Rimfire 3570

SCSI Host Bus Adapter

User's Guide

CIPRICO

Rimfire 3570

SCSI Host Bus Adapter

User's Guide

Proprietary

Publication Number: 21010331 A
December 1992

2800 Campus Drive
Plymouth, MN 55441
(612) 551-4000

2800 Campus Drive
Plymouth, MN 55441
(612) 551-4000

The information in this manual is subject to change, in part or whole, without notice or obligation.

Printed in the United States of America

Warranty

Hardware

CIPRICO, for the effective period set out below, warrants that its standard products sold hereunder will be free from defects in workmanship or material under normal use and service. CIPRICO's obligation under this warranty shall not arise until Buyer returns the defective product, freight prepaid, to CIPRICO. The only responsibilities of CIPRICO under this warranty are at its option to replace or repair, without charge, any defective component of such products.

Software and Firmware

CIPRICO warrants that Software and Firmware supplied hereunder shall conform to the then current published documentation applicable to such programs. CIPRICO, for the effective period of the warranty set out below, will upon written notice from the Buyer documenting the symptoms or the defect, expend its best efforts to resolve software bugs and/or faults. This service shall be without extra charge, and at CIPRICO's option may include on-site visit(s) if in its opinion the conditions justify such visit(s).

Effective Period of Warranty

One year from date of delivery.

Repaired or Replacement Product (Out of Warranty)

CIPRICO, for a period of 90 days, warrants that its standard products repaired or replaced hereunder shall be free from defects in workmanship or material under normal use and service. CIPRICO's obligations under this warranty shall not arise until Buyer returns the defective product, freight prepaid, to CIPRICO. The responsibilities of CIPRICO under this warranty are at its option to repair or replace, without charge, any defective component part of such products.

Any repair or replacement shall not extend the period within which the warranty can be asserted.

This warranty does not extend to and shall not apply to:

1. Products which have been repaired or altered by other than CIPRICO, unless Buyer has properly altered or repaired the products in accordance with procedures previously approved, in writing, by CIPRICO; or

2. Products which have been subject to misuse, neglect, accident or improper installation; or

3. Products not manufactured by CIPRICO.

The foregoing warranty and remedies are exclusive and are made expressly in lieu of all other warranties expressed or implied, either in fact or by operation of law, statutory or otherwise, including warranties of merchantability and fitness for use. CIPRICO neither assumes nor authorizes any other person to assume for it any other liability in connection with the sale, installation or use of its products, and CIPRICO makes no warranty whatsoever for products not manufactured by CIPRICO or with respect to any non-standard products which have been subjected to misuse, neglect, accident, or have been modified by the buyer. CIPRICO shall have no liability for incidental or consequential damages of any kind arising out of the sale, installation, or use of its products.

SERVICE POLICY

A. "Out of warranty" repairs will be accomplished expeditiously at published rates plus shipping charges. Contact your appropriate Domestic or International Sales Administrator for current pricing.

B. All products "in" or "out of warranty" returned for repair that are found to have no defect will be charged at the "out of warranty" rate.

C. Shipments arriving at CIPRICO without a Return Material Authorization (RMA) number will not be accepted and will be returned to the customer, freight collect, regardless of warranty status.

RETURN PROCEDURES

Upon determining that repair is required the Customer must:

DOMESTIC CUSTOMERS

Call CIPRICO Customer Support at (612) 559-2034 for an RMA number.

INTERNATIONAL CUSTOMERS:

Fax requests to the attention of Int'l Sales Administrator at 612-559-8799 for an RMA number .

Please have the following information available:

a) The serial number of the board(s).

b) An open P.O. number with appropriate repair charge. Requirement applicable also for "no charge" warranty repairs.

c) The reason for the return, or repairs needed.

d) State if Debit/Rebill is applicable.

SHIPMENT INSTRUCTIONS

DOMESTIC CUSTOMER

Ship prepaid to:

CIPRICO INC.
2800 Campus Drive
Suite 110
Plymouth, MN 55441

INTERNATIONAL CUSTOMERS

Ship prepaid to:

MINNEAPOLIS-ST. PAUL
INTERNATIONAL AIRPORT U.S.A.

Consigned to:

CIPRICO INC.
c/o William Griffin & Company
7830 12th Ave. So.
Minneapolis, MN 55425

Include with shipment:

a) 1 ORIGINAL commercial invoice + 3 copies

b) 1 ORIGINAL packing list + 1 copy (list S/Ns)

Commercial Invoice must show: **NOTE**

- *Complete name and address of the manufacturer.*
- *Complete name and address of the reseller.*
- *Detailed, useful and adequate description of the merchandise including:*

  Model# and name by which each item is known.
  Box markings.
  Quantities in appropriate weights & measure.
  Value of each item in the currency of purchase.
  All additional charges outside of F.O.B. value.
  Country of origin.
  Appropriate 8 digit HTS (tariff) number.

RMA NUMBERS

The RMA number is valid for 30 days after issue.

INVOICING

Upon completion of the services required, an invoice will be issued stating charges (when applicable) and work completed.

RETURN SHIPMENTS

Ciprico will prepay International return shipments (air freight consolidation) if the item was under warranty. All other shipments will be made freight collect.

Preface

Introduction

This manual is intended to instruct you on how to install and use the Rimfire® 3570 (RF3570) adapter. The adapter is essentially compatible with the Ciprico® RF3500 series of adapters. Therefore, at its simplest, this manual will instruct you how to install an RF3570 adapter to replace an existing RF3500 series adapter. Depending on your prior RF3500 implementation, this may require no change to your existing installed driver.

Since there are more features added to the RF3570, this manual also provides the information necessary to change or write a new board driver to take advantage of the advanced operation possible with this adapter.

Organization

There are two levels to the information provided in this manual:

- Reference
- Practical Usage

Chapters 3, 5, and 6 along with Appendices A and B are beneficial when you wish to find specific information about board jumpers, commands, error codes, or installation. They are meant to be used for reference.

Other chapters — 2, 4 and 7 plus parts of 3 and 5 — are effective in a different manner. These chapters explain processes or features of the board in terms of examples. It is to these chapters that you should turn when you wish to find insights into usage of the board or a feature.

A brief description of each section of the manual follows:

Chapter 1: Product Overview

Provides a topside view of the RF3570, intending to orient you to the product as a whole.

Chapter 2: Hardware Essentials

Explains, in text and illustration, the basic structure of the RF3570 hardware; port addresses and usage.

Chapter 3: Hardware Installation

Takes a practical approach to physically installing the RF3570 into a VMEbus system and attaching SCSI peripherals.

Chapter 4: Command Operation

Explains the command structures you will use and how to issue a command.

Chapter 5: SCSI Pass-through Commands

Describes the Parameter Block structure you should use for SCSI Pass-through commands. These are commands that manipulate a device on the SCSI bus.

Chapter 6: Board-control Commands Reference

A look-up reference for Board-control commands. These are commands that are interpreted by the adapter to affect its operation; they are used to tailor board operation for specific systems and applications.

Chapter 7: Details of Usage

Explains in detail how and why to use some of the features of the RF3570 adapter.

Appendix A: Error Codes

Lists the error codes returned in the Error field of the Status Block and also the codes returned in the Status Port.

Appendix B: Cables and Connectors

Provides pin-outs of connectors and specifies maximum cable length.

Appendix C: Specifications

Describes the physical and electrical characteristics of the board.

Appendix D: Defaults

Provides default values used by the RF3570: jumpers, hardware ports and command.

Conventions

This manual incorporates the following conventions:

Notational

1. An upper case letter H following a number indicates the number is a hexadecimal value (e.g., 32 = 20H and 255 = FFH).

2. In this document, a byte is defined as an 8-bit quantity, a word as a 16-bit quantity, and a double word as a 32-bit quantity.

3. This manual represents all memory ordering in the Motorola format. This means that the least significant byte of a double word is placed in the most significant memory location. For a further discussion of memory ordering and information to use Intel ordering, see Chapter 7.

Graphic

In order to explain the command structures used by the RF3570, to describe the breakdown of the structure into fields, (double-word, word, and byte wide) this manual represents the structures and fields graphically, *as they would be placed in memory.*

Throughout the RF3570 User's Guide you will find structures graphically represented similar to Figure P-1.

The shaded grey areas of the structure provide a reference for the position of the components of the structure (the white fields) in memory.

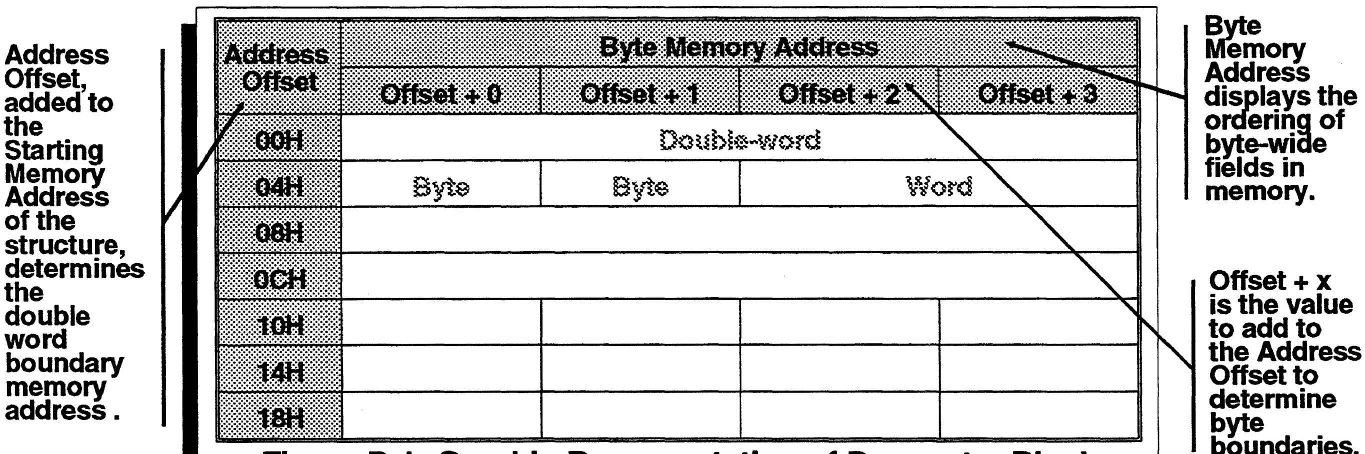


Figure P-1: Graphic Representation of Parameter Block

Address Offset

The grey-shaded area on the left is named Address Offset. Since the structures are double-word wide, these hexidecimal values, added to the Starting Memory Address, reference the beginning of each double word in the structure.

Byte Memory Address

The grey-shaded area at the top of the graphic is the Byte Memory Address. Each of the four Offset + x fields describe the relationship of byte fields to the Address Offset.

Reference Documents

These documents apply to the Ciprico Rimfire 3570 SCSI host bus adapter. They may be of interest to you:

VMEbus Specification Manual, Revision C.1 (Motorola Publication)

American National Standard: Small Computer System Interface (SCSI-1) Standard, (ANSI X3.131, 1986)

American National Standard: Common Command Set, ANSI X3 Working Committee Document, X3t9.2/85-52 Rev.4A

Draft Proposed American National Standard: Small Computer System Interface-2 , (ANSI X3.131, 199X)

User manuals for applicable SCSI devices.

TABLE OF CONTENTS

Chapter 1 — Product Overview 1-1

- Introduction 1-1
- Summary 1-1
- About the RF3570 Design 1-2
  - Characteristics 1-2
- Features 1-2
  - SCSI Port 1-2
  - VMEbus 1-3
  - Host Interface 1-3
- What next ? 1-4

Chapter 2 — Hardware Essentials 2-1

- Introduction 2-1
- Summary 2-1
- Hardware Structure and Description 2-2
  - Intel 80186 2-2
  - Emulex Fast SCSI Chip 2-2
  - Short Burst FIFO 2-2
  - Pipelined System Interface 2-2
  - Command Interface 2-3
  - Block Diagram 2-3
- Board Addressing 2-5
  - Choosing an Address Modifier 2-5
  - Choosing a VMEbus Address 2-6
- Hardware Ports: General 2-6
  - Hardware Port Addresses 2-6
  - Overview of the Ports 2-7
- Address Buffer Port 2-8
- Channel Attention Port 2-12
  - Channel Attention Port Format 2-13
- Status Port 2-14
  - Status Port: Reset Format 2-14
  - Status Port Error Codes 2-16
  - Status Port: General Format 2-17
- Reset Port 2-19
- Descriptions of Port Usage 2-19

Chapter 3 — Hardware Installation	**3-1**
Introduction	3-1
Summary	3-1
Overview of the Installation Process	3-2
Step 1: Unpacking the Board	3-4
Shipping Damage	3-4
Unpacking the Board	3-5
Step 2: Jumpers and Termination	3-6
Factory-set Jumpers	3-6
SCSI Configuration Jumpers	3-6
Bus Request/Grant Jumper	3-8
Sysfail Jumper	3-9
Exact Burst Jumper	3-9
VMEbus Address Jumpers	3-10
Example Addresses	3-11
SCSI bus Termination	3-12
Step 3: Board Insertion	3-14
General Steps	3-14
Step 4: Apply power and observe LEDs	3-16
Meaning of LED's	3-16
Power-up Sequence	3-16
Error Conditions	3-17
Step 5: Configure and Connect SCSI Devices	3-18
Selectable Options	3-18
Termination	3-19
Cabling	3-19
Chapter 4 — Command Operation	**4-1**
Introduction	4-1
Summary	4-1
What the Adapter must accomplish	4-2
The Base Parameter Block	4-4
Parameter Block Format	4-4
The Base Status Block	4-6
Status Block Format	4-6
Multiple Status Blocks	4-8
Single Command Structure	4-10
Format of Single Command Structure	4-10
Using a Single Command Structure	4-13
Set-up the Structure in memory	4-13
Prepare the Hardware Ports	4-13
Execute the Command	4-13
Determine Command Completion	4-13
Some Example Single Commands	4-14
Example 1: Start Command List Board-control command	4-14

Example 2: Inquiry Pass-through command	4-17
Command List Structure	4-20
Format of Command List Structure	4-20
Description of the Fields	4-20
Creating the Command List Structure	4-22
Using a Command List Structure	4-23
Index Guidelines	4-23
Command Guidelines	4-24
Adding a Parameter Block	4-24
Removing a Status Block	4-25
Example 'C' Routines for Command List	4-26
Chapter 5 — SCSI Pass-through Commands	**5-1**
Introduction	5-1
Summary	5-1
SCSI Pass-through Parameter Block Format	5-2
Command Format	5-2
Description of the Fields	5-2
Setting DBV, DAT and DIR Bits	5-5
Using the Table Values	5-5
Table of DAT and DIR Automatic Settings	5-7
Message Format	5-15
Description of the Fields	5-15
SCSI Pass-through Status Block Format	5-16
Status Block Format	5-16
Description of Fields	5-16
Chapter 6 — Board-control Commands Reference	**6-1**
Introduction	6-1
Summary	6-1
Using the Reference	6-2
General Status Block	6-3
General Status Block Format	6-3
Description of Fields	6-3
List of Commands	6-5
Start Command List (01H)	6-6
Command Format	6-6
Description of Fields	6-6
Stop Command List (02H)	6-8
Command Format	6-8
Description of Fields	6-8
Identify (05H)	6-10
Command Format	6-10
Description of Fields	6-10
Identify Command Status Block	6-11

Description of Fields 6-11
Board Statistics (06H) 6-14
Command Format 6-14
Description of Fields 6-14
Statistics Data Structure 6-16
Description of Fields 6-16
General Options (07H) 6-18
Command Format 6-18
Description of Fields 6-18
Unit Options (08H) 6-22
Command Format 6-22
Description of Fields 6-22
Diagnostic/Self-Test (09H) 6-30
Command Format 6-30
Description of Fields 6-30
Status Block Format 6-33
Description of Fields 6-33
SCSI Hard Reset (10H) 6-34
Command Format 6-34
Description of Fields 6-34
Board Information (15H) 6-36
Command Format 6-36
Description of Fields 6-36
Board Information Data Structure 6-38
Description of Fields 6-38
Extended Board Statistics (16H) 6-48
Command Format 6-48
Description of Fields 6-48
Extended Statistics Data Structure 6-50
Description of Fields 6-50
Extended Unit Options (18H) 6-52
Command Format 6-52
Description of Fields 6-52
Extended Unit Options Data Structure 6-54
Description of Fields 6-54

Chapter 7 — Details of Usage 7-1

Introduction 7-1
Summary 7-1
Byte and Word Swapping 7-2
Need for Swapping 7-2
RF3570 Swapping Options 7-6
Scatter/Gather Operations 7-10
What it does 7-10
How to use Scatter/Gather 7-10
Format of Scatter/Gather Descriptor Block 7-11

How Scatter/Gather is Performed 7-12
Notes on Use 7-13
Example Scatter/Gather Operation 7-14
Odd Byte Handling 7-16
What is an Odd Byte? 7-16
Other Definitions 7-17
Algorithms for Transfer Width 7-18
Example 1: A Bus-width Aligned Transfer 7-20
Example 2: Transfer using a Non-aligned Transfer Count 7-20
Example 3: Transfer using a Byte-aligned Address 7-21
Example 4: Transfer using a Word-aligned Address 7-22
Example 5: Non-aligned Transfer Count and Address 7-23
Deciding on an Algorithm 7-25
Tagged Queued Commands 7-26
How to Use Tagged Queuing 7-26
How It Works 7-28
Notes on Usage 7-28

Appendix A — Error Codes A-1

Introduction A-1
Summary A-1
Status Block Error Codes A-2
Status Port Error Codes A-6
Catastrophic Error Codes A-6
Self-test Error Codes A-7
Diagnostic/Self-test Command A-7

Appendix B — Cables and Connectors B-1

Introduction B-1
Summary B-1
Differential Connectors B-2
Maximum Cable Length B-2
Ribbon Cable Connector B-2
High-Density Connector B-3
Single-ended Connectors B-4
Maximum Cable Length B-4
Ribbon Cable Connector B-4
High-Density Connector B-5

Appendix C — Specifications C-1

Introduction C-1
Summary C-1
Specifications C-2

Appendix D — Defaults D-1

Introduction D-1
Summary D-1
Hardware Defaults D-2
Board Jumpers D-2
Hardware Ports D-3
Command Defaults D-4

List of Figures

Figure 2-1: Block Diagram of RF3570	2-4
Figure 2-2: Choosing an Address Modifier	2-5
Figure 2-3: Address Line Decoding	2-6
Figure 2-4: Hardware Port Addresses	2-6
Figure 2-5:Address Buffer Port Writes	2-8
Figure 2-6: Control Field of Address Buffer Port	2-9
Figure 2-7: Channel Attention	2-13
Figure 2-8: Status Port: Reset	2-14
Figure 2-9: Status Port: General	2-17
Figure 3-1: Locations of Rimfire 3570 Jumpers	3-6
Figure 3-2: SCSI Configuration Jumper Block	3-7
Figure 3-3: Bus Request/Grant and Sysfail Jumpers	3-8
Figure 3-4: Exact Burst and VMEbus Address Jumpers	3-10
Figure 3-5: Single-ended Board Fuse and Terminators	3-12
Figure 3-6: Terminators and Fuse on Differential Board	3-13
Figure 3-7: Inserting the RF3570	3-15
Figure 3-8: Location of LEDs	3-16
Figure 4-1: Base Parameter Block	4-4
Figure 4-2: Base Status Block	4-6
Figure 4-3: Bit-meanings of Flags Field	4-7
Figure 4-4: Single Command Structure	4-10
Figure 4-5: Bit-meanings of Interrupt Field	4-11
Figure 4-6: Example Start Command List PB	4-14
Figure 4-7: Single Command Structure in memory	4-15
Figure 4-8: Single Command Structure in memory	4-16
Figure 4-9: Example Pass-through command PB	4-17
Figure 4-10: Single Command Structure in memory	4-18

Figure 4-11: Single Command Structure in memory	4-19
Figure 4-12: Command List Structure	4-20
Figure 5-1: SCSI Pass-through Parameter Block	5-2
Figure 5-2: Bit-meanings of Address Modifier Field	5-3
Figure 5-3: Bit-meanings of Flags-1 Field	5-3
Figure 5-4: Example DAT and DIR bit settings	5-5
Figure 5-5: DIR and DAT Settings, Vendor-unique	5-6
Figure 5-6: Bit-meanings of Flags-2 Field	5-12
Figure 5-7: SCSI Pass-through Message Parameter Block	5-15
Figure 5-8: SCSI Pass-through Status Block	5-16
Figure 5-9: Bit-meanings of Flags Field	5-17
Figure 5-10: Bit-meanings of SCSI Status Field	5-18
Figure 5-11: SCSI Status Field - Unexpected SCSI Phase	5-19
Figure 5-12: Default Ordering of Sense Bytes	5-21
Figure 5-13: Ordering of 1st 16 Sequential Sense Bytes	5-21
Figure 5-14: Example of Selectable Sense Bytes	5-22
Figure 6-1: General Status Block	6-3
Figure 6-2: Bit-meanings of Flags Field	6-4
Figure 6-3: Start Command List Parameter Block	6-6
Figure 6-4: Bit-meanings of Interrupt Field	6-7
Figure 6-5: Stop Command List Parameter Block	6-8
Figure 6-6: Identify Command Parameter Block	6-10
Figure 6-7: Identify Command Status Block	6-11
Figure 6-8: Bit-meanings of Flags Field	6-11
Figure 6-9: Option Flags Field - Identify Status Block	6-12
Figure 6-10: Adapter SCSI ID Bits	6-13
Figure 6-11: Board Statistics Command Parameter Block	6-14
Figure 6-12: Bit-meanings of Statistics Options Field	6-15

Figure 6-13: Board Statistics Data Structure 6-16
Figure 6-14: General Options Command Parameter Block 6-18
Figure 6-15: Bit-meanings of Bus Throttle Field 6-19
Figure 6-16: Bit-meanings of Select Flags Field 6-20
Figure 6-17: Unit Options Command Parameter Block 6-22
Figure 6-18: Bit-meanings of Retry Control Field 6-24
Figure 6-19: Unit Flags Field of Unit Options Command 6-25
Figure 6-20: Default Ordering of Sense Bytes 6-28
Figure 6-21: Example Selection of Sense Bytes 6-29
Figure 6-22: Diagnostic/Self-test Parameter Block 6-30
Figure 6-23: Diagnostics Command Test Flags Field 6-31
Figure 6-24: Diagnostics Command Status Block 6-33
Figure 6-25: Bit-meanings of Flags Byte in Status Block 6-33
Figure 6-26: SCSI Hard Reset Parameter Block 6-34
Figure 6-27: Board Information Command Parameter Block 6-36
Figure 6-28: Board Information Data Structure 6-38
Figure 6-29: Board Information - Select Flags Field 6-39
Figure 6-30: Bit-meanings of Bus Throttle Field 6-40
Figure 6-31: Bit-meanings of Option Flags Field 6-41
Figure 6-32: Adapter SCSI ID Bits 6-41
Figure 6-33: Bit-meanings of Termination Field 6-42
Figure 6-34: SCSI Configuration Jumpers 6-43
Figure 6-35: Bit-meanings of Device Flags Field 6-44
Figure 6-36: Extended Board Statistics Parameter Block 6-48
Figure 6-37: Extended Board Statistics Options Field 6-49
Figure 6-38: Extended Board Statistics Data Structure 6-50
Figure 6-39: Extended Unit Options Parameter Block 6-52
Figure 6-40: Extended Unit Options Data Structure 6-54

Figure 6-41: Bit-meanings of Retry Control Field 6-55
Figure 6-42: Unit Flags Field of Extended Unit Options 6-57
Figure 6-43: Default Ordering of Sense Bytes 6-60
Figure 7-1: Variations in Intel and Motorola Memory 7-2
Figure 7-2: Explanation of Memory Storage Differences 7-3
Figure 7-3: Intel versus Motorola Ordering 7-4
Figure 7-4: Motorola Ordered Parameter Block 7-5
Figure 7-5: Intel Ordered Parameter Block 7-5
Figure 7-6: Scatter/Gather Descriptor Block 7-11
Figure 7-7: Example Scatter/Gather Parameter Block 7-14
Figure 7-8: First SG Descriptor Block 7-14
Figure 7-9: Last SG Descriptor Block 7-15
Figure 7-10: Meaning of Flags-2 Field 7-27
Figure A-1: Base Status Block A-2
Figure A-2: Status Port: General A-6
Figure A-3: Status Port: Reset A-7

List of Tables

Table 2-1: VMEbus Address Modifiers for RF3570 2-5

Table 2-2: Status Port Error Codes 2-16

Table 2-3: Status Port Catastrophic Error Codes 2-18

Table 3-1: Default Jumper Settings 3-7

Table 3-2: Example Base Address Jumper Settings 3-11

Table 3-3: Status Port Catastrophic Error Codes 3-17

Table 5-1: DAT and DIR Automatic Settings 5-7

Table 5-2: Meaning of TAG OPTS Bits 5-12

Table 5-3: Meaning of Status Byte Code 5-18

Table 5-4: Meaning of Status Byte - Unexpected Phase 5-19

Table 6-1: Valid Ranges of Throttle Count Values 6-20

Table 6-2: SCSI Cfig Block - Default Jumper Settings 6-43

Table 7-1: Meaning of TAG OPTS Bits 7-27

Table A-1: Status Port Catastrophic Error Codes A-6

Table A-2: Self-test Status Port Error Codes A-8

Table B-1: Cable Connector Pin-outs for the RF3574 B-2

Table B-2: Cable Connector Pin-outs for the RF3576 B-3

Table B-3: Cable Connector Pin-outs for the RF3573 B-4

Table B-4: Cable Connector Pin-outs for the RF3575 B-5

Table C-1: Specifications for the RF3570 C-2

Table D-1: Hardware Defaults - Jumpers D-2

Table D-2: Defaults for Hardware Ports D-3

Table D-3: Command Defaults D-4

Table D-3: Command Defaults (continued) D-5

Table D-3: Command Defaults (continued) D-6

Chapter
1
Product Overview

Chapter 1 — Product Overview

Introduction

This chapter provides an overview of the RF3570 product and it is intended to orient you to the product as a whole, before focusing on specific areas in the chapters that follow.

Summary

In this chapter you will find information about:

- Features of the design.
- Where to find additional information.

This chapter is most valuable to someone who wishes to find out general information about the RF3570 product.

About the RF3570 Design

With the introduction of the RF3570, Ciprico has added to the new generation of VME/SCSI adapters that began with the RF3560. The RF3560 and RF3570 adapters smoothly coordinate two separate bus structures — VME and SCSI.

The VME bus is a commonly used system bus for host computers. The SCSI bus provides a standard interface for a variety of peripheral devices, such as disk and tape drives, and is flexible enough to also support other devices such as printers, scanners, and optical disk drives.

As the SCSI standard has evolved, peripherals that support the Fast SCSI-2 specification have become available. The RF3570 also integrates the Fast SCSI-2 specification and was designed to take advantage of the higher performance offered by the Fast SCSI-2 peripherals.

Characteristics

The RF3570 shares the design goals of the RF3560 in that it strictly adheres to the bus protocols required by the SCSI and VME standards. Like the RF3560, the RF3570 is an easy-to-integrate design.

The RF3570 uses the same architicture as the RF3560. The RF3570 uses a command interface to fetch new commands in parallel with data transfers. This allows the adapter to preprocess a command during a data transfer, in effect speeding up every transaction the host initiates with the RF3570 adapter.

Thus, the RF3570 is plug-compatible with the Ciprico RF3560 product. However the RF3570 offers an enhancement to the RF3560. The RF3570 uses the Emulex Fast SCSI controller chip. Adhering to the SCSI-2 standard, this chip supports Fast SCSI transfers up to 10 MB/s. With it, the RF3570 handles commands on the SCSI bus faster for greater efficiency and overall performance.

Features

These are the features offered by the RF3570 design:

SCSI Port

- Supports up to 56 peripheral devices (7 targets with 8 logical units each).

- Supports Fast SCSI-2: Asynchronous 8-bit data rates up to 7 MB/sec. — Synchronous 8-bit data rates up to 10 MB/sec.
- Supports full SCSI-2 command set, Common Command Set (CCS) of SCSI, and command pass-through for other SCSI commands.
- Performs overlapped SCSI operations using disconnect/reconnect.
- Supports tagged, queued commands allowing multiple, simultaneous commands to be sent to a SCSI-2 target.
- Allows you to select up to 16 unique Sense Bytes (of 256 returned) to receive as status data.
- Jumper selectable SCSI Bus reset on power-up and adapter reset.
- Jumper selectable board SCSI ID on power-up and adapter reset.
- Socketed SCSI bus termination power fuse with status and proper termination report.

VMEbus

- Sustained VMEbus transfer rates at SCSI bus speed.
- Burst VMEbus capability of 20 MB/sec. , or 30 MB/sec using block mode transfers, assuming minimum memory response time of 30ns.
- Supports 16, 24, or 32 bit VMEbus addressing with no addressing restrictions.
- Compatible with Revision C.1 of the VMEbus specification.

Host Interface

- No timing restrictions on processing simultaneous host commands.
- Supports a command list for faster command I/O.
- Hardware byte and word swapping allows support of Intel i86, Motorola 68K, and National 32000 families on the VMEbus.
- Additional VMEbus Command Memory Interface.

What next ?

Depending on what you wish to do next, you should turn to one of the following chapters:

To learn more about the RF3570 hardware
— **Continue to Chapter 2.**

To learn more about the Installation process
— **Go to Chapter 3.**

To learn more about the Command Operation
— **Go to Chapter 4.**

To learn more about SCSI Pass-through commands
— **Go to Chapter 5.**

To learn more about the Board-control commands
— **Go to Chapter 6.**

To learn more about usage
— **Go to Chapter 7.**

Chapter

2

Hardware Essentials

Chapter 2 — Hardware Essentials

Introduction

This chapter explains, in text and illustrations, the basic structure of the RF3570 hardware and is provided to assist you with installing and communicating with the board.

Summary

In this chapter you will learn more about:

- Structure of the hardware based on the VLSI used.
- Hardware ports used to begin communication to the board.

This chapter is useful for anyone wishing to know what technology was used to design the RF3570 and understand the usage of the product.

Hardware Structure and Description

The R3570 hardware design takes advantage of the capabilities designed into four VLSI chips:

- Intel 80186 microprocessor
- Emulex Fast SCSI Controller chip
- Ciprico Short Burst FIFO (SBF)
- Ciprico Pipelined System Interface (PSI)

Additionally, the RF3570 design incorporates:

- A new method of command fetching through a VMEbus Command-Memory Interface

Intel 80186

At the core of the RF3570 design is the 80186 supervisory microprocessor. The 80186 is well suited to the task of overseeing board operations; it uses optimized instruction encoding for high performance and memory efficiency. RF3570 performance is further boosted by using the 16 Mhz version of the 80186.

Emulex Fast SCSI

SCSI bus protocol for the RF3570 is efficiently handled by the Emulex Fast SCSI chip. Additionally, the Emulex chip provides the capability for the RF3570 to negotiate for Fast Synchronous transfers across the bus, up to 10 MB/s. The Emulex Fast SCSI chip is a registered device that can be set-up for automated bus operations. These options are available to you via the Board-control commands discussed in Chapter 6.

Short Burst FIFO

Ciprico's Short Burst FIFO (SBF) is part of the Direct Memory Access (DMA) structure for the RF3570. The SBF works in conjunction with Ciprico's Pipelined System Interface (PSI, discussed next) to provide a high-speed method for data transfer. The SBF provides high-speed 32 byte buffered access to the system data bus. It also contains the address and status port registers discussed later in this section.

Pipelined System Interface

Ciprico created the custom VLSI Pipelined System Interface (PSI) to work in conjunction with the SBF as a high-speed DMA channel. The PSI coordinates access to the system address bus; it has registers for the DMA address count and transfer count. Also contained in the chip are the Channel Attention port register and the Interrupt Request level register.

Command Interface

In addition to the VLSI hardware just described, the RF3570 has a VMEbus Command-Memory Interface. With this additional hardware, the supervisory 80186 microprocessor is able to directly read and write system memory with command/status information without using the DMA path. This allows a new command to be fetched while a data transfer for a current command is executed. This method of parallel DMA and command-fetching boosts performance while being invisible to the user.

Block Diagram

Figure 2-1, on the next page, is a block diagram of the RF3570; it illustrates the hardware just described. The **High Speed VME Bus DMA** block includes the SBF and PSI. **The VME Bus Interrupt Generator** block is also part of the PSI.

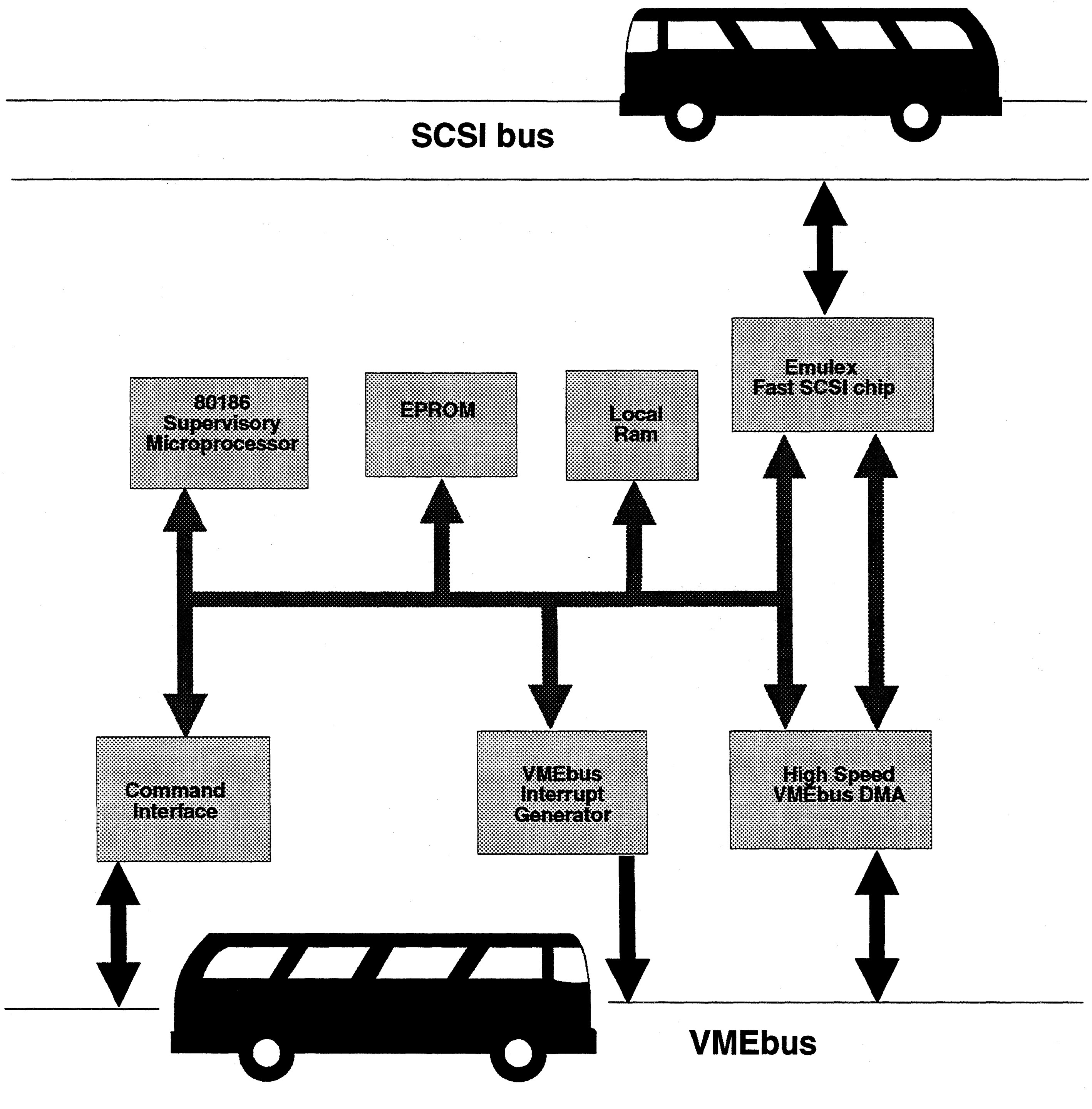


Figure 2-1: Block Diagram of RF3570

Board Addressing

The Board Address and Slave Address Modifier to which the RF3570 responds on the VMEbus are set with jumpers on the board.

