HCSG

SELFSCAN

The Next Generation

Command Description Manual

Grand Prix

PRELIMINARY
June 1994

James Byrd

Table of Contents

Overview 1
Getting Started 1
Selfscan Test Cylinder 2
- SCW Command Chain File 2
- SSW Result File 3
- Defect File 3
- Selfscan Chain Results Tail Status Word (SSW Tail) 4

SCW HEADER, Command 0x01 7
- Command Description 7
- Basic Operation 7
- Chaining Restrictions 8
- SCW Compiler Mnemonics 9
- SCW Data Parameters 9
- Ending Status 11
- SSW Print Display Example 13

WRITE ICL, Command 0x02 14
- Command Description 14
- Basic Operation 14
- Chaining Restrictions 15
- SCW Compiler Mnemonics 15
- SCW Data Parameters 15
- Ending Status 15
- SSW Print Display Example 16

ALTERNATE SEEK, Command 0x03 17
- Command Description 17
- Basic Operation 17
- Chaining Restrictions 18
- SCW Compiler Mnemonics 18
- SCW Data Parameters 18
- Ending Status 20
- SSW Print Display Example 21

SINGLE TRACK SEEK, Command 0x03 22
- Command Description 22
- Basic Operation 22
- Chaining Restrictions 23
- SCW Compiler Mnemonics 23
- SCW Data Parameters 23
- Ending Status 25
- SSW Print Display Example 27

THIRD STROKE SEEK, Command 0x03 28
- Command Description 28
- Basic Operation 28

Chaining Restrictions 29
SCW Compiler Mnemonics....29
SCW Data Parameters 29
Ending Status 31
SSW Print Display Example....33
FULL STROKE SEEK, Command 0x03 34
Command Description 34
Basic Operation 34
Chaining Restrictions 35
SCW Compiler Mnemonics....35
SCW Data Parameters 35
Ending Status 37
SSW Print Display Example....39
HEAD SWITCH, Command 0x04....40
Command Description 40
Basic Operation 40
Chaining Restrictions 41
SCW Compiler Mnemonics....41
SCW Data Parameters 41
Ending Status 42
SSW Print Display Example....44
RANDOM SEEK, Command 0x05 45
Command Description 45
Basic Operation 45
Chaining Restrictions 45
SCW Compiler Mnemonics....46
SCW Data Parameters 46
Ending Status 47
SSW Print Display Example....48
FORMAT MEDIA, Command 0x06....49
Command Description 49
Basic Operation 49
Chaining Restrictions 50
SCW Compiler Mnemonics....50
SCW Data Parameters 50
Ending Status 50
SSW Print Display Example....51
ERASE PASSWORD, Command 0x07 52
Command Description 52
Basic Operation 52
Chaining Restrictions 52
SCW Compiler Mnemonics....52
SCW Data Parameters 53
Ending Status 53
SSW Print Display Example....53

TIC, Command 0x08 54
- Command Description 54
- Basic Operation 54
- Chaining Restrictions 55
- SCW Compiler Mnemonics 55
- SCW Data Parameters 55
- Ending Status 55
- SSW Print Display Example 55

SEQUENTIAL DEFECT, Command 0x09 56
- Command Description 56
- Basic Operation 56
- Chaining Restrictions 58
- SCW Compiler Mnemonics 58
- SCW Data Parameters 58
- Ending Status 64
- SSW Print Display Example 65

STOP START, Command 0x0A 66
- Command Description 66
- Basic Operation 66
- Chaining Restrictions 67
- SCW Compiler Mnemonics 67
- SCW Data Parameters 67
- Ending Status 68
- SSW Print Display Example 69

DIGITAL DEFECT, Command 0x0B 70
- Command Description 70
- Basic Operation 70
- Chaining Restrictions 72
- SCW Compiler Mnemonics 72
- SCW Data Parameters 72
- Ending Status 78
- SSW Print Display Example 80

FIR TRAINING, Command 0x0C 81
- Command Description 81
- Basic Operation 81
- Chaining Restrictions 82
- SCW Compiler Mnemonics 82
- SCW Data Parameters 83
- Ending Status 84
- SSW Print Display Example 85

Appendix A 86
- LED Error Codes 86

Appendix B 86
- Termination Error Codes 86

Appendix C 87

SCW Error Codes....87
Appendix D....89
Error Code Index....89
Appendix E....92
Model.Def....92
Appendix F....92
Selfscan.Def....92
Appendix G....97
SSW.Def....97
Index....103

Disclaimer

NO WARRANTIES OF ANY NATURE ARE EXTENDED BY THIS DOCUMENT. Only the user of this document shall accept any financial or other responsibility (INCLUDING ANY LOST PROFITS, LOST SAVINGS, OR OTHER INCIDENTAL DAMAGES ARISING OUT OF THE USE OF OR INABILITY TO USE SAID DOCUMENT) that may be the result of your use of information contained in this document, including direct, indirect, special or consequential damages. The information contained in this document has been obtained on a "hearsay" basis and is subject to change without notice. The contents of this document may be revised without obligation to notify any person or organization of these changes.

Overview

Selfscan was introduced to reduce production costs by having the drive test itself without the need for special equipment. Drive self-test eliminates the special hardware required for function test and digital scan. This manual provides you with the knowledge of how the Selfscan command processor works, how to write a Selfscan command file, and how to interpret the Selfscan results file.

Selfscan is a command processor with a set of commands stored in a file on the drive's test cylinder. The Selfscan command processor reads the command file from the disk. Each command is executed and the results for each command are stored in a result file. The defect scanning commands maintain a defect list file.
The Selfscan command processor is not a C compiler (it does not perform do, for, while, else, or if statements. There is not enough memory, it runs one command at a time, then executes the next one. The TIC (transfer in command) command and check address in the SCW (Selfscan Command Word) allows for a "limited" program execution modification.

The Selfscan command processor requires the system cylinders to be formatted, and the Selfscan command file written on the test cylinder. The Selfscan command processor starts at power up if a valid command file has been written to the test cylinder. When the Selfscan command chain finishes the SCSI interface is enabled with a "limited" command set. The limited set does not include the MESSAGE OUT phase (i.e. synchronous transfer) the drive will "lock up", or any mode sense, mode select, read, write, read extended, or write extended commands. ECC correction is disabled for Selfscan.

James (Jay) Byrd designed the Selfscan Next Generation command processor in 1993. The interface between a Selfscan command and the command processor is similar to the interface of the IBM 370. Mark Thomas designed the Selfscan Next Generation compiler and disassember.

This document contains a description of each Selfscan command. For a theory of operation, or information how to use the compiler and disassembler see the Selfscan Next Generation User's Guide.

Getting Started

This manual describes each Selfscan command for the Grand Prix hard disk drive. Because the defines files may be altered, only the primary compiler mnemonic of each command is described.

For many Selfscan commands, the default mode and configuration pages are used as parameters. The retry count for many Selfscan commands uses the retry count in Mode Select Page 1. This value is defaulted to eight retries in manufacturing. When the Selfscan code if FLASHED in the drive, the Mode Select command is invalid. Therefore all the Mode Pages must be set correctly before FLASHING the drive with SELFSCAN code!

The Configuration Pages may be altered when the Selfscan code is loaded.

Selfscan Test Cylinder

The Selfscan command processor requires three files for operation. The first file is the Selfscan command file, which contains the commands necessary for Selfscan command processor execution. The Selfscan command file is the only file that must be written with the write physical command using the drive's interface. The second file is the Selfscan result file, which is initialized by the Selfscan command processor during the SCW header command. The Selfscan result file contains status returned from each of the Selfscan commands executed. The third file is the Selfscan defect list, which is also initialized by the SCW header command. The Selfscan defect list contains all the defects located by the Selfscan defect scanning commands. All three Selfscan files are written with all the heads of the drives Selfscan test cylinder. The Selfscan files are located on the drive's test cylinder as follows:

File Description	Cylinder	Sector, Size	Data Section
Command File	- 1	30, 6	0400h
Result File	- 1	36, 12	0800h
Defect File	- 1	48, 12	0030h

Table 1

SCW Command Chain File

The Selfscan command processor reads the command file into the drive's data buffer. The command processor starts reading at the highest head number until a file is read without errors. If an error is detected in the file, the head number is decremented and the next track is read. The SCW command file is a binary file generated from a Selfscan source file using the Selfscan compiler (See Selfscan Next Generation User's Guide). The SCW command file is divided by the Selfscan compiler into two sections. The first section (starting at byte 0000) is the command section consisting of a "chain" of commands for the command processor to execute. The data section (starting at byte 0400h) contains data parameters required for each of the Selfscan commands.

SSW Result File

The Selfscan command processor reads this file into the drive's data buffer. The SSW result file is divided by the Selfscan command processor into two sections. The first section (starting at byte 0000) is the result command section consisting of a "chain" of Selfscan Status Words (SSW) from the Selfscan Commands. The data section (starting at byte 0800h) contains results data parameters returned from each of the Selfscan commands.

The result file contains a command history of all the SCWs executed by the Selfscan command processor since the initialization of the result file. The SSW of each SCW command executed is stored in the result file with the final ending status, error codes, result data, and trace address. This list of status words from the command chain provides a "trace" of the command execution.

Defect File

The first eight bytes, "DEFECT L", identify the file as a valid Selfscan defect list. The next 32 bytes are the "Wedge Skews" for each zone (two bytes each). The "Wedge Skews" are the same for each zone and not extremely useful (this space is intended for "Sector Skew"). The defect count (0028h) indicates the number of valid defects in the defect file. There are two types of assigned defects: the sector defect and the bytes from wedge defect. The defects are assigned in the order they are located, and are not sorted in the defect file. The first defect assigned defect is at location 0030h in the defect file.

Selfscan Chain Results Tail Status Word (SSW Tail)

0	1 2	3	4 5	6	7
0Fh	Data Address	Flag	SCW Address	Status	0X8a

Selfscan Chain Results Tail Status Word

The Chain Results Tail Status Word is the final results of the Selfscan command chain ran. The SSW tail is automatically written at the end of the last SSW in the result file. The results tail data contains the final ending results of the command chain and a list of command chain statistics. The results tail statistics are kept for all the SCW chains since the results file was initialized.

The results tail SSW is loaded into the command processor's result block at the start of a command chain by the SCW header command. The error codes from the previous chain are cleared (the error codes of any failing SSW are still in the results file at the SSW), and the command chain statistical block is updated after each SCW executed by the Selfscan command processor. The results tail data presented by the SSW TAIL is as follows:

Selfscan Chain Results Tail Data

Byte	Symbol	Description
0	term_error	Termination Error Code
1	led_error	LED Error Code
2	error_code	Error Code
3	error_code_index	Error Code Index
4-5	scw_this_chain	Number of SCWs this chain
6-7	scw_all_chains	Number of SCWs all chains
8-9	time_this_chain	Total execution time this chain (4 second resolution)
10-11	time_all_chains	Total execution time all chains (4 second resolution)
12-13	soft_errors	Total soft errors all chains
14-15	hard_errors	Total hard errors all chains
16-17	all_seek_errors	Total seek errors all chains
18-57	soft_hd_errors	Total soft errors per head, (heads 0 - 19)
58-97	hard_hd_errors	Total hard errors per head, (heads 0 - 19)
98-137	wiggle_errors	Total wiggle errors per head, (heads 0 - 19)

Table 2

This section gives a detailed description of the SSW data parameter bytes by the disassembler mnemonic:

term_error

The term_error (byte 0) contains the reason the Selfscan command processor terminated the command chain. A command check, program check, or exception status must be presented for a valid error code.

led_error

The led_error (byte 1) contains the final ending status from the Selfscan command processor. A continuous flashing LED indicates a "good" ending status. The Selfscan command processor indicates an error code by turning the LED off for 4 seconds, then flashing the LED to indicate an error code. The number of times the LED flashes is equal to the error code number. A command check, program check, or exception status must be presented for a valid error code.

error_code

The error_code (byte 2) contains the reason the SCW was terminated. A command check, program check, or exception status must be presented for a valid error code.

error_code_index

The error_code_index (byte 3) is the sense key index value returned by internal subroutines in the drives operational firmware. This error code is an engineering error code to help determine the exact cause of the failure. A command check, program check, or exception status must be presented for a valid error code.

scw_this_chain

The scw_this_chain (bytes 4 - 5) is the total number of SCW commands executed in this chain. The number of previous SCW's is lost when a new chain is started.

scw_all_chains

The scw_all_chains (bytes 6 - 7) is the total number of SCW commands executed by all chains since the results file was initialized.

time_this_chain

This is the total time (bytes 8 - 9) is in four second increments that the SCW chain required to execute. The previous chain times are lost when a new chain is started.

time_all_chains

This is the total time (bytes 10 - 11) is in four second increments that all the SCW chain required to execute since the result file was initialized.

soft_errors

The soft_errors (bytes 12 - 13) is the total number of soft errors that occurred for all SCW chains since the result file was initialized. A soft error occurs when a track is read, and a sector returns with a medium error that does not occur again, or wiggle recovery fixed for the retry loops.

hard_errors

The hard_errors (bytes 14 - 15) is the total number of hard errors that occurred for all SCW chains since the result file was initialized. A hard error occurs when a track is read and a sector returns with a medium error that is repeatable. Hard error counts are incremented each time an error is detected. A defect is only assigned if the sector is not in the defect list.

all_seek_errors

The all_seek_errors (bytes 16 - 17) is the total number of recovered seek errors that occurred for all SCW chains since the result file was initialized.

soft_hd_errors

The soft_hd_errors (bytes 18 - 57) is the total number of soft errors per head that occurred for all SCW chains since the result file was initialized, and is a breakdown of the total soft errors to check for unsatisfactory heads. The data starts with head 0 (low byte, high byte) and continues to the last head in the drive.

hard_hd_errors

The hard_hd_errors (bytes 58 - 97) is the total number of hard errors per head that occurred for all SCW chains since the result file was initialized, and is a break down of the total hard errors to check for unsatisfactory heads. The data starts with head 0 (low byte, high byte) and continues to the last head in the drive.

wiggle_errors

The wiggle_errors (bytes 98 - 137) is the total number of wiggle errors per head that occurred for all SCW chains since the result file was initialized, and is a break down of the total wiggle errors to check for unsatisfactory heads. The data starts with head 0 (low byte, high byte) and continues to the last head in the drive.

SCW HEADER, Command 0x01

0	1 2	3	4 5	6	7
0x01	Data Address	Flags	0x0000	0	0x1E

Selfscan Command Word

Command Description

The SCW Header command is the "boot strap" command for the Selfscan command processor. The SCW Header command loads the command processor's chain memory block with the maximum chain limits and the previous statistical results block. The SCW initializes the result and defect files. The SCW Header command must be the first SCW in the command chain. Only one SCW Header command per chain is allowed.

Basic Operation

The SCW Header checks for a valid Selfscan password ("SELFSCAN"). If the password is valid, the command processor is initialized so that the command chain executes. The first time a Selfscan command chain is executed, the results and defect files MUST be initialized by setting the program switches in the command data SCW Header flag byte. After the results and defect files are initialized, the Selfscan command processor keeps a "command history" of all the command chains, defects, and drive statistics.

The following operations are performed to initialize the command processor:

- Check the validity of the SCW bytes.
- Check for the first SCW in chain flag.
- Check for no SCC, SPC, or SIL.
- Check to see if the "check address" equals zero.
- Check for the password "SELFSCAN".
- Check for initialize results file in header flag byte, then initialize the results file.
- Check for initialize defect file in header flag byte, then initialize the defect file.

- Check for a valid results file password ("RESULT D").
- Check for space in the result file.
- Load maximum limits in memory chain block.
- Load results file indices in the command processor's chain memory block.
- Load result tail data into the command processor's result block.
- Check for a valid defect password ("DEFECT L").
- Present ending status.

Chaining Restrictions

- The SCW Header command must be the first SCW in a command chain, otherwise a program check is presented.
- Only one SCW Header command may be in a chain, otherwise a program check is presented.
- The Suppress Program Check may not be set, otherwise a program check is presented.
- The Suppress Command Check may not be set, otherwise a program check is presented.
- The Suppress Incorrect Length may not be set, otherwise a program check is presented.
- The Check Address must be zero (no check address), otherwise a program check is presented.
- The Selfscan password in the data parameter bytes must be "SELFSCAN", otherwise a program check is presented.
- The results file must be initialized or the flag byte (byte 16) in the data parameters must specify to initialize the results file, otherwise a command check is presented.
- The defect file must be initialized or the flag byte (byte 16) in the data parameters must specify to initialize the defect file, otherwise a command check is presented.

SCW Compiler Mnemonics

The Valid mnemonics for the Selfscan Next Generation Compiler are:

SCW_HEADER	HEADER

SCW Data Parameters

The SCW data parameter bytes provide the additional specifications to execute a Selfscan command. The following table gives the byte location, compiler mnemonic, the default value, and a short description of the SCW data parameters:

SCW Header Data Parameters

Byte	Mnemonic	Default	Description
0 - 7	password	SELFSCAN	Selfscan password
8 - 15	version	VER 1.00	Version Label
16	flags	0	SCW Header flag byte
17	trace	0	Trace Byte, scope trigger
18 - 19	max_time	7200	Maximum SCW chain time (4 second increments)
20 - 21	max_scw_time	180	Maximum SCW time (4 second increments)
22 - 23	max_assign_defect	400	Maximum number of assigned defects, all chains
24 - 25	max_hard_errs	0xFFFF	Maximum hard errors
26 - 27	max_hard_head_errs	0xFFFF	Maximum hard errors per head
28 - 29	max_soft_seek_errs	0xFFFF	Maximum recovered seek errors

This next section is a detailed description of the SCW data parameters by the compiler's mnemonic:

password — The password (bytes 0 - 7) is used by the command processor to determine if the command chain file is valid. The Erase Password command changes the password to prevent the Selfscan command chain from executing again. The compiler password default value of "SELFSCAN" allows the command chain to execute.

version — The version (bytes 8 - 15) is used to describe the command chain. This eight byte string is not used by the command, the string is presented in the SSW data parameter bytes to identify the chain.

flags

The flags (byte 16) is the flag byte for the Selfscan command processor. This byte instructs the command processor to initialize the results or defect files. The flag byte is zero by defaulted so a command history is maintained by the command processor. The compiler has three equates to set the flag byte for initialization:

initialize	:	03h	Initialize the results and defect files.
init_result	:	01h	Initialize the result file only.
init_defect	:	02h	Initialize the defect file only.

trace

The trace (byte 17) is a compare byte to set and clear an oscilloscope trigger pin. The trace byte is a special engineering debug byte and should be set to 00h at all times.

max_time

The max_time (bytes 18 - 19) specifies the maximum time the command chain is allowed to execute. The time is specified in four second increments and is defaulted to eight hours by the compiler. If the time is exceeded, the current SCW command is terminated with a command check.

max_scw_time

The max_scw_time (bytes 20 - 21) specifies the maximum allowed time for each SCW executed. The time is specified in four second increments and is defaulted to five minutes by the compiler. If the time is exceeded, the SCW command is terminated with a command end, and an exception is raised. Some SCWs may alter the maximum SCW time and override the preset value.

max_assign_defect

The max_assign_defect (bytes 22 - 23) specifies the maximum number of assigned defects allowed in the Selfscan defect list. The compiler default value is set to 400 defects. If the number of defects assigned is exceeded, the current SCW command is terminated with a command check.

max_hard_errs

The max_hard_errs (bytes 24 - 25) specifies the maximum number of hard errors (repeatable read and write errors) allowed in all chains before ending an SCW. The compiler default is 0xFFFF, which is no limit. The hard error counter is incremented each time a read or write error occurs more than once on the same sector or wedge sector. If the number of hard errors is exceeded, the current SCW command is terminated with a command check.

max_hard_head_errs

The max_hard_head_errs (bytes 26 - 27) specifies the maximum number of hard errors (repeatable read and write errors) allowed per head in all chains before ending an SCW. The compiler default is

0xFFFF, which is no limit. The hard per head error counter is incremented each time a read or write error occurs more then once on the same sector or wedge sector. If the number of hard errors per head is exceeded, the current SCW command is terminated with a command check.

max_soft_seek_errs The max_soft_seek_errs (bytes 28 - 29) specifies the maximum number of allowed recovered seek errors in all chains before terminating an SCW. The compiler default is 0xFFFF, which is no limit. The seek error count is incremented each time a seek error occurs. If the number of seek errors is exceeded, the current SCW command is terminated with a command end and exception.

