:

0: Space over Blocks
1: Space over Filemarks
2: Space over Sequential Filemarks
3: Space to End Of Recorded Area
4: Space over Setmarks

**Count**

This field specifies the number if blocks to space (Code=0), the number of filemarks to space (Code=1 or Code=2) or the number of setmarks to space (Code=4). The Count field is ignored when spacing to End Of Recorded Area (Code=3). A positive Count field will cause forward positioning. A negative Count field (2's complement notation) will cause reverse positioning. A zero value in the Count field will cause no tape movement (except for Code=3).

**FAST**

When the FAST bit is set to one this indicates that the Drive will use a special fast space algorithm. When using this algorithm, the Drive is not able to test for possible bad/corrupted data blocks on the tape. See Section 28.3. for details.

## 28.3. Using Fast Space

The FAST option enables the Drive to perform FAST space to any data or filemark block on the tape. All SPACE operations except Space over Sequential Filemarks will use the FAST algorithm if the command is issued with the FAST bit set. For Space over Sequential Filemarks the FAST bit is ignored. Instead of searching in serpentine mode through every track, the Drive will use certain help information recorded on the tape to space more or less directly to the wanted position. When this option is used, the Drive will still be able to report detected filemarks during the space blocks operation. Unrecoverable blocks between the *start* and end position will, however, most likely not be detected.
Using the FAST option, all spaces will normally be significantly faster than if NORMAL SPACE is chosen.

The only case when FAST SPACE can be more time consuming is the first time when a SPACE command is issued into an area of the tape that has not been read or written since insertion of the cartridge, and where the end position is relatively close to the start position (on the *same track).*

*Typical space time for any FAST SPACE on a 600 feet cartridge is approximately 35 sec.*
*The maximum space time for any FAST SPACE on a 600 feet cartridge is approximately 160 sec.*
*SPACE REVERSE from EOT to BOT will take about 5 hours for QIC-1000 and 10 hours for QIC-2GB. The SPACE REVERSE will typically use 20 seconds per MByte.*

Instead of selecting FAST or NORMAL SPACE operation on command to command basis, it is possible to select the Drive always to use FAST SPACE by means of the MODE SELECT command. By doing so, the system drivers will not have to be changed to get the advantage of FAST SPACES.

The recorded help information needed to enable this seek function is transparent to the user and it does not violate any of the tape standards.

## 28.4. Exception Handling

### *28.4.1. General*

See sections on Error Conditions For All Commands, Deferred Errors and Error Conditions For Media Access Commands.
If the FIX bit is one and the configured Block Size is 1024, the READ command will be terminated with CHECK CONDITION status if the tape format is different from QIC-525/1000/2GB/96-34. The Error Code will be set to E$BTD_TFMT. No data will be transferred and the tape position will be at BOT.
When the SPACE command has started execution, all detected errors will set the VADD bit and the Information Bytes will hold the difference between the requested and the actual number of blocks/tapemarks spaced. Note, however, that when spacing sequential filemarks, the VADD bit is never set and the Information Bytes are never valid. See the following sections for details.
If a SPACE is aborted by an Overlapped command, the command will stop execution and the tape will be positioned at BOT on the current partition.

### *28.4.2. No Data*

If the Drive is not able to find a reference burst on the inserted cartridge, the cartridge is assumed to be blank and the SPACE command will be terminated with CHECK CONDITION. The Error Code will be set to E$TCM_NODATA. The valid (VADD) bit in the sense data list will be set to one. The Information Bytes will be set equal to the Requested Transfer Length.
A SPACE to End of Recorded Area will, however, terminate with GOOD STATUS.

### 28.4.3. Filemark/Setmark Detected

If a filemark or setmark is encountered while spacing over blocks (Code=0), the SPACE command will be terminated with CHECK CONDITION status. The Error Code will be set to E$BTD_FIMK for filemark and E$BTD_SEMK for setmark. The Information Bytes in the sense data will be set to the difference (residue) of the requested count minus the actual number of blocks spaced over (not including the tapemark). The logical position will be located at the end-of-media side (fwd. space) or beginning-of-media side (reverse space) of the tapemark.

### 28.4.4. Logical End of Partition

If the logical end of partition (End Of Recorded Area) is encountered while spacing forward over blocks, tapemarks or sequential filemarks, the Drive will return CHECK CONDITION status. The Error Code will be set to E$TEM_EOR or E$TEM_EOREW and the Sense Key will be set to BLANK CHECK. Additionally the EOM bit will be set to one if the logical end of partition was encountered at or after the physical early warning (EW) tape marker on the last track. If logical end of partition is encountered before the early warning marker, the EOM bit will be set to zero. The Valid (VADD) bit will be set to one and the Information Bytes in the sense data will be set to the difference (residue) of the requested count minus the actual number of blocks, filemarks or setmarks. The logical tape position will be so that a subsequent WRITE or COPY (copy function = backup) or WRITE FILEMARK command will append data to the last recorded block.

### 28.4.5. Physical Beginning of Partition

If the physical beginning of partition is encountered while spacing in the reverse direction, the Drive will return CHECK CONDITION status. The Error Code will be set to E$BTD_PBOP. Additionally, the Drive will set the end-of-media (EOM) and Valid (VADD) bits to one. The Information Bytes in the sense data will be set to the difference (residue) of the requested count minus the actual number of blocks, filemarks or setmarks spaced over. The logical tape position will be so that a subsequent read operation will read the first block on the tape.

### 28.4.6. Physical End of Partition

If the physical end of partition (the EOT tape marker on the last track) is encountered while spacing in the forward direction, the Drive will return CHECK CONDITION status. The Error Code will be set to E$TEM_PEOP and the Sense Key will be set to MEDIUM ERROR. The Valid (VADD) bit will be set to one and the Information Bytes in the sense data will be set to the difference (residue) of the requested count minus the actual number of blocks, filemarks and setmarks or sequential filemarks and setmarks spaced over. The logical tape position will be undefined and all new COPY (copy function = restore), READ, SEEK BLOCK and SPACE commands will be terminated immediately with CHECK CONDITION status as if they just ran into the physical end of partition. A possible COPY (copy function = backup), WRITE or WRITE FILEMARKS command will also be terminated immediately with CHECK CONDITION status. The Error Code will then be set to E$BTD_WRRD. A position type command (ERASE, LOAD/UNLOAD or REWIND) must be executed before subsequent read or write operations can be started.

### 28.4.7. Non-Recoverable Read Error During Space Forward

If a non-recoverable read error occurs during the execution of a SPACE FORWARD command, the bad block will be assumed to be a data block and the Drive will terminate the SPACE command with CHECK CONDITION status. The Error Code will be set to E$BTD_RTRY. The valid (VADD) bit the sense data list will be set to one. The Information Bytes will be set to the difference (residue) between the requested count and the actual number of blocks or filemarks and setmarks spaced over. When the command has terminated, the logical tape positions will be after the bad physical tape block.

If a new SPACE (or READ or VERIFY) command is issued after a non-recoverable read error has occurred, the space operation will begin with the physical tape block following the erroneous physical tape block. If this physical tape block was located in the middle of a variable block, this new SPACE command will space the remaining bytes of the last (truncated) variable block. This means that the length of this rest-block will be equal to the original length minus the length spaced in the previous SPACE command (including any dummy bytes). This mechanism has the effect of splitting a variable length block with non-recoverable physical tape block in the middle, into two variable blocks with total length equal to the total length of the original variable block.

### 28.4.8. Error Condition or Bad Block During Space Reverse

If a bad block or an error condition is detected during Space Reverse (blocks, filemarks, setmarks or sequential filemarks), the command terminates with CHECK CONDITION status. The Error Code will be set to E$STE_SREV. The VADD bit is set in the Sense Data and the Information Bytes will hold the residual count. Note that this count can be *negative!* In this case the error or bad block was logically detected in front of the destination position.
This can happen as the Drive is not able to read in reverse direction. It therefore has to fill the buffer with data and then test the buffer in reverse direction. The buffer will then typically also contain blocks that are logically in front of the destination position. If one of these blocks are bad, or if an error condition is detected at the time when one of the blocks should be read, a negative residual count will occur.
If an error occurs during SPACE REVERSE, the Drive will *not* allow any further Write, Read or Space commands.
An eventual Recovery will have to start from BOT.

### 28.4.9. SLR5 4.0/8.0GB and SLR4 2.5/5.0GB (TDC 4222) Only - Exception Handling

For the SLR5 4.0/8.0GB and the SLR4 2.5/5.0GB (TDC 4222), see the READ command Sections 18.3.9. Data Compression Exception Handling and 18.3.10. Reading From the Beginning of the Directory Partition.