Choosing an Address Modifier

The RF3570 adapter is an A16 SLAVE device on the VMEbus. In keeping with VMEbus Specifications, the Address Modifier for selection of this type of device can be either 29H or 2DH. Since there is only one bit difference between the two values, a single jumper is provided to select an Address Modifier.

AM5	AM4	AM3	AM2	AM1	AM0
1	0	1	Jumper In = 0 Out = 1	0	1
VMEbus Address Modifier Lines					

Figure 2-2: Choosing an Address Modifier

Choose Address Modifier 2DH by removing the jumper, when you wish to reserve the RF3570 registers for short supervisory access. Choose 29H by inserting the jumper, if you wish to allow short non-privileged access:

Hex Code	Address Modifier						Function
	5	4	3	2	1	0	
2D	1	0	1	1	0	1	Short Supervisory Access
29	1	0	1	0	0	1	Short Non-Privileged Access

Table 2-1: VMEbus Address Modifiers for RF3560

The factory setting for the RF3570 Slave Address Modifier is 2DH; the jumper is out.

The location and setting of this jumper is described in Chapter 3.

Choosing a VMEbus Address

With the address jumpers on the RF3570 board, you set the upper 7 bits of the Board Address. This effectively reserves 512 bytes of VMEbus 16- bit address space. Within the reserved area, the RF3570 adapter has four register ports that are the foundation of its communication scheme.

<table>
<tr><td>A15</td><td>A14</td><td>A13</td><td>A12</td><td>A11</td><td>A10</td><td>A9</td><td>A 8</td><td>A 7</td><td>A 6</td><td>A 5</td><td>A 4</td><td>A 3</td><td>A 2</td><td>A 1</td><td>A 0</td></tr>
<tr><td colspan="7">Address lines decoded to identify the RF3570 base address which is set via jumpers on the board.</td><td></td><td></td><td></td><td></td><td colspan="2">Address lines decoded to select one of four ports</td><td></td><td></td><td></td></tr>
</table>

Figure 2-3: Address Line Decoding

The factory setting for the RF3570 VMEbus Address is to jumper A12 making the default base address EE00H. Details about the location and setting of the jumpers can be found in **Chapter 3, Hardware Installation**.

Hardware Ports: General

There are four ports that are used to communicate with the RF3570 adapter: Address Buffer Port; Channel Attention Port; Status Port; and Reset Port. The address and function of each port is described in this section.

Hardware Port Addresses

Each of the Hardware ports can be addressed as an offset from the base address, set with jumpers on the board. Further, all ports can be accessed in 16-bit mode. To address a specific port you must add the correct offset to the base address value. The offset values for the RF3570 hardware ports are described in Figure 2- 4.

Port	Address Offset	Host Read or Write
Address Buffer	00H	Write
Channel Attention	08H	Write
Status	10H	Read
Reset	18H	Write

Figure 2-4: Hardware Port Addresses

Example Port Addresses

Using the factory set Base Address for the RF3570, EE00H, the Hardware Port Addresses would be:

Address Port = EE00H
Channel Attention Port = EE08H
Status Port = EE10H
Reset Port = EE18H

Figure 2-4 also indicates how each port can be accessed by the Host: read or write.

Overview of the Ports

An overview of the operation of all of the Hardware Ports is provided next. The overview is followed by a detailed description of each Hardware Port that focuses on the rules of operation.

Address Buffer Port

This port is instrumental in setting up the RF3570 adapter I/O operation. With it you can do two things:

1.) **Set system bus transfer characteristics such as bus width and byte/word swapping for subsequent operations.** Usually this is done only with the first command sent to the board, when you are initializing the board for operation in your system.

2.) **Pass the address of a Single Command Structure to the adapter.** Any time you wish to issue a Single Command, you must use the Address Buffer Port to pass the address structure. Most often, the first Single Command executed contains a command to Start Command List. This function of the port is only used when issuing a Single Command.

Channel Attention Port

The Channel Attention Port is your means of informing the RF3570 adapter that a command is ready to be executed. The value you write to this port tells the adapter what kind of Command Structure to expect. Before you write to the Channel Attention Port you must set-up the appropriate Command Structure in the system memory space you will be directing the adapter to read.

Status Port

This port is used to read status from the adapter. There are two types of status that can be read: **General Status** — returned after a command completes during normal operation; and **Reset Status** — returned during board self-test at reset or power-up.

Reset Port

With the Reset Port you are able to initiate, through software, a reset identical to a hardware reset.

Address Buffer Port

The Address Buffer Port is a write-only port. It is used when you wish to execute a command using a **Single Command Structure**. It is loaded with: the starting system address of the structure; an address modifier value to be used when reading the Parameter Block contained in the structure; and a byte of control information through which you may instruct the RF3570 adapter about subsequent operation.

The Address Buffer Port register is three words deep and 16 bits wide. To pass all three words requires three writes to the port:

- The first write must contain the Control byte and Address Modifier.
- The second write contains the Most Significant Word (MSW) of the **Single Command Structure** address in system memory.
- The last write contains the Least Significant Word (LSW) of the **Single Command Structure** address.

The order of these writes is illustrated in Figure 2-5:

Bits	15 8	7 0
1st Write	Control	AM Bits for SCS
2nd Write	Single Command Structure Address: MSW	
3rd Write	Single Command Structure Address: LSW	

Figure 2-5:Address Buffer Port Writes

A description of each of the fields of the Address Buffer Port follows.

Address Modifier Field

The Address Modifier (AM) Field of the Address Buffer Port is used to specify the Address Modifier value that the RF3570 adapter should use when reading the first Parameter Block.

Control Field

Use this byte-wide field to tailor the RF3570 adapter's use of the system bus. With it, you can set-up the adapter to:

- transfer data in 16-bit or 32-bit mode on the VMEbus.
- byte or word swap command structures to be compatible with the byte and word ordering of either Intel or Motorola architectures.
- byte or word swap data to change the data byte ordering.

Byte and word swapping are further discussed in **Chapter 7, Details of Usage.**

The bit meanings of the Control Field are as follows:

7	6	5	4	3	2	1	0
SET	0	0	WSD	BSD	WID	WSC	BSC

Figure 2-6: Control Field of Address Buffer Port

NOTE: *For any of the control bits to be considered valid by the adapter, the SET bit must be one.*

BSC **Byte Swapping Control:**
Defines whether high and low bytes of a word will be swapped during transfer of status and parameter blocks.
0 ⇨ No swapping.
1 ⇨ Change byte order.

WSC **Word Swapping Control:**
Defines whether high and low words of a double word will be swapped during transfer of status and parameter blocks.
0 ⇒ No swapping.
1 ⇒ Change word order.

WID **WIDth of data transfers:**
Determines whether data transfers across the system bus will be 16-bit or 32-bit wide
0 ⇒ 16-bit data transfers.
1 ⇒ 32-bit data transfers.

BSD **Byte Swap Data**
Determines whether high and low bytes of a word will be swapped during data transfer.
0 ⇒ byte order not changed.
1 ⇒ bytes swapped during transfer.

WSD **Word Swap Data**
Defines whether high and low words of a double word are swapped during data transfer.
0 ⇒ word order not changed.
1 ⇒ words swapped during transfer.

SET **select new control SETtings**
Determines whether the rest of the bits in the Control Field should be applied to adapter operation.
0 ⇒ ignore all other bits.
1 ⇒ apply bit values to next transfer.

Determining a Control Field value

These are the significant points to remember when determining the value you should use for the Control Field:

- The Control Field value is meant to be set once, usually during system initialization. Generally, its value will be set with the first command issued to the board.

- The values you choose for the Control Field will be applied immediately. They will be applicable for *the next parameter block read in* and all future system memory accesses.

- The Control Field value is *not ever* applied to the Address Buffer Port or any of the other hardware ports. Always follow the format given in this section when accessing the hardware ports.
- Attempting to change the Control Field setting after the RF3570 adapter has begun executing a command list will cause unexpected results.
- The default value for the Control Field is zero; the adapter will operate in 16-bit mode with Motorola ordering of bytes and words during data transfers and the parameter and status block structures will not be swapped.

Using the Byte and Word Swap controls with data transfers

There are some guidelines for you to observe when using the byte and word swapping capabilities of the RF3570 adapter:

1.) If you enable byte-swapping, you must be sure that all system addresses and transfer counts that you use are aligned to a *word* boundary. Otherwise the byte swap setting will be ignored.

2.) If you enable word-swapping, you must align all your system addresses and transfer counts to a *double word* boundary. Otherwise the word swap setting will be ignored.

Single Command Structure Address Fields

The Address Buffer Port is loaded with these two words to communicate to the RF3570 adapter the address of the Single Command Structure containing the command to be issued to the board.

The Address Buffer Port is only used when issuing a Single Command to the RF3570 adapter.

Channel Attention Port

The Channel Attention Port is a write-only port. It is used to direct the RF3570 adapter to execute Parameter Block(s) associated with a Command Structure you have already built in system memory. The Channel Attention Port is used for both Single and Command List issue. It is written slightly differently in each case though.

Single Command Channel Attention

To initiate execution of a Single Command you must do the following:

1.) Construct a Single Command Structure in system memory.

2.) Load the Address Buffer Port with the Control Byte, Address Modifier and the address of the Single Command Structure constructed in Step 1.

3.) Write a **0** to the Channel Attention Port, instructing the RF3570 adapter to execute the Parameter Block contained in the Single Command Structure.

Use an offset of 8H from your Base Address when writing to the Channel Attention Port.

Command List Channel Attention

To execute commands via Command List, you must do the following:

1.) Have an active Command List. (Command Lists are begun by issuing the Start Command List command via a Single Command Structure.) If you attempt to issue a command via Command List when there is not an active List, you will receive an error code of 11H in the Status Port.

2.) Fill the next available Parameter Block in the Command List with the command you wish to have executed.

3.) Adjust the Command List Parameter Block IN Index to indicate additional commands have been added.

4.) Write a **1** to the Channel Attention Port, instructing the RF3570 adapter to begin executing the Parameter Block(s) contained in the Command List structure.

If you have multiple Parameter Blocks ready in the space allocated for Command List, you may issue separate Channel Attention writes or you may issue one for the entire group.

Use an offset of 8H from your Base Address when writing to the Channel Attention Port.

For more information about how to use the Command List Structure, see Chapter 4, Command Operation.

BIT		
2	1	0
0	0	TYP

Figure 2-7: Channel Attention

Channel Attention Port Format

The Channel Attention Port is three bits wide with the following meaning:

TYP **Channel Attention Type**
Indicates the type of command being issued with a write to the Channel Attention Port.
0 ⇒ Single Command Issue.
1 ⇒ Command List Issue.

Status Port

The Status Port is a read-only port. It contains information regarding the condition of the board during power-up/reset and during normal board operation. It is used to report:

- A sequence of self-test codes while a reset is occurring.
- An error code if self-test fails or a catastrophic error occurs during normal operation.
- Readiness of the board for commands (RDY bit).
- Readiness for a new Single Command (ENT bit).

The Status Port is 16-bits wide. The bit meanings of the port will be different depending on whether the status is General information or Reset information; each has separate formats. Both types of format are described in this section.

Status Port: Reset Format

The RF3570 adapter performs a series of self-tests on reset or power-up. The Status Port is used by the RF3570 adapter to indicate the progress of the self-test as it is occurring. At power-up or reset, the Status Port will have this format:

BIT															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0								Self-test Code (STC)						RDY	STC

Figure 2-8: Status Port: Reset

Upon power-up or reset the following sequence will occur:

1. Immediately after reset the entire Status Port will be zeroed.
2. Next, self-test begins and as it progresses, the Self-test Code fields are updated with the code of the test currently executing.
3. The RDY bit is zero during self-test to indicate the board cannot accept any command. All other fields of the Port (except Self-test Code (STC) fields) also remain zero as long as self-test continues.

4. The RDY bit will change to one if the self-tests complete successfully. The self-tests take less than 1 second. If, after this time, the RDY bit is still zero, a self-test has failed.

   If a self-test fails, the RDY bit will remain zero and the code of the test that failed will remain in the Self-test Code fields of the Status Port.

5. After the RF3570 self-tests complete successfully, and the RDY bit is set to one, the Status Port format will change to the General format.

NOTE: *Whenever the board is reset, the host operations should wait a minimum of one second before testing the RDY bit. If, after the delay the RDY bit is not set to one, the value in the Self-test Code field of the Status Port will indicate the failing test. The RF3570 will not accept commands if this occurs.*

The Status Port fields during power-up or reset, have these meanings:

STC — **Self-test Code**
This bit works in conjunction with bits 2 through 7, to indicate the self-test currently executing.

RDY — **Ready**
This bit indicates when the adapter is ready to accept commands after a reset has completed.
0 ⇨ No command can be accepted
1 ⇨ Ready to accept a command

Self Test Code — **Self-test Code**
The values of these bits change as self-test progresses. After a hardware reset they are zero. As self-test begins and progresses, the values are updated to the code of the test currently executing.

See next page for a table of Self-test Codes.

Status Port Error Codes

If one of the self-tests does not properly complete, its test number will be left in the Self-test Code byte. The following table describes the test numbers:

NOTE:

Bit 1 of the Status Port is not set until the board is in a Ready condition.

Code	Description
00H	The onboard processor is not working; it cannot access firmware EPROM; cannot access the Status Port - problem with Short Burst FIFO or VMEbus data lines.
04H	Static RAM error. All words of static RAM are written with FFFFH, then zero. This test is not exhaustive.
0CH	Firmware checksum error. The calculated checksum for odd and even EPROMs does not match the value stored in EPROM.
14H	Short Burst FIFO cannot be accessed. SBF data could not be loaded or read back.
1CH	Channel Attention Port cannot be accessed, or interrupt won't clear.
90H	PSI Address/Counter registers not masked to WORD boundaries.
94H	PSI Address/Counter registers not masked to DWORD boundaries.
98H	Value Miscompare errors of PSI R/W registers in Manual Load mode.
9CH	Value Miscompare errors of PSI R/W registers in Auto Load mode.
A8H	SCSI termination power fuse is blown, but termination power is still present at the SCSI connector.
ACH	SCSI termination power fuse is blown and termination power is *not* available at the connector.
B0H	Value Miscompare error of Emulex Fast SCSI chip R/W registers.
B4H	Emulex Fast SCSI chip did not generate a reset interrupt.
B8H	Emulex Fast SCSI chip registers not zeroed after reset.
BCH	SCSI bus hung, waiting for reset from Emulex Fast SCSI chip. Could be cabling.
C0H	Short Burst FIFO 32 Bytes transfer miscompare error.
C4H	Short Burst FIFO Local Ready of System Not Ready error.
C8H	Short Burst FIFO byte transfer miscompare error.
CCH	Short Burst FIFO Local Not Ready or System Ready error.
E5H	The onboard processor failed the accumulator test.
E8H	The onboard processor failed the store instruction test.
E9H	The onboard processor failed the shift instruction test.
ECH	The onboard processor failed the compare instruction test.
EDH	The onboard processor failed the jump instruction test.

Table 2-2: Status Port Error Codes

Status Port: General Format

After the RF3570 self-tests have completed successfully and the RDY bit is set to one, the Status port is used to return general board information. The Status Port will have the following format:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Code								0			ERR	0	0	RDY	ENT

Figure 2-9: Status Port: General

The meanings of the Status Port fields are as follows during normal operation:

ENT **Entered**
This bit is used to signal when a new Single Command may be issued to the adapter via a Channel Attention. The board is ready for another Single Command write to the Channel Attention Port each time this bit value toggles. It *does not* indicate that a command has completed, only that a new Single Command may be initiated.
Initially this bit ⇨ 0
After 1st Single Command Channel Attention ⇨ 1;
After 2nd Single Command Channel Attention ⇨ 0; etc.

NOTE: This bit *does not* toggle for Command List Channel Attentions.

RDY **Ready**
This bit indicates when the adapter is ready to accept commands after a reset has completed. *During normal operation it should always be set to one.*
0 ⇨ No command can be accepted.
1 ⇨ Ready to accept a command.

ERR **Error**
This bit is used to signal that a catastrophic error occurred that could not be reported through the command interface. (Example: Using an invalid Parameter Block address

which resulted in a VMEbus error. Typically the AMBER LED will also light if this bit is set.)
0 ⇨ Normal operation.
1 ⇨ Error condition.

Code **Board Type or Error Code**
When the ERR bit is set to one, this byte will contain a catastrophic error code. If the ERR bit is zero, the code byte will contain the Ciprico board type.
02 ⇨ RF35XX board type.
Other than 02 ⇨ Catastrophic Error Code.

There are several catastrophic error codes:

Code	Name	Description
10H	**Bad PBIN Value**	The Command List PBIN (Parameter Block IN) value exceeded the value set for PBNUM (Parameter Block Number).
11H	**State wrong for Start/Stop Command List command**	This code is returned under these conditions: **Start Command List** command issued when a command list is already active. **Stop Command List** command issued when no command list is presently active. **Command List Channel Attention** issued when no Command List is active.
14H	**Software VMEbus Timeout**	The adapter times each of its VMEbus transfers; it will report this error if a VMEbus transfer takes too long to complete. This can occur for transfers of parameters as well as transfers of data.
15H	**VMEbus Error Occurred**	A VMEbus error was detected while the adapter was transferring either parameters or data.
96H	**Internal Firmware Error**	This error code indicates that a Firmware error was detected during the execution of a command.

Table 2-3: Status Port Catastrophic Error Codes

To access the Status Port. use an offset of 10H from the base address of the adapter (which is set by jumpers on the board).

Reset Port

This is a write-only port. You may write to this port when you wish to initiate a board Reset. The write can consist of any value because the data is ignored. The result of this write will be a hardware reset.

Whether or not the RF3570 adapter also resets the SCSI bus at this time is determined by how you set a jumper on the board. (Information about the configuration jumpers is in Chapter 3, Hardware Installation.)

To access the Reset Port use an offset of 18H from the base address of the adapter (set by jumpers on the board).

When a reset has completed with no self-test errors, the RDY bit of the Status Port will be set.

Descriptions of Port Usage

The following describes port usage. Assume that the adapter address jumpers are set to a base address of EE00H.

Issuing a Reset — Write to Reset Port at address EE18H. The RF3570 will begin executing self-test. Read the Status Port at address EE10H to find when self-test completes.

Issuing a Single Command — Write three words to Address Buffer Port at address EE00H. These words select system bus width/swapping options as well as the Address Modifier and memory address of the Single Command. To begin command execution, issue a Single Command Channel Attention by writing a 0 to Channel Attention Port at address EE08H. Read Status Port at EE10H to check for Enter condition before issuing the next command.

Issuing a Command List — Once an active Command List has been established, write a 1 to the Channel Attention Port (address EE08H) for a Command List Channel Attention after the new commands are added to the list.

Read Board Command Status — Read the Status Port at address EE10H to find the state of the adapter.

Chapter
3
Hardware
Installation

Chapter 3 — Hardware Installation

Introduction

Unifying the parts of a system into a whole requires a smooth transition of the theoretical into the actual. This chapter takes a very practical approach to physically installing an RF3570 board into a VME-based system and attaching SCSI devices.

Summary

In this chapter you will learn:

- The hardware installation process from unpacking to power-up.
- What configuration changes to make.
- What to look for in the documentation that comes with your SCSI device.

This chapter is most helpful for someone who is responsible for making the hardware operational.

Overview of the Installation Process

There are only five steps involved in bringing your RF3570 adapter into functionality. Each is outlined briefly below and then in more detail on the following pages.

Step 1: Unpack the board

The RF3570 adapter is carefully packaged to prevent damage and exposure to static. You should observe precautions while unpacking.

See page 3-4 for more information.

Step 2: Set jumpers and termination on the board

There are eight blocks of jumpers on the board. Three jumpers are factory set and should not be changed. You will need to know how to set the remaining five. These jumpers are for: default SCSI ID, default SCSI bus reset on power-up, enable SCSI parity-checking, Board VMEbus address and Address Modifier, SYSFAIL signal assertion, Exact Burst counter, and Bus Arbitration.

You will also need to determine whether you will terminate the SCSI bus on the board. You will need to remove termination if the RF3570 adapter is not at the physical end of the SCSI bus cable.

See page 3-6 for more information.

Step 3: Insert the board into a VME slot

You must select a VME slot and prepare it for the adapter by removing Bus Grant and Interrupt Acknowledge daisy-chain jumpers from the VMEbus backplane. There are many types of system enclosures that the RF3570 will fit into.

See page 3-14 for more information.

Step 4: Apply power and observe LEDS

The RF3570 adapter has a set of LEDs on its faceplate that will indicate its state on power-up. You should observe these LEDs and other indicators that are described in this section. If the LEDs indicate a problem, you may need to follow some of the troubleshooting hints also described.

See page 3-16 for more information.

Step 5: Configure and connect SCSI devices

There are several types of SCSI devices that you may be attaching to the RF3570 adapter. Each requires set-up and each must be cabled to the SCSI bus. There are restrictions on the length of the bus cable and termination factors that must be considered.

See page 3-18 for more information.

Step 1: Unpacking the Board

The RF3570 adapter is shipped to you in one box. You should find the following items included in the shipment:

1 RF3570 adapter board

1 Rimfire 3570 SCSI Host Bus Adapter User's Guide

Shipping Damage

Immediately after receiving it you should check your shipment for evidence of damage or mishandling in transit.

Ciprico's warranty does not cover shipping damages. Therefore, if the shipping carton is water stained or otherwise damaged, contact the carrier or shipping agent with information on the damage.

For repair or replacement of any Ciprico product damaged in transit call Ciprico for return instructions and authorization.

Unpacking the Board

Before you remove the RF3570 adapter from its box you should observe precautions to prevent damage by static: wear a wrist band and strap while handling the adapter.

Step 2: Jumpers and Termination

There are eight blocks of jumpers on the RF3570 board. Three of the blocks should remain factory set. You will need to consider each of the other five blocks to determine whether your application requires changes from the default settings.

The location of the blocks on the board are shown below:

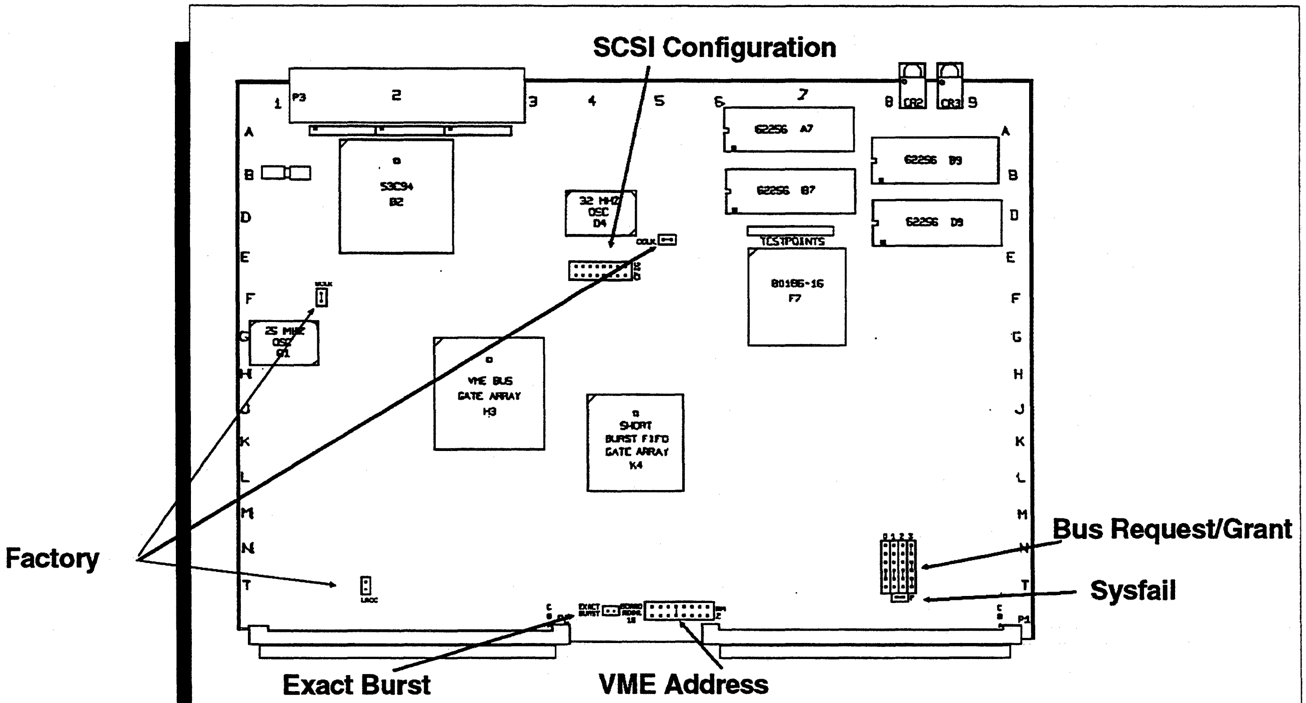


Figure 3-1: Locations of Rimfire 3570 Jumpers

Each of the jumper blocks shown above will be discussed in the following pages.

Factory-set Jumpers

These jumpers should be left as set by the Ciprico factory. They are used by the technicians who test the board before shipping; if you change the setting of any of these jumpers the board will not work.

SCSI Configuration Jumpers

This block of jumpers is used to select power-up defaults for:

- Adapter SCSI ID
- SCSI Bus reset on power-up and system reset
- SCSI Bus Parity

The jumper block is illustrated below:

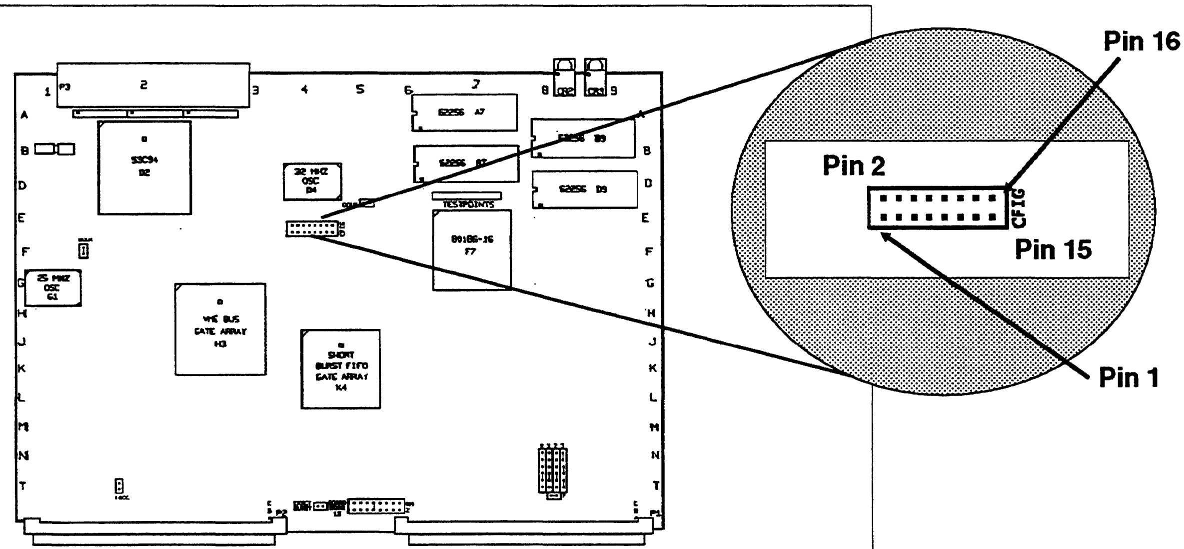


Figure 3-2: SCSI Configuration Jumper Block

The default settings of the jumper block are shown in Figure 3-2 and Table 3-1.

Pins	Description	Default
1 - 2	Reserved	OUT
3 - 4		OUT
5 - 6		OUT
7 - 8	Enable SCSI Parity Checking	OUT = Selected
9 - 10	SCSI bus Reset at Power-up	OUT =Selected
11 - 12	SCSI ID Bit 2	OUT = 0
13 - 14	SCSI ID Bit 1	OUT = 0
15 - 16	SCSI ID Bit 0	OUT = 0

Table 3-1: Default Jumper Settings

Bus Request/Grant Jumper

This jumper block is used to select the level at which the RF3570 adapter can request and be granted access to the VMEbus by the ARBITER. The four sets of BUS/GNT signals are daisy-chained separately on the VMEbus backplane. Only one of the four levels can be used by a single REQUESTER. The BUS/GNT signals that are not used are passed through to the next board on the backplane. The BUS/GNT signal will be gated on the adapter for whichever level you choose. If the RF3570 adapter has a request for the bus pending, it will take control of the VMEbus. If it has no request pending it will pass the BUS/GNT signal to the next board. The Bus Request/Grant jumper block is illustrated below.

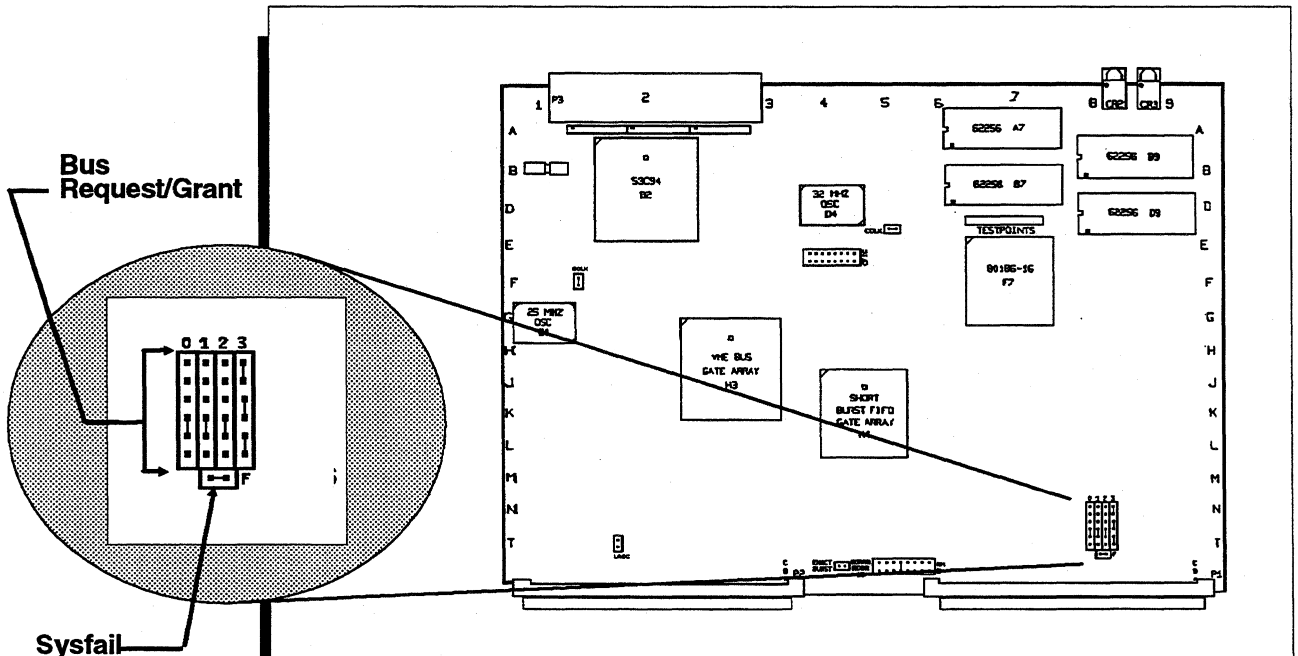


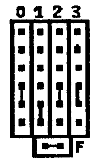
Figure 3-3: Bus Request/Grant and Sysfail Jumpers

The setting shown in Figure 3-3 is the default from the factory. It indicates level 3 for Bus Grant/Request.

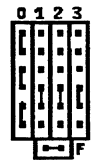
To select a Request/Grant level, place three jumpers down the level you wish to use. The unused levels should have jumpers placed from pin 4 to 5.

Bus Request and Grant on Level 2

Bus Request and Grant on Level 1 ⟶



Bus Request and Grant on Level 0 ⟶



Sysfail Jumper

The VMEbus Sysfail signal is (according to standards) to be asserted during Power-on Self-test to indicate Self-test is occurring. Some systems instead interpret the signal as a failure. For this reason, the RF3570 adapter jumpers the signal on the board.

IN Sysfail signal asserted on bus during Self-test.

OUT Sysfail signal not asserted on bus.

Figure 3-3 on the previous page illustrates the location of the Sysfail Jumper, underneath the Bus Request/Grant Jumpers. As shown, the default for this jumper is IN.

Exact Burst Jumper

With this jumper you can choose one of two ways for the RF3570 to perform DMA operations:

IN This setting will minimize the number of times that the RF3570 arbitrates for control of the VMEbus. The adapter will not initiate a VMEbus DMA operation until ready to transfer a complete burst of data (based on the bus Throttle Count setting in the General Options command as described in Chapter 6). The RF3570 *will* initiate a DMA transfer of less than a complete burst of data only when it is finishing a transfer that was not an even multiple of complete bursts.

OUT This setting usually provides the highest level of performance. In this mode the RF3570 will intitiate a VMEbus DMA operation as soon as it is ready to transfer any data.

Default for this jumper is OUT. Refer to Figure 3-4 for location of this jumper block.

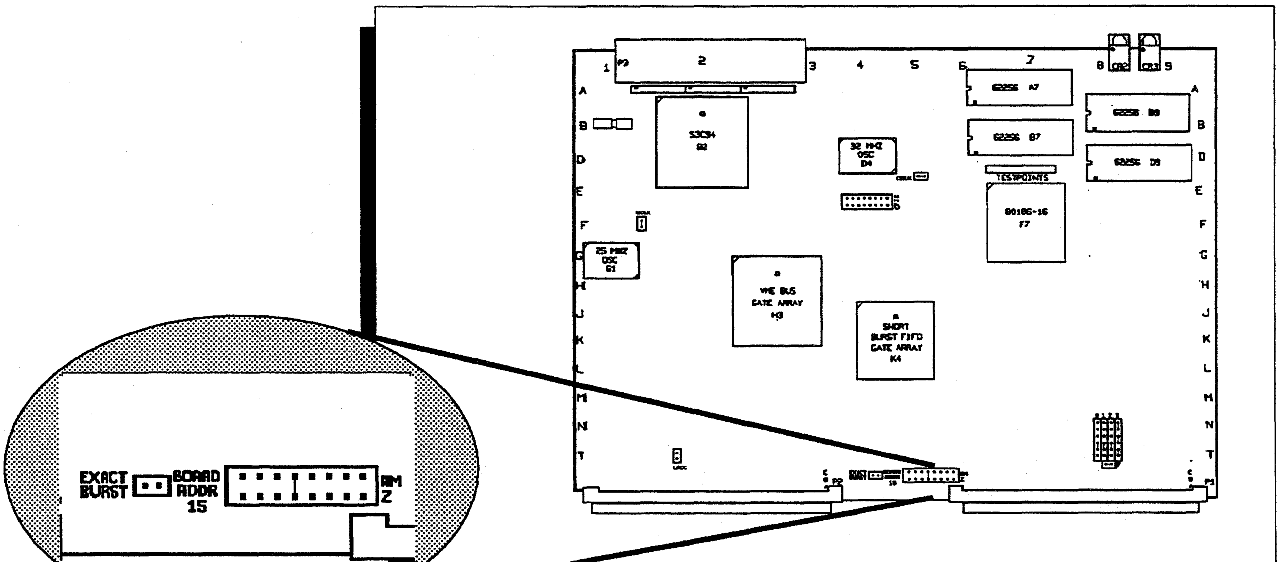


Figure 3-4: Exact Burst and VMEbus Address Jumpers

VMEbus Address Jumpers

This set of jumpers is used to select the address at which the RF3570 adapter can be addressed by the system. The jumpers set the upper 7 bits of the 16-bit address, thereby reserving 512 bytes of address space.

Also available to be set is bit 2 of the address modifier lines. With this jumper *in* you can elect to have the RF3570 adapter reserved for short supervisory access only. If the jumper is *out*, the board will be in short non-privileged access mode.

The VMEbus Address jumpers are illustrated in Fig. 3-4.

The default address setting of EE00H, with address modifier of 2DH is shown.

For the VMEbus Address Jumpers:

IN Is equal to a 0 for that address bit.

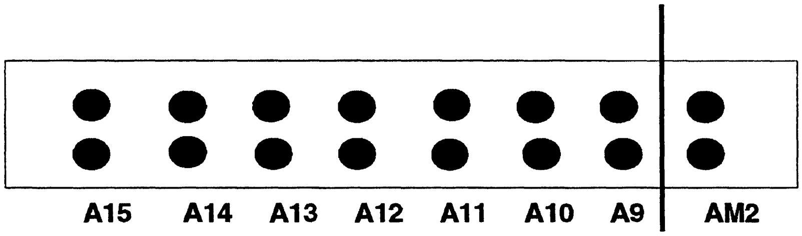
OUT Is equal to a 1 for that address bit.

