TEXAS INSTRUMENTS

TMS380 Second-Generation Token Ring

User's Guide

1990

Local Area Network Products

TMS380 Second-Generation Token Ring

User's Guide

1990

TMS380 Second-Generation Token Ring User's Guide

TEXAS INSTRUMENTS

IMPORTANT NOTICE

Texas Instruments (TI) reserves the right to make changes to or to discontinue any semiconductor product or service identified in this publication without notice. TI advises its customers to obtain the latest version of the relevant information to verify, before placing orders, that the information being relied upon is current.

TI warrants performance of its semiconductor products to current specifications in accordance with TI's standard warranty. Testing and other quality control techniques are utilized to the extent TI deems necessary to support this warranty. Unless mandated by government requirements, specific testing of all parameters of each device is not necessarily performed.

TI assumes no liability for TI applications assistance, customer product design, software performance, or infringement of patents or services described herein. Nor does TI warrant or represent that license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right of TI covering or relating to any combination, machine, or process in which such semiconductor products or services might be or are used.

Texas Instruments products are not intended for use in life-support appliances, devices, or systems. Use of a TI product in such applications without the written consent of the appropriate TI officer is prohibited.

Preface

Read This First

How to Use This Manual

This User's Guide provides information about the TMS380 Second-Generation Token-Ring COMMprocessor for hardware and software engineers, network planners, and architects. The software engineer will find IEEE 802 Communications Services discussed in Chapter 2 and the System Software Interface discussed in Chapter 4 particularly useful. Hardware engineers should focus on Chapter 3 and the Data Sheets in Appendix A, B, and C; these provide hardware information on the TMS380C16 and the TMS38053. Network planners and architects should review Chapter 2.

Chapter 1 **Introduction**
Provides information about the second-generation TMS380C16 Token-Ring COMMprocessor, including general enhancements and specific features, and compares features with the first-generation TMS380 LAN adapter chipset.

Chapter 2 **IEEE 802 Communications Services**
Describes the architecture, operation, and services provided by the TMS380 adapter as it relates to the physical, medium access control (MAC), and the logical link control (LLC) layers, network management and security services, and error reporting.

Chapter 3 **Token-Ring Adapter Hardware Design**
Describes the hardware interface as viewed from the host system, the ring interface, and expansion memory.

Chapter 4 **System Software Interface**
Describes the software interface as viewed from the host system.

Chapter 5 **Customer Information**
Provides packaging, numbering, and ordering information.

Appendix A TMS380C16 Token-Ring COMMprocessor Data Sheet
Provides product-engineering specifications.

Appendix B TMS38053 Ring Interface Data Sheet
Provides product specifications for interfacing with the TMS38053.

Appendix C TMS380SRA Source Routing Accelerator Data Sheet
Provides product specifications for interfacing with the TMS380SRA.

Appendix D MAC Frame Summary
Provides a summary of the MAC frames discussed in this user's guide.

Appendix E Test, Quality, and Reliability
Provides a summary of TMS380C16 quality assurance that includes self-test diagnostics, fault isolation and recovery, hardware error checkers, and reliability stress tests.

Appendix F Ring Interface Design Notes
Describes a set of guidelines for the layout of the ring interface circuit.

Appendix G Host-Mapped EPROM for Download of TMS380 Adapter Software
Describes the implementation of a host-mapped EPROM for a boot-time download of the adapter software to a TMS380C16-based adapter card.

Appendix H Adapter-Based EPROM for TMS380C16 Adapter Software
Describes the implementation of EPROMs on the TMS380C16 adapter local bus side, with respect to DRAMs.

Appendix I ASCII Character Sets
Lists ASCII characters that the TMS380 Token Ring recognizes, or that may be recognized and acted upon by other programs.

Appendix J Glossary of Terms
Lists and defines terms used throughout this book.

Related Documentation

Texas Instruments provides extensive documentation to support the TMS380 family devices and department tools, including user's guides, data sheets, application reports, technical articles, and a newsletter. Here is a brief list of the documents that are available:

- ❑ *TMS380 Adapter Chipset User's Guide*, Revision D (literature number SPWU001D)
- ❑ *TMS380 Adapter Chipset User's Guide Supplement*, (literature number SPWU003)

Suggested References

Additional information may be found in the following list of suggested reading.

IBM Token-Ring Network Architecture Reference (IBM, #6165877).

IBM Token-Ring Network PC Adapter Technical Reference (IBM, #69X7862).

Token Ring Access Method and Physical Layer Specification, ANSI/IEEE Standard 802.5-1989.

Logical Link Control, ANSI/IEEE Standard 802.2-1989.

Style and Symbol Conventions

This document uses the following conventions.

□ Program listings, program examples, interactive displays, filenames, and symbol names are shown in a `special font`. Examples use a **`bold`** version of the special font for emphasis. Here is a sample program listing:

```
0011  0005  0001        .field    1, 2
0012  0005  0003        .field    3, 4
0013  0005  0006        .field    6, 3
0014  0006                  .even
```

□ In syntax descriptions, the instruction, command, or directive is in a **bold face font** and parameters are in *italics*. Portions of a syntax that are in **bold face** should be entered as shown; portions of a syntax that are in *italics* describe the type of information that should be entered. Here is an example of a directive syntax:

.asect **“***section name***”**, *address*

.asect is the directive. This directive has two parameters, indicated by *section name* and *address*. When you use .asect, the first parameter must be an actual section name, enclosed in double quotes; the second parameter must be an address.

- ❑ Square brackets ([and]) identify an optional parameter. If you use an optional parameter, you specify the information within the brackets; you don't enter the brackets themselves. Here's an example of an instruction that has an optional parameter:

  LALK *16–bit constant [, shift]*

  The **LALK** instruction has two parameters. The first parameter, *16-bit constant*, is required. The second parameter, *shift*, is optional. As this syntax shows, if you use the optional second parameter, you must precede it with a comma.

  Square brackets are also used as part of the pathname specification for VMS pathnames; in this case, the brackets are actually part of the pathname (they are not optional).

- ❑ Braces ({ and }) indicate a list. The symbol | (read as *or*) separates items within the list. Here's an example of a list:

  ```
  { * | *+ | *- }
  ```

  This provides three choices: `*`, `*+`, or `*-`.

  Unless the list is enclosed in square brackets, you must choose one item from the list.

- ❑ Some directives can have a varying number of parameters. For example, the .byte directive can have up to 100 parameters. The syntax for this directive is:

  .byte *$value_1$ [, ... , $value_n$]*

  This syntax shows that .byte must have at least one value parameter, but you have the option of supplying additional value parameters, separated by commas.

- ❑ Bit, byte, and word numbering throughout this manual are based upon bit 0, byte 0, or word 0 occupying the most significant position. Thus, if a 16-bit word is being referenced, bit 0 is the most significant bit position and bit 15 is the least significant bit position.

- ❑ All hexadecimal numbers are preceded with an > symbol.

Information about Cautions and Warnings

This book may contain cautions and warnings.

□ A **caution** describes a situation that could potentially damage your software or equipment.

This is what a caution looks like.

CAUTION

□ A **warning** describes a situation that could potentially cause harm to **you**.

This is what a warning looks like.

WARNING

The information in a caution or a warning is provided for your protection. Please read each caution and warning carefully.

Trademarks

PC–DOS is a trademark of International Business Machines.
TROLI is a trademark of Pulse Engineering Incorporated.

Contents

1 **Introduction** **1-1**
1.1 TMS380 Overview 1-2
1.2 TMS380C16 Features 1-5
1.3 TMS380 Enhancements Overview 1-12
1.4 TMS380C16 Specific Enhancements 1-13
1.4.1 System Interface Block 1-13
1.4.2 Communications Processor Block 1-14
1.4.3 Protocol Handler Block 1-15
1.4.4 LAN Adapter Bus 1-15
1.4.5 Memory Interface Block 1-16
1.5 User Software Modifications for Second-Generation Chipset 1-18

2 **IEEE 802 Communications Services** **2-1**
2.1 OSI Reference Model 2-2
2.2 IEEE 802 LAN Standards 2-4
2.3 IEEE 802.5 Physical Layer Signaling 2-8
2.3.1 Differential Manchester Code 2-8
2.3.2 Ring Clocking 2-9
2.3.3 Elastic Buffer 2-9
2.4 Data Link Control Communications Services 2-11
2.4.1 Network Management Considerations 2-11
2.4.2 IEEE 802.5 Frames 2-16
2.4.3 Ring Addressing 2-21
2.4.4 MAC Services 2-28
2.4.5 IEEE 802.2 LLC Services 2-113

3 **Adapter Hardware Design** **3-1**
3.1 Ring Interface 3-2
3.1.1 Ring Interface Pins 3-2
3.1.2 Ring Interface Options 3-5
3.2 Memory Interface 3-6
3.2.1 Memory Map 3-6
3.2.2 Memory Timing Control 3-8

3.3 Enhanced Address Copy Option (EACO) Interface 3-35
3.4 Attached System Interface 3-38
3.4.1 Direct I/O for 808x Mode 3-39
3.4.2 DIO for 68xxx Interface 3-51
3.4.3 DMA Operations 3-58
3.4.4 808x Mode DMA Operation 3-59
3.4.5 68xxx-Mode DMA Operations 3-68
3.4.6 Burst/Cycle-Steal DMA Modes 3-77
3.4.7 System DMA Data Organization 3-78
3.4.8 System DMA Address Latching 3-83
3.5 Hardware Notes 3-88

4 System Software Interface 4-1
4.1 Usage of DIO and DMA 4-2
4.1.1 Description and Usage of DIO 4-3
4.1.2 Description and Usage of DMA 4-4
4.2 Adapter Data Flow 4-8
4.2.1 Receive Data Flow 4-10
4.2.2 Transmit Data Flow 4-11
4.2.3 Summary of System Buffer Requirements 4-14
4.3 DIO Register Interface 4-16
4.3.1 SIFACL—SIF Adapter Control Register 4-18
4.3.2 SIFADX, SIFADR, SIFDAT, and SIFDAT/INC Registers 4-23
4.3.3 SDMADAT Register 4-26
4.3.4 DMALEN Register 4-26
4.3.5 SDMAADR Register 4-26
4.3.6 SDMAADX Register 4-26
4.3.7 Pseudo-DMA Operation 4-27
4.3.8 8-Bit DIO 4-28
4.3.9 SIFVEC Interrupt Acknowledge Register 4-30
4.3.10 SIFCMD and SIFSTS Registers 4-30
4.4 Adapter Reset 4-37
4.4.1 Hardware Reset 4-37
4.4.2 Software Reset 4-38
4.5 Bring-up Diagnostics—BUD 4-40
4.6 Adapter Initialization 4-42
4.6.1 The Initialization Block 4-42
4.6.2 Writing the Initialization Block 4-46
4.7 Adapter Tests 4-49
4.7.1 RAM Tests 4-49
4.7.2 BIA Tests 4-49

4.8 System Command Block—SCB 4-50
4.9 System Status Block—SSB 4-52
4.10 Interface Control Block—ICB 4-53
4.11 Frame Buffers 4-57
4.12 Adapter-to-System Interrupts 4-60
4.12.1 Interrupt Handling 4-60
4.12.2 RING.STATUS 4-61
4.12.3 COMMAND.REJECT 4-64
4.12.4 ADAPTER.CHECK 4-66
4.13 Adapter MAC-Only Commands 4-70
4.14 Adapter LLC Commands 4-124

5 Customer Information 5-1
5.1 TMS380 Adapter Software Licensing Procedures 5-2
5.1.1 Second Generation Software 5-2
5.1.2 First Generation Software 5-5
5.2 TMS380 Software Ordering Information 5-6
5.2.1 Second Generation Software Options 5-6
5.2.2 First Generation to Second Generation Conversion Software .. 5-7
5.2.3 First Generation Software Options 5-7
5.3 Mechanical Package Information 5-8
5.3.1 Second Generation Packages 5-9
5.3.2 First Generation Packages 5-11
5.4 TMS380 Family Numbering and Symbol Conventions 5-14
5.4.1 Device Prefix Designators 5-14
5.4.2 Device Numbering Convention 5-15
5.4.3 Device Symbols 5-16
5.5 TMS380 Pricing and Delivery 5-19

A TMS380C16 Token-Ring COMMprocessor Data Sheet A-1

B TMS38053 Ring Interface Data Sheet B-1

C TMS380SRA Source Routing Accelerator Data Sheet C-1

D MAC Frame Summary D-1

E Test, Quality, and Reliability E-1
E.1 Network Management and Reliability E-2
E.1.1 Self-Test Diagnostics E-2
E.1.2 Fault Isolation and Recovery E-2
E.2 Network Management Services E-3

E.3 Semiconductor Quality and Reliability E-4
E.3.1 Reliability Stress Tests E-4

F Ring Interface Design Notes F-1
F.1 Circuit Diagrams F-2
F.2 Circuit Diagram Layout Notes F-8
F.2.1 Digital Data and Control F-8
F.2.2 Differential Data Signals F-8
F.2.3 Analog Signals F-8
F.2.4 Differential Drive Signals F-8
F.3 System Layout F-9
F.3.1 Isolation Buffer F-9
F.3.2 Power and Ground Layout F-10

G Host-Mapped EPROM for Download of TMS380 Adapter Software G-1
G.1 Host System Features G-2
G.2 Theory of Operation G-3
G.3 Memory Map G-4
G.4 Hardware and PAL Equations G-6
G.5 EPROM Software Program G-9

H Adapter-Based EPROM for TMS380C16 Adapter Software H-1
H.1 Theory of Operation H-2
H.2 EPROM Access Conditions H-4
H.3 Adapter Timing Considerations H-5
H.4 Hardware Schematic Description H-6
H.5 Timing Requirements H-7
H.6 Bus Loading Considerations H-8

I ASCII Character Sets I-1

J Glossary J-1

Figures

1–1 First-Generation TMS380 Typical Adapter Configuration 1-2
1–2 TMS380 First Generation – Second Generation 1-4
1–3 Industry-Standard Software Interface 1-5
1–4 IBM PC/XT Implementation with TMS380C16 1-7
1–5 TMS380 Trend: Increased Performance 1-8
1–6 Increase in Packet Size with TMS380C16 1-9
1–7 TMS380C16 Supports 255 Link Stations 1-10
1–8 16-Mbps Early Token Release 1-11
2–1 The OSI Reference Model 2-2
2–2 IEEE 802 LAN Standards 2-4
2–3 TMS380 IEEE 802.5 Physical Access Services 2-5
2–4 16-Mbps Early Token Release 2-6
2–5 TMS380 IEEE 802.5 Medium Access Control Services 2-7
2–6 TMS380 IEEE 802.2 Logical Link Control Services 2-7
2–7 Differential Manchester Code 2-8
2–8 Code Violations 2-9
2–9 Frame Format 2-17
2–10 Starting Delimiter Bit Assignments 2-18
2–11 AC Bit Assignments 2-18
2–12 FC Field 2-19
2–13 Ending Delimiter Bit Assignments 2-20
2–14 FS Bit Assignments 2-21
2–15 Destination Address Format 2-22
2–16 Source Address Format 2-22
2–17 Broadcast Address 2-24
2–18 Functional Address 2-24
2–19 Group Address 2-24
2–20 Ring-Station Address Format 2-25
2–21 Routing Information Field Format 2-26

2–22 Segment Number Field 2-27
2–23 General MAC Frame Format 2-28
2–24 Response Code Subvector 2-39
2–25 Transmit Forward MAC Frame 2-70
2–26 Model AC 2-73
2–27 Token Capture Flowchart 2-74
2–28 Token Is at Point Shown 2-80
2–29 Station A Captures Token and Begins Transmission of Frame 2-81
2–30 Station B Repeats Frame and Changes PR to 4 2-82
2–31 Station C Repeats Frame and Changes PR to 6 2-83
2–32 Station A Strips Frame from the Ring 2-84
2–33 Station A Releases a New Token 2-85
2–34 Station B Repeats Token and Changes PR to 4 2-86
2–35 Station C Captures Token and Begins Transmission of Frame 2-87
2–36 Frame Circulates with No Modification 2-88
2–37 Station C Begins Stripping the Frame from the Ring 2-89
2–38 Station C Issues a New Token of Same Priority 2-90
2–39 Station A Performs a Replace and Transmits the Token 2-91
2–40 Station B Captures Token and Transmits Frame 2-92
2–41 Station B Strips Its Frame from the Ring 2-93
2–42 Station B Issues a New Token at the Original Priority 2-94
2–43 Station A Pops Its Stacks and Issues Priority-Free Token 2-95
2–44 Solid Hard Error Recovery Time Line 2-101
2–45 Intermittent Hard Error Recovery Time Line 2-102
2–46 Soft Error Recovery Time Line 2-106
2–47 Transmitter Fault in Adapter 114 2-107
2–48 Receiver Fault in Adapter 185 2-108
2–49 Broken Ring Fault 2-109
2–50 Transmitter Streaming Fault in Adapter 185 2-110
2–51 Adapter 117 Detects a Streaming Error 2-111
2–52 Adapter 185 Streaming Claim Token Frames 2-112
2–53 A Typical LLC Session 2-116
2–54 AP/Link Station Association 2-117
2–55 LLC Frame 2-117
3–1 Chapter 0 Memory Map (RAM or ROM) 3-7

3–2 Chapter 1 Memory Map (RAM) 3-7
3–3 Read from Burned-In Address PROM (BIA) ROM 3-18
3–4 Logical Representation of BIA 3-20
3–5 Chapter 0 or 31 Read from EPROM 3-22
3–6 Read from DRAM 3-26
3–7 Write to DRAM 3-28
3–8 DRAM Refresh Cycle 3-32
3–9 TMS380C16 DRAM Memory Interconnection with BIA PAL 3-34
3–10 Internal Transparent Latch for DIO in 808x Mode 3-41
3–11 808x-Mode DIO Read Cycle 3-45
3–12 808x-Mode DIO Write Cycle 3-49
3–13 Interrupt Acknowledge Timing for 808x Mode 3-50
3–14 68xxx-Mode DIO Read Cycle Timing 3-53
3–15 68xxx-Mode DIO Write Cycle Timing 3-56
3–16 68xxx-Mode Interrupt Acknowledge Cycle Timing 3-57
3–17 808x-Mode DMA Read Timing 3-60
3–18 808x-Mode DMA Write Timing 3-61
3–19 SIF Acquires System Bus—808x Mode 3-65
3–20 SIF Returns System Bus—808x Mode 3-67
3–21 68xxx-Mode DMA Read Timing 3-69
3–22 68xxx-Mode DMA Write Timing 3-71
3–23 SIF Acquires System Bus—68xxx Mode 3-72
3–24 SIF Returns System Bus—68xxx Mode 3-74
3–25 Bit- and Byte-Numbering Conventions 3-79
3–26 Formats for Storing Character String ABCD in Memory 3-80
3–27 External TTL Latches Demultiplex Address/Data Bus 3-84
3–28 Address Extension Updates for 8- and 16- Bit Modes 3-86
3–29 Comparison of Address Bit Formats 3-87
4–1 Motorola Format 4-5
4–2 Intel Format 4-6
4–3 Intel Double-Word Format 4-7
4–4 Second-Generation TMS380 Adapter Subsystem 4-9
4–5 Adapter Data Flow 4-10
4–6 Transmitting a Frame 4-14
4–7 SIFACL—SIF Adapter Control Register 4-18

4–8 Pseudo-DMA Logic Related to SIFACL Bits 4-21
4–9 DIO Byte Transfers in 8- or 16-Bit Modes 4-29
4–10 SIFCMD and SIFSTS Register Write-Bit Assignments 4-32
4–11 SIFCMD and SIFSTS Register Read-Bit Assignments 4-34
4–12 Initialization Parameter Block 4-42
4–13 System Command Block Format 4-50
4–14 System Status Block Format 4-52
4–15 Interface Control Block 4-53
4–16 Attached System Logical Frame Format 4-57
4–17 RING_STATUS Field Bit Assignments 4-62
4–18 COMMAND_REJ_STATUS Field Bit Assignments 4-64
4–19 ADAPTER_CHECK_STATUS Field Bit Assignments 4-67
4–20 TMS380C16 COMMProcessor Command Set 4-70
4–21 Adapter Internal Buffer Format 4-82
4–22 Attached System Frame Logical Format 4-84
4–23 TRANSMIT Parameter List 4-87
4–24 TRANSMIT List Format: Example 1 4-96
4–25 TRANSMIT List Format: Example 2 4-97
4–26 TRANSMIT List Format: Example 3 4-98
4–27 TRANSMIT List Format: Example 4 4-98
4–28 RECEIVE Parameter List 4-101
4–29 Error Log Table 4-112
4–30 Host System Data Area Format 4-115
4–31 BRIDGE_PARM_BLOCK 4-120
4–32 TMS380C16 COMMProcessor Command Set 4-124
4–33 TRANSMIT Parameter List 4-138
4–34 TRANSMIT Parameter List 4-149
4–35 RECEIVE Parameter List 4-162
4–36 LLC RECEIVE Command Flow 4-166
4–37 Error Log Table 4-171
4–38 Host System Data Area Format 4-174
4–39 BRIDGE_PARM_BLOCK 4-179
4–40 SAP_ Parm_ Block 4-185
4–41 STATION_PARAMETER_BLOCK 4-191
4–42 CONNECT_PARM_BLOCK 4-196

4–43 MODIFY_PARM_BLOCK 4-199
4–44 STAT_PARM_BLOCK 4-204
5–1 TMS380C16 132-Pin Plastic Quad Flat Package,
25-Mil Pin Spacing (Type PQ Package Suffix) 5-9
5–2 TMS38053 and TMS380SRA Plastic-Leaded Chip Carrier Package,
50-Mil Pin Spacing (Type FN Package Suffix) 5-10
5–3 TMS38010, TMS38021, TMS38051, TMS38052 Plastic Dual-Inline
Packages,100-Mil Pin Spacing (Type N Package Suffix) 5-11
5–4 TMS38030 132-Pin Quad Flat Package, 25-Mil Pin Spacing
(Type PQ Package Suffix) 5-12
5–5 TMS38030 100-Pin Grid Array Package, 50-Mil Pin Spacing
(Type GB Package Suffix) 5-13
5–6 Development Flowchart 5-14
5–7 TMS380 Family Nomenclature 5-15
5–8 TI Standard Symbolization for TMS380C16 Device
in 132-Pin PQ-Type Package 5-16
5–9 TI Standard Symbolization for TMS38053 Device
in 44-Pin FNType Package 5-16
5–10 TI Standard Symbolization for TMS380SRA Device
in 44-Pin FN Type Package 5-16
5–11 TI Standard Symbolization for TMS38010 Device
in 48-Pin N-Type Package 5-17
5–12 TI Standard Symbolization for TMS38021 Device
in 48-Pin N-Type Package 5-17
5–13 TI Standard Symbolization for TMS38030 Device
in 132-Pin PQ-Type Package 5-17
5–14 TI Standard Symbolization for TMS38030 Device
in 100-Pin GB-Type Package 5-17
5–15 TI Standard Symbolization for TMS38051 Device
in 22-Pin N-Type Package 5-18
5–16 TI Standard Symbolization for TMS38052 Device
in 20-Pin N-Type Package 5-18
F–1 TMS38053 Ring Interface Circuit Diagram (Jumper Switchable) F-4
F–2 TMS38053 Ring Interface Circuit Diagram (Software Switchable) F-5
F–3 Isolation Buffer Interconnection F-9
F–4 Recommended Grounding Layout Diagram F-10
F–5 Recommended Power Layout Diagram F-11
G–1 PAL Equations for U022: G-7

G–2 Schematic of Host-Mapped EPROM G-8
G–3 Algorithm to Download From EPROM to a TMS380C16-Based Board G-9
G–4 Assembly Download Code for Adapter Software G-11
H–1 TMS380C16 EPROM-Based Interface Schematic H-3

Tables

1–1 Second-Generation Token-Ring Design Objectives 1-12
1–2 TMS380C16 SIF Enhancements 1-14
1–3 TMS380C16 Communications Processor Enhancements 1-14
1–4 TMS380C16 Protocol Handler Enhancements 1-15
1–5 TMS380C16 LAN Adapter Bus Enhancements 1-16
2–1 Network Management Functional Addresses 2-12
2–2 MAC Frame Class Designators 2-13
2–3 Source and Destination Class Applications 2-14
2–4 Major Vector Commands 2-30
2–5 Subvector Types 2-31
2–6 MAC Frames Processed by the TMS380 Adapter 2-33
2–7 Response Code Subvector Code Values 2-39
2–8 MAC Protocol Processes 2-44
2–9 Monitor Contention Process – Applicable MAC Frames 2-47
2–10 Events Triggering Monitor Contention 2-48
2–11 Ring Purge Process – Applicable MAC Frames 2-50
2–12 Ring Poll Process Applicable MAC Frames 2-52
2–13 Beacon Process – Applicable MAC Frames 2-55
2–14 Insertion Process – Applicable MAC Frames 2-62
2–15 Transmit Forward Process – Applicable MAC Frames 2-69
2–16 Remove Ring-Station MAC Frame 2-71
2–17 Network Management Request MAC Frames 2-71
2–18 Network Management Response MAC Frames 2-72
2–19 Priority Control Stimuli 2-78
2–20 Report Error MAC Frame 2-97
2–21 Isolating Error Counters 2-98
2–22 Nonisolating Error Counters 2-99
3–1 Ring Interface Pins 3-2

3–2 Recommended Clock Connections 3-4
3–3 List of Boot Bits in ACTL Register Controlling Access Timing 3-8
3–4 Adapter Address Output During Memory Cycle 3-13
3–5 Status Information on MADH0–MADH7 3-15
3–6 Device Code Status on MADH1–MADH4 3-15
3–7 EACO Interface Pins 3-35
3–8 DIO Cycles Held Off 3-41
3–9 808x DIO Read Data Bus Controls 3-43
3–10 808x DIO Write Data Bus Controls 3-47
3–11 68xxx DIO Read Data Bus Controls 3-52
3–12 68xxx DIO Write Data Bus Controls 3-54
3–13 SDTACK, SBERR, and SHALT Assertion Results 3-76
3–14 SDTACK, SBERR, and SHALT Negation Results 3-77
3–15 System DMA Cycle Description 3-82
4–1 Logical DIO Interface 4-17
4–2 SIFCMD and SIFSTS Register Write-Bit Functions 4-33
4–3 SIFCMD and SIFSTS Register Read-Bit Functions 4-35
4–4 DRAM-Based Adapter Hardware Reset Example 4-37
4–5 Bring-Up Diagnostic Error Codes (SIFCMD/SIFSTS Register) 4-41
4–6 Initialization Parameter Block Field Descriptions 4-43
4–7 Adapter Initialization Errors (SIFCMD/SIFSTS Register) 4-47
4–8 Adapter Internal Pointers 4-48
4–9 Interface Control Block Field Descriptions 4-54
4–10 Receive Frame Types 4-56
4–11 Attached System Frame Fields 4-57
4–12 Transmit Data 4-59
4–13 RING_STATUS Field Bit Functions 4-63
4–14 COMMAND_REJ_STATUS Field Bit Functions 4-65
4–15 ADAPTER_CHECK_STATUS Field Bit Definitions 4-68
4–16 OPEN Parameter List 4-72
4–17 OPEN Parameter Block Field Descriptions 4-73
4–18 OPEN_COMPLETION Bit Field Definitions 4-79
4–19 OPEN Phases and OPEN Error Codes 4-80
4–20 OPEN Error Code Descriptions 4-81

4–21 Attached System Frame Fields 4-85
4–22 TRANSMIT Parameter List Fields 4-88
4–23 XMIT_STATUS Bit Definitions4-95
4–24 RECEIVE Parameter Field Bit Definitions 4-102
4–25 ADDRESS_MATCH Codes 4-104
4–26 RECEIVE_COMPLETE Field Bit Definitions 4-108
4–27 CLOSE Return Codes 4-109
4–28 SET.GROUP.ADDRESS Return Codes 4-110
4–29 SET.FUNCTIONAL.ADDRESS Return Codes 4-111
4–30 Adapter Error Counters 4-113
4–31 READ.ERROR.LOG Return Codes 4-114
4–32 READ.ADAPTER Return Codes 4-116
4–33 MODIFY.OPEN.PARAMETERS Return Codes 4-117
4–34 RESTORE.OPEN.PARAMETERS Return Codes 4-118
4–35 SET.FIRST.16.GROUP.ADDRESS Return Codes 4-119
4–36 BRIDGE_PARM_BLOCK Field Definitions 4-121
4–37 SET.BRIDGE.PARMS Return Codes 4-122
4–38 CONFIG.BRIDGE.PARMS Return Codes 4-123
4–39 OPEN Parameter Block 4-126
4–40 OPEN Parameter Block Field Descriptions 4-127
4–41 OPEN_COMPLETION Bit Field Definitions 4-132
4–42 OPEN Phases and OPEN Error Codes 4-133
4–43 OPEN Error Code Descriptions 4-134
4–44 LLC OPEN Error Codes 4-134
4–45 TRANSMIT Parameter List Fields 4-139
4–46 TRANSMIT Frame Types 4-143
4–47 Transmit Error/Link Status Codes 4-143
4–48 XMIT_STATUS Bit Definitions 4-146
4–49 TRANSMIT Parameter List Fields 4-150
4–50 TRANSMIT Frame Types 4-154
4–51 Transmit Error/Link Status Codes 4-154
4–52 XMIT_STATUS Bit Definitions 4-157
4–53 RECEIVE Parameter Field Bit Definitions 4-163
4–54 ADDRESS_MATCH Codes 4-164

4–55 RECEIVE_COMPLETE Field Bit Definitions 4-167
4–56 CLOSE Return Codes 4-168
4–57 SET.GROUP.ADDRESS Return Codes 4-169
4–58 SET.FUNCTIONAL.ADDRESS Return Codes 4-170
4–59 Adapter Error Counters 4-172
4–60 READ.ERROR.LOG Return Codes 4-173
4–61 READ.ADAPTER Return Codess 4-175
4–62 MODIFY.OPEN.PARAMETERS Return Codes 4-176
4–63 RESTORE.OPEN.PARAMETERS Return Codes 4-177
4–64 SET.FIRST.16.GROUP.ADDRESS Return Codes 4-178
4–65 BRIDGE_PARM_BLOCK Field Definitions 4-180
4–66 SET.BRIDGE.PARMS Return Codes 4-181
4–67 CONFIG.BRIDGE.PARMS Return Codes 4-182
4–68 LLC.RESET Return Codes 4-183
4–69 SAP Parameter Block Field Definitions 4-186
4–70 OPEN.SAP Return Codes 4-189
4–71 CLOSE.SAP Return Codes 4-190
4–72 STATION_PARAMETER_BLOCK Field Descriptions 4-192
4–73 OPEN.STATION Return Codes 4-193
4–74 CLOSE.STATION Return Codes 4-194
4–75 CONNECT_PARAMETER_BLOCK Field Bit Descriptions 4-196
4–76 CONNECT.STATION Return Codes 4-197
4–77 MODIFY_PARM_BLOCK Field Definitions 4-200
4–78 MODIFY.LLC.PARMS Return Codes 4-201
4–79 FLOW_OPTIONS Field Bit Positions 4-202
4–80 FLOW.CONTROL Return Codes 4-203
4–81 STATISTICS_PARAMETER_BLOCK Field Descriptions 4-205
4–82 LLC Statistics Data 4-205
4–83 LLC.STATISTICS Return Codes 4-206
4–84 DIR.INTERRUPT Return Codes 4-207
4–85 SCB_PARM_1 Field Description 4-208
4–86 COMMAND_STATUS Return Codes 4-209
4–87 LLC.REALLOCATE Return Parameters 4-209
4–88 TIMER.SET Return Codes 4-210

4–89 TRANSMIT.I.FRAME.REQUEST Return Codes 4-212
5–1 Package Types 5-8
D–1 MAC Frame Major Vectors D-2
D–2 MAC Frame Subvectors D-5
D–3 Adapter Status Vector Decode D-8
E–1 Microprocessor and Microcontroller Tests E-8
E–2 TMS380C16 Transistors E-9
F–1 TMS38053 Signal Classification F-2
F–2 Circuit Diagram Jumpers F-3
F–3 Part Selections and Tolerances F-6
G–1 EPROM Jumper Settings G-5
I–1 ASCII Character Set I-2
I–2 Control Characters I-3

Chapter 1

Introduction

The TMS380 family of LAN adapter chipsets provides manufacturers of computer, peripheral, and telecommunications equipment with a verified connection to the industry-standard IEEE 802.5 IBM Token-Ring Network. The TMS380 is the silicon standard for this high-speed office-system LAN.

This introduction gives an overview of the first-generation chipset and of the enhancements added to the second-generation chipset, which features the TMS380C16 Token-Ring COMMprocessor. Developed using TI MegaChip Technologies, the TMS380C16 includes new features with higher performance, while maintaining complete IBM and IEEE 802.5 compatibility.

Section **Page**

1.1 TMS380 Overview 1-2
1.2 TMS380C16 Features 1-5
1.3 TMS380 Enhancements Overview 1-12
1.4 TMS380C16 Specific Enhancements 1-13
1.5 User Software Modifications for Second-Generation Chipset .. 1-18

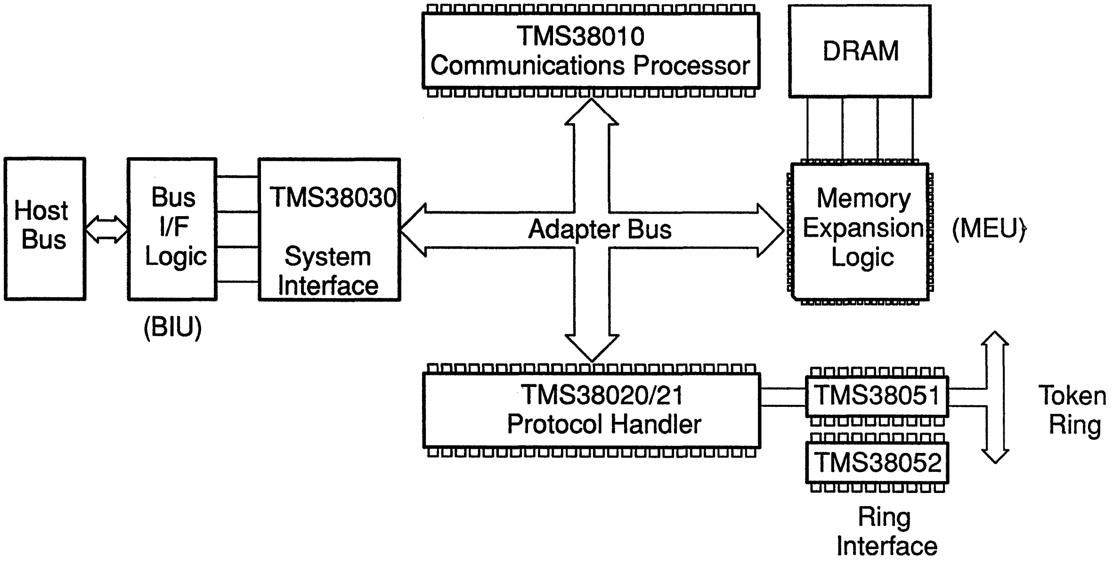
1.1 TMS380 Overview

The first-generation TMS380 integrates the functions of a LAN adapter into a five-chip set. The components of the chipset are

- ❑ TMS38030 System Interface chip,
- ❑ TMS38010 Communications Processor chip,
- ❑ TMS38020/21 Protocol Handler chip, and
- ❑ TMS38051 and TMS38052 Ring Interface pair.

Also supplied with the first-generation TMS380 are two ASIC devices, which allow low board area adapters to be implemented. The PC/ XT Bus Interface Unit (BIU) includes glue interface logic for a PC/XT adapter. The Memory Expansion Unit (MEU) provides the glue logic to interface the TMS380 adapter bus to DRAM. Figure 1–1 shows a typical PC adapter configuration with the first-generation TMS380 chipset.

Figure 1–1. First-Generation TMS380 Typical Adapter Configuration



As Figure 1–2 shows, the TMS380C16 Token-Ring COMMprocessor uses TI's EPIC 1-micron CMOS VLSI technology to integrate the TMS38010 Communications Processor, the TMS38020/21 Protocol Handler, and the TMS38030 System Interface chips of the first-generation TMS380 chipset into a single 132-pin device. In addition, it integrates the function of the TMS380MEU–DR DRAM Memory-Expansion Unit and most of the TMS380BIU-XT PC Bus-Interface Unit.

The TMS380C16 retains all of the functional capability of the first-generation TMS380 chips, including compatibility with IBM and capacity as an industry-standard software interface; in addition, the TMS380C16 offers significant new features:

❑ Lower cost, area, and power

❑ Higher performance: 4 or 16 Mbps

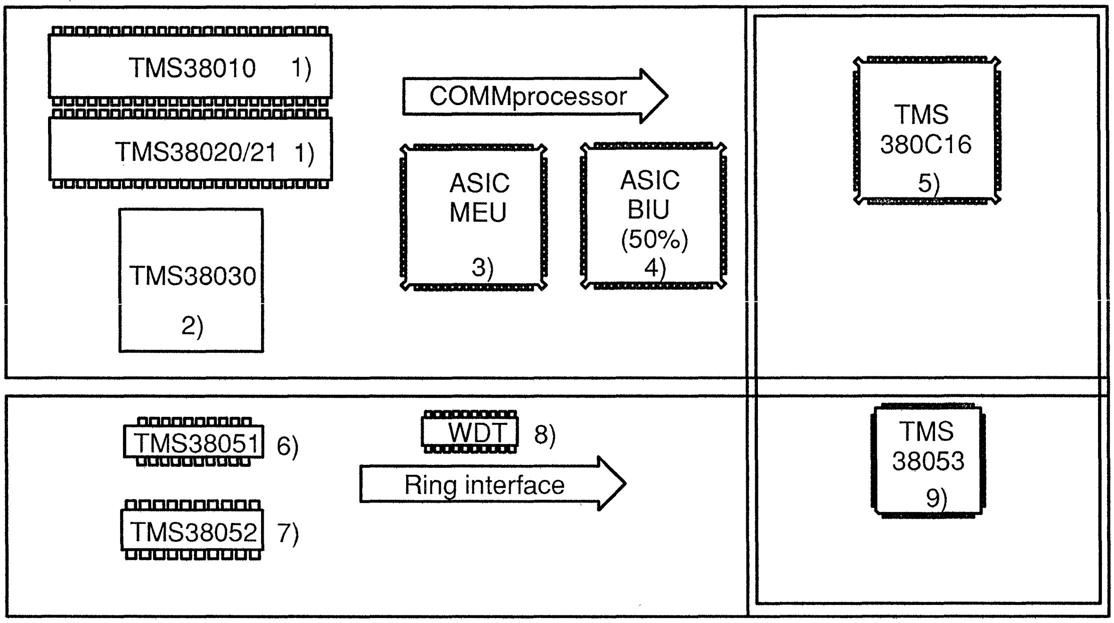
❑ Larger frame sizes

❑ Up to 255 link stations

❑ Early token release at 16 Mbps

❑ On-chip downloadable standard protocols

Figure 1–2. TMS380 First Generation – Second Generation



Notes: 1) 48 CDIP (2.4-μm NMOS)
2) 100 PGA (2.4-μm NMOS)
3) 68 PLCC (2.0-μm CMOS)
4) 100 QFP (2.0-μm CMOS)
5) 132-pin QFP (EPIC 1-μm CMOS)
6) 20 PDIP (LSTTL)
7) 22 PDIP (LSTTL)
8) 14 PDIP (LSTTL)
9) 44 PLCC (1.5-μm ALSTTL)

Also shown in Figure 1–2, the TMS38053 integrates the TMS38051, TMS38052, and a watchdog timer into a single 1.5-μm ALSTTL chip. Besides supporting compatible ring operation at 4 Mbps, this advanced device facilitates 16-Mbps ring clock recovery and signal conditioning.

1.2 TMS380C16 Features

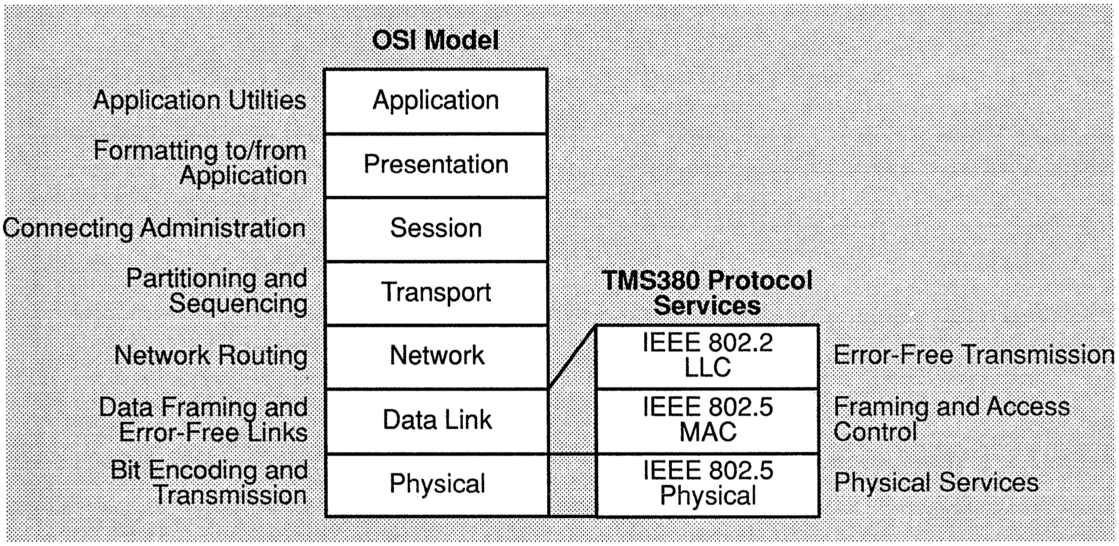
□ IBM Compatibility

The TMS380C16 has the same IBM compatibility as the first-generation TMS380, plus overall increased system performance. That compatibility is rigorously verified through original test cases applied to the first-generation chipset, and through new test cases for bridging and 16-Mbit operation, that validate the TMS380C16's advanced features.

□ Industry-Standard Software Interface

The TMS380C16 provides the same industry-standard software services of the first-generation TMS380. Figure 1–3 shows that IBM compatibility is provided at the physical layer, medium access control (MAC) layer, and the logical link control (LLC) layer of the OSI reference model. All adapters using TI's TMS380 family of products interoperate with IBM and other vendors because the MAC and LLC functions are implemented in the TMS380 silicon and associated software. In most cases, existing network software should run unchanged on the higher speed 16-Megabit-per-second (Mbps) network.

Figure 1–3. Industry-Standard Software Interface



Key:		
	LLC	Logical link control
	MAC	Medium access control

- **Lower Cost**

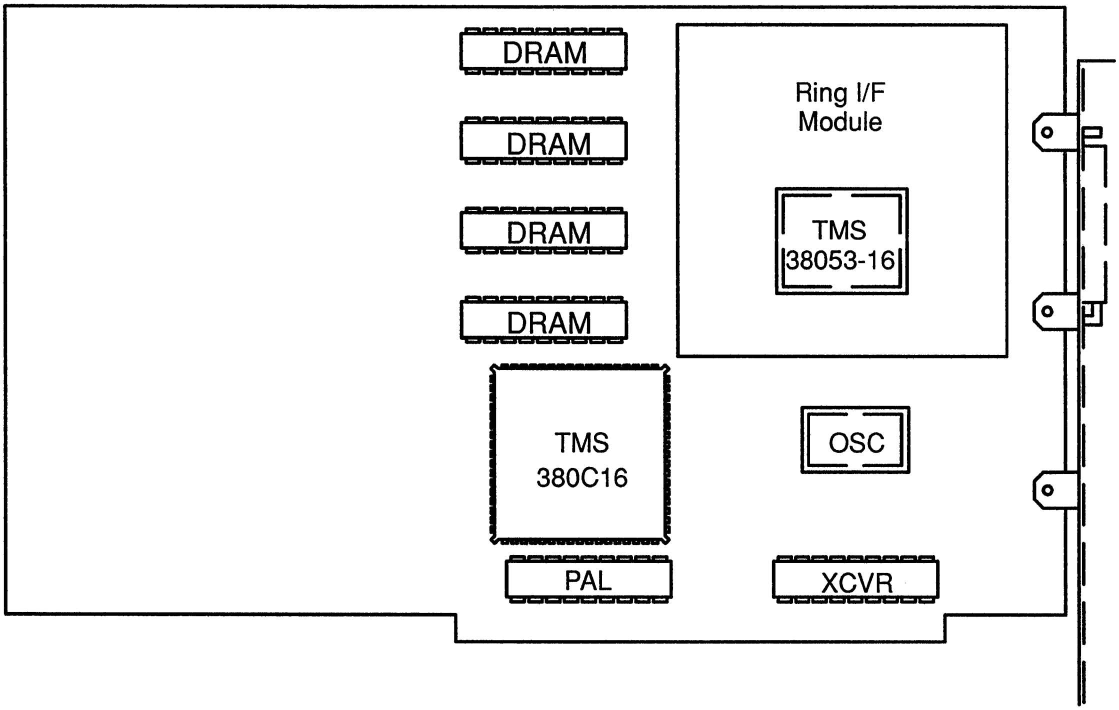
  The TMS380C16 is a highly integrated semiconductor device, incorporating functions previously performed by five chips. This high integration reduces overall cost of the token-ring product. The TMS380C16 incorporates many external functions that previously required TTL, PALs, or ASICs. They include the DRAM memory-expansion unit functions, as well as most of the bus interface unit functions. Additional manufacturing cost can be saved by reduced board size (less expensive blanks) and reduced number of drilled holes in the board for IC pins or circuit routing.

  For low-cost 4-Mbps solution, the first-generation TMS380 five-chip set, in plastic, will be the price leader into the early 1990s. If your token-ring design must fit into only 20 square inches, with a 4-Mbps data rate, the first-generation TMS380 will meet your requirements.

- **Less Area**

  You can use the TMS380C16 to design a complete token-ring LAN adapter with a minimal number of parts. Since TI has integrated most of the functions of a token-ring adapter into a single chip, all that is needed to build an adapter is the TMS380C16 Token-Ring COMMprocessor, a ring interface module containing the TMS38053, onboard memory, and minimal bus interface logic, as shown in Figure 1–4. The area requirement for a full-function, TMS380C16-based, IBM-compatible, token-ring adapter for a PC/XT is less than 10 square inches.

Figure 1–4. IBM PC/XT Implementation with TMS380C16



- ❑ **Lower Power**

  Power consumption is reduced on the second-generation chipset because the TMS380C16 uses advanced 1-micron CMOS technology. It averages less than 1 watt of power dissipation which is less than half that of first-generation NMOS chips. This translates to lower equipment operating temperatures.

- ❑ **Higher Performance**

  - ■ **Mbps Data Rate**

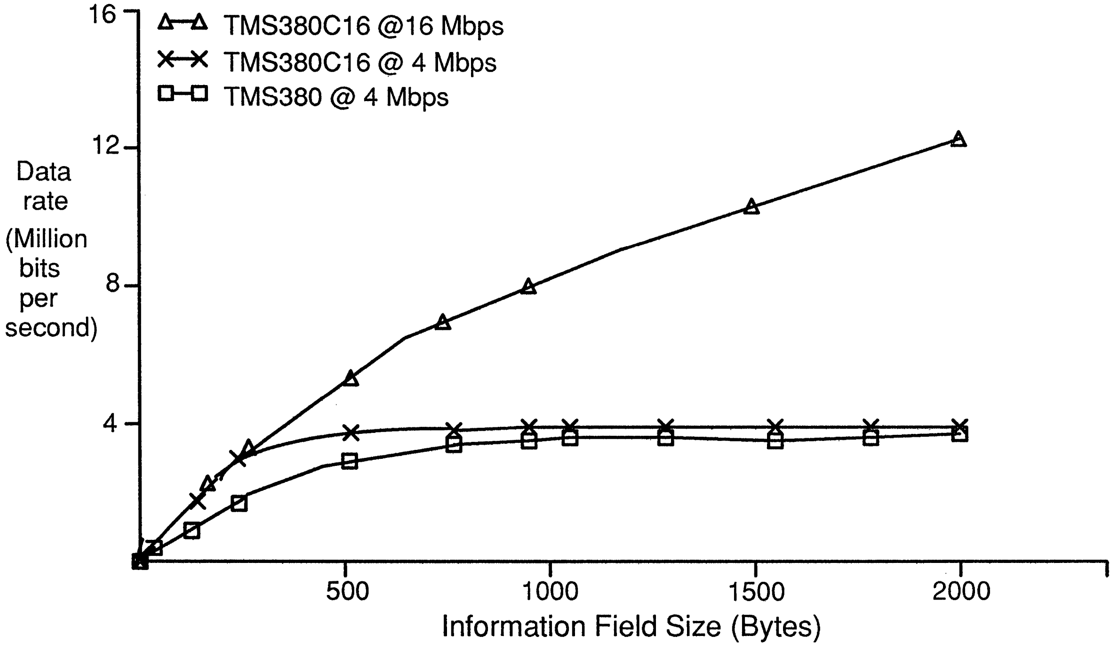
    The TMS380C16 supports the new IEEE 802.5 and IBM 16-Mbps token-ring standard, providing a 400% increase in data rate. This gives token-ring a data rate 60% faster than Ethernet, while providing users with the reliable technology of token-ring. Because the efficiency of the token passing medium access protocol enables the current 4-Mbps token-ring to perform on par with today's 10-Mbps Ethernet, it follows that 16-Mbps token-ring could deliver up to 400% the performance of Ethernet in heavily loaded networks.

■ Increased Small Packet Performance

The TMS380C16 also outperforms the first-generation chipset by providing increased small-packet performance at both 4 and 16 Mbps. This increases network throughput when connection-oriented protocols (such as those based on LLC Type II, which include IBM's protocols) are used, since they generate many small packets as acknowledgements. This performance improvement was achieved by adding direct hardware support in the TMS380C16 for many of the performance limiting tasks, such as DMA setup, that were handled by firmware in the first-generation TMS380.

Figure 1–5 represents actual performance data taken using the TMS380C16 at the medium access control layer. The squares represent the first-generation TMS380 at 4 Mbps. The cross-hatched line represents the second-generation TMS380C16 performance operating at 4 Mbps. The triangles represent the TMS380C16 operating at 16 Mbps on the token-ring.

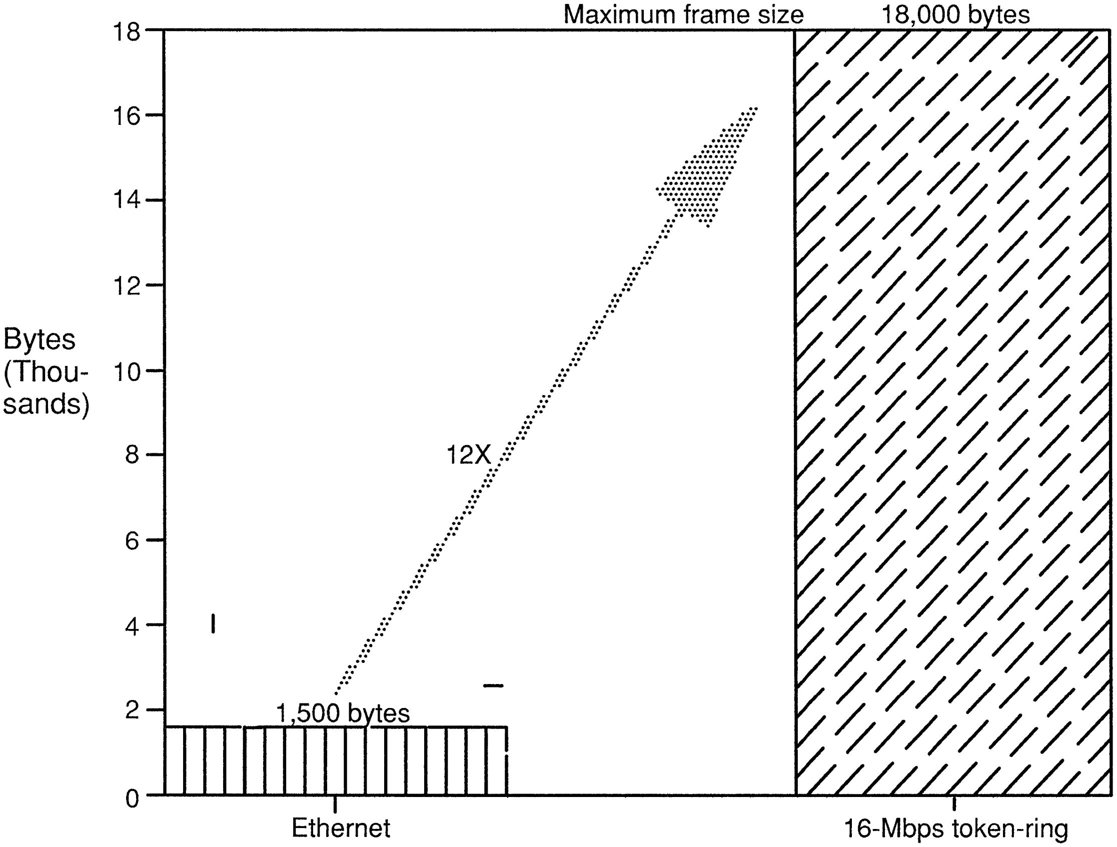
Figure 1–5. TMS380 Trend: Increased Performance



❑ Supports Larger Frames

The 400% increase in data rate and increase in small-packet performance are not the only new performance features of the TMS380C16. The 16-Mbps token-ring supports frame sizes up to 18,000 bytes. This large data transport facility can increase data transmission efficiency by as much as 300%. The 4-Mbps token-ring can support frame sizes up to 4,500 bytes; Ethernet at 10 Mbps supports frame sizes of up to 1,500 bytes. With Ethernet at 10 Mbps, a frame of 16,000 bytes must be broken up and transmitted in 11 different frames. On the token-ring at 4 Mbps, this file requires four frames. On a 16-Mbps token-ring, this 16,000-byte file requires only one frame for transmission, and that frame can be transmitted potentially at 4 times the speed of either Ethernet or 4-Mbps token-ring.

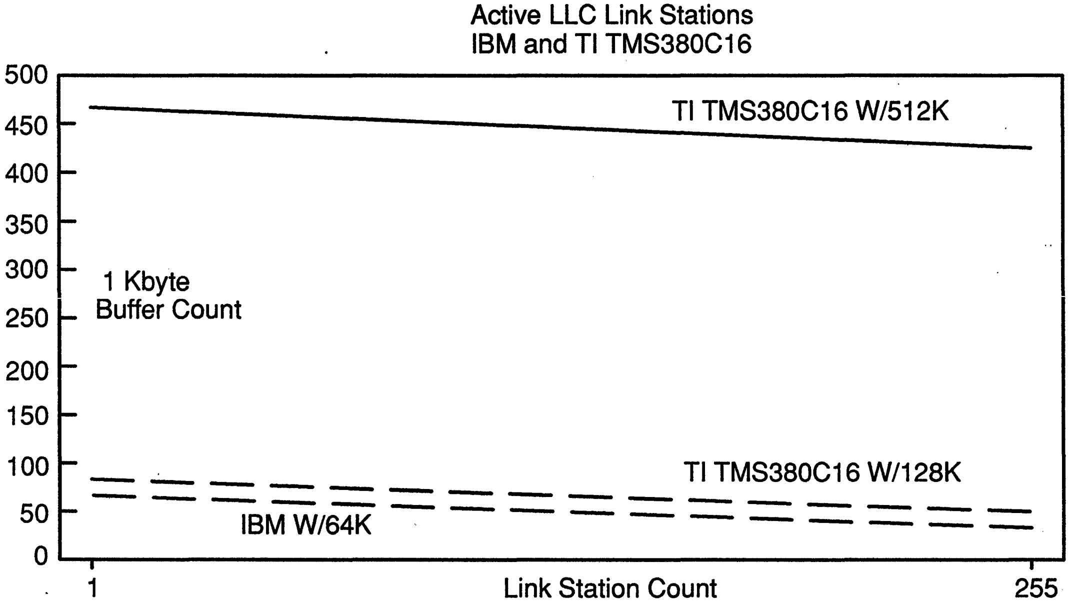
Figure 1–6. Increase in Packet Size with TMS380C16



❏ **Up to 255 Link Stations**

With 2 megabytes of extended memory capability, a TMS380C16-based adapter can support up to 255 link stations, 126 SAPs, and up to 1960 1-Kbyte buffers. With 512 kilobytes of extended memory, 390 1-Kbyte buffers are available, which is more than adequate for most applications. Figure 1–7 illustrates the number of buffers and link stations for different memory sizes. This capability enables the TMS380C16 to efficiently handle both workstation and server applications.

Figure 1–7. TMS380C16 Supports 255 Link Stations

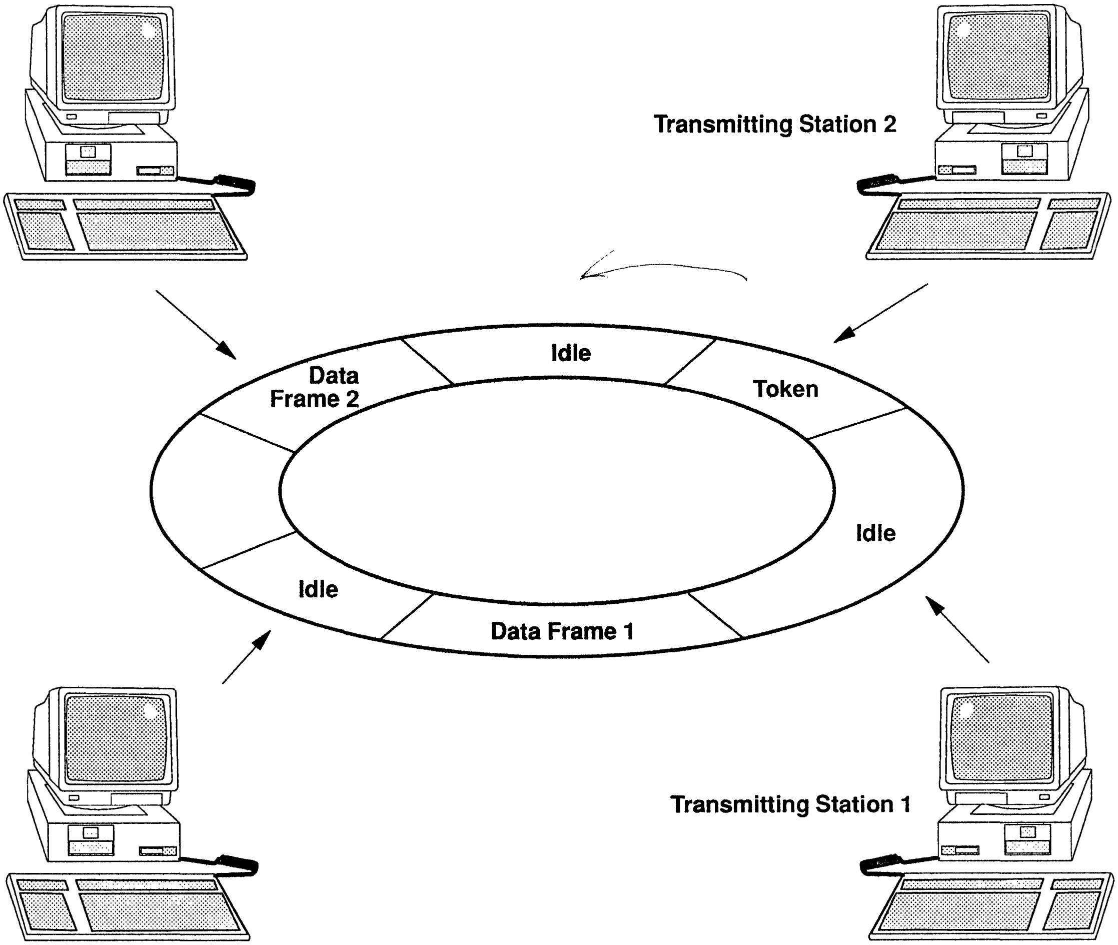


❑ **Early Token Release**

The **early token release** feature, incorporated in the IEEE 802.5 Token-Ring Standard, enhances the original token-ring architecture by allowing multiple frames on the ring at the same time. The new 16-Mbps implementation allows a station to transmit a token immediately after sending a frame, without waiting for the frame to return and be stripped from the network. Ring efficiency can be enhanced by allowing stations to place more data on the network.

Figure 1–8 shows operation of early token release. Station 1 sends a data frame, then issues a free token. Station 2 captures the token, transmits a frame, and issues a free token. Station 1 receives its frame back and strips it from the network. No token is issued because this was already done by using the early token release option. For more information on early token release, refer to the IEEE 802.5 1989 specifications.

Figure 1–8. 16-Mbps Early Token Release



1.3 TMS380 Enhancements Overview

Although the TMS380C16 is similar in many ways to the first-generation TMS380 chipset, the following enhancements have been made:

- ❑ Adapter address space expanded to 2 megabytes supports expanded link station tables with LLC, expanded data buffers, and use of downloadable standard protocols.
- ❑ System interface host address extended to 32 bits supports full address space of new high-performance 32-bit microprocessor and system buses.
- ❑ On-chip workspace register cache added to CPU significantly improves adapter performance and decreases context-switch delays.
- ❑ Choice of internal adapter bus speeds is offered: 3 MHz for lower memory costs or 4 MHz for increased performance at 16 Mbps.
- ❑ Optional parity on adapter bus can be engaged for adapter memory integrity or disengaged for lower memory cost (one less memory chip).
- ❑ Glueless DRAM interface supports 256K, 1-Mbyte, and 4-Mbyte DRAMs directly with no additional decode logic or DRAM control required.
- ❑ Improved local adapter bus provides greater flexibility in memory organization and allows DRAM, SRAM, and EPROM components to be supported with minimal glue logic.
- ❑ System interface block DMA control capabilities are improved by adding a number of hardware counters that previously were implemented externally or in software.
- ❑ Adapter control register (ACTL) added in the system interface block that allows the host to exert greater control over the adapter; this offers more system flexibility.

Table 1–1. Second-Generation Token-Ring Design Objectives

Objective	Approach
Maintain IBM compatibility	Same architecture/tests Object code compatible No change to ring task
Increase data transfer performance	Add hardware to speed up Software limited paths
Lower cost, area, and power	Integrate TMS380 chipset Integrate glue logic
Increase protocol performance	2-Mbyte memory space 3× Faster CPU with cache

1.4 TMS380C16 Specific Enhancements

As discussed earlier, the TMS380C16 integrates the functions of the first-generation TMS38010, TMS38020/21, TMS38030, the Memory Expansion ASIC, and approximately 50% of the Bus Interface ASIC. This section describes the enhancements made to each of these logic areas as they were incorporated into the TMS380C16.

1.4.1 System Interface Block

The TMS38030 System Interface (SIF) function throughput has been improved by adding more features to the DMA controller. The system interface now has hardware counters for controlling system bus retries, burst size, and for keeping track of host and adapter buffer status. Previously, these counters needed to be maintained in software. Integrating them into hardware removes software overhead and improves adapter performance.

The host can now exert greater control over the adapter with the adapter control (ACTL) register in the system interface. This register may be used to trigger adapter reset, to disable interrupts to the host, to control parity checking, to enable pseudo-DMA for host systems not capable of supporting bus master DMA, and to give the host additional status information.

The SIF contains three other new registers on the host side. These registers are Adapter DMA Length (ADMALEN), System DMA Length (SDMALEN), and System DIO Address Extension (SIFADRX). ADMALEN and SDMALEN allow the host to read the adapter and host buffer lengths. This simplifies pseudo-DMA for host systems that do not support external bus masters. Pseudo-DMA is an 8-bit DMA transfer system regardless of DMA size latches on the inactive edge of SRESET. SIFADRX is an extension to SIFADR, which allows DIO to access the full 2 megabytes of external adapter memory.

Table 1–2. TMS380C16 SIF Enhancements

Feature	1st-Generation TMS38030	2nd-Generation TMS380C16
System bus clock	10 MHz	16 MHz
System bit transfer rate	40 Mbps	64 Mbps
System address reach	16 Mbyte (24 bits)	4 GByte (32 bits)
System data bus	16 bits	16 bits
Byte parity on address/data	Yes	Program enable
System bus retry	Microcode	H/W counter
DMA burst size	Microcode	H/W counter
Adapter buffer status	Microcode	H/W register
System DMA buffer status	Microcode	H/W register
Adapter control register – pseudo DMA – interrupt control – power-up/download support	N/A External logic Limited Limited	H/W register On-chip Enhanced Enhanced

1.4.2 Communications Processor Block

The TMS380C16 CPU, previously contained on the TMS38010, now contains a register cache to speed accesses to workspace registers and save adapter bus bandwidth. Saving adapter bus bandwidth is especially important on 16-Mbps rings where DMA can use a larger portion of adapter bandwidth.

Table 1–3. TMS380C16 Communications Processor Enhancements

Feature	Benefit
16-bit CPU tailored for protocol processing	Approach physical data rate delivered to applications
4 MIPS with workspace cache	High protocol throughput
2-megabyte address space	Large buffers/data throughput
MAC (supervisor) mode	Protects IBM compatibility and RAS
Pipelined adapter bus arbitration	64-Mbps adapter throughput
Built-in timer	Protocol timers

1.4.3 Protocol Handler Block

The TMS38020/21 Protocol Handler (PH) function has been improved to speed up adapter software operations. These changes include increasing the number of transmit channels to eight, separating the interrupt requests, and supporting the extended addressing mode of the TMS38021.

The protocol handler now has 8 transmit chain pointer (TCP) registers so that frames can be queued according to their priority. Previously, there was only one TCP register and managing frame queues of different priorities was done in software, limiting transmitter performance.

Buffers are stored at 1-Kbyte boundaries. This means that only 11 bits are needed to locate a buffer anywhere in the 2 megabyte address space.

Table 1–4. TMS380C16 Protocol Handler Enhancements

Feature	1st-Generation TMS38020/21	2nd-Generation TMS380C16
Address bus (bits)	15	20
Buffer memory (Kbytes)	44	2000
Transmit Queues	1	8
Hardware-prioritized transmit	No	Yes
Interrupt levels	One, shared	Three, separate
Clock sync elastic buffer (baud)	14	15 and 63
Early token release	No	Optional
Universal/local address	38021	Yes
Enhanced Address Copy Option (EACO)	38021	Yes
Network data rate (Mbps)	4	16 or 4

1.4.4 LAN Adapter Bus

The TMS380C16 incorporates the first-generation devices onto a single silicon die, including the TMS380 LAN Adapter Bus, which interfaces these function blocks to each other and to adapter memory. The adapter bus has several enhancements; these provide increased throughput for both 4- and 16-Mbps networks.

TMS380C16 adapter bus is a glueless interface to DRAM and saves precious board area and cost. The specific details on the memory interface are discussed in section 1.4.5. Parity on data space is optional with the second-generation TMS380.

Designers now have the option of running the adapter bus at either 3 MHz or 4 MHz. At 16 Mbps on the token ring, the adapter bus must operate at 4 MHz; at 4 Mbps it can operate at either 3 or 4 MHz. 3 MHz operation allows for the use of slower DRAMs.

Table 1–5. TMS380C16 LAN Adapter Bus Enhancements

Feature	1st-Generation TMS38010/20/30	2nd-Generation TMS380C16
Dram Interface	External MEU	Built-in
Adapter bus clock	3 MHz	3 or 4 MHz
Max bit rate	48 Mbps	64 Mbps
Parity on data	Mandatory	Optional
BIA support	PH/RI	Bus
Memory Reach (KBytes)	104	2000

1.4.5 Memory Interface Block

The TMS380C16 is a highly integrated memory interface with the following enhancements:

- ❑ Integrated DRAM interface that automatically handles
  - ■ DRAM-refresh control, and
  - ■ Address decode for up to 2 megabytes.
- ❑ Glueless DRAM interface for memory expansion of 128K, 512K, and 2 megabytes, using
  - ■ 256K DRAM (64K × 4), or
  - ■ 1 MEG DRAM (256K × 4), or
  - ■ 4 MEG (1M × 4).
- ❑ Glueless burn-in-address (BIA) PROM interface
- ❑ Simplified EPROM/ROM interface with bank select

The TMS380C16 memory interface attaches to DRAMs and a burned-in-address PROM (or PAL) with no glue logic and attaches to SRAMs and EPROMs with minimal glue.The memory interface block contains a DRAM-refresh controller to ease connections to DRAMs, and code and data mapping registers to extend the TMS380C16 address to 20 bits. This enables access to 2 megabytes of memory. Dedicated chip-select output pins

($\overline{\text{MROMEN}}$ and $\overline{\text{MBIAEN}}$) are provided for selecting ROM/EPROM and burned-in address PROM.

The memory interface block allows the adapter to support an abundance of on-board memory made up of DRAMs. Consequently, the buffering techniques used for handling frames can be simplified without fear of running out of adapter memory. For example, it is now possible to use one large buffer per frame and therefore relieve the software overhead of buffer management that is required when sending and receiving multiple buffers per frame.

1.5 User Software Modifications for Second-Generation Chipset

The second-generation chipset has been designed to require minimum user software changes when upgrading from the first-generation chipset. The high-level system commands are the same, and most software using these commands can be the same. The enhancements described in the previous section are automatically incorporated without any software change. The following changes must be considered:

- ❑ User Address Space

  All address parameters specifying user memory can be 32 bits in the second-generation chipset. The first-generation chipset supported only a 24-bit address.

- ❑ Registers

  A new Adapter Control Register (ACTL) must be set properly. See section 4.3.1 for proper bit handling of this register.

  Two new psuedo-DMA mode registers (ADMALEN and SDMALEN), which are read-only registers, can be used to support systems that do not allow DMA. See sections 4.3.3 and 4.3.4 for proper use of these registers.

  One extended DIO address register (SIFADRX) allows full DIO access to the expanded TMS380C16 adapter memory map. See section 4.3.6 for proper use of this register.

- ❑ Open Command

  The OPEN command contains parameters specifying the memory size and buffer allocation in the TMS380C16 adapter DRAM. The buffer size default for the second-generation is 1024, with a minimum of 1024. The first-generation chipset default was 112, with a minimum of 96.

  The expansion RAM address and size are determined during DRAM self-test. Therefore, these parameters do not need to be specified.

- ❑ Reset Sequence

  The second-generation chipset requires that the adapter firmware (MAC or MAC/LLC) be downloaded into DRAM by a power-up and reset sequence. The first-generation chipset contained MAC code in ROM on-board the TMS38020/TMS38021. See section 4.4 for reset method and download process for the second-generation chipset.

Chapter 2

IEEE 802 Communications Services

The IEEE 802.5 and IEEE 802.2 standards provide the basepoint for the token-ring protocols. These two standards specify the Physical and Data Link Control layers in the OSI reference model. The TMS380 chipset allows the implementation of these standards in an IBM-compatible LAN. The IEEE 802.5 standard specifies the Media Access Control implementation with a rich feature set of addresses and ring management functions. The IEEE 802.2 standard specifies the Logical Link Control implementation. The details of these standards, as implemented by the TMS380 chipset, are described here.

This chapter includes the following topics:

Section **Page**

2.1 OSI Reference Model 2-2
2.2 IEEE 802 LAN Standards 2-4
2.3 IEEE 802.5 Physical Layer Signaling 2-8
2.4 Data Link Control Communications Services 2-11

2.1 OSI Reference Model

The open systems interconnection (OSI) model is a conceptual network structure defined by the International Organization for Standardization. The OSI model promotes the development of worldwide data communications standardization.

To reduce their design complexity, networks are partitioned into a series of layers. These layers are hierarchical, with each layer built upon its predecessor, thus shielding each layer from the details of how the services from the other layers are implemented. The OSI model is partitioned into seven layers. The bottom four layers of the model define the network and how it functions, and the top three layers define how the network is used.

Figure 2–1. The OSI Reference Model

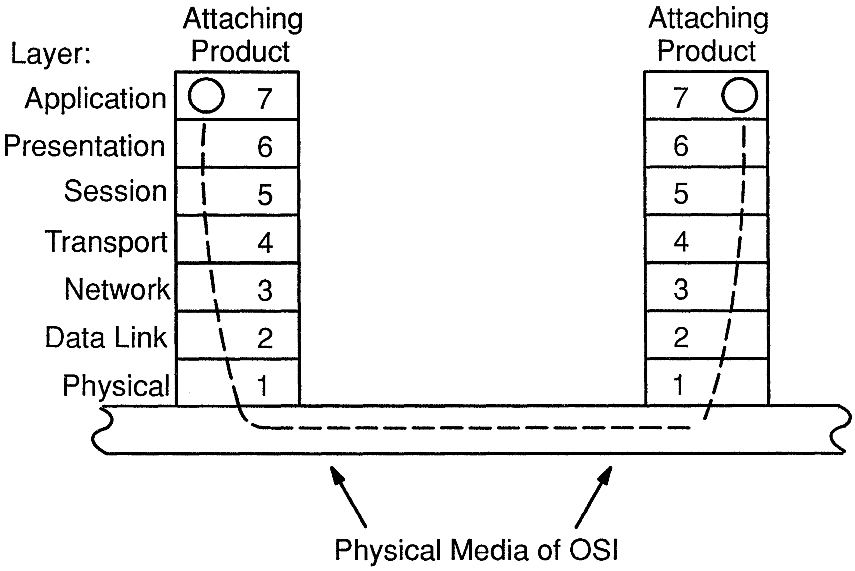


Figure 2–1 shows the relationship of the seven layers of the OSI model to one another and the path taken during communication between two attaching products.

The functions of the layers are described here, starting with the lowest.

Physical layer defines the mechanical and electrical connection.

Data link layer defines the way data is formatted for transmission and how access to the network is controlled. This layer has been separated by the IEEE 802 standards committee into two sublayers: the medium access control (MAC) and the logical link control (LLC) sublayers.

Network layer defines the routing and switching of data between networks.

Transport layer ensures that data is sent to and arrives at the destination correctly.

Session layer defines the end user's interface into the network and establishes and manages communication dialogs.

Presentation layer maps applications data structures to and from the session layer.

Application layer provides network-based services to an end user's application programs.

2.2 IEEE 802 LAN Standards

The IEEE 802 committee has developed a set of local area network standards and protocols for the physical and data link layers of the OSI model.

The data link layer has been divided by the IEEE 802 committee into two sublayers: the medium access control (MAC) sublayer and the logical link control (LLC) sublayer. To meet a variety of end-user requirements, the IEEE 802 committee has defined three MAC technologies with associated physical media:

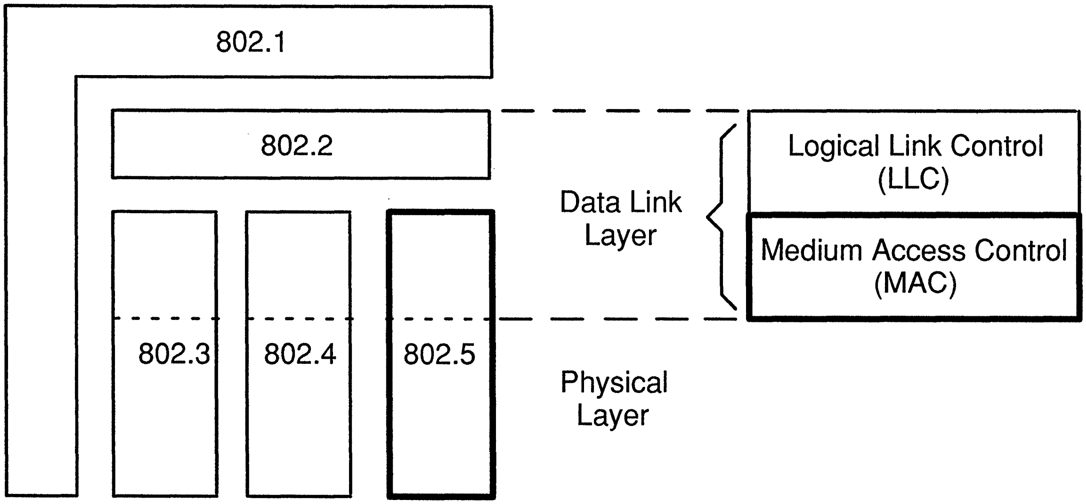
❑ IEEE Standard 802.3, a bus topology using a carrier sense multiple access/collision detect (CSMA/CD) access method

❑ IEEE Standard 802.4, a bus topology using a token-passing access method

❑ IEEE Standard 802.5, a star-wired ring topology using a token-passing access method

In addition to the published standards listed above, the IEEE 802 committee has defined a logical link control standard (802.2) and an internetworking and network management standard (802.1).

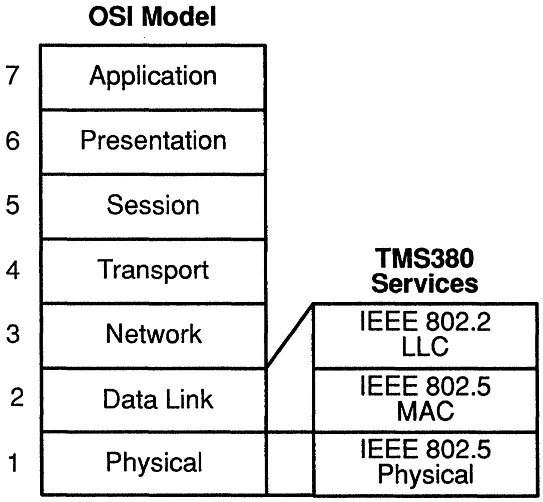
Figure 2–2. IEEE 802 LAN Standards



With its integrated communications processor architecture, the second-generation TMS380C16 provides host-independent operation of the token-ring. The TMS380 implements the IBM and IEEE international data link control layer standards. Performing the physical and MAC functions of the IEEE Standard 802.5 as well as the LLC services of IEEE Standard 802.2, the TMS380 off-loads the host and allows multi-vendor interoperability.

As Figure 2–3 shows, the TMS380 provides extensive physical layer access services and low-level error detection.

Figure 2–3. TMS380 IEEE 802.5 Physical Access Services



- □ Token-ring electrical connection
- □ Phase lock loop for clock generation from data signal
- □ 4- or 16- Mbps differential Manchester encode/decode
- □ Independent transmit and receive channels
- □ Phantom drive for physical insertion into ring
- □ Cable wire fault indication
- □ Receive data loss detection
- □ Receiver frequency equalization and low level hysteresis
- □ Loop back for self-test diagnostics
- □ Complete IBM-compatible IEEE 802.5 physical layer services

At the MAC sublayer of the data link control (DLC) layer, the TMS380 provides the medium access control services of the IEEE 802.5 Token-Ring Standard shown in Figure 2–5.

The token-ring network provides a new feature at 16 Mbps: the **early token release** feature, incorporated in the IEEE 802.5 Token-Ring Standard, enhances the original token-ring architecture by allowing multiple frames on the ring at the same time. The new 16-Mbps implementation allows a station to transmit a token immediately after sending a frame, without waiting for the frame to return and be stripped from the network. Ring efficiency can be enhanced by allowing stations to place more data on the network.

Figure 2–4 shows operation of early token release. Station 1 sends a data frame, then issues a free token. Station 2 captures the token, transmits a frame and issues a free token. Station 1 receives its frame back and strips it from the network, but does not issue a token because it already did so using the early token release option. For more information on early token release refer to the IEEE 802.5 specification.

Figure 2–4. 16-Mbps Early Token Release

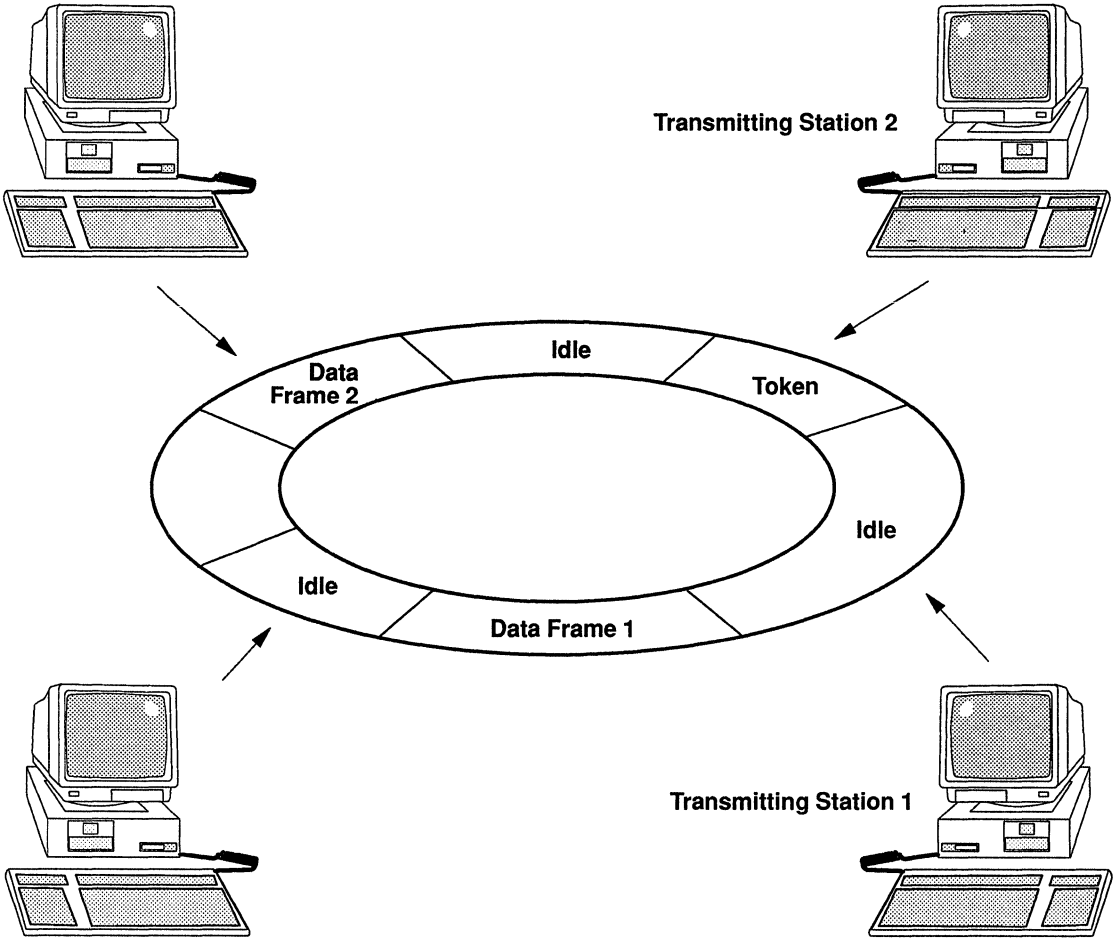
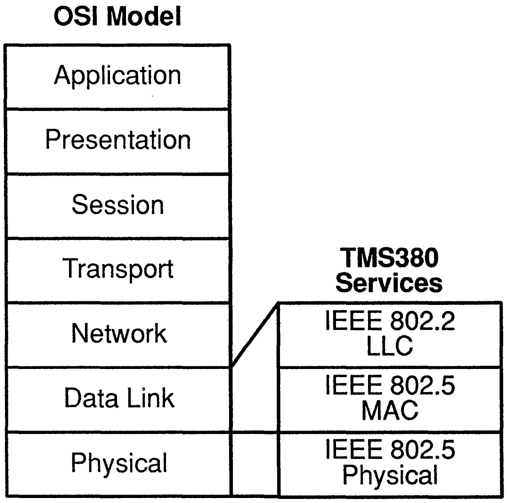


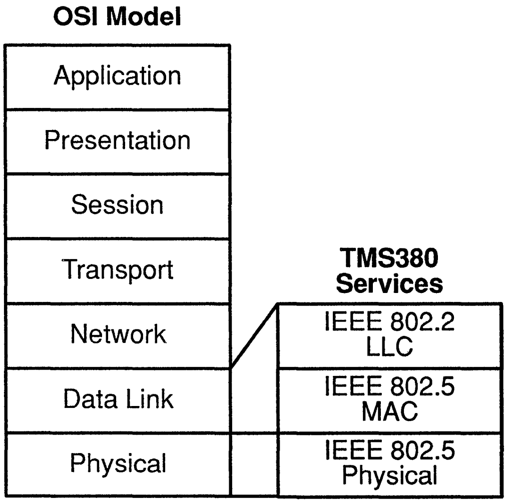
Figure 2–5. TMS380 IEEE 802.5 Medium Access Control Services



- ❑ Token establishment protocols
  - ■ Frame transmission, reception
  - ■ FCS generation and checking
- ❑ Ring maintenance and token control
  - ■ Active/standby monitor functions
  - ■ Ring error recovery and error isolation
- ❑ Ring management control
  - ■ Network management reporting services
  - ■ Network management parameter control
- ❑ Early token release at 16 Mbps
- ❑ Complete IBM-compatible IEEE 802.5 access services

The TMS380 also provides capability for LLC protocol software. This interface provides a complete set of IBM compatible IEEE 802.2 connectionless (Type 1) and connection-oriented (Type 2) LLC functions. Figure 2–6 highlights extensive LLC functions provided.

Figure 2–6. TMS380 IEEE 802.2 Logical Link Control Services



- ❑ Standard interface to IBM network software
- ❑ Standard interface between multiple upper layer protocol software and IEEE 802.5 MAC and physical layers
- ❑ Organizes physical network node into multiple logical network entities (service access points) for program communications
- ❑ Supports virtual data paths between logical network entities (links)
- ❑ Provides guaranteed delivery and proper sequencing of information between logical entities (type 2 LLC)
- ❑ Provides logical address headers and routes received frames (type 1 LLC)
- ❑ Obtains routing information for source-routed bridging
- ❑ Complete IBM-compatible IEEE 802.2 LLC services

2.3 IEEE 802.5 Physical Layer Signaling

2.3.1 Differential Manchester Code

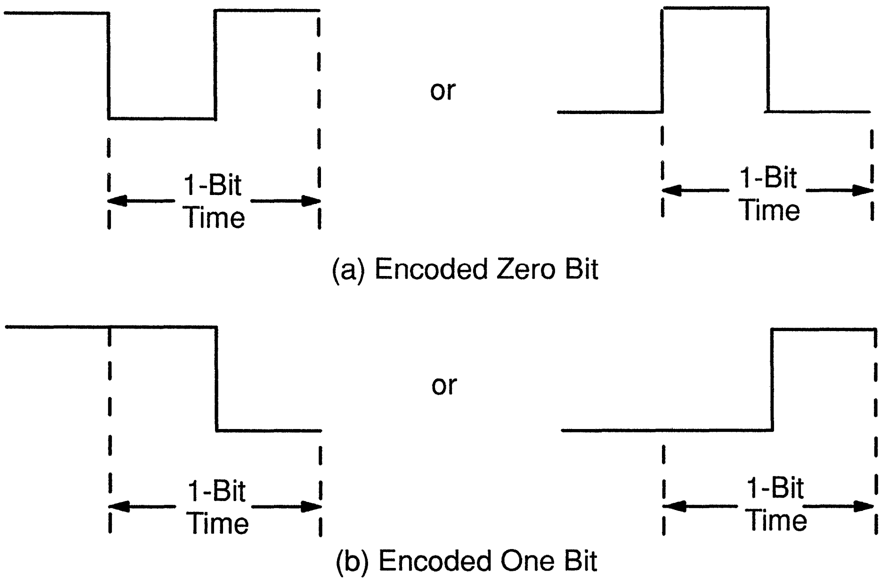
The token-ring protocol uses a ring signaling format called **differential Manchester code.** This signaling scheme follows these rules:

❑ A signal transition always occurs in the center of the bit time.

❑ A zero bit has a transition at the beginning of the bit time. A one bit has no transition during this time.

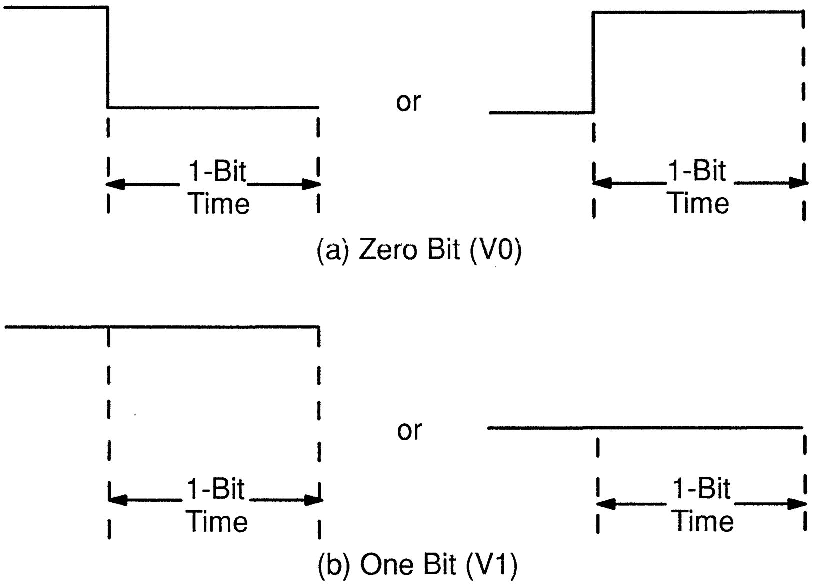
Figure 2–7 illustrates this coding scheme. The signaling transitions are symmetric around the zero-volt level, thus providing an average zero-volt DC level. This facilitates transformer coupling of the TMS380C16's transmitter and receiver to the ring.