*This Page Intentionally Left Blank*

# 29. Test Unit Ready

## 29.1. Command Description

The TEST UNIT READY command provides a means to check if the Drive is ready for a medium access command. If the Drive would accept an appropriate medium access command without returning CHECK CONDITION status, this command will return GOOD status. See the section on Command Descriptors for the list of Medium Access Commands.

The Drive will not disconnect when executing this command.

## 29.2. Command Descriptor Block

| BYTE | BIT 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 00 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 01 | Logical Unit Number (LUN) | | | RESERVED | | | | |
| 02 | RESERVED | | | | | | | |
| 03 | RESERVED | | | | | | | |
| 04 | RESERVED | | | | | | | |
| 05 | Control Byte | | | | | | | |

*Table: TEST UNIT READY Command Block*

## 29.3. Exception Handling

See section Error Conditions For All Commands.

The TEST UNIT READY command will return CHECK CONDITION status if the cartridge is not inserted or not loaded. The Error Code will be set to E$STE_NCAR (no cartridge inserted) or E$STE_NLOD (cartridge not loaded). It must, however, return GOOD status when the cartridge is auto loading/retensioning.

*This Page Intentionally Left Blank*

# 30. Verify

## 30.1. Command Description

The VERIFY command verifies one or more block(s) beginning with the next block on the tape.

The VERIFY command transfers one or more blocks from the Initiator. The Fixed (FIX) bit specifies both the meaning of the Transfer Length field and whether fixed-length or variable length block(s) are to be transferred. The data to verified will be transferred during the DATA-OUT phase of the command.

When the FIX bit is set to zero, the Drive is requested to transfer a single variable length data block. The Verification Length specifies the block length in number bytes. The block length is expected to be equal to the specified block length.

When the FIX bit is set to one, the Drive is requested to transfer the requested number of fixed length blocks. All the blocks is expected to be of the same length. The length expected is the length reported by the MODE SENSE command (the Block Size field of the Block Descriptor List). Note that a FIX bit of one is not legal when the Drive has been set into Variable Block mode. Variable Block mode is in effect when the Block Size field in the Block Descriptor List of the MODE SELECT command is set to zero (000000h). See the MODE SELECT command for further details.

If the requested transfer length is zero, then the Drive will transfer no data and the logical tape position will not be changed. This will not be considered as an error.

If the VERIFY command is the first media access command executed on a newly inserted cartridge, the verify operation will start from BOM. If the VERIFY command follows a ERASE, LOAD/UNLOAD (with Load bit set to one) or REWIND command, the verify operation will also start from BOM. If the VERIFY command follows a COPY (with Copy Function = restore), READ, SEEK BLOCK/LOCATE, SPACE or another VERIFY command, the verify operation will start with the next block on the tape.

Upon termination of a successful VERIFY command, the logical tape position will be after the last block (fixed or variable) verified (end-of-media side).

The Drive will disconnect when executing this command if the number of blocks requested for transfer exceeds the number of blocks free in the data buffer when the command has been received.

The VERIFY command will be able to do two kinds of verification; CRC/ECC check only or a byte-by-byte data compare.

If the data does not compare (Byte Compare bit equals one), the command will terminate with CHECK CONDITION status and the Error Code will be set to E$BTD_VRFY.

If the FIX is set to one, the Valid (VADD) bit will be set to one and the Information Bytes in the sense data will be set to the difference (residue) between the Verification Length and the actual number of blocks transferred. If the FIX is set to zero, the Valid (VADD) bit will be set to one and the Information Bytes in the sense data will be set equal to the Transfer Length (the whole block is considered not to compare). The tape will be logically positioned somewhere near the end-of-media side of the block containing the miscompare. Note that the Drive is only required to check for, and stop on, miscompare errors when it is about to disconnect (or go to BUS FREE). This means that when a miscompare error has been signalled to the Initiator, the actual byte(s) in error is (are) somewhere in the last transfer burst.

In the Drive the actual data compare will be done on the SCSI-bus side of the data buffer. This means that when a VERIFY command terminates, all data transferred have already been verified. This will guarantee that no deferred errors occurs due to a miscompare. This also means that the VERIFY command always executes in a "buffered" mode.

If disconnection is allowed, the Drive will disconnect when executing this command if the number of blocks requested for transfer exceeds the number of blocks free in the data buffer when the command has been received.

## 30.2. Command Descriptor Block

| BYTE | BIT 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 00 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 |
| 01 | Logical Unit Number (LUN) | | | RESERVED | | IMM | CMP | FIX |
| 02<br>03<br>04 | Verification Length | | | | | | | |
| 05 | Control Byte | | | | | | | |

*Table: VERIFY Command Block*

**IMM**

This bit is ignored by the Drive. When byte-by-byte data compare is performed (CMP bit set to one) the operation is always "immediate" because the Drive compares the data at the SCSI-bus side of the data buffer. When CRC check only is performed (CMP bit set to zero) the VERIFY command does not return with status until all the requested data has been verified.

**CMP**

A Byte Compare (CMP) bit of zero indicates that the verification will be simply a CRC/ECC verification. No data will be transferred between the Initiator and the Drive. The VERIFY command will then be functionally equivalent to a SPACE block forward command. A CMP bit of one indicates that a byte-by-byte compare of the data on tape and the data transferred from the Initiator will be performed by the Drive. Data will be transferred from the Initiator to the Drive as in a WRITE command.

***SLR5 4.0/8.0GB and SLR4 2.5/5.0GB (TDC 4222) only***

The SLR5 4.0/8.0GB and the SLR4 2.5/5.0GB (TDC 4222) only support a CRC/ECC verification. The CMP-bit must always be set to zero (0).

**FIX**

A Fixed (FIX) bit of zero indicates that a single block will be transferred with the Verification Length specifying the number of bytes to verify. A FIX bit of one indicates that the Verification Length specifies the number of blocks to be verified beginning with the next logical block. Note that a FIX bit of one is not allowed when the Drive is in Variable Block mode (see the Block Size field in the Block Descriptor List of the MODE SELECT command for further details).

**Verification Length**

This field specifies the number of bytes or blocks requested for verification. Any value in the range 0..16777215 is legal. Note, however, that for the QIC-120 and QIC-150 tape formats the maximum block size that can be written by the Drive is 32768 bytes and for the QIC-24 tape format the block length is always 512 bytes.

## 30.3. Exception Handling

See the section on Exception Handling for the READ command. Remember that data will be transferred FROM the Initiator during execution of the VERIFY command.

### *30.3.1. SLR5 4.0/8.0GB and SLR4 2.5/5.0GB (TDC 4222) Only - Exception Handling*

For the SLR5 4.0/8.0GB and the SLR4 2.5/5.0GB (TDC 4222), see the READ command Sections 18.3.9. Data Compression Exception Handling and 18.3.10. Reading From the Beginning of the Directory Partition.

If the CMP-bit is set to one (1), the SLR5 4.0/8.0GB or the SLR4 2.5/5.0GB (TDC 4222) will terminate the VERIFY command with CHECK CONDITION status. The Error Code will be set to E$STE_IFIC.

*This Page Intentionally Left Blank*

# 31. Write

## 31.1. Command Description

The WRITE command transfers one or more block(s) from the Initiator to the tape at the current tape position.

The Fixed (FIX) bit specifies both the meaning of the Transfer Length field and whether fixed-length or variable length block(s) are to be transferred. The data to be written will be transferred during the DATA-OUT phase of the command.

When the FIX bit is set to zero, the Drive is requested to transfer a single variable length data block. The Transfer Length specifies the length of the block in number of bytes. When the tape format used is QIC-525/1000/2GB/96-34 (the Density Code is set to 11h/15h/22h), the maximum block length is FFFFFFh (16777215) bytes. When the tape format is QIC-120 or QIC-150 (the density code is in the range 0Dh..10h), the maximum block length is 8000h (32768) bytes.

When the FIX bit is set to one, the Drive is requested to transfer a number of fixed length blocks. The Transfer Length specifies the number of block to transfer. All blocks will be of equal length. The length used is the length reported by the MODE SENSE command (the Block Size field of the Block Descriptor List). Note that a FIX bit of one is not legal when the Drive has been set into Variable Block mode. Variable Block mode is in effect when the Block Size field in the Block Descriptor List of the MODE SELECT command is set to zero (000000h). See the MODE SELECT command for further details.

If the requested transfer length is zero, then the Drive will transfer no data and the logical tape position will not be changed. This will not be considered an error.