For the Address Modifier Jumper:

IN Is equal to Address Modifier 29H.

OUT Is equal to Address Modifier of 2DH.

The VMEbus Address Jumper block pins correspond to the VMEbus address lines as follows:



Example Addresses

Below is a table of some common RF3570 Base Addresses. The table demonstrates the settings of the VMEbus Address Jumpers and the VMEbus Address Modifier Jumper for each of the addresses.

VMEbus Address Lines								Base Address	Address Modifier
A15	A14	A13	A12	A11	A10	A9	AM2		
Out	Out	Out	In	Out	Out	Out	Out	EE00H	2DH
Out	Out	Out	In	Out	Out	Out	In	EE00H	29H
Out	Out	In	In	Out	In	Out	Out	CA00H	2DH
Out	In	Out	In	In	In	In	In	A000H	29H
In	Out	In	Out	In	In	In	Out	5000H	2DH
In	Out	In	In	In	In	In	In	4000H	29H
In	In	Out	Out	Out	Out	Out	Out	3E00H	2DH
In	In	Out	Out	Out	Out	In	In	3C00H	29H
In	In	Out	In	Out	In	Out	Out	2A00H	2DH
In	In	Out	In	Out	In	In	In	2800H	29H

Table 3-2: Example Base Address Jumper Settings

SCSI bus Termination

Each end of the SCSI bus cable must be terminated to prevent corruption of the data and control signals. The RF3570 adapter has terminator packs on the board. These are removable in the case that the adapter is not at the end of the SCSI cable.

The adapter provides termination power on the SCSI bus cable. This line is fused on the board. The fuse is replaceable.

The location of the terminators and removable termination power fuse are shown in the following illustrations. The number of terminators depends on whether the board is single-ended or differential.

Figure 3-5 illustrates a single-ended board.

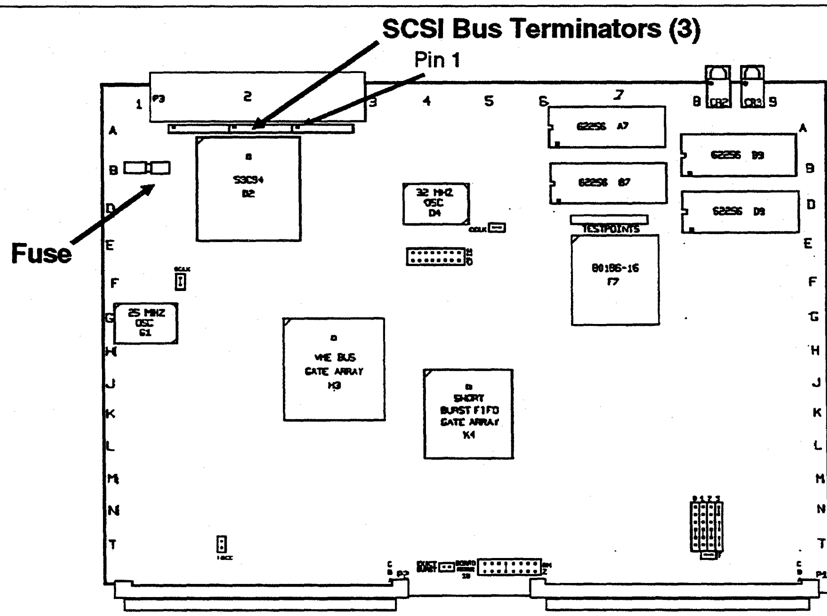


Figure 3-5: Single-ended Board Fuse and Terminators

Figure 3-6 illustrates the terminators and fuse on a differential board.

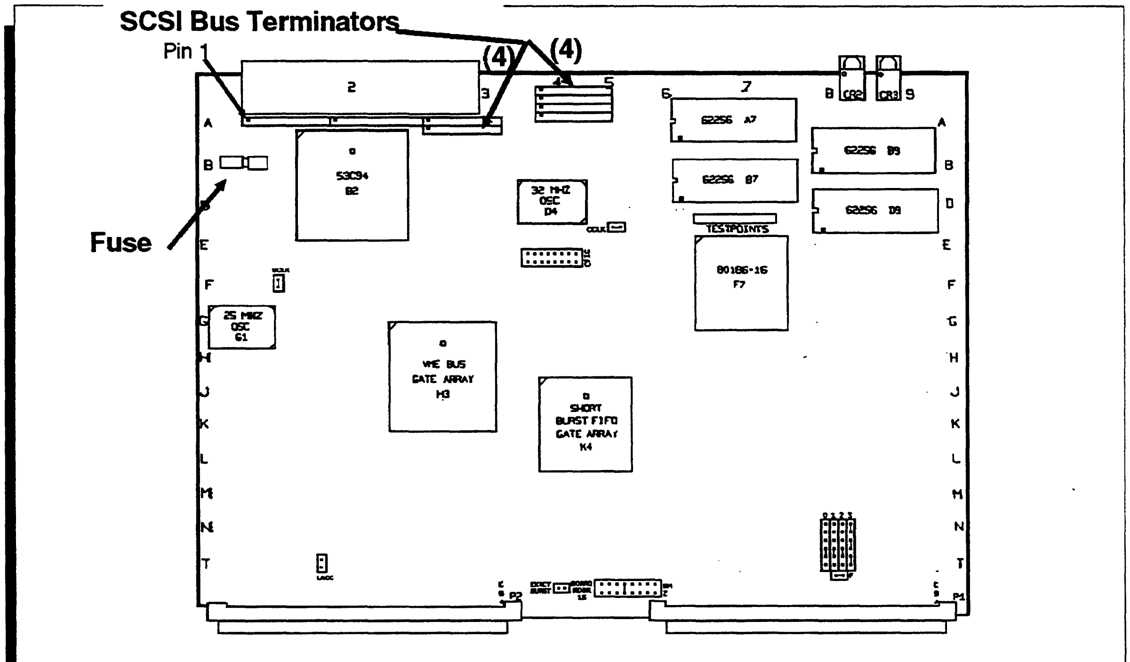


Figure 3-6: Terminators and Fuse on Differential Board

Replacement Fuse

In the event that the termination power fuse is blown, replace it with:

Littlefuse P/N 273-01.5 or equivalent

NOTE:

It is very important to replace the termination fuse with an exact equivalent. If you replace the termination fuse with a value lower or higher, you risk continued problems or damage to the RF3570 and attached equipment.

Step 3: Board Insertion

Although the VMEbus Specification has very precise mechanical and electrical requirements for the subrack, it still leaves room for variance in the design of the system enclosure. You will need to examine the enclosure that you will be working with to determine how to get access to the subrack.

General Steps

In most cases you will need to follow these steps to insert the board:

1. Remove power from the system.

2. Remove as much of the enclosure shell as is necessary to adequately access the VMEbus subrack.

3. Choose a free double-height slot for the RF3570.

4. Remove bus grant arbitration and interrupt acknowledge jumpers from backplane. You will need to remove the four Bus Grant jumpers, and the Interrupt Acknowledge daisy-chain jumper from the slot chosen for the RF3570. These jumpers must be installed only for slots that are empty.

5. Align the RF3570 with the board guides on the subrack for the slot you have chosen. Be sure the orientation is correct. (If J1 on the subrack backplane is up, the component side of the board should face right.)

6. Slide the board forward and firmly press into the J1 and J2 connectors on the backplane. You should feel it give slightly as you press inward.

7. If your system subrack accommodates them, fasten the small screws on the RF3570 into the subrack. You will need a small flat bladed screwdriver.

The illustration on the next page demonstrates insertion of the RF3570 into a VMEbus subrack. Do not reinstall the enclosure panels yet. You will need to observe the faceplate LEDs in the next step.

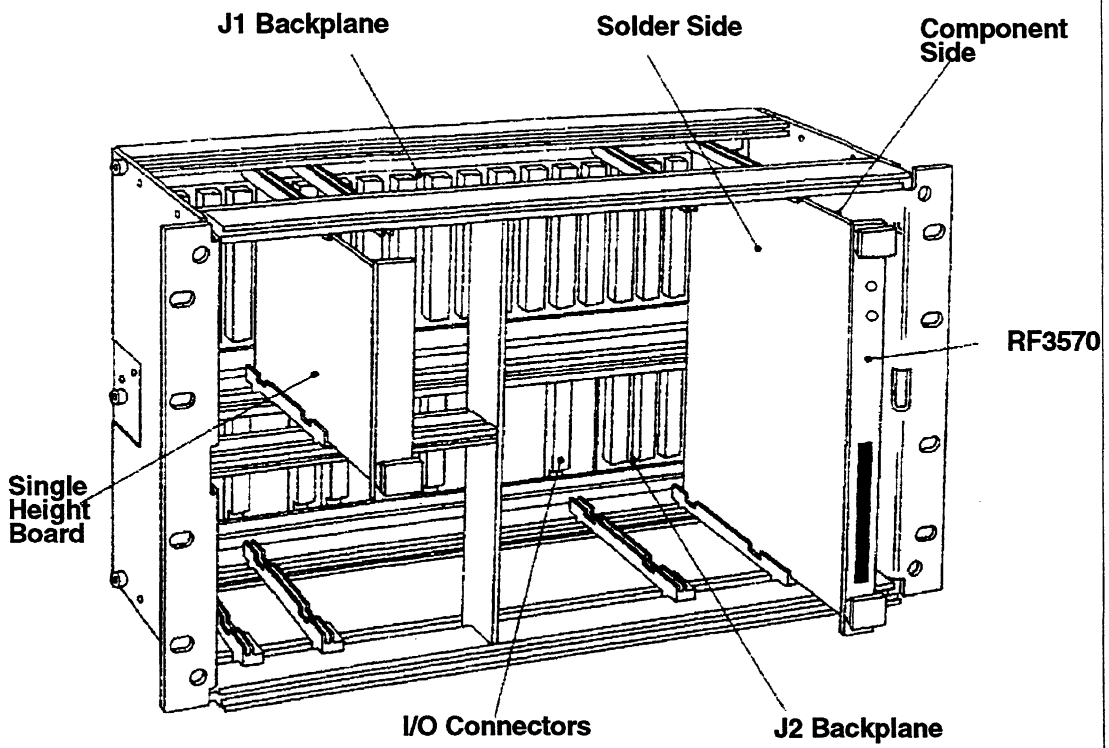


Figure 3-7: Inserting the RF3570

Step 4: Apply power and observe LEDs

After you have inserted the board into the VMEbus slot, apply power to the system.

There are two LEDs on the faceplate of the RF3570: one is green and the other is amber.

Meaning of LED's

The LED's are used in the following manner:

- The green LED indicates BUSY status when the board is processing commands.
- The amber LED indicates an ERROR status.

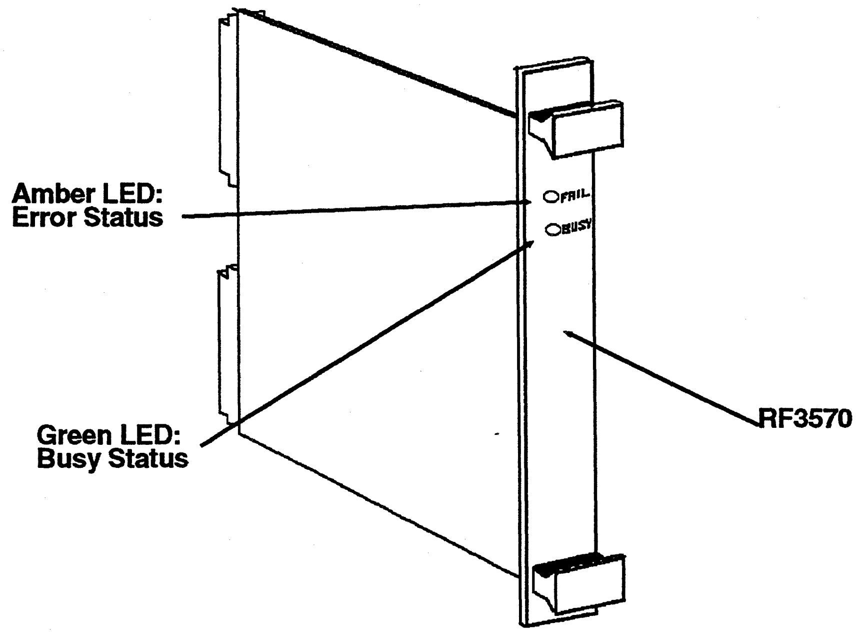


Figure 3-8: Location of LEDs

Power-up Sequence

At a normal power-up you should observe the following:

1. Both LEDs will briefly light, then the green will go out.
2. The amber will stay lit for less than 1 second, while self-test is running.
3. The amber LED will go out, indicating self-test has passed.

Error Conditions

The amber LED is used to indicate an error condition. These are the conditions indicated by the amber LED:

BLINKS

This indicates that the adapter has had a self-test failure.

STAYS ON SOLIDLY

This indicates that the board has had a catastrophic error during operation such as bus error or timeout. The Status port will contain an error code that details the type of failure that occurred. These error codes are:

Code	Name	Description
10H	**Bad PBIN Value**	The Command List PBIN (Parameter Block IN) value exceeded the value set for PBNUM (Parameter Block Number).
11H	**State wrong for Start/Stop Command List command**	This code is returned under these conditions: **Start Command List** command issued when a command list is already active. **Stop Command List** command issued when no command list is presently active. **Command List Channel Attention** issued when no Command List is active.
14H	**Software VMEbus Timeout**	The adapter times each of its VMEbus transfers; it will report this error if a VMEbus transfer takes too long to complete. This can occur for transfers of parameters as well as transfers of data.
15H	**VMEbus Error Occurred**	A VMEbus error was detected while the adapter was transferring either parameters or data.
96H	**Internal Firmware Error**	This error code indicates that a Firmware error was detected during the execution of a command.

Table 3-3: Status Port Catastrophic Error Codes

Step 5: Configure and Connect SCSI Devices

The RF3570 can communicate with any SCSI device that adheres to the SCSI Common Command Set. Each separate device on the SCSI bus *must* have a unique SCSI ID and be properly terminated. In addition, it may have a self-test sequence or options and parameters that you should know about.

You will need to examine the documentation that arrives with your selected peripheral for more information. This section outlines what to look for.

Selectable Options

Each peripheral vendor can choose a unique way to assign a SCSI ID and enable/disable options. The two most common methods are: jumpers and switches.

You will need to find the section in your peripheral documentation that explains how to set the options.

SCSI ID

You will need to decide which SCSI ID to assign to your peripheral(s). The higher SCSI ID's are given bus access priority. Each device on the SCSI bus (including the RF3570) must have a unique SCSI ID. The factory-set ID for the RF3570 adapter is zero.

Options

Your device may also have a means to select options such as: termination power source, automatic spin-up, or parity checking.

Termination Power: If your hard disk is the last peripheral on the SCSI bus, it will need to have terminators installed. Some hard drives allow you to select whether the terminators will get power locally, on the drive, or use the 5 volts available on the SCSI cable. Setting up the drive to provide it's own terminator power is the most universal solution. The RF3570 provides TERM PWR on the SCSI cable.

Automatic Spin-up: Some disk drives allow you to delay spin-up of the disk media until a SCSI command is issued to spin it up. The advantage to this is that it cuts down the power draw when the system is initially turned on.

Parity Checking: The RF3570 is flexible enough to allow you to use SCSI bus parity checking or disable it. If you wish to use it, parity checking must be enabled on each peripheral and also using the General Options command explained in Chapter 6. Note that all devices on the SCSI bus must be setup for parity the same.

You can select a power-up/reset *default* for parity checking with pins 7 and 8 of the SCSI Configuration Jumper Block. The value you use in the General Options command overrides the jumper setting, so you should be sure that they match. (You would install a jumper over the pins to *disable* parity checking.) The jumpers are further described on page 3-7. Moreover, you can use the Board Information Board-control command to discover the current setting of the parity jumper without removing the board. See Chapter 6 for more information on Board-control commands.

Termination

The devices at each physical end of the SCSI bus cable must have terminating resistors installed. If termination is not correctly applied to the bus, data corruption can occur. If you are using only two devices on the bus (the RF3570 and one peripheral) both must be terminated. If you are using multiple devices on the SCSI bus (several peripherals, in addition to the RF3570 adapter) only the two devices physically at the ends of the cable must be terminated.

Cabling

The SCSI cable connector for the RF3570 extends from the front faceplate. The type of cable you use and the connector used on the RF3570 depends on the model.

Model	SCSI mode	Max Cable Length	SCSI Connector
RF3573	Single-ended	6 meters (20 ft.)	50 pin, ribbon
RF3574	Differential	25 meters (82 ft.)	50 pin, ribbon
RF3575	Single-ended	6 meters (20 ft.)	50 pin, high density
RF3576	Differential	25 meters (82 ft.)	50 pin, high density

Chapter 4 Command Operation

Chapter 4 — Command Operation

Introduction

The purpose of this chapter is to provide you with a comprehensive view of the RF3570 firmware operation and how it works in concert with the hardware described in Chapter 2. By explaining the command structures you will use, and when to use them, this chapter teaches you how to use the board; it explains in broad terms what a board driver must do.

Summary

This chapter discusses the following:

- Format of the basic Parameter and Status blocks from which command structures are built.
- Types of command structures — Single Command and Command List.
- The process for issuing a single command.
- The process for creating and starting a command list.
- Examples of issuing both types of command structures.
- Implementation of the command structures within a 'C' routine.
- Types of commands you can issue — SCSI Pass-through and adapter Board-control commands.
- When to use each type of command.

The explanations in this chapter would be most beneficial for someone designing a driver for the RF3570 adapter or attempting to understand one existent for the RF3500 family.

What the Adapter must accomplish

There are several basic operational requirements that the RF3570 adapter must fulfill:

The Host must have a method to communicate details of command operation with the adapter — Communication from the host to the RF3570 adapter takes the form of command structures. There are two types of command structures you can use with the RF3570 adapter: Single Command and Command List. The latter is preferable in most situations because it boosts the performance of the adapter. There are instances when you *must* use the single command method of communication. Both types of structures are described in the following sections.

The adapter must have a means to communicate the outcome of a command process — Reporting the completion status of the command is accomplished via status blocks which are part of the command structure. Just as there are two types of RF3570 command structures for you to use when issuing a command, there are two ways status structures are returned from the adapter. Each uses slightly different protocol to notify the Host of returned status. This is described in more detail in a later section.

The Host must have a way to alter the behavior of the adapter to suit its environment — To meet the versatility required by differing host operating systems and application needs, the RF3570 adapter has a set of board level commands for you to use. These commands allow you to select options relating to the system bus of the host and also tailor operation for the devices attached to the board. They are referred to as **Board-control** commands.

The Host must have a defined protocol for issuing a command to the device(s) attached to the adapter — The RF3570 adapter is an interface between the VMEbus and SCSI bus architectures. When the VMEbus Host requests an operation involving a SCSI device, the RF3570 issues a command to the device using the SCSI protocol. It uses the defined SCSI command structure within its own parameter-block command structure. This allows the RF3570 to pass the command to the SCSI device without modification. This type of command is called a **Pass-through** command.

The adapter must provide status from a device regarding the command just completed by the device — Just as the defined SCSI command structure is part of the RF3570 command structure and is passed through to the device, the Sense Bytes from the SCSI device are reported to the host within the Status Block returned by the adapter.

Each of these operational requirements are discussed in more detail on the following pages.

The Base Parameter Block

The Parameter Block is one of the basic building blocks of the RF3570 command structures. It is designed to hold all facts relevant to a single command.

The same Parameter Block format is used whether the command is a **Pass-through** command directed to a SCSI device, or a **Board-control** command to be interpreted and acted on by the adapter. (The various flags and byte values may have different meanings in each case though.)

All Board-control commands are described in detail in Chapter 6. More information about how the RF3570 implements SCSI commands can be found in Chapter 5. See a SCSI specification and vendor-supplied peripheral documentation for details about particular SCSI commands.

Parameter Block Format

The Parameter Block format for all SCSI Pass-through and also adapter Board-control commands is illustrated below:

Address Offset	Byte Memory Address			
	Offset + 0	Offset + 1	Offset + 2	Offset + 3
00H	Command Identifier			
04H	Command Options, Addresses, and Transfer Count			Target/Adapter ID
08H				
0CH				
10H	Command	Command Descriptor Block		
14H				
18H				

Figure 4-1: Base Parameter Block

A brief description of each of the fields follows:

Command Identifier

When setting up a command structure, you use this field to distinguish a command in order to associate it with a returned Status Block. The value is a double-word and must be unique; the **Command Identifier** for each currently executing command should be different. Any unique four-byte value will work. You could, for example, use the command address or a Parameter Block Index pointer as the **Command Identifier**. Both SCSI Pass-through commands and adapter Board-control commands use this field for proper identification of status to be returned whether the command is issued in a Single Command structure or a Command List.

Command Options, Addresses, and Transfer Count

You will use these fields of the Parameter Block to supply an address and transfer count when a data transfer is required by the command you wish to issue. There are also some command-modifying options available to you depending on the command you are issuing.

Target/Adapter ID

The Target/Adapter ID byte is very important to the command issue process. The value you supply here determines whether the command described in the **Command Descriptor Block** is a SCSI Pass-through command or an adapter Board-control command:

- **If the Target = FFH**, the command is a Board-control command.
- **If the Target = 0-7H**, the command is destined for the SCSI target with that ID.

Command Descriptor Block

In this part of the Parameter Block you must supply a Command Code and details about how you want the command executed. Depending on the value you supply in the Target/Adapter ID field, the **Command Descriptor Block** will contain a SCSI command, or an adapter Board-control command. The RF3570 Board-control commands are explained in detail in Chapter 6. You can find more information about issuing SCSI Pass-through commands later in this chapter and also in Chapter 5. For detail about the actual SCSI commands, see a SCSI specification or your vendor-supplied peripheral documentation.

Command Code

This field will always contain a command code for either a SCSI Pass-through or an adapter Board-control command.

The Base Status Block

The Status Block is another of the building blocks of the command structures you will be using when issuing commands. Used in conjunction with the Parameter Block, its purpose is to report the outcome of a command you issued.

The Status Block format is the same whether you have issued a SCSI Pass-through command or a Board-control command. It appears the same whether the command structure you used is a Single Command or a Command List, however there is a difference in the *number* of Status Blocks you may receive for each type of command structure.

Single Command structures return only one Status Block. The Single Command structure will contain only the last Status Block — the Status Block with the CC (Command Complete) bit of the Flags byte set.

When you are using a Command List structure, multiple Status Blocks for any one command are possible depending on the retry and Sense Byte selections you have made (with the Unit Options or Extended Unit Options Board-control commands).

Status Block Format

The Status Block Format you will see used for all returned status information is illustrated in Figure 4-2:

The fields displayed in light type are the default Sense Bytes returned by SCSI Pass-through commands.

Address Offset	Byte Memory Offset			
	Offset + 0	Offset + 1	Offset + 2	Offset + 3
00H	Command Identifier			
04H	Reserved	SCSI Status	Error	Flags
08H	0 = Class/Code	1 = Segment	2 = SCSI Flags	3 = Info Bytes
0CH	4 = Info Byte	5 = Info Byte	6 = Info Byte	7 = Ex Length

SCSI Sense Bytes

Figure 4-2: Base Status Block

Following is a brief description of each of the fields:

Command Identifier

This value will identify which Parameter Block has completed.

Flags

This byte can be polled to determine whether, and in what condition, a command completed. It has this format:

BITS							
7	6	5	4	3	2	1	0
CC	ERR	RTY	DTT	0	CSB	0	0

Figure 4-3: Bit-meanings of Flags Field

CSB **Continued Status Block**
This bit indicates that the current Status Block is one of several Status Blocks generated because extended Sense Bytes are enabled. Only valid for Pass-through commands issued through Command List.
0 ⇒ Status Block is *first (or only)* Status Block.
1 ⇒ Status Block is one of a series of Status Blocks.

DTT **Data Transfer Truncated**
Indicates when a SCSI Pass-through command completes with fewer bytes transferred than requested.
0 ⇒ Data transfer was *not* truncated.
1 ⇒ Data transfer was truncated.

RTY **Retry**
Indicates whether a SCSI Pass-through command required retries to complete.
0 ⇒ No retries were required.
1 ⇒ Retries were required to complete.

ERR **Error Status**
Identifies when a Board-control or SCSI Pass-through command completes with an error.
0 ⇒ No error ocurred.
1 ⇒ An error occurred.

CC **Command Complete**
Indicates whether a Board-control or SCSI Pass-through command has finished.
0 ⇒ Command not complete.
1 ⇒ Command complete.

Error

When an adapter-detected error occurs, this byte contains an error code to describe the condition. Error Codes are described in Appendix A.

SCSI Status

When a SCSI device encounters an error it returns to the adapter a SCSI Status Byte Code. The adapter reports that value here. See **Chapter 5,** Figure 5-10 and Table 5-3 and either a SCSI specification or your vendor-supplied peripheral documentation for a complete description.

SCSI Sense Bytes

Most of the **Board-control** command Status Blocks return all zeroes in these fields. **SCSI Pass-through** command Status Blocks return Sense Byte data in these fields. If a SCSI error occurs, these eight bytes of sense data help you determine the nature of the error. When the RF3570 adapter receives a Check Condition status from a SCSI device, the adapter may issue a Request Sense command in response. The device then transfers up to 256 bytes of sense data to the adapter. The sense data contains information about the error which occurred. From this data eight bytes are returned in the status block. The default sense bytes returned are 0 - 7, but through use of the Unit Options Command or the Extended Unit Options Command , you may select up to 16 of the possible 256 bytes of Sense Data. More about the use of and difference between the Unit Option Command and Extended Unit Option Command is available in Chapter 6. If no SCSI error occurs, all bytes of Sense data in the Status Block will be zero.

Multiple Status Blocks

It is possible to receive more than one Status Block to indicate the outcome of a command. To receive multiple Status Blocks, all the following must be true:

- You have used the Unit Options or Extended Unit Options Board-control command to set up retries and/or extended Sense data for Pass-through commands.
- The command is a Pass-through command. (Board-control commands are not subject to retries and do not return Sense Data.)

- The Pass-through command is issued through use of the Command List structure. (Single command structures can only contain one Status Block.)

Setting up Unit Options

These are the options selected through the Unit Options and Extended Unit Options Board-control commands that cause the RF3570 to return multiple Status Blocks in response to an error condition:

Retries — If you 1.) supply a Retry Limit value, 2.) select a type of retry in the Retry Control field, *and also* 3.) set the ISB bit of the same field, each command retry performed will generate at least one Status Block.

Sense Data — If you select a Sense Count greater than eight, or specify over eight Selected Sense Bytes, you will receive multiple Status Blocks for each Request Sense command sent in response to a Check Condition.

NOTE:

If you enable both Retries and *over eight bytes of Sense Data, you will receive* multiple *Status Blocks for every retry performed until command completes successfully or Retry Limit is reached. This can be calculated as follows:*

For a command that exhausts Retry Limit:

# of Status Blocks = (# of retries +1) x (# of Status Blocks needed for Sense Data)

For a command that recovers before Retry Limit is reached:

# of Status Blocks = ((# of attempts + 1) x # of Status Blocks needed for Sense Data) + 1

Single Command Structure

There are two ways to issue a command to the RF3570 adapter. The simplest method involves the use of the **Single Command** structure. This 52-byte structure combines a Parameter Block and a Status Block into a single structure along with a field that specifies an interrupt level and ID. This Single Command structure is shown in Figure 4-4. The important things to remember about the Single Command structure are:

- It is designed to be used for initial set-up.
- The first command you issue to the board after power-up or reset *must* use this structure.

Format of Single Command Structure

The Single Command structure is illustrated for you below:

Address Offset	Byte Memory Address			
	Offset + 0	Offset + 1	Offset + 2	Offset + 3
00H	**Parameter Block** *(detail for Pass-through commands on page 5-2, see Chapter 6 for detail on individual Board-control commands.)*			
04H				
08H				
0CH				
10H				
14H				
18H				
1CH	Reserved		Interrupt	
20H	Reserved			
24H	**Status Block** *(detail for Pass-through status on page 5-16, see Chapter 6 for detail on individual Board-control command status.)*			
28H				
2CH				
30H				

Figure 4-4: Single Command Structure

Parameter Block

This section of the Single Command structure is where you build a Parameter Block containing information related to the command you wish to execute: Pass-through or Board-control.

Most of the fields are command-specific; if there is data to transfer, for example, an address, address modifier and transfer count must be provided. For details about the parameter block fields necessary for a SCSI Pass-through command refer to **Chapter 5,** SCSI Pass-through Commands and a SCSI specification. For details about a Board-control command see **Chapter 6,** Board-control Commands.

Reserved Fields

Reserved fields must be zeroed. If a non-zero value is permitted in a Reserved block, you may get an error.

Interrupt

This field selects the Interrupt level which the RF3570 adapter will use when the command issued through the Single Command structure completes. You can also denote the resultant interrupt acknowledge ID that the adapter will place on the data bus in response to the Host.

BITS															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	INT			ID							

Figure 4-5: Bit-meanings of Interrupt Field

The Interrupt field uses these bit-meanings:

ID — **Status/ID**
This byte denotes the Status/ID value that the adapter will provide to identify the interrupt being acknowledged by the Host.

INT — **Interrupt Level**
You may elect to use any one of the following levels for adapter interrupt:
000 — Interrupt disabled
001 — Interrupt level 1
010 — Interrupt level 2
011 — Interrupt level 3
100 — Interrupt level 4
101 — Interrupt level 5
110 — Interrupt level 6
111 — Interrupt level 7

Status Block

The Status Block is the means by which the RF3570 adapter communicates successful command completion, error conditions, retry information or SCSI Sense Bytes. Status Block information specific to the type of command you issued can be found in Chapters 5 and 6.

If the status from a command issued within a Single Command structure requires multiple Status Blocks to report the status, only the last block is saved in the structure.

NOTE: *If you have set the ISB (Issue Status Block) bit in the Retry Control field of the Unit Options or Extended Unit Options command, it will be ignored for commands issued through Single Command structure.*

Using a Single Command Structure

The Single Command structure is the fundamental means by which you can issue a command to the RF3570 adapter board. To issue a command you must perform each of these steps in the order described.

Set-up the Structure in memory

1.) Clear 52 bytes of system memory at the location you wish to create the Single Command structure.

2.) Build a Parameter Block in bytes 0 through 28. Refer to Chapter 5 for more information about SCSI Pass-through commands. Refer to Chapter 6 if you wish to issue a Board-control command like Start Command List.

3.) Assign a value to the Interrupt field of the Single Command structure, reflecting the interrupt the adapter should use and the ID by which it will identify the interrupt being acknowledged by the Host. The Reserved fields must be zero.

Prepare the Hardware Ports

4.) If you are going to use this command to set bus width and byte/word swapping, determine the appropriate value for the Control Field of the Address Buffer Port.

5.) Write the Control Field, Address Modifier and the two address words of the Single Command structure to the Address Buffer Port on the adapter.

6.) Read the Status Port. Test the RDY bit (bit 1) to confirm that the board is running. Next, verify that bit 0 (ENT) indicates the board is ready for a Single Command. (On reset or power-up, this bit is set to zero. Thereafter it toggles from its previous value to indicate readiness.)

Execute the Command

7.) Write a 0 value to the Channel Attention Port.

Determine Command Completion

8.) When the command is complete, the adapter will write a Status Block in bytes 36 through 52 of the Single Command structure and generate any interrupt you requested. If you did not specify an interrupt level in the structure, you must poll the CC bit of the Flags byte of the Status Block to determine completion. Poll the ENT bit of the Status Port to determine when you can issue another Single Command. Command pre-fetch allows you to send another command to the adapter before the previous is completed. *Therefore, the CC and ENT bits are not equivalent.*

Some Example Single Commands

The following examples will help you understand how to use the Single Command structure. Two examples are given here. The first is an example of a **Board-control** command. The second shows you how to issue a SCSI **Pass-through** command to the RF3570 adapter.

Example 1: Start Command List Board-control command

The **Start Command List** Board-control command sets up a Command List structure for you to use for subsequent commands. (Basically, the Command List uses a location in memory as a depot for multiple commands and their Status Blocks. This feature is discussed in detail later in this chapter.)

This example shows you how to use the Single Command Structure to initiate the Command List.

1.) Select a portion of system memory for the Single Command structure. You will need 52-bytes of space to hold the Parameter Block, Interrupt and Reserved fields, and the Status Block that make up a Single Command structure.

Example Memory Address = 822F4H

2.) Fill in the Parameter Block information. (Detailed Parameter Block information for *each* Board-control command is given in Chapter 6.)

These are the Parameter Block values for this example:

Address Offset	Byte Memory Address			
	Offset + 0	Offset + 1	Offset + 2	Offset + 3
00H	Command Identifier = 000822F4H			
04H	Reserved		Addr Mod = 3DH	ID = FFH
08H	Command List Memory Address = 00099450H			
0CH	Reserved		Interrupt = 300H	
10H	Command = 01H	Reserved		
14H	Reserved			
18H	Reserved			

Figure 4-6: Example Start Command List PB

Command Identifier = 822F4H
Even though the Status Block is part of the Single Command structure and easily identified it is good to provide a value here in order to be consistent with Command List structures. The value chosen is the address of the Single Command structure.
Address Modifier = 3DH
This value indicates that Standard Supervisory Data Access will be used for Command List data transfers.
ID = FFH
Indicates to the RF3570 adapter that this is a Board- control command.
Command List Memory Address = 99450H
System Memory address of the Command List structure.
Interrupt = 300H
Indicates that an Interrupt Level of 3 be used for Command List status.
Command = 01H
This value is the Start Command List command code.

3.) Fill in the Interrupt Word of the Single Command structure. This is used to indicate the Interrupt Level to be used for the Single Command. In this example we will leave it zero along with the Interrupt Acknowledge ID. This means the Flags byte of the Status Block will need to be cleared before issuing the command, and then the Command Complete (CC) bit polled to determine when this command completes.

In memory, the structure will look like this:

822F4	00	08	22	F4
822F8	00	00	3D	FF
822FC	00	09	94	50
82300	00	00	03	03
82304	01	00	00	00
82308	00	00	00	00
8230C	00	00	00	00
82310	00	00	00	00
82314	00	00	00	00
82318	00	00	00	00
8231C	00	00	00	00
82320	00	00	00	00
82324	00	00	00	00

Parameter Block
Interrupt and Reserved Fields
Status Block

Figure 4-7: Single Command Structure in memory

4.) The next step is to load the Address Buffer Port. This will require three writes to the port address. Each of the three writes is described next:
Control Field = 84H
For the first write, the Control Field, we'll select no byte or

word swapping of data or control structures, but choose to use a 32-bit system bus.
Address Modifier = 3DH
This is the Address Modifier that the adapter is to use when reading the Single Command Structure.
Single Command Structure Address = 822F4H
The other two writes to the port contain the address of the Single Command Structure.

5.) The example board is jumpered at address EE00H. The three writes to the Address Buffer Port at address EE00H will be as follows:
1.) 843DH
2.) 0008H
3.) 22F4H

6.) Read the Hardware Status Port at address EE10H. Since the ENT bit of the Status Port toggles between 0 and 1 with each Single Command Channel Attention issue, the value returned could be 0202H or 0203H depending on the previous value. We'll assume this is the first command after a reset; the initial value of the ENT bit is 0. The adapter is ready if the Status Port reads 0202H.

7.) To execute the command, write a 0 to the Channel Attention Port.

8.) Poll for command completion by reading the Flags Byte of the Status Block and comparing it to zero. When the value is non-zero, the command is complete. The value of the Flags byte in the Status Block will indicate if the command completed with an error. In this example, the command was successful; a value of 80H is returned as shown below. A returned value of C0H would indicate an error had occurred.