Figure 2–7. Differential Manchester Code



Another advantage of differential Manchester code is that a violation of the coding rules can easily be found for error detection purposes as well as for providing a convenient method of synchronizing bit streams. Figure 2–8 illustrates four possible violation patterns that are designated V0 for the zero-bit violation and V1 for the one-bit violation. These patterns are used within starting delimiters and ending delimiters for frame and token boundary synchronization. They will be referenced throughout this manual.

Figure 2–8. Code Violations



2.3.2 Ring Clocking

An adapter, randomly designated via a contention process, provides master clocking to the ring by deriving the ring's timebase from a crystal oscillator. This adapter, as will be shown later, is called the active monitor. Any adapter on the ring may assume the role of active monitor.

The remaining ring stations on the ring derive their timebase by phase-synchronizing a voltage-controlled oscillator to the incoming bit stream. This clock, derived from the phase-locked-loop (PLL) provides the necessary timebase from which the bit stream is received and transmitted by the TMS380C16.

2.3.3 Elastic Buffer

The active monitor clocks all transmissions with its crystal-controlled clock. This can present a problem when the active monitor receives the incoming bit stream, because the bit stream is no longer phase-aligned to the crystal-controlled clock. This loss of phase alignment can be caused by phase delays of the repeating stations on the ring. This misalignment is referred to as **jitter**.

To compensate for jitter, the active monitor adapter clocks the receive data with its PLL-generated clock. It clocks receive data into a first-in-first-out (FIFO) buffer, called the **elastic buffer,** with the PLL-generated clock, and clocks transmit data out of the buffer using its crystal-generated clock. The buffer holds **15 half-bits (bauds) at 4-Mbps** operation, and **63 half-bits (bauds) at 16-Mbps** ring operation to compensate for anticipated worst-case jitter of repeating adapters.

Other adapters use the elastic buffer to monitor the ring for frequency errors. If the active monitor's clocking frequency differs significantly from the crystal-controlled clock of another adapter, the elastic buffer will overflow or underflow on that adapter, indicating a frequency error.

2.4 Data Link Control Communications Services

This section describes the system architecture and communications services provided by the Texas Instruments second-generation TMS380 Token-Ring COMMprocessor. It gives details that the network planner and the software engineer need in order to design network services software.

The discussion of communications services focuses on those services provided at the data link control (DLC) layer of the ISO open system interconnect (OSI) reference model for local area networks. This layer incorporates both the IEEE 802.5 MAC sublayer and the IEEE 802.2 LLC sublayer.

The general topics covered progress from basic concepts to detailed processes:

- ❑ **Network management considerations.** Discusses the network management services provided by the TMS380. (Section 2.4.1)
- ❑ **IEEE 802.5 Frames.** Discusses the physical frame formats and the addressing scheme. (Section 2.4.2)
- ❑ **Ring addressing.** Discusses address formats on IEEE 802.5 rings. (Section 2.4.3)
- ❑ **MAC services.** Describes MAC frames on IEEE 802.5 rings. (Section 2.4.4)
- ❑ **LLC services.** Discusses LLC frames on IEEE 802.5 rings. (Section 2.4.5)

2.4.1 Network Management Considerations

As a network grows in size and complexity, it becomes necessary to determine the state of the network and to perform problem determination, isolation, and correction. A network management product is a LAN entity that specifies, watches, and modifies network configuration, and determines the state of, tests, and can remove (if faulty) individual stations. It collects error reports, logs errors, and performs fault isolation and correction based on these reports. A network management product can use all of its collected data to measure, analyze, and improve the network's reliability, efficiency, and throughput.

The TMS380 LAN COMMprocessor, in addition to providing services compatible with IEEE Standard 802.5 and IEEE 802.2 specifications, provides a number of services that support the existence of a management product in the network. These services are referred to as the network management *agent* within the adapter. Since these services are embedded within each adaptor node, networks can be expanded while earlier investments in attaching product hardware and software are preserved.

The token-ring COMMprocessor supports a network-management product in two ways:

- ❑ First, the COMMprocessor in the attaching station that implements the network-management function provides connectivity to the LAN. It recognizes information destined for, and passes the data to, the network-management product.
- ❑ Second, the TMS380 token-ring communications within attaching stations contain a logical agent for network management. Errors are automatically logged and reported; configuration changes are reported to the network manager. The TMS380C16 will respond to various requests from network management. All of these tasks are transparent to the attached station. Thus, the burden of participating in network management is removed from the attaching product. Standardizing these network-management services in silicon greatly reduces compatibility problems between different OEMs and allows end users to expand their networks and add network management without changing their installed base of equipment.

A network-management product can be divided into three logical entities:

- ❑ ring error monitor,
- ❑ network manager, and
- ❑ ring parameter server.

These logical entities communicate with the TMS380C16 via the class of frames called MAC frames. Since these functions need a well known address, a special type of address called a **functional address** is used to transmit the MAC frames to these functions. A functional address is also assigned to a bridge product and the active monitor station. Table 2–1 illustrates the functional addresses assigned to the network management logical entities.

Table 2–1. Network Management Functional Addresses

Address	Function
>01	Active monitor
>02	Ring parameter server
>08	Ring error monitor
>10	Network manager
>100	Bridge

The MAC frames that are transmitted to and received from network-management functions, as well as exchanged between adapters themselves, contain an embedded class designator as part of the basic syntax of the MAC frame. Each MAC frame contains one source class designator and one destination class designator. Each designator is four bits long, allowing up to 16 class designators. Table 2–2 lists the class designators, as defined by the network management agent within the TMS380.

Table 2–2. MAC Frame Class Designators

Class Designator	Definition
>0	Ring station class
>1	LLC manager class
>4	Network manager class
>5	Ring parameter server class
>6	Ring error monitor class

A class designator can be thought of as an embedded address that allows the network management product to route a received MAC frame to the appropriate software task, which then implements the function.

The ring station source class (>0) is used in MAC frames transmitted by the TMS380C16 software. The MAC frame transmission may be destined for a network-management function such as the ring error monitor or may be destined for another adapter to carry out ring protocols (for example, ring poll).

The ring station destination class (>0) is used in MAC frames transmitted to an adapter for processing by the adapter. The attached product will not be notified, in most cases, of the receipt of these frames. MAC frames received with a destination class of >0 include MAC frames from network-management functions requesting a specific action or report and MAC frames exchanged between adapters that support normal ring protocols.

MAC class >1 designates that the frame is passed to the LLC management entity.

Table 2–3 summarizes the combinations of source and destination class designators and their applications.

Table 2–3. Source and Destination Class Applications

Source Class	Destination Class	Application
>0	>0	Exchange of frames between adapters for basic ring protocols. – Ring poll – Ring purge – Monitor contention
>4	>0	Frames sent by the network manager to request specific action or request certain parameters from the adapter.
>0	>4	Frames sent to the network manager in response to request frames being received or to report configuration changes in the ring (new station or new active monitor).
>0	>6	Error reporting MAC frames sent to a ring error monitor.
>0	>5	A MAC frame that requests initialization parameters from a ring parameter server function.
>5	>0	MAC frames sent from the ring parameter server to the adapter, solicited or unsolicited by the adapter.

2.4.1.1 Ring Error Monitor (REM)

A ring error monitor serves as a collection point of error reports for network management. The REM does not have to be present on each ring. The network-management agent in TMS380-based adapters supports a REM function with the following:

1) Adapters within all attached stations count soft errors by the type of soft error that occurred (such as CRC errors, frame-copied errors, etc.), and automatically report these errors to the REM (via its functional address).

2) The adapter that is the active monitor reports failure of the ring poll process to the REM.

3) Errors in the active monitor, detected by the active monitor or a standby monitor, are reported to the REM.

When the information provided to the ring error monitor is used, conditions which degrade the performance of the network may be efficiently detected, diagnosed, and corrected.

2.4.1.2 Network Manager (NM)

The network manager function of the network-management product monitors and modifies the state of individual stations and that of the LAN as a whole. The network manager function is referred to by the IEEE 802.5 as the configuration report server.

The network management agent of the TMS380C16 provides NM support with the following functions:

Configuration changes Whenever a new active monitor is chosen via the monitor contention process, or whenever a station inserts or de-inserts from the ring, the event is reported to the network-management function. This allows the network-management function to maintain the configuration of the ring, including an ordered list of stations inserted at any moment.

Reporting services The TMS380C16 recognizes three requests for information from the NM and automatically reports the requested information in response.

Parameter control The TMS380C16 recognizes and responds to a command from the NM to modify internal operating parameters. This allows the NM to keep all adapters in step with network configuration changes, transparently to the attached product.

Reconfiguration control The TMS380C16 recognizes a remove command from the NM and will physically de-insert when it receives the command.

A network manager can use this support to perform its tracking and modification duties to maximize the LAN's reliability, efficiency, and overall performance.

2.4.1.3 Ring Parameter Server (RPS)

A ring parameter server is a logical function that can assign operating parameters of individual stations and of the LAN during the time the adapter is inserting into the ring. The TMS380C16 communicates with the RPS at the time of insertion by

- Requesting parameters in phase 4 of the insertion process.
- Setting these parameters on a response from the RPS.

2.4.2 IEEE 802.5 Frames

2.4.2.1 Frame Types

A message is passed through the network in a format referred to as a **frame**. The physical frame format is strictly defined by the token-ring architecture. The information carried by frames may be one of two types:

- ❑ Medium access control (MAC) information.
- ❑ Non-medium access control information (such as, protocol data units destined for the logical link control sublayer).

MAC Frames

The TMS380C16 provides a comprehensive set of problem determination, resolution, and reporting functions so that ring communication problems are rapidly diagnosed and automatically corrected. These functions are carried out through exchange of MAC frames.

Exchange of MAC frames is usually transparent to the attached system. These frames are originated by the TMS380C16 software in response to internal timeouts, error conditions on the ring, or MAC frames originated by network-server functions. MAC frames are transmitted without indication to the attached system. Furthermore, most MAC frames received are handled by the TMS380C16 itself, rather than reported to the attached system. Normally, an attached system will never need to send or receive a MAC frame. Processing the MAC layer protocol entirely within the TMS380C16 results in a lower workload on the attached system and an increase in system performance.

Higher-Layer Protocol Frames

Frames normally passed to and from the network by the attached product are used to implement the next layer of the communications protocol. By IEEE 802 conventions, the next layer would be the LLC. The LLC and MAC layers form the Data Link Control (DLC) layer of the ISO open systems interconnection reference model for a local area network communications system.

2.4.2.2 Frame Format

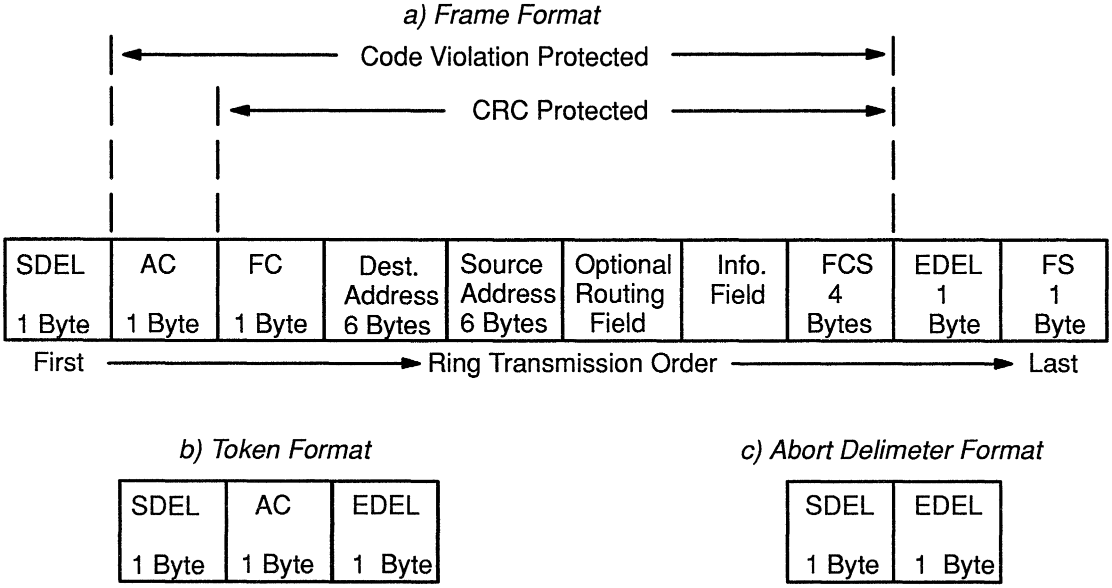
The physical frame format is strictly defined by the token-ring architecture and consists of the following fields within the frame's bit stream:

- ❑ Starting delimiter field (SDEL). See section 2.4.2.3.
- ❑ Physical control fields (PCF) consisting of the access control (AC) and frame control (FC) fields. See section 2.4.2.4.

- ❑ Source and destination address fields (SA/DA)
- ❑ Optional routing information field (RI)
- ❑ Information field (data)
- ❑ Frame check sequence field (FCS)
- ❑ Ending delimiter field (EDEL)
- ❑ Frame status field (FS)

Figure 2–9 shows the format of data on the ring. The token indicator bit is the third bit in the AC field. For frames, the token indicator bit is set to one. If the token indicator bit is set to zero, indicating a token, the token format illustrated in Figure 2–9 *(b)* is circulated on the ring. In this manual, the term *frame* refers to a data stream as illustrated in Figure 2–9 *(a)*, with the token indicator bit set to one. Also shown in Figure 2–9 *(c)*, is a structure called an abort delimiter. The abort delimiter consists of a starting delimiter and an ending delimiter sequence. The abort delimiter is transmitted by the TMS380C16 whenever the adapter detects a false free token.

Figure 2–9. Frame Format



a) Frame Format

SDEL	AC	FC	Dest. Address	Source Address	Optional Routing Field	Info. Field	FCS	EDEL	FS
1 Byte	1 Byte	1 Byte	6 Bytes	6 Bytes			4 Bytes	1 Byte	1 Byte

b) Token Format

SDEL	AC	EDEL
1 Byte	1 Byte	1 Byte

c) Abort Delimeter Format

SDEL	EDEL
1 Byte	1 Byte

2.4.2.3 Starting Delimiter (SDEL)

The SDEL is eight bits wide and employs a unique Manchester code violation, which indicates the start of the frame. Figure 2–10 defines the bits within the starting delimiter field. The figure is followed by a detailed description of these bits.

Figure 2–10. Starting Delimiter Bit Assignments

bit 0	1	2	3	4	5	6	7
V1	V0	0	V1	V0	0	0	0

bits 0–7 *Delimiter pattern.* These bits contain a delimiting pattern for the start of frames. It is through this pattern that the TMS380C16 synchronizes to the bit stream of the frame.

2.4.2.4 Physical Control Fields (PCF)

The physical control fields consist of two eight-bit fields: the access control and the frame control fields. They are used to manage the physical layer protocol.

Access Control Field (AC)

The bit assignments and bit definitions of the access control field byte are shown in Figure 2–11.

Figure 2–11. AC Bit Assignments

bit 0	1	2	3	4	5	6	7
Priority 0	Priority 1	Priority 2	Token Indicator	Monitor Count	Priority Res. 0	Priority Res. 1	Priority Res. 2

bits 0–2 *Token/frame priority.* These three bits indicate the priority of the token or frame on the ring.

bit 3 *Token indicator.* This bit is set to zero for tokens or set to one for frames.

bit 4 *Monitor count.* This bit is set by the active monitor when it repeats a frame or a token with a priority greater than zero. If the monitor receives an incoming AC with the bit already set, it assumes that the frame or token was not properly stripped and it purges the ring.

bits 5–7 *Priority reservation.* These bits indicate the priority reservation of the token and will be controlled by the TMS380C16. These bits are used by the priority control of the TMS380C16.

Frame Control Field (FC)

The bit assignments and bit definitions of the FC field byte are provided in Figure 2–12 below:

Figure 2–12. FC Field

bit 0	1	2	3	4	5	6	7
Frame Type 0	Frame Type 1	0	0	PCF Attn. 0	PCF Attn. 1	PCF Attn. 2	PCF Attn. 3

bits 0,1 *Frame type bits.* These two bits indicate the frame type:

00	MAC control frame
01	Non-MAC control frame
10	Reserved
11	Reserved

bits 2,3 *Reserved.* Bits 2 and 3 are always set to zero.

bits 4–7 *PCF attention code.* These bits indicate those frames that, upon reception, are copied into a special internal buffer called the **express buffer**. The values of the PCF attention code recognized by the TMS380C16 are listed below. The PCF attention code shown is for MAC frames only.

0001	Express buffer
0001	Remove ring station
0001	Duplicate address test
0010	Beacon
0011	Claim token
0100	Ring purge
0101	Active monitor present
0110	Standby monitor present

2.4.2.5 Source and Destination Address Fields

The source and destination address fields are each six bytes long and conform to the general address format described in Section 2.4.3.

2.4.2.6 Optional Routing Information Field (RI)

The routing information field is optional and is used only for frames that are routed between rings. This field must be included if bit 0 of the source address is set to one. See Section 2.4.3.2 for more details.

2.4.2.7 Information Field

The information field is a multi-byte field in which the data to be transported between ring stations is carried. The information field must be at least one byte in length. The maximum length of this field is on the order of 4,500 bytes on a 4-Mbps ring, and approximately 18,000 bytes on a 16-Mbps ring. This maximum frame size is set by the 10-millisecond token holding time.

2.4.2.8 Frame Check Sequence (FCS) Field

A 32-bit cyclic redundancy code (CRC) is appended to the frame to protect the frame control field, source and destination addresses, and the information field. The FCS field is generated from a polynomial and is accumulated serially as a frame is transmitted or received. When a frame is being repeated or copied by an adapter, the received FCS field is compared to the calculated value to verify that the frame was copied without error. On transmit, the FCS field is appended to the transmitted frame so that an identical comparison may be performed at the receiving station. Note that CRC generation and checking is done within the TMS380C16; the attached product does not perform this function. Following power-on, the TMS380C16 internal diagnostics check the CRC circuitry to insure proper operation.

2.4.2.9 Ending Delimiter Field (EDEL)

The ending delimiter defines the end of the frame sequence. Like the starting delimiter, it contains a code violation pattern to facilitate adapter synchronization to the bit stream. The bit assignments and bit functions are shown in Figure 2–13.

Figure 2–13. Ending Delimiter Bit Assignments

bit 0	1	2	3	4	5	6	7
V1	V0	1	V1	V0	1	Intermediate Frame	Error Detected Indicator

bits 0–5 Ending delimiting violation pattern.

bit 6 Intermediate frame. The PH always sets this bit to zero upon transmission. However, this bit may be a zero or one for reception.

bit 7 Errors detected indicator. This bit is transmitted as a zero. If received back as a zero, no errors were detected by other repeating adapters on the ring. If received back as a one, a CRC or code violation was detected by another adapter.

2.4.2.10 Frame Status Field (FS)

The frame status field indicates to the station that originated the frame, the results of the frame's circulation around the ring. This field provides two types of information:

- whether the frame's destination address was recognized, and
- whether the frame was copied by the station addressed in the destination address.

Because this field is not error-checked by the CRC or code violation detection mechanism, these bits are duplicated. Figure 2–14 shows the bit assignments and bit functions of the FS. If an adapter transmits a frame to itself, neither the ARI or FCI bits will be set when the received frame is passed to the attached system.

Figure 2–14. FS Bit Assignments

bit 0	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Address Recognized Indicator	Frame Copied Indicator	0	0	Address Recognized Indicator	Frame Copied Indicator	0	0

bits 0,4 — Address Recognized Indicator. This bit is initially transmitted as a zero. If it is received back as a zero, the frame destination address was not recognized, or a code violation or CRC error was detected by an adapter on the ring. If it is received back as a one, the destination address was recognized properly.

bits 1,5 — Frame Copied Indicator. This bit is initially transmitted as a zero. If it is received back as a zero, the frame was not copied by the adapter addressed by the destination address. If it is received back as a one, the frame was copied by the destination adapter.

bits 2,3, 6,7 — Reserved. These bits are set to zero on transmission.

2.4.3 Ring Addressing

This section describes the format of the source and destination address fields and the routing information fields.

The destination and source address fields are each six bytes in length and conform to the address format shown in Figure 2–15 and Figure 2–16, respectively.

Figure 2–15. Destination Address Format

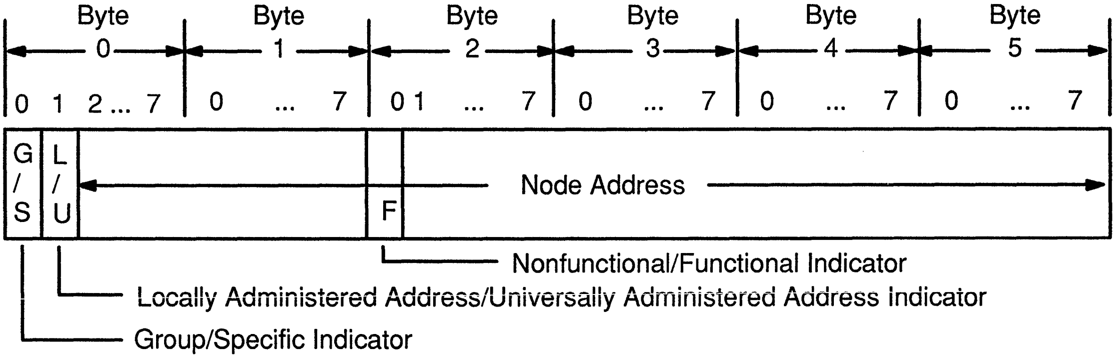
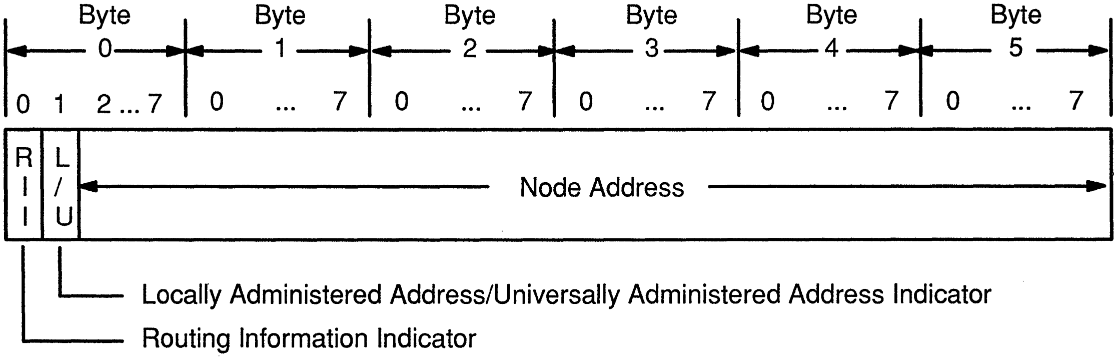


Figure 2–16. Source Address Format



The source and destination addresses differ in use of indicator bits within the frame format. The group/specific and functional/nonfunctional indicators are used for destination addresses only. All source addresses will be ring-station-specific addresses. Because all source addresses of transmitted frames will be the ring-station-specific address of the transmitting node, the most significant bit position of the source address is used to indicate the presence of a routing information field. The routing information field is a field of up to 18 bytes which follows the source address and precedes the information field of a frame.

2.4.3.1 Address Types

All addresses in the network are either locally administered addresses (byte 0, bit 1 set to one) or universally administered addresses (byte 0, bit 1 set to zero). Whether an address is locally or universally administered depends on whether address assignment is done locally (within an establishment) or administered by a standards organization for all establishments.

There are two major types of destination address: group node and ring-station-specific address.

- ❑ **Group node addresses**. A group node address can be recognized by more than one station on a ring. A group address is signified by setting bit 0 of byte 0 to one. The high-order two bytes of group addresses are assumed to be >C000 except for group broadcast address. There are three types of group node addresses: broadcast, functional, and group.

  - **Broadcast**. >FFFF FFFF FFFF or >C000 FFFF FFFF are both all stations broadcast addresses. A broadcast address is shown in Figure 2–17.

  - **Functional**. A functional address is indicated by setting bit 0 of byte 2 to zero. Bit significant decoding allows a node to recognize any or all of 31 addresses assigned by the token-ring architecture to functions in the network that need a well-known address. A functional address recognized by a node can be assigned by the attached system during the OPEN command or the SET FUNCTIONAL ADDRESS Command. See Figure 2–18.

    Several functional addresses have been defined by the IBM token-ring architecture as follows:

>0001	Active monitor
>0002	Ring parameter server
>0008	Ring error monitor
>0010	Network manager
>0080	Netbios
>0100	Bridge

  - **Group**. A group address is assigned at the discretion of the attached system by setting bit 0 of byte 2 to one. The address may be passed as one of the OPEN parameters. See Figure 2–19.

Figure 2–17. Broadcast Address

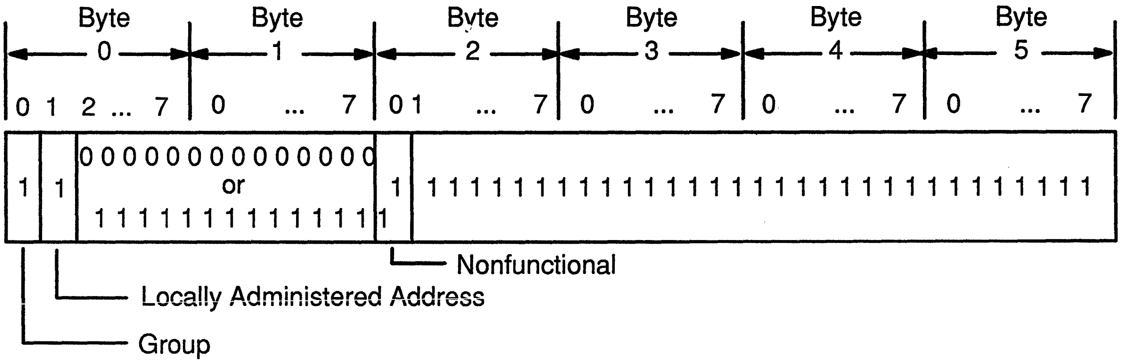


Figure 2–18. Functional Address

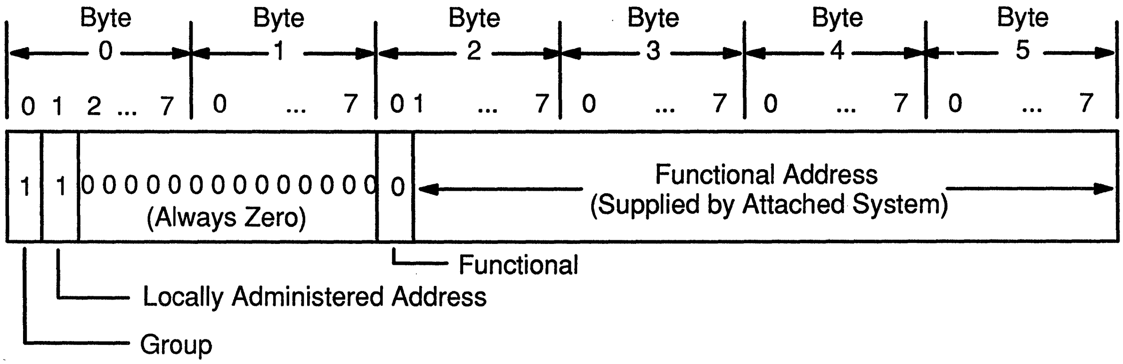
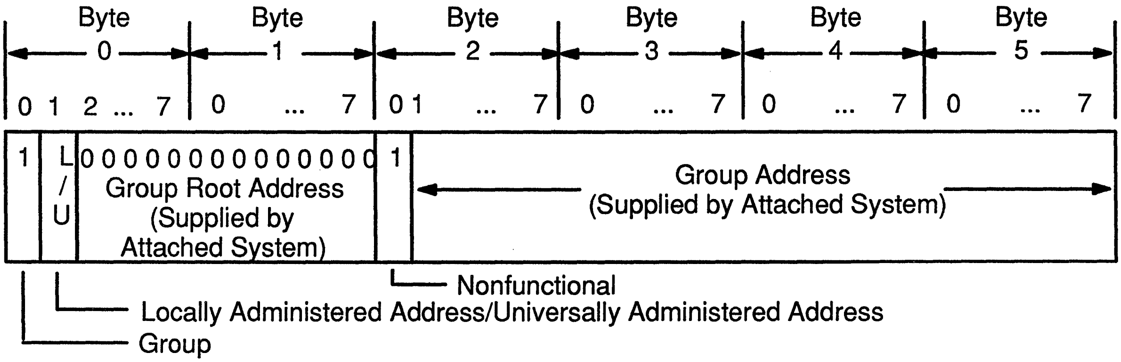


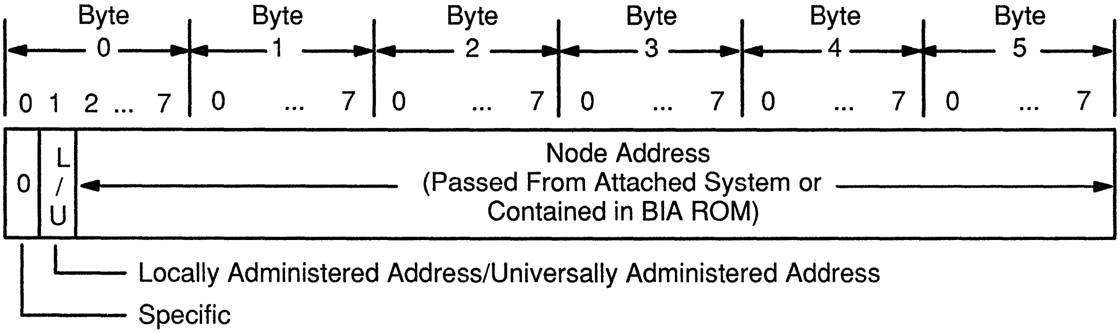
Figure 2–19. Group Address



□ **Ring-station-specific address**. This is the primary address recognized by the adapter, and is the address associated with the physical transmission layer function. This address may be passed to the adapter by the attached system during the OPEN command, or may be read from an attached ROM containing a burned-in address. If the attached system passes an all-zeros ring-station-specific (or node) address during the OPEN command, the adapter will use the burned-in address (BIA) supplied. This burned-in address allows address assignments to be integrated on the adapter's card in a nonvolatile (ROM) manner.

A ring-station address assigned by the attached system or contained in the BIA ROM must conform to the format shown in Figure 2–20. Note that bit 0 of byte 0 must be set to zero. Bit 1 of byte 0 may be set to either 0 or 1. Violating these conditions causes the adapter to reject the address assignment with a node address error.

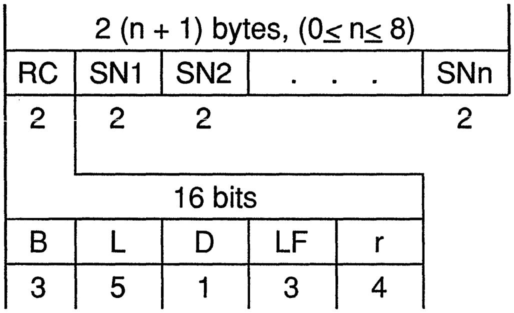
Figure 2–20. Ring-Station Address Format


Byte 0
Byte 1
Byte 2
Byte 3
Byte 4
Byte 5
0 1 2 ... 7
0 ... 7
0 ... 7
0 ... 7
0 ... 7
0 ... 7
0
L / U
Node Address (Passed From Attached System or Contained in BIA ROM)
Locally Administered Address/Universally Administered Address
Specific

2.4.3.2 Source Routing Field

If the MSB of the source address is set to one, a routing information field will follow the source address field in the frame. The format of the routing information field is shown in Figure 2–21.

Figure 2–21. Routing Information Field Format



SN = Segment number
RC = Routing control

B:	Broadcast
L:	Length
D:	Direction
LF:	Longest Frame
r:	Reserved

B (broadcast bits). These bits indicate whether the frame is to be sent along a specified path, to all rings in a network along all possible paths, or to all rings along designated bridges, which results in only one copy on a given ring.

The coding for these bits is as follows:

000 *Nonbroadcast.* If the bits are set for nonbroadcast, the segment number fields contain the specified route through the network the frame is to travel.

100 *All-routes broadcast, nonbroadcast return.* If the bits are set for all-routes broadcast, the frame will be transmitted on every route within the network resulting in multiple copies on a given ring.

110 *Single-route broadcast, all routes broadcast return.* If the bits are set for single-route broadcast with all routes broadcast return, the frame will be transmitted across the designated bridges, which will result in the frame appearing only once on each ring. The response frame is on all routes broadcast to the originator.

111 *Single-route broadcast, nonbroadcast return.* If the bits are set for single-route broadcast with nonbroadcast return, the frame will be transmitted across designated bridges, which will result in the frame appearing only once each ring.

L (length). This field indicates the length of the routing information field in bytes, including the control field.

D (direction). This indicates to a bridge whether a frame is traveling from the originating station to the target or the other way around. This bit allows the ring number segments to appear in the same order, regardless of the direction of transmission. The initial value of this bit in broadcast frames must be zero.

LF (longest frame). The LF bits specify the largest size of the MAC information field that may be transmitted between two communicating stations on a specific route. The coding of the LF indicates the following:

000 Up to 516 bytes in the information field.

001 Up to 1470 bytes in the information field.

010 Up to 2052 bytes in the information field.

011 Up to 4472 bytes in the information field.

100 Up to 8144 bytes in the information field.

101 Up to 11,407 bytes in the information field.

110 Up to 17,800 bytes in the information field.

111 Initial value of broadcast frames.

r (reserved). These bits are reserved.

SNx (segment numbers). These 16-bit fields indicate the path between nodes on different physical rings, and have the construct shown in Figure 2–22.

The ring number portion of the segment number field is different for each ring in the network. Thus, any two bridges connecting to the same ring will have identical RN portions.

The individual bridge number of the segment number field allows two bridges connecting the same two rings to provide alternative paths between the two rings.

Figure 2–22. Segment Number Field

16 bits	
RN	BN
16 – K Bits	< KBits>

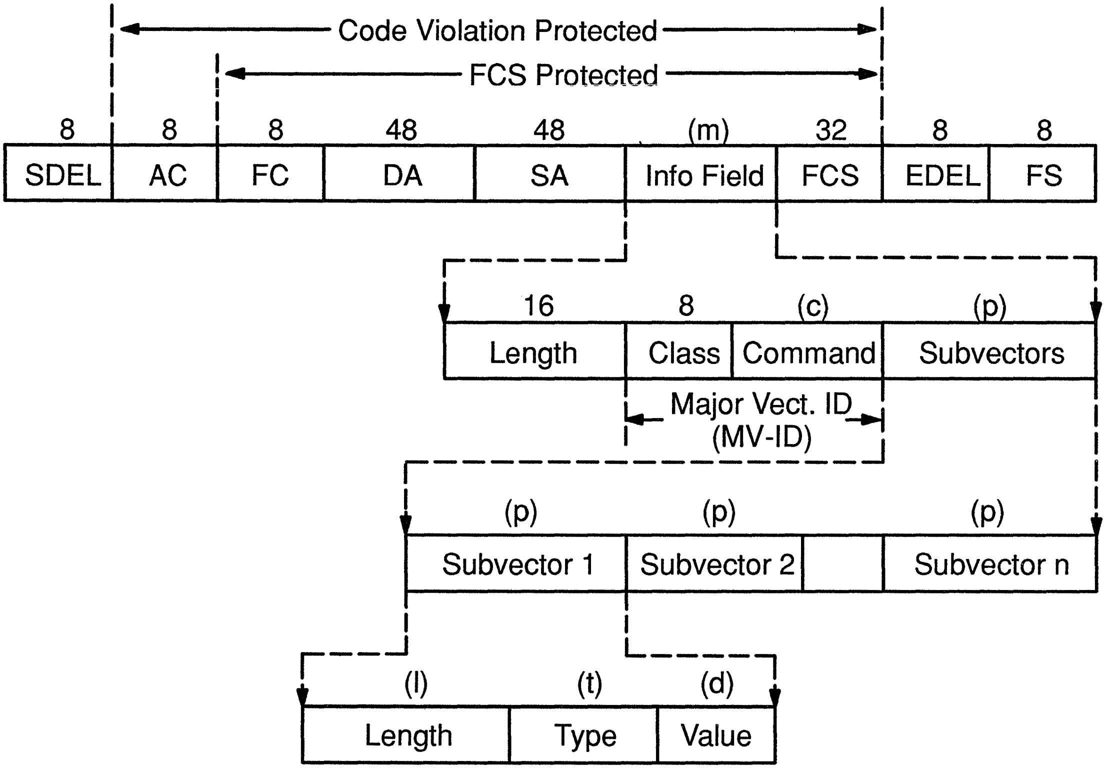
where: RN = Ring Number
BN = Bridge Number
$1 \leq K \leq 14$

2.4.4 MAC Services

2.4.4.1 IEEE 802.5 MAC Frame Format

Figure 2–23 shows the format of frames used as MAC frames.

Figure 2–23. General MAC Frame Format



Notes: 1) Numeric length notations are in bits.

2) Maximum length of (m) is restricted by the adapter.

3) Entries labeled (m), (p), and (d) contain an integral number of bytes.

4) The entries labeled (c), (l), and (t) are 8 bits each. If their value is equal to >FF, they are extended by two bytes.

2.4.4.2 MAC Frame FC

The MAC bit (bit 1) of the FC field on MAC frames is set to zero to indicate that the frame is a MAC frame. Bits 4–7 of FC compose the PCF attention code. A binary value greater than 0001 in any frame that the TMS380C16 copies or repeats causes an attention interrupt. These bits are assigned as follows:

0010	Beacon
0011	Claim token
0100	Ring purge
0101	Active monitor present
0110	Standby monitor present

2.4.4.3 *MAC Frames Source Address*

The source address field always contains the ring-station-specific address of the originator of the frame.

2.4.4.4 *MAC Frame Major Vectors*

The information field in MAC frames handled by the adapter consists of one major vector (MV). The major vector may contain one or more subvectors. When the adapter constructs a MAC frame for transmission, the subvectors are placed in the frame, in the order shown in the tables in Appendix C.

The major vector is made up of the following fields within the MAC frame information field:

MV length — Gives the length in bytes of the entire major vector, including the length field.

MV class — Defines the origin and destination class of the MV. The high-order four bits are the destination class and the low-order four bits are the source class.

The source class types are as follows:

>0	Ring station
>1	LLC manager
>4	Network manager
>5	Ring parameter server
>6	Ring error monitor

The destination class will always be ring station, 0, for MAC frames that are processed directly by the TMS380C16. Other destination classes will be passed on to the attached system if they are copied.

MV command — The major vector command defines the function that the receiver is to perform. A list of major vector commands is shown in Table 2–4.

Subvectors — Each major vector contains one or more subvectors. A subvector contains the following fields:

SV length — Gives the length of the subvector in bytes including the length field.

SV type Identifies the information found in the subvector value. The SV types recognized by the adapter are shown in Table 2–5.

SV value Contains the information used to process the subvector.

Table 2–4. Major Vector Commands

Command	Description	Command	Description
>00	Response	>03	Claim Token
>02	Beacon	>05	Active Monitor Present
>04	Ring Purge	>07	Duplicate Address Test
>06	Standby Monitor Present	>09	Transmit Forward
>08	Lobe Media Test	>0B	Remove Ring Station
>0C	Change Parameters	>0D	Initialize Ring Station
>0E	Request Station Address	>0F	Request Station State
>10	Request Station Attachment	>2A	Report Transmit Forward
>20	Request Initialization	>22	Report Station Address
>23	Report Station State	>25	Report New Monitor
>24	Report Station Attachment	>27	Report Ring Poll Failure
>26	Report SUA Change	>29	Report Error
>28	Report Monitor Error		

Table 2–5. Subvector Types

Type	Description	Type	Description
>01	Beacon type	>02	Upstream neighbor's address
>03	Local ring number	>04	Assign physical drop number
>05	Soft error report timer	>06	Enabled function classes
>07	Allowed access priority	>0A	Address of last ring poll
>20	Response code	>21	Reserved
>22	Product instance ID	>23	Adapter software level
>26	Wrap data	>0B	Physical drop number
>29	Adapter status word	>27	Frame forward
>2B	Group address	>09	Correlator
>2D	Isolation error counts	>2C	Functional address
>2F	Function request ID	>2E	Non-isolating error counts
>08	Authorized environment	>30	Error code
		>2A	Transmit status code

The major vector class and major vector command together are termed the major vector ID, MV-ID. A complete list of major vector commands and subvectors may be found in Appendix C.

2.4.4.5 IEEE 802.5 MAC Frames Processed by the TMS380

MAC frames processed automatically by the TMS380C16 are those received with a destination class of ring station. If a response to a ring station class MAC frame is required, the response is transmitted with a source class of ring station.

Some frames processed by the TMS380C16 are, in addition, passed to the attached system. Three options specified in the open command of the system software (see Chapter 4) allow MAC frames to be passed to the attached system after the adapter has been inserted onto the ring:

- **Pass attention MAC frames.** MAC frames that have an attention code (bits 4–7 of FC) greater than one are processed normally by the TMS380C16 and then passed to the attached system when one of the following conditions occurs:
  - When the attention code is different from the last attention code greater than one.
  - When the source address in the MAC frame is different from the last source address received.

  - If it is a beacon MAC frame, when the beacon type subvector value is different from the beacon type subvector value in the last beacon MAC frame.

- ❑ **Pass adapter MAC frames.** MAC frames are passed which have a major vector type not recognized by the TMS380C16. These will be passed to the attached system if this option is enabled. Otherwise, the TMS380C16 responds by transmitting a negative response MAC frame.

Table 2–6 describes briefly the MAC frames processed directly by theTMS380C16.

Table 2–6. MAC Frames Processed by the TMS380 Adapter

MAC Frame	Description
Active monitor present	This frame is transmitted by the active monitor when in the ring poll process to request a standby monitor present MAC frame from the nearest downstream neighbor from the active monitor.
Beacon	This frame is used by the TMS380C16 in the beacon process.
Change parameters	Change parameters MAC frame is sent by the network manager to change certain parameters in the adapter. This frame will be accepted in response to a request initialization MAC frame transmitted by an adapter in the insertion process. When received, this frame can set the following parameters: – Local ring number – Physical drop number – Error report timer value – Authorized transmit function classes – Authorized access priority
Claim token	This frame is used by the TMS380C16 in the monitor contention process.
Duplicate address test	This frame is sent by the TMS380C16 in phase 2 of the insertion process. It verifies that the specific address to be used by the adapter is unique to the ring in which the adapter is inserted.
Initialize ring station	Initialize ring station MAC frames are sent by the ring parameter server to set parameters in the adapter when the adapter is in the insertion process and transmits a request initialization MAC frame. When received, this frame can set the following parameters within the TMS380: – Local ring number – Physical drop number – Error report timer value
Lobe media test	This frame is used by the TMS380C16 in phase 0 of the insertion process to test the continuity of the wire in a loop back path. This occurs prior to physical insertion in the ring. It also occurs in the auto remove procedures in the beacon process, which use the insertion phase 0 process. This frame is ignored by the TMS380C16 when it is received.

Table 2–6. MAC Frames Processed by the TMS380 Adapter (Continued)

MAC Frame	Description
Remove ring station	This frame is sent by the network manager to request the TMS380C16 to de-insert itself from the ring. This frame is valid any time after phase 0 of the insertion process. If the adapter is inserted, the adapter de-inserts from the ring and indicates remove received in the ring status reported to the attached system. If this frame is received in the insertion process, the adapter removes itself from the ring and terminates the OPEN command with an error condition, indicating that a remove MAC frame was received.
Report error	This frame is used to report soft error events to the ring error monitor.
Report monitor error	This frame is used to report a problem with the active monitor or the possibility of a duplicate address of stations contending for active monitor. This frame is sent to the functional address of the ring error monitor (REM).
Report new monitor	This frame is sent by the active monitor adapter, after winning contention, to report to the network manager that the adapter is now the new action monitor.
Report ring poll failure	This frame is sent by the active monitor to the ring error monitor to report a failure in the ring poll process. This frame contains the address of the last station that responded in the ring poll process before the active monitor detected the failure.
Report station address	This frame is sent by the adapter in response to the request station address MAC frame. The request station address MAC frame is sent by the network manager.
Report station attachment	This frame is sent by the TMS380C16 in response to the request station attachment MAC frame. The request station attachment MAC frame is sent by the network manager.
Report station state	This frame is sent by the TMS380C16 in response to the request station state MAC frame (sent to the adapter by the network manager).
Report SUA change	This frame is used in the ring poll process to report a change in the stored upstream address (SUA) of the station that is upstream from the adapter generating the report SUA change MAC frame. This frame is sent to the functional address of the network manager.

Table 2–6. MAC Frames Processed by the TMS380 Adapter (Concluded)

MAC Frame	Description
Report transmit forward	This frame is sent by the adapter to the network manager functional address when a frame is forwarded and stripped by the transmit forward process.
Request initialization	This frame is transmitted in phase 4 of the insertion process to request operational parameters from the ring parameter server.
Request station address	This frame is sent by the network manager to request the adapter to respond with a report station address MAC frame.
Request station attachment	This frame is sent by the network manager to request the adapter to respond with a report station attachment MAC frame.
Request station state	This frame is sent by the network manager to request the adapter to respond with a report station state MAC frame.
Response	This frame is used to send positive responses to frames that require acknowledgement, or to report errors in syntax in MAC frame sent to the adapter.
Ring purge	This frame is used by the active monitor in the ring purge process.
Standby monitor present	This frame is used in the ring poll process to respond to an active monitor present or standby monitor present MAC frame.
Transmit forward	This frame is used in the transmit forward process.

2.4.4.6 MAC Frame Receive and Transmit Processing

MAC frame receive syntax checking and transmit processing are described below.

2.4.4.6.1 MAC Frame Receive Processing

General MAC Frame Syntax Checking

The MAC frame syntax checking procedure requires that

❑ There is only one major vector in the received frame,

❑ The major vector length agrees with the received frame length, and

❑ No required subvectors are duplicated or missing in the frame.

Only the first occurrence of an optional subvector is processed by the TMS380C16 software. Any optional subvector that has already been found in the frame is ignored. The following describes the MAC frame syntax checking procedure:

1) The routing information field is checked to determine if the length is longer than 18 bytes or odd. If the length is greater than 18 or odd, the frame is ignored.

2) The length of the frame is checked to determine if it can contain a major vector length and ID. If the length of the frame is too short (less than four bytes), a negative response of MAC Frame Data Field Incomplete is sent. Note that the source class, destination class, and the major vector command will be invalid in the response because the received frame was too short to contain them.

3) If the destination class is not zero, the frame is passed to the attached system, with no further syntax checking.

4) The command byte is checked to see if it is a transmit forward. If it is, no further syntax checking is done by this syntax checking routine.

5) The major vector length is checked to determine if it is in the range of the received frame. If the length in the major vector ID does not agree with the length of the frame, a negative response of Major Vector Length Invalid is sent.

6) The command byte in the major vector ID is checked to determine if it is greater than > 10, the highest major vector ID handled by the adapter. If the adapter has been opened without the open option to pass adapter MAC frames, a negative response is sent with a code of Major Vector Command Not Supported. If the open option is selected, the frame is passed to the attached system without further syntax checking.

7) If the source class in the class byte in the major vector ID is not valid for the command in the major vector ID, a negative response of Inappropriate source class is sent.

8) The subvectors are checked as follows:

   - If a subvector length is zero, then a negative response of Subvector Length Error is transmitted to the sender.
   - All required subvectors must be present. A response of Required Subvector Missing is transmitted if this is not the case.
   - If a subvector in the frame is marked "required" and is not a required subvector or is duplicated, a response of Unknown Required Subvector is sent.
   - Optional subvectors are ignored.

2.4.4.6.2 MAC Frame Transmit Processing

The following is a description of the queueing process the TMS380C16 uses for transmitting MAC frames.

All MAC frames make a token reservation at priority level 3 with the following exceptions:

- ❑ Immediate MAC frames (those that don't wait to capture a token) set the priority bits to 0. These are as follows:
  - ■ Beacon
  - ■ Claim token
  - ■ Ring purge
- ❑ The active monitor present MAC frame makes a token reservation at level 7.
- ❑ All response-type MAC frames are transmitted at priority level 0.

When a MAC frame is queued for transmission, the following sequence is followed.

1) Some MAC frames use a specially allocated transmit buffer. These frames are transmitted as soon as the buffer becomes available. These frames are:

   a) Report error
   b) Report monitor error
   c) Report ring poll failure
   d) Report new monitor
   e) Active monitor present
   f) Duplicate address test
   g) Report SUA change
   h) Standby monitor present
   i) Request initialization

2) Response-type MAC frames are built within the same buffer in which the associated request was received. Frames that transmit in this manner are as follows:

   a) Response
   b) Transmit forward
   c) Report transmit forward
   d) Report station address
   e) Report station state
   f) Report station attachment

2.4.4.6.3 Assured Delivery Process

The assured delivery process is intended to fulfill the following requirements:

- ❑ To significantly improve the probability of, but not guarantee, the delivery of selected MAC frames.
- ❑ To offer a high probability of preventing duplicate copies of frames delivered to the frame's destination address.

The MAC frames that use this process are the report error MAC frame and the report SUA change frame.

This procedure uses the address recognized indicator bit (ARI) and the frame copied indicator (FCI) bit of the frame status (FS) field of the frame. An internal status bit (LFED) indicates if frame errors occurred during the stripping of a transmit frame from the ring, such as code violations and frame check sequence (FCS) errors.

The assured delivery process is defined by the following steps:

1) When a MAC frame is queued for transmission, a counter called the transmit retry counter is initialized to four.
2) After a frame is transmitted and stripped, internal status is checked to determine if a corrupted free token was detected. If this did occur, proceed to step 4. If any other transmit error is detected, then the process is terminated. If the internal status indicates normal transmitter completion of the transmit operation, continue with step 3.
3) If a MAC frame must be transmitted under the assured delivery process, then the ARI, FCI, and LFED bits are checked. The table below shows the three conditions which must result in order to continue to step 4. With any other combination, the frame is not re-transmitted and the process terminated.

	LFED	ARI1	FCI1	ARI2	FCI2
a.	0	1	0	1	0
b.	1	0	0	0	0
c.	1	1	0	1	0

4) The transmit retry counter is decremented, and, if the counter is not zero, the frame is retransmitted and the process continues with step 2. If the counter is zero, the frame is not transmitted and the process is terminated.

2.4.4.7 Response MAC Frames

A response MAC frame is used to acknowledge receipt of one of the following MAC frames:

□ Change parameters
□ Initialize ring station

This frame is also used in negative responses to syntax errors detected in any adapter destination class MAC frame. The TMS380C16 will not generate negative responses to MAC frames that have a source class of >0.

A response code subvector within the response MAC frame carries additional response information. This response code subvector has the following format.

Figure 2–24. Response Code Subvector

Byte 0	1	2	3
Response Code		SRC/DES Class	M-V Command

The source class, destination class, and major vector command are generated from the contents of the major vector in the received MAC frame that caused the TMS380C16 to send the response MAC frame.

The response codes are listed in Table 2–7.

Table 2–7. Response Code Subvector Code Values

Code	Description
>0001	Positive response (ACK). Sent in response to MAC frames requiring positive acknowledgement of receipt.
>8001	MAC frame data field Incomplete. The MAC frame was too short to contain the major vector Length and major vector ID (less than 4 bytes).
>8002	The major vector Length did not agree with the length of the frame, or a subvector was found that did not fit within the major vector.
>8003	Major vector command not supported. The major vector ID was not recognized by the adapter.
>8004	Inappropriate source class. The source class in the major vector ID is not valid for the major vector.
>8005	Subvector length invalid. The length of a recognized subvector is longer than the maximum allowed.
>8006	Transmit forward frame error. An error is detected in the received transmit forward MAC frame. The frame is not forwarded.
>8007	Required subvector missing. A subvector required by the adapter is not in the frame.
>8008	Required subvector unknown. A subvector received in the MAC frame that is marked required is not known by the TMS380C16. This response is also generated if a required subvector is duplicated in the major vector.

Table 2–7. Response Code Subvector Code Values (Continued)

Code	Description
>8009	MAC frame exceeds maximum length. The received frame is rejected because it did not fit in one buffer.
>800A	Function disabled. The received MAC frame is not executed because the function requested is disabled.

2.4.4.8 Monitor Functions

An adapter on the ring can be either an active monitor or a standby monitor. Only one active monitor is present on any ring; the remaining adapters serve as standby monitors. The active monitor ensures normal token operation on the ring and provides the crystal-controlled master data clocking for data transmission. The standby monitors ensure that the active monitor is functioning properly and is still inserted on the ring.

The ring purge process is used by the active monitor to perform the recovery from a temporary error condition, to release a new token, and to return the ring to a known state.

The ring poll process is initiated periodically by the active monitor to update the upstream neighbor's address (UNA) in all adapters in the ring and to provide an indication to other stations in the ring that the active monitor is still active.

2.4.4.8.1 Active Monitor Functions

The role of the active monitor is to ensure normal token operation on the ring. An adapter becomes the active monitor by active participation in the monitor contention process. The monitor contention process is discussed in Section 2.4.4.9.2. The adapter that wins contention becomes the active monitor.

Upon successfully winning contention, the active monitor does the following:

- ❑ Sets a bit that
  - ■ Provides master clocking for data transmission.
  - ■ Inserts a 30-bit time latency to guarantee a ring length, which assures that a token can be circulated properly.
  - ■ Activates the circulating token removal hardware. The circulating token removal function operates as follows:
    1) When tokens are released on the ring, bit 4 of the AC byte (the MC bit) is transmitted as zero.

2) When a TMS380C16 changes the token into a frame, it leaves the MC bit as zero. It indicates a frame by setting only bit 3 of AC (the TI bit) to one.

3) When a frame or a token of priority greater than zero passes through the active monitor (which is in repeat mode), the monitor sets the MC bit.

4) The transmitting adapter *should* strip its frame or token off the ring and release a free token with the MC bit equal to zero. If it does not, the frame or priority token passes into the active monitor for the second time.

5) When a frame or priority token comes into the monitor with its MC bit equal to one, the monitor hardware does **not** repeat it back onto the ring. The active monitor then increments its token error counter and purges the ring.

❑ Executes the ring purge process.

❑ Starts the ring poll process by activating an internal pacing timer and queues an active monitor present (AMP) MAC frame for transmission.

❑ Transmits a free token of priority equal to the token reservation priority in the ring purge MAC frame last stripped by the adapter.

❑ Sets the monitor functional address.

❑ Activates a checking function that confirms that a good token is detected on the ring every 10 ms. This timer sets the maximum frame size on the order of 4048 bytes for a 4-Mbps ring, and 18,000 bytes for a 16-Mbps ring.

❑ Queues a report new monitor MAC frame for transmission to the network manager.

When an event occurs that requires an action by the active monitor, the active monitor does the following:

❑ When the active monitor detects a token error (either a circulating frame or priority token) or fails to detect a good token during a 10-ms period, the active monitor starts the ring purge process to recover the ring back to normal token protocols.

❑ When an internal 7-second timer expires, the active monitor starts the ring poll process.

2.4.4.8.2 *Active Monitor Exception Conditions*

The following exception conditions will cause the station to deactivate the active monitor function and take the corresponding action.

Receive ring purge If the active monitor receives a ring purge MAC frame from an address not its own, the TMS380C16 will reset the active monitor and queue a report monitor error MAC frame for transmission with an error code subvector indicating that a duplicate monitor has been detected. The TMS380C16 will start the standby monitor functions.

Receive AMP If the active monitor receives an active monitor present (AMP) MAC frame that it did not transmit, the TMS380C16 will reset the active monitor function and queue a report monitor error MAC frame with an error code subvector value indicating that a duplicate monitor has been detected. The adapter will start the standby monitor functions.

Receive claim token The TMS380C16 will queue a report monitor error MAC frame for transmission with an error code subvector indicating that a standby monitor detected an error in the active monitor, and the monitor contention process is entered in contention repeat mode. This adapter will not actively participate in the contention process by entering contention repeat mode.

Receive beacon If the adapter is inserted, the adapter will enter the beacon process in beacon repeat mode. If the adapter is in the insertion process, the OPEN command is terminated with an error.

Signal loss The adapter will enter the monitor contention process in contention transmit mode.

Wire fault If a wire fault condition is detected, the adapter deinserts from the ring and sets the lobe wire fault bit in ring status.

2.4.4.8.3 Standby Monitor Functions

Any adapter that is not the active monitor but has completed the insertion process will follow the procedures of the standby monitor. The function of the standby monitor is to monitor the events on the ring to determine if the active monitor is functioning properly.

The standby monitor functions are disabled while the adapter is in the insertion, beacon, or monitor contention processes.

A station activates its standby monitor functions as follows:

- ❑ When the adapter completes the ring insertion process and does not get contention in phase 1 of the ring insertion process.
- ❑ When the adapter receives a ring purge MAC frame while it is in contention repeat mode.
- ❑ When an inserted active monitor determines that another adapter has assumed the functions of active monitor.

As a standby monitor, the station transmits on the ring with the clock derived from the incoming signal. It also deactivates the hardware mechanism for correcting circulating priority tokens and frames.

The following ring conditions are monitored by the standby monitor:

- ❑ **Good token.** The station verifies that a good token is received at least once every 2.6 seconds. A good token is defined as a token of priority zero, or a token of priority greater than zero followed by a frame with a priority field greater than zero.
- ❑ **Periodic ring polls.** The adapter receive poll timer detects the absence of active monitor present (AMP) MAC frames. The receive poll timer is restarted when an AMP MAC frame is received. The absence of AMP MAC frames indicates there is no active monitor in the ring. If the receive poll timer expires after 15 seconds, the TMS380C16 enters the monitor contention process in the contention transmit mode.
- ❑ **Proper ring data frequency**. The TMS380C16 uses the hardware error process to check the frequency of the data on the ring. If a frequency error is detected by this process, it indicates that there is no active monitor or that the active monitor is not functioning properly. If a frequency error is detected, the TMS380C16 enters the monitor contention process in contention transmit mode.

2.4.4.8.4 Standby Monitor Exception Conditions

The following exception conditions cause the TMS380C16 to deactivate the standby monitor functions and take these actions.

Active monitor errors If the TMS380C16 detects an error in the active monitor, the adapter enters the monitor contention process in contention transmit mode.

Claim token frame If the adapter receives a claim token MAC frame, the adapter enters the monitor contention process.

Receive beacon The receiving station enters the beacon process in beacon repeat mode.

Signal loss

If a signal loss is detected by the adapter, it enters the monitor contention process in contention transmit mode.

Wire fault

If a wire fault condition is detected by the TMS380C16, the adapter de-inserts from the ring and sets the lobe wire fault bit in ring status.

2.4.4.9 MAC Processes

The token ring adapter medium access control (MAC) protocols are implemented primarily by TMS380C16 software. These protocol processes are described below:

Table 2–8. MAC Protocol Processes

MAC Frame	Process
Beacon	The beacon process is used to recover the ring when any attaching ring station has sensed that the ring is inoperable due to a hard error. If necessary, the adapter withdraws itself from the ring.
Hardware error	The hardware error process is used to detect when a wire fault, frequency error, or ring signal loss has occurred.
Monitor contention	The monitor contention process involves all adapters attached on the ring. This process establishes a ring station as the active monitor. The active monitor ensures normal operation of the ring
Ring insertion	The ring insertion process is performed by the adapter to insert the adapter into the ring. It is performed after bring-up diagnostics and initialization have been completed. The ring insertion process is initiated with an OPEN command to the adapter, issued by the attached system.
Ring poll	The ring poll process enables any adapter inserted into the ring to acquire the address of its upstream neighbor (UNA). The ring poll is initiated periodically by the active monitor to ascertain the ordered list of attached stations. It is also used to determine if a single station ring exists. This process is referred to as neighbor notification in IEEE Standard 802.5.
Ring purge	The ring purge process, initiated by the active monitor, is used to put the ring into a normal condition, which allows frames to be transmitted using token protocol.

Table 2–8. MAC Protocol Processes (Continued)

MAC Frame	Process
Soft error counting	This process logs and reports an error condition that temporarily degrades system performance, but can be tolerated if error recovery procedures of higher protocols are used.
Transmit forward	The transmit forward process allows multiple layers of frames to be sent by network management to test a path between two attached ring stations.

2.4.4.9.1 Notes on Reading the MAC Frame Tables

When each process is discussed, the MAC frames which are exchanged by the execution of that process are also provided in tabular format. These tables contain the following columns:

M-V name This is the name of the MAC frame. Also provided is the hexadecimal value of the major vector (M-V) command. Hexadecimal numbers are preceded by a > sign.

Destination class The destination class is the four-bit destination class field of the MAC frame major vector ID as previously described.

Source class The source class is the four-bit source class field of the MAC frame major vector ID. The source class value ranges from 0 through F. The source classes referenced in these tables are as follows:

0	RS	Ring station (adapter)
4	NM	Network manager
5	RPS	Ring parameter server
6	REM	Ring error monitor

Destination address This is the destination address field of the MAC frame. The terminology used in this field is described below:

all sta *All stations.* This indicates a group broadcast address allowing all adapters on the local ring to copy the frame.

F(fa) *Functional address.* This indicates that a destination address is the functional address of the “fa” function (such as network manager, ring parameter server, etc.). The functional addresses are as follow:

RPS	Ring parameter server	>0002
REM	Ring error monitor	>0008
NM	Network manager	>0010

MA *My address.* This adapter's specific address.

SA *Source address.*

Target A six-byte destination address.

Subvectors

The subvectors that are included in the MAC frame are listed in this column, along with the hexadecimal subvector type value is also provided. Two columns are used to indicate additional information.

XMT This subcolumn contains a letter indicating whether a subvector is always transmitted by the sending adapter or copied from another frame. This subcolumn is with respect to the sending adapter.

RCV This subcolumn contains a letter indicating whether a subvector is required, is optional, or is not syntax checked. A MAC frame received, which has a required subvector missing, will result in a negative response indicating required subvector missing.

2.4.4.9.2 Monitor Contention Process

The monitor contention process involves all the adapters in the ring. The monitor contention process is used to establish a ring station as an active monitor. This process is started when any station on the ring detects that no active monitor is in the ring or that the active monitor is not functioning properly. Other adapters enter the process when a claim token MAC frame is received.

The process starts when a station detects the need for monitor contention and transmits a claim token MAC frame. Other stations will join the process when they receive the claim token MAC frame. These stations join the process in an active or passive role by entering contention transmit or contention repeat mode, respectively. The active station with the highest ring station address among those contenders, will be established as the active monitor.

The stations that actively participate are

- Those detecting the need for contention.

□ Those configured to contend by the attached system's OPEN command that have not received a claim token MAC frame with a source address higher than their own ring station address, except an adapter that is the active monitor at the time the claim token MAC frame is received.

All other stations on the ring enter contention repeat mode.

stations that actively participate enter contention transmit mode. These stations repeatedly transmit a claim token MAC frame followed by ring idles to all stations on the local ring without waiting for free tokens. The frame transmission is repeated at 20 millisecond intervals. The claim token MAC frame information field contains the transmitting station's upstream neighbor's (UNA) ring station address.

Applicable MAC Frames

Table 2–9 describes the MAC frames used in this process.

Table 2–9. Monitor Contention Process – Applicable MAC Frames

Frame Description				Subvectors		
M–V (Frame Name)	Destination Class	Source Address	Destination Address	R/O		Name
				XMT	RCV	
>03 Claim token	>0	>0	All stations	A A	O R	>OB Physical drop >02 UNA
>25 Report new monitor	>4	>0	F(NM)	A A A		>OB Physical drop >02 UNA >22 Product
>28 Report monitor error	>6	>0	F(REM)	A A A A A A		>30 Error code: >0001 Monitor error >0002 Duplicate monitor >0003 Duplicate address >OB Physical drop >02 UNA

Key:	A	always transmitted
	R	required
	O	optional
	N	not syntax checked

Contention Transmit Mode

Table 2–10 describes the conditions that cause a station to enter contention transmit mode and begin the monitor contention process.

Table 2–10. Events Triggering Monitor Contention

Event	Description
Good token not received	If a standby monitor does not detect a good token within a 2.6-second period, it causes the station to enter contention transmit mode.
Ring poll timeout	If a standby monitor does not detect an active monitor present (AMP) MAC frame within a fifteen-second period, the station enters contention transmit mode.
Insertion timer timeout	If no active monitor present/standby monitor present or ring purge MAC frame is detected within 18 seconds of inserting onto the ring, the adapter enters contention transmit mode. When an active monitor is established, the adapter returns to the insertion process.
Unsuccessful purge	If the active monitor is unable to successfully purge the ring, the station enters contention transmit mode.
Frequency error	If a standby monitor detects a frequency error, it enters contention transmit mode.
Beacon transmit mode	If a station beacon transmit mode receives its own beacon frame, it enters contention transmit mode.
Beacon repeat mode	If a station in beacon repeat mode detects a circulating beacon frame, it enters contention transmit Mode.
Beacon escape timer	If an adapter in beacon repeat mode detects no beacon frames for 200 milliseconds, it enters contention transmit mode.
Signal loss	If a signal loss condition is detected by the adapter, it enters contention transmit mode.

When the adapter enters contention transmit mode, it starts the monitor contention timer and continually transmits a claim token MAC frame (followed by idles) every 20 milliseconds. Claim token MAC frames are transmitted immediately without waiting for a free token. If the adapter is in beacon repeat mode and receives a claim token MAC frame, it will exit the beacon process. When a claim token MAC frame is received with a source address higher than the station's own address, the adapter retransmits that frame and, when frame transmission has completed, enters contention repeat mode. When the adapter receives three successive claim token MAC frames with a source address equal to its own address and a UNA subvector equal to its saved UNA (i.e., receives its own frame) it wins contention. This adapter becomes the new active monitor, exits contention transmit mode, and performs the ring purge process. It then queues a report new monitor MAC frame for transmission to the network manager.

Contention Repeat Mode

When an adapter enters contention repeat mode, it starts its monitor contention timer. The monitor contention timer is a one-second timer, which serves as a watchdog timer during the contention process to prevent the process from continuing indefinitely if the contention condition cannot be resolved. If this timer expires, the beacon process is entered.

When an adapter receives a claim token MAC frame, the frame is ignored. When an adapter receives a ring purge MAC frame, it resets its monitor contention timer, resumes normal operation, and starts the standby monitor functions.

Adapters Not in Monitor Contention

An adapter not in contention transmit mode and not in contention repeat mode takes the following action when it receives a claim token MAC frame.

- ❑ If the source address in the received frame is less than this adapter's specific address, and the adapter was OPENED with the contender option, then the adapter enters contention transmit mode. Otherwise, it enters contention repeat mode.
- ❑ If the source address in the frame is equal to this adapter's specific address, the frame is ignored.

Exception Events

The following events cause an adapter to queue an error reporting MAC frame and enter contention repeat mode:

- ❑ An active monitor ring station that receives a claim token MAC frame (1) deactivates its active monitor functions, (2) enters contention repeat mode, and (3) queues the report monitor error MAC frame with an error code subvector value equal to >0001. This indicates that a standby monitor detected an error in the active monitor. This frame is transmitted after contention is resolved.
- ❑ A ring station in contention transmit mode that receives a claim token MAC frame with a source address equal to its specific address and a UNA address **not** equal to its UNA (1) enters contention repeat mode and, when contention is resolved, (2) queues the report monitor error MAC frame with an error code subvector value equal to >0003, indicating a duplicate address was detected in monitor contention.

The following exceptions cause an adapter in contention transmit or repeat mode to enter the beacon transmit mode in the beacon process or to abnormally terminate an OPEN command from the attached system.

- □ If the monitor contention timer expires (one second) and the adapter has not completed phase 2 of the insertion process, the OPEN command is terminated with an error to the attached system. The adapter is deinserted from the ring.
- □ If monitor contention timer expires and the adapter is in phase 3 or 4 of the insertion process or is inserted, the adapter enters beacon transmit mode.

If an adapter is inserted, is in contention repeat or transmit mode, and receives a beacon MAC frame, it enters the beacon process in beacon repeat mode.

2.4.4.9.3 Ring Purge Process

The ring purge process is used to put the ring into a normal condition that allows frames to be transmitted using the token protocol. The ring purge process is started by the active monitor when one of the following conditions occurs:

- □ A token error condition is detected by the active monitor.
- □ An adapter becomes the active monitor in the monitor contention process.

Applicable MAC Frames

Table 2–11 describes the ring purge MAC frame used in the ring purge process.

Table 2–11. Ring Purge Process – Applicable MAC Frames

Frame Description				Subvectors		
M–V (Frame Name)	Destination Class	Source Address	Destination Address	R/O		Name
				XMT	RCV	
>04 Ring purge	>0	>0	All stations	A A	R O	>02 UNA >OB Physical drop

Key:	**A**	always transmitted
	R	required
	O	optional
	N	not syntax checked

Ring Purge Procedure

When the active monitor enters the ring purge process, a ring purge timer starts and the active monitor transmits a ring purge MAC frame. This transmission takes place without waiting for a free token and without releasing a free token upon completion. Following transmission of the ring purge frame, the adapter sends continuous idles (zeros).

When the active monitor receives the transmission after the frame has circulated the ring, it checks for errors in the transmission. These errors could be code violation errors or frame check sequence (CRC) errors.

If an error is detected, the adapter transmits another ring purge MAC frame until a frame is received error-free or until the ring purge timer expires.

The ring purge timer functions as a watchdog timer to limit the time the adapter will continue to transmit ring purge MAC frames during the process. This one-second timer is reset when the active monitor has received one ring purge MAC frame that circulated the ring with no errors.

After an error-free frame is received, the adapter transmits a free token of priority equal to the reservation priority in the last ring purge MAC frame that was stripped by the adapter.

Ring Purge Receiver

The adapter in contention repeat mode, which receives a ring purge MAC frame returns to normal operation and starts the standby monitor functions.

If a standby monitor that is not in contention repeat mode receives a ring purge MAC frame, it resets auto-removal variables and discards the frame.

If an active monitor receives a ring purge MAC frame, it checks the source address of the frame to determine if it transmitted the frame. The reception of this frame by an active monitor, which did not transmit the frame, is an exception condition of the active monitor.

Exception Conditions

If the ring purge timer expires (one second), the adapter enters the monitor contention process in contention transmit mode. If the adapter is in the insertion process when the ring purge timer expires, the adapter is deinserted, and the OPEN command is terminated with an error.

Ring Poll Process

The ring poll process enables each adapter on the ring to acquire the six-byte specific address of its upstream neighbor station. In this process, each station transmits its specific address (6 bytes) and its physical drop number (4 bytes) to the next downstream station. Each station saves its UNA and, if different from the previously saved UNA, queues for transmission a report SUA change MAC frame to the network manager.

This process allows an ordered list of stations on the network to be maintained by the network manager for network diagnostic purposes.

Applicable MAC Frames

Table 2–12 presents the MAC frames used by the ring poll process.

Table 2–12. Ring Poll Process Applicable MAC Frames

Frame Description				**Subvectors**		
				R/O		
M–V (Frame Name)	**Destination Class**	**Source Address**	**Destination Address**	**XMT**	**RCV**	**Name**
>05 Active monitor	>0	>0	All stations	A A	O R	>OB Physical drop >02 UNA
>06 Standby monitor	>0	>0	All stations	A A	O R	>OB Physical drop >02 UNA
>27 Report pollframe failure	>6	>0	F(REM)	A		>OA/SA of last ring AMP or SMP
>26 Report SUA	>4	>0	F(NM)	A		>OB Physical drop

Key:	**A**	always transmitted
	R	required
	O	optional
	N	not syntax checked

Ring Poll Procedure

The active monitor will transmit an active monitor present (AMP) MAC frame whenever its poll timer expires (every seven seconds) or at the end of the ring purge process. The active monitor also resets an internal flag termed the poll complete flag.

The following procedure is applied to every active monitor present (AMP) or standby monitor present (SMP) MAC frame received by the adapter following phase 3 of the insertion process.

- If an active monitor present (AMP) MAC frame is received by an active monitor and the frame originated from the active monitor, the ARI and

FCI bits of the FS field are examined to see if another station received the frame. If the ARI and FCI bits are all zero, the active monitor is the only station on the ring. The adapter then sets the single-station bit in the ring status register if not previously set.

❑ If an AMP or SMP MAC frame is received by any adapter, the ARI and FCI bits are examined to see if another station is in the ring. If these bits are not all zeros, the single-station bit in the ring status register (if it was set) is reset.

❑ If an AMP MAC frame is received by a stand-by monitor, it will restart the receive poll timer.

❑ If an AMP MAC frame is received by any station, the adapter will reset to zero an internal status flag called the receive ARI/FCI flag.

❑ If an AMP or SMP MAC frame is received by the active monitor and an internal flag called the poll complete flag is set to one, the procedure terminates.

❑ If any station receives an AMP or SMP MAC frame and the ARI and FCI bits are not all zeros, then the station saves the source address of the frame as the latest poll address. This address is used by the active monitor in a report ring poll failure MAC frame if a failure is detected.

❑ If any station receives an AMP or SMP MAC frame and the ARI and FCI bits are all zeros (no other adapter has copied the frame), the adapter compares the source address of the frame to the upstream neighbor's address (UNA) previously saved. If the source address does not equal the UNA address, then the adapter saves the source address as the adapter's UNA, and a report SUA change MAC frame is queued for transmission to the network manager. Next, the adapter checks the internal receive ARI/FCI flag. If this flag is set to one, the adapter increments the ARI/FCI error counter and will not queue an SMP MAC frame for transmission. If the flag was zero, it will be set to one.

Note:

When the ARI/FCI error counter is incremented, it indicates that the upstream neighbor station is unable to set the ARI or FCI bits of received frames.

❑ If the active monitor receives an AMP or SMP MAC frame with the ARI and FCI bits all zeros, then it sets the internal poll complete flag, and the process terminates.

❑ If a standby monitor receives an AMP or SMP MAC frame with the ARI and FCI bits all zeros, the station starts the poll response timer (20 ms). At the expiration of this timer, the station queues a SMP MAC frame for transmission. This frame propagates the ring poll to the next downstream station.

Active Monitor Exception Conditions

If a poll frame from the monitor's upstream neighbor does not arrive before the poll timer expires (seven seconds), the monitor starts a new poll cycle. The monitor queues a report ring poll failure MAC frame for transmission to the ring error monitor (REM), sending to the REM the latest poll address.

If the active monitor receives an AMP MAC frame from another adapter, then two adapters think they are the active monitor. The active monitor therefore resets its active monitor functions and queues for transmission a report monitor error MAC frame to the REM with a subvector value, indicating that a duplicate monitor was detected. The adapter now responds to the received AMP MAC frame as a standby station.

In order to detect streaming in the adapter immediately upstream from the active monitor, the active monitor checks that it receives its own AMP MAC frame within 15 seconds of the last one. If it does not receive it, the active monitor will enter monitor contention in the contention transmit mode.

Standby Monitor Exception Conditions

In order to detect the failure of the active monitor, each standby monitor verifies that an AMP frame is received within 15 seconds of the last one. If this AMP is not received, the standby monitor discards any frame(s) pending transmission and enters monitor contention in contention transmit mode.

This 15-second period is timed by using the receive poll timer. This timer is active only in a standby monitor.

Exception Conditions–All Chapters

Insertion Phases 1 and 2

If the ring poll procedure requires the transmission of a report SUA change or an SMP MAC frame, the transmission of these frames is delayed until phase 3 of the insertion process.

Contention Transmit Mode or Beacon Transmit Mode

Entry into either contention transmit mode or beacon transmit mode causes the adapter to purge any AMP or SMP MAC frames pending transmission.

Poll Frame Retry

AMP or SMP MAC frames are not checked for successful transmission. An unsuccessful transmission will cause the poll cycle to terminate, but the active monitor will start a new poll cycle when the poll timer expires (seven seconds).

2.4.4.9.4 Beacon Process

The beacon process is used to recover the ring when a ring station has sensed that a hard error has occurred, rendering the ring inoperable. A station detecting a ring failure upstream transmits or beacons information in a MAC frame that isolates the error location.

The beacon process is started when an adapter inserted in the ring detects that an adapter's monitor contention timer expired in the monitor contention process, indicating that contention could not be resolved.

When a station beacons, all other stations on the ring enter beacon transmit or beacon repeat mode. In beacon transmit mode, beacon MAC frames are transmitted at 20-millisecond intervals without waiting for a token. Idle zero bits are transmitted between frames. An adapter not in the beacon process that receives a beacon MAC frame enters beacon repeat mode. An adapter in the insertion process will not enter the beacon process but will terminate the OPEN command with an error, indicating that the ring is beaconing.

Applicable MAC Frames

The MAC frames used by this process are described in Table 2–13.

Table 2–13. Beacon Process – Applicable MAC Frames

Frame Description				Subvectors		
M–V (Frame Name)	Destination Class	Source Address	Destination Address	R/O XMT	R/O RCV	Name
>02 Beacon	>0	>0	All stations	A A A	R R O	>01 beacon type >02 UNA >OB Physical drop

Key: **A** always transmitted
R required
O optional
N not syntax checked

Beacon Transmit Mode

When the adapter transmits a beacon MAC frame, the content of the frame varies, depending on the reason the beacon process was entered. All beacon frames contain a beacon-type subvector, which identifies the reason the adapter is transmitting beacon MAC frames. This parameter ranks the priority of the beacon MAC frame. The beacon MAC frame with the lowest beacon type subvector is the highest priority type beacon. The beacon MAC frame types are as follows:

- ❑ Set recovery mode (>0001, highest priority): This type of beacon MAC frame can be originated only by the process in an attached product acting as a recovery station with a function class zero authorization. (NOTE: This adapter software will not originate this type of beacon frame.)
- ❑ Signal loss (>0002): This type of beacon MAC frame is transmitted when a monitor contention timeout occurs and the contention transmit mode was entered because of a signal loss condition being detected. If a signal loss condition is detected while in beacon transmit type 0003 or 0004, the beacon type will change to 0002. If, while transmitting beacon MAC frames with this beacon-type subvector, a signal is detected, the beacon type subvector will be changed to 0003.
- ❑ Bit streaming (>0003): This type of beacon MAC frame is transmitted when a monitor contention timeout occurs in an adapter in monitor contention transmit mode, and no claim token MAC frames were received during the contention period.
- ❑ Contention streaming (>0004, lowest priority): This type of beacon MAC frame is transmitted when a monitor contention timeout occurs in an adapter in monitor contention (transmit or repeat), and one or more claim token MAC frame(s) were received during the contention period. This indicates that contention could not be resolved within one second.

When the adapter enters beacon transmit mode, it sets to one the hard error and beacon transmit bits in the ring status register.

Beacon MAC frames are transmitted every 20 ms without waiting for a token and are followed by idles. In beacon transmit mode, the adapter uses its internal crystal oscillator, rather than the clock recovered from the incoming signal, to transmit.

If the adapter receives a beacon MAC frame that was transmitted with a source address equal to this adapter's specific address (that is, a beacon MAC frame transmitted by this adapter), the adapter discontinues beaconing, clears the hard error and beacon transmit bits to zero, and enters the monitor contention process. When the beaconing adapter receives its own beacon frames, the ring hard error is recovered to the point where ring continuity is achieved, allowing for the next stage in ring recovery the monitor contention process.

If the adapter receives a beacon MAC frame that it did not transmit but has a beacon-type subvector value of equal or higher priority than the beacon-type subvector value being transmitted by this adapter, the adapter enters beacon repeat mode and clears the beacon transmit bit in the ring status register.

If the adapter receives a claim token MAC frame, the frame is ignored.

If the adapter does not proceed within 16 seconds to monitor contention mode while in beacon transmit mode, the adapter will execute the beacon transmit auto removal test. This test is executed only once while in the beacon process. The adapter must return to normal operation before the beacon transmit auto removal test can be executed again.

Beacon Repeat Mode

When the adapter enters beacon repeat mode, it sets to one the hard error bit in the ring status register. The adapter uses the recovered ring clock to repeat the incoming signal.

The adapter verifies that beacon frames continue to be received in beacon repeat mode. If no beacon MAC Frame is received for a period of 200 milliseconds, the adapter assumes that the conditions causing the beacon have been corrected. In this case, the adapter resets to zero the hard error bit in ring status, exits the beacon process, and enters the monitor contention process.

When a beacon MAC frame is received, it is inspected to determine if the frame was sent by the upstream neighbor (source address of the frame is equal to this adapter's UNA) or by the nearest downstream station (UNA in the frame is equal to this adapter's address).

When beacon frames are received by any adapter, the following actions are taken:

- ❑ If eight consecutive beacon MAC frames are received from the nearest downstream station (the UNA ID subvector in the received beacon MAC frame is equal to the adapter's specific address), the adapter executes the beacon-receive auto removal test. This test can be executed only once while the adapter is in the beacon process. The adapter must return to normal operation before the beacon-receive auto removal test can be executed again.

- ❑ If the beacon frames are not from the nearest downstream station, the following two checks are made:
  - ■ If two consecutive beacon MAC frames are received from the nearest upstream station, the adapter activates its functions for marking and removing circulating frames. This marking and removing process is described in Section 2.4.4.8.1. The adapter will also prevent itself from executing the beacon-receive auto removal test. If a circulating beacon frame is detected, the adapter enters the monitor contention process in contention transmit mode.
  - ■ If the beacon MAC frame is not from the nearest upstream station, the adapter disables its circulating frame removal functions.

If a monitor contention MAC frame is received, the adapter resets the hard error bit in ring status in the protocol handler, exits the beacon process and enters the monitor contention process.

Beacon-Transmit Auto Removal Test

The beacon-transmit auto removal test is a process for automatically withdrawing faulty stations from the ring. A station transmitting beacon MAC frames has a fault somewhere in the path between its receiver input and the transmitter of the immediate upstream station. The fault could lie either in the upstream station's lobe (the wire connecting it to the wiring concentrator), in the beaconing station's lobe, the upstream neighbor's transmitter, this adapter's receiver, or in the wiring concentrator itself. The beacon transmit auto removal test process corrects a fault if the fault is in the beaconing adapter's receiver or lobe.

If a station does not leave the beacon transmit mode within 26 seconds, it executes the beacon-transmit auto removal test, as follows:

1) It physically removes itself from the ring by deactivating the phantom circuit drive current.

2) The adapter performs the lobe media test as used in phase 0 of the insertion process. If this wrap test fails, the adapter remains removed and sets the auto removal error bit in the ring status register.

3) If the wrap test successfully completes, the adapter physically reinserts onto the ring. Its actions at that point depend on the type of beaconing it was performing prior to auto-removal, and the ring state it discovers upon reinserting.

    a) If any MAC frame is received, the adapter enters beacon-transmit mode, discards the received MAC frame, and cancels the auto removal timer.

b) If a signal loss is detected, the adapter enters beacon-transmit mode, resumes transmitting the same beacon MAC frame, and cancels the auto removal timer.

c) If the auto removal timer expires, the adapter enters beacon transmit mode, transmitting the same type beacon as before.

Beacon-Receive Auto Removal Test

The beacon-receive auto removal test corrects ring faults in the lobe of the adapter upstream of the beaconing adapter. If a fault occurs in a station's transmit path, then its downstream neighbor will beacon, and the station with the faulty transmitter will receive its neighbor's beacon frames. As described under the beacon-repeat mode section above, when eight such frames are received consecutively, a station performs the beacon-receive auto removal test, as follows:

1) The adapter physically withdraws from the ring. This wraps the transmit pair to the receive pair at the wiring concentrator.

2) The adapter executes the lobe media check phase of the insertion process. If this fails, the adapter sets the auto removal error bit in ring status and remains withdrawn from the ring.

3) If the wrap test succeeds, the adapter reinserts into the ring.

   a) If it receives any recognized MAC frame, the adapter cancels the auto removal timer, discards the MAC frame, and enters the monitor contention transmit mode.

   b) If the auto removal timer expires (18 seconds), the adapter enters the monitor contention transmit mode.

2.4.4.9.5 Hardware Error Process

The hardware error process is used to detect the following error conditions:

❑ Wire fault

❑ Frequency error

❑ Ring signal loss

Two procedures are used to detect these error conditions: the wire fault detection procedure and the ring interface error detection procedure. The wire fault detection procedure is used to detect the wire fault conditions. The ring interface error detection procedure is used to detect the frequency error condition and the signal loss condition.

Wire Fault Detection Procedure

A wire fault condition is indicated when the DC current on either wire of the transmit pair exceeds an abnormally high or low condition. This is caused, for instance, if one of the lines is open or shorted to ground. When a wire fault condition is verified to exist continuously for 5 seconds, the lobe wire fault bit of the ring status register is set. The adapter then physically deinserts from the ring.

Frequency Error Detection Procedure

When the frequency of the incoming signal differs by more than 0.6% from the local crystal oscillator, a frequency error condition is indicated. This is detected by an overflow or underflow condition of the elastic buffer.

Signal Error Detection Procedure

An incoming ring signal loss is indicated when

- ❑ The incoming signal has insufficient signal energy; or
- ❑ The incoming signal is grossly out of phase with the local phase-locked loop.

When either condition is verified to be in effect for 200 milliseconds, the software indicates a signal loss condition. if this occurs during the ring insertion process (phases 1 and 2), the OPEN command terminates with a signal loss error code. If the adapter is in the beacon-transmit auto removal process, the adapter enters beacon-transmit mode. Otherwise, the adapter enters monitor contention transmit mode.

2.4.4.9.6 Ring Insertion Process

The ring insertion process is performed after bring-up diagnostics and initialization have been completed. An OPEN command, issued by the attached system, initiates the ring insertion process. Before beginning phase 0 of the ring insertion process, the following defaults are set in the adapter:

- ❑ The physical drop number is set to zero.
- ❑ The local ring number is set to zero.
- ❑ The soft error report timer value is set to two seconds.
- ❑ The enabled function classes mask is set to permit all classes of MAC frames to be transmitted across the system interface except for ring station (0) and ring parameter server (5).
- ❑ The allowed access priority is set to three (highest priority).
- ❑ All error counters are reset to zero.

The attached system will be logically inserted in the ring when the adapter is connected into the ring and all phases of the ring insertion process have been completed successfully. Physical insertion occurs when the adapter has taken the necessary steps to establish the physical signal path from the ring through the adapter and back to the ring. In addition, the lobe media, which connects the station to the physical ring through the wiring concentrator, is verified before physical insertion occurs. Ring signal paths are switched by wiring concentrator hardware external to the adapter.

The sequence of adapter processes is divided into five phases. All five phases must be completed before the adapter is successfully inserted into the ring.

- □ Phase 0, lobe media check
- □ Phase 1, physical insertion
- □ Phase 2, address verification
- □ Phase 3, participation in ring poll
- □ Phase 4, request initialization

Applicable MAC Frames

The insertion process uses the MAC frames described in Table 2–14.

Table 2–14. Insertion Process – Applicable MAC Frames

Frame Description				Subvectors		
M–V (Frame Name)	**Destination Class**	**Source Address**	**Destination Address**	**R/O**		**Name**
				XMT	**RCV**	
>07 Duplicate address test	>0	>0	Target	A		None
>08 Lobe media test	>0	>0	All zero	A A	N	>26 Wrap data
>20 Request initialization	>5	>0	F(RPS)	A		>23 TMS380C16 software level
				A		>02 UNA
				A		>22 Product ID
				A		>21 Reserved
>OC Change parameters	>0	>4	Target		O	>09 Correlator
					O	>03 Local ring number
					O	>04 Assign phys. drop number
					O	>05 Soft error report timer value
					O	>06 Enabled function class
					O	>07 Allowed access priority
>OD Initialize ring station	>0	>5	Target		O	>09 Correlator
					O	>03 Local ring number
					O	>04 Assign phys. drop number
					O	>05 Soft error report time value

Key:

Key:	**A**	always transmitted
	R	required
	O	optional
	N	not syntax checked

Lobe Media Check: Phase 0

When the adapter is not physically inserted into the ring, the relay at the wiring concentrator wraps the transmitter's signal from a station back to its receiver. In the lobe media check phase, the adapter verifies that this lobe wrap path is functioning. The adapter transmits lobe media test MAC frames with the destination address set with a zero node address. The adapter will not copy this frame because of the destination address, but the checking of the frame for successful transmission is done when the frame is stripped. The adapter inserts a 24-bit delay into the transmit path. The subvector in the MAC frame for wrap data will have a length of 500 bytes. To complete phase 0, the adapter must successfully transmit 1500 frames and one duplicate address test (DAT) MAC frame.

This procedure is also used during the beacon-transmit and beacon-receive auto removal tests. If this procedure detects an error when used with the auto removal tests, the auto removal error bit is set in the ring status register instead of posting the OPEN failure completion codes described below.

The adapter constructs the lobe media test MAC frame and transmits a token onto the lobe.

If the subsequent token capture and transmission of the lobe media test frame does not complete within 40 milliseconds, the adapter retries the transmission. After two unsuccessful attempts, the adapter will terminate an OPEN command with an error code in the system status block (SSB).

After 511 successful transmissions of the lobe media test MAC frame, the adapter sends a duplicate address test (DAT) MAC frame to itself. This ensures that the receiver functions of the TMS380 are operational. If this frame is not received correctly, the adapter will retry once. After successful reception of this frame, the adapter proceeds to phase 1 of the insertion process.

Phase 0 Exception Events

Events that can occur asynchronously during lobe media check are as follows:

- ❑ If any MAC frame is received during this phase, the OPEN command terminates and the adapter reports a function failure to the attached system. If the frame is received when the adapter is in beacon auto removal test, the frame is ignored.
- ❑ If, after two unsuccessful transmission attempts, the lobe check fails, the OPEN command terminates and the adapter reports a function failure to the attached system.