Upon termination of a successful WRITE command, the logical tape position will be after the last block (fixed or variable) written (end-of-media side).

If the WRITE command is the first media access command executed on a newly inserted cartridge, the write operation will start from BOM. If the WRITE command follows a ERASE, LOAD/UNLOAD (with Load bit set to one) or REWIND command, the write operation will also start from BOM. If the WRITE command follows a COPY, READ, SPACE, WRITE FILEMARKS or another WRITE command, and the tape is positioned for a data append, the write operation will start at the current tape position.

The WRITE command can operate in unbuffered and buffered mode. For unbuffered operation, the Drive will not return GOOD status until all data blocks are successfully written to the medium. For buffered operation, the Drive may return GOOD status as soon as all data blocks are successfully transferred to the Drive's data buffer. See also the MODE SELECT Section.

When operating in buffered mode; a WRITE FILEMARKS command with the IMM bit set to zero will be issued when completing a WRITE operation to ensure that all the buffered data and the filemarks are written to the tape.

If disconnection is allowed, the Drive will disconnect when executing this command.

NOTE:
When writing variable blocks in the QIC-120 or QIC-150 tape formats, the Write Buffer Full Ratio (see the MODE SELECT command) does not control the maximum burst size as usual. This means that the complete variable block is always transferred with no in-block disconnects.

## 31.2. Command Descriptor Block

| BYTE | BIT 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 00 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| 01 | Logical Unit Number (LUN) | | | RESERVED | | | | FIX |
| 02 | Transfer Length | | | | | | | |
| 03 | | | | | | | | |
| 04 | | | | | | | | |
| 05 | Control Byte | | | | | | | |

*Table: WRITE Command Block*

**FIX**

A Fixed (FIX) bit of zero indicates that a single block will be transferred with the Transfer Length specifying the length of the block in bytes. A FIX bit of one indicates that the Transfer Length specifies the number of fixed length blocks to be transferred from the Initiator.

**Transfer Length**

This field specifies the number of bytes or blocks requested for transfer. Any value in the range 0..16777215 is legal when the QIC-525/1000/2GB/96-34 tape format has been selected. When the tape format is QIC-120 or QIC-150 and the FIX bit is clear, the Transfer Length must be in the range 0..32768.

NOTE:
Not all drive suppliers support this format when writing variable blocks in the QIC-120/QIC-150 tape format.

## 31.3. Terminating Write Operations

When writing on a tape, the write operation must be properly terminated before the cartridge is suitable for reading on the same or any other drive. A write operation is only terminated when following has been completed successfully:

- *the very last data block in the buffer has been filled out (in QIC-525/1000/2GB/96-34 mode - either as a complete 1024 byte physical tape block comprising two valid 512 byte SCSI blocks, or as a 1024 byte variable physical tape block holding only 512 bytes of valid SCSI data or as a completely variable block*
- *all data and filemark blocks in the buffer has been written to the tape*
- *the frame has been filled up with FILLER blocks if necessary, and the two ECC blocks in the last frame has been written to the tape (when a tape format with ECC is in effect)*
- *at least 45 in. of tape after the last block has been erased*

When a COPY (copy function = backup), WRITE or WRITE FILEMARKS command has terminated successfully, the actual write operation will not be fully terminated because:

- *When the Drive is in buffered mode, any WRITE or WRITE FILEMARKS command (with the IMMEDIATE bit set), will terminate execution (by sending a Status byte) as soon as all requested data has been transferred. There may not be enough data to fill complete physical tape blocks. Only if the WRITE command operated in the variable block mode will a possible incomplete last block be completed (marked with the number of valid data bytes). If the WRITE command operated in fixed block mode, a possible incomplete last block (filled with 512 bytes only) will not be completed. Data from the* ***next*** *WRITE command will then be used to complete (fill up) the incomplete block.*
- *When the Drive is in unbuffered mode, all data (and filemarks) will be written to the tape (writing a variable physical tape block as the very last block if necessary) and the last frame will be padded with FILLER blocks if necessary. The erasing of tape will, however, not start.*

> NOTE:
> When the Drive is in non buffered mode, the tape format is QIC-525/1000/2GB/96-34 and the transfer length of the write command is small, the actual capacity of a tape may be much less than specified. This is because FILLER blocks are used to fill the last frame of all WRITE commands.

The following commands will force all data (and filemarks) in the data buffer to be written to the tape, and the last frame to be padded, even if the Drive is in buffered mode:

- *ERASE*
- *LOAD/UNLOAD*
- *LOCATE*
- *REWIND*
- *WRITE FILEMARKS with count equal to zero*
- *WRITE FILEMARKS (if no IMMEDIATE bit set)*

The following commands will also force the erasure of 45 in. of tape after the last block written:

- *ERASE*
- *LOAD/UNLOAD*
- *LOCATE*
- *REWIND*

## 31.4. Write from BOM

Write from BOM is allowed:

- *when the Write command is the first medium access command executed on a newly inserted cartridge*
- *after an ERASE, LOAD or REWIND command*
- *after a SPACE or LOCATE command ending with BOM detected*
- *when the tape is logically positioned at BOM after SPACE and LOCATE commands*

*This is only possible when the logical counters are valid. Hence, Write from this position is not possible after LOCATE commands with Fast Space disabled and after a bad block is detected during READ or SPACE operations.*
*Note that WRITE is not allowed after a Read 0 Blocks command.*

## 31.5. Exception Handling

### 31.5.1. General

See sections on Error Conditions For All Commands, Deferred Errors, Error Conditions For Media Access Commands and Buffer Parity Errors.
When the WRITE command has started execution, all detected errors will set the VADD bit and the Information Bytes will hold the difference between the requested and the actual transfer length. See the following sections for details. If the FIX bit is one and the Drive is in Variable Block mode, the WRITE command will be terminated with CHECK CONDITION. The Error Code will be set to E$STE_IFIC. No data will be transferred.
If the Transfer Length is not in the legal range, the WRITE command will be terminated with CHECK CONDITION. The Error Code will be set to E$STE_IFIC. No data will be transferred.
If the inserted cartridge is write-protected, the WRITE command will be terminated with CHECK CONDITION status. No data will be transferred. The Error Code will be set to E$BTD_WPRO. This will be true even if the requested transfer length is zero.

### 31.5.2. Illegal Media Type

All tape formats will not be legal on all media (cartridge) types. When an illegal combination is detected, the Drive will terminate the WRITE command with CHECK CONDITION status. The Error Code will be set to E$BTD_CFMT. No data will be transferred.
The following table indicates legal/illegal media/format combinations:

| | QIC-120 | QIC-150 | QIC-525 | QIC-1000 | QIC-2GB | QIC-4GB |
|---|---|---|---|---|---|---|
| **DC300** | Illegal | Illegal | Illegal | Illegal | Illegal | Illegal |
| **DC300XLP** | Illegal | Illegal | Illegal | Illegal | Illegal | Illegal |
| **DC615** | OK | Illegal | Illegal | Illegal | Illegal | Illegal |
| **DC600A** | OK | Illegal | Illegal | Illegal | Illegal | Illegal |
| **DC6037** | OK | OK | Illegal | Illegal | Illegal | Illegal |
| **DC6150** | OK | OK | Illegal | Illegal | Illegal | Illegal |
| **DC6250** | OK | OK | Illegal | Illegal | Illegal | Illegal |
| **DC6320** | OK | OK | OK | Illegal | Illegal | Illegal |
| **DC6525** | OK | OK | OK | Illegal | Illegal | Illegal |
| **DC91XX** | Illegal | Illegal | Illegal | OK | Illegal | Illegal |
| **DC92XX** | Illegal | Illegal | Illegal | Illegal | OK | Illegal |
| **DC9400** | Illegal | Illegal | Illegal | Illegal | Illegal | OK |

*Table: Legal/Illegal Media/Format Combinations*

If the FIX bit is one and the configured Block Size is 1024, the WRITE command will be terminated with CHECK CONDITION status if the tape format is different from QIC-525/1000/2GB/96-34. The Error Code will be set to E$BTD_CFMT. No data will be transferred and the tape position will be at BOT.

### 31.5.3. Illegal Append Tape Format

When appending data on a pre-recorded tape, the only legal tape format is the format found on the tape. If the format set by the MODE SELECT command is not equal to the tape format, the Drive will terminate the WRITE command with CHECK CONDITION status. The Error Code will be set to E$BTD_TFMT. No data will be transferred.