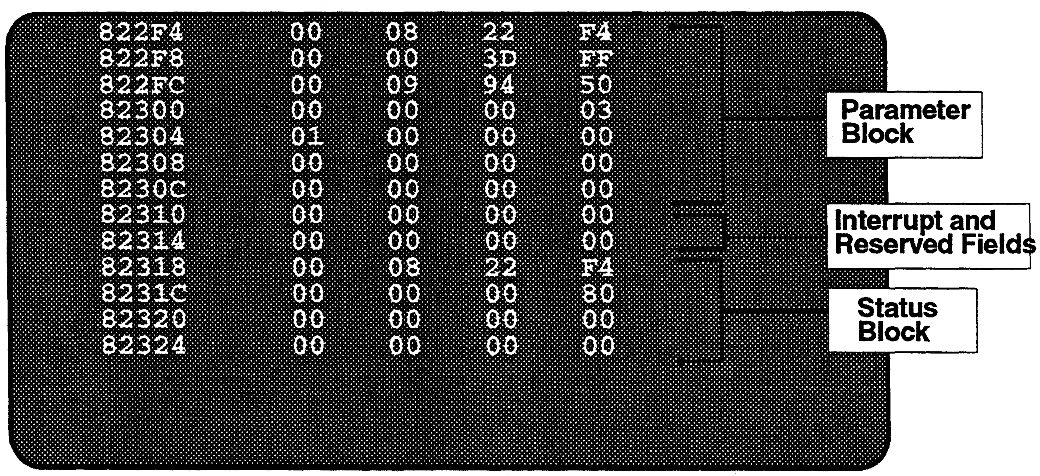


Figure 4-8: Single Command Structure in memory

Example 2: Inquiry Pass-through command

This example describes the issue of a SCSI Pass-through command. The SCSI Inquiry command is fully described in the SCSI specification.

This example shows you how to use the Single Command structure to issue a Pass-through command.

1.) The set-up of a Pass-through command is the same as for a Board-control command. You must select a portion of system memory for the Single Command structure. You will need 52-bytes of space to hold the Parameter Block, Interrupt and Reserved fields, and the Status Block that make up a Single Command structure. For this example we'll use the same memory address.

Example Memory Address = 822F4H

2.) Fill in the Parameter Block information. (Detailed Parameter Block information for Pass-through commands is given in Chapter 5. For information about the SCSI Inquiry command, you must reference a SCSI specification or drive manual.)
These are the Parameter Block values for this example:

Address Offset	Byte Memory Address			
	Offset + 0	Offset + 1	Offset + 2	Offset + 3
00H	Command Identifier = 000822F4H			
04H	Flags-2 = 0	Flags-1 = 0	Addr Mod = 3DH	ID = 01H
08H	VME Memory Address = 00099450H			
0CH	Transfer Count = 24H			
10H	Command = 12H	LUN =0	Reserved = 0	Reserved = 0
14H	Allocation = 24H	Flag/Link = 0	0	0
18H	0	0	0	0

Inquiry 6-byte Command Descriptor Block

Figure 4-9: Example Pass-through command PB

Command Identifier = 822F4H
Even though the Status Block is part of the Single Command structure and easily identified it is good to provide a value here in order to be consistent with Command List structures. The value chosen is the address of the Single Command structure.

ID = 01H
Indicates to the RF3570 adapter that this Pass-through command should be directed to device ID 1.
Address Modifier = 3DH
This value indicates Standard Supervisory Data Access mode will be used to execute the Inquiry command.
Flags-1 = 0
This byte is used to control aspects of command operation. None of the features are enabled for this command.
Flags-2 = 0
This byte offers further command control options. None of the additional features are enabled for this example.
VME Memory Address = 99450H
System Memory address to which the Inquiry data will be returned.
Transfer Count = 24H
This value is the number of bytes of data that will be transferred for this command. The Inquiry data returned by this device is contained in 36 (24H) bytes.
SCSI 6 byte Command Descriptor Block
These six bytes of command information are explained in the SCSI specification.

3.) Fill in the Interrupt Word of the Single command Structure. The Interrupt Word is used for the Single Command Interrupt Level.
In this example we'll use an interrupt level of 3, and designate a Status/ID of 3FH. This means that the adapter will interrupt the Host when the command has completed by driving IRQ3. When requested for its Status/ID during the VME interrupt cycle, the adapter will respond with the value 3FH on lines D0-D7.

In memory, the structure will look like this:

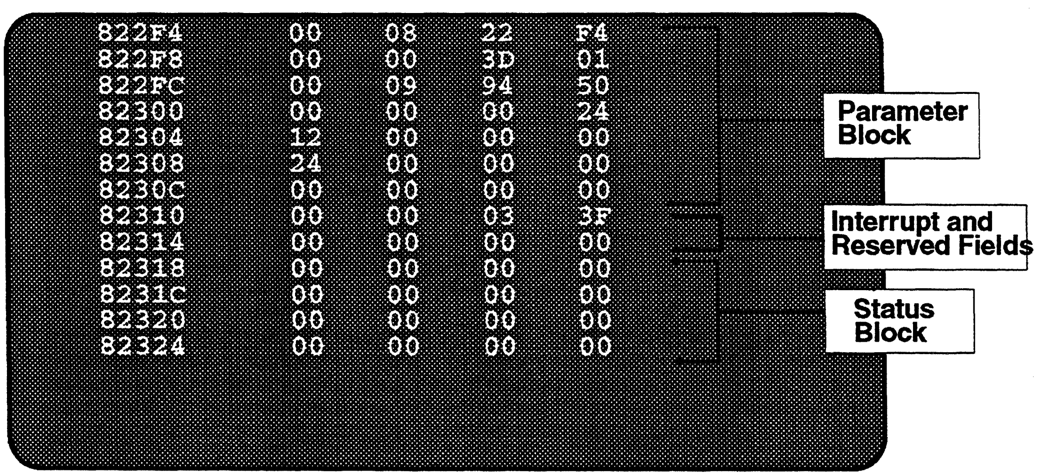


Figure 4-10: Single Command Structure in memory

4.) The next step is to load the Address Buffer Port. This will require three writes to the port address:
Control Field = 84H
For the first write, the Control Field, we'll select no byte or word swapping of data or control structures, but choose to use a 32-bit system bus.
Address Modifier = 3DH
This is the Address Modifier that the adapter is to use when reading the Single Command Structure.
Single Command Structure Address = 822F4H
The other two writes to the port contain the address of the Single Command Structure.

5.) The example board is jumpered at address EE00H. The three writes to the Address Buffer Port at address EE00H will be as follows:
1.) 843DH
2.) 0008H
3.) 22F4H

6.) Read the Hardware Status Port at address EE10H. Since the ENT bit of the Status Port toggles between 0 and 1 with each Single Command Channel Attention issue, the value returned could be 0202H or 0203H depending on the previous value. We'll assume this is the first command after a reset; the initial value of the ENT bit is 0. The adapter is ready if the Status Port reads 0202H.

7.) To execute the command, write a 0 to the Channel Attention Port.

8.) When the command has completed, the Status Block will be written to memory and an interrupt generated by the RF3570 adapter. Read the Flags byte of the Status Block to determine if there are errors, or retries associated with the command.

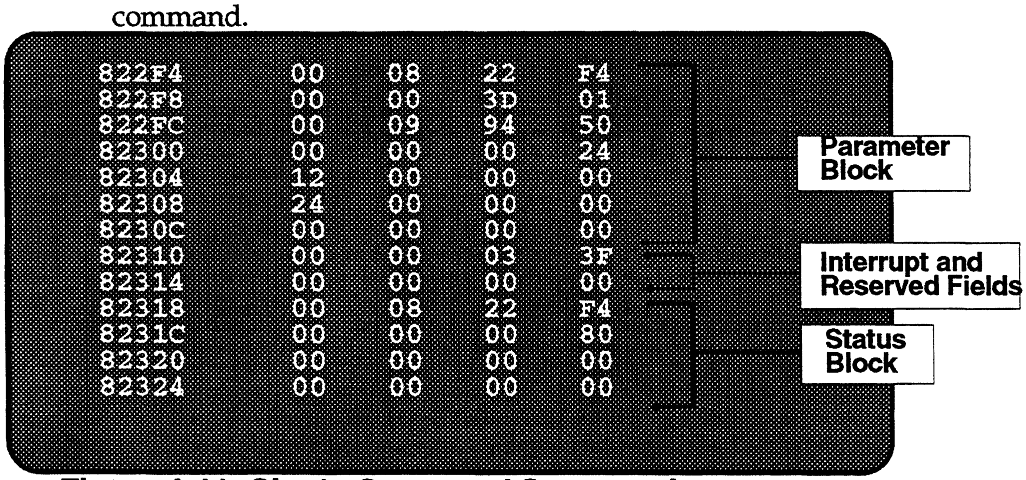


Figure 4-11: Single Command Structure in memory

Command List Structure

The previous pages have discussed the use of a Single Command Structure to issue a command to the RF3570 adapter. This section deals with the other method of operation: the **Command List Structure.** With this structure you are able to queue commands for execution; use of the Command List Structure enhances performance by decreasing command overhead and allowing both system and adapter to perform command I/O at the same time.

Format of Command List Structure

The address of the Command List Structure should be on a double word boundary. The Command List Structure is illustrated for you below:

Address Offset	Byte Memory Address			
	Offset + 0	Offset + 1	Offset + 2	Offset + 3
00H	Parameter Block IN Index			
04H	Parameter Block OUT Index			
08H	Status Block IN Index			
0CH	Status Block OUT Index			
10H	Number of Parameter Blocks *(n)* (max = 2034)			
14H	Number of Status Blocks *(y)* (max = 4096)			
18H	Reserved			
1CH	Reserved			
20H	Parameter Block #1			
3CH	Parameter Block #2			
.	Parameter Blocks (up to #2034)			
.	Status Block #1			
.	Status Block #2			
.	Status Blocks (up to #4096)			

Figure 4-12: Command List Structure

Description of the Fields

The following section describes the information you will need to properly complete a Command List Structure.

Parameter Block IN Index

This field contains an index into the Parameter Block array. It points to the next open block. When you add a Parameter Block to the Command List Structure, you must increment this number. This value is never changed from the adapter side of operations.

Parameter Block OUT Index

This field also contains an index into the Parameter Block array, but it points to the next Parameter Block in the structure to be read-in by the adapter for execution. This value will be incremented by the adapter after it reads a new Parameter Block; the value should never be changed from the Host side.

Status Block IN Index

The Status Block IN field is incremented each time a Status Block is placed into the Status Block array by the adapter. This value should never be changed from the Host side.

Status Block OUT Index

The Status Block OUT Index advances when the Host reads a Status Block from the array to indicate that a Status Block has been read. This value is never changed by the adapter.

Number of Parameter Blocks

This value indicates the maximum number of Parameter Blocks that you can have active in the list. You should pick a number that is adequate for the number of commands that you will issue. Since the Status Block array follows the Parameter Block array, you cannot increase this number once execution begins. The maximum space available to use for Parameter Blocks is 65536 bytes. The maximum number of Parameter Blocks you can have is 2034.

0 and 1 are not valid values to use in this field.

Number Of Status Blocks

This value indicates the number of Status Blocks available to be written by the adapter in the Command List. Maximum space available for Status Blocks is 65536 bytes. Since each Status Block is 16 bytes long, the maximum number of Status Blocks you can have is 4096.

0 and 1 are not valid values to use in this field.

NOTE: *Multiple Status Blocks generated from retries or extended Sense data must be factored into the value you select for this field. Retries and additional Sense data options are enabled through use of the Unit Options or Extended Unit Options commands.*

Parameter Block Space

You may have up to 2034 Parameter Blocks in the Command List Structure simultaneously. The Parameter Block area is treated as an independent circular list. First you place a Parameter Block into this area, then the adapter reads it to execute.

Status Block Space

You may have up to 4096 Status Blocks in this area simultaneously. The Status Block area is treated as a circular list separate from the Parameter Block area. When the adapter places a Status Block into this area, it is available to be read in from the Host side.

It is possible to generate multiple Command List Status Blocks from one Pass-through command if retries are enabled and more than eight bytes of Sense data are selected. Retries and Sense data are enabled with the Unit Options or Extended Unit Options commands.

Creating the Command List Structure

To create a Command List Structure you should perform the following steps in the order indicated:

1. Select and clear a portion of memory that will contain the Command List. The Command List should start on a double word boundary. Enough space should be allocated to accommodate the Index information plus the Parameter and Status Blocks. Use the following information to determine the amount of space to allocate. (The value n is the

number of Parameter Blocks and the value y is the number of Status Blocks to be held in each respective array.):

Index information = 32 bytes
Parameter Block array = n x 28 bytes
Status Block array = y x 16 bytes

Zero the indexes and set-up the Number of Parameter Blocks and Number of Status Block fields.

2. Issue a Start List command to the adapter within a Single Command Structure. An example of how to do this can be found on page 4-14.

3. Create Parameter Blocks in the reserved space, update the Parameter Block IN index. Write a 1 to the Channel Attention Port to begin execution.

4. Retrieve Status Blocks from the Status Block reserved area. Update the Status Block OUT index.

Using a Command List Structure

The Command List Structure is a fast method of sending commands to the RF3570 adapter. This section explains usage of the indexes. It also details command issuing procedures.

Index Guidelines

These guidelines explain the use of the Command List Indexes:

- The respective IN and OUT indexes for the Parameter and Status Blocks control each respective circular list. The number placed in each index points to an element in an array: a 0 is the first element in an array; 1 is the second element in an array; etc.

- The IN indexes indicate the next free block of the array to be written from the Host side (Parameter Block) or from the adapter side (Status Block).

- The OUT indexes point to the next block of an array to be read in to the Host side (Status Block) or to the adapter side (Parameter Block).

- The IN and OUT Indexes are manipulated with standard circular list algorithms; If IN is equal to OUT, the list is empty; if IN is equal to OUT minus one (modulo list size), the list is full.

- The Host side is responsible for updating the Parameter Block IN Index and the Status Block OUT Index.
- The RF3570 adapter updates the Parameter Block OUT Index and the Status Block IN Index. These indexes must not be changed from the Host side.
- The Parameter Block OUT Index is not necessarily changed at the time a command is accepted or for each command read onto the board. To reduce system bus activity the adapter performs this action only periodically to keep the list from filling up. Therefore, change in the OUT Index cannot be used to signal command acceptance.

Command Guidelines

Use the following guidelines when issuing commands via Command List:

- Several commands can be loaded into the Parameter Block array at once and issued with a single write of 1 to the Channel Attention Port.
- Similarly, the adapter can return several Status Blocks at once; there may not be an interrupt received for each Status Block. However, you may also receive several interrupts in a row.
- The issue of commands and receipt of status occurs asynchronously; there are no timing restrictions on either.
- The number of possible Parameter and Status Blocks is indicated by the two counter fields: Number of Parameter Blocks; and Number of Status Blocks.

Adding a Parameter Block

To add an entry to the Parameter Block array, follow these steps:

1. Compare the Parameter Block IN Index to the OUT Index to be sure the array is not full.
2. Calculate the offset for the Parameter Block indicated by the Parameter Block IN Index and build a Parameter Block containing the command you wish to issue.
3. Increment the Parameter Block IN Index.
4. Write a 1 to the Channel Attention Port.

Removing a Status Block

To remove an entry from the Status Block array, follow these instructions:

1. Compare the Status Block IN Index with the OUT Index to find out if the list is empty.
2. Read in the Block indicated by the Status Block IN Index.
3. Increment the Status Block OUT Index.

Example 'C' routines for Command List handling begin on the next page.

Example 'C' Routines for Command List

The following 'C' listing is included here as an example of how you may set-up the Parameter Block, Status Block and Command List structures and manipulate the Command List Indexes.

```
/* Global Defines:*/
#define   EMPTY  0

#define   NPB    100    /* NPB - Num of Parameter Blocks in Command List. */
#define   NSB    200    /* NSB - Num of Status Blocks in Command List. */

#define   BYTE   unsigned char          /* an 8 bit value */
#define   WORD   unsigned int           /* a 16 bit value */
#define   DWORD  unsigned long          /* a 32 bit value */

/* parameter_block – a structure used to pass a command to the RF3570 VME to
 * SCSI adapter. The fields are:
 */
typedef struct
   {
         DWORD     identifier;          /* identifier - command identifier. */

         BYTE      target;              /* target - target ID or adapter ID. */

         BYTE      modifier;            /* modifier - address modifier. */

         BYTE      flags-1;             /* command flags. */

         BYTE      flags-2;             /* command flags

/*  possible values for flags-1 */

#define   CF_VALID                  0x80      /* dat and dir bits valid.*/
#define   CF_FREE                   0x40      /* reserved bit, not used.*/
#define   CF_INHIBIT_CMD_COMPLETE   0x20      /* command list synch control. */
#define   CF_FREE1                  0x10      /* reserved bit, not used. */
#define   CF_IRS                    0x08      /* inhibit request-sense.*/
#define   CF_DATA                   0x04      /* data for this operation. */
#define   CF_DIR                    0x02      /* direction; 1 = (writes). */
#define   CF_SGO                    0x01      /* scatter/gather  operation */

/* possible values for flags-2 */

#define   CF_JUST_SEND_MSG          0x08      /* message-only operation.*/
#define   CF_INHIBIT_DISCONNECT     0x04      /* inhibit disconnect. */
#define   CF_TAG_MESSAGE_TYPE       0x03      /* tag message type. */
```

```
    union
    {
       WORD      w[2];
       DWORD     d;
    } sdp;                              /* sdp - source/dest. memory address */

    union
    {
       WORD      w[2];
       DWORD     d;
    } bytes;                            /* bytes - data transfer count */

    BYTE   cdb[12];                     /* cdb - Command Descriptor Block */
} parameter_block;

/* status_block — a structure used to hold the status information that is
 * returned from the adapter.
 */
typedef struct
   {
         DWORD     sb_ident;          /* sb_ident - status block identifier. */

         BYTE      sb_flag;           /* sb_flag - command completion flags. */

/* possible values for sb_flag */
#define   ST_CDONE                   0x80      /* command complete. */
#define   ST_ERROR                   0x40      /* command error. */
#define   ST_RETRY                   0x20      /* retry required. */
#define   ST_TRUNCATED               0x10      /* data transfer truncated. */
#define   ST_CONTINUED               0x04      /* block continued from previous. */

         BYTE      sb_error;          /* sb_error - error code number. */

         BYTE      sb_status;         /* sb_status - command SCSI status byte. */

         BYTE      sb_res;            /* sb_res - reserved field. */

         BYTE      sb_info[8]         /* sb_info[8] SCSI request sense info. */

   } status_block;
```

```
/* clist - a Command List structure
 */
typedef struct
   {
         DWORD     pbin;               /* parameter block IN pointer. */

         DWORD     pbout;              /* parameter block OUT pointer. */

         DWORD     sbin;               /* status block IN pointer. */

         DWORD     sbout;              /* status block OUT pointer. */

         DWORD     number_pbs;         /* number of parameter blocks.*/

         DWORD     number_sbs;         /* number of status blocks. */

         DWORD     reserved[2];        /* two RESERVED fields. */

         parameter_block  pblist[NPB];/* Parameter Block area. */

         status_block    sblist[NSB]; /* Status Block area. */
} command_list;


/* Functions: */
/*------------------------------------------------------------------------
 * put_parameter_block - A function to add a command to the Command List
 *
 *----------------------------------------------------------------------*/
void put_parameter_block(pb,clist)
parameter_block pb;
command_list *clist;
{
   /* First: check to see if the list is full */
   if ( (clist->pbin + 1) % NPB == clist->pbout)
   {
      /* If the list is full, then handle_full_list */
      if (!handle_full_list(pb,clist))
         return;
   }

   /* Second: put the Parameter Block into the next available
    * location in the list
    */
   clist->pblist[clist->pbin] = pb;

   /* Third: increment the Parameter Block IN index */
   clist->pbin = (clist->pbin +1) % NPB;

   /* Next issue a Command List Channel Attention */
   issue_cl_channel_atn();

} /* put_parameter_block */
```

This may be a function call that waits for an open entry or you could choose to return an error.

This is a separate function called to issue a Command List Channel Attention.

```
/*-------------------------------------------------------------------------
 * get_status_block - Read a Status Block returned in the Command List
 *
 *------------------------------------------------------------------------ */
status_block get_status_block(clist)
command_list *clist;
{
   status_block tsb;

   /* First: check to see if a Status Block is available */
   if (clist->sbout == clist->sbin)
   {
      /* If the list is empty there are no blocks to read. Check the adapter
       * Status Port for any error conditions, and return.
       */
      handle_empty_sb_list(clist);
      return(EMPTY);
   }

   /* Second: read the Status Block into a temporary value so that it can be
    * returned at the end of the function.
    */
   tsb = clist->sblist[clist->sbout];

   /* Third: increment the Status Block OUT index. */
   clist->sbout = (clist->sbout + 1) % NSB;

   /* Next: return with the Status Block. */
   return (tsb);

} /* get_status_block */
```

This routine could check the Status Port for 11H error code (no active command list).

Chapter 5 — SCSI Pass-through Commands

Introduction

There are two types of commands that you will issue to the RF3570 adapter: SCSI Pass-through and adapter Board-control. Board-control commands are covered in the next chapter. This chapter informs you about SCSI Pass-through commands.

When the RF3570 adapter receives a SCSI Pass-through command Parameter Block, it passes a command or message byte through to the SCSI device that must perform the work. The RF3570 will perform the necessary negotiation and report the completion status.

A SCSI Pass-through command Parameter Block can contain any SCSI command available for the peripheral you are addressing. You should refer to a SCSI and Common Command Set (CCS) specification or your peripheral vendor manual for more information. This chapter describes the Parameter Block structure you should use for Pass-through commands.

Summary

This chapter summarizes SCSI Pass-through commands and can be used as a reference for:

- Pass-through command Parameter Block format.
- Pass-through command Status Block format.
- Pass-through message Parameter Block format.
- Examples of usage.

This chapter is most useful for someone writing or modifying a driver for the RF3570 product.

SCSI Pass-through Parameter Block Format

In Chapter 4, a Base Parameter Block was introduced and explained in a general context. Parameter Blocks are the same size, with the same *general* fields, whether you are issuing a SCSI Pass-through command or an adapter Board-control command. The information specific to SCSI Pass-through commands will be covered in this section.

Command Format

Use this format when issuing a Parameter Block containing a SCSI Pass-through command:

Address Offset	Byte Memory Address			
	Offset + 0	Offset + 1	Offset + 2	Offset + 3
00H	Command Identifier			
04H	Flags-2	Flags-1	Addr. Mod.	Target ID
08H	VME Memory Address			
0CH	Transfer Count			
10H	0 (Op Code)	1	2	3
14H	4	5	6	7
18H	8	9	10	11

SCSI 6, 10 or 12 byte Command Descriptor Block

Figure 5-1: SCSI Pass-through Parameter Block

Description of the Fields

The SCSI Pass-through commands Parameter Block fields can be explained as follows:

Command Identifier

The Command Identifier field is used to identify the Status Block associated with a Parameter Block.

It must be a unique value.

Target ID

This value identifies the Target ID of the device that is to receive the command contained in the SCSI Command Descriptor Block.

Address Modifier

This value further defines the VMEbus Address specified for a data transfer.

BITS							
7	6	5	4	3	2	1	0
0	0	Addr Mod					

Figure 5-2: Bit-meanings of Address Modifier Field

Addr Mod — **Address Modifier**
These six bits inform the RF3570 adapter what Address Modifier it should use for data transfer.

Valid Address Modifier values can be found by referring to a VME Specification, Revision C.1

Flags-1

The Flags-1 byte controls the command operation. Its bits have these meanings:

BITS							
7	6	5	4	3	2	1	0
DBV	0	ICC	0	IRS	DAT	DIR	SGO

Figure 5-3: Bit-meanings of Flags-1 Field

These three bits work intra-dependently. They are automatically set by the adapter to specific values for standard SCSI and SCSI-2 commands. Their values are described in Table 5-1, following the bit descriptions.

SGO — **Scatter/Gather Operation**
Scatter/Gather operation allows you to use separate pieces of memory as if it was one contiguous piece. See Chapter 7 for more information about Scatter/Gather.
0 ⇨ This *is not* a Scatter/Gather transfer.
1 ⇨ This *is* a Scatter/Gather transfer.

DIR — **Direction of Data Transfer**
Both the DBV and the DAT bits must be set to 1 for this bit to have meaning for an RF3570 operation. Otherwise its value is ignored.
0 ⇨ Data transfer is *from* the target.
1 ⇨ Data transfer is *to* the target.

➡

DAT **Data Transfer**
This bit is available to be used to enhance efficiency of vendor-unique commands. (It is automatically set for standard commands.) If the command you are issuing is vendor-unique, this bit informs the adapter that a data-in or data-out phase will occur during the command.
0 ⇨ Command has no data phase.
1 ⇨ Data phase will occur.

IRS **Inhibit Request Sense**
Selects whether Sense Data is to be gathered automatically by the adapter (in response to a Check Condition status from a device) or given to the host to do so.
0 ⇨ Automatic Request Sense issued to device reporting Check Condition.
1 ⇨ Check Condition status of device reported to Host. Host must request the device status.

ICC **Inhibit Command Complete Interrupt**
Inhibits the interrupt that notifies the host of command completion. By inhibiting interrupts for a number of commands and then enabling an interrupt for one, you can reduce the number of interrupts to service.
0 ⇨ Interrupts enabled for this command.
1 ⇨ Interrupts inhibited for this command.

DBV **Data Bits Valid**
Use this bit when you wish to override the default values of the DAT and DIR bits to implement vendor-unique commands or alter those automatically set for standard SCSI as shown in Table 5-1.
0 ⇨ Normal operation, use values in Table 5-1.
1 ⇨ DIR and DAT values in the Flag byte are valid and override the defaults in Table 5-1.

Setting DBV, DAT and DIR Bits

The DAT and DIR bits, together, indicate whether a data transfer should be expected for a given command, and if so, which direction it will be: to or from the target. The DBV bit gates whether the values for DAT and DIR will be derived from the default table or from the values you assign.

Using the Table Values

The following table of defaults is useful for you to use when you wish to: 1.) alter the default behavior of a standard SCSI command for a specific application or 2.) implement a vendor-unique command.

An example of altering the default function of a command can be demonstrated with the SCSI Format command. If you examine the DAT, and DIR bit values for this command in the table, you see that the adapter assumes:

Opcode	DAT	DIR	Description
04H	1	1	Format Unit

Figure 5-4: Example DAT and DIR bit settings

The **ones** in the table for DIR and DAT mean that by default the adapter will assume that there is data associated with the command and the direction of data flow is to the target.

If the Format command for your device *requires no data transfer*, set the Flags-1 byte as follows when you issue the Format command:

DBV = 1, DAT = 0, DIR = 0.

Setting DBV to **one** tells the adapter to examine your settings for DIR and DAT and their **zero** settings indicate no data is associated with the command.

An example of the second case, implementing a vendor-unique command, can also be given. Suppose your SCSI device has a vendor-unique command available for Opcode 02H. When you examine the table for the default settings of that opcode you see that the RF3570 adapter assumes:

Opcode	DAT	DIR	Description
02H	0	0	not defined

Figure 5-5: DIR and DAT Settings, Vendor-unique

In this example, suppose the 02H opcode vendor-unique command involves a data transfer from the device to the adapter. To override the table values and use the opcode for a vendor-unique command set the DBV, DAT and DIR bits of the Flags-1 byte as follows:

DBV = 1; DAT = 1; DIR = 0.

This combination means: DBV is set so use Flags-1 field values for DAT and DIR; DAT is set so data will be transferred; DIR is zero so direction of transfer is from the device.

NOTE: *If you set a combination of DBV, DIR and DAT that is inconsistent with how your target executes the command, you will receive error code 24H, Unexpected SCSI Phase Encountered. You must refer to your vendor-specific documentation for information about your peripheral's command execution.*

Table of DAT and DIR Automatic Settings

The following table demonstrates the values for the DAT and DIR bits that the RF3570 associates by default with each of the SCSI opcodes:

Opcode	DAT	DIR	Description
00H	0	0	Test Unit Ready
01H	0	0	Rewind/Rezero
02H	0	0	not defined
03H	1	0	Request Sense
04H	1	1	Format Unit
05H	1	0	Read Block Limits
06H	0	0	not defined
07H	1	1	Reassign Blocks
08H	1	0	Read
09H	0	0	not defined
0AH	1	1	Write
0BH	0	0	Seek
0CH	0	0	not defined
0DH	0	0	
0EH	0	0	
0FH	1	0	Read Reverse
10H	0	0	Write Filemarks
11H	0	0	Space
12H	1	0	Inquiry
13H	0	0	Verify
14H	1	0	Recover Buffered Data
15H	1	1	Mode Select
16H	1	1	Reserve
17H	1	1	Release
18H	1	1	Copy
19H	0	0	Erase
1AH	1	0	Mode Sense
1BH	0	0	Start/Stop: Load/Unload
1CH	1	0	Receive Diagnostic
1DH	1	1	Send Diagnostic
1EH	0	0	Prevent/Allow Media Removal
1FH	0	0	not defined

Table 5-1: DAT and DIR Automatic Settings

Table 5-1: DAT and DIR Automatic Settings (continued)

Opcode	DAT	DIR	Description
20H	0	0	not defined.
21H	0	0	
22H	0	0	
23H	0	0	
24H	0	0	
25H	1	0	Read Capacity
26H	0	0	not defined
27H	0	0	
28H	1	0	Extended Read
29H	0	0	Vendor-unique
2AH	1	1	Extended Write
2BH	0	0	Extended Seek
2CH	1	0	Ciprico defined
2DH	1	0	Ciprico defined
2EH	1	1	Write and Verify
2FH	0	0	Verify
30H	1	1	Search Data Equal
31H	1	1	Search Data High
32H	1	1	Search Data Low
33H	1	1	Set Limits
34H	0	0	Pre-fetch Cache
35H	0	0	Synchronize Cache
36H	0	0	Lock/Unlock Cache
37H	1	0	Read Defect Data
38H	0	0	not defined
39H	1	1	Compare
3AH	1	1	Copy and Verify
3BH	1	1	Write Buffer
3CH	1	0	Read Buffer
3DH	0	0	not defined
3EH	1	0	Read Long
3FH	1	1	Write Long

Table 5-1: DAT and DIR Automatic Settings (continued)

Opcode	DAT	DIR	Description
40H	1	1	Change Definition
41H	1	1	Write Same
42H	0	0	not defined
43H	0	0	
44H	0	0	
45H	0	0	
46H	0	0	
47H	0	0	
48H	0	0	
49H	0	0	
4AH	0	0	
4BH	0	0	
4CH	1	1	Log Select
4DH	1	0	Log Sense
4EH	0	0	not defined
4FH	0	0	
50H	0	0	
51H	0	0	
52H	0	0	
53H	0	0	
54H	0	0	
55H	1	1	Mode Select (10)
56H	0	0	not defined
57H	0	0	
58H	0	0	
59H	0	0	
5AH	1	0	Mode Sense (10)
5BH	0	0	not defined
5CH	0	0	
5DH	0	0	
5EH	0	0	
5FH	0	0	

Table 5-1: DAT and DIR Automatic Settings (continued)

Opcode	DAT	DIR	Description
60H thru A3H	0	0	not defined
A4H	1	0	Ciprico defined
A5H	0	0	Move Medium/Play Audio
A6H	0	0	Exchange Medium
A7H	0	0	not defined
A8H	1	0	Read (12)/Get message
A9H	0	0	Play Track Relative
AAH	1	1	Write (12)/Send message
ABH	0	0	not defined
ACH	0	0	Erase (12)
ADH	0	0	not defined
AEH	1	1	Write and Verify (12)
AFH	0	0	Verify (12)
B0H	1	1	Search Data High (12)
B1H	1	1	Search Data Equal (12)
B2H	1	1	Search Data Low (12)
B3H	1	1	Set Limits
B4H	0	0	not defined
B5H	1	0	Request Volume Element Address
B6H	1	1	Send Volume Tag
B7H	1	0	Read Defect Data (12)
B8H	1	0	Read Element Status
B9H thru BFH	0	0	not defined
C0H	0	0	Ciprico defined
C1H	0	0	not defined
C2H	1	0	Ciprico defined
C3H	1	0	Ciprico defined

Table 5-1: DAT and DIR Automatic Settings (continued)

Opcode	DAT	DIR	Description
C4H thru CEH	0	0	not defined
CFH	0	0	Ciprico defined
D0H	0	0	not defined
D1H thru D5H	0	0	not defined
D6H	0	0	Ciprico defined
D7H	0	0	Ciprico defined
D8H thru E3H	0	0	not defined
E4H	1	0	Ciprico defined
E5H thru EFH	0	0	not defined
F0H	0	0	Ciprico defined
F1H F2H F3H	0	0	not defined
F4H	0	0	Ciprico defined
F5H thru FFH	0	0	not defined

Flags-2

The bits of this field are used to select SCSI-2 and additional command control features implemented by the RF3570 adapter. The meanings of the bits are as follows:

BITS							
7	6	5	4	3	2	1	0
0	0	0	0	JSM	IAD	TAG OPTS	

Figure 5-6: Bit-meanings of Flags-2 Field

TAG OPTS These two bits let you choose the tagged queuing method you wish used with the current command. You can choose one of three types of messages for the RF3570 adapter to use when issuing the command.

TAG OPT Value	Meaning:
00	**SIMPLE QUEUE TAG MESSAGE: The command may be processed by the SCSI-2 device out of sequence for efficiency. Example: device may sort commands by logical block address.**
01	**ORDERED QUEUE TAG MESSAGE: The command must be executed by the SCSI-2 device in the order sent by the adapter (in order with respect to other commands sent with the ORDERED QUEUE TAG Message).**
10	**HEAD OF QUEUE TAG MESSAGE: The command is executed by the SCSI-2 device before all other commands previously sent, including other Head of Queue Tag commands but excluding command currently in process.**
11	**Reserved**

Table 5-2: Meaning of TAG OPTS Bits

In order for these bits to apply to the current operation you must have *already enabled* tagged queuing with either the Unit Options or Extended Unit Options command. These commands are covered in detail in Chapter 6, Board-control Commands. If you do not enable tagged queuing with the Unit Options or Extended Unit Options command, the TAG OPTS bits are ignored.

For more information about the use of tagged queuing, see the usage discussion in Chapter 7.

NOTE: *Depending on the SCSI-2 device you are using, you may also need to enable tagged queuing with a MODE SELECT page 0AH, Common Device-Type Control Parameters command to the device. See your vendor-supplied peripheral documentation for more information.*

IAD **Inhibit Automatic Disconnect**
If you enable SCSI disconnect privilege with the DIS bit of the General Options Board-control command (Bit 0 of Select Flags field), disconnects will automatically be tried for any command. Use the IAD bit to *disable* SCSI device disconnect/reconnect privilege for the current command. This can be important for performance reasons.
0 ⇨ Determine disconnect from the DIS bit of General Options command.
1 ⇨ Ignore DIS bit, inhibit the disconnect privilege.

JSM **Just Send Message**
Use this bit to change the SCSI Pass-through *Command* Parameter Block to a Pass-through *Message* Parameter Block. In this Message mode, the adapter selects a device and only sends a message. This provides a measure of error recovery at the host level.
0 ⇨ Pass-through command operation as usual.
1 ⇨ Use the Pass-through Message Parameter Block format.

This message-only format is illustrated on page 5-15, following the description of the Pass-through command structure.

VME Memory Address

When the SCSI operation involves a data transfer, this field contains the beginning address for the transfer. If the SCSI command you are issuing does not require a data transfer, this field and the Address Modifier field are ignored by the adapter.

Transfer Count

This field determines the number of bytes that will be transferred to or from host memory. For some SCSI commands this is an indeterminate number. In that case, there are two ways to use this field:

1.) You can place a value in the field to represent the *maximum* number of bytes the adapter may transfer. Then, if the **target** attempts to transfer more data than this, the **adapter** will discontinue the operation and return a Status Block.

2.) Alternately, you could place a zero value in this field. This causes the adapter to continue to accept data transfer requests from the target until command is complete.

SCSI Command Descriptor Block

These 12 bytes of the SCSI Pass-through Parameter Block are reserved for the actual SCSI command that will be passed to the drive. Except for the first byte, which contains the command opcode, each SCSI command uses the bytes for a different purpose; see a SCSI specification or vendor-supplied peripheral documentation for details about the commands and fields.

SCSI commands may be 6, 10, or 12 bytes in length. Any of the 12 bytes reserved for the SCSI command that are unused are ignored. The SCSI command block is passed to the SCSI device exactly as presented. The RF3570 adapter doesn't alter any of the fields you set up.

Message Format

When you set the JSM bit in the Flags-2 field of the Pass-through Parameter Block, the format of the Parameter Block changes:

Address Offset	Byte Memory Address			
	Offset + 0	Offset + 1	Offset + 2	Offset + 3
00H	Command Identifier			
04H	Flags 2	Flags 1	Addr. Mod.	Target ID
08H	VME Memory Address			
0CH	Transfer Count			
10H	Message Code	LUN	2	3
14H	4	5	6	7
18H	8	9	10	11

Figure 5-7: SCSI Pass-through Message Parameter Block

Description of the Fields

All fields that appear in light text above are ignored when the JSM bit of the Flags-2 byte is set.