Physical Insert: Phase 1

The adapter physically inserts by impressing a DC current (phantom drive) on the transmit signal pair. This activates a relay in the wiring concentrator that connects the receive and transmit pairs into the physical ring.

After physically inserting, the adapter waits for one of the following events to occur:

- ❑ Receipt of an active monitor present (AMP) MAC frame.
- ❑ Receipt of a standby monitor present (SMP) MAC frame.
- ❑ Receipt of a ring purge MAC frame.

Any of these indicates that an active monitor is present. If none of these events occurs within 18 seconds, the adapter starts the monitor contention process.

If no active monitor is detected, the TMS380C16 software starts the monitor contention process, and the adapter that wins contention becomes the active monitor. This phase completes successfully when the TMS380C16 software determines that an active monitor is on the ring.

Upon completion of one of these events, phase 1 is complete and the adapter enters phase 2, address verification. If the OPEN command is terminated because of an error condition during this phase, the adapter is physically deinserted from the ring.

Phase 1 Exception Events

- ❑ **Timeouts.** If an active monitor is not detected within 18 seconds from the start of this process, the adapter enters the monitor contention process in contention transmit mode.

  If the adapter enters the monitor contention process and contention is not resolved within one second, the OPEN command is terminated and the adapter reports to the attached system that the ring is beaconing.

  If, after the contention process, the adapter becomes the active monitor and performs the ring purge process, and the purge process is not completed within one second, the OPEN command is terminated and the adapter reports a ring failure to the attached system.

- ❑ **Receive Frame.** If the adapter receives an AMP or SMP MAC Frame, it will follow the normal procedure for ring poll and will exit phase 1 and enter phase 2 (address verification) of the insertion process. If the transmission of a SMP or report SUA change MAC frame is required, however, the adapter will delay the transmission until phase 3 of the insertion process.

  If a claim token MAC frame is received and the adapter has not started the monitor contention process, then the adapter will begin contention repeat mode.

If a beacon MAC frame is received, the OPEN command is terminated and the adapter reports to the attached system that the ring is beaconing.

If a remove ring station MAC frame is received, the OPEN command is terminated and the adapter reports to the attached system that a remove was received.

All other MAC frames are processed normally.

Address Verification: Phase 2

Upon successful completion of phase 1, the address verification phase is entered. The ring station address must be unique to this adapter. This phase of the insertion process ensures that this address is not being used by another adapter that is inserted in the ring.

The duplicate address test MAC frame is used for performing the address check. Note that the uniqueness check for all rings in not a function of the adapter.

The adapter sends a series of duplicate address test MAC frames addressed to itself. If another station matches the local adapter's address, then it will set to one the address recognized indicator (ARI) bits of the frame. It may also set to one the frame copied indicator (FCI) bits of the frame if the frame was copied. The adapter will assume that no other station matches its address when it receives two of its duplicate address test MAC frames with both of the ARI and FCI bits set to zero.

If two frames are received with either the ARI or FCI bits set to one, the adapter deinserts itself from the ring and terminates the OPEN command with a duplicate node address error code. Any other error condition will also cause the adapter to deinsert from the ring.

After the station's address has been checked, the adapter proceeds to phase 3 (participation in ring poll) of the insertion process.

Phase 2 Exception Events

❑ **Hardware exceptions.** If a signal loss is detected, the OPEN command terminates, the adapter deinserts, and the adapter reports a signal loss to the attached system.

If a frequency error is detected, the OPEN command terminates, the adapter deinserts, and the adapter reports a frequency error to the attached system.

❑ **Timeouts.** If phase 2 does not complete within 18 seconds, the adapter terminates the OPEN command, deinserts from the ring, and reports a timeout error to the attached system.

If the adapter enters the monitor contention process and contention is not resolved within one second, the OPEN command is terminated and the adapter reports to the attached system that the ring is beaconing.

If, after the contention process, the adapter becomes the active monitor and performs the ring purge process and the purge process is not completed within one second, the OPEN command is terminated and the adapter reports a ring failure to the attached system.

- ❑ **Receive frame.** If the adapter receives an AMP or SMP MAC frame, it will follow the normal procedure for ring poll. However, if the transmission of a SMP or a Report SUA Change MAC frame is required, the adapter will delay the transmission until phase 3 of the insertion process.

  If a claim token MAC frame is received and the adapter has not started the monitor contention process, then the adapter will go to monitor contention repeat mode. Following contention, the adapter resumes the phase 2 process.

  If a beacon MAC frame is received, the OPEN command is terminated, the adapter deinserts, and the adapter reports to the attached system that the ring is beaconing.

  If a remove adapter MAC frame is received, the OPEN command is terminated, the adapter deinserts, and the adapter reports to the attached system that a remove has been received.

  All other MAC frames are processed normally.

Participation in Ring Poll: Phase 3

When the address verification phase is completed, participation in ring poll (Phase 3) will be entered. The purpose of this phase is to ensure that the adapter has participated in the ring poll process. In this process, the adapter acquires its upstream neighbor's address (UNA) and allows the nearest downstream adapter to acquire its address as that adapter's UNA.

If an SMP or report-SUA-change MAC frame was required to be transmitted because of participation in the ring poll process (in phase 1 or phase 2 of the insertion process), then these frames are transmitted at this time. The adapter then exits phase 3 and enters phase 4 of the insertion process. If neither of the frames is pending transmission, the adapter waits to participate in the ring poll process. When the report-SUA-change and the SMP MAC frames are queued for transmission, the adapter proceeds to phase 4 of the insertion process. If no AMP or SMP MAC frame is received with ARI=FCI=00 within 18 seconds, the open command terminates with a timeout error code.

Phase 3 Exception Events

□ **Hardware exceptions.** If a signal loss is detected, the OPEN command terminates, the adapter deinserts, and the adapter reports a signal loss to the attached system.

If a frequency error is detected, the OPEN command terminates, the adapter is deinserted, and it reports a frequency error to the attached system.

□ **Timeouts.** If no AMP or SMP MAC frame is received with ARI=FCI=00 within 18 seconds, the adapter terminates the OPEN command, deinserts from the ring, and reports a timeout error to the attached system.

If the adapter enters the monitor contention process and contention is not resolved within one second, the OPEN command is terminated and the adapter reports to the attached system that the ring is beaconing.

If, after the contention process, the adapter becomes the active monitor and performs the ring purge process, and the purge process is not completed within one second, the OPEN command is terminated and the adapter reports a ring failure to the attached system.

□ **Receive frame.** If a claim token MAC Frame is received and the adapter has not started the monitor contention process, then the adapter will go to monitor contention repeat mode.

If a beacon MAC frame is received, the OPEN command terminates, the adapter deinserts, and the adapter reports to the attached system that the ring is beaconing.

If a remove adapter MAC frame is received, the OPEN command terminates, the adapter deinserts, and the adapter reports to the attached system that a remove has been received.

All other MAC frames are processed normally as if the adapter were inserted.

Request Initialization: Phase 4

Upon successful completion of phase 3, the adapter enters the final step of the insertion process. The purpose of phase 4 is to request additional operational parameters. These parameters are associated with each station on the ring. The parameters received in this process replace the default parameters set at the start of the ring insertion process.

The adapter sends a series of request initialization MAC frames to the ring parameter server functional address. If one is returned with the ARI or FCI bits set, this indicates that a ring parameter server is present on the ring. In this case, the adapter waits for an initialize ring station MAC frame from the ring parameter server or a change parameter MAC frame from the network manager.

If a ring parameter server is not present in the network, indicated by four request initialization frames whose ARI and FCI bits are all zeros, the adapter leaves the parameters that could be set in this phase at the default values, and the OPEN command is completed successfully.

If an initialize ring station or a change parameters MAC frame is received, the adapter sets the parameters received in the frame and the OPEN command is completed successfully. The destination of the change parameter or initialize ring station MAC frames may not be broadcast: it must be the specific address of the adapter.

If, after receiving its request initialization MAC frame with the ARI and FCI bits set, no response is received within 2 seconds, the adapter retransmits the request initialization frame. After four such retries, the adapter will terminate the OPEN command with a request initialization error code.

Phase 4 Exception Events

If a signal loss or frequency error condition is detected by the hardware error process, the adapter will behave as if insertion was completed.

❑ **Timeouts.** If phase 4 does not complete within 18 seconds, the adapter terminates the OPEN command, deinserts from the ring, and reports a timeout error to the attached system.

  If the adapter enters the monitor contention process and contention is not resolved within one second, the beacon process is entered in beacon transmit mode.

  If, after the contention process, the adapter becomes the active monitor and performs the ring purge process, and the purge process is not completed within one second, the OPEN command is terminated and the adapter reports a ring failure to the attached system.

❑ **Receive frame.** If a claim token MAC frame is received and the adapter has not started the monitor contention process, then the adapter will go to monitor contention repeat mode. Following contention, the adapter resumes the phase 4 process.

  If a beacon MAC frame is received, the OPEN command is terminated, the adapter deinserts, and the adapter reports to the attached system that the ring is beaconing.

  All other MAC frames are processed normally.

2.4.4.9.7 Transmit Forward Process

The transmit forward process uses the transmit forward MAC frame to cause the adapter to construct a frame for transmission from the data contained in the information field of the copied transmit forward MAC frame.

The transmit forward process is initiated by an adapter's reception of a transmit forward MAC frame.

The transmit forward process allows multiple layers of frames to be sent between adapters to test a path between ring stations. The embedded frame to be forwarded must be a transmit forward MAC frame. Adapters that enter the transmit forward process (through receipt of a transmit forward MAC frame) will transmit a report transmit forward MAC frame to the network manager functional address upon stripping the forwarded frame from the ring.

Applicable MAC Frames

Table 2–15 describes the MAC frames that are used by the transmit forward process.

Table 2–15. Transmit Forward Process – Applicable MAC Frames

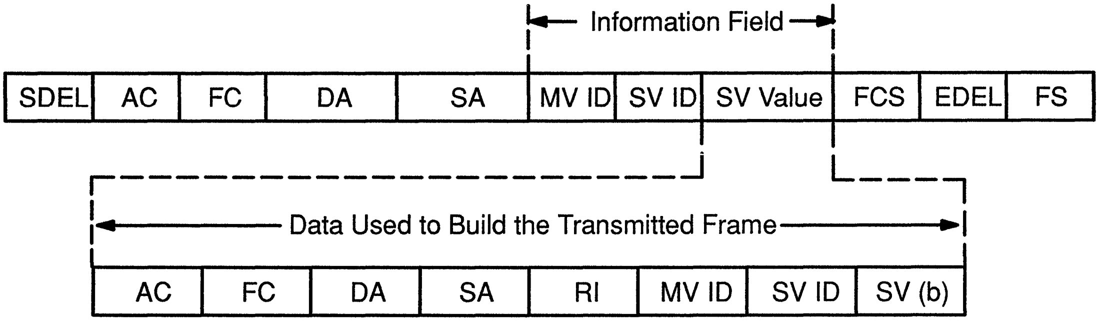
Frame Description				**Subvectors**		
M–V (Frame Name)	**Destination Class**	**Source Address**	**Destination Address**	**R/O**		**Name**
				XMT	**RCV**	
>09 Transmit forward	>0	>4	Target		R	>27 Frame forward
>2A Report transmit forward	>4	>0	F(NM)	A		>2A Transmit status code

<u>Key:</u> **A** always transmitted
R required
O optional
N not syntax checked

2.4.4.9.8 Transmit Forward MAC Frame Format

The transmit forward MAC frame is illustrated in Figure 2–25.

Figure 2–25. Transmit Forward MAC Frame



The subvector value (SV(b)) can have the same structure as the subvector value in the original frame. This data is transmitted as a frame by the ring station that receives this data.

The maximum length of a frame forward subvector is 254 bytes in a transmit forward MAC frame. The adapter does not support length extension in this subvector.

Additional Syntax Checking

After the adapter has checked the major vector length and the major vector source class, the following additional syntax checks are performed for transmit forward MAC frames.

- ❑ The first subvector in the received frame must be the frame forward subvector.
- ❑ In the frame to be forwarded, all bits in the AC and FC field must be zero.
- ❑ The major vector type in the frame to be forwarded must be transmit forward.

If the frame passes the syntax check, the remaining content of the frame after the subvector ID (SV-ID) is queued for transmission. If the frame fails the syntax checking above, a negative response of transmit forward frame error is sent to the originator of the frame.

2.4.4.10 Miscellaneous MAC Frames

This section describes MAC frames processed by the adapter and used by network management functions on the ring. This service is intended to aid in the implementation of network management functions.

2.4.4.10.1 Remove Ring Station MAC Frame

The remove ring-station MAC frame is originated by the network manager to force an adapter to deinsert from the ring. Upon receipt of a remove ring-station MAC frame, the adapter will deinsert from the ring and remain at the state following initialization (awaiting an OPEN command).

The remove ring station frame is defined in Table 2–16.

Table 2–16. Remove Ring-Station MAC Frame

Frame Description				Subvectors		
M–V (Frame Name)	Destination Class	Source Address	Destination Address	R/O		Name
				XMT	RCV	
>OB Remove ring station	>0	>4	Target			None

2.4.4.10.2 Network Management MAC Frames

The following MAC frames are originated by the network manager to request specific information from a ring station. Each request MAC frame originated by the network manager elicits a response MAC frame from a ring-station adapter.

Network Management Request MAC Frames

The request MAC frames are defined in Table 2–17.

Table 2–17. Network Management Request MAC Frames

Frame Description				Subvectors		
M–V (Frame Name)	Destination Class	Source Address	Destination Address	R/O		Name
				XMT	RCV	
>OE Request station address	>0	>4	Target		O	>09 Correlator
>OF Request station state	>0	>4	Target		O	>09 Correlator
>10 Request station attachment	>0	>4	Target		O	>09 Correlator

Key:

Key:	**A**	always transmitted
	R	required
	O	optional
	N	not syntax checked

Network Management Responses MAC Frames

The MAC frames originated by the adapter in response to the request MAC frames shown in Table 2–17 are defined in Table 2–18.

Table 2–18. Network Management Response MAC Frames

Frame Description				Subvectors		
M–V (Frame Name)	**Destination Class**	**Source Address**	**Destination Address**	**R/O**		**Name**
				XMT	**RCV**	
>22 Report station address	>4	>0	Source address of received request frame	A		>02 UNA
				A		>09 Correlator
				A		>OB Phys. drop
				A		>2B Group addr.
				A		>2C Func. addr.
				A		>21 Reserved
>23 Report station state	>4	>0	Source address of received request frame	A		>09 Correlator
				A		>23 Adapter software level
				A		>29 Adapter status vector
>24 Report station attachment	>4	>0	Source address of received request frame	A		>09 Correlator
				A		>22 Product ID
				A		>2C Functional address
				A		>06 Enabled functional classes
				A		>07 Allowed access priority
				A		>21 Reserved

Key:		
	A	always transmitted
	R	required
	O	optional
	N	not syntax checking

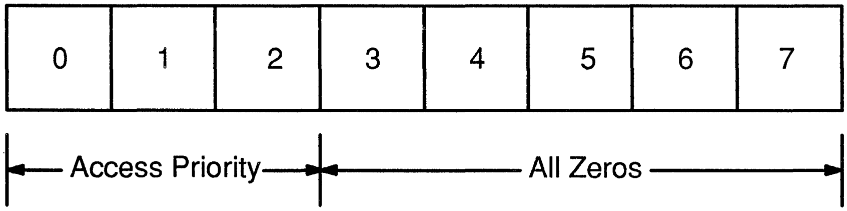
2.4.4.11 Token Transmit and Priority Control

When an attached system requests that a frame be sent to another ring station in the network, the adapter must receive and capture a token before the frame can be transmitted.

2.4.4.11.1 The Model Access Control Field

When the attached system assembles a frame in system memory (called a logical frame), the system must specify a model access control field (AC), which is passed to the adapter with the logical frame. This model AC is illustrated in Figure 2–26.

Figure 2–26. Model AC



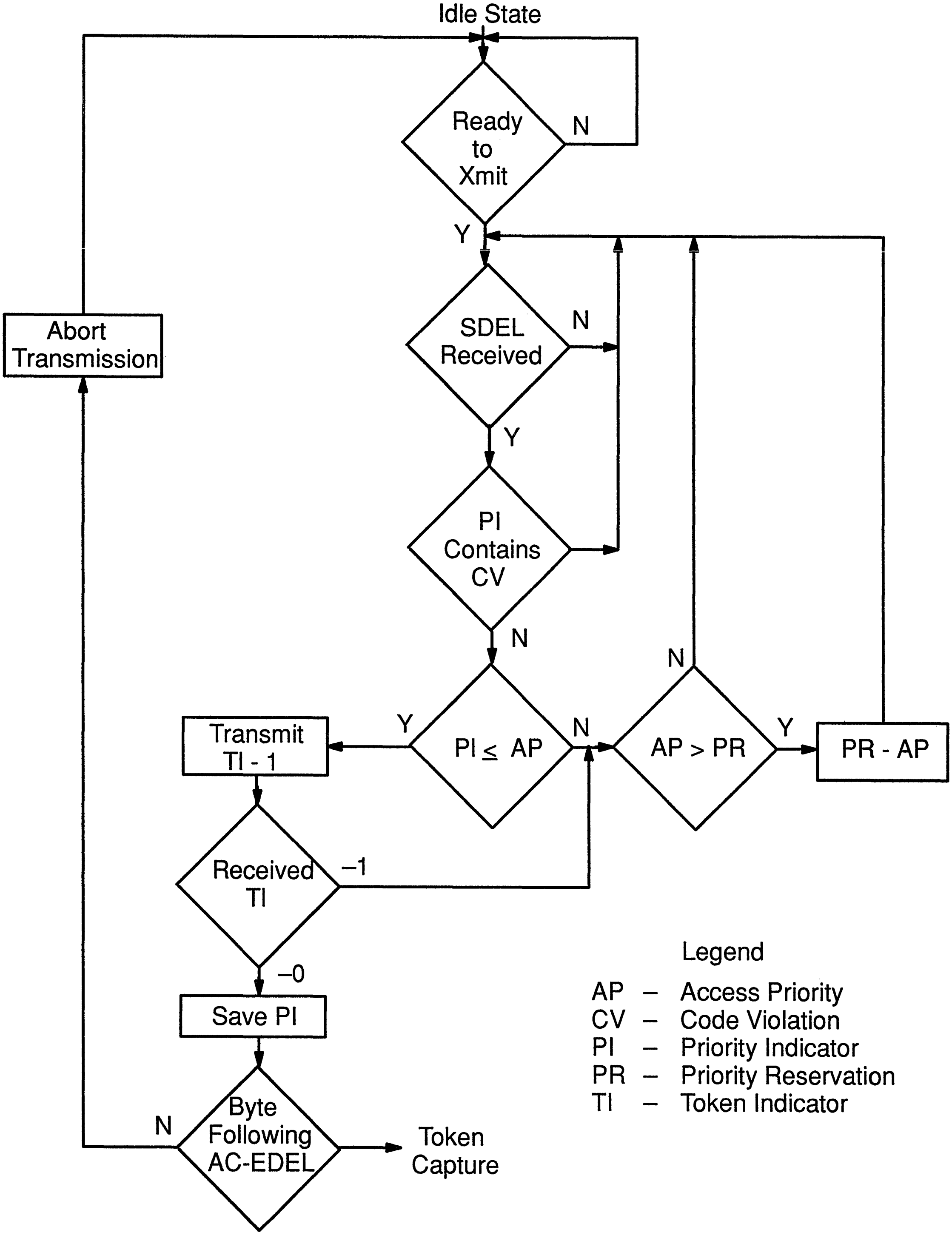
The model AC consists of the following field:

Access priority The access priority field defines the token priority level to be used in the transmission of the frame. Only priority levels 0 through 6 may be requested.

2.4.4.11.2 Token Capture

The token capture control is executed within the adapter by a state machine called the transmit token control. A flowchart, which depicts the algorithm executed by the state machine for capturing a token, is shown in Figure 2–27.

Figure 2–27. Token Capture Flowchart



Legend

AP	–	Access Priority
CV	–	Code Violation
PI	–	Priority Indicator
PR	–	Priority Reservation
TI	–	Token Indicator

Referring to Figure 2–27, when a frame is queued within the adapter for transmission, the adapter begins searching for receipt of a starting delimiter (SDEL) indicating receipt of a token or frame.

After detecting a starting delimiter, the transmit token control makes one of the following three decisions:

- ❑ If the priority indicator (PI) contains code violations, the transmit token control ignores this token and goes back to wait for another SDEL.
- ❑ If the priority indicator contains no code violations but is greater than the access priority, the transmit token control executes an algorithm for changing the priority reservation of the received token prior to repeating the token or frame.
- ❑ If the priority indicator contains no code violations and is equal to or less than the access priority, the transmit token contol transmits the token indicator of the received token as a one.

If the received token indicator (TI) is received as a one (indicating a frame), the transmit token control executes the algorithm for changing the priority reservation of the received frame.

If the received token indicator is received as a zero (indicating a token), the adapter transmits the monitor count (MC) bit as a zero and, beginning with the priority reservation field, starts transmitting the enqueued frame.

The token transmit control must still make sure that the token meets the requirement that the byte following the AC field must be an ending delimiter (EDEL).

If the byte following the AC byte is not an ending delimiter, the adapter has captured a false free token. In this case, the adapter will transmit an abort delimiter and terminate the transmission.

2.4.4.11.3 Priority Reservation Modification

The priority reservation of a received token or frame may be changed to the access priority of a queued frame only if the access priority is greater than the priority reservation of the received frame or token.

2.4.4.11.4 Token Priority Control Protocol

The token-ring protocols prioritize access on the ring according to eight priority levels: 0 is the lowest and 7 the highest priority. Each adapter on a ring network contains an independent state machine, called the priority state machine, for implementing the priority control protocol. This state machine is responsible for assuring equal access, even in the event that access to a particular priority is interrupted by a higher priority. For this reason, this priority process is sometimes referred to as the fairness function.

Note:

Although there are eight priority levels when the MAC only option is selected, the TMS380C16 software limits the maximum authorized access priority to 6 for frames transferred across the system interface for transmission.

To conceptually understand the priority control protocol, it is important to note that the adapter implements two independent but cooperating state machines: the transmit token control state machine and priority control state machine.

The priority control state machine utilizes the priority indicator (PI), token indicator (TI), and priority reservation (PR) fields of the access control (AC) field **after** the AC has been released from the transmit token control state machine of the adapter.

Note:

When the early token release option is selected in the OPEN command, the operation of token priority is less effective because the priority reservation of the received token does not comprehend additional priority reservation in the transmitted frames' priority reservation field.

Priority State Machine

The priority state machine contains three major components:

- ❑ Priority control delay.
- ❑ The new last-in-first-out (LIFO) buffer.
- ❑ The old last-in-first-out (LIFO) buffer.

The priority control delay is a nine-bit delay introduced in the repeat path to allow the priority state machine to modify the priority indicator of the transmitted token if a change is needed. This delay is inserted only when necessary to reduce this delay's effect on ring latency.

The new LIFO buffer is a four-deep LIFO. Its last entry contains the priority to which the state machine last increased the token and from which the state machine has yet to decrease the token.

The old LIFO buffer also is a four-deep LIFO. Its last entry contains the priority from which the state machine last increased the token and to which the state machine has yet to decrease the token.

With eight priority levels defined, the four-entry limit on the LIFOs is sufficient.

The adapter can be responsible for up to four increases in the priority of the token.

The priority state machine is enabled when the transmit token control issues a new token after stripping a transmitted frame from a ring. This state machine remains enabled until disabled, as described below.

Priority State Machine Functions

When the priority state machine has been enabled, it performs the following five functions:

Push This occurs when the priority control receives a token and some adapter wants the priority of the token to be greater than its present priority. The priority control pushes the priority indicator field (PI) into its old LIFO, pushes the priority reservation (PR) into its new LIFO, substitutes the priority reservation field (PR) for the PI of the token, and clears the monitor count (MC) and the PR of the token.

Pop This occurs when the priority control receives a token that has circulated the ring with a priority equal to the last value pushed onto the new LIFO. The priority control substitutes the old LIFO for the PI of the token, leaves the PR of the token as is, clears the MC of the token, and pops both its old and new LIFOs.

Replace This occurs when the priority control receives a token that has circulated the ring with the new priority, but the PR indicates that some station wants the priority of the token to be greater than the old priority. The priority control substitutes the PR for the PI of the token, substitutes the PR for the last value in the new LIFO, and clears the PR and MC of the token. The priority control does not push either of its LIFOs.

Clear This occurs when the priority control receives a token or frame for which the PI is less than the last value pushed onto the new LIFO. This condition indicates an error has occurred. The priority control leaves the token as is and clears its new and old LIFO buffers.

Remove This occurs when a token exits the functional area of the priority control and the LIFO buffers are empty. The priority control removes its delay from the repeat path and disables itself.

Priority State Machine Responses

Table 2–19 describes the operation of the priority state machine in response to stimuli conditions. The empty state refers to the condition where the priority control is active but the LIFOs have no entries. This state is entered when the state machine is initially enabled or when the state machine performs the CLEAR function. The priority control is in the not empty state when the state machine has increased the priority level of a token and has not returned the token to its original old priority.

Table 2–19. Priority Control Stimuli

Stimuli				Operation (See Note)			
				Not Empty State		Empty State	
PI<PR	PI=New	PI<New	PR>Old	TI=0	TI=1	TI=0	TI=1
0	0	0	0	Idle	Idle	Remove	Idle
0	0	0	1	Idle	Idle	Remove	Idle
0	0	1	0	Clear/remove	Clear	* *	* *
0	0	1	1	Clear/remove	Clear	* *	* *
0	1	0	0	Pop/remove	Idle	Remove	Idle
0	1	0	1	Replace	Idle	* *	* *
0	1	1	0	*	*	*	*
0	1	1	1	*	*	*	*
1	0	0	0	*	*	*	*
1	0	0	1	Push	Idle	Push	Idle
1	0	1	0	Clear/remove	Clear	* *	* *
1	0	1	1	Clear/remove	Clear	* *	* *
1	1	0	0	*	*	*	*
1	1	0	1	Replace	Idle	Push	Idle
1	1	1	0	*	*	*	*
1	1	1	1	*	*	*	*

Legend:

*	—logically inconsistent
PI	—priority indicator
TI	—token indicator
New	—last value pushed into new LIFO
Old	—last value pushed into old LIFO
Empty State	—no entries on either the new or old LIFOs
Not Empty State	—entries on both the new and old LIFOs
**	—impossible because new = 0
PR	—priority reservation

Note: If TI contains a code violation, the priority control remains idle.

Priority Operation Example

This example assumes the existence of a three-node ring with the following initial conditions.

- ❑ Station A has a frame queued for transmission with an arbitrary access priority.
- ❑ Station B has a frame queued for transmission with an access priority request of 4.
- ❑ Station C has a frame queued for transmission with an access priority request of 6.
- ❑ The initial location of the token is as shown in Figure 2–28. The token is priority-free and has no priority reservation specified.

This example uses a shorthand notation to represent the priority indicator (PI), token indicator (TI), and priority reservation (PR) of the circulating token or frame, as follows: PI/TI/PR. A zero in the TI position indicates a token; a one in this position indicates a frame. For example:

4/0/0 is a token of priority 4 with no reservation.
4/1/2 is a frame of priority 4 with a reservation of 2.

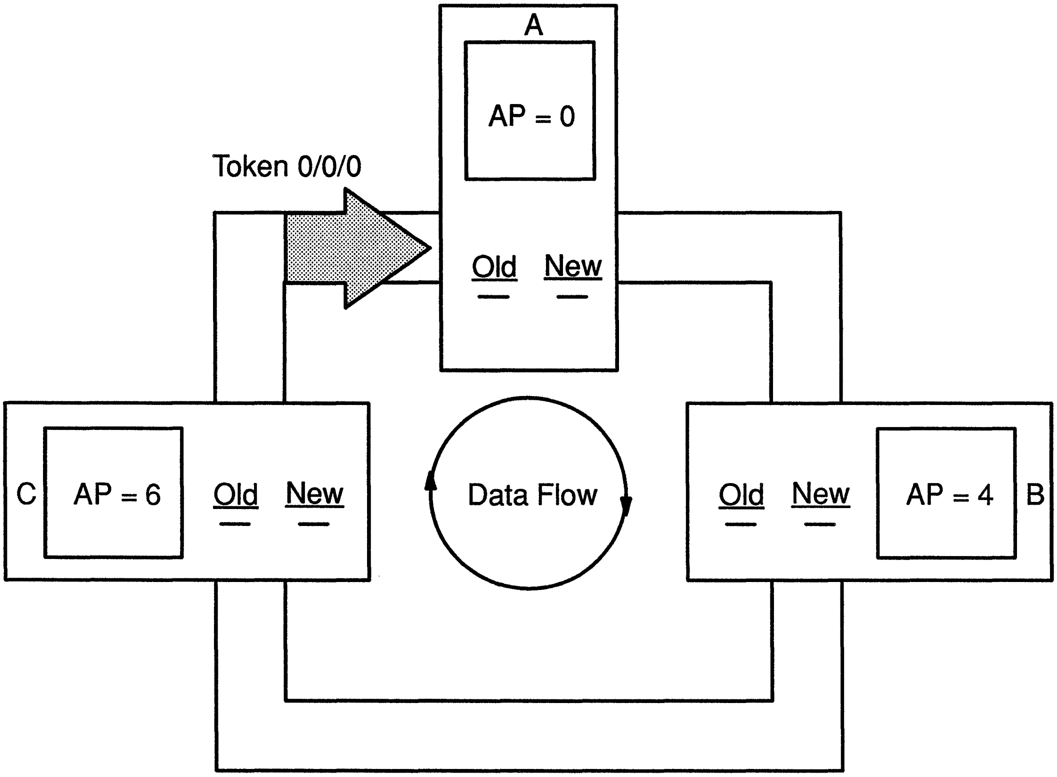
A description is provided for each following figure. Each figure illustrates a sequence of major events that occur in the overall priority control protocol as implemented by the token ring.

Example — Step 1

Figure 2–28 illustrates the initial state of the three-station ring. The token is at the position shown with the priority indicator (PI) equal to 0, the token indicator (TI) equal to zero, and the priority reservation (RP) equal to zero.

Note that the shaded block within each station represents a queued frame with the access priority shown. In addition, the contents of both the old and new stacks are shown for each station.

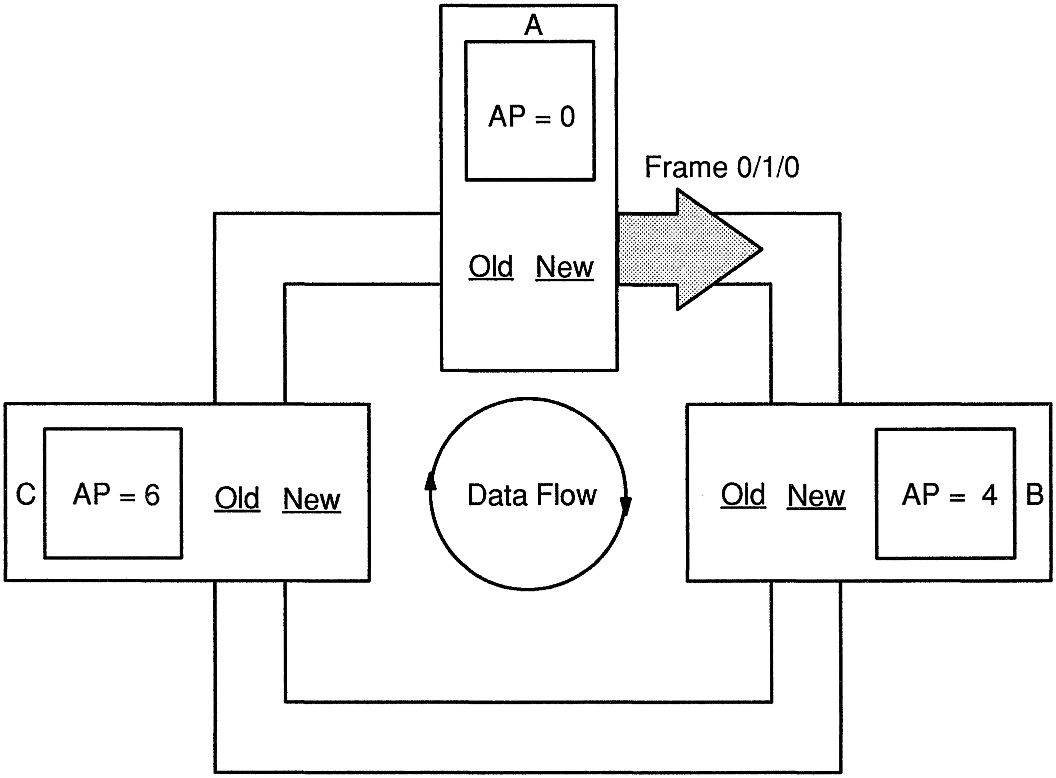
Figure 2–28. Token Is at Point Shown



Example — Step 2

Figure 2–29 shows that once the token is received by station A, it is used to transmit the queued frame. The token was captured because the PI was zero. At that point in the figure, the PI of the frame is still 0, the TI is set to one (indicating a frame), and the PR is left at zero.

Figure 2–29. Station A Captures Token and Begins Transmission of Frame

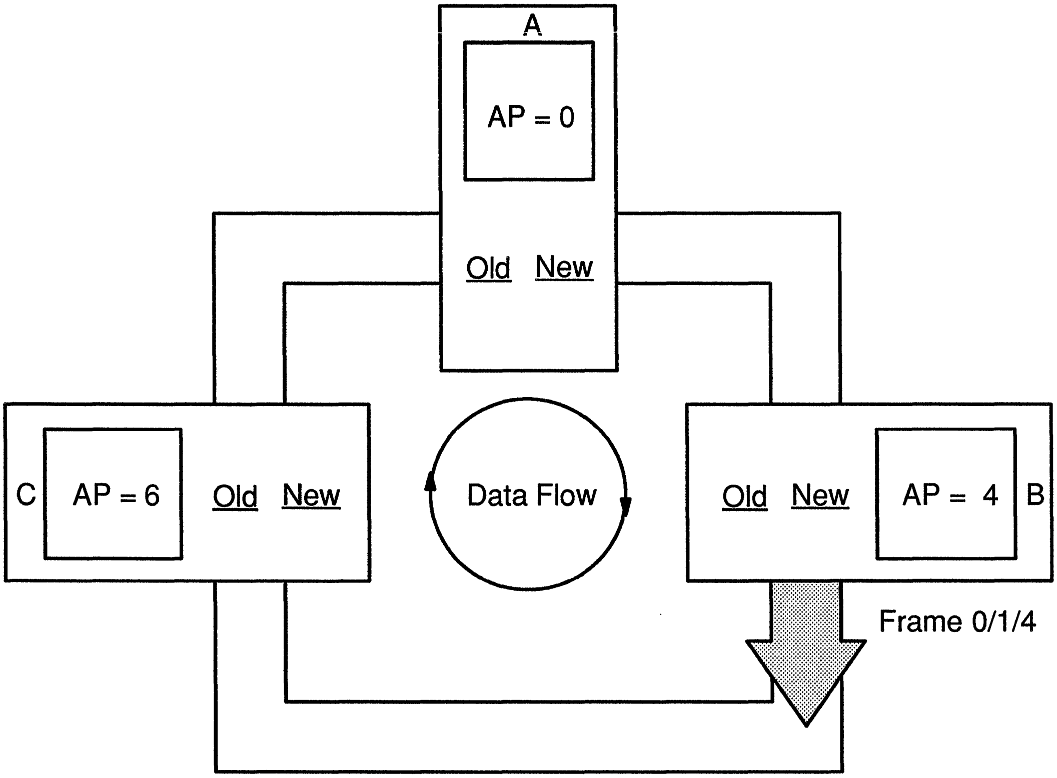


Example — Step 3

Figure 2–30 shows that station B has received station A's frame. Because the TI is equal to one and station B wants to increase the priority of a token to priority 4, station B changes the priority reservation (PR) of the frame to 4 before repeating the frame on the ring.

Thus, at this point, the PI remains at 0, the TI is still 1, but the PR is now 4.

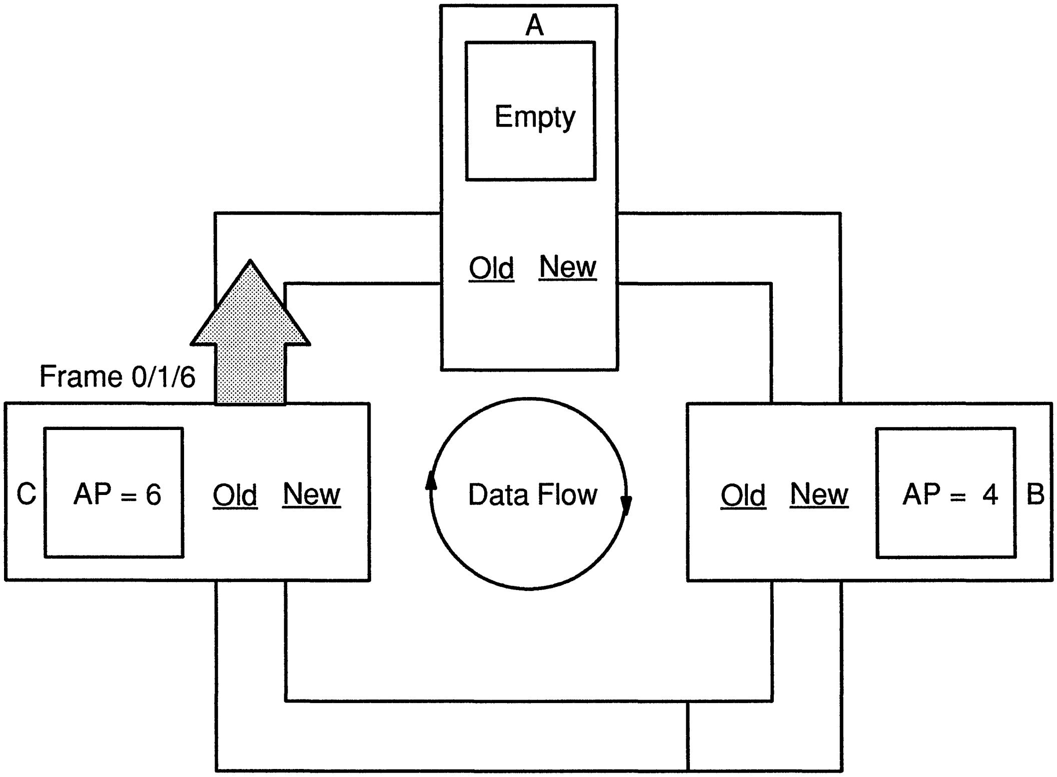
Figure 2–30. Station B Repeats Frame and Changes PR to 4



Example — Step 4

Figure 2–31 shows that when station C receives the frame, it repeats the frame with a PR equal to 6. This is because station C's queued frame has an access priority of 6, which is greater than the PR of the frame received. Thus, station C also desires a priority token, but at a higher priority than station B.

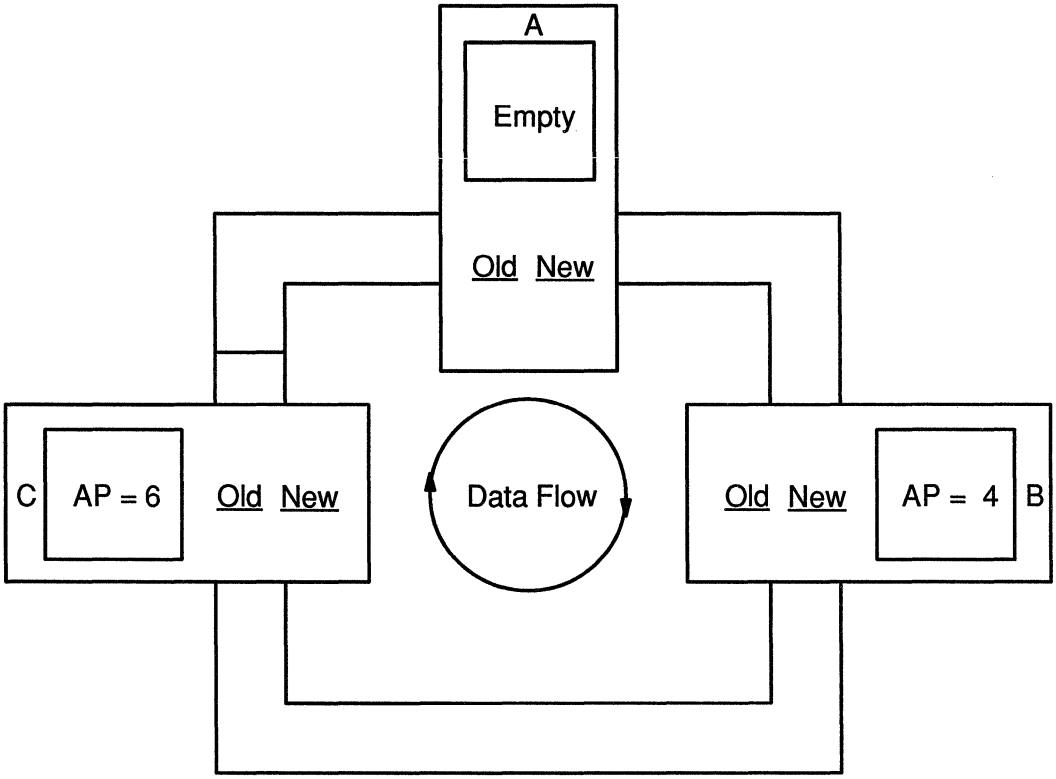
Figure 2–31. Station C Repeats Frame and Changes PR to 6



Example — Step 5

Figure 2–32 shows that when the frame originated by station A is returned to station A, it begins stripping the frame from the ring.

Figure 2–32. Station A Strips Frame from the Ring



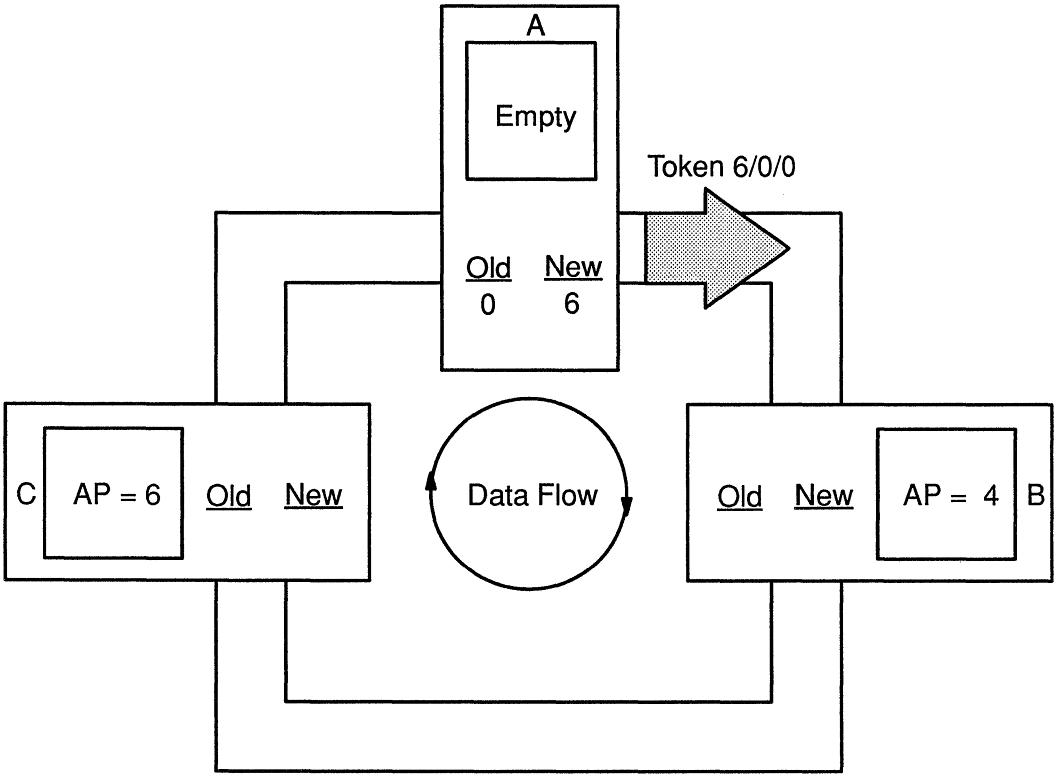
Example — Step 6

Figure 2–33 shows that station A, having now stripped its frame from the ring, activates its priority state machine and releases a new token to the state machine. The priority state machine examines the PR field of the token and performs a push operation on its new and old LIFO stacks (as can be determined from Table 2–19).

This push operation results in 6 (the new token priority level) and 0 (the old priority level) being pushed into the new and old stacks, respectively.

The transmitted token's PI now equals 6, the TI equals 0, and the PR equals 0.

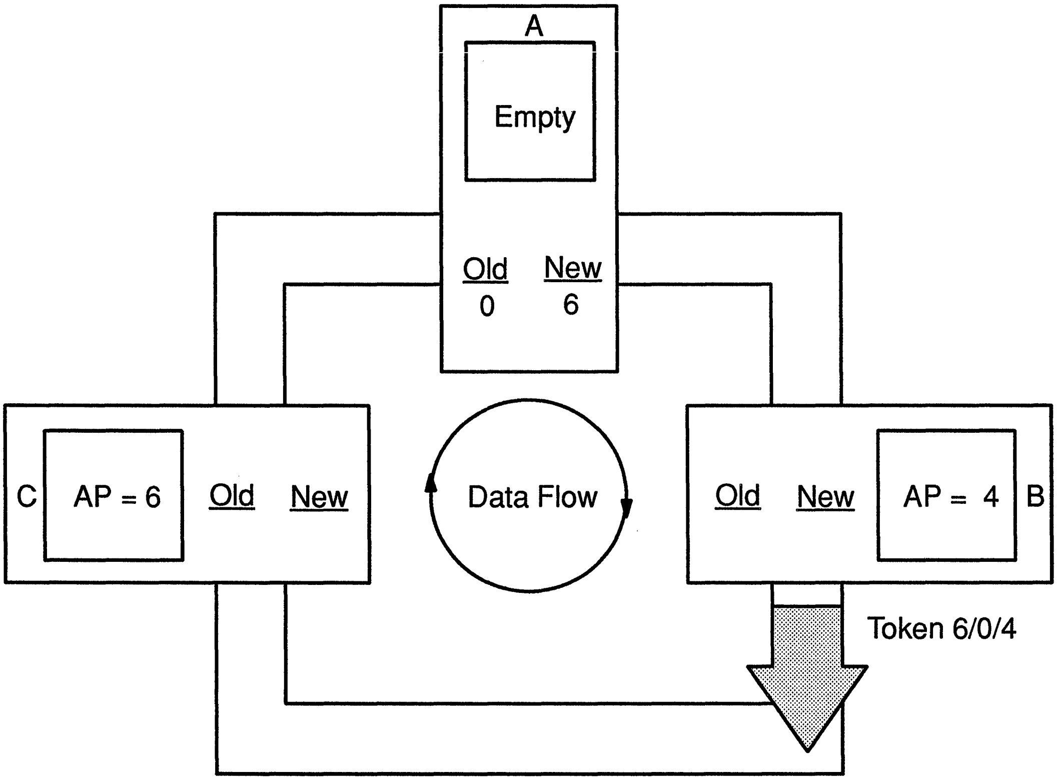
Figure 2–33. Station A Releases a New Token



Example — Step 7

Figure 2–34 shows that when station B receives the priority 6 token transmitted by station A, it cannot capture the token for transmission because its access priority is less than the PI of the token. Thus, station B changes the PR of the field, requesting again that a token of priority 4 be issued. At this point, the PI equals 6, the TI equals 0, and the PR is 4.

Figure 2–34. Station B Repeats Token and Changes PR to 4

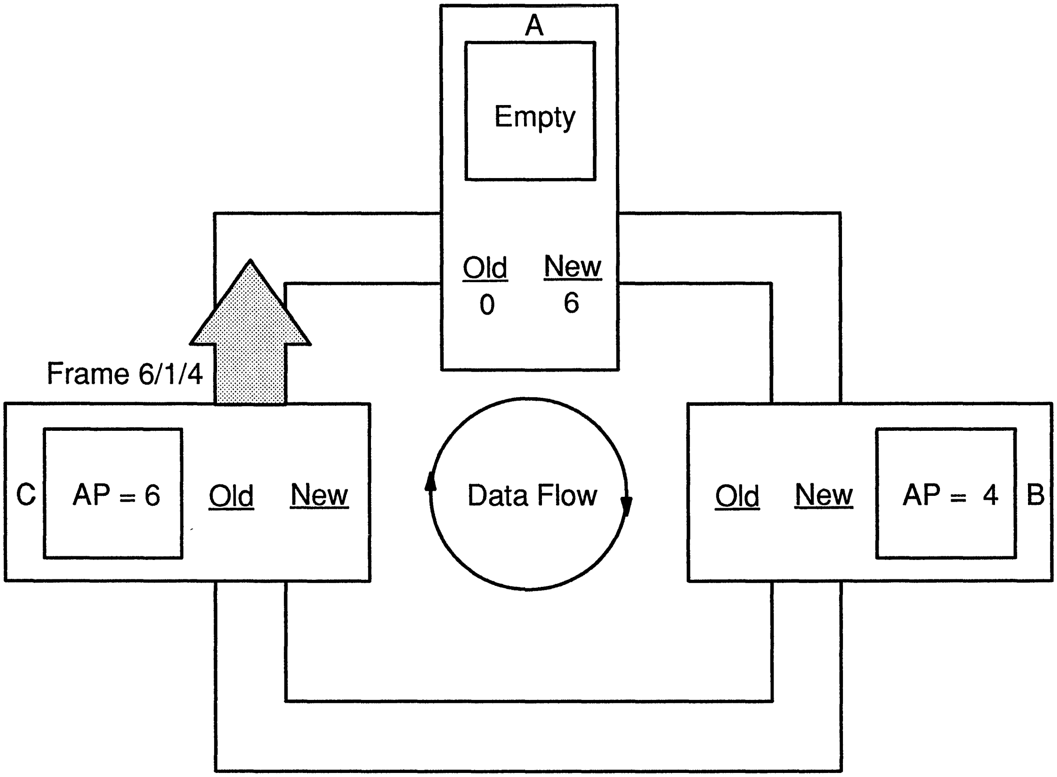


Example — Step 8

Figure 2–35 shows that when station C receives the priority 6 token, it now captures the token and begins transmitting its queued frame. The token was captured because the PI equaled the access priority of the queued frame.

Note that the PR field was left unchanged.

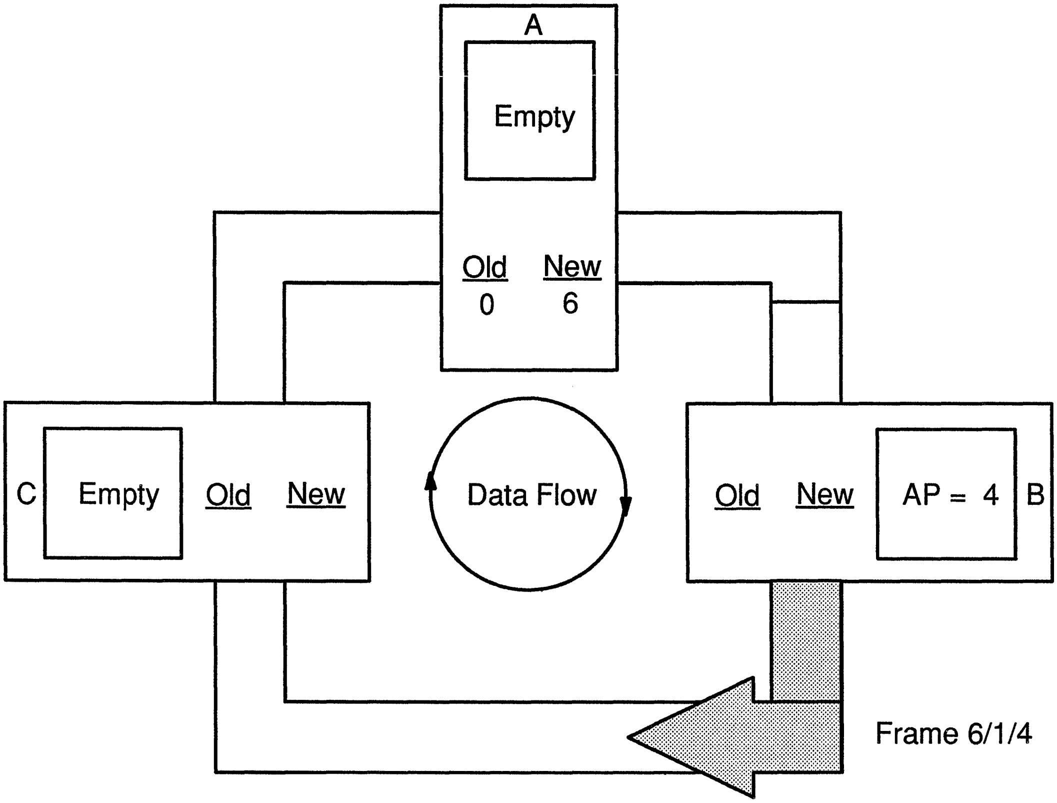
Figure 2–35. Station C Captures Token and Begins Transmission of Frame



Example — Step 9

Figure 2–36 shows that the frame transmitted by station C circulates the ring normally. Neither station A nor station B modify the PR value. Station A has no frame to transmit, and station B need not change the PR from its current value.

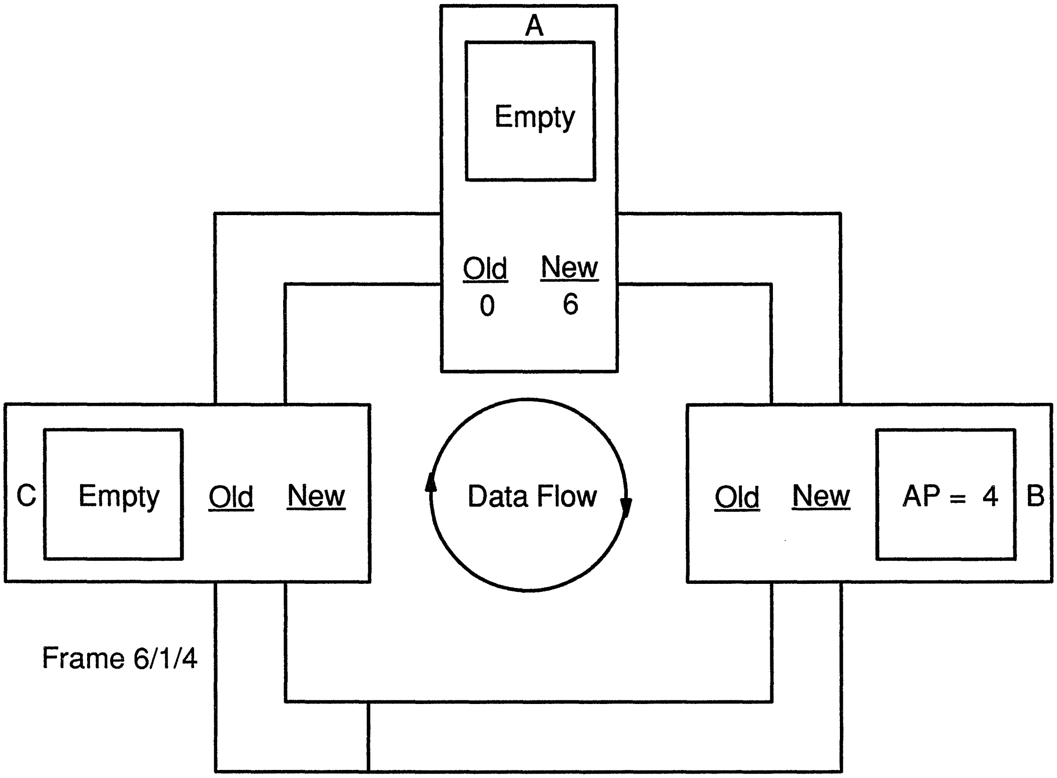
Figure 2–36. Frame Circulates with No Modification



Example — Step 10

Figure 2–37 shows that station C receives the frame it transmitted and begins stripping the frame from the ring.

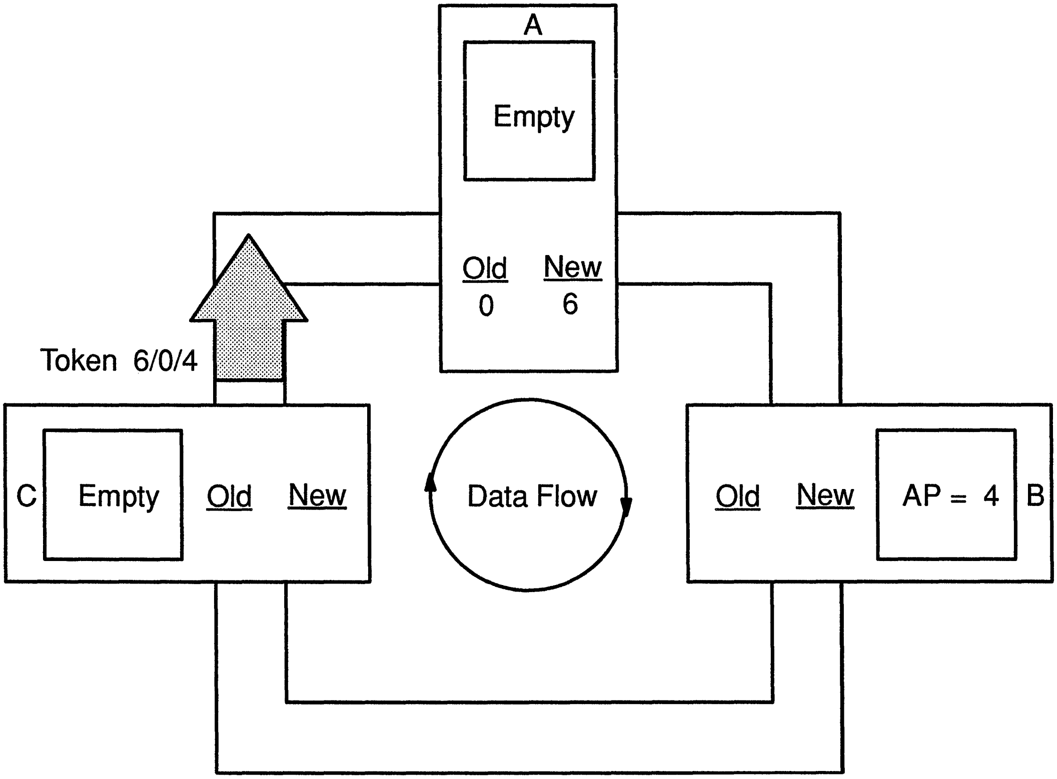
Figure 2–37. Station C Begins Stripping the Frame from the Ring



Example — Step 11

Figure 2–38 shows that once station C strips its frame, it releases a token with PI = 6 and activates its priority state machine. Because the PI is greater than the PR of the token, no stack operations are performed prior to token transmission.

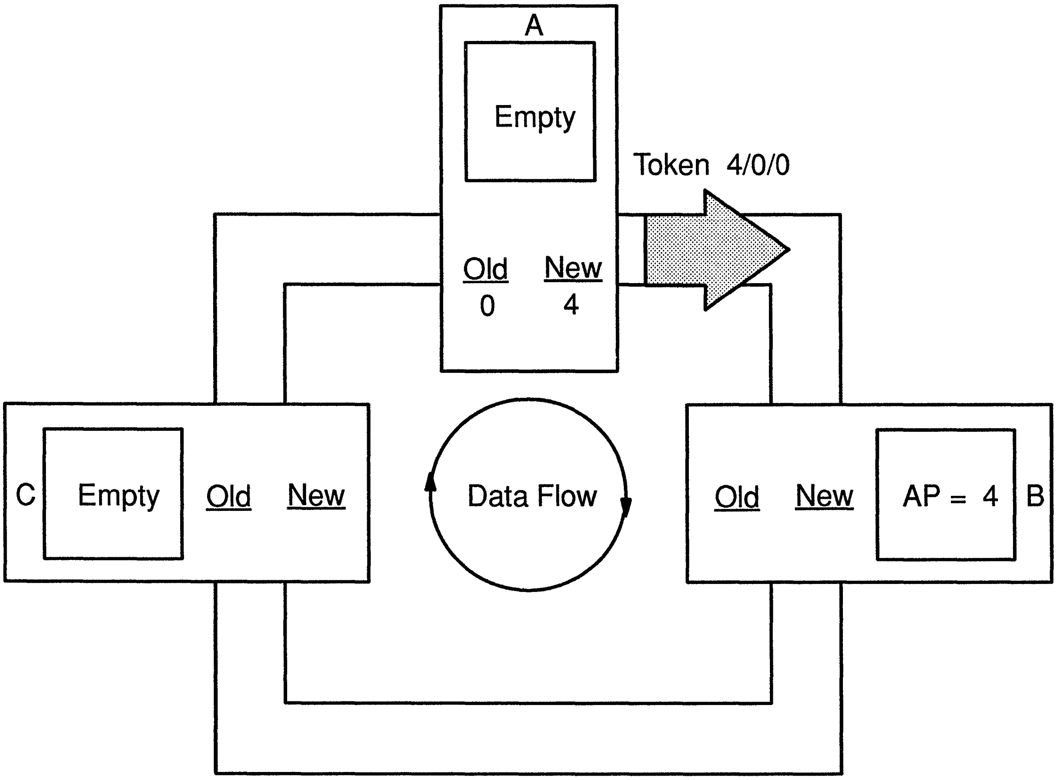
Figure 2–38. Station C Issues a New Token of Same Priority



Example — Step 12

Figure 2–39 shows that when station A receives the token, it is passed to the priority state machine, which is still active. Because the PI is greater than the PR, the state machine concludes that a station requests that a priority token less than the current priority level be circulated. Thus, the state machine performs a **replace** function on the stack (as can be determined by the stimuli conditions shown in Table 2–19), substitutes the PR for the PI, and transmits the token on the ring.

Figure 2–39. Station A Performs a Replace and Transmits the Token

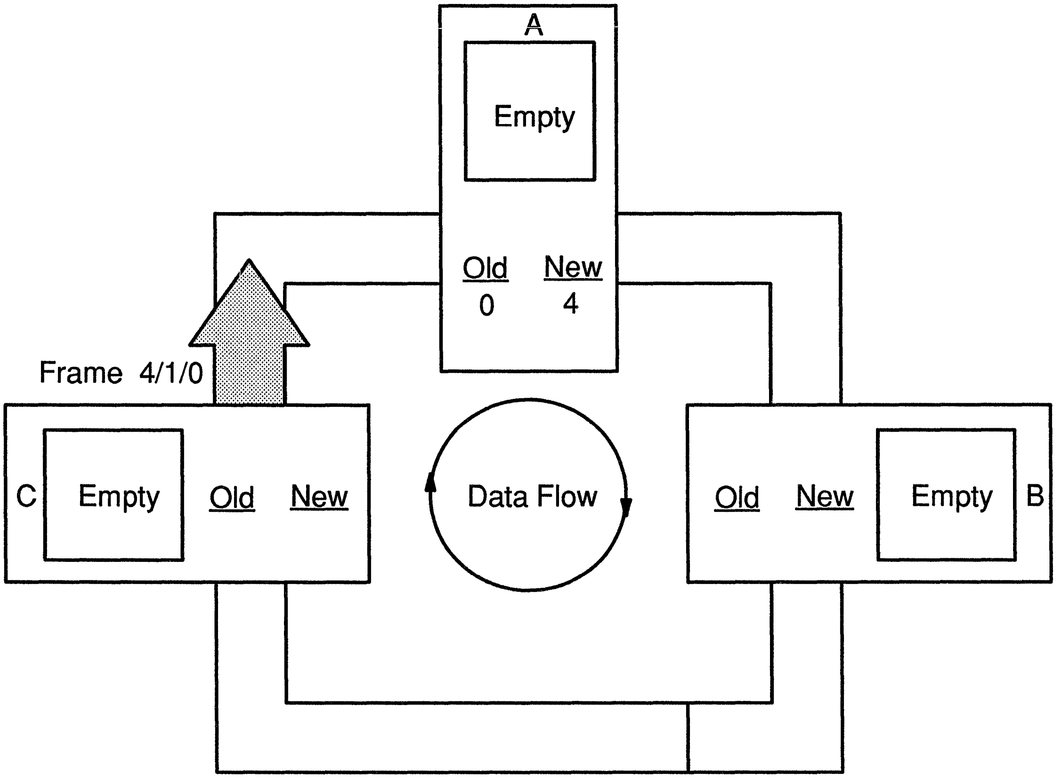


Example — Step 13

Figure 2–40 shows that station B can now capture the token because the PI is now equal to its queued frame's access priority. Thus the queued frame is transmitted on the ring.

Station C and A do not modify the PR of the frame as they currently do not have a frame queued for transmission.

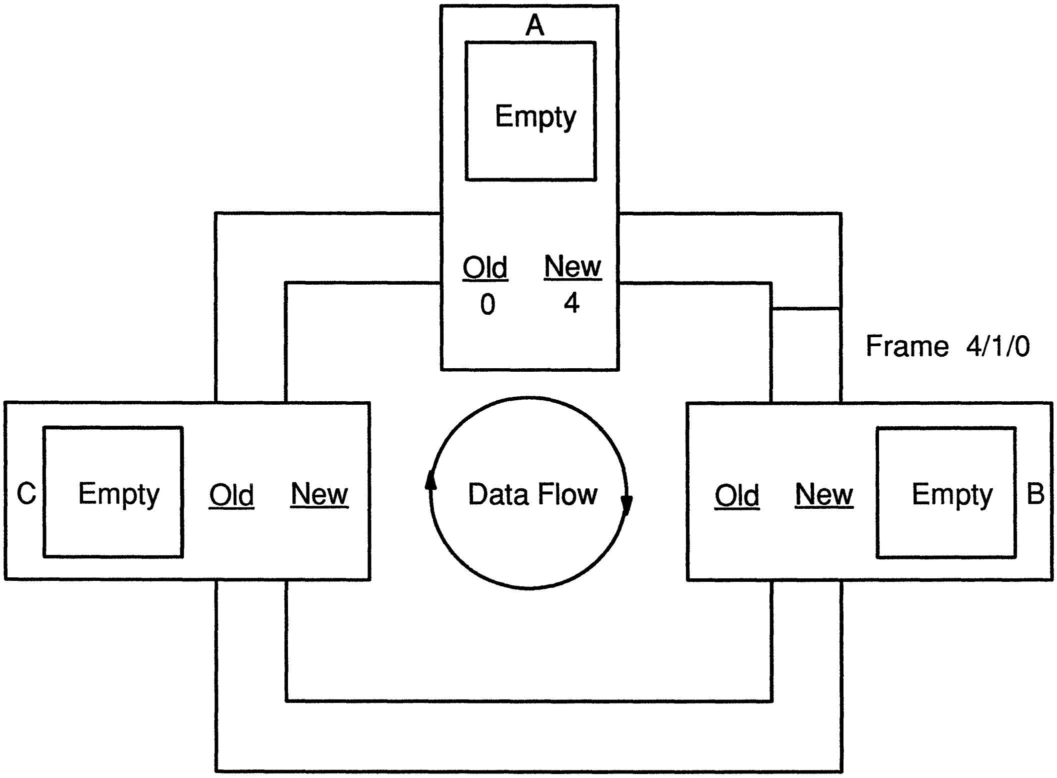
Figure 2–40. Station B Captures Token and Transmits Frame



Example — Step 14

Figure 2–41 shows that when the frame transmitted by station B has completed circulation, station B strips the frame from the ring.

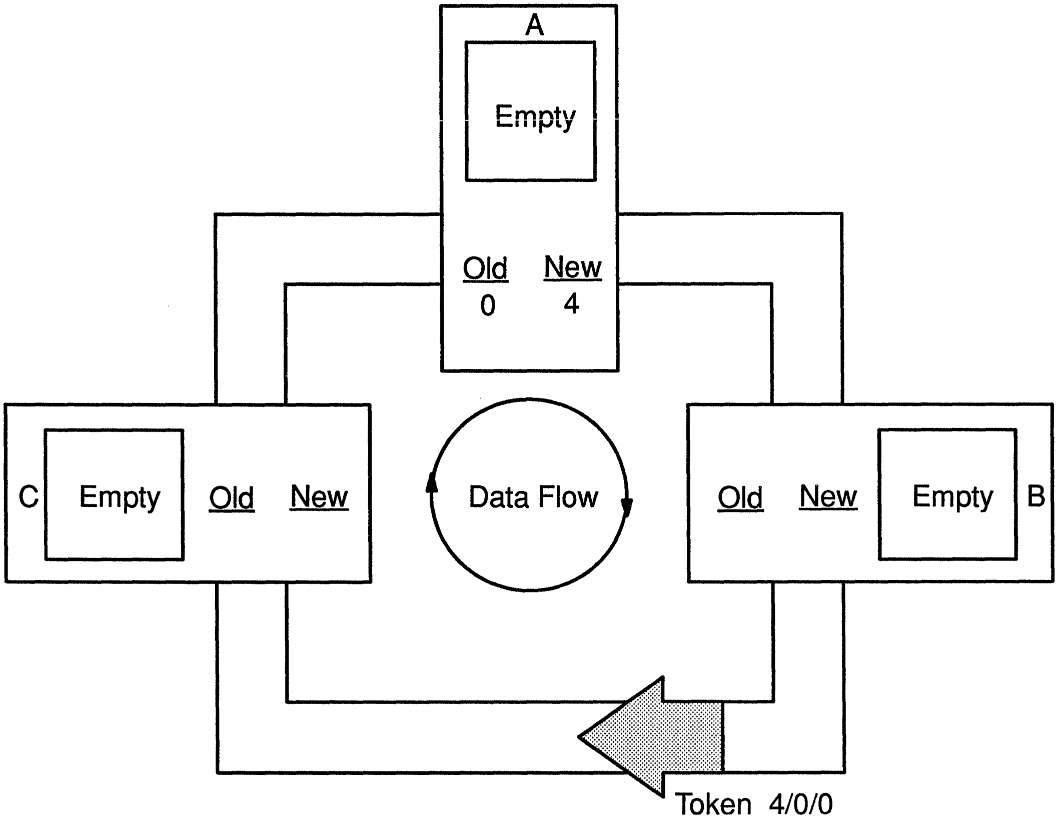
Figure 2–41. Station B Strips Its Frame from the Ring



Example — Step 15

Figure 2–42 shows that station B now issues a token of priority 4 as originally received. The priority state machine remains inactive because no other station requested a higher priority via the PR field of the frame.

Figure 2–42. Station B Issues a New Token at the Original Priority

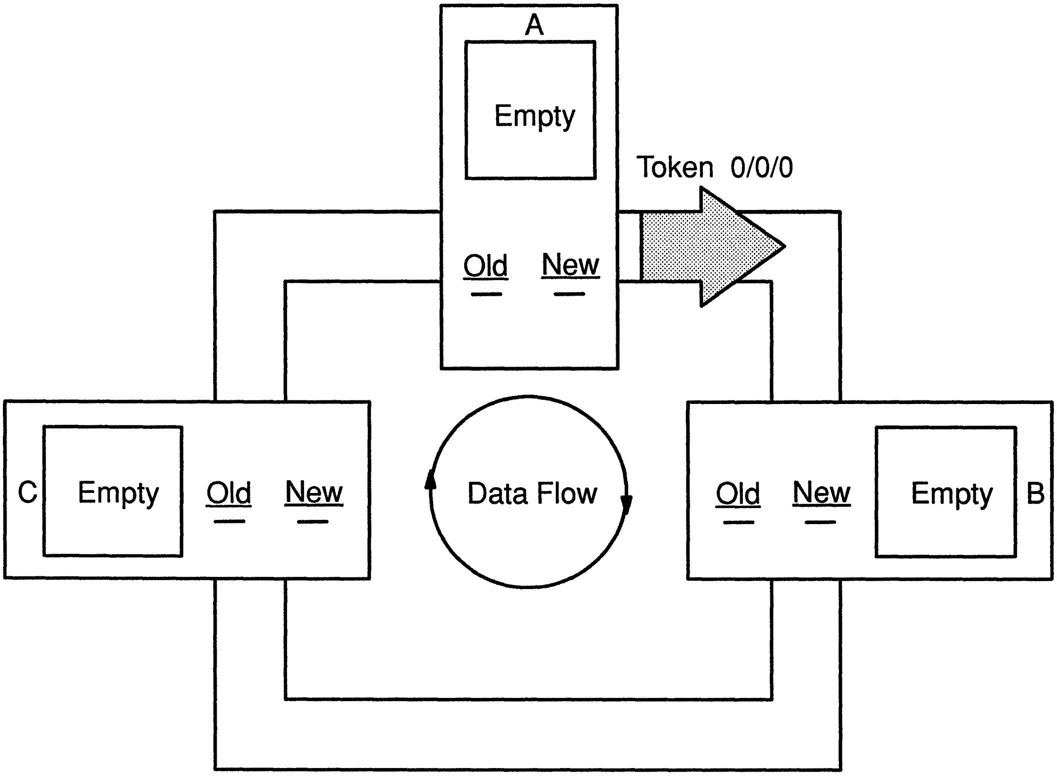


Example — Step 16

Figure 2–43 shows that when station A receives the priority 4 token as sent from station B, the token passes to the priority state machine. Because no other station requested a higher or lower priority level, the state machine performs a POP of its stacks and returns the token to its original priority-free state.

At this point, the ring is in a state similar to that shown in Figure 2–29.

Figure 2–43. Station A Pops Its Stacks and Issues Priority-Free Token



This exercise presents a basic example of the priority operation of the adapter. Note that each time the priority level of the token changes, each station on the ring is afforded equal opportunity to capture the token for frame transmission.

2.4.4.12 Soft Error Counting and Reporting

The adapter counts and reports soft errors that occur during normal operations. A soft error is an error condition that temporarily degrades system performance; however, the ring recovers by using the protocols of the adapter. The adapter counts these errors and can report them to both the attached product and the ring error monitor (section 2.4.1.1).

The error counters consists of two sets:

❑ **Soft error counters.** These are reported on the ring to a ring error monitor.

❑ **Attached product counters.** These are reported to the attached system through the system interface.

Although these counters count similar error conditions, separate sets of counters are kept due to the different way in which these counters are maintained.

2.4.4.12.1 Soft Error Counters

The soft error counters are incremented by one when

❑ The adapter detects a soft error corresponding to an error counter,

❑ The adapter is inserted,

❑ The adapter is in a normal state (normal state means that the adapter is not in the monitor contention process, not in the beacon process, not in the ring purge process), and

❑ The current value of the counter is not equal to 255.

When the adapter detects a soft error and increments any of the soft error counters, it starts the soft error report timer. When the soft error report timer expires, a report error MAC frame is queued for transmission and will be transmitted when the adapter is in a normal state. The report error MAC frame is defined in Table 2–20. The error counts transmitted in the report error MAC frame are subtracted from the soft error counters when the transmission is complete. The transmission of the report error MAC frame uses the assumed delivery process.

If a soft error occurs and the adapter enters the beacon auto-removal process, the soft error report timer is cancelled. Therefore, the soft error will not be reported until another soft error is detected after the auto-removal process has been completed.

The default value of the soft error report timer is 2.0 seconds. Its value can be changed with the receipt of an initialize ring station or change parameters MAC frame.

Table 2–20. Report Error MAC Frame

Frame Description				Subvectors		
M–V (Frame Name)	Destination Class	Source Address	Destination Address	R/O		Name
				XMT	RCV	
>29 Report error	>6	>0	F(REM)	A		>02 UNA
				A		>OB Phys. drop
				A		>2D Isolating error counts
				A		>2E Non-isolating error counts

<u>Key:</u> **A** always transmitted
R required
O optional
N not syntax checking

2.4.4.12.2 Attached Product Counters

The attached product error counters can be read by the attached system with the read error log command.

The attached product counters are incremented when

- ❑ A soft error is detected,
- ❑ The adapter is inserted,
- ❑ The adapter is in a normal state, and
- ❑ The value of the counter does not equal 255.

When an error is detected and an attached product counter is incremented from 254 to 255, the counter overflow bit is set in ring status. The attached product counters are reset when the counters are read by the attached system.

2.4.4.12.3 Isolating versus Non-Isolating Error Counters

The error counters maintained by the adapter are defined as **isolating** or **non-isolating**. Isolating error counters isolate errors to a transmitting adapter, a receiving adapter, and the components (cabling, wiring concentrators) between those two adapters. These errors are counted only by the first detecting adapter. Other adapters also detect these errors but are prevented from counting these errors by the EDI (error detected indicator) bit in the ending delimiter of the frame already being set to one by the detecting adapter.

Nonisolating error counters count errors that could have been caused by any other adapter on the ring (the fault cannot be isolated to a specific area of the ring).

2.4.4.12.4 Isolating Error Counters

The following are the isolating error counters:

- ❑ Line error counter
- ❑ Burst five error counter
- ❑ ARI/FCI set error counter

Table 2–21 defines these isolating error counters:

Table 2–21. Isolating Error Counters

Code	Description
Line error	The line error counter is contained in all adapter configurations. It is incremented no more than once per frame whenever: a) a frame is repeated or copied, or b) the error detected indicator (EDI) is zero in the incoming frame, or c) one of the following conditions exists: 1. A code violation exists between the starting delimiter (SDEL) and the ending delimiter (EDEL) of the frame. 2. A code violation exists in a token. 3. A frame check sequence (FCS) error exists.
Burst error	The burst error counter is contained in all adapter configurations and is incremented when the adapter detects the absence of transitions for five half-bit times between SDEL and EDEL. Only one adapter detects the burst five condition because the adapter that detects a burst four condition (four half-bit times without transitions) conditions its transmitter to transmit idles if the burst-five condition is detected.
ARI/FCI Set error	The ARI/FCI set error counter is incremented when an adapter receives more than one AMP or SMP MAC frame with ARI/FCI equal to zero, without first receiving an intervening AMP MAC frame. The counter indicates that the upstream adapter is unable to set its ARI/FCI bits in a frame that it has copied.

2.4.4.12.5 Nonisolating Error Counters

The following are the non-isolating error counters:

- ❑ Lost frame error counter
- ❑ Frame copied error counter
- ❑ Receive congestion error counter
- ❑ Token error counter

Table 2–22 defines the functions of these nonisolating error counters.

Table 2–22. Nonisolating Error Counters

Code	Description
Lost frame error	The lost frame error counter is contained in all adapter configurations and is incremented when an adapter is in transmit (stripping) mode and fails to receive the end of the frame it transmitted.
Frame copied	The frame copied error counter is contained in all configurations and is incremented when an adapter in the receive/repeat mode recognizes a frame addressed to its specific address, but finds the ARI bits not equal to 00 (possible line hit or duplicate address).
Receive congestion	The receive congestion error counter is contained in all adapter configurations and is incremented when an adapter in the repeat mode recognizes a frame addressed to its specific address, but has no buffer space available to copy the frame (Adapter congestion).
Token error	This one-byte counter is contained in Active Monitor Adapter configurations and is incremented when the Active Monitor function detects an error with the token protocol as follows: 1) A token with priority of nonzero and the monitor count bit equal to one. 2) A frame and the monitor count bit equal to one. 3) No token or frame is received within a 10 millisecond window. 4) The starting delimiter/token sequence has a code violation (in an area where code violations must not exist)

2.4.4.13 Token Ring Recovery

This section describes the error recovery mechanisms provided by the TMS380 token-ring protocol processes. Rather than focus on the individual processes, this discussion focuses on the interaction of these processes to effect recovery in the event of network faults.

Error conditions which may occur on a token-ring network can be classified as soft errors or hard errors:

- ❑ Hard errors prevent ring recovery protocols from restoring normal token protocols. The hard errors include streaming errors, frequency errors, signal loss errors, and internal hardware errors.

- ❑ Soft errors allow the ring recovery protocols to restore normal token protocols, but cause performance degradation due to disrupted network operation. Soft errors include line errors, lost frames, lost tokens, lost active monitor, corrupted tokens, circulating priority tokens or frames, delimiter errors, multiple monitors, and lost delimiters.

2.4.4.13.1 Hard Error Recovery

The following are the hard error types that require hard error recovery procedures to be invoked:

Streaming Error

There are two forms of streaming: bit streaming and frame streaming.

❑ Bit streaming removes (destroys) tokens and frames by writing over (repeat and transmit both occurring) or replacing (uncontrolled transmit) ring data.

❑ Frame streaming consists of the continuous transmission of tokens, abort sequences, or frames.

Both streaming types are detected by the adapter's active or standby monitor function and cause monitor contention to be entered.

Frequency Error

A frequency error is a condition in which the ring clock and the crystal clock frequency of the adapter differ by an excessive amount. Detection of this condition causes the adapter to enter the monitor contention process.

Signal Loss Error

A signal loss error occurs as the result of a broken ring, faulty wiring concentrator, transmitter malfunction, or receiver malfunction.

Internal Errors

Internal errors are detected by the adapter's hardware and/or firmware; they cause the detecting adapter to remove itself from the ring (automatic reconfiguration).

When hard errors occur on the ring, reconfiguration of the network is necessary to effect full recovery. Reconfiguration consists of removal or bypass of the faulty station(s) or cabling. Reconfiguration takes two forms: manual reconfiguration and automatic reconfiguration.

❑ Automatic reconfiguration begins when the adapter executes one of the beacon remove functions or when an internal hardware error is detected by the background diagnostics of the adapter.

❑ Manual reconfiguration is necessary if the automatic reconfiguration functions have failed to recover the ring's normal token protocol. The fault location can be easily isolated by the beacon process.

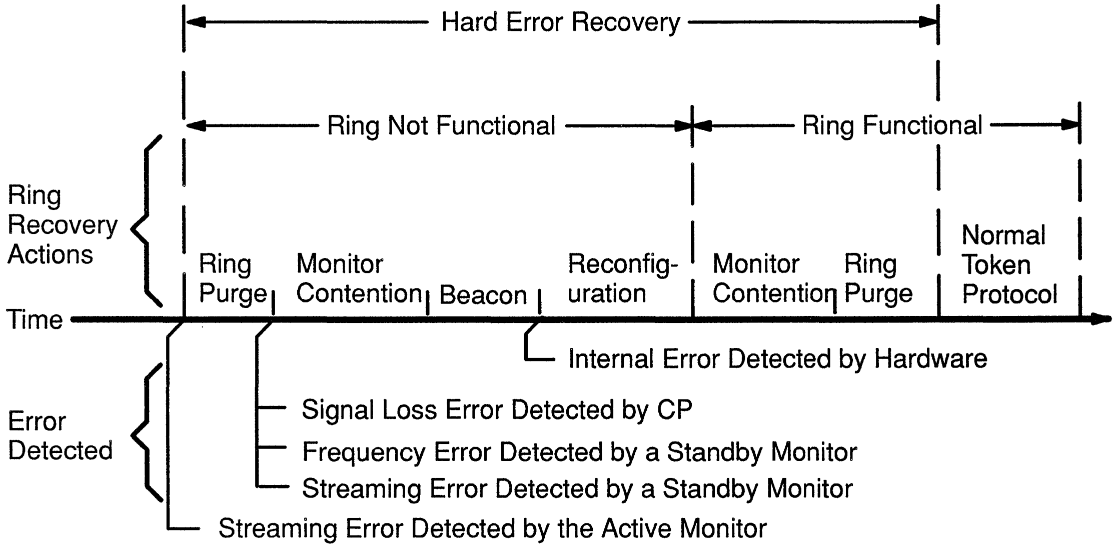
Hard Error Recovery Time Lines

Hard errors can be classified as solid (fault invokes reconfiguration) or as intermittent (fault is removed from the ring before reconfiguration is required).

Solid Hard Error Recovery Time Line

Figure 2–44 illustrates the actions to recover from a solid hard error. Note that reconfiguration (either auto or manual) is required to re-establish a functional ring.

Figure 2–44. Solid Hard Error Recovery Time Line


Hard Error Recovery
Ring Not Functional
Ring Functional
Ring Recovery Actions
Ring Purge
Monitor Contention
Beacon
Reconfig-uration
Monitor Contention
Ring Purge
Normal Token Protocol
Time
Internal Error Detected by Hardware
Error Detected
Signal Loss Error Detected by CP
Frequency Error Detected by a Standby Monitor
Streaming Error Detected by a Standby Monitor
Streaming Error Detected by the Active Monitor

The following list defines the terms used in Figure 2–45.

- ❑ **Streaming error** detection by the active monitor causes the adapter to enter ring purge, which fails, causing monitor contention, which also fails, causing beacon transmit mode. When the condition causing streaming is removed, the ring is recovered through the monitor contention and ring purge processes.
- ❑ **Frequency error** detection by the standby monitor causes the adapter to enter monitor contention, which, if it fails, causes beacon transmit mode (which requires reconfiguration). When the condition causing the frequency error is removed, the ring is recovered through the monitor contention and ring purge processes.
- ❑ **Streaming error** detection by the standby monitor causes the adapter to enter monitor contention, which fails, causing beacon transmit mode

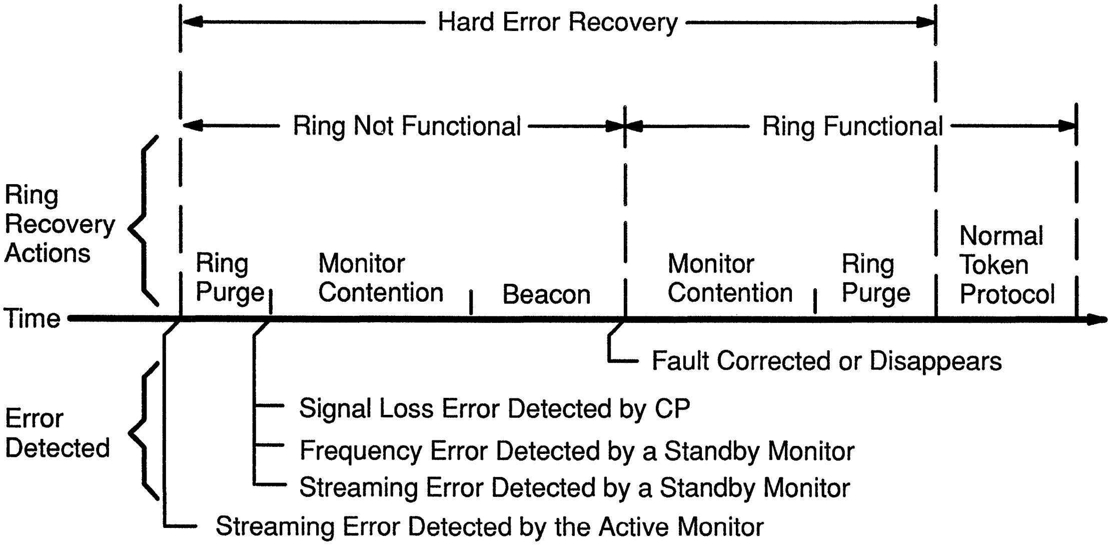
(which requires reconfiguration). When the condition causing the streaming error is removed, the ring is recovered through the monitor contention and ring purge processes.

- ❑ **Signal loss error** detection by the communications processor causes the adapter to enter monitor contention transmit mode, which, if it fails, causes beacon transmit mode to be entered. When the condition causing the signal loss error is removed, the ring is recovered through the monitor contention and ring purge processes.
- ❑ **Internal error** detection by the adapter hardware causes an adapter to deinsert from the ring.

Intermittent Hard Error Recovery Time Line

Figure 2–45 illustrates the actions taken to recover from intermittent hard errors. Note that these are the same processes as those used in solid hard error recovery except that the beacon process is terminated before reconfiguration is required.

Figure 2–45. Intermittent Hard Error Recovery Time Line


Hard Error Recovery
Ring Not Functional
Ring Functional
Ring Recovery Actions
Ring Purge
Monitor Contention
Beacon
Monitor Contention
Ring Purge
Normal Token Protocol
Time
Fault Corrected or Disappears
Error Detected
Signal Loss Error Detected by CP
Frequency Error Detected by a Standby Monitor
Streaming Error Detected by a Standby Monitor
Streaming Error Detected by the Active Monitor

Beacon Removal Functions

The beacon removal functions are executed when an adapter detects a sustained hard error. The execution of these functions causes the adapter to deinsert from the ring and execute an internal test. If this test is successful, the adapter reinserts. Otherwise, the adapter is closed and remains off-ring. If the latter occurs, the attached system is notified via ring status that a hard error has removed the adapter from the ring.

2.4.4.13.2 Soft Error Recovery

Four types of soft errors have been defined.

Type 1	These errors require no ring recovery function to be executed.
Type 2	These errors require the ring purge process to be executed.
Type 3	These errors require the monitor contention and ring purge processes to be executed.
Type 4	These errors require the beacon, monitor contention, and ring purge functions to be executed.

The detection and recovery of lost frames caused by soft error conditions are not performed by the adapter.

Type 1 Soft Errors

Type 1 errors consist of line errors, the multiple monitor condition, and the ARI/FCI set condition.

Line Errors. Each adapter checks each frame copied or repeated for a valid FCS or a Manchester code violation. Adapters detecting these errors set the EDI (Error Detected Indicator) to one in the frame or token's ending delimiter.

If the received EDI bit is equal to zero, the adapter increments its line error counter. If the received EDI bit is equal to one (line error previously detected by another adapter), the counter is not incremented.

Multiple monitors. When an active monitor receives a ring purge MAC frame or an AMP MAC frame that it did not transmit, it queues for transmission a report monitor error MAC frame with an error code indicating the multiple monitor error. It then becomes a standby monitor and enters repeat mode. Note that this action may leave the ring without an active monitor. Another standby monitor will enter the monitor contention process in this case.