If the tape format is QIC-525/1000/2GB/96-34 and the last block found on the tape is not the last ECC block in a frame, the Drive will terminate the WRITE command with CHECK CONDITION status. The Error Code will be set to E$BTD_APUF. No filemarks will be written.

If the tape format is QIC-120 or QIC-150 with ECC, the Drive will terminate the WRITE command with CHECK CONDITION status. The Error Code will be set to E$BTD_TFMT. No data will be transferred.

### 31.5.4. Pseudo Early Warning

If pseudo early warning (PSEW) tape marker is reached during the execution of a WRITE command, the Drive will respond as described below:

- *The Drive will stop transferring data. The Drive will then attempt to write all buffered data (and filemarks) to the tape. The WRITE command will then be terminated with CHECK CONDITION. If all data in the buffer was successfully written to the tape, the Error Code will be set to E$BTD_PSEW and the Sense Key will be set to NO SENSE. The EOM and VADD bits will be set to one.*

  The Information Bytes in the sense data list will be set as follows:

  1) *If the FIX bit is set to one, the Information Bytes will be set to the difference (residue) of the requested transfer length minus the actual number of transferred blocks.*

  2) *If the FIX bit is set to zero, the Information Bytes will be set to the difference (residue) of the requested transfer length minus the actual number of transferred bytes.*

     *The Drive will force unbuffered mode (regardless the state of the BM bit in the MODE SELECT/SENSE Parameter List).*

- *If a read type command (COPY - copy function = restore), READ, LOCATE, SPACE or VERIFY), has brought the tape past the PSEW marker and an Append Operation is attempted, then the Drive will terminate the first WRITE command immediately with CHECK CONDITION. No data will be transferred. The Error Code will be set to E$BTD_PSEW and the Sense Key will be set to NO SENSE. The EOM and VADD bits will be set to one. The Information Bytes in the sense data list will be set equal to the requested transfer length.*

  *The Drive will force unbuffered mode (regardless the state of the BM bit in the MODE SELECT/SENSE Parameter List).*

- *If another WRITE command is received by the Drive while the tape is positioned after PSEW (but before end of partition), the Drive will transfer and write all requested data if possible. The WRITE command will then be terminated with CHECK CONDITION (this will be true even if the requested transfer length was zero). If all data in the buffer was successfully written to the tape the Error Code will be set to E$BTD_PSEW and the Sense Key will be set to NO SENSE. The EOM and VADD bits will be set to one. The Information Bytes will be set to zero.*

### 31.5.5. End of Partition

If end of partition is reached during the execution of a WRITE command, the Drive will terminate the command with CHECK CONDITION status. The Error Code will be set to E$WRT_EOM and the Sense Key will be set to VOLUME OVERFLOW. The Valid (VADD) and End Of Media (EOM) bits will be set to one.

The Information Bytes in the sense data list will be set as follows:

**1)** If the FIX bit is set to ***one***, the Information Bytes will be set to the difference (residue) of the requested transfer length minus the actual number of transferred ***blocks***.

**2)** If the FIX bit is set to ***zero***, the Information Bytes will be set to the difference (residue) of the requested transfer length minus the actual number of transferred ***bytes***.

This error condition has priority over the Pseudo Early Warning error condition.

The Initiator will be able to read back buffered data and filemarks by issuing the RECOVER BUFFERED DATA command. The logical tape position will be undefined. Additional WRITE commands issued after an end-of-partition error has occurred will be terminated immediately with a CHECK CONDITION (no data will be transferred). The end-of-partition condition will persist until a position type command has been executed (ERASE, LOAD/UNLOAD or REWIND).

### 31.5.6. Non-Recoverable Write Error

If a non-recoverable write error occurs, the Drive will terminate the WRITE command with CHECK CONDITION status. The Error Code will be set to E$WRT_REWRITE and the Sense Key will be set to MEDIA ERROR. The Valid (VADD) bit will be set to one.

The Information Bytes in the sense data list will be set as described in Section 31.5.5.

This error condition has priority over the Pseudo Early Warning error condition.

The Initiator will be able to read back buffered data and filemarks by issuing the RECOVER BUFFERED DATA command. The logical tape position will be undefined. Additional WRITE commands issued after a non-recoverable write error has occurred, will be terminated immediately with a CHECK CONDITION (no data will be transferred). The non-recoverable write error condition will persist until a position type command has been executed (ERASE, LOAD/UNLOAD or REWIND).

### 31.5.7. Append Error

When the WRITE command is executed after a COPY (copy function = restore), READ, LOCATE, SPACE, VERIFY or WRITE command where the tape motion for some reason has stopped, the Drive must seek the last block on the tape before the write operation starts. If this seek operation fails, the Drive will terminate the WRITE command with a CHECK CONDITION status. The Error Code will be set to E$WRT_-APFAIL and the Sense Key will be set to MEDIA ERROR. The Valid (VADD) bit will be set to one.

The Information Bytes in the sense data list will be set as described in Section 31.5.5.

This error condition has priority over the Pseudo Early Warning error condition.

The Initiator will be able to read back buffered data and filemarks by issuing the RECOVER BUFFERED DATA command. The logical tape position will be undefined. Additional WRITE commands issued after a non-recoverable write error has occurred, will be terminated immediately with a CHECK CONDITION (no data will be transferred). The append error condition will persist until a position type command has been executed (ERASE, LOAD/UNLOAD or REWIND).

### 31.5.8. SLR5 4.0/8.0GB and SLR4 2.5/5.0GB (TDC 4222) Only - Data Compression Exception Handling

If an unrecoverable compression error is encountered, the Drive will terminate the WRITE command with CHECK CONDITION status. The Error Code will be set to E$DCM_MISC.

### 31.5.9. SLR5 4.0/8.0GB and SLR4 2.5/5.0GB (TDC 4222) Only - Writing From the Beginning of the Directory Partition

Writing from the start of the directory partition will make the Drive do a pre-read on the data partition to evaluate the current data compression settings with respect to those found on the tape. In this case, all exceptions mentioned in Section 18.3.9. Data Compression Exception Handling will be valid.

# 32. Write Buffer

## 32.1. Command Description

The WRITE BUFFER command is used in conjunction with the READ BUFFER command as a diagnostic function for testing the Drive's data buffer and the SCSI bus integrity. Additional modes are provided for downloading and saving of microcode.

The WRITE BUFFER Parameter List will be transferred during the DATA-OUT phase of the command.

This command will not alter the status of possible inserted cartridge in any way. The WRITE BUFFER command will only be accepted when there is no installed cartridge or if an installed cartridge is logically positioned at BOT. This restriction is included to make sure that the WRITE BUFFER command will not write over tape related data already present in the data buffer (read-ahead data after a READ command or data not written after a buffered WRITE command). The Initiator may issue a REWIND command ahead of a WRITE BUFFER command to make sure that the cartridge is logically positioned at BOT.

A mode 101b (Download Microcode and Save) WRITE BUFFER command is illegal during load or retension operation. The Drive will return CHECK CONDITION status and the Save operation will not be performed.

If disconnection is allowed, the Drive may disconnect when executing this command. When transferring data, the total data transfer will be split into smaller bursts with a maximum size. The maximum burst size (the amount of data transferred between reconnects/disconnects) is controlled by the bus ratio/threshold parameters set up by the MODE SELECT command (just as for the WRITE command).

## 32.2. Command Descriptor Block

| BYTE | BIT 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 00 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 |
| 01 | Logical Unit Number (LUN) | | | RESERVED | | Mode | | |
| 02 | Buffer ID | | | | | | | |
| 03<br>04<br>05 | Buffer Offset | | | | | | | |
| 06<br>07<br>08 | Parameter List Length | | | | | | | |
| 09 | X | X | RESERVED | | | | Flag | Link |

*Table: WRITE BUFFER Command Block*

**Mode**

This field controls the function of the WRITE BUFFER command. It also controls the meaning of the other fields within this command descriptor block. The following modes are supported:

| Mode | Description |
|---|---|
| 000b | Write Combined Header and Data |
| 010b | Write Data |
| 100b | Download Microcode |
| 101b | Download Microcode and Save |

***SLR5 4.0/8.0GB and SLR4 2.5/5.0GB (TDC 4222) only***

The data transferred to the Drive will never be compressed, even if the data compression is turned ON in the MODE SELECT command.

**Buffer ID**

The data buffer is the only buffer that can be written to by the WRITE BUFFER command. The data buffer has an ID of 0 (zero). The Buffer ID field is reserved in mode 000b (Write Combined Header and Data). It must be set to zero in mode 010b (Write Data). The Buffer ID field is ignored in mode 100b (Download Microcode) and mode 101b (Download Microcode and Save).