Ending Status

The ending status is presented to the Selfscan command processor by each Selfscan command. The ending status is presented in the status byte (byte 6) of the SSW (Selfscan Status Word). Additional error codes and command statistics are found in the SSW data parameter bytes. The following table gives the byte location, and a short description of the SSW data parameters:

SSW Header Data Parameters

Byte	Symbol	Description
0	error_code	SCW error code
1	error_code_index	Error code index
2 - 3	elapsed_time	SCW elapsed time
4 - 11	version	Version Label
12	flags	SCW Header flag byte
13	trace	Trace Byte, scope trigger
14 - 15	max_time	Maximum SCW chain time (4 second increments)
16 - 17	max_scw_time	Maximum SCW time (4 second increments)
18 - 19	max_assign_def	Maximum number of assigned defects, all chains
20 - 21	max_hard_errors	Maximum hard errors
22 - 23	max_hard_errors_hd	Maximum hard errors per head
24 - 25	max_recovered_seek_errs	Maximum recovered seek errors

This section gives a detailed description of the SSW data parameter bytes by the disassembler mnemonic:

error_code

The SCW error code (byte 0) contains the reason the SCW was terminated. A command check, program check, or exception status must be presented for a valid error code.

error_code_index

The error code index (byte 1) is the sense key index value returned by internal subroutines in the drive's operational firmware. This error code is an engineering error code to help determine the exact cause of the failure. A command check, program check, or exception status must be presented for a valid error code.

elapsed_time

This is the total time (bytes 2 - 3) is in four second increments that the SCW required to execute.

version

The version (bytes 4 - 11) is used to describe the command chain. This eight-byte string is not used by the command. The string is presented in the SSW data parameter bytes to identify the chain.

flags

The flags (byte 12) is the flag byte for the Selfscan command processor. The flag byte instructs the command processor to initialize the results or defect files. The flag byte is zero by defaulted so a command history is maintained by the command processor.

trace

The trace (byte 13) is a compare byte to set and clear an oscilloscope trigger pin. The trace byte is a special engineering debug byte.

max_time

The max_time (bytes 14 - 15) specifies the maximum allowed time the command chain is allowed to execute. The time is specified in four second increments.

max_scw_time

The max_scw_time (bytes 16 - 17) specifies the maximum allowed time for each SCW executed. The time is specified in four second increments.

max_assign_defect

The max_assign_defect (bytes 18 - 19) specifies the maximum number of assigned defects allowed in the Selfscan defect list.

max_hard_errs

The max_hard_errs (bytes 20 - 21) specifies the maximum number of hard errors (repeatable read and write errors) allowed before ending an SCW.

max_hard_head_errs The max_hard_head_errs (bytes 22 - 23) specifies the maximum number of hard errors (repeatable read and write errors) allowed per head before ending an SCW.

max_soft_seek_errs The max_soft_seek_errs (bytes 24 - 25) specifies the maximum number of recovered seek errors allowed before terminating an SCW.

SSW Print Display Example

```
>>> Start of Chain or Power Failure >>>

SCW_ADDR SCW_COMMAND  FLAG STAT SCW_ERR ERR_IDX SCW_TIME(sec)
-------- -----------  ---- ---- ------- ------- -------------
0000h    SCW_HEADER   40h  08h  00h     00h        0
Version: "EXAMPLE1",   Header Flags: 03
```

WRITE ICL, Command 0x02

0	1	2	3	4	5	6	7
0x02	Data Address		Flags	Check Address		0	0x00

Selfscan Command Word

Command Description

The Write ICL command modifies the TIC ICL (second SCW) in the command chain. The TIC command is used as a "branch instruction" that determines which SCW to execute next. Initially, the TIC command is addressed to the next SCW so that the chain runs sequentially. The Write ICL command modifies the data address field of the TIC ICL command to the next SCW address in the command chain. Modifying the data address field prevents the entire command chain from having to "run again" because of a power failure. The Write ICL command should be placed after all SCW's that require a long run time.

When Write ICL executes the initialization flags in the SCW command header are cleared. This means that a complete command history of power-ups are maintained as long as the command chain password is valid. To prevent this from happening, a ERASE PASSWORD command should be at the end of the chain or the next chain written must initialize the results and defect files.

Basic Operation

The Write ICL command performs the following operations:

- Check the validity of the SCW bytes.
- Clears the initialization flag byte in the SCW Header command, which prevents results and defect files from re-initializing.
- Check to see that the second SCW in the chain is a TIC, otherwise present a program check.
- Load new data address to TIC ICL command.
- Write command file to disk.
- Present ending status.

Chaining Restrictions

- A TIC ICL command must be the second SCW in the command chain, otherwise a program check is presented.

SCW Compiler Mnemonics

The Valid mnemonics for the Selfscan Next Generation Compiler are:

WRITE_ICL

SCW Data Parameters

No SCW data parameters are required for this command.

Ending Status

The ending status is presented to the Selfscan command processor by each Selfscan command. The ending status is presented in the status byte (byte 6) of the SSW (Selfscan Status Word). Additional error codes and command statistics are found in the SSW data parameter bytes. The following table gives the byte location, and a short description of the SSW data parameters:

SSW Write ICL Data Parameters

Byte	Symbol	Description
0	error_code	SCW error code
1	error_code_index	Error code index
2 - 3	elapsed_time	SCW elapsed time

This section gives a detailed description of the SSW data parameter bytes by the disassembler mnemonic:

error_code The SCW error code (byte 0) contains the reason the SCW was terminated. A command check, program check, or exception status must be presented for a valid error code.

error_code_index The error code index (byte 1) is the sense key index value returned by internal subroutines in the drive's operational firmware. This error code is an engineering error code to help determine the exact cause of the failure. A command check, program check, or exception status must be presented for a valid error code.

elapsed_time This is the total time (bytes 2 - 3) is in four second increments that the SCW required to execute.

SSW Print Display Example

```
SCW_ADDW SCW_COMMAND  FLAG STAT SCW_ERR ERR_IDX SCW_TIME(sec) LAST_CYL LAST_HD ASGN_DEF SOFT_ERR HARD_ERR
-------- -----------  ---- ---- ------- ------- ------------- -------- ------- -------- -------- --------
0050h    WRITE_ICL    40h  08h  00h     00h           0
```

ALTERNATE SEEK, Command 0x03

<table>
<tr><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th></tr>
<tr><td>0x03</td><td colspan="2">Data Address</td><td>Flags</td><td colspan="2">Check Address</td><td>0</td><td>0x0F</td></tr>
</table>

Selfscan Command Word

Command Description

The Alternate Seek command is a diagnostic test to measure the alternate seek characteristics of the servo. The diagnostic keeps statistics on the number of successful seek operations, the seek time, and the minimum alternate seek times.

Basic Operation

The Alternate Seek command first initializes by seeking to the specified start cylinder and head. The seek times are then measured by seeking to the ending cylinder and head and back to the specified starting cylinder and head. The diagnostic is repeated for the number of times specified in the test's "loop count" value in the SCW input data. A "loop count" of 0000h is an infinite loop, but the Selfscan's default maximum SCW time ends the loop.

The maximum seek time and average seek time statistics are then compared against the limits in the SCW input data. If the values are exceeded, an exception is returned in the status byte.

The identification byte is provided to "assign" an ID number for the different types of alternating seek operations. For example, use 0x01 for a single track seek, 0x03 for a third stroke seek, and so on.

The Alternate Seek command performs the following operations:

- Check the validity of the SCW bytes.
- Seeks to the initial cylinder and head.
- Performs a timed seek to the ending cylinder and head.
- Performs a timed seek to the starting cylinder and head.
- Check the test loop count.
- Present ending status.

Chaining Restrictions

- None

SCW Compiler Mnemonics

The Alternate Seek is used by the Selfscan Next Generation compiler as several commands. The only difference is the default SCW data parameters. Each mnemonic is the Alternate Seek command, but with different default data parameters. The Valid mnemonics for the Selfscan Next Generation Compiler are:

ALTERNATE_SEEK	ALT_SEEK	ALT_SK	AS

SCW Data Parameters

The SCW data parameter bytes provide the additional specifications to execute a Selfscan command. The following table gives the byte location, compiler mnemonic, the default value, and a short description of the SCW data parameters:

SCW Alternate Seek Data Parameters

Byte	Mnemonic	Default	Description
0 - 1	loop_cnt	1	Loop count, 0 = infinite loop
2 - 3	max_ave_seek_time	0	Maximum average seek time, 0 = infinite
4 - 5	max_seek_time_limit	0	Maximum seek time, 0 = infinite
6 - 7	delay	1	Seek delay time, 1 = no delay
8	id_byte	0	Identification byte, 0 = Alternate Seek
9 - 10	start_cyl	0	Starting Cylinder number
11	start_head	0	Starting Head number
12 - 13	end_cyl	1	Ending Cylinder number
14	end_head	last_head	Ending Head number

This next section is a detailed description of the SCW data parameters by the compiler's mnemonic:

loop_cnt

The loop_cnt (bytes 0 - 1) specifies the number of alternate seek operations to execute. This means there are two seeks for each loop count: the seek from the starting cylinder and head to the ending cylinder and head, then back again. The compiler is defaulted to one alternate seek. A value of zero performs alternating seeks until the maximum SCW time is exceeded.

max_ave_seek_time

The max_ave_seek_time (bytes 2 - 3) specifies the maximum allowed average seek time for the test. The time is specified in 2 us increments and is defaulted to zero. A value of zero specifies no limit. If the time is exceeded, the SCW command is terminated with a command end and exception.

max_seek_time_limit

The max_seek_time_limit (bytes 4 - 5) specifies the maximum allowed seek time for the test. The time is specified in 2 us increments and is defaulted to zero. A value of zero specifies no limit. If the time is exceeded, the SCW command is terminated with a command end and exception.

delay

The delay (bytes 6 - 7) specifies the delay between seek operations. The delay specifies the number of instructions in the command before starting the next seek. This is used as a scope trigger to distinguish between the start seek, read seek complete, and write seek complete. A loop count of one specifies no loops and is the compiler default.

id_byte

The id_byte (byte 8) identifies the type of alternate seek. The id_byte is provided in the ending status data parameters for identification by the disassembler. An Alternate Seek is defined as an ID byte of 0x00.

start_cyl

The start_cyl (bytes 9 - 10) specifies the starting cylinder number for the timed alternate seek. The starting cylinder number is defaulted to 0000h by the compiler.

start_head

The start_head (byte 11) specifies the starting head number for the timed alternate seek. The starting head number is defaulted to zero by the compiler.

end_cyl

The end_cyl (bytes 12 - 13) specifies the ending cylinder number for the timed alternate seek. The ending cylinder number is defaulted to 0001h by the compiler.

end_head The end_head (byte 14) specifies the ending head number for the timed alternate seek. The ending head number is defaulted to maximum head number supported.

Ending Status

The ending status is presented to the Selfscan command processor by each Selfscan command. The ending status is presented in the status byte (byte 6) of the SSW (Selfscan Status Word). Additional error codes and command statistics are found in the SSW data parameter bytes. The following table gives the byte location and a short description of the SSW data parameters:

SSW Alternate Seek Data Parameters

Byte	Symbol	Description
0	error_code	SCW error code
1	error_code_index	Error code index
2 - 3	elapsed_time	SCW elapsed time
4	id_byte	Identification byte
5 - 6	start_cyl	Starting cylinder number
7	start_hd	Starting head number
8 - 9	end_cyl	Ending cylinder number
10	end_hd	Ending head number
11 - 12	max_seek_time	Maximum seek time 2 us resolution
13 - 14	min_seek_time	Minimum seek time 2 us resolution
15 - 16	ave_seek_time	Average seek time 2 us resolution
17 - 18	total_seeks	Total number of seeks executed
19 - 20	total_seek_errors	Total number of seek errors

This section gives a detailed description of the SSW data parameter bytes by the disassembler mnemonic:

error_code The SCW error_code (byte 0) contains the reason the SCW was terminated. A command check, program check, or exception status must be presented for a valid error code.

error_code_index The error code index (byte 1) is the sense key index value returned by internal subroutines in the drives operational firmware. This error code is an engineering error code to help determine the exact cause of the

failure. A command check, program check, or exception status must be presented for a valid error code.

elapsed_time — This is the total time (bytes 2 - 3) is in four second increments that the SCW required to execute.

id_byte — The id_byte (byte 4) identifies the type of alternate seek. The id_byte is provided in the ending status data parameters for identification by the disassembler. A Alternate Seek is defined as an ID byte of 0x00.

start_cyl — The start_cyl (bytes 5 - 6) specifies the starting cylinder number for the timed alternate seek.

start_head — The start_head (byte 7) specifies the starting head number for the timed alternate seek.

end_cyl — The end_cyl (bytes 8 - 9) specifies the ending cylinder number for the timed alternate seek.

end_head — The end_head (byte 10) specifies the ending head number for the timed alternate seek.

max_seek_time — The max_seek_time (bytes 11 - 12) is the measured maximum seek time of all the seeks performed. The time is specified in 2 us increments.

min_seek_time — The min_seek_time (bytes 13 - 14) is the measured minimum seek time of all the seeks performed. The time is specified in 2 us increments.

ave_seek_time — The ave_seek_time (bytes 15 - 16) is the measured average seek time of all the seeks performed. The time is specified in 2 us increments.

total_seeks — The total_seeks (bytes 17 - 18) is the total number of seek operations performed by the test. A total of two seeks are done for each alternate seek.

total_seek_errors — The total_seek_errors (bytes 19 - 20) is the total number of seek errors that occurred during the test.

SSW Print Display Example

```
SCW_ADDR SCW_COMMAND  FLAG STAT SCW_ERR ERR_IDX SCW_TIME MAX_SEEK MIN_SEEK AVR_SEEK TOTAL_SEEK SEEK_ERR START_CYL/HD END_CYL/HD
                                                 (sec)    (us)     (us)     (us)
-------- -----------  ---- ---- ------- ------- -------- -------- -------- -------- ---------- -------- -------- --- ------ ---
0018h    ALT_SK        40h  08h  00h     00h          0     3836     2862     3170       100          0     0000   0   0001  7
```

SINGLE TRACK SEEK, Command 0x03

0	1	2	3	4	5	6	7
0x03	Data Address		Flags	Check Address		0	0x0F

Selfscan Command Word

Command Description

The Single Track Seek command is a diagnostic test to measure the single track seek characteristics of the servo. This diagnostic test is the Alternate Seek test with a new set of default SCW data parameters and keeps statistics on the number of successful seek operations, the seek time, and the minimum seek times.

Basic Operation

The Single Track Seek command first initializes by seeking to the specified start cylinder and head. The seek times are then measured by seeking to the ending cylinder and head and back to the specified starting cylinder and head. The diagnostic is repeated for the number of times specified in the test's "loop count" value in the SCW input data. A "loop count" of 0000h is an infinite loop, but the Selfscan's default maximum SCW time ends the loop.

The maximum seek time and average seek time statistics are then compared against the limits in the SCW input data. If the values are exceeded, an exception is returned in the status byte.

The identification byte is provided to "assign" an ID number for the different types of alternating seek operations. For example, use 0x01 for a single track seek, 0x03 for a third stroke seek and so on.

The Single Track Seek command performs the following operations:

- Check the validity of the SCW bytes.
- Seeks to the initial cylinder and head.
- Performs a timed seek to the ending cylinder and head.
- Performs a timed seek to the starting cylinder and head.
- Check test loop count.

- Present ending status.

Chaining Restrictions

- None

SCW Compiler Mnemonics

The Single Track Seek is an Alternate Seek test with different default SCW data parameters. The Valid mnemonics for the Selfscan Next Generation Compiler are:

SINGLE_TRACK_SEEK SST

SCW Data Parameters

The SCW data parameter bytes provide the additional specifications to execute a Selfscan command. The following table gives the byte location, compiler mnemonic, the default value, and a short description of the SCW data parameters:

SCW Single Track Seek Data Parameters

Byte	Mnemonic	Default	Description
0 - 1	loop_cnt	1	Loop count, 0 = infinite loop
2 - 3	max_ave_seek_time	0	Maximum average seek time, 0 = infinite
4 - 5	max_seek_time_limit	0	Maximum seek time, 0 = infinite
6 - 7	delay	1	Seek delay time, 1 = no delay
8	id_byte	0x01	Identification byte, 0x01 = Single Track Seek
9 - 10	start_cyl	0	Starting Cylinder number
11	start_head	0	Starting Head number
12 - 13	end_cyl	1	Ending Cylinder number
14	end_head	last_head	Ending Head number

This next section is a detailed description of the SCW data parameters by the compiler's mnemonic:

loop_cnt

The loop_cnt (bytes 0 - 1) specifies the number of single track seek operations to execute. This means there are two seeks for each loop count: the seek from the starting cylinder and head to the ending cylinder and head, then back again. The compiler is defaulted to one single track seek. A value of zero performs single track seeks until the maximum SCW time is exceeded.

max_ave_seek_time

The max_ave_seek_time (bytes 2 - 3) specifies the maximum allowed average seek time for the test. The time is specified in 2 us increments and is defaulted to zero. A value of zero specifies no limit. If the time is exceeded, the SCW command is terminated with a command end and exception.

max_seek_time_limit

The max_seek_time_limit (bytes 4 - 5) specifies the maximum allowed seek time for the test. The time is specified in 2 us increments and is defaulted to zero. A value of zero specifies no limit. If the time is exceeded, the SCW command is terminated with a command end and exception.

delay

The delay (bytes 6 - 7) specifies the delay between seek operations. The delay specifies the number of instructions in the command before starting the next seek. This is used as a scope trigger to distinguish between the start seek, read seek complete, and write seek complete. A loop count of one specifies no loops and is the compiler default.

id_byte

The id_byte (byte 8) identifies the type of alternate seek. The id_byte is provided in the ending status data parameters for identification by the disassembler. A Single Track Seek is defined as an ID byte of 0x01.

start_cyl

The start_cyl (bytes 9 - 10) specifies the starting cylinder number for the timed single track seek. The starting cylinder number is defaulted to 0000h by the compiler.

start_head

The start_head (byte 11) specifies the starting head number for the timed single track seek. The starting head number is defaulted to 0 by the compiler.

end_cyl

The end_cyl (bytes 12 - 13) specifies the ending cylinder number for the timed single track seek. The ending cylinder number is defaulted to 0001h by the compiler.

end_head The end_head (byte 14) specifies the ending head number for the timed single track seek. The ending head number is defaulted to maximum head supported.

Ending Status

The ending status is presented to the Selfscan command processor by each Selfscan command. The ending status is presented in the status byte (byte 6) of the SSW (Selfscan Status Word). Additional error codes and command statistics are found in the SSW data parameter bytes. The following table gives the byte location, and a short description of the SSW data parameters:

SSW Single Track Seek Data Parameters

Byte	Symbol	Description
0	error_code	SCW error code
1	error_code_index	Error code index
2 - 3	elapsed_time	SCW elapsed time
4	id_byte	Identification byte
5 - 6	start_cyl	Starting cylinder number
7	start_hd	Starting head number
8 - 9	end_cyl	Ending cylinder number
10	end_hd	Ending head number
11 - 12	max_seek_time	Maximum seek time 2 us resolution
13 - 14	min_seek_time	Minimum seek time 2 us resolution
15 - 16	ave_seek_time	Average seek time 2 us resolution
17 - 18	total_seeks	Total number of seeks executed
19 - 20	total_seek_errors	Total number of seek errors

This section gives a detailed description of the SSW data parameter bytes by the disassembler mnemonic:

error_code The SCW error code (byte 0) contains the reason the SCW was terminated. A command check, program check, or exception status must be presented for a valid error code.

error_code_index The error code index (byte 1) is the sense key index value returned by internal subroutines in the drive's operational firmware. This error code is an engineering error code to help determine the exact cause of the

failure. A command check, program check, or exception status must be presented for a valid error code.

elapsed_time

This is the total time (bytes 2 - 3) is in four second increments that the SCW required to execute.

id_byte

The id_byte (byte 4) identifies the type of alternate seek. The id_byte is provided in the ending status data parameters for identification by the disassembler. A Single Track Seek is defined as an ID byte of 0x01.

start_cyl

The start_cyl (bytes 5 - 6) specifies the starting cylinder number for the timed single track seek.

start_head

The start_head (byte 7) specifies the starting head number for the timed single track seek.

end_cyl

The end_cyl (bytes 8 - 9) specifies the ending cylinder number for the timed single track seek.

end_head

The end_head (byte 10) specifies the ending head number for the timed single track seek.

max_seek_time

The max_seek_time (bytes 11 - 12) is the measured maximum seek time of all the seeks performed. The time is specified in 2 us increments.

min_seek_time

The min_seek_time (bytes 13 - 14) is the measured minimum seek time of all the seeks performed. The time is specified in 2 us increments.

ave_seek_time

The ave_seek_time (bytes 15 - 16) is the measured average seek time of all the seeks performed. The time is specified in 2 us increments.

total_seeks

The total_seeks (bytes 17 - 18) is the total number of seek operations performed by the test. A total of two seeks are done for each single track seek.

total_seek_errors

The total_seek_errors (bytes 19 - 20) is the total number of seek errors that occurred during the test.

SSW Print Display Example

```
SCW_ADDR SCW_COMMAND  FLAG STAT SCW_ERR ERR_IDX SCW_TIME MAX_SEEK MIN_SEEK AVR_SEEK TOTAL_SEEK SEEK_ERR START_CYL/HD END_CYL/HD
                                                 (sec)    (us)     (us)     (us)
-------- -----------  ---- ---- ------- ------- -------- -------- -------- -------- ---------- -------- -------- --- ------ ---
0018h    SINGLE_SK     40h  08h  00h     00h           0    3836     2862     3170      100        0       0000     0   0001   7
```

THIRD STROKE SEEK, Command 0x03

0	1	2	3	4	5	6	7
0x03	Data Address		Flags	Check Address		0	0x0F

Selfscan Command Word

Command Description

The Third Stroke Seek command is a diagnostic test to measure the one third stroke seek characteristics of the servo. This diagnostic test is the Alternate Seek test with a new set of default SCW data parameters. The diagnostic keeps statistics on the number of successful seek operations, the seek time, and the minimum seek times.

The number following the test (0, 1, 2) determines which third of the drive is tested. The outer third is specified with the number 0, the middle third is specified with the number 1, and the inner third is specified with the number 2.

Basic Operation

The Third Stroke Seek command first initializes by seeking to the specified start cylinder and head. The seek times are then measured by seeking to the ending cylinder and head and back to the specified starting cylinder and head. The diagnostic is repeated for the number of times specified in the test's "loop count" value in the SCW input data. A "loop count" of 0000h is an infinite loop, but the Selfscan's default maximum SCW time ends the loop.

The maximum seek time and average seek time statistics are then compared against the limits in the SCW input data. If the values are exceeded, an exception is returned in the status byte.

The identification byte is provided to "assign" an ID number for the different types of alternating seek operations. For example, use 0x01 for a single track seek, 0x03 for a third stroke seek and so on.

The Third Stroke Seek command performs the following operations:

- Check the validity of the SCW bytes.
- Seeks to the initial cylinder and head.
- Performs a timed seek to the ending cylinder and head.
- Performs a timed seek to the starting cylinder and head.