ARI/FCI set error. Reception of more than one AMP or SMP MAC frame with ARI/FCI equal to zero, without first receiving an intervening AMP MAC frame, causes the adapter to

❑ Increment the ARI/FCI set error counter, and

❑ Terminate the ring poll process (does not queue the SMP MAC frame).

This process leaves the adapter downstream of a malfunctioning adapter with an incorrect UNA (when the ring poll process terminates), but adapters downstream from the active monitor to the defective adapter will have a correct UNA.

When an adapter, upon receiving a frame, is unable to set the ARI/FCI bits, the following LAN functions do not operate correctly:

- ☐ Higher-level protocols may not work correctly because of the inability of the insertion process to detect duplicate addresses.
- ☐ A beacon transmitter may identify an incorrect upstream adapter.
- ☐ An adapter MAC frame requiring assured delivery will not be assured when the destination adapter fails to set the ARI/FCI bits.

Type 2 Soft Error

Type 2 soft errors consist of the burst-5 error, lost frame, multiple token, corrupted token, lost token, lost delimiter, circulating token, or circulating frame error conditions. These error conditions cause the active monitor to execute its ring purge function and may cause frames to be lost.

Burst-5 error. An adapter detects the burst-5 condition when five half-bits of Manchester-coded data are received without a phase change.

A type 2 soft error is detected when the burst error is long enough to remove the token or frame from the ring, but not long enough (due to signal loss) to start a hard error recovery.

Lost frames. When a transmitting adapter has transmitted the physical trailer, the adapter strips the ring data until EDEL is detected or 4.1 milliseconds expires. If the 4.1-ms timer expires, then the adapter

- ☐ Enters repeat mode without generating a token, and
- ☐ Increments its lost frame error counter.

This error causes the active monitor to detect a lost token and to perform a ring purge to restore normal token protocols.

Corrupted token. When an adapter has one or more frames to transmit and receives a token but does not detect an EDEL after the access control field, a corrupted token has been detected. The adapter transmits an abort delimiter (an SDEL/EDEL sequence), queues for transmission the frame that was being transmitted, and does not generate a token.

This error causes the active monitor to detect a lost token and to restore the ring through the ring purge process.

Lost token. When the active monitor fails to detect a token or a frame physical header once every 10 milliseconds, the adapter increments the token error counter and restores the ring through the ring purge process.

Circulating frame or priority token/multiple monitor. Each time an adapter transmits a frame or a priority token, it sets the monitor bit (AC bit

4) to zero. Each time the active monitor repeats a frame or a priority token, it examines the AC bit 4.

- ❑ If equal to zero, it sets AC bit 4 to one.
- ❑ If equal to one, then a circulating frame, a circulating priority token, or a multiple active monitor condition exists. The active monitor increments its token error counter and restores the ring through the ring purge process.

Type 3 Soft Errors

The type 3 soft error may be caused by the nonexistence of an active monitor and initiates the monitor contention and ring purge processes.

These error conditions or their resultant recovery techniques may cause frames to be lost.

Lost monitor. The standby monitors monitor the ring for good tokens to be received at least once every 2.6 seconds. The standby monitors also monitor to assure that a ring poll process is executed at least once every 15 seconds.

If either of these times expire, the adapter assumes the ring's active monitor is not functional or not present and enters the monitor contention process in contention transmit mode.

Frequency error. The standby monitors' detection of a frequency error causes the adapter to enter monitor contention in contention transmit mode. This failure may be resolved by soft error recovery or may cause the hard error recovery function to be executed.

Type 4 Soft Errors

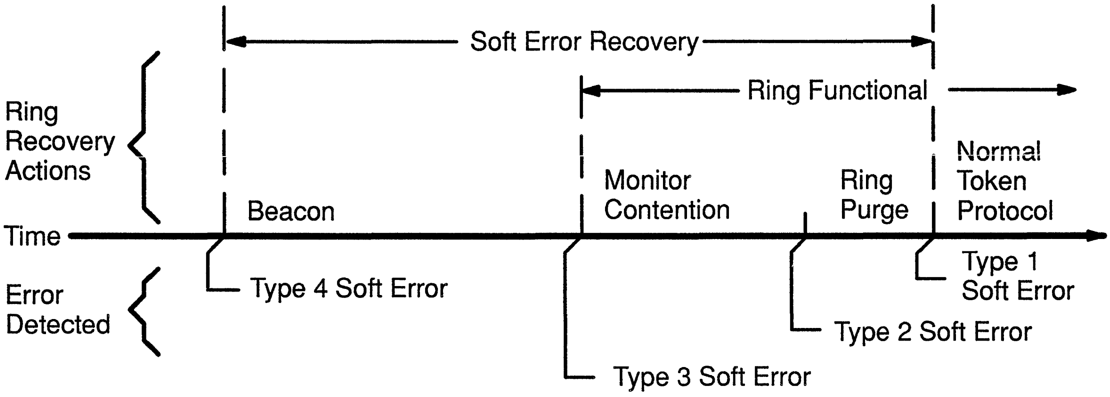
The type 4 soft error is caused when monitor contention cannot be resolved. This error condition may also require hard error recovery.

This error condition or its resultant recovery may cause frames to be lost.

Soft Error Recovery Time Line

Figure 2–46 illustrates the recovery actions that take place as a function of time when one of the soft errors is detected.

Figure 2–46. Soft Error Recovery Time Line



2.4.4.13.3 Hard Error Recovery Examples

This section presents several examples of the hard error recovery process as previously described. The following examples are provided:

- ❑ A signal loss is detected.
- ❑ Transmit streaming is detected (not claim token MAC frames).
- ❑ Receive streaming is detected (not claim token MAC frames).
- ❑ Transmit streaming is detected (claim token MAC frames).
- ❑ Two adapters are transmitting streaming data (transmitter fault).
- ❑ Two adapters are receiving streaming data (receiver fault).

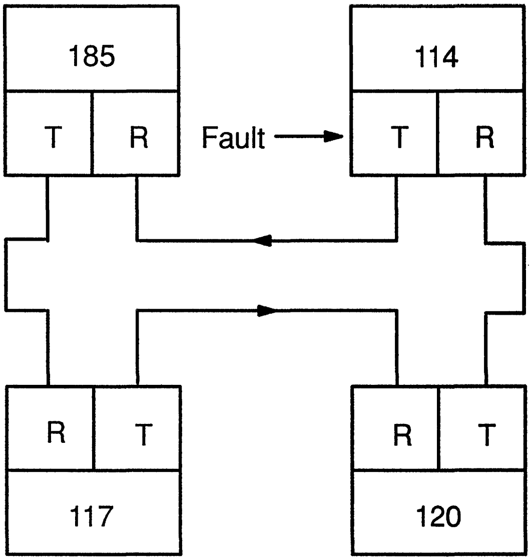
Signal Loss

Three examples of signal loss recovery are presented.

Case 1 — Transmitter Malfunction

Adapter 185 detects a signal loss error caused by a transmitter malfunction in adapter 114.

Figure 2–47. Transmitter Fault in Adapter 114

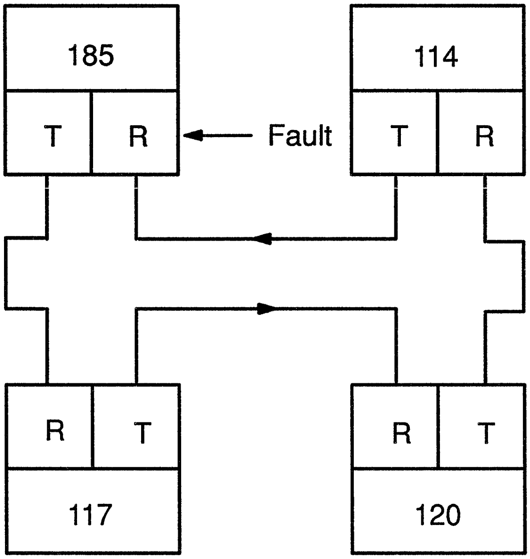


1) Adapter 185 detects a burst-4 error and transmits idles.
2) Adapter 185 detects a burst-5 error.
3) The state of adapters 114, 120, and 117 will vary, depending upon which adapter's token protocol timers expire first.
4) Adapter 185 detects the signal loss error, causing it to enter contention-transmit mode.
5) If monitor contention is not resolved (times out), adapter 185 enters beacon-transmit mode and transmits a beacon MAC frame with a beacon-type subvector equal to 0002 (signal loss).
6) Adapters 114, 120, and 117 enter beacon-repeat mode upon receiving the beacon MAC frame from 185 with a lower error code.
7) Adapter 185 remains in beacon-transmit mode until the signal is restored to adapter 185 by the removal of adapter 114 through the beacon-receive auto removal function.
8) Adapter 185 detects its own beacon MAC frame and enters the monitor contention process to recover the ring.

Case 2 — Receiver Malfunction

Adapter 185 detects a signal loss error caused by its own receiver malfunction.

Figure 2–48. Receiver Fault in Adapter 185



1) Adapter 185 detects a burst-4 error and transmits idles.
2) Adapter 185 detects a burst-5 error.
3) The state of adapters 114, 120 and 117 will vary, depending upon which adapter's token protocol timers expire first.
4) Adapter 185 detects the signal loss error, causing it to enter contention-transmit mode.
5) If Adapter 185 cannot resolve monitor contention, it enters beacon-transmit mode and transmits a beacon MAC frame with a beacon-type subvector equal to 0002 (signal loss).
6) Adapters 114, 120, and 117 enter beacon-repeat mode upon receiving the beacon MAC frame from 185 with a lower error code.
7) Adapter 185 remains in beacon-transmit mode until it is removed from the ring by the beacon-transmit remove function.

Note:

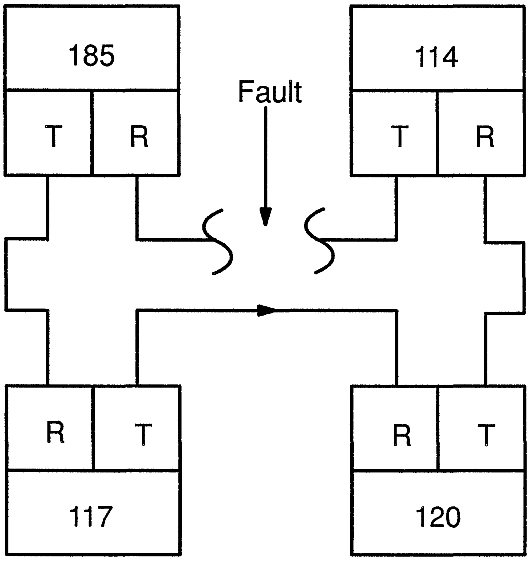
Adapter 114 did execute the beacon-receive remove function, but reinserted into the ring because the adapter's tests were successful.

8) Adapter 117 receives no beacon frames for 200 milliseconds and so starts the monitor contention process to recover the ring.

Case 3 — Broken Ring

Adapter 185 detects a signal loss caused by a ring break within the wiring concentrator between adapters 185 and 114.

Figure 2–49. Broken Ring Fault



1) Adapter 185 detects a burst-4 error and transmits idles.

2) Adapter 185 detects a burst-5 error.

3) The state of adapters 114, 120, and 117 will vary, depending upon which adapter's token protocol timers expire first.

4) Adapter 185 detects the signal loss error, causing it to enter contention-transmit mode.

5) If adapter 185 does not resolve monitor contention (timeout), it enters beacon-transmit mode and transmits a beacon MAC frame with a beacon-type subvector equal to 0002 (signal loss).

6) Adapters 114, 120, and 117 enter beacon-repeat mode upon receiving the beacon MAC frame from 185 with a lower error code.

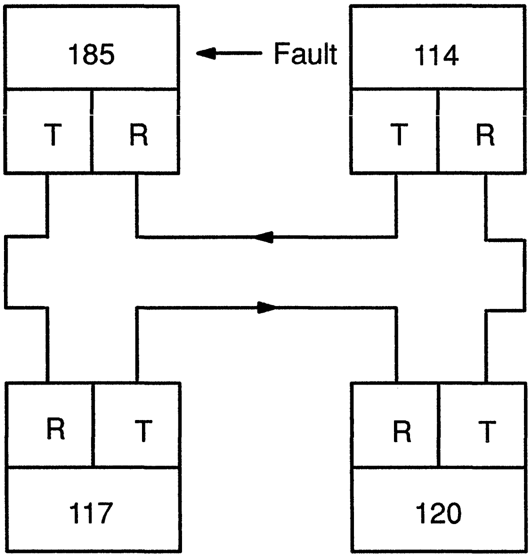
7) Adapter 185 remains in beacon-transmit mode until the broken ring segment is repaired.

8) Adapter 185 receives its own beacon MAC frame and starts the monitor contention process to recover the ring.

Case 4 — Transmit Data Streaming

Adapter 185 is transmitting data without regard to ring protocol (streaming data of any kind except claim token MAC frames). Adapter 185 does not respond to ring purge, but does recognize the beacon MAC frame.

Figure 2–50. Transmitter Streaming Fault in Adapter 185



1) Depending on when their timers expire, any combination of adapters may enter monitor contention repeat or transmit mode and start their internal watchdog timers for monitor contention (1 second).

2) Depending on when their contention timers expire, any combination of adapters may enter beacon-transmit mode and transmit a beacon MAC frame with a beacon-type subvector equal to 0003 (adapters 117 never received a claim token MAC frame) or 0004 (adapters received a claim token MAC frame).

3) Adapter 117 detects a streaming data error, enters beacon-transmit mode, and transmits a beacon MAC frame with an error code of 0003 and a UNA identifying adapter 185.

4) Adapter 114 and 120 enter beacon-repeat mode upon receiving the beacon frame from adapter 117 with a lower error code.

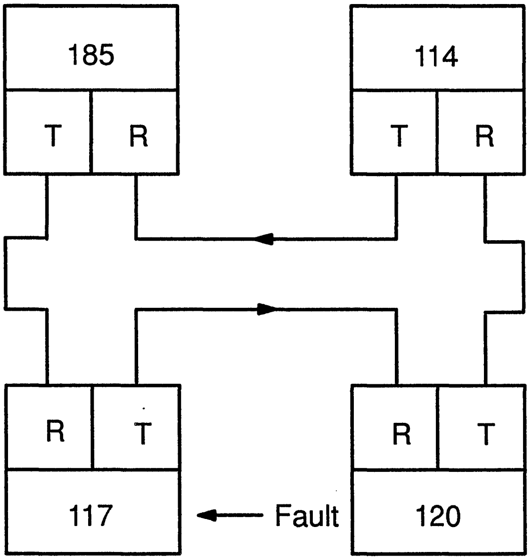
5) Adapter 117 remains in beacon-transmit mode until the streaming adapter 185 is removed through the beacon-receive remove function.

6) Adapter 117 detects its own beacon frames and enters the monitor contention process to recover the ring.

Case 5 — Receive Data Streaming

Adapter 117 is receiving a signal without recognizing token or frame protocols (streaming data), but does detect a streaming error.

Figure 2–51. Adapter 117 Detects a Streaming Error



1) Depending on when their timers expire, any combination of adapters may enter monitor contention repeat or transmit mode and start their internal watchdog timers for monitor contention (1 second).

2) Depending on when their contention timers expire, any combination of adapters may enter beacon-transmit mode and transmit a beacon MAC frame with a beacon-type subvector equal to >0003 (adapters never received a claim token MAC frame) or >0004 (adapters received a claim token MAC frame).

3) Adapter 117 detects a streaming data error, enters beacon-transmit mode and transmits a beacon MAC frame with an error code of >0003 and a UNA-identifying adapter 185.

4) Adapters 114 and 120 enter beacon-repeat mode upon receiving the beacon frame from adapter 117 with a lower error code.

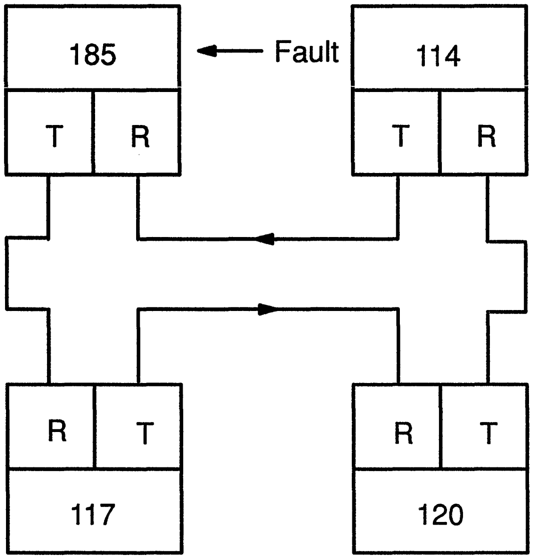
5) Adapter 117 remains in beacon-transmit mode until an internal 26-second timer expires; adapter 117 is then removed by the beacon-transmit remove function.

6) Adapter 120 experiences 200 milliseconds with no beacon frames and enters the monitor contention process to recover the ring.

Case 6 — Transmit Monitor Contention Streaming

Adapter 185 is transmitting claim token MAC frames without regard to ring protocol (streaming), and does not recognize the ring purge MAC frame (normal for a monitor contention transmitter), but receives the beacon MAC frame.

Figure 2–52. Adapter 185 Streaming Claim Token Frames



1) Adapters 114, 120, 117 all enter monitor contention repeat mode and start their watchdog timers (1 second), which limit the duration of the monitor contention process.

2) Depending on when their timers expire, any combination of adapters may enter beacon-transmit mode and transmit a beacon MAC frame with a beacon-type subvector (error code) equal to >04 (monitor contention MAC frames have been received).

3) Adapter 117 detects a streaming claim token MAC frame error, enters beacon-transmit mode, and transmits a beacon MAC frame with an error code of >0004 and a UNA-identifying adapter 185.

4) Adapters 114 and 120 enter beacon-repeat mode upon receiving adapter 117's beacon MAC frame of an equal error code.

5) Adapter 117 remains in beacon-transmit mode until the streaming adapter 185 is removed through the beacon-receive remove function.

6) Adapter 117 detects its own beacon frames and enters the monitor contention process to recover the ring.

2.4.5 IEEE 802.2 LLC Services

This section describes the LLC services presented by the TMS380 when the IEEE 802.22 logical link control (LLC) software is resident on the adapter.

Because of its use by IBM and other major implementers of communications protocols, logical link control software has become an important factor on the token ring. LLC is the cornerstone on which IBM's Netbios and APPC, and MAP/TOP's OSI protocols are based.

LLC software executing on the TMS380 adapter has several advantages over LLC on the attached system. First, with the LLC on the TMS380, a user can guarantee the same LLC across an entire product line. For host-based LLC, different code would have to be developed for different hosts, thus introducing the possibility of incompatibilities. TI has provided on the TMS380 a single version of LLC for compatibility and portability across all product lines, including different host processors and operating systems, such as MS-DOS, UNIX, and VMS. Thus, equipment developers using the TMS380 are spared significant development costs through simplified system integration. With LLC on the TMS380, the LAN protocols are independent of the attached product and operating system.

A second advantage of having the LLC on the adapter is that system performance can be increased up to 100% over host-based solutions. This increased performance results largely from the decreased data movement. With host-based LLC, received frames must be transferred to the attached system, an acknowledgement packet built into the attached system, and the acknowledgement packet transferred to the adapter for transmission. With LLC in the adapter, received frames are examined while still in the adapter, and acknowledgement packets are built and sent from within the adapter. This decreased data movement increases overall network throughput.

Another factor in performance improvement is increased available processing time for the attached system. When the attached system is relieved of the LLC protocols, more bandwidth is available for data processing. Thus, frames are queued to the adapter for transmission faster than when the LLC is host-resident. Because the LLC protocols can be handled on the adapter, the attached system is freed from time consuming operations such as sequencing, acknowledgements, link session control, and automatic retries. Therefore, more of the attached system's processor power and memory space is available for user functions.

By putting 802.2 logical link control on the TMS380 Communications Processor, TI has increased performance, confidence of compatibility, portability, and ease of use of the protocols.

2.4.5.1 Logical Link Control

This section describes the logical link control functions and parameters necessary to use the LLC programming interface. For more detailed information on the LLC protocols, refer to the *IBM Token-Ring Architecture Reference* (IBM 1987), and the IEEE/ANSI Standard 802.2–1985.

The IEEE 802.2 logical link control sublayer is the top sublayer in the data link layer of the OSI reference model. The LLC sublayer is common to all the IEEE 802 medium access control layers and provides services for the network layer protocol. There are currently three types of logical link control operation defined by the IEEE 802 standard: type 1, type 2, and type 3. They satisfy a broad range of applications.

2.4.5.1.1 Type 1—Connectionless LLC

Type 1 LLC provides a connectionless data link service. This type of protocol provides a data link with minimum protocol complexity. Type 1 LLC could be used where the upper layers provide sufficient error detection and recovery, and the user does not wish to duplicate these functions at the data link layer. Type 1 LLC could be used in an application where it is not necessary to guarantee all data link layer transmissions. Type 1 LLC also includes the protocols for determining which type of LLC protocol a node supports. Type 1 is used in the MAP/TOP implementation of OSI protocols.

When using Type 1 communications, each node must open a service access point (SAP) through which communication takes place. A separate SAP is opened for each protocol stack running above the LLC in a node. The SAP is an identifier for the LLC, a pointer to the application above it. When a Sap is opened, the attached system indicates the SAP_VALUE which is to be used. This SAP_VALUE is placed in the source SAP location of each frame sent from this SAP, and is compared against the destination SAP in received frames. In turn, then the SAP is opened, the adapter provides a STATION_ID to the attached system.This STATION_ID is used for communications between the attached system and the adapter. It is used in commands in both directions between the adapter and attached system as an identifier. The STATION_ID consists of two bytes. The first byte indicates a SAP, and the second indicates a link station. Link stations will be discussed later. The adapter assigns the STATION_ID in a round robin fashion. For example, the first SAP opened will have the STATION_ID of >0100.

When a SAP is opened, the user has several configuration options for that SAP. The first is a choice of whether the adapter will respond to XID command frames for this SAP, or whether these frames should be passed on to the host. This is controlled by the XID_HANDLER bit in the OPEN.SAP options. See the description of the OPEN.SAP command for more details.

If the attached system selects to receive XID frames, then all received XID commands are passed to the attached system. Conversely, if the SAP is opened such that the adapter will respond to XID commands, then the adapter will respond to all XID commands. All XID response frames received will be passed to the attached system regardless of the state of the XID_HANDLER bit.

The other option a user has when opening a SAP is whether the group SAP corresponding to this SAP should be opened. A group SAP is indicated by a one in the least significant bit of the SAP_VALUE. A group SAP can be used in the destination SAP field of a frame to send frames to more than one SAP at a node. When a frame to a group SAP is received, the adapter duplicates the frame and issues a RECEIVE.PENDING interrupt to each of the individual SAPs belonging to the group SAP.

Once the SAP has been opened, the user of that SAP can transmit and receive any type 1 frame through that SAP. Type 1 frames are listed below. The frames are described later.

❑ XID

❑ TEST

❑ UI

All communication directly through the SAP is connectionless.

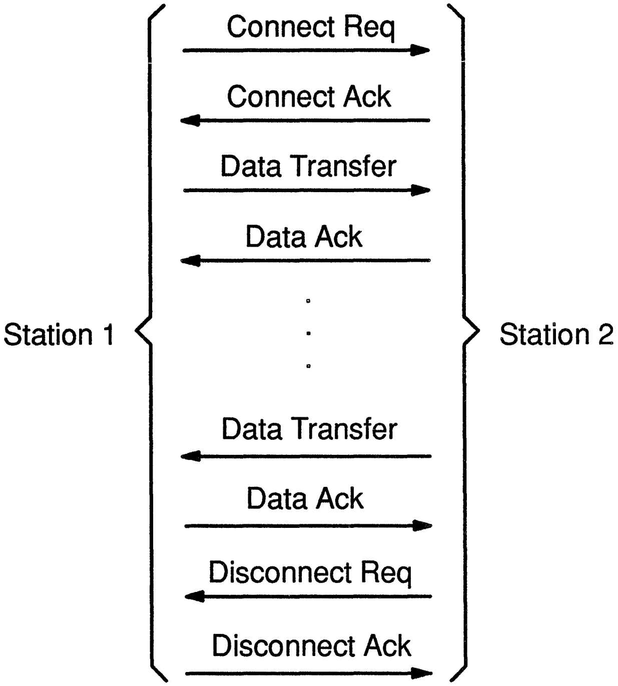
2.4.5.1.2 Type 2—Connection-Oriented LLC

For connection-oriented services, the attached system must open further connections after opening the SAP.

IEEE 802.2 type 2 LLC provides a connection-oriented data link service. Type 2 LLC is similar to HDLC protocols in use today. Type 2 LLC guarantees the delivery of all data link transmissions with sequencing, acknowledgements, and automatic retries. With type 2 LLC, connections are established between nodes wishing to communicate prior to any data transmissions. These are referred to as link stations. Type 2 LLC is the data link layer of IBM's protocols for local area networks.

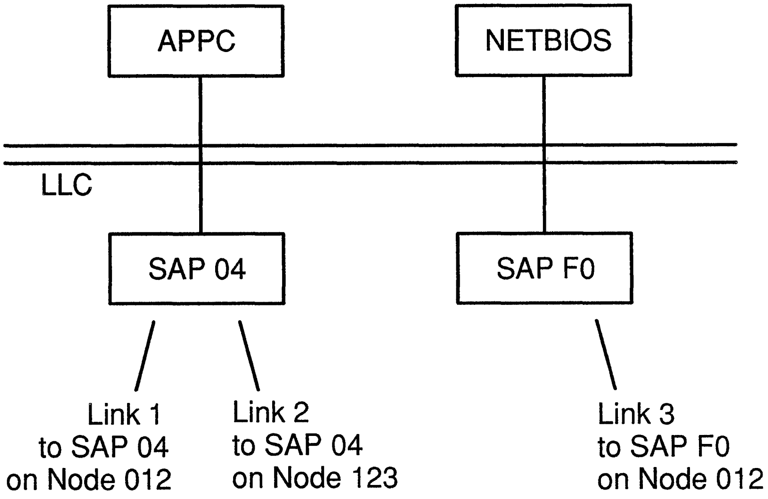
For illustration, a typical LLC type 2 session is shown in Figure 2–53. A station wishing to communicate (station 1) sends out a connection request frame to the station with which it wants to communicate (station 2). If station 2 is able (has the resources, is authorized) to establish communications, then station 2 returns a positive acknowledgement to the connection request. Otherwise, a negative acknowledgement is returned and no communication link is established. Assuming that the acknowledgement is positive, the link is established, and data transfer single or multiple, can take place in either direction. All data is acknowledged; however, not every packet need be acknowledged with a separate acknowledgement. A single acknowledgement may acknowledgement multiple packets, and acknowledgements can be piggybacked onto data packets. Once all data has been transferred, either station can send a disconnect request to close the link. This frees resources in both stations for other communications.

Figure 2–53. A Typical LLC Session



To establish communications for type 2 operation, the attached system must open a SAP and then open a link station associated with that SAP. The link station creates a link from the SAP in this node to another SAP in a different node. One link station can be associated with only one local SAP and only one remote SAP on one remote node. A SAP may have multiple link stations associated with it. This hierarchy is illustrated in Figure 2–54.

Figure 2–54. SAP/Link Station Association



As with the opening SAPs, when the attached system opens a link station, the adapter assigns a STATION_ID for use in commands between the attached system and the adapter. The first byte of this STATION_ID will be the SAP STATION_ID, and the second will be the link station identifier. That is, a STATION_ID of >0102 indicates the second link station opened, and indicates that it was opened under the first SAP.

2.4.5.1.3 *Type 3—Connectionless-Acknowledged LLC*

The third type of LLC is specifically intended for use in manufacturing applications. Type 3 is known as connectionless acknowledged service. Type 3 supports a protocol where packets are sent and acknowledged one at a time, without establishing the link stations of type 2. This provides a certainty of delivery without the overhead of a type 2 link station. For traffic that tends to be bursts of one frame, this type of protocol is suitable. The TMS380 LLC does not implement type 3 service.

2.4.5.2 LLC Frames

The following is a description of the frames used for communications with LLC. The LLC frame has the format shown in Figure 2–55.

Figure 2–55. LLC Frame

MAC Header	DSAP	SSAP	LLC Control	Data	MAC Trailer

The MAC Header consists of the starting delimiter (SDEL), access control (AC), frame control (FC), destination address (DA), source address (SA), and routing information (RI) fields. The MAC Trailer consists of the frame check sequence (FCS) and ending delimiter (EDEL) fields. The LLC header consists of the destination SAP (DSAP), the source SAP (SSAP), and the LLC control fields. These fields are described below:

DSAP This byte indicates the SAP to which the frame is destined. The receiving node copies the frame because of an address match on the destination address (DA), and then routes the frame to the appropriate attached system software according to the DSAP field value.

SSAP This byte indicates the SAP from which the frame originated. By examining the SSAP and source address (SA) fields, the adapter can determine to which link station under the DSAP the frame is destined. Since the source SAP must be an individual SAP, the least significant bit of the SSAP field is used to distinguish commands from responses. If this bit is zero, the frame is a command. If the bit is one, the frame is a response.

LLC Control This field is either one or two bytes in length. For unnumbered LLC frames, it is one byte; for numbered frames, it is two bytes. This field is the command field of the LLC. It determines what type of LLC frame this is. The LLC control field can take on the values described in the following pages. For more information on these frames, see the IEEE/ANSI Standard 802.2–1985 and the *IBM Token-Ring Network Architecture Reference* (IBM, 1987).

Unnumbered information. The unnumbered information (UI) frame is the means of transferring data in a type 1 environment. For this frame, the LLC control field takes on the binary value 00000011.

Exchange identification. The exchange ID (XID) is a frame used by LLC to identify the properties of a SAP and request the properties of a remote SAP. This frame identifies the type(s) of LLC that are supported by a SAP, and also identifies the receive window size (N3), which is discussed in section 2.4.5.4. The LLC control field for this frame contains the binary value 101p1111. The p value is the poll/final bit of LLC. The poll/final bit is used to check that the remote station is responding to frames. If the poll bit in a command is set to one, then the final bit (same bit position) in the response to that command must also be set to one. This mechanism can be used to match a response with the polled command.The least significant bit of the SSAP field will be zero, indicating a command, in the command XID. The bit will be a one in the response XID. For more information on the poll/final bit and the XID command, see the *IBM Token-Ring Network Architecture Reference* (IBM, 1987) and the IEEE/ANSI Standard 802.2–1985.

Test. Sending the test command frame causes the remote station to send a test response. Any data sent in a test command is echoed in the test response. For the test frame, the LLC control field takes on the value 111p0011, where the p bit is the poll/final bit. The least significant bit of the SSAP field will be zero to indicate a command, or or one to indicate a response, as appropriate.

Set asynchronous balanced mode extended. The set asynchronous balanced mode extended (SABME) frame is sent to initiate a link station between two SAPs. This frame is sent as a result of a CONNECT.STATION command issued by the attached system. The value of the LLC control field is 011p1111.

Disconnect. The disconnect command (DISC) is sent to request that a link between two SAPs be shut down. This frame is sent as a result of a CLOSE.STATION command from the attached system, or various error conditions in the link station. The LLC control field for the DISC is 010p0011.

Unnumbered acknowledgement. The unnumbered acknowledgement (UA) is a response sent to positively acknowledge received unnumbered commands. The UA is sent as a response to the SABME and DISC commands. The LLC control field for the UA response is 011f0011. The f bit is the poll/final bit described earlier.

Disconnected mode. The disconnected mode (DM) frame is a response sent to negatively acknowledge received unnumbered commands. The DM is sent as a response to the SABME when resources are not available to establish the requested link. The DM is sent also as a response to the DISC command when the receiving adapter is in the disconnected mode. This usually has resulted from error conditions in the link station. The LLC control field for the DM response is 000f1111.

Frame reject. The frame reject (FRMR) is a response sent when an illegal frame is received. An illegal frame is one of the following:

1) An information field in a UA, DM, SABME, or DISC frame, none of which allows information fields.

2) The final bit set to one when no poll bit was set.

3) An unsolicited UA frame.

4) An invalid N(R) from the remote station.

5) An unexpected N(S) from the remote station.

The LLC control field for the FRMR response is 100f0111.

Information frame. The I-frame is used to transfer data in an LLC type 2 connection. Each I-frame contains a send sequence number, N(S), that is the sequence number of this I-frame. It also contains a received sequence number, N(R), which has the same meaning as in the RR and RNR frames. Thus, acknowledgments can be piggybacked on I-frames. The format of the I-frame is shown below. Note that bit 7 of the control field set to zero indicates an I-frame

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	N(S)							0	N(R)							p/f

Receive ready. The receive ready (RR) response is sent to acknowledge receipt of an I-frame and indicates that the receiving adapter is ready to receive additional frames. Contained within the RR frame is the N(R) number, which is the sequence number of the I-frame that the station expects to receive next. That is, an RR frame with N(R) = 3 acknowledges all I-frames through the I-frame with sequence number 2. The RR frame can also be used without the receipt of an I-frame to periodically check that the link station is still functioning. This process is known as checkpointing and is described later. The format of the RR frame is shown below. The p/f bit is the poll/final bit.

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	0	0	0	0	0	0	0	1	N(R)							p/f

Receive not ready. The receive not ready (RNR) response is sent to acknowledge receipt of an I-frame and indicates that the receiving adapter is not ready to receive additional frames. Contained within the RNR frame is the N(R) number, which is the sequence number of the I-frame that the station expects to receive next. That is, an RNR frame with N(R) = 3 acknowledges all I-frames through the I-frame with sequence number 2. The RNR frame can also be used in the checkpointing process. The format of the RNR frame is shown below.

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	0	0	0	0	0	1	0	1	N(R)							p/f

Reject. The reject (REJ) response is sent to request the retransmission of I-frames, beginning with the frame whose sequence number is contained in the N(R) field. The format of the REJ frame is shown as follows.

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	0	0	0	0	1	0	0	1	N(R)							p/f

2.4.5.3 LLC Protocol States

In type 2 communications, link stations can take on any of eight (8) primary and seven (7) secondary states. The descriptions of these states follows. For further information on link station states, refer to the *IBM Token-Ring Architecture Reference*.

2.4.5.3.1 *Primary States*

Closed. This is the default state of all nonexistent link stations. In this state, all received frames except the SABME will be ignored. This state is entered when

1) A CLOSE.STATION command issued by the attached system is completed.
2) The attached system issues an LLC.RESET command.

This state is exited to the disconnect state when the attached system issues an OPEN.STATION command or a SABME has been received and the adapter is waiting for the attached system's response to the connection request.

Disconnected. This state is entered when

1) The attached system issues an OPEN.STATION command.
2) The adapter has received a SABME frame and opened the station.
3) A DISC command has been received from the remote node, and a UA response has been sent.
4) A DM response to a transmitted DISC command is received from the remote node.
5) The T1 timer expires when in the disconnecting state.

This state is exited to the opening state when the attached system issues a CONNECT.STATION command, or when a SABME is received; the state is exited to the closed state when the attached system issues a CLOSE.STATION command.

Disconnecting. This state is entered when

1) The attached system issues a CLOSE.STATION command, and a DISC command has been transmitted.

2) The remote node has responded to transmitted I-frames but has not accepted them, and the retry counts have been exceeded.

This state is exited to the disconnected state if T1 expires or a response has been received to the transmitted DISC command.

Opening. This state is entered when the attached system issues a CONNECT.STATION command, and the adapter has either transmitted a SABME or a UA response to a SABME from the remote node. This state is exited to opened when an I-frame or RR response is received from the remote node. Since I-frames can be transmitted only in the opened state, the two adapters in the opening state will exchange a pair of RR frames before I-frame transmission begins.

Resetting. This state is entered when the adapter receives a SABME frame for an opened link station. This state is exited to the opening state when a response to the SABME is transmitted. Exit is to the disconnected state when a CLOSE.STATION command is issued, or when a DISC command or DM response is received.

Frame reject sent. This state is entered when an illegal frame has been received and a FRMR frame has been sent. This state is exited to the opening state if an SABME is received. It is exited to the disconnecting state if T1 expires, or if the attached system issues a CLOSE.STATION command. Exit is to the disconnected state if a DISC command or DM response is received.

Frame reject received. This state is entered when a FRMR frame has been received in response to a sent I-frame. This state is exited to the opening state if an SABME is received. It is exited to the disconnecting state if T1 expires or if the attached system issues a CLOSE.STATION command. Exit is to the disconnected state if a DISC command or DM response is received.

Opened. This state is entered when the SABME/UA sequence is completed after the attached system issues a CONNECT.STATION command. This is the only state in which data transfer can occur. When in the opened state, a link station may be in any of seven secondary states. If a link station is in either a checkpointing or remote busy state, then I-frame transmission is temporarily suspended.

2.4.5.3.2 Secondary States

Checkpointing. The checkpointing state is entered because of a period of inactivity in the link station. When this state is entered, an RR command (or RNR if in local busy) with the poll bit set to one is transmitted. The adapter waits to receive an RR (or RNR) response with the final bit set to one before resuming I-frame transmission.

Local busy (user). This state indicates that the attached system on the local node is temporarily unable to receive frames. This state is entered when a FLOW.CONTROL command with local_busy_status = 0 option is issued by the attached system.

Local busy (buffer). This state indicates that the attached system on the local node is temporarily unable to receive frames. This state is entered when a frame has been canceled by the attached fsystem, indicating that the SAP is out of buffers.

Remote busy. This state indicates that the remote node of a link station is temporarily unable to receive frames. This condition is indicated by the receipt of an RNR frame from the remote station. This state will be exited when an RR frame is received from the remote node.

Rejection. An REJ frame has been sent to the remote station. This state is cleared when the requested I-frame is received.

Clearing. This state is entered when a local busy condition is cleared, yet the link is in a checkpointing state and so cannot inform the remote station of the cleared local busy condition. As soon as the response is received, removing the station from the checkpointing state, the remote station will be notified of the local busy clear condition.

Dynamic window. The station is processing the dynamic window algorithm. This occurs when the remote station is on another ring and there is congestion in the bridge.

2.4.5.4 LLC Parameters

Familiarity with operational parameters of LLC facilitates understanding the programming interface to LLC. The following section describes those parameters. For further information on the LLC parameters, the user should refer to the *IBM Token-Ring Architecture Reference*, the *IBM Token-Ring PC Adapter Technical Reference*, and the IEEE/ANSI Standard 802.2–1985.

2.4.5.4.1 Timers

The LLC protocols make use of three timers:

❑ The response timer (T1)

❑ The acknowledge timer (T2), and

❑ The inactivity timer (TI)

The T1 timer is started by a link station whenever an I-frame is transmitted. If the timer expires before an acknowledgement to the I-frame is received, then the station enters a checkpointing state and transmits an RR frame (or RNR frame) with the poll bit set to 1. The T1 timer should be set greater than the expected delay of the network in order to avoid unnecessary polls. Normal setting for the T1 timer is 1–2 seconds.

The T2 timer is started by a link station whenever an I-frame is received and the maximum number of I-frames that can be received (N3) has not been reached. If this timer expires before N3 has been reached, then an acknowledgement is sent to the transmitting station. The value of T2 must be less than the value of the remote station's T1. Normal values for T2 are 80–256 milliseconds.

The T1 timer is running whenever T1 is not running. If T1 expires, the checkpointing state is entered, and an RR frame (or RNR frame) is sent to the remote station with the poll bit set to 1. This solicits a response from the remote station and ensures that the link is still active. The T1 timer should be 5–10 times greater than T1. Normal values for the T1 timer are 5–20 seconds.

When timers are set at the software interface, the timer value is selected in increments of a tick count. The tick counts are set by the OPEN.ADAPTER command. The tick counts themselves are numbers of 40-millisecond intervals. See the section on OPEN.ADAPTER command for further details. When selecting a timer value, the attached system selects a number of tick counts. The timer value parameters are a number between 1 and 10. If a number between 1 and 5 is selected, the number is multiplied by the short tick count and by 40 milliseconds to achieve the timer value. If a number between 6 and 10 is selected, the number minus 5 is multiplied by the long tick count and by 40 milliseconds to reach the timer value. For example, if in the OPEN.ADAPTER command, the attached system sets the TIMER_T1_2 parameter to 25, and the TIMER_T1 parameter in the OPEN.SAP command to 10, then the value of the T1 timer is (10–5) x 25 x 40 = 5 seconds.

2.4.5.4.2 Numbers

The LLC protocols use six number parameters, mostly as counters. These parameters are

❑ The maximum I-frame length (N1),

❑ The maximum number of retransmissions (N2),

❑ The maximum number of received I-frames (N3),

❑ The maximum number of outstanding I-frames (Tw),

❑ The working window (Ww), and

❑ The window increment (Nw).

The maximum length of an I-frame (N1) is the maximum frame size of I-frames for a given SAP. This value is set when the SAP is opened. By using this value, upper layer protocols can determine the maximum frame size on a link.

The maximum number of retransmissions (N2) defines the maximum number of times an adapter will try the checkpointing poll when T1 expires. If the N2 count is exceeded without a successful poll, then the link is closed. This parameter is set in the OPEN.SAP and OPEN.STATION commands. Normal values for N2 will be less than 10.

The maximum number of received I-frames (N3) is the number of I-frames that a link station can receive without transmitting a response. This number must always be less than or equal to the Tw value. This number should be set low enough so that the adapter has enough receive buffers to hold all received frames.

The maximum number of outstanding I-frames (Tw) is the number of I-frames that a link station can transmit without receiving an acknowledgement. This number should be set so that the receiving adapter can hold all the transmitted I-frames in its buffers to avoid unnecessary retransmissions. This number should always be greater than or equal to the remote station's (N3) value, or the remote link station will wait until T2 expires before sending an acknowledgement.

The working window (Ww) and window increment (Nw) are two counts associated with the dynamic windowing algorithm. The dynamic windowing algorithm alters the maximum number of outstanding I-frames when congestion on a network is detected. When congestion is detected, the maximum number of outstanding I-frames value, held temporarily in the working window (Ww), is set to one. As conditions improve, the Ww is incremented each time Nw frames are acknowledged. When Ww is equal to Tw, the dynamic windowing state is exited.

Chapter 3

Adapter Hardware Design

This chapter describes the hardware design interfaces for the TMS380C16. The hardware interface includes four sections:

- ❑ The ring interface circuit, which provides the analog signal conditioning for connection to the ring, is described in section 3.1. The TMS38053 data sheet for this analog interface is given in Appendix B, and layout guidelines are given in Appendix E.
- ❑ The local adapter memory interface is discussed in section 3.2. Examples of specific DRAM and EPROM schematics are given in Appendices G and H.
- ❑ Enhanced address copy option, described in section 3.3, is used by the TMS380SRA chip (Appendix C). Source routing bridges that use this interface are described in Appendix F.
- ❑ Detailed system interface bus structures are described in section 3.4.

Section **Page**

3.1 Ring Interface 3-2
3.2 Memory Interface 3-6
3.3 Enhanced Address Copy Option (EACO) Interface 3-35
3.4 Attached System Interface 3-38
3.5 Hardware Notes 3-88

3.1 Ring Interface

The TMS38053 bipolar MSI device and several discrete components are required for 16-Mbps and 4-Mbps token-ring operation with the TMS380C16. For ease of applications, the pulse engineering token-ring optimized line interface (TROLI) can be used. The TROLI module family provides complete signal conditioning between the twisted pair cabling and the TMS380C16. The modules contain the TMS38053, isolation transformers, and the passive-ring interface components.

The adapter connects to the token ring through two twisted pairs of wires: one set to transmit and one set to receive. These twisted pairs connect through a wiring concentrator to form lobes. Each lobe is serially connected to the network to form a ring topology. The wiring concentrator provides the electrical and mechanical function necessary to physically insert and deinsert the ring station onto the ring.

3.1.1 Ring Interface Pins

The following TMS380C16 pins are connected to the ring interface device.

Table 3–1. Ring Interface Pins

Signal	Description

FRAQ	**Frequency acquisition**. This pin is an input to the ring interface, and is interpreted as follows:
	FRAQ = 0 Token-ring input pair (data acquisition)
	FRAQ = 1 Crystal oscillator input (frequency acquisition)
	On every transition of FRAQ, the ring interface will deassert $\overline{\text{REDY}}$ until the ring interface receive circuit detects sufficient signal energy and is locked onto the incoming signal. Refer to the full ring interface specification for energy detect and PLL lock times.

$\overline{\text{REDY}}$	**Ready (PLL GO).** Input from the ring interface $\overline{\text{REDY}}$ is asserted low by the ring interface when minimum signal energy is detected on the input pair. If the energy falls below the minimum value, then $\overline{\text{REDY}}$ is deasserted to one.
RCLK	**Received clock**. This is the output of the ring interface's VCO. RCLK has a period equal to one baud on the ring. Each half-bit of data is provided on the FALLING edge of RCLK, in accordance with EIA synchronous signaling protocols, which specify bit data changes on the falling edge of data clocks (with sampling on the rising edge).

Table 3–1. Ring Interface Pins (Continued)

Signal	Description
RCVR	**Received data in**. Input from ring interface. This is the extracted Manchester data from the ring-receive pair. The RCVR signal changes on the falling edge of RCLK.
DRVR	**Transmitted data out positive**. This signal is the positive-going data output of TMS380C16, which is connected to the ring interface. It is used by the ring interface in conjunction with $\overline{\text{DRVR}}$. The ring interface provides current amplification of the DRVR signals and drives data onto the ring transmit pair.
$\overline{\text{DRVR}}$	**Transmitted data out negative.** This signal is the negative-going ring data output of TMS380C16, which is connected to the ring interface. It is used by the ring interface in conjunction with DRVR. The ring interface provides current amplification of the DRVR signals and drives data onto the ring transmit pair.
$\overline{\text{NSRT}}$	**Insert into ring.** This pin is an active-low output to the ring interface. When asserted, it forces the ring interface to activate the appropriate ring insertion mechanism. For copper signaling, this will cause the ring interface to activate the relays that insert the ring transmit/receive pair into the ring by applying a constant DC bias current on both wires of the transmitting pair. Insertion on fiber media is performed simply by the assertion of signal energy to the external loop lobe.
$\overline{\text{WRAP}}$	**Internal wrap**. This pin is an active low output to the ring interface. When asserted, it causes the ring interface to activate an attenuated internal pin-to-pin feed-back path, and not to assert any signal transitions onto the output pair itself.
$\overline{\text{WFLT}}$	**Wire fault detect.** This pin is an active low input from the ring interface. When set to zero, it indicates the detection of a wire fault in either the receive or transmit pair. When set to one, it indicates valid current drives on the receive and transmit pairs.

Table 3–1. Ring Interface Pins (Concluded)

Signal	Description
PXTALIN	PXTALIN is used by the PH to clock data out of the elastic buffer. In a 16-Mbps adapter, PXTALIN is connected to a 32-MHz external oscillator, which can be provided by the ring interface. PXTALIN will not be connected to the ring interface in a 4-Mbps adapter.When a TMS380C16 reset occurs, the pins going to the ring interface are statically driven as follows: FRAQ: high DRVR: high $\overline{\text{NSRT}}$: high $\overline{\text{WRAP}}$: low

3.1.1.1 Ring Frequency Clock Options

Both the TMS38053 and TMS380C16 (via PXTALIN) require an input clock at the ring baud frequency. This clock may be provided by one of the following:

- ❑ The OSC output from TMS380C16, or
- ❑ A separate ring baud clock generator.

Table 3–2 describes recommended clock connections for various local bus cycle and ring speed combinations:

Table 3–2. Recommended Clock Connections

Local Bus Cycle	Ring Speed	TMS380C16 OSCIN	TMS380C16 CLKDIV	TMS380C16 PXTALIN	TMS38053 FEXTAL
333 ns	4 Mbps	48 MHz	Low	8 MHz	8 MHz
250 ns	4 Mbps	64MHz	High	OSC	OSC
250 ns	16 Mbps	64 MHz	High	32MHz	32MHz

3.1.2 Ring Interface Options

The token-ring standard data rate is at either 4 Mbps or 16 Mbps. When a token-ring product is implemented, the choices for ring interface include

- ❑ 4-Mbps operation only,
- ❑ 16-Mbps operation only, and
- ❑ Switchable 16/4 operation.

The specific circuit implementations for these three options are described in Appendix E. The switchable option requires that the external component values for the TMS38053 be changed by using either a jumper or an FET switch. At any time, the token ring must operate at a single clock frequency. In order to change frequency, the ARESET bit in the SIFACL register (section 4.3.1) must be held high. The state of the TEST0 (pin 79) indicates to the adapter microcode whether 16 Mbps or 4 Mbps is used. The signal may be read by the host via a READ ADAPTER command (address 0A0C).

3.2 Memory Interface

This section describes the interface between the LAN adapter memory bus and the TMS380C16. This interface provides four main functions:

❑ Generates internal clocks required for the TMS380C16 from an external clock input.

❑ Performs an address mapping function for accesses to external LAN adapter bus memory with 2 Mbytes address (20 bits) from the processor address span, which is 64 Kbytes (16 bits). The memory is divided into 32 pages from the processor viewpoint. Segment registers are provided to map the virtual 64 Kbytes to a real 2-Mbyte address.

❑ Acts as an external memory bus controller by providing all necessary signals to easily interface to DRAM. Additional memory types (SRAM, EPROM, or PROM) are supported with interface logic.

❑ Optionally, checks parity on internal data transfers to/from memory to generate parity (if not already generated) for writes to memory, and to add parity to internal data transfers (if not already generated).

3.2.1 Memory Map

The TMS380C16 has a 2-Mbyte address range on the LAN adapter bus. The memory map is divided into 32 chapters of 64K bytes. Adapter ad–dresses are specified in the general form XX.YYYY, where XX is a hexadecimal number in the range of 0 to 1F and YYYY is a hexadecimal number in the range from 0 to FFFF that indicates the byte address.

Figure 3–1 shows the chapter 0 memory map. The first 16 bytes of chapter 0 are reserved for an external burned-in address PROM or PAL. The remainder of chapter 0 memory map is reserved for code and buffers. The code area from 00.8000 to 00. FFFE is reserved for MAC code.

Figure 3–2 shows the chapter 1 memory map. The first 64 bytes are reserved for hardware and software interrupt vectors. The internal TMS380C16 registers are located from 01.0100 to 01.01FE. The locations 01.0200 to 01.03FE are reserved for an external address checker chip, such as the source routing accelerator, TMS380SRA. The remaining portion of chapter 1 is used for data storage.

Chapters 2 through 32 are used for either code or data areas. The amount of memory used is implementation-specific. The adapter bring-up diagnostics will check all memory before use. For a minimum memory adapter with 128 Kbytes, all code and data resides in chapter 0 and chapter 1.

Figure 3–1. Chapter 0 Memory Map (RAM or ROM)

Address	Contents
00.0000	Burned-in Address
00.000E	Code
00.8000 00.FFFE	MAC Code

Figure 3–2. Chapter 1 Memory Map (RAM)

Address	Contents
01.0000	Interrupt vectors
01.007E	Reserved
01.0100 01.01FE	TMS380C16/4 Registers
01.0200	Reserved for External Address Checker
01.02FE 01.FFFE	Data Area

3.2.2 Memory Timing Control

The LAN adapter bus memory access timings are determined from the specified address and whether the implementer is using DRAM or EPROM. The SIFACL register (section 4.3.1) contains a boot bit to control memory access timing. Table 3–3 summarizes this relationship and lists the sections in this manual where detailed timing information is found.

Table 3–3. List of Boot Bits in ACTL Register Controlling Access Timing

Section	Action	Address	Boot Bit (SIFACL)
3.2.2.2	Read from BIA-PROM	00.0000–00.000E	X
3.2.2.3	Read from EPROM	00.0010–00.FFFE 1F.0000–1F.FFFE	0
3.2.2.4 and 3.2.2.5	Read from DRAM or write to DRAM	00.0010–00.FFFE 01.0000–01.00FE 01.0200–01.02FE 01.0300–1E.FFFE 1F.0000–1F.FFFE	1 0 or 1 See Note 0 or 1 1

Note: $\overline{\text{MOE}}$ is not asserted during this address range to allow external address checker to be accessed.

3.2.2.1 LAN Adapter Bus Status and Control

The adapter bus status and control functions are shown in Table 3–4 and are described below.

3.2.2.1.1 Data Multiplexer

Data is transferred on the memory bus as 16-bit words being supplied to or read from memory during the second half of the memory cycle on the MADL0–MADL7 and MADH0–MADH7 buses after the address is supplied during the first half. The MSbyte is on MADH0–MADH7, with bit 0 the MSB; the LSbyte is on MADL0–MADL7, with bit 7 as the LSB.

3.2.2.1.2 Parity Checker

The parity checker verifies every data transfer that has associated parity for correct odd-byte parity by checking the parity and data present on the internal bus at the end of the memory cycle, regardless of whether it is a read or a write cycle, or whether the transfer is internal only or passes through the MIF to or from the memory bus. See section 4.2.1 and PRTYEN in Appendix A.

The data and parity are sampled on the internal bus at the very end of the cycle. At this stage, parity will always accompany the data even if the sourcing device for the data has not provided it, because the MIF's parity generator/passer will have provided it in this case. The analysis is performed in the first half of the following cycle and the parity error signal driven to the appropriate value during the second half of this cycle.

The result of the parity check will, however, be disregarded under one condition; if the PEN bit in the SIF's ACTL register is zero, and/or the APEN bit in the ASF's ASFCTL register is zero. Either bit being zero indicates that adapter parity (but not necessarily host parity) is turned off.

If a parity error is detected and the above case exists, a parity error signal is issued to halt the SIF DMA and deinsert the PH from the ring.

3.2.2.1.3 Parity Generator/Passer

The parity generator provides parity for all data transfers from the internal bus to the memory bus that do not already have parity provided, and for all internal transfers that do not already have parity. It performs this parity generation (if required) very quickly at the start of the data phase of the cycle and drives the calculated parity onto the internal parity bus. This is then passed out to the memory bus with the data and is also available on the internal bus if it is an internal transfer only. The MIF's parity generator knows whether the sourcing device will supply parity because it knows who is the internal bus master and from where the data is being sourced. In addition, if adapter parity is disabled, the parity generator will ignore whatever parity is on the MAXPH and MAXPL lines during a read from memory, and will, instead, add its own parity to the internal data bus so that the SIF and PH parity checkers will still receive good parity.

3.2.2.1.4 Refresh Control

The MIF can refresh DRAMs when a DRAM-refresh cycle is required. Once the refresh logic is granted bus ownership, it will supply the refresh address on the internal bus, which the MIF will output on the memory bus for memories supporting $\overline{\text{RAS}}$-only refresh. The MIF will also generate $\overline{\text{CAS}}$-before-$\overline{\text{RAS}}$ refreshes so that DRAMs of either refresh type can be supported. By taking the MREF pin high, the MIF will indicate that a refresh cycle is occurring. This can be used to disable $\overline{\text{CAS}}$ to devices that don't support $\overline{\text{CAS}}$-before-$\overline{\text{RAS}}$ refresh, or to enable $\overline{\text{CAS}}$ to all devices requiring $\overline{\text{CAS}}$-before-$\overline{\text{RAS}}$ refresh. The refresh cycle is described more fully in the timing diagrams later in this section.

The refresh rate is set to support either a 4- or 3-MHz adapter bus using CLKDIV. The rate is internally set to 62 or 46 adapter bus cycles by CLKDIV. This supports DRAMs with a 4-ms maximum that have 256 refresh addresses.

3.2.2.1.5 Memory Control

The memory control block provides all of the control signals required to interface directly with DRAM, EPROM, PROM, and SRAM, and also provides several internal control signals. Its function is best described by explaining the signals entering and leaving the block.

$\overline{\text{MRAS}}$ This signal, output by the MIF, is the row address strobe for DRAMs. It is taken low while the row address is valid on MADL0–MADL7, MAXPH, and MAXPL. It is also taken low during refresh cycles when the refresh address is valid on MADL0–MADL7.

$\overline{\text{MCAS}}$ This signal, output by the MIF, is the column address strobe for DRAMs. It is taken low during every cycle while the column address is valid on MADL0–MADL7, MAXPH, and MAXPL except when one of the following conditions occurs:

- ☐ When the address is reserved for BIA ROM (00.0000–00.000F).
- ☐ When the address is assigned to on-chip ROM (such as when the ROM bit in the SIF's ACTL register is one and an access to an address in the range 00.C000–00.FFFF is made with $\overline{\text{MACS}}$ asserted low).
- ☐ When the address is assigned to be EPROM (such as when the BOOT bit in the SIF's ACTL register is zero and an access is made to 00.XXXX or 1F.XXXX).
- ☐ When the address is one of the on-chip registers (01.0100–01.01FF in normal mode, or 01.0100–01.07FF, when $\overline{\text{MACS}}$ is asserted. $\overline{\text{MCAS}}$ will be issued if $\overline{\text{MCAS}}$ is not asserted).
- ☐ The cycle is a refresh cycle, in which case $\overline{\text{MCAS}}$ is driven out at the start of the cycle before $\overline{\text{MRAS}}$.

$\overline{\text{MW}}$ This signal, output by the MIF, indicates that the cycle is a read if it remains high or a write if it is driven low. The data on MADH0–MADH7 and MADL0–MADL7 buses and the parity on MAXPH and MAXPL are valid on the falling edge of $\overline{\text{MW}}$ during a write cycle.

$\overline{\text{MAL}}$ This signal, output by the MIF, is a strobe signal for transparent latches used to sample the address output at the start of the cycle in order to create a static address for address decode, SRAM address, or EPROM address. At the falling edge of this signal, the full 20-bit word address is valid on MAX0, MAXPH, MAX2, MAXPL, MADH0–MADH7, and MADL0–MADL7, and the EPROM enable signal is valid on $\overline{\text{MROMEN}}$. Three 8-bit transparent latches can therefore be used to retain a 20-bit static address and EPROM-enable signal throughout the cycle.

MDDIR This signal is an output when an internal device has bus mastership. It serves as a direction control signal for bidirectional buffers used to reduce loading on the MADH and MADL buses.

When it is low, the cycle is a bus master read; when high, the cycle is a bus master write cycle.

$\overline{\text{MBEN}}$ This signal, output by the MIF, is used in conjunction with MDDIR and enables the outputs of bidirectional buffers on the MADH and MADL buses in the direction selected by MDDIR. This signal will not be issued during DRAM refreshes or on-chip address accesses (registers or ROM).

$\overline{\text{MROMEN}}$ This signal, output by the MIF, provides an active low chip-select signal for EPROMs during the first quarter of a memory cycle if the BOOT bit of the SIF's ACTL register is zero and a read from an address in the range 00.0010–00.BFFF, 1F.0000–1F.FFFF, or 00.C000–00.FFFF (when the ROM bit in the SIF's ACTL register is zero) is performed. It will remain inactive high during the first quarter of the cycle for writes to these addresses, accesses of other addresses, or access of any address when BOOT is one. During the final three quarters of the memory cycle, this pin outputs the A13 address signal, which can be used in conjunction with A12 and A14, output on MAX0 and MAX2, respectively, to address a BIA PROM.

$\overline{\text{MBIAEN}}$

This signal, output by the MIF, is used to provide a chip-select signal for a ROM containing the adapter's burned-in address (BIA). The signal will be taken low for any read from addresses in the range >00.0000–00.000F. The signal will remain inactive high for writes to these addresses or accesses to any other address.

$\overline{\text{MOE}}$

This signal, output by the MIF, enables the outputs of DRAMs during a read cycle.

On DRAMs with a 1 orientation (such as TMS4256 256K ×1), $\overline{\text{MOE}}$ drives the output-enable signal on the buffer (a 244-type circuit) on the Q outputs, which isolates the Q outputs from the data bus. This is required as the DRAM's Q outputs are invalid during write cycles. On 4 DRAMs (such as TMS4464s; 64K × 4), $\overline{\text{MOE}}$ connects directly to the DRAM's output-enable pin. $\overline{\text{MOE}}$ is taken low during read cycles under the same conditions the $\overline{\text{MACS}}$ is taken low; though during write cycles and refresh cycles, it will remain inactive high.

MBCLK1
MBCLK2

These two signals are always outputs, even when bus access is granted to an external bus master.

$\overline{\text{MACS}}$

This signal is always an input, and is used as an adapter chip-select signal when an address corresponding to on-chip registers is accessed. This pin should be wired low. CP is the only device allowed to perform on-chip register accesses.

MREF

This signal is an output at all times and is an indication that the cycle occurring on the memory bus is a DRAM refresh cycle. It has two uses:

- ❑ To disable $\overline{\text{MCAS}}$ to DRAMs that don't support $\overline{\text{CAS}}$-before-$\overline{\text{RAS}}$ refresh. This is simply achieved by OR-ing together $\overline{\text{MCAS}}$ and MREF and using the result as the $\overline{\text{CAS}}$ signal for the DRAMs; and
- ❑ To enable $\overline{\text{MCAS}}$ to all DRAMs that use $\overline{\text{CAS}}$-before-$\overline{\text{RAS}}$ refreshing when bank decoding on $\overline{\text{MCAS}}$ is being used.

$\overline{\text{MRESET}}$

This signal, which is an output at all times, is simply the internal reset signal passed through an output buffer.

Table 3–4. Adapter Address Output During Memory Cycle

Pin	First Quarter	Second Quarter	Rest of Cycle
MAX0	AX0	A12	A12
MAXPH	AX1	AX0	Parity
MAX2	AX2	A14	A14
MAXPL	AX3	AX2	Parity
MADH(0)	AX4	Status	Data
MADH(1)	A0	Status	Data
MADH(2)	A1	Status	Data
MADH(3)	A2	Status	Data
MADH(4)	A3	Status	Data
MADH(5)	A4	Status	Data
MADH(6)	A5	Status	Data
MADH(7)	A6	Status	Data
MADL(0)	A7	AX4	Data
MADL(1)	A8	A0	Data
MADL(2)	A9	A1	Data
MADL(3)	A10	A2	Data
MADL(4)	A11	A3	Data
MADL(5)	A12	A4	Data
MADL(6)	A13	A5	Data
MADL(7)	A14	A6	Data
$\overline{\text{MROMEN}}$	ROMEN	A13	A13

AX0–AX4 are the 5 most significant address signals (corresponding to the chapter number). A0–A14 are the 15 least significant address signals (corresponding to the chapter address).

Note that although the row address is said to be valid for the first quarter, and the column address for the second quarter, these are, strictly speaking, the first 5/16ths and following 3/16ths, respectively, as a result of the extension of the row address hold time.

3.2.2.1.6 On-chip Pullups

All of the following signals have on-chip pullups: MAX0, MAX2, MAXPH, MAXPL, MADH0–MADH7, MADL0–MADL7, $\overline{\text{MRAS}}$, $\overline{\text{MCAS}}$, $\overline{\text{MW}}$, MDDIR, $\overline{\text{MAL}}$, $\overline{\text{MOE}}$, $\overline{\text{MBEN}}$, $\overline{\text{MROMEN}}$, $\overline{\text{MBIAEN}}$, and MBRQ.

MIF pins requiring 1-kΩ pull-ups are $\overline{\text{MBRQ}}$ and $\overline{\text{EXTINT}}$ 0–3.

3.2.2.1.7 Effect of Reset on MIF Outputs and Inputs/Outputs

When reset is asserted (either by the $\overline{\text{SRESET}}$ pin or the ARESET bit), the MIF's outputs and input/outputs will behave as follows:

1) MBCLK1, MBCLK2, and FEOSC continue to output clock signals.

2) MAXO, MAX2, MAXPH, MAXPL, MADH0–MADH7, MADL0–MADL7, $\overline{\text{MRAS}}$, $\overline{\text{MCAS}}$, $\overline{\text{MW}}$, MDDIR, $\overline{\text{MAL}}$, $\overline{\text{MOE}}$, $\overline{\text{MBEN}}$, $\overline{\text{MROMEN}}$, and $\overline{\text{MBIAEN}}$ all become high impedance and are pulled up by on-chip pullups.

3) $\overline{\text{MBGR}}$ outputs a driven one except in CP-less mode, when it is high impedance and pulled up by an on-chip pullup.

4) $\overline{\text{MBRQ}}$ outputs a driven one if in CP-less mode; otherwise, it is high impedance and pulled up by an on-chip pull-up.

5) $\overline{\text{MRESET}}$ outputs a driven zero.

6) MREF outputs a driven zero.

When reset is deasserted, the $\overline{\text{MRESET}}$ pin outputs a driven one. The $\overline{\text{MRESET}}$ pin can be used as a reset signal to another TMS380C16 by connecting it to the other TMS380C16's $\overline{\text{SRESET}}$ pin and can also be used to reset any external glue or other peripherals. $\overline{\text{MRESET}}$ is simply the internal reset signal buffered and driven out at M8.

3.2.2.1.8 Status

The status block provides information on the MADH0–MADH7 bus during the second quarter of the memory cycle. This indicates which internal device has ownership of the internal and memory buses, and, if the CP has ownership, whether it is fetching code (instructions or immediate operands) or transferring data. This data is not required for normal adapter operation but is very useful for test and debug purposes, allowing particular device operation tracing. It additionally provides two bits of information required for use by an external adapter node address checker. One bit indicates that the data on the memory bus is the first data of a new received frame; the second bit indicates that the cycle is transferring any data of a received frame to memory.

During the second quarter of the cycle, MADH0–MADH7 are used as listed in Table 3–5.

Table 3–5. Status Information on MADH0–MADH7

MADH Signal	Function
0	Code/$\overline{\text{data}}$
1	Device code 0
2	Device code 1
3	Device code 2
4	Device code 3
5	SIF DMA active
6	PH RX DMA cycle
7	New RX frame

The code/$\overline{\text{data}}$ signal is high only when the CP is fetching code. When it transfers data, or another device has bus ownership, it is low. The SIF DMA active signal is provided by the SIF when its DMA is not complete.

The PH RX DMA cycle signal is high whenever the PH's RX DMA machine is transferring any received frame data to memory, and low at all other times. The new RX frame signal is taken high when the PH RX DMA machine is transferring the first data of a new received frame to memory, and low at all other times. These last two signals can be used by an external adapter node address checker device to examine received frame data for specific information.

The device codes are detailed in Table 3–6.

Table 3–6. Device Code Status on MADH1–MADH4

Device Code				
0	1	2	3	Representation
0	0	0	0	DRAM controller
0	0	0	1	– not assigned –
0	0	1	0	PH TX DMA machine
0	0	1	1	PH RX DMA machine
0	1	0	0	PH TX buffer manager
0	1	0	1	PH RX buffer manager
0	1	1	0	SIF DIO machine
0	1	1	1	SIF DMA machine
1	0	0	0	CP (uses bus)
1	0	0	1	CP (doesn't use bus)
1	0	1	0	– not assigned –
1	0	1	1	– not assigned –

Table 3–6. Device Code Status on MADH1–MADH4 (Continued)

Device Code				Representation
1	1	0	0	– not assigned –
1	1	0	1	– not assigned –
1	1	1	0	– not assigned –
1	1	1	1	– no memory access

The 1111 status code means that no device owns the bus and that a default read from 00.0000 is performed. (This occurs on the first cycle after reset, or whenever no device requests the bus and the CP is halted with the CPHALT bit in the SIF's ACTL register.) Code 1001 means that the CP has been granted bus ownership but doesn't use the bus because it is performing an operation internal to itself. A default read of >00.0000 is performed under this condition also. The arbitration logic provides information on which device has bus ownership.

3.2.2.1.9 Clocks

The clock logic provides all of the internal clocks required by the chip for synchronization to internal adapter events. There is no on-chip oscillator; it is necessary to supply an external oscillator signal to the OSCIN pin, as specified in Appendix A. This external frequency will be 48 MHz for a 4-Mbps token-ring, and 64 MHz for a 4- or 16- Mbps token-ring, as shown in Table 3–2. The externally supplied signal is divided down, as indicated, to supply the internal clocks. All dividers are assumed to synchronize to rising edges, except where stated.

3.2.2.2 Read from BIA ROM

The burned-in address (BIA) PROM is a 8-byte by 8-bit PROM placed on the MADH bus to allow the adapter software to obtain the node address.

The memory bus timings for an BIA ROM read are given in Figure 3–3. This type of access occurs when an address in the range >00.0000–00.000F is read.

MREF is taken low at the start of M8 of the previous cycle to indicate to any external logic attached to DRAM $\overline{\text{CAS}}$ inputs that the cycle is not a refresh cycle.

The $\overline{\text{MAL}}$ signal is typically supplied to a transparent latch. This samples the address on the falling edge of $\overline{\text{MAL}}$, retaining it as a static address throughout the rest of the cycle. $\overline{\text{MROMEN}}$ is also sampled on the falling edge of $\overline{\text{MAL}}$ to latch the inactive EPROM enable signal and retain it throughout the cycle to prevent EPROM access.

At the center of M3, MAXO, $\overline{\text{MROMEN}}$, and MAX2 output A12–A14 of the address and maintain these throughout the rest of the cycle. Since the BIA is 8 bytes long, only three address lines are required to address the PROM. These three lines can therefore connect directly to the address lines of the PROM, without the need to latch the address.

MDDIR indicates to any bidirectional buffers used to buffer BIA data onto the data bus and parity lines that the MIF is performing a read (the $\overline{\text{MW}}$signal remaining high also does this). $\overline{\text{MBIAEN}}$ is taken low after the MIF has three-stated the MADH, MADL, MAXPH, and MAXPL signals to enable the outputs of the PROM containing the BIA. $\overline{\text{MBEN}}$ is also taken low at the same time to enable any bidirectional buffers to drive the data and parity onto the memory bus.

Data and parity read from the BIA are sampled at the end of M6 and driven onto the internal data bus during M7. If adapter parity is disabled (PEN bit in the SIF's ACTL register is zero, or the APEN bit in ASFCTL is zero), then the values applied to the parity lines are don't care. $\overline{\text{MBIAEN}}$ and $\overline{\text{MBEN}}$ are taken high at the start of M7 to three-state the PROM and buffer outputs before the next address is output in the following M1.

$\overline{\text{MCAS}}$ is retained at the inactive high level throughout the cycle to prevent accesses to DRAM. $\overline{\text{MOE}}$ is retained inactive high to keep DRAM outputs three-stated. $\overline{\text{MRAS}}$ is issued, however, because this does no harm and actually performs an additional DRAM refresh of the row address presented on A7–A14.

This cycle is also performed when no internal or external device uses the cycle for a data transfer. The address output will be 00.0000. In this case, the MIF parity checker does not check for correct parity, since no BIA ROM need be present and the data and address lines are floating.

Figure 3–3. Read from Burned-In Address PROM (BIA) ROM

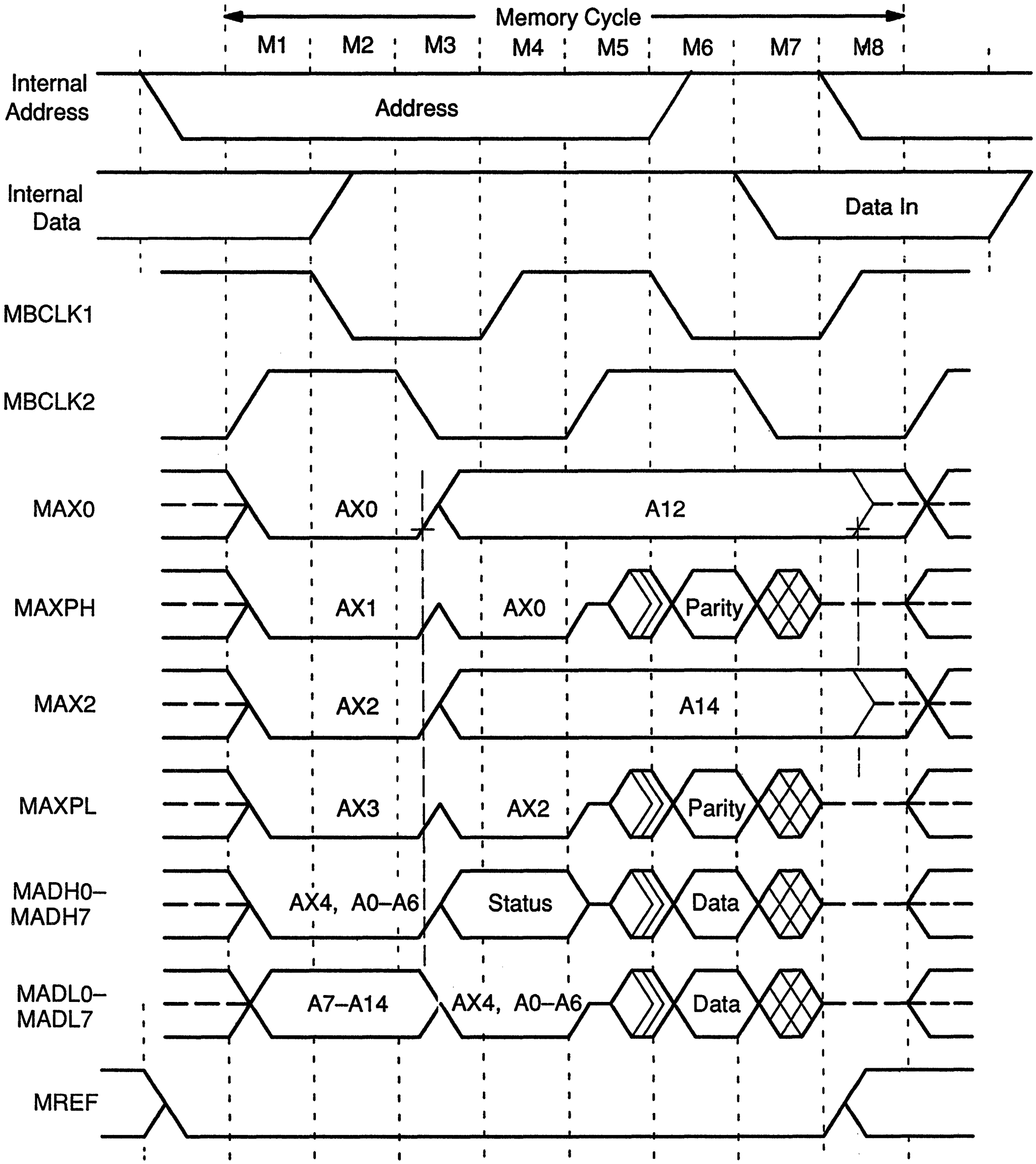
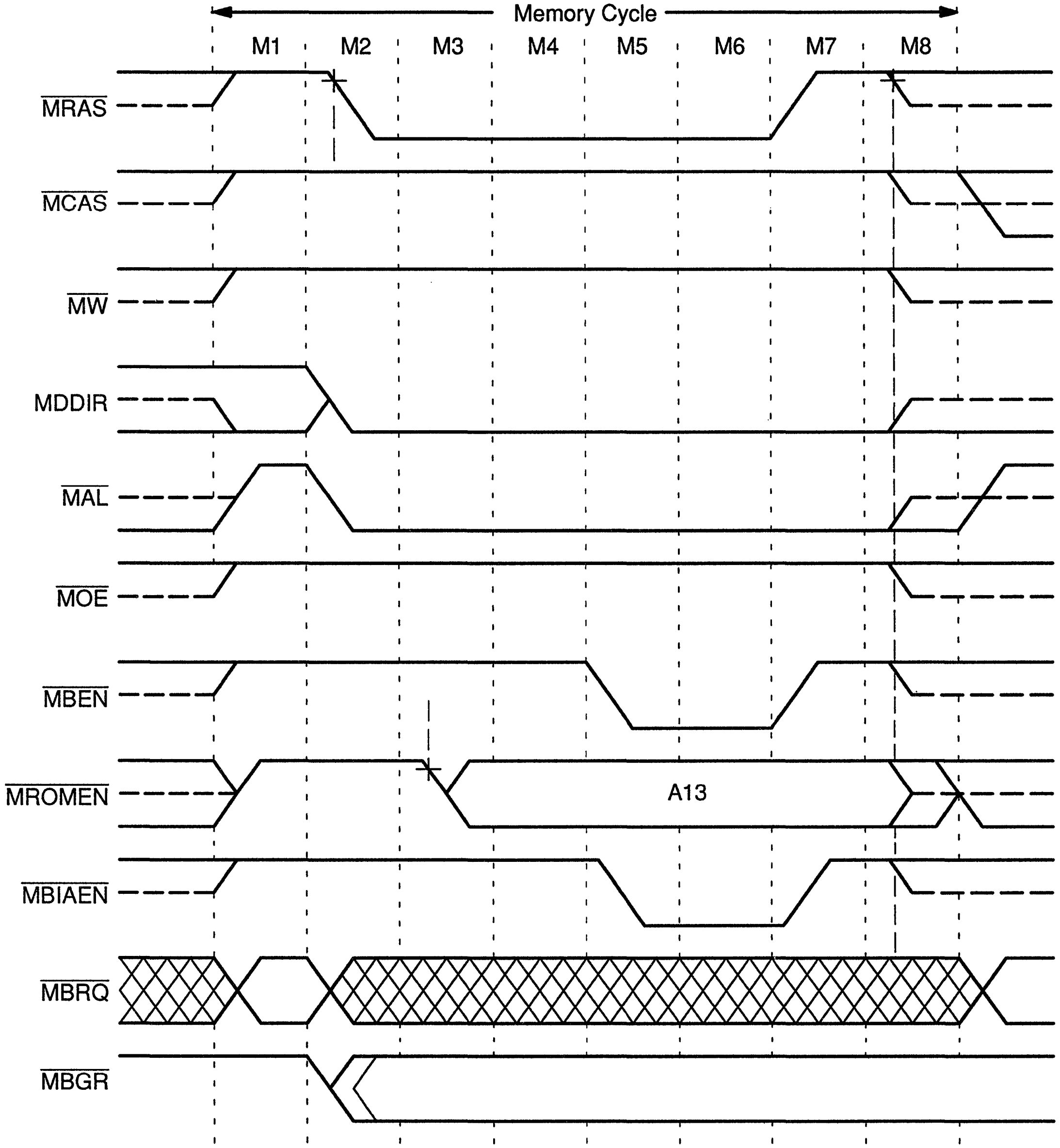


Figure 3–3. Read from Burned-In Address PROM (BIA) ROM (Concluded)



A read from BIA address 00.0000 to 00.000F does not require an external parity bit. The BIA can be read from MADH0–MADH7, the most significant byte of the bus, by the adapter software or the host system at any time.

The BIA is read by adapter software. If it is all zeros or the word checksum is incorrect, the address will not be used, and the host system must supply the node address at open.

The BIA is electrically connected to the most significant byte of the TMS380C16 memory interface bus (such as MADH0–MADH7) and is read by the adapter software in bytes. It is then placed in word format, and a checksum is generated and checked. Figure 3–4 shows a BIA PROM representation and checksum rules.

Figure 3–4. Logical Representation of BIA

MSB	LSB	
0	1	
2	3	Node address
4	5	
6	7	Checksum

Byte 0 = MS byte of adapter address
1
2
3
4
Byte 5 = LS byte of adapter address
Byte 6, 7 = Word checksum of 6, 7 = 0, 1 + 2, 3 + 4, 5 (in words)

BIA Programming Procedure

The following is a description of the BIA PROM programming procedure. The BIA may be contained in a 32 x 8 PROM (TBP18S30 or equivalent).

BIA PROM Memory Map

The BIA PROM memory map is the following:

Byte 0	Node address byte 0
Byte 1	Node address byte 1
Byte 2	Node address byte 2
Byte 3	Node address byte 3
Byte 4	Node address byte 4
Byte 5	Node address byte 5
Byte 6	Checksum byte 0
Byte 7	Checksum byte 1
x	Unused bytes

Calculating BIA Checksum

The BIA checksum stored in bytes 6 and 7 is calculated as a 16-bit checksum. Bytes 0 and 1, bytes 2 and 3, and bytes 4 and 5 compose the three 16-bit quantities used in the checksum calculation. All remaining PROM bytes beyond byte 7 are unused.

3.2.2.3 Chapter 0 or 31 Read from EPROM

The memory bus timings for an EPROM read are given in Figure 3–5. This is the type of access that occurs when the following two conditions are true:

❑ The BOOT in the SIF's ACTL register is zero, and

❑ A read from an address in the range >00.0010–>00.FFFF or >1F.0000–>F.FFFF is performed.

MREF is taken low at the start of M8 of the previous cycle. This indicates to any external logic attached to DRAM CAS inputs that the cycle is not a refresh cycle.

The MAL signal is typically supplied to a transparent latch. This samples the address on the falling edge of MAL, retaining it as a static address throughout the rest of the cycle. The MROMEN signal is also taken low on M1 and latched externally by the falling edge of MAL to maintain an EPROM enable signal throughout the cycle.

MDDIR indicates to any bidirectional buffers used to buffer EPROM data onto the data bus and parity lines that the MIF is performing a read (MW signal remaining high also does this). MBEN is taken low after the MIF has three-stated the MADH, MADL, MAXPH, and MAXPL signals to drive the EPROM data and parity onto these signals.

Data and parity read from the EPROM are sampled at the end of M6 and driven onto the internal data bus during M7. If adapter parity is disabled (PEN bit in the SIF's ACTL register is zero, or APEN bit in ASFCTL is zero), then the values applied to the parity lines are don't care. MBEN is taken high at the start of M7 to three-state the buffer outputs before the next address is output in the following M1.

MCAS and MBIAEN are retained at the inactive high level throughout the cycle to prevent accesses to DRAM and BIA PROM, respectively. MOE is also retained inactive high to keep DRAM outputs three-stated. MRAS is issued, however, because this does no harm and actually performs an additional DRAM refresh of the row address presented on A7–A14. MRAS is normally connected to the OE pin of the EPROMs.

Figure 3–5. Chapter 0 or 31 Read from EPROM

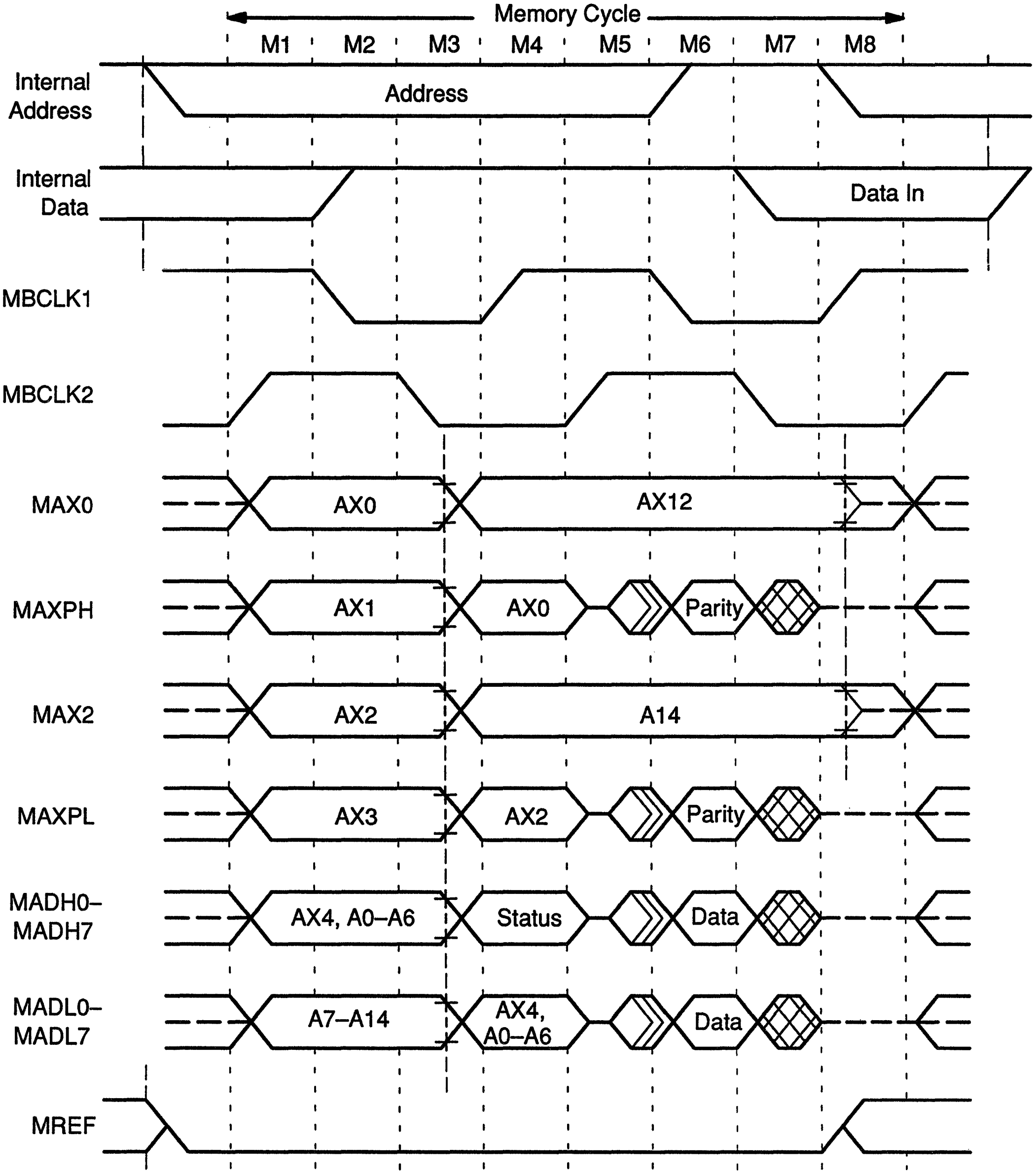
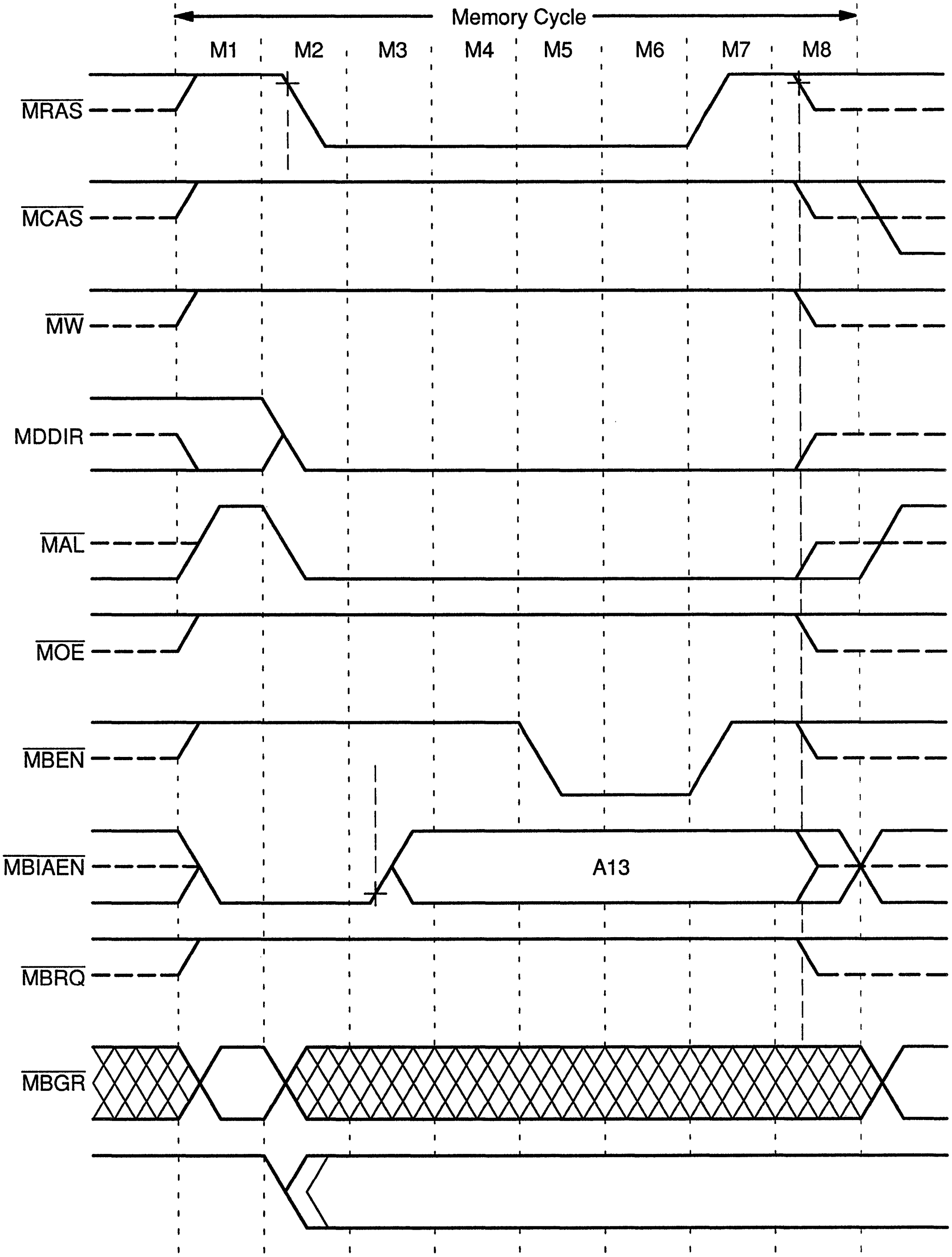


Figure 3–5. Chapter 0 or 31 Read from EPROM (Concluded)



3.2.2.4 Read from DRAM Address

The memory bus timings for a read from a non-BIA ROM, a non-EPROM, and a non-on-chip address are given in Figure 3–6. This is the type of access that occurs when one of the following conditions is true:

- □ A read from an address in the range >00.0010–>00.FFFE or >1F.0000–>F.FFFE is performed when the BOOT bit in the ACTL register is a one.
- □ A read from an address in the range >01.0000–>01.00FF.
- □ A read from an address in the range >01.0200–>01.02FE. However, $\overline{\mathrm{MOE}}$ is not asserted; this allows for external address checker at this location.
- □ A read from an address in the range >01.0300–>E.FFFE.
- □ A read from an address in the range >1F.0000–>F.FFFE when BOOT bit in the ACTL register is one.