**Buffer Offset**

The Buffer Offset field specifies an offset into the buffer given by the Buffer ID field. The Buffer Offset is always a *byte* offset into the buffer. If the Buffer Offset is set to *N*, then the first data byte transferred by the WRITE BUFFER command will be put into byte *N*, relative to the first byte of the specified buffer. If the Parameter List Length plus the Buffer Offset exceeds the capacity of the specified buffer, the WRITE BUFFER command will be terminated with a CHECK CONDITION status (no data will be transferred). The Buffer Offset field can only be used in mode 010b (Write Data), mode 100b (Download Microcode) and mode 101b (Download Microcode and Save). It is reserved in mode 000b (Write Combined Header and Data).

**Parameter List Length**

The Parameter List Length field specifies the number of bytes that shall be transferred during the DATA OUT phase. The transferred bytes will be stored in the specified buffer beginning at the Buffer Offset. If the Parameter List Length is zero the WRITE BUFFER command will have no DATA OUT phase and no data will be transferred. The Initiator should attempt to ensure that the Parameter List Length plus the Buffer Offset do not exceed the capacity of the specified buffer (the capacity of the buffer can be determined by the Buffer Capacity field in the READ BUFFER descriptor). If the Parameter List Length plus the Buffer Offset fields specify a transfer that would exceed the buffer capacity, the Drive will terminate the WRITE BUFFER command with a CHECK CONDITION status (no data will be transferred). In mode 000b (Write Combined Header and Data) the maximum Parameter List Length is 245764 bytes (Buffer Capacity + 4). In mode 010b (Write Data) the maximum Parameter List Length is 245760 bytes. In mode 100b (Download Microcode) and mode 101b (Download Microcode and Save) the maximum Parameter List Length is 131076 bytes.

**Flag**

See Section 4.2. Command Control Byte.

**Link**

The Link-bit *must be set to zero* in mode 101b (Download Microcode and Save). See Section 4.2. Command Control Byte for the use of the Link bit in other modes.

## 32.3. Combined Header and Data Mode (000b)

In this mode, data to be transferred is preceded by a four-byte header list:

| BYTE | BIT 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 00 | RESERVED | | | | | | | |
| 01 | RESERVED | | | | | | | |
| 02 | RESERVED | | | | | | | |
| 03 | RESERVED | | | | | | | |

*Table: WRITE BUFFER Header List*

The four-byte header consists of all reserved bytes. The Buffer ID and the Buffer Offset fields must be zero. The Parameter List Length field specifies the max. number of bytes that must be transferred. This number includes the four bytes of header and the *data* length to be stored in the Drive's data buffer, which is the Parameter List Length minus four. The transferred header will be validated and discarded, and the buffer data will be transferred to the Drive's data buffer. The data will be stored from buffer address 0 and up. The maximum legal Parameter List Length is 245764 bytes, the header included.

## 32.4. Write Data Mode (010b)

In this mode, the DATA OUT phase contains buffer data only. Data is written to the Drive's data buffer starting at the location specified by the Buffer Offset. The Parameter List Length field specifies the number of transferred bytes. Maximum Parameter List Length is 245760 bytes.

## 32.5. Download Microcode Mode (100b)

In this mode, drive-specific microcode can be transferred to the Drive data buffer. The microcode change will not take effect until after a WRITE BUFFER with mode 101b (Download Microcode and Save) has been executed. Note that the transferred microcode will be lost after a power-cycle or reset as long as it have not been saved.

Note that the WRITE BUFFER command in this mode does not perform any validation of the transferred microcode data. The validation is performed when a mode 101b (Download Microcode and Save) WRITE BUFFER command is executed

The Buffer ID field is ignored in this mode. Data is written to the Drive data buffer starting at the location specified by the Buffer Offset. The Parameter List Length field specifies the number of bytes to be transferred. The maximum Parameter List Length is 131076 bytes.

## 32.6. Download Microcode and Save Mode (101b)

In this mode, drive-specific microcode can be transferred to the Drive data buffer. If the WRITE BUFFER command is completed successfully the microcode will be saved in a non-volatile memory space (a Flash EPROM). The downloaded code will then be effective after each power-cycle and reset until it is supplanted in another download microcode and save operation.

The Buffer ID field is ignored in this mode. Data is written to the Drive data buffer starting at the location specified by the Buffer Offset. The Parameter List Length field specifies the number of bytes to be transferred. The maximum Parameter List Length is 131076 bytes.

The Initiator may either transfer a complete microcode in one single WRITE BUFFER command (using mode 101b with a Parameter List Length of 131076 and Buffer Offset of zero) or to split the download operation into several (smaller) WRITE BUFFER commands (using first mode 100b and then 101b). In the latter case, it is up to the Initiator to make sure that the use of the Buffer Offset and Parameter List Length in mode 100b results in continuous set of microcode data, starting at offset zero, before saving the microcode with one final mode 101b command. The final mode 101b command may have Parameter List Length of zero if all microcode data is downloaded by mode 100b commands.

Before saving any microcode the Drive will perform several tests on the microcode data to make sure that the data really are microcode data intended for the Drive (see Section 32.7. for details on the microcode data format):

- The Microcode Version field must match the Drive version identifier located in the Drive Flash EPROM. This prevents the possibility of changing the microcode version of the Drive.
- The Product ID must match the Product ID found in the Standard Inquiry Data. This prevents the possibility of saving microcode meant for a different product.

- The Mainboard Revision located in the EEPROM must be higher or equal the Mainboard Revision in the Microcode Header. This prevents the possibility of saving microcode meant for a different hardware

- The Microcode CRC check must be correct.

If any of these tests fails the WRITE BUFFER command will be terminated with a CHECK CONDITION status. The microcode in the data buffer is not saved and the current microcode is left unchanged.

Before staring the save operation, the Drive will disconnect from the SCSI bus. While performing the save operation (during Flash EPROM programming), the Drive will *not* respond to any SCSI activity except for a hard SCSI reset. A hard SCSI reset will most likely make the save operation fail and the Drive will then need servicing. This is *not* a recommended procedure.

When the save operation has completed successfully, the Drive will reconnect to the Initiator with a GOOD status. The Drive will then generate an Unit Attention condition - "Microcode Has Been Changed" for all Initiators except the Initiator that issued the mode 101b WRITE BUFFER command. For the Initiator that issued the mode 101b WRITE BUFFER command the Drive will generate an Unit Attention condition - "Power-Up/Reset".

If disconnection is not allowed, the mode 101b WRITE BUFFER command will give GOOD status *before* the save operation is executed.

If the save operation fails, the Drive will indicate the error with the front LED as in Selftest (see drive's reference manual for details on the LED). It will not be possible to access the Drive via the SCSI bus after a save failure. The Drive needs to be serviced since a power-up will not clear this error condition.

The time required for saving the microcode (Flash EEPROM programming) are:

**Time for saving of the Microcode:**

| | |
|---|---|
| Typical: | 20 seconds |
| Maximum: | 150 seconds |

## 32.7. Microcode Data Format

The microcode consists of a 45-byte Microcode Header followed by 131027 bytes of Microcode Data. 4 bytes of Microcode CRC follow last.

The microcode has the following format:

| Byte | Size | Description |
|---|---|---|
| 00 - 03 | 4 | Don't Care |
| 04 - 07 | 4 | RESERVED |
| 08 - 11 | 4 | Don't Care |
| 12 - 15 | 4 | Microcode Revision |
| 16 - 23 | 8 | RESERVED |
| 24 - 31 | 8 | Don't Care |
| 32 | 1 | Microcode Version |
| 33 - 35 | 3 | Don't Care |
| 36 - 39 | 4 | Product ID |
| 40 | 1 | Microcode Branch |
| 41 | 1 | Release Status |
| 42 - 43 | 2 | Mainboard Revision |
| 44 | 1 | Function Code |
| 45 - 131071 | 131027 | Microcode Data |
| 131072 - 131075 | 4 | Microcode CRC |

**Microcode Revision** This 4-byte field holds 4 ASCII characters that identify the new microcode revision level.

**Microcode Version** This 1-byte field holds an ASCII character that identifies a specific version of the microcode.

**Product ID** This 4-byte field holds 4 ASCII characters that represent the name of the product.

**Microcode Branch** This 1-byte field holds a number that represents a branch from the standard Microcode Revision.

**Release Status** This 1-byte field holds an ASCII character that represents the microcode release status.

**Mainboard Revision** This 2-byte field holds a number that represents the minimum hardware revision level required by this microcode.

