

ARTISAN®
TECHNOLOGY GROUP

Your **definitive** source for quality pre-owned equipment.

Artisan Technology Group
(217) 352-9330 | sales@artisantg.com | artisantg.com

Full-service, independent repair center
with experienced engineers and technicians on staff.

We buy your excess, underutilized, and idle equipment
along with credit for buybacks and trade-ins.

Custom engineering
so your equipment works exactly as you specify.

- Critical and expedited services
- In stock / Ready-to-ship
- Leasing / Rentals / Demos
- ITAR-certified secure asset solutions

Expert team | Trust guarantee | 100% satisfaction

All trademarks, brand names, and brands appearing herein are the property of their respective owners.

Find the *Interphase V/FDDI 5211* at our website: ***Click HERE***

V/FDDI 5211 Adapter Installation and Software Developers Guide

Document No. UG05211-000-E

Print Date: January 2000

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com



NOTE

See Appendix E for Regulatory Statements/Conditions that affect the operation of this product.
The CE Declaration of Conformity can be found at www.iphase.com.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Copyright Notice

© 2000 by Interphase Corporation. All rights reserved.

Printed in the United States of America, 2000.

This manual is licensed by Interphase to the user for internal use only and is protected by copyright. The user is authorized to download and print a copy of this manual if the user has purchased one or more of the Interphase adapters described herein. All copies of this manual shall include the copyright notice contained herein. No part of this manual, whether modified or not, may be incorporated into user's documentation without prior written approval of

Interphase Corporation
13800 Senlac
Dallas, Texas 75234

Phone: (214) 654-5000
Fax: (214) 654-5500

Disclaimer

Information in this manual supersedes any preliminary specifications, preliminary data sheets, and prior versions of this manual. While every effort has been made to ensure the accuracy of this manual, Interphase Corporation assumes no liability resulting from omissions, or from the use of information obtained from this manual. Interphase Corporation reserves the right to revise this manual without obligation to notify any person of such revision. Information available after the printing of this manual will be in one or more Read Me First documents. Each product shipment includes all current Read Me First documents. All current Read Me First documents are also available on our web site.

THIS MANUAL IS PROVIDED "AS IS." INTERPHASE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING THOSE OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL INTERPHASE BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Trademark Acknowledgments

Interphase® and Syncard® are registered trademarks and CellView™, (i)chip™, ADSLWatch™, ADSLEye™, SynWatch™, SynEye™, FibreView™, and the Interphase logo are trademarks of Interphase Corporation.

Microsoft®, MS-DOS®, Windows®, and Windows NT® are registered trademarks of Microsoft Corp.

Novell® and NetWare® are registered trademarks of Novell, Inc.

Solaris® and NFS® are registered trademarks and SunOS™ and ONC™ are trademarks of Sun Microsystems, Inc. Sun is a trademark or registered trademark of Sun Microsystems, Inc.

SPARC® is a registered trademark of SPARC International, Inc. SPARCstation™ and UltraSPARC™ are trademarks of SPARC International, Inc., licensed exclusively to Sun Microsystems, Inc.

LattisCell™, EtherCell™, Bay Networks™, and SAHI™ are trademarks of Bay Networks, Inc.

UNIX® is a registered trademark in the United States and other countries, licensed exclusively through X/Open Company Ltd.

AT®, IBM®, and OS/2® are registered trademarks and AIX™ and PowerPC™ are trademarks of International Business Machines Corporation.

HP-UX® is a registered trademark and Tachyon™ and Precision Bus™ are trademarks of Hewlett-Packard Company.

Intel® and Pentium® are registered trademarks and i386™, i486™, Intel386™, and Intel486™ are trademarks of Intel Corporation.

TI® is a registered trademark of Texas Instruments.

Compu-shield® is a registered trademark of Stewart Connectors Systems, Inc.

Tundra® is a registered trademark and Universe™ is a trademark of Tundra Semiconductor Corporation.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

DG/UX® and AViiON® are registered trademarks of Data General Corporation.

Apple® and Power Macintosh® are registered trademarks and Macintosh™, MacOS™, Mac™, AppleTalk™, and Open Transport™ are trademarks of Apple Computer, Inc.

NCR® is a registered trademark of NCR Corporation.

Silicon Graphics® is a registered trademark and SGI™, Indigo™, Indy™, Indigo²™, IRIX™, IRIS™, IRIS Indigo™, Challenge™, and Challenge M™ are trademarks of Silicon Graphics, Inc.

Alpha™ and DIGITAL™ are trademarks of Digital Equipment Corporation.

Gadzoox™ is a trademark of Gadzoox Microsystems, Inc.

Seagate™ and Barracuda™ are trademarks of Seagate Technology, Inc.

ST® is a registered trademark of Lucent Technologies.

SCO, The Santa Cruz Operation, SCO OpenServer, and UnixWare are trademarks or registered trademarks of The Santa Cruz Operation, Inc.

SUPERNET™ is a trademark of Advanced Micro Devices, Inc.

Cisco® is a registered trademark and Cisco Systems™ is a trademark of Cisco Systems, Inc.

Adobe® and Acrobat® are registered trademarks of Adobe Systems Incorporated.

CompactPCI® and PICMG® are registered trademarks of the PCI Industrial Computer Manufacturers Group.

Assistance

Product Purchased from Reseller

Contact the reseller or distributor if

- You need ordering, service or any technical assistance.
- You received a damaged, incomplete or incorrect product.

Product Purchased Directly from Interphase Corporation

Contact Interphase Corporation directly for assistance with this, or any other Interphase Corporation product. Please have your purchase order and serial numbers ready.

Customer Service

United States:	Telephone:	(214) 654-5555
	Fax:	(214) 654-5506
	E-Mail:	intouch@iphase.com
Europe:	Telephone:	+ 33 (0) 1 41 15 44 00
	Fax:	+ 33 (0) 1 41 15 12 13

World Wide Web

http://www.iphase.com

Anonymous FTP Server

ftp.iphase.com

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Contents

Using This Guide xv

Audience xv
Icon Conventions xv
Text Conventions xv

CHAPTER 1 INTRODUCTION

OVERVIEW 1
RELATED PUBLICATIONS AND STANDARDS 1
PRODUCT OVERVIEW 2
Design Flexibility 2
6U or 9U Compatibility 2
Single Attachment or Dual Attachment 3
Fiber and Copper Support 3
OVERVIEW OF HARDWARE/FIRMWARE DESIGN 4
Backward Compatibility 5
Motorola Chipset 6
On-Board Processor 7
32-Entry CAM 7
Communications Buffer 7
FLASH EPROM Storage 7
High-Performance VMEbus Interface 7
The FIFO Buffer 8
The Host Bus State Machine 8
On-Board Station Management (SMT) 8
CONVENTIONS 8

CHAPTER 2 INSTALLATION

INSTALLATION OVERVIEW 11
VISUAL INSPECTION 11
POWER SYSTEM OFF 11
SET ON-BOARD JUMPERS 12
On-Board Jumper Settings 15
INSTALLING BOARD(S) 17
CABLING PROCEDURE 18
Required Cables and Connectors 18
Considerations for Cabling a Dual Attachment Station 19
Keying FDDI Connectors 20
Preparation for Cabling Procedure 21
Cabling a Single Attachment Station 22
Cabling a Dual Attachment Station 23
POWER SYSTEM ON 25

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

CHAPTER 3 FUNCTIONAL OVERVIEW

OVERVIEW OF THE 5211 INTERFACE 27
SYSTEM REQUIREMENTS 28
USING THE COMMON BOOT INTERFACE 29
BOOTUP AND RESET SEQUENCE 31
RESETTING THE 5211 33
5211 DIAGNOSTICS 33
DIAG COMMAND 34
COMMAND FIELDS 34
COMMAND LENGTH 34
MAJOR TEST 34
MINOR TEST 34
TEST DATA 35
TEST RESULTS 36
DIAGNOSTIC ERROR CODES 37
USING THE REPORT/COMMAND INTERFACE 37
RC-Specific Terminology 38
Available Commands 38
SHORT I/O AND THE RC INTERFACE 39
DEFINITION OF RC CHANNELS 39
GENERAL FORMAT OF RC CHANNELS 39
CHANNEL DESCRIPTORS 40
UPDATING READ AND WRITE POINTERS 41
Reading a Pointer 42
WRITING A POINTER 43
MANAGING SEGMENT MEMORY 44
Segment Ownership 44
Host-Owned Segment Memory 44
Controller-Owned Segment Memory 45
Assigning Segment Numbers 45
Location of Segments 45
Reusing Segments 46
Rollover of Segment Numbers to 0 47
INITIALIZING THE RC INTERFACE 47
STRUCTURE OF RC INTERFACE IN SHARED MEMORY 48
MANAGING COMMANDS AND RESPONSES 48
Available RC Commands 48
Location of Commands and Tx/Rx Data 48
Managing Controller Commands 48
Linking from One Segment to the Next 51
Reading Command Responses 52
INTERRUPTS 53
Enabling Interrupts 53
Generating an Interrupt 54
Servicing Interrupts 54
Using the Interrupt Timer 55

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Channel Polling....56
Channel Polling (with Interrupts Suspended)....56
Channel Polling (with Interrupts Not Suspended)....57
ADDITIONAL CONSIDERATIONS....57
Creating Dedicated Channels....57
BUILDING COMMANDS....57
Location of Commands and Tx/Rx Data....58
COMMONLY USED COMMAND FIELDS....58
Command Control Block Format....58
Channel ID Fields....59
Transfer Options Word....59
CONNECTING THE STATION TO THE RING....61
TRANSMITTING DATA....62
Queuing Frames for Transmission....62
Synchronous vs. Asynchronous Frames....62
RECEIVING DATA....63
Alignment of Received Frames....63
Allocating And Managing Receive Buffers....63
Creating A Buffer List....63
Ownership of Receive Buffers....64
Managing the Buffer List....64
Balancing the Load of Rx Frame Processing....66
Insufficient Receive Buffers....66
SCATTER/GATHER CAPABILITIES....66
Frame Scattering....67
Frame Gathering....67
ACCESSING THE CAM....67

CHAPTER 4 5211 COMMAND SET

COMMON BOOT COMMAND SET....69
RC COMMAND SET....70
FDDI COMMAND SET....71
COMMON BOOT COMMANDS....72
DOWNLOADING CODE....74
BOOT....75
FUNCTION....75
CODE....76
PARAMETERS....76
PROGRAMMING SEQUENCE....76
ERROR RETURNS....76
DIAG....77
FUNCTION....77
CODE....77
PARAMETERS....78
PROGRAMMING SEQUENCE....78
RESPONSE BLOCK....78

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

FILL 80
FUNCTION 80
CODE 80
PARAMETERS 80
PROGRAMMING SEQUENCE 81
ERROR RETURNS 81
FLSH 82
FUNCTION 82
CODE 82
PARAMETERS 82
PROGRAMMING SEQUENCE 82
NEVER FAILS 82
GRNL 83
FUNCTION 83
CODE 83
PARAMETERS 83
PROGRAMMING SEQUENCE 83
NEVER FAILS 83
JUMP 84
FUNCTION 84
CODE 84
PARAMETERS 84
PROGRAMMING SEQUENCE 84
PEEK 85
FUNCTION 85
CODE 85
PARAMETERS 85
PROGRAMMING SEQUENCE 85
NEVER FAILS 85
POKE 86
FUNCTION 86
PARAMETERS 86
PROGRAMMING SEQUENCE 86
NEVER FAILS 86
DNLD 87
FUNCTION 87
CODE 87
PARAMETERS 87
MOTOROLA S_RECORD 87
PROGRAMMING SEQUENCE 88
ERROR RETURNS 88
BURN 89
FUNCTION 89
CODE 89
PROGRAMMING SEQUENCE 89
ERROR RETURNS 89
REDL 90
FUNCTION 90

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

CODE 90
PARAMETERS 90
PROGRAMMING SEQUENCE 90
NEVER FAILS 90
STUF 91
FUNCTION 91
CODE 91
PARAMETERS 91
PROGRAMMING SEQUENCE 92
ERROR RETURNS 92
HGET 93
FUNCTION 93
CODE 93
PARAMETERS 93
EXAMPLE 94
PROGRAMMING SEQUENCE 94
ERROR RETURNS 94
HPUT 95
FUNCTION 95
CODE 95
PARAMETERS 95
EXAMPLE 96
PROGRAMMING SEQUENCE 96
ERROR RETURNS 96
INFO 97
FUNCTION 97
CODE 97
PARAMETERS 97
PROGRAMMING SEQUENCE 98
ERROR RETURNS 98
REPORT/COMMAND (RC) COMMANDS 99
TYPES OF RC COMMANDS 100
SET DMA BURST 101
FIELDS 102
SET BUS TIMEOUT 103
FIELDS 103
REQUEST STATISTICS 104
FIELDS 104
SET INTERRUPT 105
FIELDS 105
INTERRUPT VECTOR (4 bytes) 105
CREATE HOST-TO-BOARD CHANNEL 106
FIELDS 106
CREATE BOARD-TO-HOST CHANNEL 108
FIELDS 108
LINK HOST-TO-BOARD SEGMENT 109
FIELDS 109
ALLOCATE BOARD-TO-HOST SEGMENT MEMORY 110

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

FIELDS 110
REPORT PRINTF CALL 112
FIELDS 112
ERROR MESSAGE 113
FIELDS 113
ERROR CODES 114
FIELDS 114
REPORT STATISTICS 115
FIELDS 115
REPORT NEW HOST-TO-BOARD CHANNEL 116
FIELDS 116
REPORT NEW BOARD-TO-HOST CHANNEL 117
FIELDS 117
REPORT LINK TO NEXT SEGMENT 118
FDDI-SPECIFIC COMMANDS 119
TRANSMIT RAW DATA 121
TRANSMIT RAW DATA (Embedded) 124
TRANSMIT DATAGRAM 127
TRANSMIT DATAGRAM (EMBEDDED) 130
SET UP RECEIVE BUFFERS 134
REQUEST STATION ADDRESS 136
SET STATION ADDRESS 137
RING CONTROL 139
PERFORM MAINTENANCE 140
GET MIB ATTRIBUTE 142
SET MIB ATTRIBUTE 144
SET CAM ADDRESS 148
CLEAR CAM ADDRESS 150
FLUSH CAM ADDRESSES 151
SMT TRACE 152
COMMAND CONTROL BLOCK (8 bytes) 152
TRACE CONTROL (4 bytes) 152
SET HARDWARE PARAMETERS 153
GET HARDWARE PARAMETERS 157
GET CAM ADDRESS 159
Tx RESPONSE 160
Rx FRAME 162
REPORT STATION ADDRESS 164
RING STATUS INDICATION 165
CAM ADDRESS RESPONSE 166
MIB ATTRIBUTE RESPONSE 168
SMT EVENT INDICATION 170
REPORT HARDWARE PARAMETERS 171
REPORT CAM ADDRESS 174

CHAPTER 5 SPECIFICATIONS

ARCHITECTURE 175

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

VMEBUS SPECIFICATIONS 175
FDDI SPECIFICATIONS 175
TIMER DEFAULTS 176
POWER REQUIREMENTS 176
MECHANICAL (NOMINAL) 176
OPERATING ENVIRONMENT 177
STORAGE ENVIRONMENT 177
LEDs 177
DEFAULT REPORT/COMMAND PARAMETERS 178
OPTICAL BYPASS SWITCH PIN ASSIGNMENTS 179
P1 AND P2 CONNECTORS PINOUTS (VMEbus) 179

APPENDIX A DIAGNOSTICS

INTRODUCTION 181
ISSUING COMMANDS 182
SHORT I/O 183
TEST FAILURES 183
COMMAND/RESPONSE STRUCTURE 183
MEMORY TEST DIAGNOSTICS 184
MEMORY TESTS - MAJOR TEST CODE 0x1 184
OPTIONS - 0x0 185
MEMORY TEST ERROR CODES 185
POWER-UP DIAGNOSTIC TEST DESCRIPTION 186
EXTENDED DIAGNOSTIC TEST DESCRIPTION 186
FRONT END FDDI DIAGNOSTICS 187
FRONT END FDDI CHIPSET TESTS - MAJOR TEST CODE 0x2 187
FRONT END TEST DESCRIPTION 187
0x4 - 0x7 — EXTERNAL LOOPBACK TEST 188
POWER-UP DIAGNOSTIC DESCRIPTION 189
TEST RESULTS 189
DMA TEST DIAGNOSTICS 193
DIAGNOSTIC DMA TEST - MAJOR TEST 0x3 193
ADDGEN STATUS REGISTERS 193
AVAILABLE DMA DIAGNOSTIC COMMANDS 194
0x0 - PUP_TEST 194
0x1 - SHIO 195
0x2 - STANDARD DMA 195
0x3 - SWAP 195
0x4 - RAW DMA 196
0x5 - READ BUFFER 196
0x6 - WRITE BUFFER 198
0x7 - VIRQ 199
0x8 - DUMP ASIC REGISTERS 199
ERROR CODES 200
MEMORY MAP 201
CAM TEST DIAGNOSTICS 201

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

CAM TEST - MAJOR TEST 0x7 201
CAM TEST DESCRIPTION 201
TEST RESULTS 202

APPENDIX B DOWNLOADING FIRMWARE

DOWNLOAD FORMATS 203
DOWNLOAD AND EXECUTE A CB-FORMAT BINARY IMAGE 203
STORE A DOWNLOADED PROGRAM IN FLASH MEMORY 205
DOWNLOAD CB IMAGE FORMAT 206
MOTOROLA S-RECORDS 208
FLASH EPROM LAYOUT 209
SECONDARY IMAGE UPDATES 209
CB ERROR STATUS CODES I 209
CB ERROR STATUS CODES II 210
CB FAIL STATUS BLOCK FORMAT 212
MEMORY TEST POWER-UP CODES 212
CPU TEST POWER-UP CODES 213
EPROM TEST POWER-UP CODES 213
HARDWARE CRITICAL POWER-UP CODES 213
ILLEGAL INTERRUPT POWER-UP CODES 215

APPENDIX C CB ERROR CODES

CB ERROR STATUS CODES I 217
CB ERROR STATUS CODES II 218
CB FAIL STATUS BLOCK FORMAT 219
MEMORY TEST POWER-UP CODES 220
CPU TEST POWER-UP CODES 221
EPROM TEST POWER-UP CODES 222
HARDWARE CRITICAL POWER-UP CODES 222
ILLEGAL INTERRUPT POWER-UP CODES 223

APPENDIX D JUMPER SETTINGS

APPENDIX E Regulatory Statements

Interphase Fiber Products' Compliance 231
EN60950–IEC950 Safety Standard 231
Canadian 231
Regulatory Information for Europe 231

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

List of Figures

Figure 1-1. Hybrid FDDI Network 4
Figure 1-2. V/FDDI 5211 Architecture 6
Figure 2-1. Jumper Settings for V/FDDI 5211 DAS 14
Figure 2-2. VMEbus Request Priority Jumper Setting 17
Figure 2-3. ST- and FDDI-Standard Connector 18
Figure 2-4. Flow of Tx/Rx Signals between PHYs of three 5211 DASs (No Fault in Ring) 20
Figure 2-5. Cabling a Single Attachment Station (ST-to-MIC) 22
Figure 2-6. Cabling a Dual Attachment Station 24
Figure 2-7. Example Optical Bypass Switch 25
Figure 3-1. Short I/O as a Window into the 5211 Communications Buffer Address Space 28
Figure 3-2. Common Boot Command/Response Format 30
Figure 3-3. Polling CBOK for Commands or Responses 30
Figure 3-4. CB_HERALD Format 31
Figure 3-5. Format of Shared Memory after Controller Initialization 32
Figure 3-6. Format of Shared Memory after FAIL 32
Figure 3-7. Format of the DIAG command 34
Figure 3-8. “FAIL” Returned Status Block 36
Figure 3-9. Channel Descriptor Format 41
Figure 3-10. Reading a Pointer 43
Figure 3-11. Control Structure of RC Interface 49
Figure 3-12. Example of Issuing a 16-Byte Host-to-Board Command 51
Figure 3-13. Linked Segments 52
Figure 3-14. Command Control Block Structure 58
Figure 3-15. Transfer Options Word for Data Transfers 59
Figure 3-16. Transfer Options for DMA Transfer in SUN 3/260 4.1 Operating System 61
Figure 3-17. Format of Buffer List Element 64
Figure 4-1. BOOT Command and Response Format 75
Figure 4-2. Format of the DIAG command 77
Figure 4-3. “FAIL” Response Block 77
Figure 4-4. FILL Command and Response Format 80
Figure 4-5. FLSH Command and Response Format 82
Figure 4-6. GRNL Command and Response Format 83
Figure 4-7. JUMP Command and Response Format 84
Figure 4-8. PEEK Command and Response Format 85
Figure 4-9. POKE Command and Response Format 86
Figure 4-10. DNLD (DOWNLOAD) Command and Response Format 87
Figure 4-11. BURN Command and Response Format 89
Figure 4-12. REDL Command and Response Format 90
Figure 4-13. STUF Command and Response Format 91
Figure 4-14. HGET Command and Response Format 93
Figure 4-15. HPUT Command and Response Format 95
Figure 4-16. Information Command (INFO) Format 97
Figure 4-17. CB Information Structure Type 0 98
Figure 4-18. EXEC 0 Information Structure Type 1 98

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Figure 4-19. Format of Set DMA Burst Directive 102
Figure 4-20. Format of Set Bus Timeout Directive 103
Figure 4-21. Format of Request Statistics 104
Figure 4-22. Format of Set Interrupt Directive 105
Figure 4-23. Format of Create Host-to-Board Channel Request 106
Figure 4-24. Format of Create Board-to-Host Channel Request 108
Figure 4-25. Format of Link Host-to-Board Segment Directive 109
Figure 4-26. Format of Allocate Board-to-Host Segment Memory 110
Figure 4-27. Format of Report Printf Call 112
Figure 4-28. Format of Error Message Indication 113
Figure 4-29. Format of Report Statistics Response 115
Figure 4-30. Format of Report New Host-to-Board Channel 116
Figure 4-31. Format of Report New Host-to-Board Channel 117
Figure 4-32. Format of Link Board-to-Host Segment Indication 118
Figure 4-33. Request Class Field for Transmit Raw Data 122
Figure 4-34. Request Class Field for Transmit Raw Data 125
Figure 4-35. Request Class Field for Transmit Datagram 129
Figure 4-36. Request Class Field for Transmit Datagram 132
Figure 4-37. Buffer List Element 134
Figure 4-38. C Code for Set MIB Example 147
Figure 4-39. Frame Status Field 163
Figure A-1. Structure of the DIAG Command183
Figure A-2. “FAIL” Returned Status Block184
Figure A-3. Memory Test Command Format185
Figure A-4. Memory Test Command Options Field.185
Figure A-5. CB Diagnostic Error Data Structure186
Figure A-6. Bit Structure of the EXTERNAL LOOPBACK TEST Data Options188
Figure A-7. External Loopback Test Command Format189
Figure A-8. Power-up Test Command Format195
Figure A-9. Standard DMA Test Command Format195
Figure A-10. RAW DMA Test Command Format196
Figure A-11. Bit Structure of the READ BUFFER Data Options197
Figure A-12. Read Buffer Command Format198
Figure A-13. VIRQ Command Format199
Figure A-14. Dump ASIC Registers Command Format200
Figure B-1. Using POKE Command to Download Code204
Figure B-2. Using BOOT command to run downloaded program205
Figure B-3. Using BURN to store a downloaded program in flash memory205

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

List of Tables

Table 2-1. V/FDDI 5211 On-Board Jumpers 12
Table 2-2. Short I/O Base Address Jumpers 16
Table 2-3. Jumper settings for Short I/O Base Address 16
Table 2-4. Sample Parts for Cabling the V/FDDI 5211 19
Table 2-5. Keying Scheme on AMP FDDI Connectors 21
Table 3-1. Major Test Numbers 35
Table 3-2. Diagnostic Error Codes 37
Table 3-3. Generic RC Commands 38
Table 3-4. 5211-Specific Commands 38
Table 3-5. Address Modifiers Allowed in Data Transfers 60
Table 3-6. Memory Type for Data Transfers 60
Table 4-1. V/FDDI 5211 Common Boot Commands 72
Table 4-2. RC Command Set 99
Table 4-3. Types of RC Commands 100
Table 4-4. Error Codes 114
Table 4-5. V/5211 FDDI-Specific Commands 119
Table 4-6. Transmit Raw Data Request 121
Table 4-7. Transmit Raw Data (Embedded) Request 124
Table 4-8. Transmit Datagram Request 127
Table 4-9. Transmit Datagram (Embedded) Request 130
Table 4-10. Set Up Receive Buffers Directive 134
Table 4-11. Request Station Address 136
Table 4-12. Set Station Address Directive 137
Table 4-13. Type Codes in Set Station Address Directive 138
Table 4-14. Ring Control Directive 139
Table 4-15. Perform Maintenance Directive 140
Table 4-16. Line States Available for Perform Maintenance Directive 140
Table 4-17. Get MIB Attribute Request 142
Table 4-18. Valid Entries in MIB Attribute Index Field 143
Table 4-19. Set MIB Attribute Directive 144
Table 4-20. Valid Entries in MIB Attribute Index Field 145
Table 4-21. Example Set MIB Command (*FDDI SMTConnectionPolicy*) 146
Table 4-22. Set CAM Address Request 148
Table 4-23. Clear CAM Address Request 150
Table 4-24. Flush CAM Addresses Directive 151
Table 4-25. SMT Trace Directive 152
Table 4-26. Values in Trace Control Field 152
Table 4-27. Set Hardware Parameters Directive 153
Table 4-28. Host-Settable Registers and Functions 154
Table 4-29. Set MAC Mode 154
Table 4-30. Set PMode 155
Table 4-31. Set Bridge Stripping 155
Table 4-32. FCS Generation Mode Control 155
Table 4-33. Optical Bypass Relay Control 155
Table 4-34. Get Hardware Parameters Request 157
Table 4-35. Host-Readable Registers and Functions 158
Table 4-36. Get CAM Address Request 159
Table 4-37. Tx Response 160

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table 4-38. Transmit Completion Status Field 160
Table 4-39. Rx Frame Indication 162
Table 4-40. Report Station Address 164
Table 4-41. Ring Status Indication 165
Table 4-42. Ring Status Codes 165
Table 4-43. CAM Address Response 166
Table 4-44. Status Codes in CAM Address Response 166
Table 4-45. MIB Attribute Response 168
Table 4-46. SMT Event Indication 170
Table 4-47. Report Hardware Parameters Response 171
Table 4-48. Parameter Types for Report Hardware Parameters 172
Table 4-49. MAC Mode 172
Table 4-50. PMode 172
Table 4-51. Bridge Stripping 172
Table 4-52. Returned FCS Generation Modes 173
Table 4-53. Returned Optical Bypass Control Modes 173
Table 4-54. Report CAM Address 174
Table 5-1. Ring State LEDs (LED5-LED7) 177
Table 5-2. P1 Connector - VMEbus 179
Table 5-3. P2 Connector - VMEbus (P2 Row B) 180
Table A-1. Quick Reference to Power-up Diagnostics 182
Table A-2. Minor Test Codes for Memory Tests 184
Table A-3. Minor Test Codes for Front-End Tests 187
Table A-4. Error Codes for Front End Tests 190
Table A-5. NO_RX_INDICATION 190
Table A-6. RX_ERROR 191
Table A-7. RX_CRC_ERROR 191
Table A-8. RX_IER_ERROR (FSI Internal Error) 191
Table A-9. RX_FRAME_MISCOMPARE 191
Table A-10. RX_FRAME_SIZE_ERROR 192
Table A-11. BIST Test Data 192
Table A-12. Addgen General Status Register - High True Values 193
Table A-13. Addgen I/O Status Register - High True Values 193
Table A-14. MBC Interrupt Port - Low True Values 193
Table A-15. Minor Test Codes for DMA Diagnostic Tests 194
Table A-16. Data Pattern Values 198
Table A-17. DIAGNOSTICS Error Codes 200
Table A-18. 5211 Hardware Memory Map 201
Table A-19. CAM Test Error Codes 202
Table A-20. CAM_COMPARE_FAIL 202
Table B-1. CB Image Download Format 206
Table B-2. Motorola S-Record Fields 208
Table 4-3. Layout of FLASH EPROM 209
Table B-4. CB Error Status Codes I 209
Table B-5. CB Error Status Codes II 210
Table B-6. Memory Test Power-Up Codes 212
Table B-7. CPU Test Power-Up Codes 213
Table 4-8. 213
Table B-9. Hardware Critical Power-up Codes 213
Table B-10. Illegal Interrupt Power-Up Codes 215
Table C-1. CB Error Status Codes I 217
Table C-2. CB Error Status Codes II 218

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table C-3. Memory Test Power-Up Code 220
Table C-4. CPU Test Power-Up Codes 221
Table C-5. EPROM Test Power-Up Codes 222
Table C-6. Hardware Critical Power-Up Codes 222
Table C-7. llegal Interrupt Power-up Codes 223
Table D-1. Base Address Jumper Settings 225

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Using This Guide

Audience

This manual assumes that its audience has a general understanding of computing and networking terminology.

Icon Conventions

Icons draw your attention to especially important information:



NOTE

The Note icon indicates important points of interest related to the current subject.



CAUTION

The Caution icon brings to your attention those items or steps that, if not properly followed, could cause problems in your machine's configuration or operating system.



WARNING

The Warning icon alerts you to steps or procedures that could be hazardous to your health, cause permanent damage to the equipment, or impose unpredictable results on the surrounding environment.

Text Conventions

The following conventions are used in this manual. Computer-generated text is shown in typewriter font. Examples of computer-generated text are: program output (such as the screen display during the software installation procedure), commands, directory names, file names, variables, prompts, and sections of program code.

```
Computer-generated text example
```

Commands to be entered by the user are printed in **`bold Courier`** type. For example:

```
cd /usr/tmp
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Pressing the return key (↵ **Return**) at the end of the command line entry is assumed, when not explicitly shown. For example:

```
/bin/su
```

is the same as:

```
/bin/su  ↵ Return
```

Required user input, when mixed with program output, is printed in **bold Courier** type. References to UNIX programs and manual page entries follow the standard UNIX conventions.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

INTRODUCTION 1

OVERVIEW

This manual is intended for users who need to write a software driver for the V/FDDI 5211. It can also be used to install and/or test the 5211.

Readers are assumed to have extensive knowledge of the following;

- FDDI specifications, including FDDI Station Management (SMT) specifications (ANSI X3T9.5/84-49)
- C programming language, including experience writing and installing drivers
- Operating system of the host computer
- VMEbus specifications and hardware installation procedures

The level of expertise will, of course, depend on the task to be performed.

RELATED PUBLICATIONS AND STANDARDS

The following publications are excellent resources of information when working with this product. If you do not have access to the Interphase publications, call Interphase Corporation, Customer Service.

- Motorola Fiber Distributed Data Interface User's Manual, MC68836 (FDDI Clock Generator)
  Motorola Fiber Distributed Data Interface User's Manual, MC68837 (Elasticity Buffer and Link Management)
  Motorola Fiber Distributed Data Interface User's Manual, MC68838 (Media Access Controller)
  Motorola Fiber Distributed Data Interface MC68839 (FDDI System Interface)
- The VME64 Specification
- VME International Trade Association
  10229 North Scottsdale Rd., Suite B
  Scottsdale, AZ 85253
  VMEbus specifications and terminology.
- VMEbus Specification Rev. D, IEEE
  IEEE Service Center
  Publications Sales Dept.
  445 Hoes Lane
  Picscataway, NJ 08854-4150
- Writing a Unix® Device Driver
- Janet I. Egan/Thomas J Teixeira
  John Wiley and Sons, Inc. 1988

  An excellent introduction to writing Unix drivers. Discusses how Unix performs input/output, how device drivers relate to the hardware I/O architecture, and how

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

to incorporate the device driver into the Unix kernel. It also contains an excellent discussion of how to test drivers.

- FDDI references:

  ANSI X3.229: Information Systems FDDI Station Management (SMT)

  ANSI X3.139: FDDI Token Ring Media Access Control (MAC)

  ANSI X3.148: Token Ring FDDI Physical Layer Protocol (PHY)

  ANSI X3.166: Token Ring FDDI Physical Layer Medium Dependent (PMD)

  TP-PMD/ANSI X3.263: Twisted Pair Physical Layer Medium Dependent (TP-PMD)

For more information about these standards, see
http://sholeh.nswc.navy.mil/x3t12/overview.html

For information about which revisions of these standards the 5211 complies with, see
http://www.iphase.com/Public/AppEng/techsup/fddi_tech

The FDDI specifications listed above may be ordered from either of the following organizations:

Primary source:	Secondary source:
Global Engineering 2805 McGraw Ave. Irvine, CA 92714 (800) 854-7179	American National Standards Institute 1430 Broadway New York, NY 10018 (212) 354-3300

PRODUCT OVERVIEW

The V/FDDI 5211 is a high-performance node processor for 125 Mbps fiber optic FDDI (Fiber Distributed Data Interface) networks. Besides connecting a workstation or computer system based on VMEbus to an FDDI network, the 5211 can perform much of the communications protocol processing, plus certain network management functions required by FDDI.

Design Flexibility

The 5211 can be configured to support any FDDI implementation, including the use of FDDI as a high-speed local area network or as a backbone connecting other networks.

6U or 9U Compatibility

Because of its 6U form factor, the 5211 can be integrated into 6U and, with an adapter, 9U VMEbus card cages.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Single Attachment or Dual Attachment

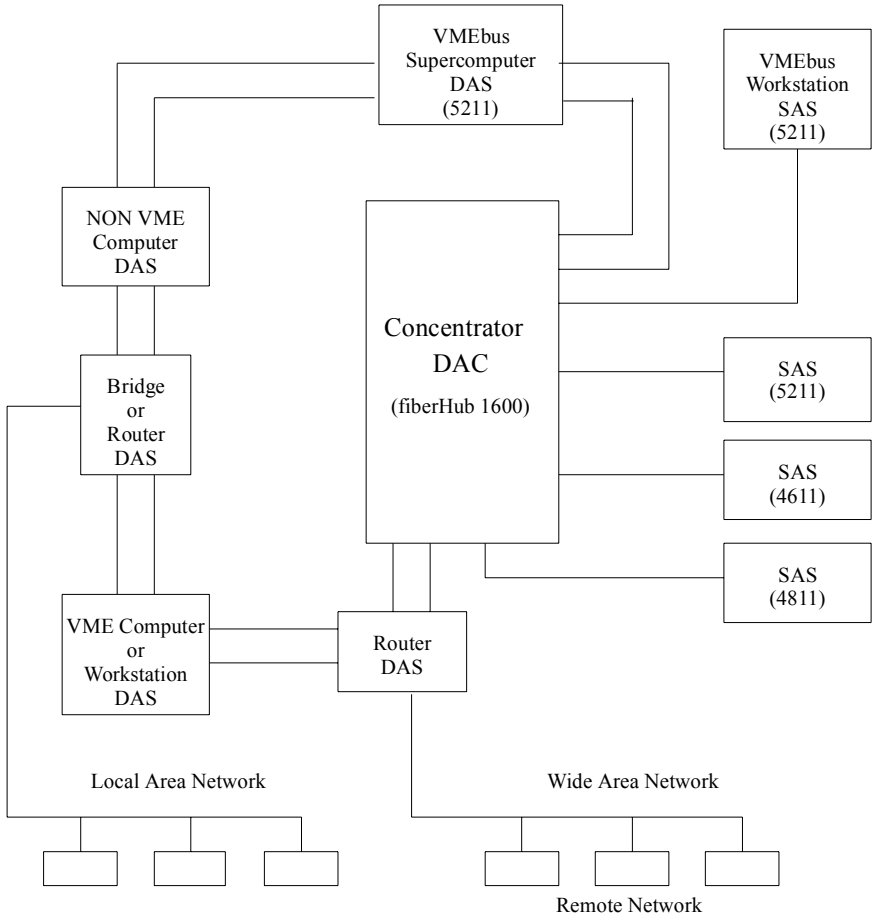
The 5211 supports both Class A Dual Attachment and Class B Single Attachment stations as defined in ANSI's FDDI specification.

Combined with other Interphase products, hybrid configurations are built using Single Attach Stations (SAS), Dual Attach Stations (DAS), Concentrators, and other standard FDDI devices. An example of a hybrid system is shown in Figure 1-1.

Fiber and Copper Support

The 5211 is designed to accomodate several standard physical interfaces including ST® and MIC fiber plus copper interface modules defined by the ANSI TP-PMD specification.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com



SAS = Single Attachment Station
DAS = Dual Attachment Station
DAC = Dual Attach Concentrator

Figure 1-1. Hybrid FDDI Network

The 5211 supports either single attach or dual attach stations in compliance with the FDDI specification.

OVERVIEW OF HARDWARE/FIRMWARE DESIGN

Key design features of the V/FDDI 5211 include:

- Backward Compatibility with 4211
- High Performance VMEbus Interface

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

- Motorola Chip Set
- Motorola 68EC040 processor
- 1 or 4 Megabyte Communications Buffer
- FLASH EPROM Storage
- High-speed proprietary VMEbus interface
- VME64
- Unaligned transfers
- Queued Host Bus State Machine
- On-Board Station Management (SMT)

Figure 1-2 is a diagram of the V/FDDI 5211 architecture.

Backward Compatibility

The 5211 is driver compatibility with the 4211 so no additional development should be required for existing 4211 customers to use the 5211.



NOTE

Some hardware-dependent Diagnostic code (not part of the driver) may not run properly on the 5211.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

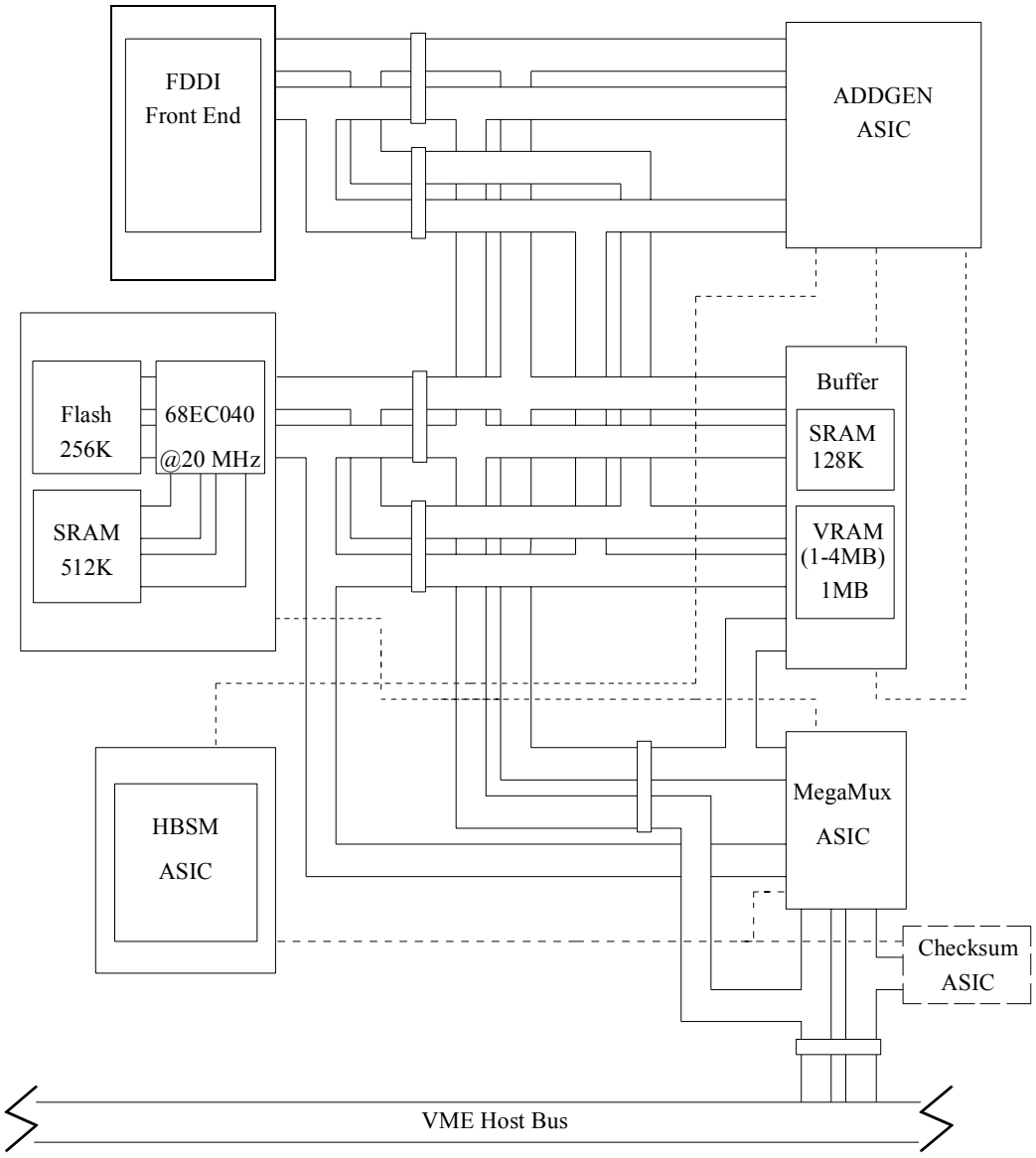


Figure 1-2. V/FDDI 5211 Architecture

Motorola Chipset

For its interface to the FDDI network, the 5211 utilizes members of Motorola's FDDI chipset.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

On-Board Processor

The 5211 uses Motorola's 68EC040 processor. This processor runs at 25 MHz, zero wait-states. To maximize performance, the 5211 includes 512K of dedicated execution RAM. This frees memory and enables the processor to execute at rates approaching its peak performance. If desired, the host can download protocol software to the 5211 for execution by the processor. Contact Interphase Corporation for a list of available protocols.

32-Entry CAM

The 5211 supports on-board content-addressable memory (CAM). This feature enables the board to receive frames addressed to 32 different destinations.

Communications Buffer

The 5211 communications buffer consists of 1MB VRAM and 128K SRAM in the default configuration. the 5211 firmware utilizes the characteristics of both memory types for optimum performance.

Memory may be expanded for custom applications such as downloadable protocols. The maximum memory capacities for the 5211 are: 512K Buffer SRAM, 4MB VRAM, and 512 bytes (optionally 2 K) of VMEbus shared “short I/O” space is also defined in the communications buffer SRAM.

FLASH EPROM Storage

256K of FLASH programmable EPROM is provided to permanently store 5211 firmware images. Firmware updates are greatly simplified and host memory storage of download images are no longer required. Up to 2 MB of FLASH memory is possible for very large images.

High-Performance VMEbus Interface

The 5211 features Interphase’s exclusive Host Bus State Machine (HBSM) ASIC for its interface to the VMEbus, and is capable of performing VMEbus data transfers in either standard 32-bit or high speed 64-bit mode.

Essentially, the HBSM ASIC decouples VMEbus activity from other activity on the 5211 through the use of video RAMs and an asynchronous state machine that monitors VMEbus signalling. The HBSM ASIC formats packets of data in the FIFO side of the video RAM before seizing control of the VMEbus. The FIFOs are emptied at speeds that are limited only by the speed of the host system's memory.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The FIFO Buffer

The HBSM ASIC controls a very high speed 2K FIFO. It can write a packet to (or read a packet from) the V/FDDI 5211's communications buffer with a fast VMEbus Block Transfer. Because the FIFO itself is very fast, once it is connected to the VMEbus, the DMA can proceed at speeds exceeding 40 megabytes/second. The speed of the 5211 on the VMEbus is totally independent of the speed of the communications buffer.

The Host Bus State Machine

The VMEbus interface of an FDDI network controller must be able to transfer data at rates significantly faster than the FDDI network. Otherwise, data coming from the network to the host will "hog" bus bandwidth. The 5211's VMEbus data transfer capabilities approach 30 Megabytes/second in 32-bit mode and 40 Megabytes/second in 64-bit mode. This high-speed operation is critical in FDDI applications, due to the speed of the FDDI network (100 Megabits/second or 12.5 Megabytes/second). The high-speed DMA transfer of the HBSM ASIC implements an asynchronous state machine that runs off a tapped delay line. This configuration avoids metastable states without incurring the delays caused by synchronizing the VMEbus signals.

On-Board Station Management (SMT)

The 5211 supports FDDI Station Management (SMT) capabilities in on-board firmware. By running SMT protocols on-board, the 5211 frees the host to perform other tasks.

FDDI's SMT can be thought of as a vertical communications protocol environment stretching across the first two levels of the OSI models. It includes functions such as connection management, which affects ring formation and fault isolation/recovery, and frame-based protocols for network management.

CONVENTIONS

This section details many of the writing conventions used throughout the manual. In addition, it gives many of the technical conventions.

- The terms "controller" and "5211" are synonymous.
- The term "short I/O" refers to a VMEbus-addressed block of memory in which only the lower 16 VMEbus address lines are used for transactions. The upper 16 VMEbus address lines are not used.
- The term "byte" represents 8 bits of data; "word" represents 16 bits (2 bytes); and "long-word" represents 32 bits (2 words/4 bytes).
- Binary (single bit) data is represented as either '1' or '0'.
- When used in the context of a single bit of data, the term "set" means that the bit is a one ('1'). Similarly, the term "cleared" means that the bit is a zero ('0').
- To represent hexadecimal numbers, the manual adopts the C language notation. Decimal numbers are shown as decimal digits. For example:

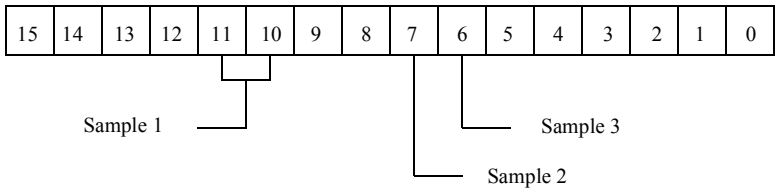
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

0x29 = 29 hex
41 = 41 decimal

- Many commands contain bits, bytes, or words which are marked “RESERVED”. Such reserved fields fall into two categories:
- Reserved for future use, must be cleared ('0', 0x00, or 0x0000).
- Reserved for future use, do not write to this field.

In this manual, when a data structure contains reserved field(s), a statement about how the host should treat the reserved field(s) is included. The statement is placed immediately after the data structure.

- When showing binary representations of bytes or words, the diagrams may have many bits which do not have names. These are RESERVED. As an example:



Bits 10 and 11 are called Sample 1, bit 7 is called Sample 2, and bit 6 is called Sample 3. All other bits are RESERVED.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

INSTALLATION 2

INSTALLATION OVERVIEW

Before attempting installation, read this chapter to ensure the safe and proper installation of the 5211 in your system. If you have any questions regarding installation which are not answered in this guide, please contact Interphase Customer Service.

The 5211 is installed into the VMEbus system using the following six steps:

1. Visual Inspection
2. Power System OFF
3. Set On-Board Jumpers
4. Install Board
5. Cabling Procedure
6. Power System ON



CAUTION

Do not install or apply power to a damaged board. Failure to observe this warning could result in extensive damage to the board and/or system.

The 5211 is sensitive to electrostatic discharge (ESD), and the board can be damaged if handled improperly. Interphase ships the board enclosed in a special anti-static bag. Upon receipt of the board, take the proper measures to eliminate board damage due to ESD (that is, wear a wrist ground strap or other grounding device).

VISUAL INSPECTION

Before attempting the installation of this board, make sure you are wearing an anti-static or grounding device.

Remove the 5211 from the anti-static bag and visually inspect it to ensure no damage has occurred during shipment. If the board is undamaged and all parts are accounted for, proceed with the installation.

POWER SYSTEM OFF

Ensure that the host system and peripherals are turned OFF.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

SET ON-BOARD JUMPERS

Set all on-board jumpers so that the 5211 is properly configured for operation within your system.



NOTE

Some jumpers' default settings should not or cannot be altered. You can use the other jumpers to help configure the board for your particular system.

The on-board jumpers are summarized in Table 2-1. To locate the jumpers, refer to the board layout in Figure 2-1.

Table 2-1. V/FDDI 5211 On-Board Jumpers

Jumper	Function
JA1 - JA4	Reserved
JA5	In=SAS; Out=DAS
JA6-JA8	Reserved
JA9	In=Auto-Boot Enabled (default); Out=Auto-Boot Disabled
JA10-JA12	Reserved
JA13	Noisy DTACK Filter
JA14	Fairness
JA15	VME Arbitration Configuration
JA16 - JA17	Reserved
JA18	Short I/O size
JA19	VRAM Slave Enable
JA20	VRAM Slave Address Modifier
JA21	VRAM Slave Address (Va23)
JA22	VRAM Slave Address (Va22)
JA23	SRAM Short I/O Address Modifier
JA24	Short I/O Address (Va15)
JA25	Short I/O Address (Va14)
JA26	Short I/O Address (Va13)
JA27	Short I/O Address (Va12)
JA28	Short I/O Address (Va11)

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table 2-1. V/FDDI 5211 On-Board Jumpers (cont)

Jumper	Function
JA29	Short /O Address (Va10)
JA30	Short I/O Address (Va9)
JA31 - JA42	VMEbus Request Level

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

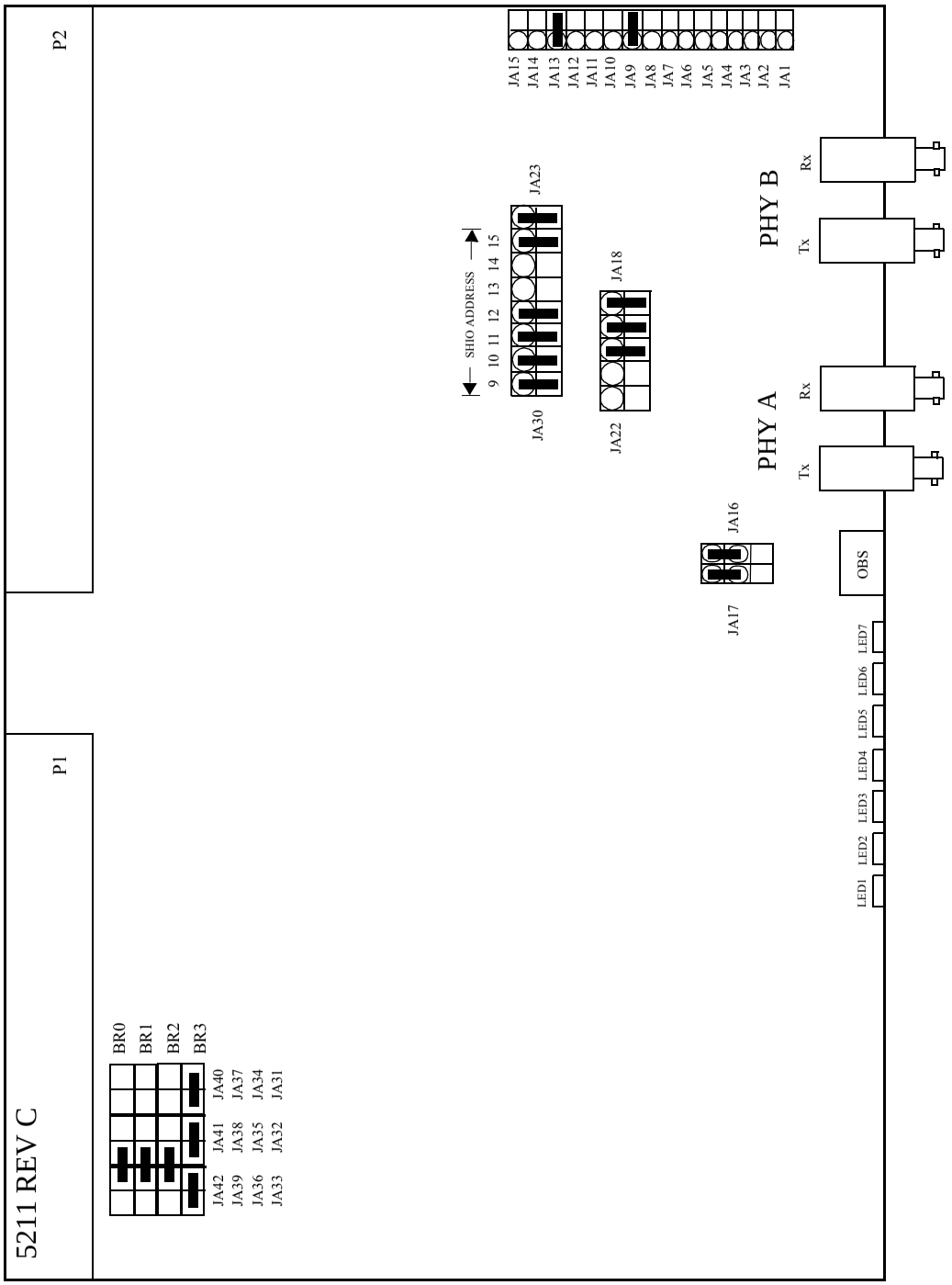


Figure 2-1. Jumper Settings for V/FDDI 5211 DAS

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

On-Board Jumper Settings

JA1 - JA4 Reserved

JA5 Determines whether the 5211 is configured as a Single Attached Station (SAS) or a Dual Attached Station (DAS). If the components for PHY A are not populated (see Figure 2-1), then jumper JA5 **must** be **installed** and denotes a SAS configuration. If both PHY layer components are populated, then jumper JA5 **must** be **removed**, indicating a DAS configuration.

JA6 - JA8 Reserved

JA9 Determines whether Auto-Boot is enabled (default) or disabled. Auto-Boot controls the firmware boot process. The default setting works with all Interphase firmware configurations. Disable Auto-Boot only if you are using a custom firmware configuration.

JA10 - JA12 Reserved

JA13 Noisy DTACK Filter. If this jumper is IN (Default), DTACK is not filtered (that is, there is no additional immunity against noisy DTACK). This can save 20ns/cycle for systems that are dependent on the release of DTACK for the next cycle to start. If this jumper is OUT, DTACK is filtered. Some systems have considerable ringing on the trailing edge of DTACK. Since the 5211 is an asynchronous design, the circuitry may interpret this noise as a response to the next transfer. Removing this jumper provides extra filtering and causes the trailing edge of DTACK to be sampled by a clock, eliminating the illegal transition.

JA14 Fairness. This jumper installed will cause the 5211 to not re-request the VME bus until it has seen its bus request level become inactive.

JA15 VME Arbitration Configuration. In, early; out, standard. This jumper determines whether the 5211 will use Early or Standard release when releasing the bus during VME mastership. Early release will allow the 5211 to release the bus after the last VAS of ownership. Standard release will hold BBSY until all 5211 VME control signals are inactive.

JA16 - JA17 Reserved

JA18 Short I/O Size. If in, 512 bytes; if out, 2K. This jumper should always be installed.

JA19 VRAM slave enable. If JA19 is not installed, the entire 4MB internal VRAM buffer is available in the VME A24 space. This jumper should normally be installed.

NOTE: This option is not available on default hardware configurations. If you require VRAM access, contact Interphase for configuration information.

JA20 VRAM slave address modifiers. Installed, this jumper allows the vram address compare circuitry to respond to either privileged or non-privileged address modifiers (0x3D and 0x39). If not installed, this jumper allows only privileged address modifiers.

JA21 VRAM Slave Address. VME address 23.

JA22 VRAM Slave Address. VME address 22.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

JA23 SRAM Short I/O Address Modifiers. This jumper installed allows the short I/O compare circuitry to accept either privileged or non-privileged address modifiers (0x29 and 0x2D). This jumper not installed allows privileged accesses only (0x2D).

JA24 - JA30 Short I/O Base Address. Jumpers JA24 through JA30 set the base address of the 512 bytes of short I/O space RAM on the 5211. See the Table 2-2 to determine the Base Address settings.

Table 2-2. Short I/O Base Address Jumpers

Jumper	VMEbus Address Bit
JA24	A15
JA25	A14
JA26	A13
JA27	A12
JA28	A11
JA29	A10
JA30	A9

Removing a jumper sets the corresponding address bit to ‘1’. Installing a jumper clears the address bit to ‘0’. The short I/O base address must be a multiple of 0x200. Table 2-3 illustrates possible Short I/O Base Address settings.

Table 2-3. Jumper settings for Short I/O Base Address

Jumper Settings							VMEbus Address							Base Address
JA24	JA25	JA26	JA27	JA28	JA29	JA30	A15	A14	A13	A12	A11	A10	A9	
IN	IN	IN	IN	OUT	IN	IN	0	0	0	0	1	0	0	0x0800
IN	OUT	IN	OUT	OUT	IN	OUT	0	1	0	1	1	0	1	0x5A00
IN	OUT	OUT	IN	IN	IN	IN	0	1	1	0	0	0	0	0x6000
OUT	IN	OUT	IN	OUT	IN	OUT	1	0	1	0	1	0	1	0xAA00

JA31 - JA42 VMEbus Request Level. Jumper fields JA30 through JA42 are used to set the 5211’s VMEbus request priority. See Figure 2-2.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Figure 2-2 shows possible valid configurations for this jumper block.

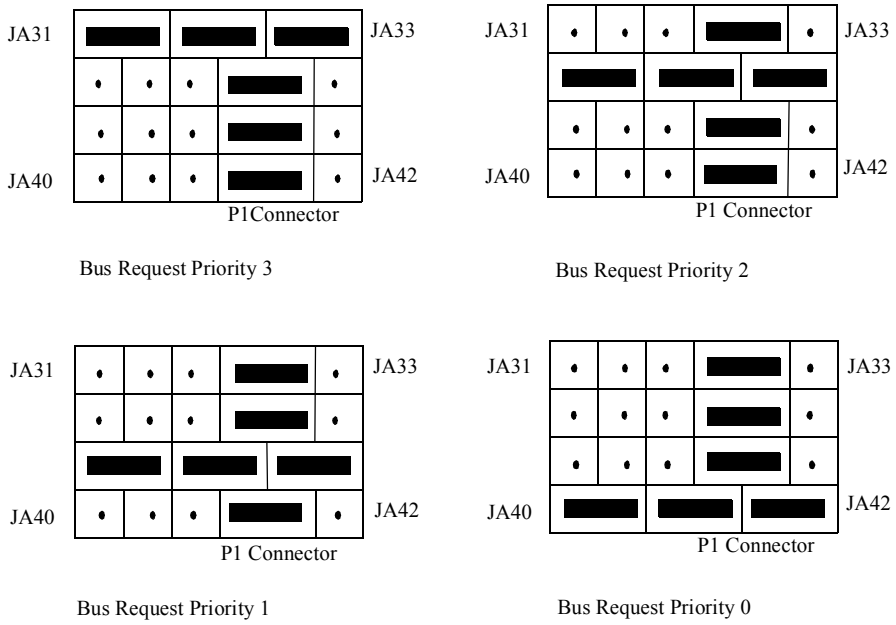


Figure 2-2. VMEbus Request Priority Jumper Setting

INSTALLING BOARD(S)

Prior to installing the 5211 in the VMEbus, configure the backplane jumpers, if required, for each board to be installed. Consult your system documentation for backplane configuration requirements. Carefully slide the 5211 into the VMEbus card slot. The board should slide in and seat without difficulty. If there is a difficulty, remove the board out and check for obstructions.

Once the board(s) are properly seated in the slot), tighten the captive mounting screws on each end of the board front panel(s).

CAUTION

System power and peripherals power must be turned OFF before attempting to install the 5211. Failure to do so may result in severe damage to the board and/or system.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

CABLING PROCEDURE

This section provides generic cabling instructions. Other configurations are possible. For information to select the optimum cables and connectors for your installation, contact your FDDI cable manufacturer. The cabling procedure varies depending on your setup:

- Single attachment station (see page 22)
- Dual attachment station (see page 23)

Required Cables and Connectors

Either ST-compatible optical connectors and FDDI-standard MIC (Media Interface Connector) receptacles are used on the 5211. Both fiber and copper PMD (Physical Medium Dependent) standard interfaces are available on the 5211. The 9U version of the V/FDDI 5211 uses FDDI-standard MIC receptacles. The 6U version of the 5211 uses either ST-compatible optical connectors or FDDI-standard MIC receptacles. Figure 2-3 shows all connector types.

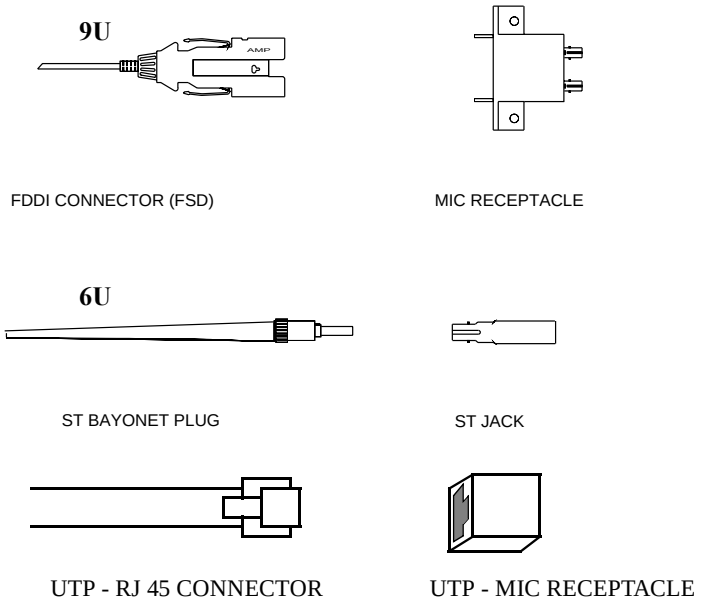


Figure 2-3. ST- and FDDI-Standard Connector

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The following table lists sample part numbers of components that can be used to set up various types of FDDI stations. These components are referenced in the cabling instructions in this chapter.

Table 2-4. Sample Parts for Cabling the V/FDDI 5211

Type of Part[a]	Sample Part Number	Purpose
ST-to-MIC Cable	AMP 502125-7 [b]	Routes Tx and Rx signals from the ST connectors of one 5211 PHY. The FSD connector on the cable plugs into the MIC receptacle of a concentrator.
MIC-to-MIC Cable	AMP 502122-7 [c]	In the subsections describing how to cable the different types of dual attachment stations, this cable connects the optical bypass switch to the data link.
Optical Bypass Switch	Interphase Part # [d] SA0000070	Provides the optical bypassing mechanism described in FDDI specifications.

a. These components are ***representative*** of the types of parts used to cable various types of FDDI stations. Contact your FDDI cable manufacturer for details on selecting the optimum cables and connectors for your installation. All FDDI connectors must be correctly keyed for your specific setup (that is, M-S key for a single attachment station, or A-B for a dual attachment station). In general, you can specify the desired keying when ordering your cables.

b. This part number is for a 15-meter dual-fiber cable with two 2.5mm bayonet ST connectors on one end and a MIC connector on the other. Its cable size is 62.5/125 µm.

c. This part number is for a 15-meter light-duty dual-fiber cable with two FDDI connectors at each end. Its cable size is 62.5/125 µm.

d. This part is an AMP 502443-3 optical bypass switch. The switch has two pairs of ST connectors that mate with the Tx/Rx connectors on a dual-PHY 5211. It also has a 6-pin mini DIN plug on its optical bypass control cable that mates with the 5211's OBS connector.

Considerations for Cabling a Dual Attachment Station

When cabling the 5211 as a dual attachment station, carefully note how Tx/Rx signals are routed to/from the board. For example, assume that you have three dual attachment stations connected to form a standard, counter-rotating FDDI ring. Assuming that no fault has occurred, the flow of Tx and Rx signals between PHYs is as illustrated in Figure 2-4.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

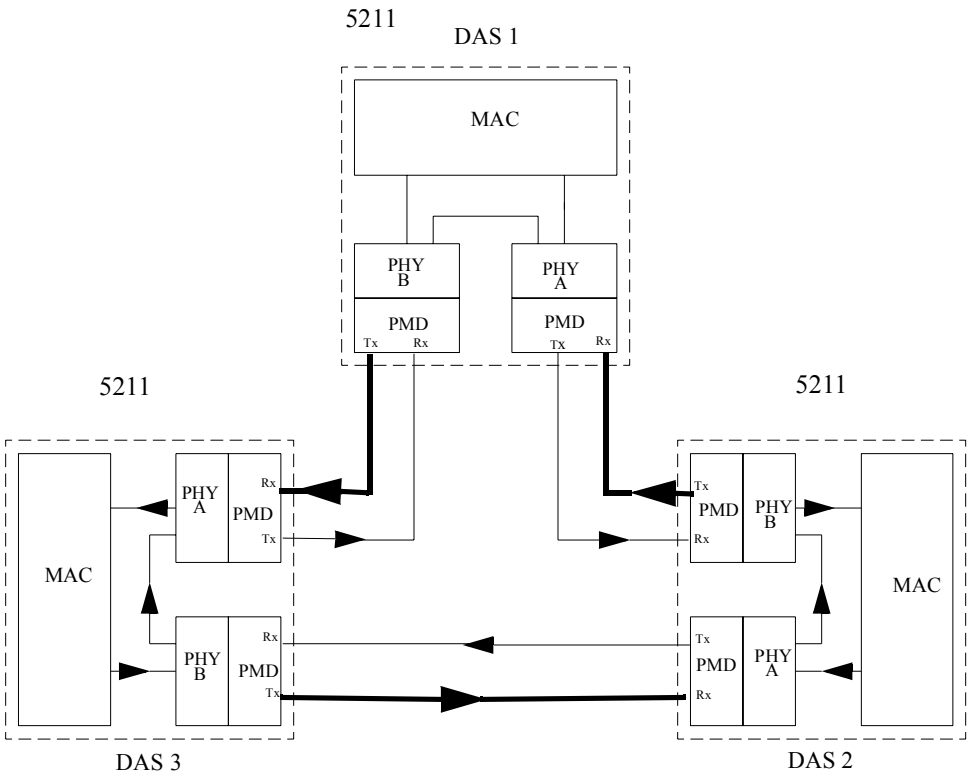


Figure 2-4. Flow of Tx/Rx Signals between PHYs of three 5211 DASs (No Fault in Ring)

In Figure 2-4 outgoing data from one station's PHY A feeds into its downstream neighbor's PHY B as incoming data. Data that comes in through a station's PHY B goes out through its PHY A. This allows the ring to be reconfigured if a fault occurs.

Keying FDDI Connectors

Before cabling the 5211 as a dual attachment station, it is necessary to key some of the FDDI-standard connectors on the cables. For example, the ST-to-MIC cable specified in Table 2-4 is a "universal" FDDI cable. The FDDI end of the cable will mate with *any* MIC receptacle.



NOTE

It is essential for all FDDI connectors to be properly keyed. Otherwise, it is likely that signals will get crossed during installation or when network cables are later detached/ reconnected.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Deciding How to Key the Connectors. For information to key the FDDI connectors of a dual attachment station, refer to the appropriate cabling diagram in this chapter.

The drawings only show keying which is local to the 5211 station. The connectors required by other physical entities in the network will vary depending on your setup.

Identifying and Installing Connector Keys. To determine the keying of an FDDI connector, examine it for an identifying mark or aperture for the key. For example, for an AMP FDDI cable connector, remove the protective cover. Locate the key aperture approximately 1/4 inch from the end of the connector. If the aperture is empty, the connector is unkeyed ("universal"). If a key is installed, it is identified by color (see Table 2-5).

Table 2-5. Keying Scheme on AMP FDDI Connectors

Key Color	Key Type	Function
Red	A	Connect to PHY A
Blue	B	Connect to PHY B
Green	M	Connect to concentrator or single attachment station.

To key an unkeyed AMP FDDI cable connector:

1. Get the desired key. Keys for an AMP FDDI connector are stored in connector's protective cover. To remove a key from the protective cover, pry it out carefully.
2. Press the key into the key aperture. Orient the key so that the T-shaped end fits into the rectangular groove on the connector. The other end of the key is flush with the smooth side of the connector.

To remove a key from an AMP FDDI cable connector, turn the smooth side of the connector face up, place the tip of a small, flat-headed screwdriver on the key, and press firmly.

Preparation for Cabling Procedure

Keep the dust caps on the cable ends, transmitter(s), and receiver(s) until the connections are made. This prevents dirt and oils from soiling any important surfaces. Do not attempt to polish the connectors with a cloth made of synthetic fibers: this charges the fiber and attracts dust.

Connect the fiber optic cable(s) to each system to be connected. Be sure not to stretch, puncture, or crush the cable(s) with staples or heavy equipment or doors, etc. Always maintain the minimum bend radii specified by the cable manufacturer. The fiber in fiber optic cable can suffer damage if the cables are bent into small radii. The minimum bend radii is usually 10-20 times a cable's outer diameter.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Cabling a Single Attachment Station

This subsection describes one method for cabling the 5211 as a single attachment station. It assumes that you are using the sample FDDI cables and connectors listed in Table 2-4. The only sample parts required are:

- ST-to-MIC Cable with M keying
- MIC receptacle with M keying

The cabling procedure is described below. Refer to Figure 2-5 for an illustration of the setup.

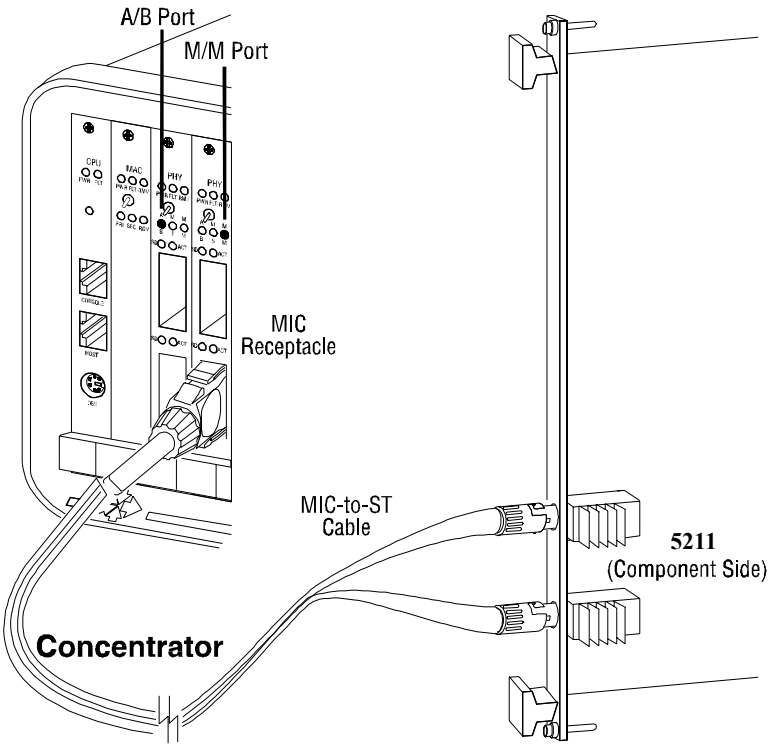


Figure 2-5. Cabling a Single Attachment Station (ST-to-MIC)

1. Connect the ST bayonet end of the ST-to-MIC cable to the transmit and receive connectors on the 5211.

   a. Locate the Tx and Rx connectors on the 5211. The single-PHY version of the 5211 has a single Tx/Rx connector pair. In the board layout on page 14, this is the pair of Tx/Rx connectors located further from the J7 connector (optical bypass control).

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

b. Identify the ST connector to be plugged into the Tx connector and the one to be plugged into the Rx connector. Signals must be routed such that the Tx output of one station is routed to the Rx input of its downstream neighbor.

c. There is no physical difference between the ST connectors on the cable, but they are usually marked in some way to distinguish them. These marks should be used consistently to cable all the stations in your installation. For example, if you decide to use the ST connector marked with a blue ring to route Tx signals from this station, do so for all other stations.

d. Each transmitter and receiver on the 5211 has a gold jack with a slit running from the edge to its base. Key the ST connectors to the appropriate jack and push them on. Turn the connectors clockwise to lock the bayonet mechanism.

2. Connect the FDDI end of the ST-to-MIC cable to a MIC receptacle. This adapter is typically installed inside the concentrator.

3. Recheck your connections. Ensure that each connector is plugged into the correct Tx or Rx receptacle or jack as discussed in Step 1b.

Cabling a Dual Attachment Station

This subsection describes a method for cabling the 5211 as a dual attachment station using an optical bypass switch. (An optical bypass switch is optional in a dual-attached configuration.) It assumes that:

The sample FDDI cables and connectors listed in Table 2-4 are used. Sample parts required are:

- Optical Bypass Switch
- 2 MIC-to-MIC Cables, one with “type A” keying on one end and the other with “type B” keying on one end.[1]

The procedure for cabling a dual attachment station is:

1. Connect the optical bypass switch to the 5211.

a. Locate the 5211’s Tx/Rx connector pairs for PHY A and PHY B. For location information, see Figure 2-6 on page 24.

1. These cables connect the optical bypass switch to the rest of the network. Keying is only specified for end of the cable which connects to the bypass switch, since the connector requirements at the other end can vary.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

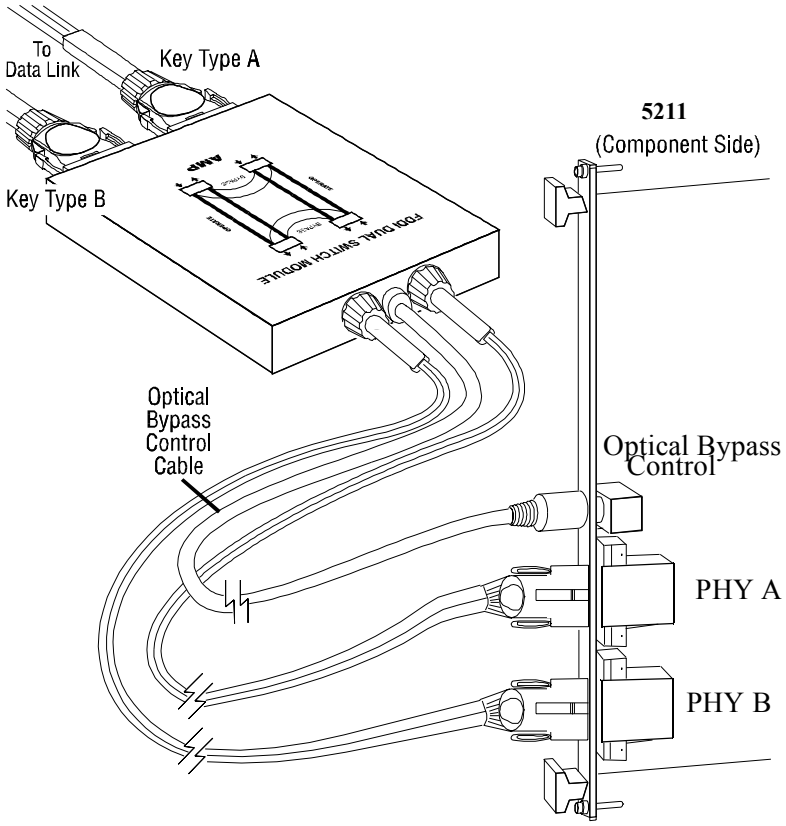


Figure 2-6. Cabling a Dual Attachment Station

b. Plug the connectors on the bypass switch into the appropriate Tx or Rx connectors. Route signals so that the PHY A Tx output for each station feeds to the PHY B Rx input of its upstream neighbor and each station's PHY B Tx output feeds to the PHY A Rx input of its downstream neighbor.

If you are using ST style connectors, identify the connectors by examining the two strand-pairs of ST connectors on the optical bypass switch. They should be marked by the cable manufacturer (for example, with a number or color coding).

Figure 2-7 illustrates typical connections to the 5211 and dual ring:

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

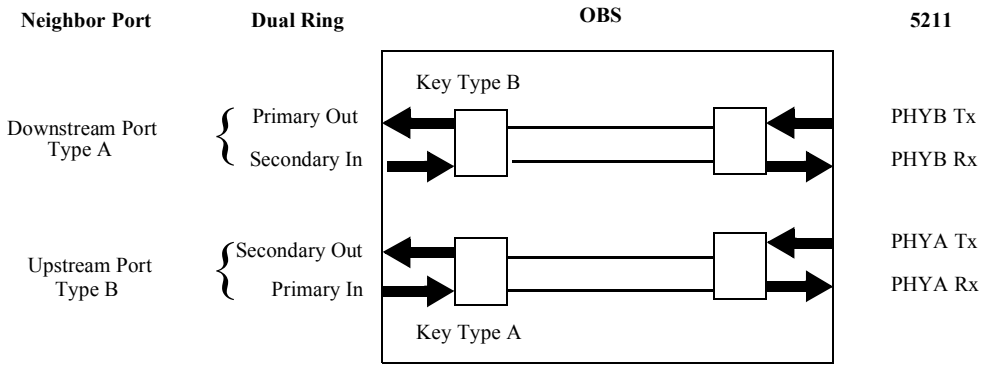


Figure 2-7. Example Optical Bypass Switch

c. Remove the dust caps from the transmitters, receivers and connectors.

d. If you are using ST style connectors, each transmitter and receiver jack is keyed to align the bayonet connectors to the appropriate jack. Align the connectors with the jack, and then turn the connectors clockwise to lock the bayonet mechanism.

e. The bypass switch uses a control cable with a 6-pin mini DIN connector on the end. Plug this connector into the optical bypass control connector on the 5211. To locate the optical bypass switch connector, refer to Figure 2-6 on page 24 or the board layout, Figure 2-1 on page 14.

2. Connect the optical bypass switch to the dual FDDI ring. (For information about OBS pin assignments, see *OPTICAL BYPASS SWITCH PIN ASSIGNMENTS on page 179*.)

a. Locate the two MIC receptacles on the optical bypass switch. These are on the end of the switch opposite from the cables discussed in Step 1. One socket has “type A” keying, and the other has “type B.”

b. Using the MIC-to-MIC cable with “type A” keying on one end, connect the socket with “A type” keying to the data link.

c. Using the MIC-to-MIC cable with “type B” keying on one end, connect the socket with “B type” keying to the data link.

3. Recheck the connections. Ensure that each connector is plugged into the correct Tx or Rx ST receptacle.

POWER SYSTEM ON

Once the 5211 is installed in the chassis and properly cabled, turn the host on. *If your other system documentation (host/peripherals) contains precautionary statements about powering up the system, please follow these precautions*.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

After powering up the system, observe the LED1, which is located on the edge of the board opposite the VMEbus connectors. Once the power-up diagnostics complete successfully, the LED1 will stop toggling and the green light is visible. At this point, the 5211's Common Boot Interface is booted up and ready to accept commands. (See *BOOTUP AND RESET SEQUENCE on page 31*.) For more information on the LEDs, see *LEDs on page 177*.



NOTE

Once the host boots the RC Interface, LED1 status toggles green and red.

If one of the power-up tests fails, the LEDs will start repeating a pattern of on/off signals of varying duration. If the LED1 fails to turn green, the onboard 68EC040 processor has failed. If either of these events occur, contact Interphase Customer Service for assistance.

Alternately, if the LED fails to turn green, the 5211 may not be fully seated in the chassis. Turn off the system and ensure that the 5211 is firmly inserted into the backplane slot. If the LED does not turn green after reapplying power to the system, call Customer Service at Interphase for assistance.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

FUNCTIONAL OVERVIEW 3

OVERVIEW OF THE 5211 INTERFACE

The 5211 firmware interface consists of two major parts used for initializing and operating the controller. These are called the Common Boot (CB) Interface and the Report/Command (RC) Interface.

The Common Boot Interface is a monitor which begins to run when the controller is first powered up or reset. A general-purpose protocol, it provides a deterministic start-up sequence for resident or downloadable host/board interfaces. It functions as a distinct entity in the controller's firmware architecture.

The Common Boot interface provides a mechanism to boot, initialize and run controller diagnostics on the 5211. It is also used to download and execute optional firmware images and for permanently storing images in "Flash Memory." The host uses the Common Boot interface to submit the characteristics of the initial RC channel pair needed to boot the RC Interface on the 5211.

The host issues commands to the 5211 via the intelligent Report/Command (RC) Interface. This interface is implemented in the 512-byte *Short I/O* space on the 5211. (The term "short I/O" refers to a block of on-board memory configured to operate in the VMEbus Short Supervisory I/O address space.) This memory is shared across the bus between the host CPU and the 5211 firmware. Figure 3-1 depicts how short I/O and the RC channel descriptors can be viewed as part of the Communications Buffer.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

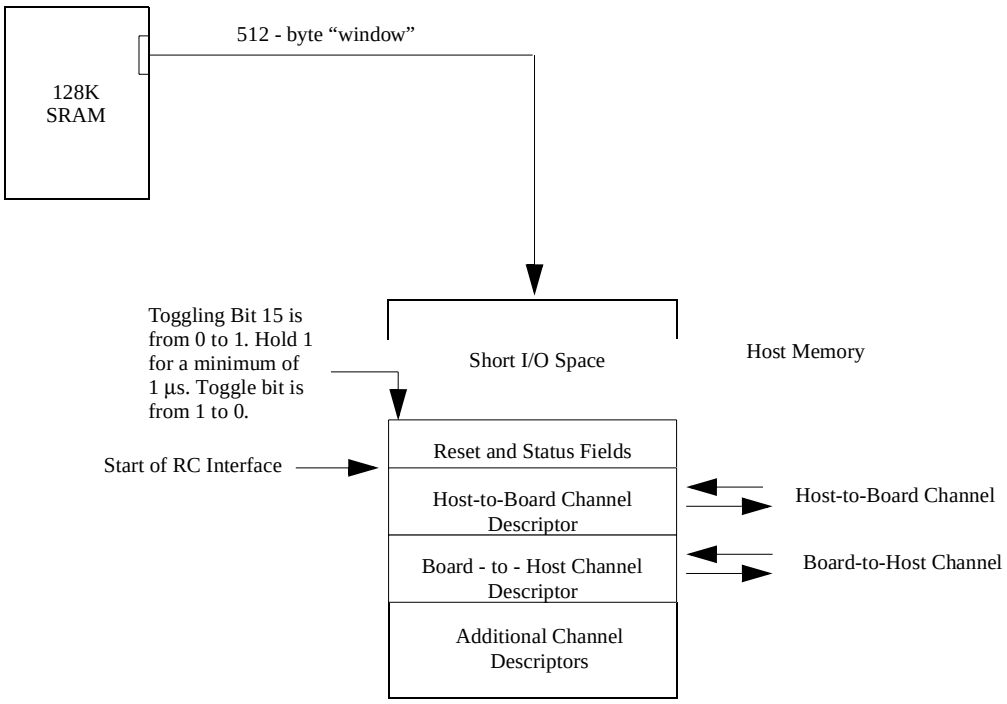


Figure 3-1. Short I/O as a Window into the 5211 Communications Buffer Address Space

SYSTEM REQUIREMENTS

To interface with the 5211, the host must meet following requirements:

- Although the 5211 supports 32-bit access to short I/O, some systems will only perform 16-bit transfers. If the system breaks short I/O into 16-bit transfers, the following statement must be observed.
- To transfer a 32-bit quantity to/from short I/O, the host must place the MSW (Most Significant Word) in the low-order address and the LSW (Least Significant Word) in the high address.
- The host must have at least 16-bit access into the controller’s on-board shared memory (“short I/O”) space. In addition, it must be able to read or write a 16-bit quantity in one operation without leaving an access window between bytes. Unlike the 4211, the 5211 will allow the host to transfer a 32-bit quantity in one operation. If the host supports this type of transfer without leaving access windows between 16-bit words, the host may simplify the way pointers are updated. These issues are discussed later in this chapter.
- Host-to-Board and Board-to-Host segments must be longword-aligned.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

- At least 512 bytes of VMEbus short I/O address space must be available on the system VMEbus for use by the RC Interface.
- The byte ordering of commands must be “big-endian” (that is, Most Significant Byte first).

This section details the capabilities and limitations of the 5211 firmware. Later sections describe specific aspects of the firmware when they are relevant to the topic being covered.

USING THE COMMON BOOT INTERFACE

The host uses Common Boot to submit the characteristics of the initial RC channel pair needed to boot the RC Interface on the 5211. This initial channel pair can be used to create additional channels after RC is running. Once RC boots successfully, the Common Boot interface exits until the board is reset. The host can also perform a variety of other tasks when in Common Boot mode. These include:

- performing extended diagnostics
- downloading code for on-board execution.

  This feature is for special applications only. It is not needed to boot the native RC Interface stored in controller firmware.

This section describes how to use the Common Boot interface to get the RC Interface up and running on the controller. It also details various commands that you can use when booting the controller.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

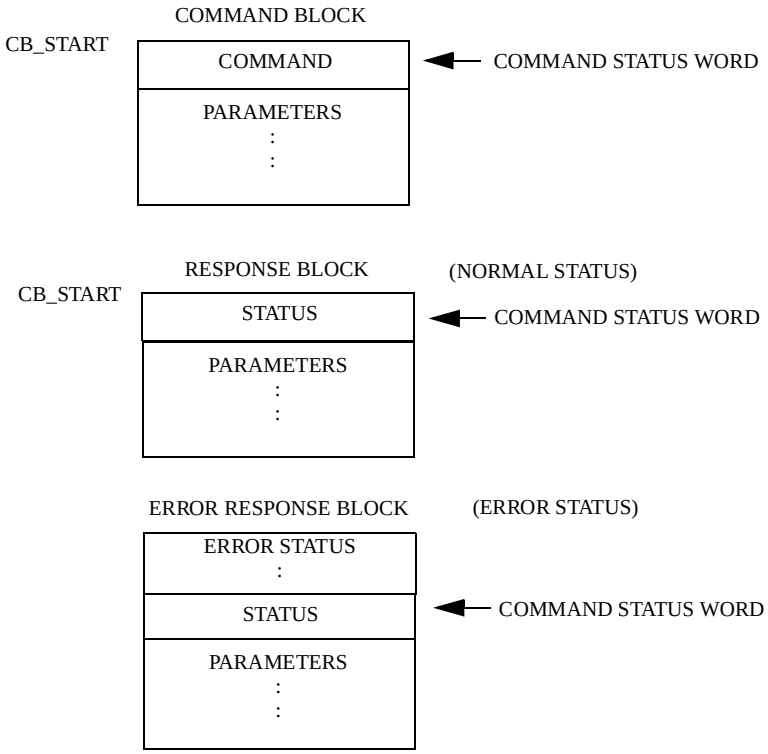


Figure 3-2. Common Boot Command/Response Format

Commands are issued by writing over the Command Status Word, “CBOK,” with a Common Boot command. If the supplied command has parameters, then these parameters must be written into shared memory ***before*** the command code is entered. The Common Boot interface acquires a new command when CBOK changes. After command input, the host must poll the Command Status Word for “CBOK” or “FAIL.” When the board detects the command Status Word has changed, it will compare the value read with a list of acceptable commands. If the issued command is valid, CB will record the command in the ERROR STATUS BLOCK (See Figure 3-2), then execute it. Commands execute until completion, and no timeout facility is provided.

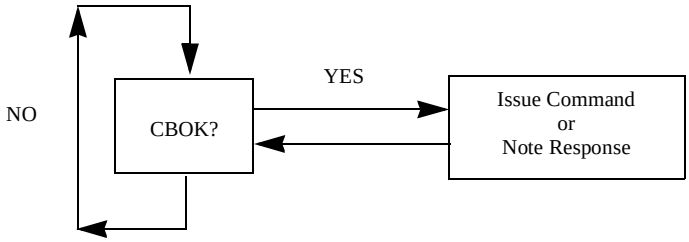


Figure 3-3. Polling CBOK for Commands or Responses

Unless you need to perform diagnostics, or download code, the Common Boot may simply be used to boot the RC Interface. With this scenario, the following occurs when the controller is powered up or reset:

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

- The Common Boot starts up and fills shared memory with CB_HERALD.
- The host issues the BOOT command to boot the RC Interface.

The BOOT command allows you to create the initial host-to-board channel and board-to-host channel. These structures enable the host to interact with the controller using the RC Interface. Additional channels may be established once RC has booted up.

BOOTUP AND RESET SEQUENCE

The following sequence of events occurs each time the 5211 is powered up or reset.

1. The 5211 executes its bootstrap code and performs a series of power-up diagnostics. Once the 5211 Common Boot interface is active, LED1 turns green.
2. The Common Boot Interface starts up and fills the 512-byte *short I/O* space with a 32-bit value called CB_HERALD. The format of CB_HERALD is shown below. Next, the Common Boot Interface fills the entire *shared memory* region with a 32-bit pattern known as the CB_HERALD. CB_HERALD contains information about the start of the Common Boot Interface and the total size of the controller's short I/O space (i.e. shared memory region). The format of CB_HERALD is shown in Figure 3-4.

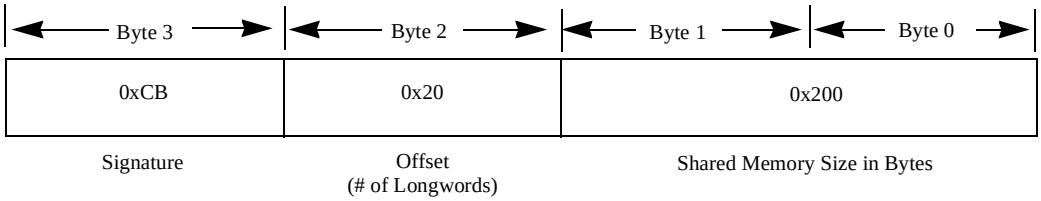


Figure 3-4. CB_HERALD Format

- Bytes 1 and 0 contain the size of 5211's 512 byte shared memory space.
- Byte 2 specifies the offset (in longwords) at which the Common Boot interface starts in shared memory. The first longword of the Common Boot command area is referred to as the **Command Status Word**.
- Byte 3 contains the Common Boot signature, which is a hex CB (0xCB) or 203.

If the power-up diagnostics completed successfully, **CBOK** will be written to the Command Status Word. CBOK is defined to be a 32-bit ASCII "CBOK," or 0x43424F4B.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Figure 3-5 depicts the controller's shared memory space after the controller initializes successfully.

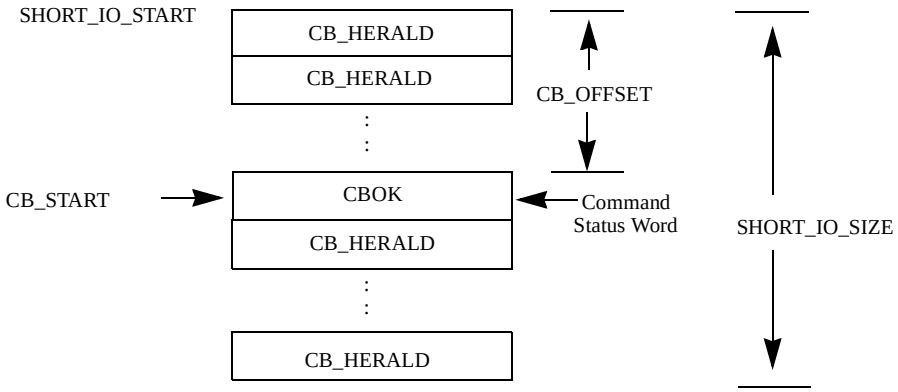


Figure 3-5. Format of Shared Memory after Controller Initialization

3. After the 5211 power up is complete, the firmware will call the "INFO" command to display information about the 5211 board. This information may be used to distinguish the 5211 board from the 4211 board. See *INFO on page 97* for more information.

   If any power-up test fails, the controller writes the ASCII value **FAIL** (0x4641494C) to the Command Status Word, followed by a set of fields describing the failure. Figure 3-6 depicts the format of shared memory when FAIL occurs during power-up diagnostics.

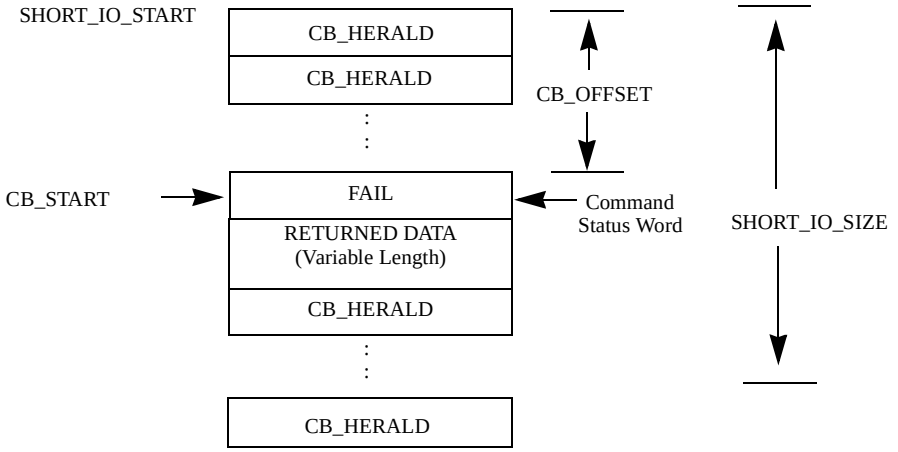


Figure 3-6. Format of Shared Memory after FAIL

4. The host reads Command Status Word. It should contain one of the following 32-bit ASCII values: "CBOK" (0x43424F4B) or "FAIL" (0x4641494C). If it contains CBOK, the host may issue any Common Boot command. If it contains FAIL, the host may issue the DIAG command to identify the problem.

5. The board polls the Command Status Word. When it changes to a command code, the board executes the command and writes the results (CBOK or FAIL) back to the Command Status Word. It polls for the next command.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

No interrupt is generated at the completion of a Common Boot command, so the host polls the Command Status Word to determine whether the command has completed.

6. To boot the RC Interface, the host issues the Common Boot command “BOOT”. Once the RC Interface boots up, LED1 alternately toggles red and green. The LED will continue to toggle as long as RC is running normally.

7. The RC Interface starts up and creates the initial host-to-board and board-to-host channel. Common Boot exits until the board is reset.

RESETTING THE 5211

To reset the 5211, toggle bit 15 of the Hardware Control field from 0 to 1 to 0. The Hardware Control field is a 16-bit field located at the start of short I/O (that is, bytes 0 and 1 of the 512-byte short I/O space). When the 5211 is reset, it goes through its bootup sequence as described in the previous section. The reset bit must hold the “1” for a minimum of 1 µsec.



NOTE

After toggling the *reset bit* the host should wait at least 500 ms before looking for CB_HERALD. Or the user may write a value in the command/status location (normally zero or 0xDEAD) and reset the controller. The zero or 0xDEAD value will be replaced by CBOK or FAIL.

5211 DIAGNOSTICS

For complete details of the various diagnostic tests see Appendix A.

The 5211 supports a variety of power-up and host-controlled diagnostics. Power-up diagnostics execute automatically each time the board is reset. Host-controlled diagnostics may be invoked individually while the Common Boot interface is running (i.e., before the host has booted the RC Interface using the BOOT command).

The 5211 diagnostics provide the following controller tests.

- Major Test 0x1 - Memory Tests
- Major Test 0x2 - Front End FDDI Chipset Tests
- Major Test 0x3 -DMA Diagnostic Tests
- Major Test 0x7 - CAM (Content Addressable Memory) Tests

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

DIAG COMMAND

Most of the commands in the Common Boot command set are generic. Their definition is the same for any controller that supports the Common Boot interface. The DIAG command, however, is an exception. It is used to perform a variety of controller-specific tests. Therefore, the definition of its fields varies depending on the controller and on the type of test being performed. Figure 3-7 shows the format of the DIAG command.

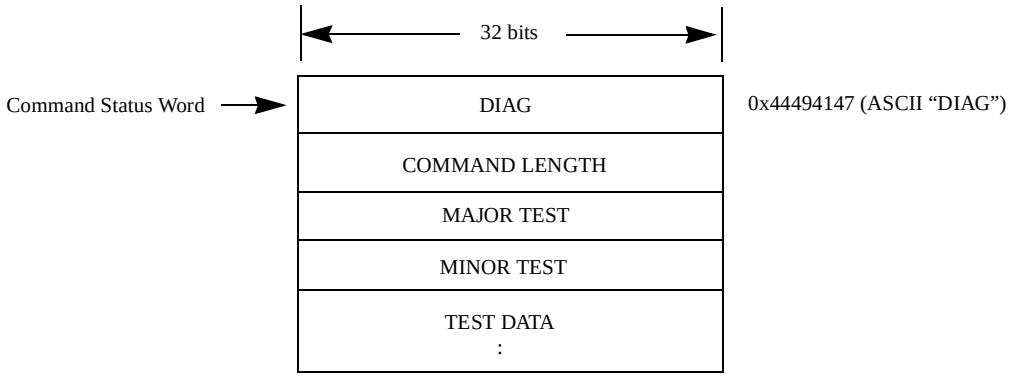


Figure 3-7. Format of the DIAG command

COMMAND FIELDS

The meaning of the fields in the DIAG command is as follows:

COMMAND LENGTH

This field contains the total number of bytes in the test command, including itself, but excluding the Command Status Word (which contains the ASCII value “DIAG”).

MAJOR TEST

Each major hardware section of the 5211 has a separate test which is assigned a number. See “DIAG” command in CB Commands for test numbers. A major test number of 0x0 will execute all the major tests.

MINOR TEST

Some of the major tests are divided into a number of minor tests. A minor test number of 0x0 will execute all the minor tests in the associated major test.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com



NOTES

If the Major Test field contains 0x0, the Minor Test field is ignored.

If no minor tests are defined for the major test, the host may enter either 0x0 or 0x1 in the minor test field. If the 5211 returns a status block the returned Minor Test field will contain the value 0x1.

TEST DATA

Only VMEbus DMA tests require test-specific data. These fields are provided in the VMEbus DMA test description.

Table 3-1. Major Test Numbers

Code	Test Type
0x0	Execute ALL major test (1 through 7)
0x1	Memory Test
0x2	Front End FDDI Test
0x3	VME DMA Test
0x4	Not Used †
0x5	Not Used †
0x6	Not Used †
0x7	CAM Test

† For compatibility with the 4211, these tests return a passing status, although no test is actually executed.

Details on each major test and its minor test(s) are provided in Appendix A.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

TEST RESULTS

If the test completes successfully, the Command Status Word will contain the ASCII value “CBOK” (0x43424F4B). No other data is returned. If the test fails, the 5211 returns the structure shown in Figure 3-8.

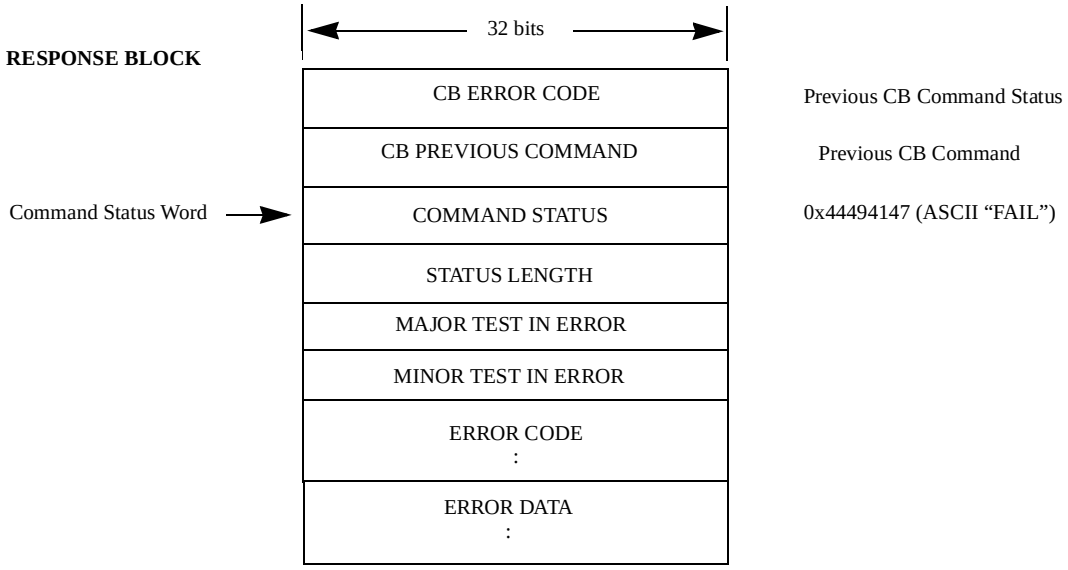


Figure 3-8. “FAIL” Returned Status Block

The meaning of the fields in the resulting structure shown in Figure 3-8 is as follows:

- CB Error Code

  This field reports the status of the previously issued CB command.

- CB Previous Command

  This reports the previously issued CB command which failed.

- Status Length

  This field contains the total number of bytes in the status block, including itself, but excluding the Command Status Word (which contains the ASCII “FAIL”).

- Major Test in Error

  This reports the major test number of the failed test.

- Minor Test in Error

  This reports the minor test number of the failed test, if any. If there is no minor test, this field contains 0x1.

- Error Code

  This field reports the test error code. Table A-17 on page 200 lists the diagnostic error codes.

- Error Data

  This field contains test-specific information about the error. Some errors have data associated with them, while others do not. Consult the individual test description for details.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

If a given error does not include error data, the returned structure will end after the Error Code field. It will consist of four 32-bit fields (Status Length, Major Test in Error, Minor Test in Error, and Error Code).

DIAGNOSTIC ERROR CODES

Table 3-2 lists the diagnostic Error Codes.

Table 3-2. Diagnostic Error Codes

Value	Description
0x0	PASS
0x1	Data miscompare
0x2	VME bus time out reported by the addgen
0x3	VME buserr reported by the addgen
0x4	Addgen interrupt time-out, no done, error, or tick interrupt occurred
0x5	Premature interrupts, interrupts present before the test started
0x6	The addgen tick timer elapsed before the DMA done or error interrupts
0x7	Addgen status register lacked sanity
0x8	Addgen spurious interrupt, and interrupt occurred when not expected
0x9	VIRQ time-out, the VIRQ inport bit was never seen
0xA	VIRQ IACK time-out, the IACK inport bit was never seen
0xB	Premature VIRQ interrupt, the VIRQ inport bit was present before the test was started
0xFE	Bad command parameter supplied
0xFF	Illegal test code

USING THE REPORT/COMMAND INTERFACE

The Report/Command (RC) Interface is an asynchronous mechanism for queuing commands and responses between the host and controller. It is designed to minimize the overhead involved in issuing commands to and reading responses from the controller.

Most of the interaction between the host and controller takes place via **channels**. An RC channel is a data pathway through which message traffic can be moved from one subsystem to another. It is not a piece of hardware. Rather, it is a set of techniques or methods whereby host-based software and controller firmware can make efficient use of hardware devices for the purpose of communicating with each other.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com



NOTE

The 5211 expects queued commands as well as transmit/receive data to reside in host memory.

RC-Specific Terminology

The term “channel” has a specific definition in the RC Interface. Other terms with an RC-specific meaning include: segment, channel descriptor, read and write pointer, and the four RC command types (request, directive, response, and indication).

Available Commands

The 5211 FDDI command set consists of “generic” RC commands plus commands specific to the 5211 board. The generic RC commands are those that are also common to other Interphase controllers.

Table 3-3 shows a listing of the generic RC Commands.

Table 3-3. Generic RC Commands

Set DMA Burst	Allocate Board-to-Host Segment Memory
Set Bus Timeout	Report Printf Call
Request Statistics	Error Message
Set Interrupts	Report Statistics
Create Host-to-Board Channel	Report New Host-to-Board Channel
Create Board-to-Host Channel	Report new Board-to-Host Channel
Link Host-to-Board Segment	Report Link to Next Segment

For a complete description of each command see *5211 COMMAND SET on page 69*.

Table 3-4 shows the available 5211-Specific Commands.

Table 3-4. 5211-Specific Commands

Transmit Raw Data	Transmit Datagram
Set up Receive Buffers	Request Station Address
Set Station Address	Ring Control
Perform Maintenance	Get MIB Attribute
Set MIB Attribute	Set CAM Address
Clear CAM Address	Flush CAM Address
SMT Trace	Set Hardware Parameters

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table 3-4. 5211-Specific Commands (cont)

Get Hardware Parameters	Get CAM Address
Rx Frame	Report Station Address

For a complete description of each command see *5211 COMMAND SET on page 69*.

SHORT I/O AND THE RC INTERFACE

Short I/O memory is initially used for the purposes of booting the controller and starting up the RC Interface. Once the RC Interface is running, short I/O will hold *channel descriptors*. These descriptors point to areas of host memory containing RC commands and responses, which point to transmit/receive data elsewhere in memory. They act as control elements of a low-overhead handshake mechanism for queuing commands asynchronously between the controller and the host.

Channels are data structures used for host-to-board or board-to-host communication. Commands are written to the board through a *host-to-board channel*. If an issued command needs to report command completion status to the host, it does so through a *board-to-host channel*. Channels are discussed below.

For a detailed memory map of the 5211's short I/O space, see Table 3-11.

DEFINITION OF RC CHANNELS

In the context of the RC Interface, a channel consists of one or more blocks of memory used for host-to-board or board-to-host communication. Commands are written to the board through a structure referred to as a **host-to-board channel**. If the controller needs to report command completion or other statuses to the host, it does so through another structure called a **board-to-host channel**.

The host can create multiple host-to-board and board-to-host channels. It determines any associations between these channels. For example, the host can create a board-to-host channel and instruct the controller to use it only when responding to commands issued through a specific host-to-board channel. Or, if desired, multiple host-to-board channels can issue commands to which the controller responds via a single board-to-host channel.

GENERAL FORMAT OF RC CHANNELS

A channel is composed of two or more **segments**, where a segment is defined to be a block of contiguous memory up to 65,535 (64K - 1) bytes in length. These segments may be discontiguous with each other. A link command placed at or near the end of each segment provides information which links segments together.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The linking of the segments is done by the owner (**writer**) of the channel. The host owns host-to-board channels, and the controller owns board-to-host channels. The non-owner (**reader**) of the channel follows the command list through a segment until it encounters the link command. It then spans to the next segment in the channel.

An RC channel may be thought of as being infinite in size, because the segments may be linked such that there is no return to the starting point.

Theoretically, the simplest case of an RC channel would be a single segment which links to itself. In practice, however, this setup is impractical due to the overhead associated with bounds checking. Thus, the minimum configuration for a channel is two segments which link to each other.

The RC Interface requires at least two channels in order to function – one for host-to-board communication and the other for the board-to-host path. As discussed later in the chapter, the host establishes these two initial channels when it initializes the RC Interface. Once RC boots, the host may create additional channels.

The maximum number of channels is determined by how much on-board shared memory space can be dedicated to holding “channel descriptors.” These structures are described in the next section.

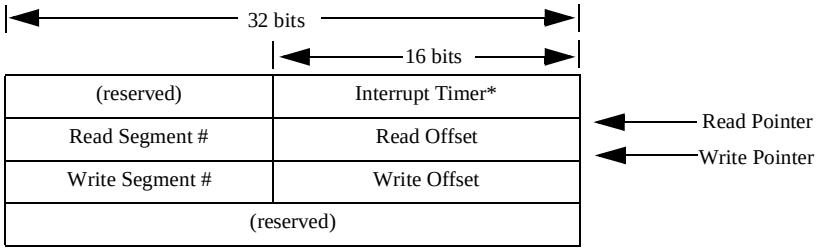
CHANNEL DESCRIPTORS

For each RC channel the host creates, the controller will write a **channel descriptor** into its shared memory space. The channel descriptors begin at an offset of +16 bytes from the start of shared memory. The first and second channel descriptors belong to the host-to-board and board-to-host channels, respectively, specified by the host when it boots RC.

A channel descriptor is a 16-byte control structure containing the information needed to read/write a specific channel. It resides in shared memory at a location assigned by the controller when it executes the command to create the channel.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Figure 3-9 shows the format of a channel descriptor.



* The Interrupt Timer has no meaning in host-to-board channel descriptors. For a description of this field, see the section on interrupts at the end of this chapter.

The host should **not** write to the reserved fields in a channel descriptor.

Figure 3-9. Channel Descriptor Format

Each channel descriptor contains a read and write pointer. The owner of the channel uses the **write pointer** to keep track of its place in the channel's segments. The non-owner uses the **read pointer** to mark its location in the channel. Thus, the owner must update the write pointer, while the non-owner updates the read pointer.

For example, when the host issues one or more commands, it updates the write pointer in the channel descriptor of the host-to-board channel that will pass the command(s) to the controller. The controller updates the read pointer once it has extracted the command(s) from the channel and queued them for execution.

Read and write pointers are composed of two 16-bit quantities — a segment number and an offset. The segment number is used to keep track of which segment is currently being accessed. The offset points to the next memory location in the segment that will be written/read.

The offset field of a pointer is set to zero at the beginning of each segment. As the owner and non-owner work their way through the segment (one providing information and the other consuming information), each updates its offset to point past the data it has just processed. When each encounters the end of the segment (set using the Link Segment directive and indication), it links to the new segment.

UPDATING READ AND WRITE POINTERS

As discussed previously, a channel descriptor contains both a read and write pointer. These pointers in turn consist of a segment number and offset which identify where the next read or write operation is to take place.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

This section discusses the procedure for updating the read and write pointers. It focuses on what the host needs to do to keep track of where it is in a channel. If the driver software properly creates and maintains the channels, the controller will update its read and write pointers automatically.

Reading a Pointer

The 5211 controller reads and writes segment and offset fields in a single 32-bit operation. If the host can read the segment and the offset field in a single 32-bit operation than this assures that valid pointer values are being read. However, if the host reads the segment and offset field in two 16-bit operations, the following steps must be performed to ensure that valid pointer values are read:

1. Read the segment number.
2. Read the offset.
3. Read the segment number again.
4. If the segment numbers read in step 1 and step 3 have changed, go to step 1.
5. If the segment numbers are identical, record the valid pointer.

An algorithm depicting a read of the pointer is shown in Figure 3-10.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

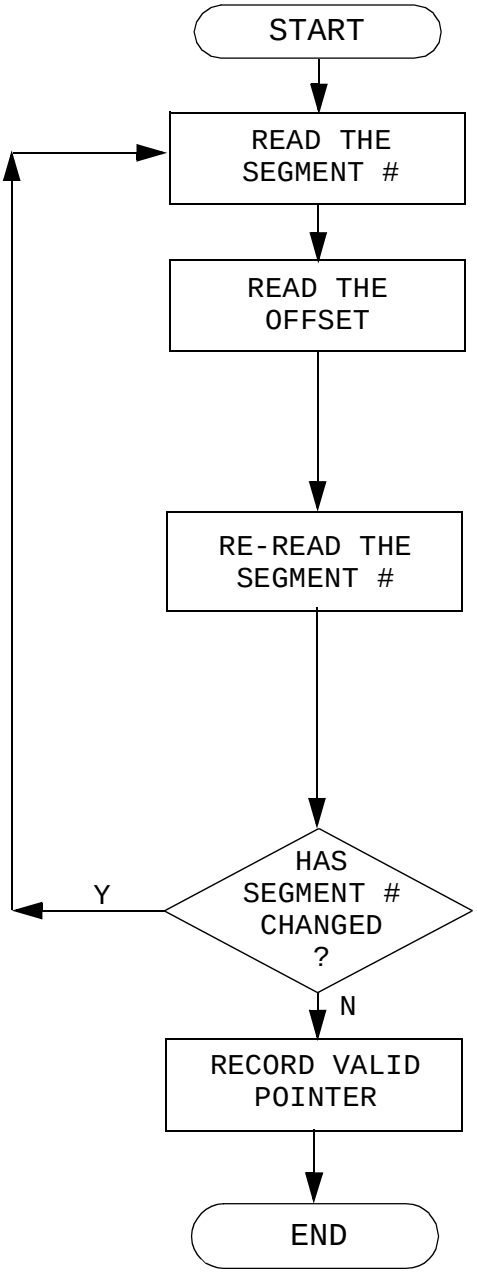


Figure 3-10. Reading a Pointer

WRITING A POINTER

If a host can update the segment and the offset field in a single 32-bit operation no further handshaking or protocol is required. However, if the host updates the segment and offset field as two separate 16-bit operations, the following procedure must be followed:

1. Write the value 0xFFFF into the offset field.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

2. Increment the segment number field by 1.
3. Write a valid offset (0 or some other positive integer less then 0xFFFF) into the offset field.



NOTE

The method for reading and writing the channel pointers used for the 4211 will also work with the 5211. This ensures that the drivers written for the 4211 will also work with the 5211.

MANAGING SEGMENT MEMORY

Segments typically reside in host memory. The host must set up two segment "pools" – one for host-to-board channels and the other for board-to-host channels. The maximum size of a segment is 65,535 bytes. The maximum legal offset is 0xFFFE. (Note: As discussed previously in this chapter, 0xFFFF cannot be used as an offset since it indicates that the segment number is being updated.) The host may choose to match segment sizes to system page sizes for efficient memory utilization.