MREF is taken low at the start of M8 of the previous cycle to indicate to any external logic attached to DRAM $\overline{\mathrm{CAS}}$ inputs that the cycle is not a refresh cycle.

The $\overline{\mathrm{MAL}}$ signal is typically supplied to a transparent latch. This samples the address on the falling edge of $\overline{\mathrm{MAL}}$, retaining it as a static address throughout the rest of the cycle. $\overline{\mathrm{MROMEN}}$ is also sampled on the falling edge of $\overline{\mathrm{MAL}}$ to sample the inactive EPROM enable signal and retain it throughout the cycle.

The $\overline{\mathrm{MRAS}}$ signal is taken low in the center of M2 to latch the row address into the DRAMs. Depending upon which size of DRAM is used, the row address is taken from MADL0–MADL7 only; MADL0–MADL7 and MAXPL; or MADL0–MADL7, MAXPL, and MAXPH.

$\overline{\mathrm{MCAS}}$ is taken low at the start of M4 to latch the column address into the DRAMs. The signal lines used for the column address are the same as for the row address.

MDDIR indicates to any bidirectional buffers used to buffer memory data onto the data bus and parity lines that the MIF is performing a read; $\overline{\mathrm{MW}}$ signal remaining high also does this. $\overline{\mathrm{MBEN}}$ is taken low after the MIF has three-stated the MADH, MADL, MAXPH, and MAXPL signals at the start of M5 to drive the memory data and parity onto these signals. $\overline{\mathrm{MOE}}$ also taken low at the same time to enable the output of the DRAMs onto the data bus.

Data and parity read from memory are sampled at the end of M6 and driven onto the internal data bus during M7. If adapter parity is disabled (PEN bit in the SIF's ACTL register is zero, or the APEN bit in ASFCTL is zero), then the values applied to the parity lines are don't care. MRAS, MCAS, and MOE are all taken high at the start of M7 to terminate the DRAM read, and MBEN is taken high to three-state the buffer outputs.

MBIAEN is retained at the inactive high level throughout the cycle to prevent accesses to BIA ROM.

Note that although this cycle is described as non-EPROM (because the EPROM enables signal on MROMEN remains inactive high), there is no restriction to prevent EPROMs from being used at addresses causing this type of memory cycle. However, if EPROMs are used, extra chip-select logic must be used to prevent MCAS from being issued to DRAMs when addresses used by EPROM are accessed.

Figure 3–6. Read from DRAM

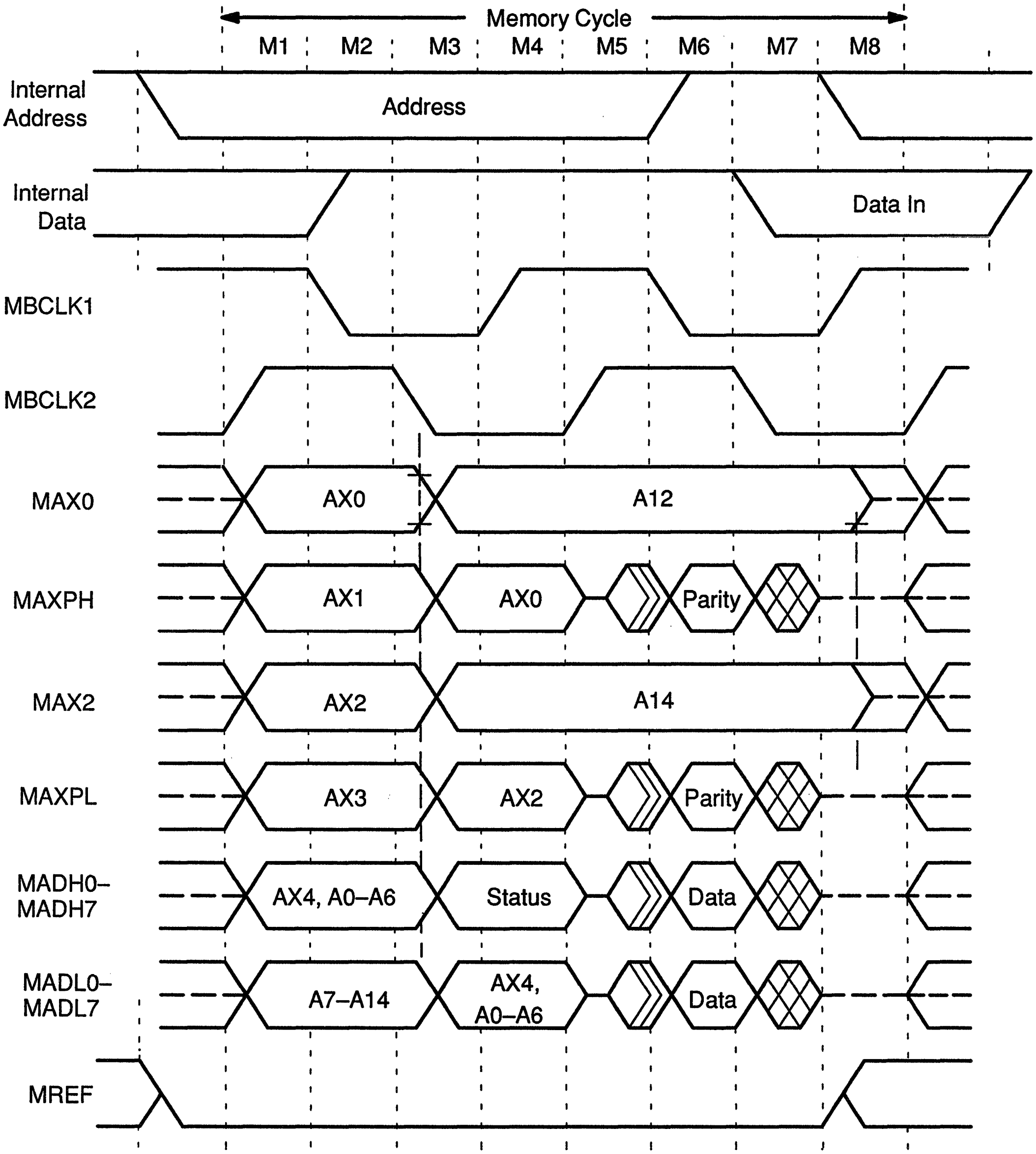
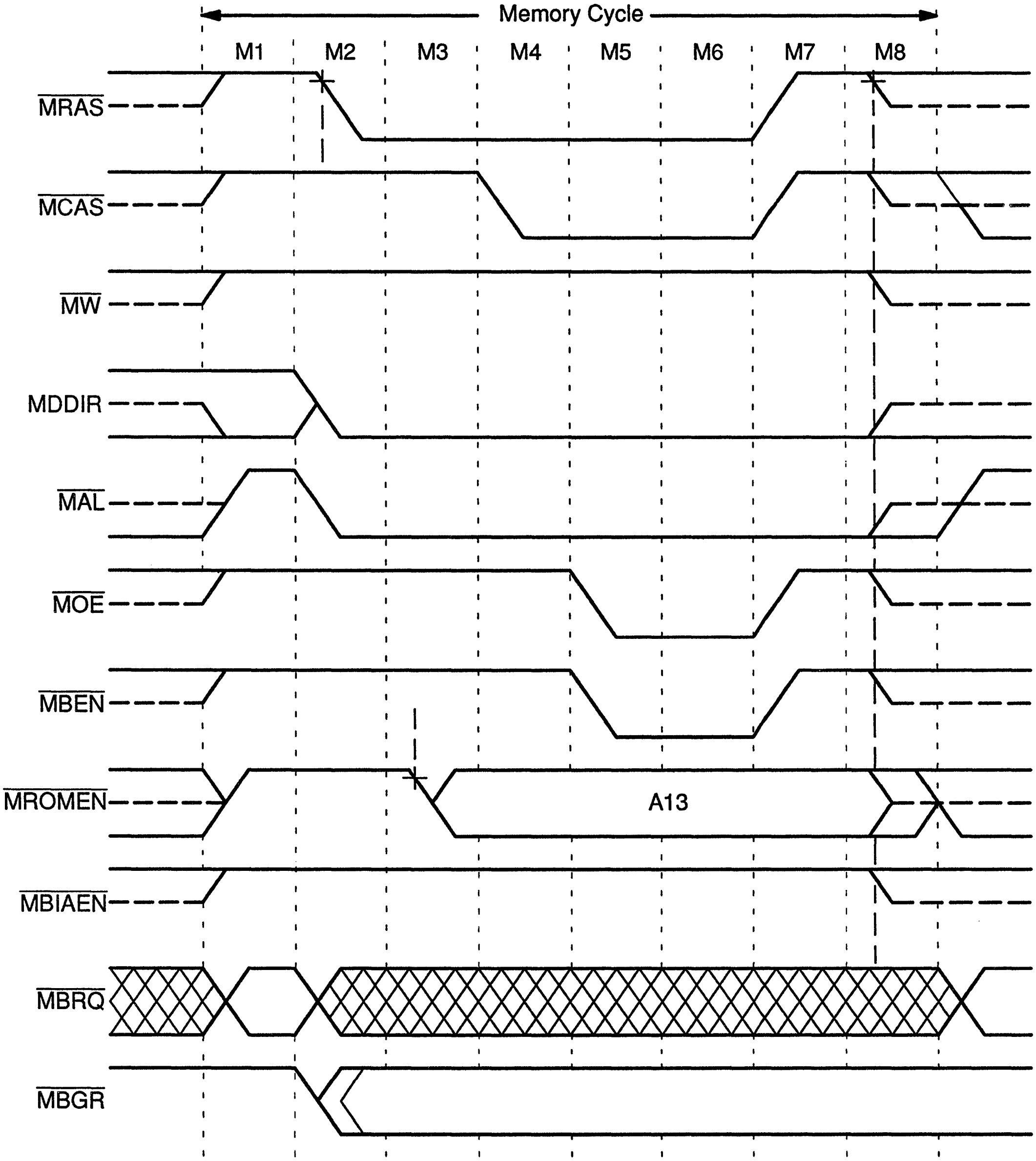


Figure 3–6. Read from DRAM (Concluded)



3.2.2.5 Write to DRAM

The memory bus timings for a write to a non-BIA ROM, non-EPROM, and non-on-chip address are given in Figure 3–7.

Figure 3–7. Write to DRAM

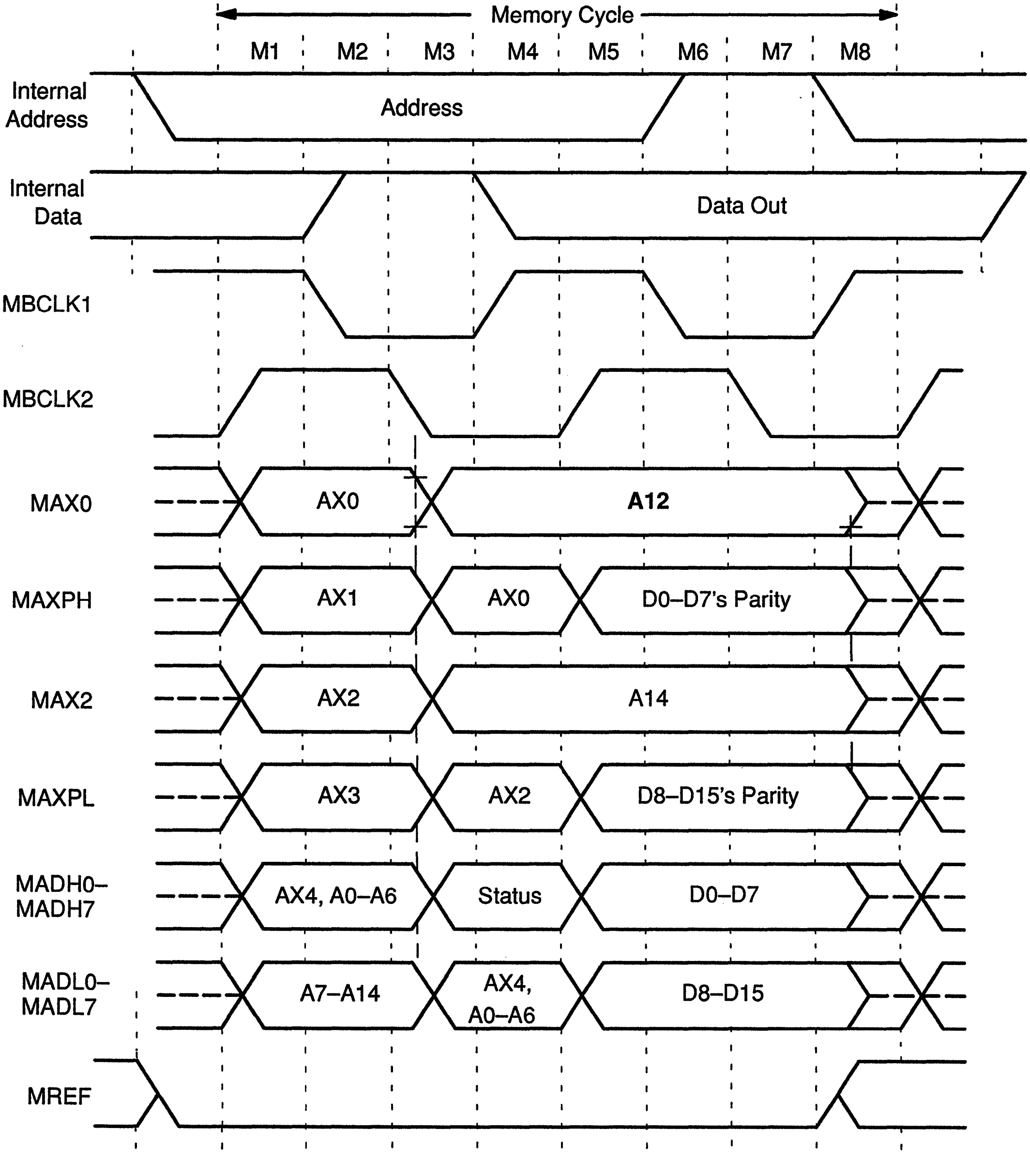
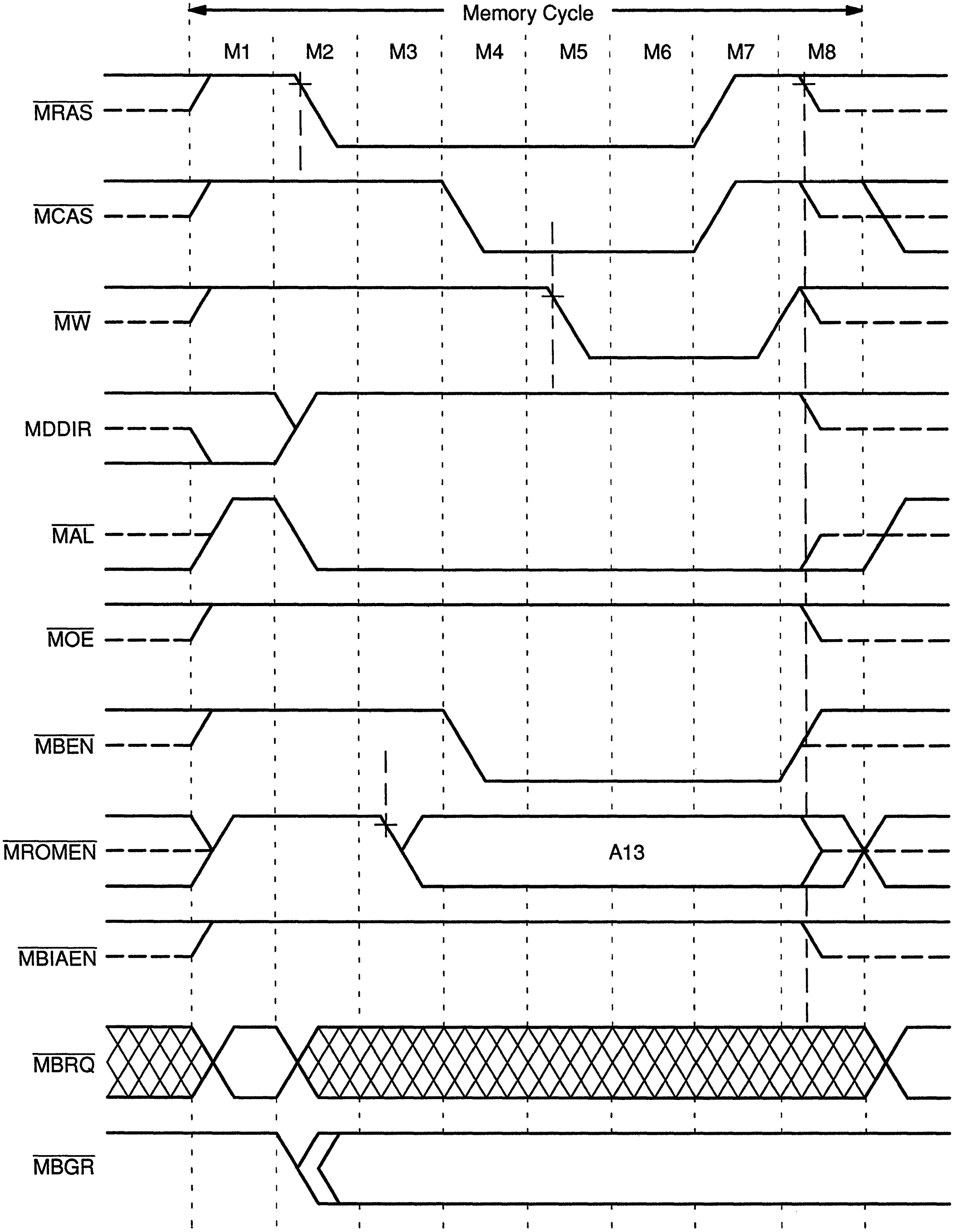


Figure 3–7. Write to DRAM (Concluded)



The type of access shown in Figure 3–7 occurs when one of the following conditions is true:

❑ A write to an address is the range >00.0010–>00.FFFE or >1F.0000–>1F.FFFF when the BOOT bit in the SIF's ACTL register is a one.

❑ A write to an address in the range >01.0000–>01.00FF.

❑ A write to an address in the range >01.0200–>01.02FF. However, MOE is not asserted for external address checker at this location.

❑ A write to an address in the range >01.0300–>1E.FFFE.

MREF is taken low at the start of M8 of the previous cycle to indicate to any external logic attached to DRAM CAS inputs that the cycle is not a refresh cycle.

The MAL signal is typically supplied to a transparent latch. This samples the address on the falling edge of MAL, retaining it as a static address throughout the rest of the cycle. MROMEN is also sampled on the falling edge of MAL to sample the inactive EPROM enable signal and maintain it throughout the cycle.

The MRAS signal is taken low in the center of M2 to latch the row address into the DRAMs. Depending upon which size of DRAM is used, the row address is taken from MADL0–MADL7 only; MADL0–MADL7 and MAXPL; or MADL0–MADL7, MAXPL, and MAXPH.

MCAS is taken low at the start of M4 to latch the column address into the DRAMs. The selection of signals used for the column address is the same as for the row address.

MDDIR indicates to any bidirectional buffers used to buffer memory data onto the data bus and parity lines that the MIF is performing a write. MBEN is taken low at the start of M4 to allow the maximum setup time on the data presented to memory.

Data and parity are written to DRAMs on the falling edge of MW which occurs in the center of M5, and are written to SRAMs on the rising edge of MW which occurs at the start of M8. MRAS and MCAS are taken high at the start of M7to terminate the DRAM write. MBEN is taken high to three-state the buffer outputs in the center of M8.

MBIAEN is retained at the inactive high level throughout the cycle to prevent accesses to BIA ROM. MOE is also retained inactive high to keep the DRAM outputs three-stated.

3.2.2.6 DRAM Refresh Cycle

The memory bus timings for a DRAM refresh cycle are given in Figure 3–8. This is the type of access that occurs when the DRAM refresh controller has been granted bus ownership.

The MREF output is taken high at the start of M8 of the previous cycle. This is an indication to any bank decode logic on $\overline{\text{MCAS}}$ that a refresh cycle is about to occur, and to allow $\overline{\text{MCAS}}$ to pass through to all devices requiring $\overline{\text{CAS}}$-before-$\overline{\text{RAS}}$ refresh signals. It also allows $\overline{\text{MCAS}}$ to be inhibited to any memory that does not support $\overline{\text{CAS}}$-before-$\overline{\text{RAS}}$ refresh.

The $\overline{\text{MAL}}$ signal is issued for the duration of M1 to allow the address to be sampled by transparent latches. The WORD address output is 088XX, in which XX is the refresh address. It doesn't matter what memory is allocated to this address, because any data read is ignored by the MIF, and a write to the address does not occur. The inactive EPROM enable signal on $\overline{\text{MROMEN}}$ is also sampled on the falling edge of $\overline{\text{MAL}}$ to retain it throughout the cycle.

The $\overline{\text{MCAS}}$ signal is taken low at the start of M1 to allow $\overline{\text{CAS}}$-before-$\overline{\text{RAS}}$ refreshing to those DRAMs that support this refresh method.

The $\overline{\text{MRAS}}$ signal is taken low in the center of M2 to latch the refresh address into the DRAMs that use $\overline{\text{RAS}}$-only refreshing, and to perform the refresh for DRAMs supporting $\overline{\text{CAS}}$-before-$\overline{\text{RAS}}$ refreshing. The 8-bit refresh address is placed on MADL0–MADL7, and MAXPL and MAXPH are both zero. DRAMs with more than 8 address lines should be wired so that the 8 address inputs expecting the refresh addresses are connected to MADL0–MADL7. Devices that require more than 8 bits of refresh address should instead use $\overline{\text{CAS}}$-before-$\overline{\text{RAS}}$ refreshing.

MDDIR is taken low, indicating a MIF read, but $\overline{\text{MBEN}}$ remains high throughout the cycle so that any bidirectional buffers remain three-stated in both directions throughout the cycle.

$\overline{\text{MRAS}}$ and $\overline{\text{MCAS}}$ are taken high again at the start of M7 to complete the refresh cycle. MREF is taken low at the start of M8.

$\overline{\text{MBIAEN}}$, $\overline{\text{MOE}}$, $\overline{\text{MW}}$, and $\overline{\text{MBEN}}$ are retained at the inactive high level throughout the cycle to prevent any accidental memory accesses. Note that the first cycle after reset is not a DRAM refresh so that there is no discontinuity on MREF, which is driven low during reset.

Figure 3–8. DRAM Refresh Cycle

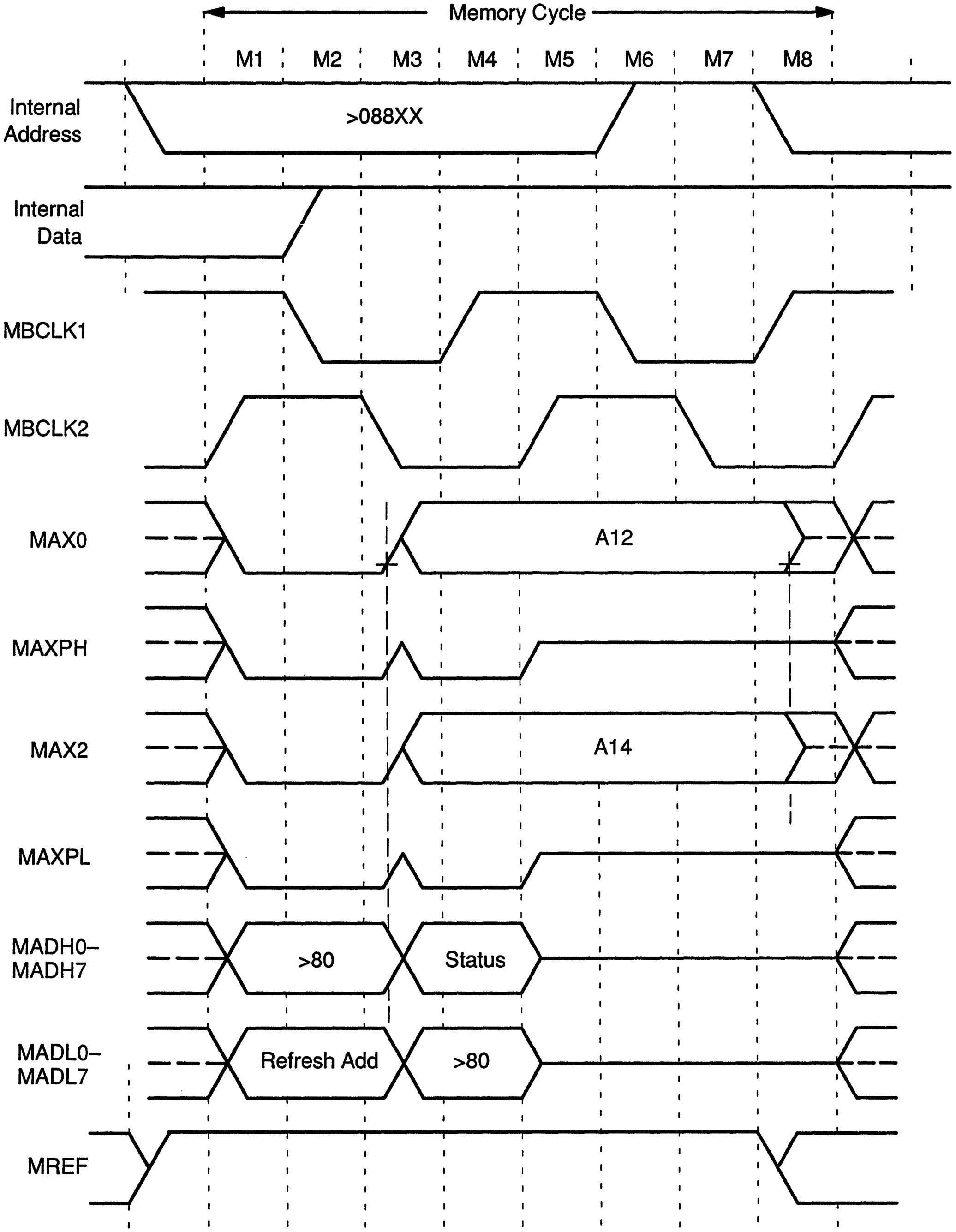


Figure 3–8. DRAM Refresh Cycle (Concluded)

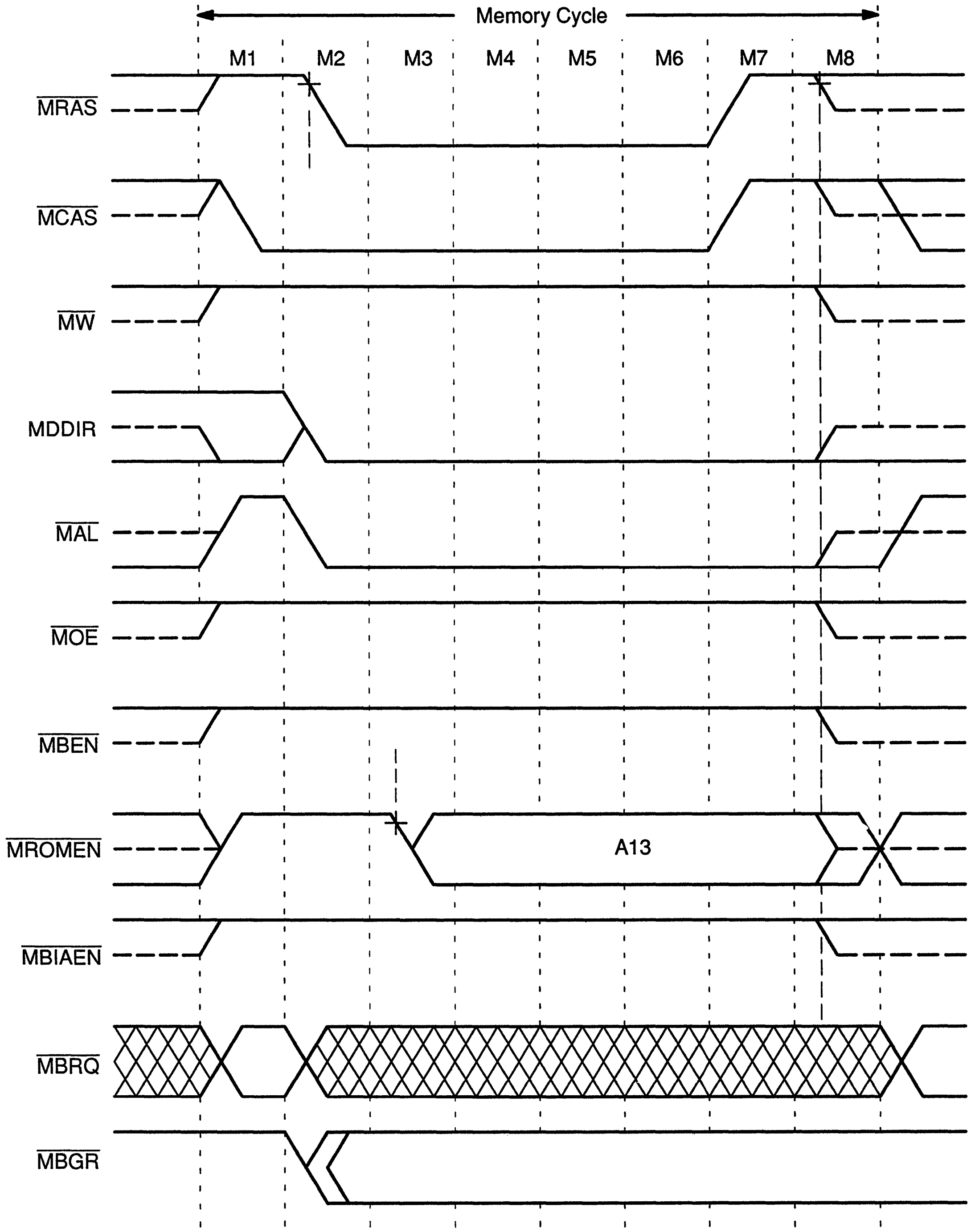
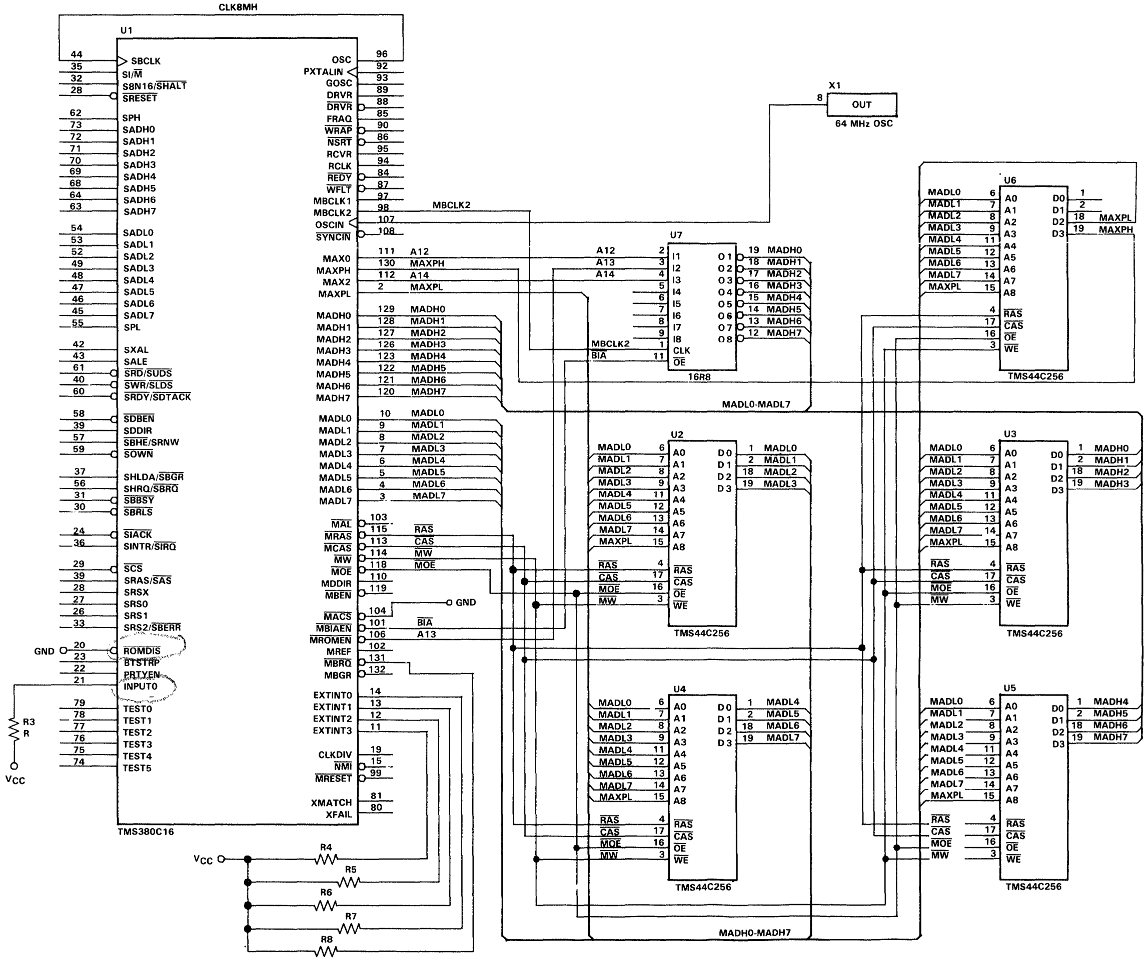


Figure 3–9. TMS380C16 DRAM Memory Interconnection with BIA PAL



3.3 Enhanced Address Copy Option (EACO) Interface

The TMS380C16 supports an Enhanced Address Copy Option (EACO) which enhances the address recognition function when frames are copied from the ring. The EACO interface **allows an adapter to copy frames that otherwise would not be copied** by the TMS380C16's internal address compare logic. The EACO hardware interface consists of the adapter bus signals and the following TMS380C16 pins:

- ❑ XMATCH
- ❑ XFAIL

Table 3–7. EACO Interface Pins

Signal	Description
XMATCH	If this signal is asserted by external device, TMS380C16 copies the current frame to adapter memory, provided XFAIL is low. This active high signal has an internal pull-up.
XFAIL	This signal is used in conjunction with XMATCH. An external address checker uses XFAIL to signal that it does not wish to copy the current frame. This active high signal has an internal pull-up. If both XMATCH and XFAIL are asserted, XFAIL takes precedence. If the pins are left unconnected, XFAIL is always asserted.

The TMS380C16 uses a recoverable buffering technique to receive frames from the ring. A received frame is entered into buffer memory by DMA until the following decision is completed:

- ❑ Do not copy the frame; the buffer will be recovered, or
- ❑ Copy the frame; the remainder of the frame will be copied to the buffer(s) and accepted.

During reception of a frame, the frame is available on the adapter bus for examination by external hardware. The source, destination address, and routing information fields determine if the frame is to be copied.

The TMS380C16 places status information on the adapter bus during the second quarter of a memory cycle (see section 3.2.2.1.8). This status indicates to external hardware what information is on the adapter bus. The status information identifies either **New RX Frame** or **PH RX DMA Cycle.**

New RX Frame — This indicates that the PCF word (see section 2.4.2.2) of a new frame is now being entered into the present buffer by DMA.

PH RX DMA Cycle — This indicates that the next received data word of the frame is now being entered into the present buffer by DMA.

The TMS380C16 is not able to accept or reject the frame until both the internal address compare is completed and an external signal is received. The timing of the external hardware cannot be assumed to be the same as the internal address compare logic. Therefore, the TMS380C16 will continue transferring the data into the buffer until the external hardware indicates completion by setting either XMATCH or XFAIL active to indicate that the frame should be copied or not be copied, respectively.

If a frame is externally matched (indicated by XMATCH and not XFAIL), then both ARI and FCI bits will be set in the passing frame's FS (see Section 2.4.2.10). If a frame is externally matched, but there is no buffer available for the frame, then only the ARI bits of the frame's FS will be set. This is considered a receive congestion error. When there are no buffers available to copy a frame, the frame is loaded to address 01.0200 on the adapter bus.

These are additional notes about the EACO interface:

- ❑ If the XMATCH and XFAIL are left unconnected (internally pulled high), the adapter assumes XFAIL.
- ❑ The TMS380C16 will not chain past the first buffer given. Therefore, the external hardware must indicate whether to copy or not to copy at least four bytes before the end of the first buffer.
- ❑ The buffer recovery logic is delayed until both the internal address checker and external hardware check are complete. The external hardware must finish its check in time to allow the buffer to be recovered. Therefore, the result of an external check should be posted prior to transfer of the next to last word.
- ❑ When a receive frame error is detected and the frame is no longer loaded to the adapter memory, XFAIL is internally set (in other words, when a new RX frame status indicator is set, the external EACO chip should start the recognition process again, aborting any previous frame recognition in progress).
- ❑ When a frame is externally matched, the external address match bit of the received CSTAT field is set to one.
- ❑ If a MAC frame is both externally and internally matched, the adapter processes the frame as if it were an internal match only.

- ❑ If a non-MAC frame is received and LLC is enabled, the following occurs:
  - ■ Externally matched frames are sent to the attached system as bridge data.
  - ■ Externally and internally matched frames are copied and the external copy is sent to the attached system as bridge data, with only the external address match bit set (see section 4.10). The internal copy is processed as an internally matched frame only. Therefore, it is possible that two frames may be sent to the attached system.
- ❑ Internally matched frames are always copied, regardless of the external hardware indication to copy or not to copy.

3.4 Attached System Interface

The TMS380C16 interfaces to the attached system by using two selectable modes: 808x mode and the 68xxx mode. The 808x mode gives a compatible interface to the 808x series of CPUs and buses, using either 8-bit or 16-bit data bus widths. The 68xxx mode gives a compatible interface to the 68xxx series of CPUs or buses, using a 16-bit data bus width. Both modes support 32-bit attached system address and optional parity on address or data paths. Of course, the designer is not constrained to use these processor types in the attached system but should select the interface mode which best fits his system hardware.

The data transfer between the TMS380C16 and the attached system is supplied by direct I/O and bus master DMA. Direct I/O is used to initialize the TMS380C16 and provide handshaking between attached system and adapter. Sixteen registers are accessible by direct I/O. All message transfers (both commands and frame data) are accomplished using adapter DMA or adapter pseudo-DMA. The DMA interface allows the adapter to gain control of the attached system bus and furnish address and data for direct transfer to/from attached system memory. In most applications, this DMA interface gives the highest speed message transfer rate.

In attached systems where direct bus control is awkward or not permitted, the TMS380C16 offers an alternate means of message transfer called pseudo-DMA. In pseudo-DMA, the adapter supplies to the attached system the specific parameters to allow the attached system to control message transfer. Direct I/O register settings establish the length of transfer and the attached system starting address to the attached system. The TMS380C16 provides the actual message information to a single direct I/O location, which can also be mapped into the attached system memory space. The attached system, using the above information furnished by the TMS380C16, then transfers the data from this I/O location to the appropriate system memory.

In the following sections, timing information is given for each mode of operation. Direct I/O hardware interface is described in section 3.4.1 for the 808x mode and section 3.4.2 for the 680xx mode. Interrupt handshaking timing is also explained in these sections. Chapter 4 has a description of the direct I/O registers and their control of adapter operations. DMA operations are discussed in section 3.4.3, and the DMA timing is presented in section 3.4.4 for the 808x mode and section 3.4.5 for the 680xx mode. Burst DMA or cycle-steal DMA modes are summarized in section 3.4.6 and bit and byte numbering organization for DMA transfers is found in section 3.4.7. Finally, section 3.4.8 describes how the address control signals are organized to provide full 32-bit DMA.

3.4.1 Direct I/O for 808x Mode

3.4.1.1 Multiplexed and Nonmultiplexed Address

Two DIO interfacing options are provided in 808x mode. The nonmultiplexed option is suitable for expanded configurations in which the address/ data bus from the user processor has been demultiplexed onto physically separate address and data lines. In this case, the chip-select and signals presented to the SIF are required to remain stable and valid for the duration of the DIO access cycle. The multiplexed option is provided to reduce the chip count in minimum-chip configurations in which the SIF is interfaced directly to the multiplexed address/data lines of the user processor. In this case, the chip-select and register-select inputs are latched on chip at the beginning of the cycle; therefore, they need remain valid only while they are being latched.

The mechanism for implementing the two interfacing options discussed above is shown in Figure 3–10. The SRAS/$\overline{\text{SAS}}$ signal is called SRAS because of its interpretation as a register address strobe in 808x mode. As the chip-select and register-select signals come on chip, they are input to a transparent latch. The SRAS input serves as the active-high register address strobe that enables the latch. As long as SRAS remains high, the outputs of the latch follow the chip-select and register-select inputs. When SRAS goes low, the outputs of the latch are frozen at the levels present at the time of the falling edge of SRAS. In a minimum-chip system, the multiplexed option is implemented as follows.

The chip-select and register-select inputs of the CP chip are connected to the selected address/data pins of the processor. The falling edge of ALE causes the selected address bits to be latched inside the SIF. This means that external latches are not required. In an expanded system, the address/ data bus from the user processor is demultiplexed, and the nonmultiplexed option is used. To implement this option, the SRAS input is strapped high.

Note:

In reality, the register-select bits are latched internally, even if SRAS/$\overline{\text{SAS}}$ is tied high. The latch enable is turned off as soon as both $\overline{\text{SCS}}$ and $\overline{\text{SWR/SLDS}}$ or $\overline{\text{SRD/SUDS}}$ are asserted and certain internal busy signals are de-asserted. SRAS/$\overline{\text{SAS}}$ would normally occur first, if used, and would provide the more accurate timing for latching an address from a multiplexed bus. In the 68xxx mode, the SRAS signal is ignored, and only the internal latching process is used.

- Internal latching (by use of the SRAS/$\overline{\text{SAS}}$ pin) of chip-select and register-select inputs is an option provided in 808x mode only.
- $\overline{\text{SCS}}$ is disabled when the SIF controls the system bus ($\overline{\text{SOWN}}$ active).

3.4.1.1.1 DIO Ready Control

The delay of $\overline{\text{SRDY}}$ going low in a second cycle to the same register class is due to the fact that the previous operation may not have completed, causing some hold-off time as described below.

Class:
- DIO–SIFDAT/INC, SIFADR, SIFADX
- SIF command/status (1 register)
- Pseudo-DMA (1 register)
- Others such as ACTL, DMALEN, SDMAADR, SDMAADX

The worst-case $\overline{\text{SRDY}}$ delay calculation for back-to-back read or write to a register of the same class is defined below.

Class DIO	17Q + 70 ns + cycle loss
Class SIFCMD/SIFSTS	8Q + 70 ns
Class SDMADAT	8(0.5T) + 70 ns + cycle loss
Class Others	70 ns

$$\text{Where } Q = \frac{1}{\text{Adapter Busclk Speed} \times 4}$$

The 70 ns for each register class is the minimum strobe time.

The cycle loss is the number of adapter cycles the DIO operation may be held off as defined in Table 3–8. This may be up to 6 cycles (1.5 μs) for the PH at 4 Mbits and the adapter at 4 MHz.

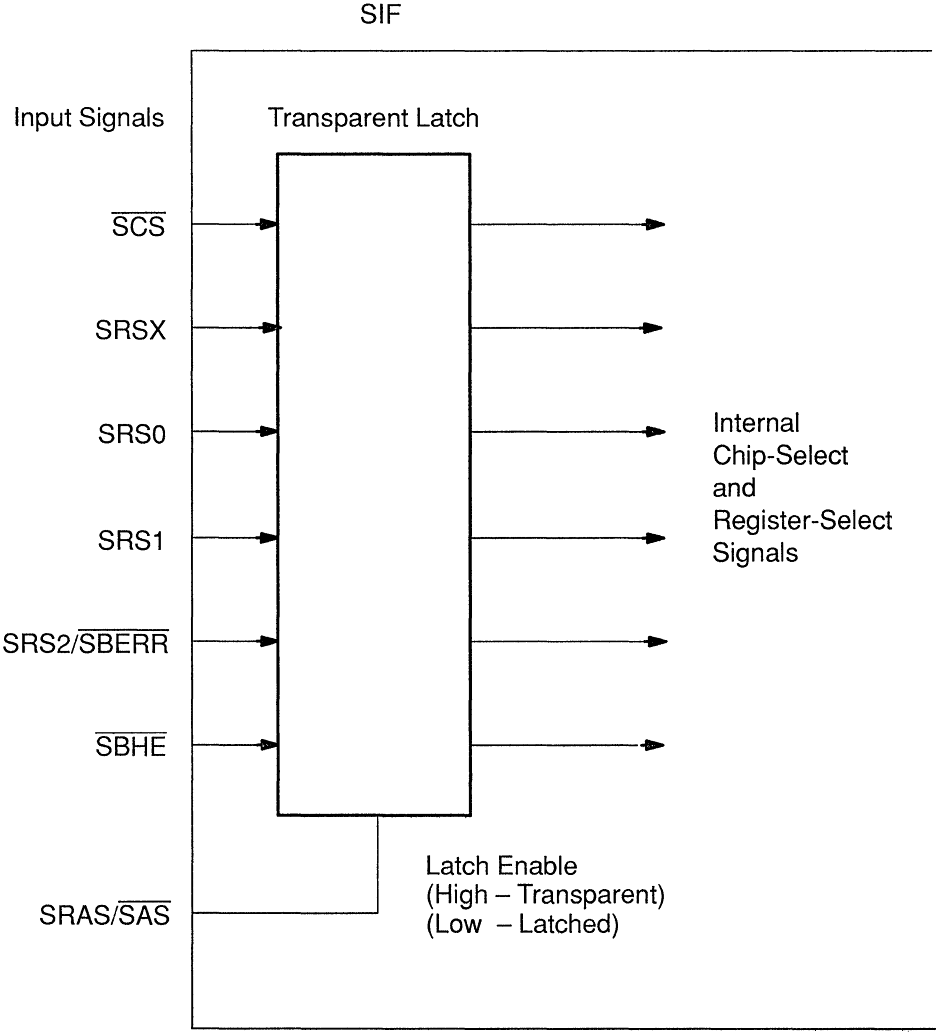
Table 3–8. DIO Cycles Held Off

Ring Speed	Local Bus Frequency	Number of Cycles Held Off
4 Mbit	3 MHz	Up to 8 cycles
4 Mbit	4 MHz	Up to 6 cycles
16 Mbit	4 MHz	Up to 12 cycles

Q represents one quarter of an adapter bus cycle (that is, 62.5 ns if the adapter is running at 4 MHz bus speed).

0.5T implies half-system (or host) clock cycles.

Figure 3–10. Internal Transparent Latch for DIO in 808x Mode



In the discussion that follows, the description of DIO timing is based on the nonmultiplexed option described in the preceding paragraph. The basic timing presented, however, applies to the multiplexed option as well, with the exception that the chip-select and register-select inputs must remain valid only while they are being latched.

3.4.1.2 808x DIO Read Cycle

Figure 3–11 depicts an 808x-mode DIO read cycle. The sequence of events is explained below.

1) In the quiescent state, the system deasserts SCS, SIACK, SRD, and SWR. The SIF floats the SAD bus, and drives both SDDIR and SDBEN high. The SIF also floats its SRDY pin, but because SRDY has a pullup resistor (nominal 4.7 kΩ) attached to it, the signal floats high (negated).
2) Starting the DIO cycle, the system (bus master) asserts a register address on the SBHE, SRSX, and SRS0–SRS2 lines, and asserts SRAS high. In 8-bit mode,SBHE is ignored.
3) The system asserts SCS low, keeping SIACK negated high.
4) The system optionally deasserts SRAS low, latching the SRSX, SRS0–SRS2, SBHE, and SCS inputs. SRAS may be tied high if the SRSX, SRS0–SRS2, and SCS inputs can be held throughout the cycle.
5) The system asserts SRD low, keeping SWR negated high. The system need not necessarily drive SCS before SRD.
6) When the SIF detects SCS and SRD asserted and DIO busy is not asserted, the SIF starts the internal register access.
7) Once the SIF has started the internal register access, the SIF drives the output read data, with odd parity on the SPL and SPH lines. If the system is reading only a single byte, the SIF drives stable data and parity on the other byte.
8) The SIF (bus slave) drives SDBEN low asynchronously when it detects the internal access has started. Since SDDIR remains high, this enables the SAD outputs onto the system bus.

Table 3–9. 808x DIO Read Data Bus Controls

808x 16-Bit Mode Read			
BUS Controls		**SAD Outputs**	
$\overline{\text{SBHE}}$	SRS2	SADH, SPH	SADL, SPL
0	0	Data MSB	Data LSB
0	1	Data MSB	Don't care
1	0	Don't care	Data MSB
1	1†	Undefined	Undefined
808x 8-Bit Mode Read			
BUS Controls		**SAD Outputs**	
$\overline{\text{SBHE}}$	SRS2	SADH, SPH	SADL, SPL
X	0	Don't care	Data LSB
X	1	Don't care	Data MSB

† Illegal input combination

9) When the SIF detects the register access in progress and the recognition of both $\overline{\text{SCS}}$ and $\overline{\text{SRD}}$ still active, the SIF drives $\overline{\text{SRDY}}$ low. The SIF guarantees that output data is valid before driving $\overline{\text{SRDY}}$.

10) While $\overline{\text{SRDY}}$ is asserted, the system should not remove the register address($\overline{\text{SCS}}$, SRSX, SRS0–SRS2, and $\overline{\text{SBHE}}$) unless they have been latched by SRAS.

11) After $\overline{\text{SRDY}}$ is asserted, the system deasserts $\overline{\text{SRD}}$ and takes $\overline{\text{SCS}}$ high.

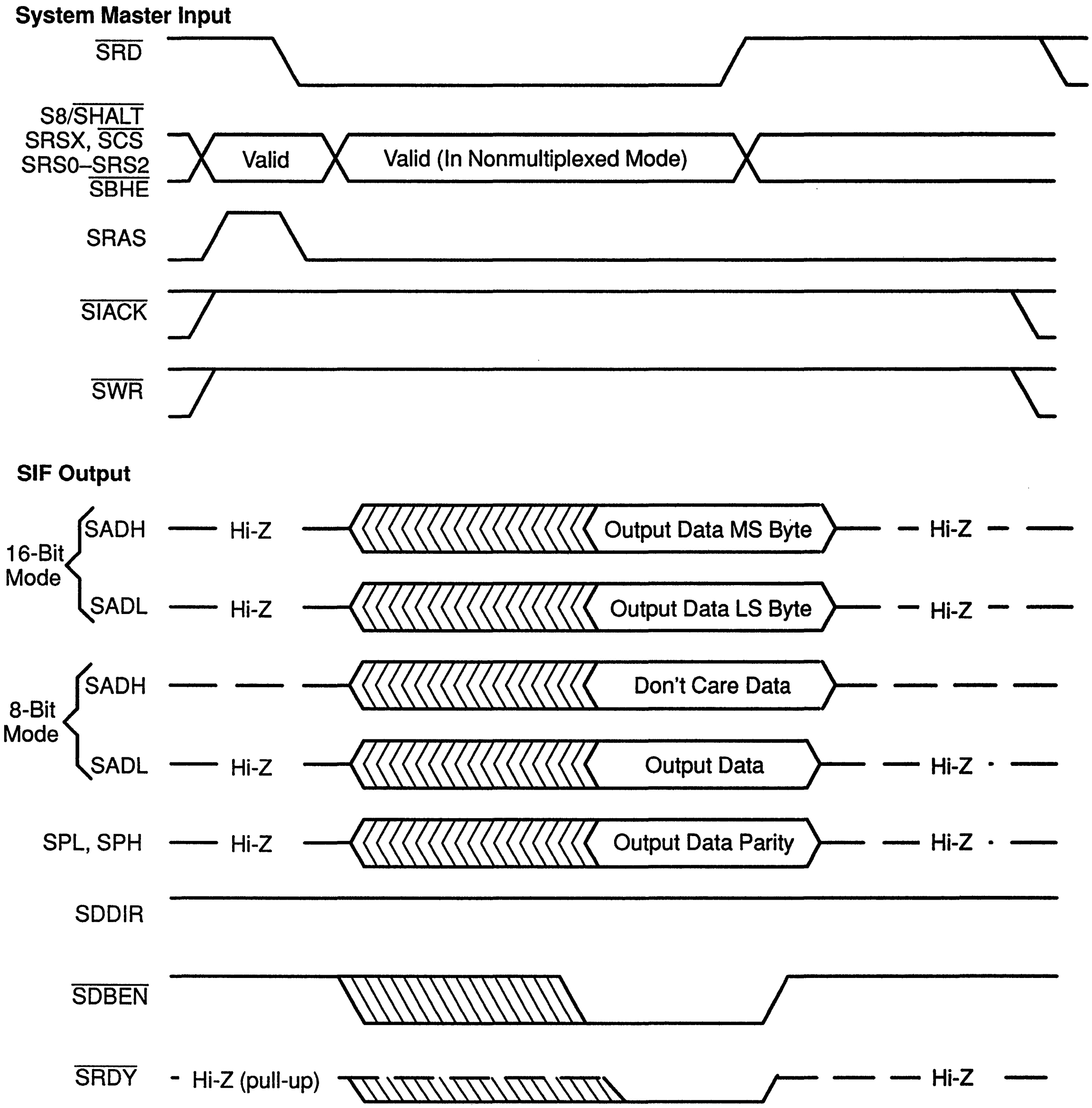
12) When either $\overline{\text{SRD}}$ or $\overline{\text{SCS}}$ are negated high, the SIF asynchronously deasserts $\overline{\text{SDBEN}}$ high. The SIF also asynchronously drives $\overline{\text{SRDY}}$ high at this time.

13) When the SIF detects that it has driven its $\overline{\text{SRDY}}$ signal high at the pin, it floats the signal, returning to the quiescent state.

14) The system must hold $\overline{\text{SRD}}$ negated for a minimum period before reaccessing the SIF; also, SCS need not be negated between consecutive accesses to the same 16-bit register address (such as SRSX and SRS0–SRS2 do not change). $\overline{\text{SCS}}$ may be kept asserted, however, between accesses to the two bytes of the same 16-bit register address (such as only SRS2 changes) or between accesses to two different 16-bit register addresses (such as SRSX, SRS0, or SRS1 change). The system must also hold the previously inactive signals $\overline{\text{SWR}}$ and $\overline{\text{SIACK}}$ negated for a minimum period following a read operation.

As explained earlier, the SIF reloads the SIFDAT register after every write to the high byte of SIFADR, or read of SIFDAT, and writes the contents of SIFDAT to adapter memory after every write to the SIFDAT register. If the system attempts to read SIFDAT before this operation is complete, the SIF delays its assertion of $\overline{\text{SRDY}}$, $\overline{\text{SDBEN}}$, and valid DATA. If a read of SIFCMD/STS is performed immediately after a write to either byte, the read delays the assertion of $\overline{\text{SRDY}}$, $\overline{\text{SDBEN}}$, and valid DATA.

Figure 3–11. 808x-Mode DIO Read Cycle



3.4.1.3 808x DIO Write Cycle

The sequence of events in an 808x DIO write cycle is explained below.

1) In the quiescent state, the system deasserts $\overline{\text{SCS}}$, $\overline{\text{SIACK}}$, $\overline{\text{SRD}}$, and $\overline{\text{SWR}}$. The SIF floats the SAD bus, and drives both SDDIR and $\overline{\text{SDBEN}}$ high. The SIF also floats its $\overline{\text{SRDY}}$ pin, but because $\overline{\text{SRDY}}$ has a pull-up resistor (nominal 4.7 kΩ) attached to it, the signal floats high (negated).

2) Starting the DIO cycle, the system (bus master) asserts a register address on the $\overline{\text{SBHE}}$, SRSX, and SRS0–SRS2 lines, and asserts SRAS high. In 8-bit mode,$\overline{\text{SBHE}}$ is ignored.

3) The system asserts $\overline{\text{SCS}}$ low, keeping $\overline{\text{SIACK}}$ negated high.

4) The system optionally deasserts SRAS low, latching the SRSX, $\overline{\text{SBHE}}$, SRS0–SRS2, and $\overline{\text{SCS}}$ inputs. SRAS may be tied high if the $\overline{\text{SBHE}}$, SRS0–SRS2, and $\overline{\text{SCS}}$ inputs can be held throughout the cycle.

5) The system asserts $\overline{\text{SWR}}$ low, keeping $\overline{\text{SRD}}$ negated high. The system need not necessarily drive $\overline{\text{SCS}}$ before $\overline{\text{SWR}}$.

6) The SIF (bus slave) drives SDDIR low asynchronously from the activation of $\overline{\text{SWR}}$ and $\overline{\text{SCS}}$. This sets up the direction for the system bus drivers.

7) When the SIF detects SDDIR and $\overline{\text{SWR}}$ in their active states, the SIF asserts $\overline{\text{SDBEN}}$ low, enabling the system bus lines onto the SIF SAD lines.

8) If the SIF is busy performing a read or write to adapter memory as a result of an earlier DIO operation, it waits at this point until the internal operation is complete.

9) When the SIF asynchronously samples both $\overline{\text{SCS}}$ and $\overline{\text{SWR}}$ low and DIO busy is not asserted, it starts the internal register access.

10) Once the SIF has detected ready to write to a register, it drives $\overline{\text{SRDY}}$ to its active state (low).

Table 3–10. 808x DIO Write Data Bus Controls

808x 16-Bit Mode Write			
BUS Controls		**SAD Inputs**	
$\overline{\text{SBHE}}$	SRS2	SADH, SPH	SADL, SPL
0	0	Data MSB	Data LSB
0	1	Data MSB	Don't care
1	0	Don't care	Data LSB
1	1†	Don't care	Don't care
808x 8-Bit Mode Write			
BUS Controls		**SAD Inputs**	
$\overline{\text{SBHE}}$	SRS2	SADH, SPH	SADL, SPL
X	0	Don't care	Data LSB
X	1	Don't care	Data MSB

† Illegal input combination

11) When $\overline{\text{SRDY}}$ is asserted (low), the system performs the following actions in this order:

    a) Deasserts the $\overline{\text{SWR}}$ and $\overline{\text{SCS}}$ strobes first.

    b) Removes the register address (SRSX, SRS0–SRS2, and $\overline{\text{SBHE}}$) if it has not already done so.

    c) Removes the write data.

12) When the SIF detects the $\overline{\text{SWR}}$ deasserting, it latches the DATA internally to the register. After latching the input data, the SIF deasserts the $\overline{\text{SDBEN}}$ and $\overline{\text{SRDY}}$ signals high, disabling the system bus onto the SAD bus. The SIF effectively holds the SDDIR signal active low beyond the deassertion of the $\overline{\text{SDBEN}}$ signal.

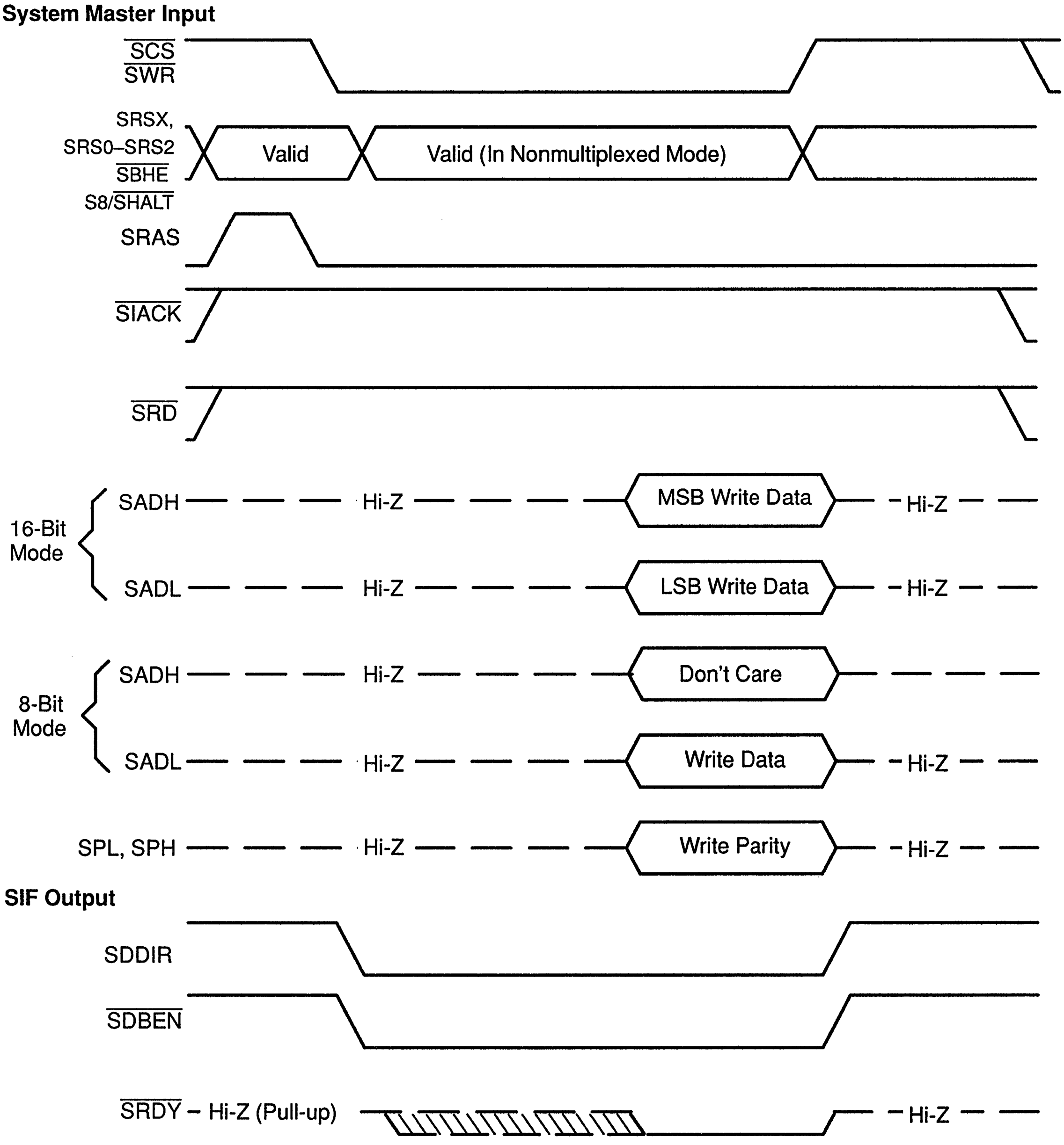
13) When the SIF detects that it has driven its $\overline{\text{SRDY}}$ signal high at the pin, it floats the signal, allowing the external pull-up to maintain the signal's high level.

14) The system must hold $\overline{\text{SWR}}$ negated for a minimum period before reaccessing the SIF. $\overline{\text{SCS}}$ need not be deasserted between accesses to the high and low bytes of the same 16-bit register or between accesses to different 16-bit registers. The system must also hold the previously inactive signals $\overline{\text{SRD}}$ and $\overline{\text{SIACK}}$ negated for a minimum period.

If the host attempts to access (such as read or write) any DIO register before the internal SIF logic has synchronized the contents of an earlier write cycle to SIFCMD or SIFSTS to the adapter internal data bus, then the SIF will delay its internal start of that second DIO cycle (see section 3.4.1.1.1).

As explained earlier, the SIF reloads the SIFDAT register after every write to the high byte of SIFADR, and will write the contents of SIFDAT to adapter memory after every write to the SIFDAT register. If the system attempts to write to SIFADR or to SIFDAT before this operation is complete, the SIF delays its assertion of $\overline{\text{SRDY}}$.

Figure 3–12. 808x-Mode DIO Write Cycle



3.4.1.4 808x Interrupt Acknowledge Sequence

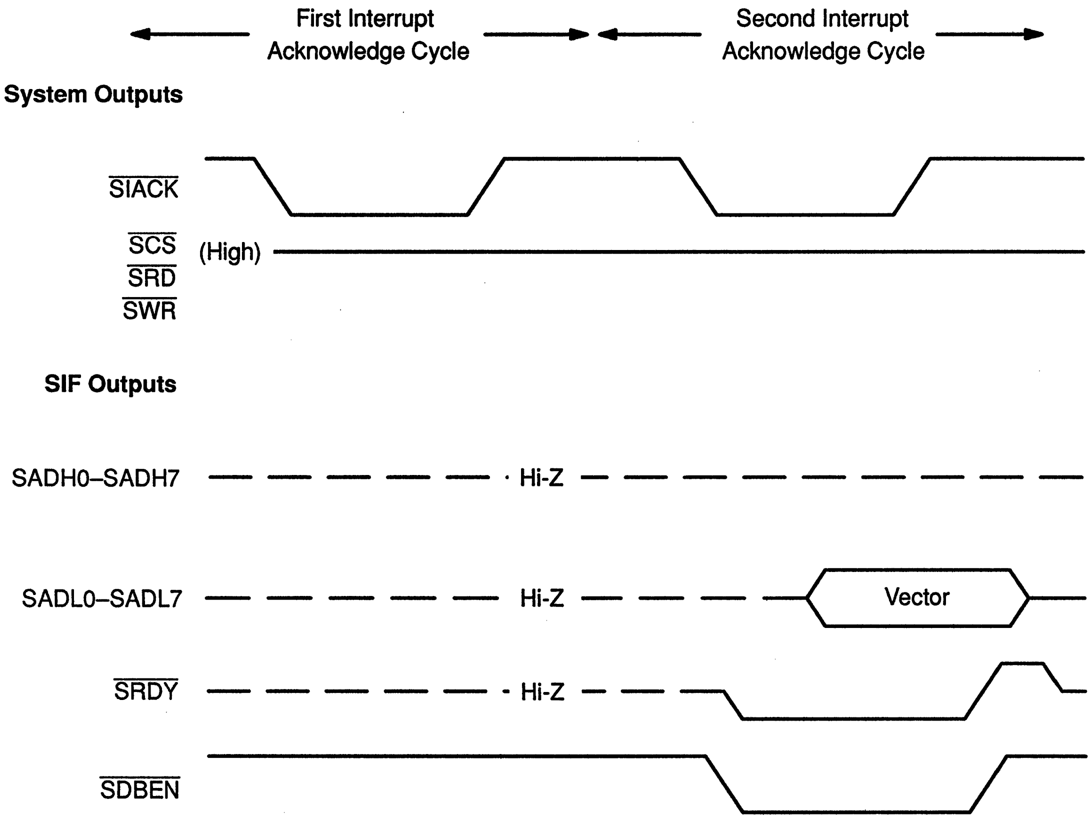
The SIF has the option of placing an 8-bit interrupt vector on the system bus during a special system bus operation known as an interrupt acknowledge cycle. The adapter microcode initializes the 8-bit vector by writing to the

SIFVEC register. The system reads the interrupt vector on the system bus, with the timing shown in Figure 3–13.

No synchronization is performed between the adapter internal bus write and the system read. Adapter microcode must write to SIFVEC at least one LBCLK cycle before the system can read it reliably.

In 808x mode, the SIF outputs the interrupt vector on the second of two consecutive low-going pulses of the $\overline{\text{SIACK}}$ pin. **On the first pulse, the SIF does not drive any of its I/O pins.** On the second pulse, the SIF operates according to the DIO read operation described above, where the $\overline{\text{SIACK}}$ signal, rather than the combination of $\overline{\text{SCS}}$ and $\overline{\text{SRD}}$, serves as the data strobe.

Figure 3–13. Interrupt Acknowledge Timing for 808x Mode



The interrupt acknowledge sequence consists of two back-to-back interrupt acknowledge (IACK) cycles. $\overline{\text{SRD}}$ or $\overline{\text{SWR}}$ must not be asserted between these cycles. During the second cycle, the SIF outputs the contents of the SIFVEC (SIF vector) register onto SADL0–SADL7. Timing for the second interrupt acknowledge cycle is identical to that for the DIO read, with $\overline{\text{SIACK}}$ used instead of the AND of $\overline{\text{SCS}}$ and $\overline{\text{SRD}}$. The SINTR output is not affected by the IACK sequence. The SCS input must be negated while $\overline{\text{SIACK}}$ is asserted. The input signals on $\overline{\text{SRD}}$ and $\overline{\text{SWR}}$ must remain inactive-high for the duration of the sequence shown in Figure 3–13. The SRSX and SRS0–SRS2 inputs are ignored. The SIF does not check that SINTR is also asserted when responding to a $\overline{\text{SIACK}}$ pulse.

3.4.2 DIO for 68xxx Interface

In 68xxx mode, the SIF bus control lines are defined to be similar to a 68xxx–type microprocessor. The 68xxx interface does not have an address bit A0 to select which byte of a word is accessed; rather, it provides separate upper and lower data strobes. In 68xxx mode, the $\overline{\text{SWR/SLDS}}$ pin serves as the (lower) data strobe, and the $\overline{\text{SRD/SUD}}$ pin serves as the upper data strobe. The system processor may access a single byte by asserting either $\overline{\text{SUDS}}$ or $\overline{\text{SLDS}}$ separately.

3.4.2.1 68xxx-Mode DIO Read Cycle

The timing for a 68xxx mode system interface DIO read is shown in Figure 3–14. The sequence of events is explained below.

1) In the quiescent state, the system deasserts $\overline{\text{SCS}}$, $\overline{\text{SIACK}}$, $\overline{\text{SUDS}}$, and $\overline{\text{SLDS}}$. SRNW is undefined. The SIF floats the SAD bus and drives both SDDIR and $\overline{\text{SDBEN}}$ high. The SIF also floats its $\overline{\text{SDTACK}}$ pin, but because the pin has a pull-up resistor (nominal 4.7k Ω) attached to it, the signal floats high (negated).

2) Starting the DIO cycle, the system (bus master) asserts a register address on the SRSX and SRS0–SRS1 lines and drives SRNW high, indicating a read cycle. The $\overline{\text{SAS}}$ signal is an input to the SIF but is not used in controlling the $\overline{\text{SRSX}}$ and SRS0–SRS1, $\overline{\text{SCS}}$ transparent latch.

3) The system asserts $\overline{\text{SCS}}$ low, keeping $\overline{\text{SIACK}}$ negated high.

4) The system asserts either $\overline{\text{SUDS}}$ or $\overline{\text{SLDS}}$ low for an 8-bit access, or both low for a 16-bit access.

5) When the SIF detects $\overline{\text{SCS}}$, when either $\overline{\text{SUDS}}$ or $\overline{\text{SLDS}}$, or both, have been asserted, and when DIO busy is not asserted, this condition starts the internal register access.

6) The SIF (bus slave) drives $\overline{\text{SDBEN}}$ low asynchronously when it detects that an internal access has started. Because SDDIR remains high, this enables the SAD outputs onto the system bus.

7) When the SIF has detected the internal register access, it drives data with odd parity on the SPH and SPL lines and asserts $\overline{\text{SDTACK}}$. If the

system is reading only a single byte, the SIF floats the half of SAD not containing data, as shown below:

Table 3–11. 68xxx DIO Read Data Bus Controls

68xxx-Mode Read			
BUS Controls		**SAD Outputs**	
$\overline{\text{SUDS}}$	$\overline{\text{SLDS}}$	SADH, SPH	SADL, SPL
0	0	Data MSB	Data LSB
0	1	Data MSB	Hi-Z
1	0	Hi-Z	Data MSB
1	1†	Undefined	Undefined

† Illegal input combination

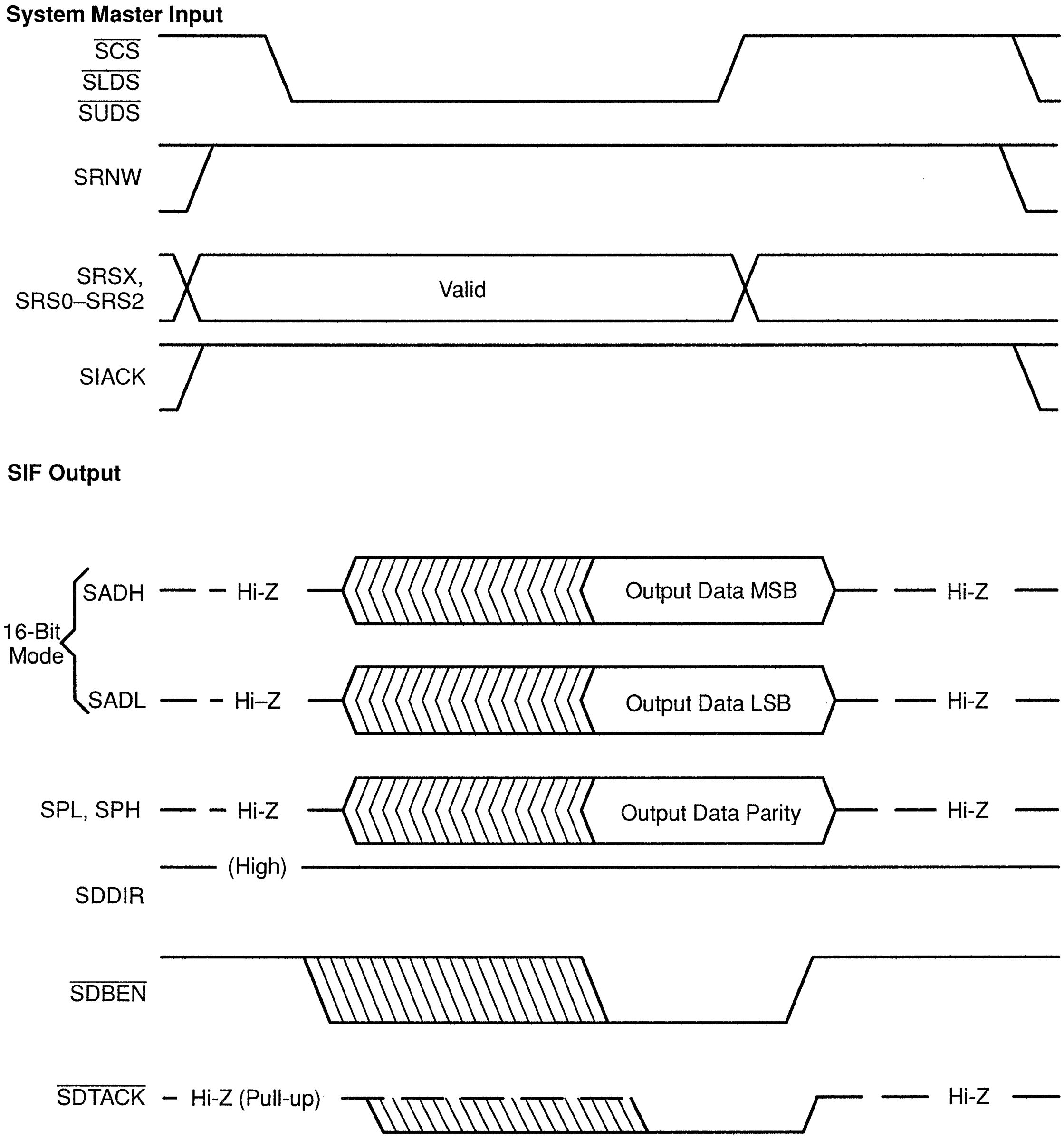
8) When $\overline{\text{SDTACK}}$ is asserted, the system may deassert the register strobes $\overline{\text{SUDS}}$ and $\overline{\text{SLDS}}$ and remove the register address (SRSX, SRS0–SRS1, and SRNW).

9) When both data strobes and $\overline{\text{SCS}}$ are negated high, the SIF asynchronously deasserts $\overline{\text{SDBEN}}$ and $\overline{\text{SDTACK}}$ high.

10) When the SIF detects that it has driven its $\overline{\text{SDTACK}}$ signal high at the pin, it floats the signal, returning to the quiescent state.

11) The system must hold both data strobes negated for a minimum period before reaccessing the SIF. $\overline{\text{SCS}}$ may be kept asserted between accesses to the high and low bytes of the same 16-bit register, but must be negated between accesses to different 16-bit registers. The system must also hold the previously inactive signal $\overline{\text{SIACK}}$ negated for a minimum period.

Figure 3–14. 68xxx-Mode DIO Read Cycle Timing



3.4.2.2 68xxx-Mode DIO Write Cycle Timing

The sequence of events in a 68xxx DIO write cycle is explained below.

1) In the quiescent state, the system deasserts SCS, SIACK, SUDS, and SLDS. SRNW is a don't care. The SIF floats the SAD bus and drives both SDDIR and SDBEN high. The SIF also floats its SDTACK pin, but because the pin has a pull-up resistor (nominal 4.7 kΩ) attached to it, the signal floats high (negated).

2) Starting the DIO cycle, the system (bus master) asserts a register address on the SRSX and SRS0–SRS1 lines, and drives SRNW low, indicating a write cycle. The SAS pin is an input to the SIF but is not used in controlling the SRSX, SRS0–SRS1, SCS transparent latch.

3) The system asserts SCS low, keeping SIACK negated high.

4) The system asserts either SUDS or SLDS low for an 8-bit access, or both low for a 16-bit access.

5) The SIF (bus slave) drives SDDIR low asynchronously from the activation of SCS and either SUDS or SLDS. This sets up the direction for the system bus drivers.

6) Once the SIF detects SDDIR valid at the pin, it asserts the SDBEN low, enabling the system bus lines onto the SIF SAD lines.

7) If the SIF is busy performing a read or write to adapter memory as a result of an earlier DIO operation, it waits at this point until the internal operation is complete.

8) Once the SIF has detected the access is in progress, it asserts SDTACK low.

Table 3–12. 68xxx DIO Write Data Bus Controls

68xxx-Mode Write			
BUS Controls		**SAD Outputs**	
SUDS	SLDS	SADH, SPH	SADL, SPL
0	0	Data MSB	Data LSB
0	1	Data MSB	Don't care
1	0	Don't care	Data MSB
1	1†	Undefined	Undefined

† Illegal input combination

9) When $\overline{\text{SDTACK}}$ is low, the system performs the following actions.

   a) Deasserts the $\overline{\text{SUDS}}$ and $\overline{\text{SLDS}}$ strobes.

   b) Removes the register address (SRSX and SRS0–SRS2) and SRNW.

   c) Removes the write data; and

10) When both data strobes and $\overline{\text{SCS}}$ are negated high, the SIF latches the incoming data into the register accessed and asynchronously drives $\overline{\text{SDBEN}}$ and $\overline{\text{SDTACK}}$ high.

11) When SDBEN is seen high at the pin, the SIF drives SDDIR high.

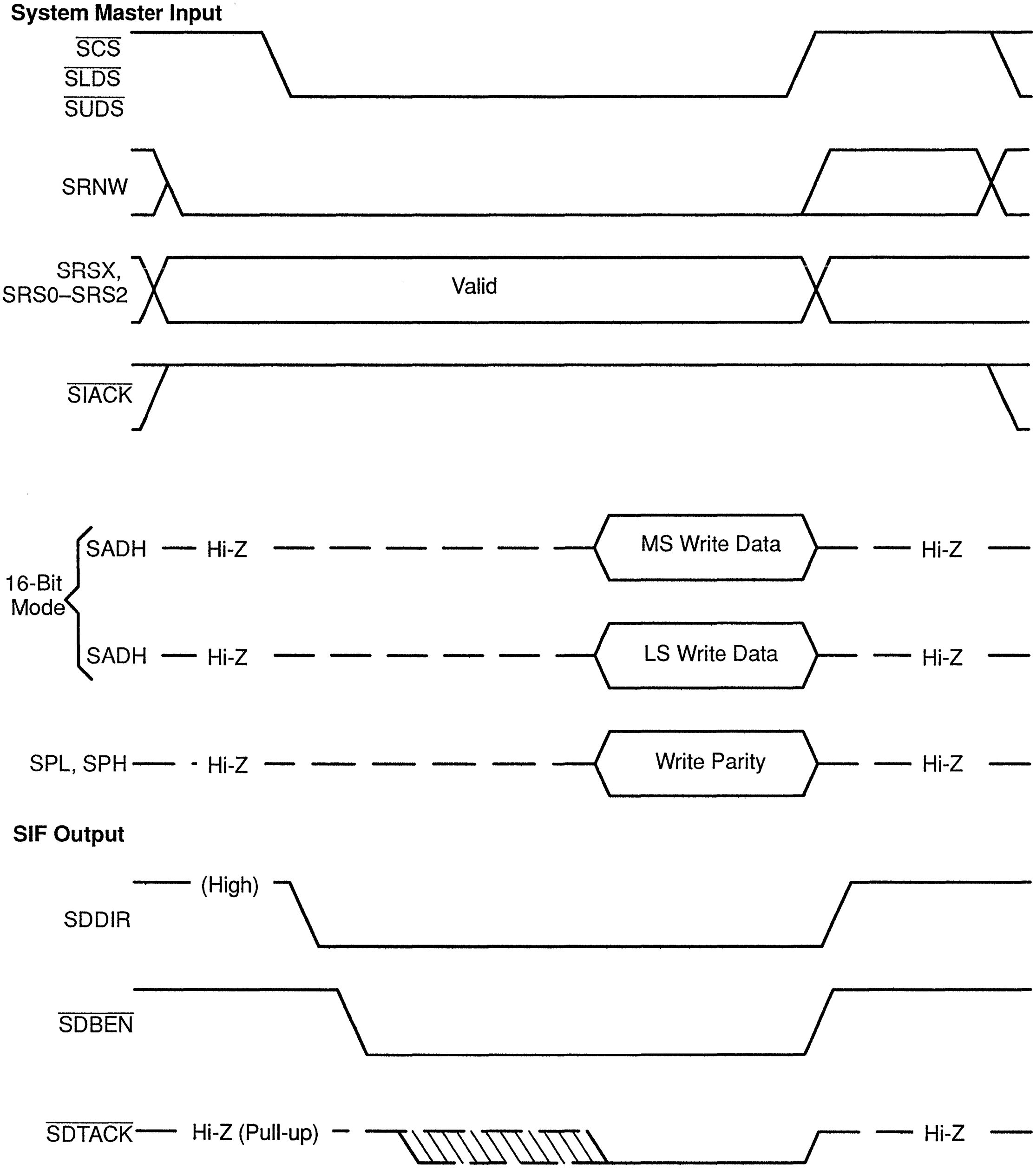
12) When the SIF detects that it has driven its $\overline{\text{SDTACK}}$ signal high at the pin, it floats the signal, allowing the external pull-up to maintain a high level.

13) The system must hold both data strobes negated for a minimum period before reaccessing the SIF. $\overline{\text{SCS}}$ need not be deasserted between accesses to the high and low bytes of the same 16-bit register, or between accesses to different 16-bit registers. The system must also hold the previously inactive signal $\overline{\text{SIACK}}$ negated for a minimum period.

If the host attempts to access (such as read or write) any DIO register before internal SIF logic has synchronized the contents of an earlier write cycle to SIFCMD or SIFSTS to the adapter internal data bus, then the SIF will delay its internal start of that second DIO cycle (see section 3.4.1.1.1).

As also mentioned earlier, if the SIF has not completed reloading SIFDAT or writing its value into adapter memory, it delays its assertion of $\overline{\text{SDTACK}}$ until the operation is complete.

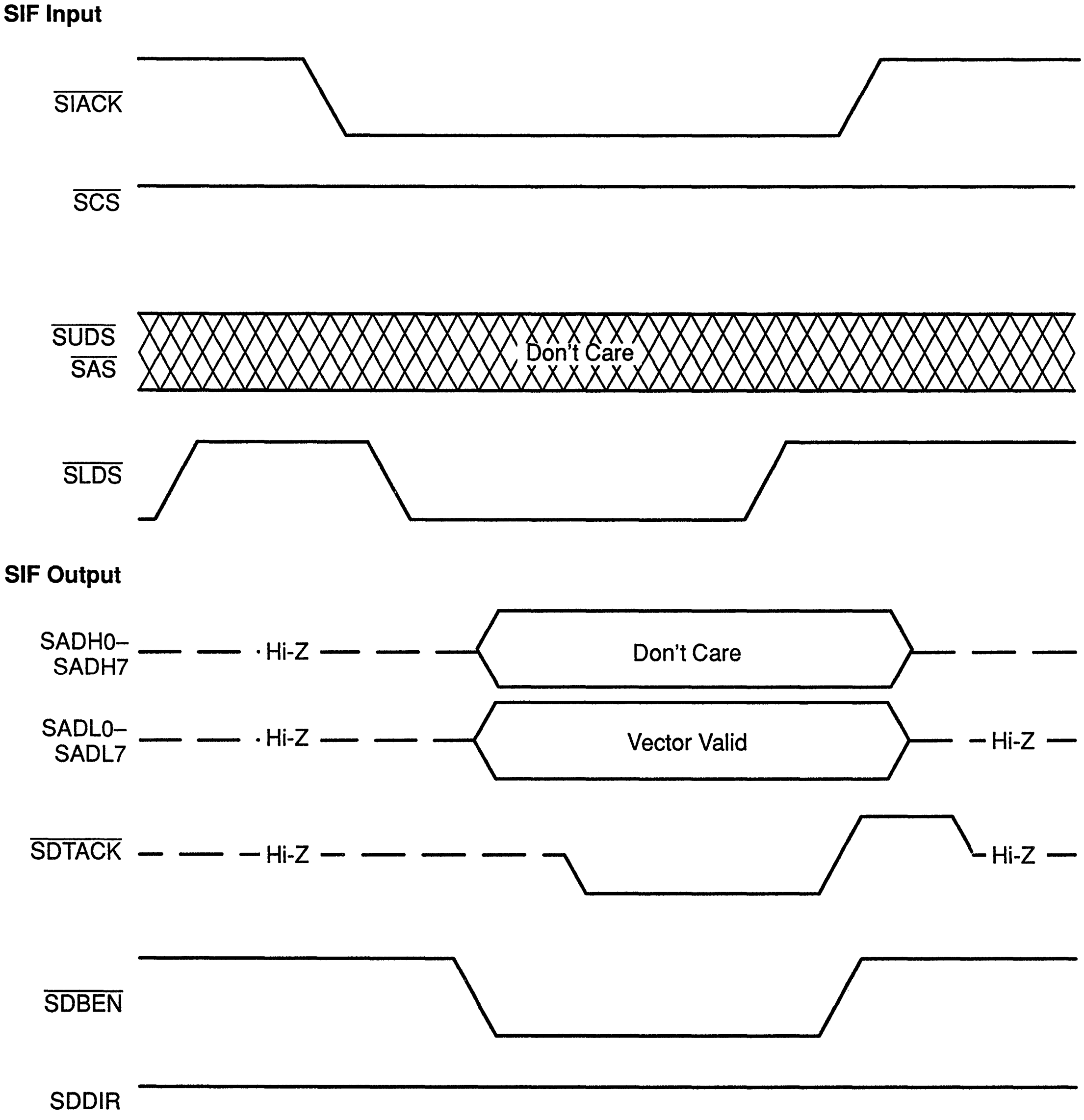
Figure 3–15. 68xxx-Mode DIO Write Cycle Timing



3.4.2.3 68xxx Interrupt Acknowledge Cycle

The timing for the 68xxx-mode interrupt acknowledge cycle is shown in Figure 3–16. This timing is almost identical to a 68xxx 8-bit read cycle, with $\overline{\text{SIACK}}$ used instead of $\overline{\text{SCS}}$ as a chip select. The signal presented to the $\overline{\text{SIACK}}$ input on the SIF is decoded from the FC(0–2) and A(1–3) outputs of the 68xxx microprocessor. The SIF does not check that $\overline{\text{SIRQ}}$ is asserted when responding to an $\overline{\text{SIACK}}$ pulse.

Figure 3–16. 68xxx-Mode Interrupt Acknowledge Cycle Timing



3.4.3 DMA Operations

The DMA channel of the system interface provides a full 32 bits of address with which to access up to 4 gigabytes of system memory. The throughput capability of the DMA channel is matched to that of the host system and can exceed the actual ring transmission speed. The maximum DMA transfer rate corresponds to one word per four user-system clock periods. System DMA occurs concurrently with ring DMA and communications processor program execution. Arbitration for the internal adapter buses is managed by the MIF. The DMA channel can be programmed for either burst-mode or cycle-steal operation.

The 32-bit system address used for DMA transfers is maintained in the system DMA address registers internal to the SIF. The contents of these registers are brought out onto the multiplexed address/data pins of the SIF and can be latched externally for presentation to the user system.

Since only 16 bits of address can be output at any instant on the 16 address/ data lines, the 16 most significant address bits must be multiplexed (and latched) separately from the lower 16 bits of address. These are called the extended address bits of the SIF DMA address. Two separate latch-enable signals are provided by the SIF for the demultiplexing of the address. The extended address latch is updated only when the SIF controls the sytems bus and is performed in an extra T phase (TX) of the system bus memory transfer cycle. For systems requiring only 16 bits of address, no external latch is needed for the 16 address extension bits, although they will still continue to be output by the SIF, as before.

Due to detailed timing considerations, the SIF performs a TX cycle every time it acquires the system bus. In cycle steal mode, then, the SIF always performs a TX cycle; in burst mode, the SIF performs a TX cycle on the first memory cycle, and thereafter only when the 16 most significant bits of address are changed due to a carry propagated from the lower 16 bits. On a TX cycle, the 16 most significant bits of the address are placed on the SAD bus. In the following T1 cycle, the 16 least significant address bits are placed on the SAD bus.