**Function Code** This 1-byte field holds a number that identifies the microcode function code. The code tells if the microcode can be upgraded, downgraded or both.

**Microcode Data** The data to be programmed into the Flash EPROM in addition to the Microcode Header.

**Microcode CRC** The last 4 bytes of CRC cover the 131072 bytes of Microcode Header and Data. The following generating polynomial will be used:

$$X^{32} + X^{28} + X^{26} + X^{19} + X^{17} + X^{10} + X^{6} + X^{2} + 1$$

## 32.8. Exception Handling

See sections on Error Conditions For All Commands, Deferred Errors and Buffer Parity Errors.

If the WRITE BUFFER command is received while the inserted tape is not logically positioned at BOT, the Drive will terminate the WRITE BUFFER with CHECK CONDITION. The Drive Error Code will be set to E$STE_CSEQ. No data will be transferred.

If the Mode is not in the set of legal modes, the Drive will terminate the WRITE BUFFER with CHECK CONDITION. The Drive Error Code will be set to E$STE_IFIC. No data will be transferred.

If the Buffer ID and Buffer Offset fields are not set to zero in mode 000b (Write Combined Header and Data), the Drive will terminate the WRITE BUFFER command with CHECK CONDITION status. The Drive Error Code will be E$STE_IFIC. No data will be transferred.

If the Parameter List Length exceeds the available length in mode 000b (Write Combined Header and Data), the Drive will terminate the WRITE BUFFER command with CHECK CONDITION status. The Drive Error Code will be E$STE_IFIC. No data will be transferred.

If the reserved fields in the Header List are not set to zero, the Drive will return CHECK CONDITION status after having transferred all header and data bytes. The Drive Error Code will be E$STE_IFIP. Note that in this case the transferred header and data will overwrite data already in the data buffer.

If the Buffer ID is not set to zero in mode 010b (Write Data), the Drive will terminate the WRITE BUFFER command with CHECK CONDITION status. The Drive Error Code will be E$STE_IFIC. No data will be transferred

If the Parameter List Length plus the Buffer Offset in mode 010b (Write Data), mode 100b (Download Microcode) and mode 101b (Download Microcode and Save) exceeds the capacity of the specified buffer, the WRITE BUFFER command will be terminated with a CHECK CONDITION status. The Drive Error Code will be E$STE_IFIC. No data will be transferred.

If the Link bit is set in mode 101b (Download Microcode and Save), the WRITE BUFFER command will be terminated with a CHECK CONDITION status. The Drive Error Code will be E$STE_IFIC. No data will be transferred.

If the validity checks made on the microcode data in mode 101b (Download Microcode and Save) fails, the WRITE BUFFER command will be terminated with a CHECK CONDITION status. The Drive Error Code will be E$STE_IFIP. The Sense Key Specific field of the Sense Data will indicate the microcode field in error.

If the save operation fails (an error is detected during Flash EPROM programming), the Drive will indicate the error with the front LED (see the drive's reference manual for details).

If a Mode 101b command is issued while the Drive is in a load/retension sequence, the Drive will return CHECK CONDITION status, and the Error Code will be set to E$STE_ILOD.

*This Page Intentionally Left Blank*

# 33. *Write Filemarks*

## 33.1. Command Description

The WRITE FILEMARKS command causes the specified number of filemarks to be written beginning at the current tape position. A zero value of filemarks indicates that no filemarks are to be written. This can be used to force any buffered write data or filemarks to be written to the tape.

If the WRITE FILEMARKS command is the first media access command executed on a newly inserted cartridge, the write filemark operation will start from BOM. If the WRITE FILEMARKS command follows a ERASE, LOAD/UNLOAD (with Load bit set to one) or REWIND command, the write filemark operation will also start from BOM. If the WRITE FILEMARKS command follows a COPY, READ, SPACE, LOCATE, WRITE or another WRITE FILEMARKS command, and the tape is in an append position, the write filemark operation will start at the current tape position.

Upon termination of a successful WRITE FILEMARKS command, the logical tape position will be after the last filemark written (end-of-partition side).

See also the WRITE command for details on terminating write operations.

If disconnection is allowed, the Drive will disconnect when executing this command.

## 33.2. Command Descriptor Block

| BYTE | BIT 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 00 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 01 | Logical Unit Number (LUN) | | | RESERVED | | | WSmk | IMM |
| 02 | Number of Filemarks | | | | | | | |
| 03 | | | | | | | | |
| 04 | | | | | | | | |
| 05 | Control Byte | | | | | | | |

*Table: WRITE FILEMARKS Command Block*

**WSmk**

A Write Setmark (WSmk) bit of one indicates that the Write Filemarks command should write the requested number of *setmarks*. A Write Setmark (WSmk) bit of zero indicates that the Write Filemarks command should write the requested number of *filemarks*.
Setmarks are useful to help structure the tape contents. One tape may contain several *sets* separated by SETMARKS. Each set may contain several *files* separated by FILEMARKS. Each file will again contain several *blocks*. Note that *setmarks* are only legal when operating in *QIC-525/1000/2GB/96-34 mode.*

**IMM**

An Immediate (IMM) bit of one indicates that the Target will return status as soon as the Command Descriptor Block has been validated. An IMM bit of one is only valid if buffered mode is reported in the mode parameter header. An IMM bit of zero indicates that the Target will not return status until the Write operation has completed. Any buffered data, filemarks and setmarks will be written to the medium prior to completing the command.

**Number Of Filemarks**

This field specifies the number of filemarks to be written. A zero value will write no filemarks. This can be used to force any buffered write data or filemarks to be written to the tape. When this field is zero the Drive will not return status until all data and filemarks have been written (the IMM bit is just ignored in this case). The range of legal values are 0..65535.

## 33.3. Terminating Write Operations

See the WRITE command section.

## 33.4. Write Filemarks from BOM

See the WRITE command section.

## 33.5. Exception Handling

### *33.5.1. General*

See sections on Error Conditions For All Commands, Deferred Errors and Error Conditions For Media Access Commands.

If both the IMM and the Link bits are set to one the Drive will terminate the WRITE FILEMARKS command with CHECK CONDITION status. The Error Code will be set to E$STE_IFIC. The logical tape position will not be changed.

When the WRITE FILEMARKS command has started execution, all detected errors will set the VADD bit and the Information Bytes will hold the difference between the requested and the actual filemark count. See the following sections for details.

If the inserted cartridge is write-protected, the WRITE FILEMARKS command will be terminated with CHECK CONDITION status. No filemarks will be written. The Error Code will be set to E$BTD_WPRO. This will be true even if the requested Number Of Filemarks is zero.

If the WSmk (Write Setmark) bit is set to one when the current tape format is not set to QIC-525/1000/2GB/96-34, the Drive will terminate the WRITE FILEMARKS command with CHECK CONDITION status. The Error Code will be set to E$STE_IFIC.

### 33.5.2. Illegal Media Type

All tape formats will not be legal on all media (cartridge) types. When an illegal combination is detected, the Drive will terminate the WRITE FILEMARKS command with CHECK CONDITION status. The Error Code will be set to E$BTD_CFMT. No filemarks will be written.

See the WRITE command for illegal media/format combinations.

### 33.5.3. Illegal Append Tape Format

When appending data on a pre-recorded tape the only legal tape format is the format found on the tape. If the format set by the MODE SELECT command is not equal to the tape format, the Drive will terminate the WRITE FILEMARKS command with CHECK CONDITION status. The Error Code will be set to E$BTD_TFMT. No filemarks will be written.

If the tape format is QIC-525/1000/2GB/96-34 and the last block found on the tape is not the last ECC block in a frame, the Drive will terminate the WRITE FILEMARKS command with CHECK CONDITION status. The Error Code will be set to E$BTD_APUF. No filemarks will be written.