- Checks the test loop count.
- Present ending status.

Chaining Restrictions

- None

SCW Compiler Mnemonics

The Third Stroke Seek is an Alternate Seek test with different default SCW data parameters. The Valid mnemonics for the Selfscan Next Generation Compiler are:

THIRD_STROKE_SEEK_0	THIRD_SEEK_0	TSS0
THIRD_STROKE_SEEK_1	THIRD_SEEK_1	TSS1
THIRD_STROKE_SEEK_2	THIRD_SEEK_2	TSS2

SCW Data Parameters

The SCW data parameter bytes provide the additional specifications to execute a Selfscan command. The following table gives the byte location, compiler mnemonic, the default value, and a short description of the SCW data parameters:

SCW Third Stroke Seek Data Parameters

Byte	Mnemonic	Default	Description
0 - 1	loop_cnt	1	Loop count, 0 = infinite loop
2 - 3	max_ave_seek_time	0	Maximum average seek time, 0 = infinite
4 - 5	max_seek_time_limit	0	Maximum seek time, 0 = infinite
6 - 7	delay	1	Seek delay time, 1 = no delay
8	id_byte	0x01	Identification byte, 0x01 = Single Track Seek
9 - 10	start_cyl	start of third	Starting Cylinder number
11	start_head	0	Starting Head number
12 - 13	end_cyl	end of third	Ending Cylinder number
14	end_head	last_head	Ending Head number

This next section is a detailed description of the SCW data parameters by the compiler's mnemonic:

loop_cnt
The loop_cnt (bytes 0 - 1) specifies the number of third stroke seek operations to execute. This means there are two seeks for each loop count, the seek from the starting cylinder and head to the ending cylinder and head, then back again. The compiler is defaulted to one third stroke seek. A value of zero performs third stroke seeks until the maximum SCW time is exceeded.

max_ave_seek_time
The max_ave_seek_time (bytes 2 - 3) specifies the maximum allowed average seek time for the test. The time is specified in 2 us increments and is defaulted to zero. A value of zero specifies no limit. If the time is exceeded, the SCW command is terminated with a command end and exception.

max_seek_time_limit
The max_seek_time_limit (bytes 4 - 5) specifies the maximum allowed seek time for the test. The time is specified in 2 us increments and is defaulted to zero. A value of zero specifies no limit. If the time is exceeded, the SCW command is terminated with a command end and exception.

delay
The delay (bytes 6 - 7) specifies the delay between seek operations. The delay specifies the number of instructions in the command before starting the next seek. This is used as a scope trigger to distinguish between the start seek, read seek complete, and write seek complete. A loop count of one specifies no loops and is the compiler default.

id_byte
The id_byte (byte 8) identifies the type of alternate seek. This byte is provide in the ending status data parameters for identification by the disassembler. A Third Stroke Seek is defined as an ID byte of 0x03.

start_cyl
The start_cyl (bytes 9 - 10) specifies the starting cylinder number for the timed third stroke seek. The starting cylinder number is defaulted to the starting third of the drives cylinder range by the compiler.

start_head
The start_head (byte 11) specifies the starting head number for the timed third stroke seek. The starting head number is defaulted to zero by the compiler.

end_cyl
The end_cyl (bytes 12 - 13) specifies the ending cylinder number for the timed third stroke seek. The ending cylinder number is defaulted to the ending third of the drives cylinder range by the compiler.

end_head The end_head (byte 14) specifies the ending head number for the timed third stroke seek. The ending head number is defaulted to maximum head supported.

Ending Status

The ending status is presented to the Selfscan command processor by each Selfscan command. The ending status is presented in the status byte (byte 6) of the SSW (Selfscan Status Word). Additional error codes and command statistics are found in the SSW data parameter bytes. The following table gives the byte location and a short description of the SSW data parameters:

SSW Third Stroke Seek Data Parameters

Byte	Symbol	Description
0	error_code	SCW error code
1	error_code_index	Error code index
2 - 3	elapsed_time	SCW elapsed time
4	id_byte	Identification byte
5 - 6	start_cyl	Starting cylinder number
7	start_hd	Starting head number
8 - 9	end_cyl	Ending cylinder number
10	end_hd	Ending head number
11 - 12	max_seek_time	Maximum seek time 2 us resolution
13 - 14	min_seek_time	Minimum seek time 2 us resolution
15 - 16	ave_seek_time	Average seek time 2 us resolution
17 - 18	total_seeks	Total number of seeks executed
19 - 20	total_seek_errors	Total number of seek errors

This section gives a detailed description of the SSW data parameter bytes by the disassembler mnemonic:

error_code The SCW error code (byte 0) contains the reason the SCW was terminated. A command check, program check, or exception status must be presented for a valid error code.

error_code_index The error code index (byte 1) is the sense key index value returned by internal subroutines in the drives operational firmware. This error code is an engineering error code to help determine the exact cause of the

failure. A command check, program check, or exception status must be presented for a valid error code.

elapsed_time

This is the total time (bytes 2 - 3) is in four second increments that the SCW required to execute.

id_byte

The id_byte (byte 4) identifies the type of alternate seek. This byte is provide in the ending status data parameters for identification by the disassembler. A Third Stroke Seek is defined as an ID byte of 0x03.

start_cyl

The start_cyl (bytes 5 - 6) specifies the starting cylinder number for the timed third stroke seek.

start_head

The start_head (byte 7) specifies the starting head number for the timed third stroke seek.

end_cyl

The end_cyl (bytes 8 - 9) specifies the ending cylinder number for the timed third stroke seek.

end_head

The end_head (byte 10) specifies the ending head number for the timed third stroke seek.

max_seek_time

The max_seek_time (bytes 11 - 12) is the measured maximum seek time of all the seeks performed. The time is specified in 2 us increments.

min_seek_time

The min_seek_time (bytes 13 - 14) is the measured minimum seek time of all the seeks performed. The time is specified in 2 us increments.

ave_seek_time

The ave_seek_time (bytes 15 - 16) is the measured average seek time of all the seeks performed. The time is specified in 2 us increments.

total_seeks

The total_seeks (bytes 17 - 18) is the total number of seek operations performed by the test. A total of two seeks are done for each third stroke seek.

total_seek_errors

The total_seek_errors (bytes 19 - 20) is the total number of seek errors that occurred during the test.

SSW Print Display Example

```
SCW_ADDR SCW_COMMAND  FLAG STAT SCW_ERR ERR_IDX SCW_TIME MAX_SEEK MIN_SEEK AVR_SEEK TOTAL_SEEK SEEK_ERR START_CYL/HD END_CYL/HD
                                                 (sec)    (us)     (us)     (us)
-------- -----------  ---- ---- ------- ------- -------- -------- -------- -------- ---------- -------- -------- --- ------ ---
0028h    THIRD_SK     40h  08h  00h     00h         0    13224     9446    10824        40          0   0000      0  0957   7
0030h    THIRD_SK     40h  08h  00h     00h         0    12250    10988    11306        40          0   0957      0  1914   7
0038h    THIRD_SK     40h  08h  00h     00h         0    11938    10730    11148        40          0   1914      0  2873   7
```

FULL STROKE SEEK, Command 0x03

0	1	2	3	4	5	6	7
0x03	Data Address		Flags	Check Address		0	0x0F

Selfscan Command Word

Command Description

The Full Stroke Seek command is a diagnostic test to measure the full stroke seek characteristics of the servo. This diagnostic test is the Alternate Seek test with a new set of default SCW data parameters. The diagnostic keeps statistics on the number of successful seek operations, the seek time, and the minimum seek times.

Basic Operation

The Full Stroke Seek command first initializes by seeking to the specified start cylinder and head. The seek times are then measured by seeking to the ending cylinder and head and back to the specified starting cylinder and head. The diagnostic is repeated for the number of times specified in the test's "loop count" value in the SCW input data. A "loop count" of 0000h is an infinite loop, but the Selfscan's default maximum SCW time ends the loop.

The maximum seek time and average seek time statistics are then compared against the limits in the SCW input data. If the values are exceeded, an exception is returned in the status byte.

The identification byte is provided to "assign" an ID number for the different types of alternating seek operations. For example, use 0x01 for a single track seek, 0x03 for a third stroke seek, and so on.

The Full Stroke Seek command performs the following operations:

- Checks the validity of the SCW bytes.
- Seeks to the initial cylinder and head.
- Performs a timed seek to the ending cylinder and head.
- Performs a timed seek to the starting cylinder and head.
- Checks the test loop count.

- Present ending status.

Chaining Restrictions

- None

SCW Compiler Mnemonics

The Full Stroke Seek is an Alternate Seek test with different default SCW data parameters. The Valid mnemonics for the Selfscan Next Generation Compiler are:

FULL_STOKE_SEEK FSS

SCW Data Parameters

The SCW data parameter bytes provide the additional specifications to execute a Selfscan command. The following table gives the byte location, compiler mnemonic, the default value, and a short description of the SCW data parameters:

SCW Full Stroke Seek Data Parameters

Byte	Mnemonic	Default	Description
0 - 1	loop_cnt	1	Loop count, 0 = infinite loop
2 - 3	max_ave_seek_time	0	Maximum average seek time, 0 = infinite
4 - 5	max_seek_time_limit	0	Maximum seek time, 0 = infinite
6 - 7	delay	1	Seek delay time, 1 = no delay
8	id_byte	0x01	Identification byte, 0x01 = Single Track Seek
9 - 10	start_cyl	0	Starting Cylinder number
11	start_head	0	Starting Head number
12 - 13	end_cyl	max cylinder	Ending Cylinder number
14	end_head	last head	Ending Head number

This next section is a detailed description of the SCW data parameters by the compiler's mnemonic:

loop_cnt

The loop_cnt (bytes 0 - 1) specifies the number of full stroke seek operations to execute, which means there are two seeks for each loop count, the seek from the starting cylinder and head to the ending cylinder and head, then back again. The compiler is defaulted to one full stroke seek. A value of zero performs full stroke seeks until the maximum SCW time is exceeded.

max_ave_seek_time

The max_ave_seek_time (bytes 2 - 3) specifies the maximum allowed average seek time for the test. The time is specified in 2 us increments and is defaulted to zero. A value of zero specifies no limit. If the time is exceeded, the SCW command is terminated with a command end and exception.

max_seek_time_limit

The max_seek_time_limit (bytes 4 - 5) specifies the maximum allowed seek time for the test. The time is specified in 2 us increments and is defaulted to zero. A value of zero specifies no limit. If the time is exceeded, the SCW command is terminated with a command end and exception.

delay

The delay (bytes 6 - 7) specifies the delay between seek operations. The delay specifies the number of instructions in the command before starting the next seek. This is used as a scope trigger to distinguish between the start seek, read seek complete, and write seek complete. A loop count of one specifies no loops and is the compiler default.

id_byte

The id_byte (byte 8) is used to identify the type of alternate seek. The id_byte is provided in the ending status data parameters for identification by the disassembler. A Full Stroke Seek is defined as an ID byte of 0xFF.

start_cyl

The start_cyl (bytes 9 - 10) specifies the starting cylinder number for the timed full stroke seek. The starting cylinder number is defaulted to 0000h by the compiler.

start_head

The start_head (byte 11) specifies the starting head number for the timed full stroke seek. The starting head number is defaulted to zero by the compiler.

end_cyl

The end_cyl (bytes 12 - 13) specifies the ending cylinder number for the timed full stroke seek. The ending cylinder number is defaulted to the maximum cylinder number.

end_head The end_head (byte 14) specifies the ending head number for the timed full stroke seek. The ending head number is defaulted to maximum head supported.

Ending Status

The ending status is presented to the Selfscan command processor by each Selfscan command. The ending status is presented in the status byte (byte 6) of the SSW (Selfscan Status Word). Additional error codes and command statistics are found in the SSW data parameter bytes. The following table gives the byte location and a short description of the SSW data parameters:

SSW Full Stroke Seek Data Parameters

Byte	Symbol	Description
0	error_code	SCW error code
1	error_code_index	Error code index
2 - 3	elapsed_time	SCW elapsed time
4	id_byte	Identification byte
5 - 6	start_cyl	Starting cylinder number
7	start_hd	Starting head number
8 - 9	end_cyl	Ending cylinder number
10	end_hd	Ending head number
11 - 12	max_seek_time	Maximum seek time 2 us resolution
13 - 14	min_seek_time	Minimum seek time 2 us resolution
15 - 16	ave_seek_time	Average seek time 2 us resolution
17 - 18	total_seeks	Total number of seeks executed
19 - 20	total_seek_errors	Total number of seek errors

This section gives a detailed description of the SSW data parameter bytes by the disassembler mnemonic:

error_code The SCW error code (byte 0) contains the reason the SCW was terminated. A command check, program check, or exception status must be presented for a valid error code.

error_code_index The error code index (byte 1) is the sense key index value returned by internal subroutines in the drives operational firmware. This error code is an engineering error code to help determine the exact cause of the

failure. A command check, program check, or exception status must be presented for a valid error code.

elapsed_time

This is the total time (bytes 2 - 3) is in four second increments that the SCW required to execute.

id_byte

The id_byte (byte 4) is used to identify the type of alternate seek. The id_byte is provided in the ending status data parameters for identification by the disassembler. A Full Stroke Seek is defined as an ID byte of 0xFF.

start_cyl

The start_cyl (bytes 5 - 6) specifies the starting cylinder number for the timed full stroke seek.

start_head

The start_head (byte 7) specifies the starting head number for the timed full stroke seek.

end_cyl

The end_cyl (bytes 8 - 9) specifies the ending cylinder number for the timed full stroke seek.

end_head

The end_head (byte 10) specifies the ending head number for the timed full stroke seek.

max_seek_time

The max_seek_time (bytes 11 - 12) is the measured maximum seek time of all the seeks performed. The time is specified in 2 us increments.

min_seek_time

The min_seek_time (bytes 13 - 14) is the measured minimum seek time of all the seeks performed. The time is specified in 2 us increments.

ave_seek_time

The ave_seek_time (bytes 15 - 16) is the measured average seek time of all the seeks performed. The time is specified in 2 us increments.

total_seeks

The total_seeks (bytes 17 - 18) is the total number of seek operations performed by the test. A total of two seeks are done for each full stroke seek.

total_seek_errors

The total_seek_errors (bytes 19 - 20) is the total number of seek errors that occurred during the test.

SSW Print Display Example

```
SCW_ADDR SCW_COMMAND  FLAG STAT SCW_ERR ERR_IDX SCW_TIME MAX_SEEK MIN_SEEK AVR_SEEK TOTAL_SEEK SEEK_ERR START_CYL/HD END_CYL/HD
                                                (sec)    (us)     (us)     (us)
-------- -----------  ---- ---- ------- ------- -------- -------- -------- -------- ---------- -------- -------- --- ------ ---
0040h    FULL_SEEK    40h  08h  00h     00h         0    22616    18074    19454      100        0      0000      0 2873   7
```

HEAD SWITCH, Command 0x04

0	1	2	3	4	5	6	7
0x04	Data Address		Flags	Check Address		0	0x0A

Selfscan Command Word

Command Description

The Head Switch command is a diagnostic test to measure the head switch seek characteristics of the servo. The diagnostic keeps statistics on the number of successful seek operations, the number of recovered seek operations, the average head switch time, the maximum head switch time, and the minimum head switch time.

Basic Operation

The Head Switch command first initializes by seeking to the specified start cylinder (head 0). The seek times are then measured by seeking to the next head until the maximum head number is reached. The diagnostic is repeated for the number of times specified in the test's "loop count" value in the SCW input data. A "loop count" of 0000h is an infinite loop, but the Selfscan's default maximum SCW time ends the loop.

The maximum head switch time and average head switch time statistics are then compared against the limits in the SCW input data. If the values are exceeded, an exception is returned in the status byte.

The Head Switch command performs the following operations:

- Checks the validity of the SCW bytes.
- Seeks to the initial cylinder, head 0.
- Performs a timed seek to cylinder, head + 1 until the maximum head is reached.
- Checks the test loop count.
- Presents ending status.

Chaining Restrictions

- None

SCW Compiler Mnemonics

The Valid mnemonics for the Selfscan Next Generation Compiler are:

HEAD_SWITCH	HS

SCW Data Parameters

The SCW data parameter bytes provide the additional specifications to execute a Selfscan command. The following table gives the byte location, compiler mnemonic, the default value, and a short description of the SCW data parameters:

SCW Head Switch Data Parameters

Byte	Mnemonic	Default	Description
0 - 1	loop_cnt	1	Loop count, 0 = infinite loop
2 - 3	max_ave_sw_time	0	Maximum average head switch time, 0 = infinite
4 - 5	max_sw_time_limit	0	Maximum head switch time, 0 = infinite
6 - 7	delay	1	Seek delay time, 1 = no delay
8 - 9	cyl	0	Cylinder number

This next section is a detailed description of the SCW data parameters by the compiler's mnemonic:

loop_cnt

The loop_cnt (bytes 0 - 1) specifies the number of head switch operations to execute. The drive seeks to each head once for each loop count. The compiler is defaulted to one head switch test loop. A value of zero will perform head switches until the maximum SCW time is exceeded.

max_ave_sw_time

The max_ave_sw_time (bytes 2 - 3) specifies the maximum allowed average head switch time for the test. The time is specified in 2 us increments and is defaulted to zero. A value of zero specifies no limit.

If the time is exceeded, the SCW command is terminated with a command end and exception.

max_sw_time_limit	The max_sw_time_limit (bytes 4 - 5) specifies the maximum allowed head switch time for the test. The time is specified in 2 us increments and is defaulted to zero. A value of zero specifies no limit. If the time is exceeded, the SCW command is terminated with a command end and exception.

delay	The delay (bytes 6 - 7) specifies the delay between seek operations. The delay specifies the number of instructions in the command before starting the next head switch. This is used as a scope trigger to distinguish between the start seek, read seek complete, and write seek complete. A loop count of one specifies no loops and is the compiler default.

cyl	The cyl (bytes 8 - 9) specifies the cylinder number for the timed head switch seek. The cylinder number is defaulted to 0000h by the compiler.

Ending Status

The ending status is presented to the Selfscan command processor by each Selfscan command. The ending status is presented in the status byte (byte 6) of the SSW (Selfscan Status Word). Additional error codes and command statistics are found in the SSW data parameter bytes. The following table gives the byte location and a short description of the SSW data parameters:

SSW Head Switch Data Parameters

Byte	Symbol	Description
0	error_code	SCW error code
1	error_code_index	Error code index
2 - 3	elapsed_time	SCW elapsed time
4 - 5	cyl	Starting cylinder number
6 - 7	max_hd_time	Maximum head switch time 2 us resolution
8 - 9	min_hd_time	Minimum head switch time 2 us resolution
10 - 11	ave_hd_time	Average head switch time 2 us resolution
12 - 13	total_switches	Total number of head switches executed

14 - 15	total_sw_errors	Total number of head switch errors

This section gives a detailed description of the SSW data parameter bytes by the disassembler mnemonic:

error_code
The SCW error code (byte 0) contains the reason the SCW was terminated. A command check, program check, or exception status must be presented for a valid error code.

error_code_index
The error code index (byte 1) is the sense key index value returned by internal subroutines in the drive's operational firmware. This error code is an engineering error code to help determine the exact cause of the failure. A command check, program check, or exception status must be presented for a valid error code.

elapsed_time
This is the total time (bytes 2 - 3) is in four second increments that the SCW required to execute.

cyl
The cyl (bytes 4 - 5) specifies the cylinder number for the timed head switch.

max_hd_time
The max_hd_time (bytes 6 - 7) is the measured maximum head switch time of all the seeks performed. The time is specified in 2 us increments.

min_hd_time
The min_hd_time (bytes 8 - 9) is the measured minimum head switch time of all the seeks performed. The time is specified in 2 us increments.

ave_hd_time
The ave_hd_time (bytes 10 - 11) is the measured average head switch time of all the seeks performed. The time is specified in 2 us increments.

total_switches
The total_switches (bytes 12 - 13) is the total number of head switch operations performed by the test. A total of maximum head number minus one seeks are done for each head switch.

total_sw_errors
The total_sw_errors (bytes 14 - 15) is the total number of head switch errors that occurred during the test.

SSW Print Display Example

```
SCW_ADDR SCW_COMMAND  FLAG STAT SCW_ERR ERR_IDX SCW_TIME MAX_SEEK MIN_SEEK AVR_SEEK TOTAL_SEEK SEEK_ERR START_CYL/HD END_CYL/HD
                                                (sec)    (us)     (us)     (us)

-------- -----------  ---- ---- ------- ------- -------- -------- -------- -------- ---------- -------- -------- --- ------ ---
0020h    HEAD_SWITCH  40h  08h  00h     00h        0      1960      566      896      350        0      0000
```

RANDOM SEEK, Command 0x05

0	1	2	3	4	5	6	7
0x05	Data Address		Flags	Check Address		0	0x08

Selfscan Command Word

Command Description

The Random Seek command is a diagnostic test to measure the seek characteristics of the servo. The diagnostic keeps statistics on the number of successful seek operations, the number of recovered seeks, and maximum, minimum, and average seek times.

Basic Operation

The seek times are then measured by seeking to the next random cylinder and head. The diagnostic is repeated for the number of times specified in the test's "loop count" value in the SCW input data. A "loop count" of 0000h is an infinite loop, but the Selfscan's default maximum SCW time ends the loop.

The maximum seek time and average seek time statistics are then compared against the limits in the SCW input data. If the values are exceeded, an exception is returned in the status byte.

The Random Seek command performs the following operations:

- Checks the validity of the SCW bytes.
- Performs a timed seek to random cylinder and head.
- Checks the test loop count.
- Present ending status.