A segment should not link back to itself. Therefore, the host must allow sufficient memory to support ***at least two*** segments for ***each*** channel to be created. The controller will enforce this rule for the board-to-host segments (but not the host-to-board segments). It does this when it processes requests to create new board-to-host channels. If there is insufficient segment memory available to the controller when it encounters such a request, it will respond with an error message.

Segment Ownership

Segments are owned and managed by the "writer" of the channel. This means that the host is responsible for tracking and reusing the host-to-board segments. Likewise, the board-to-host segments are tracked and reused only by the controller.

Host-Owned Segment Memory

The host, as it writes commands to a host-to-board channel and fills up segments, links to new or previously allocated segments. Allocation of these segments is completely under the control of the host. For example, different-sized segments can be used in the same channel.

The only restrictions on host-owned segment memory are as follows:

- All segments must begin on a longword boundary.
- The maximum segment size is 65,535.
- All segments must be I/O mapped in a manner that permits the controller to perform DMA operations on them.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Controller-Owned Segment Memory

The host must dedicate memory for sole use by the controller as board-to-host segments. The controller independently manages its segments, reusing them when the host has extracted the responses they contain.

To inform the controller about what memory is assigned to it, the host issues an Allocate Board-to-Host Segment Memory directive. This is done at RC boot time, so that the board-to-host segment memory is in place after RC initialization. The directive may also be reissued after booting RC, if necessary. For more information, see Initializing the RC Interface, page 47.

Unlike segments in host-to-board channels, all board-to-host channel segments are same-sized. The host sets this size when it allocates memory to the controller.

Assigning Segment Numbers

The RC Interface has some important rules concerning the “Segment #” fields in the read and write pointer. These rules are:

- Segment numbers are assigned on a per-channel basis (that is, the segment numbers associated with one channel are not affected by those of another).
- Segment numbers start at 0 and are incremented by 1 until they exceed 0xFFFF. At that point, they roll over to 0 and start incrementing again.
- When the writer of the channel links to a new segment, it increments the “Write Segment #” field. When the reader of the channel links to the segment, it increments the “Read Segment #” field.

It is quite possible for the writer of a channel to be several segments ahead of the reader. Segment numbers provide a way for the writer to determine where the reader is in the channel. They thus enable the writer to reclaim segments after the reader has finished extracting information from them (see *Reusing Segments on page 46*).

The writer of a channel “owns” the segments of that channel. It must therefore keep track of what segment number (if any) is currently associated with a given segment in physical memory.

The reader, on the other hand, cannot use segment numbers to find a segment in memory. It only uses segment numbers to determine whether the writer is still pointing to the same segment which it (the reader) is accessing. This information affects how the reader will process any remaining information in the segment, as will be discussed later in the chapter.

Location of Segments

When looking for a segment in host memory, the reader of a channel is faced with one of three scenarios:

1. **At RC bootup time**, it must find the first segment in the initial channel that it is supposed to read.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

When the host boots the RC Interface, it creates an initial host-to-board and board-to-host channel. The location of the first segment in these initial channels is as follows:

The first segment in the initial host-to-board channel is specified in the command used to create the channel.

The first segment in the initial board-to-host channel is, by convention, the first item in the list of memory resources allocated to the controller by the host.

See “Initializing the RC Interface” for more information.

2. **When accessing a newly created channel**, the reader must be able to find the first segment in the channel.

   This depends on whether the channel is a host-to-board or board-to-host channel. For host-to-board channels, the location of the first segment is specified in the channel creation command. For board-to-host channels, is returned by the controller in response to the channel creation command.

3. **When linking to a new segment in a channel**, the reader must be able to find the next segment.

   Each time the writer of the channel links to a new segment, it places a link command at the end of the old segment. The link command contains the location of the next segment in the channel.

Reusing Segments

Once the reader of a channel has linked from one segment to the next, the writer can reuse the old segment to pass new information to the reader. This section describes what the host needs to do to reuse its segments. The controller follows a similar procedure without host intervention.

Before the host can reuse a segment, it must make sure that the controller has extracted all the commands from the segment. The controller indicates that it is finished with a segment when it updates the “Read Segment #” field in the channel descriptor. It updates this field after executing the Link Host-to-Board Segment directive placed by the host at the end of the segment.

The procedure for identifying such freed-up segments is relatively straightforward, provided that you adhere to the convention that segment numbers always increment. One method is for the host to keep a table of the segment numbers currently associated with memory being used to pass commands to the controller. Data for the table would extracted from the host-to-board channel descriptors in shared memory. For each host-to-board channel, the host can assume that a segment is reusable if its segment number is less then the current value of the channel's “Read Segment #” field.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Rollover of Segment Numbers to 0

Segment numbers are 16-bit quantities and eventually roll over to 0. In general, the reader of the channel need not concern itself with this event. It simply increments the “Read Segment #” field as necessary.

On the other hand, the writer of a channel needs to know when the “Read Segment #” and “Write Segment #” fields roll over in the channel descriptor. Otherwise, after the “Write Segment #” rolls over, there will be an unnecessary delay before some of the segments in the channel can be reused. This is because all segments which have not been processed when the Write Segment # rolls over will be associated with large segment numbers.

Since the controller manages its own segments, the host only needs to check for segment number rollover in host-to-board segments. One method is to allocate an array of segments whose total quantity divides evenly into 0x10000, and then use this array as a circular queue. To find the array index of the segment currently being read, apply the current segment number in the read pointer as a modulus against the total number of segments. This procedure would be repeated each time you link to another segment in the array. If the indices were not equal, the candidate link segment in the array would be assumed to be free.

INITIALIZING THE RC INTERFACE

The RC Interface is initialized via the Common Boot command BOOT. This command informs RC of the host virtual address of shared memory, thus enabling the controller to convert host-referenced addresses. BOOT requires the following three RC commands as parameters:

1. Allocate Board-to-Host Segment Memory
2. Create Board-to-Host Channel
3. Create Host-to-Board Channel

The first command provides the controller with a list of segments for use in board-to-host channels. The latter two commands submit the characteristics of two RC channels – a host-to-board and a board-to-host channel. This initial pair of channels can then be used to create additional channels as necessary. Both channels must be in place when RC boots up, because the host must have a path to receive a response from any additional channel creation commands it may issue.

There is an important convention to note concerning the BOOT command. When the controller executes a Create Board-to-Host Channel directive, it normally returns to the host the logical address of the first segment for the new channel. This is not possible at boot time since the channel for response is not yet established. Therefore, after RC boots up, the controller examines the list of segments provided in the Allocate Board-to-Host Segment Memory directive. It then uses the first segment in the list as the **initial segment** of the **first board-to-host channel**.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

STRUCTURE OF RC INTERFACE IN SHARED MEMORY

RC channel descriptors are located in the controller's shared memory space, starting at an offset of +16 bytes (0x10). The first word in shared memory is typically used for board reset purposes. Refer to "Resetting the 5211" for details. The RC control structure is aligned on 32-bit longwords. Figure 3-11 shows how this space is organized:

MANAGING COMMANDS AND RESPONSES

Once the RC Interface is booted, the host can issue commands to the 5211 through the initial host-to-board channel and read 5211-generated messages from the initial board-to-host channel.

Available RC Commands

The 5211 FDDI command set consists of RC commands ***specific*** to the 5211 board. Commands specific to the 5211 are described in the next chapter.

In addition to the FDDI command set, the 5211 supports all members of the generic RC command set. The generic RC command set is documented in the following section.

Before performing FDDI-related tasks using the 5211, it is necessary to have a good understanding of the RC Interface. Concepts such as "segment linking" and "working one's way through a channel as the writer (or reader) of a channel" are integral to writing a driver for the 5211.

Location of Commands and Tx/Rx Data

The 5211 expects queued commands as well as transmit/receive data to reside in host memory.

COMMANDS AND RESPONSES

Once RC has booted up, the host can issue commands to the controller through the initial host-to-board channel and read controller-generated data from the board-to-host channel. See Chapter 4 for a description of the commands in the generic RC command set.

Managing Controller Commands

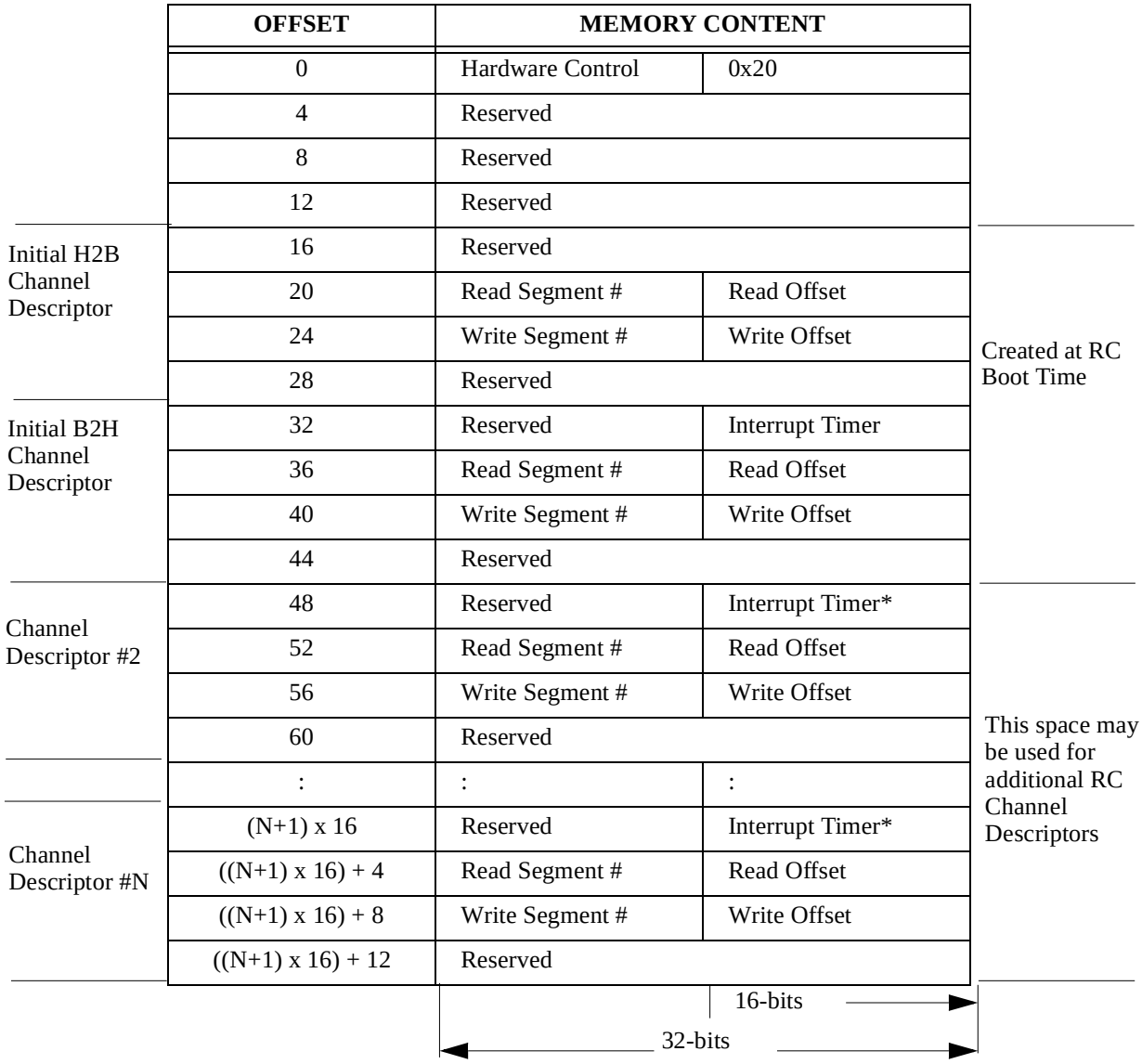
To issue one or more commands to a host-to-board channel, the host should:

1. **Examine the appropriate channel descriptor**

   This refers to the channel descriptor of the host-to-board channel that will be used to issue the command. Two fields in the channel descriptor are of particular interest: 1) the Write Segment # (which identifies the segment that is currently being used by this channel to pass commands to the board), and 2) the Write Offset

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Figure 3-11. Control Structure of RC Interface



NOTES: [1] H2B = Host-to-Board
B2H = Board-to-Host

[2] * The Interrupt Timer field is not used in host-to-board channel descriptors.

[3] The host should NOT write to the reserved fields in a channel descriptor.

[4] The “Hardware Control” field contains the bit to reset the 5211.

[5] The 16-bit field located at the base of short I/O +2 bytes is hard-coded to 0x20 to maintain backward compatibility with previous RC firmware releases.

(which specifies the offset into the segment at which the next command should be written).

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

2. **Make sure the current segment has enough available memory for the command(s)**

   A command cannot be split between two segments. Moreover, there must always be enough space in the segment to issue the Link Host-to-Board Segment directive. Otherwise, the controller will not be able to find the next segment in the channel.

   To determine whether the command(s) fit into this segment, compute the following values:

   Required Space=[Total length of command(s) in bytes] + 20 bytes (the length of the Link-to-Board Segment directive.

   Available Space=[Total length of segment in bytes] – [Current value of Write Offset field]

   The command(s) can be written to this segment if the Available Space exceeds the Required Space. If not, the host must determine how many of the commands can fit into the current segment, including the 20-byte link directive. The "excess" commands should be written to the next segment.

3. **Issue the command(s)**

   Write the command(s) into the segment, starting at the offset indicated by the write pointer. Link to a new segment if necessary.



NOTE

If linking to a new segment, update the Write Pointer *before* issuing the link directive. Refer to the next step and also to the section "Linking from One Segment to the Next."

4. **Update the Write Pointer**

   After the commands are written to the segments. Channel write pointer can be updated. See the discussion in the section on WRITING A POINTER for the proper procedure to update the write pointer.

In addition to queuing up commands in a single channel, the host can also issue multiple commands to the board at the same time. This is done by writing each command to a different host-to-board channel, and then updating write pointers in the corresponding channel descriptors.

The controller polls the host-to-board channel descriptors starting at the low end of shared memory. When the board detects that the read pointer in a host-to-board channel descriptor is not the same as the write pointer, it responds by:

1. Identifying the memory that contains the new command(s). If the Write Segment # is still the same as the Read Segment #, this is simply a matter of subtracting the Read Offset from the Write Offset.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

2. DMAing the command(s) into on-board memory for processing. When processing the commands, the controller uses the “Command Length” field in each command to determine where one command ends and another begins.

   If the commands span between segments, the board will fetch all the remaining bytes in this segment (since segments are assumed to be discontiguous). It then uses the link directive at the end of the segment to locate the next segment. When linking to the new segment, it increments the Read Segment # and resets the Read Offset. It will continue to DMA entire segments onto the controller until the read and write segment numbers match.

3. Updating the Read Pointer. Once the controller has processed the command(s) in the channel, it updates the Read Offset to point past the last processed command.

Figure 3-12 shows an overview of what takes place in memory when the host issues a command.

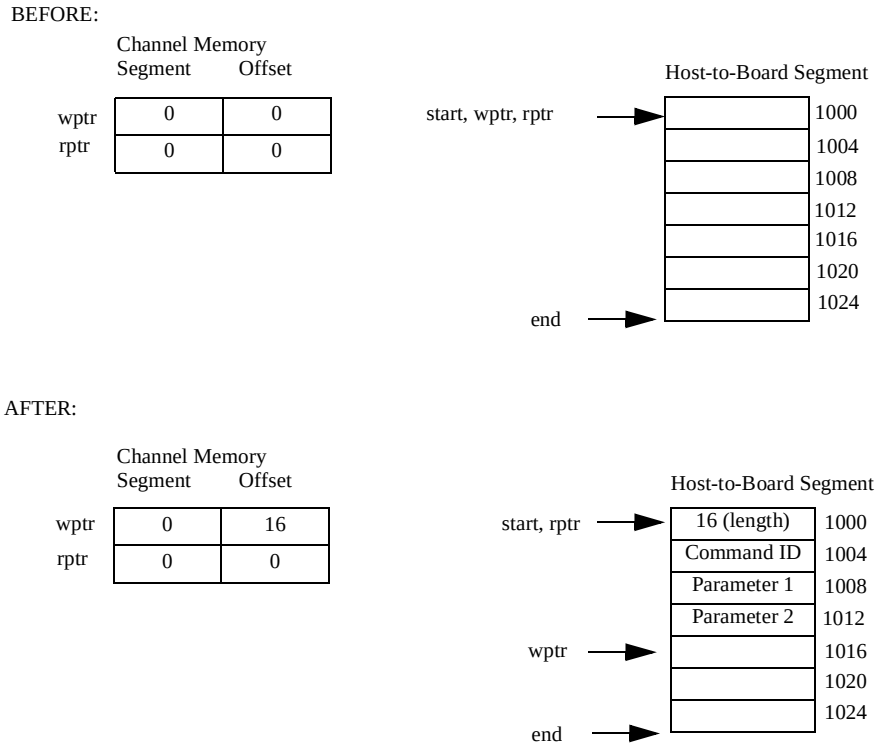


Figure 3-12. Example of Issuing a 16-Byte Host-to-Board Command

Linking from One Segment to the Next

When issuing commands, the host should link to a new segment whenever the number of bytes between the current Write Offset and the end of the segment is less than:

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

(length of proposed next command in bytes) ***plus*** 20 bytes[1]

The procedure to link one host-to-board segment to another is as follows:

1. Obtain memory in the host system for the new segment. This is typically done by:
   a. finding previously used segments to reuse
   b. searching pre-allocated memory tables in the driver, or
   c. allocating new memory from the operating system.
2. Write a Link Host-to-Board Segment directive into the space remaining in the current segment. This tells the controller where the new segment is located.
3. Update the write segment and the write offset of the channel descriptor.

.

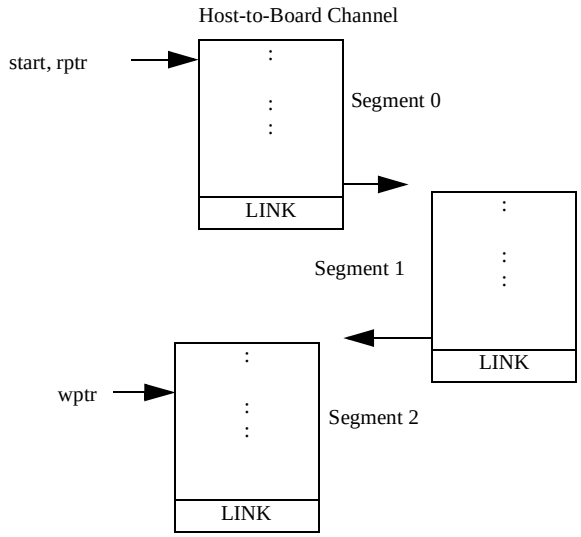


Figure 3-13. Linked Segments

Reading Command Responses

RC communicates the following types of information via the board-to-host channel(s):

- Responses to host-supplied RC commands
- Asynchronous indications concerning board status, transmitted/received data, etc.

Since there may be multiple board-to-host channels, there is a convention for determining which channel will be used to convey a particular response or indication.

1. The length of the Link Host-to-Board Segment directive.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

- *Responses* are posted to the board-to-host channel identified in the command. If no channel is identified, the controller will use the default response channel. The default response channel is the initial board-to-host channel created at RC boot time.
- Most *Indications* are automatically posted to the default response channel. The exception is Report Link to Next Segment, which the controller places in a given channel to inform the host that it is time to span to another segment.

The controller passes messages to the host the same way that the host issues commands to the controller. That is, it writes its next message at the physical address indicated by the write pointer and then updates the write pointer to point just past the newly written message.

When a segment becomes full, the controller spans to a new segment by incrementing the write segment field in the channel descriptor:

When the segment and offset numbers of the read pointer match those of the write pointer, the host may assume that it has read all of the pending messages in that channel.

The controller writes to its board-to-host channel(s) asynchronously. New reports may come to the host from the controller at any time, and the host does not engage in per-message handshaking. It merely processes the messages between the read pointer and write pointer of a given channel, and then updates the read pointer. You can arrange for the host to be interrupted periodically by a given board-to-host channel. In addition, the host can poll the channel(s) for new messages. These topics are discussed in the section on interrupts, below.

Note that segment memory is allocated to the controller as a pool of buffers (using the Allocate Board-to-Host Segment Memory directive). These segments are not assigned to any particular channel. If the controller runs out of available segments, it queues its messages internally until the host frees up some of the segments by reading the pending messages.

INTERRUPTS

The host enables interrupts from the 5211 on a per-channel basis. This is done by issuing separate Set Interrupt directives for the appropriate board-to-host channels. Such channels will cause the 5211 to interrupt the host when the necessary internal conditions are met.

Like the host, the 5211 writes to its board-to-host channels asynchronously. New messages may come from the 5211 at any time. The host need not worry about synchronization: It processes the messages between the channel descriptor's read pointer and write pointer and then updates the board-to-host read pointer.

Enabling Interrupts

After creating a board-to-host channel, the host can enable interrupts for the channel using the Set Interrupt directive. Interrupts are enabled on a per-channel basis.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

If interrupts are enabled for a given channel, it will cause the controller to interrupt the host when either of the following conditions is met:

- The controller has just placed a response in a channel which previously had no unprocessed responses ***and*** whose Interrupt Timer is zero.
- A channel's Interrupt Timer field decrements to zero (times out), and the controller determines that there are unprocessed responses in the channel.

The Interrupt Timer field is located in the channel descriptor of each board-to-host channel. As discussed in the following sections, it serves several purposes. These include suspending interrupts from the associated channel, re-enabling them, and fine-tuning how often the channel interrupts the host.

Generating an Interrupt

To generate an interrupt, the controller:

1. Writes the special value 0xFFFF into the Interrupt Timer field of the channel which caused the interrupt. This suspends further interrupts from the channel until the host resets the timer.
2. Interrupts the host.

Servicing Interrupts

To service interrupts:

1. The host acknowledges each interrupt as it occurs on a hardware level.
2. As soon as convenient, the host polls the board-to-host channel descriptors to determine which Interrupt Timer field(s) contain 0xFFFF.
3. For each board-to-host channel containing 0xFFFF in its Interrupt Timer field, the host:
   a. Processes all pending responses in the channel, updating the read pointer when done (or whenever it links to a new segment, if the responses are written to more than one segment).
   b. Resets the channel's Interrupt Timer field to any value from 0 and 0xFFFE. This re-enables the countdown-to-interrupt function for that channel.
4. If time allows, the host may check other board-to-host channels for pending responses. If there are, and the host chooses to service them, it should temporarily suspend interrupts from the channel by writing 0xFFFF to its Interrupt Timer field.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com



NOTES

The host should only write 0xFFFF to the Interrupt Timer field when it wishes to suspend interrupts from a channel whose timer has *not* expired. If the Interrupt Timer field already contains 0xFFFF, interrupts are suspended and the host should not rewrite this value.

It may not be necessary to poll the channels to determine the source of the interrupt(s). If the host assigns separate interrupt vectors to each board-to-host channel (using the Set Interrupt directive), it is possible for the interrupt service routine to already "know" the location of the channel descriptor associated with a given interrupt.

Using the Interrupt Timer

Each board-to-host channel has a 16-bit Interrupt Timer field as part of its channel descriptor. The host can dynamically modify this timer to meter its interrupt load on a per-channel basis.

The host sets the timer by writing any value from 0 through 0xFFFE into the field. The controller interprets this value as units of 100 microseconds. The internal 5211 resolution of this timer is 1 ms. This field is checked and updated every millisecond. Therefore values 1 to 10 correspond to a 1ms wait, values 11 to 20 correspond to a 2 ms wait, and so on. Once the specified amount of time expires, the controller will interrupt the host if (or as soon as) the channel contains pending messages.

Writing 0xFFFF to the Interrupt Timer field causes interrupts to be suspended from the associated channel. This value is used by both the controller and the host, as follows:

- After interrupting the host, the controller writes 0xFFFF to the timer of the channel which caused the interrupt. This suspends further interrupts from the channel until the interrupt service routine resets the timer.
- Before polling a board-to-host channel for pending messages, the host may write 0xFFFF to its timer field to suspend interrupts from the channel. It then resets the timer once the polling is completed. See "Channel Polling," below, for more information.



NOTE

***The Interrupt Timer field is irrelevant in descriptors for host-to-board channels since they are polled instead of interrupt-driven. It is only meaningful in board-to-host channel descriptors*.**

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Channel Polling

If the host can spare time to poll the board-to-host channels, it may be able to reduce the number of controller-generated interrupts. A board-to-host channel has pending messages whenever the write pointer in its channel descriptor differs from the read pointer. The host may extract these messages and update the read pointer as described in “Reading Command Responses.”

In general, the host suspends interrupts from a board-to-host channel before reading response(s) it has discovered in the channel. However, this is not required for all systems. Both cases are discussed in the following two subsections.

Channel Polling (with Interrupts Suspended)

The host may suspend interrupts from a board-to-host channel before processing messages that it has discovered by polling the channel. To do so, it writes 0xFFFF into the channel's Interrupt Timer field. Once the controller sees this value, it suspends interrupts from the channel until the host resets the timer to a value from 0x0 – 0xFFFE.



NOTE

If the Interrupt Timing field already contains 0xFFFF, as is the case after the channel has produced an interrupt, interrupts are already suspended and the host should not rewrite the value.

Be aware that there is a small “timing window” during which the 0xFFFF can be inadvertently overwritten by the controller. The sequence of events might be:

1. The controller reads the Interrupt Timer field to decrement it and determine if it has reached zero.
2. The host writes the 0xFFFF value to the Interrupt Timer field.
3. The controller writes the decremented value to the Interrupt Timer field, overwriting the 0xFFFF value placed there by the host.

To rule out this possibility, the host should “debounce” the channel's Interrupt Timer field when writing the 0xFFFF value. This is done by:

1. Writing 0xFFFF to the Interrupt Timer field
2. Reading the Interrupt Timer field
3. Comparing the value just read to 0xFFFF
4. Rewriting 0xFFFF to the field if the values don't match

The above sequence should be repeated until the 0xFFFF “sticks.”

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The benefit of explicitly suspending the interrupts associated with a channel is that the host does not need to mask controller interrupts during the entire interrupt service routine. Otherwise, masking might be necessary to insure that the board did not interrupt a routine which was already servicing the channel.

Channel Polling (with Interrupts Not Suspended)

In systems where masking interrupts from the controller is not a performance problem, the timer protocol described in the preceding section is not needed. If interrupts are masked in the host, the host may simply process the pending messages in the channel and write the new timer value afterwards.

This method causes “empty” interrupts to be generated if the timer expires while the host is processing messages. In such a situation, the controller generates an interrupt, but the host has interrupts masked and therefore does not “see” it until interrupts are unmasked. At that time, there are no more messages pending in the channel. Ordinarily, the timing window in which this can occur is small enough to make the frequency of empty interrupts insignificant. However, if the host uses very small values in the Interrupt Timer fields, the controller may generate an excessive number of empty interrupts. In this case, it is advisable to suspend interrupts before polling the channels.

ADDITIONAL CONSIDERATIONS

Creating Dedicated Channels

RC channels exist as independent entities and do not require any fixed relationship for correct operation. Most applications, however, will typically create separate channels dedicated to specific purposes. For example, you will probably want to set up board-to-host channels to respond to specific kinds of requests (such as requests to transmit data over a network). The “ID # of Response Channel” field in request-type commands allows you to select which board-to-host channel will be used by the controller to respond to a given request.

BUILDING COMMANDS

Once the host has booted the RC Interface it can issue commands to the 5211 through the initial host-to-board channel, and read 5211-generated messages from the initial board-to-host channel.

Host-to-board directed information is described by **requests** and **directives**. Requests have responses associated with them (a reply in the board-to-host direction). Directives, on the other hand, have no associated reply.

Board-to-host directed information is described by **responses** and **indications**. Responses are paired with requests that are host-initiated. Indications are asynchronous events which have no pre-initiated host-to-board trigger.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Location of Commands and Tx/Rx Data

The 5211 expects queued commands as well as transmit/receive data to reside in host memory, transferring commands, responses, and transmit/receive data between the host and controller-resident buffers using a high-speed, on-board DMA channel.

COMMONLY USED COMMAND FIELDS

The fields in a command are explained in the section on that command except for the following fields, which occur in most 5211 commands:

- Command Control Block (CCB)
- Channel ID
- Transfer Options Word

Their descriptions follow.

Command Control Block Format

The Command Control Block (CCB) is an 8-byte data structure placed at the start of any request, directive, response, or indication sent through RC channels. The command format is shown in Figure 3-14.

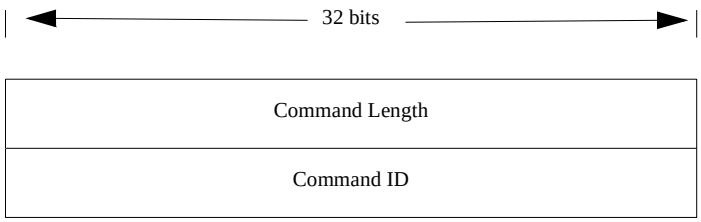


Figure 3-14. Command Control Block Structure

The fields in the Command Control Block are:

- Command Length (4 bytes)

  This field contains the number of bytes in the entire command, including the CCB, the parameters, and any extra padding. This value is used by the reader of a channel to find the next command in the channel.

- Command ID (4 bytes)

  This field identifies the specific command (request, directive, response, or indication) to be processed. The command IDs of the 5211-specific command set are shown in Table 3-4.



NOTE

There is no relationship between a "command ID" and a "channel ID."

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Channel ID Fields

Many RC commands include a field that identifies a specific channel. The name of the field varies slightly depending on its purpose. For example, the Set Interrupt directive has a "Target Channel ID" field that identifies the channel for which interrupts are being enabled. Request-type commands (such as Request Statistics) contain the field "ID # of Response Channel" to tell the controller where to write its response.

In all cases, the ID of a channel is the offset (in bytes) at which its channel descriptor starts in shared memory. Examples of channel IDs include:

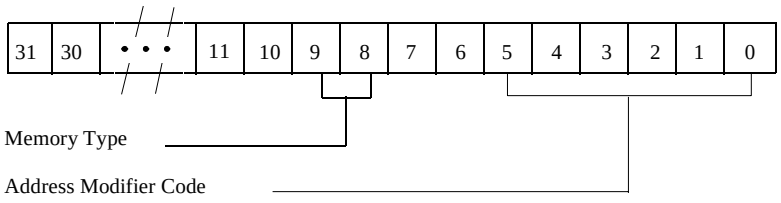
- The ID of the initial board-to-host channel created at RC boot time. This value is always 0x20, since the channel descriptor of the initial board-to-host channel always begins at an offset of +0x20 (32 bytes) from the start of shared memory.
- The ID of the initial host-to-board channel created at RC boot time. This value is always 0x10, since the channel descriptor of the initial board-to-host channel always begins at an offset of +0x10 (16 bytes) from the start of shared memory.
- The ID of any channel created after RC boot time. The RC Interface assigns an ID to a channel upon successful completion of a Create Channel request (either host-to-board or board-to-host). It returns the ID of the new channel via the appropriate response (Report New Host-to-Board Channel or Report New Board-to-Host Channel).

Transfer Options Word

The Transfer Options Word is used to specify the type of VMEbus transaction used for data transfers. The bits in the transfer options word indicate whether the transfer should be 16, 32, or 64 bits wide and which address modifier should be used.

The Transfer Options Word is defined as a 32-bit unsigned integer, even though only the least significant word of the field (bits 0 -15) is used to specify options. The most significant word in the field (bits 16 - 31) is a reserved 16-bit field.

The format of the Transfer Options Word is shown in Figure 3-15.



Note: All unused bits are reserved and must be cleared to 0.

Figure 3-15. Transfer Options Word for Data Transfers

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Bits 0-5 Address Modifier

This field contains the VMEbus address modifier used by the controller for all VMEbus data transfers associated with this data transfer. It is not modified in any way by the controller. Valid entries in this field are listed in Table 3-5:

Table 3-5. Address Modifiers Allowed in Data Transfers

Code	Function
0x09	Ext. Non-Privileged Data Access
0x08	Ext. Non-Privileged Block Transfer
0x0D	Ext. Supervisory Data Access
0x0F	Ext. Supervisory Block Transfer
0x29	Short Non-Privileged I/0
0x2D	Short Supervisory I/0
0x39	Std. Non-Privileged Data Access
0x3B	Std. Non-Privileged Block Transfer
0x3D	Std. Supervisory Data Access
0x3F	Std. Supervisory Block Transfer

Bits 6-7 Reserved

These bits ar reserved and must be cleared to 0.

Bits 8-9 Memory Type

This two-bit field specifies the width of the data transfer. Permitted values are shown in Table 3-6.

Table 3-6. Memory Type for Data Transfers

Bit 9	Bit 8	Memory Type
0	0	(reserved)
0	1	16-bit transfers
1	0	32-bit transfers
1	1	64-bit transfers

Bits 10-31 Reserved

These bits are reserved and must be cleared to 0.

For example:

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

For a DMA transfer in a SUN 3/260 4.1 operating system, the transfer options word shown in Figure 3-16 would be appropriate.

Bit #	Bits 31 - 10	9	8	7	6	5	4	3	2	1	0
Setting	(all 0's)	1	0	0	0	1	1	1	1	0	1

Figure 3-16. Transfer Options for DMA Transfer in SUN 3/260 4.1 Operating System

This specifies 32-bit addressing and a VME address modifier of 0x3D.

CONNECTING THE STATION TO THE RING

The host instructs the 5211 to connect to (or disconnect from) the FDDI ring by issuing a Ring Control directive. If the command completes successfully, the 5211 issues a Ring Status Indication to report the new ring status – ring "up" or ring "down."

Be aware that the 5211 issues a Ring Status Indication only *when the ring status changes*. If the 5211 fails in its attempt to connect (or disconnect), it does *not* issue the indication. It does not issue the indication if it receives a directive to connect/disconnect when it is already in the specified state.

If the station is not currently connected to the ring, a successful connection attempt proceeds as follows:

1. The host issues a Ring Control directive with the value 0x1 (Connect to Ring) in the options field.
2. After connecting to the ring, the 5211 issues a Ring Status Indication with the value 0x1 (Ring Up) in the status field. This indication is written to the default response channel. The default response channel is the initial board-to-host channel created by the host when it boots the RC Interface. The host can detect pending messages by polling the channel and/or enabling interrupts.

Step 2 should occur a few seconds after Step 1. If it does, the station has successfully connected to the ring and the host can proceed to transmit and receive frames.

If Step 2 does *not* occur within 5 - 10 seconds, the station has failed to connect to the FDDI ring. The connection attempt will succeed only if a wide range of parameters are properly set for the ring. These requirements, which involve multiple layers of the network interface, are documented in the FDDI SMT specifications.

Additional information about the state of the ring is signalled by LED5, LED6, and LED7. LEDs are described on *LEDs on page 177*.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

TRANSMITTING DATA

To transmit a frame over the network, the host writes a transmit command (either Transmit Raw Data or Transmit Datagram) to a host-to-board channel. The command points to the outgoing data in the host memory. If desired, the frame may be stored as fragments in discontiguous memory locations (see *SCATTER/GATHER CAPABILITIES on page 66*).

For details on the two data transmission commands, refer to *TRANSMIT RAW DATA on page 121* and *TRANSMIT DATAGRAM on page 127*.

Queuing Frames for Transmission

When the 5211 processes a transmit command, it: 1) attempts to move the data to be transmitted into its internal communications buffer, and 2) issues a Tx Response. This response tells the host either that: 1) the data has been successfully transmitted over the network, or 2) an error has occurred.

Once the host has received the Tx Response, it can reuse the memory space from which the 5211 obtained the frame.

The 5211 queues outgoing frames in its communications buffer until it gets the token. The 5211 transmits as many frames as possible until the token holding time in the MAC expires. Any untransmitted frames are held until the token returns to the 5211.

Synchronous vs. Asynchronous Frames

Generally, all queued synchronous frames are transmitted before asynchronous frames. Note that the RC Interface does not guarantee that a synchronous frame queued for transmission in a host-to-board channel will go out before any asynchronous frames queued in the channel. RC processes commands in the channel in a first-come first-served manner. As the 5211 executes each Tx command, it pulls the corresponding frame across the bus. At this point, it still has not determined which frames should go first.

For example, suppose the host queues up ten asynchronous frames followed by a synchronous frame in the same host-to-board channel. Each frame is transferred on-board and queued internally at the FDDI System Interface (FSI) level.

If the token has *not* been captured at this time, the synchronous frame is transmitted before the ten asynchronous ones once the 5211 gets the token.

If the 5211 captures the token at the same time that one or more synchronous frames are being queued on board, any asynchronous frames which are already queued up may be transmitted before the synchronous one(s).



NOTE

Synchronous Frames are not supported in our implementation of SMT.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

RECEIVING DATA

For the 5211 to upload received frames, the host must supply the controller with a **buffer list** consisting of **buffer list elements** which point to **receive buffers** in host memory. These terms have a meaning specific to the RC Interface and are defined in this section.

To upload a received frame, the controller does the following:

1. The 5211 DMAs the frame to a host-resident receive buffer. If the frame is larger than the buffer, the 5211 scatters the frame across multiple buffers (see *SCATTER/GATHER CAPABILITIES on page 66* for details on frame scattering).

   Only one frame (or frame fragment) is uploaded per buffer. If no receive buffers are available, the 5211 holds the frame internally until it can upload it.

2. The 5211 then issues a **receive indication** which points to the receive buffer(s) containing the uploaded frame. The 5211 command set currently supports one receive indication – Rx Frame.

All receive indications are posted to the default response channel (that is, the initial board-to-host channel created at RC boot time).

Note that receive indications are written to the default response channel, but the actual received frames are not. The host must separately allocate memory space for these two purposes. The considerations involved in allocating memory for RC channels (that is, segment memory) are documented in the *RC Interface* section of this user's guide. Guidelines for setting up receive buffers are discussed below.

Alignment of Received Frames

When the 5211 uploads a frame to a receive buffer, it aligns the information field of the frame on a longword boundary. It places a 3-byte pad preceding the MAC frame header. This 3-byte pad is *not* included in the "Frame Length" field of the receive indication. The MAC frame header is 13 bytes, including the Frame Control field (1 byte) and the Source and Destination addresses (6 bytes each).

Allocating And Managing Receive Buffers

To allocate host memory for incoming frames, the host issues the Set Up Receive Buffers directive (). This directive is issued at initialization time. It defines a **buffer list** which points to **receive buffers** – a set of same-sized, longword-aligned buffers to which the 5211 may upload received frames.

Creating A Buffer List

To set up a buffer list, the host allocates a block of contiguous memory (usually a page). It writes **buffer list elements** into this space, terminated by the value 0xFFFFFFFF.

Each buffer list element points to a specific receive buffer somewhere in host memory. All receive buffers must be of the same size and longword-aligned. Valid buffer sizes range from 512 bytes to 8K bytes, inclusive.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Figure 3-17 shows the format of a buffer list element.

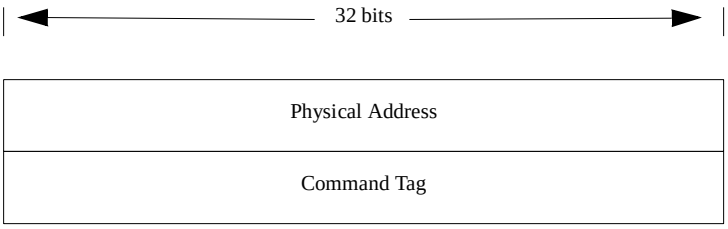


Figure 3-17. Format of Buffer List Element

The “Physical Address” field is used by the 5211 to DMA data to the buffer. Receive buffers must be aligned on a longword boundary.

The “Command Tag” field is a value assigned by the host which is echoed back by the controller via a receive indication. This tag is usually the address of a host memory descriptor or virtual memory address for the buffer.

Marking the Last Valid Element in the Buffer List. The host does not have to fill the buffer list completely with valid elements (elements which point to an available receive buffer). Instead, it maintains a *delimiter* located immediately after the last valid element. The delimiter has the same format as a buffer list element, but has the special value 0xFFFFFFFF in its Physical Address field. To create a buffer list containing N elements, set the number of elements specified using the Set Up Receive Buffers directive to N + 1 to allow for the delimiter.

Ownership of Receive Buffers

Each valid element in the buffer list points to a receive buffer in host memory. The 5211 “owns” the receive buffer until it has placed data in it and notified the host by issuing a receive indication. At that time, ownership of the buffer returns to the host. The host must post new buffers to the list to keep the 5211 supplied with available receive buffers. See “Host-Managed Tasks” on page 65 for details.

Managing the Buffer List

Board-Managed Tasks. When the 5211 executes the Set Up Receive Buffers directive, it DMAs a copy of the entire buffer list on-board. As it processes received frames coming off the net, it starts working its way down the list, using successive buffer list elements to identify receive buffers to which it can upload frames.

Each time it uploads a frame to one or more buffers, the 5211 moves its list pointer ahead to the next available buffer list element. It then issues a receive indication to notify the host that: 1) received data is ready for the host to process, and 2) one or more buffers can be reused (or newly allocated) and posted to the buffer list. The method for posting buffers to the list is described in “Host-Managed Tasks”.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

If the buffer size is 4.5K bytes or larger, there is always a one-to-one correspondence between the number of posted buffers and the number of received frames uploaded to the host. This is because the 5211 only places one frame (or frame fragment) in each buffer. If a smaller buffer size is being used, large frames will be scattered over multiple buffers. The buffer size is defined using the "Length in Bytes of Each Buffer" field in *SET UP RECEIVE BUFFERS on page 134*.

The 5211 continues moving down the buffer list until one of the following conditions occurs:

- It reaches the physical end of the list as defined by the Set Up Receive Buffers directive. In this case, the 5211 simply wraps back to the top of the list and continues to work down the list, or
- It runs out of valid elements (i.e., its list pointer reaches the delimiter). It re-reads the host's copy of the buffer list to obtain new buffers that the host has posted to the list. If the host has not updated the list, the 5211 polls the list at successively larger intervals starting at 1 ms. until it determines that the delimiter has moved. The on-board copy of the buffer list is refreshed and uploading frames is resumed.

Each time the 5211 gets the buffer list from the host, it wipes out its old copy of the buffer list. However, it does not return its list pointer to the top of the list. Instead, it continues to point to the same buffer list element that it was pointing to before it re-read the list. If that specific element in the refreshed list *is not* the delimiter, the 5211 assumes that it is a valid element and proceeds to upload data as usual. If that element *is* the delimiter, the 5211 assumes there are no free buffers for uploading frames. It polls the buffer list until the host posts new buffers.

Host-Managed Tasks. To manage the buffer list, the host typically keeps a reader pointer and writer pointer into the list. The reader pointer points to the "next" buffer list element from which the host expects to obtain data. The writer pointer points to the delimiter.

As the 5211 passes receive indications back to the host, the host extracts the data from the buffers and moves the reader pointer down the list to the next free buffer.

To post a buffer back to the list, the host should:

1. Write the delimiter (0xFFFFFFFF) into the Physical Address field of the buffer list element immediately after the old delimiter. (The list contains *two* delimiters in consecutive buffer list elements.)
2. Insert the new buffer at the writer pointer, overwriting the old delimiter. Fill in the Command Tag field before the Physical Address field. This eliminates a window during which the 5211 could pick up the new physical address before the new command tag becomes valid.
3. Update the writer pointer to point to the new delimiter.

This procedure prevents the 5211 from inadvertently reading past the delimiter.

It is safe for the host to wrap to the top of the list so long as it posts a new buffer only ***after*** the 5211 has passed back one of "its" buffers via a receive indication.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Balancing the Load of Rx Frame Processing

On occasion, frames may arrive in large batches causing the 5211 and host to process the data inefficiently. For example, suppose that 64 frames addressed to the host arrive in a very small time frame. If the 5211 tries to process and upload all 64 frames at once, other important tasks (such as pending transmits) are left hanging. The buffer list is likely to become depleted. The 5211 has to wait for the host to process the uploaded data and post new buffers to the list before it can get buffers for additional frames.

This situation can be avoided by placing an appropriate value in the “Rx Load Adjustment” field of the Set Up Receive Buffers directive. This field sets the maximum number of received frames that the 5211 will process at a given time. The default value is 32.

Using the previous example, assume that the 5211 has 64 frames to upload and the “Rx Load Adjustment” field is set to 16. The 5211 loads 32 frames at a time. Other tasks, such as pending transmits, are attended to at this time. Control is returned to the receive routine to process the next 32-frame chunk.

The optimum value for the “Rx Load Adjustment” field depends on a wide range of variables. These include the speed of the host processor, the size of the receive buffers, the amount of network traffic addressed to the host, and other factors. To find the best setting, test different values in the field.

The Rx Load Adjustment feature is disabled if the host clears the field to 0 or enters an out-of-bounds value. If the feature is disabled, the receive frame handling routine will attempt to process all the received frames it knows about, even if it has to use up all available buffers in the buffer list. The 5211 will not refresh its copy of the buffer list until it encounters the delimiter (0xFFFFFFFF). Values currently supported are discussed in the *RX LOAD ADJUSTMENT (4 bytes) on page 135*.

Insufficient Receive Buffers

Since the 5211 relinquishes a buffer after it has placed received data in it and notified the host, it is the host’s responsibility to post new buffers to the buffer list. If it fails to do so, the 5211 will eventually use up all of the buffers in the list. It buffers frames internally until it runs out of available RAM. Subsequent incoming frames addressed to the host will be lost. No incoming data can be received until the host allocates new buffers to the list, thereby enabling the board to free up its on-board memory by uploading previously received data to the host.

SCATTER/GATHER CAPABILITIES

The scatter/gather feature enables the 5211 to:

- Upload a received frame to discontiguous locations in host memory (“frame scattering”)
- Download a transmit frame from discontiguous locations in host memory (“frame gathering”)

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The considerations involved in frame scattering are different from those of frame gathering. This is because the rules for allocating and using receive buffers are much stricter than those concerning outgoing data.

Frame Scattering

If the size of a received frame exceeds that of the 5211's receive buffers, the 5211 will scatter the frame across multiple buffers. It reports the location of the frame fragments in the receive indication. The 5211 only writes one frame (or frame fragment) to a receive buffer before passing it back to the host.

Frame Gathering

If the host issues a transmit request containing multiple frame fragments, the 5211 gathers the frame. The 5211 places no restrictions on the alignment of the data on any fragment.



NOTE

The maximum number of gather elements in a transmit is 16.

ACCESSING THE CAM

The 5211 includes a 32-entry CAM (Content-Addressable Memory) in which the host can store up to 32 destination addresses. The host can add and remove addresses from the CAM, read specific entries in the CAM. These tasks are performed using certain commands in the 5211's controller-specific command set, as documented in the next chapter.



NOTE

Be careful to distinguish between Report CAM Address and CAM Address Response. The former reports the current address stored in a CAM entry. The latter is issued in response to an attempt by the host to add or remove a CAM entry.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

ACCESSING THE MIB

Since SMT functions (as specified in FDDI Station Management, revision 6.2 or 7.3) are fully implemented in the 5211's firmware, the Management Information Base (MIB) is also stored on-board. The MIB contains a large set of parameters (MIB attributes) which are used by the 5211 when performing SMT-level tasks.

The default MIB parameters on the 5211 are correct for most installations. Generally they should be left in their default settings.

The 5211 supports commands for reading or changing the value of MIB attributes. These commands, which are documented in the next chapter, may be issued any time after the RC Interface is initialized (see *BOOTUP AND RESET SEQUENCE on page 31*).

Since MIB attributes should usually not be modified, the main reason for accessing the MIB is to acquire data for monitoring, controlling and statistics-gathering purposes.

Reading the Value of a MIB Attribute

To determine the current value of a MIB attribute:

1. Issue the command Get MIB Attribute.
2. Read the returned MIB Attribute Response.

Setting the Value of a MIB Attribute

To assign a value to a MIB attribute, issue the command Set MIB Attribute. Since this is an RC directive, no response is returned by the 5211. The command cannot be used to change "read-only" attributes.

Changing the MIB attributes requires an *expert-level* understanding of the FDDI Station Management (SMT) specifications. It is not recommended for most applications.

Resetting the 5211 or cycling the power restores the default MIB attributes.

SMT FRAME AND EVENT TRACING

If the host needs to monitor the network closely, it can instruct the 5211 to upload SMT frames and/or events as they are received by the board. See the SMT Trace directive and SMT Event Indication for details.

Like the commands for accessing the MIB, the SMT trace commands require a good understanding of the SMT specifications. The SMT 6.2 and 7.3 standards define the attributes associated with SMT frames and events.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5211 COMMAND SET 4

This chapter describes the 5211 command set which includes

- Common Boot (CB) Command Set
- Report/Command (RC) Command Set
- FDDI Command Set

The CB command set consists of low-level commands used primarily for booting the RC interface. It is also used for other tasks such as performing diagnostics and downloading firmware. See USING THE COMMON BOOT INTERFACE in the previous chapter for detailed information on using the Common Boot Interface.

Both the RC and FDDI command sets are classified as RC commands, and use the same command format described in *COMMONLY USED COMMAND FIELDS on page 58*. They are listed in separate groups according to their function.

The RC commands are “generic” commands used to create and manage the RC interface feature of several Interphase Controllers. See *RC Command Set on page 99* for a list of RC commands.

The FDDI command set consists of those used to transmit and receive data, as well as to perform other network-related tasks such as monitoring the FDDI MIB (Management Information Base). See *FDDI COMMAND SET on page 71* for a list of FDDI commands.

RC Commands are divided into four types:

1. Requests
2. Directives
3. Responses
4. Indications

RC Host-to-board directed information is described by **requests** and **directives**. Requests have responses associated with them (a reply in the board-to-host direction). Directives have no associated reply.

RC Board-to-host directed information is described by **responses** and **indications**. Responses are paired with requests that are host-initiated. Indications are asynchronous events which have no pre-initiated host-to-board trigger.

COMMON BOOT COMMAND SET

The following Common Boot commands are available:

- BOOT
- DIAG
- FILL

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

- FLSH
- GRNL
- JUMP
- PEEK
- POKE
- DNLD
- BURN
- REDL
- STUF
- HGET
- HPUT
- INFO

See *COMMON BOOT COMMANDS on page 72* for a detailed description of each command.

RC COMMAND SET

The following RC commands are available:

- SET DMA BURST
- SET BUS TIMEOUT
- REQUEST STATISTICS
- SET INTERRUPTS
- CREATE HOST-TO-BOARD CHANNEL
- CREATE BOARD-TO-HOST CHANNEL
- LINK HOST-TO-BOARD CHANNEL
- ALLOCATE BOARD-TO-HOST SEGMENT MEMORY
- REPORT PRINTF CALL
- ERROR MESSAGE
- REPORT STATISTICS
- REPORT NEW HOST-TO-BOARD CHANNEL
- REPORT NEW BOARD-TO-HOST CHANNEL
- REPORT LINK TO NEXT SEGMENT

See *REPORT/COMMAND (RC) COMMANDS on page 99* for a detailed description of each command.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

FDDI COMMAND SET

The FDDI controller-specific command set for the 5211 is:

- TRANSMIT RAW DATA
- TRANSMIT DATAGRAM
- SETUP RECEIVE BUFFERS
- REQUEST STATION ADDRESS
- SET STATION ADDRESS
- RING CONTROL
- PERFORM MAINTENANCE
- GET MIB ATTRIBUTE
- SET MIB ATTRIBUTE
- SET CAM ADDRESS
- CLEAR CAM ADDRESS
- FLUSH CAM ADDRESS
- SMT TRACE
- SET HARDWARE PARAMETERS
- GET HARDWARE PARAMETERS
- GET CAM ADDRESS
- Tx RESPONSE
- Rx RESPONSE
- REPORT STATION ADDRESS
- RING STATUS INDICATION
- CAM ADDRESS RESPONSE
- MIB ATTRIBUTE RESPONSE
- SMT EVENT INDICATION
- REPORT HARDWARE PARAMETERS
- REPORT CAM ADDRESS

See *FDDI-SPECIFIC COMMANDS on page 119* for a detailed description of each command.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

COMMON BOOT COMMANDS

This section describes the CB Commands. The following table shows the available CB commands.

Table 4-1. V/FDDI 5211 Common Boot Commands

COMMAND	DESCRIPTION	SEE PAGE
BOOT	This command is used to boot an alternate firmware image	75
DIAG	This command performs a series of tests whose definition is specific to the 5211.	77
FILL	This command allows the host to modify controller memory or hardware	80
FLSH	This command allows the host to control the state of the controller's on-board LED	82
GRNL	This command allows the host to control the state of the controller's on-board LED	83
JUMP	This command allows the host to force the CPU to do an unconditional jump	84
PEEK	This command allows the host to examine controller memory	85
POKE	This command allows the host to modify controller memory	86
DNLD	This command is used to download an image to the controller	87
BURN	This command is used to store a saved downloadable image into FLASH EPROM to be used as the EXEC 0 image	89
REDL	This command allows the host to control the state of the controller's on-board LED	90
STUF	This command allows the host to modify controller memory or hardware	91
HGET	This command allows the host to examine controller memory or hardware	93
HPUT	This command allows the host to modify controller memory or hardware	95
INFO	This command allows the host to identify a controller as well as obtain various controller specific information	97

Common Boot commands support the following functions:

- Boot resident and downloadable interface
- Controller LED commands
- Controller memory read and write
- Controller diagnostic commands
- Burn image into FLASH memory

The following define statements describe the Common Boot command set, along with the CBOK and FAIL responses:

```
#define MAKE_LONG(a,b,c,d) (((u32)a << 24) | ((u32)b << 16) | ((u32)c << 8) | (u32)d)

#define CB_BOOT  MAKE_LONG('B','O','O','T')

#define CB_EXEC  MAKE_LONG('E','X','E','C')

#define CB_DIAG  MAKE_LONG('D','I','A','G')
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

```
#define CB_FILL    MAKE_LONG('F','I','L','L')
#define CB_FLSH    MAKE_LONG('F','L','S','H')
#define CB_GRNL    MAKE_LONG('G','R','N','L')
#define CB_JUMP    MAKE_LONG('J','U','M','P')
#define CB_PEEK    MAKE_LONG('P','E','E','K')
#define CB_POKE    MAKE_LONG('P','O','K','E')
#define CB_DNDL    MAKE_LONG('D','N','L','D')
#define CB_BURN    MAKE_BURN('B','U','R','N')
#define CB_REDL    MAKE_LONG('R','E','D','L')
#define CB_STUF    MAKE_LONG('S','T','U','F')
#define CB_CBOK    MAKE_LONG('C','B','O','K')
#define CB_FAIL    MAKE_LONG('F','A','I','L')
```

All of the above-listed defines are unique in both the upper and lower words. This enables the host to make both 16-bit and 32-bit accesses to shared memory when using Common Boot, provided that the controller supports both types of accesses.

A description of the Common Boot command set appears latter in this chapter.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

DOWNLOADING CODE

A download mechanism is provided on Interphase controllers that support the Common Boot interface. This feature allows the host to download code to the 5211 firmware for execution. Applications of the feature include:

- Booting an interface other than the native RC Interface
- Making firmware upgrades in the field

Interphase provides utilities which automate the download process. These utilities are tied to specific controller types (for example, 4211, 4207, and so on) and hardware/firmware revision levels. They typically use the Common Boot commands “POKE” and “BOOT” to download and execute the code.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

BOOT

Group: Common Boot

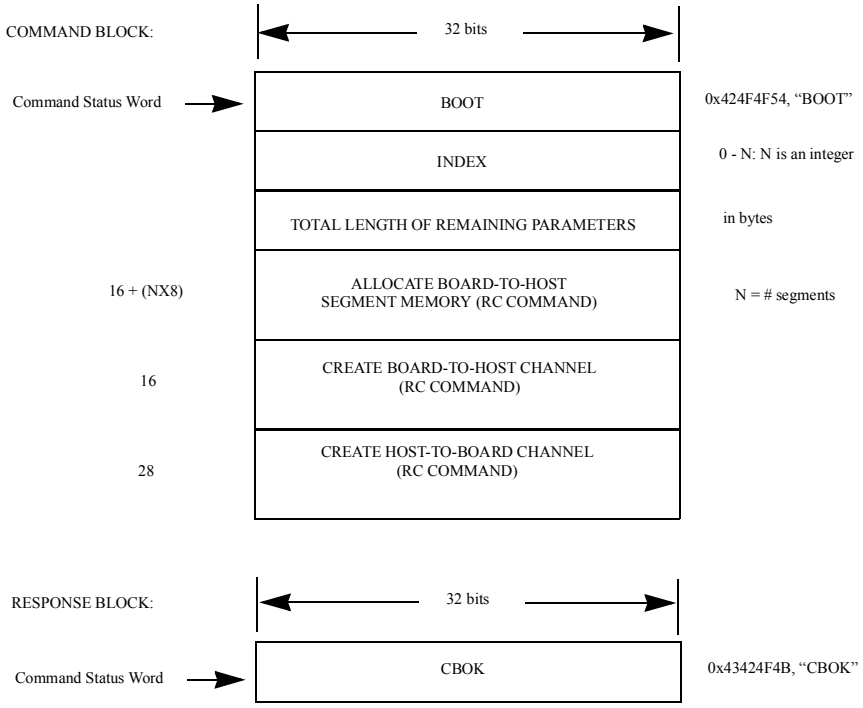


Figure 4-1. BOOT Command and Response Format

FUNCTION

Boot a native or downloaded interface. This section describes the format of the BOOT command when booting the controller's native Report/Command (RC) Interface (that is, the default interface stored in EPROM). For information on booting a downloaded interface, see *DOWNLOADING CODE on page 74*.

If you are booting the native RC Interface, the BOOT command includes three RC commands:

- Allocate Board-to-Host Segment Memory
- Create Board-to-Host Channel
- Create Host-to-Board Channel

These commands establish the initial pair of RC channels needed to issue commands to and obtain responses from the controller. See *FUNCTIONAL OVERVIEW on page 27* for a detailed discussion of segments, channels, and related topics.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

CODE

0x424F4F54 (ASCII “BOOT”)

PARAMETERS

- INDEX

  Boot Entry Point Index (0,1).
  With a supplied 0, the native RC Interface will boot.
  With a supplied 1, the downloaded image is moved to the execution RAM and the control is passed to the new image. No other parameters are required for index 1.

- TOTAL LENGTH OF REMAINING PARAMETERS

  Specifies the aggregate length, in bytes, of all RC commands which the 5211 may expect to find in shared memory following the Total Length field in this command

- ALLOCATE BOARD-TO-HOST SEGMENT MEMORY

  Allocates host-resident channel segment memory to the board.

  The command must be executed ***before*** the two channel creation commands. It must therefore precede these two commands in the BOOT command structure, as shown in Figure 4-1. This is necessary because the controller requires at least two board-to-host segments to create a board-to-host channel.

- CREATE BOARD-TO-HOST CHANNEL

  Establishes the initial board-to-host channel. The position of the descriptor for the created channel is written to a fixed location in shared memory. The offset of the initial channel descriptors is shown in the shared memory map in Figure 3-5 on page 32. Create Board-to-Host is an RC command.

- CREATE HOST-TO-BOARD CHANNEL

  Establishes the initial host-to-board channel. The position of the descriptor for the created channel is written to a fixed location in shared memory. Create Host-to-Board is an RC command.

PROGRAMMING SEQUENCE

If CBOK, 1) supply the parameters, then 2) supply the BOOT command. When CBOK occurs or the booted interface responds, the Common Boot Interface “goes away” until the host performs a hardware reset on the board.

As discussed in the previous chapter, the Create Channel directives normally return to the host the logical address of the first segment for the new channel. This is not possible at boot time since the channel for response is not yet established. Therefore, the controller will use the first segment in the list provided by the Allocate Board-to-Host Segment Memory command as the initial board-to-host channel segment.

ERROR RETURNS

If FAIL, the CB ERROR status block will be filled in.
BOOT_0_ATTEMPT (0x0D)

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

DIAG

Group: Common Boot.

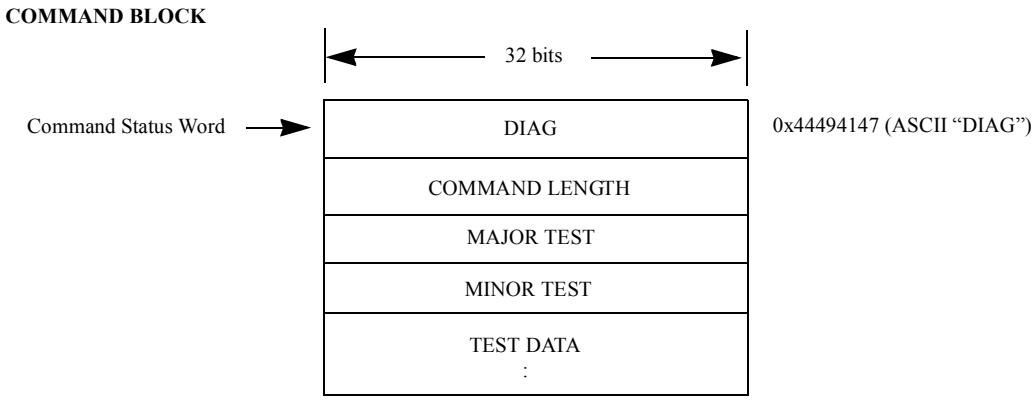


Figure 4-2. Format of the DIAG command

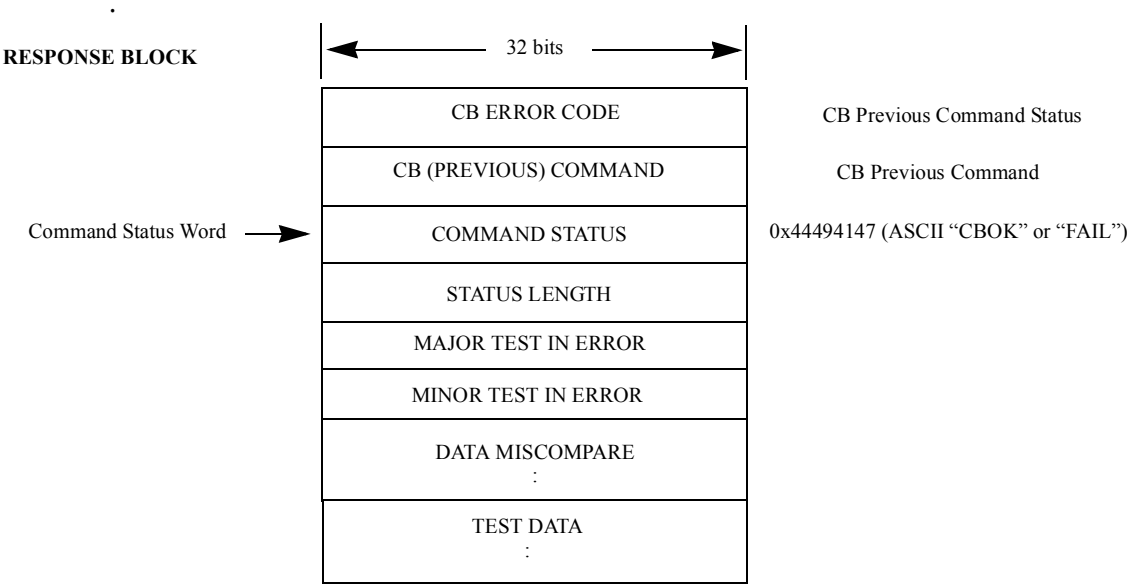


Figure 4-3. "FAIL" Response Block

FUNCTION

Execute Diagnostics. This command performs a series of tests whose definition is specific to the 5211. For a description of how the fields are defined, consult the section on power-up diagnostics in the Functional Overview chapter and refer to *DIAGNOSTICS on page 181*.

CODE

0x44494147 (ASCII "DIAG")

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

PARAMETERS

- COMMAND LENGTH

  This field contains the total number of bytes in the test command, including itself, but excluding the Command Status Word (which contains the ASCII “DIAG”).
- MAJOR TEST

  Major test to be executed.
- MINOR TEST

  Minor test (if any) to be executed.
- TEST DATA

  Test-specific data to be used in the test.

PROGRAMMING SEQUENCE

If CBOK, supply the parameters, then supply the DIAG command. When CBOK occurs, the test has been successfully completed. If FAIL occurs, the 5211 returns the response structure shown in Figure 4-3.

RESPONSE BLOCK

The Response Block contains the Command Status (CBOK or FAIL). If the response is normal, the board will place CBOK in the Command Status Word field and the other fields are irrelevant (that is, Don’t Cares). If the response is “FAIL,” additional information is provided in the fields as shown in Figure 4-3 and described below:

- CB ERROR CODE

  This field gives the error code of previous CB command. See *CB ERROR CODES on page 217* for error codes.
- CB PREVIOUS COMMAND

  Previously issued Common Boot command.
- COMMAND STATUS

  Status (CBOK or FAIL) of the issued command.
- STATUS LENGTH

  This field contains the total number of bytes in the status block, including itself, but excluding the Command Status Word (which is the ASCII “FAIL”). Contains 0x10 (or 0x18) if no error data is returned.
- MAJOR TEST IN ERROR

  This reports the major test number of the failed test.
- MINOR TEST IN ERROR

  This reports the minor test number, if any, of the failed test.
- DATA MISCOMPARE

  Error data (if any).
- TEST DATA

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

This field contains test-specific information about the error. Some errors have data associated with them, while others do not. Consult the individual test description for details.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

FILL

Group: Common Boot

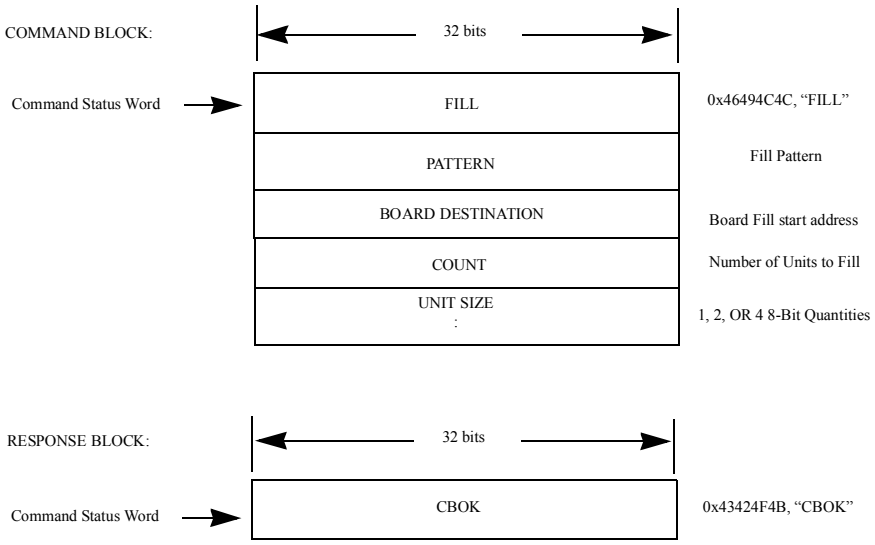


Figure 4-4. FILL Command and Response Format

FUNCTION

Fill area with pattern.

Starting at the provided DESTINATION ADDRESS, this function will write to the contents of memory the value contained in the command parameter PATTERN. This transfer will be made in data widths provided by the UNIT SIZE parameter. The range of memory checked will be determined by the parameter UNIT COUNT.

CODE

0x46494C4C (ASCII "FILL")

PARAMETERS

- PATTERN

  The pattern used for filling memory.
- BOARD DESTINATION

  The starting board address for pattern filling.
- COUNT

  The number of units to fill with pattern.
- UNIT SIZE

  The size of unit (1=8, 2=16, or 4=32 bits).

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

PROGRAMMING SEQUENCE

If CBOK, supply the parameters, then supply the FILL command. When CBOK occurs, the pattern has been successfully laid down.

ERROR RETURNS

If FAIL, then CB ERROR block will be filled in.
BAD_SIZE (0x0B)

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

FLSH

Group: Common Boot

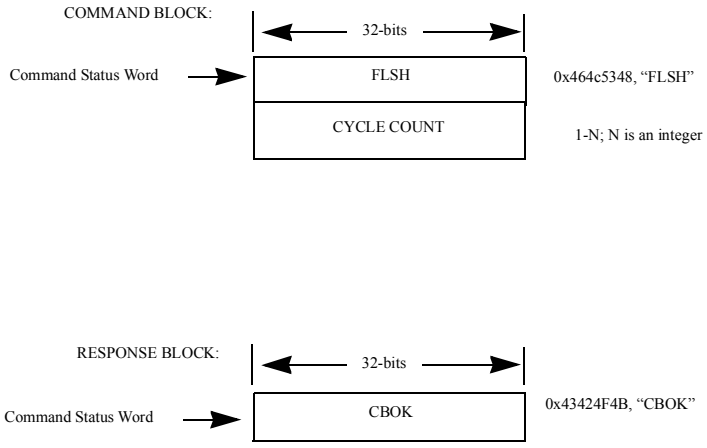


Figure 4-5. FLSH Command and Response Format

FUNCTION

Flash the LED(s).

CODE

0x464C5348 (ASCII "FLSH")

PARAMETERS

- CYCLE COUNT

  The number of times an LED red to green transition occurs.

PROGRAMMING SEQUENCE

If CBOK, then supply the FLASH cycle parameter, then supply the FLSH command. When CBOK occurs, FLSH is finished.

NEVER FAILS

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

GRNL

Group: Common Boot

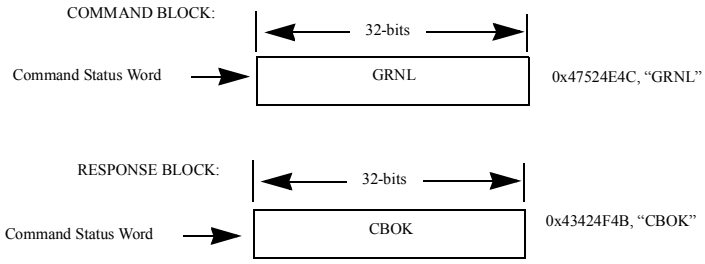


Figure 4-6. GRNL Command and Response Format

FUNCTION

Turn on the green LED

CODE

0x47524E4C (ASCII “GRNL”)

PARAMETERS

None required

PROGRAMMING SEQUENCE

If CBOK, supply the GRNL command. When CBOK occurs or the LED turns green, GRNL is finished.

NEVER FAILS

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

JUMP

Group: Common Boot

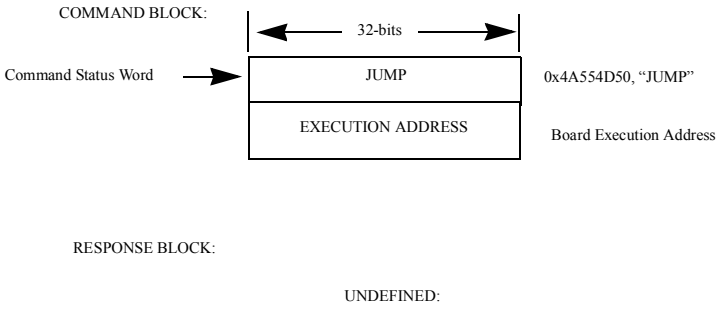


Figure 4-7. JUMP Command and Response Format

FUNCTION

Transfer execution to address.

CODE

0x4A554D50 (ASCII "JUMP")

PARAMETERS

- EXECUTION ADDRESS

  Board execution address.

PROGRAMMING SEQUENCE

If CBOK, supply the target execution address, then supply the JUMP command.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

PEEK

Group: Common Boot

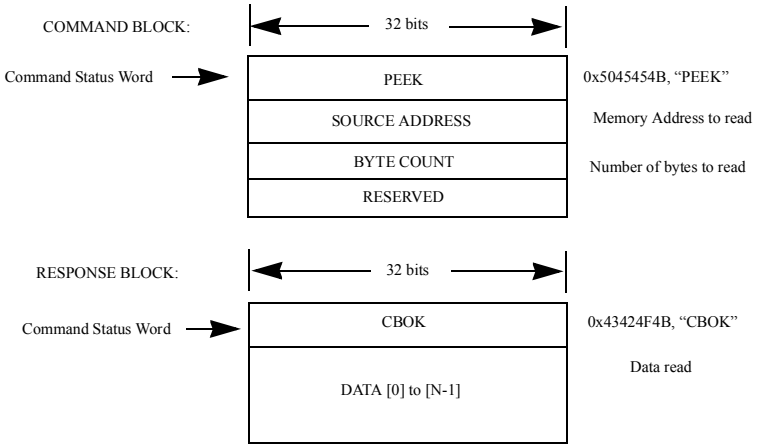


Figure 4-8. PEEK Command and Response Format

FUNCTION

Examine controller memory.

CODE

0x5045454B (ASCII "PEEK")

PARAMETERS

- SOURCE ADDRESS

  Board memory address to read.
- BYTE COUNT

  Number of units to read.
- RESERVED

  This field is reserved and should be 0.

PROGRAMMING SEQUENCE

If CBOK, supply the parameters, then supply the PEEK command. When CBOK occurs, PEEK is finished and memory will be dumped at the location immediately following CB_START + 4

NEVER FAILS

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

POKE

Group: Common Boot

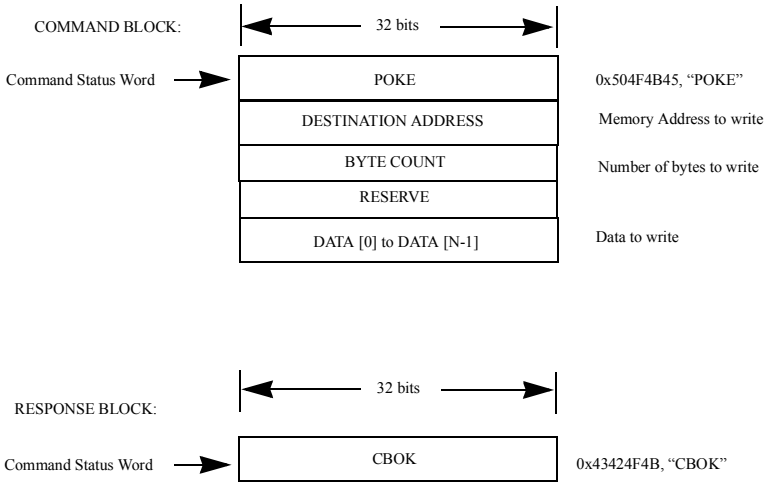


Figure 4-9. POKE Command and Response Format

FUNCTION

Modify controller memory. For an application of this command see *DOWNLOADING CODE on page 74*.

0x504F4B45 (ASCII "POKE")

PARAMETERS

- DESTINATION ADDRESS

  Where data is to be written.
- BYTE COUNT

  Number of bytes to write.
- RESERVED

  This field is reserved and should be 0.

PROGRAMMING SEQUENCE

If CBOK, supply the parameters, then supply the POKE command. The data to be poked immediately follows the RESERVED field. When CBOK occurs, POKE is finished and written memory exists on the controller at the destination address.

NEVER FAILS

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

DNLD

Group: Common Boot

The CB Download command DNLD allows the host to download new code to the controller.

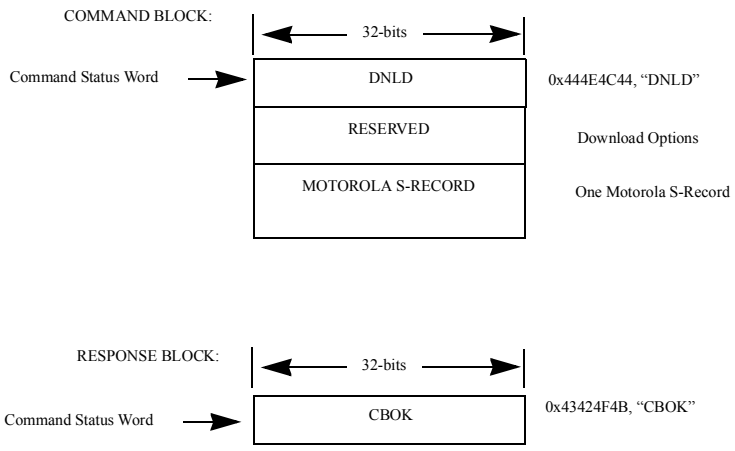


Figure 4-10. DNLD (DOWNLOAD) Command and Response Format

FUNCTION

This command allows the host to download Material S-Records, one record at a time, into the units download memory. The function will track the SD-Record type (0-9), will convert the input from ASCII to binary format, will checksum each record, will track if the image being downloaded will download the image into buffer RAM with the appropriate offset. Note: The function will do an overall checksum of the header image and keep a record of the expected program image size and checksum.

CODE

0x444E4C44 (ASCII "DNLD")

PARAMETERS

None

MOTOROLA S_RECORD

This entry consists of a single Motorola S-Record. Record types S0, S1, S2, S3, S7, S8 and S9 are supported.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

PROGRAMMING SEQUENCE

If CBOK, supply the parameters, then supply the DNLD command. The S-Record to be downloaded immediately follows the RESERVED field. When CBOK occurs, DNLD is finished and the next DNLD command can be issued until all the S-Records are downloaded.

If the data was downloaded with the FLASH option, the HOST may issue an BOOT 1 to start executing the code or a BURN command to save the code in FLASH EPROM.

ERROR RETURNS

If FAIL, the CB ERROR status block will be filled in

CB_SREC_BAD_TYPE(0x01)	CB_IMG_CHKSUM(0x02)
CB_SREC_CHKSUM(0x03)	CB_BUFFER_OVERRUN(0x04)
CB_IMG_OVERRUN(0x05)	CB_BAD_SORE(0x06)
CB_BAD_LENGTH(0x08)	CB_NOT_SREC(0x09)
IHE_DHRCHKSUM(0x17)	CB_DNLD_SEQC(0x23)

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

BURN

Group: Common Boot

The CB BURN command is used to update FLASH EPROM with a down loaded image that was saved in RAM using the CB DNLD command.

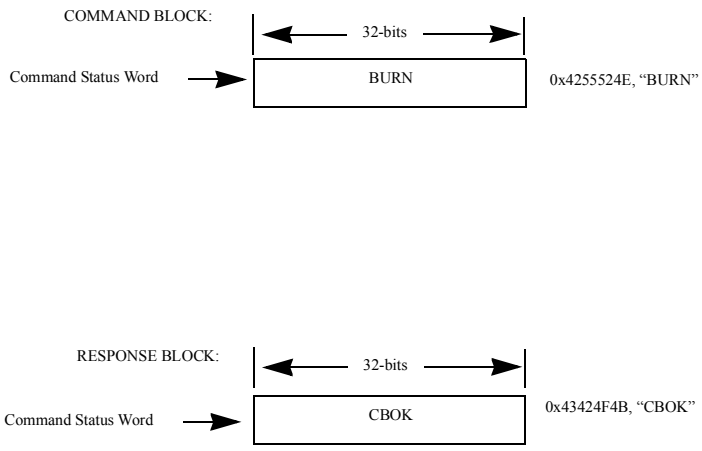


Figure 4-11. BURN Command and Response Format

FUNCTION

This function allows for the updating of FLASH EPROM with new program images.

CODE

0x4255524E (ASCII “BURN”)

PROGRAMMING SEQUENCE

If CBOK, supply the parameters, then the BURN command. The data to be burned into FLASH EPROM should have been previously downloaded using the CB DNLD or CB POKE command.

ERROR RETURNS

If FAIL, the CB ERROR status block will be filled in. See *CB ERROR CODES on page 217* for a table of CB Error Codes.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

REDL

Group: Common Boot

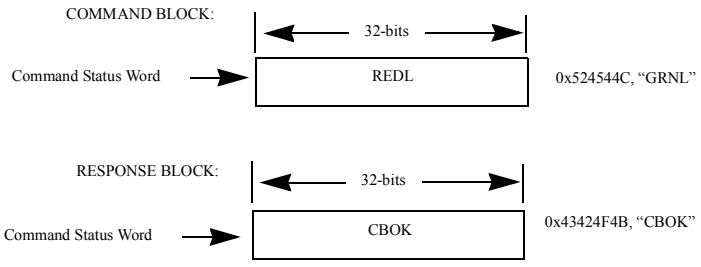


Figure 4-12. REDL Command and Response Format

FUNCTION

Turn on the red LED (if present on controller)

CODE

0x5245444C (ASCII "REDL")

PARAMETERS

None

PROGRAMMING SEQUENCE

If CBOK, then supply the REDL command. When CBOK occurs or the LED turns red, REDL is finished.

NEVER FAILS

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

STUF

Group: Common Boot

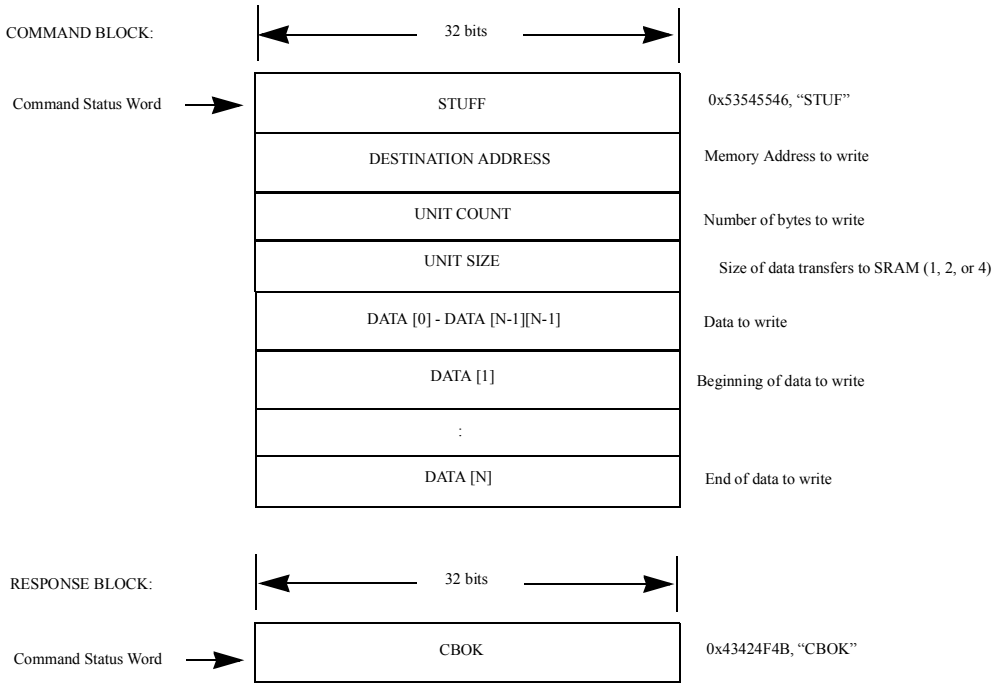


Figure 4-13. STUF Command and Response Format

FUNCTION

Modify controller memory by stuffing multiple data units at the same destination address. This routine is used to stuff data into a single, non-incrementing address location. The data is transferred into the address location in the size requested. The data to be written is read from shared memory in the format of (byte, word, or longword) the parameter UNIT SIZE.

CODE

0x53545546 (ASCII "STUF")

PARAMETERS

- DESTINATION ADDRESS

  Where data is written. This address does not increment. This value is an absolute address.

- UNIT COUNT

  Number of units to write. The value placed in UNIT COUNT should not exceed the size in Longword

- UNIT SIZE

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Size in bytes of units (1, 2, or 4).

PROGRAMMING SEQUENCE

If CBOK, supply the parameters and then supply the STUF command. The data to be stuffed immediately follows the UNIT SIZE field. When CBOK occurs, STUF is finished and the specified data exists on the controller at the destination address.

ERROR RETURNS

If FAIL, the CB ERROR status block will be filled in.
BAD_SIZE (0x0B)

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

HGET

Group: Common Boot

The CB HGET command is used to examine a series of address locations.

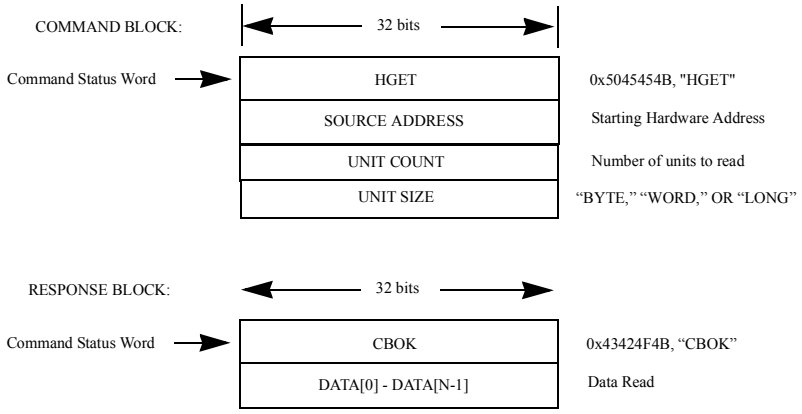


Figure 4-14. HGET Command and Response Format

FUNCTION

Examine a series of address locations.

The function simply transfers data from the source addresses to the addresses of the shared memory data fields. Both the shared memory and the source address are accessed and incremented as given by the UNIT SIZE input variable.

CODE

0x5045454B (ASCII "HGET")

PARAMETERS

- SOURCE ADDRESS

  Where data is to be read from.

- UNIT COUNT

  Number of units to read.