The TMS380C16 software controls the DMA interface. Although the DMA channel hardware is capable of performing transfers in only one direction at a time, the software may present a more powerful logical interface to the system. The software, for instance, may transfer a frame as a sequence of separate buffers or maintain the appearance of separate receive and transmit channels. When writing to user memory, the software can interrupt the user processor when its DMA channel has written the last buffer of the frame.

To set up a DMA operation, the communications processor

1) Loads the starting address of the system memory block.
2) Loads the number of bytes to transfer.
3) Loads the starting address of adapter memory.
4) Starts DMA by writing a word to the SIF control register.

The system DMA (SDMA) channel is composed of two separate DMA controllers. One controller manages the interface to the system bus (for accesses of the system memory), and the other controller manages the interface to the adapter's internal address and data buses. These two controllers exchange data asynchronously through a system DMA data FIFO. The SDMA FIFO, along with the other registers in the DMA contoller that manage transfers over the system bus, cannot be written to by the user processor and only the system address and DMA-length registers can be read by the user processor; they are, however, **all managed by the TMS380C16 microcode**.

The SDMA FIFO maintains a parity bit for each byte. On system-to-adapter transfers, the parity bit read from the system bus (such as SPH or SPL) is checked but not stored. Rather, the SIF regenerates parity internally and places it with the data into the system side of FIFO. This regenerated parity bit is supplied with the data it corresponds to when a write is performed to adapter memory. On adapter-to-system transfers, the parity bit read from the adapter memory is checked but not stored; rather, parity is regenerated and placed into the FIFO on the adapter internal bus side. This parity is driven onto the system bus when the FIFO data is written to system memory. The parity checker on adapter-to-system transfers can be disabled by setting the PEN bit in ACTL to zero. In this case, the parity generator still operates to enable parity to be supplied to the system memory.

3.4.4 808x Mode DMA Operation

3.4.4.1 808x Bus Transfer

In 808x mode, the SIF controls the bus in the same manner that an intel-type microprocessor would. The general system timing for DMA data transfers between the SIF pins and the system memory is as shown in Figure 3–17.

Figure 3–17. 808x-Mode DMA Read Timing

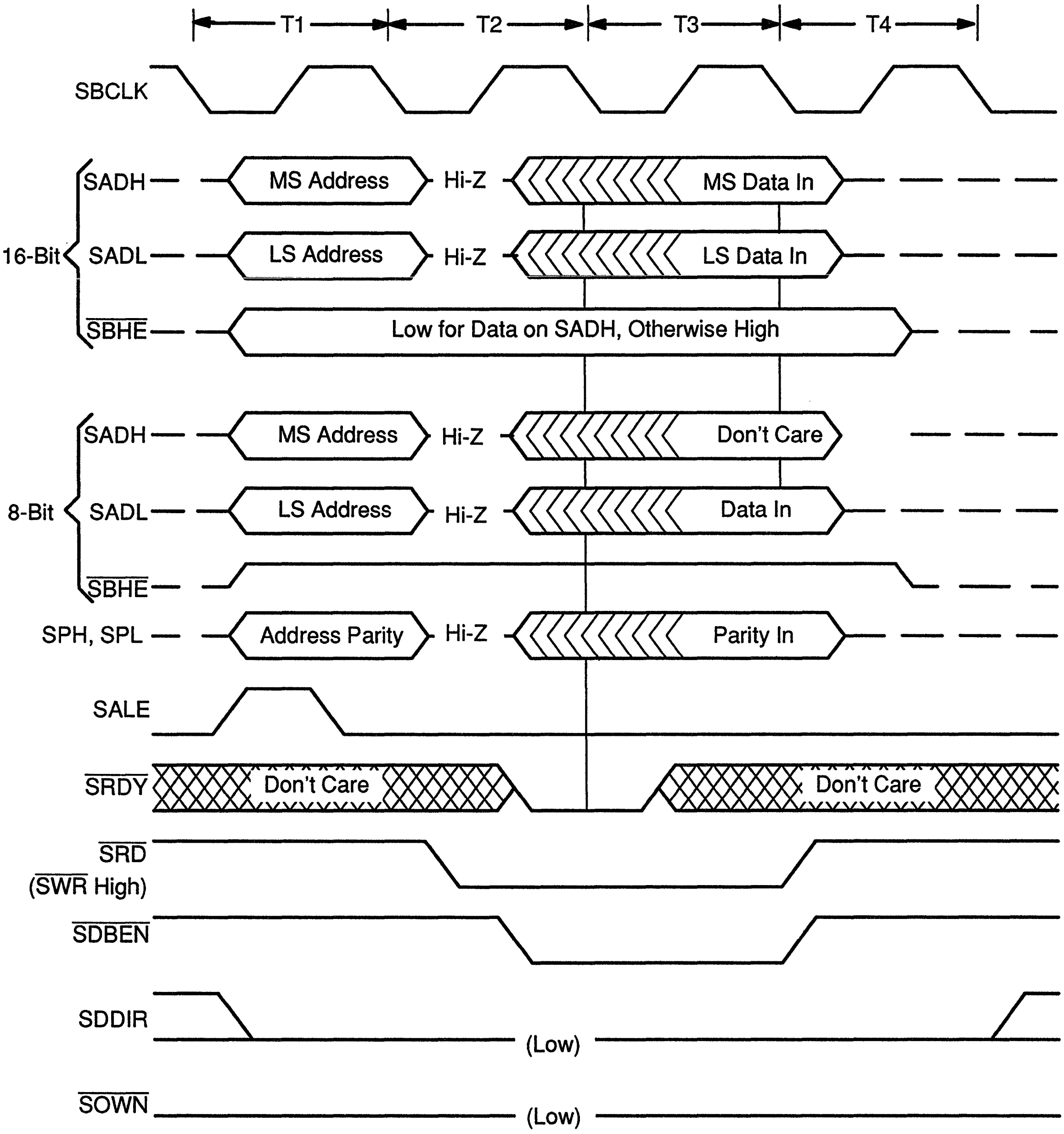
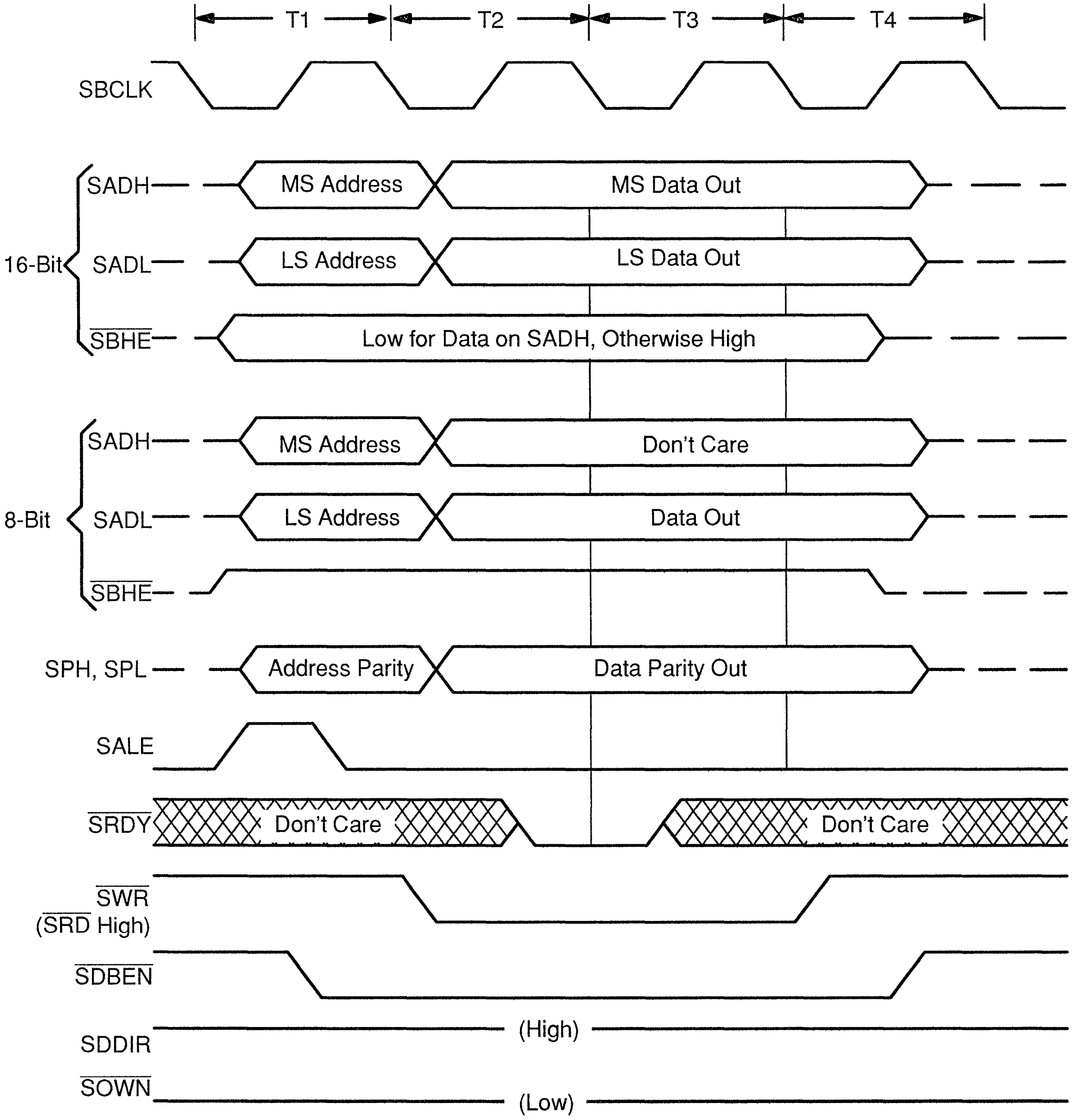


Figure 3–18. 808x-Mode DMA Write Timing



At the start of the DMA cycles shown in Figure 3–17 and Figure 3–18, the SALE output strobes the contents of address/data lines SADH0–SADH7 and SADL0–SADL7 into an external latch (such as 373). 808x processors multiplex address and data in the same manner. With the SIF's SALE output, the address/data bus can be demultiplexed with external latches to present to the system bus a nonmultiplexed interface with separate address and data buses. Assuming that a transparent latch is used to capture the address from SADH0–SADH7 and SADL0–SADL7, the demultiplexed address presented to the system becomes valid prior to the falling edge of SALE. This is recommended in order to provide memories with as long an access time (from address valid) as possible. In 808x 16-bit mode, both byte and word DMA cycles are supported. The least-significant address bit, SADL(7) is the equivalent of the 808x microprocessor's A0 address bit, used to select the low byte; the $\overline{\text{SBHE}}$/SRNW output is the equivalent of a16-bit 808x processor's $\overline{\text{BHE}}$ signal. On single byte transfers (either reads or writes), the unused half of the SAD lines is placed in the high-impedance state.

In 808x 8-bit mode, data is transferred only over SADL0–SADL7, the $\overline{\text{SBHE}}$ signal is kept high, and the SADH15–SADH8 bus contains don't care data.

A DMA read cycle is shown in Figure 3–17. After the address has been latched, the SIF forces its address/data lines, SADH0–SADH7 and SADL0–SADL7, to high impedance and also drives $\overline{\text{SRD}}$ active-low to enable the read data from the slave device onto the address/data lines. The read data is strobed into the SIF on the falling edge of the clock cycle marked T3, and $\overline{\text{SRD}}$ goes high to cause the slave device to remove its data from the bus.

Note:

In this document, a system interface T state is specified to start following the falling edge of SBCLK, continue past its rising edge, and include the subsequent falling edge. For example, T3 high refers to the rising edge in the middle of state T3, and T3 low refers to the falling edge at the end of T3.

A DMA write cycle is shown in Figure 3–18. After the address has been latched, the SIF drives the write data onto the address/data lines and also drives $\overline{\text{SWR}}$ active-low to indicate to slave devices that write data is available on the address/data lines. The $\overline{\text{SWR}}$ strobe shown in the figure is roughly two SBCLK periods in duration. The rising edge of $\overline{\text{SWR}}$ should be used by slave devices to latch the data from the address/data lines.

Slave devices that require longer access times should drive $\overline{\text{SRDY}}$ high prior to the falling clock edge at the end of T2 in order to generate wait states. As long as $\overline{\text{SRDY}}$ remains high, the SIF continues to generate wait states. (T3 is repeated for each wait state requested by the slave device.) When the slave has had sufficient time to complete the access, it drives $\overline{\text{SRDY}}$ low to allow the SIF to complete the DMA cycle.

The SDBEN and SDDIR outputs permit the SIF to interface easily to a system bus through a set of external data bus transceivers (such as the 74LS245). $\overline{\text{SDBEN}}$ is an active-low data bus enable signal used to turn on the transceivers, and SDDIR indicates the direction in which the data is to be transmitted. SDDIR is driven low to enable read data from the system bus into the SIF, and high to enable write data from the SIF onto the system bus.

Before the SIF can control the bus, however, it must first obtain the bus from the current bus master and eventually return it. This bus arbitration mechanism is described below.

3.4.4.2 808x Bus Arbitration

The bus arbitration for 808x mode is shown in Figure 3–19 and is similar to the minimum-mode arbitration conventions for the 808x microprocessors.

In Figure 3–19, the SIF arbitrates for and obtains control of the system bus in preparation for one or more DMA cycles. The sequence of operations is described below:

1) The SIF asserts SHRQ high on the rising edge of SBCLK.

2) When the system bus master completes its operation, it asserts SHLDA high.

3) The SIF samples several lines at the falling edge of SBCLK: its own SHRQ, the asynchronous inputs SHLDA, and $\overline{\text{SBBSY}}$. If SHRQ, SHLDA, and $\overline{\text{SBBSY}}$ are all high at a sample time, the SIF asserts $\overline{\text{SOWN}}$ low on the first rising edge of SBCLK following the sample (I2 low).

4) SDDIR is driven to its proper state when $\overline{\text{SOWN}}$ is driven active.

5) The subsequent falling edge of SBCLK starts state TX of the first SDMA cycle. The SIF drives the 8 most significant bits of the 32-bit DMA address on the SADH bus and the next 8 most significant bits on the SADL bus. Figure 3–19 depicts the address driven out on TX cycles.

6) On the next rising edge of SBCLK (TX high), the SIF drives the bus controls $\overline{\text{SRD}}$ and $\overline{\text{SWR}}$ high.

7) On the next falling edge (TX low), the SIF drives the lowest 8 bits of address onto the SADL bus, the next 8 most significant bits onto the SADH bus, and $\overline{\text{SBHE}}$ to its value for the transfer.

8) On the next falling edge of SBCLK (T1 low), the SIF asserts $\overline{\text{SWR}}$ low for a write cycle or $\overline{\text{SRD}}$ low for a read cycle.

Note that a minimum of two idle SBCLK cycles occur in the bus handoff, where no data transfers are taking place.

Figure 3–19. SIF Acquires System Bus—808x Mode

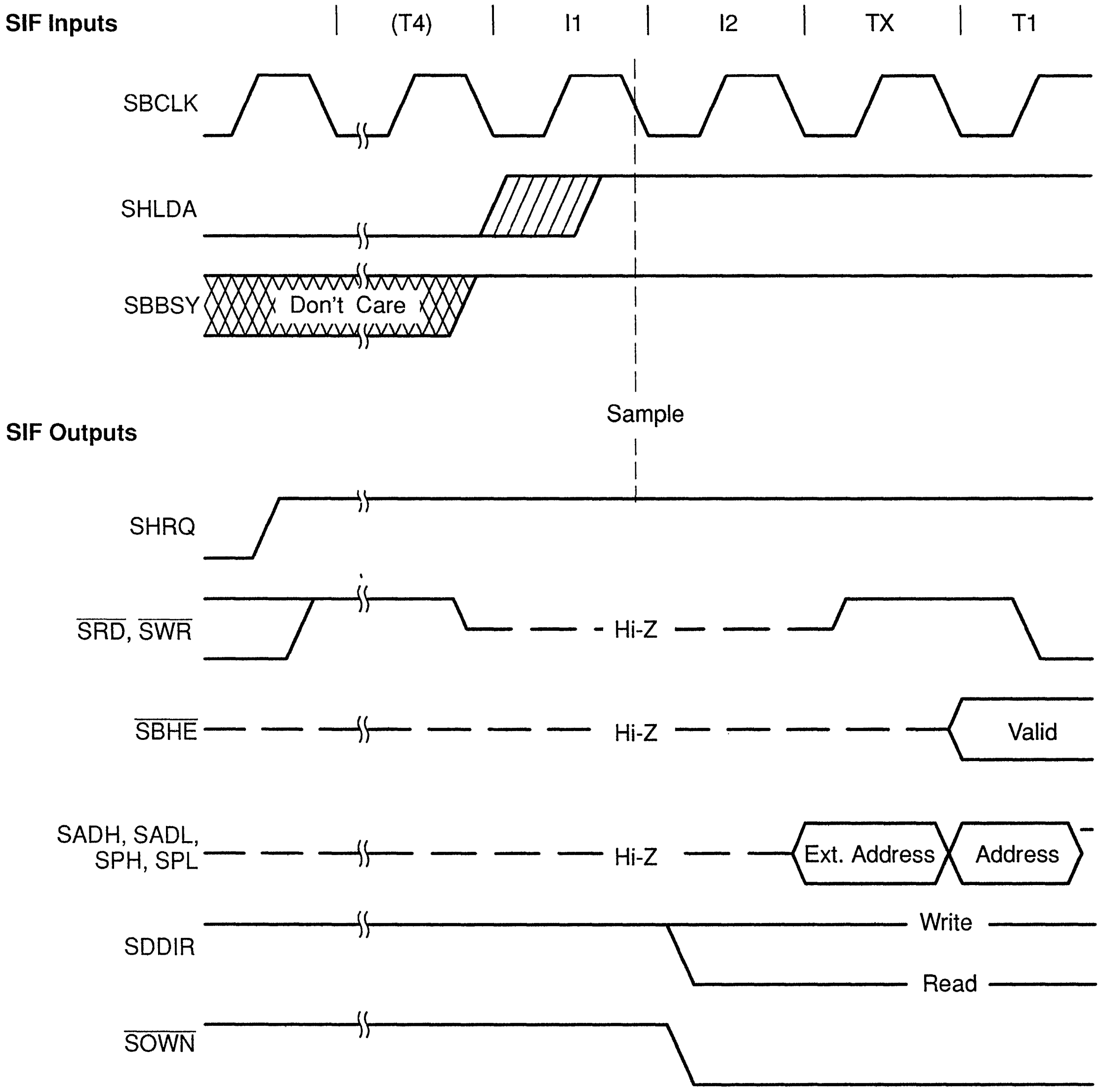


Figure 3–20 demonstrates the timing when the SIF returns control of the system bus to the user processor. The sequence of operations is described below:

1) On the falling edge of SBCLK in state T4 of the last DMA transfer, the SIF does the following:

    a) It floats the bus controls $\overline{\text{SRD}}$ and $\overline{\text{SWR}}$;

    b) It drives $\overline{\text{SBHE}}$ high, and when it has detected that the pin has reached high level, three-states the signal; and

    c) It three-states the SAD bus.

2) On the next rising edge of SBCLK (I1 high), the SIF deasserts SHRQ by driving it low.

3) On the subsequent falling edge (I2 high), the SIF deasserts $\overline{\text{SOWN}}$ by driving it high.

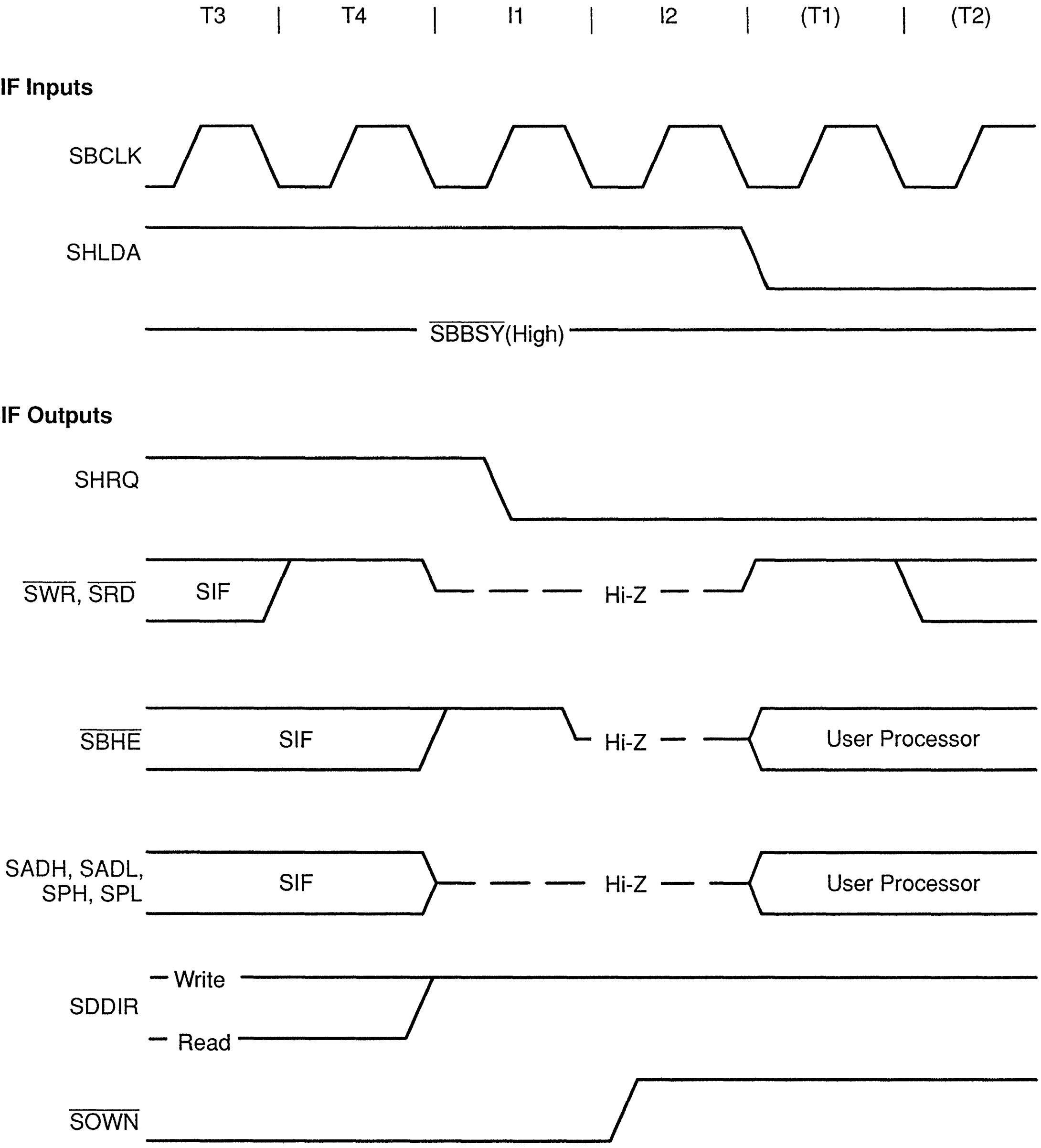
4) It drives SDDIR back high (if previously driven low for a read cycle).

5) Sometime after detecting that SHRQ is negated, the system deasserts SHLDA.

Note:

Care must be taken that SHLDA be driven inactive before the SIF arbitrates for the next DMA request.

Figure 3–20. SIF Returns System Bus—808x Mode



3.4.5 68xxx-Mode DMA Operations

This subsection describes the system bus cycle and bus arbitration timing when the SIF is configured in 68xxx mode.

3.4.5.1 68xxx DMA Transfer

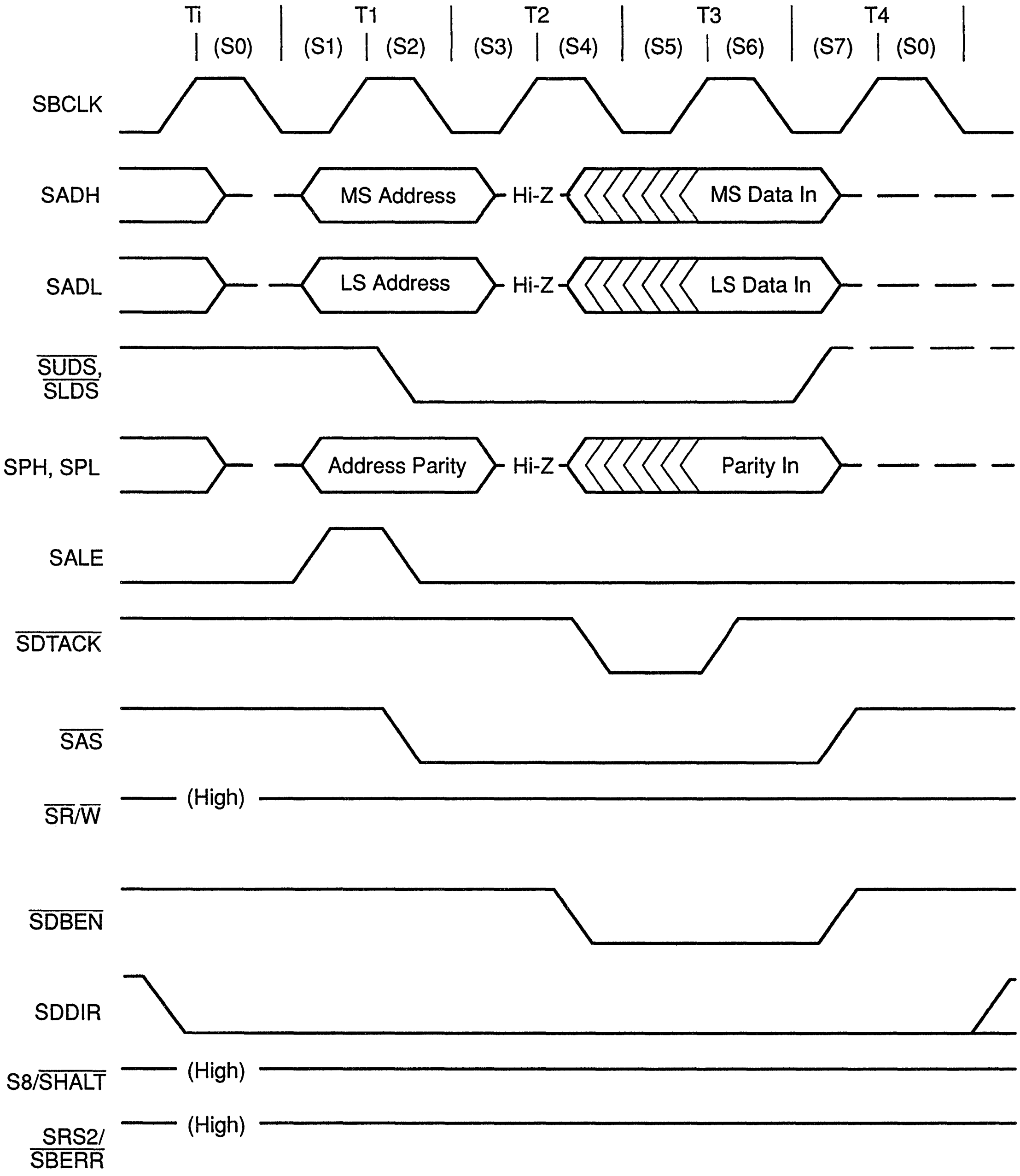
The general system timing for DAM transfers between the SIF pins and the user memory when the SIF is in 68xxx mode is shown in Figure 3–21 and Figure 3–22.

68xxx documentation describes its bus transfers as a sequence of S states, each consisting of one half-phase of the SBCLK cycle. For consistency of this specification, it describes a 68xxx mode DMA cycle as a sequence of T states, each consisting of one SBCLK cycle. This convention is the same one used for 808x-mode DMA cycles.

At the start of DMA cycles shown in Figure 3–21 and Figure 3–22, output SALE (system interface address latch enable) strobes the contents of SADH0–SADH7 and SADL0–SADL7 into an external latch (such as 373). Unlike the 68xxx microprocessor, the SIF multiplexes address and data over the same physical I/O pins. However, the timing of SALE is such that the address/data bus can be demultiplexed with external latches to present to the system bus a nonmultiplexed interface whose signals and timing are similar to those of the 68xxx. Assuming that a transparent latch is used to capture the address from the SAD pins of the SIF, the address at the outputs of the external latch becomes valid prior to the falling edge of SAS (system interface address strobe).

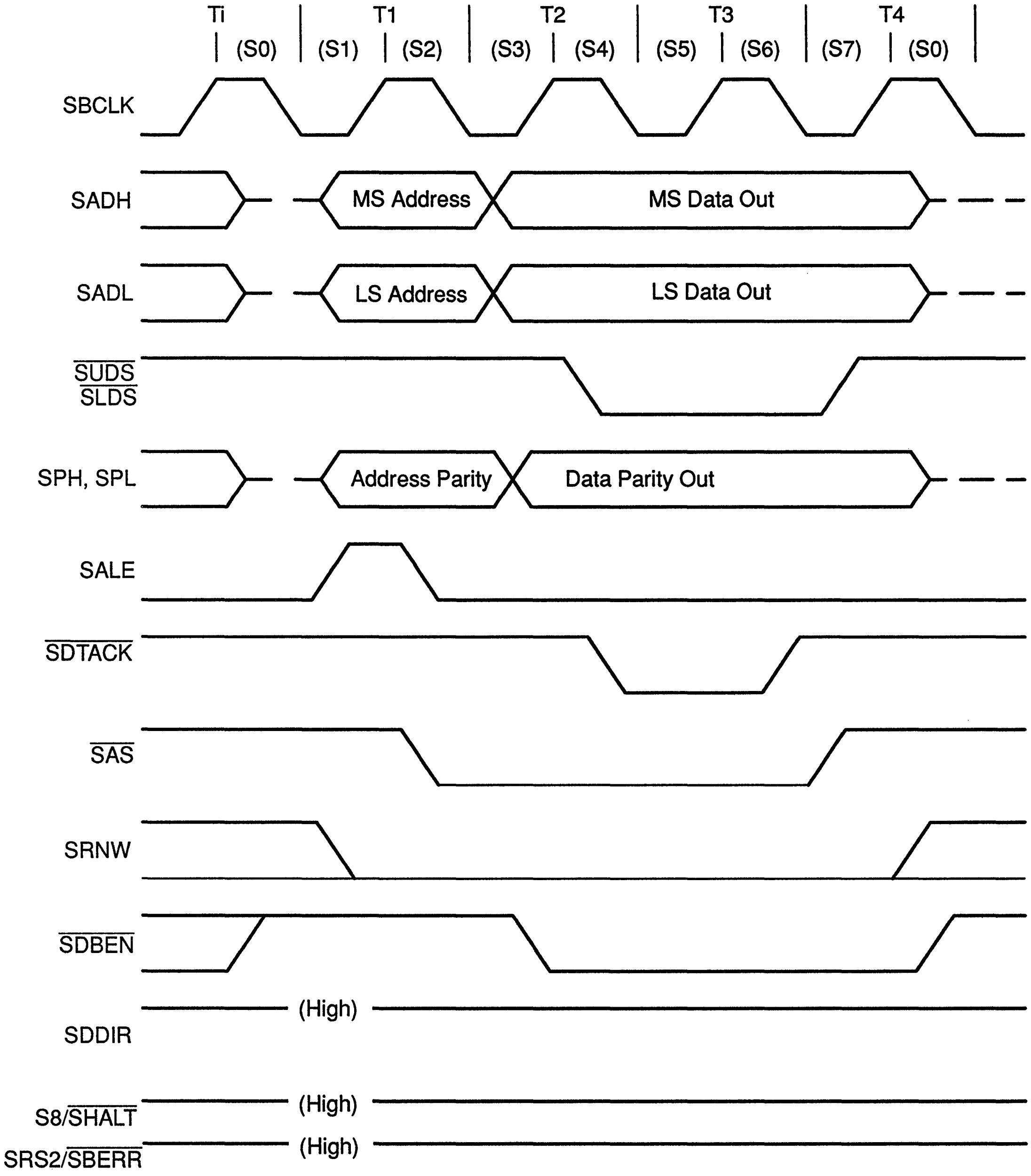
A DMA read cycle is shown in Figure 3–21. Data strobes SUDS and SLDS have the same timing as SAS. The SBHE/SRNW signal (which in 808x mode functions as the system interface upper data strobe indicator) remains high throughout the cycle to indicate that a read operation is taking place. SDDIR, used to indicate the direction of the data transfer to the 245-type external data bus transceivers, remains at its default low level throughout the cycle. SDBEN, the signal used to turn the data bus transceivers on (if SDBEN is low) and off (if SDBEN is high), goes low in the middle of the cycle to gate the contents of the system data bus into the SIF.

Figure 3–21. 68xxx-Mode DMA Read Timing



A DMA write cycle is shown in Figure 3–22. $\overline{\text{SBHE}}$/SRNW goes low to indicate that a write operation is taking place. The falling edges of data strobes $\overline{\text{SUDS}}$ and $\overline{\text{SLDS}}$ occur one full clock after the falling edge of $\overline{\text{SAS}}$. SDDIR remains high during the cycle, and $\overline{\text{SDBEN}}$ goes low to enable the external databus transceivers to drive the data output from the SADH0–SADH7 and SADL0–SADL7 pins of the SIF onto the system data bus. In certain cases, only a single byte is transferred. In these cases, the unused half of SAD is floated.

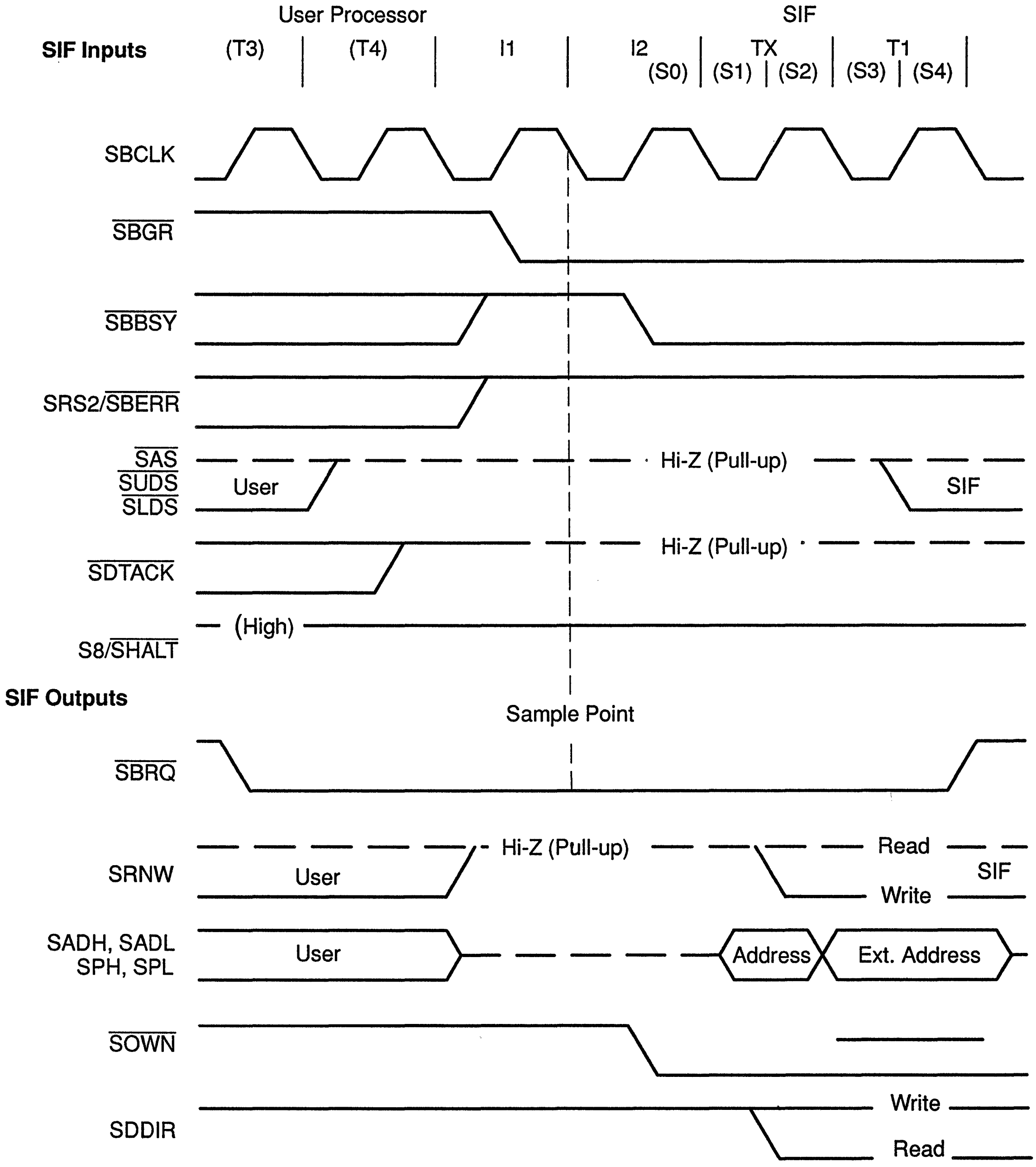
Figure 3–22. 68xxx-Mode DMA Write Timing



3.4.5.2 68xxx Bus Arbitration

The bus arbitration timing for 68xxx mode is shown in Figure 3–23 and Figure 3–24. The SBRQ, $\overline{\text{SBGR}}$, and $\overline{\text{SBBSY}}$ signals of the SIF correspond to the $\overline{\text{BR}}$, $\overline{\text{BG}}$, and $\overline{\text{BGACK}}$ signals of the 68xxx.

Figure 3–23. SIF Acquires System Bus—68xxx Mode



In Figure 3–23, the SIF arbitrates for and obtains control of the system bus in preparation for one or more DMA cycles. Before the SIF requests the bus, it verifies that SBGR is high at the falling edge of SBCLK. On the second rising edge of SBCLK, following the SIF's sample of SBGR high, SBRQ (system bus request) is driven low by the SIF. SBRQ (system bus request) is driven low by the SIF to request the bus. SBGR (system bus grant) is driven low by the processor to indicate its readiness to yield bus control to the SIF. The SIF samples several signals on the falling edge of SBCLK to determine when it may control the bus: its own SBRQ must be low, SBGR must be low, and the following signals must be high: SBBSY, SBERR, SAS, SUDS, SLDS, SDTACK, and SBERR. The asynchronous inputs SBBSY, SBGR, SBERR, SDTACK, and SHALT are sampled on the falling edge of SBCLK. The SIF starts phase TX of its bus cycle on the first falling edge from the sample point at which all inputs are at the appropriate level.

The SIF input pin SBBSY is connected to the bus grant acknowledge pin of the system bus. As long as a system bus master holds BGACK low, no other device is allowed to take control of the bus. The SIF must sample BGACK (SBBSY) negated before it takes control of the bus. When it does so (by driving SOWN low), external glue logic drives BGACK low. The SIF, having pulled BGACK low, can release SBRQ (such as return it to its inactive-high level) and still retain exclusive mastership of the bus. Other DMA devices can now arbitrate among themselves by using the SBRQ and SBGR lines to determine which of them will be the next to gain control of the bus. But none of these devices can actually take control of the bus until the SIF releases BGACK (by negating its SOWN signal).

The SIF is shown returning control of the bus to the user processor in Figure 3–24. The timing shown represents the simple case in which no other DMA device is presently requesting the bus; hence, the processor again assumes control of the bus.

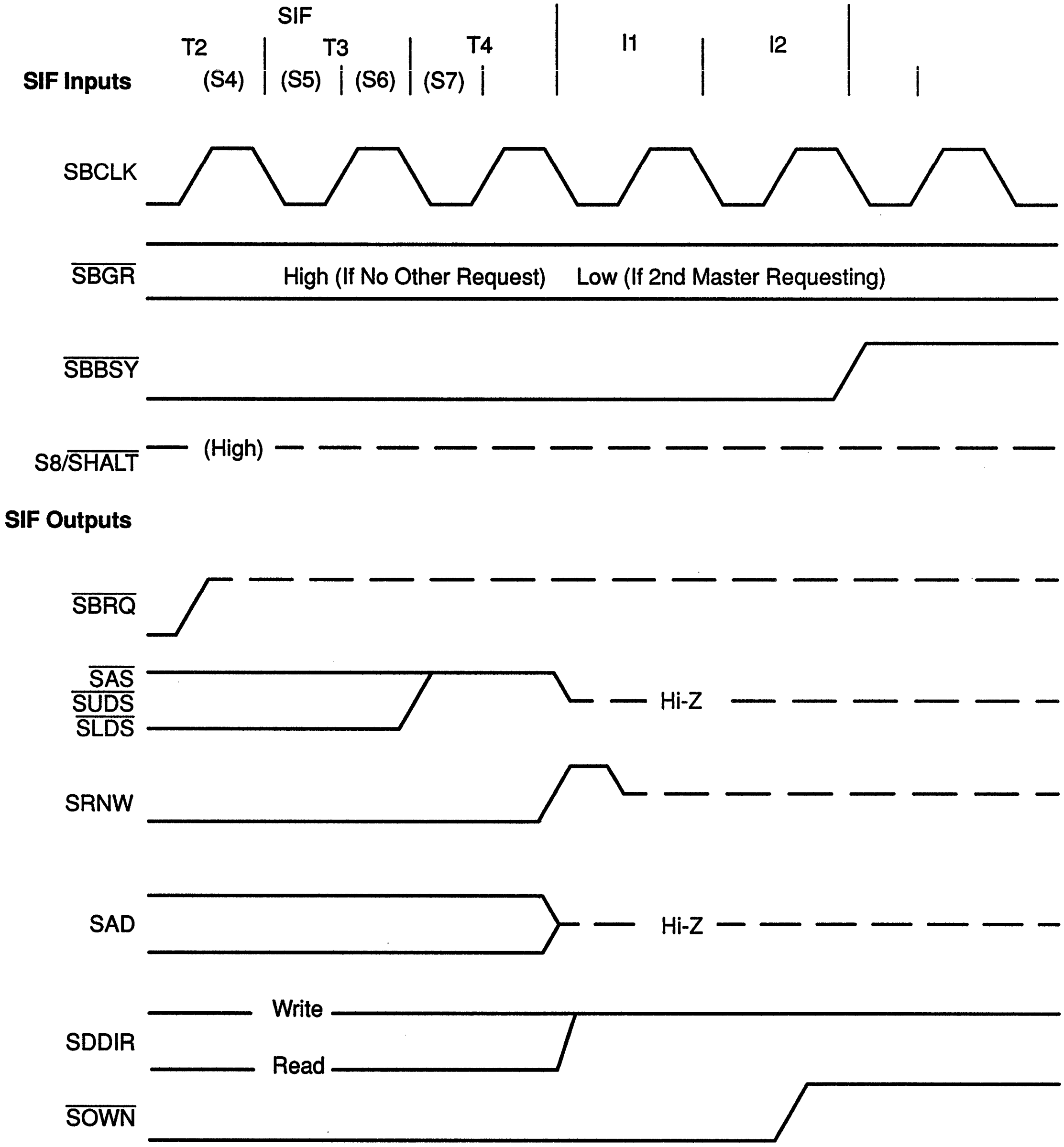
The bus arbitration signals of the SIF have been designed to permit minimum-chip configurations to be built using as little additional glue logic as possible. The SBBSY input of the SIF connects to the bus grant acknowledge signal of the system bus. If left unconnected, SBBSY floats high due to the pullup device connected internal to the chip.

In order to monitor asynchronous inputs on the SBGR and SBBSY pins, the SIF contains internal double-buffered synchronizer latches to insure reliable operation in all system configurations. In larger systems the signals to these inputs will in fact be asynchronous; also, some external glue logic is required to provide buffering and support daisy-chaining of the bus grant signals. In a small system in which the user processor and SIF are connected directly to the same bus and share the same clock, a glueless bus arbitration configuration can be constructed by connecting the BR, BG, and BGACK pins of the processor directly to the corresponding pins of the SIF. In support of this synchronous, minimum-chip configuration, the SIF follows timing conventions on its bus arbitration pins similar to the 68xxx microprocessor:

- ❑ Inputs are always sampled on falling clock edges, and
- ❑ Outputs always change on rising clock edges.

This can be verified by comparing Figure 3–23 with published 68xxx microprocessor material.

Figure 3–24. SIF Returns System Bus—68xxx Mode



3.4.5.3 68xxx DMA Bus Error: Rerun and Halt Conditions

The SIF supports the bus error, rerun, and halt conditions required by devices interfacing with 68xxx-based systems. These are conditions that can occur during any bus cycle.

- Bus errors indicate that something went wrong during the cycle, causing an error condition.
- Rerun indicates that something other than an error necessitates rerunning the cycle.
- Halt indicates that it became necessary to suspend execution for some reason, such as to single-step operation during system debug.

These three conditions are controlled by the $\overline{\text{SBERR}}$ and $\overline{\text{SHALT}}$ pins working in conjunction with $\overline{\text{SDTACK}}$. These three signals are sampled by the SIF at the start of every T state (falling edge of SBCLK), and the values sampled are then used in conjunction with the values sampled at the start of the previous T state. This process is repeated for every T state while the SIF retains ownership of the system bus.

When one of the three described conditions occurs during a bus cycle, the SIF recognizes this and performs an appropriate response during the following bus cycle. In the case of detection of a bus error, the SIF checks BERETRY for zero and, if nonzero, decrements it and reattempts the transfer; if BERETRY is zero, the transfer is aborted as described in an earlier subsection. If a rerun condition is detected, the cycle is repeated without checking or decrementing BERETRY. If a halt condition is detected, the SIF completes the cycle in which the condition is detected and then performs idle T states until the halt condition is removed.

Two special conditions exist relating to bus acquisition and relinquish. The first occurs if the SIF acquires the system bus with $\overline{\text{SHALT}}$ active low. In this case, it performs idle T states until the halt condition is removed and then begins DMA operation with its first TX state. The second condition occurs when a bus error or rerun condition is detected, but the SIF is forced to relinquish bus ownership on completion of the cycle. In this case, the SIF remembers the condition and acts upon it after regaining bus ownership, unless DMA is halted by the CP before the bus is regained. In this case, the error or rerun condition is forgotten. The repetition of the cycle is not necessarily the first cycle after regaining the bus, since a TX cycle is always the first memory cycle executed after bus acquisition, but the cycle to be repeated may not be a TX cycle.

Table 3–13 indicates the values of $\overline{\text{SDTACK}}$, $\overline{\text{SBERR}}$, and $\overline{\text{SHALT}}$ for the detection of the four cycle termination conditions. (normal, normal then halt, bus error, and rerun):

Table 3–13. $\overline{\text{SDTACK}}$, $\overline{\text{SBERR}}$, and $\overline{\text{SHALT}}$ Assertion Results

Signal	Values Present at SBCLK Rising Edge		Result
	R	T + 1	
$\overline{\text{SDTACK}}$	0	WA	Normal cycle termination. Operation continues in the next cycle.
$\overline{\text{SBERR}}$	1	X	
$\overline{\text{SHALT}}$	1	X	
$\overline{\text{SDTACK}}$	0	0	Normal cycle termination, but next cycle does not begin until $\overline{\text{SHALT}}$ is taken high.
$\overline{\text{SBERR}}$	1	X	
$\overline{\text{SHALT}}$	0	0	
$\overline{\text{SDTACK}}$	1	0	Normal cycle termination, but next cycle does not begin until $\overline{\text{SHALT}}$ is taken high.
$\overline{\text{SBERR}}$	1	1	
$\overline{\text{SHALT}}$	0	0	
$\overline{\text{SDTACK}}$	X	X	Cycle terminates with a bus error detected. Cycle is retried if BERETRY nonzero.
$\overline{\text{SBERR}}$	0	0	
$\overline{\text{SHALT}}$	1	1	
$\overline{\text{SDTACK}}$	1	X	Cycle terminates with a rerun condition detected. Cycle is rerun when $\overline{\text{SHALT}}$ is taken high.
$\overline{\text{SBERR}}$	0	0	
$\overline{\text{SHALT}}$	1	0	
$\overline{\text{SDTACK}}$	X	X	Cycle terminates with a rerun condition detected. Cycle is rerun when $\overline{\text{SHALT}}$ is taken high.
$\overline{\text{SBERR}}$	0	0	
$\overline{\text{SHALT}}$	0	0	
$\overline{\text{SDTACK}}$	1	X	Cycle terminates with a rerun condition detected. Cycle is rerun when $\overline{\text{SHALT}}$ is taken high.
$\overline{\text{SBERR}}$	1	0	
$\overline{\text{SHALT}}$	0	0	

Notes: 1) T and T + 1 are the two consecutive T states in which the signals are sampled.
2) WA indicates was active in previous T state.

Table 3–14 indicates the values of $\overline{\text{SDTACK}}$, $\overline{\text{SBERR}}$, and $\overline{\text{SHALT}}$ for the removal of the termination conditions given in Table 3–13 above.

Table 3–14. $\overline{\text{SDTACK}}$, $\overline{\text{SBERR}}$, and $\overline{\text{SHALT}}$ Negation Results

Condition of Termination (Table 3–9)	Signal	Values Present at SBCLK Rising Edge			Result (Next Cycle)
		T		T + 2	
Normal	SBERR	*			May delay start of next cycle.
	SHALT	*	or	*	
Normal	$\overline{\text{SBERR}}$			*	If next cycle has started, it terminates with a bus error detect.
	$\overline{\text{SHALT}}$	*			
Bus-error	SBERR	*	or	*	Reruns cycle if BERETRY is nonzero.
	SHALT	*	or	*	
Rerun	SBERR	*	or	*	Illegal sequence; treated as a bus error. Reruns cycle if BERETRY is nonzero.
	SHALT	*			
Rerun	SBERR	*			Reruns the cycle.
	SHALT			*	

Note: * signifies that the signal is negated high in this state.

3.4.6 Burst/Cycle-Steal DMA Modes

Two modes of bus arbitration may be selected by the CP software: burst mode or cycle-steal mode. CP software selects DMA burst mode by setting the DMABURST bit in the INIT options to a one. If the DMABURST bit is zero, the SIF is in cycle-steal mode for that transfer. See Section 4.5.

3.4.6.1 Burst Mode

During a burst-mode block DMA transfer, the SIF arbitrates for and obtains control of the user bus and then retains the bus until one of the following conditions occurs:

1) The block transfer is completed;
2) The bus error retry or parity error retry counter set in INIT has reached zero following the detection of bus or parity error(s), and the system bus parity is enabled;
3) The bus-release line from the user system ($\overline{\text{SBRLS}}$) is asserted; or
4) The DMA burst count from INIT has decremented to zero.

During the time that the continuous transfer is in progress, successive word or byte transfers occur back to back, such as with no idle SBCLK cycles between bus transfers. An exception is the occasional SBCLK period (phase TX) used to update the extended-address latch, which occurs when the increment of the 32-bit DMA address counter causes a carry to ripple into its 16 MSBs.

If the SIF is unable to maintain the back-to-back DMA transfer rate, it will release the bus temporarily to allow other devices to use the bus.

While the SIF is performing a burst-mode DMA block transfer, a user device can cause the SIF to momentarily release the system bus, with the DMA transfers resuming normally after the user device relinquishes the bus. An external device requests that the SIF release the bus by asserting an active-low signal to the SIF $\overline{\text{SBRLS}}$ (bus release) input. The external device holds SBRLS active low until the SIF indicates the system bus has been released by deasserting $\overline{\text{SOWN}}$ high (to set $\overline{\text{SBBSY}}$ high).

The number of consecutive cycles that can be transferred in burst mode is contained in burst count from initialization. When an SDMA transfer is initiated, it continues until one of the halt conditions has been detected or the burst count decrementer has reached zero. In the later case, the SDMA transfer does not cease, but rather pauses by relinquishing control of the system bus to allow another device access to it. The SIF again requests ownership of the system bus and when it has regained it, begins to transfer another burst of data of the length set by initialization. A value of >00 in the burst count of INIT is regarded as infinite; the SIF does not relinquish the system bus until a SDMA halt condition has been detected.

3.4.6.2 Cycle-Steal Mode

In cycle-steal mode, the SIF contends for the system bus for each individual system DMA data transfer (that is, one 16- or 8-bit transfer), releasing the bus when the transfer is complete. This mode decreases the latency of higher-priority devices in obtaining bus cycles, but results in a longer time required to transfer a block to or from the adapter. A TX cycle is performed on every bus transfer in cycle steal mode.

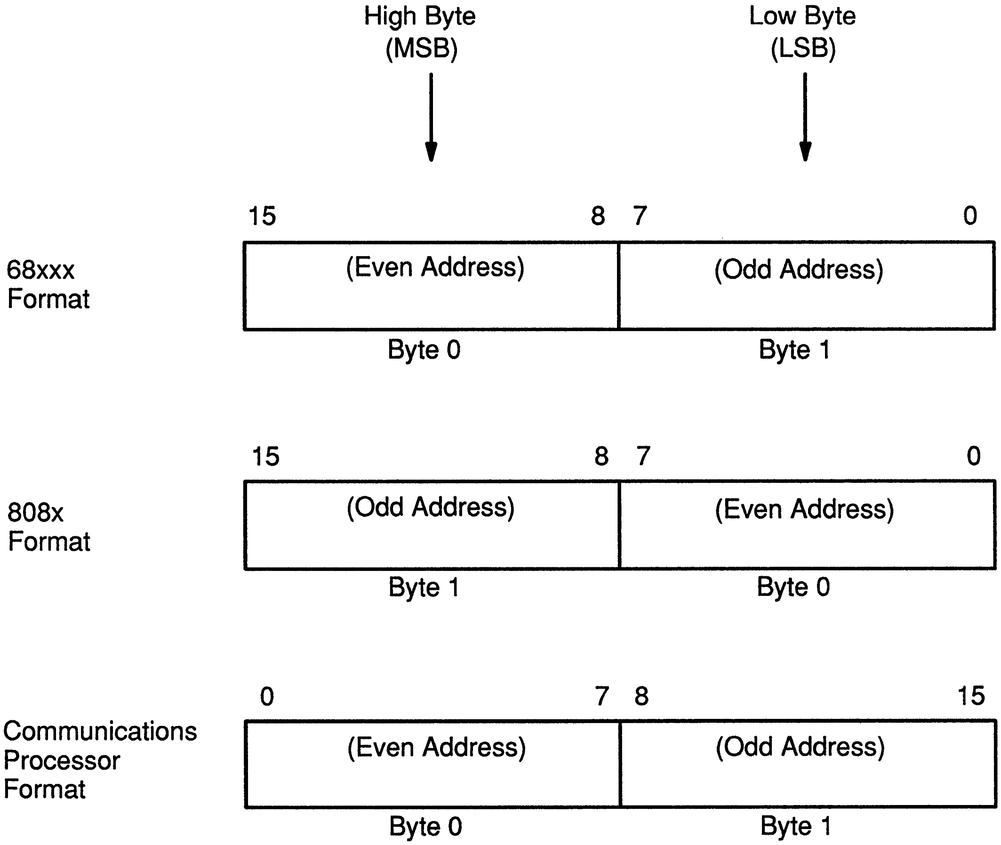
3.4.7 System DMA Data Organization

The SIF DMA controller operates on the principle that sequentially transmitted bytes (to and from the ring) are placed in system memory in increasing byte address order according to the byte numbering conventions for the user processor. When performing DMA with 16-bit data, the SIF aligns bytes so that the first byte received from the ring is transferred to or from the byte of processor memory that corresponds to the lower processor byte address.

This operation is somewhat complex because of the opposite conventions used by 68xxx and Intel-type processors for the addressing of bytes within a 16-bit word. We repeat, however: in all interface modes, the adapter transfers sequentially transmitted bytes to sequentially increasing byte addresses in the system.

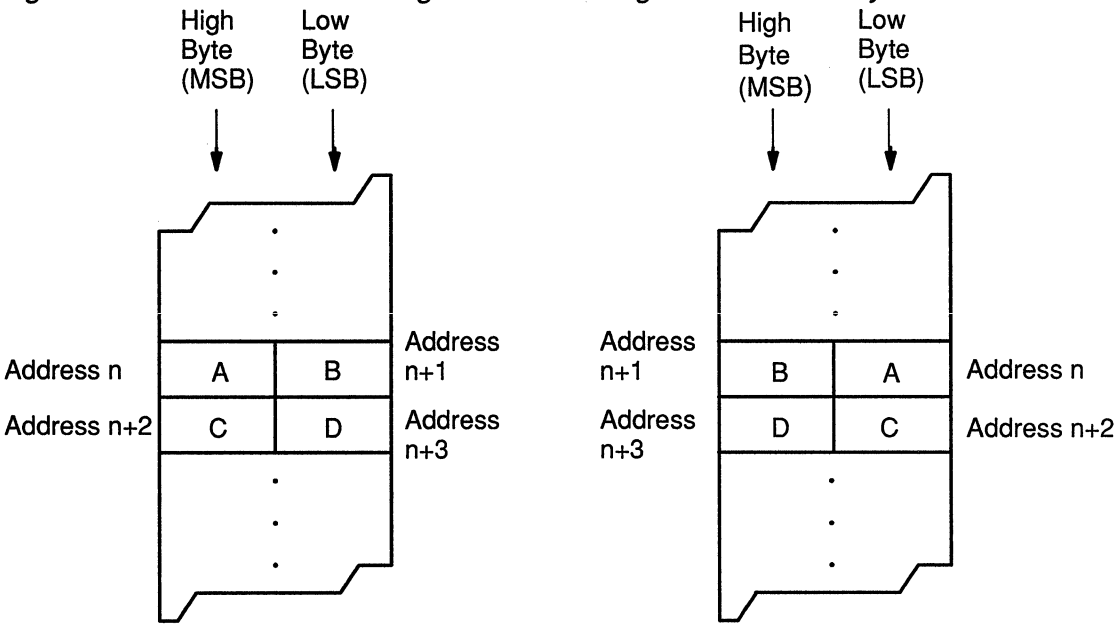
This is desirable because most messages sent over the ring consist of a sequence of single-byte ASCII character codes that must be kept in proper sequence in order to be read (displayed, printed, stored etc.) correctly. This requires the 808x processors to actually swap bytes before they can treat two adjacent bytes as a word entity. Figure 3–25 contrasts the bit-and-byte-numbering conventions for 68xxx and 808x microprocessors with those of the communications processor, as they appear, based on even boundary word architecture.

Figure 3–25. Bit- and Byte-Numbering Conventions



The two bytes in each word are referred to uniformly; the byte on the left is designated the high byte, and the byte on the right is designated the low byte. The impact of these conventions on the ordering of data bytes in memory is illustrated in Figure 3–26. The storage of the string ABCD is shown for both the 68xxx and 808x.

Figure 3–26. Formats for Storing Character String ABCD in Memory



68xxx Format

808x Format

In 808x 8-bit mode, each DMA word transferred from the SIF to the user system is transmitted as a series of two bytes over the SADL0–SADL7 pins. An internal byte swapper is used to multiplex the internal high byte of the SDMA FIFO to/from the SADL bus pins. DMA bytes are transferred in increasing byte address order on the system data bus.

The SIF transfers the data in the minimum number of cycles, adding single-byte transfers at the beginning and/or end if necessary, so that all word transfers will be even-address aligned. This is accomplished on the adapter bus by a read-modify-write cycle, and on the host bus by a single-byte transfer. If the host is strapped for 8-bit mode, all transfers to or from the host are 8-bit. This allows the SIF to transfer any number of bytes up to 64K from any byte address to any byte address, from or to the host.

There are 32 possible cases for system DMA transfer, because of five independent binary cases:

- ❑ The host processor may use either 808x or 68xxx data organization (selected with the SI/$\overline{\text{M}}$ strap pin);
- ❑ The DMA starting address in system memory may be even or odd;
- ❑ The DMA starting address in adapter memory may be even or odd;
- ❑ The number of bytes to transfer may be even or odd; and
- ❑ The transfer may be a read (system to adapter) or write (adapter to system) operation.

If the host is in 808x 8-bit mode, then all data transfers on the system bus are performed on the SADL pins (with the MSB of the address asserted on SADH), and one system bus cycle is used for each byte to be transferred. In 16-bit mode, however, either 8- or 16-bit data may be transferred in a cycle, and 8-bit data may appear on either the SADL or SADH pins. DMA operation with 16-bit systems is detailed for each case in Table 3–15. In each example, the byte string ABCD or ABC is to be transmitted/received across the system interface. For example, in 808x mode, at an even starting system address, for three bytes (odd) to write to the system, the SIF performs two transfers:

- ❑ The SIF first reads a word from adapter memory with A in the MSB and B in the LSB. This is written onto the system bus as a word with A in the LSB and B in the MSB. Note that in both adapter and system memory, A is in the lower (even) address and B is in the upper (odd) address. As always, sequentially transmitted bytes are stored in sequential byte addresses.
- ❑ The SIF then reads a word from adapter memory with C in the MSB and anything (–) in the LSB. The SIF performs a single-byte cycle on the system bus, writing the byte C to the lower (even) byte of the system bus. In 808x mode, the lower addressed byte is connected to SADL, so C is written on these lines. The SADH lines are driven during this cycle.

In the 808x 16-bit mode, the bytes are actually swapped during DMA transfers; that is, the n^th^ byte of the DMA FIFO is connected to the SADL (LSB) system bus and the n1^th^ byte of the DMA FIFO is connected to the SADH (MSB) system bus. In the 808x 8-bit mode, both bytes go over the SADL system bus, but the time sequence (n^th^ byte first, n1 ^th^ byte second) is opposite to the normal sequence of a 808x 16-bit instruction run on an 8-bit bus. This requires a byte-swap instruction to be performed by the 808x processor before two adjacent bytes can be used as a word entity. In the 68xxx 16-bit mode, the high byte is connected to SADH, and the low byte is connected to SADL. No byte-swap instruction is necessary to treat two adjacent bytes as a word entity (see section 4.1 for more details).

It should also be noted that no byte swapping occurs in either mode for DIO operations. The DIO byte structure was discussed in section 3.4.1.

Table 3–15. System DMA Cycle Description

Host	Starting System Address	Number of Bytes to Move	Read or Write to System	16-Bit System Bus Transfer		Adapter Internal Bus Transfers	
				SADH 0–7	SADL 0–7	HIGH BYTE	LOW BYTE
808x	even	even	read	B D	A C	A C	B D
			write	B D	A C	A C	B D
		odd	read	B –	A C*	A C	B #
			write	B –	A C*	A C	B –
	odd	even	read	A* C –	– B D*	A C	B D
			write	A* C –	– B D*	A C	B D
		odd	read	A* C	– B	A C	B #
			write	A* C	– B	A C	B –
68xxx	even	even	read	A C	B D	A C	B D
			write	A C	B D	A C	B D
		odd	read	A C	B –	A C	B #
			write	A C*	B –	A C	B –
	odd	even	read	– B D*	A* C –	 A C	 B D
			write	– B D*	A* C –	A C	B D
		odd	read	– B	A* C	A C	B #
			write	– B	A* C	A C	B –

The bytes are transmitted across the ring in alphabetical order. Whenever an odd number of bytes is read from the system to the adapter, the extra byte in the adapter is maintained by a read-modify-write cycle. Adapter transfers are always 16 bits wide. A * symbol indicates that this byte is transferred via a single-byte operation on the system bus. In other words, the strobe for that byte only is asserted. A – symbol indicates a don't care value. A # symbol indicates a read value form the RMW cycle. Within the system and adapter columns, each row indicates a single bus cycle. On 8-bit write cycles in 16-bit mode or 808x in 8-bit mode, the unused bus (SADH or SADL) is driven but don't care.

3.4.8 System DMA Address Latching

Figure 3–27 shows how the latches used to capture address information from the SADL0–SADL7 and SADH0–SADH7 pins are connected to the SIF.

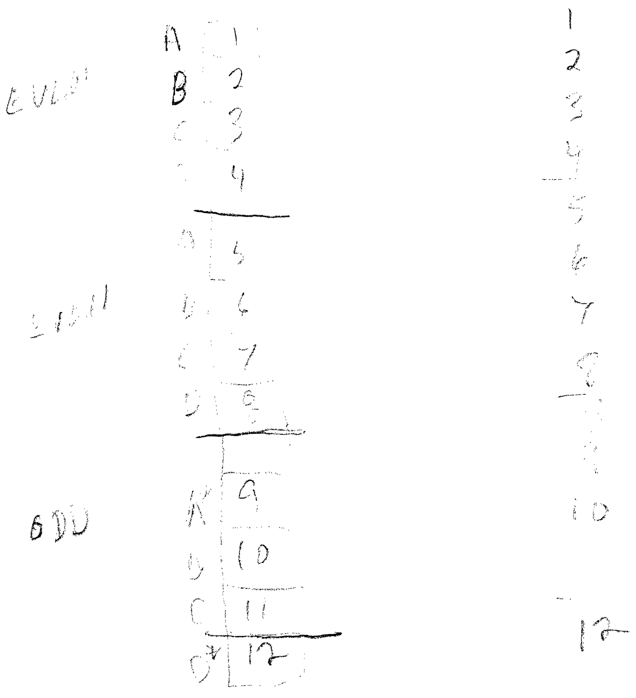
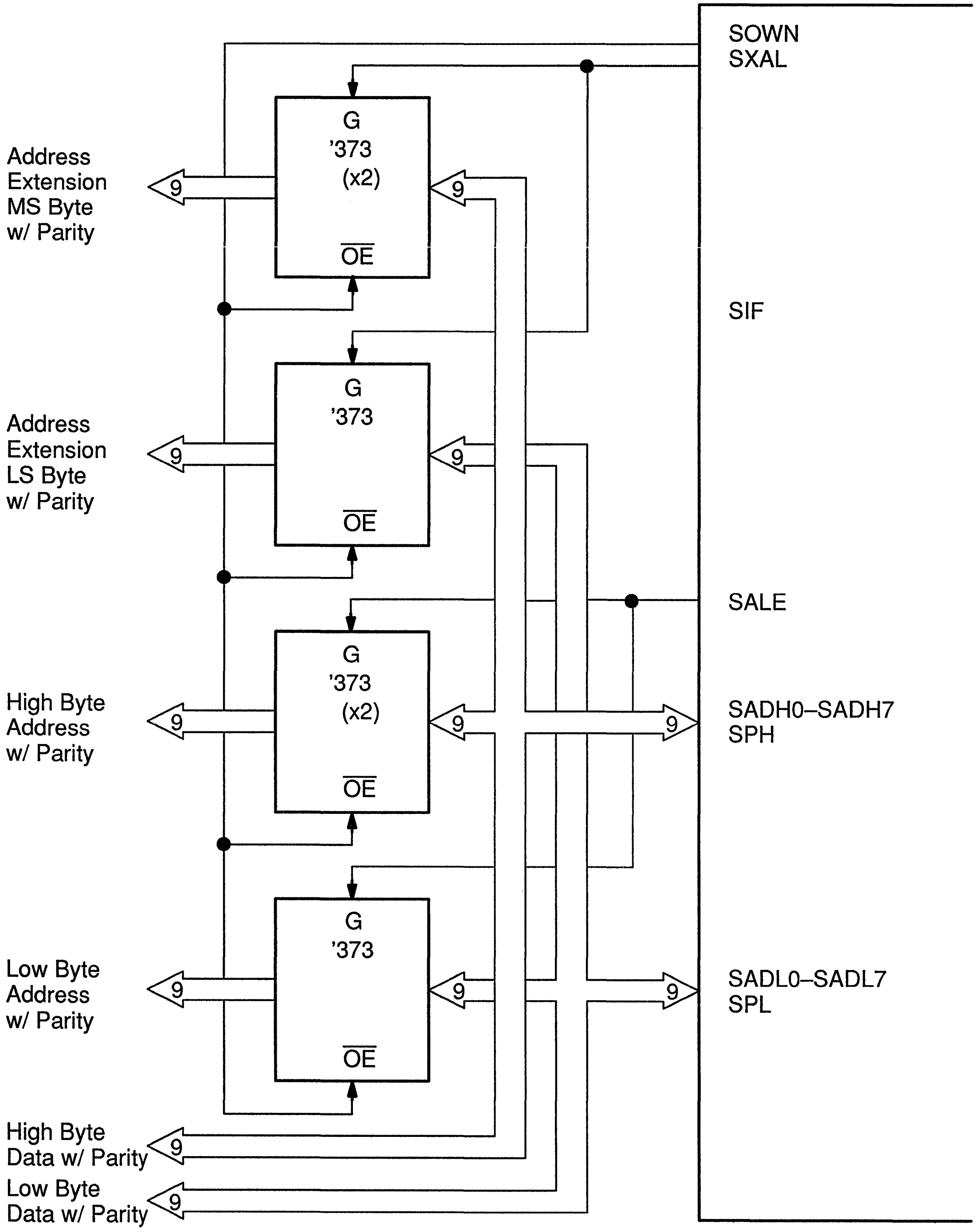
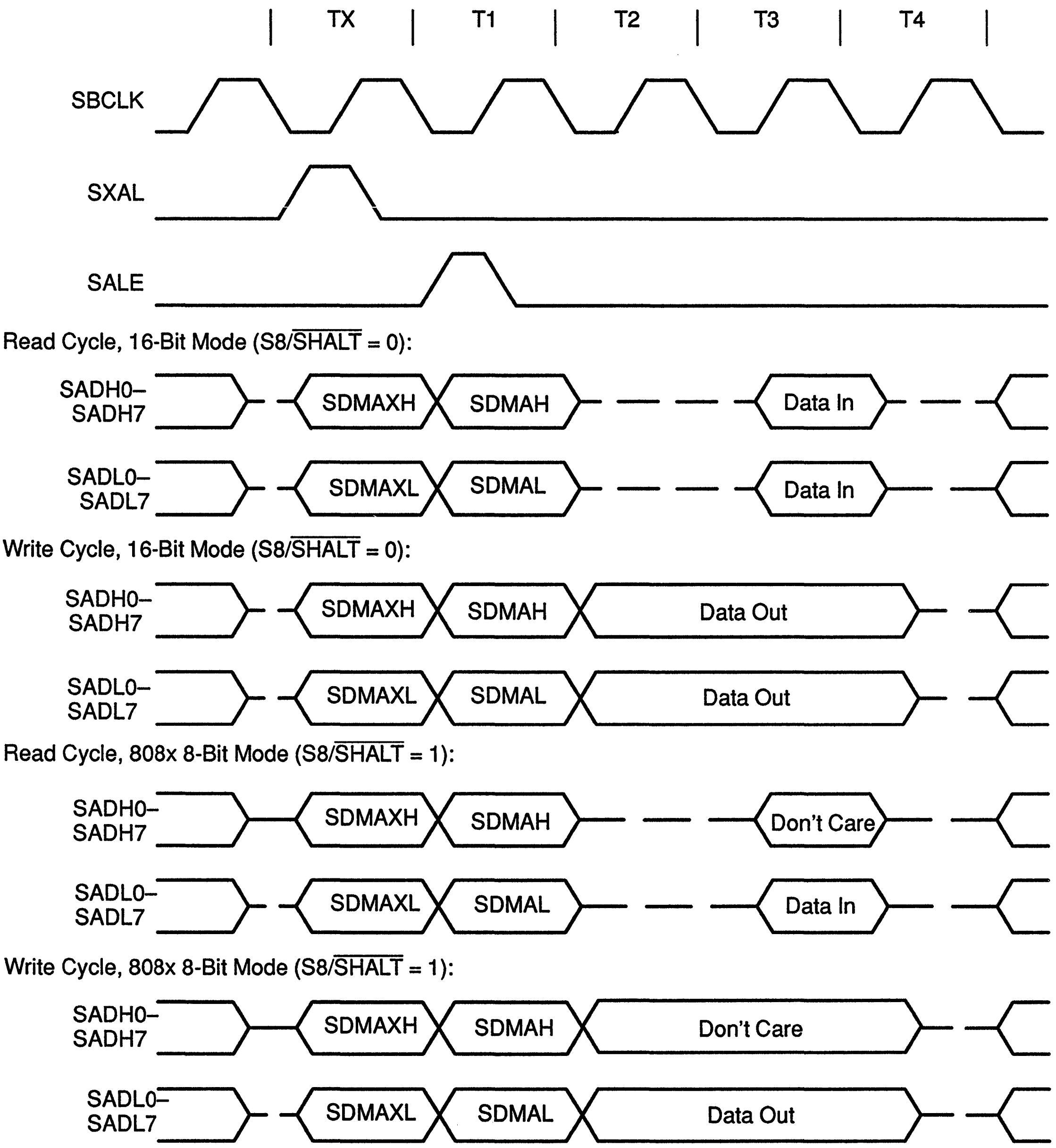


Figure 3–27. External TTL Latches Demultiplex Address/Data Bus



These latches are typically TTL devices, such as the 74ALS373 or 74ALS573. The SIF's $\overline{\text{SOWN}}$ signal is used to enable or disable the three-state outputs of the latches. Two address-latch-enable signals, SALE (system address latch enable) and SXAL (extended address latch enable), are provided by the SIF to demultiplex the low-order 16 and high-order 16 address bits, respectively. When the extended address latch needs to be changed, a special TX bus clock phase is used to update the latch, with the new data being asserted on the SADL lines. The timing of the SALE and SXAL signals in both 8- and 16-bit modes is shown in Figure 3–28.

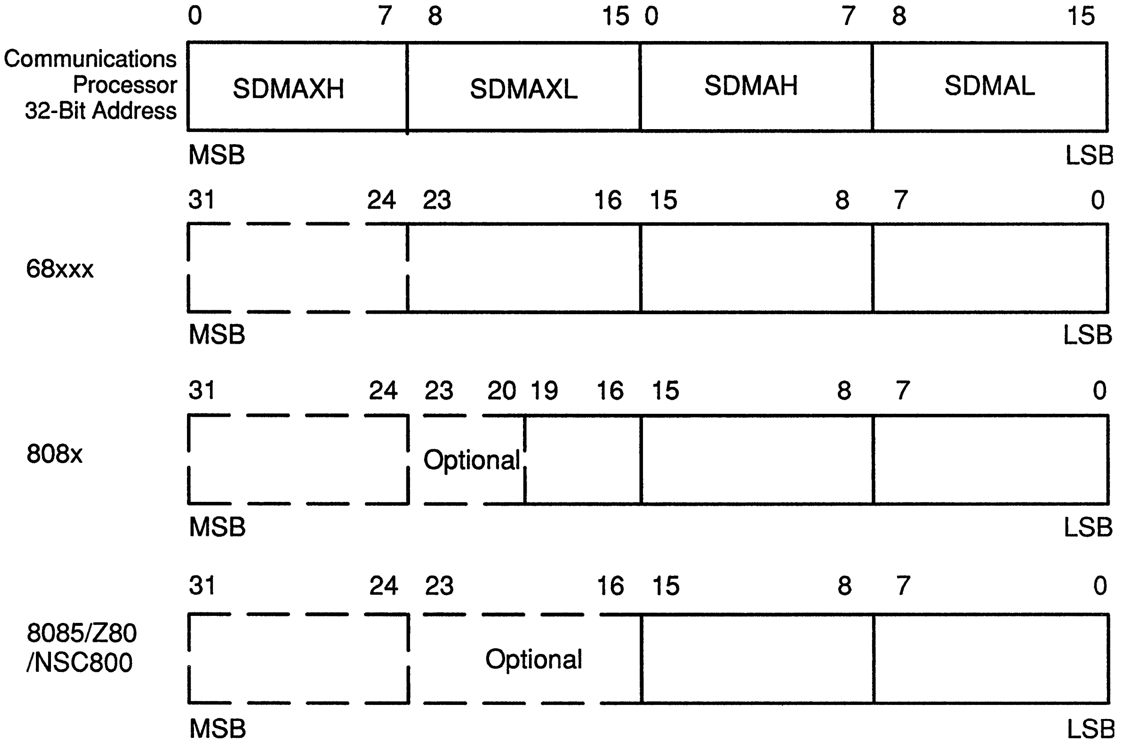
Figure 3–28. Address Extension Updates for 8- and 16-Bit Modes



The 32-bit system bus DMA address is formed by concatenating the four 8-bit values SDMAXH, SDMAXL, SDMAH, and SDMAL from the system address registers, as indicated in Figure 3–29.

The bit-by-bit correspondence between the SDMA address stored adapter memory, and the representation of the same address according to the conventions of particular user processors is shown in Figure 3–29.

Figure 3–29. Comparison of Address Bit Formats



In general, any of the microprocessors mentioned in Figure 3–29, with the exception of the 68xxx, specifies addresses with the least significant byte or word preceding the most significant byte or word, and requires conversion to the format used by the communications processor. For this reason, host system software must take care during setup of DIO commands to place its system addresses into adapter memory in the correct byte order.

It is recommended that adapter software document its data structures and commands via 16-bit-wide data structures, clearly indicating the most significant and least significant bytes of every word. Host system software, then, loads the most significant and least significant bytes of SIFDAT exactly as the word should appear in adapter memory. Because the MSbyte and LSbyte of SIFDAT are attached to the MSbyte and LSbyte, respectively, of the host system, it is not necessary for the host system to swap the bytes of addresses passed across the DIO interface. Addresses are given to the adapter exactly as they are stored in memory with the most significant byte of the address in the MSbyte of the word.

3.5 Hardware Notes

The following pins should be pulled high during three-state:

System Interface	Resistor Value
$\overline{\text{SRDY}}$-$\overline{\text{SDTACK}}$	4.7 kΩ
$\overline{\text{SRD}}$-$\overline{\text{SUDS}}$	4.7 kΩ
$\overline{\text{SWR}}$-$\overline{\text{SLDS}}$	4.7 kΩ
SRAS-$\overline{\text{SAS}}$	4.7 kΩ

The following pins should each be pulled to V_{CC} separately during normal operation when not used.

Memory Interface	Resistor Value
$\overline{\text{MBRQ}}$	1 kΩ
$\overline{\text{EXTINT0}}$-$\overline{\text{EXTINT3}}$	1 kΩ

Chapter 4

System Software Interface

This section describes the software interface between the attached system's processor and the adapter. This software interface controls the operation of the adapter and data transfer to and from the network.

The adapter is controlled through direct access to eight registers and a DMA channel contained within the TMS380C16 System Interface (SIF) function.

The system interface registers are used to initialize the adapter, read the cause of interrupts posted to the attached system, and post interrupts to the adapter to initiate DMA transfers to and from the attached system's memory.

The DMA channel is used to pass commands, parameters, and frames to the adapter and to receive completion codes and frames from the adapter. All data movement to and from the ring is via DMA only.

Section **Page**

4.1 Usage of DIO and DMA 4-2
4.2 Adapter Data Flow 4-8
4.3 DIO Register Interface 4-16
4.4 Adapter Reset 4-37
4.5 Bring-up Diagnostics—BUD 4-40
4.6 Adapter Initialization 4-42
4.7 Adapter Tests 4-49
4.8 System Command Block—SCB 4-50
4.9 System Status Block—SSB 4-52
4.10 Interface Control Block—ICB 4-53
4.11 Frame Buffers 4-57
4.12 Adapter-to-System Interrupts 4-60
4.13 Adapter MAC-Only Commands 4-70
4.14 Adapter LLC Commands 4-124

4.1 Usage of DIO and DMA

There are two methods for passing information between the adapter and the attached system:

- ❑ Direct input/output (DIO), and
- ❑ Direct memory access (DMA)

Take care when using these two methods because of the differences in how information is stored in the attached system's memory (in other words, the differences between the 808x and 68xxx mode's natural order). The natural order of a 68xxx system transferring a word in bytes is Byte +0 (MSB) first, and then Byte +1 (LSB) second, as shown below.

Natural Order for 68xxx Mode

	Byte + 0	Byte +1
Word + 0	MSB	LSB

The natural order of a 808x system transferring a word in bytes is Byte +0 (LSB) first, and then Byte +1 (MSB) second, as shown below.

Natural Order for 808x Mode

	Byte +1	Byte + 0
Word + 0	MSB	LSB

4.1.1 Description and Usage of DIO

DIO is a word-based transfer mechanism. That is, all transfer of data by the attached system via DIO is made in the attached system's natural order. If the attached system transfers information via DIO through word transfers, then the order of word byte storage does not matter, regardless of the storage format (808x or 68xxx).This assumes the attached system's machine is capable of word transfers. **However, if the machine can transfer only bytes, then the first byte transferred must be the Byte +0, and the second byte transferred must be the Byte +1**. This is in line with the 808x and 68xxx mode's natural order (that is, Byte +0 first, followed by Byte +1).

4.1.1.1 An Example of DIO Usage

If the attached system writes the word >55AA to an adapter register by a word transfer, the MSB is placed on the SADH bus and the LSB is placed on the SADL bus. This is true for both the 808x and 68xxx modes. Therefore, in both 808x and 68xxx modes, >55 (MSB) is placed on the SADH bus, and >AA (LSB) is placed on the SADL bus, as shown below.

808x	Byte + 1	Byte + 0
68xxx	Byte + 0	Byte + 1
Word + 0	>55AA	

The format of information storage does not matter for DIO word transfers. However, if DIO is done through byte transfers, Byte +0 must be transferred first and then Byte +1. Therefore, in the above example, to transfer a word (>55AA) in bytes, the following applies. In 68xxx mode, the MSB(>55) is transferred first, followed by the LSB(>AA). In 808x mode, the LSB(>AA) is transferred first, and then the MSB(>55). This allows a programmer to use word instructions to access DIO registers, regardless of physical bus size.

4.1.2 Description and Usage of DMA

Note:

All DMA data is transferred to and from the host in byte order.

DMA is a byte-based transfer mechanism. This is important because every transfer of information is done byte-by-byte. In other words, DMA transfers Byte +0, Byte +1, Byte +2, etc.

First Byte Transferred				Last Byte Transferred
Byte n	Byte n + 1	Byte n + 2	. . . .	Byte n + m

m = the number of bytes to be transferred
n = the attached system DMA base address

The attached system must ensure that when it loads or reads structures used by the adapter (such as SCBs, lists, etc.) in words and double words, the order of the bytes is as shown above. This means that if the attached system is 68xxx-based, the byte order will be correct whether using bytes, words, or longer structures. However, with 808x-based systems, the order will be correct only if the adapter structures are manipulated with byte structures. But when these structures are manipulated with anything larger than byte structures (such as a word or a double word), **the bytes within these larger structures must be byte-swapped**. Also, if a double word is used for storage, then the words, as well as the bytes, must be swapped.

In addition, the following rules apply when DMA is used:

□ All TMS380 adapter commands are handled through DMA

□ All parameter blocks and lists are handled through DMA

□ All reads and writes by the adapter, to and from the SCB (System Control Block) and SSB (System Status Block), are handled through DMA

□ All frame data is handled through DMA

□ For Intel mode, byte swapping within a word will automatically be performed by the adapter. In double word format, the host software is required to swap the words.

4.1.2.1 Example of DMA Usage (Example 1)

The following illustration is an example of most structures used in interfacing with the TMS380 adapter software by the SCB, SSB, and all command parameter tables. In this illustration, all elements are defined as words. Therefore, in 68xxx format, the table can be manipulated by the attached system in bytes, words, or double words. But in 808x-based systems, the bytes of each word must be swapped, if manipulated in structures larger than a byte.

Figure 4–1. Motorola Format

		Byte + 0	Byte + 1
Word 0	+ 0	Byte 0	Byte 1
Word 1	+ 2	Byte 2	Byte 3
Word 2	+ 4	Byte 4	Byte 5
Word 3	+ 6	Byte 6	Byte 7

<table>
<tr><th></th><th></th><th>Byte + 0</th><th>Byte + 1</th></tr>
<tr><td>Word 0</td><td>+ 0</td><td colspan="2">&gt;0001</td></tr>
<tr><td>Word 1</td><td>+ 2</td><td colspan="2">&gt;0203</td></tr>
<tr><td>Word 2</td><td>+ 4</td><td colspan="2">&gt;0405</td></tr>
<tr><td>Word 3</td><td>+ 6</td><td colspan="2">&gt;0607</td></tr>
</table>

Byte 0	=	>00
Byte 1	=	>01
Byte 2	=	>02
.		
.		
.		
Byte 7	=	>07

Note that the bytes are not swapped.

Figure 4–2. Intel Format

		Byte + 1	Byte + 0
Word 0	+ 0	Byte 1	Byte 0
Word 1	+ 2	Byte 3	Byte 2
Word 2	+ 4	Byte 5	Byte 4
Word 3	+ 6	Byte 7	Byte 6

therefore

		Byte + 1	Byte + 0
Word 0	+ 0	>0100	
Word 1	+ 2	>0302	
Word 2	+ 4	>0504	
Word 3	+ 6	>0706	

Byte 0	=	>00
Byte 1	=	>01
Byte 2	=	>02
.		
.		
.		
Byte 7	=	>07

Note that the bytes are swapped when viewed as words.

4.1.2.2 Example of DMA Usage (Example 2)

This example shows how to deal with double words when DMA is used with an 808x mode format.

Figure 4–3. Intel Double-Word Format

Intel storing of bytes in double word format

		Byte + 1	Byte + 0	
Double word 0	+ 0	Byte 1	Byte 0	Word 0
	+ 2	Byte 3	Byte 2	Word 1

In an 808x mode format, the table should look like the following.

Byte order that the adapter expects

		Byte + 1	Byte + 0
Double word 0	+ 0	>0302	
	+ 2	>0100	

Byte 0	=	>00
Byte 1	=	>01
Byte 2	=	>02
Byte 3	=	>03

Both the bytes and words are swapped.

4.2 Adapter Data Flow

The TMS380C16, at a high level of detail, performs these basic operations on the ring data stream:

- Repeats the received data without copying the data.
- Repeats and copies the received data.
- Changes the state of single bits in the received data before retransmitting it.
- Originates the transmission of data on the ring.
- Removes messages from the ring that it has previously transmitted.

Two devices make up the major functional blocks of the adapter: the TMS380C16 and the TMS38053.

The TMS380C16 combines the functions previously contained in the TMS38021 Protocol Handler (PH), the TMS38010 Communications Processor (CP), and the TMS38030 System Interface (SIF). The TMS380C16 also integrates much of the glue logic, which was required for bus interface and memory expansion with the first-generation TMS380. The logic block for memory expansion control is called the memory interface block (MIF), and it includes all of the functions contained in the DRAM memory expansion unit (MEU) used with the first-generation TMS380. The block of logic, which assists in host interface, is incorporated into the system interface functions, and it replaces approximately 50% of the bus interface unit (BIUs) used with the first-generation TMS380.

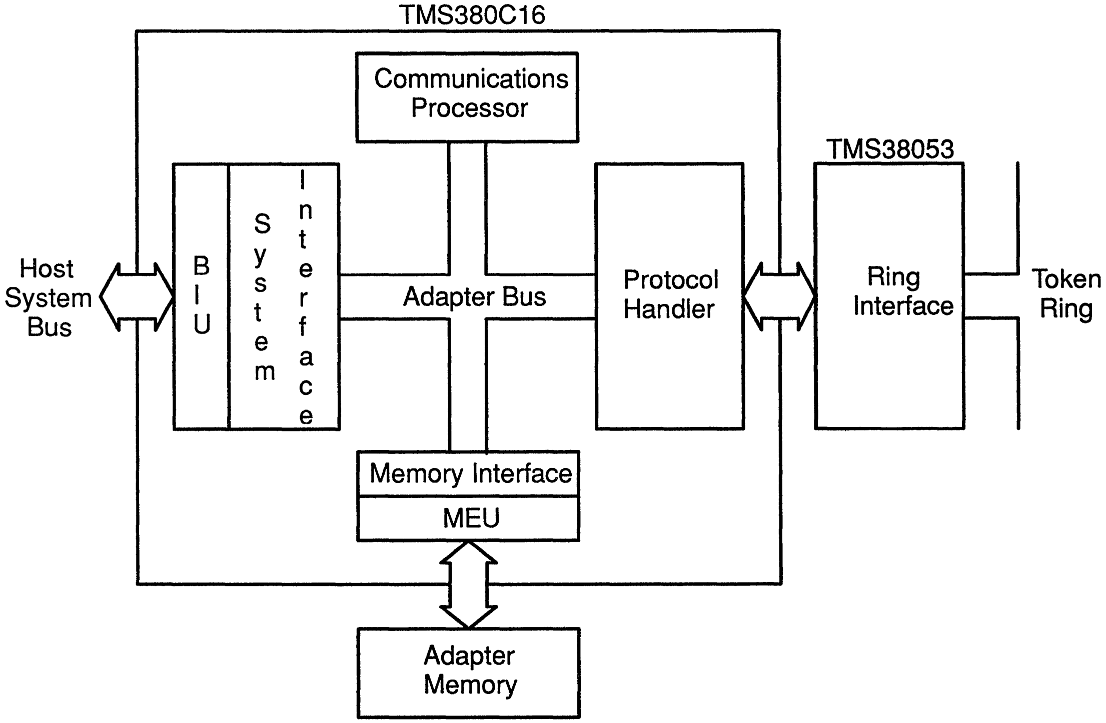
The TMS38053 combines the functions of the TMS38051 ring interface transceiver, the TMS38052 ring interface controller, and a watchdog timer.

Figure 4–4 shows the second-generation TMS380 adapter subsystem. The adapter presents a full-duplex interface to the host system with separate receive and transmit channels between the adapter and the LAN. The adapter hardware has been designed to handle reception of back-to-back frames on the LAN and to allow simultaneous reception and transmission of data at the full ring data rate.

The adapter chipset has separate receive and transmit lines to the LAN, two separate pairs of DMA channels in the protocol handler function for both receive and transmit, and an option for either 48-MHz or 64-MHz adapter bus with single-cycle arbitration for internal adapter transfers. A 64-MHz DMA controller connects the LAN adapter bus to the host system bus.