Chaining Restrictions

- None

SCW Compiler Mnemonics

The Valid mnemonics for the Selfscan Next Generation Compiler are:

RANDOM_SEEK	RS

SCW Data Parameters

The SCW data parameter bytes provide the additional specifications to execute a Selfscan command. The following table gives the byte location, compiler mnemonic, the default value, and a short description of the SCW data parameters:

SCW Random Seek Data Parameters

Byte	Mnemonic	Default	Description
0 - 1	loop_cnt	1	Loop count, 0 = infinite loop
2 - 3	max_ave_seek_time	0	Maximum average seek time, 0 = infinite
4 - 5	max_seek_time_limit	0	Maximum seek time, 0 = infinite
6 - 7	delay	1	Seek delay time, 1 = no delay

This next section is a detailed description of the SCW data parameters by the compiler's mnemonic:

loop_cnt — The loop_cnt (bytes 0 - 1) specifies the number of random seek operations to execute. The compiler is defaulted to one random seek. A value of zero performs random seeks until the maximum SCW time is exceeded.

max_ave_seek_time — The max_ave_seek_time (bytes 2 - 3) specifies the maximum allowed average seek time for the test. The time is specified in 2 us increments and is defaulted to zero. A value of zero specifies no limit. If the time is exceeded, the SCW command is terminated with a command end and exception.

max_seek_time_limit — The max_seek_time_limit (bytes 4 - 5) specifies the maximum allowed seek time for the test. The time is specified in 2 us increments and is defaulted to zero. A value of zero specifies no limit. If the time is exceeded, the SCW command is terminated with a command end and exception.

delay — The delay (bytes 6 - 7) specifies the delay between seek operations. The delay specifies the number of instructions in the command before starting the next seek. This is used as a scope trigger to distinguish between the start seek, read seek complete, and write seek complete. A loop count of one specifies no loops and is the compiler default.

Ending Status

The ending status is presented to the Selfscan command processor by each Selfscan command. The ending status is presented in the status byte (byte 6) of the SSW (Selfscan Status Word). Additional error codes and command statistics are found in the SSW data parameter bytes. The following table gives the byte location and a short description of the SSW data parameters:

SSW Random Seek Data Parameters

Byte	Symbol	Description
0	error_code	SCW error code
1	error_code_index	Error code index
2 - 3	elapsed_time	SCW elapsed time
4 - 5	max_seek_time	Maximum seek time 2 us resolution
6 - 7	min_seek_time	Minimum seek time 2 us resolution
8 - 9	ave_seek_time	Average seek time 2 us resolution
10 - 11	total_seeks	Total number of seeks executed
12 - 13	total_seek_errors	Total number of seek errors

This section gives a detailed description of the SSW data parameter bytes by the disassembler mnemonic:

error_code — The SCW error code (byte 0) contains the reason the SCW was terminated. A command check, program check, or exception status must be presented for a valid error code.

error_code_index — The error code index (byte 1) is the sense key index value returned by internal subroutines in the drives operational firmware. This error code is an engineering error code to help determine the exact cause of the failure. A command check, program check, or exception status must be presented for a valid error code.

elapsed_time — This is the total time (bytes 2 - 3) is in four second increments that the SCW required to execute.

max_seek_time The max_seek_time (bytes 4 - 5) is the measured maximum seek time of all the seeks performed. The time is specified in 2 us increments.

min_seek_time The min_seek_time (bytes 6 - 7) is the measured minimum seek time of all the seeks performed. The time is specified in 2 us increments.

ave_seek_time The ave_seek_time (bytes 8 - 9) is the measured average seek time of all the seeks performed. The time is specified in 2 us increments.

total_seeks The total_seeks (bytes 10 - 11) is the total number of seek operations performed by the test.

total_seek_errors The total_seek_errors (bytes 12 - 13) is the total number of seek errors that occurred during the test.

SSW Print Display Example

```
SCW_ADDR SCW_COMMAND FLAG STAT SCW_ERR ERR_IDX SCW_TIME MAX_SEEK MIN_SEEK AVR_SEEK TOTAL_SEEK SEEK_ERR START_CYL/HD END_CYL/HD
                                                (sec)    (us)     (us)     (us)
-------- ----------- ---- ---- ------- ------- -------- -------- -------- -------- ---------- -------- -------- --- ------ ---
0048h    RANDOM_SEEK 40h  08h  00h     00h         0    17192      5224   11002        50         0
```

FORMAT MEDIA, Command 0x06

0	1	2	3	4	5	6	7
0x06	Data Address		Flags	Check Address		0	0x00

Selfscan Command Word

Command Description

The Format Media command is used to format the customer data cylinders with an ID after wedge sector format. This command simply calls the super-set SCSI command format track (FF18) with the options to format the entire drive.

Basic Operation

First, the Format Media command clears the working (W list), grown (G list), and the primary (P list) defects lists. The data cylinders are then formatted using the super-set command format track. All the data sectors are written with 0xF6 for data. After the format operation is complete, the idle call vectors are run to initialize the drive mode and configuration pages.

This command does not check the maximum limits of time or seek errors because the super-set command format track or idle call vectors can't be terminated .

The Format Media command performs the following operations:

- Checks the validity of the SCW bytes.
- Clears the W list.
- Clears the P list.
- Clears the G list.
- Formats the drive from cylinder 0 to the maximum cylinder.
- Restarts the idle call vectors to initialize the drive.
- Present ending status.

Chaining Restrictions

- None.

SCW Compiler Mnemonics

The Valid mnemonics for the Selfscan Next Generation Compiler are:

FORMAT_MEDIA	FORMAT	FMT

SCW Data Parameters

No SCW data parameters are required for this command.

Ending Status

The ending status is presented to the Selfscan command processor by each Selfscan command. The ending status is presented in the status byte (byte 6) of the SSW (Selfscan Status Word). Additional error codes and command statistics are found in the SSW data parameter bytes. The following table gives the byte location, and a short description of the SSW data parameters:

SSW Format Media Data Parameters

Byte	Symbol	Description
0	error_code	SCW error code
1	error_code_index	Error code index
2 - 3	elapsed_time	SCW elapsed time
4 - 5	cylinder	Current cylinder number
6	head	Current head number

This section gives a detailed description of the SSW data parameter bytes by the disassembler mnemonic:

error_code — The SCW error code (byte 0) contains the reason the SCW was terminated. A command check, program check, or exception status must be presented for a valid error code.

error_code_index The error code index (byte 1) is the sense key index value returned by internal subroutines in the drives operational firmware. This error code is an engineering error code to help determine the exact cause of the failure. A command check, program check, or exception status must be presented for a valid error code.

elapsed_time This is the total time (bytes 2 - 3) is in four second increments that the SCW required to execute.

cylinder The cylinder (bytes 4 - 5) is the current cylinder number when the command terminates.

head The head (byte 6) is the current head number when the command terminates.

SSW Print Display Example

```
SCW_ADDR SCW_COMMAND  FLAG STAT SCW_ERR ERR_IDX SCW_TIME(sec)
-------- -----------  ---- ---- ------- ------- -------------
0098h    FORMAT_MEDIA 40h  08h  00h     00h      672
```

ERASE PASSWORD, Command 0x07

<table>
<tr><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th></tr>
<tr><td>0x07</td><td colspan="2">Data Address</td><td>Flags</td><td colspan="2">Check Address</td><td>0</td><td>0x00</td></tr>
</table>

Selfscan Command Word

Command Description

The Erase Password command changes the password in the SCW Header command to "ERSEPWRD" in the command chain. Changing the password prevents the Selfscan command processor from running the SCW chain again. The Erase Password command should only be executed after the chain completes, or as a check address for errors from other SCWs that have failed.

Basic Operation

The Erase Password command performs the following operations:

- Checks the validity of the SCW bytes.
- Load the new SCW Header password ("ERSEPWRD").
- Writes the command file to disk.
- Present ending status.

Chaining Restrictions

- None.

SCW Compiler Mnemonics

The Valid mnemonics for the Selfscan Next Generation Compiler are:

ERASE_PASSWORD

SCW Data Parameters

No SCW data parameters are required for this command.

Ending Status

The ending status is presented to the Selfscan command processor by each Selfscan command. The ending status is presented in the status byte (byte 6) of the SSW (Selfscan Status Word). Additional error codes and command statistics are found in the SSW data parameter bytes. The following table gives the byte location and a short description of the SSW data parameters:

SSW Write ICL Data Parameters

Byte	Symbol	Description
0	error_code	SCW error code
1	error_code_index	Error code index
2 - 3	elapsed_time	SCW elapsed time

This section gives a detailed description of the SSW data parameter bytes by the disassembler mnemonic:

error_code — The SCW error code (byte 0) contains the reason the SCW was terminated. A command check, program check, or exception status must be presented for a valid error code.

error_code_index — The error code index (byte 1) is the sense key index value returned by internal subroutines in the drives operational firmware. This error code is an engineering error code to help determine the exact cause of the failure. A command check, program check, or exception status must be presented for a valid error code.

elapsed_time — This is the total time (bytes 2 - 3) is in four second increments that the SCW required to execute.

SSW Print Display Example

```
SCW_ADDR SCW_COMMAND  FLAG STAT SCW_ERR ERR_IDX SCW_TIME(sec)
-------- -----------  ---- ---- ------- ------- -------------
00c8h    ERS_PASSWRD  00h  08h  00h     00h           0
```

TIC, Command 0x08

0	1	2	3	4	5	6	7
0x08	Data Address		Flags	0x0000		0	0x00

Selfscan Command Word

Command Description

The TIC command (Transfer In Command) is similar to a branch instruction in a processor. The TIC uses the address field to indicate the location of the next SCW to execute (provided command chain continue) if the "Status Modifier" bit is reset in the previous SSW status byte. If the "Status Modifier" bit is set the next SCW is executed. This is useful for an SCW that is waiting for an event to occur. This could be an SCW that waits for intervention from the host interface before continuing with the chain. The second SCW in a command chain is a TIC, which contains the ICL (Initial Command Load) address. This allows the command chain to be modified during its execution with the Write ICL command. This way a power failure does not start the chain at the beginning again, instead the command chain continues where it was executing.

Basic Operation

The TIC command changes the execution address of the next SCW if the "Status Modifier" bit in the previous SSW status byte is reset. If the transfer address is located back eight bytes (TIC -8), the command raises an internal program flag (disable_result) in the command processor chain block to disable the writing of the next SCW in the result file. Disabling the writing of the next SCW keeps the result file from "filling up" with the trace.

The TIC (Transfer In Command) command performs the following operations:

- Checks the validity of the SCW bytes.
- Checks the Status Modifier. If it is set, the TIC loads the next SCW. If the Status Modifier is not set, the TIC loads the command processor's program counter with the data address field.
- Checks if the branch address is located back one SCW. If it is, the TIC sets disable result flag. If not, it resets the disable result flag.
- Checks the maximum time limits to terminate the SCW.

- Present ending status.

Chaining Restrictions

- The second SCW in a command chain should be TIC ICL, which allows the Write ICL command to execute. This feature allows the command chain to continue execution where it "left off" after a power failure.

- A TIC one SCW back halts the results file trace.

- The Check Address must be zero (no check address), otherwise a program check is presented.

SCW Compiler Mnemonics

The Valid mnemonics for the Selfscan Next Generation Compiler are:

TIC JMP GOTO

SCW Data Parameters

No SCW data parameters are required for this command.

Ending Status

The ending status is presented to the Selfscan command processor by each Selfscan command. The ending status is presented in the status byte (byte 6) of the SSW (Selfscan Status Word). Additional error codes and command statistics are found in the SSW data parameter bytes.

No SSW data parameters are presented by this command.

SSW Print Display Example

```
SCW_ADDR SCW_COMMAND  FLAG STAT
-------- -----------  ---- ----
0008h    TIC          40h  08h
```

SEQUENTIAL DEFECT, Command 0x09

0	1	2	3	4	5	6	7
0x09	Data Address		Flags	Check Address		0	0x30

Selfscan Command Word

Command Description

The sequential defect command is a diagnostic test to search for defective sectors on a cylinder sequentially from the specified starting cylinder to the specified ending cylinder. If a defect is located, it is added to the Selfscan defect list file. The reading of the sectors may be stressed by offtrack reading, threshold, and boost in the R/W channel. Sequential defect scan keeps statistics on the number of times a defect has been located, the number of soft and hard errors, the number of wiggle recovered errors, the number of recovered seek operations, and the total time the diagnostic test ran.

All non-medium errors in the sequential defect command are retried the number of times specified in the retry count of MODE PAGE 1 (the default is eight retries). Setting the retry count in MODE PAGE 1 to zero halts all retries of non_medium errors (i.e. write faults). Setting the retry count in MODE PAGE 1 to 255 retries all non_medium errors 255 times.

Basic Operation

The criteria for locating a defect is defined in the SCW data block. Each track is read the number of times specified (search read loop count) for each search write loop count. The read channel values (offtrack, boost, Viterbi threshold, and DPD threshold) should be "stressed" (i.e. close to the edge of failure) to locate marginal medium defects. This "search" algorithm is used to locate a possible defect on the track.

Should a medium error (write or read) be detected by the "search" algorithm, the next user-specified criteria determines if the medium error is a "soft" or "hard (assigned defect)" error. The suspect sector is first read the number of times specified in the retry read verify loop count. If an error occurs, "wiggle" recovery is initiated to check for a wiggle head problem. The "wiggle" recovery count is then compared with the defect threshold count specified in the SCW. If the "wiggle" recovery count is greater than or equal to the defect threshold count, the sector is assigned as a "hard" error and the defect added to the Selfscan defect list. The read channel values can be "stressed" with the retry offtrack and retry margin values to locate the defect.

If no medium errors are detected in the retry read verify loop, the next step in the "defect" algorithm is the retry write/read verify loop. The suspect sector is written and then read back using the retry margin values for the number of times specified in the retry write/read verify loop count. If an error occurs, "wiggle" recovery is initiated to check for a wiggle head problem. The "wiggle" recovery count is then compared with the defect threshold count specified in the SCW. If the "wiggle" recovery count is greater than or equal to the defect threshold count, the sector is assigned as a "hard" error and the defect added to the Selfscan defect list. If no medium errors were detected in the suspect sector after the write/verify loop count, the sector is assumed to be a "soft" error and the soft error statistics are updated. The defect "search" continues until the ending cylinder has been reached.

Sequential defect scan SCW data also has several maximum limits to terminate the Selfscan test early should a value be exceeded. The maximum SCW time, the maximum total number of "soft" and "hard" errors, the maximum number of "soft" and "hard" errors per head, the maximum number of recovered seek errors, and the maximum number of assigned defects are all specified in the SCW data block. If a limit is exceeded, the SCW is terminated.

The Sequential defect scan command performs the following operations:

- Checks the validity of the SCW bytes.

- Checks the validity of the SCW data bytes.

- Checks for valid cylinders.

- Fill the write data buffer with specified data bytes.

- Initializes counters, timers, starting cylinder, head, and sectors numbers.

- Initializes debug trace

**** Defect Scan Loop

- Checks to see if maximum limits have been exceeded.

- Checks to see if the maximum number of hard errors has been exceeded.

- Check to see if the maximum number of soft errors has been exceeded.

- Writes and reads back each track the number of times specified.

- If a medium error occurs the sector is re-read the number of times specified, then written and read the number of times specified to determine if a defect exits.

- If a defect is located, the defect is added to the Selfscan Defect list file and track statistics updated.

- Increments the head, cylinder and continue with the Defect Scan Loop.

**** End Defect Scan Loop

- Checks the maximum recovered seek error limit.

- Present ending status.

Chaining Restrictions

- The drive must be formatted with sectors, either executing the Format Media command in selfscan or calling the SCSI Format Media command from the host interface.

SCW Compiler Mnemonics

The Valid mnemonics for the Selfscan Next Generation Compiler are:

SEQUENTIAL_DEFECT	SEQUENTIAL_SCAN	SDS	DS

SCW Data Parameters

The SCW data parameter bytes provide the additional specifications to execute a Selfscan command. The following table gives the byte location, compiler mnemonic, the default value, and a short description of the SCW data parameters:

SCW Sequential Defect Data Parameters

Byte	Mnemonic	Default	Description
0 - 1	max_time	900	Maximum execution time, 4 second intervals
2	write_loop	1	Write loop, number of write data pattern
3	read_loop	1	Read loop, number of reads after each write loop
4	retry_read_loop	8	Retry read loop, number of reads to verify data checks

5	retry_write_loop	4	Retry write/read loop, number of write/read loops after retry read loop
6	defect_threshold	2	Number of medium errors to assign defect
7	max_assign_defect	0xff	Maximum number of assigned defects, 0ffh - no limit
8 - 9	max_soft_head	0xffff	Maximum number of soft errors per head
10	max_hard_head	0xff	Maximum number of hard errors per head, 0ffh - no limit
11 - 12	max_soft_errors	0xffff	Maximum number of soft errors
13 - 14	max_hard_errors	0xffff	Maximum number of hard errors
15	margin_enable	0x00	Margin enable flags
16 - 17	offtrack	0x0000	Offtrack
18	boost	0x00	Boost, flag: enb_boost
19	vit_threshold	0x00	Viterbi threshold, flag: enb_vitthres
20	dpd_threshold	0x00	DPD threshold, flag: enb_dpdthres
21 - 22	retry_offtrack	0x0000	Retry offtrack
23	retry_boost	0x00	Retry Boost, flag: enb_rty_boost
24	retry_vit_thres	0x00	Retry Viterbi threshold, flag: enb_rty_vitthres
25	retry_dpd_thres	0x00	Retry DPD threshold, flag: enb_rty_dpdthres
26 - 27	start_cyl	0x0000	Starting cylinder
28 - 29	end_cyl	0xffff	Ending cylinder
30	data_flag	0x00	Data flag byte, flag: random
31	data_length	16	Data repeat pattern length
32 - 47	data	pattern1	Data, 16 bytes of 0x66

This next section is a detailed description of the SCW data parameters by the compiler's mnemonic:

max_time

The max_time (bytes 0 - 1) specifies the maximum allowed time the command is allowed to execute. The time is specified in four second increments and is defaulted to one hour. If the time is exceeded, the test is terminated with a command check.

write_loop

The write_loop (byte 2) specifies the number of times to write a track to search for defects. The sectors are written with the data specified.

After each write, the track is read to search for defects. The write_loop is defaulted to one write by the compiler. Specifying a value of zero, prevents any writing of the track.

The write operation has a debug trace of 0xff starting at buffer location 0x50000. See the Theory of Operation in the Selfscan Next Generation User's Guide.

read_loop

The read_loop (byte 3) specifies the number of times to read a track to search for defects after each track write. The read_loop is defaulted to one full track read by the compiler. Specifying a value of zero prevents any reading of the track.

The read operation has a debug trace of 0x7f starting at buffer location 0x50000. See the Theory of Operation in the Selfscan Next Generation User's Guide.

retry_read_loop

The retry_read_loop (byte 4) specifies the number of times to read verify a suspect sector. If an error occurs, "wiggle" recovery is initiated to check for a wiggle head problem. The "wiggle" recovery count is then compared with the defect threshold count. If the "wiggle" recovery count is greater than or equal to the defect threshold count, the sector is assigned as a "hard" error and the defect added to the Selfscan defect list. The retry_read_loop is default to eight reads by the compiler. Specifying a value less than the defect threshold count prevents the sector from defect assignment with this loop.

The retry read operation has a debug trace of 0x7e starting at buffer location 0x50000. See Theory of Operation in the Selfscan Next Generation User's Guide.

retry_write_loop

The retry_write_loop (byte 5) specifies the number of times to write then read a suspect sector. This loop is started only after the retry read verify loop. If an error occurs, "wiggle" recovery is initiated to check for a wiggle head problem. The "wiggle" recovery count is then compared with the defect threshold count. If the "wiggle" recovery count is greater than or equal to the defect threshold count, the sector is assigned as a "hard" error and the defect added to the Selfscan defect list. If no medium errors are found the "soft" error counts are updated. The retry_write_loop is default to eight writes then read by the compiler. Specifying a value less than the defect threshold count prevents the sector from defect assignment with this loop.

The retry write operation has a debug trace of 0xfe starting at buffer location 0x50000. See the Theory of Operation in the Selfscan Next Generation User's Guide.

defect_threshold

The defect_threshold (byte 6) count specifies the number of allowed medium errors before assigning a defect to the Selfscan defect list. The defect threshold count is used in both retry loops: the retry read loop, and the retry write/read loop. The compiler default of two, allows for one wiggle recovery before assigning a defect.

max_assign_defect

The max_assign_defect (byte 7) specifies the maximum number of assigned defect to the Selfscan defect list allowed by this SCW. The compiler default value is set to 0xff maximum assigned defects. A value of 0xff specifies "no limit" to the number of assigned defects. If the number of maximum assigned defects is exceeded, the command is terminated with command end and exception.