- UNIT SIZE

  "BYTE", "WORD", or "LONG"

  This parameter determines the width in bytes of the CPU reads to the controller hardware
  ("BYTE"(8 bits), "WORD"(16 bits), or "LONG"(32 bits)).

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

It is represented as a 32 bit ASCII value. Byte 0x42595445, “WORD” 0x574F5244, “LONG” 0x4C4F4E47.

EXAMPLE

For memory source with the first for longword addresses containing the following: A0=0x00112233 A1=0x44556677 A2=0x8899AABB A3=0xCCDDEEFF.

Assuming a UNIT COUNT = 4 and a VMEbus address of COMMAND STATUS WORD = 0xff4080 the following results will appear in the controller’s short I/O memory:

- UNIT SIZE = “BYTE”

  0xff4084 = 0x00 0xff4085 = 0x11 0xff4086 = = 0x22 0xff4087 = 0x33
- UNIT SIZE = “WORD”

  0xff4087 = 0x0011 0xff4086 = 0x2233 0xff4088 = 0x4455 0xff408a = 0x6677
- UNIT SIZE = “LONG”

  0xff4084 = 0x00112233 0xff4088 = 0x44556677 0xff408c = 0x8899AABB 0xff490 0xCCDDEEFF

PROGRAMMING SEQUENCE

If CBOK, supply the parameters, then supply the HGET command.

ERROR RETURNS

IF FAIL, the CB ERROR status block will be filled in.
BAD_SIZE(0x0B)

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

HPUT

GROUP: Common Boot

The CB HPUT command is used to modify a series of address locations.

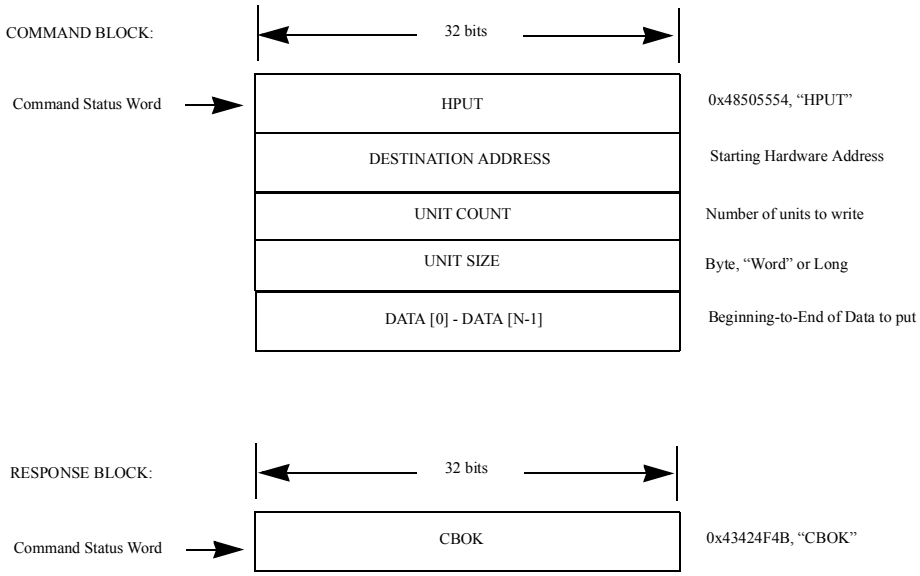


Figure 4-15. HPUT Command and Response Format

FUNCTION

Modify a series of address locations.

The function simply writes to the destination address the contents of the shared memory data fields. Both the shared memory and the source address are accessed and incremented as given by the UNIT SIZE input variable.

CODE

0x48505554 (ASCII "HPUT")

PARAMETERS

- DESTINATION ADDRESS

  Starting address where data is to be written.
- UNIT COUNT

  Number of units to write.
- UNIT SIZE

  "BYTE," "WORD," or "LONG"

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

This parameter determines the width in bytes of the CPU writes to the controller hardware (Byte (8 bits), Word (16 bits), or Long (32 bits)).

It is represented as a 32 bit ASCII value. Byte 0x42595445, Word 0x574F5244, Long 0x4C4F4E47.

EXAMPLE

For a HPUT command with the first four longword data parameters containing the following: D0 = 0x00112233 D1 = 0x44556677 D2 = 0x8899AABB D3 = 0xCCDDEEFF.

Assuming a UNIT COUNT = 4 and a DESTINATION ADDRESS of 0x0c00000 the following results will appear in the controller's memory:

- UNIT SIZE = “BYTE”

  xc00000 = 0x00 0xc00001 = 0x11 0xc00002 = 0x22 0xc00003 = 0x3.

- UNIT SIZE = “WORD”

  0xc00000 = 0x0011 0xc00002 = 0x2233 0xc00004 = 0x4455 0xc00006 = 0x6677.

- UNIT SIZE =”LONG”

  0xc00000 = 0x00112233 0xc00004 = 0x44556677 0xc00008 = 0x8899AABB 0xc0000c = 0xCCDDEEFF.

PROGRAMMING SEQUENCE

If CBOK, supply the parameters, then supply the HPUT command. The data to be put immediately follows the UNIT SIZE field.

ERROR RETURNS

IF FAIL, the CB ERROR status block will be filled in.
BAD_SIZE(0x0B)

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

INFO

The CB board information command (INFO) allows the host to identify the controller.

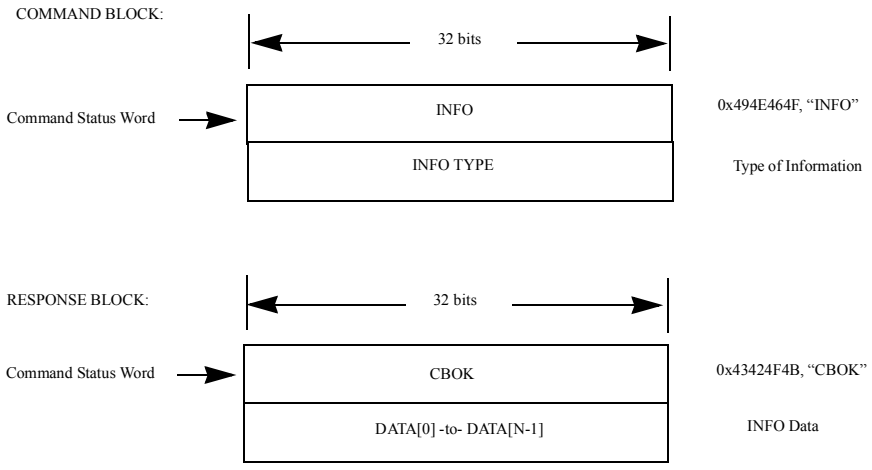


Figure 4-16. Information Command (INFO) Format

FUNCTION

Function returns board identification, boot image information or controller specific information.

CODE

0x42444944 (ASCII "INFO")

PARAMETERS

- INFO TYPE:Specifies which information the controller is to make available to the host.

  0 **CB information:** Setting this value to 0 tells the controller to return CB information.

  1 **EXEC 0 ID:** Setting this value to 1 tells the controller to return the EXEC 0 header information.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

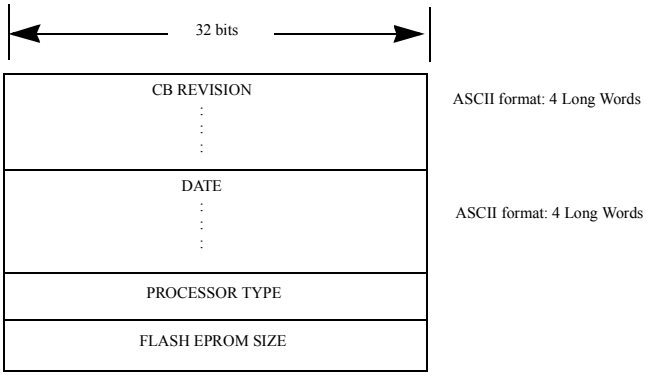


Figure 4-17. CB Information Structure Type 0

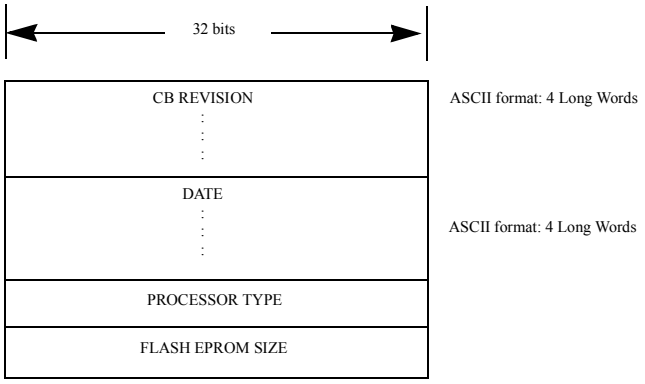


Figure 4-18. EXEC 0 Information Structure Type 1

PROGRAMMING SEQUENCE

If CBOK, supply the INFO TYPE; then supply the INFO command. When CBOK occurs, the host may examine the information.

ERROR RETURNS

IF FAIL, the CB ERROR status block will be filled in.
BAD_SIZE(0x0B)

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

REPORT/COMMAND (RC) COMMANDS

This section describes the RC Commands. The RC Interface has a generic command set, plus extensions which are specific to your controller. This chapter documents the generic RC commands.

As discussed in SYSTEM REQUIREMENTS in the previous chapter, the byte ordering of commands must be “big-endian” (that is, data is read/written Most Significant Byte first). Table 4-2 shows the generic RC command set.

Table 4-2. RC Command Set

Requests And Directives			
Command ID	**Name**	**Type**	**Page # †**
0x0	(Not Used; invalid)	-	-
0x1	Set DMA Burst	Directive	101
0x2	Set Bus Timeout	Directive	103
0x3	Request Statistics	Request	104
0x4	Set Interrupt	Directive	105
0x5	Create Host-to-Board Channel	Request	106
0x6	Create Board-to-Host Channel	Request	108
0x7	Link Host-to-Board Segment	Directive	109
0x8	(reserved)	-	-
0x9	Allocate Board-to-Host Segment Memory	Directive	110
Responses And Indications			
0x80	(Not used; invalid)	-	-
0x81	Report Printf Call	Indication	112
0x82	Error Message	Response	113
0x83	Report Statistics	Response	115
0x84	Report New Host-to-Board Channel	Response	117
0x85	Report New Board-to-Host Channel	Response	117
0x86	Report Link to Next Segment	Indication	118

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

TYPES OF RC COMMANDS

The RC command set is divided into four groups – requests, directives, responses, and indications.

Requests and **directives** are issued from the host to the controller. “Requests” have “responses” associated with them (that is, a reply in the board-to-host direction). “Directives,” on the other hand, have no associated reply.

Responses and **indications** are passed from the controller to the host. “Responses” are paired with requests that are host-to-board initiated. “Indications” are asynchronous events which have no pre-initiated host-to-board trigger.

The four RC command types are summarized in Table 4-3:

Table 4-3. Types of RC Commands

Type	Description	Type
Request	A Host-to-Board command that requires a response from the controller as a handshake.	Response
Directive	A Host-to-Board command that does not require a response from the controller.	None
Response	A structure passed from the controller to the host in response to a host-initiated request.	None
Indication	A structure passed from the controller to the host reporting unsolicited information or status.	None

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

SET DMA BURST

Group: Report/Command
Type: Directive
Command ID: 0x1

This directive programs the transfer counter which meters the number of bytes that go out per DMA burst. The actual number of bytes transferred in a given DMA burst is determined as follows:

BYTES PER BURST = (COUNT x TSIZE) ÷ 8

MNEMONIC:

COUNT = "Count Value" programmed using Set DMA Burst directive, or the controller's default transfer count

TSIZE = Transfer size (e.g. 16 bits)

For example, the V/FDDI 5211 supports 8-bit, 16-bit, 32-bit and 64-bit DMA transfers and has a 10-bit transfer counter. Using the default transfer count (256), the DMA burst sizes are as follows:

- Burst size (8-bit transfers) = (256 * 8) ÷ 8 = 256 bytes per burst
- Burst size (16-bit transfers) = (256 * 16) ÷ 8 = 512 bytes per burst
- Burst size (32-bit transfers) = (256 * 32) ÷ 8 = 1024 bytes per burst
- Burst size (64-bit transfers) = (256 * 64) ÷ 8 = 2048 bytes per burst

If the host does *not* issue this directive (or if it issues the directive with a Count Value of 0), the controller will use its default transfer count. See COUNT VALUE on the following page.

After completing a DMA burst, the controller relinquishes the bus for the use of other devices on the bus. The controller's DMA engine will attempt to regain control of the bus as soon as possible in order to do another burst.



NOTE

Although the 5211 will hold the bus for the entire transfer count, the hardware will de-assert VME address strobe (*VAS) between memory page boundaries as required by the VMEbus specifications.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The format of the Set DMA Burst Directive is shown in Figure 4-19.

Size (in bytes)	Description
8	CCB
4	Count Value

Figure 4-19. Format of Set DMA Burst Directive

The RC Interface does not pass a reply message to the host for this command.

FIELDS

The fields in the Set DMA Burst directive are as follows:

- COMMAND CONTROL BLOCK (8 bytes)

  This field contains the Command Control Block, as defined in the *Command Control Block Format on page 58*. The command ID must be 0x1 for the Set DMA Burst directive.

- **C**OUNT VALUE (4 bytes)

  This field sets the maximum number of transfers that the transfer counter will perform per DMA burst.

  The set of legal values in this field = (0 ... 0xFF) inclusive. A value of 0 in this field disables the burst. If burst is disabled the controller relinquishes the bus only after the entire DMA is complete.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

SET BUS TIMEOUT

Group: Report/Command
Type: Directive
Command ID: 0x2

The Set Bus Timeout directive sets the amount of time the controller will wait for a bus cycle to complete. If a bus cycle does not finish within this time, a DMA timeout message (error code 0x1) is sent to the host (see *ERROR MESSAGE on page 113*).

The RC Interface does not pass a reply message to the host for this command.

The format the Set Bus Timeout Directive is shown in Figure 4-20.

Size (in bytes)	Description
8	CCB
4	Number of Milliseconds

Figure 4-20. Format of Set Bus Timeout Directive

FIELDS

The fields in the Set Bus Timeout directive are as follows:

- COMMAND CONTROL BLOCK (8 bytes)

  This field contains the Command Control Block. The command ID must be set to 0x2 for the Set Bus Timeout directive.

- NUMBER OF MILLISECONDS (4 bytes)

  This field sets the number of milliseconds the controller should wait for the bus cycle to complete. If this field is 0, the controller will use the default timeout value encoded in its firmware. The default value is 100 milliseconds.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

REQUEST STATISTICS

Group: Report/Command
Type: Request
Command ID: 0x3

Request Statistics causes the controller to pass back various firmware parameters, including the firmware's version number, release date, and memory size.

In reply to this command, the RC Interface generates a Report Statistics response. For a general description of this response, see *REPORT STATISTICS on page 115*.

The format of the Request Statistics request is shown in Figure 4-21.

Size (in bytes)	Description
8	CCB
4	ID # of Response Channel
4	Command Tag

Figure 4-21. Format of Request Statistics

FIELDS

The fields in Request Statistics are as follows:

- COMMAND CONTROL BLOCK (8 bytes)

  This field contains the Command Control Block. The command ID must be set to 0x3 for Request Statistics.

- ID # OF RESPONSE CHANNEL (4 bytes)

  This field contains the Channel ID of the board-to-host channel to which the controller should write the statistics. For a discussion of channel IDs, refer to *Channel ID Fields on page 59*.

- COMMAND TAG (4 bytes)

  This field may contain a host-generated command tag, if needed for the application. Command tags are typically generated by the host's device driver and can be used to notify host processes associated with a command. The RC Interface does not use the command tag in any way. It simply returns the value in the Report Statistics response.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

SET INTERRUPT

Group: Report/Command
Type: Directive
Command ID: 0x4

As discussed in the section on Interrupts in the previous chapter, the controller may be directed to interrupt the host for a particular channel when certain conditions are met.

The Set Interrupt directive is used to assign a bus interrupt level and vector to a specific board-to-host channel. Different board-to-host channels can have different interrupt vectors and/or levels. This provides a mechanism for separating different classes of responses.

The controller will not interrupt the host for any reason until this directive has been issued to the RC Interface. The format of the Set Interrupt Directive is shown in Figure 4-22.

Size (in bytes)	Description
8	CCB
4	Target Channel ID
4	Interrupt Level
4	Interrupt Vector

Figure 4-22. Format of Set Interrupt Directive

FIELDS

The fields in the Set Interrupt directive are as follows:

- COMMAND CONTROL BLOCK (8 bytes)

  This field contains the Command Control Block. The command ID must be set to 0x4 for the Set Interrupt directive.

- TARGET CHANNEL ID (4 bytes)

  This field identifies the board-to-host channel to which the interrupt level and vector are being assigned. For a discussion of Channel IDs, refer to *Channel ID Fields on page 59*.

- INTERRUPT LEVEL (4 bytes)

  This field sets the bus interrupt level used by the controller to notify the host that it has completed command(s) in a given board-to-host channel (where the board-to-host channel is specified in the “Target Channel ID” field). Valid entries in this field are 0x1 through 0x7.

INTERRUPT VECTOR (4 bytes)

This field sets the bus interrupt vector that the controller will use to report command completion.

The legal entries in this field are 0x0 through 0xFF.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

CREATE HOST-TO-BOARD CHANNEL

Group: Report/Command
Type: Request
Command ID: 0x5

Create Host-to-Board Channel allows the host to create a new host-to-board channel. This command is issued only by the host, and has the effect of determining who owns which read/write pointers in the channel descriptor.

In reply to this command, the RC Interface: 1) assigns an ID to the channel, and 2) returns the new ID via a Report New Host-to-Board Channel response (see *REPORT NEW BOARD-TO-HOST CHANNEL on page 117*). The returned channel ID is the byte offset from the shared memory base address to the channel descriptor for the new channel. The host will update the write pointer of the new channel descriptor, and the controller will manage the read pointer. The format of the Create Host-to-Board Channel Request is shown in Figure 4-23.

Size (in bytes)	Description
8	CCB
4	ID # of Response Channel
4	Command Tag
4	Transfer Options of First H2B Segment
4	Physical Address of First H2B Segment
4	Length in bytes of First H2B Segment

Figure 4-23. Format of Create Host-to-Board Channel Request

FIELDS

The fields in the Create Host-to-Board Channel request are as follows:

- COMMAND CONTROL BLOCK (8 bytes)

  This field contains the Command Control Block. The command ID must be set to 0x5 for Create Host-to-Board Channel.

- ID # OF RESPONSE CHANNEL (4 bytes)

  This value is the ID of the board-to-host channel to which the controller should write the Report New Host-to-Board Channel response.

- COMMAND TAG (4 bytes)

  This field may contain a host-generated command tag, if needed for the application. Command tags are typically generated by the host's device driver and can be used to notify host processes associated with a command. The RC Interface does not use the command tag in any way. It simply returns the value in the Report New Host-to-Board Channel response.

- TRANSFER OPTIONS OF FIRST HOST-TO-BOARD (H2B) SEGMENT (4 bytes)

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

This field contains the Transfer Options Word, as defined in *Transfer Options Word on page 59*. The controller uses this information to perform DMA accesses on the ***first*** segment in the channel. The transfer characteristics of subsequent segments are given when the host spans to each new segment. (This is specified using the Transfer Options Word in the Link Host-to-Board Segment directive, see *LINK HOST-TO-BOARD SEGMENT on page 109*)

- PHYSICAL ADDRESS OF FIRST HOST-TO-BOARD (H2B) SEGMENT (4 bytes)

  This field provides the controller with the address by which the onboard DMA mechanism can reach the first segment. This address may very well be different from the logical or virtual address by which host-resident software accesses the memory.

- LENGTH IN BYTES OF FIRST H-TO-BOARD (H2B) SEGMENT (4 bytes)

  This field contains the number of bytes in the channel's first segment. The controller uses this value in cases where the "Write Segment #" field in the channel descriptor increments past the first segment before that segment has been processed by the controller.

  The physical addresses and byte lengths of subsequent segments in the channel are provided to the controller using the Link Host-to-Board Segment directive.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

CREATE BOARD-TO-HOST CHANNEL

Group: Report/Command
Type: Request
Command ID: 0x6

Create Board-to-Host Channel allows the host to create a new board-to-host channel. It is issued only by the host, and has the effect of determining who owns which read/write pointers in the channel descriptor.

In reply to this command, the RC Interface: 1) assigns an ID to the channel, and 2) returns the new ID via a Report New Board-to-Host Channel response (see *REPORT NEW BOARD-TO-HOST CHANNEL on page 117*). The returned channel ID is the byte offset from the start of shared memory to the channel descriptor for the new channel. The controller will update the write pointer of the new channel descriptor, and the host will manage the read pointer. The format of the Create Board-to-Host Channel Request is shown in Figure 4-24.

Size (in bytes)	Description
8	CCB
4	ID # of Response Channel
4	Command Tag

Figure 4-24. Format of Create Board-to-Host Channel Request

FIELDS

The fields in the Create Board-to-Host Channel request are as follows:

- COMMAND CONTROL BLOCK (8 bytes)

  This field contains the Command Control Block. The command ID must be set to 0x6 for Create Board-to-Host Channel.

- ID # OF RESPONSE CHANNEL (4 bytes)

  This value is the ID of the board-to-host channel to which the controller should write the Report New Board-to-Host Channel response.

- COMMAND TAG (4 bytes)

  This field may contain a host-generated command tag, if needed for the application. Command tags are typically generated by the host's device driver and can be used to notify host processes associated with a command. The RC Interface does not use the command tag in any way. It simply returns the value in the Report New Board-to-Host Channel response.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

LINK HOST-TO-BOARD SEGMENT

Group: Report/Command
Type: Request
Command ID: 0x7

The Link Host-to-Board Segment directive is used to link one host-to-board segment to another. It instructs the controller to begin drawing commands from a different block of memory. This directive is always the **last** command the host places in a host-to-board channel segment.

The format of the Link Host-to-Board Segment Directive is shown in Figure 4-25.

Size (in bytes)	Description
8	CCB
4	Transfer Options Word
4	Physical Address of Next Segment
4	Length of Next Segment in Bytes

Figure 4-25. Format of Link Host-to-Board Segment Directive

FIELDS

The fields in the Link Host-to-Board Segment directive are as follows:

- COMMAND CONTROL BLOCK (8 bytes)

  This field contains the Command Control Block. The command ID must be set to 0x7 for the Link Host-to-Board Segment directive.

- TRANSFER OPTIONS WORD (4 bytes)

  This field contains the Transfer Options Word, as defined in chapter 3. It provides information required in order for the controller to have DMA access to the next segment in this host-to-board channel.

- PHYSICAL ADDRESS OF NEXT SEGMENT (4 bytes)

  This field contains the DMA-able byte address of the start of the next segment. The physical address of the segment may be different from the virtual address to which the host software writes.

- LENGTH OF NEXT SEGMENT IN BYTES (4 bytes)

  This field gives the number of total bytes in the next segment. This information is needed by the controller in order to DMA the segment into onboard memory.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

ALLOCATE BOARD-TO-HOST SEGMENT MEMORY

Group: Report/Command
Type: Directive
Command ID: 0x9

This directive supplies the controller with a list of host DMA buffer resources for board-to-host channel segments. The segments are not assigned by the host to any particular channel. Each segment is identified by a pair of addresses. The first is the physical address for board-to-host DMA purposes. The second address is the virtual, or "logical" address by which the host will access the segment. The controller uses this virtual address to tell the host the location of the next segment in a board-to-host channel. It does so by placing the address into a Report Link to Next Segment indication, which the host uses to span from one channel segment to another. For details on Report Link to Next Segment, see *REPORT LINK TO NEXT SEGMENT on page 118*.

Allocate Board-to-Host Segment Memory must be issued as a parameter to "BOOT," which is the Common Boot command used to boot the RC Interface. This is necessary so that the board-to-host segment memory is in place after RC initialization. For details, see *BOOT on page 75*.

The host must allocate ***at least two*** channel segments to the controller for ***each*** board-to-host channel to be created. The controller enforces this rule when it processes requests to create new board-to-host channels. If there is insufficient segment memory available to the controller when it encounters such a request, it will respond with an error message.

This directive may be issued again by the host if it appears that the board is bottlenecking on channel segment memory. If the board does not have enough segment memory to pass responses or indications to the host as they are generated, it will buffer them up in board firmware until segment memory becomes free.

The format of the Allocate Board-to-Host Segment Memory is shown in Figure 4-26:

Size (in bytes)	Description
8	CCB
4	Transfer Options Word (Same for all segments)
4	Segment Length (Same for all segments)
4	Physical Address of First Segment
4	Command Tag or Virtual Address of First Segment
:	:
4	Physical Address of Segment N
4	Command Tag or Virtual Address of Segment N

Figure 4-26. Format of Allocate Board-to-Host Segment Memory

FIELDS

The fields in the Allocate Board-to-Host Segment Memory directive are as follows:

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

- COMMAND CONTROL BLOCK (8 bytes)

  This field contains the Command Control Block. The command ID must be set to 0x9 for the Allocate Board-to-Host Segment Memory directive.

- TRANSFER OPTIONS WORD (4 bytes)

  This field contains the Transfer Options Word, as defined in chapter 3. It provides information required in order for the controller to have DMA access to the board-to-host channel segments. All segments in a given board-to-host channel must have the same transfer characteristics.

- SEGMENT LENGTH (4 bytes)

  This field contains the length of the segments in bytes. All segments must have the same size. The maximum legal value in this field is 65,534 bytes (0xFFFE).

- PHYSICAL ADDRESS OF SEGMENT *n* (4 bytes)

  This field provides the controller with the DMA-able byte address of the start of segment n. The physical address of the segment may be different from the virtual address to which the host software writes.

- COMMAND TAG OR VIRTUAL ADDRESS OF SEGMENT *n* (4 bytes)

  This field contains a value which enables the host to identify segment ***n*** when it is used by the controller. The controller reports this value when it links to segment **n** (see *REPORT LINK TO NEXT SEGMENT on page 118*).

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

REPORT PRINTF CALL

Group: Report/Command
Type: Indication
Command ID: 0x81

Report Printf Call is delivered to the host whenever an on-board software module calls the *printf()* routine. It is typically used to report boot-time configuration information and non-fatal anomalies. It can also provide debugging information which is useful in the development of firmware or downloaded software. The body of this message is a null-terminated string that should be treated as a status message for the operator/user.

The controller writes Report Printf Call to the default response channel (that is, the initial board-to-host channel created at RC boot time). The format of the Report Printf Call is shown in Figure 4-27:

Size (in bytes)	Description
8	CCB
N	Variable Length ASCII String
1	Null Termination (Zero-Valued Byte)
0-3	Padding to a longword boundary

Figure 4-27. Format of Report Printf Call

FIELDS

The fields in Report Printf Call are as follows:

- COMMAND CONTROL BLOCK (8 bytes)

  This field contains the Command Control Block. The controller sets the Command ID field to 0x81 for Report Printf Call.

- VARIABLE-LENGTH ASCII STRING

  This field contains a variable-length, null-terminated string that should be treated as a status message for the operator/user. It is followed by 0 to 3 bytes to pad the message to a longword boundary.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

ERROR MESSAGE

Group: Report/Command
Type: Indication
Command ID: 0x82

The controller issues an Error Message indication when a condition arises that prevents completion of a host-issued command. It writes the indication to the default response channel (that is, the initial board-to-host channel created at RC boot time). The format of the Error Message is shown in Figure 4-28.

Size (in bytes)	Description
8	CCB
4	Error Code
8	CCB in Error
N	Variable Length ASCII String
1	Null Termination (Zero-Valued Byte)
0-3	Padding to a longword boundary, or an extra word of 0's if the string ends on a longword boundary

Figure 4-28. Format of Error Message Indication

FIELDS

The fields in the Error Message indication are as follows:

- COMMAND CONTROL BLOCK (8 bytes)

  This field contains the Command Control Block. The controller sets the Command ID field to 0x82 for Error Message.

- ERROR CODE (4 BYTES)

  This value indicates the type of error that has occurred. Table 4-4 shows the possible value which may occur in this field, as well as a brief description of each error.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

ERROR CODES

Table 4-4. Error Codes

Code	Description
0x1	The controller timed out during a DMA transfer. This usually indicates a bad physical address. The starting address of the transfer that failed is given in the error string.
0x2	An internal error has occurred on the controller.
0x3	The command identified in the “CCB in Error” field contains invalid parameter(s)
0x4	The controller made an unsuccessful malloc attempt. †
0x5	This command is not implemented
0x6	The host has specified an unknown command.
0x7	The controller’s non-volatile RAM (NOVRAM) has failed.
0x8	A DMA bus error has occurred. The controller tried to DMA data to a location that is “not there” (from the controller’s perspective). This indicates either a bad address, or failure by the host to map the memory properly for DMA.
0x9	The controller tried to allocate an mblk but was unable to do so. This means that the on-board static RAM is full.
0xB	A byte-alignment error has taken place. For information on how the controller expects data to be aligned, (see SYSTEM REQUIREMENTS in chapter 3), as well as the Functional Overview and Specifications chapters in your controller manual.
0xD	The controller has run out of available board-to-host segments. To handle this situation, the host can take either (or both) of the following actions: 1.) The host can free up “controller-owned” segments by extracting data from the board-to-host channel(s) more rapidly. 2.) The host can give the board more segment memory (*ALLOCATE BOARD-TO-HOST SEGMENT MEMORY on page 110*)
0xE	The controller made an unsuccessful alloc attempt. †
0xF	The controller made an unsuccessful salloc attempt. †
0x10	The controller has run out of channel space. This means that there is no room left in shared memory to write new channel descriptors.

FIELDS

- CCB IN ERROR (8 BYTES)

  This field contains the CCB of the command which caused the error.

- VARIABLE-LENGTH ASCII STRING

  This field contains a variable-length, null-terminated string describing the error. It is followed by 0 to 3 bytes (if needed) to pad the message to a longword boundary. Or, if the string ends on a longword boundary, it is followed by an extra word of zeros.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

REPORT STATISTICS

Group: Report/Command
Type: Response
Command ID: 0x83

The controller issues Report Statistics in response to a host-issued Request Statistics. It provides information on the specific version of firmware on your controller. The first two values following the Command Control Block will always be the firmware ID and release date.The format of the Report Statistics Response is shown in Figure 4-29.

Size (in bytes)	Description
8	CCB
4	Command Tag
16	Firmware ID (ASCII)
16	Release date (ASCII)
4	Buffer RAM Size in Bytes
4	Static RAM Size in Bytes

Figure 4-29. Format of Report Statistics Response

FIELDS

The fields in the Report Statistics response are as follows:

- COMMAND CONTROL BLOCK (8 bytes)

  This field contains the Command Control Block. The controller sets the Command ID field to 0x83 for Report Statistics.

- COMMAND TAG (4 bytes)

  This value is the host-supplied command tag (if any) contained in the command to which the controller is responding.

- FIRMWARE ID (16 bytes)

  This field contains a null terminated ASCII string which identifies the ID number of the firmware.

- RELEASE DATE (16 bytes)

  This field contains a null-terminated ASCII string stating the release date of the firmware.

- BUFFER RAM SIZE IN BYTES (4 bytes)

  This is the size of the dynamic buffer RAM.

- STATIC RAM SIZE IN BYTES (4 bytes)

  This is the size of the static buffer RAM.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

REPORT NEW HOST-TO-BOARD CHANNEL

Group: Report/Command
Type: Response
Command ID: 0x84

Report New Host-to-Board Channel is generated in response to a Create Host-to-Board Channel request. The format of the response is shown in Figure 4-30.

Size (in bytes)	Description
8	CCB
4	Command Tag
4	ID # of Created Channel

Figure 4-30. Format of Report New Host-to-Board Channel

FIELDS

The fields in the Report New Host-to-Board Channel response are as follows:

- COMMAND CONTROL BLOCK (8 bytes)

  This field contains the Command Control Block. The controller sets the Command ID field to 0x84 for Report New Host-to-Board Channel.

- COMMAND TAG (4 bytes)

  This value is the host-supplied command tag (if any) contained in the command to which the controller is responding.

- ID # OF CREATED CHANNEL (4 bytes)

  This field contains ID number of the newly created channel.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

REPORT NEW BOARD-TO-HOST CHANNEL

Group: Report/Command
Type: Response
Command ID: 0x85

Report New Board-to-Host Channel is generated in response to a Create Board-to-Host Channel request. The format of the response is shown in Figure 4-31:

Size (in bytes)	Description
8	CCB
4	Command Tag
4	ID # of Created Channel
4	Virtual Address of Initial Segment
4	Length in Bytes of Initial Segment

Figure 4-31. Format of Report New Host-to-Board Channel

FIELDS

The fields in the Report New Host-to-Board Channel response are as follows:

- COMMAND CONTROL BLOCK (8 bytes)

  This field contains the Command Control Block. The controller sets the Command ID field to 0x84 for Report New Host-to-Board Channel.

- COMMAND TAG (4 bytes)

  This value is the host-supplied command tag (if any) contained in the command to which the controller is responding.

- ID # OF CREATED CHANNEL (4 bytes)

  This field contains ID number of the newly created channel.

- VIRTUAL ADDRESS OF INITIAL SEGMENT (4 bytes)

  This field contains the logical address at which the initial segment in the channel begins. This address was originally generated by the host and passed to the controller via the Allocate Board-to-Host Segment Memory directive.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

REPORT LINK TO NEXT SEGMENT

Group: Report/Command
Type: Indication
Command ID: 0x86

Report Link to Next Segment is issued by the controller when it needs to span to the next segment in a board-to-host channel. This indication tells the host to start drawing controller-generated responses/indications for this channel from a different block of memory.

This indication is always the **last** command the controller places in a board-to-host channel segment. The format of the Link Board-to-Host Segment is shown in Figure 4-32:

Size (in bytes)	Description
8	CCB
4	Command Tag or Virtual Address of Next Segment
4	Length of Next Segment in Bytes

Figure 4-32. Format of Link Board-to-Host Segment Indication

FIELDS

The fields in Report Link to Next Segment are as follows:

- COMMAND CONTROL BLOCK (8 bytes)

  This field contains the Command Control Block. The controller sets the Command ID field to 0x86 for Report Link to Next Segment.

- COMMAND TAG OR VIRTUAL ADDRESS OF NEXT SEGMENT (4 bytes)

  This field identifies the segment to which the controller is linking. The value in this field was originally generated by the host and passed to the controller via the Allocate Board-to-Host Segment Memory directive. (See *ALLOCATE BOARD-TO-HOST SEGMENT MEMORY on page 110*.)

- LENGTH OF NEXT SEGMENT IN BYTES (4 bytes)

  This field gives the number of total bytes in the next segment. This information is provided to the host as a convenience – it may not be needed other than for sanity checking of segment boundaries. In most applications, the host simply reads RC commands from the beginning of the next segment until it either encounters another Link Board-to-Host directive, or determines that the read pointer matches the write pointer.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

FDDI-SPECIFIC COMMANDS

A summary of the command set for the 5211 is shown in Table 4-5.

Table 4-5. V/5211 FDDI-Specific Commands

REQUESTS AND DIRECTIVES			
Command ID	**Name**	**Type**	**See Page**
0x101	Transmit Raw Data	Request	121
0x101	Transmit Embedded Raw Data	Request	124
0x102	Transmit Datagram	Request	127
0x102	Transmit Embedded Datagram	Request	127
0x103	Set Up Receive Buffers	Directive	134
0x104	Request Station Address	Request	136
0x105	Set Station Address	Directive	137
0x106	Ring Control	Directive	139
0x107	Perform Maintenance	Directive	140
0x108	Get MIB Attribute	Request	142
0x109	Set MIB Attribute	Directive	144
0x10A	Set CAM Address	Request	148
0x10B	Clear CAM Address	Request	150
0x10C	Flush CAM Addresses	Directive	151
0x10D	SMT Trace	Directive	152
0x10E	Set Hardware Parameters	Directive	153
0x10F	Get Hardware Parameters	Request	157
0x110	Get CAM Address	Request	159
RESPONSES AND INDICATIONS			
0x181	Tx Response	Response	160
0x182	Rx Frame	Indication	162
0x183	(reserved)	-	-
0x184	Report Station Address	Response	164
0x185	Ring Status Indication	Indication	165
0x186	(reserved)	-	-
0x187	CAM Address Response	Response	166
0x188	MIB Attribute Response	Response	168
0x189	SMT Event Indication	Indication	170
0x18A	Report Hardware Parameters	Response	171

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table 4-5. V/5211 FDDI-Specific Commands (cont)

REQUESTS AND DIRECTIVES			
RESPONSES AND INDICATIONS			
0x18B	Report CAM Address	Response	-174

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

TRANSMIT RAW DATA

Group: FDDI
Type: Request
Command ID: 0x101

Transmit Raw Data is a non-FDDI-standard request. It provides a mechanism to pass one frame from host-to-fiber with absolutely no interpretation or analysis performed by the board. This ability is necessary in applications where the Logical Link Control (LLC) implementation is responsible for all FDDI frame construction and data encapsulation.

Data for the frame can be "gathered" from blocks of host memory that are not contiguous. The host typically writes the frame header in the first fragment and the actual data in subsequent fragments.

If necessary, the host may pad the frame header to align it on a word boundary. Your system requirements determine if padding is used. If used, it should not be included in the "Total # of Bytes for Transmission" and "Physical Address of Data Fragment" fields.

After executing Transmit Raw Data, the 5211 issues a Tx Response (see page 160) if it has been enabled to do so in the Request Class field.

The data structure of the Transmit Raw Data request is detailed Table 4-6.

Table 4-6. Transmit Raw Data Request

Size (in bytes)	Description
8	Command Control Block
4	ID number of Response Channel
4	Command Tag
4	Request Class
4	Transfer Options Word
4	Total number of bytes for transmission
4	Number of fragments for transmission
4	Physical Address of data fragment 1
4	Length in bytes of data fragment 1
	.
	.
4	Physical Address of data fragment n
4	Length in bytes of data fragment n

The fields in the Transmit Raw Data request are:

- COMMAND CONTROL BLOCK (8 bytes)

  This field contains the Command Control Block. The command ID must be set to 0x101 for the Transmit Raw Data request.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

- ID # OF RESPONSE CHANNEL (4 bytes)

  If a response is enabled for this command (see Request Class field below), this field identifies the board-to-host channel to which the 5211 will write the Tx Response. If this field is 0, the 5211will use the default response channel (that is, the initial board-to-host channel crated at RC boot time).

- COMMAND TAG (4 bytes)

  This field contains a host-supplied tag generated by the device driver. It is usually the address of a host memory descriptor or the virtual address of a data buffer. This value is useful for releasing host memory resources associated with the transmission. The 5211 echoes the command tag to the host via the Tx Response (see page 160), provided that the response bit is set in the Request Class field.

- REQUEST CLASS (4 bytes)

  This field (shown in Figure 4-33) determines the type of operation or response required by the transmit request.

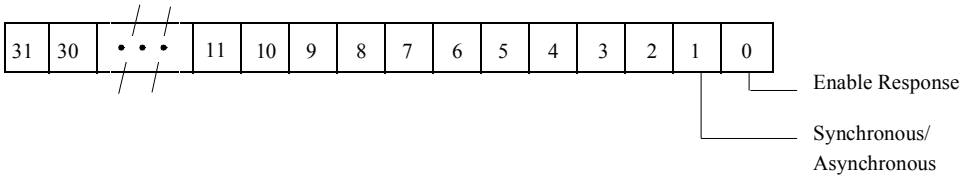


Figure 4-33. Request Class Field for Transmit Raw Data

  – BIT 0 ENABLE RESPONSE

    Setting this bit to 1 causes the 5211 to issue a Tx Response after executing the command. Clearing the bit disables the response.

  – BIT 1 SYNCHRONOUS/ASYNCHRONOUS

This bit is defined as:

0 = send data out as a synchronous transmission

1 = send data out as an asynchronous transmission



NOTE

The controller does not force the Frame Control field to conform to this bit setting. The host can send an asynchronous frame in a synchronous frame time. This is not in accordance with FDDI standards and is *not* recommended.

  – BITS 2 - 31 RESERVED

    These bits are reserved and must be cleared to 0 by the host.

- TRANSFER OPTIONS WORD (4 bytes)

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

This field contains the transfer options word. The data transfer options in this field apply to all data fragments associated with this command.

- TOTAL # OF BYTES FOR TRANSMISSION (4 bytes)

  This value is the total number of bytes for transmission, including the frame header.



NOTE

If the frame header is preceded by padding to word-align the header, the pad should *not* be included in the byte count.

- NUMBER OF FRAGMENTS FOR TRANSMISSION (4 bytes)

  This value specifies the number of fragments (that is, Physical Address/Length pairs) following this field.

  The maximum number of gather elements in a transmit is sixteen (16).

- PHYSICAL ADDRESS OF DATA FRAGMENT *n* (4 bytes)

  This field contains the memory address which the 5211 DMA channel uses to locate and move Data Fragment *n* on-board.



NOTE

If the frame header is preceded by padding to word-align the header, the physical address of Data Fragment 1 must point *past* the pad to the Frame control field.

This field contains the number of bytes in Data Fragment *n*.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

TRANSMIT RAW DATA (Embedded)

Group: FDDI
Type: Request
Command ID: 0x101

Transmit Raw Data (Embedded) is a special case of TRANSMIT RAW DATA where the Transfer Options Word is zero. Zero in the Transfer Options Word indicates that the data is embedded in the command. If the data is embedded in the transmit request, data is placed after the "Number of fragments for transmission" field. For embedded data "Number of fragments for transmission" also has a different meaning and contains the number of pad bytes. Padding is provided so that data in the segment can have the same byte alignment as the data in the host buffer. The total length of the command must be a longword multiple, or an error will occur. It is recommended that the size of the embedded data in the embedded transmit request should not exceed 256 bytes including the pad bytes.

Transmit Raw Data is a non-FDDI-standard request. It provides a mechanism to pass one frame from host-to-fiber with absolutely no interpretation or analysis performed by the board. This ability is necessary in applications where the Logical Link Control (LLC) implementation is responsible for all FDDI frame construction and data encapsulation.

After executing Transmit Raw Data, the 5211 issues a Tx Response (see page 160) if it has been enabled to do so in the Request Class field.

The data structure of the Transmit Embedded Raw Data request is detailed Table 4-6.

Table 4-7. Transmit Raw Data (Embedded) Request

Size (in bytes)	Description
8	Command Control Block
4	ID number of Response Channel
4	Command Tag
4	Request Class
4	Transfer Options Word = 0
4	Total number of bytes for transmission
4	Number of pad bytes
0-3	Pad
	data :

The fields in the Transmit Raw Data request are:

- COMMAND CONTROL BLOCK (8 bytes)

  This field contains the Command Control Block. The command ID must be set to 0x101 for the Transmit Raw Data request.

- ID # OF RESPONSE CHANNEL (4 bytes)

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

If a response is enabled for this command (see Request Class field below), this field identifies the board-to-host channel to which the 5211 will write the Tx Response. If this field is 0, the 5211will use the default response channel (that is, the initial board-to-host channel crated at RC boot time).

- COMMAND TAG (4 bytes)

  This field contains a host-supplied tag generated by the device driver. It is usually the address of a host memory descriptor or the virtual address of a data buffer. This value is useful for releasing host memory resources associated with the transmission. The 5211 echoes the command tag to the host via the Tx Response (see page 160), provided that the response bit is set in the Request Class field.

- REQUEST CLASS (4 bytes)

  This field (shown in Figure 4-34) determines the type of operation or response required by the transmit request.

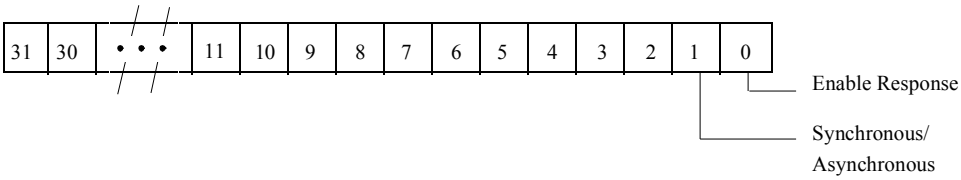


Figure 4-34. Request Class Field for Transmit Raw Data

  – BIT 0 ENABLE RESPONSE

    Setting this bit to 1 causes the 5211 to issue a Tx Response after executing the command. Clearing the bit disables the response.

  – BIT 1 SYNCHRONOUS/ASYNCHRONOUS

    This bit is defined as:

    0 = send data out as a synchronous transmission

    1 = send data out as an asynchronous transmission



NOTE

The controller does not force the Frame Control field to conform to this bit setting. The host can send an asynchronous frame in a synchronous frame time. This is not in accordance with FDDI standards and is *not* recommended.

  – BITS 2 - 31 RESERVED

    These bits are reserved and must be cleared to 0 by the host.

- TRANSFER OPTIONS WORD (4 bytes)

  This field contains zero to indicate that this is an embedded request.

- TOTAL # OF BYTES FOR TRANSMISSION (4 bytes)

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

This value is the total number of bytes for transmission, including the frame header.



NOTE

If the frame header is preceded by padding to word-align the header, the pad should *not* be included in the byte count.

- NUMBER OF PAD BYTES (4 bytes)

  If necessary, the host may pad the frame header to align it on a word boundary. Your system requirements determine if padding is used. If padding is used, it should not be included in the “Total Number of Bytes for Transmission” field.

- DATA

  This field contains the embedded data.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

TRANSMIT DATAGRAM

Group: FDDI
Type: Request
Command ID: 0x102

The Transmit Datagram request is used to pass one IP datagram over FDDI in automatic conformance with RFC 1042 and RFC 1103. The FDDI headers are built by the 5211's firmware and should not be supplied by the host application. The headers built by the firmware consist of the following fields:

- 13-byte MAC header (FC field, destination address, and source address)
- 3-byte LLC header (DSAP, SSAP, and Control). The contents of these bytes are:
  - DSAP = 170 (0xAA)
  - SSAP = 170 (0xAA)
  - Control = 3 (0x03)
- 3-byte SNAP header contains all 0's
- 2-byte Ethertype

Data for the frame can be "gathered" from blocks of host memory that are not contiguous.

After executing Transmit Datagram, the 5211 issues a Tx Response if it has been enabled to do so in the Request Class field. The data structure of the Transmit Datagram request is detailed in Table 4-8.

Table 4-8. Transmit Datagram Request

Size (in bytes)	Description
8	Command Control Block
4	ID Number of Response Channel
4	Command Tag
6	Destination Address
2	Ethertype
4	Request Class
4	Transfer Options Word
4	Total number of bytes for transmission
4	Number of fragments for transmission
4	Physical Address of data fragment 1
4	Length in bytes of data fragment 1
	.
4	Physical address of data fragment n
4	Length in bytes of data fragment n

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The fields in the Transmit Datagram request are:

- COMMAND CONTROL BLOCK (8 bytes)

  This field contains the Command Control Block. The command ID must be set to 0x102 for the Transmit Datagram request.

- ID # OF RESPONSE CHANNEL (4 bytes)

  If a response is enabled for this command (see Request Class field below), this field indicates which board-to-host channel the controller should use when it responds to the command. If the field is 0, the controller uses the default response channel. The default response channel is the initial board-to-host channel created at RC boot time.

- COMMAND TAG (4 bytes)

  This field contains a host-supplied tag generated by the device driver. It is usually the address of a host memory descriptor or the virtual address of a data buffer. This value is useful for releasing host memory resources associated with the transmission. The command tag is echoed back to the host via a Tx Response (see page 160) if the response bit is set in the Request Class field.

- DESTINATION ADDRESS (6 bytes)

  This field identifies the destination address for insertion into the FDDI MAC header constructed by the firmware. The bit order of the destination address must be "big-endian," meaning that it is in Internet (or "canonical" order), with the Group bit positioned as the low-order bit of the first octet.

  FDDI addresses as they appear on the fiber (which is in IEEE or "little-endian" order) have the Group bit in the high order, or most-significant, bit position.



NOTE

The Transmit Datagram command assumes that the source address is already stored in onboard firmware. The source address is either the factory default address or a user-specified address. The factory default address is obtained by issuing Request Station Address. To specify a different station address, execute the Set Station Address directive.

- ETHERTYPE (2 bytes)

  This field contains the 2-byte Ethertype field that is inserted into the Logical Link Control SNAP header (RFC 1103) constructed by the 5211.

- REQUEST CLASS (4 bytes)

  This field determines the type of operation or response required by the transmit request.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

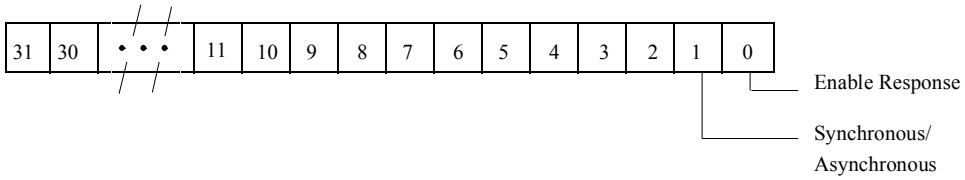


Figure 4-35. Request Class Field for Transmit Datagram

– BIT 0 ENABLE RESPONSE

Setting this bit to 1 causes the 5211 to issue a Tx Response after executing the command. Clearing the bit disables the response.

– BIT 1 SYNCHRONOUS/ASYNCHRONOUS

This bit is defined as:

0 = send data out as a synchronous transmission

1 = send data out as an asynchronous transmission



NOTE

The controller forces the Frame Control field to conform to this bit setting.

For more information Synchronous vs. Asynchronous transmissions, see chapter 3.