The Command Identifier and Target ID fields work the same as explained in the preceeding pages. The JSM bit of the Flags-2 field is the only bit in that field that is valid.

Message Code

Currently there are three messages that are valid to appear in this field. They are:

06H - ABORT
0CH - BUS DEVICE RESET
0EH - CLEAR QUEUE

Other messages will not be passed to the device and will create an error status. Error code returned is 01H (invalid command).

LUN

This field has the same format as specified in the SCSI specification for LUN (Logical Unit Number); the 3 most significant bits hold the LUN value.

SCSI Pass-through Status Block Format

In Chapter 4, a Base Status Block was introduced and explained in general terms. The Status Block for a SCSI Pass-through command encompasses the same fields used for the Board-control commands, plus it holds information that is particular to a SCSI transaction: a SCSI status byte, and eight bytes of Sense data returned from the SCSI device.

The Sense Data fields are an important part of the Status Block. Sense data is acquired directly from the SCSI device; you can enable the adapter to automatically request Sense Data in response to an error condition reported from the device. What you will see in these fields is affected by your use of the Unit Options or Extended Unit Options Board-control commands. If you use one of these commands to have more than eight bytes of Sense Data returned, you will receive multiple Status Blocks.

This section includes information about all the fields you will see when examining a Status Block returned from a SCSI Pass-through command.

Status Block Format

This is the format you can expect from a SCSI Pass-through Status Block:

The fields displayed in light type are the default Sense Bytes returned by SCSI Pass-through commands

Address Offset	Byte Memory Address			
	Offset + 0	Offset + 1	Offset + 2	Offset + 3
00H	Command Identifier			
04H	Reserved	SCSI Status	Error	Flags
08H	0 = Class/Code	1 = Segment	2 = SCSI Flags	3 = Info Bytes
0CH	4 = Info Byte	5 = Info Bytes	6 = Info Byte	7 = Ex Length

Selected Sense Bytes

Figure 5-8: SCSI Pass-through Status Block

Description of Fields

Following is a brief description of each of the fields:

Command Identifier

This value links a Status Block with a Parameter Block.

Flags

This byte can be polled to ascertain whether, and in what condition, a command completed. It has the following format:

BITS							
7	6	5	4	3	2	1	0
CC	ERR	RTY	DTT	0	CSB	0	0

Figure 5-9: Bit-meanings of Flags Field

CSB **Continued Status Block**
This bit indicates that the current Status Block is one of several Status Blocks generated because extended Sense Bytes are enabled. Only valid for Pass-through commands issued through Command List.
0 ⇔ Status Block is *first (or only)* Status Block
1 ⇔ Status Block is one of a series of Status Blocks.

DTT **Data Transfer Truncated**
Indicates when a SCSI Pass-through command completes with fewer bytes transferred than requested in the Transfer Count field.
0 ⇔ Data transfer was *not* truncated.
1 ⇔ Data transfer was truncated.

RTY **Retry**
Indicates whether a SCSI Pass-through command required retries to complete.
0 ⇔ No retries were required.
1 ⇔ Retries were required to complete.

ERR **Error Status**
Identifies when a command, Board-control or Pass-through, completes with an error.
0 ⇔ No error occurred.
1 ⇔ An error occurred.

CC **Command Complete**
Indicates whether a command has finished.
0 ⇔ Command not complete.
1 ⇔ Command complete.

Error

When an adapter-detected error occurs, this byte contains an error code to describe the condition. See Appendix A.

SCSI Status

When a SCSI device completes a command, during the STATUS phase of the SCSI bus it returns a SCSI Status Byte Code to the adapter. The adapter reports that value in this field. The field has these bit meanings:

BITS							
7	6	5	4	3	2	1	0
RES	RES	Status Byte Code					RES

Figure 5-10: Bit-meanings of SCSI Status Field

RES SCSI reserved.

Status Byte Code Implementation of these bits is required for all devices that adhere to SCSI specifications. See Table 5-3 (below) for information about what they mean.

Status	HEX Code*	7	6	5	4	3	2	1	0
Good	00H	R	R	0	0	0	0	0	R
Check Condition	02H	R	R	0	0	0	0	1	R
Condition Met/Good	04H	R	R	0	0	0	1	0	R
Busy	08H	R	R	0	0	1	0	0	R
Intermediate/Good	10H	R	R	0	1	0	0	0	R
Intermediate/Condition Met/Good	14H	R	R	0	1	0	1	0	R
Reservation Conflict	18H	R	R	0	1	1	0	0	R
Queue Full	28H	R	R	1	0	1	0	0	R

R = Reserved
* Assumes that R values are zero.

Table 5-3: Meaning of Status Byte Code

Further information about the Status Byte Code can be found in the SCSI or SCSI-2 specification.

SCSI Status Field during Unexpected SCSI Phase

In the event that the adapter encounters an unexpected SCSI phase it does two things: posts an error (24H) in the Error field of the Status Block, and reports the phase encountered in the last three bits of the SCSI Status field. The bits correspond to the MSG, C/D, and I/O signals. At the time of an Unexpected SCSI Phase Encountered error, the SCSI Status field has the following bit-meaning:

BITS							
7	6	5	4	3	2	1	0
X	X	X	X	X	M	C	I

Figure 5-11: SCSI Status Field - Unexpected SCSI Phase

I	I/O (Input/Output) Signal from Target
C	C/D (Control/Data) Signal from Target
M	MSG (Message) Signal from Target
X	May be either 0 or 1

The unexpected phase encountered can be determined by interpreting the state of the three signals:

Signal			Phase	Direction of Transfer
MSG	C/D	I/O		
0	0	0	Data Out	Initiator to target
0	0	1	Data In	Initiator from target
0	1	0	Command	Initiator to target
0	1	1	Status	Initiator from target
1	0	0	Reserved	—
1	0	1	Reserved	—
1	1	0	Message Out	Initiator to target
1	1	1	Message In	Initiator from target

Table 5-4: Meaning of Status Byte - Unexpected Phase

See your SCSI or SCSI-2 specification for more information about SCSI phases.

Sense Bytes

When enabled to do so (the IRS bit of the Flags-1 field of the Pass-through command is *not* set), the RF3570 adapter automatically responds to Check Condition status from a device, with a Request Sense command.

The device answers the Request Sense command by returning information about its condition. This information is called Sense Bytes. Depending on the peripheral, up to 256 Sense Bytes can be returned in response to the Request Sense command.

There are three ways that Sense Bytes can be reported to you by the RF3570 via the Sense Bytes fields of the Status Block:

- The first eight bytes of Sense data returned (this is the default).
- Up to 32 of the first sequential Sense Bytes.
- Up to 16 of any of the 256 possible Sense Bytes returned.

You can specify one of these methods, for Sense Bytes to be reported in the Status Block of SCSI Pass-through commands, by using the Unit Options or Extended Unit Options Board-control command (See Chapter 6).

If no Check Condition status occurred, or the automatic Request Sense capability of the adapter is not enabled, the Sense Bytes fields will all be zero. See page 5-4 for more information about how to inhibit the automatic Request Sense command (using the IRS bit).

Default Sense Bytes

You will receive the default eight bytes of Sense data if you place a zero in the Sense Count field and zeroes in all fields of the Selected Sense Bytes of the Unit Options or Extended Unit Options Board-control command.

The default values returned are:

Byte Memory Address			
Offset + 0	Offset + 1	Offset + 2	Offset + 3
Command Identifier			
Reserved	SCSI Status	Error	Flags
0 = Class/Code	1 = Segment	2 = SCSI Flags	3 = Info Byte
4 = Info Byte	5 = Info Byte	6 = Info Byte	7 = Ex Length

Figure 5-12: Default Ordering of Sense Bytes

Only one Status Block is required to return the default Sense Bytes.

Up to 32 Sequential Sense Bytes

This option is chosen by placing a number (up to 32) in the Sense Count field of either the Unit Options or Extended Unit Options command. The Selected Sense Bytes fields are ignored.

This is useful if all pertinent information occurs within the first 32 bytes of Sense data returned from your device.

If you enable more than eight bytes of sequential Sense data to be returned, they will be returned in *multiple* Status Blocks. For example, the first sixteen would appear in this order:

Byte Memory Offset				
Offset + 0	Offset + 1	Offset + 2	Offset + 3	
Command Identifier				
Reserved	SCSI Status	Error	Flags	
0 = Class/Code	1 = Segment	2 = SCSI Flags	3 = Info Byte	Sense Bytes in Status Block #1
4 = Info Byte	5 = Info Byte	6 = Info Byte	7 = 8H	
Command Identifier				
Reserved	SCSI Status	Error	Flags	
8	9	10	11	Sense Bytes in Status Block #2
12	13	14	15	

(Command or Vendor-Unique Values)

Figure 5-13: Ordering of 1st 16 Sequential Sense Bytes

Selectable Sense Bytes

You also have the option to select which (up to 16) of the 256 possible Sense bytes be returned in the Status Block and in what order they should appear.

- The Unit Options command allows you to choose 8 *selectable* Sense Bytes.
- The Extended Unit Options command allows you to choose 16 *selectable* Sense Bytes. You will receive multiple Status Blocks.

You can specify Selectable Sense Bytes by placing a zero in the Sense Count field and filling each of the Selected Sense fields with the Sense Byte number you wished returned in that field.

For example, the Sense Bytes fields of a Status Block could return the following Sense Bytes in the following order if the Extended Unit Options command had been used.

	Byte Memory Offset			
	Offset + 0	Offset + 1	Offset + 2	Offset + 3
	Command Identifier			
	Reserved	SCSI Status	Error	Flags
Sense Bytes in Status Block #1	0	1	2	6
	5	4	3	7
	Command Identifier			
	Reserved	SCSI Status	Error	Flags
Sense Bytes in Status Block #2	56	57	13	19
	122	33	12	23

Figure 5-14: Example of Selectable Sense Bytes

Status Block #1 contains the first eight bytes with the Info Bytes arranged from LSB to MSB, and Status Block #2 returns Sense Bytes that are selected from among the other 248 Sense Bytes, based on information in the vendor-supplied peripheral documentation.

See Chapter 6 for more information about the Unit Options and Extended Unit Options commands and how to set-up the Status Block Sense bytes the way you wish.

Chapter

6

Board-control Commands

Chapter 6 — Board-control Commands Reference

Introduction

There are two types of commands that you will issue to the RF3570 adapter: SCSI Pass-through commands, and adapter Board-control commands. This chapter concentrates on Board-control commands.

When the RF3570 adapter receives a Board-control command it must interpret the command, then apply to its operation whatever was specified in the command. Board-control commands allow you to communicate set-up information and operational guidelines to the RF3570 adapter, as well as request current information from the board, perform diagnostics, and selectively reset the SCSI bus. This chapter is intended to be used as a look-up reference for these Board-control commands issued to the adapter.

If you need information about the SCSI Pass-through commands available, see a current SCSI or SCSI-2 specification and CCS addendum. See Chapter 5 for information about initiating a Pass-through command.

Summary

This chapter provides detail about each individual Board-control command:

- The command code
- A description of the command
- Explanation of unique flags and Parameter Block fields
- Examples of command usage where helpful

This chapter is most helpful to someone writing or modifying a driver for the RF3570 product.

Using the Reference

The RF3570 adapter recognizes the type of command it is receiving by the contents of the *Target/Adapter ID* field of its parameter block. If the *Target/Adapter ID* field contains a number between 0 and 7 (indicating SCSI target addresses) the command is a SCSI Pass-through command. If the *Target/Adapter ID* field contains an FFH, the adapter recognizes it as a Board-control command. This chapter explains only the Board-control commands. The *Target/Adapter ID* field will always be set to FFH for these commands. In this chapter it is referred to as just the **Adapter ID** field.

Each of the Board-control commands are completely described in the following pages; each description follows the same format:

- First the Parameter Block format is illustrated for each command. The fields that are valid appear in bold type.
- Next, each field used by the command is explained.
- Last, if there is a special type of Status Block returned by the command, it is displayed and all of its fields explained.

General Status Block

When a Board-control command does not return a special type of Status Block, it is assumed that its Status Block will take the general format. That format is described here for reference.

General Status Block Format

The format of the Status Block returned by most Board-control commands takes this form:

Address Offset	Byte Memory Address			
	Offset + 0	Offset + 1	Offset + 2	Offset + 3
00H	Command Identifier			
04H	Reserved		Error	Flags
08H	Reserved			
0CH				

Figure 6-1: General Status Block

Description of Fields

Following is a brief description of each of the fields used to return General Status:

Command Identifier

This value is used to match a Status Block to its Parameter Block.

Flags

This field can be polled to ascertain whether, and in what condition, a command completed. It has this format:

BITS							
7	6	5	4	3	2	1	0
CC	ERR	0	0	0	0	0	0

Figure 6-2: Bit-meanings of Flags Field

ERR **Error Status**
Identifies when a command completes with an error.
0 ⇨ No error occurred.
1 ⇨ An error occurred.

CC **Command Complete**
Indicates whether a command has finished.
0 ⇨ Command not complete.
1 ⇨ Command complete.

NOTE: *The Flags byte returned in the Status Block of a Board-control command reserves the bits other than ERR and CC. However, three of the reserved bit fields are used in the Flags byte returned in the Status Block of a Pass-through command. See Chapter 5 for more information.*

Error

When an adapter-detected error occurs, this byte contains an error code to describe the condition. See Appendix A.

These are the Board-control commands that you can use to set-up the RF3570 operation in your system.

List of Commands

Command Code	Command Name	Page #
01H	Start Command List	6-6
02H	Stop Command List	6-8
05H	Identify	6-10
06H	Board Statistics	6-14
07H	General Options	6-18
08H	Unit Options	6-22
09H	Diagnostic/Self-test	6-30
10H	SCSI Hard Reset	6-34
15H	Board Information	6-36
16H	Extended Board Statistics	6-48
18H	Extended Unit Options	6-52

Start Command List (01H)

With this command you can instruct the RF3570 adapter to begin operating from a Command List. Since with Command List operation you can attain higher performance from your SCSI peripheral, this is one of the first commands you should issue to the adapter. It must be issued by means of a Single Command Structure.

See **Chapter 4, Command Operation** for information about *using* Command Lists.

Command Format

The Parameter Block Format for the Start Command List command looks like this:

Address Offset	Byte Memory Address			
	Offset + 0	Offset + 1	Offset + 2	Offset + 3
00H	Command Identifier			
04H	Reserved		Addr Mod	Adapter ID = FFH
08H	Command List Memory Address			
0CH	Reserved		Interrupt	
10H	Command = 01H	Reserved		
14H	Reserved			
18H	Reserved			

Figure 6-3: Start Command List Parameter Block

Description of Fields

Each of the fields used by the Start Command List command are explained in the following section:

Command Identifier

The Command Identifier field is used to identify the Status Block associated with a Parameter Block. It must be a unique value.

Adapter ID

This field must always contain the ID of FFH to indicate a Board-control command.

Address Modifier

This is the value that the adapter is to drive onto the VMEbus Address Modifier Lines when selecting the memory address of the Command List.

Start Command List (01H)

Command List Memory Address

This field contains the address of the Command List in Memory. It points to the beginning of the structure: Parameter Block IN index.

Interrupt

This field defines the interrupt level for the adapter to use when reporting command completion. You can also denote the resultant interrupt acknowledge ID that the adapter will place on the data bus in response to the host. The Interrupt field of the Start Command List Parameter Block uses these bit-meanings:

BITS															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	INT			Status/ID							

Figure 6-4: Bit-meanings of Interrupt Field

Status/ID **Interrupt Status/ID**
This byte denotes the value the adapter will provide to identify the interrupt being acknowledged by the Host.

INT **Interrupt Level**
You may elect to use any one of the following levels for adapter interrupt:
000 — Interrupt disabled
001 — Interrupt level 1
010 — Interrupt level 2
011 — Interrupt level 3
100 — Interrupt level 4
101 — Interrupt level 5
110 — Interrupt level 6
111 — Interrupt level 7

Command Code

This field indicates the command to be executed. For the Start Command List command it is always equal to 01H.

NOTE: *An Error Code of 11H will be returned in the Status Block if you attempt to start an already started Command List.*

Stop Command List (02H)

Stop Command List (02H)

The Stop Command List command is used to gracefully halt Command List operations. Upon receipt of this command, the adapter will accept no further additions to the Command List queue.

After all other existent commands are executed from the Command List, a Status Block for this command is returned.

The Stop Command List command is useful when you are shutting down your system. It processes and returns status of commands current at the time of its issue, therefore you can be sure all data is transferred properly before shutdown.

This command should be issued via a Single Command Structure.

Command Format

The format of the Stop Command List Parameter Block is shown below:

Address Offset	Byte Memory Address			
	Offset + 0	Offset + 1	Offset + 2	Offset + 3
00H	Command Identifier			
04H	Reserved			Adapter ID = FFH
08H	Reserved			
0CH	Reserved			
10H	Command = 02H	Reserved		
14H	Reserved			
18H	Reserved			

Figure 6-5: Stop Command List Parameter Block

Description of Fields

Each of the fields used by the Stop Command List command are explained in the following section:

Command Identifier

The Command Identifier field is used to identify the Status Block associated with a Parameter Block. It must be a unique value.

Adapter ID

The Stop Command List command is a Board-control command. To identify it as such to the RF3570 adapter, this field must contain the value FFH.

Command Code

The command code to use to indicate the Stop Command List command is 02H.

NOTE:

An error code of 11H will be returned in the Status Block for this command if you attempt to stop a Command List that is stopped (or has not been started).

Identify (05H)

Identify (05H)

This command returns a special Status Block that contains information about the Firmware installed on the board. It can be used to determine board compatibility and support of options (especially useful if your application depends on certain functions to be in place on every product).

The Identify command can be issued with either type of command structure. Its primary use may be during initialization, which may require use of a Single Command Structure (due to limitations on available memory at power-up).

Command Format

The Parameter Block format for the Identify command looks like this:

Address Offset	Byte Memory Address			
	Offset + 0	Offset + 1	Offset + 2	Offset + 3
00H	Command Identifier			
04H	Reserved			Adapter ID = FFH
08H	Reserved			
0CH	Reserved			
10H	Command = 05H	Reserved		
14H	Reserved			
18H	Reserved			

Figure 6-6: Identify Command Parameter Block

Description of Fields

The Identify command requires only three fields as explained in the following descriptions.

Command Identifier

The Command Identifier field is used to identify the Status Block associated with a Parameter Block. It must be a unique value.

Adapter ID

An FFH indicates that this is a Board-control command.

Command Code

The Command Code for the Identify command is 05H.

Identify Command Status Block

The Status Block returned by the Identify command contains fields that are slightly different from the standard Status Block. The fields of this special Status Block are described below:

Address Offset	Byte Memory Address			
	Offset + 1	Offset + 1	Offset + 2	Offset + 3
00H	Command Identifier			
04H	FW Rev.	Eng. Rev.	Error	Flags
08H	Option Flags	Day	Month	Year
0CH	Firmware Number of Even Prom			

Figure 6-7: Identify Command Status Block

Description of Fields

The Status Block returned from an Identify command contains these fields:

Command Identifier

The value in this field will be identical to the Parameter Block Value.

Flags Field

This byte is used to indicate command completion. It has this format:

BITS							
7	6	5	4	3	2	1	0
CC	ERR	0	0	0	0	0	0

Figure 6-8: Bit-meanings of Flags Field

ERR **Error Status**
Indicates if the command completed with an error.
0 ⇨ No error occurred.
1 ⇨ A command execution error occurred.

CC **Command Complete**
Indicates a command has finished.
0 ⇨ Command is not complete.
1 ⇨ Command has completed.

Error

When an adapter-detected error occurs, this byte contains an error code to describe the condition. See Appendix A for descriptions of all codes that could occur in this field.

Engineering Revision

This field indicates the revision level used by Ciprico to identify its source code. This field is useful only for Ciprico and should not be used to identify the revision level of the board firmware.

Firmware Revision

This field contains a value that identifies the released level of Firmware for the product.

Option Flags

The Options field of the Identify command Status Block has the following format:

BITS							
7	6	5	4	3	2	1	0
SCSI ID			RST	0	TAG	0	0

Figure 6-9: Option Flags Field - Identify Status Block

TAG Tagged Queuing
Indicates whether the firmware installed in the board supports tagged queuing.
0 ⇨ Tagged queuing *not* supported.
1 ⇨ Tagged queuing supported.

RST — **Reset Jumper**
Indicates whether the SCSI bus will be reset at power-up and when adapter is reset through the reset port.
0 ⇒ SCSI bus will not be reset.
1 ⇒ SCSI bus will be reset.

SCSI ID — **SCSI ID of Adapter**
Indicates the current SCSI bus ID of the RF3570 adapter.
0 ⇒ Bit of ID not set.
1 ⇒ Bit of ID set.

Bit → / SCSI ID ↓	7	6	5
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

Figure 6-10: Adapter SCSI ID Bits

Day, Month, Year

These three fields, when combined, indicate the date (in hex) that the firmware in the EPROM was compiled for production.

Firmware Number of Even EPROM

There is a set of two EPROMs on the board. This number is the Firmware number of the EVEN EPROM.

Board Statistics (06H)

Board Statistics (06H)

The RF3570 adapter keeps track of several types of board statistics. This particular command maintains compatibility with earlier RF35XX products; it is used to report three types of statistics. The Extended Board Statistics command, described in a following section, reports additional board statistics.

The statistics are valid only as long as power is kept to the system. This command reports the statistics by recording them in the area of system memory specified in the command Parameter Block. (This statistics structure is returned in addition to the normal Status Block.)

You can clear the internal statistics table at any time by issuing the command with the CLR bit of the Options field set. No data transfer occurs when the CLR bit is set.

Command Format

The Parameter Block format for the Board Statistics command looks like this:

Address Offset	Byte Memory Address			
	Offset + 0	Offset + 1	Offset + 2	Offset + 3
00H	Command Identifier			
04H	Reserved		Addr. Mod.	Adapter ID = FFH
08H	VME Memory Address			
0CH	Reserved			
10H	Command = 06H	Options	Reserved	
14H	Reserved			
18H	Reserved			

Figure 6-11: Board Statistics Command Parameter Block

Description of Fields

Each of the fields of the Parameter Block are explained below:

Command Identifier

The Command Identifier field is used to identify the Status Block associated with a Parameter Block. It must be a unique value.

Adapter ID

The Board Statistics command is a Board-control command. This field must be set to FFH.

Address Modifier

This is the value to be used by the adapter on the Address Modifier lines when writing statistics to memory. This field is not used when the CLR bit is set.

VME Memory Address

This is the system memory address at which you wish to have the adapter place the board statistics. This field is not used when the CLR bit of the Options byte is set.

Command Code

The value for this field will always be 06H to indicate a Board Statistics command.

Options

This field has the following bit-meanings:

BITS							
7	6	5	4	3	2	1	0
0							CLR

Figure 6-12: Bit-meanings of Statistics Options Field

CLR This bit allows you to clear the internal statistic values currently stored. (Statistics returned by both the 06H and 16H commands will be cleared.) It is useful if you wish to keep track of statistics during a specific period. The only other time statistic values get cleared is during adapter reset.
0 ⇨ Command is used to return internal statistics to specified system memory address.
1 ⇨ Command is used to clear the internal statistics table. VMEbus Address fields are not used.

The Data Structure returned from the Board Statistics command is illustrated on the next page. ➡

Board Statistics (06H)

Statistics Data Structure

The Statistics Data Structure returned from a Board Statistics command will have this format:

Address Offset	Byte Memory Address			
	Offset + 0	Offset + 1	Offset + 2	Offset + 3
00H	Total Commands Processed			
04H	Total SCSI Commands Issued to ID 0			
08H	Total SCSI Commands Issued to ID 1			
0CH	Total SCSI Commands Issued to ID 2			
10H	Total SCSI Commands Issued to ID 3			
14H	Total SCSI Commands Issued to ID 4			
18H	Total SCSI Commands Issued to ID 5			
1CH	Total SCSI Commands Issued to ID 6			
20H	Total SCSI Commands Issued to ID 7			
24H	Number of Times SCSI Check Condition Status Received			
28H : 58H	Reserved			

Figure 6-13: Board Statistics Data Structure

Description of Fields

The Board Statistics Data Structure returned from the Board Statistics command contains these fields:

Total Commands Processed

This field reports the number of commands, SCSI Pass-through and Board-control, that have been processed by the adapter since power-up or since the CLR option has been used.

Total SCSI Commands issued to an ID

Each of these fields reports the number of commands the adapter has issued to the respective SCSI ID.

Number of SCSI Check Condition Status Received

This field reports the number of times a SCSI device reported a Check Condition status.

General Options (07H)

The General Options command allows you to select how the RF3570 adapter treats system and SCSI operation. The options you can select are general to the entire SCSI bus. Other options are available to you on a per unit basis with the Unit Options command and Extended Unit Options command.

The options of this command need only be set once during initialization; they will remain in effect until the next General Options command is issued or until the adapter is reset.

Command Format

The Parameter Block for the General Options command looks like this:

Address Offset	Byte Memory Address			
	Offset + 0	Offset + 1	Offset + 2	Offset + 3
00	Command Identifier			
04	Select Flags	Bus Throttle	SCSI Bus ID	Adapter ID = FFH
08H	Reserved			
0CH	Reserved			
10H	Command = 07H	Reserved		
14H	Reserved			
18H	Reserved			

Figure 6-14: General Options Command Parameter Block

Description of Fields

Each field of the Parameter Block is explained below:

Command Identifier

The Command Identifier field is used to identify the Status Block associated with a Parameter Block. It must be a unique value.

Adapter ID

The General Options command is a Board-control command. This value must be FFH.

SCSI Bus ID

Each device that is connected to the SCSI bus must have an ID in order to communicate on the bus; the initial SCSI ID of the RF3570 adapter is set through jumpers. (See Chapter 3 for information on setting the jumpers.)

This field of the General Options command allows you to select a different ID than the one jumpered for the adapter, without changing the jumpers. This can be useful to avoid conflicts caused by duplicated ID's at power-up.

Valid values are 0 - 7H.

NOTE: *You can use the Board Information command (page 6-36) to find out what ID the jumpers are set to without physically examining the board.*

Bus Throttle

The bus throttle control allows you to regulate the flow of SCSI data on the VMEbus. The flow of data is regulated by limiting the activity allowed by the adapter each time it acquires the bus. The RF3570 provides two ways to limit its VME bus activity. You may specify a maximum number of *bytes* to transfer for each bus acquisition or a maximum number of *transfers* to perform for each bus acquisition.

BITS							
7	6	5	4	3	2	1	0
B/T	Throttle Count						

Figure 6-15: Bit-meanings of Bus Throttle Field

The bit-meanings of this field are as follows:

Throttle Count **Throttle Count**
The throttle count specifies either a number of bytes or a number of transfers, based on the B/T flag setting. There are specific ranges of values that are allowable for this field depending on your system bus width and whether you are throttling bytes or transfers. These ranges are given in the table on the next page.

Bus Width	Byte Count Throttle	Transfer Throttle
8-bit	1 to 32	1 to 32
16-bit	2 to 32	1 to 16
32-bit	4 to 32	1 to 8

Table 6-1: Valid Ranges of Throttle Count Values

NOTE: *If you use a value greater than the given range, the adapter will default to the maximum in the range. If you use zero as a value, it will default to the minimum.*

If you do not issue a General Options command to setup Throttle Count, the RF3570 defaults to Byte throttles and sets the throttle count to 32.

B/T **Byte or Transfer Throttle**
A **Transfer Throttle** counts the number of data transfers. Each time the adapter acquires the VMEbus, it will perform a number of transfers that is equal to or less than the transfer count value specified in Throttle Count.
A **Byte Count Throttle** counts the number of bytes transferred. Each time the adapter acquires the VMEbus, it will transfer a number of bytes that is equal to or less than the byte count value specified in Throttle Count.
0 ⇨ Transfer Count throttle method enabled; Throttle Count equals number of transfers.
1 ⇨ Byte Count Throttle method enabled; Throttle Count equals number of bytes.

Select Flags

The Select Flags are used to determine several operating factors for the RF3570 adapter. One factor determined by Select Flags is whether the adapter will perform VMEbus transfers in Block Mode. Other factors pertain to the SCSI bus: parity checking, odd-byte handling, and disconnect/reconnect.

The format of the Select Flags byte is as follows:

BITS							
7	6	5	4	3	2	1	0
0	0	0	0	OBH	BMT	PAR	DIS

Figure 6-16: Bit-meanings of Select Flags Field

DIS **Disconnect/Reconnect**
With this bit you are able to select whether the RF3570 adapter will allow peripherals to disconnect from the SCSI bus while performing a command.
0 ⇒ Disallow Disconnect/Reconnect.
1 ⇒ Allow Disconnect/Reconnect.
The default value for this bit is one — allow disconnects on all commands.

PAR **Check SCSI Bus Parity**
This bit allows you to select whether the adapter checks errors in parity that occurred on the SCSI bus. (The default is set by SCSI configuration jumpers. See Chapter 3.)
0 ⇒ Do not check for parity.
1 ⇒ Check for SCSI parity errors.

BMT **Block Mode Transfers**
This bit allows you to set the adapter to operate in Block Mode. This bit affects transfers of data. *It does not affect the transfer of control structures.*
0 ⇒ Data transfers will not be in Block Mode.
1 ⇒ Perform all data transfers in Block Mode regardless of Address Modifier used.

NOTE: *The RF3570 will perform Block Mode Transfers when this bit is set regardless of the Address Modifier used.*

OBH **Odd Byte Handling**
This bit selects how transfers with an odd transfer count or address will use the system bus.
0 ⇒ Use 8 bit system bus.
1 ⇒ Use Command Interface to transfer the odd bytes and use the high speed DMA port with current system bus width for the remaining data.

More information on Odd Byte Handling is available in Chapter 7.

Command Code

The command code for this command is 07H.

Unit Options (08H)

The Unit Options command allows you to individually tailor how the RF3570 adapter operates with each SCSI device on the bus.

Usually these options need be set only once, at power-up, to create the proper operating environment. Each unit you need to set-up requires a separate Unit Options command.

Command Format

The Parameter Block for the Unit Options command looks like this:

Address Offset	Byte Memory Address			
	Offset + 0	Offset + 1	Offset + 2	Offset + 3
00H	Command Identifier			
04H	Disconnect Timeout		Unit SCSI ID	Adapter ID = FFH
08H	Select Timeout		Retry Control	Retry Limit
0CH	Reserved	Reserved	Sense Count	Unit Flags
10H	Command = 08H	Reserved		
14H	Sense Byte #	Sense Byte #	Sense Byte #	Sense Byte #
18H	Sense Byte #	Sense Byte #	Sense Byte #	Sense Byte #

Selected Sense Bytes

Figure 6-17: Unit Options Command Parameter Block

Description of Fields

Each of the fields of the Parameter Block are described below:

Command Identifier

This field is used if you are issuing the Unit Options command through the structure of a Command List. It is used to identify the Status block returned from the command. It should be a unique value.

Adapter ID

This command performs no action on a specific SCSI device. It is a Board-control command and so this field must be FFH.

Unit SCSI ID

This field must contain a number from 0 - 7 representing a SCSI device on the bus. *Do not* use the RF3570 adapter SCSI ID you assign with the General Options command.

Disconnect Timeout

If you have enabled SCSI bus Disconnect/Reconnect with the General Options command (DIS bit of Select Flags is set), this field selects — for the specified unit — the amount of time the adapter will wait for a reconnect. If a reconnect does not occur within the time defined, the adapter will post a timeout error (1FH) in the appropriate command Status Block.

The time appropriate for a disconnect depends on the action and the unit performing the task (seeks, rewinds, etc.). For this reason, the Disconnect Timeout value is tailored per unit with a latitude of 100 milliseconds to almost two hours. You may also choose the option of no timeout for this unit. In that case, the device or SCSI bus must be reset to recover from the failure.

0 ⇔ No timeout occurs; SCSI bus or device must be reset to clear a failure.
Each increment ⇔ 100 milliseconds.

Retry Limit

If retries are enabled in the Retry Control field, (discussed next) this field specifies the maximum number of retries to attempt before reporting an error. Once the retry limit is reached, the error condition reported on the last retry is returned in the Status Block.

You must provide a value in this field or retries will not occur even if enabled (in the Retry Control field).

NOTE: *The sequentially recorded nature of tape media is such that positioning for retries requires extra commands. For this reason, do not enable the retry function for SCSI tape devices.*

Retry Control

This field is used to particularize the retry operation for this unit. This field allows you to select the type of errors to retry and how they are reported.

BITS							
7	6	5	4	3	2	1	0
0	0	0	RBE	RCE	RPE	ISB	INT

Figure 6-18: Bit-meanings of Retry Control Field

INT **Issue Interrupt**
When ISB bit is set, allows you to select whether the adapter posts an interrupt for retry Status Block(s) when returned. Valid only when ISB bit is set.
0 ⇒ Do not interrupt for retry Status Block(s).
1 ⇒ Interrupt when retry Status Block(s) returned.

ISB **Issue Status Block**
Allows you to select that a Status Block be returned for each command retry performed. Enabling this will cause *multiple* Status Blocks to be returned for a command that requires retries.
0 ⇒ Do not issue a Status Block for each retry.
1 ⇒ Issue a Status Block for each retry.

RPE **Retry Parity Errors**
Allows you to retry SCSI commands that failed with a parity error.
0 ⇒ Do not retry for parity errors.
1 ⇒ Retry parity errors.

RCE **Retry Command Errors**
Allows you to select whether the adapter will respond to a device error with a retry of the failed command.
0 ⇒ Do not retry for device errors.
1 ⇒ Retry commands that error from the device.

RBE **Retry Bus (SCSI) Errors**
Enables retries in the event that a command fails.
0 ⇒ Do not perform Bus Error retries.
1 ⇒ Perform Bus Error retries.

Select Timeout

According to SCSI specifications, a target must respond to initiator selection by asserting the BSY signal in response to SEL, with its ID on the bus. With this field you can select the amount of time that the RF3570 adapter will wait for response before timing out with an error.

The select timeout period is specified in 1 millisecond increments. The default value is 250 (FAH), which selects a period of 250 milliseconds.

The longest timeout period possible is 419 milliseconds, selected with a value of 419 (1A3H) Any larger value placed in this field will still select 419 milliseconds.

A value of 0 will select the shortest timeout period possible, which is about 1 millisecond.

Unit Flags

The Unit Flags field of the Unit Options Parameter Block is where you can define command operation for the unit. With bit settings you can address the issues of: synchronous data transfer, tagged commands, and use of the ATN signal.

BITS							
7	6	5	4	3	2	1	0
0	TAG	0	0	0	IAT	SYN	UNQ

Figure 6-19: Unit Flags Field of Unit Options Command

UNQ **Untagged Queuing**
The SCSI-1 specification allows issue of multiple, simultaneous, commands to a target that has multiple LUN's, one command per LUN. This bit allows you to select whether the RF3570 will queue commands to a target with multiple LUN's.
0 ⇨ Send one command at a time to this target.
1 ⇨ Queue commands to this target; it has multiple LUN's.

NOTE: *If you set the TAG flag (explained below) the adapter will ignore the UNQ bit. Setting the TAG bit implies that you are using a SCSI-2 device. The UNQ bit is intended for SCSI-1 devices.*

SYN **Synchronous Negotiation**
Either the initiator or the target may introduce negotiation for synchronous operation. How you set this bit determines whether the RF3570 adapter will attempt to negotiate with the device on the first command. (Even if you do not set this bit to allow negotiation, if the *target* begins negotiation, the adapter will respond.)
0 ⇒ Do not initiate negotiation for synchronous data transfer.
1 ⇒ Negotiate for synchronous data transfer during first SCSI command.

IAT **Inhibit ATN Signal**
With this bit you can select whether the adapter will assert the ATN signal during selection of a target. (Some older SCSI devices do not respond to the ATN signal and may stop executing commands if it is asserted.) Setting this bit will prevent the adapter from negotiating for synchronous transfers and SCSI-2 features. If you enable either of these features with this bit set, you will get errors.
0 ⇒ ATN *will* be asserted.
1 ⇒ ATN *will not* be asserted during target selection.

TAG **Tagged Queuing**
One of the design options offered with the SCSI-2 specification is the capability to issue multiple simultaneous commands to each logical unit. If you have a peripheral that is designed to this specification, you can use this bit to tell the RF3570 adapter to negotiate for tagged queuing. The tag "type" is specified in the Flags-2 field of the Pass-through command Parameter Block. Refer to Chapter 5.
0 ⇒ Tagged queuing operation not negotiated.
1 ⇒ Peripheral supports tagged queuing - negotiate for it.