This feature allows a defect scan with a simulated "inline sparing", as long as the defect is in the defect list no defect is assigned, only the error statistics are updated. If the maximum number is exceeded, the defect list may be considered not constant.

max_soft_head

The max_soft_head (byte 8 - 9) specifies the maximum number of soft errors per head allowed before terminating the SCW. The compiler default is 0xFFFF, which is no limit. The soft error per head count is incremented each time a soft error is located. If the number of soft errors per head is exceeded the SCW command is terminated with a command end and exception.

max_hard_head

The max_hard_head (byte 10) specifies the maximum number of hard errors (repeatable read and write errors) per head allowed before terminating the SCW. The compiler default is 0xFF, which is no limit. The hard error per head count is incremented each time a hard error is located. If the number of hard errors per head is exceeded the SCW command is terminated with a command end and exception.

max_soft_errors

The max_soft_errors (byte 11 - 12) specifies the maximum number of soft errors allowed before terminating the SCW. The compiler default is 0xFFFF, which is no limit. The soft error count is incremented each time a soft error is located. If the number of soft errors is exceeded the SCW command is terminated with a command end and exception.

max_hard_errors

The max_hard_errors (byte 13 - 14) specifies the maximum number of hard errors (repeatable read and write errors) allowed before terminating the SCW. The compiler default is 0xFFFF, which is no limit. The hard error count is incremented each time a hard error is located. If the number of hard errors is exceeded the SCW command is terminated with a command end and exception.

margin_enable

The margin_enable (byte 15) flag specifies which R/W channel register margin values are loaded. Any margin values not enabled use the default value in the drive's zone tables. The compiler default value is 0x00, no margins enabled. The following symbols may be used to enable the margin bytes:

Bit	Symbol	Description
0	enb_boost	Enable boost value, R/W synthesizer, register 2
1	enb_vitthres	Enable Viterbi threshold value, register 1ch
2	enb_dpdthres	Enable DPD threshold value, register 1dh
3		Unused
4	enb_rty_boost	Enable retry boost value, R/W synthesizer, register 2
5	enb_rty_vitthres	Enable retry Viterbi threshold value, register 1ch
6	enb_rty_dpdthres	Enable retry DPD threshold value, register 1dh
7		Unused

Enabling the bits in the margin enable flag only allows the margin byte to load into the R/W register, NO BITS IN THE R/W REGISTER ARE CONTROLLED BY THIS FLAG.

offtrack

The offtrack (bytes 16 - 17) specifies the amount of offtrack to the DSP. The offtrack is defaulted to 0x0000 by the compiler. A value of 0x7fff specifies to move the servo plus one half track. A value of 0x8000 specifies to move the servo minus one half track. This margin value is always enabled and not controlled by the margin enable flag.

boost

The boost (byte 18) specifies the value of the R/W synthesizer register 2, bits 7 - 5. The margin enable flag (bit 0) must be enabled for this margin value to load in the R/W synthesizer register 2.

vit_threshold

The vit_threshold (byte 19) specifies the value of the digital R/W channel register 1ch. The margin enable flag (bit 1) must be enabled for this margin value to load into the register.

dpd_threshold

The dpd_threshold (byte 20) specifies the value of the digital R/W channel register 1dh. The margin enable flag (bit 2) must be enabled for this margin value to load into the register.

retry_offtrack

The retry_offtrack (bytes 21 - 22) specifies the amount of retry offtrack to the DSP. The retry offtrack is defaulted to 0x0000 by the compiler. A value of 0x7fff specifies to move the servo plus one half track. A value of 0x8000 specifies to move the servo minus one half track. This margin value is always enabled, and not controlled by the margin enable flag.

retry_boost

The retry_boost (byte 23) specifies the value of the R/W synthesizer register 2, bits 7 - 5. The margin enable flag (bit 4) must be enabled for this margin value to load in the R/W synthesizer register 2.

retry_vit_thres

The retry_vit_thres (byte 24) specifies the value of the digital R/W channel register 1ch. The margin enable flag (bit 5) must be enabled for this margin value to load into the register.

retry_dpd_thres

The retry_dpd_thres (byte 25) specifies the value of the digital R/W channel register 1dh. The margin enable flag (bit 6) must be enabled for this margin value to load into the register.

start_cyl

The start_cyl (bytes 26 - 27) specifies the starting cylinder number for the defect scan. The starting cylinder number is defaulted to 0x0000 by the compiler.

end_cyl

The end_cyl (bytes 28 - 29) specifies the ending cylinder number for the defect scan. The ending cylinder number is defaulted to 0xFFFF by the compiler. A value of 0xFFFF specifies the maximum cylinder of drive.

data_flag

The data_flag (byte 30) specifies the data pattern options. There are two options currently available: a user defined data pattern, and a random data pattern. The data flag byte is defaulted to zero, user defined data pattern.

Bit	Symbol	Description
0	random	Enable random pattern generator
1		This bit reserved for write sine, program check if set.

data_length

The data_length (byte 31) specifies the number of bytes in the user defined data pattern before repeating the sequence. A data pattern length may be defined from 1 to 16 bytes. The compiler is defaulted to a 16 byte repeating pattern. Any value not within the range presents a program check.

data

The data (bytes 32 - 47) specifies the user defined data pattern. The frequency of the data pattern is controlled by the data_length. The compiler is defaulted to a 0x66 pattern. See the Selfscan Next Generation defines file for predefined user data pattern symbols.

Ending Status

The ending status is presented to the Selfscan command processor by each Selfscan command. The ending status is presented in the status byte (byte 6) of the SSW (Selfscan Status Word). Additional error codes and command statistics are found in the SSW data parameter bytes. The following table gives the byte location and a short description of the SSW data parameters:

SSW Sequential Defect Seek Data Parameters

Byte	Symbol	Description
0	error_code	SCW error code
1	error_code_index	Error code index
2 - 3	elapsed_time	SCW elapsed time
4 - 5	last_cyl	Ending cylinder number
6	last_head	Ending head number
7 - 8	seek_recovered	Number of recovery seek errors
9 - 10	assign_defect	Number of assigned defects
11 - 12	soft_errors	Number of soft errors detected
13 - 14	hard_errors	Number of hard errors detected
15 - 44	soft_hd_errors (2 bytes / head)	Number of soft errors per head
45 - 74	hard_hd_errors (1 byte / head)	Number of hard errors per head

This section gives a detailed description of the SSW data parameter bytes by the disassembler mnemonic:

error_code

The SCW error code (byte 0) contains the reason the SCW was terminated. A command check, program check, or exception status must be presented for a valid error code.

error_code_index

The error code index (byte 1) is the sense key index value returned by internal subroutines in the drives operational firmware. This error code is an engineering error code to help determine the exact cause of the

failure. A command check, program check, or exception status must be presented for a valid error code.

elapsed_time

This is the total time (bytes 2 - 3) is in four second increments that the SCW required to execute.

last_cyl

The last_cyl (bytes 4 - 5) is the current cylinder number when the command terminates.

last_head

The last_head (byte 6) is the current head number when the command terminates.

seek_recovered

The seek_recovered (bytes 7 - 8) is the total number of recovered seek errors that occurred during the test.

assign_defect

The assign_defect (bytes 9 - 10) is the total number of assigned defects to the Selfscan Defect list.

soft_errors

The soft_errors (bytes 11 - 12) is the total number of soft errors that occurred during the test. A soft error occurs when a track is read, and a sector returns with a medium error that does not occur again, or wiggle recovery fixed for the retry loops.

hard_errors

The hard_errors (bytes 13 - 14) is the total number of hard errors that occurred during the test. A hard error occurs when a track is read and a sector returns with a medium error that is repeatable. Hard error counts are incremented each time an error is detected. A defect is only assigned if the sector is not in the defect list.

soft_hd_errors

The soft_hd_errors (bytes 15 - 44) is the total number of soft errors per head that occurred during the test and is a breakdown of the total soft errors to check for unsatisfactory heads. The data starts with head 0 (low byte, high byte) and continues to the last head in the drive.

hard_hd_errors

The hard_hd_errors (bytes 45 - 74) is the total number of hard errors per head that occurred during the test and is a break down of the total hard errors to check for unsatisfactory heads. The data starts with head 0 and continues to the last head in the drive.

SSW Print Display Example

```
SCW_ADDW SCW_COMMAND  FLAG STAT SCW_ERR ERR_IDX SCW_TIME(sec) LAST_CYL LAST_HD ASGN_DEF SOFT_ERR HARD_ERR
-------- -----------  ---- ---- ------- ------- ------------- -------- ------- -------- -------- --------
00a8h    SEQ_DEFECT   40h  08h  40h     00h     1136          2874     0        14      297       15
```

STOP START, Command 0x0A

0	1	2	3	4	5	6	7
0x0A	Data Address		Flags	Check Address		0	0x08

Selfscan Command Word

Command Description

The Stop Start command is a diagnostic test to exercise and measure the spin down and spin up motor characteristics. Since this test stops the motor, the failing test results may not be available on the drive's Selfscan test cylinder. Only the LED error code is available.
The SSW status may be read using the read micro memory command (RDMM) in SCSIDIAG. The location of the SSW may vary between code releases, so an updated listing must be obtained.

Basic Operation

The Stop Start command performs the following operations:

- Checks the validity of the SCW bytes.
- Sets the maximum SCW execution time.
- Checks the Maximum SCW limits.

***** Start of test loop

- Stops the drive spindle.
- Waits for a specified stop time.
- Starts the drive spindle.
- Checks to see if the drive "spun up" under specified Maximum time limit.
- Checks the test loop count.
- Present ending status.

Chaining Restrictions

- None

SCW Compiler Mnemonics

The Valid mnemonics for the Selfscan Next Generation Compiler are:

STOP_START	SS

SCW Data Parameters

The SCW data parameter bytes provide the additional specifications to execute a Selfscan command. The following table gives the byte location, compiler mnemonic, the default value, and a short description of the SCW data parameters:

SCW Stop Start Data Parameters

Byte	Mnemonic	Default	Description
0 - 1	loop_cnt	1	Loop count
2 - 3	max_time	180	Maximum SCW execution time, 4 second resolution
4 - 5	stop_time	4	Spindle off time, 4 second resolution
6 - 7	max_start_time	5	Maximum start time, 4 second resolution

This next section is a detailed description of the SCW data parameters by the compiler's mnemonic:

loop_cnt — The loop_cnt (bytes 0 - 1) specifies the number of Stop Start operations to execute. The compiler is defaulted to one Stop Start test. A value of zero performs 65,535 Stop Start tests or until the maximum SCW time is exceeded.

max_time — The max_time (bytes 2 -3) specifies the maximum allowed time for the SCW. The time is specified in four second increments and is defaulted to 12 minutes by the compiler. If the time is exceeded, the SCW command is terminated with a command end and exception raised. An attempt is made to start the motor so the results may be written to the disk.

stop_time — The stop_time (bytes 4 - 5) specifies the time the spindle remains off before starting the spindle. The time is specified in four second increments and is defaulted to 16 seconds by the compiler.

max_start_time — The max_start_time (bytes 6 - 7) specifies the maximum allowed starting time for the spindle motor. The time is specified in four second increments and is defaulted to 20 seconds. If the time is exceeded, the SCW command is terminated with a command end and exception.

Ending Status

The ending status is presented to the Selfscan command processor by each Selfscan command. The ending status is presented in the status byte (byte 6) of the SSW (Selfscan Status Word). Additional error codes and command statistics are found in the SSW data parameter bytes. The following table gives the byte location and a short description of the SSW data parameters:

SSW Stop Start Data Parameters

Byte	Symbol	Description
0	error_code	SCW error code
1	error_code_index	Error code index
2 - 3	elapsed_time	SCW elapsed time

This section gives a detailed description of the SSW data parameter bytes by the disassembler mnemonic:

error_code — The SCW error code (byte 0) contains the reason the SCW was terminated. A command check, program check, or exception status must be presented for a valid error code.

error_code_index — The error code index (byte 1) is the sense key index value returned by internal subroutines in the drive's operational firmware. This error code is an engineering error code to help determine the exact cause of the failure. A command check, program check, or exception status must be presented for a valid error code.

elapsed_time — This is the total time (bytes 2 - 3) is in four second increments that the SCW required to execute.

SSW Print Display Example

```
SCW_ADDR SCW_COMMAND  FLAG STAT SCW_ERR ERR_IDX SCW_TIME(sec)
-------- -----------  ---- ---- ------- ------- -------------
0010h    STOP_START   40h  08h  00h     00h       40
```

DIGITAL DEFECT, Command 0x0B

0	1	2	3	4	5	6	7
0x0B	Data Address		Flags	Check Address		0	0x30

Selfscan Command Word

Command Description

The Digital Defect command is a diagnostic test to search for defective wedge sectors on a cylinder sequentially from the specified starting cylinder to the specified ending cylinder. If a defect is located, the defect is added to the Selfscan defect list file. The reading of the sectors may be stressed by offtrack reading, threshold, and boost in the R/W channel. Sequential defect scan keeps statistics on the number of times a defect has been located, the number of soft and hard errors, the number of wiggle recovered errors, the number of recovered seek operations, and the total time the diagnostic test ran.

All non-medium errors in the Digital Defect command are retried the number of times specified in the retry count of MODE PAGE 1 (default is eight retries). Setting the retry count in MODE PAGE 1 to zero halts all retries of non_medium errors (i.e. write faults). Setting the retry count in MODE PAGE 1 to 255 retries all non_medium errors 255 times.

Basic Operation

The criteria to locate a "wedge" defect is defined in the SCW data block. Each track is read the number of times specified (search read loop count) for each search write loop count. The read channel values (offtrack, boost, Viterbi threshold, and DPD threshold) should be "stressed" (i.e. close to the edge of failure) to locate marginal medium defects. This "search" algorithm is used to locate a possible defect on the track.

Should a medium error (write or read) be detected by the "search" algorithm, the next user-specified criteria determines if the medium error is a "soft" or "hard (assigned defect)" error. The suspect "wedge sector" is first read the number of times specified in the retry read verify loop count. If an error occurs, "wiggle" recovery is initiated to check for a wiggle head problem. The "wiggle" recovery count is then compared with the defect threshold count specified in the SCW. If the "wiggle" recovery count is greater than or equal to the defect threshold count, the "wedge sector" is scaned for all defects, then assigned as "hard" errors and the defects added to the Selfscan defect list in bytes from wedge format. The read channel values can be "stressed" with the retry offtrack and retry margin values to locate the defect.

If no medium errors are detected in the retry read verify loop, the next step in the "defect" algorithm is the retry write/read verify loop. The suspect "wedge sector" is written and then read back using the retry margin values for the number of times specified in the retry write/read verify loop count. If an error occurs, "wiggle" recovery is initiated to check for a wiggle head problem. The "wiggle" recovery count is then compared with the defect threshold count specified in the SCW. If the "wiggle" recovery count is greater than or equal to the defect threshold count, the "wedge sector" is scanned for all defects, then assigned as "hard" errors and the defects added to the Selfscan defect list in bytes from wedge format. If no medium errors were detected in the suspect "wedge sector" after the write/verify loop count, the "wedge sector" is assumed to be a "soft" error and the soft error statistics are updated. The defect "search" continues until the ending cylinder has been reached.

Digital defect scan SCW data also has several maximum limits to terminate the Selfscan test early should a value be exceeded. The maximum SCW time, the maximum total number of "soft" and "hard" errors, the maximum number of "soft" and "hard" errors per head, the maximum number of recovered seek errors, and the maximum number of assigned defects are all specified in the SCW data block. If a limit is exceeded, the SCW is terminated.

The Digital Defect scan command performs the following operations:

- Checks the validity of the SCW bytes.
- Checks the validity of the SCW data bytes.
- Checks for valid cylinders.
- Fills the write data buffer with specified data bytes.
- Initializes the counters, timers, starting cylinder, head, and wedge numbers.
- Initializes the debug trace

**** Defect Scan Loop

- Checks to see if the maximum limits have been exceeded.
- Checks to see if maximum number of hard errors has been exceeded.
- Checks to see if maximum number soft errors has been exceeded.
- Writes and reads back each track the number of times specified.
- If a medium error occurs, the "wedge sector" is re-read the number of times specified, then written and read the number of times specified to determine if a defect exits.

- If a defect is located, the defect is added to the Selfscan Defect list file and track statistics are updated.

- Increments head, cylinder, and continues with the Defect Scan Loop.

**** End Defect Scan Loop

- Checks the maximum recovered seek error limit.

- Present ending status.

Chaining Restrictions

- This command destroys all track formats. The Format Media command must be called before any sector defect scanning can be performed.

SCW Compiler Mnemonics

The Valid mnemonics for the Selfscan Next Generation Compiler are:

DIGITAL_DEFECT DIGITAL_SCAN DDS

SCW Data Parameters

The SCW data parameter bytes provide the additional specifications to execute a Selfscan command. The following table gives the byte location, compiler mnemonic, the default value, and a short description of the SCW data parameters:

SCW Digital Defect Data Parameters

Byte	Mnemonic	Default	Description
0 - 1	max_time	900	Maximum execution time, 4 second intervals
2	write_loop	1	Write loop, number of write data pattern
3	read_loop	1	Read loop, number of reads after each write loop
4	retry_read_loop	8	Retry read loop, number of reads to verify data checks

5	retry_write_loop	4	Retry write/read loop, number of write/read loops after retry read loop
6	defect_threshold	2	Number of medium errors to assign defect
7	max_assign_defect	0xff	Maximum number of assigned defects, 0ffh - no limit
8 - 9	max_soft_head	0xffff	Maximum number of soft errors per head
10	max_hard_head	0xff	Maximum number of hard errors per head, 0ffh - no limit
11 - 12	max_soft_errors	0xffff	Maximum number of soft errors
13 - 14	max_hard_errors	0xffff	Maximum number of hard errors
15	margin_enable	0x00	Margin enable flags
16 - 17	offtrack	0x0000	Offtrack
18	boost	0x00	Boost, flag: enb_boost
19	vit_threshold	0x00	Viterbi threshold, flag: enb_vitthres
20	dpd_threshold	0x00	DPD threshold, flag: enb_dpdthres
21 - 22	retry_offtrack	0x0000	Retry offtrack
23	retry_boost	0x00	Retry Boost, flag: enb_rty_boost
24	retry_vit_thres	0x00	Retry Viterbi threshold, flag: enb_rty_vitthres
25	retry_dpd_thres	0x00	Retry DPD threshold, flag: enb_rty_dpdthres
26 - 27	start_cyl	0x0000	Starting cylinder
28 - 29	end_cyl	0xffff	Ending cylinder
30	data_flag	wr_sine	Data flag byte, flag: random, wr_sine
31	data_length	16	Data repeat pattern length
32 - 47	data	pattern3	Data, 16 bytes of 0xff

This next section is a detailed description of the SCW data parameters by the compiler's mnemonic:

max_time — The max_time (bytes 0 - 1) specifies the maximum allowed time the command is allowed to execute. The time is specified in four second increments and is defaulted to one hour. If the time is exceeded, the test is terminated with a command check.

write_loop — The write_loop (byte 2) specifies the number of times to write a track to search for defects. The wedge sectors are written with the data specified. After each write, the track is read to search for defects. The write_loop is defaulted to one write by the compiler. Specifying a value of zero prevents any writing of the track.

The write operation has a debug trace of 0xff starting at buffer location 0x50000. See the Theory of Operation in the Selfscan Next Generation User's Guide.

read_loop

The read_loop (byte 3) specifies the number of times to read a track to search for defects after each track write. The read_loop is defaulted to one full track read by the compiler. Specifying a value of zero, will prevent any reading of the track.

The read operation has a debug trace of 0x7f starting at buffer location 0x50000. See the Theory of Operation in the Selfscan Next Generation User's Guide.

retry_read_loop

The retry_read_loop (byte 4) specifies the number of times to read verify a suspect wedge sector. If an error occurs, "wiggle" recovery is initiated to check for a wiggle head problem. The "wiggle" recovery count is then compared with the defect threshold count. If the "wiggle" recovery count is greater than or equal to the defect threshold count, the "wedge sector" is scanned for all defects, then assigned as "hard" errors and the defects added to the Selfscan defect list. The retry_read_loop is defaulted to eight reads by the compiler. Specifying a value less than the defect threshold count prevents the "wedge sector" from defect assignment with this loop.

The retry read operation has a debug trace of 0x7e starting at buffer location 0x50000. See the Theory of Operation in the Selfscan Next Generation User's Guide.

retry_write_loop

The retry_write_loop (byte 5) specifies the number of times to write then read a suspect wedge sector. This loop is started only after the retry read verify loop. If an error occurs, "wiggle" recovery is initiated to check for a wiggle head problem. The "wiggle" recovery count is then compared with the defect threshold count. If the "wiggle" recovery count is greater than or equal to the defect threshold count, the wedge sector is scanned for all defects, then assigned as a "hard" errors and the defects added to the Selfscan defect list. If no medium errors are found the "soft" error counts are updated. The retry_write_loop is defaulted to eight writes then reads by the compiler. Specifying a value less than the defect threshold count prevents the "wedge sector" from defect assignment with this loop.

The retry write operation has a debug trace of 0xfe starting at buffer location 0x50000. See the Theory of Operation in the Selfscan Next Generation User's Guide.

defect_threshold

The defect_threshold (byte 6) count specifies the number of allowed medium errors before assigning a defect to the Selfscan defect list. The defect threshold count is used in both retry loops: the retry read loop, and the retry write/read loop. The compiler default of two, allows for one wiggle recovery before assigning a defect.

max_assign_defect

The max_assign_defect (byte 7) specifies the maximum number of assigned defects to the Selfscan defect list allowed by this SCW. The compiler default value is set to 0xff maximum assigned defects. A value of 0xff specifies "no limit" to the number of assigned defects. If the number of maximum assigned defects is exceeded, the command is terminated with command end and exception.