### 33.5.4. Pseudo Early Warning

If pseudo early warning (PSEW) tape marker is reached during the execution of a WRITE FILEMARKS command, the Drive will respond as described below:

- *The Drive will attempt to write all buffered filemarks. The WRITE FILEMARKS command will then be terminated with CHECK CONDITION. If all filemarks and data in the buffer were successfully written to the tape, the Error Code will be set to E$BTD_-PSEW and the Sense Key will be set to NO SENSE. The EOM and VADD bits will be set to one. The Information Bytes will be set to zero.*
  *The Drive will force unbuffered mode (regardless the state of the BM bit in the MODE SELECT/SENSE Parameter List).*

- *If a read type command (COPY - copy function = restore), READ, LOCATE, SPACE or VERIFY), has brought the tape past the PSEW marker and an Append Operation is attempted, then the Drive will terminate the first WRITE FILEMARKS command immediately with CHECK CONDITION. No filemarks will be written. The Error Code will be set to E$BTD_PSEW and the Sense Key will be set to NO SENSE. The EOM and VADD bits will be set to one. The Information Bytes in the sense data list will be set equal to the requested transfer length.*
  *The Drive will force unbuffered mode (regardless the state of the BM bit in the MODE SELECT/SENSE Parameter List).*

- *If another WRITE FILEMARKS command is received by the Drive while the tape is positioned after PSEW (but before end of partition), the Drive will transfer and write all requested data if possible. The WRITE FILEMARKS command will then be terminated with CHECK CONDITION (this will be true even if the* *requested Number of Filemarks is zero). If all filemarks were successfully written to the tape the Error Code will be set to E$BTD_PSEW and the Sense Key will be set to NO SENSE. The EOM and VADD bits will be set to one. The Information Bytes will be set to zero.*

### 33.5.5. End of Partition

If end of partition is reached during the execution of a WRITE FILEMARKS command, the Drive will terminate the command with CHECK CONDITION status. The Error Code will be set to E$WRT_EOM and the Sense Key will be set to VOLUME OVERFLOW. The Valid (VADD) and End Of Media (EOM) bits will be set to one.

The Information Bytes in the sense data list will be set to the difference between the requested number of filemarks and the actual number of filemarks transferred to the Drive's data buffer.

This error condition has priority over the Pseudo Early Warning error condition.

The Initiator will be able to read back buffered filemarks or data by issuing the RECOVER BUFFERED DATA command. The logical tape position will be undefined. Additional WRITE FILEMARKS commands issued after an end-of-partition error has occurred will be terminated immediately with a CHECK CONDITION (no filemarks will be transferred). The end-of-partition condition will persist until a position type command has been executed (ERASE, LOAD/UNLOAD or REWIND).

### 33.5.6. Non-Recoverable Write Error

If a non-recoverable write error occurs, the Drive will terminate the WRITE FILEMARKS command with CHECK CONDITION status. The Error Code will be set to E$WRT_REWRITE and the Sense Key will be set to MEDIA ERROR. The Valid (VADD) bit will be set to one.

The Information Bytes in the sense data list will be set to the difference between the requested number of filemarks and the actual number of filemarks transferred to the Drive's data buffer.

This error condition has priority over the Pseudo Early Warning error condition.

The Initiator will be able to read back buffered filemarks and data by issuing the RECOVER BUFFERED DATA command. The logical tape position will be undefined. Additional WRITE FILEMARKS commands issued after a non-recoverable write error has occurred, will be terminated immediately with a CHECK CONDITION (no filemarks will be transferred). The non-recoverable write error condition will persist until a position type command has been executed (ERASE, LOAD/UNLOAD or REWIND).

### 33.5.7. Append Error

When the WRITE FILEMARKS command is executed after a COPY (copy function = restore), READ, LOCATE, SPACE, VERIFY or WRITE FILEMARKS command where the tape motion for some reason has stopped, the Drive must seek the last block on the tape before the write operation starts. If this seek operation fails, the Drive will terminate the WRITE FILEMARKS command with a CHECK CONDITION status. The Error Code will be set to E$WRT_APFAIL and the Sense Key will be set to MEDIA ERROR. The Valid (VADD) bit will be set to one.

The Information Bytes in the sense data list will be set to the difference between the requested number of filemarks and the actual number of filemarks transferred to the Drive's data buffer.

This error condition has priority over the Pseudo Early Warning error condition.

The Initiator will be able to read back buffered filemarks and data by issuing the RECOVER BUFFERED DATA command. The logical tape position will be undefined. Additional WRITE FILEMARKS commands issued after a non-recoverable write error has occurred, will be terminated immediately with a CHECK CONDITION (no data will be transferred). The append error condition will persist until a position type command has been executed (ERASE, LOAD/UNLOAD or REWIND).

*This Page Intentionally Left Blank*

# A. Application Note 1: SCSI-Bus Communication During the Copy Command

This application note is intended for technical personnel that is familiar to the SCSI interface and particularly the Copy command.
The application note will try to describe all possible activity on the SCSI-bus from the point where the Host selects the Drive in order to issue a COPY command, until the Copy operation is terminated.
It is recommended that the user also refers to Chapter 8. Copy.

## A.1. Host - Tape Drive Communication

### *A.1.1. Start - Issuing the Copy Command With Parameters*

#### A.1.1.1. Arbitration/Selection Phase

Normal, ATN must be asserted in order to enable for a Message Out Phase.

#### A.1.1.2. Message Out Phase

The DscP (Disconnect Privilege) bit must be set in the Message Out byte which specifies that the Host has granted the Drive the privilege of disconnecting.

#### A.1.1.3. Command Phase

Normal, six byte Command Descriptor Block

#### A.1.1.4. Data Out Phase

If the number of parameters, specified in the three byte "Length of Parameter List" field of the CDB is greater than zero, these bytes are transferred in this phase. No matter if there are some wrong values in the parameters nor if the number of parameter bytes are invalid, all bytes are transferred. There is, however, one restriction: No more than 3076 bytes are transferred, even though the number specified in the CDB is larger.
The bytes are DMA transferred into the buffer RAM of the Drive.

#### A.1.1.5. No Problems Detected After the Data Out Phase

**Message In Phase (2 bytes)**

If the Drive (management part) does not find any problems with the parameters (Segment Descriptor Lists and Header) or the Message Out byte, the following two Message In bytes will be transferred to the Host:

02 - Save Data pointer
04 - Disconnect

**Bus Free**

Then Bus Free is entered. The Host will now expect the Drive to reconnect when the Copy operation is finished or aborted. See Section A.1.2 "End - Tape Drive Reconnecting and Status Reporting".

#### A.1.1.6. Problems Detected After the Data Out Phase

**Status Phase**

If the Drive (management part) does find an error in the parameters (Segment Descriptor Lists and Header) or the Message Out byte a Check Condition Status byte will be transferred to the Host:

02 - Check Condition

**Message In**

Lastly, a Command Complete Message will be sent to the Host:

00 - Command complete

### *A.1.2. End - Tape Drive Reconnecting and Status Reporting*

#### A.1.2.1. No Problems Detected During the Copy Operation

If the Drive has performed the complete Copy operation and not detected any errors, the phases to follow will be:

**Status Phase**

00 - Good Status

**Message Phase**

00 - Command Complete

**Bus Free**

If the Copy operation performed was a Backup, and no more writing operations is to be done, it is recommended that the whole operation is terminated by a Rewind Command.

#### A.1.2.2. Problems Detected During the Copy Operation

If any problems are detected by the Drive during the Copy operation, the phases to follow will be:

**Status Phase**

02 - Check Condition

**Message In Phase**

00 - Command Complete

**Bus Free**

At this point it is wise (for the Host) to issue a Request Sense Command to get hold of the reason why the Copy operation was aborted. The number in the "Allocation Length" field of the Request Sense CDB should be large enough so that possible Source/Destination Bytes will be transferred.

# A.2. Tape Drive - Hard Drive Communication

## A.2.1. General Rules

### A.2.1.1. Tape Drive, the Initiator

The Drive will switch from Target mode (when it receives the Copy Command and parameter bytes from the Host) to Initiator mode. It is as an Initiator it performs the Copy operation, and the Hard Drive will be the Target of the Drive.

When the Copy operation is completed the Drive resumes the Target role, and reselects the Host.

### A.2.1.2. Tape Drive Selection of Hard Drive

After the Drive has won the arbitration of the SCSI-bus the Selection Phase is entered. If the selection attempt is non-successful (e.g. the Target does not respond within 250 ms) the Drive will, up to 255 times, re-enter the arbitration phase (of course after Bus Free is detected) and try again. If the 256th attempt also is negative, the Copy operation is aborted. Internal error code will be: E$SIP_CISE.

NOTE:
Due to the fact that the selection timeout period is 0.25 seconds, 255 selection attempts will be quite time consuming (64 seconds). This will be the case if the Hard Drive specified in the current Segment Descriptor List (SDL) is not powered up or not present on the SCSI-bus.

### A.2.1.3. Hard Drive Return BUSY Status

If the Hard Drive returns a BUSY status as a response to a command issued by the Drive, the procedure will be as follows:

After the Message In message (Command Complete) is received after the BUSY status, Bus Free Phase is entered. The Drive will then try to issue the command again (go through ARB/SEL/MSG OUT/CMD phases). The Drive accepts up to 255 BUSY statuses on the same command before the Copy operation is aborted. Internal error code will be: E$SIP_CIBS.