Unit Options (08H)

Sense Count

The RF3570 adapter automatically responds to a device Check Condition with a SCSI Request Sense command. (Unless inhibited on a per command basis with the IRS bit of the Flags-1 field which is embedded in the SCSI Pass-through command Parameter Block. See Chapter 5.)

The Request Sense command, issued by the adapter to the device with the Check Condition, can garner Sense information of up to 256 bytes. The number returned depends on the number asked for in the SCSI Request Sense command. The Sense Count field is used to define the number of Sense bytes the adapter will ask for when automatically issuing a Request Sense command.

Valid values for this field are 0 to 32 (20H).

If more than 32 bytes of sequential sense information is requested, the count will be set to 8 and error code A1H will be returned.

Otherwise, the value you place in the Sense Count field is used in conjunction with the Selected Sense Bytes fields. There are three ways you can use the Sense Count and Selected Sense Bytes fields to get the result you want:

☆ **To receive the default — first eight bytes of Request Sense Data:**

- Place a zero value in Sense Count field.
- Place zeroes in all fields of Selected Sense Bytes.

☆ **To receive any eight bytes of Request Sense Data from the possible 256:**

- Place a zero in the Sense Count field.
- Place a number in each of the eight fields of the Selected Sense Bytes, in the order desired.

☆ **To receive any number of the first 32 bytes of Request Sense Data:**

(any number over eight will generate multiple Status Blocks for each Request Sense command the adapter issues.)

- Place the number desired in the Sense Count field.
- Selected Sense Bytes fields are ignored.

➡

If you wish to hand-pick more than eight bytes of Request Sense Data to be returned in a specific order, you must use the Extended Unit Options command; with it you can specify up to sixteen selected Request Sense bytes.

Command Code

The command code for this command is 08H.

Selected Sense Bytes

This part of the Unit Options Parameter Block is a combination of eight fields, each containing the byte number of one of 256 possible SCSI Sense bytes. These fields are only valid when the value of the Sense Count field of this command is zero.

If the Sense Count field contains a 0 *and* all eight fields that make up the Selected Sense Bytes field are set to 0, the default Sense bytes are returned

The default Sense bytes are bytes 0 to 7. Returned in a Status Block, they would appear as follows:

Byte Memory Address			
Offset + 0	Offset + 1	Offset + 2	Offset + 3
Command Identifier			
Reserved	SCSI Status	Error	Flags
0 = Class/Code	**1 = Segment**	**2 = SCSI Flags**	**3 = Info Byte**
4 = Info Byte	**5 = Info Byte**	**6 = Info Byte**	**7 = Ex Length**

Figure 6-20: Default Ordering of Sense Bytes

If the Sense Count field is set to 0, but there are non-zero values in the Selected Sense bytes fields, the byte numbers and order you define in the Selected Sense Bytes fields are used to return automatic Request Sense information for this device.

If you define more than eight bytes to be returned as Sense Data, you will receive two SCSI Pass-through command Status Blocks. See Chapter 5 for more information about the format of the Status Blocks that will be returned.

Example Use of Selected Sense Bytes Fields

In this example your device has valuable information, in bytes other than the first eight, that you wish to substitute:

- Byte 12 is an Additional Sense Code.
- Bytes 15 through 17 include additional error information.

Assume bytes 16 and 17 are pointers that you wish to read in as a word. You will arrange them in the group of Sense Bytes to make this easy.

For the example, the Sense Count field must be set to zero. The Selected Sense Bytes fields of the Unit Options command would look like this:

Address Offset	Byte Memory Address			
	Offset + 0	Offset + 1	Offset + 2	Offset + 3
00H	Command Identifier			
04H	Disconnect Timeout		Unit SCSI ID	Adapter ID = FFH
08H	Select Timeout		Retry Control	Retry Limit
0CH	Reserved	Reserved	**Sense Count = 0**	Unit Flags
10H	Command = 08H	Reserved		
14H	**0H (Class/Code)**	**01H (Segment)**	**02H (SCSI Flags)**	**03H (Info Byte)**
18H	**0CH (Byte 12)**	**0FH (Byte 15)**	**11H (Byte 17)**	**10H (Byte 16)**

Figure 6-21: Example Selection of Sense Bytes

Diagnostic/Selftest (09H)

Diagnostic/Self-Test (09H)

The Diagnostic/Self-Test command duplicates most of the tests performed at power-up. You can select which tests will be performed. The command returns a special Status Block with testing results.

This command writes test data into board memory.

The adapter will not execute the Diagnostic/Self-Test command until it completes all preceding commands in order to avoid writing over valid data in memory. Also, the adapter will not accept other commands while it is executing Diagnostic/Self-test. When the testing is done, the adapter will service all pending Channel Attentions.

Command Format

The Parameter Block for the Diagnostic/Self-test command has this format:

Address Offset	Byte Memory Address			
	Offset + 0	Offset + 1	Offset + 2	Offset + 3
00H	Command Identifier			
04H	Reserved			Adapter ID = FFH
08H	Reserved			
0CH	Reserved			
10H	Command = 09H	Test Flags	Reserved	
14H	Reserved			
18H	Reserved			

Figure 6-22: Diagnostic/Self-test Parameter Block

Description of Fields

Each field of the Diagnostic/Self-test Parameter Block is described below:

Command Identifier

The Command Identifier field is used to identify the Status Block associated with a Parameter Block. It must be a unique value.

Adapter ID

This field identifies the type of command being issued. This is a Board-control command . The value in this field must be FFH.

Test Flags

This field is used to determine which of the RF3570 board tests will be performed.

BITS							
7	6	5	4	3	2	1	0
0	EMX	STT	PSI	SBF	186	PCS	0

Figure 6-23: Diagnostics Command Test Flags Field

PCS **PROM Checksum Test**
This test performs a checksum of all bytes programmed in the EPROMs, compares it to the checksum stored in the highest EPROM memory location, and reports the result in a Status Block.
0 ⇨ Skip the PROM Checksum test.
1 ⇨ Perform the PROM Checksum test.

186 **80186 Processor Test**
This routine verifies that the internal registers of the 80186 can be loaded with a sequence of data patterns, checks the arithmetic and logical data handling for correct results and flags, confirms that the program jump instructions function, plus verifies the 80186 can read and write memory.
0 ⇨ Skip the 80186 Processor test.
1 ⇨ Perform the 80186 Processor test.

SBF **Short Burst FIFO Test**
This test writes various data patterns to the Short Burst FIFO Gate Array, then reads it back to verify. This test is repeated for each SBF mode.
0 ⇨ Skip the Short Burst FIFO test.
1 ⇨ Perform the Short Burst FIFO test.

PSI **Pipelined System Interface Test**
This test writes a variety of data patterns to the registers and counters of the Pipelined System Interface and then reads them back to verify. It also confirms that the address and transfer counters accurately reflect the proper DMA transfer mode. (i.e. adjusted for word or longword values).
0 ⇨ Skip the Piplined System Interface test.
1 ⇨ Perform the Pipelined System Interface test.

STT **SCSI Termination Test**
This test checks for a blown SCSI termination power fuse and also verifies a good voltage at TERM PWR pin on the SCSI connector.
0 ⇨ Skip the SCSI Termination test.
1 ⇨ Perform the SCSI Termination test.

EMX **Emulex Fast SCSI chip Test**
This test writes a progression of data patterns to the registers and counters of the Emulex Fast SCSI chip, then reads each back to verify the pattern.
0 ⇨ Skip the Emulex Fast SCSI chip test.
1 ⇨ Perform the Emulex Fast SCSI chip test.

Command Code

The command code for this operation is 09H.

Status Block Format

The Status Block returned from the Diagnostic/Self-test command looks like this:

Address Offset	Byte Memory Address			
	Offset + 0	Offset + 1	Offset + 2	Offset + 3
00H	Command Identifier			
04H	Reserved		Error	Flags
08H	Reserved			
0CH	Reserved			

Figure 6-24: Diagnostics Command Status Block

Description of Fields

The Status Block returned from a Diagnostic/Self-test command contains these fields:

Command Identifier

The value in this field will be identical to the Parameter Block Value.

Flags Byte

This byte is used to indicate command completion.:

BITS							
7	6	5	4	3	2	1	0
CC	ERR	0	0	0	0	0	0

Figure 6-25: Bit-meanings of Flags Byte in Status Block

ERR **Error Status**
Indicates whether the command completed with an error.
0 ⇨ No error occurred.
1 ⇨ An error occurred.

CC **Command Complete**
Indicates if the command has finished.
0 ⇨ Command not complete.
1 ⇨ Command complete.

Error

This field contains error codes specific to the tests performed as well as those error codes generated from system and SCSI bus errors. See Appendix A.

SCSI Hard Reset (10H)

The SCSI Hard Reset command simply asserts the reset (RST) signal on the SCSI bus. It will abort all commands being executed, even those being executed by devices that are currently disconnected. For this reason, it should only be used as a drastic means of recovery.

Command Format

The Parameter Block for the SCSI Hard Reset command has this format:

Address Offset	Byte Memory Address			
	Offset + 0	Offset + 1	Offset + 2	Offset + 3
00H	Command Identifier			
04H	Reserved			Adapter ID = FFH
08H	Reserved			
0CH	Reserved			
10H	Command = 10H	Reserved	Reserved	
14H	Reserved			
18H	Reserved			

Figure 6-26: SCSI Hard Reset Parameter Block

Description of Fields

Each field of the SCSI Hard Reset Parameter Block is described below:

Command Identifier

The Command Identifier field is used to identify the Status Block associated with a Parameter Block. It must be a unique value.

Adapter ID

This field identifies the type of command being issued. This is a Board-control command . The value in this field must be FFH.

Command Code

The command code for this Board-control command is 10H.

Board Information (15H)

Board Information (15H)

This command returns a special Board Information Data Structure (in addition to the normal status block), that provides information about the current configuration of the RF3570 adapter. The structure, returned at the address you specify in the Board Information command Parameter Block, includes information about the options set with the General Options and Unit Options commands as well as the configuration set in the hardware. It can be used as a snapshot of the board set-up. Additionally, the same information received from the Identify command is included in the structure returned.

This command can be issued with either a Single Command structure or in a Command List.

Command Format

The Parameter Block for the Board Information command has this format:

Address Offset	Byte Memory Address			
	Offset + 0	Offset + 1	Offset + 2	Offset + 3
00H	Command Identifier			
04H	Reserved		Addr. Mod.	Adapter ID = FFH
08H	VME Memory Address			
0CH	Reserved			
10H	Command = 15H	Reserved		
14H	Reserved			
18H	Reserved			

Figure 6-27: Board Information Command Parameter Block

Description of Fields

Each field of the Board Information Parameter Block is described below:

Command Identifier

The Command Identifier field is used to identify the Status Block associated with a Parameter Block. It must be a unique value.

Adapter ID

This is a Board-control command . The value in this field must be FFH.

Address Modifier

This is the value to be used by the adapter on the Address Modifier lines when writing the Board Information Data Structure to memory.

VME Memory Address

This is the system memory address at which you wish to have the adapter place the Board Information Data Structure.

Command Code

The value for this field will always be 15H to indicate a Board Information command.

Board Information (15H)

The format of the Data Structure returned from the Board Information command is illustrated on the next page.

Board Information Data Structure

The Data Structure returned by the Board Information command has two parts. The first part of the structure consists of a special header primarily containing General Board Information (including hardware configuration).

The second part of the returned structure consists of eight blocks of data that describe the Unit Options set for each target attached to the RF3570. The first block corresponds to SCSI ID 0, the second to SCSI ID 1, etc.

Each of the fields of this Data Structure are described below and on the following pages:

Address Offset	Byte Memory Address			
	Offset + 0	Offset + 1	Offset + 2	Offset + 3
00H	FW Rev.	Eng. Rev.	Bus Throttle	Select Flags
04H	Option Flags	Day	Month	Year
08H	Firmware Number of Even Prom			
0CH	Reserved		SCSI Cfig Block	Termination
10H	Reserved			
14H	Sense Count	Retry Limit	Disconnect Timeout	
18H	Synch Offset	Synch Period	Device Flags	
1CH	Reserved			
20H	Sense Byte #	Sense Byte #	Sense Byte #	Sense Byte #
24H	Sense Byte #	Sense Byte #	Sense Byte #	Sense Byte #
28H	Sense Byte #	Sense Byte #	Sense Byte #	Sense Byte #
2CH	Sense Byte #	Sense Byte #	Sense Byte #	Sense Byte #
:				
F0H	Target information repeated seven more times			

Figure 6-28: Board Information Data Structure

Description of Fields

The Data Structure returned from a Board Information command contains the fields shown on the following pages.

Select Flags

This field displays the selections currently valid for the Select Flags field of the General Options command. The format of the Select Flags byte is as follows:

BITS							
7	6	5	4	3	2	1	0
0	0	0	0	OBH	BMT	PAR	DIS

Figure 6-29: Board Information - Select Flags Field

DIS **Disconnect/Reconnect**
This bit indicates whether the RF3570 adapter will allow peripherals to disconnect from the SCSI bus while performing a command.
0 ⇔ Disconnect/Reconnect is disallowed.
1 ⇔ Disconnect/Reconnect is allowed.

PAR **Check SCSI Bus Parity**
This bit indicates whether the adapter reports errors in parity that occurred on the SCSI bus.
0 ⇔ Parity will not be checked.
1 ⇔ SCSI parity will be checked.

BMT **Block Mode Transfers**
This bit indicates whether the adapter is set to operate in Block Mode. This bit affects transfers of data. *It does not affect the transfer of control structures.*
0 ⇔ Data transfers will not be in Block Mode.
1 ⇔ All data transfers occur in Block Mode regardless of Address Modifier used.

OBH **Odd Byte Handling**
This bit indicates how transfers with an odd transfer count or address will use the system bus.
0 ⇔ 8 bit system bus is used.
1 ⇔ Command Interface used to transfer the odd bytes and the high speed DMA port with current system bus width used for the remaining data.

Bus Throttle

This field *displays* the currently selected throttle option in an identical manner as the field used to *select* throttle in the General Options command.

BITS							
7	6	5	4	3	2	1	0
B/T	Throttle Count						

Figure 6-30: Bit-meanings of Bus Throttle Field

Throttle Count This part of the Throttle field represents the throttle count currently selected for the board. It is either the *number of transfers* that the adapter will perform or the *number of bytes* transferred, before the adapter surrenders the system bus.

B/T **Byte or Transfer Throttle**
This bit identifies the type of throttle count currently selected for transfer of data.
0 ⇔ Transfer Count throttle method enabled; throttle count equals number of transfers.
1 ⇔ Byte Count Throttle method enabled; throttle count equals number of bytes.

Engineering Revision

This field indicates the revision level used by Ciprico to identify its source code. This field is useful only for Ciprico and should not be used to identify the revision level of the board firmware.

Firmware Revision

This field contains a value that identifies the released level of Firmware for the product.

Day, Month, Year

These three fields, when combined, indicate the date (in hex) that the firmware in the EPROM was compiled.

Option Flags

The Options Flags byte returned here is identical to the byte returned with the Identify command. It is used to indicate whether a particular option is enabled or available in the revision of firmware installed.

BITS							
7	6	5	4	3	2	1	0
SCSI ID			RST	0	TAG	0	0

Figure 6-31: Bit-meanings of Option Flags Field

The bit-meanings of the Option Flags field are as follows:

TAG **Tagged Queuing**
Indicates whether the firmware installed in the board supports tagged queuing.
0 ⇒ Tagged queuing *not* supported.
1 ⇒ Tagged queuing supported.

NOTE: *Even if support of Tagged Queuing is indicated here, you will still need to* ***enable*** *it with the Unit Options command to use it.*

RST **Reset Jumper**
Indicates whether the SCSI bus will be reset at power-up and when adapter is reset through the reset port.
0 ⇒ SCSI bus will not be reset.
1 ⇒ SCSI bus will be reset.

SCSI ID **SCSI ID of Adapter**
Indicates the current ID of the adapter.
0 ⇒ Bit of ID not set.
1 ⇒ Bit of ID set.

Bit→ SCSI ID ↓	7	6	5
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

Figure 6-32: Adapter SCSI ID Bits

Firmware Number of Even EPROM

There is a set of two EPROMs on the board. This number is the checksum of the EVEN EPROM.

Termination

This field returns the status of the SCSI fuse on the RF3570 and the condition of terminator power on the SCSI cable connector. It has this format:

BITS							
7	6	5	4	3	2	1	0
0	0	0	0	0	BSF	BTP	0

Figure 6-33: Bit-meanings of Termination Field

BTP **Bad SCSI Termination Power**
Indicates whether there is sufficient voltage on the TERM PWR pin of the SCSI connector.
0 ⇒ Proper 4 - 5.25 volts available.
1 ⇒ Less than 4 volts available.

BSF **Blown SCSI Fuse**
Indicates whether the Termination Power Fuse on the adapter is intact.
0 ⇒ Fuse is good.
1 ⇒ Fuse is blown.

SCSI Cfig Block

This field reports the status of the SCSI Configuration Block of hardware jumpers. You can use it to determine how the jumpers are set without removing the board from your system:

BITS							
7	6	5	4	3	2	1	0
1 - 2	3 - 4	5 - 6	7 - 8	9 - 10	11 -12	13 - 14	15 -16

Figure 6-34: SCSI Configuration Jumpers

Each bit corresponds to a hardware jumper.

0 ⇒ Jumper is not present.
1 ⇒ Jumper is installed.

The purpose for each jumper is described in the table below:

Pins	Description	Default
1 - 2	Reserved	OUT
3 - 4		OUT
5 - 6		OUT
7 - 8	Enable SCSI Parity Checking	OUT = Selected
9 - 10	SCSI bus Reset at Power-up	OUT =Selected
11 - 12	SCSI ID Bit 2	OUT = 0
13 - 14	SCSI ID Bit 1	OUT = 0
15 - 16	SCSI ID Bit 0	OUT = 0

Table 6-2: SCSI Cfig Block - Default Jumper Settings

Target - Disconnect Timeout

If Disconnects are enabled for the target with the General Options command, this field represents the amount of time the adapter will wait for a reconnect before taking error recovery action. For more information, refer to the section of this chapter that explains the General Options, Unit Options, and Extended Unit Options commands.

Board Information (15H)

Target - Retry Limit

This field specifies the maximum number of retries the RF3570 will attempt before reporting an error. This value is set for each target with the Unit Options or Extended Unit Options command.

Target - Sense Count

This field reports the number of Sense bytes that the adapter will return for this target. This number is defined by the Unit Options or Extended Unit Options commands.

Target - Device Flags

The Device Flags field is a word wide. It reports the options that you have selected for this target with the Unit Options or Extended Unit Options command. The format is as shown below:

BITS															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	SS	0	RBE	RCE	RPE	ISB	INT	0	TQE	0	0	0	IAT	SYN	UNQ

Figure 6-35: Bit-meanings of Device Flags Field

UNQ **Untagged Queuing**
This bit indicates whether the RF3570 will queue commands to a target with multiple LUN's.
0 ⇨ One command at a time will be sent to this target.
1 ⇨ Commands to this target will be queued; it has multiple LUN's.

SYN **Synchronous Negotiation**
This bit indicates whether negotiations for synchronous transfers will be initiated for this target.
0 ⇨ Synchronous transfer negotiation will *not* be initiated by adapter.
1 ⇨ Adapter will initiate synchronous transfer negotiations.

IAT **Inhibit ATN Signal**
This bit indicates if the adapter is to assert the ATN signal.
0 ⇨ Adapter will assert ATN for this target.
1 ⇨ Adapter will *not* assert ATN for this target.

TQE **Tagged Queuing Enabled**
This bit indicates whether you have enabled tagged queuing operation with this target.
0 ⇨ Tagged Queuing is not enabled.
1 ⇨ Tagged Queuing is enabled.

NOTE: *The setting of this bit indicates only that you have enabled tagged queuing operation for this target using the Unit Options (or Extended Unit Options) command. It does not indicate that the device is actually operating in that mode.*

INT **Issue Interrupt**
Used with the ISB bit, the INT bit indicates whether the adapter will post an interrupt for retry Status Block(s) returned for this target.
0 ⇨ Will *not* interrupt for retry Status Block(s).
1 ⇨ Will interrupt when retry Status Block(s) returned.

ISB **Issue Status Block**
This bit indicates whether a Status Block will be returned for each command retry performed on this target.
0 ⇨ Will *not* issue a Status block for each retry.
1 ⇨ Will issue a Status Block for each retry.

RPE **Retry Parity Errors**
This bit indicates whether the adapter will retry SCSI commands that failed with a parity error.
0 ⇨ Will *not* retry for parity errors.
1 ⇨ Will retry parity errors.

Board Information (15H)

RCE **Retry Command Errors**
This bit indicates whether the adapter will respond to a device error with a retry of the failed command.
0 ⇒ Will *not* retry for device errors.
1 ⇒ Will retry commands that error from the device.

RBE **Retry (SCSI) Bus Errors**
This bit indicates whether the adapter will retry a command that fails.
0 ⇒ Will *not* retry SCSI Bus Errors.
1 ⇒ Will retry commands that failed due to SCSI Bus Errors.

SS **Selected Sense**
This bit indicates whether you are using the Selected Sense byte option to return up to 16 bytes of specially ordered and selected Sense data.
0 ⇒ Sense data defaults to first eight bytes returned from the device.
1 ⇒ Selected Sense option in effect; Sense data returned according to selection.

Target - Synch Period

The Synchronous Transfer Period is negotiated between initiator and target when the target is able to operate in synchronous mode. The common value is agreed upon via Messages between initiator and target. The Synchronous Transfer Period is the minimum time allowed between leading edges of each successive REQ pulse and each successive ACK pulse.

The transfer period (in nanoseconds) is calculated by multiplying the negotiated value by 4:

Transfer period(nanoseconds) = negotiated value X 4

Target - Synch Offset

This value is negotiated between initiator and target when the target is able to operate in synchronous mode. The common value is agreed upon via Messages between initiator and target. The Synch Offset is the maximum number of REQ pulses allowed to be outstanding before the corresponding ACK pulses are received. This value is usually a function of the buffering available on the device.

Target - Selected Sense Bytes

If the Selected Sense option is in effect, these 16 fields indicate which Sense bytes, in which order, will be returned from the device.

If you have enabled 16 Selected Sense Bytes using the Extended Unit Options command you will receive the Selected Sense bytes in two Status Blocks. In that case, the first eight Sense byte Fields displayed in the Board Information Data Structure will be returned with the first Status Block. The last eight will appear in the second Status Block.

Extended Board Statistics (16H)

The RF3560 adapter keeps track of several types of board statistics. With the Extended Board Statistics command you can retrieve the extended set of statistics kept by the adapter. The statistics are valid only as long as power is kept to the system. This command reports the statistics by recording them in the area of system memory specified in the command Parameter Block. This statistics structure is returned in addition to the normal Status Block.

You can clear the internal statistics table at any time by issuing the command with the CLR bit of the Options field set. No data transfer occurs when the CLR bit is set.

Command Format

The Parameter Block format for the Extended Board Statistics command looks like this:

Address Offset	Byte Memory Address			
	Offset + 0	Offset + 1	Offset + 2	Offset + 3
00H	Command Identifier			
04H	Reserved		Addr. Mod.	Adapter ID = FFH
08H	VME Memory Address			
0CH	Reserved			
10H	Command = 16H	Options	Reserved	
14H	Reserved			
18H	Reserved			

Figure 6-36: Extended Board Statistics Parameter Block

Description of Fields

Each of the fields of the Parameter Block are explained below:

Command Identifier

The Command Identifier field is used to identify the Status Block associated with a Parameter Block. It must be a unique value.

Adapter ID

The Extended Board Statistics command is a Board-control command. This field must be set to FFH.

Address Modifier

This is the value to be used by the adapter on the Address Modifier lines when writing extended statistics to memory. This field is not used when the CLR bit is set.

VME Memory Address

This is the system memory address at which you wish to have the adapter place the Extended Statistics Data Structure. This field is not used when the CLR bit is set.

Options

The options field is one byte wide with the following format and bit-meanings:

BITS							
7	6	5	4	3	2	1	0
0							CLR

Figure 6-37: Extended Board Statistics Options Field

CLR This bit allows you to clear the internal statistic values currently stored. (Statistics returned by both the 06H and 16H commands will be cleared.) It is useful if you wish to keep track of statistics during a specific period. The only other time statistic values get cleared is during adapter reset.
0 ⇒ Command is used to return internal statistics to specified system memory address.
1 ⇒ Command is used to clear the internal statistics table.

Command Code

The value for this field will always be 16H to indicate an Extended Board Statistics command.

The format of the Extended Data Structure returned from the Extended Board Statistics command is illustrated on the next page.

Extended Statistics Data Structure

The Extended Statistics Data Structure returned from an Extended Board Statistics command will have this format:

General Board Information Header

Per Target Information; Targets 0-7

Address Offset	Byte Memory Address			
	Offset + 0	Offset + 1	Offset + 2	Offset + 3
00H	Total Commands Issued			
04H	Number of SCSI Bad Status Conditions			
08H	Number of SCSI Parity Errors			
0CH	Reserved			
10H	Reserved			
14H	SCSI Target ID 0: Number of Commands			
18H	Queue Full Count		Maximum Queued	
1CH	Reserved			
.	.			
.	.			
.	.			
70H	SCSI Target ID 7: Number of Commands			
74H	Queue Full Count		Maximum Queued	
78H	Reserved			

Figure 6-38: Extended Board Statistics Data Structure

Description of Fields

The Extended Statistics Data Structure returned from the Extended Board Statistics command contains these fields:

Total Commands Issued

This field reports the number of commands, SCSI Pass-through and Board-control, that have been processed by the adapter since power-up, reset, or the CLR bit option was used.

Number of SCSI Bad Status Conditions

This field reports the number of times a status other than Good was reported by all SCSI devices.

Number of SCSI Parity Errors

This field reports the total number of SCSI Parity Errors that have occurred since power-up, reset, or the CLR bit option was used.

Per SCSI Target — Number of Commands

This field is repeated for each of the eight possible SCSI Target ID's. It reports the number of commands received by that target since power-up, reset, or the CLR bit option was used.

Per SCSI Target — Queue Full Count

This field reports the number of times a target reported a Queue Full status. It may be useful in evaluating the performance of tagged queuing operation with a SCSI-2 target.

Per SCSI Target — Maximum Queued

This field reports the maximum number of commands that were queued to a target and may also be useful in evaluating the performance of SCSI-2 tagged queuing operation.

Extended Unit Options (18H)

This command is designed as an enhancement to the Unit Options command. The primary difference between the commands is the opportunity to select 16 bytes of Sense Data to be returned per unit, instead of the eight offered in the Unit Options command.

Another difference is that the Extended Unit Options for the SCSI device are built into a Data Structure whose address is simply passed in the command.

Command Format

The format of the Extended Unit Options command is as follows:

Address Offset	Byte Memory Address			
	Offset + 0	Offset + 1	Offset + 2	Offset + 3
00H	Command Identifier			
04H	Reserved		Addr. Mod.	Adapter ID = FFH
08H	VME Memory Address			
0CH	Reserved			
10H	Command = 18H	Reserved		
14H	Reserved			
18H	Reserved			

Figure 6-39: Extended Unit Options Parameter Block

Description of Fields

Each of the fields of the Parameter Block are described below:

Command Identifier

The Command Identifier field is used to identify the Status Block associated with a Parameter Block. It must be a unique value.

Adapter ID

This is a Board-control command. The value in this field must be FFH.

Address Modifier

This is the value to be used by the adapter on the Address Modifier lines when reading the Extended Unit Options Data Structure from memory.

VME Memory Address

This is the system memory address at which you have built the Extended Unit Options Data Structure.

Command Code

The value for this field will always be 18H to indicate a Extended Unit Options command.

The format of the Data Structure to be used for the Extended Unit Options command is illustrated on the following pages.

Extended Unit Options Data Structure

The Data Structure that you will use to pass the Extended Unit Option parameters to the RF3570 adapter is similar to the format of the Unit Options Parameter Block.

The Data Structure must contain these fields:

Address Offset	Byte Memory Address				
	Offset + 0	Offset + 1	Offset + 2	Offset + 3	
00H	Disconnect Timeout		Unit SCSI ID	0	
04H	Select Timeout		Retry Control	Retry Limit	
08H	Reserved	Reserved	Sense Count	Unit Flags	
0CH	Sense Byte #	Sense Byte #	Sense Byte #	Sense Byte #	Status Block #1
10H	Sense Byte #	Sense Byte #	Sense Byte #	Sense Byte #	
14H	Sense Byte #	Sense Byte #	Sense Byte #	Sense Byte #	Status Block #2
18H	Sense Byte #	Sense Byte #	Sense Byte #	Sense Byte #	
1CH	0				

Selected Sense Bytes

Figure 6-40: Extended Unit Options Data Structure

Description of Fields

Each of the fields of the Data Structure are described below:

Unit SCSI ID

This field must contain a number from 0 - 7 representing a SCSI Unit on the bus. *Do not* use the RF3570 adapter SCSI ID you assign with the General Options command. (If you do you will receive the 02H (Bad Unit) error code.)

Disconnect Timeout

If you have enabled SCSI bus Disconnect with the General Options command, this field selects — for the specified unit — the amount of time the adapter will wait for a reconnect. If a reconnect does not occur within the time defined, the adapter will post a timeout error (1FH) in the appropriate command Status Block.

The time appropriate for a disconnect depends on the action and the unit performing the task (seeks, rewinds, etc.). For this reason, the Disconnect Timeout value is tailored per unit with a latitude of 100 milliseconds to almost two hours. You may also choose the option of no timeout for this unit. In that case, the device or SCSI bus must be reset to recover from the failure.

0 ⇒ No timeout occurs; SCSI bus or device must be reset to clear a failure.
Each increment ⇒ 100 milliseconds.

Retry Limit

If retries are enabled in the Retry Control field, (discussed next) this field specifies the maximum number of retries to attempt before reporting an error. Once the retry limit is reached, the error condition reported with the last retry is returned in the Status Block.

You must provide a value in this field or retries will not occur even if enabled (in the Retry Control field).

NOTE: *The sequentially recorded nature of tape media is such that positioning for retries requires extra commands. For this reason, do not enable the retry function for SCSI tape devices.*

Retry Control

This field is used to particularize the retry operation for this unit. This field allows you to select the type of errors to retry and how they are reported.

BITS							
7	6	5	4	3	2	1	0
0	0	0	RBE	RCE	RPE	ISB	INT

Figure 6-41: Bit-meanings of Retry Control Field

INT **Issue Interrupt**
When ISB bit is set, allows you to select whether the adapter posts an interrupt for retry Status Block(s) when returned.Valid only when ISB is set.
0 ⇒ Do not interrupt for retry Status Block(s).
1 ⇒ Interrupt when retry Status Block(s) returned.

➡

ISB **Issue Status Block**
Allows you to select that a Status Block be returned for each command retry performed. Enabling this will cause *multiple* Status Blocks to be returned for a command that requires retries.
0 ⇒ Do not issue a Status Block for each retry.
1 ⇒ Issue a Status Block for each retry.

RPE **Retry Parity Errors**
Allows retries of SCSI commands that failed with a parity error.
0 ⇒ Do not retry for parity errors.
1 ⇒ Retry parity errors.

RCE **Retry Command Errors**
Allows you to select whether the adapter will respond to a device error with a retry of the failed command.
0 ⇒ Do not retry for device errors.
1 ⇒ Retry commands returning bad SCSI status.

RBE **Retry Bus (SCSI) Errors**
Enables retries in the event that a command fails.
0 ⇒ Do not perform Bus Error retries.
1 ⇒ Perform Bus Error retries.

Select Timeout

According to SCSI specifications, a target must respond to initiator selection by asserting the BSY signal in response to SEL, with its ID on the bus. With this field you can select the amount of time that the RF3570 adapter will wait for response before timing out with an error.

The Select Timeout period is specified in increments of 1 millisecond. The default value is 250 (FAH), which selects a period of 250 milliseconds.

The longest timeout period possible is 419 milliseconds, selected with a value of 419 (1A1H). Any larger value placed in this field will still select 419 milliseconds.

A value of 0 will select the shortest timeout period possible, which is about 1 millisecond.

Unit Flags

The Unit Flags field of the Extended Unit Options Parameter Block is where you can define command operation for the unit. With bit settings you can address the issues of: synchronous data transfer, tagged commands, and use of ATN signal.

BITS							
7	6	5	4	3	2	1	0
0	TAG	0	0	0	IAT	SYN	UNQ

Figure 6-42: Unit Flags Field of Extended Unit Options

UNQ **Untagged Queueing**
The SCSI-1 specification allows issue of multiple, simultaneous, commands to a target that has multiple LUN's, one command per LUN. This bit allows you to select whether the RF3570 will queue commands to a target with multiple LUN's.
0 ⇒ Send one command at a time to this target.
1 ⇒ Queue commands to this target; it has multiple LUN's.

NOTE: *If you set the TAG flag (explained on next page) the adapter will ignore the UNQ bit. Setting the TAG bit implies that you are using a SCSI-2 device. The UNQ bit is intended for SCSI-1 devices.*

SYN **Synchronous Negotiation**
Either the initiator or the target may introduce negotiation for synchronous operation. How you set this bit determines whether the RF3570 adapter will attempt to negotiate with the device on the first command. Even if you do not set this bit to allow negotiation, if the target begins negotiation, the adapter will respond.
0 ⇒ Do not initiate negotiation for synchronous data transfer.
1 ⇒ Negotiate for synchronous data transfer during first command.

IAT **Inhibit ATN Signal**
With this bit you can select whether the adapter will assert the ATN signal during selection of a target. (Some older SCSI devices do not respond to the ATN signal and may stop executing if it is asserted.) Setting this bit will prevent the adapter from negotiating for synchronous transfers and SCSI-2 features. If you enable either of these features with this bit set, you will get errors.
0 ⇔ ATN *will* be asserted.
1 ⇔ ATN *will not* be asserted during target selection.

TAG **Tagged Commands**
One of the design options offered with the SCSI-2 specification is the capability to issue multiple simultaneous commands to each logical unit. If you have a peripheral that is designed to this specification, you can use this bit to tell the RF3570 adapter to negotiate for queued tagged commands. The tag "type" is specified in the Flags-2 field of the Pass-through command Parameter Block. Refer to Chapter 5.
0 ⇔ Tagged commands not negotiated.
1 ⇔ Peripheral supports tagged commands - negotiate for it.

Sense Count

The RF3570 adapter automatically responds to a device Check Condition with a SCSI Request Sense command. (Unless inhibited on a per command basis with the IRS bit of the Flags-1 field which is embedded in the SCSI Pass-through command Parameter Block. See Chapter 5.)

The Request Sense command, issued by the adapter to the device with the Check Condition, can garner Sense information of up to 256 bytes. The number returned depends on the number asked for in the SCSI Request Sense command that is issued by the adapter. The Sense Count field is used to define the number of Sense bytes the adapter will ask for when automatically issuing a Request Sense command.

Valid values for this field are 0 to 32 (20H).

If more than 32 bytes of *sequential* sense information is requested, the count will be set to 8 and error code A1H will be returned.

Otherwise, the value you place in the Sense Count field is used in conjunction with the Selected Sense Bytes Fields. There are three ways you can use the Sense Count and Selected Sense Bytes fields to get the result you want:

☆ **To receive the default — first eight bytes of Request Sense Data:**

- Place a zero value in Sense Count field.
- Place zeroes in all fields of Selected Sense Bytes.

☆ **To receive any sixteen bytes of Request Sense Data from the possible 256:**

- Place a zero in the Sense Count field.
- Place the Sense Byte number in each of the sixteen fields of the Selected Sense Bytes, in the order desired.

☆ **To receive any number of the first 32 bytes of Request Sense Data** (any number over eight will generate multiple Status Blocks for each Request Sense command the adapter issues.):

- Place the number desired in the Sense Count field.
- Selected Sense Bytes fields are ignored.

Selected Sense Bytes

This part of the Extended Unit Options Parameter Block is a combination of sixteen fields, each containing the byte number of one of 256 possible SCSI Sense bytes. These fields are only valid when the value of the Sense Count field of the Data Structure is zero.

If the Sense Count field contains a 0 *and* all sixteen fields that make up the Selected Sense Bytes section of the Data Structure are set to 0, the default of eight Sense bytes is returned.