This feature allows a defect scan with a simulated "inline sparing", as long as the defect is in the defect list no defect is assigned, only the error statistics are updated. If the maximum number is exceeded, the defect list may be considered not constant.

max_soft_head

The max_soft_head (byte 8 - 9) specifies the maximum number of soft errors per head allowed before terminating the SCW. The compiler default is 0xFFFF, which is no limit. The soft error per head count is incremented each time a soft error is located. If the number of soft errors per head is exceeded, the SCW command is terminated with a command end and exception.

max_hard_head

The max_hard_head (byte 10) specifies the maximum number of hard errors (repeatable read and write errors) per head allowed before terminating the SCW. The compiler default is 0xFF, which is no limit. The hard error per head count is incremented each time a hard error is located. If the number of hard errors per head is exceeded the SCW command is terminated with a command end and exception.

max_soft_errors

The max_soft_errors (byte 11 - 12) specifies the maximum number of soft errors allowed before terminating the SCW. The compiler default is 0xFFFF, which is no limit. The soft error count is incremented each time a soft error is located. If the number of soft errors is exceeded the SCW command is terminated with a command end and exception.

max_hard_errors

The max_hard_errors (byte 13 - 14) specifies the maximum number of hard errors (repeatable read and write errors) allowed before terminating the SCW. The compiler default is 0xFFFF, which is no limit. The hard error count is incremented each time a hard error is located. If the number of hard errors is exceeded the SCW command is terminated with a command end and exception.

margin_enable

The margin_enable (byte 15) flag specifies which R/W channel register margin values are loaded. Any margin values not enabled use the default value in the drive's zone tables. The compiler default value is 0x00, which is no margins enabled. The following symbols may be used to enable the margin bytes:

Bit	Symbol	Description
0	enb_boost	Enable boost value, R/W synthesizer, register 2
1	enb_vitthres	Enable Viterbi threshold value, register 1ch
2	enb_dpdthres	Enable DPD threshold value, register 1dh
3		Unused
4	enb_rty_boost	Enable retry boost value, R/W synthesizer, register 2
5	enb_rty_vitthres	Enable retry Viterbi threshold value, register 1ch
6	enb_rty_dpdthres	Enable retry DPD threshold value, register 1dh
7		Unused

Enabling the bits in the margin enable flag only allows the margin byte to load into the R/W register, THIS FLAG CONTROLS NO BITS IN THE R/W REGISTER.

offtrack

The offtrack (bytes 16 - 17) specifies the amount of offtrack to the DSP. The offtrack is defaulted to 0x0000 by the compiler. A value of 0x7fff specifies to move the servo plus one half track. A value of 0x8000 specifies to move the servo minus one half track. This margin value is always enabled and not controlled by the margin enable flag.

boost

The boost (byte 18) specifies the value of the R/W synthesizer register 2, bits 7 - 5. The margin enable flag (bit 0) must be enabled for this margin value to load in the R/W synthesizer register 2.

vit_threshold

The vit_threshold (byte 19) specifies the value of the digital R/W channel register 1ch. The margin enable flag (bit 1) must be enabled for this margin value to load into the register.

dpd_threshold

The dpd_threshold (byte 20) specifies the value of the digital R/W channel register 1dh. The margin enable flag (bit 2) must be enabled for this margin value to load into the register.

retry_offtrack

The retry_offtrack (bytes 21 - 22) specifies the amount of retry offtrack to the DSP. The retry offtrack is defaulted to 0x0000 by the compiler. A value of 0x7fff specifies to move the servo plus one half track. A value of 0x8000 specifies to move the servo minus one half track. This margin value is always enabled, and not controlled by the margin enable flag.

retry_boost

The retry_boost (byte 23) specifies the value of the R/W synthesizer register 2, bits 7 - 5. The margin enable flag (bit 4) must be enabled for this margin value to load in the R/W synthesizer register 2.

retry_vit_thres

The retry_vit_thres (byte 24) specifies the value of the digital R/W channel register 1ch. The margin enable flag (bit 5) must be enabled for this margin value to load into the register.

retry_dpd_thres

The retry_dpd_thres (byte 25) specifies the value of the digital R/W channel register 1dh. The margin enable flag (bit 6) must be enabled for this margin value to load into the register.

start_cyl

The start_cyl (bytes 26 - 27) specifies the starting cylinder number for the defect scan. The starting cylinder number is defaulted to 0x0000 by the compiler.

end_cyl

The end_cyl (bytes 28 - 29) specifies the ending cylinder number for the defect scan. The ending cylinder number is defaulted to 0xFFFF by the compiler. A value of 0xFFFF specifies the maximum cylinder of drive.

data_flag

The data_flag (byte 30) specifies the data pattern options. There are three options currently available: a user defined data pattern, the write sine pattern, and a random data pattern. The write sine pattern is a special pattern for the R/W channel. The data flag byte is defaulted to write sine.

Bit	Symbol	Description
0	random	Enable random pattern generator
1	wr_sine	Enable write sine in R/W unit

data_length

The data_length (byte 31) specifies the number of bytes in the user defined data pattern before repeating the sequence. A data pattern length may be defined from 1 to 16 bytes. The compiler is defaulted to a 16 byte repeating pattern. Any value not within the range presents a program check.

data

The data (bytes 32 - 47) specifies the user defined data pattern. The frequency of the data pattern is controlled by data_length. The compiler is defaulted to a 0xff pattern, the input data required for write sine. See the Selfscan Next Generation defines file for predefined user data pattern symbols.

Ending Status

The ending status is presented to the Selfscan command processor by each Selfscan command. The ending status is presented in the status byte (byte 6) of the SSW (Selfscan Status Word). Additional error codes and command statistics are found in the SSW data parameter bytes. The following table gives the byte location and a short description of the SSW data parameters:

SSW Digital Defect Seek Data Parameters

Byte	Symbol	Description
0	error_code	SCW error code
1	error_code_index	Error code index
2 - 3	elapsed_time	SCW elapsed time
4 - 5	last_cyl	Ending cylinder number
6	last_head	Ending head number
7 - 8	seek_recovered	Number of recovery seek errors
9 - 10	assign_defect	Number of assigned defects
11 - 12	soft_errors	Number of soft errors detected
13 - 14	hard_errors	Number of hard errors detected
15 - 44	soft_hd_errors (2 bytes / head)	Number of soft errors per head
45 - 74	hard_hd_errors (1 byte / head0	Number of hard errors per head

This section gives a detailed description of the SSW data parameter bytes by the disassembler mnemonic:

error_code

The SCW error code (byte 0) contains the reason the SCW was terminated. A command check, program check, or exception status must be presented for a valid error code.

error_code_index

The error code index (byte 1) is the sense key index value returned by internal subroutines in the drives operational firmware. This error code is an engineering error code to help determine the exact cause of the failure. A command check, program check, or exception status must be presented for a valid error code.

elapsed_time

This is the total time (bytes 2 - 3) is in four second increments that the SCW required to execute.

last_cyl

The last_cyl (bytes 4 - 5) is the current cylinder number when the command terminates.

last_head

The last_head (byte 6) is the current head number when the command terminates.

seek_recovered

The seek_recoverd (bytes 7 - 8) is the total number of recovered seek errors that occurred during the test.

assign_defect

The assign_defect (bytes 9 - 10) is the total number of assigned defects to the Selfscan Defect list.

soft_errors

The soft_errors (bytes 11 - 12) is the total number of soft errors that occurred during the test. A soft error occurs when a track is read and a wedge sector returns with a medium error that does not occur again, or wiggle recovery fixed for the retry loops.

hard_errors

The hard_errors (bytes 13 - 14) is the total number of hard errors that occurred during the test. A hard error occurs when a track is read and a wedge sectors returns with a medium error that is repeatable. Hard error counts are incremented each time an error is detected. A defect is only assigned if the wedge sector is not in the defect list.

soft_hd_errors

The soft_hd_errors (bytes 15 - 44) is the total number of soft errors per head that occurred during the test. Soft_hd_errors is a break down of the total soft errors and is used to check for unsatisfactory heads. The data starts with head 0 (low byte, high byte) and continues to the last head in the drive.

hard_hd_errors

The hard_hd_errors (bytes 45 - 75) is the total number of hard errors per head that occurred during the test. Hard_hd_errors is a break down of the total hard errors and is used to check for unsatisfactory heads. The data starts with head 0 and continues to the last head in the drive.

SSW Print Display Example

```
SCW_ADDW SCW_COMMAND  FLAG STAT SCW_ERR ERR_IDX SCW_TIME(sec) LAST_CYL LAST_HD ASGN_DEF SOFT_ERR HARD_ERR
-------- -----------  ---- ---- ------- ------- ------------- -------- ------- -------- -------- --------
0078h    DIG_DEFECT   40h  08h  40h     03h     1100          2874     0         6      179        9
```

FIR TRAINING, Command 0x0C

0	1	2	3	4	5	6	7
0x0C	Data Address		Flags	Check Address		0	0x07

Selfscan Command Word

Command Description

The FIR Training command is a diagnostic to "train" the PRML (Partial Response Maximum Likelihood) R/W channel for each head in each zone. This command requires that the "untrained" FIR coefficients are loaded in the buffer at power-up (i.e. a drive with "trained" FIR coefficients can't be trained again).

All non-medium errors in the Digital Defect command are retried the number of times specified in the retry count of MODE PAGE 1 (the default is eight retries). Setting the retry count in MODE PAGE 1 to zero halts all retries of non_medium errors (i.e. write faults). Setting the retry count in MODE PAGE 1 to 255 retries all non_medium errors 255 times.

Basic Operation

For each head in each zone, the FIR training command writes a full track of "wedge sectors" (256 bytes long) using the write sine data option. The test searches for one "wedge sector" that is defect free (NO ECC ERRORS). That "wedge sector" is written with the PRML R/W channel training pattern (a fix random pattern). The PRML channel is setup for FIR coefficient training and the "wedge sector" is read. The coefficients are checked for value limits, and the "wedge sector" reread with training if necessary. The "trained FIR coefficients" are saved in the "trained FIR coefficient table". When the last head of the last zone is "trained" the "trained FIR coefficient table" is written to the system cylinder and the drive's FIR coefficient tables are updated with "trained" coefficients.

The FIR Training command performs the following operations:

- Checks the validity of the SCW bytes.
- Checks for valid FIR tables.
- Copies the "untrained" FIR table to the "trained" FIR table.

***** FIR Training Loop

- Write "wedge sectors" for a full track with write sine data.
- Search for first "wedge sector" with no defects (NO ECC ERRORS).
- Write "wedge sector" with training pattern.
- Read "wedge sector" to "train" the FIR coefficients.
- Checks FIR coefficient limits, and re-train if necessary.
- Zero out any unnecessary taps as specified.
- Load "trained" FIR coefficients in "trained" FIR table.
- Repeat for each head, all zones.
- Write "trained" FIR coefficients on system cylinder.
- Set valid status for "trained" FIR tables.
- Present ending status.

Chaining Restrictions

- The "untrained" FIR coefficients must be in the FIR coefficient tables. If the "trained" FIR coefficients are loaded the command is terminated with a command end and exception raised.
- This test destroys the track format in all zones. The Format Media command must be executed to restore the track formats.

SCW Compiler Mnemonics

The Valid mnemonics for the Selfscan Next Generation Compiler are:

FIR_TRAINING FIR

SCW Data Parameters

The SCW data parameter bytes provide the additional specifications to execute a Selfscan command. The following table gives the byte location, compiler mnemonic, the default value, and a short description of the SCW data parameters:

SCW FIR Training Data Parameters

Byte	Mnemonic	Default	Description
0 - 1	max_time	900	Maximum execution time, 4 second intervals
2 - 3	max_sum	0x26	Maximum FIR coefficient sum
4 - 5	retry_count	8	FIR training retry count
6	tap_count	5	Number of taps. Limit: 1, 3, or 5

This next section is a detailed description of the SCW data parameters by the compiler's mnemonic:

max_time

The max_time (bytes 0 - 1) specifies the maximum allowed time the command is allowed to execute. The time is specified in four second increments and is defaulted to one hour. If the time is exceeded, the test is terminated with a command check.

max_sum

The max_sum (bytes 2 - 3) specifies the maximum allowed sum (even or odd) of the FIR coefficients after training. The training of the PRML channel continues until the FIR coefficient sum is less than the maximum FIR coefficient sum or the retry count is exceeded. The maximum FIR coefficient sum is defaulted to 0x26 for both the even and odd sums. If the retry count is exceeded the command terminates with a command check.

retry_count

The retry_count (bytes 4 - 5) specifies the retry count for the FIR training sequence only. The retry count is defaulted to eight retries for FIR training. The retry count for other read and write sequences is controlled by the MODE PAGE 1 retry count.

tap_count

The tap_count (byte 6) specifies the number of taps in the FIR filter. The tap_count may be specified as 1, 3, or 5 and is defaulted to 5 taps. If the tap_count is not within the specified range, the command terminates with a program check.

Ending Status

The ending status is presented to the Selfscan command processor by each Selfscan command. The ending status is presented in the status byte (byte 6) of the SSW (Selfscan Status Word). Additional error codes and command statistics are found in the SSW data parameter bytes. The following table gives the byte location and a short description of the SSW data parameters:

SSW FIR Training Data Parameters

Byte	Symbol	Description
0	error_code	SCW error code
1	error_code_index	Error code index
2 - 3	elapsed_time	SCW elapsed time
4 - 5	last_cyl	Current cylinder number
6	last_head	Current head number
7	last_wedge	Current wedge number

This section gives a detailed description of the SSW data parameter bytes by the disassembler mnemonic:

error_code — The SCW error code (byte 0) contains the reason the SCW was terminated. A command check, program check, or exception status must be presented for a valid error code.

error_code_index — The error code index (byte 1) is the sense key index value returned by internal subroutines in the drive's operational firmware. This error code is an engineering error code to help determine the exact cause of the failure. A command check, program check, or exception status must be presented for a valid error code.

elapsed_time — This is the total time (bytes 2 - 3) is in four second increments that the SCW required to execute.

last_cyl — The last_cyl (bytes 4 - 5) is the current cylinder number when the command terminates.

last_head — The last_head (byte 6) is the current head number when the command terminates.

last_wedge — The last_wedge (byte 7) is the current wedge number when the command terminates.

SSW Print Display Example

```
SCW_ADDW SCW_COMMAND  FLAG STAT SCW_ERR ERR_IDX SCW_TIME(sec) LAST_CYL LAST_HD LAST_WEG
-------- -----------  ---- ---- ------- ------- ------------- -------- ------- --------
0068h    FIR_TRAIN    40h  09h  43h     00h        0          -001     0       36
```

Appendix A

LED Error Codes

Error Code	Mnemonic	Description
0x00	led_normal	No error
0x03	led_por	Failed power-up
0x04	led_cmd_check	Command check
0x05	led_prgm_check	Program check
0x06	led_rd_file	Failed read command chain file
0x07	led_wr_file	Failed writing command chain file
0x08	led_result_full	Result file full
0x09	led_spindle	Failed spin-up

Appendix B

Termination Error Codes

Error Code	Mnemonic	Description
0x00	tr_no_term	Normal ending
0x10	tr_result_full	Result file full
0x11	tr_invalid_status	Invalid SSW ending status
0x12	tr_chain_timeout	Overall chain timeout
0x14	tr_cmd_check	Command check
0x15	tr_prgm_check	Program check
0x16	tr_max_assign_def	Maximum number of assigned defects exceeded
0x17	tr_failed_spin	Failed spin-up

Appendix C

SCW Error Codes

Error Code	Mnemonic	Description
0x00	ss_no_error	No error
0x1f	ss_scw_terminated	SCW terminated by command processor
0x20	ss_one_header	More than one SCW header command in chain
0x21	ss_no_header	No SCW header command at start of chain
0x22	ss_invalid_cmd	Invalid SCW command
0x23	ss_invalid_password	Invalid password
0x24	ss_initial_result	Failed initialization results file
0x25	ss_initial_defect	Failed initialization defect file
0x26	ss_invalid_defect	Invalid defect file
0x27	ss_invalid_result	Invalid result file
0x28	ss_chain_timeout	Chain timeout
0x29	ss_scw_timeout	SCW timeout
0x2a	ss_hard_errors	Maximum overall hard errors reached
0x2b	ss_seek_errors	Maximum overall seek errors reached
0x2c	ss_sect_rd_error	Failed reading selfscan file
0x2d	ss_sect_wr_error	Failed writing selfscan file
0x2e	ss_no_icl	No TIC ICL in chain
0x2f	ss_idle_call	Fatal error from idle call vector
0x30	ss_avg_seek_timeout	Average seek timeout
0x31	ss_max_seek_timeout	Maximum seek timeout
0x32	ss_max_seek_error	Maximum number of seek errors exceeded
0x33	ss_recal_failure	Recal failure
0x34	ss_format_track	Format track failure
0x35	ss_clear_defect	Failed to clear defect list
0x36		Unused
0x37	ss_hard_limit_scw	Maximum number of hard errors reached this scw
0x38	ss_hard_limit_hd_scw	Maximum number of hard errors per head reached this scw
0x39	ss_soft_limit_scw	Maximum number of soft errors reached this scw
0x3a	ss_soft_limit_hd_scw	Maximum number of soft errors reached this scw

0x3b	ss_track_scan_fail	Track scan failure
0x3c	ss_medium_fail	Failed medium error
0x3d	ss_writing_defect	Failed writing defect file
0x3e	ss_defect_file_full	Defect file full
0x3f		Unused
0x40	ss_defect_assigned	Defect assigned
0x41	ss_fir_sum	Maximum FIR coefficient sum exceeded
0x42	ss_no_fir_coeff	No FIR coefficients loaded
0x43	ss_trained_fir_cof	Trained FIR coefficients loaded
0x44	ss_no_fir_wedge	No FIR wedge found for training
0x45	ss_fail_training	Failed FIR training
0x4e	ss_max_def_scw	Maximum number of assigned defects exceeded per SCW
0x4f	ss_max_assign_def	Maximum number of assigned defects exceeded
0x50		Unused
0x51	ss_start_spin	Failed start spin up
0x52	ss_spin_timeout	Start spin up timeout

Appendix D

Error Code Index

The error code index values are from the TABLES.A file in the source files. Any errors not found in this table must check with the current TABLES.A for your project.

Error Code	Mnemonic	Description
0x00	ec_no_error	No error
0x01	ec_no_index	No index signal
0x02	ec_settle_timeout	Timeout in settling
0x03	ec_write_fault	Write fault
0x08	ec_data_ecc	Uncorrectable data ECC error
0x0a	ec_too_many_bumps	Bump retry counter expired
0x0c	ec_data_sync_tmo	Data field sync timeout
0x0e	ec_no_record_found	No record found
0x10	ec_seek_error	Seek error
0x12	ec_data_sync_mrkr	Data sync or marker not found
0x15	ec_fmt_failure	Drive format did not complete
0x16	ec_bad_dfct_list	Bad defect list
0x22	ec_motor_rpm_error	Motor never gets up to speed
0x26	ec_not_ready	Drive not ready
0x27	ec_dfct_list_full	Defect table is full, no more entries
0x28	ec_buffer_ram	RAM error
0x29	ec_ram_parity	RAM parity error
0x2a	ec_id_sync_tmo	AM mark not found for ID field
0x2d	ec_assert_error	Logical assertion (firmware consistency)
0x2f	ec_rom_chksum	Internal ROM checksum error
0x32	ec_prom_chksum	External PROM checksum error
0x33	ec_write_system	Error writing a system sector

0x34	ec_read_system	Error reading a system sector
0x35	ec_motor_fault	Motor drops out of legal speed range
0x37	ec_seq_ram_fail	Fail writing to sequencer format RAM
0x39	ec_unxpctd_seq_err	Unexpected sequencer error
0x3c	ec_bad_head_amp	Bad head amplifier
0x3d	ec_hd_miscompare	Head miscompare
0x3e	ec_invalid_head	Invalid head specified
0x3f	ec_invalid_cyl	Invalid cylinder specified
0x41	ec_bad_bps_bpb	Bytes/block, bytes/sector gives remainder
0x43	ec_recaling	Drive is recalibration
0x44	ec_spinning	Drive is spinning up
0x45	ec_stopped	Drive has not been told to spin up
0x47	ec_invalid_sector	Invalid sector specified
0x49	ec_fifo_unload	FIFO unload error
0x4a	ec_fifo_load	FIFO load error
0x4b	ec_fifo_pred_full	FIFO predicted full error
0x4d	ec_seq_timeout	Sequencer timeout
0x4f	ec_bump_timeout	Bump timeout
0x52	ec_seq_rollover	Sequencer rollover register failure
0x53	ec_external_sram	External SRAM failure
0x54	ec_external_ram	External RAM failure
0x58	ec_id_err	No record found
0x5a	ec_offtrack_timeout	Offtrack timeout
0x5b	ec_crc_cont	ID CRC error
0x61	ec_underrun	Underrun error
0x63	ec_spin_fail	Failed to spin up
0x64	ec_cal_fail	Unable to calibrate
0x65	ec_dsp_fail	DSP failed to report ready

0x66	ec_dsp_tune_dac	Recal failure during DSP DAC offset turn
0x67	ec_cal_init	DSP stopped operation during initialization
0x68	ec_nec_tune_dac	Failure DAC offset tune
0x69	ec_pes_bias_cal	Failure PES gain/bias calibration
0x6a	ec_kt_cal	Failure KT/J calibration
0x6b	ec_rro_cal	Failure once around calibration
0x6c	ec_hdo_cal	Failure head offset calibration
0x6d	ec_cleanup_cal	Failure during clean up calibration
0x6e	ec_rcal_no_servo	Can't servo on one or more heads at recal
0x7c	ec_recal	Recalibrate failure
0x7e	ec_dsp_not_rdy	DSP has quit, or is unable to start up
0x80	ec_seek_timeout	Seek timeout error
0x82	ec_dsp_com_failed	Unsuccessful communication with DSP
0x84	ec_dsp_start_er	DSP failed to go ready at start up
0x86	ec_med_corrupted	Medium format corrupted
0x87	ec_fatal_servo_error	Not ready - fatal servo error
0x88	ec_fatal_stack_overflow	Stack overflow

Appendix E

Model.Def

```
;
; SELFSCAN MODEL DEFINITION FILE
;
; This file contains the descriptions of the Selfscan model definitions for the Selfscan Next Generation SCW compiler.
;
;
$DEFINE MODEL="1440S"                                                   ;empire 1440 drive
;
; SELFSCAN MODEL DEFINIONS:
;
$IF MODEL="1440S" THEN $DEFINE MAX_CYL=3052 : $DEFINE LAST_HEAD=7       ;maximum cylinder, head number
$IF MODEL="2160S" THEN $DEFINE MAX_CYL=3052 : $DEFINE LAST_HEAD=11      ;maximum cylinder, head number
```