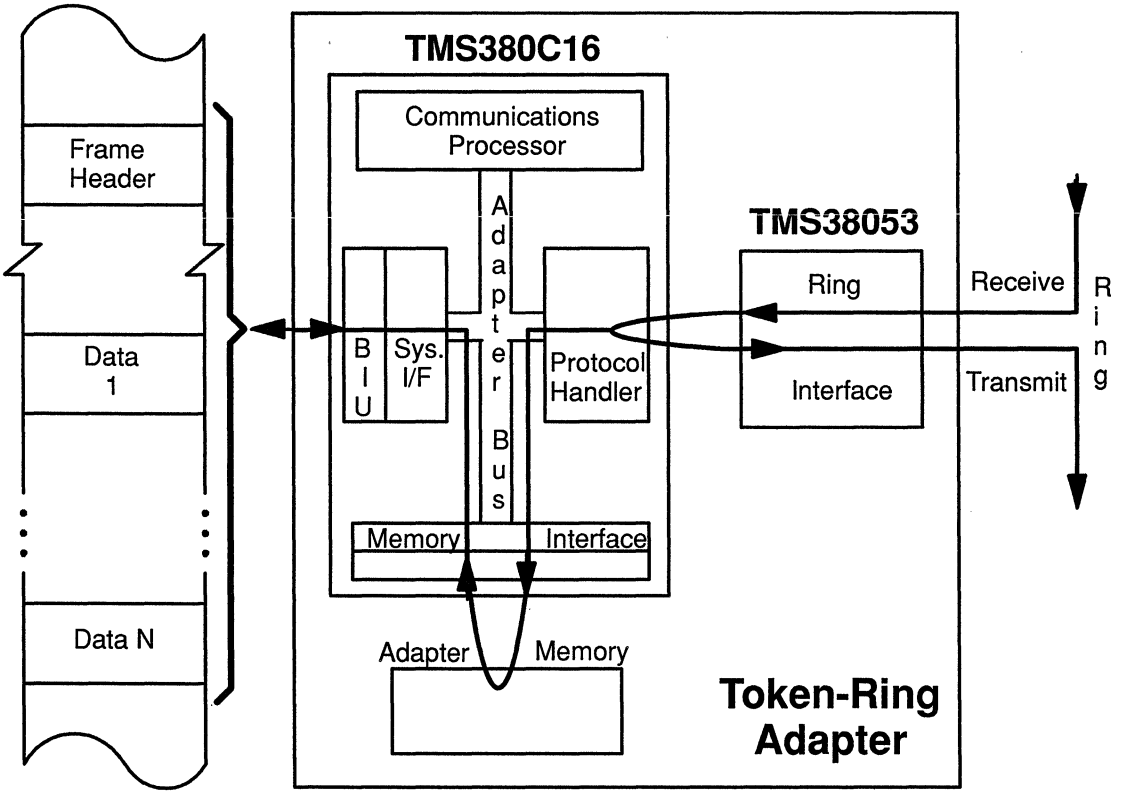
The chipset offers the designer configurable options, such as cycle-steal and programmable-burst mode operation, a linked-list interface controlled by high-level commands, polled or interrupt operation, and list-suspension capability to allow the host system to interrogate portions of an incoming frame. The adapter chipset can also scatter-write/gather-read to and from discontiguous memory locations in the host during DMA transfers. Up to three high-level commands (such as TRANSMIT, RECEIVE, and READ ERROR LOG) can be queued simultaneously on the adapter chipset at any time, for MAC code only. For LLC code, the adapter can support a TRANSMIT command, a RECEIVE command, multiple CLOSE.STATION commands, multiple CONNECT.STATION commands, and one other command simultaneously. Host system protocol efficiency is improved with the second-generation TMS380.

Figure 4–4. Second-Generation TMS380 Adapter Subsystem



The flow of data among host system, the LAN adapter chipset, and the ring is shown in Figure 4–5.

Figure 4–5. Adapter Data Flow



To insure integrity of the data flow between the attached system and the adapter, each byte may be protected with a parity bit. In addition, the integrity of the data flow within the adapter can be assured by use of optional parity. Parity generation and checking to the system interface is performed on both address and data. Parity checking in this manner, coupled with protocol-level data integrity checks, assures a high level of confidence in the validity of data transported on the ring.

A description of the data flow through the adapter is described in the following paragraphs.

4.2.1 Receive Data Flow

When receiving data is taken from the ring into the ring interface, where it is reshaped into distortion-free digital signals. The ring interface performs no code conversion, because its primary function is to provide synchronization to the received data stream and level translations to levels compatible with the protocol handler (TTL). Synchronization is achieved via an integrated phase-locked loop (PLL), which locks to the bit stream signaling rate. This clock (RCLK) is boundary-aligned to the bit stream and is passed to the protocol handler for data extraction.

The protocol handler function, as the first step during data receive, provides deserialization of the bit stream to 16-bit data words by counting clock pulses received from the ring interface. Parity bits are generated on the received data, so that the integrity of the received data may optionally be continuously monitored as it passes from the protocol handler to local storage and ultimately to the attached system. The destination address contained within the bit stream is compared against addresses stored within the protocol handler to determine if the frame is to be copied by the adapter. If so, the protocol handler transfers the frame being received to the adapter bus RAM.

The physical control fields, the destination and source addresses, the data portions, and the CRC field of the in-progress received message are now passed to the local RAM in sequence. When the end of the CRC-protected field is received, the calculated value is compared to the CRC value received as part of the message. If these two values disagree, the protocol handler signals the communications processor that an error has occurred and the frame should be ignored. When the CRC values agree, the protocol handler changes the value of the appropriate bits within the re-transmitted data stream to indicate that the frame was properly received.

After a valid frame has been received and buffered to the adapter bus RAM, the adapter begins transfer of the frame to the attached system via direct memory access. When the transfer is complete, the adapter completes the receive operation by reporting the status of the transfer to the attached system. Specifics about the DMA options and other initial conditions, as they relate to the adapter/attached system interface, are provided via parameters sent to the adapter during initialization.

4.2.2 Transmit Data Flow

Data flow on transmit operations is essentially the reverse of data flow on receive operations. The frame to be transmitted onto the ring is in the attached system memory. The adapter is set up with information pertaining to the starting location and length so that a direct memory access can take place. The adapter then does DMA accesses to transfer the frame, including the destination address, from the system memory to adapter memory. Once the frame has been transferred from the host system to the adapter, the communications processor sets up the protocol handler to initiate transfer of the frame from the adapter RAM buffers out onto the ring.

Upon sensing that a transmission is pending, the protocol handler begins to transfer the message from the adapter RAM buffers into the PH's internal registers. After sufficient data has been transferred to sustain continuous transfers to the ring, the protocol handler begins to search for a token on the ring. When a token is received, the protocol handler modifies the token into a frame, appends applicable control bits, including the source and destination addresses, and begins a sequential transmission of the message onto the ring. Concurrent with this process, the protocol handler begins to accumulate the CRC field to be transmitted at the appropriate time. The concurrent transfer of the message from the adapter buffer RAM to the protocol handler, and the subsequent transmission of the message onto the ring, continue until the entire message has been transmitted. At this time, the protocol handler transmits the accumulated CRC. The CRC field is followed with encoded ending fields.

After transmission has begun, the protocol handler begins to remove (strip) data from the ring that is being received. The adapter searches the incoming data stream for a match of the source address of the frame transmitted by the adapter. After a match is found and transmission is complete, the adapter encodes and releases a token onto the ring. The transmission of the token is followed by a continuous transmission of zeros. After receipt of the end of the transmitted message, the adapter begins to repeat the received data stream, and the transmit operation is complete. The adapter passes status back to the attached system, indicating the completion status of the transmission.

If the early token release option is selected, the adapter's token release procedure is altered. The transmitting station sends a data frame, then issues a token. The token is then available for other stations to use for transmission. The original station receives its frame back, strips it from the network, but does not issue a token, because it already had done so using the early token release option. For more information on early token release, refer to the IEEE 802.5–1989 specifications.

4.2.2.1 Transmitting and Receiving a Frame

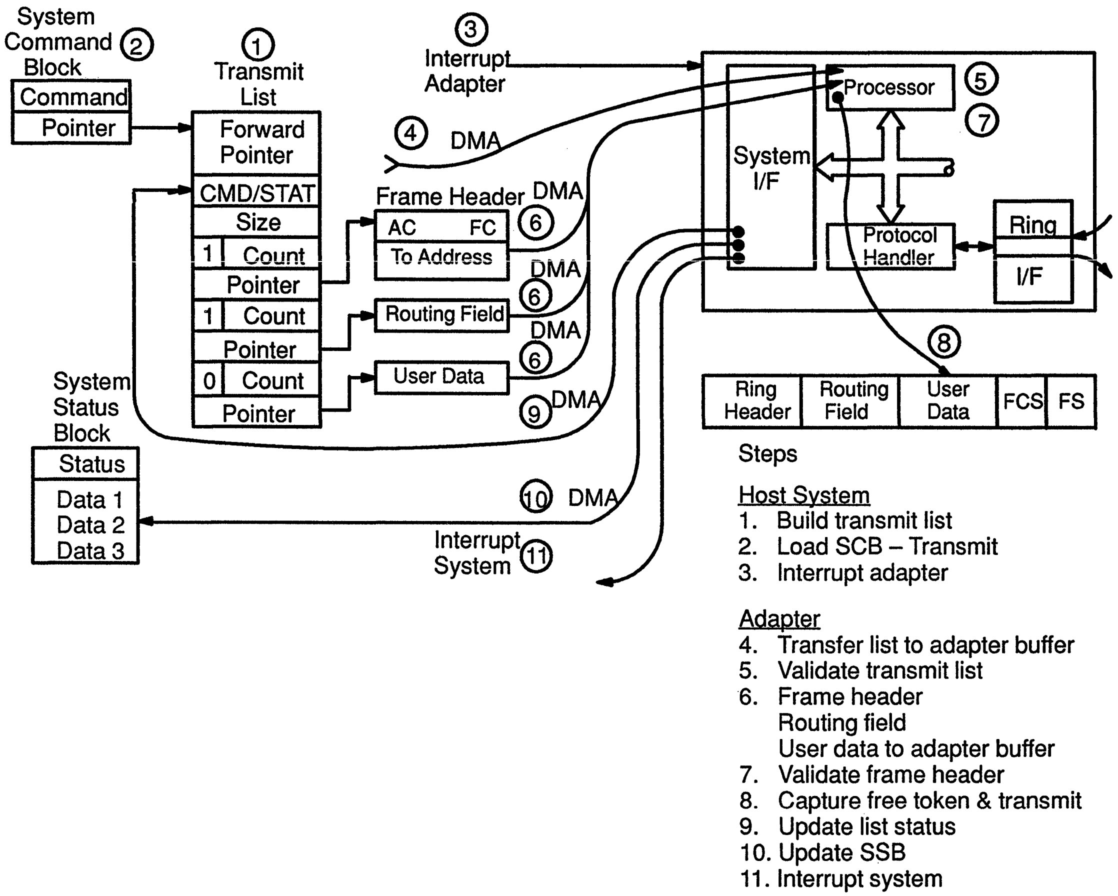
The TMS380C16 presents a linked-list interface to the host system bus that is controlled by high-level commands. During adapter chipset initialization, a host system configures the adapter interface. The adapter chipset can be initialized to match specific host system bus requirements. This adapter-to-host interface is controlled via eight 16-bit, logical DIO registers described in section 4.3. Command and command status information is passed between the adapter and host system via two control blocks:

- ❑ system command block (SCB)
- ❑ system status block (SSB)

To transmit a frame, the host system first interrupts the adapter chipset. The adapter then reads the system command block by DMA. The SCB contains the transmit command and the starting address of the transmit list that resides in host system memory. The adapter chipset then transfers the transmit list from the host system to the adapter's memory by DMA. Next, the adapter transfers the entire frame to the adapter memory, and the adapter chipset captures a token and transmits the frame onto the LAN. The frame circulates to the destination address, which copies the data and returns the frame to circulate on the LAN. When the frame returns to the originating adapter chipset, it is stripped and a token is released. Note that when the early token release option is selected, the adapter releases a token after transmission has occurred. The system status block is updated by the adapter and transferred to the host system after the frame has been stripped.

Figure 4–6 illustrates this procedure.

Figure 4–6. Transmitting a Frame



Steps

Host System
1. Build transmit list
2. Load SCB – Transmit
3. Interrupt adapter

Adapter
4. Transfer list to adapter buffer
5. Validate transmit list
6. Frame header
   Routing field
   User data to adapter buffer
7. Validate frame header
8. Capture free token & transmit
9. Update list status
10. Update SSB
11. Interrupt system

A frame is received in a manner similar to the way it is transmitted. An SCB is issued to the adapter pointing to the receive list in host system memory. Upon reception of a frame, the adapter transfers the receive list from the host system to the memory and then transfers the received data to the appropriate location in host system memory. The SSB is then updated.

4.2.3 Summary of System Buffer Requirements

The integration of the adapter into an attached system requires that several system memory buffers be allocated and reserved for adapter use. The following list summarizes the necessary system memory allocations; it does not describe their application in detail.

System command block The system command block (SCB) is a six-byte buffer that holds the command to be executed by the adapter and a 32-bit address pointer to a parameter block or buffer.

System status block The system status block (SSB) is an eight-byte buffer that holds status codes returned upon completion of adapter commands or adapter status interrupt.

Command parameter lists Certain commands (like the OPEN command) require that a block of memory be designated as a parameter block. Once the command has completed execution, this buffer allocation may be released for other uses.

Receive list The RECEIVE command requires that receive list(s) be allocated within system memory. The memory allocation size is dependent upon the size and number of lists used in the application. The size of a receive list may be selected upon opening the adapter to be either 14, 20, or 26 bytes in length. The number of lists is application-dependent.

Transmit list The TRANSMIT command requires that transmit list(s) be allocated within system memory. The memory allocation size is dependent upon the size and number of lists used in the application. The size of a transmit list may be selected upon opening the adapter to be either 14, 20, or 26, bytes in length. The number of lists is application-dependent.

Product ID The OPEN command requires a pointer to an 18-byte product ID as part of the open parameter list. The system software designer should reserve 18 bytes of system memory for this function. After the OPEN command completes, this memory may be released for other uses. Additional information on product IDs may be found in the IBM token-ring architecture reference.

Note:

The adapter must have a minimum of three transmit and three receive lists, plus associated buffers for circular list processing.

4.3 DIO Register Interface

The LAN adapter appears in the memory space of the user system as byte addresses. In the 16-bit mode, the system processor may access either one or two bytes per system bus cycle. In 808x 8-bit mode, it can access only one byte at a time.

When the host processor is operating in the 16-bit mode (with 16-bit transfers), each 16-bit register (both bytes) is written to or read from in a single DIO system access cycle. When it is operating in 8-bit mode, or doing single-byte transfers in a 16-bit mode, each 16-bit register is written to or read from as a sequence of two successive DIO access cycles. When the processor is executing a 16-bit transfer instruction in an 8-bit mode, the even byte (processor address A0 = 0) is always accessed first, and the odd byte (A0 = 1) is accessed last.

The physical register selected on each byte cycle depends on the SI/M control signal. In the 808x mode (SI/M = 1), the LSbyte of a 16-bit DIO register is selected when the processor A0 bit = 0; and the MSbyte is selected when the processor A0 bit = 1. In a 808x system, the A0 address bit is connected directly to the SRS2/SBERR DIO address pin. Hence, in the 808x mode, the SIF determines which byte is being selected by the value on the SRS2/SBERR pin.

In 68xxx mode (SI/M = 0), the MSbyte of a 16-bit DIO register is selected when the processor A0 bit = 0; and the LSbyte is selected when the processor A0 = 1. In a 68xxx system, the A0 bit is not brought out to the address bus. Instead, the SIF determines which byte is being selected by sensing the state of the SUDS and SLDS strobes: SUDS = 0 for A0 = 0; SLDS = 0 for A0 = 1. When reading from or writing to a 16-bit DIO register as a sequence of two 8-bit operations on a 16-bit bus, the host system should, in general, access the even byte (A0 = 0) first and the odd byte (A0 = 1) second. This is compatible with the natural order of operations performed by a 16-bit transfer instruction running on an 8-bit bus (such as the Intel 8088 processor). See Section 4.3.8, **8-Bit DIO**, for more details on when the byte sequence must be in the natural order.

Table 4–1. Logical DIO Interface

808x 16-Bit Mode: (SI/$\overline{\text{M}}$ = 1, S8/$\overline{\text{SHALT}}$ = 0)						
Word Transfers			**Normal Mode** $\overline{\text{SBHE}}$ = 0 SRS2 = 0		**Pseudo-DMA Mode Active** $\overline{\text{SBHE}}$ = 0 SRS2 = 0	
Byte Transfers			$\overline{\text{SBHE}}$ = 0 SRS2 = 1	$\overline{\text{SBHE}}$ = 1 SRS2 = 0	$\overline{\text{SBHE}}$ = 0 SRS2 = 1	$\overline{\text{SBHE}}$ = 1 SRS2 = 0
SRSX	**SRS0**	**SRS1**				
0	0	0	SIFDAT MSB	SIFDAT LSB	—	SDMADAT
0	0	1	SIFDAT/INC MSB	SIFDAT/INC LSB	DMALEN MSB	DMALEN LSB
0	1	0	SIFADR MSB	SIFADR LSB	SDMAADR MSB	SDMAADR LSB
0	1	1	SIFCMD	SIFSTS	SDMAADX MSB	SDMAADX LSB
1	0	0	SIFACL MSB	SIFACL LSB	SIFACL MSB	SIFACL LSB
1	0	1	SIFADR MSB	SIFADR LSB	SIFADR MSB	SIFADR LSB
1	1	0	SIFADX MSB	SIFADX LSB	SIFADX MSB	SIFADX LSB
1	1	1	DMALEN MSB	DMALEN LSB	DMALEN MSB	DMALEN LSB

(SBHE = 1 and SRS2 = 1 is not defined)

808x 8-Bit Mode: (SI/$\overline{\text{M}}$ = 1, S8/$\overline{\text{SHALT}}$ = 1)					
SRSX	**SRS0**	**SRS1**	**SRS2**	**Normal** $\overline{\text{SBHE}}$ = X	**Pseudo-DMA** $\overline{\text{SBHE}}$ = X
0	0	0	0	SIFDAT LSB	SDMADAT
0	0	0	1	SIFDAT MSB	—
0	0	1	0	SIFDAT/INC LSB	DMALEN LSB
0	0	1	1	SIFDAT/INC MSB	DMALEN MSB
0	1	0	0	SIFADR LSB	SDMAADR LSB
0	1	0	1	SIFADR MSB	SDMAADR MSB
0	1	1	0	SIFSTS	SDMAADX LSB
0	1	1	1	SIFCMD	SDMAADX MSB
1	0	0	0	SIFACL LSB	SIFACL LSB
1	0	0	1	SIFACL MSB	SIFACL MSB
1	0	1	0	SIFADR LSB	SIFADR LSB
1	0	1	1	SIFADR MSB	SIFADR MSB
1	1	0	0	SIFADX LSB	SIFADX LSB
1	1	0	1	SIFADX MSB	SIFADX MSB
1	1	1	0	DMALEN LSB	DMALEN LSB
1	1	1	1	DMALEN MSB	DMALEN MSB

Table 4–1. Logical DIO Interface (Concluded)

68xxx Mode: (SI/$\overline{\text{M}}$ = 0)						
Word Transfers			**Normal Mode** **$\overline{\text{SUDS}}$ = 0** **$\overline{\text{SLDS}}$ = 0**		**Pseudo-DMA Mode Active** **$\overline{\text{SUDS}}$ = 0** **$\overline{\text{SLDS}}$ = 0**	
Byte Transfers			**$\overline{\text{SUDS}}$ = 0** **$\overline{\text{SLDS}}$ = 1**	**$\overline{\text{SUDS}}$ = 1** **$\overline{\text{SLDS}}$ = 0**	**$\overline{\text{SUDS}}$ = 0** **$\overline{\text{SLDS}}$ = 1**	**$\overline{\text{SUDS}}$ = 1** **$\overline{\text{SLDS}}$ = 0**
SRSX	**SRS0**	**SRS1**				
0	0	0	SIFDAT MSB	SIFDAT LSB	—	SDMADAT
0	0	1	SIFDAT/INC MSB	SIFDAT/INC LSB	DMALEN MSB	DMALEN LSB
0	1	0	SIFADR MSB	SIFADR LSB	SDMAADR MSB	SDMAADR LSB
0	1	1	SIFCMD	SIFSTS	SDMAADX MSB	SDMAADX LSB
1	0	0	SIFACL MSB	SIFACL LSB	SIFACL MSB	SIFACL LSB
1	0	1	SIFADR MSB	SIFADR LSB	SIFADR MSB	SIFADR LSB
1	1	0	SIFADX MSB	SIFADX LSB	SIFADX MSB	SIFADX LSB
1	1	1	DMALEN MSB	DMALEN LSB	DMALEN MSB	DMALEN LSB

4.3.1 SIFACL—SIF Adapter Control Register

The SIFACL register (see Figure 4–7) allows the host processor to control and to some extent reconfigure the adapter under software control.

Figure 4–7. SIFACL—SIF Adapter Control Register

Bit	**Name**	**Description**
0—3	—	Unimplemented
4	SWHLDA	Software hold acknowledge
5	SWDDIR	Current SDDIR signal value
6	SWHRQ	Current SHRQ signal value
7	PSDMAEN	Pseudo system DMA enable
8	ARESET	Adapter reset
9	CPHALT	Communication processor halt
10	BOOT	Bootstrapped CP code
11	—	Reserved; must be set to zero
12	SINTEN	System interrupt enable
13	PEN	Adapter parity enable
14	INPUT0	Reserved
15	—	Unimplemented

(Adapter address >01.011C)

Bits 0—3: Unimplemented

These bits always read as zero. Writing to them has no effect.

Bit 4: SWHLDA—Software Hold Acknowledge

This bit, which can be read from and written to, is relevant only when the pseudo-system-DMA enable bit (PSDMAEN) is a one; otherwise, it is a don't care. When PSDMAEN is one, this bit replaces the SHLDA pin function of the SIF host interface, enabling the host to generate a hold acknowledge through software, rather than in hardware. (ANY value applied to the SHLDA pin is ignored by the SIF.) This bit is used in conjunction with the SWHRQ bit, which is also in the SIFACL register. The SWHLDA bit has an active high sense. When used in Motorola mode, SWHLDA will replace the function of the SBGR signal (instead of the SHLDA signal of Intel mode). SWHLDA is set to zero by a reset and can be set to a one only when the ARESET bit is zero, PSDMAEN bit is one, and SWHRQ bit is a one. SWHLDA is automatically cleared upon the end of pseudo-DMA operation. The operation of SWHLDA is detailed in Figure 4–8.

Bit 5: SWDDIR—Current SDDIR Signal Value

This bit, which is read only (writing to it causes no effect), returns the current value of the SDDIR host interface signal when read. This enables the host to easily determine the direction of DMA data transfer to allow system DMA to be controlled by system software. This bit can be read at any time, even in reset. The operation of this bit is detailed in Figure 4–8.

Value	**Direction**
0	Host system to adapter DMA (input to adapter)
1	Adapter to host system DMA (output from adapter)

Bit 6: SWHRQ—Current SHRQ Signal Value

This bit, which is read only (writing to it causes no effect), returns the current value of the SHRQ host interface signal when read in Intel mode. This enables the host to know that an adapter DMA is pending. When in Motorola mode, this bit returns the inverse value of the same pin and indicates the value of $\overline{\text{SBRQ}}$; that is, this bit is always active high. The SHRQ/$\overline{\text{SBRQ}}$ pin and SWHRQ bit remain active throughout the pseudo-DMA operation and go inactive when the entire DMA between adapter and attached system is complete. This bit can be read at any time, even in reset, and its operation is detailed in Figure 4–8.

Bit 7: PSDMAEN—Pseudo-System-DMA Enable

This bit, which can be read from and written to, and is set to zero by reset, is a system-interrupt enable signal for the SHRQ/$\overline{\text{SBRQ}}$ signal. When this bit is a one, the SINTR/$\overline{\text{SIRQ}}$ signal generates a system-interrupt request whenever the SHRQ/$\overline{\text{SBRQ}}$ signal is active. This interrupt function is in addition to the existing interrupt function related to $\overline{\text{SIRQ}}$. The interrupt signal now, therefore, becomes active whenever $\overline{\text{SIRQ}}$ and SINTEN are both one, or when PSDMAEN is one and SWHRQ is one.

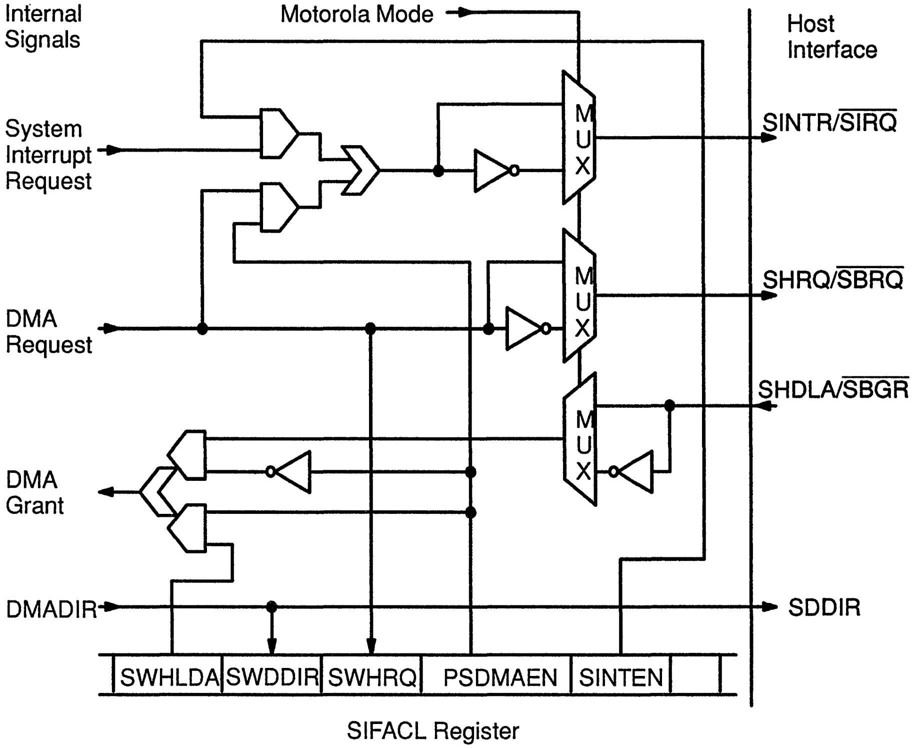
PSDMAEN performs another function in addition to the interrupt enable described above. It acts as a multiplexer control signal upon the host DMA grant internal signal. Whenever the bit is zero, the value on the SHLDA/$\overline{\text{SBGR}}$ pin is the signal used by the SIF internally for SDMA control. When the bit value is a one, the internal signal is taken from the SWHLDA bit, and the SHLDA/$\overline{\text{SBGR}}$ pin value has no meaning.

This bit can be read from at any time, regardless of the value of ARESET. It can be written to only with a one when ARESET is zero. The operation of this bit is detailed in Figure 4–8.

Note:

Bits 4 to 7 of SIFACL are used only when it is necessary to perform a pseudo-DMA on a system bus that will not support a true DMA (in other words, it will not allow an external master to take control of the system bus). Bits 4 to 7, together with SDMAADR, SDMAADX, DMALEN, and SDMADAT registers of the SIF, can be used to perform a software handshake routine that allows the SIF to think that it is performing a true DMA; but in reality it is latching its DMA address in the SDMAADR and SDMAADX registers and transferring its data to/from the SDMADAT registers. The system then reads the address and length registers of the SIF and transfers the data from/to the SDMADAT register. Pseudo-DMA is an 8-bit DMA system, regardless of host DIO size or DMA size latched at reset.

Figure 4–8. Pseudo-DMA Logic Related to SIFACL Bits



Bit 8: ARESET—Adapter Reset

When this bit is set to one, TMS380C16 is reset and held in a reset condition for as long as the bit remains one. It has exactly the same effect as the $\overline{\text{SRESET}}$ pin except that the DIO interface to the SIFACL register is maintained both to allow this bit to be reset to zero, as well as to allow the other bits in the SIFACL register to be changed. This bit can be read from or written to and is set to zero by the $\overline{\text{SRESET}}$ signal driven active low. This bit also affects the operation of the BOOT, CPHALT, and PEN bits in this register, as described in their subsections. After ARESET is released, the adapter software must be redownloaded.

This bit can also be set to one if any clock inputs to the TMS380C16 are not valid (such as OSCIN, PXTALIN, RCLK, or SBCLCK).

Bit 9: CPHALT—Communications Processor Halt

In adapters where the CP's code is RAM-based, it is necessary to prevent the CP from attempting to execute code before the code has been downloaded from the host system. This is achieved by setting the CPHALT bit to one. This signal is passed to the MIF arbitration logic and prevents the CP from being granted access to the internal adapter buses. When this bit is zero, the CP is granted bus access whenever no other device requires it.

Whenever SRESET is low, or when ARESET is changed from zero to one, this bit is loaded with a copy of the value on the BTSTRP pin. This automatically sets the CPHALT bit to the correct default value to allow the CP to gain bus accesses if its code is RAM-based and to prevent it from gaining bus accesses if its code is ROM-based.

Whenever the ARESET bit is one, this bit can be written to with one or with zero. After the ARESET bit has been set to a zero, this bit can be set only to a zero. Writing a one has no effect. The CP cannot therefore be halted without also resetting the adapter. It can be read, however, at any time.

Bit 10: BOOT—Bootstrap CP Code

This bit indicates to the MIF's memory control whether the memory existing in chapters 0 and 31 of the adapter memory space is RAM-based or ROM-based. (A one means RAM-based, a zero means ROM-based.) The MIF then uses this information to control the MCAS and MROMEN signals to select either one or the other memory type.

Whenever SRESET is low, or when ARESET is changed from a zero to a one, this bit is loaded with a copy of the value on the BTSTRP pin. This automatically sets the boot bit to the correct default value to indicate to the MIF the CP code memory type.

Whenever the ARESET bit is one, this bit can be written to with one or with zero. After the ARESET bit has been set to a zero, this bit becomes write-protected and cannot be changed. It can, however, be read at any time.

Bit 11: Reserved; must be set to zero

Note:

Undefined events may occur if this bit is set to one.

Bit 12: SINTEN—System-Interrupt Enable

This bit, which is set to one by reset, allows the host processor to enable or disable system interrupt requests from the CP. When the bit is set to one, system interrupts are issued by the SIF logic; when it is set to zero, system interrupts are not requested, even though the SYSTEM_INTERRUPT bit in the SIFSTS register may be a one. System interrupts are generated even with this bit at zero if the PSDMAEN bit is a one and the SHRQ/$\overline{\text{SBRQ}}$ pin is active. This bit can be read by the host processor at any time, regardless of the value of ARESET. It can be written with either a one or a zero when ARESET is zero. This operation is indicated in Figure 4–8.

Bit 13: PEN—Adapter Parity Enable

This bit determines whether or not data transfers within the adapter are checked for correct parity. When this bit is one, the parity checkers in the MIF, SIF, and PH, which connect to the internal data bus, check data they receive for correct odd parity. When zero, all internal data transfers are not checked for parity. (This bit does not control the parity checking on the system bus.)

Whenever $\overline{\text{SRESET}}$ is low, or when ARESET is changed from zero to one, this bit is loaded with a copy of the value on the PRTYEN pin. This automatically sets the PEN bit to the correct default mode.

Whenever the ARESET bit is one, this bit can be written to with one or with zero. After the ARESET bit has been set to zero, this bit becomes write- protected, and cannot be changed. It can, however, be read from at any time.

Bit 14: INPUT0—Reserved; read-only bit.

Bit 15: Not Implemented

This bit is always read zero. Writing to it has no effect.

4.3.2 SIFADX, SIFADR, SIFDAT, and SIFDAT/INC Registers

The system interface is designed to allow the user processor to directly access the SIF registers and to indirectly access the adapter memory without intervention by the adapter software executed by the communications processor function. For this purpose, the DIO portion of the system interface is equipped with its own internal DMA controller with which to perform accesses of the adapter memory. This DMA logic is separate from and independent of the DMA channel used for DMA transfers between the system memory and the adapter memory. Contention between the user processor and the communications processor for access to the adapter memory is handled by on-chip arbitration.

The DIO section of the SIF has been designed to support the convenient loading or fetching of a series of contiguous buffer locations in the adapter memory by using the SIFADR autoincrement feature. Alternately, the non-increment option can be used to read from and then write to a single memory location, such as performing a software-controlled handshake between the user processor and the CP. The same memory location can be read from repeatedly, each access returning to the user processor a more recent copy of the contents of the memory location. Any adapter address, including the SIF control registers, can be accessed via DIO.

The DIO section of the SIF cannot write to the PH registers. Reading internal PH or CP registers on TMS380C16 with SIF DIO in normal operations may cause the adapter software to fault.

Note:

The SIFADR is duplicated in the SIF DIO registers. They access the same internal register.

Indirect access of adapter memory is performed by accessing either of two SIF register addresses, SIFDAT or SIFDAT/INC. These operate in conjunction with the SIF address pointer registers (SIFADX and SIFADR) and the I/O write enable, a function controlled by adapter microcode. SIFADX and SIFADR together contain a 21-bit adapter address; the user processor loads this address by accessing the SIFADX and SIFADR registers individually. The 16 LSBs are written to SIFADR and the 5 MSBs to the 5 LSBs of SIFADX. The LSB of SIFADR is a don't care because adapter memory accesses are always a 16-bit word. (Individual byte addressing of adapter memory is not supported.)

In general, the SIF maintains a copy of the word addressed by SIFADX and SIFADR in the SIFDAT register. Reading from the SIFDAT register returns the word in the adapter memory pointed to by SIFADX and SIFADR. Writing to the SIFDAT register causes the word pointed to by SIFADX and SIFADR to be modified. Reading from or writing to the SIFDAT/INC register has the same effect as reading from or writing to the SIFDAT register, except that following each word access, the address contained in the SIFADX/SIFADR combination is automatically incremented by two in order to point to the next word in the memory. Before the autoincrement feature is used, the SIFADX and SIFADR combination is set up to point to the first location of a buffer region in memory; following each subsequent access of the memory using the SIFDAT/INC register, SIFADX/SIFADR is automatically incremented by two to point to the next word location in memory. In this manner, a series of contiguous locations can be accessed following a single load of the SIFADX and SIFADR registers without additional pointer-management overhead.

When the system writes to SIFADR or reads from the SIFDAT register, the SIF's DIO logic acquires access to the adapter memory and reads the word addressed by SIFADX/SIFADR into the physical SIFDAT and SIFDAT/INC registers. The adapter will prefetch the next data value pointed to by the SIFADR and SIFADX registers into the SIFDAT and SIFDAT/INC registers.

To read a block of *N* words starting at address START in adapter memory, the system processor performs the following actions:

- ❑ Writes the value START into SIFADX and then into SIFADR, in that order.
- ❑ Reads the SIFDAT/INC register location *N* times.

When the system processor writes to the SIFDAT register, the DIO logic acquires access to the adapter memory and rewrites the data stored in SIFDAT to the adapter memory location addressed by SIFADX/SIFADR.

When the system writes to SIFDAT/INC, the DIO block again writes the SIFDAT/INC data into adapter memory, but it also increments SIFADX/SIFADR by two after the operation.

Each subsequent write to SIFDAT/INC stores data into sequential adapter memory addresses. After DIO logic increments SIFADX and SIFADR, the adapter location addressed by SIFADX and SIFADR is not read back into SIFDAT. If the system writes to SIFDAT or SIFDAT/INC and then reads it back without modifying SIFADX and SIFADR, the original write data is returned.

To write a block of *N* words starting at address START in adapter memory, and then read a word from address STATUS, the system processor performs the following actions:

1) Writes the value of START into SIFADX and then into SIFADR, in that order.
2) Writes data to the SIFDAT/INC register location *N* times. After each write, the DIO DMA logic writes the data into adapter memory and increment the SIFADX/SIFADR combination by two.
3) Writes the address value of STATUS into SIFADX and then SIFADR, in that order. This causes the DIO to prefetch the word at address STATUS.
4) Reads the SIFDAT register to obtain the prefetched value.

4.3.3 SDMADAT Register

During pseudo-DMA to/from the host, the host processor may read from, or write to, this register to complete the pseudo-DMA operation. During pseudo-DMA to the host, this register contains the data written to the host memory. Writing to this register during an adapter-to-host transfer has no effect. Data is transferred from the adapter memory in byte order. During pseudo-DMA from the host, this register awaits data from the host, and reading this register has no effect; it is read as zeros. Data is transferred to the adapter memory in byte order. Together with SDMAADR, SDMAADX, and DMALEN, this register can be used to pseudo-DMA data to/from the host when the host is unable to support DMA masters. This register cannot be accessed in any mode other than pseudo-DMA mode: that is, when PSDMAEN, SWHRQ, and SWHLDA bits of SIFACL register are all one. The SDMADAT register replaces the SIFDAT register during pseudo-DMA operation.

4.3.4 DMALEN Register

During pseudo-DMA to/from the host, the host processor may read from, but not write to, the register indicating the present DMA's transfer length. The DMALEN register replaces the SIFDAT/INC register during pseudo-DMA operation.

4.3.5 SDMAADR Register

During pseudo-DMA to/from the host, the host processor may read from, but not write to, the register indicating the present DMA's sixteen least significant bits of host address. Together with SDMAADX, this register can be used to determine the address of the DMA transfer in progress. This register cannot be accessed in any mode other than pseudo-DMA mode: that is, when PSDMAEN, SWHRQ, and SWHLDA bits of SIFACL register are all one. The SDMAADR register replaces the SIFADR register during pseudo-DMA operation.

4.3.6 SDMAADX Register

During pseudo-DMA to/from the host, the host processor may read from, but not write to, the register indicating the present DMA's sixteen most significant bits of host address. Together with SDMAADR this register can be used to determine the address of the DMA transfer in progress. This register cannot be accessed in any mode other than pseudo-DMA mode: that is, when PSDMAEN, SWHRQ, and SWHLDA bits of SIFACL register are all one. The SDMAADX register replaces the SIFCMD/STS register during pseudo-DMA operation.

4.3.7 Pseudo-DMA Operation

Pseudo-DMA is a method of performing DMA in a host that does not support bus master DMA. The DMA is accomplished by the host interrupt service routine, which transfers data from/to the adapter to/from host memory, using only DIO registers. The pseudo-DMA interface is an 8-bit interface; this means that only 8 bits of data at a time are read from or written to the adapter.

During adapter setup, the host sets the PSDMAEN bit (bit 7) of the SIFACL register, enabling pseudo-DMA operations. When the adapter requires a DMA to be performed, a host interrupt is set. When this interrupt is serviced by the host, it checks the SWHRQ (bit 6) and PSDMAEN bits of the SIFACL register to first determine if there is a pseudo-DMA request. If the SWHRQ bit is not set, the host processes SIFSTS interrupts as normal (the SYSTEM_INTERRUPT bit of SIFSTS is set.) When the SWHRQ and PSDMAEN bits of SIFACL are set to a one, this indicates a pseudo-DMA request. The host then sets the SWHLDA bit (bit 4) in the SIFACL register and reads the DMA length from the DMALEN register and the 32-bit host system's physical start address from the SDMAADX and SDMAADR registers, MSW and LSW, respectively.

The host then determines the direction of DMA by the SWDDIR bit (bit 5) of the SIFACL register. If the SWDDIR bit is clear (0), specifying DMA from host to adapter, the host reads a byte of data from the physical address, increments the address for the next cycle, and writes it to the SDMADAT register for DMA-length number of times, incrementing the host address each time. If SWDDIR bit is set (1), specifying DMA from adapter to host, the host reads a byte of data from the SDMADAT register, writes it to the physical address, and increments the address for the next cycle for DMA-length number of times.

Once the pseudo-DMA is complete, the SWHRQ and SWHLDA bits are automatically cleared. The host then issues the End-of-Interrupt (EOI) to the interrupt controller and returns from the interrupt.

Note:

When the pseudo-DMA interface is used, the interrupt acknowledge vector is not changed; that is, the interrupt vector presented on the bus during an interrupt acknowledge cycle for a pseudo-DMA will be whatever it was on the last SIFSTS interrupt. So, when pseudo-DMA is used with the interrupt vectors, the first operation of each vector is to check whether pseudo-DMA is required.

The SHRQ/$\overline{\text{SBRQ}}$ pin is the hardware representation of the SWHRQ bit (SHRQ/$\overline{\text{SBGR}}$ is active when SWHRQ is active). When the PSDMAEN bit is set, the SHLDA/$\overline{\text{SBGR}}$ pin is ignored and the $\overline{\text{SOWN}}$ pin is always inactive.

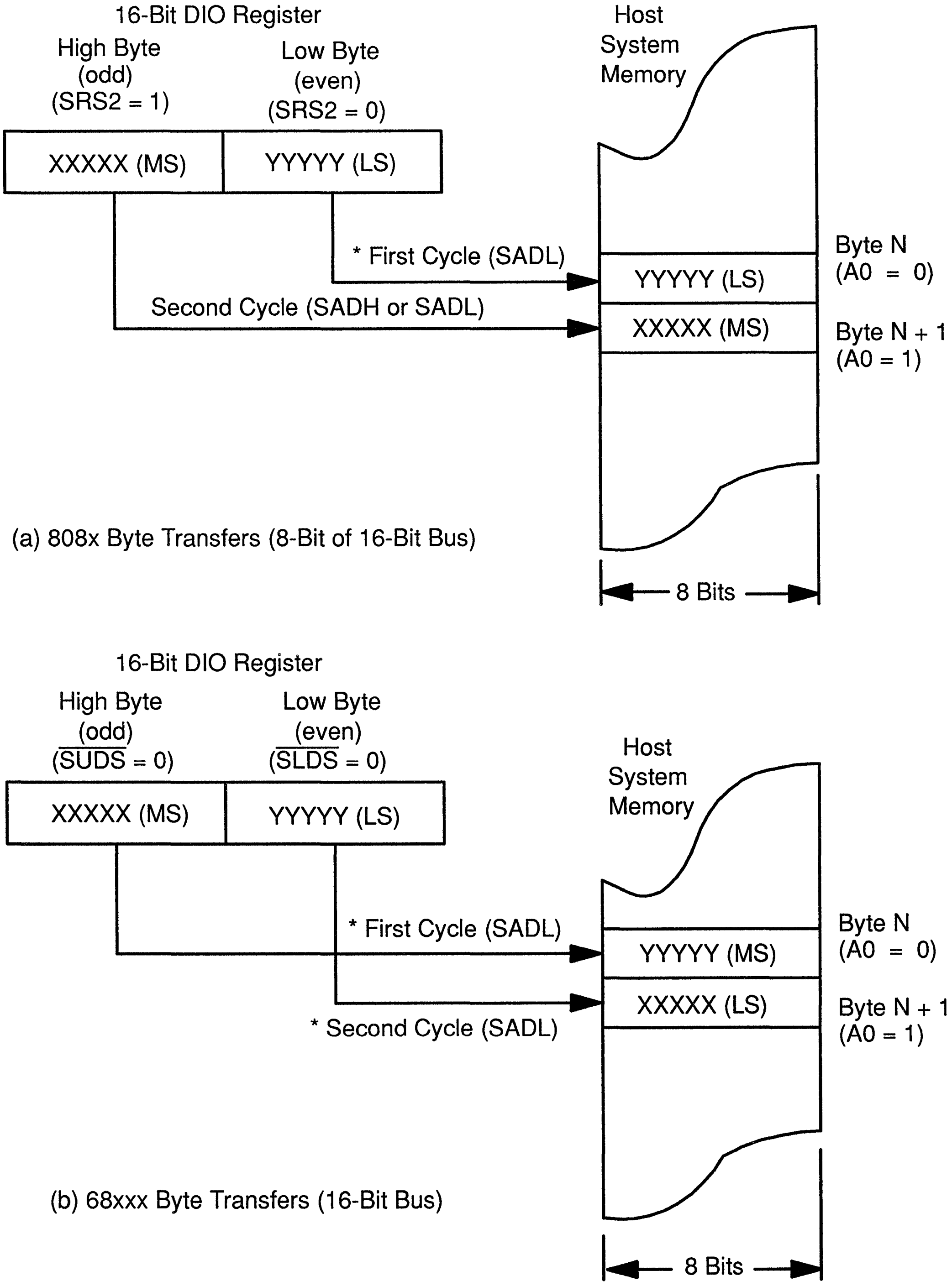
4.3.8 8-Bit DIO

When the system interface is configured in 808x 8-bit mode, or when 8-bit data transfers are performed in 16-bit mode, the user processor must be careful to transfer each byte to/from the correct corresponding 8 bits of the DIO register. Internal to the chip, transfers are still based on the movement of words; that is, each DIO register is accessed over the adapter memory bus as a word, and not as a sequence of two bytes.

In the case of a SIFDAT or SIFDAT/INC data transfer or a SIFADR write, take additional care to insure that the attached processor reads or writes the two bytes in the sequence expected by the adapter. The adapter expects the byte sequence to be the same as one that would occur when an 8-bit processor executes a 16-bit mode instruction. Even though the SIF does not support an 8-bit 68xxx interface, the sequence of bytes expected still corresponds to the sequence that would be generated by an 8-bit 68xxx processor. **The basic byte sequence rule is the even byte (the one addressed when the processor LS address bit A0 = 0) is accessed first, and the odd byte (addressed when A0 = 1) is accessed second.**

Figure 4–9(a) shows how the host memory byte addresses correspond to the two bytes of a DIO register for a 808x microprocessor operating in either 8-bit or 16-bit mode; Figure 4–9(b) shows the byte correspondence for a 68xxx microprocessor operating in the 16-bit mode. In the 808x mode, the LSbyte (SRS2 = 0) corresponds to the even byte (A0 = 0) of host memory, and the MSbyte (SRS2 = 1) corresponds to the odd byte (A0 = 1) of the host memory; while in the 68xxx Mode, the MSbyte (SUDS = 0) corresponds to the even byte(A0 = 0) of the host memory, and the LSbyte (SLDS = 0) corresponds to the odd byte (A0 = 1) of the host memory. Figure 4–9 also shows the required byte sequence for the SIFDAT, SIFDAT/INC, and SIFADR registers. Accessing the even byte of one of these three registers will transfer the data of the even byte of the register but has no other effect. Accessing the odd byte of one of these three registers has the same effect as previously described for 16-bit word transfers to these registers in terms of transferring SIFDAT to the adapter memory, updating the SIFDAT register with the adapter data, incrementing the SIFADR/SIFADRX register, etc. Although byte transfers would not normally be used in 16-bit mode to load 16-bit registers, and although all registers except SIFDAT, SIFDAT/INC, and SIFADR can be byte addressed in either byte sequence, it is recommended to use the above sequence for all DIO registers if byte transfers are used to load all 16 bits. Regardless of the byte sequence used, or if only one byte of a register is addressed, the byte correspondence must be strictly followed.

Figure 4–9. DIO Byte Transfers in 8- or 16-Bit Modes



(a) 808x Byte Transfers (8-Bit of 16-Bit Bus)

(b) 68xxx Byte Transfers (16-Bit Bus)

* This order is required only for SIFDAT or SIFDAT/INC transfer or SIFADR write.

4.3.9 SIFVEC Interrupt Acknowledge Register

The SIFVEC (SIF vector) register contains the 8-bit interrupt vector, internally managed by the adapter from vector values specified at initialization time.The SIFVEC register cannot be accessed directly by a host DIO read or write cycle. The contents of SIFVEC are output onto the system data bus **only** during an interrupt acknowledge cycle. See sections 3.4.1.4 and 3.4.2.3 for more details.

4.3.10 SIFCMD and SIFSTS Registers

Low-level command and status information is passed via the SIFCMD (SIF command) and SIFSTS (SIF status) registers. The low-level information passed through SIFCMD and SIFSTS is used by the adapter software to facilitate the passing of high-level command and status information through the SIFDAT (SIF data) and SIFDAT/INC (SIF data with automatic increment of adapter address) registers.

Note:

The INTERRUPT register of the first-generation TMS380 is referred to as the SIFCMD and SIFSTS registers on the TMS380C16. The name has been changed to more accurately reflect its functions.

SIFCMD and SIFSTS are individual 8-bit registers that share the same word register location in a 16-bit interface mode but are accessed separately in an 8-bit mode. The attached processor transmits low-level commands to the adapter CPU by writing to the SIFCMD Register. The adapter CPU responds by interpreting the commands, and transmitting the appropriate command-completion status codes to the attached processor through the SIFSTS register. When the attached system sets the INTERRUPT_ADAPTER (adapter interrupt request) bit of SIFCMD, the SIF interrupts the adapter CPU.

The eight SIFCMD bits can be read, set, but not reset by the attached processor. Adapter CPU microcode resets the SIFCMD bits. The seven LSBs of SIFCMD contain a command code whose meaning is determined by adapter microcode rather than by hard-wired logic. By writing a one to the MSb of SIFCMD (the interrupt adapter flag), the attached processor activates the command code by causing an interrupt service request to be issued to the adapter CPU; the adapter CPU responds by reading and interpreting the command. After reading the code, the adapter CPU writes a zero to the interrupt adapter bit to indicate to the attached processor that the SIFCMD register is available for the next command. The attached processor may poll this flag to check whether the previous adapter interrupt has been serviced. The seven LSBs of SIFCMD can be read and reset, but not set, by the adapter.

The seven SIFSTS bits can be both written to and read from by the adapter CPU. The seven LSBs of SIFSTS contain a status code whose meaning is determined by microcode rather than by hardwired logic. When the adapter CPU writes a one to the MSb of SIFSTS, the SINTR/SIRQ pin is asserted (if the SINTEN bit of the SIFACL register is set to one), signaling an interrupt request to the attached processor to notify it that the new status code has been posted.

The attached processor responds to the interrupt by reading the status code and then writing a zero to the SYSTEM_INTERRUPT bit to indicate to the adapter CPU that the SIFSTS register is available for the next status code. The adapter CPU polls this flag to check whether the previous interrupt to the system has been serviced. Both the INTERRUPT_ADAPTER and SYSTEM_INTERRUPT bits are cleared at system reset. The seven LSBs of SIFSTS can be only read from and not written to by the attached processor.

4.3.10.1 Writing to the SIFCMD and SIFSTS

A direct I/O (DIO) write to the SIFCMD and SIFSTS registers transfers a 16-bit word, which is used to post interrupts within the adapter as well to reset the adapter-to-system interrupt level on the SINTR/SIRQ pin.

Figure 4–10 shows the bit assignments of the SIFCMD and SIFSTS registers when written by the attached system. Table 4–2 defines the functions of each bit.

Figure 4–10. SIFCMD and SIFSTS Register Write-Bit Assignments

Bit	0	INTERRUPT_ADAPTER	
	1	ADAPTER_RESET	
	2	SSB_CLEAR	
	3	EXECUTE	SIFCMD
	4	SCB_REQUEST	
	5	RECEIVE_CONTINUE/CANCEL	
	6	RECEIVE_VALID	
	7	TRANSMIT_VALID	
	8	SYSTEM_INTERRUPT	
	9	RESERVED	
	10	RESERVED	
	11	RESERVED	
	12	RESERVED	SIFSTS
	13	RESERVED	
	14	RESERVED	
LSB	15	RESERVED	

Note: X denotes don't care. Writing to these bits has no effect.

Table 4–2. SIFCMD and SIFSTS Register Write-Bit Functions

Bit	Definition
0	INTERRUPT_ADAPTER: Bit 0, when set to one, causes an internal adapter interrupt. This bit has no effect when set to zero. This bit is cleared by the adapter when the adapter is ready for another interrupt. The purpose of the interrupt is defined by the ADAPTER_RESET, SSB_CLEAR, EXECUTE, SCB_REQUEST, RECEIVE_CANCEL, RECEIVE_VALID, and TRANSMIT_VALID bits described below.
1	ADAPTER_RESET: Setting bit 1 to one forces an adapter reset if bits 2—7 are also set to one. Following an adapter reset, the initialization procedure outlined in Section 4.6 should be followed. This reset function is a software command, and certain conditions of hardware failure may prevent its execution. Setting this bit and the INTERRUPT_ADAPTER bit without bits 2—7 set, will cause the adapter to deinsert from the ring, if inserted. Once in this state, a soft reset, writing >FF to SIFCMD will restart the adapter.
2	SSB_CLEAR: This interrupt request is used by the system to acknowledge adapter-to-host interrupts and to notify the adapter that the System Status Block (SSB) is available for posting additional status information, even if the SSB was not used for that particular interrupt (for example, SCB_CLEAR interrupt).
3	EXECUTE: This interrupt is used to initiate an adapter command contained in the System Command Block (SCB). This block will be DMA-read and executed by the adapter.
4	SCB_REQUEST: This interrupt is used to request the adapter to interrupt the attached system when the SCB is available for another command. The adapter returns the SCB.CLEAR interrupt code.
5	RECEIVE_CONTINUE/CANCEL: The function of this bit depends on the setting of the LLC_ENABLE bit in the adapter initialization options discussed later in this chapter. If the LLC_ENABLE bit is not set, this bit is used to signal the adapter that buffers have been added to the receive list chain. If the LLC_ENABLE bit is set, this interrupt causes the adapter to discard remaining data in the frame being transferred to the attached system.
6	RECEIVE_VALID: The function of this bit depends on the setting of the LLC_ENABLE bit in the adapter initialization options discussed later in this chapter. If the LLC_ENABLE bit is not set, this bit is used to signal the adapter that an invalid list causing list processing suspension during receive has been validated. If the LLC_ENABLE bit is set, this interrupt signals the adapter that the condition causing list processing suspension (odd forward pointer or invalid list) during receive has been cleared.
7	TRANSMIT_VALID: This interrupt request signals the adapter that the condition causing list processing suspension during transmit has been cleared.
8	SYSTEM_INTERRUPT: Writing a zero to bit 8 resets the adapter-to-attached-system interrupt (that is, clears the SINTR/$\overline{\text{SIRQ}}$ line). Writing a one to this bit has no effect. See Section 4.12.1 to determine correct use of this bit.
9—15	RESERVED: These bits cannot be written by the attached system.

4.3.10.2 Reading from the SIFCMD and SIFSTS Registers

Direct I/O (DIO) read of the SIFCMD and SIFSTS registers transfers a 16-bit word, which is used to examine status of the adapter.

Figure 4–11 shows the bit assignments of the SIFCMD and SIFSTS registers when read by the attached system. Table 4–3 defines the function of each bit.

Figure 4–11. SIFCMD and SIFSTS Register Read-Bit Assignments

Bit	0	INTERRUPT_ADAPTER	SIFCMD
	1	ADAPTER_RESET	
	2	SSB_CLEAR	
	3	EXECUTE	
	4	SCB_REQUEST	
	5	RECEIVE_CONTINUE/CANCEL	
	6	RECEIVE_VALID	
	7	TRANSMIT_VALID	
	8	SYSTEM_INTERRUPT	SIFSTS
	9	INITIALIZE	
	10	TEST	
	11	ERROR	
	12	INTERRUPT_CODE 0/ERROR_CODE 0	
	13	INTERRUPT_CODE 1/ERROR_CODE 1	
	14	INTERRUPT_CODE 2/ERROR_CODE 2	
LSB	15	INTERRUPT_CODE 3/ERROR_CODE 3	

Note: Bits 12 through 15 are used to report bring-up diagnostic and initialization errors.

Table 4–3. SIFCMD and SIFSTS Register Read-Bit Functions

Bit	Definition
0	INTERRUPT_ADAPTER: Bit 0 reflects the current state of the system-to-adapter interrupt.
1	ADAPTER_RESET: Bit 1 reflects the current state of the ADAPTER_RESET bit.
2	SSB_CLEAR: Bit 2 reflects the current state of the SSB_CLEAR bit.
3	EXECUTE: Bit 3 reflects the current state of the EXECUTE bit.
4	SCB_REQUEST: Bit 4 reflects the current state of the SCB_REQUEST bit.
5	RECEIVE_CONTINUE/CANCEL: Bit 5 reflects the current state of the RECEIVE_CONTINUE/CANCEL bit.
6	RECEIVE_VALID: Bit 6 reflects the current state of the RECEIVE_VALID bit.
7	TRANSMIT_VALID: Bit 7 reflects the current state of the TRANSMIT_VALID bit.
8	SYSTEM_INTERRUPT: Bit 8 is set to one if the adapter-to-attached-system interrupt is valid. In systems not implementing hardware interrupt control, this bit may be polled under software control. The adapter cannot reset this bit to zero. This must be done by the attached system writing a zero to this bit location.
9	INITIALIZE: Bit 9 is set to one when the bring-up diagnostics have completed and the adapter is ready to start the initialization process. This bit is cleared to zero when the initialization process is complete.
10	TEST: Bit 10 is set to one by the bring-up diagnostics following an adapter reset. This bit is cleared when the bring-up diagnostics complete successfully.
11	ERROR: Bit 11 is set if the bring-up diagnostics detect an error or if there is an error during the initialization process. The error condition is specified in bits 12 —15. The error codes are detailed in sections 4.5 and 4.6.

Table 4–3. SIFCMD and SIFSTS Register Read-Bit Functions (Continued)

Bit	Definition
12—15	INTERRUPT_CODE: Bits 12—15 define the adapter-to-attached-system interrupt reason code. The interrupt code indicates if the SSB is used. The valid 4-bit interrupt codes are shown below: 0000 Adapter Check: This interrupt code is used when the adapter has encountered an unrecoverable hardware or software error. See adapter Check Interrupt description. 0100 Ring Status: This interrupt code is used if the SSB is updated with ring status. 0101 LLC Status: This interrupt code is used if the status of the adapter's LLC has been changed. The status code can be read from the Interface Control Block (ICB). This interrupt is generated if the LLC_ENABLE bit is set in the initialization options. 0110 SCB Clear: This interrupt code is used following an SCB.REQUEST interrupt when the SCB is clear. 0111 Timer: This interrupt is generated at regular intervals set by the attached system with the TIMER_SET command. This interrupt is generated only if the LLC_ENABLE bit is set in the initialization options. 1000 Command Status: This interrupt code is used when the SSB is updated with COMMAND_STATUS for commands other than TRANSMIT and RECEIVE. This includes COMMAND_ REJECT. 1010 Receive Status: This interrupt code is used if the SSB is updated with RCV_STATUS. 1100 Transmit Status: This interrupt code is used if the SSB is updated with XMIT_STATUS. 1110 Receive Pending: This interrupt code is used to indicate that a frame is ready to be transferred across the system interface, and that the ICB has been updated with destination information. This interrupt is generated only if the LLC_ENABLE bit is set in the initialization options.

Bits 9—15 are set or reset only by the adapter. The attached system cannot set or reset these bits; these bits can only be read by the attached system. See section 4.6 for a list of error codes.

4.4 Adapter Reset

The operation of the reset process is affected by the strapping pins, which determine the type of memory expansion implemented on the TMS380C16 adapter bus.

4.4.1 Hardware Reset

1) SRESET line of the system interface is asserted active low.
   - ❑ In DRAM-based adapters, when the $\overline{\text{SRESET}}$ line is asserted and then deasserted, the TMS380C16 CPU is placed in a halted state to allow the download of adapter software. When the CPHALT bit (bit 9) of the SIFACL register is cleared, the adapter software bring-up diagnostics are executed.
   - ❑ In EPROM-based adapters, when $\overline{\text{SRESET}}$ is asserted and then deasserted, the adapter resets and the adapter software bring-up diagnostics are executed.
2) A one is written to bit 8 (ARESET) of the adapter control register (SIFACL).
   - ❑ In DRAM-based adapters, when the ARESET bit of the SIFACL register is set, the adapter is placed in a reset state. Clearing ARESET and setting the CPHALT and BOOT bit in the same write causes the adapter to enter a halted state and wait for download of the adapter software. Refer to Table 4–4 for an example of hardware reset with a DRAM-based adapter.
   - ❑ In EPROM-based adapters, writing to the ARESET bit of the SIFACL register causes the adapter to reset. Clearing the ARESET, CPHALT, and BOOT bit in the same write causes the adapter to begin execution of the adapter software bring-up diagnostics.

Table 4–4. DRAM-Based Adapter Hardware Reset Example

DRAM Configuration	— BTSTRP and PRTYEN pins tied high or not connected — INPUTO pin pulled high
Hardware Reset	1) Set the ARESET bit in the SIFACL register to one (write >00EE) 2) Write a >006E to the SIFACL register to clear the reset and cause the DRAM-based adapter to be placed in a halted state. 3) Write address to SIFADX and SIFADR. 4) Write data to SIFDAT/INC to download software. 5) Once the software is downloaded, write a >002E to the SIFACL register to cause the adapter to execute the bring-up diagnostics.

Note: See Section 3.4.1.1.1 and Appendix A, System DIO Timing.

4.4.2 Software Reset

The adapter software must be resident to perform a software reset. Therefore, in DRAM-based adapters, the adapter software must already have been downloaded to perform a software reset.

Writing a one to bits 0 through 7 (>FF00) of the SIFCMD/SIFSTS register causes the adapter to software reset. Bring-up diagnostics are executed as part of this software reset.

The following list is the sequence of steps for a power-up reset:

1) Turn power on.
2) Wait two ms for V_{DD} to reach minimum high level and for the oscillator to start.
3) Hardware reset: $\overline{\text{SRESET}}$ asserted active low.
4) Wait five ms; then deassert $\overline{\text{SRESET}}$.
5) If BTSTRP is tied low (EPROM-based), then go to #17; if it is tied high (RAM-based), then reset: write(0x0080) to SIFACL register.
6) Wait 14 µs, the minimum duration of ARESET active high.
7) COPY = Read(SIFACL) ; get parity and CPU strappings.
8) If pseudo-DMA is needed, then COPY = (COPY | 0x0100); set pseudo-DMA.
9) COPY = (COPY & 0x0FF7F); clear the ARESET bit to clear the reset.
10) COPY = (COPY | 0x0040); halt adapter CPU to prevent the adapter CPU from obtaining the adapter bus and thus allowing download of the adapter software.
11) Write(COPY) to SIFACL register: this clears the reset, halts the adapter CPU, and enables parity and the pseudo-DMA logic, if required.
12) Wait 14 µs, minimum time for reset to clear after writing ARESET bit low to the SIFACL register.
13) Download adapter software through DIO interface.
14) COPY = Read(SIFACL).
15) COPY = (COPY & 0x0FFBF); clear the CPHALT bit and start execution of the Bring-Up-Diagnostics (BUD).
16) Write(COPY) to SIFACL register: execute BUD.
17) RETRY_CNTR = 3. The number of retries that should be performed.
18) LOOP_CNTR = 6. The maximum number of 0.5 s waits needed for BUD to complete.

```
19) While (LOOP_CNTR ! = 0) /* Timeout has not occurred */
    {
        Wait 0.5 s
        STATUS = (Read (SIFCMD/SIFSTS) & 0x00F0)
        LOOP_CNTR = LOOP_CNTR – 1;
        If (STATUS == 0x0040), then go to # 26
    }
20) Error Occurred!
    If RETRY_CNTR == 0 then
        Go to # 23: Hardware Error prevented successful completion of BUD
    else
        (Retry) : RETRY_CNTR = RETRY_CNTR –1.
21) Write(0x0FF00) to SIFCMD/SIFSTS register. Software reset.
22) Go to #18 to retry execution of BUD.
23) INTERRUPT_CODE = (Read(SIFCMD/SIFSTS) & 0x000F)
24) Hardware error detected! Check INTERRUPT CODE BITS (12—15) to determine the cause.
    Switch(INTERRUPT_CODE)
    {
        CASE   0  :  printf("Initial Test Error");
        CASE   1  :  printf("Adapter Software Checksum Error");
        CASE   2  :  printf("Adapter RAM Error");
        CASE   3  :  printf("Instruction Test Error");
        CASE   4  :  printf("Context/Interrupt Test Error");
        CASE   5  :  printf("Protocol Handler/RI Hardware Error");
        CASE   6  :  printf("System Interface Register Error");
    }
25) BUD Failed to complete successfully! STOP!
26) BUD completed successfully!!
27) If EPROM-based and pseudo-DMA is needed, then set PSDMA bit of the SIFACL register to enable pseudo-DMA.
28) Initialize adapter.
29) Open  adapter.
```

Note:

The notation used in the sequence above is standard C programming notation. For more information, refer to ***The C Programming Language*** by Kernighan and Ritchie.

4.5 Bring-up Diagnostics—BUD

These diagnostics are executed independently of the state of the system interface pins or the lobe media.

The bring-up diagnostics perform extensive testing of the TMS380C16 hardware and software. Initial tests confirm the existence and validity of executable code. Next, a checksum of adapter code is performed to determine its validity. Adapter RAM is also tested as further described in Section 4.7.1.

The adapter CPU is then self-tested: first the instructions are tested, and then the interrupts are tested. The protocol handler/ring interface is then thoroughly tested. The PH registers are tested first. Then, the CRC generation and checking, code violation detection, receive and transmit function, MAC frame reception, and buffer chaining are tested using wrap mode internal to the ring interface. Also in wrap mode, the monitor and priority state machines, and PH parity checkers are tested. Finally, the SIF registers on the adapter bus are tested.

Bring-Up Diagnostics (BUD) Verification

Before the adapter can be initialized for proper operation, the attached system must verify that the BUDs terminated normally. For verification, follow this procedure:

1) After adapter reset (see section 4.4), the attached system should wait 0.5 s and then read the SIFCMD/SIFSTS register until one of the following conditions has occurred:

   a) If the INITIALIZE bit is set to one, the TEST bit is zero, and the ERROR bit is zero, then the INTERRUPT CODE bits (12—15) should also be zero. This indicates that the BUDs completed successfully and that the adapter may now be initialized. The SIFCMD/SIFSTS register will contain the value >0040.

   b) If the TEST and the ERROR bits are set to one, the diagnostics have detected an unrecoverable hardware error. The bring-up error code may be read from bits 12—15. Table 4–5 lists the definitions of these error codes. The SIFCMD/SIFSTS register will contain the value >003x.

   c) If neither of the above conditions occurs within three seconds of reset, there is a hardware error preventing completion of the diagnostic routines. It is recommended that the attached system reset and retry this procedure three times. If the retries are still unsuccessful, an unrecoverable hardware error has occurred, and the adapter should be checked.

Table 4–5. Bring-Up Diagnostic Error Codes (SIFCMD/SIFSTS Register)

Error Code				Error Condition
12	13	14	15	
0	0	0	0	Initial test error
0	0	0	1	Adapter software checksum error
0	0	1	0	Adapter RAM error (First 128 Kbyte)
0	0	1	1	Instruction test error
0	1	0	0	Context/interrupt test error
0	1	0	1	Protocol handler hardware error
0	1	1	0	System interface register error

After verification of the BUDs, the attached system software may now continue with adapter initialization.

4.6 Adapter Initialization

After successful completion of the adapter's bring-up diagnostics, the system software must initialize the adapter. This initialization involves the transfer of parameters to the adapter via the DIO interface. These parameters specify

- The address in the system memory of the system command block (SCB) and system status block (SSB).
- Interrupt control parameters.

Before the completion of the initialization process, the adapter initiates a test of the DMA interface. These tests include

- DMA writes to both the SCB and SSB.
- DMA reads from both the SCB and SSB to compare expected data.

These tests may not require any attached system software to execute; however, in the event these tests fail, the adapter returns an error in the SIFSTS register. A working host system driver is typically needed for pseudo-DMA operation.

4.6.1 The Initialization Block

The initialization block is 22 bytes in length, and the entire block must be transferred to the adapter. Figure 4–12 defines the 22 bytes of this block:

Figure 4–12. Initialization Parameter Block

		High byte	Low byte
Word	0	INIT_OPTIONS	
	1	CMD_STATUS_VECTOR	TRANSMIT_VECTOR
	2	RECEIVE_VECTOR	RING_STATUS_VECTOR
	3	SCB_CLEAR_VECTOR	ADAPTER_CHK_VECTOR
	4	RECEIVE_BURST_SIZE	
	5	TRANSMIT_BURST_SIZE	
	6	DMA_ABORT_THRESHOLD	
	7	SCB_ADDRESS (High)	
	8	SCB_ADDRESS (Low)	
	9	SSB_ADDRESS (High)	
	10	SSB_ADDRESS (Low)	

Table 4–6. Initialization Parameter Block Field Descriptions

Field	Definition
INIT_OPTIONS	This 16-bit field is used to specify the desired initialization options. The bit assignments of the 16-bit initialization options field is shown below. Bit 0 RESERVED: This bit must be set to one. Bits1—2 PARITY_ENABLE: These bits should be set to one if the system bus provides odd parity on its data. If parity checking is not desired, these bits should be set to zero. If enabled, parity checking is performed on both DIO and DMA transfers between the adapter and attached system. Bit 3 BURST_SCB_SSB: If this bit is set to one, the adapter transfers the SCB from the system and the SSB to the system in DMA burst mode. The burst size is six bytes for the SCB read, two bytes for SCB clear, and eight bytes for SSB write. If this bit is set to zero, then these transfers occur in cycle steal mode. Bit 4 BURST_LIST: If this bit is set to one, the adapter transfers transmit and receive lists from the system in DMA burst mode. The burst size is 26 bytes. If this bit is set to zero, then cycle steal mode is selected. Bit 5 BURST_LIST_STATUS: If this bit is set to one, the adapter transfers list CSTAT and length information to the system in DMA burst mode. If this bit is set to zero, cycle steal mode is selected. Bit 6 BURST_RECEIVE_DATA: If this bit is set to one, the adapter transfers receive data to the system in DMA burst mode. The burst size is specified in the RECEIVE_BURST_SIZE field of the initialization block. If this bit is set to zero, cycle steal mode is selected. Bit 7 BURST_TRANSMIT_DATA: If this bit is set to one, the adapter transfers transmit data from the system in DMA burst mode. The burst size is specified in the TRANSMIT_BURST_SIZE field of the initialization block. If this bit is set to zero, cycle steal mode is selected. Bit 8 RESERVED: This bit must be reset to zero.

Table 4–6. Initialization Parameter Block Field Descriptions (Continued)

Field	Definition
INIT_OPTIONS (Cont.)	Bit 9 LLC_ENABLE: This bit must be set to one to enable the adapter's LLC interface. If this bit is reset to zero, then only the MAC interface is provided to the user. If this bit is set, then the LLC software must be present in the adapter. The LLC software may be downloaded into DRAMs or present in EPROMs on the adapter bus. Bits10—15 RESERVED: These bits must be reset to zero.
CMD_STATUS_VECTOR	This byte contains the interrupt vector that the adapter places on the attached system bus when the SSB is updated with command status for commands other than transmit or receive.
TRANSMIT_VECTOR	This byte contains the interrupt vector that the adapter places on the attached system bus when the SSB is updated with TRANSMIT_STATUS.
RECEIVE_VECTOR	This byte contains the interrupt vector that the adapter places on the attached system bus when the SSB is updated with RECEIVE_STATUS or when the ICB is updated with RECEIVE_PENDING data.
RING_STATUS_VECTOR	This byte contains the interrupt vector that the adapter places on the attached system bus when the SSB is updated with RING_STATUS, when the ICB is updated with LLC_STATUS data, or when a timer interrupt is indicated in the interrupt register.
SCB_CLEAR_VECTOR	This byte contains the interrupt vector that the adapter places on the attached system bus if an SCB_CLEAR interrupt is generated.
ADAPTER_CHK _VECTOR	This byte contains the interrupt vector that the adapter places on the attached system bus if an ADAPTER_CHECK interrupt is generated.
RECEIVE_BURST_SIZE	This 16-bit field contains a count of the maximum number of transfer cycles that the adapter will process during one DMA burst cycle when receive data is written to the attached system memory. If this field is zero, the adapter will set the burst size equal to the amount of data to be transferred. This parameter is ignored if the BURST_RECEIVE_DATA bit of the initialization options field is set to zero, indicating cycle steal mode. The parameter must be even.

Table 4–6. Initialization Parameter Block Field Descriptions (Continued)

Field	Definition
TRANSMIT_BURST_SIZE	This 16-bit field contains a count of the maximum number of transfer cycles that the adapter will process during one DMA burst cycle when transmit data is read from the attached system memory. If this field is zero, the adapter will set the burst size equal to the amount of data to be transferred. This parameter is NOT ignored if the BURST_TRANSMIT_DATA bit of the initialization options field is set to zero, indicating cycle steal mode. Even in cycle steal mode, TRANSMIT_BURST_SIZE is used to limit transmit data bus utilization so that higher priority receive-DMA operations can be initiated. This parameter must be even.
DMA_ABORT_THRESHOLD	This 16-bit field contains counts for the number of times the adapter will attempt a DMA operation that is terminated abnormally by a bus error or parity error. The high-order byte (bits 0—7) contains the count for bus errors, and the low-order byte (bits 8—15) contains the count for parity errors.
SCB_ADDRESS	This 32-bit field contains the 32-bit address of the SCB in attached system memory. This value must be an even address aligned on a word boundary.
SSB_ADDRESS	This 32-bit field contains the 32-bit address of the SSB in attached system memory. This value must be an even address aligned on a word boundary.

4.6.2 Writing the Initialization Block

The initialization block is passed to the adapter by following the procedure below:

1) Verify that the bring-up diagnostics completed successfully as described in section 4.5.

2) Write the value of >0001 to the SIFADX extended address register and >0A00 into the SIFADR register.

3) Begin transfer of the initialization block by writing each byte or 16-bit word to the SIFDAT/INC register. This causes the block to be written to successive adapter RAM locations beginning at RAM address >01.0A00.

4) Write the bit pattern >9080 to the SIFCMD/SIFSTS register. This sets the INTERRUPT_ADAPTER and EXECUTE bits, prevents resetting the SYSTEM_INTERRUPT bit, and clears all others.

5) Continue to read the SIFCMD/SIFSTS register until one of the following occurs:

   - ❑ The INITIALIZE, TEST, and ERROR bits are all zero. This condition indicates that initialization is complete without error. The SCB should contain >0000C1E2D48B in byte order, and the SSB should contain >FFFFD1D7C5D9C3D4 in byte order.

     Table 4–8 illustrates some of the internal adapter pointers accessible via the READ.ADAPTER or DIO commands. These pointers reside beginning at location >01.0A00 in adapter memory. These pointers must be read following initialization but before issuing the OPEN command.

   - ❑ If the ERROR bit is set, the initialization process has failed. The INTERRUPT_CODE (bits 12—15) contains the initialization error code. These error codes are listed in Table 4–7. The initialization procedure must be restarted from adapter reset.

   - ❑ If neither of the above conditions occurs within eleven seconds of loading the initialization parameter block, then it is recommended that the attached system reset the adapter and retry the initialization procedure three times. If still unsuccessful, then the adapter should be checked for errors.

Table 4–7. Adapter Initialization Errors (SIFCMD/SIFSTS Register)

Error Code Bits				Error Condition
12	13	14	15	
0	0	0	1	Invalid initialization block. 22 bytes were not passed.
0	0	1	0	Invalid options. This code is returned if the PARITY_ENABLE bits are not equal or if reserved bits are not zero in the INIT_OPTIONS. This code is also returned if the LLC_ENABLE bit is set and the LLC is not present.
0	0	1	1	Invalid receive burst size. RECEIVE_BURST_SIZE is odd.
0	1	0	0	Invalid transmit burst count. TRANSMIT_BURST_SIZE is odd.
0	1	0	1	Invalid DMA abort threshold. DMA_ABORT_THRESHOLD is zero.
0	1	1	0	Invalid SCB. SCB_ADDRESS is odd.
0	1	1	1	Invalid SSB. SSB_ADDRESS is odd.
1	0	0	0	DIO parity. A parity error occurred during a DIO write operation.
1	0	0	1	DMA timeout. The adapter timed out (11 seconds) waiting for a test DMA transfer to complete.
1	0	1	0	DMA parity error. A parity error occurred during the DMA tests, and the operation was tried as specified by DMA_ABORT_THRESHOLD.
1	0	1	1	DMA bus error. The DMA test encountered a bus error, and the operation was tried as specified by DMA_ABORT_THRESHOLD.
1	1	0	0	DMA data error. Initialize DMA test failed because of a data-compare error.
1	1	0	1	Adapter check. The adapter encountered an unrecoverable hardware error.

Table 4–8. Adapter Internal Pointers

Address	Description
>01.0A00	pointer to burned-in address.
>01.0A02	pointer to software level.
>01.0A04	pointer to adapter addresses: pointer + 0 node address. pointer + 6 group address. pointer + 10 functional address.
>01.0A06	pointer to adapter parameters: pointer + 0 physical drop number. pointer + 4 upstream neighbor address. pointer + 10 upstream physical drop number. pointer + 14 last ring poll address. pointer + 20 reserved. pointer + 22 transmit access priority. pointer + 24 source class authorization. pointer + 26 last attention code. pointer + 28 last source address. pointer + 34 last beacon type. pointer + 36 last major vector. pointer + 38 ring status. pointer + 40 soft error timer value. pointer + 42 ring interface error counter. pointer + 44 local ring number. pointer + 46 monitor error code. pointer + 48 beacon transmit type. pointer + 50 beacon receive type. pointer + 52 frame correlator save. pointer + 54 beaconing station UNA. pointer + 60 reserved. pointer + 64 beaconing station physical drop number.
>01.0A08	pointer to MAC buffer (a special buffer used by the software to transmit adapter generated MAC frames).
>01.0A0A	pointer to LLC counters: pointer + 0 MAX_SAPs. pointer + 1 open SAPs. pointer + 2 MAX_STATIONs. pointer + 3 open stations. pointer + 4 available stations. pointer + 5 reserved.
>01.0A0C	pointer to 4-/16-Mbps word flag. If zero, then 4 Mbps. If nonzero, then the adapter is set to run at 16-Mbps data rate.
>01.0A0E	pointer to total adapter RAM found in Kbytes in RAM allocation test.

4.7 Adapter Tests

4.7.1 RAM Tests

Adapter RAM availability determination and testing is performed dynamically by the adapter software in a two-stage process. The first stage occurs after reset when the adapter software performs the BUD tests. During the adapter RAM test portion of BUD tests, the adapter software tests the first 128K bytes of memory in the adapter. This memory includes memory used for instruction and data space by the adapter software itself if implemented in a RAM-based adapter. If the RAM tests are completed successfully, the remaining BUD tests will commence.

The second stage of adapter RAM availability determination and testing occurs as part of the background diagnostics once the BUD tests have completed. The background diagnostics are normally used to verify adapter software code integrity and the processor's instruction set. However, immediately following the BUD tests, the background diagnostics determine the availability of RAM beyond the first 128K bytes of memory. Upon determining the amount of RAM present on the adapter bus, the adapter software performs tests to verify the RAM and configure the memory into buffers to be added to the internal free pool dynamically. The RAM test consists of address verification, a checkerboard test, and a parity test.

Because the TMS380C16 has the capability to support up to two megabytes of memory, the testing time could take several seconds. Therefore, the adapter software could potentially be finding, testing, and configuring RAM while the adapter is being initialized and opened onto the network. The advantage of this procedure is that the RAM test and configuration time do not increase the amount of time a host system must wait for the BUD tests to complete if all the testing and configuration were to be performed during the BUD tests.

If a fatal error is discovered any time during the RAM tests, the adapter enters the adapter check state with a RAM test error.

4.7.2 BIA Tests

During initialization, the adapter copies the BIA to the RAM at pointer address >01.0A00 and performs a BIA checksum validation. If the BIA address is zero or the checksum is incorrect, the first word at pointer address >01.0A00 of the BIA address is set to >FFFF. This BIA copy can be read after initialization by reading the initialization pointers and using the DIO or READ.ADAPTER command to get the RAM copy of the BIA.

4.8 System Command Block—SCB

The attached system issues a command to the adapter by loading the request in the SCB and interrupting the adapter. The adapter then downloads the command (and any required parameters) through the system interface DMA channel. An interrupt can be used as an indication that the SCB is available for additional commands. The SCB is six bytes in length, and the adapter always DMA-reads six bytes. The SCB appears in host memory in byte order. The SCB format is shown in Figure 4–13.

Figure 4–13. System Command Block Format

			+0	+1
SCB Address	+	0	SCB_CMD	
	+	2	SCB_PARM_0	
	+	4	SCB_PARM_1	

The SCB_CMD field contains the 16-bit command code request to the adapter. The SCB_PARM_0 and SCB_PARM_1 fields are used to hold command-dependent parameters. Usually, SCB_PARM_0 and SCB_PARM_1 are used as a 32-bit address field containing a 32-bit pointer to a command parameter block. Some commands do not have additional parameters, and only the 16-bit command code must be written.

The attached system initiates an adapter command by following the sequence shown below:

1) The attached system writes the command request code into SCB_CMD and any necessary parameters to SCB_PARM_0 and SCB_PARM_1.

2) The attached system writes to the adapter's SIFCMD/SIFSTS register and sets the INTERRUPT_ADAPTER, EXECUTE, and SYSTEM_INTERRUPT bits to one. Bit 8 (SYSTEM_INTERRUPT) should be set to one so that incoming interrupts from the adapter are not accidentally reset.

This sequence causes an interrupt internal to the adapter. The adapter DMA-reads the SCB and, if required, the command parameter block. Once the SCB is downloaded, the adapter writes a zero to the SCB_CMD field. This can be used as an indication that the command has been recognized and the attached system can write to the SCB. If the SCB_REQUEST bit (bit 4) of the SIFCMD/SIFSTS register is set to one, an adapter-to-system interrupt is posted after the parameters are read into the adapter. In most cases, this interrupt occurs on command completion. The attached system is responsible for clearing this interrupt by writing a zero to the SYSTEM_INTERRUPT bit (bit 8) of the adapter SIFCMD/SIFSTS register.

It is recommended that SCB_REQUEST be set coincident with EXECUTE if SCB_CLEAR interrupt is desired.

There cannot be more than one transmit or receive command executed at one time. Multiple CONNECT.STATION, TRANSMIT.I.FRAME, or CLOSE.STATION commands can be executing. One other command can be executing while these commands are active. Thus, the adapter can support a TRANSMIT command, a RECEIVE command, multiple CLOSE.STATION commands, multiple TRANSMIT.I.FRAME commands, multiple CONNECT.STATION commands, and one other command, simultaneously. CONNECT.STATION, TRANSMIT.I.FRAME, and CLOSE.STATION commands can exist only if LLC code exists and if the LLC_ENABLE bit in the initialization options was set to one.

4.9 System Status Block—SSB

When the status of any command is to be returned to the attached system, the adapter DMA-writes this information to the SSB. An interrupt is then posted (if enabled). After the attached system has read the SSB, the adapter must be notified that the SSB is clear and available for additional status posting. This is done by writing a one to the INTERRUPT_ADAPTER (bit 0) and the SSB_CLEAR bit (bit 2) of the SIFCMD/SIFSTS register. The SSB is eight bytes in length and the adapter always DMA-writes the entire eight bytes, regardless of the actual length of the returned status. The SSB appears in host memory in byte order. The SSB format is defined in Figure 4–14.

Figure 4–14. System Status Block Format

			+0	+1
SSB Address	+	0	SSB_CMD	
	+	2	SSB_PARM_0	
	+	4	SSB_PARM_1	
	+	6	SSB_PARM_2	

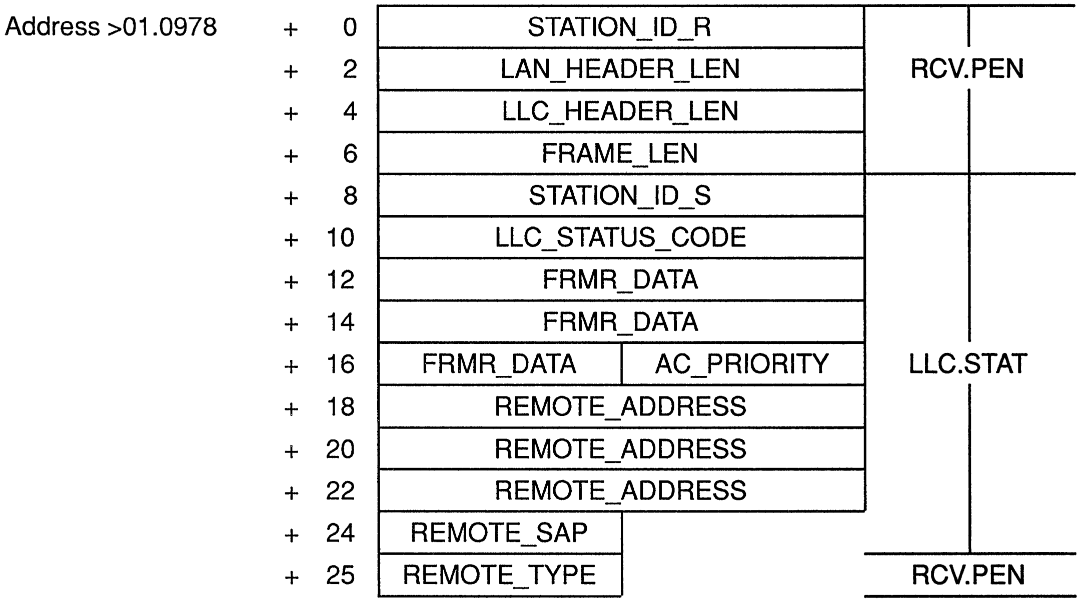
The SSB_CMD field contains a 16-bit code, which reflects the status type being returned. Valid status types that can be returned to the SSB are RING_STATUS, COMMAND_REJ_STATUS, COMMAND_STATUS, XMIT_STATUS, or RCV_STATUS. The command status field is written to SSB_PARM_0. The bit field definitions for the different status types can be found in the descriptions of the following interrupts and commands: RING.STATUS, COMMAND.STATUS, COMMAND.REJECT, TRANSMIT, and RECEIVE.

4.10 Interface Control Block—ICB

The ICB resides in adapter memory and is read by the attached system via DIO operations. This block contains LLC frame routing and LLC status information for use by the attached system. The ICB exists only if the LLC_ENABLE bit is set to one in the initialization options. Portions of this block are updated when RECEIVE.PENDING and LLC.STATUS interrupts are generated. The adapter updates the relevant portion of the ICB before either of these interrupts occurs. The format of the ICB is shown in Figure 4–15.

Access to the ICB is gained through use of the DIO registers. Load the SIFADX register with the value >0001 followed by loading the value >0978 into the SIFADR register.

Figure 4–15. Interface Control Block



Address >01.0978	+	0	STATION_ID_R		RCV.PEN
	+	2	LAN_HEADER_LEN		
	+	4	LLC_HEADER_LEN		
	+	6	FRAME_LEN		
	+	8	STATION_ID_S		LLC.STAT
	+	10	LLC_STATUS_CODE		
	+	12	FRMR_DATA		
	+	14	FRMR_DATA		
	+	16	FRMR_DATA	AC_PRIORITY	
	+	18	REMOTE_ADDRESS		
	+	20	REMOTE_ADDRESS		
	+	22	REMOTE_ADDRESS		
	+	24	REMOTE_SAP		
	+	25	REMOTE_TYPE		RCV.PEN

Table 4–9. Interface Control Block Field Descriptions

Field	Definition
STATION_ID_R	This field contains the station ID to which a received frame is destined. This field is updated with a RECEIVE.PENDING interrupt.
LAN_HEADER_LEN	This field contains the length of the MAC header in a received frame. The MAC header includes the AC, FC, DA, SA, and, if present, the routing information fields. This field is updated with a RECEIVE.PENDING interrupt.
LLC_HEADER_LEN	This field contains the length of the LLC header in a received frame. This field is updated with a RECEIVE.PENDING interrupt. The LLC header includes the DSAP, SSAP, and LLC control fields.
FRAME_LEN	This field contains the length of the entire received frame, including data, LAN header, and LLC header. This count does not include the frame's CRC. If the PASS_CRC option is set for the receive, the attached system must add four bytes to this value to find the frame length. This field is updated with a RECEIVE.PENDING interrupt.
STATION_ID_S	This field contains the station ID to which an LLC_STATUS_CODE belongs. This field is updated with a LLC.STATUS interrupt.
LLC_STATUS_CODE	The adapter uses this field to indicate the nature of the LLC status change. This field is updated with a LLC.STATUS interrupt. The bit definitions are as follows: Bit 0 LOST_LINK: When set to one, indicates that the link between the adapter and a remote link station has been lost. Bit 1 DISC: When set to one, indicates that a disconnected mode (DM) or disconnect (DISC) frame has been received, or that a previously sent DISC frame has been acknowledged by the remote station. Bit 2 FRMR_RCVD: When set to one, indicates that the adapter has received a Frame Reject (FRMR) frame. Bit 3 FRMR_SENT: When set to one, indicates that the adapter has sent an FRMR frame. Bit 4 SABME_RCVD: When set to one, indicates that the adapter has received a Set Asynchronous Balanced Mode Extended (SABME) frame for an open link station.