– BITS 2 - 31 RESERVED

These bits are reserved and must be cleared to 0 by the host.

- TRANSFER OPTIONS WORD (4 bytes)

  This field contains the transfer options word. The data transfer options in this field apply to all data fragments associated with this command.

- TOTAL # OF BYTES FOR TRANSMISSION (4 bytes)

  This value is the total number of bytes for transmission, excluding the frame header (i.e., data only).

- NUMBER OF FRAGMENTS FOR TRANSMISSION (4 bytes)

  This value specifies the number of fragments (that is, Physical Address/Length pairs) following this field. The maximum number of gather elements in a transmit is sixteen (16).

- PHYSICAL ADDRESS OF DATA FRAGMENT *n* (4 bytes)

  This field contains the memory address which the 5211's DMA channel uses to locate and move Data Fragment *n* onboard.

- LENGTH IN BYTES OF DATA FRAGMENT *n* (4 bytes)

  This field contains the number of bytes in Data Fragment *n*.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

TRANSMIT DATAGRAM (EMBEDDED)

Group: FDDI
Type: Request
Command ID: 0x102

Transmit Datagram (Embedded) is a special case of the TRANSMIT DATAGRAM command where the Transfer Options Word is set to zero. Zero in the Transfer Options Word indicates that the data is embedded in the command. If the data is embedded in the transmit request, data is placed after the “Number of fragments for transmission” field. For embedded data “Number of fragments for transmission” also has a different meaning and contains the number of pad bytes. Padding is provided so that data in the segment can have the same byte alignment as the data in the host buffer. The total length of the command must be a longword multiple, or an error will occur. It is recommended that the size of the embedded data in the embedded transmit request should not exceed 256 bytes including the pad bytes. See *TRANSMIT DATAGRAM on page 127* for additional information on Transmit Datagram.

Data for the frame can be “gathered” from blocks of host memory that are not contiguous. After executing Transmit Datagram, the 5211 issues a Tx Response if it has been enabled to do so in the Request Class field. The data structure of the embedded Transmit Datagram request is detailed in Table 4-8.

Table 4-9. Transmit Datagram (Embedded) Request

Size (in bytes)	Description
8	Command Control Block
4	ID Number of Response Channel
4	Command Tag
6	Destination Address
2	Ethertype
4	Request Class
4	Transfer Options Word = 0
4	Total number of bytes for transmission
4	Number of Pad Bytes
0-3	Pad
	data :

The fields in the Transmit Datagram request are:

- COMMAND CONTROL BLOCK (8 bytes)

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

This field contains the Command Control Block. The command ID must be set to 0x102 for the Transmit Datagram request.

- ID # OF RESPONSE CHANNEL (4 bytes)

  If a response is enabled for this command (see Request Class field below), this field indicates which board-to-host channel the controller should use when it responds to the command. If the field is 0, the controller uses the default response channel. The default response channel is the initial board-to-host channel created at RC boot time.

- COMMAND TAG (4 bytes)

  This field contains a host-supplied tag generated by the device driver. It is usually the address of a host memory descriptor or the virtual address of a data buffer. This value is useful for releasing host memory resources associated with the transmission. The command tag is echoed back to the host via a Tx Response (see page 160) if the response bit is set in the Request Class field.

- DESTINATION ADDRESS (6 bytes)

  This field identifies the destination address for insertion into the FDDI MAC header constructed by the firmware. The bit order of the destination address must be "big-endian," meaning that it is in Internet (or "canonical" order), with the Group bit positioned as the low-order bit of the first octet.

  FDDI addresses as they appear on the fiber (which is in IEEE or "little-endian" order) have the Group bit in the high order, or most-significant, bit position.



NOTE

The Transmit Datagram command assumes that the source address is already stored in onboard firmware. The source address is either the factory default address or a user-specified address. The factory default address is obtained by issuing Request Station Address. To specify a different station address, execute the Set Station Address directive.

- ETHERTYPE (2 bytes)

  This field contains the 2-byte Ethertype field that is inserted into the Logical Link Control SNAP header (RFC 1103) constructed by the 5211.

- REQUEST CLASS (4 bytes)

  This field determines the type of operation or response required by the transmit request.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

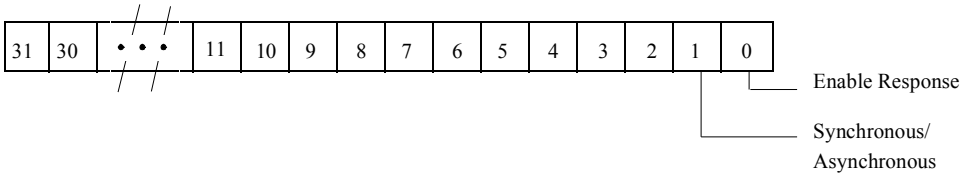


Figure 4-36. Request Class Field for Transmit Datagram

– BIT 0 ENABLE RESPONSE

Setting this bit to 1 causes the 5211 to issue a Tx Response after executing the command. Clearing the bit disables the response.

– BIT 1 SYNCHRONOUS/ASYNCHRONOUS

This bit is defined as:

0 = send data out as a synchronous transmission

1 = send data out as an asynchronous transmission

NOTE

The controller forces the Frame Control field to conform to this bit setting.

For more information Synchronous vs. Asynchronous transmissions, see chapter 3.

– BITS 2 - 31 RESERVED

These bits are reserved and must be cleared to 0 by the host.

- TRANSFER OPTIONS WORD (4 bytes)

  This field contains the transfer options word. The data transfer options in this field apply to all data fragments associated with this command.

- TOTAL # OF BYTES FOR TRANSMISSION (4 bytes)

  This value is the total number of bytes for transmission, including the frame header.

NOTE

If the frame header is preceded by padding to word-align the header, the pad should *not* be included in the byte count.

- NUMBER OF PAD BYTES (4 bytes)

  If necessary, the host may pad the frame header to align it on a word boundary. Your system requirements determine if padding is used. If padding is used, it should not be included in the “Total Number of Bytes for Transmission” field.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

- DATA

  This field contains the embedded data.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

SET UP RECEIVE BUFFERS

Group: FDDI
Type: Directive
Command ID: 0x103

The Set Up Receive Buffers directive supplies the 5211 with a buffer list of host memory resources for received data. The buffer list is a block of contiguous memory containing "buffer list elements." The format of this element is shown in Figure 4-37.

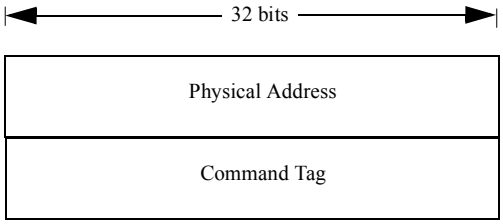


Figure 4-37. Buffer List Element

The Physical Address of a buffer list element is used by the 5211 to DMA data to the buffer. The address must be aligned on a longword boundary. The Command Tag of a buffer list element is a host-assigned value which will be echoed back by the controller via a Rx Frame indication (page 162).

The host delimits the end of the buffer list by writing the special value 0xFFFFFFFF into the Physical Address field of the element beyond the last valid buffer descriptor. For additional information on the buffer list, see chapter 3.

The 5211 does not pass a response message to the host for the Set Up Receive Buffers directive.

The data structure format of this directive is shown in Table 4-10.:

Table 4-10. Set Up Receive Buffers Directive

Size (in bytes)	Description
8	Command Control Block
4	Transfer Options Word for Buffer List
4	Physical Address of Buffer List
4	Length in Bytes of Buffer List
4	Rx Load Adjustment
4	Transfer Options Word (for all buffers)
4	Length in Bytes of Each Buffer

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The fields in the Set Up Receive Buffers directive are:

- COMMAND CONTROL BLOCK (8 bytes)

  This field contains the Command Control Block. The command ID must be set to 0x103 for the Set Up Receive Buffers directive.

- TRANSFER OPTIONS WORD FOR BUFFER LIST (4 bytes)

  This field contains the transfer options word. The 5211 uses these transfer options to DMA the buffer list onboard. It is not used to transfer individual buffers in the buffer list.

- PHYSICAL ADDRESS OF BUFFER LIST (4 bytes)

  This is the address that the 5211 uses to DMA the buffer list onboard. It must be aligned on a longword boundary.

- LENGTH IN BYTES OF BUFFER LIST (4 bytes)

  This field contains the number of bytes for the 5211 to DMA on-board. Each buffer list element is 8 bytes long. Therefore, to allow for the delimiter, the number of bytes in the buffer list is:

  8 * (N + 1)

  (where N is the number of receive buffers in the list)

  The recommended maximum list size is 2 Kilobytes. Creating a larger list will result in excessive overhead for the 5211.

- RX LOAD ADJUSTMENT (4 bytes)

  This value specifies the maximum number of received frames that the 5211will process simultaneously. Default value is 32.

  For example, placing 0x10 in this field causes the 5211 to process no more than 16 incoming frames at once.

  The set of valid adjustment values = {0x1 ... 0x50}. The 5211 can currently process up to 80 (0x50) received frames at once. Entering a value between 0x51 – 0xFF (inclusive) causes the 5211 to use an adjustment value of 0x50. Clearing the field to 0 or entering a value greater than 0xFF disables the adjustment feature.

- TRANSFER OPTIONS WORD FOR ALL BUFFERS (4 bytes)
- This field contains the transfer options word, as defined on p. 4-4. the 5211 uses these transfer options to transfer data to the individual receive buffers in the buffer list.

- LENGTH IN BYTES OF EACH BUFFER (4 bytes)

  This field specifies the size, in bytes, of the individual receive buffers in the list. All buffers in the list must be this length. The buffer size should be between 512 bytes and 8K bytes (inclusive) and must be a longword multiple (that is, divisible by 4).

  The recommended minimum buffer size is 1 Kilobyte. Using smaller buffers results in excessive bandwidth to transfer the frames across the bus.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

REQUEST STATION ADDRESS

Group: FDDI
Type: Request
Command ID: 0x104

Request Station Address is used to obtain the controller's current station address. Interphase assigns a unique address to each individual 5211. The host may change this factory default address using the Set Station Address directive (page 137).

The controller returns its current address by issuing Report Station Address (page 164).

The data structure of Request Station Address is detailed in Table 4-11.

Table 4-11. Request Station Address

Size (in bytes)	Description
8	Command Control Block
4	ID Number of Response Channel
4	Command Tag

The fields in the Request Station Address request are:

- COMMAND CONTROL BLOCK (8 bytes)

  This field contains the Command Control Block. The command ID must be set to 0x104 for Request Station Address.

- ID # OF RESPONSE CHANNEL (4 bytes)

  This field identifies the board-to-host channel to which the 5211 should write the Report Station Address response. If the field is 0, the controller will use the default response channel. The default response channel is the initial board-to-host channel created at RC boot time.

- COMMAND TAG (4 bytes)

  This field contains a host-supplied tag that is echoed by the controller in response to the command.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

SET STATION ADDRESS

Group: FDDI
Type: Directive
Command ID: 0x105

The Set Station Address directive is used to change the 5211's default station address stored in permanent memory. The permanent memory also contains a write-protected copy of the factory-assigned station address. The Set Station Address directive can be used to restore the factory-assigned address from this backup copy, if necessary.

After the 5211 executes the directive, the host must reset the board or cycle the power in order to make the new address effective.

The controller does not pass a response message to the host for the Set Station Address directive. To verify the updated station address, issue a Request Station Address to the 5211 (page 136), and then read the controller-generated Report Station Address response (page 164).

The data structure format of the directive is described in Table 4-12.

Table 4-12. Set Station Address Directive

Size (in bytes)	Description
8	Command Control Block
6	Station Address
2	Type

The fields in the Set Station Address directive are:

- COMMAND CONTROL BLOCK (8 bytes)

  This field contains the Command Control Block. The command ID must be set to 0x105 for the Set Station Address directive.

- STATION ADDRESS (6 bytes)

  This field sets the desired station address. The address must be represented using the format specified by Internet Protocol (IP). It should be stored byte-wise, with the Group bit positioned as the low order bit of the first octet.

  For efficient data transmission, the address given with this command is bit-reversed within each byte before being written to permanent memory. The effective MAC address for the station (that is, the one placed into the MAC header by a Transmit Datagram request) is drawn directly from memory without requiring modification for FDDI format.

  The Short Addressing mode, which uses a 16-bit address instead of the more common 48-bit long addresses, is not currently supported.

- TYPE (2 bytes)

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

This field specifies how the on-board address memory is to be manipulated. Legal values are described in Table 4-13.

Table 4-13. Type Codes in Set Station Address Directive

Code	Purpose
0x0	Restore the factory default station address. This option causes any value in the Station Address field to be ignored.
0x1	Change the current station address to the value specified in the Station Address field.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

RING CONTROL

Group: FDDI
Type: Directive
Command ID: 0x106

The Ring Control directive is used to connect or disconnect the 5211 from the FDDI ring. The 5211 does not insert itself into an FDDI network unless it is explicitly instructed to do so.

Issuing this directive causes the board to automatically perform all Station Management functions necessary for entering an FDDI network in accordance with the X3T9.5 standards.

The 5211 does not pass a response message to the host for the Ring Control directive. It issues a Ring Status Indication asynchronously when the FDDI ring changes state. If the host succeeds in its attempt to connect to (or disconnect from) the ring, the 5211 will report the new ring status by issuing a Ring Status Indication. If the attempt fails (or the 5211 is already in the state specified by the Ring Control directive), the indication is *not* issued.

The data structure format of the Ring Control directive is described in Table 4-14.

Table 4-14. Ring Control Directive

Size (in bytes)	Description
8	Command Control Block
4	Ring Control Options

The fields in the Ring Control directive are:

- COMMAND CONTROL BLOCK (8 bytes)

  This field contains the Command Control Block. The command ID must be set to 0x106 for the Ring Control directive.

- RING CONTROL OPTIONS (4 bytes)

  Two legal values are currently defined for this field:

  0x1 Connect to the ring

  0x2 Disconnect from the ring

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

PERFORM MAINTENANCE

Group: FDDI
Type: Directive
Command ID: 0x107

It is convenient to be capable of generating specific FDDI line states upon demand in order to debug certain classes of network problems. The Perform Maintenance directive places the 5211 in a maintenance state in which generate a desired line state. The command can be reissued to remove the controller from the maintenance state.

The controller does not pass a response message to the host for the Perform Maintenance directive.

The data structure of the directive is described in Table 4-15.

Table 4-15. Perform Maintenance Directive

Size (in bytes)	Description
8	Command Control Block
4	Maintenance Control Options
4	Maintenance State

The fields in the Perform Maintenance directive are:

- COMMAND CONTROL BLOCK (8 bytes)

This field contains the Command Control Block. The command ID must be set to 0x107 for the Perform Maintenance directive.

MAINTENANCE CONTROL OPTIONS (4 bytes)

Two values are currently defined for this field:

0x0Put the PCM state machine into the Maintenance State and issue a specified line state.

0x1Remove the PCM state machine from the Maintenance State and return to the OFF state.

- MAINTENANCE STATE (4 bytes)

  This field is used to specify a line state when the Maintenance Control field value is 0x0. Legal entries in this field are described in Table 4-16.

Table 4-16. Line States Available for Perform Maintenance Directive

Value	State
0x4	Master
0x8	Idle
0x10	Halt
0x20	Quiet

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table 4-16. Line States Available for Perform Maintenance Directive

Value	State
0x100	Transmit PHY Data Request

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

GET MIB ATTRIBUTE

Group: FDDI
Type: Request
Command ID: 0x108

Get MIB Attribute is a multi-purpose request which can be used to obtain an element (or group of elements) stored in the Management Information Base (MIB) onboard the controller. Each MIB attribute or group has a specific name.

To use this command, you must be familiar with the TLV parameter structure of MIB attributes. For details, refer to Section 6 (“Services”) of the SMT specifications.

The 5211 returns the requested information to the host by issuing a MIB Attribute Response (page 168).

The data structure of Get MIB Attribute is described in Table 4-17.

Table 4-17. Get MIB Attribute Request

Size (in bytes)	Description
8	Command Control Block
4	ID Number of Response Channel
4	Command Tag
4	Attribute Name
4	Data Area Needed for Attribute(s)
4	Attribute Index

The fields in the Get MIB Attribute request are:

- COMMAND CONTROL BLOCK (8 bytes)

  This field contains the Command Control Block. The command ID must be set to 0x108 for the Get MIB Attribute request.

- ID # OF RESPONSE CHANNEL (4 bytes)

  This field identifies the board-to-host channel to which the 5211 should write the MIB Attribute Response. If the field is 0, the controller uses the default response channel. The default response channel is the initial board-to-host channel created at RC boot time.

- COMMAND TAG (4 bytes)

  This field may contain a host-generated command tag, if needed for the application. The 5211 does not use the tag in any way. It is echoed back to the host in the MIB Attribute Response.

- ATTRIBUTE NAME (4 bytes)

  This field contains the Parameter_Type of the attribute or MIB group, as defined in the SMT revision 6.2/7.3 specifications. The high byte contains the object and

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

sub-object ID, and the lower byte is the attribute registration ID. For example, to request the attribute fddiSMTStationid, use 0x100B in this field.

- DATA AREA NEEDED FOR ATTRIBUTE(S) (4 bytes)

  This field tells the controller how many bytes of on-board data area it needs to handle the requested attribute(s). The host should allow 36 (0x24) bytes for each requested attribute. 36 bytes is the maximum attribute size, as defined in the SMT 6.2/7.3 specifications. Familiarity with the TLV parameter structure of MIB attributes is necessary. For details, see Section 6 (“Services”) of the SMT 6.2/7.3 specifications.

  This field should contain the following value:

  Data area = 0x24 * the number of attributes requested

- ATTRIBUTE INDEX (4 bytes)

  This contains the MAC, Port, Path, or Attachment index as defined in the SMT specifications. Valid entries are described in Table 4-18.

Table 4-18. Valid Entries in MIB Attribute Index Field

Type of Attribute	Legal Value(s)
SMT	0x0 (SMT attributes do not have an index, so the field must be cleared.)
MAC	0x1
PATH	0x1 or 0x2
PORT	0x1 or 0x2
ATTACHMENT	0x1 or 0x2 (SMT 6.2 only)

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

SET MIB ATTRIBUTE

Group: FDDI
Type: Directive
Command ID: 0x109

The Set MIB Attribute directive allows the host to assign a value to a specific attribute in the 5211's on-board Management Information Base (MIB). The command can only be used to change attributes with read/write access, as defined in revision 6.2/7.3 of the FDDI Station Management specifications.

Since the default settings of the 5211's MIB attributes are correct for most applications, this command is *not* required in most drivers.



CAUTION

Changing the MIB attributes requires an Expert-Level understanding of revision 6.2/7.3 of the FDDI Station Management specifications.

To use this command, familiarity with the TLV parameter structure of MIB attributes is necessary. For details, refer to the SMT 6.2/7.3 specifications.

If SMT rejects the proposed MIB attribute value the MIB attribute is not changed. Instead, the 5211 informs the host about the problem by issuing Report Printf Call (an indication in the generic RC command set).

MIB attributes are stored in static RAM on the 5211. Cycling the power or performing a reset causes the 5211 to restore the default MIB attributes.

The driver must check and make sure the value is set in the MIB. The controller does not pass a response message to the host for the Set MIB Attribute directive. To check the new attribute definition, issue a Get MIB Attribute request.

The data structure of Set MIB Attribute is shown on page 144. Its format is shown in Table 4-19.

Table 4-19. Set MIB Attribute Directive

Size (in bytes)	Description
8	Command Control Block
4	Attribute Name
4	Data Area Needed for Attribute
4	Attribute Index
n	Attribute Definition (Variable-length)

The fields in the Set MIB Attribute directive are.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

- COMMAND CONTROL BLOCK (8 bytes)

  This field contains the Command Control Block. The command ID must be set to 0x109 for the Set MIB Attribute directive.

- ATTRIBUTE NAME (4 bytes)

  This field contains the Parameter_Type of the attribute, as defined in the SMT 6.2/7.3 specifications. The high byte contains the object and sub-object ID, and the lower byte is the attribute registration ID. For example, to specify the attribute *fddiSMTConnectionPolicy*, write 0x101B to this field.

- DATA AREA NEEDED FOR ATTRIBUTE (4 bytes)

  This field should contain the attribute's Parameter_Length, as defined in the SMT 6.2/7.3 specs. For example, write 0x0004 to this field if setting the attribute *fddiSMTConnectionPolicy*.

- ATTRIBUTE INDEX (4 bytes)

  This field contains the MAC, Port, Path, or Attachment index as defined in the SMT specifications. Valid entries are shown in Table 4-20.

Table 4-20. Valid Entries in MIB Attribute Index Field

Type of Attribute	Legal Value(s)
SMT	0x0 (SMT attributes do not have an index, so the field must be cleared.)
MAC	0x1
PATH	0x1 or 0x2
PORT	0x1 or 0x2
ATTACHMENT	0x1 or 0x2 (SMT 6.2 only)

- ATTRIBUTE DEFINITION (Variable-Length)

  This field contains the new attribute definition. The host must write the data as a byte stream, using exactly the same TLV encoding as given in the SMT specifications. No termination is necessary.

  For example: You wish to set the attribute *fddiSMTConnectionPolicy*. The TLV encoding of this attribute (as defined in the SMT revision 6.2 specifications) is shown below.

  Parameter_TypeX'10 1B
  Parameter_LengthX'00 04'
  PadX'00 00'
  Parameter_ValueX'00 00'-X'00 7F'

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

To assign a new value to *fddiSMTConnectionPolicy*, execute a Set MIB Attribute command with the fields completed as shown in Table 4-21.

Table 4-21. Example Set MIB Command (*FDDI SMTConnectionPolicy*)

Field	Longword	Contents	Comments
Command Control Block (8 bytes)	0x1 0x2	0x00000038 0x00000109	Total Command length is 0x38 bytes. Command ID is 0x109.
Attribute Name (4 bytes)	0x3	0x0000101B	Attribute's Parameter_Type as defined in the SMT 6.2 specifications.
Data Area Needed for Attribute (4 bytes)	0x4	0x00000004	Attribute's Parameter_Length as defined in the SMT 6.2 specifications.
Attribute Index (4 bytes)	0x5	0x00000000	SMT attributes have no index (see Table 4-18, page 143)
Attribute Definition (16 bytes for FDDI SMT ConnectionPolicy)	0x6 0x7 - 0xD	0x0000*yyyy* 0x*zzzzzzzz*	*yyyy* = Attribute value *zzzz*... = Don't care values in TLVPParamType



NOTE

If the 5211 undergoes a hard reset, the default parameters will be restored.

To better understand how to fill in the command fields, examine the C code listingshown in Figure 4-38. This code produces in host memory a hex dump matching the "Contents" fields shown in the preceding example. It uses the TLVParamStruct structure.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

```
extern   ULONG    *rc_ptr;              /* host->board channel pointer */
set_smtconnectpolicy(void)
{
ULONG cmd_length,  *cmd_ptr;
USHORT   reject;
FDDI_SET_MIB      *ccb_ptr;
TLVParamType      *smt_ptr;

                  reject = get_policy();

                  cmd_length =
                  (((sizeof(FDDI_SET_MIB))-(sizeof(u32)))+(sizeof(TLVPParamType)));
                  cmd_length = align_it(cmd_length);     /* insure long word alignment   */
                  will_it_fit(cmd_length);              /* check for link necessary     */
                  cmd_ptr = rc_ptr;
                  ccb_ptr = (FDDI_SET_MIB *)rc_ptr;
                  smt_ptr = (TLVParamType *)&ccb_ptr->attribute:

                  ccb_ptr->ccb.len = cmd_length;
                  ccb_ptr->ccb.cmd = FDDI_SET_MIB_DIRECTIVE;
                  ccb_ptr->attr_name = fddiSMTConnectionPolicy;
                  ccb_ptr->attr_length = (sizeof(TLV168itType));
                  ccb_ptr->attr_index = 0;              /* index = 0 for smt          */

                  smt_ptr->paramType = fddiSMTConnectionPolicy;
                  smt_ptr->paramLen = (sizeof(TLV168itType));
                  smt_ptr->SMTPARAM16 = reject;

                  rc_ptr += (cmd_length/(sizeof(u32)));
                  do_h2b( cmd_ptr );                    /* display and execute command */
}
```

Figure 4-38. C Code for Set MIB Example

The hex code produced by the above code is:

Longword	Data	Explanation
0x0	0x00000038	See field descriptions in Table 4-21.
0x1	0x00000109	
0x2	0x0000101b	
0x3	0x00000004	
0x4	0x00000000	
0x5	0x101b0004	
0x6	0x0000yyyy	
0x7 - 0xD	0xzzzzzzzz	

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

SET CAM ADDRESS

Group: FDDI
Type: Request
Command ID: 0x10A

Set CAM Address is used to store an address in the 5211's content-addressable memory (CAM). This feature allows the 5211 to receive data addressed to up to 32 different destinations, in addition to the normal base station address embedded in permanent memory. As frames on the ring move through the 5211, their destination addresses are matched against addresses stored in the CAM. When a match is found, the 5211 copies that frame into onboard memory and passes it up to the host as an ordinary receive indication.

The command will complete with error if the host attempts to set an address which is already present in the CAM or if the CAM already contains 32 addresses. To remove an individual address from the CAM, issue the Clear CAM Address request (page 150). To clear out all CAM addresses, issue the directive Flush CAM Addresses (page 151).

After executing the request, the 5211 issues CAM Address Response (page 166). The data structure of Set CAM Address is detailed in Table 4-22:

Table 4-22. Set CAM Address Request

Size (in bytes)	Description
8	Command Control Block
4	ID Number of Response Channel
4	Command Tag
6	Station Address
2	Reserved

The fields in the Set CAM Address request are:

- COMMAND CONTROL BLOCK (8 bytes)

  This field contains the Command Control Block. The command ID must be set to 0x10A for the Set CAM Address request.

- ID # OF RESPONSE CHANNEL (4 bytes)

  This field identifies the board-to-host channel to which the 5211 should write the CAM Address Response. If the field is 0, the controller uses the default response channel. The default response channel is the initial board-to-host channel created at RC boot time.

- COMMAND TAG (4 bytes)

  This field may contain a host-generated command tag, if needed for the application. The 5211 does not use the tag in any way. It is echoed back to the host in the CAM Address Response.

- STATION ADDRESS (6 bytes)

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The station address should be given in Internet bit order. The 5211 will bit-swap it to IEEE bit-transmission order before placing it in the CAM.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

CLEAR CAM ADDRESS

Group: FDDI
Type: Request
Command ID: 0x10B

Clear CAM Address is used to remove a destination address which the host has previously stored in the 5211's content-addressable memory (CAM). After executing the request, the 5211 issues CAM Address Response (page 166). If the specified address is not present in the CAM, the response will flag the error. The format of Clear CAM Address is shown in Table 4-23.

Table 4-23. Clear CAM Address Request

Size (in bytes)	Description
8	Command Control Block
4	ID Number of Response Channel
4	Command Tag
6	Station Address
2	Reserved

The fields in the Clear CAM Address request are:

- COMMAND CONTROL BLOCK (8 bytes)

  This field contains the Command Control Block. The command ID must be set to 0x10B for the Clear CAM Address request.

- ID # OF RESPONSE CHANNEL (4 bytes)

  This field identifies the board-to-host channel to which the 5211 should write the CAM Address Response. If the field is 0, the controller uses the default response channel. The default response channel is the initial board-to-host channel created at RC boot time.

- COMMAND TAG (4 bytes)

  This field may contain a host-generated command tag, if needed for the application.

- STATION ADDRESS (6 bytes)

  This field contains the station address that is to be removed from the CAM. It should specified in Internet bit order.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

FLUSH CAM ADDRESSES

Group: FDDI
Type: Directive
Command ID: 0x10C

The Flush CAM Addresses directive is used to remove *all* addresses that the host stored in the 5211's content-addressable memory (CAM).

The format of this command is detailed in Table 4-24:

Table 4-24. Flush CAM Addresses Directive

Size (in bytes)	Description
8	Command Control Block

The field in the Flush CAM Addresses directive is:

- COMMAND CONTROL BLOCK (8 bytes)

  This field contains the Command Control Block. The command ID must be set to 0x10C for the Flush CAM Addresses directive.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

SMT TRACE

Group: FDDI
Type: Directive
Command ID: 0x10D

The SMT Trace directive is used to instruct the 5211 to route SMT frames and/or events to the host. The command must be issued twice in order to enable both frame and event tracing. The default is for the 5211 to route neither of these SMT entities to the host.

If the host enables frame tracing, the 5211 will route each incoming SMT frame to the host by issuing an Rx Frame indication (see page 162). If event tracing is enabled, the board issues an SMT Event Indication (page 170) to notify the host of each SMT event.

To disable frame or event tracing, the host reissues the directive with the appropriate value in the Trace Control field, as described below. The format of the command is shown in Table 4-25:

Table 4-25. SMT Trace Directive

Size (in bytes)	Description
8	Command Control Block
4	Trace Control

The fields in the SMT Trace directive are:

COMMAND CONTROL BLOCK (8 bytes)

This field contains the Command Control Block. The command ID must be set to 0x10D for the SMT Trace directive.

TRACE CONTROL (4 bytes)

Table 4-26 shows the legal values for this field.

Table 4-26. Values in Trace Control Field

Value	Description
0x1	Turn SMT frame tracing on. Route each incoming SMT message to the host via an Rx Frame Indication.
0x2	Turn SMT frame tracing off.
0x3	Turn SMT event tracing on. Route each incoming SMT event to the host via an SMT Event Indication.
0x4	Turn SMT event tracing off.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

SET HARDWARE PARAMETERS

Group: FDDI
Type: Directive
Command ID: 0x10E

Set Hardware Parameters allows manipulation of some hardware features on the 5211. These registers control various modes of operation and enable/disable specific functions of the controller. One or any combination of the supported hardware registers may be written in a single directive. No specific ordering of parameters is required.

The controller does not pass a response message to the host for the Set Hardware Parameters directive. The format of the directive is detailed in Table 4-27.

Table 4-27. Set Hardware Parameters Directive

Size (in bytes)	Description
8	Command Control Block
4	Command Tag
4	Parameter Count
2	Parameter Type
2	Parameter Data
	:
	:
2	Parameter Type
2	Parameter Data

The fields in the Set Hardware Parameters directive are:

- COMMAND CONTROL BLOCK (8 bytes)

  This field contains the Command Control Block. The command ID must be set to 0x10E for the Set Hardware Parameters directive.

- COMMAND TAG (4 bytes)

  This field contains a host-supplied tag generated by the device driver. It is usually the address of a host memory descriptor or the virtual address of a data buffer.

- PARAMETER COUNT (4 bytes)

  This field contains the number of parameter changes supplied in the remainder of this command. For each change, there is one pair of Parameter Type and Parameter Data fields.

  Example: If there are two pairs of Parameter Type/Parameter Data fields in the directive, the Parameter Count field should contain 0x2.

- PARAMETER TYPE *n* (2 bytes)

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The code in this field identifies the type of parameter to be manipulated. Table 4-28 lists the codes for the available registers and functions. No specific ordering of parameters is required.

Table 4-28. Host-Settable Registers and Functions

Code	Parameter Type	Valid Parameter Data
0x0001	Reserved	
0x0002	Reserved	
0x0003	Reserved	
0x0004	Set MAC Mode	See Table 4-29, Table 4-30, and Table 4-31
0x0005	Disable/Enable FCS Generation	See Table 4-32
0x0006	Optical Bypass Relay Control	See Table 4-33
0x0007	Set MAC Control Register A	
0x0008	Reset MAC Control Register A	

- PARAMETER DATA *n* (2 bytes)

  This field contains either:

  – A mapped field of control bits which will be written to a specific register, or

  – A code enabling/disabling a specific function. The data in this field must be valid for the Parameter Type field with which it is paired (that is, the 2-byte Parameter Type field immediately preceding it). Valid entries for the parameter types listed inTable 4-28 are described in the following tables.

  Set MAC Mode (0x0004)

  This parameter type allows the 5211 to copy additional frames that would otherwise be ignored, depending on the promiscuous mode setting. In addition, bridge stripping mode allows the 5211 to remove frames transmitted by this station that have a source address that is not on the local ring.

Table 4-29. Set MAC Mode

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													BS	PMode	

- MAC MODE Bit Assignments:

  Bits 0 - 1: PMode - These bits define the level of promiscuity that the MAC will operate in when receiving LLC frames.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table 4-30. Set PMode

Code	PMode
0x0	Normal mode, receive if the destination address matches the local MAC address*
0x1	Receive if the destination or source address matches the local MAC address*
0x2	Receive all frames except source address matched frames*
0x3	Receive all frames

* Note: These are compatible with the 4211 “Set Promiscuous Mode” modes of operation.

Bit 2: BS - This bit, if set, will cause the MAC to implement Bridge Stripping mode when transmitting frames.

Table 4-31. Set Bridge Stripping

Value	BS
0x0	Disable bridge stripping mode
0x1	Enable bridge stripping mode

Bits 3 - 15: Reserved - These bits should always be set to zero.

- Disable/Enable FCS Generation (0x0005)

Table 4-32. FCS Generation Mode Control

Code	Mode
0x0	Frame Check Sequence is sent on transmit frames
0x1	Inhibits the generation of the Frame Check Sequence on transmitted frames

- Optical Bypass Relay Control (0x0006)

  Provides control of the optical bypass relay.

  SMT running onboard the 5211 provides OBS control according to the FDDI specifications. This parameter is not required.

Table 4-33. Optical Bypass Relay Control

Code	Mode
0x0	Enable bypass mode

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table 4-33. Optical Bypass Relay Control

Code	Mode
0x1	Enable THRU mode

- Set MAC Control Register A (0x0007)

  Bits set in this data field will be set in the Motorola Media Access Control chip (MC68838), MAC Control Register A. All bits that are not set will remain in their previous state. For example, a value of 0x0010 will enable the “copy-grp-llc” bit. This parameter is optional. For more information, see the Motorola data book.

- Reset MAC Control Register A (0x0008)

  Bits set in this data field will *reset* or disable the corresponding bit in the Motorola Media Access Control chip (MC68838), MAC Control Register A. All bits that are not set will remain in their previous state. For example, a value of 0x0010 disables the “copy-grp-llc” bit. This parameter is optional. For more information, see the Motorola data book.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

GET HARDWARE PARAMETERS

Group: FDDI
Type: Request
Command ID: 0x10F

Get Hardware Parameters is used to read various hardware registers on the controller. One or any combination of the supported hardware parameters may be read in a single request. No specific ordering of parameters is required.

The controller issues Report Hardware Parameters (page 171) in response to this request. The format of the Get Hardware Parameters is detailed in Table 4-34:

Table 4-34. Get Hardware Parameters Request

Size (in bytes)	Description
8	Command Control Block
4	Command Tag
4	Parameter Count
2	Parameter Type
2	Reserved
2	Parameter Type
2	Reserved
	:
	:
2	Parameter Type
2	Reserved

The fields in the Get Hardware Parameters request are:

- COMMAND CONTROL BLOCK (8 bytes)

  This field contains the Command Control Block. The command ID must be set to 0x10F for the Get Hardware Parameters request.

- COMMAND TAG (4 bytes)

  This field contains a host-supplied tag generated by the device driver. It is usually the address of a host memory descriptor or the virtual address of a data buffer.

- PARAMETER COUNT (4 bytes)

  This field contains the number of parameters requested in the remainder of this command. The same parameter can be requested multiple times in the same command, if desired.

- PARAMETER TYPE *n* (2 bytes)

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The code in this field identifies the type of parameter being requested. No specific ordering of parameters is required.

Table 4-35. Host-Readable Registers and Functions

Code	Parameter Type
0x0001	Reserved
0x0002	Reserved
0x0003	Reserved
0x0004	MAC Mode
0x0005	Disable/Enable FCS Generation
0x0006	Optical Bypass Relay Control
0x0007	Set MAC Control Register A
0x0008	Reset MAC Control Register A

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

GET CAM ADDRESS

Group: FDDI
Type: Request
Command ID: 0x110

Get CAM Address is used to query a specific entry in the 5211's 32-entry content-addressable memory (CAM). The host returns the requested CAM entry by issuing Report CAM Address (page 174).

The format of Get CAM Address is detailed in Table 4-36:

Table 4-36. Get CAM Address Request

Size (in bytes)	Description
8	Command Control Block
4	ID Number of Response Channel
4	Command Tag
2	CAM Address
2	Reserved

The fields in the Get CAM Address request are:

- COMMAND CONTROL BLOCK (8 bytes)

  This field contains the Command Control Block. The command ID must be set to 0x110 for the Get CAM Address request.

- ID # OF RESPONSE CHANNEL (4 bytes)

  This field identifies the board-to-host channel to which the 5211 should write the Report CAM Address response. If the field is 0, the controller uses the default response channel. The default response channel is the initial board-to-host channel created at RC boot time.

- COMMAND TAG (4 bytes)

  This field may contain a host-generated command tag, if needed for the application. The 5211 does not use the tag in any way. It is echoed back to the host in the CAM Address Response.

- CAM ADDRESS (2 bytes)

  This field identifies which entry in the 32-entry CAM is queried. Possible values are 0x0000 – 0x1F.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Tx RESPONSE

Group: FDDI
Type: Response
Command ID: 0x181

Tx Response is generated by the 5211 after it executes a Transmit Raw Data or a Transmit Datagram request. It will not be issued unless responses are enabled in the Response Class field of the completed command.

This response indicates whether the 5211 has successfully queued the data for transmission in its communications buffer. The format of Tx Response is shown below in Table 4-37:

Table 4-37. Tx Response

Size (in bytes)	Description
8	Command Control Block
4	Command Tag
4	Transmit Completion Status

The fields in Tx Response are:

- COMMAND CONTROL BLOCK (8 bytes)

  This field contains the Command Control Block. The controller sets the command ID field to 0x181 for Tx Response.

- COMMAND TAG (4 bytes)

  This field contains the host-supplied command tag from the completed transmit command. It is echoed unchanged back to the host by the controller. Typically, the tag identifies the host memory resources associated with the transmit command.

- TRANSMIT COMPLETION STATUS (4 bytes)

  This field reports whether the 5211 was able to copy the frame specified by the transmit command into its communications buffer. Possible status codes are shown in Table 4-38.

Table 4-38. Transmit Completion Status Field

Code	Description
0x0	Normal. Frame queued for transmission.
0x1	Reserved.
0x2	Ring is down. Frame is discarded.
0x3	VMEbus error. Frame is discarded.

If the status code is 0, the host may reuse any host memory space containing outgoing data associated with the transmit command. Otherwise, the data should

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

be either retained in host memory for later transmission or discarded, depending on the problem.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Rx FRAME

Group: FDDI
Type: Indication
Command ID: 0x182

Rx Frame is used by the 5211 to pass a received frame from fiber-to-host without interpretation or analysis. Media-specific header data, such as the FDDI MAC header, appears at the beginning of the receive buffer.[1]

In addition to uploading link-level packets, Rx Frame is also used to upload SMT frames, provided the host has enabled SMT frame tracing via the SMT Trace directive (page 152). The receive buffer contains an actual SMT packet as it was received from the fiber.

The 5211 writes the Rx Frame indication itself into the default response channel (that is, the initial board-to-host channel created at RC boot time). The received frame is DMAed into a host-resident "receive buffer." If the frame cannot fit into a single receive buffer, the 5211 "scatters" the frame across multiple buffers that are not contiguous.

When the 5211 uploads a received frame to the host, it aligns the frame on a longword boundary. Since the FDDI MAC frame header is 13 bytes long, some padding is necessary to longword-align the frame. The 5211 places a 3-byte pad at the beginning of the first frame fragment. Subsequent fragments of that frame are not offset. Their contents are written to the beginning of receive buffers.

The format is shown in Table 4-39:

Table 4-39. Rx Frame Indication

Size (in bytes)	Description
8	Command Control Block
2	Receive Status
2	Frame Status
2	Length in bytes of received frame
2	Number of fragments in received frame
4	Command Tag for data fragment 1
4	Length in bytes of data fragment 1
4	Command Tag for data fragment 2
4	Length in bytes of data fragment 2
	:
	:
4	Command Tag for data fragment n
4	Length in bytes of data fragment n

1. The term "receive buffer" refers to an area of host memory which the host allocates to the 5211 for use when uploading frames to the host.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The fields in the Rx Frame indication are:

- COMMAND CONTROL BLOCK (8 bytes)

  This field contains the Command Control Block. The controller sets the command ID field to 0x182 for the Rx Frame indication.

- RECEIVE STATUS (2 bytes)

  This field provides information concerning the ability of the 5211 to receive the frame and move it to the host. Currently, no status codes are defined; and the field is cleared to 0.

- FRAME STATUS (2 bytes)

  This field contains the E, A, & C bits as defined in the MAC specification. Its format is shown in Figure 4-39:

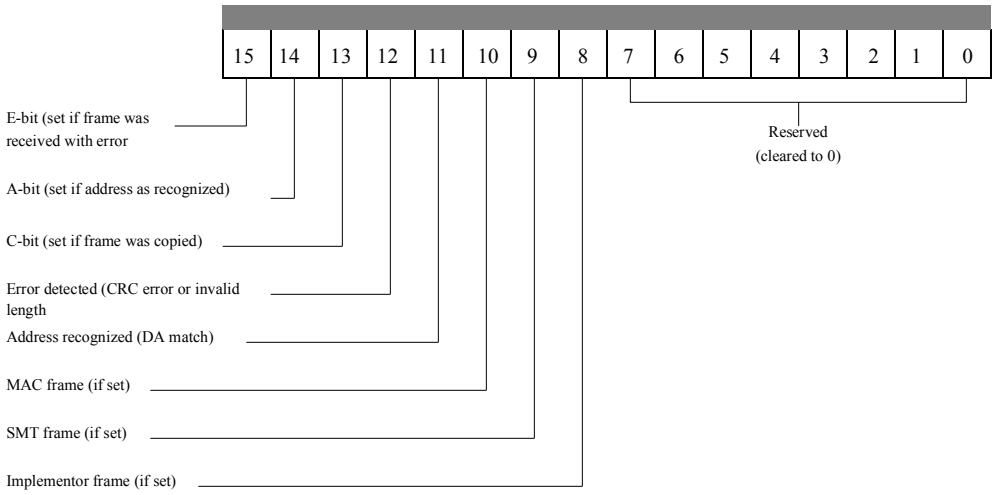


Figure 4-39. Frame Status Field

- LENGTH IN BYTES OF RECEIVED FRAME (2 bytes)

  This specifies the total byte length of the received frame. This value includes the 13-byte MAC header and 8-byte LLC header. It does *not* include the 3-byte pad used to longword-align the frame.

- NUMBER OF FRAGMENTS IN RECEIVED FRAME (2 bytes)

  The value is the number of fragments listed in the following list.

- COMMAND TAG FOR DATA FRAGMENT *n* (4 bytes)

  This is the command tag of a buffer list element which points to the received data. This tag was supplied by the host to the controller via the Set Up Receive Buffers directive (see page 130).

- LENGTH IN BYTES OF DATA FRAGMENT *n* (4 bytes)

  This field contains the length of the data fragment referred to by the corresponding Command Tag parameter.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

REPORT STATION ADDRESS

Group: FDDI
Type: Response
Command ID:0x184

The 5211 issues Report Station Address as a result of having received a Request Station Address from the host. This response contains the controller's station address in both Internet and bit transmission format. The format is detailed in Table 4-40.

Table 4-40. Report Station Address

Size (in bytes)	Description
8	Command Control Block
4	Command Tag
6	Station Address in Internet Format (Canonical)
6	Station Address in Bit Transmission Format (Non-Canonical)

The fields in Report Station Address response are:

- COMMAND CONTROL BLOCK (8 bytes)

  This field contains the Command Control Block. The controller sets the command ID field to 0x184 for Report Station Address.

- COMMAND TAG (4 bytes)

  This field contains the host-supplied command tag from the Request Station Address command to which the 5211 is responding.

- STATION ADDRESS, INTERNET FORMAT (6 bytes)

  This field contains the controller's station address, represented using the format specified by Internet Protocol (IP). It is stored byte-wise, with the most significant byte first.

- STATION ADDRESS, BIT TRANSMISSION FORMAT (6 bytes)

  This field contains the controller's station address in bit transmission format. It is stored byte-wise, with the least significant byte first.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

RING STATUS INDICATION

Group: FDDI
Type: Indication
Command ID: 0x185

The Ring Status Indication reports a change in the state of the FDDI ring. The 5211 writes this message asynchronously to the default response channel. The default response channel is the initial board-to-host channel created by the host when it boots the RC Interface.The host can detect pending messages by polling the channel and/or enabling interrupts.

This indication is not issued until the host attempts to connect the station to the ring (see Ring Control directive, page 139). Once the attempt is made, the board issues the indication each time it successfully completes a command to connect to (or disconnect from) the ring. It also issues the indication when the ring goes down for any reason.

Since the Ring Status Indication is used to signal a *change* in the ring status, it is not issued if the 5211 fails in an attempt to connect to the ring. It is not issued if the host instructs the 5211 to connect or disconnect when the board is already in the specified state. The format of the Ring Status Indication is detailed below in Table 4-41:

Table 4-41. Ring Status Indication

Size (in bytes)	Description
8	Command Control Block
4	Ring Status

The fields in the Ring Status Indication are:

- COMMAND CONTROL BLOCK (8 bytes)

  This field contains the Command Control Block. The controller sets the command ID field to 0x185 for Ring Status Indication.

- RING STATUS (4 bytes)

  This field contains the status of the ring. Table 4-42 shows the possible status codes.

Table 4-42. Ring Status Codes

Status Code	Description
0x1	Ring up. Station is connected to the network and may transmit and receive frames.
0x2	Ring down. Station is disconnected from ring.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

CAM ADDRESS RESPONSE

Group: FDDI
Type: Response
Command ID: 0x187

The 5211 issues a CAM Address Response as a result of having received either a Set CAM Address or Clear CAM Address request from the host. It reports whether the host was successful in its attempt to modify the CAM.



NOTE

CAM Address Response is not the same as Report CAM Address. The latter is only issued in response to the Get CAM Address request. For an overview of the CAM-related commands, see page 148.

The format is shown in Table 4-43.

Table 4-43. CAM Address Response

Size (in bytes)	Description
8	Command Control Block
4	Command Tag
4	Status

The fields in the CAM Address Response are:

- COMMAND CONTROL BLOCK (8 bytes)

  This field contains the Command Control Block. The controller sets the command ID field to 0x187 for CAM Address Response.

- COMMAND TAG (4 bytes)

  This field contains the host-supplied command tag from the CAM-related command to which the 5211 is responding.

- STATUS (4 bytes)

  This field reports the success or failure of the host's attempt to modify the CAM contents. Possible values are shown in Table 4-44.

Table 4-44. Status Codes in CAM Address Response

Code	Description
0x0	Request successful.
0x1	CAM table full. Request failed.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table 4-44. Status Codes in CAM Address Response (cont)

Code	Description
0x2	CAM address already present when trying to SET. Request failed.
0x3	CAM address not present when trying to CLEAR. Request failed.
0x4	CAM feature not supported on this version of the board. Request failed.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

MIB ATTRIBUTE RESPONSE

Group: FDDI
Type: Response
Command ID: 0x188

The 5211 issues a MIB Attribute Response in reply to a host-issued Get MIB Attribute request. To interpret the data returned by this response, you must be familiar with the TLV parameter structure of MIB attributes. For details, refer to the SMT 6.2/7.3 specifications. The format is shown in Table 4-45.

Table 4-45. MIB Attribute Response

Size (in bytes)	Description
8	Command Control Block
4	ID number of Response Channel
4	Command Tag
4	Attribute Name
4	Data Area Needed for Attribute(s)
4	Attribute Index
N	Attribute Definition (Variable-length)

The fields in the MIB Attribute Response are:

- COMMAND CONTROL BLOCK (8 bytes)

  This field contains the Command Control Block. The controller sets the command ID field to 0x188 for MIB Attribute Response.

- ID # OF RESPONSE CHANNEL (4 bytes)

  This field echoes back the “ID # of Response Channel” field from the Set MIB Attribute request to which the 5211 is responding.

- COMMAND TAG (4 bytes)

  This field echoes back the host-supplied command tag from the Set MIB Attribute request to which the 5211 is responding.

- ATTRIBUTE NAME (4 bytes)

  This field contains the Parameter_Type of the attribute or MIB Group supplied by the Get MIB Attribute request.

- DATA AREA NEEDED FOR ATTRIBUTE(s) (4 bytes)

  This field contains a multiple of 0x24 (36 bytes for each attribute being reported). The host does *not* use this value to “unpack” the attribute(s).

- ATTRIBUTE INDEX (4 bytes)

  This field echoes back the Attribute Index field specified in the request to which the 5211 is responding.

- ATTRIBUTE DEFINITION (Variable-Length)

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

This field contains a byte stream with the requested attribute(s). The data is formatted as follows:

- The data for each attribute is presented *exactly* as defined in the SMT 6.2/7.3 specifications, starting with the Parameter_Type and continuing to the end of the attribute definition.
- If multiple attributes are being returned, the data is packed. No terminator is placed between attributes or at the end of the data.

  To extract the data from the Attribute Definition field, use the first eight bytes of the field to identify the Parameter_Type and Parameter_Length of the first attribute. Use the Parameter_Length value to determine where the attribute definition ends. If the controller is reporting only one MIB attribute, this should correspond with end of the MIB Attribute Response, excluding any pad bytes used to longword-align the response.