### A.2.1.4. Hard Drive Return RESERVATION CONFLICT Status

If the Hard Drive returns a RESERVATION CONFLICT status as a response to a command issued by the Drive, the procedure will be as follows:

After the Message In message (Command Complete) is received after the RESERVATION CONFLICT status, Bus Free Phase is entered. The Drive will then try to issue the command again (go through ARB/SEL/MSG OUT/CMD phases). The Drive accepts up to 255 RESERVATION CONFLICT statuses on the same command before the Copy operation is aborted. Internal error code will be: E$SIP_CIBS.

### A.2.1.5. Hard Drive Disconnect Privilege

After the Drive successfully has selected the Hard Drive it will, since the ATN signal is set along with the SEL signal, enter the Message Out phase and **always** set the DscP bit in the Message Out byte. This means that the Drive allows the Hard Drive to disconnect whenever it likes.

### A.2.2. Initial Procedure Tape Drive - Hard Drive

In both Backup and Restore mode, the first task of the Drive will be to establish the block size of the Hard Drive. This is done as follows:

Issue a Read Capacity command to the Hard Drive. If the Hard Drive supports this command, and no problems occur, the Data In phase is entered after the Command phase. Eight bytes are transferred in this Data In phase, and by analyzing these bytes the block size is found.

If the status Phase is entered after the Command phase, and the status byte is 02 (Check Condition) the following procedure is followed: (If status is BUSY or RESERVATION CONFLICT, see Sections A.2.1.3. and A.2.1.4).

Issue a Request Sense command. The number of requested bytes is four. The reason why only four bytes are requested is simply that on expects that the reason to the Check Condition is a Unit Attention message.

Then a new Read Capacity command is issued. If a new Check Condition status is received, the code assumes that the Read Capacity command is not supported by the Hard Drive. A Request Sense command (again four bytes is requested) is then issued to simply clear this message.

A Read (one block) command is then issued to the Hard Drive. If no error occur, one Hard Drive block of data is transferred to the Drive in the Data In phase following the Command phase. The Drive counts the received bytes and take no interest in the data. In this way the Drive establishes the Hard Drive block size simply by counting.

If, for any odd reason, the Hard Drive responds to the Read command with a Check Condition, a new Request Sense command is issued to the Hard Drive. This time however, the number requested is according to the Drive's COPY SENSE ALLOCATION value (see Mode Select chapter in manual). The Copy operation is now to aborted since the Hard Drive block size was not found, and in order to inform the Host in the best possible way, a Request Sense Parameter List is built according to the Copy Sense Allocation value. (See Section A.1.2.2. "Problems Detected During the Copy Operation" for more information).

If the Drive, by using the above described procedure, establishes the Hard drive block size, the Backup/Restore operation is ready to begin.

### A.2.3. Copy Operation - Overview

The parameters passed to the Drive in the initial Data Out phase will normally include one header (4 bytes) and one or more (up to 256) Segment Descriptor Lists, each of 12 bytes. If no problems are found regarding the parameters (see Section A.1.1.6. "Problems Detected After the Data Out Phase"), the Copy operation starts.

As described earlier, the parameters are initially stored in the buffer RAM. From now on, one SDL at the time is copied to the work RAM and then processed. The initial procedure, described in Section A.2.2, will be performed on every SDL even though the block size of the Hard Drive is established earlier in the Copy operation.

The type of Copy operation is determined by the first byte of the header block. The current supported Copy functions are Backup (read data from Hard Drive and store on Drive) or Restore (read data on Drive and store on Hard Drive). Note: only one type of Copy function can be performed in a single Copy command.

### A.2.3.1. Copy Function: Backup

When performing a backup operation Read commands are sent to the Hard Drive, and the received data are stored on the Drive's tape. The number of Hard Drive blocks to be read and the address of these blocks are found in the current processed SDL.

If the number of blocks to be read exceed the selectable variable Copy Threshold (see Mode Select chapter in manual), the read operation is done in portions, the Copy Threshold number of blocks at the time. The last Read Command will possibly request less than Copy Threshold blocks.

If the Hard Drive Block address is greater than ((2**21)-1) then a 10 byte CDB is used by the Drive. Else a 6 byte CDB is used.

### A.2.3.2. Copy Function: Restore

When performing a restore operation the Drive reads data from the tape, and transfers/writes the data on the Hard Drive. The number of Hard Drive blocks to be written and the address of these blocks are found in the current processed SDL.

If the number of blocks to be written exceed the selectable variable Copy Threshold (see Mode Select chapter in manual), the write operation is done in portions, the Copy Threshold number of blocks at the time. The last Write Command will possibly request less than Copy Threshold blocks.

If the Hard Drive Block address is greater than ((2**21)-1) then a 10 byte CDB is used by the Drive. Else a 6 byte CDB is used.

## A.3. Parity Error Handling

The parity detection is only operative when the parity jumper is installed.

### A.3.1. Parity Error During the Data In Phase

If a parity error is detected by the Drive during a data transfer from the Hard Drive, the ATN signal is asserted. This will force the Hard Drive to enter the Message Out phase. The message INITIATOR DETECTED ERROR is then transferred to the Hard Drive. According to the SCSI standard the target (Hard Drive) should then retransfer the last sent data(burst).

### A.3.2. Parity Error During the Data Out Phase

If a parity error is detected by the Hard Drive during a data transfer from the Drive, the target is free to ask for a retransfer of these last sent data. The retransfer is initiated by the Hard Drive by sending a RESTORE POINTER message (Message-In phase) to the Drive. When the Hard Drive next enters the Data Out phase, the last sent data is retransferred.

### A.3.3. Parity Error During the Message In Phase

If a parity error is detected by the Drive when receiving a message from the Hard Drive, the ATN signal is asserted. This will force the Hard Drive to enter the Message Out phase. The message MESSAGE PARITY ERROR is then transferred to the Hard Drive. According to the SCSI standard the target (Hard Drive) should then retransfer the last sent message.

### A.3.4. Parity Error During the Message Out Phase

If a parity error is detected by the Hard Drive when receiving a message from the Drive, the target is free to ask for a retransfer of this message. The retransfer is simply initiated (by the Hard Drive) by continuing the Message Out phase. The Drive will, when it finishes the original message transfer, detect that the Hard Drive still holds the Message Out phase and for that reason retransfer the Message byte(s).

### A.3.5. Parity Error During the Status Phase

If a parity error is detected by the Drive when receiving a status byte from the Hard Drive, the ATN signal is asserted. This will force the Hard Drive to enter the Message Out phase. The message INITIATOR DETECTED ERROR is then transferred to the Hard Drive. According to the SCSI standard the target (Hard Drive) should then retransfer the status byte.

### A.3.6. Parity Error During the Command Phase

If a parity error is detected by the Hard Drive during the transfer of the Command Descriptor Block from the Drive, the Target is free to ask for a retransfer of the CDB. The retransfer is initiated by the Hard Drive by sending a RESTORE POINTER message (Message-In phase). When the Hard Drive next enters the Command phase, the last sent CDB is retransferred.

# A.4. Limitations and Other Curiosities

## A.4.1. Test of Parameter List (Header and SDLs)

The following items are **not** tested (before the actual SDL is processed):

Test to see if the Hard Drive is present on the SCSI-bus.
Test to see if the total number of blocks in all SDLs is an even number if the Hard drive block size is 256. This would cause the Copy operation to start, but when the Drive detects the total number of Hard Drive blocks is odd (right before it's to process the very last SDL) it has to abort the whole Copy operation.

## A.4.2. Copy Operation With More Than One Hard Drive

As long as the hard Drive block sizes are 512 bytes or greater, this should not cause any problems. However, if the one of the Hard Drives involved has a block size of 256 byte blocks and the other(s) have a blocks size 512 or greater on can run into problems. This is related to the fact that the Drive's block size is 512 bytes (always) and two and two 256 byte block are put together to make a 512 bye block.
A good rule is to assure that when involving a Hard Drive with a block size of 256 bytes, one should always have an even number of blocks specified in the Direct Access Device Number of Blocks field.

## A.4.3. Zero in the SDL Direct Access Device Number of Blocks Field

The initial procedure, described in Section A.2.2, will be performed even though no blocks are to be transferred to/from the Drive.
Such a Copy operation, not involving any reading or writing on the Drive is not considered as an error, and will subsequently report a GOOD status.

*This Page Intentionally Left Blank*