The eight default Sense bytes are displayed as follows:

Byte Memory Address			
Offset + 0	Offset + 1	Offset + 2	Offset + 3
Command Identifier			
Reserved	SCSI Status	Error	Flags
0 = Class/Code	1 = Segment	2 = SCSI Flags	3 = Info Byte
4 = Info Byte	5 = Info Byte	6 = Info Byte	7 = Ex Length

Figure 6-43: Default Ordering of Sense Bytes

If the Sense Count field is set to 0, but there are non-zero values in the Selected Sense bytes fields, the byte numbers

and order you define in the Selected Sense Bytes fields are used to return automatic Request Sense information for this device.

If you define more than eight bytes to be returned as Sense Data, you will receive two SCSI Pass-through command Status Blocks.

See Chapter 5 for more information about the format of the Status Blocks that will be returned.

Chapter

7

Usage

Chapter 7 — Details of Usage

Introduction

The previous chapters informed you about the details you will need to install the board, set up command structures, and issue commands to the board and the peripherals attached to it. This chapter will provide information you can use to make decisions about how the RF3570 adapter can be used in your system. It explains in detail how and why to use some of the features of the adapter.

Summary

This chapter explains how to use the following operations for the RF3570 adapter:

- Byte and Word swapping of data and command structures.
- Scatter/Gather operation.
- Odd Byte Handling.
- Tagged Queuing.

This chapter is useful to use as you are making decisions about your system and set-up.

Byte and Word Swapping

The RF3570 adapter is flexible enough to allow you to request Byte Swapping, Word Swapping or both, during transfers of data structures, command structures or both.

This section will first give you some examples to explain why the swapping feature is offered to you, then provide a perspective on implementation with the RF3570 for both Command Structures (used by the RF3570) and data.

Need for Swapping

The memory architecture of your system depends on the type of processor you are using. There are several major manufacturers that each use different arrangements of bytes, words, and double-words in memory. The two most common formats are: Intel and Motorola. Their differences are illustrated in the example below:

Example of Memory Differences

If you were to define the following variables in a program:

char byte[ ] = "test";
int word[2] = {0x0102, 0x0304};
double dword = 0xAABBCCDD;

They would appear this way in memory (beginning at location *n*) for each respective format:

variable	Byte # of memory	Intel Order				Motorola Order			
byte	*n*	t	e	s	t	t	e	s	t
word	n + 4	02	01	04	03	01	02	03	04
dword	n + 8	DD	CC	BB	AA	AA	BB	CC	DD

Figure 7-1: Variations in Intel and Motorola Memory

Essentially the two processors store a *string of bytes* in the same order. However, the bytes that make up *word* and *double-word* information are stored in an order *opposite* to each other:

Processor Type	Byte of word or double-word	Order in memory
Motorola	LSB *	stored in highest memory address
	MSB *	stored in lowest memory address
Intel	LSB	stored in lowest memory address
	MSB	stored in highest memory address

* LSB = Least Significant Byte of a word or double-word value
* MSB = Most Significant Byte of a word or double-word value

Figure 7-2: Explanation of Memory Storage Differences

There are two types of information stored in system memory that the RF3570 adapter must correctly access: Command Structures and Data.

Throughout this manual, the Parameter and Status Blocks that make up the command structures are represented double-word wide, in Motorola ordering.

If you have an Intel processor, you will need to translate these displays to your format. The following example will help explain this.

Example Pass-through Command

The information on the following page would be required to complete a parameter block that issues a SCSI Pass-through command. The example shown on the next page uses:

- Logical block size is 128 bytes per block.
- Extended Write SCSI command.
- Command issued to LUN 1 of Target 2.
- Transferring data from VME memory address 5A9320H.
- Transferring data to Logical block 8F4E9H on the device.
- Parameter Block written to memory address 89FF4H.

Example parameters for a Pass-through command:

Name of Parameter Block field	Value used for example	Comments
Command Identifier	04030201H	
Target ID	02H	
Address Modifier	3EH	
Flags-1	86H	
Flags-2	01H	
VME Memory Address	005A9320H	
Transfer Count	00010280H	
SCSI Command Byte 0	2AH	Extended Write Command
SCSI Command Byte 1	40H	Logical Unit 1
SCSI Command Byte 2	00H	Logical Block Address (MSB)
SCSI Command Byte 3	08H	Logical Block Address
SCSI Command Byte 4	F4H	Logical Block Address
SCSI Command Byte 5	E9H	Logical Block Address (LSB)
SCSI Command Byte 6	00	Reserved
SCSI Command Byte 7	02H	Transfer Count (MSB)
SCSI Command Byte 8	05H	Transfer Count (LSB)
SCSI Command Byte 9	00	Control Byte
SCSI Command Byte 10	00	Not Used
SCSI Command Byte 11	00	Not Used

If written to VME memory in the order given above, these command parameters would be written differently (as shown below) for both types of processors:

Intel Ordering

89FF4	01	02	03	04
89FF8	02	3E	86	01
89FFC	20	93	5A	00
8A000	80	02	01	00
8A004	08	00	40	2A
8A008	02	00	E9	F4
8A00C	00	00	00	05

Motorola Ordering

89FF4	04	03	02	01
89FF8	01	86	3E	02
89FFC	00	5A	93	20
8A000	00	01	02	80
8A004	2A	40	00	08
8A008	F4	E9	00	02
8A00C	05	00	00	00

Figure 7-3: Intel versus Motorola Ordering

It becomes apparent when examining the memory displays, that the Command Structures of the RF3570 though divided into a variety of byte, word, and double-word fields, must be considered as a grouping of double-words.

In a Motorola environment, the Parameter Block would remain as graphically represented throughout this manual, and as shown below:

Address Offset	Byte Memory Address			
	Offset + 0	Offset + 1	Offset + 2	Offset + 3
00H	Command Identifier = 04030201H			
04H	Flags-2 = 01H	Flags-1 = 86H	Addr. Mod = 3EH	Target ID = 02H
08H	VME Memory Address = 005A9320H			
0CH	Transfer Count = 00010280H			
10H	0 = 2AH	1 = 40H	2 = 0	3 = 08H
14H	4 = F4H	5 = E9H	6 = 0	7 = 02H
18H	8 = 05H	9 = 0	10 = 0	11 = 0

Figure 7-4: Motorola Ordered Parameter Block

However, if you have an Intel processor, you must reinterpret the representations in this manual to an order like this:

Address Offset	Byte Memory Address			
	Offset + 3	Offset + 2	Offset + 1	Offset + 0
00H	Command Identifier = 04030201H			
04H	Flags-2 = 01H	Flags-1 = 86H	Addr. Mod = 3EH	Target ID = 02H
08H	VME Memory Address = 005A9320H			
0CH	Transfer Count = 00010280H			
10H	0 = 2AH	1 = 40H	2 = 0	3 = 08H
14H	4 = F4H	5 = E9H	6 = 0	7 = 02H
18H	8 = 05H	9 = 0	10 = 0	11 = 0

Offset + x values are in opposite order for Intel processors

Figure 7-5: Intel Ordered Parameter Block

In order for this transititon to properly occur, more than just your interpretation of the graphics in the manual must change. The RF3570 must be told to also reinterpret the structures it uses.

RF3570 Swapping Options

Byte and Word Swapping methodology for the RF3570 adapter is configured with the Control field of the Address Buffer Port. This field is in the first word written to the Address Buffer Port. (For bit-specific information on the Address Buffer Port see Chapter 2, Hardware Essentials.)

Swapping is a function of the hardware on the RF3570 adapter and is enabled and performed without detriment to performance.

There are four bits in the Control field that allow you to separately control byte swapping and word swapping — 2 bits for **Command Structures** and 2 bits for **data** to be transferred. If you set *none* of the swapping bits in the Control field, the Command Structures and the data will be ordered for a standard Motorola processor.

Command Structure Swapping

As explained, the need to byte-swap and word-swap the fields of the RF3570 Command Structures is due to the differences inherent in various processor architectures. Since the RF3570 defaults to using Motorola ordering for its structures, if your system uses a Motorola processor, no byte and word swapping is necessary. However, if your system uses an Intel processor, the Command Structures of the RF3570 must be adjusted.

This *could* be done entirely in software — the Host processor could re-order the Command Structures. However this extra burden for the system is not necessary because, by simply enabling the Byte and Word Swap controls, the RF3570 *hardware* will automatically re-order the Command Structures. The re-ordering occurs at "no cost" since the hardware automatically performs this feature with no additional overhead.

Data Swapping Options

Another aspect of the Byte and Word swapping ability of the RF3570 is for conversion of the format with which **data** is written to media.

The demonstrated difference in memory ordering between Intel and Motorola for Command Structures can be equally valid for data transfers. Since most data transfers will naturally occur using the full bus width available, the issue again becomes the difference in the way words and double-words are stored in memory (and then transferred to media).

This is only important if you are sharing data between two different types of processors.

Sharing Tapes

For example, if you are writing tapes with an Intel-based machine and only Intel-based machines will be reading them, *no swapping is necessary*.

If, however, a Motorola-based machine was to read the tape, it may need to re-order the data. Again, this could be done in software by having the Motorola processor re-order the data structures. Again, with the RF3570 it is not necessary to burden the Host processor.

By simply enabling the Byte and Word Swap controls, the RF3570 adapter hardware will automatically re-order the data structures with no additional overhead.

Dual Initiators

Another case for data swapping would be when dual RF3570's are used in separate machines but share the same SCSI bus. If both systems use the same disk media, and each is based on a different processor, one of the RF3570's may need to have swapping enabled.

Implementation of Data swapping

To accomplish Byte and Word swapping conversion for a data transfer, the WSD *and/or* the BSD bits of the Control field of the Address Buffer Port must be set to 1.

(See Chapter 2 for more information about the Address Buffer Port.)

NOTE: *If you set the WID bit of the Control field to 1, you will enable 32-bit data transfers and will need to swap both words and bytes. If you set it to 0, 16 bit transfers will occur and you will only need to setup for byte swapping.*

The use of Byte and Word swapping for data transfers can easily be used to provide flexibility for your system. Follow these guidelines to make it work for you properly:

1.) If you enable byte-swapping, you must be sure that all system addresses and transfer counts that you use are aligned to a word boundary. If you do not, byte swapping will not be performed even if you set the proper bit.

For the RF3570 to know which bytes to properly swap, it must have a point of orientation to work from. Therefore you must arrange addresses and counts to preserve the word boundary.

2.) If you enable word-swapping, you must align all system addresses and transfer counts to a double-word boundary. If you do not, word swapping will not be performed even if enabled.

Again, the RF3570 requires proper address orientation and proper byte counts to be able to judiciously swap words.

3.) You may need to allocate more memory for a transfer or adjust the counts you use in order to accomodate the need to preserve word or double-word boundaries. This may require an additional routine in your driver that checks addresses returned from dynamic allocation.

Scatter/Gather Operations

The Scatter/Gather feature is an alternate means to utilize the memory space available in your system. It can be enabled, for any command that transfers data to or from a SCSI device, by setting a bit in the Pass-through Parameter Block of the command.

What it does

Briefly, Scatter/Gather operations allow more flexibility in allocating memory space for your transfers. With Scatter/Gather enabled a single command can:

- Use separate pieces of memory. (Pieces that are non-contiguous, non-adjacent, not next to each other.)
- Use pieces of memory that are sized smaller than the block size of the command.

How to use Scatter/Gather

Scatter/Gather Operations (SGO) must be enabled and set-up properly. Each step in the process is explained briefly below.

1. **Build a chain of Descriptor Blocks** that contain pointers to locations in memory that you wish to use. Each SGO Descriptor Block defines up to eight segments of memory by providing the address and length of each segment. To build the chain:

   a.) Build each Block in memory. Up to eight segments can be described in each block; provide the address and length of each segment.

   b.) Point to the first Descriptor Block from the VME MemoryAddress field of the Parameter Block issuing the command.

   c.) Place the address for each subsequent Descriptor Block in the first double-word of the current Descriptor Block.

   d.) Identify the last Descriptor Block by placing an end value (FFFFFFFFH) in its first double-word.

2. **Set the SGO bit in the Flags-1 byte** to a one in the SCSI Pass-through Parameter block containing the command. (See page 5-3 for more information.)

3. **Issue the command** using the normal means.

Format of Scatter/Gather Descriptor Block

The Descriptor Block is an important piece of the Scatter/Gather operation. Each must be set up in the format that follows:

Address Offset	Byte Memory Address			
	Offset + 0	Offset + 1	Offset + 2	Offset + 3
00H	Next SG Descriptor Block Address			
04H	Add Mod 1	Data Length - Segment 1		
08H	Data Address -Segment 1			
0CH	Add Mod 2	Data Length - Segment 2		
10H	Data Address - Segment 2			
14H	Add Mod 3	Data Length - Segment 3		
18H	Data Address - Segment 3			
1CH	Add Mod 4	Data Length - Segment 4		
20H	Data Address - Segment 4			
24H	Add Mod 5	Data Length - Segment 5		
28H	Data Address - Segment 5			
2CH	Add Mod 6	Data Length - Segment 6		
30H	Data Address - Segment 6			
34H	Add Mod 7	Data Length - Segment 7		
38H	Data Address - Segment 7			
3CH	Add Mod 8	Data Length - Segment 8		
40H	Data Address - Segment 8			

Figure 7-6: Scatter/Gather Descriptor Block

Next SG Descriptor Block Address

This field is used to point to other Scatter/Gather descriptor blocks in the series or to indicate that this block is the *last* in the series.

Valid Address = Next Block Address.
FFFFFFFFH = This is the last descriptor block (of memory segments) in the series.

Address Modifier -

For each of the eight segments of the Descriptor Block you will need to provide the Address Modifier value to be used for the transfer.

Data Length - Segment

With this field you indicate, for each of the eight segments, the number of bytes available at the indicated address in memory. This field is also used to indicate when there are no more segments used in this block.

> **Valid number** = Number of bytes available at memory location.
> **0** = There are no more segments used in this block.

Data Address - Segment

This field indicates the beginning address of the segment for each of the eight segments in the SGO Descriptor Block.

How Scatter/Gather is Performed

When you issue a Read or Write command to the RF3570 adapter with the SGO bit of the Flags byte set to one, the adapter expects a Scatter/Gather operation. The adapter follows this sequence:

1. At the address, specified in the Pass-through command Parameter Block the adapter reads in the first SGO Descriptor Block.

2. Beginning at the **Data Address - Segment 1** of the block, the adapter transfers data (Read or Write) until the **Data Length - Segment 1** count is reached.

3. The **Data Length** of the next segment indicated in the SGO Descriptor block is examined for a zero value. If there is a non-zero value, the adapter continues the transfer operation, at the next segment address, until the next segment Data Length is exhausted.

4. This process continues until a zero **Data Length** is reached, or the limit of eight segments is reached. The adapter then reads in the next SGO Descriptor block. The address of the next block is read from the **Next SGO Descriptor Block Address** field.

5 Steps 2, 3, and 4 are repeated for each SGO Descriptor Block.

6. The Scatter/Gather operation is terminated when the adapter identifies a value of FFFFFFFFH in the **Next SGO Descriptor Block Address** field.

Notes on Use

These are the things to remember as you are using the Scatter/Gather feature offered by the RF3570 adapter:

- The RF3570 always reads the entire Descriptor Block, regardless of the number of segments that are actually used. You do not have to use all eight segments available in a single SGO block — indicating a zero in the Data Length field of any segment causes subsequent segments to be ignored during processing of the block. It is to your benefit to use all the segments of a block rather than have more blocks because each new block read by the adapter requires a bus transfer. Therefore more VMEbus activity is incurred if the blocks are mostly empty.

- You should build all SGO blocks before issuing the command that will use them. This will prevent the adapter from being given invalid parameters.

- To maximize performance when using Scatter/Gather you should be sure that the Data Length and Data Address of all segments in a Descriptor Block cause alignment to the same byte/word/double-word boundary. The RF3570 hardware design incorporates proprietary VLSI that preloads the starting address and transfer count of the next segment in the Descriptor Block. This pipelining is only effective when all the segments in a single Descriptor Block are aligned to the same address boundary.

- The transfer count in the Parameter Block should contain a value that reflects the *total* number of bytes to be transferred in the *entire* SGO operation.

- All other fields of the Pass-through Command Parameter Block are used the same as in a non-SGO operation.

Example Scatter/Gather Operation

In this example, two Descriptor Blocks are used to define eleven areas of memory that together contain 400H bytes of information to be written to SCSI device zero.

The example Parameter Block for this Gather Write command contains this information:

The SGO bit of this field must be set to 1.

This is the address of the first SGO Descriptor Block.

The Transfer Count should be the total of all Descriptor Block transfers.

Address Offset	Byte Memory Address			
	Offset + 0	Offset + 1	Offset + 2	Offset + 3
00H	Command Identifier = 00001A84H			
04H	Flags-2 = 00H	Flags-1 = 01H	Addr. Mod =3DH	Target ID = 00H
08H	VME Memory Address = 00345678H			
0CH	Transfer Count = 00000400H			
10H	0 = 2AH	1 = 40H	2 = 0	3 = 08H
14H	4 = F4H	5 = E9H	6 = 0	7 = 00H
18H	8 = 02H	9 = 0	10 = 0	11 = 0

Figure 7-7: Example Scatter/Gather Parameter Block

Follow these steps to issue the Scatter/Gather command.

1.) Build a Chain of Descriptor Blocks.

For this example, two blocks will be built, the first is at the address specified in the Parameter Block — 345678H:

Address Modifiers, Segments 1-8

Next SG Descriptor Block Address

345678	00	23	33	34		
34567C	3D	00	00	10	Data Length	Segment 1
345680	00	60	00	00	Data Address	
345684	3D	00	01	00	Data Length	Segment 2
345688	00	66	66	66	Data Address	
34568C	3D	00	00	01	Data Length	Segment 3
345690	00	70	00	00	Data Address	
345694	3D	00	00	23	Data Length	Segment 4
345698	00	71	00	00	Data Address	
34569C	3D	00	00	02	Data Length	Segment 5
3456A0	00	72	00	00	Data Address	
3456A4	3D	00	00	12	Data Length	Segment 6
3456A8	00	73	00	00	Data Address	
3456AC	3D	00	00	24	Data Length	Segment 7
3456B0	00	73	33	33	Data Address	
3456B4	3D	00	00	01	Data Length	Segment 8
3456B8	00	74	00	00	Data Address	

Figure 7-8: First SG Descriptor Block

The next Descriptor Block is at the address specified in the **Next SG Descriptor Block Address** field of the first Decriptor block — 233334H:

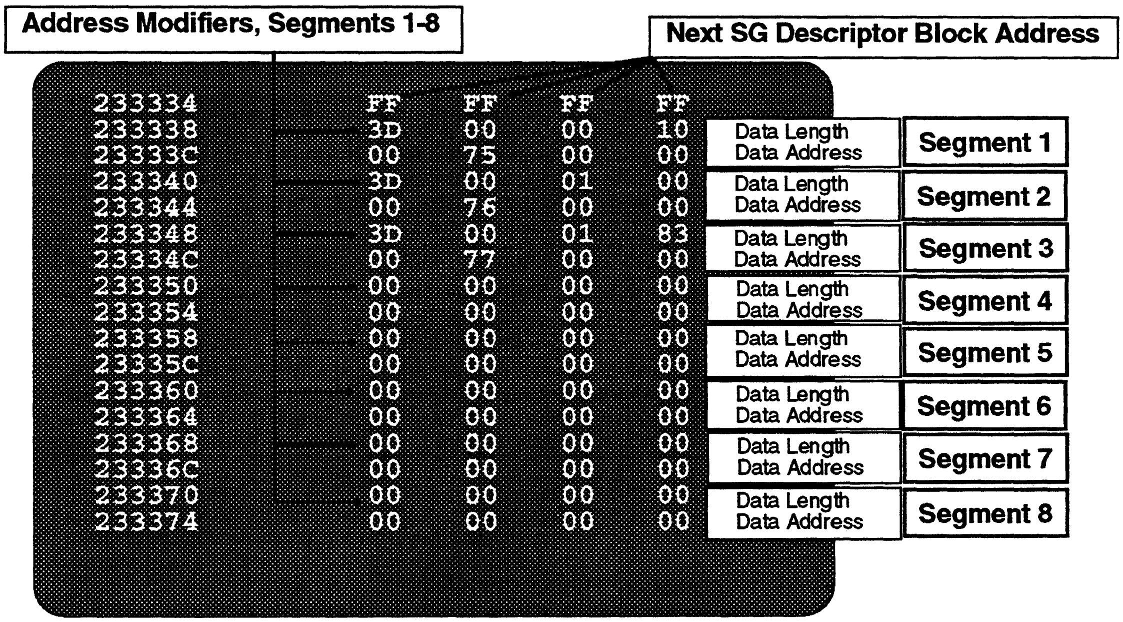


Figure 7-9: Last SG Descriptor Block

Note that this Descriptor Block has the value FFFFFFFFH in the first field to indicate that this is the last Descriptor Block. Additionally, only three of the eight segments are used in this Descriptor Block; the rest of the segments are zeroed.

2.) Be sure the SGO bit of the Parameter Block is set to one.

3.) The command is ready to be issued.

Odd Byte Handling

The highest throughput and least amount of overhead is acomplished during data transfers by: aligning all memory addresses to an even boundary of the system bus width, and using transfer counts divisible by the number of bytes in the system bus width. This allows the adapter to always transfer data using the full width of the system bus. Although desirable, this may not always be possible. When not possible, the RF3570 must decide how much of the available bus it *can* use for the transfer. The Odd Byte Handling feature of the RF3570 is an alternate algorithm for determining transfer width that attempts to mitigate effects on performance when the address and/or transfer count used in a command are not aligned.

The RF3570 can be set-up to use either the Standard or Odd Byte Handling algorithm for determining transfer width. Each method has trade-offs. Which one you choose should be decided based on the types of transfers typical to your system and availability of bus resources. The Odd Byte Handling feature must be enabled using the General Options Board-control command. (See Chapter 6 for more information.) The RF3570 defaults to the Standard algorithm.

After defining some terms, this section describes each of the algorithms used by the RF3570 to determine bus transfer width, then provides examples that illustrate the possible advantages inherent in each method.

What is an Odd Byte?

The RF3570 can perform 8, 16, or 32 bit transfers of data across the system bus. The choice of which bus width to use is based on the following three factors:

- available system bus width
- transfer count
- memory address of the transfer

The term "Odd Byte" is derived from the situation that occurs when an address and/or transfer count cause the data transfer to either begin at an address not aligned to the full system bus width, and/or finish at an address that is not aligned to such a boundary.

In such cases there are remainder or "odd" bytes that must be handled. These odd bytes can occur at the beginning of a transfer (because the address of a transfer is not bus-width aligned) and also at the end of a transfer (when a transfer count contains a non-aligned number of bytes).

Other Definitions

A key to understanding the information in this section is to know the definition of terms that are used throughout.

Bus-width Aligned Address

In a system with a 32-bit system bus, four bytes of data can be transferred at once across the system bus. An address that is bus-width aligned to a 32-bit system bus, then, would be an even multiple of 4. On a 16-bit system bus, two bytes can be transferred at once; bus-width aligned addresses would be an even multiple of 2.

Non-aligned Number of Bytes (or Non-aligned Transfer Count)

These terms are used to describe a transfer count that is not evenly divisible by the number of bytes in the system bus width. A non-aligned transfer count can cause the transfer to end at an address that is not bus-width aligned. For a 32-bit system bus, this term applies to values that are not a multiple of 4. On a 16-bit bus it applies to values that are not a multiple of 2.

Non-aligned Data

This term is used to describe the bytes that cause a transfer to be non-aligned. They are the remainder bytes at the end of a transfer of a non-aligned number of bytes, or the extra bytes of data that occur at the beginning of a transfer when an address is not bus-width aligned.

Algorithms for Transfer Width

Both systems used by the RF3570 adapter to determine bus width for data transfer involve the three factors already mentioned: system bus width, transfer count, and memory address. The system bus width is a predetermined constant, but the transfer count and address can change with each command.

Standard Algorithm

The Standard algorithm only sets up the hardware *once* for the *entire* data transfer operation. The transfer width is detemined by performing logical operations on the transfer count and address of the transfer. With a 32-bit system bus width, these are the possible results:

> If the data is aligned to a 32-bit bus, the RF3570 sets up the hardware for 32-bit wide transfer.

> If the non-aligned data is a word, the RF3570 sets up the hardware for 16-bit wide transfer.

> If the non-aligned data is an odd number of bytes, the RF3570 sets up the hardware for 8-bit wide transfer.

Advantages

- There is little adapter overhead per operation because there is only one hardware setup required for the entire transfer and only one interrupt to service.
- The adapter will achieve the greatest throughput with minimal overhead when the transfers and addresses *are* aligned to system bus width.

Disadvantage

- The adapter will require more frequent use of the system bus resource because it will be using less than the maximum bus width during transfers that have non-aligned data.

Odd Byte Handling Algorithm

The Odd Byte Handling algorithm allows the RF3570 to perform *multiple* setups of the hardware for each transfer. The result of the logical operation performed on the transfer count and memory address determines what and how many setups are used.

If the transfer turns out to be aligned with the system bus width, the adapter performs a single setup, as with the Standard algorithm. If the transfer is not so aligned, the adapter can perform up to three different setups for the transfer, depending on the situation. These are the possible setups that may be required on a 32-bit system bus:

If the address is not bus-width aligned, the adapter will setup to transfer (in 8-bit mode *or* 16-bit mode) the 1 to 3 bytes of initial non-aligned data.

The next setup performed by the adapter will take advantage of the full system bus width. This transfer will move all data that is aligned to the full bus width.

If the transfer count contained a non-aligned number of bytes, the next possible setup would be 8-bit or 16-bit to move the 1 to 3 remainder bytes of non-aligned data.

Advantage

- For transfers that have non-aligned data, the adapter will utilize the minimum amount of available bus resource; the adapter will transfer most of the data using the full system bus width.

Disadvantage

- A greater amount of overhead is incurred because up to three different setups may be required, and up to three different interrupts may need servicing.

Examples

The five examples that follow further illustrate the differences, advantages, and disadvantages of the Odd Byte Handling and Standard algorithms.

Example 1: A Bus-width Aligned Transfer

Bus Width = 32 bit
Address = 0000
Transfer Count = 12

This is the perfect scenario; the address is aligned to the system bus width, and the transfer count is evenly divisible by the number of bytes in same:

$12 \div 4 = 3$

How would each algorithm handle a transfer that is perfectly aligned?

Standard Algorithm

The RF3570 performs a single hardware setup for three double-word transfers.

Odd-Byte Handling Algorithm

The RF3570 performs a single hardware setup for three double-word transfers.

Conclusion

For transfers that are entirely aligned, there is no difference in how each algorithm handles the transfer.

Example 2: Transfer using a Non-aligned Transfer Count

Bus Width = 32 bit
Address = 0000
Transfer Count = 13

This example illustrates a non-aligned transfer. The address of this transfer is bus-width aligned, but the transfer count contains a non-aligned number of bytes:

$13 \div 4 = 3$, with a remainder of 1.

How will each algorithm handle the non-aligned data resulting from a non-aligned number of bytes in the transfer count?

Standard Algorithm

The RF3570 sets up the hardware to perform 13 transfers of one byte each.

Odd Byte Handling Algorithm

The RF3570 performs the following set-ups:

1. Hardware is setup for 3 double-word transfers.

2. Hardware is setup for 1 byte transfer.

Conclusion

The Standard Algorithm uses less overhead because it sets the hardware once for the entire operation, but it uses more of the bus resource by doing only byte-wide transfers. The Odd Byte Handling Algorithm spends more overhead (in hardware set-up) but transfers data more efficiently.

Example 3: Transfer using a Byte-aligned Address

Bus Width = 32 bit
Address = 0001
Transfer Count = 12

This is a non-aligned transfer. The system bus is 32-bit, but the address in this example is aligned to a *byte* boundary. The transfer count however, is evenly divisible by the bytes in the system bus width:

$12 \div 4 = 3$

How will each algorithm handle the non-aligned data resulting from an address that is not aligned to the bus-width?

Standard Algorithm

The RF3570 performs a single hardware setup for 12 transfers of one byte each.

Odd Byte Handling Algorithm

The RF3570 performs the following setups:

1. Hardware is set for 3 byte-wide transfers. When complete, the address is equal to 4 and the transfer count remaining is equal to 9.

2. Hardware is set for 2 doubleword-wide transfers. When complete, the address is equal to 12 and the transfer count remaining is equal to 1.

3. Hardware is set for 1 byte-wide transfer.

Conclusion

Again, the Standard Algorithm uses little overhead but more bus resource. The Odd Byte Handling Algorithm used 3 separate setups in this example and so involves more overhead.

Example 4: Transfer using a Word-aligned Address

Bus Width = 32 bit
Address = 0002
Transfer Count = 12

This is another example of a non-aligned transfer. In this example, the address is aligned on a *word* boundary on a system with a 32 bit bus. However the transfer count is divisible by the number of bytes in the system bus width.

$12 \div 4 = 3$

How will each algorithm handle the non-aligned data that results from using a word aligned address instead of a double-word aligned address?

Standard Algorithm

The RF3570 performs a single hardware setup for 6 word-wide transfers.

Odd Byte Handling Algorithm

The RF3570 performs the following setups:

1. Hardware is set for 1 word-wide transfer. When complete, the address is equal to 4 and the transfer count remaining is equal to 10.

2. Hardware is set for 2 doubleword-wide transfers. When complete, the address is equal to 12 and the transfer count remaining is equal to 2.

3. Hardware is set for 1 word-wide transfer.

Conclusion

The Standard Algorithm is somewhat more effective because it is able to use word-wide transfers. The Odd Byte Handling Algorithm again uses 3 separate setups which incurs overhead, but it was able to transfer the majority of data using the full bus width.

Example 5: Non-aligned Transfer Count *and* Address

Bus Width = 32 bit
Address = 0003
Transfer Count = 13

This example of a non-aligned transfer illustrates what can happen when both the address and the transfer count are not aligned. The address is aligned to an odd byte. The transfer count is not divisible by the number of bytes in the system bus:

$13 \div 4 = 3$, with a remainder of 1

How will each algorithm handle any non-aligned data that results from this combination of factors?

Standard Algorithm

The RF3570 performs a single setup for transfer of 13 bytes.

Odd Byte Handling Algorithm

The RF3570 performs the following setups:

1. Hardware is set for 1 byte-wide transfer. When complete, the address is equal to 4 and the transfer count remaining is equal to 12.

2. Hardware is set for 3 doubleword-wide transfers.

Conclusion

Although the Standard Algorithm still takes less overhead and more bus resource, in this example the Odd Byte Handling Algorithm takes less overhead than previous examples by needing only two setups, and it still uses the bus efficiently.

Deciding on an Algorithm

There are several major factors that should influence the decision you make about which transfer width algorithm to use:

- whether the majority of transfers are bus-width aligned
- the size of a typical transfer
- bus availability

The first consideration is always bus-width alignment. If your transfers are mostly aligned, the Standard Algorithm is the best to use. If bus-width aligned transfers are not clearly in a majority, *then* you must look at the typical size of your transfers balanced with available bus resource. If your transfers tend to be small, the Odd Byte Handling Algorithm would incur greater overhead and degrade performance. If, on the other hand, you have medium to large transfers, the Odd Byte Handling Algorithm would provide better performance. Anytime that bus resource is at a premium, the Odd Byte Handling Algorithm is useful. A tradeoff exists between the two algorithms: better performance using the full bus width, versus better performance with reduced overhead.

Tagged Queuing

The SCSI-2 specification provides a protocol for queuing commands at the device level. The device is responsible for managing the order of the commands, basing the order on message information from the initiator; the tag message used by the initiator instructs the device where to place the command in its queue. The initiator also uniquely numbers each command to identify it.

The RF3570 fully supports the tagged queuing protocol. This section describes the RF3570 implementation of tagged queuing and how to use it.

How to Use Tagged Queuing

The RF3570 implementation of this SCSI-2 option is very simple to use. These are the steps:

1. Enable Tagged Queuing

In order to use the SCSI-2 option of Tagged Queuing, you must be sure that it is enabled on both the adapter and SCSI device.

A. Enabling on the RF3570

For the RF3570, tagged queuing is enabled for use with a device by issuing a Unit Options or Extended Unit Options command with the TAG bit of the Unit Flags field set *before* you issue any commands you want to tag. The RF3570 will only send tagged messages to a device if this flag has been set. This could be done during initialization.

B. Enabling on a SCSI Device

You may also need to enable tagged queuing on your SCSI device with a Page 0AH Mode Select. See your vendor-supplied documentation for more information. The RF3570 will not send an error to the Host if your device rejects the queue tag message.

NOTE: *If your SCSI-2 device is operating with tagged queuing, the Queue Full Count and Maximum Queued fields of the Extended Board Statistics command (16H) will contain values. See Chapter 6.*

2. Choose a tag message.

There are three tag messages specified by the SCSI-2 protocol. You need only set the appropriate TAG OPTS bits in the Flags-2 field of the Pass-through Parameter Block for the command you are issuing to the SCSI device.

BITS							
7	6	5	4	3	2	1	0
0	0	0	0	JSM	IAD	TAG OPTS	

Figure 7-10: Meaning of Flags-2 Field

These are the three messages defined by the SCSI-2 specification, their meaning, and the TAG OPTS bit assignments for the Flags-2 field of the RF3570 command:

TAG OPT Value	Meaning:
00	**SIMPLE QUEUE TAG MESSAGE:** The command may be processed by the SCSI-2 device out of sequence for efficiency. Example: device may sort commands by logical block address.
01	**ORDERED QUEUE TAG MESSAGE:** The command must be executed by the SCSI-2 device in the order sent by the adapter (in order with respect to other commands sent with the ORDERED QUEUE TAG Message).
10	**HEAD OF QUEUE TAG MESSAGE:** The command is executed by the SCSI-2 device before all other commands previously sent, including other Head of Queue Tag commands but excluding the command currently in process.
11	Reserved

Table 7-1: Meaning of TAG OPTS Bits

As you determine the types of tag messages you wish to use, you should be aware that there are rules of execution that your SCSI-2 device must follow for each tag message or combination of tag message commands. For example, determination of priority in the case of mixed tag messages, which executes first — an ORDERED QUEUE TAG or a SIMPLE QUEUE TAG? This situation and others are detailed in your SCSI-2 specification.

3. The RF3570 sends the tag message.

The RF3570 decodes these two bits, assigns a tag number and then manages the protocol for sending the proper message for the command.

How It Works

This is how the RF3570 processes commands that use tag messages:

1. The tag message is sent after the identify message and before the SCSI command.
2. The adapter sends the commands in the exact order received from the host.
3. The queue tag number that is sent with the tag message is assigned by the adapter in ascending order beginning with 1, maximum of 255. When 255 is reached, the RF3570 starts over again with 1.

Notes on Usage

The following provides additional detail about how tagged queuing works for the RF3570.

Queue Tag Numbers

- The RF3570 will never have duplicate queue tag numbers assigned at the same time to any single device.
- The maximum number out to a device at one time will be 255.
- If all 255 tag numbers have been assigned to commands, the RF3570 simply waits for previous commands to complete before sending any more tagged commands to that device.

Queue Full Status

- If a device reports Queue Full status before it has received all possible 255 commands, the RF3570 sets an internal threshold flag, (MAXIMUM_COMMANDS) equal to the number of commands out at the time of the Queue Full status.
- The command that generated the Queue Full status will once more be sent out to the device when the number of commands out to the device drops below the threshold set in MAXIMUM_COMMANDS. (One or more disconnected commands complete). This cuts down on unnecessary SCSI bus activity.
- Once below the MAXIMUM_COMMANDS threshold, additional commands will continue to be sent to the SCSI device until another Queue Full status is reported, at which time the MAXIMUM_COMMANDS threshold is set again.

Message Reject

- On the intitial command to a SCSI device for which tagged queuing has been enabled, the adapter verifies that the device accepted the queue tag message.
- If the device rejects the queue tag message, the RF3570 does not send further queue tag messages in order to avoid unnecessary SCSI bus activity.
- In the case of a message reject of a tag message, no error is reported to the host.

Tagged Queueing Statistics

The Extended Board Statistics command (16H) returns two fields, per target, that may be useful in evaluating the performance of tagged queuing with a SCSI-2 device. The fields are: Queue Full Count and Maximum Queued. See Chapter 6 for details.

Appendix

A

Error Codes

Appendix A — Error Codes

Introduction

When the RF3570 adapter encounters a command or operation error, it will return a value in the Error byte of the Status Block. This value can be used to determine the nature of the problem the adapter encountered. A list of these error codes is included in this appendix along with the error codes returned by Self-test.

Summary

This appendix provides information on the following:

- Error Codes returned in the Error field of the Status Block.
- Error Codes returned in the Status Port as a result of Self-test.

This appendix is helpful for someone writing a driver for use with the RF3570.