  If board is returning multiple attributes, use each successive Parameter_Length to locate the start of the next attribute definition. According to SMT 6.2/7.3 specifications, attributes are order-independent. Therefore, the host should not assume that the attributes will be presented in the same order that they appear in the SMT specifications.

  It is advisable for the host to maintain tables of useful information such as the size of MIB groups and the Parameter_Type and Parameter_Length associated with specific attributes. This information is useful for performing sanity checking.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

SMT EVENT INDICATION

Group: FDDI
Type: Indication
Command ID: 0x189

If the host has enabled event tracing via the SMT Trace directive, the 5211 issues an SMT Event Indication each time it detects an SMT event.

The 5211 writes the SMT Event Indication into the default response channel (that is, the initial board-to-host channel created at RC boot time). The format is shown below in Table 4-46:

Table 4-46. SMT Event Indication

Size (in bytes)	Description
8	Command Control Block
4	Attribute ID
N	Variable-Length SMT message

The Fields in the SMT Event Indication are:

- COMMAND CONTROL BLOCK (8 bytes)

  This field contains the Command Control Block. The controller sets the command ID field to 0x189 for SMT Event Indication.

- ATTRIBUTE ID (4 bytes)

  This value is the ID of the MIB attribute associated with tis event. These IDs are given in the SMT 6.2/7.3 specifications. For example, if the controller is reporting an event concerning *fddiMACNeighborChange*, it writes the value 0x208F to this field.

- VARIABLE-LENGTH SMT MESSAGE

  This field consists of data containing the event message. The data in this field matches the SMT 6.2/7.3 definition of the MIB attribute. For example, the format of a message concerning the attribute *fddiMACNeighborChange* is shown in the SMT 6.2/7.3 specifications.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

REPORT HARDWARE PARAMETERS

Group: FDDI
Type: Response
Command ID: 0x18A

Report Hardware Parameters is issued in response to a Get Hardware Parameters request. It reports the current mode of operation of the hardware registers specified in the request. The format of response is shown below in Table 4-47:

Table 4-47. Report Hardware Parameters Response

Size (in bytes)	Description
8	Command Control Block
4	Command Tag
4	Parameter Count
2	Parameter Type
2	Parameter Data
2	Parameter Type
2	Parameter Data
	:
	:
2	Parameter Type
2	Parameter Data

The fields in the Report Hardware Parameters response are:

- COMMAND CONTROL BLOCK (8 bytes)

  This field contains the Command Control Block. The host sets the command ID to 0x18A for Report Hardware Parameters.

- COMMAND TAG (4 bytes)

  This field contains the host-supplied command tag from the command to which the 5211 is responding.

- PARAMETER COUNT (4 bytes)

  This field contains the number of parameter reports supplied in the remainder of this command. There is one pair of Parameter Type and Parameter Data fields for each parameter reported. For example: If this field contains 0x2, there are two pairs of Parameter Type/Parameter Data fields in the remainder of the response.

- PARAMETER TYPE *n* (2 bytes)

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The code in this field identifies the type of parameter reported. Table 4-48 lists the codes for the available registers and functions.

Table 4-48. Parameter Types for Report Hardware Parameters

Code	Parameter Type	Valid Parameter Data
0x0004	MAC Mode	See Table 4-49
0x0005	Disable/Enable FCS Generation	See Table 4-52
0x0006	Optical Bypass Relay Control	See Table 4-53
0x0007	Set MAC Control Register A	
0x0008	Reset MAC Control Register A	

- PARAMETER DATA *n* (2 bytes)

  This field contains either: 1) a mapped field of control bits written to a specific register, or 2) a code indicating the mode of a specific function. The data written to this field depends on the Parameter Type field with which it is paired (that is, the 2-byte Parameter Type field immediately preceding it). The following tables describe the data for each of the parameter types listed in Table 4-48.

Table 4-49. MAC Mode

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													BS	PMode	

- MAC Mode Assignments

Table 4-50. PMode

Code	Description
0x0	Receive if the destination address matches the local MAC address
0x1	Receive if the destination or source address matches the local MAC address
0x2	Receive all except source address frames
0x3	Receive all frames

Table 4-51. Bridge Stripping

Value	BS
0x0	Bridge Stripping mode disabled.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table 4-51. Bridge Stripping (cont)

Value	BS
0x1	Bridge Stripping mode enabled.

- FCS Generation

Table 4-52. Returned FCS Generation Modes

Code	Description
0x0	CS is enabled. Frame Check Sequence is sent on transmit frames.
0x1	FCS is disabled.

- Optical Bypass Control

Table 4-53. Returned Optical Bypass Control Modes

Code	Description
0x0	Optical Bypass Relay not present.
0x1	Optical Bypass Relay present and bypass mode enabled.
0x2	Optical Bypass Relay present and THRU mode enabled.

- Set MAC Control Register A

  The return values reflect those specified in Set Hardware Parameters (page 153).
- Reset MAC Control Register A

  The return values reflect those specified in Set Hardware Parameters (page 153).

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

REPORT CAM ADDRESS

Group: FDDI
Type: Response
Command ID: 0x18B

The 5211 issues Report CAM Address in response to a host-issued Get CAM Address request. The 5211 returns the 48-bit MAC address stored at the requested location.



NOTE

CAM Address Response is not the same as Report CAM Address. The latter is only issued in response to the Get CAM Address request. For an overview of the CAM-related commands, see page 148.

The format of the response is detailed in Table 4-54:

Table 4-54. Report CAM Address

Size (in bytes)	Description
8	Command Control Block
4	Command Tag
2	Reserved (returns zero)
6	MAC Address

The fields in Report CAM Address are:

- COMMAND CONTROL BLOCK (8 bytes)

  This field contains the Command Control BlocK. The controller sets the command ID field to 0x18B for Report CAM Address.

- COMMAND TAG (4 BYTES)

  This field contains the host-supplied command tag from the CAM-related command to which the 5211 is responding.

- MAC ADDRESS (6 bytes)

  This field contains the 48-bit MAC address stored at the requested CAM location.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

SPECIFICATIONS 5

ARCHITECTURE

Processor: Motorola 68EC040

Network Interface:Motorola chipset

Data Communications Buffer:1 Mbyte/128K (expandable)

VMEBUS SPECIFICATIONS

- Capable of VMEbus data transfers in excess of 30 Mbytes/sec. @ 32-bits
- Capable of VMEbus data transfers in excess of 40 Mbytes/sec. @ 64-bits
- Supports any VME alignment.
- Options (VMEbus Master):
  - Addressing: 16-bit, 24-bit, and 32-bit addressing
    16-bit, 32-bit, and 64-bit data transfers
  - Address Modifiers: all options
  - Requester options: any one of R(0), R(1), R(2), R(3) (jumper-selectable)
  - RWD (Release When Done)
- Options (VMEbus Slave)
  - Master Addressing: A16, A24, A32, D8, D16, D32, D64
  - Slave (short I/O): A16, D8, D16, D32
  - Slave (Buffer): A24, D8, D16, D32
  - Address Modifiers: 0x29, 0x2D
  - Interrupter options: D08 any vector



NOTE

Slave access to VRAM buffer memory is enabled by a jumper and is OFF by default. Host access is not required nor recommended when using Interphase firmware.

FDDI SPECIFICATIONS

- Configurations
  - Single PHY single MAC (SAS-SM)
  - Dual PHY single MAC (DAS-SM)

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

- SMT
  - FDDI Station Management (SMT) standards, Revision 6.2 or 7.3.
- Optical Bypass
  - I/O port with up to 300 mA current sink for control of external optical bypass switch.
- Connectors
  - ST-compatible optical connectors or MIC (Media Interface Connector) connectors and receptacles 6-pin mini-DIN optical bypass control

TIMER DEFAULTS

- The default value of TVX
  - 2.5 milliseconds
- The default for T_req
  - 165 milliseconds
- Value of T_Min
  - 4 milliseconds
- Value of T_Max
  - 165 milliseconds

POWER REQUIREMENTS

In hardware variation levels H05211-008, REVxxx and earlier, the requirements are:

- 4.0 A typical @ +5VDC/25 degrees C
- 100 mA typical @ +12V DC/ 5 degrees C

MECHANICAL (NOMINAL)

The 5211 occupies one 6U single-height VMEbus slot.

- Width:
  - 9.20 in (233 mm)
- Height:
  - 6.30 in (160 mm)
- Thickness:
  - 20 mm
- Weight:
  - 566 grams

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

OPERATING ENVIRONMENT

- Temperature:
  - 0 - 55 degrees C
- Relative Humidity
  - 10 - 95% noncondensing
- Altitude
  - 0 - 15,000 feet
- Air Flow
  - 250 LFM minimum

STORAGE ENVIRONMENT

- Storage Temperature
  - 0 - 125 degrees C
- Storage Humidity
  - 5% - 95% noncondensing
- Storage Altitude
  - -1000 to 50,000 feet

LEDs

The 5211 has seven LEDs — labelled LED1 through LED7 in Figure 2-1.

LED1 is either red or green. When the board is powered-up or reset, the LEDs will toggle on and off for about 10 seconds while the 5211 performs its power-up diagnostics. Once the 5211 firmware has booted up and the Common Boot Interface is active, LED1 turns green. Once the host boots the RC Interface, LED1 returns to toggling red/green again.

LED5, LED6 and LED7 indicate the state of the ring. LED6 signals THRU, LED5 signals the status of the primary ring, and LED7 signals the status of the secondary ring. If the THRU bit is set (that is, LED6 is lit), one of the other bits indicates the THRU condition. If the THRU bit is not set, the other bits indicate WRAP condition. All bits off indicates a "ring down" condition. Table 5-1 is a summary of all possible ring conditions which can be signalled by these LEDs. Ring states are defined in the SMT Revision specifications.

Table 5-1. Ring State LEDs (LED5-LED7)

Ring State	PRI LED5	THRU LED6	SEC LED7
Isolated	Off	Off	Off
Wrap_A	On	Off	Off
Wrap_B	Off	Off	On

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table 5-1. Ring State LEDs (LED5-LED7) (cont)

Ring State	PRI LED5	THRU LED6	SEC LED7
THRU_A	On	On	Off

LED4 indicates the Master DMA hardware has data available to transfer. LED3 indicates an active request to the VRAM buffer while another requestor is the owner. LED2 indicates an active request to the SRAM buffer while another requestor is the owner.



NOTE

LEDs 2 through 4 may not be present in future revisions of the 5211.

DEFAULT REPORT/COMMAND PARAMETERS

This section states the default values of parameters which can be adjusted by using the generic RC command set. The generic RC commands referred to below are documented in chapter 4.

- Default Size of DMA Transfers

  The 5211 supports 8-bit, 16-bit, 32-bit, and 64-bit DMA burst transfers and has a 10-bit transfer counter. The default is to not use burst, which means it will relinquish the bus only after the DMA transfer is complete. See SET DMA BURST in Chapter 4 for more information.

  – Burst size (8-bit transfers) = 256 bytes per burst
  – Burst size (16-bit transfers) = 512 bytes per burst
  – Burst size (32-bit transfers) = 1024 bytes per burst
  – Burst size (64-bit transfers) = 2048 bytes per burst

    The burst sizes can be adjusted using the Set DMA Burst directive.

- Default VMEBus Master Timeout

  The 5211's default bus timeout is 100 ms. To set an explicit timeout value, issue the Set Bus Timeout directive.

- Error Messages

  The generic RC command set includes two indications which can be used by the controller to pass error messages to the host. These indications are issued asynchronously by the 5211 to the default response channel (that is, the initial board-to-board channel created at RC boot time). The two indications are:

  – **Error Message (Command ID 0x82)**

    No 5211-specific errors are currently defined for the Error Message indication. For information on the generic error messages, Refer to the command description in Chapter 4.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

– **Report Printf Call (Command ID 0x81)**

Report Printf Call is delivered to the host whenever an on-board software module calls the *printf()* routine. It is typically used to report boot-time configuration information and non-fatal anomalies. It can also provide debugging information which is useful in the development of firmware or downloaded software. The body of this message is a null-terminated string that should be treated as a status message for the operator/user.

The controller writes Report Printf Call to the default response channel (i.e. the initial board-to-host channel created at RC boot time).

- Maximum Slave Access Time

The 5211's Maximum Slave Access time is 4 microseconds.

For more information, refer to the command description of Report Printf Call in chapter 4.

OPTICAL BYPASS SWITCH PIN ASSIGNMENTS

Pin	Signal Description
1	Secondary switch power
2	Primary switch power
3	Primary switch drive
4	Secondary switch drive
5	OBS sense
6	Ground (connected to pin 5 when OBS is plugged in)

P1 AND P2 CONNECTORS PINOUTS (VMEbus)

All versions of the 5211 have the same VME P1 connector configuration. See Table 5-2 for a list of the P1 signals. Some versions of the 5211 utilize only row B on the P2 connector. (See Table 5-3.) Others use all three rows on the P2 connector. (See Table 5-3.) To determine which board version you have, look at the P2 connector on your board. If it has one row of pins (32 pins), it uses P2 row B only. If it has three rows of pins (96 pins), it also uses P2 rows A and C.

Table 5-2. P1 Connector - VMEbus

Pin	Row A Signal Mnemonic	Row B Signal Mnemonic	Row C Signal Mnemonic
1	D00	BBSY*	D08
2	D01	BCLR*	D09
3	D02	ACFAIL*	D10
4	D03	BG0IN*	D11
5	D04	BG0OUT*	D12
6	D05	BG1IN*	D13
7	D06	BG1OUT*	D14

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table 5-2. P1 Connector - VMEbus (cont)

Pin	Row A Signal Mnemonic	Row B Signal Mnemonic	Row C Signal Mnemonic
8	D07	BG2IN*	D15
9	GND	BG2OUT*	GND
10	SYSCLK	BG3IN*	SYSFAIL*
11	GND	BG3OUT*	BERR*
12	DS1*	BR0*	SYSRESET*
13	DS0*	BR1*	LWORD*
14	WRITE*	BR2*	AM5
15	GND	BR3*	A23
16	DTACK*	AM0	A22
17	GND	AM1	A21
18	AS*	AM2	A20
19	GND	AM3	A19
20	IACK*	GND	A18
21	IACKIN*	SERCLK	A17
22	IACKOUT*	SERDAT*	A16
23	AM4	GND	A15
24	A07	IRQ7*	A14
25	A06	IRQ6*	A13
26	A05	IRQ5*	A12
27	A04	IRQ4*	A11
28	A03	IRQ3*	A10
29	A02	IRQ2*	A09
30	A01	IRQ1*	A08
31	-12V	+5VSTDBY	+12V
32	+5V	+5V	+5V

For a description of the signals on the P1 connector, refer to the VMEbus specifications.

Table 5-3. P2 Connector - VMEbus (P2 Row B)

Pin	Row B Signal Mnemonic
1	+5v
2	GND
3	RESERVED
4	A24
5	A25
6	A26
7	A27
8	A28
9	A29
10	A30
11	A31
12	GND

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

DIAGNOSTICS A

INTRODUCTION

The 5211 diagnostic commands are compatible with the 4211 command/response structure, with some additional commands to help isolate failures.

The 5211 supports a variety of power-up and host-controlled diagnostics. Power-up diagnostics execute automatically each time the system is powered-up or the board is reset. Host-controlled diagnostics may be invoked while the Common Boot interface is running.

- In power-up mode, quick memory, frontend, and DMA tests are performed. Memory is zeroed out if all tests pass. With host-controlled diagnostics, extended memory tests are available. Refer to the Power-up test in each major test category (Memory, Frontend, DMA).
- VME DMA tests are not executed in power-up mode, but can be run in host-controlled mode.
- External loopback tests are not executed during power-up diagnostics, but can be invoked by the host after power-up.
- VMEbus is not accessed during the power-up DMA test, but can be invoked by the host after power-up.
- Some CAM tests are not executed during power-up diagnostics, but can be invoked by the host after power-up.

At reset, LED1 is RED The first thing the CPU does is turn LED1 GREEN. This indicates that the CPU is functional and has started executing code. The board then performs a buffer RAM test. If the buffer RAM test fails, both the LED alternately toggles RED and GREEN and the controller stops. If the RAM test passes, the controller runs all the remaining power-up diagnostics. While the tests are executing, the LED alternately toggles RED and GREEN.

For the purpose of setting system timeouts, most tests take less than 1 millisecond.

Once the diagnostics finish, the LED stops toggling and control is passed to the Common Boot interface. If all the tests complete successfully, the Command Status Word will contain the ASCII value “CBOK” (0x43424F4B), and the LED will remain GREEN. If any of the power-up tests fail, the Command Status Word will contain the ASCII value “FAIL” (0x4641494C), and LED1 will remain RED. In this event, the Command Status Word will be followed by a series of test-specific fields describing the failure. The returned fields of a failed power-up test typically match those of its host-invoked counterpart. Therefore, to identify which power-up test failed, compare the actual returned data to the returned data structures described in the DIAG command.The major test numbers are listed in the following table:

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table A-1. Quick Reference to Power-up Diagnostics

Major Test	Major Test Type	Minor Test	Possible Errors
0x0	Execute ALL Major Tests 1 through 7	NA	If test fails, the returned status block will identify the major test, minor test (if any), and error code of the failed test
0x1	Memory Tests	0x0 PUP_TEST 0x1 PRAM 0x2 SRAM 0x3 BRAM	0x1 Test Failed
0x2	FDDI Front End	0x0 PUP_TEST 0x1 “Reserved” 0x2 ELM_SHORT_LOOP 0x3 ELM_EB_LOOP 0x4 EXT_LOOP 0x5 ELM_SHORT_LOOP2 0x6 ELM_EB_LOOP2 0x7 EXT_LOOP2 0x8 ELM_LOOP 0x9 ELM_LOOP2 0xA FCG_LOOP 0xB FCG_LOOP2 0xC REPORT_BIST	0x2E DIAG_ERROR is written to the CB Last Command Status, FAIL is written to the Command Status Word, followed by the appropriate error data
0x3	VME DMA Diagnostics	0x0 PUP_TEST 0x3 SWAP 0x4 RAW DMA 0x5 READ BUFFER 0x6 WRITE BUFFER 0x7 VIRQ 0x8 DUMP ASIC REGS	See Table A-4 for error codes
0x4	Not Used	NA	NA
0x5	Not Used	NA	NA
0x6	Not Used	NA	NA
0x7	CAM Test	NA	0x15 CAM_COMPARE_FAIL 0x16 CAM_TIMEOUT 0x17 CAM_TIMEOUT2

ISSUING COMMANDS

Diagnostic commands are issued through Short I/O (SHIO). The procedure for the issuing commands is outlined below.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

SHORT I/O

1. Build the command block directly behind the “Command Status Word” in short I/O (SHIO).
2. Write the ASCII word “DIAG” or “TEST” over the command status word, which should currently read as ASCII “CBOK.”
3. Wait for the Command Status Word to read either “CBOK” or “FAIL.”

TEST FAILURES

If any test fails, the 5211 writes the ASCII value “FAIL” (0x464149C) to the Command Status Word, followed by a set of fields describing the failure.



NOTE

If a diagnostic test fails, please make a note of the returned data. This will enable Interphase to provide you with better technical assistance and faster repair turnaround.

COMMAND/RESPONSE STRUCTURE

Most of the commands in the Common Boot command set are generic. Their definition is the same for any controller. Their definition is the same for any controller that supports the Common Boot Interface. The DIAG command, however, is an exception. It is used to perform a variety of controller-specific tests. Therefore, the definition of its fields varies depending on the controller *and* on the type of test being performed.

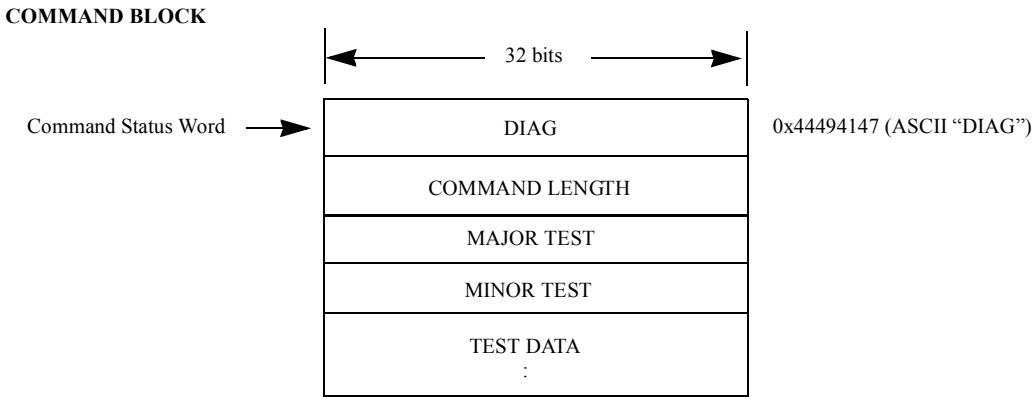


Figure A-1. Structure of the DIAG Command

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

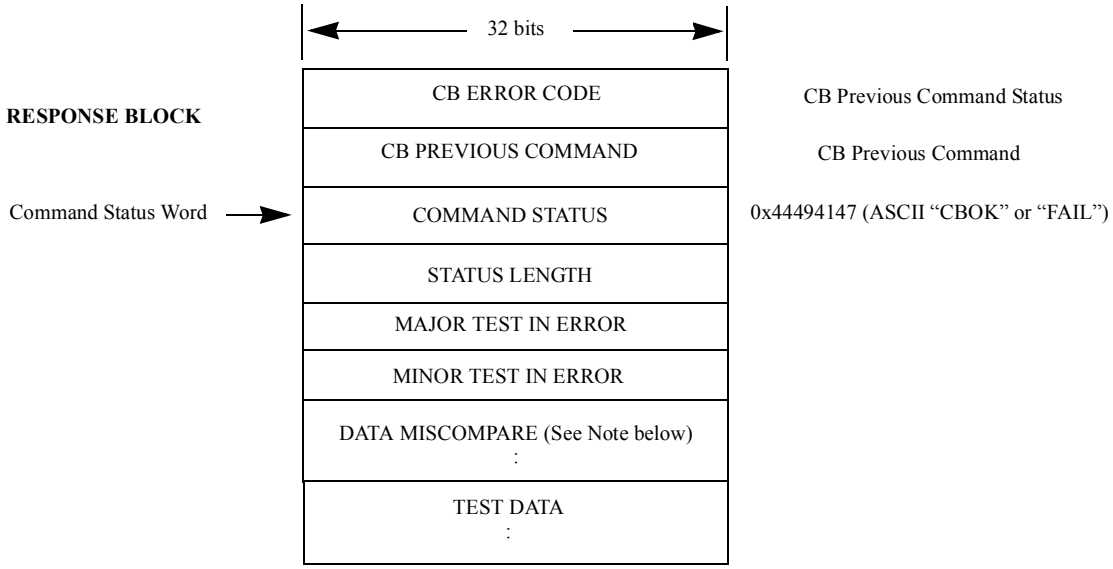


Figure A-2. "FAIL" Returned Status Block



NOTE

The following fields are for miscompares only: Base Internal Address, Address Offset, Expected Data, Data Found

The remainder of this section describes the various diagnostic tests.

MEMORY TEST DIAGNOSTICS

MEMORY TESTS - MAJOR TEST CODE 0x1

There are three distinct memory areas on the 5211. The CPU operates from 512K bytes of static RAM referred to as PRAM. There is another set of static RAMs used by the board referred to as SRAM. The size of this bank of memory is variable up to 1 Mbyte. On-board buffer space and on-board shared memory (short I/O space) are implemented in video RAM chips abbreviated BRAM. The BRAM size is variable up to 4 Mbytes.

There are four tests written to test memory. One is a quick test run at power-up or reset, and the other three are extended diagnostics which can be run by the host using Common Boot commands.

Table A-2. Minor Test Codes for Memory Tests

Minor Test Code	Name	Function
0x0	PUP_TEST ONLY	Quick check of all 3 memory types

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Minor Test Code	Name	Function
0x1	PRAM	Test 512K CPU Static Ram
0x2	SRAM	Test up to 1Mbyte of Static Ram
0x3	BRAM	Test up to 4Mbytes of Video Ram

If the test completes successfully, the 5211 will move the CB **DIAG** command to the **Previous Command** position in short I/O, and write **CBOK** into the **Command Status Word**. See Figure A-3.

Command example:

Name	Value in Hex
Length	0x10
Major	0x1
Minor	0x2
Options	0x0

Figure A-3. Memory Test Command Format

The Options field is shown in Figure A-4:

31	. . .						2	1	0
rsvd	rsvd	rsvd	rsvd	rsvd	rsvd	rsvd	rsvd	rsvd	option

Figure A-4. Memory Test Command Options Field.

OPTIONS - 0x0

- Bit 0

  If set (bit 0 = 1) then controller memory is cleared when the test is complete. If cleared (bit 0 = 0) then the test patterns remain in memory after the test is complete.

- Bits 1 - 31

  These bits are reserved and should be 0.

MEMORY TEST ERROR CODES

The only error code returned by the memory tests is 0x1. Along with this error code, the test will report where the error was found, the data read from that location and the data that was expected. Common Boot defines a location for the previous command (in this case

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

DIAG) and a location for a status for that command. When reporting an error from a diagnostic routine, that status will indicate the CB error code **DIAG_ERROR (0x2e)**. **See Appendix C for CB Error Codes**.

The DIAG error code and results will be found in the normal data structure for CB diagnostics given below.

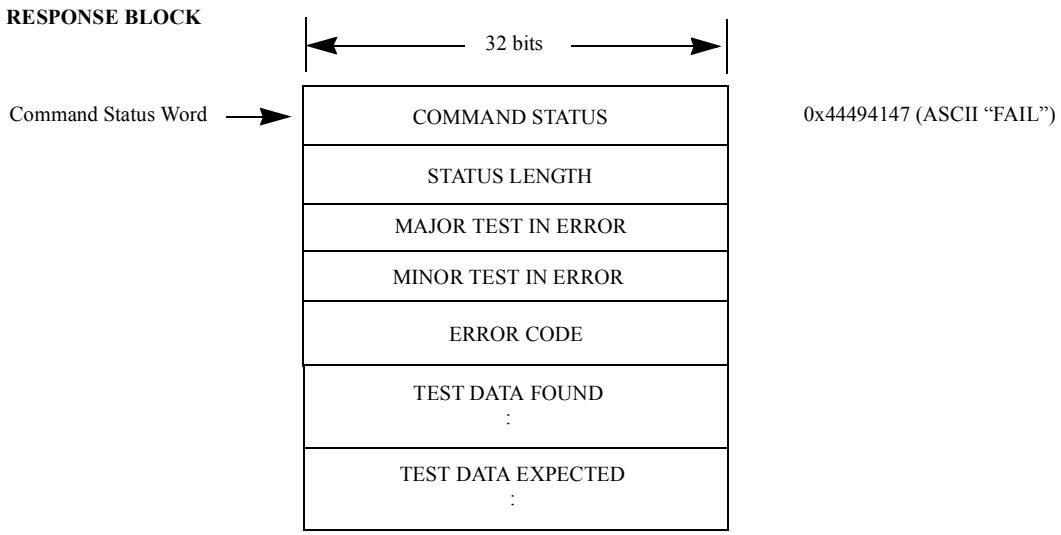


Figure A-5. CB Diagnostic Error Data Structure

POWER-UP DIAGNOSTIC TEST DESCRIPTION

This test is run at power-up or system reset. It is an abbreviated test run on all three segments of memory. The first test checks the integrity of the data bus. If no problem is discovered, BRAM, SRAM, and then the PRAM are tested. The address bus, to each memory space, is exercised to find address lines which are stuck, shorted, or disconnected.

EXTENDED DIAGNOSTIC TEST DESCRIPTION

Each of the extended memory tests perform the same operations. They will write each longword with its address, then return and check that each is correct. If no problem is discovered, a walking 1 test is performed.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

FRONT END FDDI DIAGNOSTICS

FRONT END FDDI CHIPSET TESTS - MAJOR TEST CODE 0x2

The front end tests check the Motorola FDDI chipset, the associated memory, data paths, hardware, and the physical layer transmitters and receivers.

Table A-3. Minor Test Codes for Front-End Tests

Code	Name	Function
0x0	PUP_TEST	Quick power-up diagnostic.
0x2	ELM_SHORT_LOOP	Port A ELM shortest loopback path.
0x3	ELM_EB_LOOP	Port A ELM longest loopback path. Includes framer and elasticity buffer.
0x4	EXT_LOOP	External loopback. Tests the entire data path.
0x5	ELM_SHORT_LOOP2	Port B ELM shortest loopback path.
0x6	ELM_EB_LOOP2	Port B ELM longest loopback path. Includes framer and elasticity buffer.
0x7	EXT_LOOP2	External loopback. Tests the entire data path.
0x8	ELM_LOOP	Port A ELM path. Includes complete Tx path and Rx path excluding framer and elasticity buffer.
0x9	ELM_LOOP2	Port B ELM path. Includes ELM-FCG data paths.
0xA	FCG_LOOP	Port A path. Includes ELM-FCG data paths.
0xB	FCG_LOOP2	Port B path. Includes ELM-FCG data paths.
0xC	REPORT_BIST	Reports BIST data on MAC and ELMs.
0xD	REPORT_FSI	Returns FSI internal register data.
0xE	REPORT_MAC	Returns MAC internal register data.
0xF	REPORT_ELM	Returns ELM internal register data (both ELMs).

FRONT END TEST DESCRIPTION

The front end diagnostics test the FSI, the MAC, the FDDI ring configuration multiplexers, and associated memory. The depth of the frame path tested is dependant upon the minor test selected.

The external loopback tests require fiber connections from the transmitters to the receivers. For DAS controllers, the A port optical transmitter connects to the B port optical receiver, and the B port optical transmitter to the A port optical receiver. For SAS controllers, the A port optical transmitter connects to the A port optical receiver. If a B port is available, the B port optical transmitter connects to the B port optical receiver. The entire FDDI data path is tested for both DAS and SAS controllers.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The internal loopback tests test one port or the other with no fiber connections required. The short loopback tests send a maximum length frame from buffer memory (via the FSI) through the MAC which adds the MAC header, and finally to the selected ELM. The ELM loops its incoming transmit data back to the MAC.

The ELM_LOOP2 tests take the frame data just before the Tx data output to the FCG and loop it back into its receive data path just after the framer and elasticity buffer. The ELM_EB_LOOP2 tests function the same except that the framer and elasticity buffer are included in the receive data path.

The FCG_LOOP2 tests include the output stage of the ELM, the FCG, and the receive data path of the ELM, including the framer and elasticity buffer. The frame's destination address is matched in the MAC, then receive frame data is passed to FSI which places the data in buffer memory and informs the CPU. The data is compared to the transmitted data. If the data is correct, a second frame is sent with an incorrect destination address to insure that it will not be received and the test is completed.

0x4 - 0x7 — EXTERNAL LOOPBACK TEST

This test verifies the integrity of the FDDI path for both dual attach controllers (DAS) and single attach controllers (SAS).

- Parameters

  Data options:

31 - 16	15 - 2	1	0
loop count	reserved	loop	path

Figure A-6. Bit Structure of the EXTERNAL LOOPBACK TEST Data Options

- Loop Count

  If the loop bit is set and this value is 0x0, then the test will run until a failure occurs or the test is reset. If this field is nonzero, then the test will run the specified number of iterations entered in this field, or until a failure occurs.

- Reserved

  This value should be set to 0x0.

- Loop

  If this bit is set, the loopback test will run until a failure occurs or until the test is reset. If this bit is zero, then the test will execute the number of times specified by the Loop Count.

- Path

  This bit selects the desired loopback path. If the bit is 0, data loops to/from both ports and requires that fiber be connected from Port A TX to Port B RX and Port B TX to Port A RX. If the bit is set to 1, data loops to/from a single port and requires that fiber be connected from Port A TX to Port A RX, or Port B TX to Port B RX.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Command example:

Name	Value in Hex
Length	10
Major	02
Minor	04
Option	1

Figure A-7. External Loopback Test Command Format

Response - see *CB ERROR CODES on page 217*.

POWER-UP DIAGNOSTIC DESCRIPTION

On power-up or reset, the board runs the built-in-self-test (BIST) on the FSI, MAC, and both ELMs. If the parts fail to respond, the error will be reported, otherwise there is no pass/fail criteria for these tests. Test signatures are retrieved from the parts, stored, and compared against known good values. If unrecognized values are discovered, a miscompare error is noted in the stored data. The data can be examined using the REPORT_BIST test. After the BISTs, ELM_SHORT_LOOP tests are run on both ELMs. The test stops on the first failure. The REPORT_BIST function returns a PASS if all tests passed and a FAIL if any test failed. Each test also reports an error code and data. Table A-11 describes the BIST test results.

The test stops on the first failure. There are built-in-self-test (BIST) features on the MAC and ELMs. These tests are also run at power-up or reset. Failure of these tests do not prevent the board from completing its initialization and startup. Results of the tests can be examined using the REPORT_BIST minor test. The REPORT_BIST function returns a PASS if all tests passed and a FAIL if any test failed. Each test also reports an error code and data.

TEST RESULTS

The DIAG command is moved to the CB Previous Command Word. If the test runs successfully, CBOK is written into the Command Status Word. On a failure the DIAG_ERROR code (0x2e) is written to the CB Previous Command Status, FAIL is written to the Command Status Word followed by error data appropriate to the test failure. Figure A-1 describes the CB structure in short I/O that holds commands, status, and error data. Table A-2 through Table A-8 list the error codes and error data for those errors that return error data.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The REPORT_FSI, REPORT_EM, and REPORT_MAC test always indicate a PASS, but still include data retrieved from the parts. The data from each register is written as a 32 bit word regardless of the actual width of the register. They are returned in the order listed in the Motorola documentation for each component.

Table A-4. Error Codes for Front End Tests

Code	Error	Description	Error Data Returned
0x3	BIST_TIMEOUT	BIST failed to complete)MAC or ELM)	yes
0x4	FSI_FAIL	No Response from FSI	no
0x5	FSI_COMMAND_FAIL	FSI command completion time-out	no
0x6	FSI_TIMEOUT1	FSI Tx Ring time-out	no
0x7	FSI_TIMEOUT2	FSI Tx Ring time-out second frame	no
0x8	FSI_TIMEOUT3	FSI Rx Ring time-out	no
0x9	FSI_TIMEOUT4	FSI define Tx Ring time-out	no
0xA	RING_OP_TIMEOUT	MAC RingOp time-out	no
0xB	NO_RX_INDICATION	No Rx indication from FSI	yes (Table A-5)
0xC	RX_ERROR	Rx error indication from MAC	yes (Table A-6)
0xD	RX_CRC_ERROR	Frame FCS error from MAC	yes (Table A-7)
0xE	RX_IER_ERROR	MAC internal error	yes (Table A-8)
0xF	RX_FRAME_MISCOMPARE	Frame data incorrect	yes (Table A-9)
0x10	RX_FRAME_SIZE_ERROR	Rx frame size is not equal to Tx frame size	yes (Table A-10)
0x11	MAC_ADDR_FAIL	Cannot program MAC address	no
0x12	ELM_ERROR1	ELM PCM time-out	no
0x13	ELM_ERROR2	PCM failed to go to NEXT	no
0x14	ELM_ERROR3	ELM failed to reset properly	no

Table A-5. NO_RX_INDICATION

Offset	Data	Description
0x80	0x4641494C	ASCII “FAIL”
0x84	0x18	Length in bytes of error data including length
0x88	0x2	Major Test (front end tests)
0x8C	xxxxxxxx	Minor Test
0x90	0xB	Error Code (NO_RX_INDICATION)
0x94	xxxxxxxx	First Rx Ring Descriptor
0x98	xxxxxxxx	Second Rx Ring Descriptor

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table A-6. RX_ERROR

Offset	Data	Description
0x80	0x4641494C	ASCII “FAIL”
0x84	0x18	Length in bytes of error data including length
0x88	0x2	Major Test (front end tests)
0x8C	xxxxxxxx	Minor Test
0x90	0xC	Error Code (RX_ERROR)
0x94	xxxxxxxx	First Rx Ring Descriptor
0x98	xxxxxxxx	Second Rx Ring Descriptor

Table A-7. RX_CRC_ERROR

Offset	Data	Description
0x80	0x4641494C	ASCII “FAIL”
0x84	0x18	Length in bytes of error data including length
0x88	0x2	Major Test (front end tests)
0x8C	xxxxxxxx	Minor Test
0x90	0xD	Error Code (RX_CRC_ERROR)
0x94	xxxxxxxx	First Rx Ring Descriptor
0x98	xxxxxxxx	Second Rx Ring Descriptor

Table A-8. RX_IER_ERROR (FSI Internal Error)

Offset	Data	Description
0x80	0x4641494C	ASCII “FAIL”
0x84	0x18	Length in bytes of error data including length
0x88	0x2	Major Test (front end tests)
0x8C	xxxxxxxx	Minor Test
0x90	0xE	Error Code (RX_IER_ERROR)
0x94	xxxxxxxx	FSI internal error status

Table A-9. RX_FRAME_MISCOMPARE

Offset	Data	Description
0x80	0x4641494C	ASCII “FAIL”
0x84	0x18	Length in bytes of error data including length
0x88	0x2	Major Test (front end tests)

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table A-9. RX_FRAME_MISCOMPARE (cont)

Offset	Data	Description
0x8C	xxxxxxxx	Minor Test
0x90	0xF	Error Code (RX_FRAME_MISCOMPARE)
0x94	xxxxxxxx	Address of Tx frame data
0x98	xxxxxxxx	Address of Rx frame data
0x9C	xxxxxxxx	Data sent
0xA0	xxxxxxxx	Data received

Table A-10. RX_FRAME_SIZE_ERROR

Offset	Data	Description
0x80	0x4641494C	ASCII “FAIL”
0x84	0x18	Length in bytes of error data including length
0x88	0x2	Major Test (front end tests)
0x8C	xxxxxxxx	Minor Test
0x90	0x10	Error Code (RX_FRAME_SIZE_ERROR)
0x94	xxxxxxxx	Frame size sent
0x98	xxxxxxxx	Frame size received

Table A-11. BIST Test Data

Offset	Data	Description
0x80	0x4641494C	ASCII “FAIL”
0x84	0x18	Length in bytes of error data including length
0x88	0x2	Major Test (front end tests)
0x8C	xxxxxxxx	Minor Test
0x90	0x0 or 0x10	Error Code (PASS or FAIL)
0x94	xxxxxxxx	MAC BIST signature for test 1
0x98	xxxxxxxx	MAC BIST signature for test 2
0x9C	xxxxxxxx	MAC BIST signature for test 3
0xA0	0x0 - 0x3	MAC BIST Pass/Fail code
0xA4	xxxxxxxx	Port A ELM BIST signature
0xA8	xxxxxxxx	Port B ELM BIST signature
0xAc	0x0 - 0x3	Port A ELM Pass/Fail code
0xB0	0x0 - 0x3	Port B ELM Pass/Fail code

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

DMA TEST DIAGNOSTICS

DIAGNOSTIC DMA TEST - MAJOR TEST 0x3

The following sections describe the available diagnostic DMA commands, options, error codes, and power up diagnostics.

The format of the DIAGNOSTIC DMA command block is as shown in Figure A-1. The format of the response block is shown in Figure A-2. The following fields are a part of the response block for the Diagnostic DMA test:

- Data miscompare information - these fields are for miscompares only
  - Base internal address
  - Address offset
  - Expected data
  - Data found
- Addgen general status register - High true values

ADDGEN STATUS REGISTERS

The ADDGEN status registers are shown below.

Table A-12. Addgen General Status Register - High True Values

31-16	15-14	13-12	11-9	8-7	6-4	3	2	1	0
rsvd	rev = 0x01	rsvd	dma chanl has work	rsvd	dma chanl busy	bus tmo intr	bus error intr	count exhst intr	dma done intr

Table A-13. Addgen I/O Status Register - High True Values

31-21	20	19-18	17-15	14-13	12-10	9-8	7-5	4-3	2-0
rsvd	tick timer active	rsvd	dma chanl tmo intr	rsvd	dma chanl bus error	rsvd	dma chnl exhst intr	rsvd	dma chnl done intr

Table A-14. MBC Interrupt Port - Low True Values

31-19	18	17	16	15	14	13	12	11	10
rsvd	Big timer	uart A break	uart A xmit	uart A rcv	uart B break	uart B xmit	uart B rcv	acfail	dbug abort
9	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table A-14. MBC Interrupt Port - Low True Values (cont)

31-19	18	17	16	15	14	13	12	11	10
dboard hi-pri	fsi	dboard lo-pri	mac	elm 0	elm 1	addgen iop	mmux mbox	addgen timer	vme iack

AVAILABLE DMA DIAGNOSTIC COMMANDS

The following DMA Diagnostic commands are available.

Table A-15. Minor Test Codes for DMA Diagnostic Tests

CODE	NAME	FUNCTION
0x0	PUP_TEST	Quick power-up diagnostic.
0x1	SHIO	4211 only; not supported.
0x2	STANDARD DMA	Executes a VME write followed by a VME read. With auto data build and compare.
0x3	SWAP	2211 only, Not supported.
0x4	RAW DMA	Executes a VME read followed by a VME write of the same buffer. No data manipulation.
0x5	READ BUFFER	Executes a VME write with the specified on-board buffer. Builds data according to user parameters.
0x6	WRITE BUFFER	Executes a VME read into the specified on-board buffer. Compares data according to the user parameters.
0x7	VIRQ	Issues a VME interrupt on the given level and with the given vector.
0x8	DUMP ASIC REGS	Reads and prints all registers of the specified ASIC. UART output only.

0x0 - PUP_TEST

This test executes a Read Buffer using the ADDGEN special mode of no VME bus involvement from the first page of VRAM. A data pattern of 0x55555555 with the incrementing/inverting option is built. This is followed by a Write Buffer to the next page of VRAM using the same data and data options to compare the data.

This test can be executed from the host/uart also.

- Parameters

  None, only the major and minor test codes are needed.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Command example:

Name	Value in Hex
Length	0C
Major	03
Minor	00

Figure A-8. Power-up Test Command Format

0x1 - SHIO

4211 only, executes a BUS DMA as a master into its own short I/O space.

Response - PASS.

0x2 - STANDARD DMA

Builds a byte-wide incrementing pattern in the first page of VRAM and does a VME write DMA followed by a VME read DMA to the first page of SRAM and compares the data.

Parameters

- System address (physical)

  DMA count - in bytes
- Transfer options

  standard usage

Command example:

Name	Value in Hex
Length	18
Major	03
Minor	02
Sys address	100000
DMA count	200
Transfer options	023D

Figure A-9. Standard DMA Test Command Format

- Response - see CB Error Codes in *CB ERROR CODES on page 217*.

0x3 - SWAP

2211 only - tests the byte swap function for Intel compatibility, same function as the standard test.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Response - PASS.

0x4 - RAW DMA

Executes a VME read DMA followed by a VME write DMA. No data is built or compared.

Parameters (same as the Standard DMA test)

Command example:

Name	Value in Hex
Length	18
Major	03
Minor	04
Sys address	100000
DMA count	200
Transfer options	023D

Figure A-10. RAW DMA Test Command Format

Response - see CB Error Codes in *CB ERROR CODES on page 217*.

0x5 - READ BUFFER

Builds an on-board buffer according to the host supplied parameters and executes a VME write DMA.

Parameters:

- System address (physical)
- DMA count - in bytes

  If this value is set to 0x0 then the special ADDGEN mode will be used where no HBSM or VME bus involvement will take place.

  A 2k buffer will be will be transferred from the random access side of VRAM to the serial bank. All normal addgen interrupts and handshake still take place. This is useful for determining if the failure is in the CPU-ADDGEN area or the ADDGEN-HBSM-VME area.
- Transfer options - standard usage (see *Transfer Options Word on page 59*)
- Data seed - seed value for the data pattern to build
- Data options:

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Option Word	31 - 16	15 - 12	11 - 9	8-0
1	reserved	buffer offset	burst control	reserved
Option Word	**31 - 16**	**15 - 10**	**9**	**8**
0	buffer #	reserved	reset on fail	ram select

Figure A-11. Bit Structure of the READ BUFFER Data Options

- Buffer #

  This value is for the on-board buffer offset in 512 byte increments (that is, if the value was 0x8, then the on-board buffer used would be (0x8 * 0x200) + base address of the ram selected).

- Reserved

  This value should be set to 0x0.

- Reset on fail

  If this bit is set to 0x1 then the hardware will NOT be reset after a failure condition. This allows the hardware to be interrogated in the failure condition.

- Ram select

  This bit defines whether to use VRAM or SRAM for the on-board ram. A 0x0 selects VRAM and a 0x1 selects SRAM.

- Data width

  These bits define how the data is to be built or compared by the processor. This is useful for determining where data bit failures are between the CPU and buffers. Values are 0x0, 0x1 and 0x2 for byte, word and long.

- Data pattern

  These bits define how data is to be built or compared. The following table assumes that the data seed is 0x55, the data width is bytes and the length is 4 bytes.

- Reserved

  This value should be set to 0x0.

- Buffer offset

  This 4 bit field is used to determine on which of the first 16 bytes of the selected buffer to start. This allows changing the byte boundary of the internal buffer being used.

- Burst control

  If these 3 bits are cleared, no burst control is requested; otherwise, the value in the burst control field * 16 is selected (for example, burst control = 2 will cause a burst value of 32 transfers to be selected).

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

- Reserved

  This value should be set to 0x0.

Table A-16. Data Pattern Values

Value	Function	Example
0x0	Solid	0x55555555
0x1	Increment	0x55565758
0x2	Inverted	0x55aa55aa
0x3	Increment and Invert	0x55aa56a9
0xF	No manipulation	Buffer used as is

- Command example:

Name	Value in Hex
Length	20
Major	03
Minor	05
Sys address	100000
DMA count	200
Transfer options	023D
Data seed	55555555
Data option 0	80023
Data option 1	1200

Figure A-12. Read Buffer Command Format

Response - see CB Error Codes in *CB ERROR CODES on page 217*.

0x6 - WRITE BUFFER

Executes a VME read DMA to the on-board buffer location specified and compares data according to the host supplied parameters.

- Parameters

  Same as Read buffer except for the 0x0 byte count mode. In the special 0x0 byte count mode the serial bank of the VRAM is transferred to the random access side of VRAM and compared according to the supplied parameters.

- Command example: same as Read buffer except for the Minor code.
- Response - see CB Error Codes in *CB ERROR CODES on page 217*.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

0x7 - VIRQ

A VME interrupt will be generated on the level and with the vector supplied in the command.

- Parameters

31 - 11	10 - 8	7 - 0
Reserved	IRQ level	IRQ vector

- Reserved - Bits 11-31

  should be set to 0x0
- IRQ level - Bit 8-10

  legal values are 0x1 - 0x7
- IRQ vector - Bits 0-7

  bus vector supplied during IACK

Command example:

Name	Value in Hex
Length	10
Major	03
Minor	07
IRQ options	0155

Figure A-13. VIRQ Command Format

Response - see CB Error Codes in *CB ERROR CODES on page 217*.

0x8 - DUMP ASIC REGISTERS

Used only as a console command, this command will dump the register contents of the specified ASIC. The format of the dump is register # - value (that is, 10 = 1234).

Parameters

Selected ASIC

1 = Addgen
2 = Megamux
3 = MBC

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Command example:

Name	Value in Hex
Length	10
Major	03
Minor	08
ASIC select	01

Figure A-14. Dump ASIC Registers Command Format

Response - see CB Error Codes in *CB ERROR CODES on page 217*.

ERROR CODES

The following table lists the diagnostic Error Codes:

Table A-17. DIAGNOSTICS Error Codes

Value	Description
0x0	PASS
0x1	Data miscompare
0x2	VME bus time out reported by the addgen
0x3	VME buserr reported by the addgen
0x4	Addgen interrupt time-out, no done, error, or tick interrupt occurred
0x5	Premature interrupts, interrupts present before the test started
0x6	The addgen tick timer elapsed before the DMA done or error interrupts
0x7	Addgen status register lacked sanity
0x8	Addgen spurious interrupt, and interrupt occurred when not expected
0x9	VIRQ time-out, the VIRQ inport bit was never seen
0xA	VIRQ IACK time-out, the IACK inport bit was never seen
0xB	Premature VIRQ interrupt, the VIRQ inport bit was present before the test was started
0xFE	Bad command parameter supplied
0xFF	Illegal test code

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

MEMORY MAP

The following table shows a memory map of the 5211 hardware:

Table A-18. 5211 Hardware Memory Map

Value	Description
0x20000000	PRAM - program space 512K
0x36000000	SRAM - static ram 128K
0x38000000	VRAM - video ram 1M to 4M (board option)
0x9A000000	General output register (MBC)
0x9A800000	Iport register - status of on-board interrupts (1 = set)
0xB4000000	Addgen base address
0x3601FF40	DLE space for the addgen (4 * 20)
0x3601FFF0	SGE space for the DLE translation (1 * 10)

CAM TEST DIAGNOSTICS

CAM TEST - MAJOR TEST 0x7

The CAM is actually an integral part of the Motorola FDDI chipset, specifically the MAC and FSI. The 4211 CAM is comprised of discreet logic located on a plug-in companion board, and the diagnostic was written as a separate major test. In order to maintain driver and firmware compatibility, the separate test is extended to the 5211 instead of being incorporated into the Front End tests (major test 0x2). The CAM function allows frame matching on multiple MAC addresses stored in the FSI. Special handshake lines to the MAC inform it when an external match (CAM match) has been detected.

The CAM test has no mnor test code.