Appendix F

Selfscan.Def

```
;
; SELFSCAN DEFINITION FILE
;
; This file contains the descriptions of the Selfscan Command Words (SCW) for the Selfscan Next Generation Compiler
;
; SELFSCAN BINARY FILE SPECIFICATIONS:
;
$BUFFER_SIZE=0x0c00                               ; size of command/data buffer file
$CMD_FILE_DATA=0x0400                             ; start of data in buffer file
;
; MODEL DEFINES
;
$INCLUDE model.def
;
; GLOBAL DEFINES:
;
$DEFINE PATTERN0="\0x00\0x00\0x00\0x00\0x00\0x00\0x00\0x00\0x00\0x00\0x00\0x00\0x00\0x00\0x00\0x00"
$DEFINE PATTERN1="ffffffffffffffff"               ;defect scan default pattern (66h)
$DEFINE PATTERN2="wwwwwwwwwwwwwwww"               ;defect scan default pattern (77h)
$DEFINE PATTERN3="\0xff\0xff\0xff\0xff\0xff\0xff\0xff\0xff\0xff\0xff\0xff\0xff\0xff\0xff\0xff\0xff"
$DEFINE PATTERN4=16*"\001"                        ;defect scan default pattern (01h)
;
; SELFSCAN COMMAND DESCRIPTIONS:
;
$START
$NAME=SCW_HEADER, HEADER
$OPCODE=01h
$LENGTH=30                                        ; 30 bytes of data
$define init_result=1                             ;scw header flag byte: initialize result file
$define init_defect=2                             ;scw header flag byte: initialize defect list
$define initialize=init_result | init_defect      ;scw header flag byte: initialize results, defect list
;
; define options:
; name, # of bytes      =default value
;
password                ="SELFSCAN"               ; 8 chars
version                 ="VER 1.00"               ; 8 chars
flags,1                 =0                        ; init result block, init defect list
trace,1                 =0                        ; trace byte, scope trigger
max_time,2              =7200                     ; 8 hours (4 sec increments)
max_scw_time,2          =180                      ; 5 minutes (4 sec increments)
max_assign_defect,2     =400                      ; maximum number of assigned defects, all chains
max_hard_errs,2         =0FFFFh                   ; 0FFFFh == no limit
max_hard_head_errs,2    =0FFFFh                   ; 0FFFFh == no limit
max_soft_seek_errs,2    =0FFFFh                   ; 0FFFFh == no limit
$END


$START
$NAME=WRITE_ICL
$OPCODE=02h
$LENGTH=0
$END
```

```
$START
$NAME=ALT_SEEK, ALTERNATE_SEEK, ALT_SK, AS
$OPCODE=03h
$LENGTH=15                                       ; 15 bytes of data
;
; define options:
; name, # of bytes      =default value
;
loop_cnt,2              =1                       ;Loop count, 0 == infinite loop
max_ave_seek_time,2     =0                       ;max average seek time, 0 == infinite
max_seek_time_limit,2   =0                       ;max seek time limit, 0 == infinite
delay,2                 =1                       ;seek delay time, 0001h == no delay
id_byte,1               =0                       ;idenification byte, 0 - alternate seek
start_cyl,2             =0                       ;starting cylinder
start_head,1            =0                       ;starting head
end_cyl,2               =1                       ;ending cylinder
end_head,1              =LAST_HEAD               ;ending head
$END


$START
$NAME=SINGLE_TRACK_SEEK, SST
$OPCODE=03h
$LENGTH=15                                       ; 15 bytes of data
;
; define options:
; name, # of bytes      =default value
;
loop_cnt,2              =1                       ;Loop count, 0 == infinite loop
max_ave_seek_time,2     =0                       ;max average seek time, 0 == infinite
max_seek_time_limit,2   =0                       ;max seek time limit, 0 == infinite
delay,2                 =1                       ;seek delay time, 0001h == no delay
id_byte,1               =0x01                    ;idenification byte, 0x01 - single track seek
start_cyl,2             =0                       ;starting cylinder
start_head,1            =0                       ;starting head
end_cyl,2               =1                       ;ending cylinder
end_head,1              =LAST_HEAD               ;ending head
$END


$START
$NAME=THIRD_STROKE_SEEK_0, THIRD_SEEK_0, TSS0
$OPCODE=03h
$LENGTH=15                                       ; 15 bytes of data
;
; define options:
; name, # of bytes      =default value
;
loop_cnt,2              =1                       ;Loop count, 0 == infinite loop
max_ave_seek_time,2     =0                       ;max average seek time, 0 == infinite
max_seek_time_limit,2   =0                       ;max seek time limit, 0 == infinite
delay,2                 =1                       ;seek delay time, 0001h == no delay
id_byte,1               =0x03                    ;idenification byte, 0x03 - third stroke seek
start_cyl,2             =0                       ;starting cylinder
start_head,1            =0                       ;starting head
end_cyl,2               =MAX_CYL/3               ;ending cylinder
end_head,1              =LAST_HEAD               ;ending head
$END


$START
$NAME=THIRD_STROKE_SEEK_1, THIRD_SEEK_1, TSS1
$OPCODE=03h
$LENGTH=15                                       ; 15 bytes of data
;
; define options:
; name, # of bytes      =default value
;
loop_cnt,2              =1                       ;Loop count, 0 == infinite loop
max_ave_seek_time,2     =0                       ;max average seek time, 0 == infinite
max_seek_time_limit,2   =0                       ;max seek time limit, 0 == infinite
delay,2                 =1                       ;seek delay time, 0001h == no delay
id_byte,1               =0x03                    ;idenification byte, 0x03 - third stroke seek
start_cyl,2             =MAX_CYL/3               ;starting cylinder
start_head,1            =0                       ;starting head
end_cyl,2               =MAX_CYL/3 * 2           ;ending cylinder
end_head,1              =LAST_HEAD               ;ending head
$END
```

```
$START
$NAME=THIRD_STROKE_SEEK_2, THIRD_SEEK_2, TSS2
$OPCODE=03h
$LENGTH=15                                       ; 15 bytes of data
;
; define options:
; name, # of bytes      =default value
;
loop_cnt,2              =1                       ;Loop count, 0 == infinite loop
max_ave_seek_time,2     =0                       ;max average seek time, 0 == infinite
max_seek_time_limit,2   =0                       ;max seek time limit, 0 == infinite
delay,2                 =1                       ;seek delay time, 0001h == no delay
id_byte,1               =0x03                    ;idenification byte, 0x03 - third stroke seek
start_cyl,2             =MAX_CYL/3 * 2           ;starting cylinder
start_head,1            =0                       ;starting head
end_cyl,2               =MAX_CYL                 ;ending cylinder
end_head,1              =LAST_HEAD               ;ending head
$END


$START
$NAME=FULL_STROKE_SEEK, FSS
$OPCODE=03h
$LENGTH=15                                       ; 15 bytes of data
;
; define options:
; name, # of bytes      =default value
;
loop_cnt,2              =1                       ;Loop count, 0 == infinite loop
max_ave_seek_time,2     =0                       ;max average seek time, 0 == infinite
max_seek_time_limit,2   =0                       ;max seek time limit, 0 == infinite
delay,2                 =1                       ;seek delay time, 0001h == no delay
id_byte,1               =0xff                    ;idenification byte, 0xff - full stroke seek
start_cyl,2             =0                       ;starting cylinder
start_head,1            =0                       ;starting head
end_cyl,2               =MAX_CYL                 ;ending cylinder
end_head,1              =LAST_HEAD               ;ending head
$END


$START
$NAME=HEAD_SWITCH, HS
$OPCODE=04h
$LENGTH=10                                       ; 10 bytes of data
;
; define options:
; name, # of bytes      =default value
;
loop_cnt,2              =1                       ;Loop count, 0 == infinite loop
max_ave_sw_time,2       =0                       ;max average head switch time, 0 == infinite
max_sw_time_limit,2     =0                       ;max head switch time limit, 0 == infinite
delay,2                 =1                       ;seek delay time, 0001h == no delay
cyl,2                   =0                       ;cylinder number
$END


$START
$NAME=RANDOM_SEEK, RS
$OPCODE=05h
$LENGTH=8                                        ; 8 bytes of data
;
; define options:
; name, # of bytes      =default value
;
loop_cnt,2              =1                       ;Loop count, 0 == infinite loop
max_ave_seek_time,2     =0                       ;max average seek time, 0 == infinite
max_seek_time_limit,2   =0                       ;max seek time limit, 0 == infinite
delay,2                 =1                       ;seek delay time, 0001h == no delay
$END


$START
$NAME=FORMAT_MEDIA, FORMAT, FMT
$OPCODE=06h
$LENGTH=0                                        ; 0 bytes of data
$END


$START
$NAME=ERASE_PASSWORD
$OPCODE=07h
$LENGTH=0                                        ; 0 bytes of data
$END


$START
$NAME=TIC, GOTO, JMP, JUMP
$OPCODE=08h
$LENGTH=0                                        ; 0 bytes of data
$END
```

```
$START
$NAME=SEQUENTIAL_DEFECT, SEQUENTIAL_SCAN, SDS, DS
$OPCODE=09h
$LENGTH=48                                       ;48 bytes of data
$define enb_boost=01h                            ;margin enable flag: enable boost
$define enb_vitthres=02h                         ;margin enable flag: enable viterbi threshold
$define enb_dpdthres=04h                         ;margin enable flag: enable dpd threshold
$define enb_rty_boost=10h                        ;margin enable flag: enable retry boost
$define enb_rty_vitthres=20h                     ;margin enable flag: enable retry viterbi threshold
$define enb_rty_dpdthres=40h                     ;margin enable flag: enable retry dpd threshold
$define random=01h                               ;data flag         : enable random data
;
; define options:
; name, # of bytes      =default value
;
max_time,2              =900                     ;maximum execution time, (4 second intervals)
write_loop,1            =1                       ;write loop, number of write with data pattern
read_loop,1             =1                       ;read loop, number of reads after each write write loop
retry_read_loop,1       =8                       ;retry read loop, number of reads to verify data check
retry_write_loop,1      =4                       ;retry write/read verify loop, number of write/read loops after retry read loop
defect_threshold        =2                       ;defect threshold count, number of medium errors to assign defect
max_assign_defect,1     =0ffh                    ;maximum number of assigned defects per SCW (0ffh - no limit)
max_soft_head,2         =0ffffh                  ;maximum number of soft errors per head
max_hard_head,1         =0ffh                    ;maximum number of hard errors per head
max_soft_errors,2       =0ffffh                  ;maximum number of soft errors
max_hard_errors,2       =0ffffh                  ;maximum number of hard errors
margin_enable,1         =0                       ;margin enable flags
offtrack,2              =0                       ;offtrack
boost,1                 =0                       ;boost
vit_threshold,1         =0                       ;viterbi threshold
dpd_threshold,1         =0                       ;dpd threshold
retry_offtrack,2        =0                       ;retry offtrack
retry_boost,1           =0                       ;retry boost
retry_vit_thres,1       =0                       ;retry viterbi threshold
retry_dpd_thres,1       =0                       ;retry dpd threshold
start_cyl, 2            =0000h                   ;starting cylinder
end_cyl, 2              =0ffffh                  ;ending cylinder, 0ffffh : last user cylinder
data_flag,1             =0                       ;data flag
data_length,1           =16                      ;data length
data                    =PATTERN1                ;data, 16 bytes of 66h
$END


$START
$NAME=STOP_START, SS
$OPCODE=0ah
$LENGTH=8                                        ; 8 bytes of data
;
; define options:
; name, # of bytes      =default value
;
loop_cnt,2              =1                       ;Loop count, 0 == infinite loop
max_time,2              =180                     ;maximum scw execution time ( 4 second resolution)
stop_time,2             =4                       ;stop time (4 second resolution)
max_start_time,2        =5                       ;maximum start time (4 second resolution)
$END


$START
$NAME=DIGITAL_DEFECT, DIGITAL_SCAN, DDS
$OPCODE=0bh
$LENGTH=48                                       ;48 bytes of data
$define enb_boost=01h                            ;margin enable flag: enable boost
$define enb_vitthres=02h                         ;margin enable flag: enable viterbi threshold
$define enb_dpdthres=04h                         ;margin enable flag: enable dpd threshold
$define enb_rty_boost=10h                        ;margin enable flag: enable retry boost
$define enb_rty_vitthres=20h                     ;margin enable flag: enable retry viterbi threshold
$define enb_rty_dpdthres=40h                     ;margin enable flag: enable retry dpd threshold
$define random=01h                               ;data flag         : enable random data
$define wr_sine=02h                              ;data flag         : enable write sine
```

```
;
; define options:
; name, # of bytes      =default value
;
max_time,2              =900                    ;maximum execution time, (4 second intervals)
write_loop,1            =1                      ;write loop, number of write with data pattern
read_loop,1             =1                      ;read loop, number of reads after each write write loop
retry_read_loop,1       =8                      ;retry read loop, number of reads to verify data check
retry_write_loop,1      =4                      ;retry write/read verify loop, number of write/read loops after retry read loop
defect_threshold        =2                      ;defect threshold count, number of medium errors to assign defect
max_assign_defect,1     =0ffh                   ;maximum number of assigned defects per SCW (0ffh - no limit)
max_soft_head,2         =0ffffh                 ;maximum number of soft errors per head
max_hard_head,1         =0ffh                   ;maximum number of hard errors per head
max_soft_errors,2       =0ffffh                 ;maximum number of soft errors
max_hard_errors,2       =0ffffh                 ;maximum number of hard errors
margin_enable,1         =0                      ;margin enable flags
offtrack,2              =0                      ;offtrack
boost,1                 =0                      ;boost
vit_threshold,1         =0                      ;viterbi threshold
dpd_threshold,1         =0                      ;dpd threshold
retry_offtrack,2        =0                      ;retry offtrack
retry_boost,1           =0                      ;retry boost
retry_vit_thres,1       =0                      ;retry viterbi threshold
retry_dpd_thres,1       =0                      ;retry dpd threshold
start_cyl, 2            =0000h                  ;starting cylinder
end_cyl, 2              =0ffffh                 ;ending cylinder, 0ffffh : last user cylinder
data_flag,1             =wr_sine                ;data flag
data_length,1           =16                     ;data length
data                    =PATTERN3               ;data, 16 bytes of ffh
$END


$START
$NAME=FIR_TRAINING, FIR
$OPCODE=0Ch
$LENGTH=2                                       ; 2 bytes of data
;
; define options:
; name, # of bytes      =default value
;
max_sum, 2              =26h                    ;maximum allowed fir sum
$END


$START
$NAME=SET_AGC_MODE, SET_AGC, AGC
$OPCODE=0Eh
$LENGTH=0                                       ; 0 bytes of data
;
; define options:
; name, # of bytes      =default value
;
$END
```