Status Block Error Codes

The Error Codes on the following pages are the codes that you will see returned in the Error field of the Status Block.

The Status Block has this basic format:

Address Offset	Byte Memory Address			
	Offset + 0	Offset + 1	Offset + 2	Offset + 3
00H	Command Identifier			
04H	Reserved	SCSI Status	Error	Flags
08H	SCSI Sense Bytes			
0CH				

Figure A-1: Base Status Block

The Error Codes are meant to give you an indication of the conditions surrounding an adapter-detected error. There are two types of errors that can occur: system operation errors and SCSI bus errors. Both types of codes are fully described in order to assist you in determining the reason for a failure.

hexadecimal values that do not appear in the current list of Error Codes are reserved by Ciprico for future use.

Codes Reported in Status Block *and* Status Port

Codes in the following list that are asterisked (* **XXH**) are considered catastrophic errors that will also be reported through the **Status Port**.

Code	Name	Description
01H	**Invalid Command**	This code will be reported for the following conditions: **In a Board-control Parameter Block:** code used in the Command field is not a valid Board-control command. Occurs when Target/Adapter ID field is equal to FFH (indicating a Board-control command). **In a Pass-through Parameter Block:** An invalid tag type was specified in the Flags-2 field. **In a Scatter/Gather operation:** The initial descriptor list of a Scatter/Gather command had no valid entries. **In a Message Pass-through Parameter Block:** A non-supported message was specified for a SCSI operation.
02H	**Bad Unit Number**	This code will be reported for the following conditions: **Target/Adapter ID field:** The value specified in the Target/Adapter ID field of a Parameter Block was not valid (valid values are 0-7 and FFH). **In a Pass-through Parameter Block**: The value specified in the Target ID field is the same as the SCSI ID assigned to the RF3570. This is not a valid SCSI ID to issue commands to. **In a Pass-through Parameter Block:** the value used in the LUN field of the SCSI command contained in the Parameter Block is invalid. (Valid values are 0-7.) **In General Options Parameter Block:** the value specified in the SCSI Bus ID field (used to designate an RF3570 SCSI ID other than that set with jumpers) is invalid. (Valid values are 0-7.) **In Unit Options or Extended Unit Options Parameter Block:** Value specified in the Unit SCSI ID field was invalid. (Valid values are any number between 0 and 7, not being used by the RF3570.)
0FH	**Bad Command List Size**	This code will be reported when the **Start Command List** command terminates for the following reasons: • Number of Parameter Blocks field contains invalid value. (Valid values are 2-2034.) • Number of Status Blocks field contains invalid value. (Valid values are 2 -4096.)
* 11H	**State wrong for Start/Stop Command List command**	This code is returned under these conditions: **Start Command List** command issued when a command list is already active. **Stop Command List** command issued when no command list is presently active. **Command List Channel Attention** issued when no Command List is active.

* 14H	**General Firmware Timeout**	The adapter times each of its transfers; it will report this error if a VMEbus or SCSI bus transfer takes too long to complete. On the VMEbus, this error can occur for transfers of parameters as well as transfers of data.
* 15H	**VMEbus Error Occurred**	A VMEbus error was detected while the adapter was transferring either parameters or data.
1EH	**SCSI Select Timeout**	For SCSI **Pass-through** operations, this code is reported when no SCSI device responds to the selection sequence within the timeout period. The timeout period is specified in the Unit Options command or Extended Unit Options command.
1FH	**SCSI Disconnect Timeout**	For SCSI **Pass-through** commands, this code is reported when a disconnected SCSI device does not reselect the adapter within the timeout period. The timeout period is specified in the Unit Options command or Extended Unit Options command.
20H	**SCSI Parity Error**	For SCSI **Pass-through** operations, this code is reported for any SCSI information transfer (message, command, data or status) that results in a parity error.
21H	**Unexpected Disconnect**	If a SCSI device unexpectedly goes to the bus free state, this code will be reported.
23H	**Unit Returned Bad SCSI Status**	If the device returns a bad status (status other than 0) in response to a SCSI Pass-through command, this code will be reported. The SCSI Status field of the Status Block will contain the Status Byte returned by the device (Busy, Reservation Conflict, Check Condition, etc.). If the SCSI status is Check Condition, and the automatic request sense feature of the adapter is not inhibited, the Selected Sense Bytes fields of the Status Block will contain the Status Bytes returned by the device. These will further describe the Check Condition. If the automatic Request Sense feature is inhibited (with the IRS bit of the Flags-1 byte), the Selected Sense Bytes fields of the Status Block will contain zeroes.
24H	**Unexpected SCSI Phase Entered**	This code is reported if the SCSI device being accessed enters a phase that is unexpected for the command sequence issued.

25H	**Data Transfer Truncated**	If an automatic Request Sense command results in fewer bytes transferred than required to satisfy the Selected Sense Bytes fields of the Status Block, this code will be reported. The Selected Sense Bytes fields of the Status Block that have been truncated will contain value FFH to aid the host in determining which Selected Sense Bytes are valid.
27H	**SCSI Bus Reset Asserted or Detected**	This code can be reported for either of the following reasons: 1.) Certain conditions will arise that will force the adapter to assert RST on the SCSI bus in order to get to bus free. The SCSI command that was executing at the time of the RESET and any commands that were disconnected at the time of the RESET will report this error. A SCSI bus RESET is a last-resort measure that the adapter uses only if all prior error recovery attempts to get the SCSI bus to a free state have failed. 2.) The adapter detected a SCSI bus reset. The SCSI command currently executing and any commands that were disconnected at the time of the reset will report this code.
31H	**Message Queue Clear or Device Reset Occurred**	After executing a **Message Pass-through** operation of Bus Device Reset or Clear Queue, the adapter clears its internal queue of commands that were disconnected when the message was sent. This code will be reported in the Status Block of each disconnected command that was terminated due to the message-only operation.
8EH	**Bad Task**	This error code indicates the occurence of an Internal Firmware error detected by the adapter.
* 96H	**Internal Firmware Error**	This error code indicates that a Firmware error was detected during the execution of a command.
A1H	**Invalid Unit Options Sense Count Value**	A value greater than 32 was placed in the Sense Count field of a Unit Options or Extended Unit Options command. Valid values for this field are 0 to 32.

Status Port Error Codes

In addition to the Error Codes that are reported in the Status Block of the command, there is a group of codes that are used to report Self-test anomalies and also Catastrophic Error Conditions. These are reported in the Status Port.

Catastrophic Error Codes

These are Error Codes that are returned in the Status Port because conditions are such that it is unlikely that a Status Block could be read. The Status Port format at these times is as shown below:

BIT															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Code								0			ERR	0	0	RDY	ENT

Figure A-2: Status Port: General

The Catastrophic Error Code field is highlighted above.

The codes that can be returned in this field of the Status Port are described below:

Code	Name	Description
10H	Bad PBIN Value	The Command List PBIN (Parameter Block IN) value exceeded the value set for PBNUM (Parameter Block Number).
11H	State wrong for Start/Stop Command List command	This code is returned under these conditions: **Start Command List** command issued when a command list is already active. **Stop Command List** command issued when no command list is presently active. **Command List Channel Attention** issued when no Command List is active.
14H	Software VMEbus Timeout	The adapter times each of its VMEbus transfers; it will report this error if a VMEbus transfer takes too long to complete. This can occur for transfers of parameters as well as transfers of data.
15H	VMEbus Error Occurred	A VMEbus error was detected while the adapter was transferring either parameters or data.
96H	Internal Firmware Error	This error code indicates that a Firmware error was detected during the execution of a command.

Table A-1: Status Port Catastrophic Error Codes

Self-test Error Codes

Self-test occurs each time the RF3570 is powered on or reset. If one of the Self-tests fail, an Error Code is reported in the Status Port. The format of the Status Port is as follows:

BIT															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0								Self-test Code (STC)						RDY	STC

Figure A-3: Status Port: Reset

The Self-test Code (STC) fields are highlighted in the illustration above.

If one of the self-tests does not properly complete, its test number will be left in the Self-test Code fields. The table on the next page lists all the Self-test codes and describes the test performed.

Note that bit 1 of the Status Port is not set until the board is in a Ready condition. Therefore, during Self-test it will remain zero.

Diagnostic/Self-test Command

The Diagnostic/Self-test Board-control command allows you to selectively perform the board Self-tests. In the event that one of the Self-tests does not complete properly, the Error Codes shown on the next page will be used to report the area of difficulty.

It is important to note, however, that the codes are reported in the *Error field* of the *Status Block* returned from the Diagnostics/Self-test command. In that case, these codes are returned in the Status Block Error field *in addition* to the normal Status Block Error Codes.

Code	Description
00H	The onboard processor is not working; it cannot access firmware EPROM; cannot access the Status Port - problem with Short Burst FIFO or VMEbus data lines.
04H	Static RAM error. All words of static RAM are written with FFFFH, then zero. This test is not exhaustive.
0CH	Firmware checksum error. The calculated checksum for odd and even EPROMs does not match the value stored in EPROM.
14H	Short Burst FIFO cannot be accessed. SBF data could not be loaded or read back.
1CH	Channel Attention Port cannot be accessed, or interrupt won't clear.
90H	PSI Address/Counter registers not masked to WORD boundaries.
94H	PSI Address/Counter registers not masked to DWORD boundaries.
98H	Value Miscompare errors of PSI R/W registers in Manual Load mode.
9CH	Value Miscompare errors of PSI R/W registers in Auto Load mode.
A8H	SCSI termination power fuse is blown, but termination power is still present at the SCSI connector.
ACH	SCSI termination power fuse is blown and termination power is *not* available at the connector.
B0H	Value Miscompare error of Emulex Fast SCSI chip R/W registers.
B4H	Emulex Fast SCSI chip did not generate a reset interrupt.
B8H	Emulex Fast SCSI chip registers not zeroed after reset.
BCH	SCSI bus hung, waiting for reset from Emulex Fast SCSI chip. Could be cabling.
C0H	Short Burst FIFO 32 Bytes transfer miscompare error.
C4H	Short Burst FIFO Local Ready of System Not Ready error.
C8H	Short Burst FIFO byte transfer miscompare error.
CCH	Short Burst FIFO Local Not Ready or System Ready error.
E5H	The onboard processor failed the accumulator test.
E8H	The onboard processor failed the store instruction test.
E9H	The onboard processor failed the shift instruction test.
ECH	The onboard processor failed the compare instruction test.
EDH	The onboard processor failed the jump instruction test.

Table A-2: Self-test Status Port Error Codes

Appendix

B

Cables and Connectors

Appendix B — Cables and Connectors

Introduction

The SCSI specification allows devices to operate with either a single-ended or a differential SCSI interface. The difference between interfaces is the manner in which SCSI signals are driven on the cable.

The differential interface uses two lines for each signal (+SIGNAL and -SIGNAL). A signal is true when +SIGNAL is more positive than -SIGNAL. This interface provides better noise immunity than the single-ended interface and so allows a longer cable length to be used.

The single-ended interface uses one line for each SCSI signal.

All devices on a single SCSI bus must operate with the same interface.

The RF3570 is available with either a single-ended or differential interface, with either low density ribbon cable connectors, or high density shielded connectors.

Summary

This appendix provides the following information:

- Pin-outs of the differential connectors used for the RF3570 adapter.
- Pin-outs of the single-ended connectors used for the RF3570 adapter.
- Cable Lengths allowed for each interface.

This appendix is most useful for those integrating devices on a SCSI bus.

Differential Connectors

There are two models of RF3570 that support the differential SCSI interface. One is available with a ribbon cable connector and the other uses a high-density shielded connector.

Maximum Cable Length

Maximum cable length for differential operation is 25 meters (or 82 feet). This length includes internal cabling and cable stubs.

Ribbon Cable Connector

The **RF3574** model supports the differential SCSI interface with a ribbon cable connector. Its connector pin-outs are as follows:

Signal Name	Pin Number		Signal Name
Shield Ground	1	2	Ground
+DB (0)	3	4	-DB (0)
+DB (1)	5	6	-DB (1)
+DB (2)	7	8	-DB (2)
+DB (3)	9	10	-DB (3)
+DB (4)	11	12	-DB (4)
+DB (5)	13	14	-DB (5)
+DB (6)	15	16	-DB (6)
+DB (7)	17	18	-DB (7)
+DB (P)	19	20	-DB (P)
Diffsens	21	22	Ground
Ground	23	24	Ground
Termpwr	25	26	Termpwr
Ground	27	28	Ground
+ATN	29	30	-ATN
Ground	31	32	Ground
+BSY	33	34	-BSY
+ACK	35	36	-ACK
+RST	37	38	-RST
+MSG	39	40	-MSG
+SEL	41	42	-SEL
+C/D	43	44	-C/D
+REQ	45	46	-REQ
+I/O	47	48	-I/O
Ground	49	50	Ground

Table B-1: Cable Connector Pin-outs for the RF3574

High-Density Connector

The **RF3576** supports the differential SCSI interface with a high-density shielded connector. Its pin-outs are as follows:

Signal Name	Pin Number		Signal Name
Shield Ground	1	26	Ground
+DB (0)	2	27	-DB (0)
+DB (1)	3	28	-DB (1)
+DB (2)	4	29	-DB (2)
+DB (3)	5	30	-DB (3)
+DB (4)	6	31	-DB (4)
+DB (5)	7	32	-DB (5)
+DB (6)	8	33	-DB (6)
+DB (7)	9	34	-DB (7)
+DB (P)	10	35	-DB (P)
Diffsens	11	36	Ground
Ground	12	37	Ground
Termpwr	13	38	Termpwr
Ground	14	39	Ground
+ATN	15	40	-ATN
Ground	16	41	Ground
+BSY	17	42	-BSY
+ACK	18	43	-ACK
+RST	19	44	-RST
+MSG	20	45	-MSG
+SEL	21	46	-SEL
+C/D	22	47	-C/D
+REQ	23	48	-REQ
+I/O	24	49	-I/O
Ground	25	50	Ground

Table B-2: Cable Connector Pin-outs for the RF3576

Single-ended Connectors

There are two models of RF3570 that support the single-ended SCSI interface. One is available with a ribbon cable connector and the other uses a high-density shielded connector.

Maximum Cable Length

Maximum cable length for single-ended operation is 6 meters (or 20 feet). This length includes internal cabling and cable stubs.

Ribbon Cable Connector

The **RF3573** model supports the single-ended SCSI interface with a ribbon cable connector. Its connector pin-outs are as follows:

Signal Name	Pin Number		Signal Name
Ground	1	2	-DB (0)
Ground	3	4	-DB (1)
Ground	5	6	-DB (2)
Ground	7	8	-DB (3)
Ground	9	10	-DB (4)
Ground	11	12	-DB (5)
Ground	13	14	-DB (6)
Ground	15	16	-DB (7)
Ground	17	18	-DB (P)
Ground	19	20	Ground
Ground	21	22	Ground
Ground	23	24	Ground
Open	25	26	Termpwr
Ground	27	28	Ground
Ground	29	30	Ground
Ground	31	32	-ATN
Ground	33	34	Ground
Ground	35	36	-BSY
Ground	37	38	-ACK
Ground	39	40	-RST
Ground	41	42	-MSG
Ground	43	44	-SEL
Ground	45	46	-C/D
Ground	47	48	-REQ
Ground	49	50	-I/O

Table B-3: Cable Connector Pin-outs for the RF3573

High-Density Connector

The **RF3575** supports the single-ended SCSI interface with a high-density shielded connector. Its pin-outs are as follows:

Signal Name	Pin Number		Signal Name
Ground	1	26	-DB (0)
Ground	2	27	-DB (1)
Ground	3	28	-DB (2)
Ground	4	29	-DB (3)
Ground	5	30	-DB (4)
Ground	6	31	-DB (5)
Ground	7	32	-DB (6)
Ground	8	33	-DB (7)
Ground	9	34	-DB (P)
Ground	10	35	Ground
Ground	11	36	Ground
Ground	12	37	Ground
Open	13	38	Termpwr
Ground	14	39	Ground
Ground	15	40	Ground
Ground	16	41	-ATN
Ground	17	42	Ground
Ground	18	43	-BSY
Ground	19	44	-ACK
Ground	20	45	-RST
Ground	21	46	-MSG
Ground	22	47	-SEL
Ground	23	48	-C/D
Ground	24	49	-REQ
Ground	25	50	-I/O

Table B-4: Cable Connector Pin-outs for the RF3575

Appendix C — Specifications

Introduction

The RF3570 is designed to meet certain criteria: physical, electrical, environmental. The specifications are listed on the following page.

Summary

This appendix lists the pertinent specifications for the RF3570 SCSI-Host bus adapter.

This appendix is most helpful for someone choosing hardware for a system.

Specifications

Physical	Single slot, double height VME Eurocard form factor board: Dimensions: 233 mm by 160 mm
Electrical	Voltage: 4.75 Vdc to 5.25 Vdc Current: 4.0 Amps typical (at +5 Vdc)
Capacity	Up to seven SCSI devices
Transfer Rate	SCSI data rate to 7 Mbytes per second in Asynchronous mode. SCSI data rate to 10 Mbytes per second in Synchronous mode.
Environmental	
Operating	Temperature: 0° to +55° C Air Flow: 200 linear feet per minute Humidity: 10% to 80% (non-condensing) Elevation: 0 feet to 10,000 feet
Non-Operating	Temperature: - 40° to + 85° C Humidity: 10% to 95% (non-condensing) Elevation: 40,000 feet maximum
Bus Interface	VMEbus Standard (Revision C.1)
Device Interface	Draft Proposed American National Standard: Small Computer Systems Interface-2, (ANSI X3.131 - 199X

Table C-1: Specifications for the RF3570

Appendix

D

Defaults

Appendix D — Defaults

Introduction

Each Rimfire 3570 is shipped with the same factory settings. If you change nothing when you receive your board, it will power-up with the defaults that are listed in this appendix.

Summary

This appendix provides information on the following:

- Hardware defaults as shipped from the factory.
- Firmware defined defaults for these Board-control commands: Identify, General Options, Unit Options and Extended Unit Options, and Board Information.

This appendix is helpful when you wish to know the values the RF3570 will assume if you do not change an option.

Hardware Defaults

This section describes the defaults you will find for both the jumpers on the board, and the Hardware Ports you will use to set-up the board and operate it.

Board Jumpers

These are the factory settings for the jumpers on the board. For more information see Chapter 3.

Jumper	Default Setting	Meaning
VMEbus Address Modifier	No jumper at AM2	2D, Short Supervisory Access.
VMEbus Address	Jumper at A12	EE00H
SCSI Configuration Jumper Block:		
Parity Checking	OUT	Enabled
SCSI Bus Reset at Power-up	OUT	Enabled
RF3570 SCSI ID	OUT	ID = 0
Bus Request/Grant Level	6 jumpers installed: pins 4-5 under 0 pins 4-5 under 1 pins 4-5 under 2 pins 1-2, 3-4, 5-6 under 3	Level 3
Sysfail	IN	Sysfail signal asserted on bus during Reset.
Exact Burst	OUT	Initiate VMEbus DMA operation as soon as data ready to transfer.
SCSI Bus Terminators	IN	RF3570 provides SCSI Bus Termination.

Table D-1: Hardware Defaults - Jumpers

Defaults

Hardware Ports

These are the values that will be written if you do not specify other values. See Chapter 2 for more information.

Port	Default	Meaning
Address Buffer Port:		
Byte Swapping Control	0	The adapter will operate in 32-bit mode with Motorola ordering of bytes and words during data transfers; the parameter and status block structures will not be swapped.
Word Swapping Control	0	
Width	1	
Byte Swapping Data	0	
Word Swapping Data	0	
Status Port:		
Entered Bit	0	Initial value.
Ready Bit	1	After Reset has completed.
Code	02H	Whenever ERR bit is zero, this should be 02H.
Channel Attention Port	There is no default: Write a 1 to initiate a Command List, write a 0 to issue a Single Command.	
Reset Port	There is no default. Write any value to this port to get a Reset.	

Table D-2: Defaults for Hardware Ports

Command Defaults

If you do nothing to change the board operation by using the available Board-control commands, the adapter uses defaults. These are listed below. For more information see Chapter 6.

Command Parameters	Default Values	Meaning
***Identify* - returned in the status block:**		
Engineering Revision	XX	Depends on revison.
Firmware Revision	XX	
Day, Month, Year	XXXXXX	
Option Flags = 14H		
Tag	bit 2 =1	Tagged commands supported.
Reset	bit 4 =1	SCSI bus will reset at adapter reset - set by board jumper default setting.
SCSI ID	bits 5, 6, 7 = 0	Set by board jumper default settings.
FW # of Even Prom	XXXXXXXX	Depends on revision.
***General Options* - these are used if you set nothing yourself**		
Bus Throttle	20H	Transfer Count throttle of 32.
Select Flags = 03H		
Disconnect/Reconnect	bit 0 = 1	Allow Disconnect/Reconnect.
SCSI Bus Parity	bit 1 = 1	Check for SCSI parity errors.
Block Mode Transfers	bit 2 = 0	Data transfers will not be block mode.
Odd Byte Handling	bit 3 = 0	Use Standard algorithm.
***Unit Options and Extended Unit Options* - used if you set nothing:**		
Disconnect Timeout	0H	No timeout occurs.
Retry Limit	0H	No retries enabled.
Retry Control = 0H		
Issue Interrupt	bit 0 = 0	No interrupt on retry.
Issue Status Block	bit 1 = 0	No Status Block on retry.
Retry Parity Errors	bit 2 = 0	No retry for parity errors.
Retry Command Errors	bit 3 = 0	No retry for device errors.
Retry Bus Errors	bit 4 = 0	No retry for bus errors.

Table D-3: Command Defaults

Command Parameters	Default Values	Meaning
Unit Options and Extended Unit Options (cont):		
Select Timeout	FAH	250 milliseconds.
Unit Flags = 0H		
Untagged Queue	bit 0 = 0	Send one command at a time.
Synch Negotiation	bit 1 = 0	Will not initiate synchronous negotiation
Inhibit ATN Signal	bit 2 = 0	ATN will be asserted.
Tagged Commands	bit 6 = 0	Tagged commands not negotiated.
Sense Count	08H	
Selected Sense Bytes	all = 0	
Board Information **- these are returned in a Data Structure**		
Select Flags = 03H		
Disconnect/Reconnect	bit 0 = 1	Allow Disconnect/Reconnect.
SCSI Bus Parity	bit 1 = 1	Check for SCSI parity errors.
Block Mode Transfers	bit 2 = 0	Data transfer will not be block mode.
Odd Byte Handling	bit 3 = 0	Use Standard algorithm.
Bus throttle	20H	Transfer count throttle of 32.
Engineering Revision	XX	Depends on revision.
Firmware Revision	XX	
Day, Month, Year	XXXXXX	
Option Flags = 14H		
TAG	bit 2 = 1	Tagged commands supported.
Reset	Bit 4 = 1	SCSI bus will reset at adapter reset - set by board jumper default setting.
SCSI ID	bits 5, 6, 7 = 0	Set by board jumper default settings.
FW # of Even Prom	XXXXXXXX	Depends on revision.
Termination = 0H		
Bad SCSI Term Power	bit 1 = 0	TERM PWR is good.
Blown SCSI Fuse	bit 2 = 0	Fuse is good.

Table D-3: Command Defaults (continued)

Command Parameters	Default Values	Meaning
Board Information (cont):		
Jumper Configuration	0H	All Jumpers removed.
Target - Disconnect Timeout	0H	No timeout occurs.
Target - Retry Limit	0H	No retries occur.
Target - Sense Count	08H	First eight bytes of Sense Data.
Target - Device Flags = 0		
Untagged Queueing	bit 0 = 0	Allow 1 Command at a time.
Synch Negotiation	bit 1 = 0	Synchronous negotiation not initiated.
Inhibit ATN Signal	bit 2 = 0	ATN will be asserted.
SCSI - 2	bit 6 = 0	Target is not SCSI-2 device.
Issue Interrupt	bit 8 = 0	No interrupt for retry.
Issue Status Block	bit 9 = 0	No Status Block per retry.
Retry Parity Errors	bit 10 = 0	No retry for Parity Errors.
Retry Command Errors	bit 11 = 0	No retry for command errors.
Retry Bus Errors	bit 12 = 0	No retry for bus errors.
Selected Sense	bit 14 = 0	First 8 bytes of sense data returned.
Target - Synch Period	† 1FH	Equates to 125 nanoseconds.
Target - Synch Offset	†† 0FH	Emulex chip limit.
Target - Selected Sense Bytes	all = 0	First eight bytes of Sense Data returned.

† This value is returned when Synchronous transfer rate is 8 MB/sec.
†† This value is returned if the full Emulex FIFO is in use.
Both values are zero if operation is in Asynchronous mode.

Table D-3: Command Defaults (continued)

INDEX

A

Address Buffer Port
  address fields, 2-11
  address modifier field, 2-9
  control field, 2-9
  use of, 2-8
Address Modifier
  choosing, 2-5
  use and location of jumper, 3-11

B

Block Diagram, 2-4
BMT (Block Mode Transfer) bit, 6-21, 6-39
Board Information (15H), 6-36
  data structure, 6-38
Board Statistics (06H), 6-14
  data structure, 6-16
Board-control Commands, 6-1
  a list of available commands, 6-5
  board information, 6-36
  board statistics, 6-14
  defaults, D-4
  diagnostic/self-test, 6-30
  error, 6-4
  extended board statistics, 6-48
  extended unit options, 6-52
  format of general status block, 6-3
  general options, 6-18
  identify, 6-10
  SCSI hard reset, 6-34
  start command list, 6-6
  stop command list, 6-8
  unit options, 6-22
BSC (Byte Swapping Control) bit
  in control field of address buffer port, 2-9
BSD (Byte Swap Data) bit
  in control field of address buffer port, 2-10
Bus Request/Grant Jumper, 3-8
Bus Throttle
  See Throttle
Byte Swapping
  BSC (Byte Swapping Control) bit, 2-9
  BSD (Byte Swap Data) bit, 2-10
  use of feature, 7-2

C

Cabling, 3-19
  lengths, B-1
Catastrophic Errors
  LEDs and, 3-17
CC (Command Complete) bit, 4-7, 5-17, 6-4
Channel Attention Port
  format of, 2-13
  use of for command list, 2-12
  use of for single command, 2-12
CLR bit
  in extended board statistics, 6-49
  in options field of board statistics, 6-15
Command Identifier
  use of in parameter block, 4-4
  use of in status block, 4-6
Command Interface
  description of, 2-3
Command List Structure
  creating, 4-22
  format of, 4-20
  number of parameter blocks, 4-21
  number of status blocks, 4-22
  parameter block in index, 4-21
  parameter block out index, 4-21
  parameter block space, 4-22
  status block in index, 4-21
  status block out index, 4-21
  status block space, 4-22
  using, 4-23
Command Operation, 4-1
Connectors
  pin-outs, B-1
CSB (Continued Status Block) bit, 4-7, 5-17

D

DAT (Data Transfer) bit, 5-4
  use of with DIR and DBV bits, 5-5
DBV (Data Bits Valid) bit, 5-4
  use of with DIR and DAT bits, 5-5
Defaults, D-1
  board jumpers, D-2
  command, D-4
  hardware ports, D-3
Device Flags Field
  in board information command, 6-44
Diagnostic/Self-test (09H), 6-30
DIR (Direction of Data Transfer) bit, 5-3
  use of with DBV and DAT bits, 5-5
DIS (Disconnect/Reconnect) bit, 6-20, 6-39
Disconnect/Reconnect
  inhibiting with IAD bit, 5-13
  timeout, 6-23, 6-43, 6-54
  use of DIS bit, 6-20
DTT (Data Transfer Truncated) bit, 4-7, 5-17

E

ERR (Error Status) bit, 4-7, 5-17, 6-4
Error Codes, A-1
  01H, 5-15
  10H, 2-18, 3-17
  11H, 2-18, 3-17, 6-7, 6-9
  14H, 2-18, 3-17
  15H, 2-18, 3-17
  1FH, 6-23
  24H, 5-19
  96H, 2-18, 3-17
  reported in status block, A-2
  reported in status port - catastrophic, A-6
  reported in status port - selftest, A-7
Error Field
  in pass-through status block, 5-18
  in status block, 4-8
Exact Burst Jumper, 3-9
Examples
  'c' routines for command list, 4-26
  memory differences, 7-2
  odd byte handling, 7-20
  scatter/gather operation, 7-14
  SCSI inquiry (single command), 4-17
  start command list (single command), 4-14
Extended Board Statistics (16H), 6-48
  data structure, 6-50
Extended Unit Options (18H), 6-52
  data structure, 6-54

F

Features, 1-2
Flags Field
  in base status block, 4-7
  in identify command status block, 6-11
  in pass-through status block, 5-17
Flags-1, 5-3
Fuse, 3-13

G

General Options (07H), 6-18

H

Hardware Essentials, 2-1
  address buffer port, 2-8
  block diagram, 2-3
  board addressing, 2-5
  channel attention port, 2-12
  hardware ports, 2-6
  port usage, 2-19
  reset port, 2-19
  status port, 2-14
Hardware Installation, 3-1
  board insertion, 3-14
  jumpers, 3-6
  LEDs, 3-16

overview, 3-2
SCSI devices, 3-18
termination, 3-12
unpacking the board, 3-4
Hardware Ports
addresses, 2-6
defaults, D-3
overview, 2-7
usage, 2-19

I

IAD (Inhibit Automatic Disconnect) bit, 5-13
IAT (Inhibit ATN Signal) bit, 6-26, 6-45, 6-58
ICC (Inhibit Command Complete Interrupt) bit, 5-4
Identify (05H), 6-10
INT (Issue Interrupt) bit, 6-24, 6-45, 6-55
Intel Ordering, 7-4
Interrupts
inhibiting with ICC bit, 5-4
single command structure, 4-11
use of INT bit for retries, 6-24, 6-55
IRS (Inhibit Request Sense) bit, 5-4
ISB (Issue Status Block) bit, 4-12, 6-24, 6-45, 6-56

J

JSM (Just Send Message) bit, 5-13
Jumpers
bus request/grant, 3-8
defaults, D-2
exact burst, 3-9
factory set, 3-6
SCSI configuration, 3-6
sysfail, 3-9
VMEbus address, 3-10

L

LEDs
at power-up, 3-16
indication of error, 3-17
meaning of, 3-16

M

Motorola Ordering, 7-4
Multiple Status Blocks, 4-8
computing the number of blocks, 4-9
CSB bit in flags field, 4-7
retry generated, 4-9
sense data generated, 4-9
with selectable sense data, 5-22
with sequential sense data, 5-21

O

OBH (Odd Byte Handling) bit, 6-21, 6-39
Odd Byte Handling, 7-16
enabling with OBH bit in general options, 6-21
examples, 7-20
Option Flags Field
in identify command status block, 6-12
Options Field
in board statistics command, 6-15
in extended board statistics, 6-49
Options Flags Field
in board information, 6-40
Overview of RF3570, 1-1
design, 1-2
features, 1-2

P

PAR (Check SCSI Bus Parity) bit, 6-21, 6-39
Parameter Block
  basic format of, 4-4
  command descriptor block, 4-5
  command identifier, 4-4
  command options, addresses, transfer count, 4-5
  explanation of, 4-4
Pass-through Commands, 5-1
Pass-through Message
  format, 5-15
  LUN, 5-15
  message code, 5-15
Pass-through Parameter Block
  address modifier, 5-3
  command format, 5-2
  command identifier, 5-2
  flags-1, 5-3
  flags-2, 5-12
  message format, 5-15
  SCSI command descriptor block, 5-14
  target ID, 5-2
  transfer count, 5-14
  VME memory address, 5-14
Pass-through Status Block
  command identifier, 5-16
  error field, 5-18
  flags, 5-17
  format, 5-16
  SCSI status, 5-18
  sense bytes, 5-20
Pipelined System Interface (PSI)
  description of, 2-2
PSI
  See Pipelined System Interface

R

RBE (Retry Bus Errors-SCSI) bit, 6-24, 6-46, 6-56
RCE (Retry Command Errors) bit, 6-24, 6-46, 6-56
Repair
  shipping damage, 3-4
Request Sense
  default sense bytes, 5-20
  inhibiting with IRS bit, 5-4
  selectable sense bytes, 5-22
  sequential sense bytes, 5-21
Reset Port
  description of, 2-19
Retries
  determining retry limit set, 6-44
  retry control field, 6-24, 6-55
  retry limit field, 6-23, 6-55
Return Procedures, iii
  shipment instructions, iv
RMA Numbers, v
RPE (Retry Parity Errors) bit, 6-24, 6-45, 6-56
RST (Reset Jumper) bit, 6-41
  in identify command status block, 6-13
RTY (Retry) bit, 4-7, 5-17

S

SBF
  See Short Burst FIFO
Scatter/Gather
  example, 7-14
  format of descriptor block, 7-11
  notes on use, 7-13
  SGO (scatter/gather operation) bit, 5-3
  use of feature, 7-10
SCSI Configuration Jumpers, 3-6
  default settings, 3-7
  reading with board information command, 6-43
SCSI Hard Reset (10H), 6-34
SCSI ID
  choosing for adapter, 6-19, 6-54

C

choosing for peripherals, 3-18
reported for adapter, 6-13, 6-41
SCSI Peripheral Options, 3-18
SCSI Status
during unexpected SCSI phase, 5-19
SCSI Status Byte Code, 5-18
SCSI Timeout, 6-25, 6-56
Select Flags Field
in board information, 6-38
in general options, 6-20
Select Timeout Field
in extended unit options, 6-56
in unit options, 6-25
Self-test Codes, 2-15
Sense Count
determining as set per target, 6-44
setting per target, 6-27, 6-58
Sense Data
specifying in extended unit options, 6-58
specifying in unit options, 6-27
Service Policy, iii
SET (new control setting) bit
in control field of address buffer port, 2-10
SGO (Scatter/Gather Operation) bit, 5-3
Shipping Damage, 3-4
Short Burst FIFO (SBF)
description of, 2-2
Single Command Structure
examples of use, 4-14
format of, 4-10
interrupt/ID field, 4-11
parameter block in, 4-10
status block in, 4-12
using, 4-13
Specifications, C-1
SS (Selected Sense) bit, 6-46
Start Command List (01H), 6-6
Status Block
basic format of, 4-6
command identifier, 4-6
error field, 4-8
explanation of, 4-6
flags field, 4-7, 6-4
for identify command, 6-11
multiple, 4-8
SCSI sense bytes, 4-8
SCSI status field, 4-8
Status Port
errors reported, 2-18
general format, 2-17
reset format, 2-14
Stop Command List (02H), 6-8
SYN (Synchronous Negotiation) bit, 6-26, 6-44, 6-57
Synchronous Negotiation
determining synch period, offset negotiated, 6-46
enabling, 6-26, 6-57
Sysfail Jumper, 3-9

T

TAG (Tagged Queuing) bit, 6-12, 6-26, 6-41, 6-58
TAG OPTS (Tag Options) bits, 5-12, 7-27
Tagged Queuing
enabling negotiation in extended unit options, 6-58
enabling negotiation in unit options, 6-26
how to use, 7-26
notes on use, 7-28
Termination
determining if good using BTP and BSF bits, 6-42
location of fuse, 3-13
location of resistor packs, 3-12
on SCSI peripheral, 3-19
use of STT bit in diagnostics test, 6-32
Test Flags Field
in diagnostics/selftest, 6-31
Throttle, 6-19
byte or transfer throttle selection, 6-20
throttle count, 6-19
values reported in board information, 6-40
TQE (Tagged Queuing Enabled) bit, 6-45

U

Unit Flags Field
  in extended unit options, 6-57
  in unit options, 6-25
Unit Options (08H), 6-22
UNQ (Untagged Queuing) bit, 6-25, 6-44, 6-57
Untagged Queuing, 6-25
Usage, 7-1
  byte and word swapping, 7-2
  odd byte handling, 7-16
  scatter/gather, 7-10
  tagged queuing, 7-26

V

VMEbus Address
  choosing, 2-6
  use and location of jumpers, 3-10

W

Warranty, i
WID (Width of Data Transfers) bit
  in control field of address buffer port, 2-10
Word Swapping
  use of feature, 7-2
  WSC (Word Swapping Control) bit, 2-10
  WSD (Word Swap Data) bit, 2-10
WSC (Word Swapping Control) bit
  in control field of address buffer port, 2-10
WSD (Word Swap Data) bit
  in control field of address buffer port, 2-10

21010331 A

21010331 A

CIPRICO

2800 Campus Drive, Plymouth, MN 55441
(612) 551-4000

CIPRICO LISTENS. AND RESPONDS.