CAM TEST DESCRIPTION

The CAM test begins by programming all 32 address locations with distinct values. Programming is done via the FSI command ring. All values are then read back via 32 CAM COMPARE commands also issued through the FSI's command ring. If no failure is detected, an ELM_SHORT_LOOP test is run using an address programmed into the CAM as a destination address. Failure to receive the frame correctly is a failure. A second frame is sent with a source address that has been programmed into the CAM. Receiving this frame is a failure.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

TEST RESULTS

the DIAG command is moved to the CB Previous Command Word. If the test runs successfully, CBOK is written into the Command Status Word. On a failure, the DIAG_ERROR code (0x2E) is written to the CB Previous Command Status, ASCII “FAIL” is written to the Command Status Word followed by error data appropriate to the test failure. Table A-19 lists the error codes.Table A-20 lists error data for each failure.

Because the CAM test also runs the ELM_SHORT_LOOP, it is also possible to get error codes returned from that test. (See Front End Tests - Major Test 0x02).

Table A-19. CAM Test Error Codes

Code	Error	Description	Error Data
0x15	CAM_COMPARE_FAIL	CAM data read from FSI was incorrect	Yes
0x16	CAM_TIMEOUT	FSI failed to respond to CAM write commands	No
0x17	CAM_TIMEOUT2	FSI failed to respond to CAM compare commands	No

Table A-20. CAM_COMPARE_FAIL

Offset	Data	Description
0x80	0x4641494C	ASCII “FAIL”
0x84	0x14	Length in bytes of error data including length
0x88	0x7	Major Test (CAM test)
0x8C	0x0	Minor test - No minor test for CAM
0x90	0x15	Error Code
0x94	xxxxxxxx	Data written to CAM
0x98	xxxxxxxx	Data read from CAM

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

DOWNLOADING FIRMWARE B

The host system can download firmware to 5211 via the board's Common Boot interface. The downloaded firmware can then be programmed into the on-board flash memory. Most 5211 applications do not require users to be familiar with this feature to integrate the controller into the host system. Applications of the download feature include:

- making firmware upgrades in the field
- booting an interface other than the native RC Interface as documented in Chapter 3.

Once the firmware is downloaded, the user has the option of executing the downloaded firmware or programming it into the flash memory.

DOWNLOAD FORMATS

Downloadable Firmware image formats for the 5211 are:

- Binary CB image
- Motorola S-record CB image

The CB image format is shown later in this section.When using either a binary image or a Motorola S-record image, the firmware is downloaded to the 5211 where it can be executed or burned into the on-board flash memory.

Either the binary image or the Motorola S-record image may be downloaded to the 5211. However, the download method selected will determine which format you should use. In general, if you have a 4211 driver that uses **POKE** and **BOOT** to download and execute, use the binary image to allow backward compatibility.

If you are developing new software, or want to update the 4211 driver use the **DNLD** and the **BURN** commands, which provide better control over data integrity, as well as allowing the user to permanently store the downloaded program into the flash memory, are described in chapter 4. The **DNLD** command transfers Motorola S-record format files to the board. The **BURN** command stores the downloaded program into the flash memory.

DOWNLOAD AND EXECUTE A CB-FORMAT BINARY IMAGE

The sequence for downloading and executing code is as follows:

1. The controller starts up in Common Boot mode each time it is powered up or reset. In CB mode the Command Response Word should contain the ASCII value "CBOK."

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

2. The host executes a series of POKE commands to download firmware into controller memory. Figure B-1 shows how the fields of this Common Boot command are filled in.

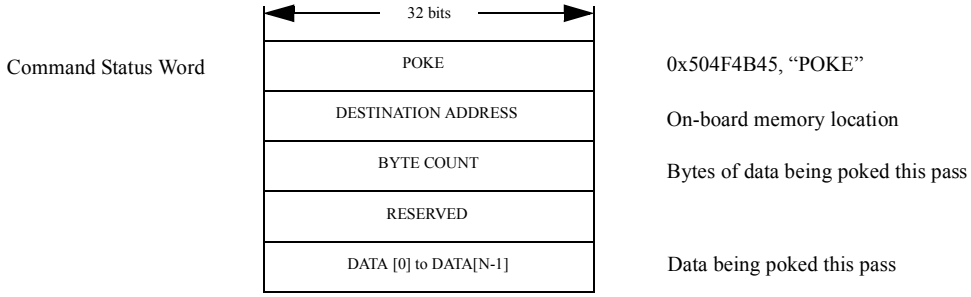


Figure B-1. Using POKE Command to Download Code

The fields are defined as follows:

- COMMAND STATUS WORD

  This field contains the ASCII value POKE (0x504F4b45). This field is filled in after the other command fields (described below) have been completed.

- DESTINATION ADDRESS

  On the first pass, this field contains address of the controller buffer memory location where the downloaded program is assembled. Address 0x38000000, the start of internal buffer memory address, should be used. As successive batches of data are poked on-board, this address is incremented to a point past previously downloaded data.

- BYTE COUNT

  This is the number of bytes being poked this pass. The maximum byte count is determined by the available space in short I/O.

- RESERVED

  This field is ignored by the controller.

- DATA

  This is the "BYTE COUNT" number of data bytes being poked into memory this pass.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3. The host executes a BOOT command to move the downloaded program to the execution RAM and transfer control to it. The following diagram shows how the fields of the Common Boot command are filled in:

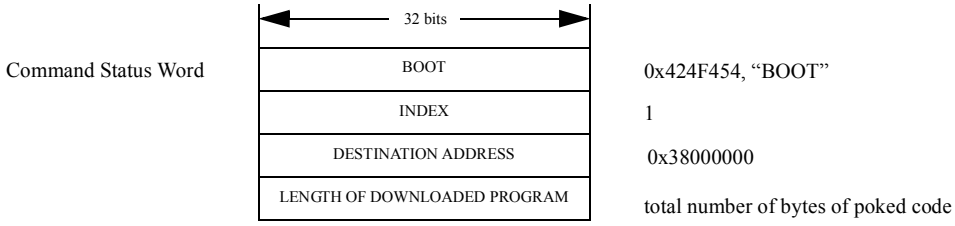


Figure B-2. Using BOOT command to run downloaded program

The fields are defined below:

- COMMAND STATUS WORD

  This field contains the ASCII value BOOT (0x424F454). This field is filled in after the other command fields (described below) have been completed.

- INDEX

  A boot index of 1 has a special meaning which indicates that the downloaded program should be moved to the execution RAM and the control should be transferred to it.

- DESTINATION ADDRESS

  This filed contains the start address of the buffer memory where the program is downloaded. A value of 0x38000000 should be used.

- LENGTH OF DOWNLOADED PROGRAM

  This is the total length in bytes of the downloaded program.

STORE A DOWNLOADED PROGRAM IN FLASH MEMORY

After a program is downloaded into the controller it can be permanently stored in the flash memory using the **BURN** command as shown below.

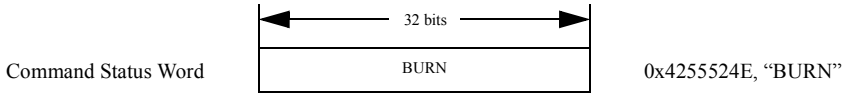


Figure B-3. Using BURN to store a downloaded program in flash memory

- COMMAND STATUS WORD

  This filed contains the ASCII value BURN (0x4255524E).

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

DOWNLOAD CB IMAGE FORMAT

The CB download image format is shown in Table B-1. The image can be in binary format or Motorola S-record format. Note: If the image is in the form of Motorola S-records, refer to Table B-2 for additional information. The image offset and byte counts in Table B-1 do not reflect S-record addresses and checksums.

The data within the image consists of two sections. The first section is a 128 byte image header that allows the firmware to make sanity checks and verify that the download data is of the proper format. The second section is the program text and data to be downloaded.

Table B-1. CB Image Download Format

Section	Image Offset	Description	Size (Bytes)
Header	0	Magic Number	4
	4	Board ID Code	4
	8	Processor Type	4
	12	ASCII Identifier	16
	28	Image Checksum	4
	32	Image Size	4
	36	Text Start	4
	40	Text End	4
	44	Data Start	4
	48	BSS Start	4
	52	BSS End	4
	56	Entry	4
	60	Reserved	4
	68	Reserved	4
	72	Stack Pointer	4
	76	Reserved	4
	80	Reserved	4
	84-123	Reserved	40
	124	Image Header Checksum	4
Image	128 : N+128	Image Starts : Image Ends	N

- MAGIC NUMBER

  This value allows the controller to quickly determine if the download header is valid. This value is currently 0x12345678.

- BOARD ID CODE

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The value is unique for every controller. It allows the controller to verify that the image is for proper controller.

- PROCESSOR TYPE

  This value contains the processor type for the download image. Values might be 0x29000, 0x68040, 0x68020, and so on. This enables the controller to sanity check that the download image is proper.

- ASCII IDENTIFIER

  This field consists of a NULL terminated string of bytes that is not used by the controller. It is included in the header to allow the host to be able to identify various download images.

- IMAGE CHECKSUM

  This value is a longword one's complement of a running checksum of the download or executable image. This checksum does not include the header in its calculation.

- IMAGE SIZE

  This value contains the size in bytes of the image section of the download data.

- TEXT START

  Start Address of the executable image's test.

- TEXT END

  Ending Address of the test of the executable image.

- DATA START

  Starting Address of the initialized data of the executable program.

- DATA END

  End Address of the initialized data.

- BSS START

  Starting Address of the program's uninitialized data.

- BSS END

  Ending Address of the uninitialized data.

- ENTRY

  Entry Address of executable image.

- RESERVED

  This area is reserved for future expansion of the header.

- RESERVED

  This area is reserved for future expansion of the header.

- STACK POINTER

  Address of the stack pointer is to be set at the time of entry to the executable image.

- RESERVED

  This area is reserved for future expansion of the header.

- RESERVED

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

This area is reserved for future expansion of the header.

- RESERVED

  This area is reserved for future expansion of the header.

- IMAGE HEADER CHECKSUM

  This field is a check for the download image header. This allows the controller to be sure that the header is valid, consist of a longword one's complement of a running of the header. The checksum is calculated by adding 8 bit quantities of the image header to produce a 32 bit sum. The 32 bit sum is then one's complemented.

- IMAGE STARTS

  Start of the program text and data.

- IMAGE ENDS

  Ends of the program text and data.

MOTOROLA S-RECORDS

If the download image is in the form of Motorola S-records, the following table shows the format of each S-record field in the image file.

Table B-2. Motorola S-Record Fields

Fields	No. of Characters	Contents
Type	2	S-record type (S0, S1, S2, S3, S7, S8, S9)
Record Length	2	The count of the character pairs in the record, excluding the type and record length.
Address	4, 6, or 8	The 2-, 3-, or 4-byte address at or which the data field is to be loaded into memory. S0 uses 2-, S1 uses 2-, S2 uses 3-, S3 uses 4-, S7 uses 4-, S8 uses 3-, and S9 uses 2-byte address.
Code/Data	0-2n	From 0 to n bytes of executable code, memory-loaded data, or descriptive information.
Checksum	2	The least significant byte of the one's complement of the sum of the values represented byte the pairs of characters up the record length, address, and the code/data fields.

All addresses for the header and the image start at location 0 and are sequential. Each download section must be preceded by a S0 record type. The exact S0 records distinguishing the header and image sections are:

S00600004844521B For the header (Address 0 and data "HDR")

S0060000494D471C For the image (Address 0 and data "IMG")

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The remaining portions of each section would consists of S1, S2, or S3 records, and are terminated with a S7, S8 or S9 record. The address field in the termination record should contain the start address for the image. This address matches the address given in the header. The exact S9 record with a 0 start address is:

S9030000FC

FLASH EPROM LAYOUT

The following table is a layout of FLASH EPROM. This layout consists of two sections. One section that holds the base level code known as the primary image and a second section that holds an alternate image.

Table 4-3. Layout of FLASH EPROM

Primary Image	Common Boot and Diagnostics
Secondary (Exec0) Image	Download Image Header
	Image

The primary image is CB. This is the code that starts on power up or reset. It is responsible for all the power-up diagnostics as well as maintaining the location of the secondary image.

SECONDARY IMAGE UPDATES

The secondary image can be updated by simply using the **POKE/DNLD** and **BURN** commands. CB also offers a command to return the header information of the secondary image stored in the flash EPROM. For more information refer to the **INFO** command in chapter 4.

CB ERROR STATUS CODES I

The following table lists the CB error codes.

Table B-4. CB Error Status Codes I

Codes	Descriptions
01	Bad S-record Type
02	Image Checksum Error
03	S-record Checksum Error
04	Download Buffer Overrun
05	Image Overrun
06	Bad S0 S-record
07	Image Header Checksum Error

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table B-4. CB Error Status Codes I (cont)

Codes	Descriptions
08	Bad Length
09	Not a valid S-record
0A	Bad Compare
0B	Bad Size -- Bad UNIT Size
0C	Bad INFO Type
0D	Boot 0 Attempted
0E	Invalid BOOT Type
0F	Major Test Unknown
10	Minor Test Unknown
11	RAM Type Unknown
12	Memory Test Failure
13	Unknown Option
14	Illegal Return
15	Minimum Execution RAM Not Found

CB ERROR STATUS CODES II

The following table shows CB Error Codes generated during the DNLD, EXEC and BURN commands.

Table B-5. CB Error Status Codes II

Codes	Descriptions
16	Bad MAGIC NUMBER in Image Header
17	Error in Image Checksum
18	Image Header Length Error
19	Overlap in Text, Data or BSS Addresses
1A	Text Address Bad
1B	Data Address BAd
1C	BSS Address Bad
1D	Stack Address Bad
1E	Entry Address Error
1F	No Text Section
20	Regions Overlap

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table B-5. CB Error Status Codes II (cont)

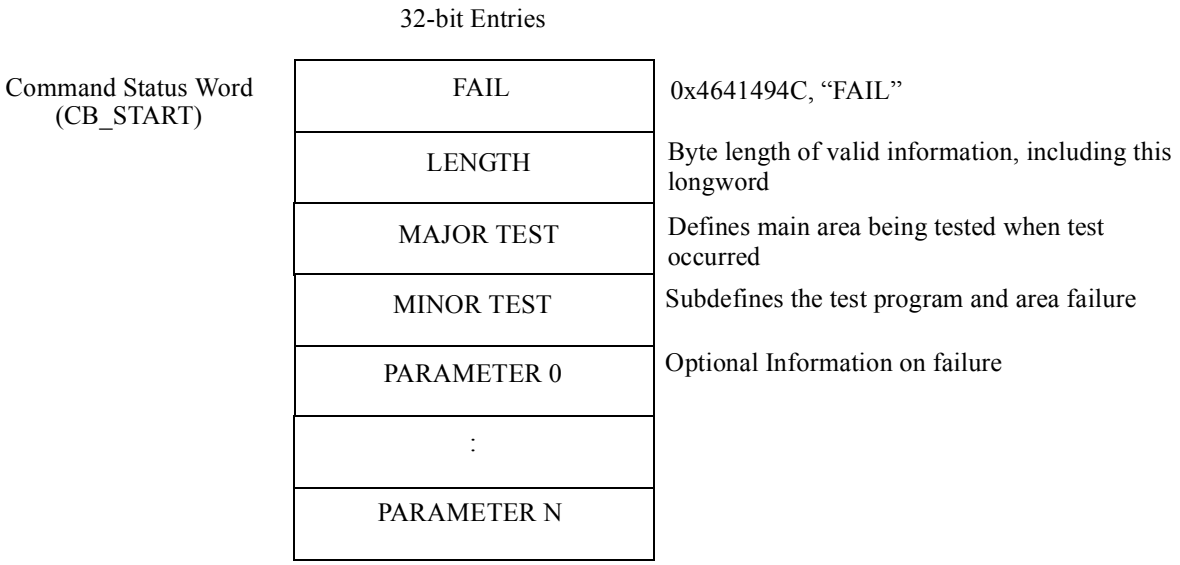
Codes	Descriptions
21	Regions Lengths do not Sum
22	Image Checksum is Bad
23	Sequence Error
24	Header Overrun
25	Burn Algorithm Field
26	Burn did not Verify
27	Burn Image is too Large
28	Invalid Offset for Burn
29	No EXEC 0 Image
2A	Invalid Type on INFO Command
2B	No BOOT 0 Image Available
2C	Execution Type Unrecognized
2D	Bad Copy of Execute Image
2E	DIAG Command Failed

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

CB FAIL STATUS BLOCK FORMAT

The following is the format for a CB FAIL Status Block for the power-up/reset test sequence.

FAIL STATUS BLOCK



This status block allows the host to examine the cause of the power-up/reset. NOTE: The TEST command uses the same error codes as the MEMORY TEST section.

MEMORY TEST POWER-UP CODES

Memory Test Numbers are shown in the following table:

Table B-6. Memory Test Power-Up Codes

MEMORY MAJOR TEST NUMBER	
MEMORY_TEST	0x00000001
MEMORY MINOR TEST NUMBER	
STACK_RAM	0x00000001
EXECUTION_RAM	0x00000002
BUFFER_RAM	0x00000003
SHORT_IO_RAM	0x00000004
EVENT_RAM	0x00000005

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table B-6. Memory Test Power-Up Codes (cont)

TYPES OF MEMORY TESTS (goes into error code field)	
EXIST_TEST	0x00000001
ADDRESS_FAULTS_TEST	0x00000002
MARCH_TEST	0x00000003
SLIDER_TEST	0x00000004
BLOCK_TEST	0x00000005
ADDR_PTR_TEST	0x00000006
SIZING_TEST	0x00000007

CPU TEST POWER-UP CODES

CPU Test Numbers are shown in the following table:

Table B-7. CPU Test Power-Up Codes

CPU MAJOR TEST NUMBERS	
CPU_TEST	0x00000002
CPU MINOR TEST NUMBERS	
REG_TEST	0x000000001

EPROM TEST POWER-UP CODES

EPROM Test Numbers are shown in the following table:

Table 4-8.

EPROM MAJOR TEST NUMBERS	
EPROM_TEST	0x00000003
EPROM MINOR TEST NUMBERS	
CHECK_SUM	(0x01)

HARDWARE CRITICAL POWER-UP CODES

Hardware Critical Test Numbers are shown in the following table:

Table B-9. Hardware Critical Power-up Codes

HARDWARE CRITICAL MAJOR TEST NUMBERS	
HW_CRITICAL	0x00000004

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table B-9. Hardware Critical Power-up Codes (cont)

HARDWARE CRITICAL MINOR TEST NUMBERS	
SHADOW	0x00000001

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

ILLEGAL INTERRUPT POWER-UP CODES

Table B-10. Illegal Interrupt Power-Up Codes

ILLEGAL INTERRUPT MAJOR TEST NUMBER	
ILLEGAL_INTERRUPT	0x00000005
ILLEGAL INTERRUPT MINOR TEST NUMBERS	
STDBUS_ERROR	2 BUS ERROR
STD_ADDRESS_ERROR	3 ADDRESS ERROR
STD_ILLEGAL_INSTRUCTIONS	4 ILLEGAL INSTRUCTION
STD_ZERO_DIVIDE	6 CHK AND CHK2 INSTRUCTIONS
STD_CHK_CHK2	7 TRAPV INSTRUCTION
STD_PRIVILEGE	8 PRIVILEGE VIOLATION
STD_TRACE	9 TRACE
STD_EMUL_A	10 EMULATE A
STD_EMUL_B	11 EMULATE B
STD_HDW_BREAKPOINT	12 HARDWARE BREAKPOINT
STD_COPROCESSOR_VIOLATE	13 COPROCESSOR VIOLATION
STD_FORMAT_ERROR	14 FORMAT ERROR
STD_UNINITIALIZED_INT	15 UNINITIALIZED INTERRUPT
STD_RESERVED	16 RESERVED INTERRUPT 16-23, 59-63
STD_SPURIOUS	24 SPURIOUS INTERRUPT
STD_LEVEL_1_AUTO	25 LEVEL 1 AUTOVECTOR
STD_LEVEL_2_AUTO	26 LEVEL 2 AUTOVECTOR
STD_LEVEL_3_AUTO	27 LEVEL 3 AUTOVECTOR
STD_LEVEL_4_AUTO	28 LEVEL 4 AUTOVECTOR
STD_LEVEL_5_AUTO	29 LEVEL 5 AUTOVECTOR
STD_LEVEL_6_AUTO	30 LEVEL 6 AUTOVECTOR
STD_LEVEL_7_AUTO	31 LEVEL 7 AUTOVECTOR
STD_TRAP_0	32 TRAP 0
STD_TRAP_1	33 TRAP 1
STD_TRAP_2	34 TRAP 2
STD_TRAP_3	35 TRAP 3
STD_TRAP_4	36 TRAP 4
STD_TRAP_5	37 TRAP 5
STD_TRAP_6	38 TRAP 6

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table B-10. Illegal Interrupt Power-Up Codes (cont)

ILLEGAL INTERRUPT MINOR TEST NUMBERS	
STD_TRAP_7	39 TRAP 7
STD_TRAP_8	40 TRAP 8
STD_TRAP_9	41 TRAP 9
STD_TRAP_10	42 TRAP 10
STD_TRAP_11	43 TRAP 11
STD_TRAP_12	44 TRAP 12
STD_TRAP_13	45 TRAP 13
STD_TRAP_14	46 TRAP 14
STD_TRAP_15	47 TRAP 15
STD_COPR_RESERVED	48 COPROCESSOR RESERVED 48-58
/* See STD_RESERVED for 59-63 */	
STD_USER	64 USER INTERRUPTS 64-255

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

CB ERROR STATUS CODES I

The following lists the CB error codes.

Table C-1. CB Error Status Codes I

Code	Description
01	Bad S-record Type
02	Image Checksum Error
03	S-record Checksum Error
04	Download Buffer Overrun
05	Image Overrun
06	Bad S0 S-record
07	Image Header Checksum Error
08	Bad Length
09	Not a valid S-record
0A	Bad Compare
0B	Bad Size -- Bad UNIT Size
0C	Bad INFO Type
0D	Boot 0 Attempted
0E	Invalid BOOT Type
0F	Major Test Unknown
10	Minor Test Unknown
11	RAM Type Unknown
12	Memory Test Failure
13	Unknown Option
14	Illegal Return
15	Minimum Execution RAM Not Found

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

CB ERROR STATUS CODES II

The following is a table of CB Error Codes generated during the DNLD, EXEC, and BURN commands.

Table C-2. CB Error Status Codes II

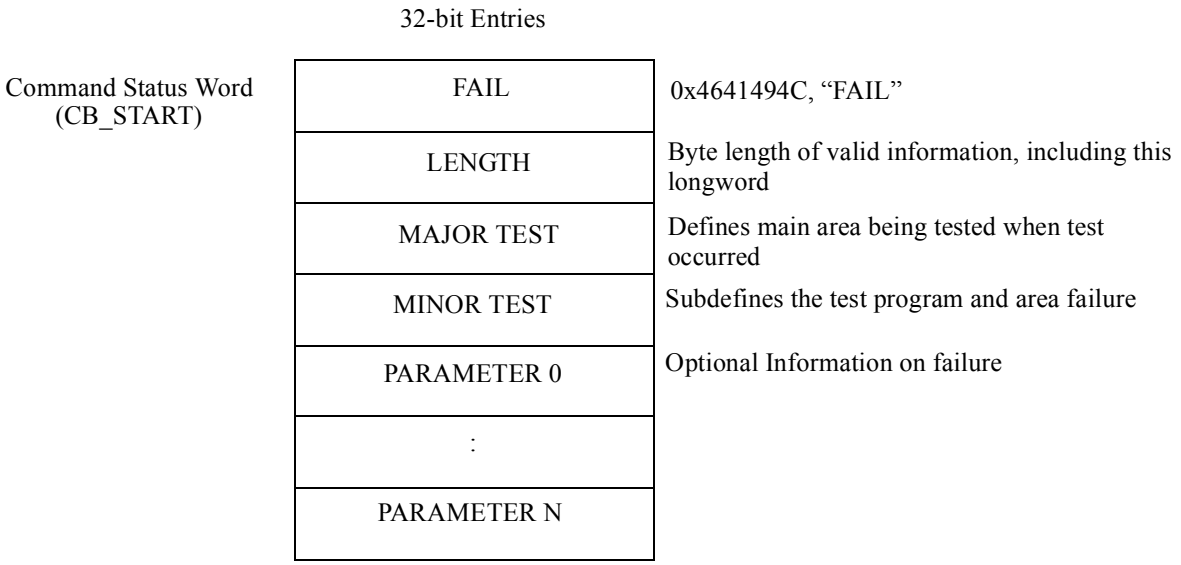
Code	Description
16	Bad MAGIC NUMBER in Image Header
17	Error in Image Checksum
18	Image Header Length Error
19	Overlap in Text, Data or BSS Addresses
1A	Text Address Bad
1B	Data Address BAd
1C	BSS Address Bad
1D	Stack Address Bad
1E	Entry Address Error
1F	No Text Section
20	Regions Overlap
21	Regions Lengths do not Sum
22	Image Checksum is Bad
23	Sequence Error
24	Header Overrun
25	Burn Algorithm Field
26	Burn did not Verify
27	Burn Image is too Large
28	Invalid Offset for Burn
29	No EXEC 0 Image
2A	Invalid Type on INFO Command
2B	No BOOT 0 Image Available
2C	Execution Type Unrecognized
2D	Bad Copy of Execute Image
2E	DIAG Command Failed

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

CB FAIL STATUS BLOCK FORMAT

The following is the format for a CB FAIL Status Block for the power-up/reset test sequence.

FAIL STATUS BLOCK



This status block allows the host to examine the cause of the power-up/reset. NOTE: The TEST command uses the same error codes as the MEMORY TEST section described below.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

MEMORY TEST POWER-UP CODES

Memory Test Numbers are shown in the table below:

Table C-3. Memory Test Power-Up Code

MEMORY MAJOR TEST NUMBER	
MEMORY_TEST	0x00000001
MEMORY MINOR TEST NUMBER	
PROGRAM_RAM	0x00000001
STATIC_RAM	0x00000002
BUFFER_RAM	0x00000003
TYPES OF MEMORY TESTS (goes into error code field)	
EXIST_TEST	0x00000001
ADDRESS_FAULTS_TEST	0x00000002
MARCH_TEST	0x00000003
SLIDER_TEST	0x00000004
BLOCK_TEST	0x00000005
ADDR_PTR_TEST	0x00000006
SIZING_TEST	0x00000007

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

CPU TEST POWER-UP CODES

CPU Test Numbers are shown in the table below:

Table C-4. CPU Test Power-Up Codes

CPU MAJOR TEST NUMBERS	
CPU_TEST	0x00000002
CPU MINOR TEST NUMBERS	
REG_TEST	0x000000001

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

EPROM TEST POWER-UP CODES

EPROM Test Numbers are shown in the following table:

Table C-5. EPROM Test Power-Up Codes

EPROM MAJOR TEST NUMBERS	
EPROM_TEST	0x00000003
EPROM MINOR TEST NUMBERS	
CHECK_SUM	(0x01)

HARDWARE CRITICAL POWER-UP CODES

Hardware Critical Test Numbers are shown in the following table:

Table C-6. Hardware Critical Power-Up Codes

HARDWARE CRITICAL MAJOR TEST NUMBERS	
HW_CRITICAL	0x00000004
HARDWARE CRITICAL MINOR TEST NUMBERS	
SHADOW	0x00000001

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

ILLEGAL INTERRUPT POWER-UP CODES

Table C-7. Ilegal Interrupt Power-up Codes

ILLEGAL INTERRUPT MAJOR TEST NUMBER	
ILLEGAL_INTERRUPT	0x00000005
ILLEGAL INTERRUPT MINOR TEST NUMBERS	
STDBUS_ERROR	2 BUS ERROR
STD_ADDRESS_ERROR	3 ADDRESS ERROR
STD_ILLEGAL_INSTRUCTIONS	4 ILLEGAL INSTRUCTION
STD_ZERO_DIVIDE	6 CHK AND CHK2 INSTRUCTIONS
STD_CHK_CHK2	7 TRAPV INSTRUCTION
STD_PRIVILEGE	8 PRIVILEGE VIOLATION
STD_TRACE	9 TRACE
STD_EMUL_A	10 EMULATE A
STD_EMUL_B	11 EMULATE B
STD_HDW_BREAKPOINT	12 HARDWARE BREAKPOINT
STD_COPROCESSOR_VIOLATE	13 COPROCESSOR VIOLATION
STD_FORMAT_ERROR	14 FORMAT ERROR
STD_UNINITIALIZED_INT	15 UNINITIALIZED INTERRUPT
STD_RESERVED	16 RESERVED INTERRUPT 16-23, 59-63
STD_SPURIOUS	24 SPURIOUS INTERRUPT
STD_LEVEL_1_AUTO	25 LEVEL 1 AUTOVECTOR
STD_LEVEL_2_AUTO	26 LEVEL 2 AUTOVECTOR
STD_LEVEL_3_AUTO	27 LEVEL 3 AUTOVECTOR
STD_LEVEL_4_AUTO	28 LEVEL 4 AUTOVECTOR
STD_LEVEL_5_AUTO	29 LEVEL 5 AUTOVECTOR
STD_LEVEL_6_AUTO	30 LEVEL 6 AUTOVECTOR
STD_LEVEL_7_AUTO	31 LEVEL 7 AUTOVECTOR
STD_TRAP_0	32 TRAP 0
STD_TRAP_1	33 TRAP 1
STD_TRAP_2	34 TRAP 2
STD_TRAP_3	35 TRAP 3
STD_TRAP_4	36 TRAP 4
STD_TRAP_5	37 TRAP 5
STD_TRAP_6	38 TRAP 6

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table C-7. Ilegal Interrupt Power-up Codes (cont)

ILLEGAL INTERRUPT MINOR TEST NUMBERS	
STD_TRAP_7	39 TRAP 7
STD_TRAP_8	40 TRAP 8
STD_TRAP_9	41 TRAP 9
STD_TRAP_10	42 TRAP 10
STD_TRAP_11	43 TRAP 11
STD_TRAP_12	44 TRAP 12
STD_TRAP_13	45 TRAP 13
STD_TRAP_14	46 TRAP 14
STD_TRAP_15	47 TRAP 15
STD_COPR_RESERVED	48 COPROCESSOR RESERVED 48-58
/* See STD_RESERVED for 59-63 */	
STD_USER	64 USER INTERRUPTS 64-255

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

JUMPER SETTINGS D

Base Address Jumper Settings

Jumper fields 8 through 14 are used to set the base address of the 5211's short I/0 space. The following table shows the jumper settings for all possible base addresses. To locate the jumper fields, refer to the 5211 board diagram in Figure 2-1 on page 14.

Table D-1. Base Address Jumper Settings

Address	JA24	JA25	JA26	JA27	JA28	JA29	JA30
0x0000	In	In	In	In	In	In	In
0x0200	In	In	In	In	In	In	Out
0x0400	In	In	In	In	In	Out	In
0x0600	In	In	In	In	In	Out	Out
0x0800	In	In	In	In	Out	In	In
0x0A00	In	In	In	In	Out	In	Out
0x0C00	In	In	In	In	Out	Out	In
0x0E00	In	In	In	In	Out	Out	Out
0x1000	In	In	In	Out	In	In	In
0x1200	In	In	In	Out	In	In	Out
0x1400	In	In	In	Out	In	Out	In
0x1600	In	In	In	Out	In	Out	Out
0x1800	In	In	In	Out	Out	In	In
0x1A00	In	In	In	Out	Out	In	Out
0x1C00	In	In	In	Out	Out	Out	In
0x1E00	In	In	In	Out	Out	Out	Out
0x2000	In	In	Out	In	In	In	In
0x2200	In	In	Out	In	In	In	Out
0x2400	In	In	Out	In	In	Out	In
0x2600	In	In	Out	In	In	Out	Out
0x2800	In	In	Out	In	Out	In	In
0x2A00	In	In	Out	In	Out	In	Out
0x2C00	In	In	Out	In	Out	Out	In
0x2E00	In	In	Out	In	Out	Out	Out
0x3000	In	In	Out	Out	In	In	In

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table D-1. Base Address Jumper Settings (cont)

Address	JA24	JA25	JA26	JA27	JA28	JA29	JA30
0x3200	In	In	Out	Out	In	In	Out
0x3400	In	In	Out	Out	In	Out	In
0x3600	In	In	Out	Out	In	Out	Out
0x3800	In	In	Out	Out	Out	In	In
0x3A00	In	In	Out	Out	Out	In	Out
0x3C00	In	In	Out	Out	Out	Out	In
0x3E00	In	In	Out	Out	Out	Out	Out
0x4000	In	Out	In	In	In	In	In
0x4200	In	Out	In	In	In	In	Out
0x4400	In	Out	In	In	In	Out	In
0x4600	In	Out	In	In	In	Out	Out
0x4800	In	Out	In	In	Out	In	In
0x4A00	In	Out	In	In	Out	In	Out
0x4C00	In	Out	In	In	Out	Out	In
0x4E00	In	Out	In	In	Out	Out	Out
0x5000	In	Out	In	Out	In	In	In
0x5200	In	Out	In	Out	In	In	Out
0x5400	In	Out	In	Out	In	Out	In
0x5600	In	Out	In	Out	In	Out	Out
0x5800	In	Out	In	Out	Out	In	In
0x5A00	In	Out	In	Out	Out	In	Out
0x5C00	In	Out	In	Out	Out	Out	In
0x5E00	In	Out	In	Out	Out	Out	Out
0x6000	In	Out	Out	In	In	In	In
0x6200	In	Out	Out	In	In	In	Out
0x6400	In	Out	Out	In	In	Out	In
0x6600	In	Out	Out	In	In	Out	Out
0x6800	In	Out	Out	In	Out	In	In
0x6A00	In	Out	Out	In	Out	In	Out
0x6C00	In	Out	Out	In	Out	Out	In
0x6E00	In	Out	Out	In	Out	Out	Out
0x7000	In	Out	Out	Out	In	In	In
0x7200	In	Out	Out	Out	In	In	Out

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table D-1. Base Address Jumper Settings (cont)

Address	JA24	JA25	JA26	JA27	JA28	JA29	JA30
0x7400	In	Out	Out	Out	In	Out	In
0x7600	In	Out	Out	Out	In	Out	Out
0x7800	In	Out	Out	Out	Out	In	In
0x7A00	In	Out	Out	Out	Out	In	Out
0x7C00	In	Out	Out	Out	Out	Out	In
0x7E00	In	Out	Out	Out	Out	Out	Out
0x8000	Out	In	In	In	In	In	In
0x8200	Out	In	In	In	In	In	Out
0x8400	Out	In	In	In	In	Out	In
0x8600	Out	In	In	In	In	Out	Out
0x8800	Out	In	In	In	Out	In	In
0x8A00	Out	In	In	In	Out	In	Out
0x8C00	Out	In	In	In	Out	Out	In
0x8E00	Out	In	In	In	Out	Out	Out
0x9000	Out	In	In	Out	In	In	In
0x9200	Out	In	In	Out	In	In	Out
0x9400	Out	In	In	Out	In	Out	In
0x9600	Out	In	In	Out	In	Out	Out
0x9800	Out	In	In	Out	Out	In	In
0x9A00	Out	In	In	Out	Out	In	Out
0x9C00	Out	In	In	Out	Out	Out	In
0x9E00	Out	In	In	Out	Out	Out	Out
0xA000	Out	In	Out	In	In	In	In
0xA200	Out	In	Out	In	In	In	Out
0xA400	Out	In	Out	In	In	Out	In
0xA600	Out	In	Out	In	In	Out	Out
0xA800	Out	In	Out	In	Out	In	In
0xAA00	Out	In	Out	In	Out	In	Out
0xAC00	Out	In	Out	In	Out	Out	In
0xAE00	Out	In	Out	In	Out	Out	Out
0xB000	Out	In	Out	Out	In	In	In
0xB200	Out	In	Out	Out	In	In	Out
0xB400	Out	In	Out	Out	In	Out	In

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table D-1. Base Address Jumper Settings (cont)

Address	JA24	JA25	JA26	JA27	JA28	JA29	JA30
0xB600	Out	In	Out	Out	In	Out	Out
0xB800	Out	In	Out	Out	Out	In	In
0xBA00	Out	In	Out	Out	Out	In	Out
0xBC00	Out	In	Out	Out	Out	Out	In
0xBE00	Out	In	Out	Out	Out	Out	Out
0xC000	Out	Out	In	In	In	In	In
0xC200	Out	Out	In	In	In	In	Out
0xC400	Out	Out	In	In	In	Out	In
0xC600	Out	Out	In	In	In	Out	Out
0xC800	Out	Out	In	In	Out	In	In
0xCA00	Out	Out	In	In	Out	In	Out
0xCC00	Out	Out	In	In	Out	Out	In
0xCE00	Out	Out	In	In	Out	Out	Out
0xD000	Out	Out	In	Out	In	In	In
0xD200	Out	Out	In	Out	In	In	Out
0xD400	Out	Out	In	Out	In	Out	In
0xD600	Out	Out	In	Out	In	Out	Out
0xD800	Out	Out	In	Out	Out	In	In
0xDA00	Out	Out	In	Out	Out	In	Out
0xDC00	Out	Out	In	Out	Out	Out	In
0xDE00	Out	Out	In	Out	Out	Out	Out
0xE000	Out	Out	Out	In	In	In	In
0xE200	Out	Out	Out	In	In	In	Out
0xE400	Out	Out	Out	In	In	Out	In
0xE600	Out	Out	Out	In	In	Out	Out
0xE800	Out	Out	Out	In	Out	In	In
0xEA00	Out	Out	Out	In	Out	In	Out
0xEC00	Out	Out	Out	In	Out	Out	In
0xEE00	Out	Out	Out	In	Out	Out	Out
0xF000	Out	Out	Out	Out	In	In	In
0xF200	Out	Out	Out	Out	In	In	Out
0xF400	Out	Out	Out	Out	In	Out	In
0xF600	Out	Out	Out	Out	In	Out	Out

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table D-1. Base Address Jumper Settings (cont)

Address	JA24	JA25	JA26	JA27	JA28	JA29	JA30
0xF800	Out	Out	Out	Out	Out	In	In
0xFA00	Out	Out	Out	Out	Out	In	Out
0xFC00	Out	Out	Out	Out	Out	Out	In
0xFE00	Out	Out	Out	Out	Out	Out	Out

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Regulatory Statements

E

Interphase Fiber Products' Compliance

This Interphase fiber product uses an LED and complies with IEC825-1, IEC825-2, and FDA 21 CFR 1040.10 and 1040.11.

EN60950–IEC950 Safety Standard

This equipment complies with the EN60950–IEC950 safety standard.

Canadian

Tested to Comply with Canadian Standards

This Class A digital apparatus complies with Canadian ICES-003.

Cet appareil numérique de la classe A est conforme à la norme NMB-003 du Canada.

Regulatory Information for Europe

This equipment displays the CE mark to show that it has been found to be in full compliance with the requirements of the EMC and Low Voltage Directives (89/336/EEC and 72/23/EEC, as amended by Directive 93/68/EEC).

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Limited Warranty Policy

Interphase Corporation (“Seller”) warrants that (i) the hardware provided to Buyer (“Products”) shall, at the F.O.B. point, be free from defects in materials and workmanship for a period of one (1) year from the date of shipment to Buyer; (ii) the software and/or firmware associated with or embedded in the Products shall comply with the applicable specifications for a period of six (6) months from the date of shipment to Buyer; and (iii) its services will, when performed, be of good quality. Defective and nonconforming Products and software must be held for Seller’s inspection and returned at Seller’s request, freight prepaid, to the original F.O.B. point.

Upon Buyer’s submission of a claim in accordance with Seller’s Return and Repair Policy, Seller will, at its option either (i) repair or replace the nonconforming Product; (ii) correct or replace the software/firmware; (iii) rework the nonconforming services; or (iv) refund an equitable portion of the purchase price attributable to such nonconforming Products, software, or services. Seller shall not be liable for the cost of removal or installation of products or any unauthorized warranty work, nor shall Seller be responsible for any transportation costs, unless expressly authorized in writing by Seller. This warranty does not cover damage to the Product resulting from accident, disaster, misuse, negligence, improper maintenance, or modification or repair of the Product other than by Seller. Any Products or software replaced by Seller will become the property of Seller.

REMEDIES AND EXCLUSIONS. THE SOLE LIABILITY OF SELLER AND BUYER’S SOLE REMEDY FOR BREACH OF THESE WARRANTIES SHALL BE LIMITED TO REPAIR OR REPLACEMENT OF THE PRODUCTS OR CORRECTION OF THAT PART OF THE SOFTWARE, WHICH FAILS TO CONFORM TO THESE WARRANTIES. EXCEPT AS EXPRESSLY STATED HEREIN, AND EXCEPT AS TO TITLE, THERE ARE NO OTHER WARRANTIES, EXPRESS OR IMPLIED, INCLUDING WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE, IN CONNECTION WITH OR ARISING OUT OF ANY PRODUCT OR SOFTWARE PROVIDED TO BUYER.

IN NO EVENT SHALL SELLER HAVE ANY LIABILITY FOR INDIRECT, INCIDENTAL, SPECIAL OR CONSEQUENTIAL DAMAGES, HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, ARISING OUT OF THESE WARRANTIES, INCLUDING BUT NOT LIMITED TO LOSS OF ANTICIPATED PROFITS, LOSS OF DATA, USE OR GOODWILL, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

IC-199, 1/97

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

THIS MANUAL IS PROVIDED “AS IS.” INTERPHASE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING THOSE OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL INTERPHASE BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Index

When using this index, keep in mind that a page number indicates only where referenced material begins. It may extend to the page or pages following the page referenced.

Numerics

68EC040 processor 7, 175
6U 2
9U 2

A

address
- destination address 128, 131, 150
- FDDI format 128, 131
- in CAM 148, 150, 151
- in Command Tag 122, 125, 128, 131
- in command tag 129, 136, 142, 148, 150, 153, 157, 159, 160, 163, 164, 166, 168, 171, 174
- modifiers 15
- of buffer list 135
- of receive buffer 134
- Report Station Address 164
- Request Station Address 136
- Set CAM Address 148
- Set Station Address 137
- short I/O 16
- slave address modifiers 15
- source address 128, 131
- VMEbus address modifier 60
- VMEbus master options 175
- VRAM address 15

address modifier 15, 60
address modifiers 175
ANSI X3.139 - 1987 FDDI Token Ring Media Access Control (MAC) 2
ANSI X3.166 - 1990 Token Ring FDDI Physical Layer Medium Dependent (PMD) 2
ANSI X3T9.5/84-49 FDDI Station Management (SMT) 2
ANSI X3T9.5/92 - TP-PMD/220 - Rev 0.3, Feb. 17, 1993 2
asynchronous 7, 62, 122, 125, 129, 132, 178
- specifying for Tx frame 122, 125, 129, 132

attribute ID 170
attribute index 143, 145

B

bandwidth 8, 135
big-endian 128, 131
binary representation 9
board-to-host
- channel 39
- channel ID 59
- initial channel 59, 122, 125, 128, 131, 136, 142, 148, 150, 159, 162, 165, 170

board-to-host channel 53
boot-up 31
buffer list 63
- command tag 134
- creating 134
- delimiting end of 134
- format 134
- length of 135
- physical address 135

buffer list elements 63, 64
buffer size 65

C

cable connectors
- keying 20

cabling 18, 21
- dual attachment stations 19, 23
- single attachment stations 22

CAM (Content-Addressable Memory) 7, 119, 148
- adding address 148
- CAM Address Response 150, 159, 166
- Clear CAM Address 150
- Flush CAM Address 151
- Get CAM Address 159
- removing station address 150
- Report CAM Address 174
- Set CAM Address 148

CAM address 159
CAM Address Response 148, 150, 166
canonical 128, 131, 164
channel
- board-to-host 39, 53, 59
- descriptors 39, 59
- DMA 123, 129
- host-to-board 39, 59
- ID of response channel 122, 124, 128, 131, 136, 142, 148, 150, 159, 168
- initial 122, 125
- reader of a channel 58
- response 122, 124, 128, 131, 136, 142, 148, 150, 159, 162, 165, 168, 170

channel ID 58, 59
- definitions 59

Clear CAM Address 150
cleared 8
Command Control Block (CCB) 128, 130, 135, 136, 137, 139, 140, 142, 145, 148, 150, 151, 152, 153, 157, 159, 160, 163, 164, 165, 166, 168, 170, 171, 174
- format 58

command tag 64, 65

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

(field) 122, 125, 128, 131, 136, 142, 148, 150, 153, 157, 159, 160, 163, 164, 166, 168, 171, 174
in Tx commands 122, 125, 128, 131
of a buffer list element 134
of Rx indication 163
commands
5211-specific command set 69
CAM Address Response 166
Clear CAM Address 150
Command Control Block (CCB) 58
Flush CAM Addresses 151
generic RC command set 178
Get CAM Address 159
Get Hardware Parameters 157
Get MIB Attribute 142
MIB Attribute Response 168
padding 58
Perform Maintenance 140
Report CAM Address 174
Report Hardware Parameters 171
Report Station Address 164
Request Station Address 136
Ring Control 139
Ring Status Indication 165
Rx Frame 162
Set CAM Address 148
Set Hardware Parameters 153
Set MIB Attribute 144
Set Station Address 137
Set Up Receive Buffers 134
SMT Event Indication 170
SMT Trace 152
Transmit Datagram 127
Tx Response 160
Common Boot 27, 29, 31, 33
communications buffer 5, 160
connection management 8
connector 21

D

DAS 3
data communications buffer 175
data fragment
command tag 163
length 129, 163
physical address 123, 129
data transfers 7, 8, 60, 123
transfer options 59, 123, 129, 132, 135
width of (5211-to-host) 60
datagram 127, 160
Default Bus Timeout 178
default response channel 123, 129, 137, 150, 159, 162
destination address 128, 131, 148
DMA 8, 63, 123, 129, 134, 135, 178
DMA Transfers 178
driver 1
driver compatibility 5
Dual Attach Stations 3
dual attachment 3, 19, 20, 23

E

environment 177
Ethertype 128, 131

F

FDDI
bus bandwidth issues 8
connectors 21
network 6
specifications 4
Station Management functions 8
station types 3
FDDI Configurations 175
FDDI references 2
FDDI specifications 1
FIFO 7
Flush CAM Address 151
frame status 163

G

generic RC command set 48
Get CAM Address 159
Get Hardware Parameters 157
Get MIB Attribute 142

H

host-to-board
channel 39
channel ID 59

I

Insufficient Receive Buffers 66

J

jumper settings
JA1 15
JA15 15
JA16 15
JA17 15
JA19 15
JA2 15
JA3 15
JA4 16
JA8 - JA14 16

L

LEDs 177

M

MAC address 137, 174
maintenance control options 140
Management Information Base (MIB) 68
MIB Attribute Response 168
MIC 3
MIC (Media Interface Connector) 18
MIC receptacles 18, 20, 23, 25

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

N

network interface 175
numbers
representation of 9

O

on-board processor 7
optical bypass switch 23, 24, 25, 176
sample part number 19
to data link 19
OSI 8

P

Perform Maintenance 140
phsycial attributes 176
physical address 64, 65
physical layer 2
power requirements 176
processor 175

R

receive buffers 63, 64
receive status 163
received frame
length 163
number of fragments 163
receivers 23, 25
RELATED PUBLICATIONS AND STANDARDS 1
Report CAM Address 159, 166, 174
Report Hardware Parameters 171
Report Station Address 164
Request Station Address 136
requester options 175
Reserved
bits 9
fields 9
response channel 162
Ring Control 61, 139
ring control options 139
ring status 165
Ring Status Indication 61, 165
Rx Frame 162
Rx frame processing 66
Rx Load Adjustment 135
Rx load adjustment 135
Rx/Tx connector 22, 23, 24

S

SAS 3
Scatter/Gather Capabilities 66
set 8
Set CAM Address 148
Set Hardware Parameters 153
Set Interrupt 53
Set MIB Attribute 144
Set Station Address 137
Set Up Receive Buffers 63, 64, 66, 134
short I/O 7, 16
definition 8
setting base address 16
short I/O space 27, 39
Single Attach Stations 3
single attachment 3, 22
slave address modifiers 175
SMT
message 170
SMT Event Indication 170
SMT Trace 152
SRAM 7
ST 3
ST connector 22
ST connectors 18, 23, 24
station address 137, 148, 150, 164
Station Management 8
synchronous 62, 122, 125, 129, 132
specifying for Tx frame 122, 125, 129, 132

T

timer defaults 176
trace control 152
Transfer Options Word 59
Transmit Datagram 62, 127, 160
Transmit Raw Data 62, 160
transmit status 160
transmitters 23, 25
Tx Response 62, 160
Tx/Rx connector 22, 23, 24

V

VMEbus 7, 8
address modifier 60
connectors 179
data transfers 175
VMEbus Master 175
address modifiers 175
VMEbus slave
options 175
VRAM 7

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Artisan Technology Group is an independent supplier of quality pre-owned equipment

Gold-standard solutions

Extend the life of your critical industrial, commercial, and military systems with our superior service and support.

We buy equipment

Planning to upgrade your current equipment? Have surplus equipment taking up shelf space? We'll give it a new home.

Learn more!

Visit us at **artisantg.com** for more info on price quotes, drivers, technical specifications, manuals, and documentation.

Artisan Scientific Corporation dba Artisan Technology Group is not an affiliate, representative, or authorized distributor for any manufacturer listed herein.

We're here to make your life easier. How can we help you today?

(217) 352-9330 | sales@artisantg.com | artisantg.com

ARTISAN® TECHNOLOGY GROUP