Appendix G

SSW.Def

```
;
; SELFSCAN STATUS WORD (SSW) DEFINITIONS FILE
;
; This file contains the Selfscan Status Word (SSW) definitions for the SSW results disassembler.
;
;
; SSW COMMAND RESULT TAIL OPERATION CODE:
;
$CMD_TAIL=0FH
;
; MODEL DEFINES
;
$INCLUDE MODEL.DEF
;
; DEFINE HEADERS & FORMATS FOR ALL SSW's:
;
$LINES_PER_PAGE=0
$PAGE_HEADER ">>>                    QUANTUM CORPORATION SELFSCAN RESULTS                    <<<"
$PAGE_HEADER ">>>                                                                            <<<"
$PAGE_HEADER "\n\n\n\n"
$HEADER 1 "\n\n"
$HEADER 1 "SCW_ADDR SCW_COMMAND  FLAG STAT\n"
$HEADER 1 "-------- -----------  ---- ----\n"
$FORMAT 1 "%04xh %s %02xh %02xh\n"
$HEADER 2 "\n\n"
$HEADER 2 "SCW_ADDR SCW_COMMAND  FLAG STAT SCW_ERR ERR_IDX SCW_TIME(sec)\n"
$HEADER 2 "-------- -----------  ---- ---- ------- ------- -------------\n"
$FORMAT 2 "%04xh %s %02xh %02xh %02xh %02xh %4u\n"
$HEADER 3 "\n\n"
$HEADER 3 "SCW_ADDR SCW_COMMAND  FLAG STAT SCW_ERR ERR_IDX SCW_TIME MAX_SEEK MIN_SEEK AVR_SEEK TOTAL_SEEK SEEK_ERR START_CYL/HD END_CYL/HD\n
$HEADER 3 "                                                (sec)    (us)     (us)     (us)                                               \n
$HEADER 3 "-------- -----------  ---- ---- ------- ------- -------- -------- -------- -------- ---------- -------- -------- --- ------ ---\n
$FORMAT 3 "%04xh %s %02xh %02xh %02xh %02xh %4u %5u %5u %5u %5u %3u %04d %d %04d %d\n"
$HEADER 4 "\n\n"
$HEADER 4 "SCW_ADDW SCW_COMMAND  FLAG STAT SCW_ERR ERR_IDX SCW_TIME(sec) LAST_CYL LAST_HD ASGN_DEF SOFT_ERR HARD_ERR\n"
$HEADER 4 "-------- -----------  ---- ---- ------- ------- ------------- -------- ------- -------- -------- --------\n"
$FORMAT 4 "%04xh %s %02xh %02xh %02xh %02xh %4u %04d %d %3d %3d %3d\n"
$HEADER 5 "\n\n"
$HEADER 5 "SCW_ADDR SCW_COMMAND  FLAG STAT TERM_ERR SCW_ERR ERR_IDX SOFT_ERR HARD_ERR\n"
$HEADER 5 "-------- -----------  ---- ---- -------- ------- ------- -------- --------\n"
$FORMAT 5 "%04xh %s %02xh %02xh %02xh %02xh %02xh %3d %3d\n"
$HEADER 6 "\n\n"
$HEADER 6 "SCW_ADDW SCW_COMMAND  FLAG STAT SCW_ERR ERR_IDX SCW_TIME(sec) LAST_CYL LAST_HD LAST_WEG\n"
$HEADER 6 "-------- -----------  ---- ---- ------- ------- ------------- -------- ------- --------\n"
$FORMAT 6 "%04xh %s %02xh %02xh %02xh %02xh %4u %04d %d %d\n"

;
; SELFSCAN STATUS WORD DEFINITIONS:
;
$START
$NAME=RESULT_HEADER
$OPCODE=0xFF
$LENGTH=16                  ; # bytes of data
PASSWORD,8                  ; result file password
INTERNAL,8
;
; define header & format # to use
;
$print "NEXT GENERATION SELFSCAN RESULTS FILE\n\n"
$print "%s\n", $name
$END


$START
$NAME=SCW_HEADER
$OPCODE=01h
$LENGTH=26                  ; # bytes of data
```

```
;
; define output: name, # of bytes
;
ERROR_CODE, 1                ; scw error code
ERROR_CODE_INDEX, 1          ; error code index
ELAPSED_TIME, 2              ; scw elapsed time
VERSION, 8                   ; ascii version number of test
FLAGS, 1                     ; scw header flags
TRACE,1                      ; trace byte, scope trigger
MAX_TIME, 2                  ; max time for chain (4 sec)
MAX_SCW_TIME, 2              ; default max scw time (4 sec)
MAX_ASSIGN_DEF, 2            ; maximum number of assigned defects
MAX_HARD_ERRORS, 2           ; max # of hard errors, all chains
MAX_HARD_ERRORS_HD, 2        ; max # of hard errors per head, all chains
MAX_RECOVERED_SEEK_ERRS, 2   ; max # of recovered seek errors, all chains
;
; define header & format # to use
;
$print "\n>>> Start of Chain or Power Failure >>>\n"
$display 2, $scw_addr, $name, $flag, $status, error_code, error_code_index, elapsed_time * 4
$print "Version: %s,   Header Flags: %02x\n", version, flags
$END


$START
$NAME=WRITE_ICL
$OPCODE=02h
$LENGTH=4
;
; define output: name, # of bytes
;
ERROR_CODE, 1                  ; scw error code
ERROR_CODE_INDEX, 1            ; error code index
ELAPSED_TIME, 2                ; scw elasped time
;
; define header & format # to use
;
$display 4, $scw_addr, $name, $flag, $status, error_code, error_code_index, elapsed_time * 4
$END


$START
$NAME=ALT_SEEK
$OPCODE=03h
$LENGTH=21                   ; # bytes of data
;
; define output: name, # of bytes
;
ERROR_CODE, 1                ; scw error code
ERROR_CODE_INDEX, 1          ; error code index
ELAPSED_TIME, 2              ; scw elapsed time
ID_BYTE,1                    ; identification byte
START_CYL, 2                 ; starting cylinder
START_HD, 1                  ; starting head
END_CYL, 2                   ; ending cylinder
END_HD, 1                    ; ending head
MAX_SEEK_TIME, 2             ; max seek time (2 us resolution)
MIN_SEEK_TIME, 2             ; min seek time
AVG_SEEK_TIME, 2             ; average seek time
TOTAL_SEEKS, 2               ; total number of seeks
TOTAL_SEEK_ERRORS, 2         ; total number of seek errors
;
; define header & format # to use
;
$set test="ALT_SEEK"
$if id_byte=0x01 then $set test="SINGLE_SK"
$if id_byte=0x03 then $set test="THIRD_SK"
$if id_byte=0xff then $set test="FULL_SEEK"
$DISPLAY 3, $scw_addr, test, $flag, $status, error_code, error_code_index, elapsed_time * 4, max_seek_time * 2 , min_seek_time * 2, \
            avg_seek_time * 2, total_seeks, total_seek_errors, start_cyl, start_hd, end_cyl, end_hd
$END


$START
$NAME=HEAD_SWITCH
$OPCODE=04h
$LENGTH=16                   ; # bytes of data
;
; define output: name, # of bytes
;
ERROR_CODE, 1                ; scw error code
ERROR_CODE_INDEX, 1          ; error code index
ELAPSED_TIME, 2              ; scw elapsed time
CYL, 2                       ; cylinder number
MAX_HD_TIME, 2               ; max head switch time (2 us resolution)
MIN_HD_TIME, 2               ; min head switch time
AVG_HD_TIME, 2               ; average head switch time
TOTAL_SWITCHES, 2            ; total number of head switches
TOTAL_SW_ERRORS, 2           ; total number of head switch errors
```

```
;
; define header & format # to use
;
$display 3, $scw_addr, $name, $flag, $status, error_code, error_code_index, elapsed_time * 4, max_hd_time * 2, min_hd_time * 2, \
            avg_hd_time * 2, total_switches, total_sw_errors, cyl
$END


$START
$NAME=RANDOM_SEEK
$OPCODE=05h
$LENGTH=14                    ; # bytes of data
;
; define output: name, # of bytes
;
ERROR_CODE, 1                 ; scw error code
ERROR_CODE_INDEX, 1           ; error code index
ELAPSED_TIME, 2               ; scw elapsed time
MAX_SEEK_TIME, 2              ; max seek time (2 us resolution)
MIN_SEEK_TIME, 2              ; min seek time
AVG_SEEK_TIME, 2              ; average seek time
TOTAL_SEEKS,2                 ; total number of seeks
TOTAL_SEEK_ERRORS,2           ; total number of seek errors
;
; define header & format # to use
;
$display 3, $scw_addr, $name, $flag, $status, error_code, error_code_index, elapsed_time * 4, max_seek_time * 2, min_seek_time * 2, \
            avg_seek_time * 2, total_seeks, total_seek_errors
$END


$START
$NAME=FORMAT_MEDIA
$OPCODE=06h
$LENGTH=7                     ; # bytes of data
;
; define output: name, # of bytes
;
ERROR_CODE, 1                 ; scw error code
ERROR_CODE_INDEX, 1           ; error code index
ELAPSED_TIME, 2               ; scw elapsed time
CYLINDER, 2                   ; ending cylinder
HEAD, 1                       ; ending head
;
; define header & format # to use
;
$display 2, $scw_addr, $name, $flag, $status, error_code, error_code_index, elapsed_time * 4
$END


$START
$NAME=ERS_PASSWRD
$OPCODE=07h
$LENGTH=4                     ; # bytes of data
;
; define output: name, # of bytes
;
ERROR_CODE, 1                 ; scw error code
ERROR_CODE_INDEX, 1           ; error code index
ELAPSED_TIME, 2               ; scw elapsed time
;
; define header & format # to use
;
$display 2, $scw_addr, $name, $flag, $status, error_code, error_code_index, elapsed_time * 4
$END


$START
$NAME=TIC
$OPCODE=08h
$LENGTH=0                     ; # bytes of data
;
; define header & format # to use
;
$display 1, $scw_addr, $name, $flag, $status
$END


$START
$NAME=SEQ_DEFECT
$OPCODE=09h
$LENGTH=LAST_HEAD + 1 * 3 + 15     ; # bytes of data
```

```
;
; define output: name, # of bytes
;
ERROR_CODE, 1                ; scw error code
ERROR_CODE_INDEX, 1          ; error code index
ELAPSED_TIME, 2              ; scw elapsed time
LAST_CYL, 2                  ; last cylinder number
LAST_HEAD, 1                 ; last head number
SEEK_RECOVERED, 2            ; total seek recovered errors
ASSIGN_DEFECT,2              ; total assigned defect
SOFT_ERRORS, 2               ; total soft errors
HARD_ERRORS, 2               ; total hard errors
SOFT_HD_ERRORS, LAST_HEAD + 1 * 2 ; total soft errors per head
HARD_HD_ERRORS, LAST_HEAD + 1     ; total hard errors per head
;
; define header & format # to use
;
$display 4, $scw_addr, $name, $flag, $status, error_code, error_code_index, elapsed_time * 4, last_cyl, last_head, assign_defect,\
            soft_errors, hard_errors
$END


$START
$NAME=STOP_START
$OPCODE=0Ah
$LENGTH=4                    ; # bytes of data
;
; define output: name, # of bytes
;
ERROR_CODE, 1                ; scw error code
ERROR_CODE_INDEX, 1          ; error code index
ELAPSED_TIME, 2              ; scw elapsed time
;
; define header & format # to use
;
$display 2, $scw_addr, $name, $flag, $status, error_code, error_code_index, elapsed_time * 4
$END


$START
$NAME=DIG_DEFECT
$OPCODE=0bh
$LENGTH=LAST_HEAD + 1 * 3 + 15    ; # bytes of data
;
; define output: name, # of bytes
;
ERROR_CODE, 1                ; scw error code
ERROR_CODE_INDEX, 1          ; error code index
ELAPSED_TIME, 2              ; scw elapsed time
LAST_CYL, 2                  ; last cylinder number
LAST_HEAD, 1                 ; last head number
SEEK_RECOVERED, 2            ; total seek recovered errors
ASSIGN_DEFECT,2              ; total assigned defect
SOFT_ERRORS, 2               ; total soft errors
HARD_ERRORS, 2               ; total hard errors
SOFT_HD_ERRORS, LAST_HEAD + 1 * 2 ; total soft errors per head
HARD_HD_ERRORS, LAST_HEAD + 1     ; total hard errors per head
;
; define header & format # to use
;
$display 4, $scw_addr, $name, $flag, $status, error_code, error_code_index, elapsed_time * 4, last_cyl, last_head, assign_defect, \
            soft_errors, hard_errors
$END


$START
$NAME=FIR_TRAIN
$OPCODE=0ch
$LENGTH=8                    ; # bytes of data
;
; define output: name, # of bytes
;
ERROR_CODE, 1                ; scw error code
ERROR_CODE_INDEX, 1          ; error code index
ELAPSED_TIME, 2              ; scw elapsed time
LAST_CYL, 2                  ; last cylinder number
LAST_HEAD, 1                 ; last head number
LAST_WEDGE, 1                ; last wedge number
;
; define header & format # to use
;
$display 6, $scw_addr, $name, $flag, $status, error_code, error_code_index, elapsed_time * 4, last_cyl, last_head, last_wedge
$END


$START
$NAME=SET_AGC
$OPCODE=0Eh
$LENGTH=4                    ; # bytes of data
```

```
;
; define output: name, # of bytes
;
ERROR_CODE, 1                 ; scw error code
ERROR_CODE_INDEX, 1           ; error code index
ELAPSED_TIME, 2               ; scw elapsed time
;
; define header & format # to use
;
$display 2, $scw_addr, $name, $flag, $status, error_code, error_code_index, elapsed_time * 4
$END


$START
$NAME=RESULT_TAIL
$OPCODE=0fh
$LENGTH=104                   ; # bytes of data
;
; define output: name, # of bytes
;
TERM_ERROR, 1                 ; termination error code
LED_ERROR,1                   ; led error code
ERROR_CODE, 1                 ; scw error code
ERROR_CODE_INDEX, 1           ; error code index
SCW_THIS_CHAIN, 2             ; number of scw this chain
SCW_ALL_CHAINS, 2             ; total number of scw, all chains
TIME_THIS_CHAIN, 2            ; total time this chain (4 second resolution)
TIME_ALL_CHAINS, 2            ; total time all chains (4 second resolution)
SOFT_ERRORS, 2                ; total soft errors all chains
HARD_ERRORS, 2                ; total hard errors all chains
ALL_SEEK_ERRORS, 2            ; total seek errors all chains
SOFT_HD_ERRORS, 28, word[]    ; total soft errors per head
HARD_HD_ERRORS, 28, word[]    ; total hard errors per head
WIGGLE_ERRORS, 28, word[]     ; total wiggle errors per head
SPACE,2                       ; reserved space
;
; define header & format # to use
;
$display 5, $scw_addr, $name, $flag, $status, term_error, error_code, error_code_index, soft_errors, hard_errors
$print "\n\nTotal Chain Result:\n\n"
$print "Total SCW's this chain       : %4d       ,  Total SCW's all chains         : %4d\n", scw_this_chain, scw_all_chains
$print "Total execute time this chain: %4d (sec),  Total execution time all chains: %4d (sec)\n", time_this_chain * 4, time_all_chains * 4
$print "Total seek errors all chains : %4d\n", all_seek_errors
;
; DECODE ERROR CODES:
;
$set rc = 0
$if term_error = 00h $print "\n\nNo Errors                              \n\n" : $set rc = 0
$if term_error = 10h $print "\n\nTermination Error: Result File Full   \n" : $set rc = 1
$if term_error = 11h $print "\n\nTermination Error: Invalid Status     \n" : $set rc = 1
$if term_error = 12h $print "\n\nTermination Error: Chain Timeout\n    \n" : $set rc = 1
$if term_error = 14h $print "\n\nTermination Error: Command Check\n    \n" : $set rc = 1
$if term_error = 15h $print "\n\nTermination Error: Program Check\n    \n" : $set rc = 1
$if term_error = 16h $print "\n\nTermination Error: Max Assign Defects \n" : $set rc = 1
$if term_error = 17h $print "\n\nTermination Error: Failed Spinup      \n" : $set rc = 1
$if $flag&40h  = 40h $print "\n\nSCW CHAIN ERROR: SCW Chain is Incomplete, Command Chain Continue in Flag byte\n\n" \
    : $set rc = 1
$if error_code = 1fh $print "SCW Error: SCW Terminated by Command Processor\n\n"
$if error_code = 20h $print "SCW Error: More than one SCW Header in Chain\n\n"
$if error_code = 21h $print "SCW Error: No SCW Header in Chain\n\n"
$if error_code = 22h $print "SCW Error: Invalid SCW Command\n\n"
$if error_code = 23h $print "SCW Error: Invalid Password\n\n"
$if error_code = 24h $print "SCW Error: Failed Initialization Results File\n\n"
$if error_code = 25h $print "SCW Error: Failed Initialization Defect File\n\n"
$if error_code = 26h $print "SCW Error: Invalid Defect List\n\n"
$if error_code = 27h $print "SCW Error: Invalid Result File\n\n"
$if error_code = 28h $print "SCW Error: Chain Timeout\n\n"
$if error_code = 29h $print "SCW Error: SCW Timeout\n\n"
$if error_code = 2ah $print "SCW Error: Maximum Overall Hard Errors\n\n"
$if error_code = 2bh $print "SCW Error: Maximum Overall Seek Errors\n\n"
$if error_code = 2Ch $print "SCW Error: Failed Reading File\n\n"
$if error_code = 2dh $print "SCW Error: Failed Writing File\n\n"
$if error_code = 2eh $print "SCW Error: No TIC ICL in Chain\n\n"
$if error_code = 2fh $print "SCW Error: Fatal Error From Idle Call Vector\n\n"
$if error_code = 30h $print "SCW Error: Average Seek Timeout\n\n"
$if error_code = 31h $print "SCW Error: Maximum Seek Timeout\n\n"
$if error_code = 32h $print "SCW Error: Maximum Number of Seeks Exceeded\n\n"
$if error_code = 33h $print "SCW Error: Recal Failure\n\n"
$if error_code = 34h $print "SCW Error: Format Track Failure\n\n"
$if error_code = 35h $print "SCW Error: Failed to Clear Defect List\n\n"
$if error_code = 37h $print "SCW Error: Maximum Hard Errors this SCW\n\n"
$if error_code = 38h $print "SCW Error: Maximum Hard Errors per Head\n\n"
$if error_code = 39h $print "SCW Error: Maximum Soft Errors this SCW\n\n"
$if error_code = 3ah $print "SCW Error: Maximum Soft Errors per Head\n\n"
$if error_code = 3bh $print "SCW Error: Track Scan Failure\n\n"
$if error_code = 3ch $print "SCW Error: Medium Error Failure\n\n"
$if error_code = 3dh $print "SCW Error: Failed Writing Defect List\n\n"
$if error_code = 3eh $print "SCW Error: Defect List Full\n\n"
$if error_code = 40h $print "SCW Error: Defect Assigned\n\n"
$if error_code = 41h $print "SCW Error: Maximum FIR sum exceeded\n\n"
$if error_code = 42h $print "SCW Error: No FIR coefficients loaded\n\n"
$if error_code = 43h $print "SCW Error: Trained FIR coefficients loaded\n\n"
$if error_code = 44h $print "SCW Error: No FIR wedge found for training\n\n"
```

```
$if error_code = 45h $print "SCW Error: Failed FIR training\n\n"
$if error_code = 4eh $print "SCW Error: Maximum Assigned Defects per SCW\n\n"
$if error_code = 4fh $print "SCW Error: Maximum Assigned Defects\n\n"
$if error_code = 51h $print "SCW Error: Failed Spinup\n\n"
$if error_code = 52h $print "SCW Error: Start Spinup Timeout\n\n"
;
; PRINT ERROR STATISTICS:
;
$print "\nError Statistics Per Head:\n\n"
$print "                0        1        2        3"
$if last_head > 3  then $print "        4        5"
$if last_head > 5  then $print "        6        7"
$if last_head > 7  then $print "        8        9"
$if last_head > 9  then $print "       10       11"
$if last_head > 11 then $print "       12       13"
$print "\n"
$print "--------------------------------------------"
$if last_head > 3  then $print "------------------"
$if last_head > 5  then $print "------------------"
$if last_head > 7  then $print "------------------"
$if last_head > 9  then $print "------------------"
$if last_head > 11 then $print "------------------"
$print "\n"
$print "Soft Errors: %4d     %4d     %4d     %4d", soft_hd_errors[0], soft_hd_errors[1], soft_hd_errors[2], soft_hd_errors[3]
$if last_head > 3  then $print "     %4d     %4d", soft_hd_errors[4], soft_hd_errors[5]
$if last_head > 5  then $print "     %4d     %4d", soft_hd_errors[6], soft_hd_errors[7]
$if last_head > 7  then $print "     %4d     %4d", soft_hd_errors[8], soft_hd_errors[9]
$if last_head > 9  then $print "     %4d     %4d", soft_hd_errors[10], soft_hd_errors[11]
$if last_head > 11 then $print "     %4d     %4d", soft_hd_errors[12], soft_hd_errors[13]
$print "\n"
$print "Hard Errors: %4d     %4d     %4d     %4d", hard_hd_errors[0], hard_hd_errors[1], hard_hd_errors[2], hard_hd_errors[3]
$if last_head > 3  then $print "     %4d     %4d", hard_hd_errors[4], hard_hd_errors[5]
$if last_head > 5  then $print "     %4d     %4d", hard_hd_errors[6], hard_hd_errors[7]
$if last_head > 7  then $print "     %4d     %4d", hard_hd_errors[8], hard_hd_errors[9]
$if last_head > 9  then $print "     %4d     %4d", hard_hd_errors[10], hard_hd_errors[11]
$if last_head > 11 then $print "     %4d     %4d", hard_hd_errors[12], hard_hd_errors[13]
$print "\n"
$print "Wiggle Errs: %4d     %4d     %4d     %4d", wiggle_errors[0], wiggle_errors[1], wiggle_errors[2], wiggle_errors[3]
$if last_head > 3  then $print "     %4d     %4d", wiggle_errors[4], wiggle_errors[5]
$if last_head > 5  then $print "     %4d     %4d", wiggle_errors[6], wiggle_errors[7]
$if last_head > 7  then $print "     %4d     %4d", wiggle_errors[8], wiggle_errors[9]
$if last_head > 9  then $print "     %4d     %4d", wiggle_errors[10], wiggle_errors[11]
$if last_head > 11 then $print "     %4d     %4d", wiggle_errors[12], wiggle_errors[13]
$print "\n"
$return rc
$end
```

Index

A
all_seek_errors, 6
ALT_SEEK, 18
ALT_SK, 18
ALTERNATE_SEEK, 18
AS, 18
assign_defect, 65, 79
ave_hd_time, 43
ave_seek_time, 21, 26, 32, 38, 48
B
boost, 62, 76
boot strap, 7
C
chain results, 4
command file, 2
command history, 3, 7
command processor, 1, 2
configuration pages, 2
cyl, 42, 43
cylinder, 51
D
data, 64, 78
data_flag, 63, 77
data_length, 63, 77
DDS, 72
defect count, 3
defect file, 2
delay, 19, 24, 30, 36, 42, 46
DIGITAL_DEFECT, 72
DIGITAL_SCAN, 72
dpd_threshold, 63, 77
DS, 58
E
elapsed_time, 12, 16, 21, 26, 32, 38, 43, 47, 51, 53, 65, 68, 79, 84
enb_boost, 62, 76
enb_dpdthres, 62, 76
enb_rty_boost, 62, 76
enb_rty_dpdthres, 62, 76
enb_rty_vitthres, 62, 76
enb_vitthres, 62, 76
end_cyl, 19, 21, 24, 26, 30, 32, 36, 38, 63, 77
end_head, 20, 21, 25, 26, 31, 32, 37, 38
ending status, 3

ERASE_PASSWORD, 52
Error Code, 4
error code, 3
Error Code Index, 4, 89, 92, 97
error code index, 4
error_code, 5, 12, 15, 20, 25, 31, 37, 43, 47, 50, 53, 64, 68, 78, 84
error_code_index, 5, 12, 16, 20, 25, 31, 37, 43, 47, 51, 53, 64, 68, 78, 84
F
FIR, 82
FIR table, 82
FIR_TRAINING, 82
flags, 10, 12
FMT, 50
FORMAT, 50
Format Media, 58, 72
FORMAT_MEDIA, 50
FSS, 35
FULL_STROKE_SEEK, 35
G
GOTO, 55
H
hard error, 57, 71
hard errors, 4
hard_errors, 6, 65, 79
hard_hd_errors, 6, 65, 79
head, 51
HEAD_SWITCH, 41
HEADER, 9
HS, 41
I
id_byte, 19, 21, 24, 26, 30, 32, 36, 38
J
JMP, 55
L
last_cyl, 65, 79, 84
last_head, 65, 79, 84
last_wedge, 84
LED, 5
LED Error Code, 4
LED error code, 4
LED Error Codes, 86
led_error, 5
loop_cnt, 19, 24, 30, 36, 41, 46, 67
M
margin values, 56

margin_enable, 62, 76
max sum, 83
max time, 59, 73
max_assign_defect, 10, 12, 75
max_ave_seek_time, 19, 24, 30, 36, 46
max_ave_sw_time, 41
max_hard_errors, 61, 75
max_hard_errs, 10, 12
max_hard_head, 61, 75
max_hard_head_errs, 10, 13
max_hd_time, 43
max_scw_time, 10, 12
max_seek_time, 21, 26, 32, 38, 48
max_seek_time_limit, 19, 24, 30, 36, 46
max_soft_errors, 61, 75
max_soft_head, 61, 75
max_soft_seek_errs, 11, 13
max_start_time, 68
max_sw_time_limit, 42
max_time, 10, 12, 67
medium error, 70
min_hd_time, 43
min_seek_time, 21, 26, 32, 38, 48
MODE PAGE 1, 56, 70, 81
mode select, 2
N
Number of SCWs all chains, 4
Number of SCWs this chain, 4
O
offtrack, 62, 70, 76
P
password, 7, 9
PRML, 81
R
random, 63, 77
RANDOM_SEEK, 46
read_loop, 60, 74
recovered seek, 6, 57, 70
result file, 2
results tail, 4
retry_boost, 63, 77
retry_count, 83
retry_dpd_thres, 63, 77
retry_offtrack, 63, 77
retry_read_loop, 60, 74

retry_vit_thres, 63, 77
retry_write_loop, 60, 74
RS, 46
S
SCW Command Chain File, 2
SCW Error Codes, 87
SCW Header, 7
SCW Header Command, 2
scw_all_chains, 5
SCW_HEADER, 9
scw_this_chain, 5
SDS, 58
sector defect, 3
seek errors, 4
seek_recovered, 65, 79
Selfscan test cylinder, 2
SEQUENTIAL_DEFECT, 58
SEQUENTIAL_SCAN, 58
SINGLE_TRACK_SEEK, 23
soft error, 57, 71
soft errors, 4
soft_errors, 5, 65, 79
soft_hd_errors, 6, 65, 79
SS, 67
SST, 23
SSW, 3
SSW Result file, 3
SSW tail, 4
start_cyl, 19, 21, 24, 26, 30, 32, 36, 38, 63, 77
start_head, 19, 21, 24, 26, 30, 32, 36, 38
STOP_START, 67
stop_time, 68
T
tap_count, 83
term_error, 5
Termination Error Code, 4
termination error code, 4
Termination Error Codes, 86
THIRD_SEEK_0, 29
THIRD_SEEK_1, 29
THIRD_SEEK_2, 29
THIRD_STROKE_SEEK_0, 29
THIRD_STROKE_SEEK_1, 29
THIRD_STROKE_SEEK_2, 29
TIC, 55

TIC ICL, 14
time_all_chains, 5
time_this_chain, 5
Total execution time all chains, 4
Total execution time this chain, 4
Total hard errors all chains, 4
Total hard errors per head, 4
Total seek errors all chains, 4
Total soft errors all chains, 4
Total soft errors per head, 4
Total wiggle errors per head, 4
total_seek_errors, 21, 26, 32, 38, 48
total_seeks, 21, 26, 32, 38, 48
total_sw_errors, 43
total_switches, 43
trace, 10, 12
trace address, 3
training pattern, 82
TSS0, 29
TSS1, 29
TSS2, 29
V
version, 9, 12
vit_threshold, 62, 76
W
wedge defect, 3, 70
wedge sectors, 70
wedge skews, 3
wiggle errors, 4
wiggle recovery, 56, 70
wiggle_errors, 6
wr_sine, 77
WRITE_ICL, 15
write_loop, 59, 73