Table 4–9. Interface Control Block Field Descriptions (Continued)

Field	Definition	
LLC_STATUS_CODE (Cont.)	Bit 5	SABME_OPEN_LINK: When set to one, indicates that the adapter has received a SABME and has opened a new link station.
	Bit 6	REMOTE_BUSY: When set to one, indicates that the remote station on a link has entered a local busy state.
	Bit 7	REMOTE_NOT_BUSY: When set to one, indicates that the remote station on a link has exited a local busy state.
	Bit 8	TI_EXPIRED: When set to one, indicates that the adapter's inactivity timer (TI) has expired. This interrupt is generated only for a link in the disconnected state.
	Bit 9	LLC_CTR_OVERFLOW: When set to one, indicates that one of the adapter's LLC counters has reached half its maximum value.
	Bit 10	ACCESS_PRIORITY: When set to one, indicates that the access priority for a local SAP or link station has been reduced. This interrupt is generated when the user first attempts to transmit from the SAP after the access priority has been reduced.
	Bits 11—14	RESERVED: These bits are reserved and are set to zero.
	Bit 15	LOCAL_STATION_ENTERED_LOCAL_BUSY: This bit, when set, indicates that the local link station has entered a local busy state. This interrupt is not generated if the local busy state was entered because of a FLOW.CONTROL command from the attached system.
FRMR_DATA	This field contains the FRMR response contained in a transmitted or received FRMR frame. This field is updated with a LLC.STATUS interrupt.	
AC_PRIORITY	This field contains the new access priority for the SAP/Link station specified in STATION_ID_S. This field is updated with a LLC.STATUS interrupt.	
REMOTE_ADDRESS	This field contains the 6-byte node address of the remote station. This field is updated with a LLC.STATUS interrupt.	

Table 4–9. Interface Control Block Field Descriptions (Continued)

Field	Definition
REMOTE_SAP	This field contains the remote SAP connected to the link specified by the STATION_ID_S field. This field is updated with a LLC.STATUS interrupt.
REMOTE_SAP	This field contains the remote SAP connected to the link specified by the STATION_ID_S field. This field is updated with a LLC.STATUS interrupt.
FRAME_TYPE	This field indicates to the attached system what type of frame has been received. This field takes a value as shown in Table 4–10. A frame type of bridge data (20) indicates a frame that is not addressed to this adapter. That is, it was received because of an external address match. For more information on the other frame types, refer to the section on the TRANSMIT command. This field is updated with a RECEIVE.PENDING interrupt.

Table 4–10. Receive Frame Types

Value (hex)	Frame Type
02	MAC frame
04	I-frame
06	UI frame
08	XID cmd (poll = 1)
0A	XID cmd (poll = 0)
0C	XID resp (final = 1)
0E	XID resp (final = 0)
10	TEST resp (final = 1)
12	TEST resp (final = 0)
14	Other (non-MAC)
20	Bridge data

4.11 Frame Buffers

The attached system must allocate a portion of its memory for frame buffering. The amount of memory allocated varies according to the application. The information required to be supplied by the host system varies, depending on whether the TMS380C16 is supporting a MAC-only interface, or a MAC/LLC interface.

If a MAC-only interface is supported by the TMS380C16, the attached system logical frame format is as shown in Figure 4–16. Table 4–11 describes the frame fields of Figure 4–16. Table 4–12 summarizes the data that the attached system must provide to the adapter for various frame types.

Figure 4–16. Attached System Logical Frame Format

	H	ACCESS_CONTROL	FRAME_CONTROL	2 bytes
L	E	DESTINATION_ADDRESS		6 bytes
A	A	SOURCE_ADDRESS		6 bytes
N	D	ROUTING_FIELD (Optional)		2—30 bytes
	E	LLC_HEADER (Optional)		3—4 bytes
	R	Data		4K bytes at 4 Mbps (max)/ 18K bytes at 16 Mbps (max)

Table 4–11. Attached System Frame Fields

Field	Description	
ACCESS_CONTROL	Bits 0—2	Access priority: Bits 0—2 select the access prior ity for the frame. This value (0—7) must be less than or equal to the allowed access priority. The adapter's authorized access priority is 6 for MAC (3 for LLC) unless modified by network manage ment.
	Bits 3—7	RESERVED: These bits must be cleared (0) by the host system.
	Note:	If LLC is enabled, then for all LLC type 1 frames, the adapter fills in the correct value for the ACCCESS_CONTROL field, though the attached system must leave space for the AC in the MAC header.

Table 4–11. Attached System Frame Fields (Continued)

Field	Description
FRAME_CONTROL	This 8-bit field is defined in Chapter 2. If the MAC indicator bit is set to zero, the authorized function class parameter is used to validate the major vector source class. If LLC is enabled, then MAC frames must be sent as a direct frame. For all LLC type 1 frames, the adapter fills in the correct value for the FC byte, though the attached system must leave space for the FC in the MAC header.
DESTINATION _ADDRESS	This field is 48 bits wide and contains the address of the destination. The address format for this field is explained in Chapter 2. The address contained within this six-byte field must be organized in system memory with the highest order byte occupying the lowest host system byte address and the lowest order byte occupying the highest host system byte address.
SOURCE_ADDRESS	Unless the PASS_SOURCE_ADDRESS option is selected, the adapter stores the node address into the six bytes of the SOURCE_ADDRESS with the exception of byte 0 bit 0 (the routing information indicator). The node address is that address supplied by the BIA or passed during the OPEN command.
ROUTING _INFORMATION	This field must be included if bit 0 of SOURCE_ADDRESS is set to one. The format of the ROUTING_INFORMATION is described in Chapter 2.
LLC_HEADER	This field consists of the DSAP, SSAP, and LLC control fields as described in Chapter 2. If LLC is enabled, then this field is provided by the adapter for all type 1 and 2 LLC frames; space does not need to be provided for this header in attached system memory. This field is optionally provided by the attached system for direct frames.
DATA	The data portion of the frame is transmitted as specified by the attached system. The CRC, ending delimiter, and frame status (FS) are appended to the data by the adapter. Note that if the attached system selects the TRANSMIT_CRC option, then the four byte CRC must be included at the end of the DATA field by the host system. See the section on TRANSMIT for more information.

If the MAC/LLC software is installed, all frames, except LLC type 2 I-frames, are constructed according to the logical frame format defined in Table 4–11. I-frames consist of only the DATA field; the adapter is responsible for attaching the LAN header. Table 4–12 summarizes the data that the attached system must provide to the adapter for various frame types.

Table 4–12. Transmit Data

MAC-Only Interface

Frame Type	MAC Header Supplied By	LLC Header Supplied By	Data Field Supplied By	Notes
IEEE 802.5 MAC frame	Adapter	N/A	Adapter	1
Host-generated MAC frame	Attached system	N/A	Attached system	2
IEEE 802.2 LLC frames	Attached system	Attached system	Attached system	3

LLC Interface

Frame Type	MAC Header Supplied By	LLC Header Supplied By	Data Field Supplied By	Notes
IEEE 802.5 MAC frame	Adapter	N/A	Adapter	
Host-generated MAC frames	Attached system	N/A	Attached system	
Direct frame	Attached system	not used	Attached system	
UI frame	Attached system	Adapter	Attached system	4
XID command	Attached system	Adapter	Attached system or adapter	4, 5
XID response final	Attached system	Adapter	Attached system	4
XID response not final	Attached system	Adapter	Attached system	4
TEST command	Attached system	Adapter	Attached system (optional)	4
I-frame	Adapter	Adapter	Attached system	3, 6

Notes: 1) The TMS380C16 handles all IEEE 802.5 MAC frame operations.

2) The attached system (host) has the option to transmit MAC frames other than class 0 MAC frames.

3) When the MAC-only software is installed on the adapter, the attached system (host) must generate LLC frames.

4) The adapter constructs the LLC header information from the STATION_ID parameter and the REMOTE_SAP parameter in the transmit list.

5) If the XID_HANDLER bit in the OPEN.SAP command parameters list was set, the attached system must provide the XID data field. If this bit was not set, the adapter writes over the first three bytes of the data field provided by the attached system. The attached system must save these three bytes of space for the adapter.

6) I-frames may not be transmitted until the attached system has issued a CONNECT.STATION command. The adapter constructs the MAC and LLC headers from information provided in the OPEN.STATION and CONNECT.STATION commands.

4.12 Adapter-to-System Interrupts

4.12.1 Interrupt Handling

This section is intended to give the programmer information on how to handle interrupts from the adapter. After an adapter-to-attached-system interrupt, the adapter must be re-enabled in two ways for further interrupts. The SYSTEM_INTERRUPT bit (bit 8) of the SIFCMD/SIFSTS register should be cleared. This re-enables the system interface hardware for interrupts. Secondly, the adapter should be interrupted with the INTERRUPT_ADAPTER (bit 0) and the SSB_CLEAR bit (bit 2) of the SIFCMD/SIFSTS register set to one. Issuing the SSB_CLEAR interrupt to the adapter indicates to the adapter's controlling software that the attached system is ready for another interrupt. When the SYSTEM_INTERRUPT bit has been cleared but an SSB_CLEAR has not been issued, the adapter interrupts the system for DMA adapter check and SCB clear only. All other interrupts are held until the attached system issues an SSB_CLEAR interrupt.

When writing host software, consider the following guidelines for enabling adapter interrupts.

- ❑ Interrupts should be enabled by clearing the SYSTEM_INTERRUPT bit as soon as possible after entering the interrupt handling routine. Note that interrupt code must be read before clearing the interrupt.
- ❑ For those interrupts that use the SSB (RING.STATUS, TRANSMIT.STATUS, COMMMAND.REJECT, RECEIVE.STATUS, and COMMAND.STATUS), the SSB_CLEAR interrupt should not be issued until the contents of the SSB are read and stored.
- ❑ When the interrupt in progress uses the Interface Control Block (RECEIVE.PENDING, LLC.STATUS, or TIMER), the SSB_CLEAR interrupt should not be issued until the contents of the Interface Control Block (ICB) are read and stored. Otherwise, information may be lost.

4.12.2 RING.STATUS

The SSB is loaded with RING.STATUS and an interrupt posted to the attached system when any of the following conditions occur:

- The adapter detects a signal loss on the ring.
- The adapter is transmitting or receiving beacon frames to/from the ring. This interrupt to the host system may be disabled by setting bit 1 of the OPEN command options to one.
- The adapter transmits a report error MAC frame. This interrupt error condition can be disabled by setting bit 2 of the OPEN command option to one.
- An open or short circuit fault on the lobe is detected by the adapter.
- The adapter receives a remove MAC frame request.
- One of the adapter's error counters has incremented from 254 to 255.
- The adapter has been opened and is the only station on the ring.

Because of the dynamic nature of the report indications and the possibility that the ring status could change before the system can respond to a previous RING.STATUS interrupt, the current ring status could possibly equal the last ring status report. No RING.STATUS interrupts occur until the adapter has been opened. When any condition changes state, an interrupt is issued. The ring status field bits reflect the current state of respective status conditions; consequently, more than one bit may be set. A ring status interrupt vector is output as set in initialization.

	+0	+1
SSB_CMD	>0001	
SSB_PARM_0	RING_STATUS	
SSB_PARM_1	>XXXX	
SSB_PARM_2	>XXXX	

Figure 4–17. RING_STATUS Field Bit Assignments

MSb	Bit 0	SIGNAL_LOSS
	1	HARD_ERROR
	2	SOFT_ERROR
	3	TRANSMIT_BEACON
	4	LOBE_WIRE_FAULT
	5	AUTO_REMOVAL_ERROR
	6	RESERVED
	7	REMOVE_RECEIVED
	8	COUNTER_OVERFLOW
	9	SINGLE_STATION
	10	RING_RECOVERY
	11	0
	12	0
	13	0
	14	0
LSb	15	0

Table 4–13. RING_STATUS Field Bit Functions

Bit	Definition
0	SIGNAL_LOSS: When set to one, indicates that the adapter has detected a loss of signal on the ring.
1	HARD_ERROR: When this bit is set to one, it indicates that the adapter is presently transmitting or receiving beacon frames to or from the ring.
2	SOFT_ERROR: When this bit is set to one, it indicates that the adapter has transmitted a report error MAC frame.
3	TRANSMIT_BEACON: When this bit is set to one, it indicates that the adapter is transmitting beacon frames to the ring.
4	LOBE_WIRE_FAULT: When this bit is set to one, it indicates that the adapter has detected an open or short circuit in the cable between the adapter and the wiring concentrator. The adapter is closed and at the state following adapter initialization.
5	AUTO_REMOVAL_ERROR: When this bit is set to one, it indicates that the adapter has failed the lobe wrap test resulting from the beacon auto-removal process and has deinserted from the ring. The adapter is closed and remains at the state following adapter initialization.
6	RESERVED: This bit is undefined.
7	REMOVE_RECEIVED: When this bit is set to one, it indicates that the adapter has received a remove ring station MAC frame request and has deinserted from the ring. The adapter is closed and remains at the state following adapter initialization.
8	COUNTER_OVERFLOW: When this bit is set to one, it indicates that the one of the adapter's error counters has incremented from 254 to 255. The attached system should use the READ.ERROR.LOG command to determine which counter has overflowed.
9	SINGLE_STATION: When this bit is set to one, it indicates that the adapter has sensed that it is the only station on the ring. This bit is reset to zero when another station inserts into the ring.
10	RING_RECOVERY: This bit is set to one when the adapter observes claim token MAC frames on the ring. The adapter may be transmitting the claim token frames. This bit is reset when a ring purge frame is received or transmitted.
11—15	RESERVED: Are set to zero.

4.12.3 COMMAND.REJECT

The SSB is loaded with COMMAND_REJECT_STATUS and a command status interrupt is posted if an SCB is passed to the adapter with an error in SCB_CMD, SCB_PARM_0, or SCB_PARM_1. The adapter sets SSB_CMD to >0002 and SSB_PARM_1 with the SCB_CMD field of the offending SCB. The bit positions within the COMMAND_REJ_STATUS indicate the error that caused the adapter to interrupt, and an interrupt vector is output as set in initialization.

	+0	+1
SSB_CMD	>0002	
SSB_PARM_0	COMMAND_REJ_STATUS	
SSB_PARM_1	REJ_COMMAND	
SSB_PARM_2	>XXXX	

Figure 4–18. COMMAND_REJ_STATUS Field Bit Assignments

Bit 0	ILLEGAL_COMMAND
1	ADDRESS_ERROR
2	ADAPTER_OPEN
3	ADAPTER_CLOSED
4	SAME_COMMAND
5	0
6	0
7	0
8	0
9	0
10	0
11	0
12	0
13	0
14	0
LSB 15	0

Table 4–14. COMMAND_REJ_STATUS Field Bit Functions

Bit	Definition
0	ILLEGAL_COMMAND: This bit is set to one when an unknown command is issued to the adapter. This bit is set when commands that are valid only when the LLC interface is enabled are issued and the LLC interface is not enabled.
1	ADDRESS_ERROR: This bit is set to one if any address field in the SCB is odd (not word-aligned).
2	ADAPTER_OPEN: This bit is set to one if a command is issued when an adapter is open and the command is honored only when the adapter is closed.
3	ADAPTER_CLOSED: This bit is set to one if a command is issued when the adapter is closed and the command is honored only when the adapter is open.
4	SAME_COMMAND: This bit is set to one if a command is issued and the same command is already executing.
5—15	RESERVED: These bits are set to zero.

4.12.4 ADAPTER.CHECK

The ADAPTER.CHECK interrupt is generated when the adapter has encountered an unrecoverable hardware or software error. The SSB is not altered when the ADAPTER.CHECK Interrupt is generated. The adapter is in a closed state waiting to be reset, and an interrupt vector is output as set in initialization.

ADAPTER.CHECK information can be obtained by writing the value >0001 to the extended address register SIFADX, followed by writing the value >05E0 to the SIFADR register and then reading the ADAPTER_CHECK_BLOCK (8 bytes) through the SIFDAT/INC register.

Address >01.05E0	Byte +0	Byte +1
ADAPTER_CHECK_BLOCK + 0	ADAPTER_CHECK_STATUS	
+ 2	ADAPTER_CHECK_PARM_0	
+ 4	ADAPTER_CHECK_PARM_1	
+ 6	ADAPTER_CHECK_PARM_2	

Figure 4–19. ADAPTER_CHECK_STATUS Field Bit Assignments

Bit 0	DIO_PARITY
1	DMA_READ_ABORT
2	DMA_WRITE_ABORT
3	ILLEGAL_OP_CODE
4	PARITY_ERRORS
5	RESERVED
6	RESERVED
7	RESERVED
8	RAM_DATA_ERROR
9	RAM_PARITY_ERROR
10	RING_UNDERRUN
11	RESERVED
12	INVALID_INTERRUPT
13	INVALID_ERROR_INTERRUPT
14	INVALID_XOP
LSB 15	RESERVED

Table 4–15. ADAPTER_CHECK_STATUS Field Bit Definitions

Bit	Definition
0	DIO_PARITY: This bit is set to one if the adapter detects bad parity on data passed from the attached system to the adapter through a direct I/O access.
1	DMA_READ_ABORT: This bit is set to one if the adapter aborts a DMA-read operation (from the system). This can be a result of parity errors in excess of the parity abort threshold set during initialization, bus errors in excess of the bus error abort threshold also set during initialization, or if the adapter times out (11 seconds) waiting for the completion of a DMA bus operation (with or without an error). ADAPTER_CHECK_PARM_0 contains the following information: >0000 Indicates a timeout abort. >0001 Indicates a parity error abort. >0002 Indicates a bus error abort. ADAPTER_CHECK_PARM_1—2: Contains the host system address at failure. This address can be within ±12 bytes of the actual address.
2	DMA_WRITE_ABORT: This bit is set to one if the adapter aborts a DMA-write operation (to the system). This can be a result of parity errors in excess of the parity abort threshold set during initialization, bus errors in excess of the bus error abort threshold also set during initialization, or if the adapter times out (10 seconds) waiting for the completion of a DMA bus operation (with or without an error). ADAPTER_CHECK_PARM_0 contains the following information: >0000 Indicates a timeout abort. >0001 Indicates a parity error abort. >0002 Indicates a bus error abort. ADAPTER_CHECK_PARM_1—2: Contains the host system address at failure. This address can be within ±12 bytes of the actual address.
3	ILLEGAL_OP_CODE: This bit is set to one if the adapter's communications processor detects an illegal operation code in the adapter's internal program. ADAPTER_CHECK_PARM_0—2: Contains the communications processor registers R13, R14, and R15, respectively.
4	PARITY_ERRORS: This bit is set to one if the adapter detects a bus parity error on the adapter's internal bus. ADAPTER_CHECK_PARM_0—2 contains the communications processor registers R13, R14, and R15, respectively.
5—7	RESERVED. These bits are not set.
8	RAM_DATA_ERROR: This bit is valid only during RAM testing and indicates a RAM data error. ADAPTER_CHECK_PARM_0, 1 contains the MSW/LSW address of the RAM location.
9	RAM_PARITY_ERROR: This bit is valid only during RAM testing and indicates a RAM parity error. ADAPTER_CHECK_PARM_0, 1 contains the MSW/LSW address of the RAM location.

Table 4–15. ADAPTER_CHECK_STATUS Field Bit Definitions (Continued)

Bit	Definition
10	RING_UNDERRUN: This bit is set to one if the adapter detects an internal DMA underrun when transmitting onto the ring.
11	RESERVED.
12	INVALID_INTERRUPT: This bit is set to one if an unrecognized interrupt was generated internal to the adapter. ADAPTER_CHECK_PARM_0—2 contains communications processor registers R13, R14, and R15, respectively.
13	INVALID_ERROR_INTERRUPT: This bit is set to one if an unrecognized error interrupt was generated. ADAPTER_CHECK_PARM_0—2 contains adapter register R13, R14, and R15, respectively.
14	INVALID_XOP: This bit is set to one if an unrecognized XOP request was generated in the communications processor code. ADAPTER_CHECK_PARM_0—2 contains the communications processor registers R13, R14, and R15, respectively.
15	RESERVED: The value of this bit position is undefined.

4.13 Adapter MAC-Only Commands

This section describes the TMS380C16 command interface when only the IEEE 802.5 Medium Access Control (MAC) services are supported by the adapter software or when the LLC_ENABLE bit in the initialization options is set to zero. This interface is referred to as the MAC-Only I/F. Figure 4–20 lists the available commands and the corresponding command codes.

Figure 4–20. TMS380C16 COMMProcessor Command Set

Command	Code
OPEN	>0003
TRANSMIT	>0004
TRANSMIT.HALT	>0005
RECEIVE	>0006
CLOSE	>0007
SET.GROUP.ADDRESS	>0008
SET.FUNCTIONAL.ADDRESS	>0009
READ.ERROR.LOG	>000A
READ.ADAPTER	>000B
MODIFY.OPEN.PARAMETERS	>000D
RESTORE.OPEN.PARAMETERS	>000E
SET.FIRST.16.GROUP.ADDRESS	>000F
SET.BRIDGE.PARMS	>0010
CONFIG.BRIDGE.PARMS	>0011

Command OPEN

Description Before the adapter can be used for data communications, the attached system must first open the adapter by issuing an OPEN command. The OPEN command sets the adapter's addresses and enables the receipt and transmission of frames on the ring. A RECEIVE command should be the first command issued after a successful OPEN completion. During the OPEN process, the adapter suspends processing of all interrupt requests except reset.

The options that are set by this command can be modified by using the MODIFY.OPEN.PARAMETERS command or by closing the adapter via the CLOSE command and then reopening the adapter with the desired options.

Initiation

	Byte + 0	Byte + 1
SCB_CMD	>0003	
SCB_PARM_0	OPEN_PARM_BLOCK (High)	
SCB_PARM_1	OPEN_PARM_BLOCK (Low)	

Completion

SSB_CMD	>0003	
SSB_PARM_0	Status	Error

Initiation The SCB_PARM_0 and SCB_PARM_1 fields contain a 32-bit address that points to the OPEN parameter list, which is used by the adapter to configure itself for operation. The size of this block is dependent on whether the LLC interface was enabled during initialization of the adapter (LLC_ENABLE = 1). If LLC_ENABLE bit was set to zero, the adapter reads a 32-byte block of system memory via DMA; otherwise, the block is 44 bytes. Table 4–16 gives the OPEN parameter block field descriptions, and Table 4–17 gives a detailed description of each field.

Table 4–16. OPEN Parameter List

		Byte + 0	Byte + 1
OPEN_PARM_BLOCK	+ 0	OPEN_OPTIONS	
	+ 2	NODE_ADDRESS (High)	
	+ 4	NODE_ADDRESS	
	+ 6	NODE_ADDRESS (Low)	
	+ 8	GROUP_ADDRESS (High)	
	+ 10	GROUP_ADDRESS (Low)	
	+ 12	FUNCTIONAL_ADDRESS (High)	
	+ 14	FUNCTIONAL_ADDRESS (Low)	
	+ 16	RECEIVE_LIST_SIZE	
	+ 18	TRANSMIT_LIST_SIZE	
	+ 20	BUFFER_SIZE	
	+ 22	RESERVED	
	+ 24	RESERVED	
	+ 26	XMIT_BUF_MIN	XMIT_BUF_MAX
	+ 28	PRODUCT_ID_ADDRESS	
	+ 30	PRODUCT_ID_ADDRESS	

Table 4–17. OPEN Parameter Block Field Descriptions

Field	Definition	
OPEN_ OPTIONS	Bit 0	WRAP_INTERFACE: If this bit is set, ring insertion process is omitted, and all user-transmit data appears as user-receive data. The data is transmitted on the lobe from the attached product to the wiring concentrator. This option can be used for system interface debug, system interface DMA testing, or lobe media testing. MODIFY.OPEN.PARA-METERS cannot be used to alter the value of this bit; a CLOSE command must be issued to terminate WRAP mode.
	Bit 1	DISABLE_HARD_ERROR: If this bit is set, a RING.STATUS interrupt for HARD_ERROR and TRANSMIT_BEACON is not generated.
	Bit 2	DISABLE_SOFT_ERROR: If this bit is set, a RING.STATUS interrupt for SOFT_ERROR is not generated.
	Bit 3	PASS_ADAPTER_MAC_FRAMES: The value of this bit determines the action to be taken by the adapter when unsupported adapter MAC frames are received. If this bit is set to one, these MAC frames are passed to the attached system as normal receive data. If this bit is reset to zero, the adapter purges these frames and transmits a negative response MAC frame to the originating station.
	Bit 4	PASS_ATTENTION_MAC_FRAMES: If this bit is set to one, all attention MAC frames that are not equal to the last attention MAC frame received are passed to the system as receive data, following normal processing by the adapter.

Table 4–17. OPEN Parameter Block Field Descriptions (Continued)

Field	Definition	
OPEN_ OPTIONS (Cont.)	Bit 5	PAD_ROUTING_FIELD: If this bit is set to one, the adapter pads the routing field to 18 bytes. If no routing field is present in the received frame, the entire field is padded to 18 bytes. This option is voided if the current list's data count is not at least 32 bytes. In this case, the frame is transferred as if the bit were set to zero.
	Bit 6	FRAME_HOLD: If this bit is set to one, the adapter waits for an entire frame to be read from the ring before initiating the DMA transfer of the frame to the system. If this bit is reset to zero, then a DMA transfer is initiated whenever an adapter internal buffer is filled. This bit is forced to one if the LLC interface is enabled or if either bit 13 or 14 of the OPEN_OPTIONS is set to one.
	Bit 7	CONTENDER: If this bit is set to one, the adapter participates in the monitor contention process when another adapter detects the need for contention and initiates the process. This bit has no effect if this adapter detects the need for contention and initiates the monitor contention process.
	Bit 8	PASS_BEACON_MAC_FRAMES: If this bit is set, the adapter passes beacon MAC frames to the attached system after processing them. After passing the beacon MAC frame, the next beacon MAC frame is passed only if the source address or the beacon-type subvector changes.
	Bits 9—10	RESERVED: Must be reset to zero.
	Bit 11	EARLY_TOKEN_RELEASE: This bit is valid only in 16-Mbps operation. When this bit is set to zero, the TMS380C16 operates in the early token release mode. This bit has no effect in 4 Mbps operation.
	Bit 12	RESERVED: Must be set to zero.

Table 4–17. OPEN Parameter Block Field Descriptions (Continued)

Field	Definition	
OPEN_ OPTIONS (Cont.)	Bit 13	COPY_ALL_MAC_FRAMES: Setting this bit to one causes the adapter to copy and pass to the attached system all MAC frames that occur on the ring. Note that MAC frames are copied, whether or not they are addressed to the adapter. The following MAC frames are **not** passed to the attached system when they are addressed to this adapter: — Request station state — Request station address — Request station attachment — Transmit forward — Change parameters — Initialize ring station — Lobe media test These frames are passed, however, if they are destined for another adapter on the ring. Note that MAC frames addressed to the adapter are processed by the adapter before being passed to the attached system. Thus, frames may not be passed to the attached system in the same order in which they were received. Bit 3, bit 4 (PASS_ATTENTION_MAC _FRAMES), and bit 8 (PASS_BEACON_MAC_FRAMES) are forced to zero when bit 13 is set to one. Bit 3 (PASS_ADAPTER_MAC_FRAMES) functions in a normal manner. The Duplicate Address Test (DAT) MAC frame, the last frame of the lobe media check for insertion, is the first frame copied and passed to the attached system after the adapter is opened with this option and after the Receive command is issued. The lobe media test MAC frames are **not** copied by the adapter. **See Notes 1, 2, and 3.**
	Bit 14	COPY_ALL_NON_MAC_FRAMES: Setting this bit to one causes the adapter to pass all non-MAC frames that appear on the ring to the attached system. All non-MAC frames are copied, whether or not they are addressed to the adapter. **See Notes 1, 2, and 3.**

Table 4–17. OPEN Parameter Block Field Descriptions (Continued)

Field	Definition	
OPEN_ OPTIONS (Cont.)	Bit 15	PASS_FIRST_BUFFER_ONLY: Setting this bit to one causes the adapter to pass to the attached system only the first internal buffer of each received frame. The internal buffer size is specified by the BUFFER_SIZE field of the Open Parameter list. Regardless of the size of frame received, the adapter transfers one internal buffer to the attached system. If the frame received is smaller than the internal buffer size, the data from the last byte of the frame's information field to the end of the buffer is filled with indeterminate data. If the frame received is larger than the internal buffer size, only the first buffer is transferred. When this bit is set, the FRAME SIZE field of the Receive list, which starts a frame, always contains the internal buffer size. It will not contain the frame size. The 16-bit frame size value is written in the last two bytes of the buffer passed to the attached system. The frame is also marked with an elapsed time value in the third and fourth bytes from the end of the buffer transferred to the host system. This is the value of a 16-bit counter, which is started at adapter reset and incremented every 10 milliseconds. It thus rolls over to zero approximately every 655 seconds. This value is useful for measuring the relative time of appearance of frames on the ring. The PASS_FIRST_BUFFER_ONLY option is independent of the options taken in the rest of the bits in the open options field. Thus, this bit affects the manner in which receive frames are passed to the host, regardless of the reason the frames are passed to the attached system. This bit is forced off when the LLC Interface is enabled. **See Notes 1 and 3.**

Table 4–17. OPEN Parameter Block Field Descriptions (Continued)

Field	Definition
NODE_ADDRESS	This 6-byte field specifies the node address for the adapter. If this address is all zeros, the adapter uses the BIA read from the BIA PROM. Byte 0 and bit 0 must be set to zero.
GROUP_ADDRESS	This 32-bit field specifies the adapter group address and causes the adapter to receive messages that are sent to this address. GROUP_ADDRESS can be any value, but bit 0 is forced to one by the adapter. The two high-order bytes of the group address are set to >C000.
FUNCTIONAL_ADDRESS	This 32-bit field specifies the functional address and causes the adapter to receive messages that are sent to the functional address. Bits 0 and 31 are ignored by the adapter. The two high-order bytes of the functional address are set to >C000.
RECEIVE_LIST_SIZE	This 16-bit field indicates the number of bytes that must be read by the adapter when it obtains a receive list from the attached system. A decimal value of 0, 14, 20, or 26 is required. If zero, the default value of 26 is used.
TRANSMIT_LIST_SIZE	This 16-bit field indicates the number of bytes that must be read by the adapter when it obtains a transmit list from the attached system. A decimal value of 0, 14, 20, or 26 is required. If zero, the default value of 26 is used. If the LLC Interface is enabled, the adapter automatically adds four bytes to the value of the parameter for start-of-frame lists.
BUFFER_SIZE	The TMS380C16 adapter software rounds the BUFFER_SIZE parameter up to the nearest 1K-byte boundary (such as 1024, 2048, etc.) because internal adapter buffers must reside on 1K-byte boundaries.
RESERVED	These fields are reserved and ignored by the adapter.
TRANSMIT_BUFFER _MINIMUM_COUNT	This byte parameter contains the number of adapter buffers that are to be reserved as transmit buffers. These buffers are reserved for transmit only and are never used for receive. If zero is specified, no buffers are reserved for transmit. The minimum transmit buffer count must be less than or equal to the TRANSMIT_BUFFER_MAXIMUM_COUNT.

Table 4–17. OPEN Parameter Block Field Descriptions (Continued)

Field	Definition
TRANSMIT_BUFFER _MAXIMUM_COUNT	This byte parameter contains the maximum number of adapter buffers that are to be used for transmit. A minimum of two buffers must be available for receive. If this byte is zero, a default value of six is used. The product of TRANSMIT_BUFFER_MAXIMUM_COUNT and BUFFER_SIZE–8 determines the largest frame that the adapter can transmit.
PRODUCT_ID_ADDRESS	This 32-bit field contains a 32-bit address of the attached system product ID. Eighteen bytes are read, starting from the location specified during the OPEN command processing. After the OPEN command is complete, these 18 bytes in attached system memory can be released for other purposes. The bytes read are included in the product ID subvector of the attachment MAC frame. This frame is transmitted in response to the request station attachment MAC frame.

Notes: 1) The copy-all-frames feature is not included in the standard MAC or LLC software provided with the TMS380C16. This software is available from Texas Instruments upon special request. These bits must be set for MAC and LLC software.

2) Occasionally, when the adapter is opened with the copy-all-frames bits set, the OPEN command fails because of ring poll failure during the insertion process. This can occur when the ring receiving the insertion has heavy traffic, which causes extensive receive activity within the adapter during insertion process operations. If this open failure should occur, reset the adapter and try again. This problem can be avoided by opening the adapter without the copy-all-frames bits set and using the MODIFY.OPEN.PARAMETERS command (see MAC SET.GROUP.ADDRESS and MODIFY.OPEN.PARAMETERS commands) to set the desired copy options after the adapter has inserted into the ring.

3) The use of these bits is independent of XMATCH/XFAIL.

Completion Upon completion of the OPEN command, a COMMAND.STATUS interrupt is posted and the SSB updated with command completion information. The OPEN command code is loaded into SSB_CMD, and the OPEN_COMPLETION code is loaded into SSB_PARM_0. The bit definitions of the OPEN_COMPLETION field are listed in Table 4–18.

Table 4–18. OPEN_COMPLETION Bit Field Definitions

Bit	Definition
0	OPEN_SUCCESSFUL: Bit 0 is set to one if the OPEN command completed successfully. All other bits will be set to zero.
1	NODE_ADDRESS_ERROR: Bit 1 is set to one if an error was detected in the NODE_ADDRESS field of the open parameters, or if the BIA was read as all zeros, or if there is no BIA and the node address was all zeros.
2	LIST_SIZE_ERROR: Bit 2 is set to one if the RECEIVE_LIST_SIZE and/or the TRANSMIT_LIST_SIZE are not equal to 0, 14, 20, or 26.
3	BUFFER_SIZE_ERROR: Bit 3 is set to one if fewer than two buffers of the specified size can be created from available memory.
4	RESERVED.
5	TRANSMIT_BUFFER_COUNT_ERROR: Bit 5 is set to one if the number of receive buffers (total number of buffers minus the TRANSMIT_BUFFER_MAXIMUM_COUNT) is less than two.
6	OPEN_ERROR: Bit 6 is set to one if an error is detected during insertion onto the ring. Bits 8—15 in the SSB_PARM_0 field can be read to determine the cause of an error. Bits 8—15 are effectively divided into 4-bit entities. The first 4-bit field, open phase, is set to the OPEN command processing phase in which the error defined in the second 4-bit field occurred. The second 4-bit field, open error code, is set to an error code that reflects the ring-related error that occurred during OPEN command processing. The open phases and open error codes are listed in Table 4–19, and Table 4–20 describes the error codes.
7	INVALID_OPEN_OPTION: If bit 7 is set to one, the open options selected are not supported by the adapter.

Note:

Under heavy ring loading conditions, especially when traffic is broadcast address frames or frames addressed to the inserting adapter, the OPEN command may fail due to a ring poll failure during the insertion process. This failure occurs because of excessive receive loading on the inserting station. The OPEN command should be attempted again after a short delay. If the problen persists, the source of the receive traffic to the inserting adapter should be identified and corrected before another OPEN command is attempted.

Table 4–19. OPEN Phases and OPEN Error Codes

Bits				
8	**9**	**10**	**11**	**OPEN PHASES**
0	0	0	1	Lobe media test
0	0	1	0	Physical insertion
0	0	1	1	Address verification
0	1	0	0	Participation in ring poll
0	1	0	1	Request initialization
Bits				
12	**13**	**14**	**15**	**OPEN ERROR CODES**
0	0	0	1	Function failure
0	0	1	0	Signal loss
0	1	0	1	Timeout
0	1	1	0	Ring failure
0	1	1	1	Ring beaconing
1	0	0	0	Duplicate node address
1	0	0	1	Request initialization
1	0	1	0	Remove received

Table 4–20. OPEN Error Code Descriptions

Error Code	Definition
FUNCTION_FAILURE	This code is returned when the adapter is unable to transmit to itself while wrapped through its lobe at the wiring concentrator, or if any MAC frames are received before physical insertion.
SIGNAL_LOSS	This code is returned if a signal loss condition is detected at the adapter receiver input during the open process (when either wrapped or inserted onto the ring).
TIMEOUT	This code is returned if the adapter fails to logically insert onto the ring before the insertion timer expires. Each phase of the insertion process must complete before expiration of the 18-second insertion timer.
RING_FAILURE	This code is returned if the adapter times out when attempting a ring purge after becoming the active monitor: that is, when the adapter is unable to receive its own ring purge MAC frames.
RING_BEACONING	This code is returned if the adapter receives a beacon MAC frame after physically inserting into the ring.
DUPLICATE_NODE _ADDRESS	This code is returned if the adapter finds that another station on the ring already has the address that the adapter wishes to use.
REQUEST _PARAMETERS	This code is returned if the adapter determines that a Ring Parameter Server (RPS) is present on the ring but does not respond to a request initialization MAC frame. (If no RPS is present, the adapter will not return this code.)
REMOVE_RECEIVED	This code is returned if the adapter receives a remove adapter MAC frame during the insertion process.

Adapter Buffer Management

Frame data is transferred into the adapter local memory before transmission on the ring. Data is stored in the adapter local memory as a linked list of buffers. Because the chosen buffer size can affect overall adapter performance, the local buffer size is a user-programmable option through the OPEN parameters. The default adapter internal buffer size is 1K byte. The adapter's internal buffer format is shown in Figure 4–21. Note that the buffer size chosen for the adapter internal buffer is independent of the data buffers used in the attached system memory. The adapter automatically divides or combines internal memory buffers to form the minimum number of internal buffers required to represent a frame.

Figure 4–21. Adapter Internal Buffer Format

Buffer header 6 bytes	Forward Pointer	2 bytes
	Buffer status	2 bytes
	Data length	2 bytes
Frame Header (32 bytes max, First frame buffer only)	AC FC	2 bytes
	Destination address	6 bytes
	Source address	6 bytes
	Routing field (optional)	18 bytes (max)
Data field	Data	Variable
Received CRC	CRC	4 bytes
Buffer trailer	Reserved	2 bytes

Buffer Allocation

When the attached system requests a frame transmission, buffers are taken from the buffer pool one at a time until the frame has been transferred to the adapter. The user can specify a maximum number of these buffers to be used for transmission. The rest are dedicated receive buffers.

The TRANSMIT_BUFFER_MAXIMUM_COUNT of the OPEN parameters must leave at least two buffers available to receive frames. A maximum of two transmit frames can be processed simultaneously by the adapter. One will be enqueued for transmission while the other is transferred across the system interface, or both frames can be enqueued for transmission.

The maximum number of buffers that can be taken for frame transmission is specified in the TRANSMIT_BUFFER_MAXIMUM_COUNT of the OPEN parameters. If the system requests transmission of a single frame that causes the number of buffers required to transmit the frame to exceed the TRANSMIT_BUFFER_MAXIMUM_COUNT, the TRANSMIT command is terminated with LIST_ERROR.

To avoid the possibility of transmit congestion, causing transmit operations to suspend, it is highly recommended that a RECEIVE command and corresponding valid receive lists be issued as soon as possible following the OPEN command.

The limit on transmit frame size is specified by (BUFFER SIZE–8) x TRANSMIT_BUFFER_MAXIMUM_COUNT. The buffer size must be evenly divisible by 1K and a minimum of two buffers must be allocated.

Additional RAM can be added to the adapter to increase the number of buffers and/or increase the buffer size. The RAM expansion is determined automatically by the adapter during the adapter RAM test.

TRANSMIT Operation (MAC)

The TRANSMIT command is used to transmit frames to other nodes. These frames are passed from the attached system to the adapter using the logical format shown in Figure 4–22.

Figure 4–22. Attached System Frame Logical Format

AC	FC	2 bytes
Destination address		6 bytes
Source address		6 bytes
Routing field (optional)		18 bytes (max)
Data		

Table 4–21 describes the logical frame fields.

Table 4–21. Attached System Frame Fields

Field	Description
AC	This control field consists of the following bit functions: Bits 0—2 ACCESS PRIORITY. Bits 0—2 select the Access Priority for the frame. This value must be less than or equal to 6, or as modified by receipt of a lower Allowed Access Priority MAC parameter. Bits 3—7 RESERVED. These bits must be reset to zero.
FC	This 8-bit field is defined in detail in Section 2.4.2.2. If the MAC indicator bit is set to zero, the Enabled Function Class parameter is used to validate the Major Vector Source Class.
DESTINATION ADDRESS	This field is 48 bits wide and contains the address of the destination. The address format for this field is explained in Section 2.4.2.2. The address contained within this six-byte field must be organized in system memory with the highest order byte occupying the lowest system byte address and the lowest order byte occupying the highest system byte address.
SOURCE ADDRESS	This field is ignored with the exception of byte 0 bit 0 (the Routing Information Indicator) unless PASS_SOURCE_ADDRESS is set in bit 7 of transmit CSTAT request. The node address that is supplied by the Burned-in Address (BIA) or passed during the OPEN command will be used for the remaining bits of the source address of the frame.
ROUTING FIELD	The routing field must be included if bit 0 of the source address field is set to one.
DATA	The data portion is transmitted as specified by the attached system. The CRC, Ending Delimiter, and FS are appended to the data by the adapter unless pass CRC option is set, bit 6 of transmit CSTAT.

Note:

The only restrictions on transmitting MAC frames from the host is that a source class of ring station 0 is not permitted, and PCF attention of greater than 1 is prohibited.

MAC 0004 TRANSMIT Command

Command TRANSMIT

Description This TRANSMIT command is used to transmit frames to other nodes. A COMMAND.REJECT interrupt is posted if the adapter has not been opened, if there is already an executing TRANSMIT command, or if the address passed in the SCB_PARM_0 and SCB_PARM_1 is not aligned on a word boundary.

Initiation

	Byte + 0	Byte + 1
SCB_CMD	>0004	
SCB_PARM_0	XMIT_PARM_BLOCK (High)	
SCB_PARM_1	XMIT_PARM_BLOCK (Low)	

Completion

SSB_CMD	>0004
SSB_PARM_0	XMIT_STATUS
SSB_PARM_1	XMIT_PARM_LIST (High)
SSB_PARM_2	XMIT_PARM_LIST (Low)

Initiation The SCB_PARM_0 and SCB_PARM_1 contain a 32-bit word-aligned address that points to a transmit parameter list. The format of the transmit parameter list is shown in Figure 4–23, and the definitions of each field are given in Table 4–22. Transmit parameter lists must be aligned on even word boundaries. Note that one transmit parameter list cannot be used to transmit more than one frame, but several transmit parameter lists can be used to transmit a single frame.

Figure 4–23. TRANSMIT Parameter List

		Byte + 0	Byte + 1
XMIT_PARM_LIST	+ 0	FORWARD_POINTER (High)	
	+ 2	FORWARD_POINTER (Low)	
	+ 4	TRANSMIT_CSTAT	
	+ 6	FRAME_SIZE	
	+ 8	DATA_COUNT	
	+ 10	DATA_ADDRESS (High)	
	+ 12	DATA_ADDRESS (Low)	
	+ 14	DATA_COUNT	
	+ 16	DATA_ADDRESS (High)	
	+ 18	DATA_ADDRESS (Low)	
	+ 20	DATA_COUNT	
	+ 22	DATA_ADDRESS (High)	
	+ 24	DATA_ADDRESS (Low)	

Table 4–22. TRANSMIT Parameter List Fields

Field	Definition
FORWARD_POINTER	This 32-bit field contains a 32-bit pointer to the next transmit parameter list in the chain. When the pointer is odd, the current transmit parameter list is the last in the chain. The adapter continues to process transmit parameter lists until it reads an odd FORWARD_POINTER, at which time the adapter waits for the last frame (list with odd address) to be transmitted before exiting the command. If the system updates the FORWARD_POINTER before the transmission of the last frame is completed, the adapter continues to process transmit parameter lists. The system must update the FORWARD_POINTER from the most significant byte to the least significant byte to ensure that the address is valid before changing to an even address. The FORWARD_POINTER should not be initialized to point to itself; this could cause problems because of the pipelined nature of the adapter's list processing. The adapter does not alter this parameter.
TRANSMIT_CSTAT	This 16-bit parameter is set by the attached system when the transmit parameter list is created. It is overwritten by the adapter to report frame completion status. When initially set by the attached system, this parameter field is referred to as the TRANSMIT_CSTAT_REQUEST field. After a frame completes transmission, the adapter writes TRANSMIT_CSTAT_COMPLETE to the list that has START_FRAME set to one. These bits indicate the completion status of the frame, not the TRANSMIT command itself. **Note that the system must update the least significant byte prior to the most significant byte to ensure the frame type is set before the valid bit is set.**

Table 4–22. TRANSMIT Parameter List Fields (Continued)

Field	Definition	
TRANSMIT_ CSTAT_ REQUEST	Bit 0	VALID: The adapter waits for bit 0 to be set to a one before processing the current transmit parameter list. The attached system must issue a TRANSMIT.VALID interrupt request when changing bit 0 from zero to one. This bit is ignored unless the list is an anticipated start of frame (follows the end of frame or is first list of command).
	Bit 1	FRAME_COMPLETE: Must be reset to zero.
	Bit 2	START_FRAME: Must be set to one for a list that defines the start of a frame.
	Bit 3	END_FRAME: Must be set to one for a list that defines the end of a frame.
	Bit 4	FRAME_INTERRUPT: When this bit is set to one, an adapter-to-attached-system interrupt is generated when the frame has been transmitted, rather than after all frames on the chain have been transmitted. This bit is ignored unless START_FRAME (bit 2) is a one.
	Bit 5	RESERVED: This bit must be reset to zero.
	Bit 6	PASS_CRC: When this bit is set to one, the adapter assumes that the CRC to be transmitted with the frame is contained in the last four bytes of the frame data. In this case, the adapter does not generate the CRC on transmit but uses the CRC passed with frame data. Note that the CRC value is not checked. This option can be used by adapters, such as MAC-layer bridges, that must forward frames without altering the CRC. This bit is ignored unless the START_FRAME bit (bit 2) is set to one.
	Bit 7	PASS_SOURCE_ADDRESS: When this bit is set to one, the adapter transmits the frame with the source address as given by the host. The adapter does not overwrite the source address with the adapter node address. This option can be used by adapters, such as MAC-layer bridges, that must forward frames without altering the source address. This bit is ignored unless the START_FRAME bit (bit 2) is set to one.
	Bits 8—15	RESERVED: These bits must be reset to zero.

Table 4–22. TRANSMIT Parameter List Fields (Continued)

Field	Definition
TRANSMIT_ CSTAT_ COMPLETE	The completion code for the transmitted frame is written to this field in the list that has START_FRAME bit equal to one when the adapter has completed transmission of the frame. CSTATs that are not in a list defining the start of a frame are not altered by the adapter. The TRANSMIT_CSTAT_COMPLETE bit definitions are shown below. Bit 0 VALID: Reset to zero. Bit 1 COMPLETE: Set to one. Bit 2 FRAME_START: Bit 2 has the same value as specified by the attached system in TRANSMIT_CSTAT_REQUEST. Bit 3 FRAME_END: Bit 3 has the same value as specified by the attached system in TRANSMIT_CSTAT_REQUEST. Bit 4 INTERRUPT_FRAME: Bit 4 has the same value as specified by the attached system in TRANSMIT_CSTAT_REQUEST. Bit 5 TRANSMIT_ERROR: Bit 5 is set to one if the frame transmit or strip process was in error. Bit 6 PASS_CRC: This bit reflects the state of bit 6 in the TRANSMIT_CSTAT_REQUEST. This bit is valid only if the START_FRAME bit (bit 2) is set to one. Bit 7 PASS_SOURCE_ADDRESS: This bit reflects the state of bit 7 in the TRANSMIT_CSTAT_RQUEST. This bit is valid only if the START_ FRAME bit (bit 2) is set to one. Bits 8—15 STRIP_FS: If TRANSMIT_ERROR (bit 5) is set to zero, then this field contains a copy of the FS byte returned when the transmitted frame is stripped off the ring. If TRANSMIT_ERROR is set to one, this field should be ignored.

Table 4–22. TRANSMIT Parameter List Fields (Continued)

Field	Definition
FRAME_SIZE	This 16-bit field contains the number of bytes to be transmitted as a frame. The FRAME_SIZE value includes AC/FC, DESTINATION and SOURCE ADDRESS, the Routing Field, and the Data Field. FRAME_SIZE does not include CRC, FS, or EDEL. This parameter is valid only for the transmit list that has the FRAME START bit set; however, FRAME_SIZE must **not** be zero in any list. The adapter does not alter this parameter. A frame size of zero is not valid. The maximum frame size that can be transmitted is (BUFFER SIZE – 8) times TRANSMIT_BUFFER_MAXIMUM_COUNT.
DATA_COUNT	This 16-bit field contains the number of bytes to be read at the address defined by DATA_ADDRESS. There can be a maximum of three DATA_COUNT/DATA_ADDRESS parameters per transmit parameter list. If bit 0 is zero, then this DATA_COUNT is the last DATA_COUNT in the transmit parameter list. Bit 0 of the third DATA_COUNT is ignored. A DATA_COUNT of 0 is permitted (with or without bit 0 set). The sum of the used DATA_COUNT parameters must equal the FRAME_SIZE specified in the list that has START_FRAME set to one. The DATA_COUNT can be even or odd. The adapter does not alter this parameter.
DATA_ADDRESS	This 32-bit field contains a 32-bit pointer to a portion of (or the entire) logical frame residing in attached system memory. DATA_ADDRESS may be even or odd.

TRANSMIT CSTAT REQUEST Fields

Bit #	0	1	2	3	4	5	6	7
MSB	Valid	Frame Complete = 0	Start of Frame	End of Frame	Frame Interrupt	0	Pass CRC	Pass Source Address

Bit #	8	9	10	11	12	13	14	15
LSB	0	0	0	0	0	0	0	0

TRANSMIT CSTAT COMPLETE Fields

Bit #	0	1	2	3	4	5	6	7
MSB	Valid = 0	Frame Complete	Start of Frame	End of Frame	Frame Interrupt	Transmit Error	Pass CRC	Pass Source Address

Bit #	8	9	10	11	12	13	14	15
LSB	Strip FS bit 0	Strip FS bit 1	Strip FS bit 2	Strip FS bit 3	Strip FS bit 4	Strip FS bit 5	Strip FS bit 6	Strip FS bit 7

Execution The attached system can create a circular chain of transmit parameter lists by setting the FORWARD_POINTER of the last transmit parameter list to the address of the first list. The valid bit of TRANSMIT_CSTAT_REQUEST is manipulated to initiate and suspend frame transmission. When the adapter reads a list with the FRAME_START bit set to one and the valid bit reset to zero, it suspends processing until a TRANSMIT.VALID interrupt request is issued by the attached system. The attached system is not notified of a transmit suspended condition by the adapter.

If a fixed transmit chain technique is utilized and more than one list is used to transmit a single frame, lists that do not have the FRAME_START bit set should have the valid bit reset to zero, because the adapter does not alter the TRANSMIT_CSTAT field for lists that do not have the FRAME_START bit set. Revalidating the start of frame list also releases the remaining frame lists if the valid bits were initially set.

A circular chain of one or two frames should not be used, because of the pipelined nature of transmit parameter list processing. An implementation of this nature can cause the adapter to send the same frame twice. A circular chain should contain enough lists to hold two entire frames plus an extra list.

Since transmit lists may be added dynamically to the transmit parameter list chain, a test should be made, following a TRANSMIT.STATUS interrupt, to determine if the adapter has processed all frames that the attached system has placed in the chain. If frames have been added to the chain subsequent to the TRANSMIT command completion, another TRANSMIT command should be executed with the SCB_PARM_0 and SCB_PARM_1 set to the address of the first valid list.

When the last list in a chain contains an odd FORWARD POINTER, the adapter performs the following operations in order:

1) Updates the CSTAT field of the last start of frame list.
2) Posts the FRAME COMPLETE status if requested.
3) Rereads the FORWARD POINTER field of the list containing the odd FORWARD POINTER.
4) Posts a COMMAND COMPLETE status if FORWARD POINTER is still odd.

Because the adapter rereads the FORWARD POINTER after posting the FRAME COMPLETE status, the host software should not modify the FORWARD POINTER of the last list except to add another transmit list to the chain. Therefore, this list containing the odd FORWARD POINTER should not be considered free until one of the following occurs:

- ❑ A COMMAND COMPLETE status is received for this list, or
- ❑ Any transmit status (such as FRAME COMPLETE, COMMAND COMPLETE, or LIST ERROR) occurs, specifying a list further down the chain.

Completion A TRANSMIT.STATUS interrupt is generated when one of the following conditions occurs during processing of the TRANSMIT command:

- ❑ All the frames specified by the transmit parameter list chain have been transmitted,
- ❑ A TRANSMIT.HALT command has been issued and completed,
- ❑ A frame has been transmitted that had FRAME INTERRUPT set in CSTAT, or
- ❑ A fatal list error is detected.

Prior to issuing the TRANSMIT.STATUS interrupt, the adapter updates SSB_PARM_0 with the XMIT.STATUS completion code and the SSB_PARM_1and SSB_PARM_2 with the address of the last transmit parameter list processed by the adapter. The bit definitions of XMIT_STATUS are listed in Table 4–23.

Table 4–23. XMIT_STATUS Bit Definitions

Bit	Definition
0	COMMAND_COMPLETE: This bit is set to one to indicate that the TRANSMIT command has completed. The system must issue another TRANSMIT command to transmit additional frames. SSB_PARM_1 and SSB _PARM_2 contain a pointer to the transmit parameter list of the last frame transmitted. This bit is also set as a result of a TRANSMIT.HALT command. If a TRANSMIT.HALT command is issued, and no frames have been transmitted, SSB_PARM_1 and SSB_PARM_2 are cleared by the adapter. The COMMAND_COMPLETE and FRAME_COMPLETE bits are not set at the same time.
1	FRAME_COMPLETE: When set to one, this bit indicates that a frame has been transmitted and the INTERRUPT_FRAME bit was set in TRANSMIT_CSTAT_REQUEST. Since frames in the transmit parameter list chain may be transmitted faster than the system can respond to the interrupts and/or faster than the adapter can cause the interrupts, this bit can report the completion of more than one frame at a time. SSB_PARM_1 and SSB_PARM_2 contain a pointer to the transmit parameter list of the last frame transmitted. If lists with the FRAME_INTERRUPT bit set are intermixed with lists that do not have the FRAME_INTERRUPT bit set, FRAME_COMPLETE can include frames that did not have FRAME_INTERRUPT set. The attached system should check the TRANSMIT_ERROR bit (bit 5) of the CSTAT in the returned list for possible non-fatal errors.
2	LIST_ERROR: Bit 2 is set to one if there is an error in one of the lists that compose the frame. Bits 8—13 define the error. The TRANSMIT command is terminated, and the system must issue another TRANSMIT command to continue transmission. SSB_PARM_1 and SSB_PARM_2 contain a pointer to the list that starts the frame with errors. The TRANSMIT.STATUS interrupt for LIST_ERROR is not generated until the status of every other transmit has been posted. The CSTATs of lists found to be in error are not altered by the adapter. Neither FRAME_COMPLETE nor COMMAND_COMPLETE bit is set with LIST_ERROR.
3—7	RESERVED: Bits 3—7 are reset to zero.
8	FRAME_SIZE_ERROR: Bit 8 is set to one if FRAME_SIZE does not equal the sum of the valid DATA COUNT fields, if FRAME_SIZE is less than the required header plus one byte of information field (15 bytes plus routing field), or if FRAME_SIZE was specified as zero in any list.
9	TRANSMIT_THRESHOLD: Bit 9 is set to one if FRAME_SIZE is greater than the product of (BUFFER_SIZE – 8) times the TRANSMIT_BUFFER_MAXIMUM_COUNT parameters from the OPEN command.

Table 4–23. XMIT_STATUS Bit Definitions (Continued)

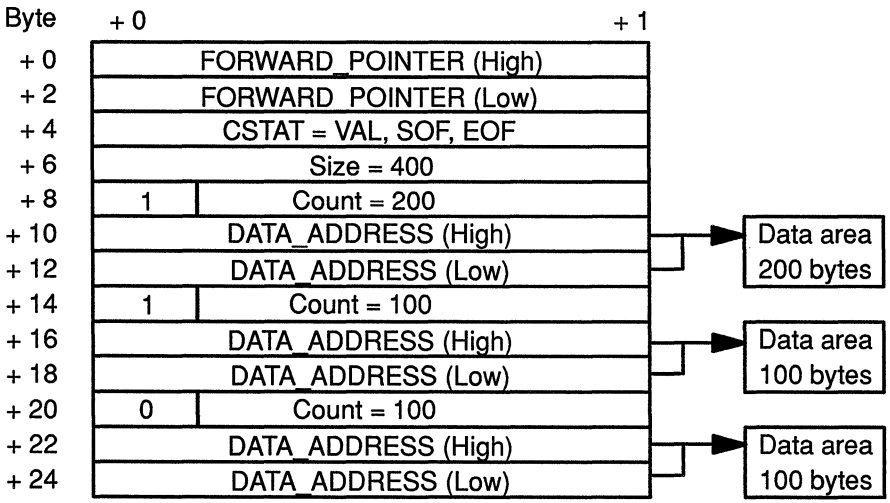
Bit	Definition
10	ODD_ADDRESS: Bit 10 is set to one if an odd FORWARD_POINTER value is read on a list that does not have END_FRAME set to 1.
11	FRAME_ERROR: Bit 11 is set to one if the START_FRAME bit is set to one on a list that is not an anticipated start of frame or if the START_FRAME bit is zero on an anticipated start of frame.
12	ACCESS_PRIORITY_ERROR: Bit 2 is set to one if the Access Priority requested has not been allowed.
13	UNENABLED_MAC_FRAME: Bit 13 is set to one if the MAC frame has a source class of zero, or if the MAC frame PCF ATTN field is greater than one.
14	ILLEGAL_FRAME_FORMAT: Bit 14 is set to one if bit 0 of the FC field was set to one.
15	RESERVED: Reset to zero.

TRANSMIT List Examples

Four examples (Figure 4–24 to Figure 4–27) of list formats are shown. The first three result in the transmission of a single 400-byte frame. Figure 4–26 is configured so that the attached system buffer space is appended to a 14-byte list.

The fourth example (Figure 4–27) illustrates two lists chained together to form the transmit chain for transmission of two frames.

Figure 4–24. TRANSMIT List Format: Example 1



Byte	+0	+1
+0	FORWARD_POINTER (High)	
+2	FORWARD_POINTER (Low)	
+4	CSTAT = VAL, SOF, EOF	
+6	Size = 400	
+8	1	Count = 200
+10	DATA_ADDRESS (High)	
+12	DATA_ADDRESS (Low)	
+14	1	Count = 100
+16	DATA_ADDRESS (High)	
+18	DATA_ADDRESS (Low)	
+20	0	Count = 100
+22	DATA_ADDRESS (High)	
+24	DATA_ADDRESS (Low)	

Figure 4–25. TRANSMIT List Format: Example 2

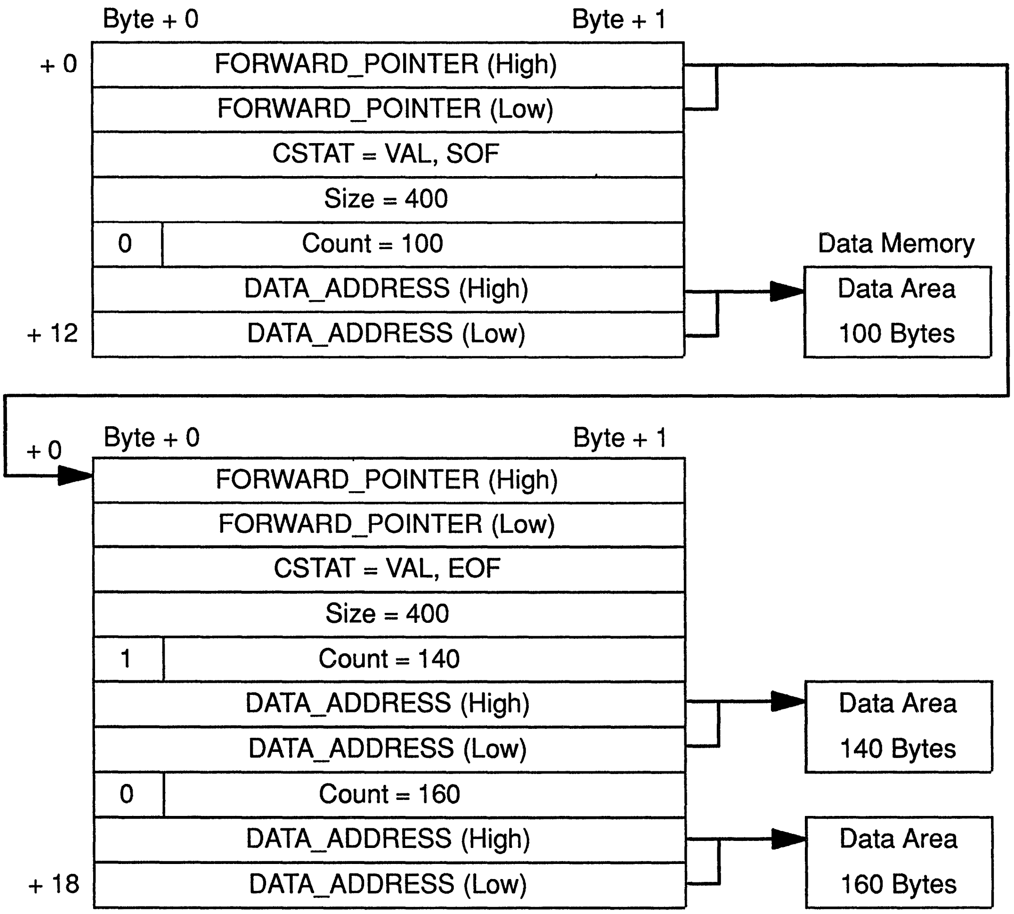


Figure 4–26. TRANSMIT List Format: Example 3

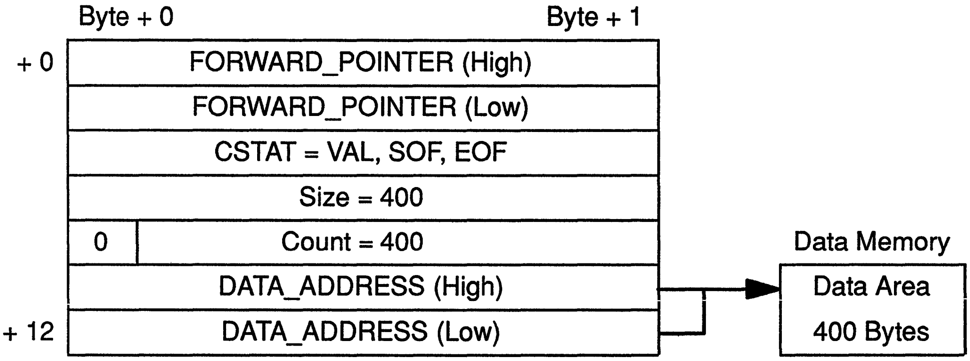
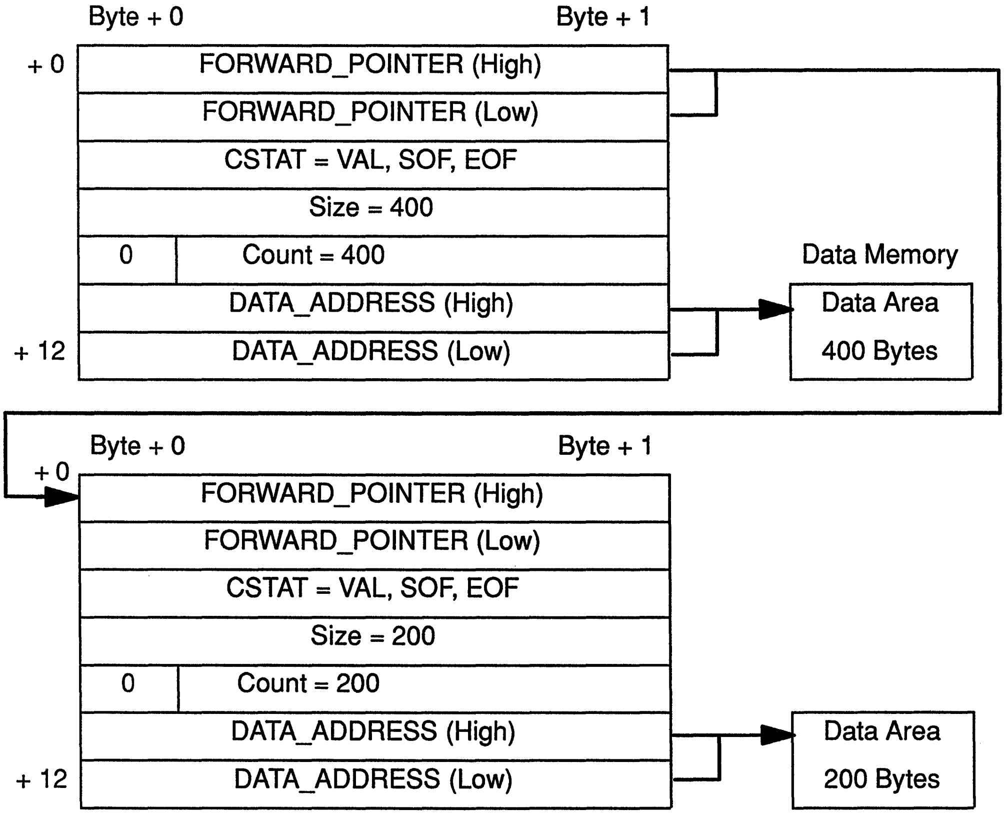


Figure 4–27. TRANSMIT List Format: Example 4



MAC 0005 TRANSMIT.HALT Command

Command TRANSMIT.HALT

Description The TRANSMIT.HALT command is used to interrupt the transmit list chain. Following recognition of this command, the adapter terminates the transmit chain as soon as possible. Any frames queued in the adapter are purged, and the TRANSMIT command is terminated by posting a TRANSMIT.STATUS interrupt and updating SSB_PARM_0 with XMIT_STATUS and SSB_PARM_1 and SSB_PARM_2 with a pointer to the last transmit parameter list processed by the adapter. If TRANSMIT.HALT is issued and no frames have been transmitted, SSB_PARM_1 and SSB _PARM_2 are cleared. This command is ignored by the adapter if there is not an executing TRANSMIT command.

Initiation

	Byte + 0	Byte + 1
SCB_CMD	>0005	
SCB_PARM_0	>0000	
SCB_PARM_1	>0000	

Completion

SSB_CMD	>0004
SSB_PARM_0	XMIT_STATUS
SSB_PARM_1	XMIT_PARM_LIST (High)
SSB_PARM_2	XMIT_PARM_LIST (Low)

Initiation TRANSMIT.HALT does not require a command parameter block. Both the SCB_PARM_0 and SCB_PARM_1 fields are ignored but should be set to zero.

Completion Upon completion of TRANSMIT.HALT, the adapter generates a COM- MAND.STATUS interrupt, sets SSB_CMD to >0004, updates SSB_CMD_0 with COMMAND_STATUS of the TRANSMIT command, and updates SSB_PARM_1 and SSB_PARM_2 with a pointer to the last transmit parameter list processed by the adapter.

Command RECEIVE

Description The RECEIVE command is used to receive frames from other stations on the ring. This command is normally issued only once (after OPEN) because receive data may be dynamically added to a RECEIVE parameter list chain. The RECEIVE command can be terminated because of a list error or by issuing a CLOSE command.

The Routing Field is passed to the attached system for all frames, if received. If the PAD ROUTING FIELD option is specified during OPEN, the routing field is padded to 18 bytes. If the frame does not contain a Routing Field, the field is still padded to 18 bytes. The padding does not alter the contents of the system's data buffer.

The logical format of received frames passed from the adapter to the attached system is identical to the logical format shown in Figure 4–16. The access control, frame control, destination address, source address, any routing information, and LLC fields are transferred to the attached system as they were received from the ring.

The RECEIVE command is rejected with adapter COMMAND REJECT STATUS under the following conditions:

□ The adapter has not been opened,
□ A RECEIVE command has already been issued, or
□ The address passed in the SCB is not word-aligned.

Initiation

	Byte + 0	Byte+ 1
SCB_CMD	>0006	
SCB_PARM_0	RCV_PARM_LIST (High)	
SCB_PARM_1	RCV_PARM_LIST (Low)	

Receive Status Interrupt

SSB_CMD	>0006
SSB_PARM_0	RCV_STATUS
SSB_PARM_1	RCV_PARM_LIST (High)
SSB_PARM_2	RCV_PARM_LIST (Low)

Initiation The SCB_PARM_0 and SCB_PARM_1 contain a 32-bit word-aligned address, which points to a RECEIVE parameter list. The format of the RECEIVE parameter list is shown in Figure 4–28. RECEIVE parameter lists must be aligned on even word boundaries.

Note:

When issuing this command, wait for the first word of the SCB to be cleared (>0000) before issuing another command.

Figure 4–28. RECEIVE Parameter List

<table>
<tr><th></th><th></th><th>Byte + 0</th><th>Byte + 1</th></tr>
<tr><td>RCV_PARM_LIST</td><td>+ 0</td><td colspan="2">FORWARD_POINTER (High)</td></tr>
<tr><td></td><td>+ 2</td><td colspan="2">FORWARD_POINTER (Low)</td></tr>
<tr><td></td><td>+ 4</td><td colspan="2">RECEIVE_CSTAT</td></tr>
<tr><td></td><td>+ 6</td><td colspan="2">FRAME_SIZE</td></tr>
<tr><td></td><td>+ 8</td><td colspan="2">DATA_COUNT</td></tr>
<tr><td></td><td>+ 10</td><td colspan="2">DATA_ADDRESS (High)</td></tr>
<tr><td></td><td>+ 12</td><td colspan="2">DATA_ADDRESS (Low)</td></tr>
<tr><td></td><td>+ 14</td><td colspan="2">DATA_COUNT</td></tr>
<tr><td></td><td>+ 16</td><td colspan="2">DATA_ADDRESS (High)</td></tr>
<tr><td></td><td>+ 18</td><td colspan="2">DATA_ADDRESS (Low)</td></tr>
<tr><td></td><td>+ 20</td><td colspan="2">DATA_COUNT</td></tr>
<tr><td></td><td>+ 22</td><td colspan="2">DATA_ADDRESS (High)</td></tr>
<tr><td></td><td>+ 24</td><td colspan="2">DATA_ADDRESS (Low)</td></tr>
</table>

The definitions of the FORWARD_POINTER, DATA_COUNT, and DATA_ADDRESS fields in Figure 4–28 are the same as for the transmit parameter list in Figure 4–23 and are given in Table 4–23. The RECEIVE_CSTAT field definition is given in Table 4–24.

Table 4–24. RECEIVE Parameter Field Bit Definitions

Field	Definition	
RECEIVE_CSTAT	This 16-bit parameter is set by the attached system when the receive parameter list is created. It is overwritten by the adapter to report frame completion status. When initially set by the attached system, this parameter field is referred to as the RECEIVE_CSTAT_REQUEST field. After a frame has been received, the adapter writes RECEIVE_CSTAT_COMPLETE to lists that start or end a frame. These bits indicate the completion status of the frame, not the RECEIVE command itself. However, if the forward pointer is odd, this field is not updated until a new list is given to the adapter. That is, the forward pointer is made even by the host and a RECEIVE_CONTINUE interrupt is issued to the adapter.	
RECEIVE_CSTAT_REQUEST	Bit 0	VALID: The adapter waits for bit 0 to be set to a one before processing the current receive parameter list. The attached system must issue a RECEIVE.VALID interrupt request when changing bit 0 from zero to one. This bit is examined for every receive parameter list.
	Bit 1	FRAME_COMPLETE: Must be reset to zero.
	Bit 2	START_FRAME: Must be reset to zero.
	Bit 3	END_FRAME: Must be reset to zero.
	Bit 4	FRAME_INTERRUPT: If this bit is set to one, an adapter-to-attached system interrupt is generated when a frame has been received. This bit is ignored unless a list starts a frame.
	Bit 5	INTERFRAME_WAIT: If this bit is set to one, the adapter interrupts the attached system when a frame has been received. The FRAME_COMPLETE bit (bit 0) sets the SSB when this occurs. The adapter then assumes a receive-suspended state, waiting for the attached system to issue a RECEIVE.CONTINUE interrupt. The next list to be used is addressed by the FORWARD POINTER of the list that has the FRAME_END bit set.This bit is ignored unless a list starts or ends a frame.
	Bit 6	PASS_CRC: When this bit is set, the adapter includes the frame's CRC in the information passed to the attached system. The CRC is the last four bytes passed to the attached system. The additional four bytes are reflected in the FRAME_SIZE field of the receive list. This bit is ignored in lists that do not start a frame. The CRC is passed only for externally matched frames.
	Bits 7—15	RESERVED: Must be reset to zero.

Table 4–24. RECEIVE Parameter Field Bit Definitions (Continued)

Field	Definition
RECEIVE_CSTAT_COMPLETE	The completion code for the received frame is written to this field in lists that start and end a frame. RECEIVE_CSTATs that are not in lists that define the start or end of a frame are not altered by the adapter. The RECEIVE_CSTAT_COMPLETE bit definitions are shown below. Bit 0 VALID: Reset to zero. Bit 1 FRAME_COMPLETE: Set to one. Bit 2 FRAME_START: Set to one on the list that starts the frame. Bit 3 FRAME_END: Set to one on the list that ends the frame. Bit 4 FRAME_INTERRUPT: Bit 4 has the same value as specified by the attached system in RECEIVE_CSTAT_REQUEST. Bit 5 INTERFRAME_WAIT: Bit 5 has the same value as specified by the attached system in RECEIVE_CSTAT_REQUEST. Bit 6 CRC_PASSED: This bit is set only if the PASS_CRC bit of the RECEIVE_CSTAT_RE-QUEST is set and if the CRC is contained within the data. This bit is set only in lists t hat have the FRAME_START bit (bit 2) set to one, and only if the frame received is externally matched. Bit 7 RESERVED: Reset to zero. Bits 8—13 RECEIVE_FS: On lists with START_FRAME set to one, this field contains the frame status field from the received frame. If START_FRAME is cleared (0), this field is cleared (0). Bits 14—15 ADDRESS_MATCH: These bits contain the codes shown in Table 4–25.
FRAME_SIZE	This 16-bit field contains the number of bytes in the received frame. The adapter stores this count in the receive list, which starts a new frame. FRAME_SIZE includes AC, FC, Destination and Source Address, Routing Field (if any), pad length (if PAD ROUTING FIELD specified), and the data fields. FRAME_SIZE does not include CRC, FS, or EDL. FRAME_SIZE should be used in conjunction with the DATA _COUNT Fields to determine how far the frame data extends into the last buffer data area.

Table 4–24. RECEIVE Parameter Field Bit Definitions (Continued)

Field	Definition
DATA_COUNT	This 16-bit field contains the maximum number of bytes that can be stored, starting at the address defined in the DATA_ADDRESS parameter. There can be a maximum of three DATA_COUNT and DATA_ADDRESS parameters to provide a scatter-write capability per received list (not frame). If bit 0 is clear (0), it is the last DATA_COUNT in the receive list. Bit 0 of the third DATA_COUNT is ignored. A zero DATA_COUNT is not permitted (regardless of the value of bit 0). The DATA_COUNT can be even or odd. The adapter does not alter this parameter. If the PAD ROUTING FIELD option is specified during the OPEN command, then the first DATA_COUNT in a receive list used for start of frame, must be at least 32. This allows space for the AC, FC, DESTINATION ADDRESS, SOURCE ADDRESS, and the ROUTING FIELD, which is padded to 18 bytes. If the DATA_COUNT is less than 32, the option is voided.
DATA_ADDRESS	This 32-bit field contains the address of the data to be received. DATA_ADDRESS may be even or odd. The adapter does not alter this parameter.

Table 4–25. ADDRESS_MATCH Codes

Code		Definition
Bit 14	Bit 15	
0	0	This code can be observed in frames copied by the copy all frames option, but not internally or externally matched.
0	1	Internally address matched
1	0	Externally matched via XMATCH/XFAIL interface
1	1	Internally and externally matched

RECEIVE CSTAT REQUEST Fields

Bit #	0	1	2	3	4	5	6	7
MSB	Valid	Frame Complete = 0	Start of Frame = 0	End of Frame = 0	Frame Interrupt	Interframe Wait	Pass CRC	0

Bit #	8	9	10	11	12	13	14	15
LSB	0	0	0	0	0	0	0	0

RECEIVE CSTAT COMPLETE Fields

Bit #	0	1	2	3	4	5	6	7
MSB	Valid = 0	Frame Complete = 1	Start of Frame	End of Frame	Frame Interrupt	Interframe Wait	CRC Passed	0

Bit #	8	9	10	11	12	13	14	15
LSB	Receive FS bit 1	Receive FS bit 2	Receive FS bit 3	Receive FS bit 4	Receive FS bit 5	Receive FS bit 6	External Address Match	Internal Address Match

Execution When the LLC interface is not enabled, the attached system can create a circular chain of receive parameter lists by setting the FORWARD_POINTER of the last receive parameter list to the address of the first list. The valid bit of RECEIVE_CSTAT_REQUEST is manipulated to control the flow of data to the attached system. When the adapter reads a list with the valid bit reset, it suspends the RECEIVE command until a RECEIVE.VALID interrupt request is issued. The attached system is not notified of this receive suspended condition by the adapter. If the adapter reads an odd FORWARD_POINTER or if the FRAME_INTERRUPT bit is set in the RECEIVE_CSTAT, the adapter suspends the receive command and interrupts the attached system. The adapter resumes processing the RECEIVE command when the attached system issues a RECEIVE.CONTINUE interrupt.

When you create a circular chain of receive lists, the sum of the DATA_COUNT fields must be equal to or greater than the largest frame to be received. Since the CSTAT fields are not updated by the adapter until a frame has been transferred, violating this rule causes the adapter to overwrite portions of the same frame being transferred (in other words, the frame will wrap back upon itself). Thus, data will be lost.

The RECEIVE.CONTINUE interrupt can be issued at any time, and the adapter will ignore the interrupt if it is not waiting for a FORWARD_POINTER transition from an odd to even address.

Header Routing

Some systems may need to receive only a frame header (or header and a portion of the data) and route the remainder of the frame data according to the contents of the header. This can be accomplished as follows:

1) Set FRAME HOLD in the open command.
2) Post a receive list that has an odd FORWARD_POINTER and one DATA_COUNT and DATA_ADDRESS parameter sufficient to hold the desired header.
3) The adapter uses the list and interrupts the system with RECEIVE_SUSPENDED, leaving the CSTAT unchanged.
4) Following RECEIVE_SUSPENDED, the system can examine the header and determine the frame destination. The frame size field of the receive parameter list is not updated by the adapter and is not valid.
5) Post additional lists to receive the data by writing a non-odd address in the FORWARD_POINTER field, followed by another header list with an odd FORWARD_POINTER.
6) Issue a RECEIVE CONTINUE interrupt request.
7) When the frame has been transferred, a FRAME COMPLETE interrupt occurs (if requested).

RECEIVE Frame Hold Operation

Receive processing is affected by the value of the FRAME_HOLD bit of the OPEN_OPTIONS. If the FRAME_HOLD bit is set, the adapter waits until an entire frame has been received before sending it to the host. If the FRAME_HOLD bit is not set, the adapter processes the host list until it sees an end-of-frame adapter buffer. Since this processing occurs while the frame is being received (dynamic piping), the adapter is not aware of the frame status until the end-of-frame buffer is processed. If the end-of-frame buffer has good status, the adapter updates the start and end list CSTATs with their respective statuses. Otherwise, the adapter rereads the start-of-frame list, effectively recovering the list, and processes the next received frame.

Because the adapter does not update the host's receive lists' CSTATs until after the whole frame is DMAed to the host (regardless of the FRAME_HOLD bit value), the host must make sure that there are enough receive lists so that the adapter can DMA the entire received frame. If the FRAME_HOLD bit is set, the host must make sure that there are enough receive lists so that the sum of the data count fields is equal to or greater than the largest frame to be received. If the FRAME_HOLD bit is not set, there should be enough receive lists so that the sum of the DATA_COUNT fields is equal to or greater than the largest erred frame received.

In the event that an erred frame is received so that it is not terminated by an EDEL or a burst-5 error, its length is determined by the token timer of the MAC code (10 ms). This means that the largest frame that could be be seen, if FRAME_HOLD bit is not set, is less than 20-ms long, or up to 10Kbyte in length, before the error recovery of the list can be started. At the 16-Mbps rate, the error frame can be up to 40Kbytes in length. Take care when not using the FRAME_HOLD function because more host receive buffers are necessary. Note that this is applicable only when MAC software is used (in other words, LLC is NOT enabled).

Interrupt RECEIVE.STATUS

A RECEIVE.STATUS interrupt is generated when the RECEIVE parameter list chain has ended (odd address in FORWARD_POINTER) or when a frame is copied into a list that has the FRAME_INTERRUPT bit set in the RECEIVE_CSTAT field. The adapter updates SSB_PARM_0 with the RECEIVE_COMPLETE code and SSB_PARM_1 and SSB_PARM_2 with a 32-bit pointer to the last receive parameter list processed by the adapter. The RECEIVE_COMPLETE bit definitions are listed in Table 4–26.

Table 4–26. RECEIVE_COMPLETE Field Bit Definitions

Bit	Definition
0	FRAME_COMPLETE: Bit 0 is set to one when a frame has been received and the FRAME_INTERRUPT bit is set in RECEIVE_CSTAT. Since frames may be received and transferred faster than the attached system can respond to the interrupts and/or faster than the adapter can cause the interrupts, the RECEIVE.STATUS interrupt may report the arrival of more than one frame at a time. The SSB_PARM_1 and SSB_PARM_2 contain the receive parameter list address of the last frame transferred to the system. If lists with the FRAME_INTERRUPT bit set are intermixed with lists that do not have the FRAME_INTERRUPT bit set, RECEIVE.STATUS can include frames that do not have the FRAME_INTERRUPT bit set. The FRAME_COMPLETE bit will not be set with the RECEIVE_SUSPENDED bit also set.
1	RECEIVE_SUSPENDED: Bit 1 is set to one when the adapter detects an odd address in the FORWARD_POINTER field of a receive parameter list. SSB_PARM_1 and SSB_PARM_2 contain the address of the list that has an odd FORWARD_POINTER. The attached system must update the FORWARD_POINTER and issue a RECEIVE.CONTINUE interrupt or a RECEIVE.CANCEL to resume the RECEIVE command processing. The RECEIVE_SUSPENDED bit is not set with the FRAME_COMPLETE bit also set.
2—15	RESERVED: Reset to zero.

Command CLOSE

Description The CLOSE command is used to terminate transmission on the ring or to terminate OPEN with the wrap option. Any frames residing in adapter internal buffers are purged. After this command has completed, the adapter must be reset and then reinitialized before issuing an OPEN command.

A COMMAND.REJECT interrupt is issued if the adapter has not been opened.

Initiation

	Byte + 0	Byte + 1
SCB_CMD	>0007	
SCB_PARM_0	>0000	
SCB_PARM_1	>0000	

Completion

SSB_CMD	>0007
SSB_PARM_0	COMMAND_STATUS
SSB_PARM_1	>0000
SSB_PARM_2	>0000

Initiation CLOSE does not require a command parameter block. Both the SCB_PARM_0 and SCB_PARM_1 fields are ignored but should be set to zero.

Completion Upon completion of CLOSE, the adapter generates a COMMAND.STATUS interrupt, sets SSB_CMD to >0007, and updates SSB_PARM_0 with COMMAND_STATUS, as shown in Table 4–27.

Table 4–27. CLOSE Return Codes

Value	Explanation
>8000	Good completion

MAC 0008 SET.GROUP.ADDRESS Command

Command SET.GROUP.ADDRESS

Description The SET.GROUP.ADDRESS command is used to alter the adapter group address after an OPEN command has been issued.

A COMMAND.REJECT interrupt is issued if the adapter has not been opened.

Initiation

	Byte + 0	Byte + 1
SCB_CMD	>0008	
SCB_PARM_0	GROUP_ADDRESS_HIGH	
SCB_PARM_1	GROUP_ADDRESS_LOW	

Completion

SSB_CMD	>0008
SSB_PARM_0	COMMAND_STATUS
SSB_PARM_1	>0000
SSB_PARM_2	>0000

Initiation The SCB_PARM_0 and SCB_PARM_1 fields contain a 32-bit address, which is the group address to be stored in the adapter group address register. Bit 0 is forced to one by the adapter. The most significant two bytes of the 48-bit group address are >C000.

Completion Upon completion of SET.GROUP.ADDRESS, the adapter generates a COMMAND.STATUS interrupt. The SSB_CMD field is set to >0008, and the SSB_PARM_0 field is updated with COMMAND_STATUS, as shown in Table 4–28.

Table 4–28. SET.GROUP.ADDRESS Return Codes

Value	Explanation
>8000	Good completion

Command SET.FUNCTIONAL.ADDRESS

Description The SET.FUNCTIONAL.ADDRESS command is used to alter the adapter functional address after an OPEN command has been issued.

A COMMAND.REJECT interrupt is issued if the adapter has not been opened.

Initiation

	Byte + 0	Byte + 1
SCB_CMD	>0009	
SCB_PARM_0	FUNCTIONAL_ADDRESS_HIGH	
SCB_PARM_1	FUNCTIONAL_ADDRESS_LOW	

Completion

SSB_CMD	>0009
SSB_PARM_0	COMMAND_STATUS
SSB_PARM_1	>0000
SSB_PARM_2	>0000

Initiation The SCB_PARM_0 and SCB_PARM_1 fields contain a 32-bit address, which is the functional address to be stored in the adapter's internal functional address register. Bits 0 (most significant bit) and 31 (least significant bit) of the functional address are ignored. The most significant two bytes of the 48-bit functional address are >C000.

Completion Upon completion of SET.FUNCTIONAL.ADDRESS, the adapter generates a COMMAND.STATUS interrupt and sets the SSB_CMD field to >0009. The SSB_PARM_0 field is updated with COMMAND_STATUS, as shown in Table 4–29.

Table 4–29. SET.FUNCTIONAL.ADDRESS Return Codes

Value	Explanation
>8000	Good completion

MAC 000A READ.ERROR.LOG Command

Command READ.ERROR.LOG

Description The READ.ERROR.LOG command is used to read the adapter's error counters. These error counters are reset after this command has completed.

This command is ignored if the adapter has not been opened.

Initiation

	Byte + 0	Byte + 1
SCB_CMD	>000A	
SCB_PARM_0	ERROR_LOG_ADDR (High)	
SCB_PARM_1	ERROR_LOG_ADDR (Low)	

Completion

SSB_CMD	>000A	
SSB_PARM_0	COMMAND_STATUS	
SSB_PARM_1	>0000	
SSB_PARM_2	>0000	

Initiation The SCB_PARM_0 and SCB_PARM_1 fields contain a 32-bit pointer to a 14-byte area in attached system memory where the error log table is to be written. The 14-byte error log table format is shown in Figure 4–29. The error counters are explained in Table 4–30.

Figure 4–29. Error Log Table

	Byte + 0	Byte + 1
+ 0	LINE_ERROR	RESERVED
+ 2	BURST_ERROR	ARI/FCI_ERROR
+ 4	RESERVED	RESERVED
+ 6	LOST_FRAME _ERROR	RECEIVE_CONGESTION _ERROR
+ 8	FRAME_COPIED_ERROR	RESERVED
+ 10	TOKEN_ERROR	RESERVED
+ 12	DMA_BUS_ERRORS	DMA_PARITY_ERRORS

Table 4–30. Adapter Error Counters

Counter	Definition
LINE_ERROR	The line error counter is incremented whenever 1) A frame is repeated or copied, and 2) The Error Detected Indicator (EDI) is zero in the incoming frame, and 3) At least one of the following conditions exists: a) A code violation between the starting delimiter and the ending delimiter of the frame. b) A code violation in a token. c) A Frame Check Sequence (FCS) error. When the line error is incremented, the EDI of the frame is set to one so that no further adapters count the error.
BURST_ERROR	The burst error counter is contained in all adapter configurations and is incremented when the adapter detects the absence of transitions for five half-bit times between SDEL and EDEL, or EDEL and SDEL. Only one adapter detects the burst five condition because the adapter that detects a burst four condition (four half-bit times without transition) directs its transmitter to transmit idles if the burst five condition is detected.
ARI/FCI_ERROR	The ARI/FCI error counter is incremented when an adapter receives an Active Monitor Present (AMP) MAC frame with the ARI/FCI bits equal to zero and a Standby Monitor Present (SMP) MAC frame with the ARI/FCI bits equal to zero, or more than one SMP MAC frame with the ARI/FCI bits equal to zero, without receiving an intervening AMP MAC frame. This condition indicates that the upstream neighbor is unable to set the ARI/FCI bits in a frame that it has copied.
LOST_FRAME_ERROR	The lost frame error counter is incremented when an adapter is in transmit (stripping) mode and fails to receive the end of the frame it transmitted.
RECEIVE_CONGESTION_ERROR	The receive congestion error counter is incremented when an adapter in the repeat mode recognizes a frame addressed to it but has no buffer space available to copy the frame.
FRAME_COPIED_ERROR	The frame copied error counter is incremented when an adapter in the receive/repeat mode recognizes a frame addressed to its specific address but finds the ARI bits not equal to zero. This indicates a possible line hit or duplicate address.

Table 4–30. Adapter Error Counters (Continued)

Counter	Definition
TOKEN_ERROR	The token error counter is active only in the active monitor station. It is incremented when the active monitor detects an error with the token protocol as follows: 1) The MONITOR_COUNT bit of a token with nonzero priority equals one. 2) The MONITOR_COUNT bit of a frame equals one. 3) No token or frame is received within a 10-ms window. 4) The starting delimiter/token sequence has a code violation in an area where code violations must not exist.
DMA_BUS_ERRORS	The DMA bus error counter counts the occurrences of DMA bus errors that do not exceed the abort thresholds as specified in the Initialization parameters.
DMA_PARITY _ERRORS	The DMA parity error counter counts the occurrences of DMA parity errors that do not exceed the abort thresholds as specified in the Initialization parameters.

Completion Upon completion of READ.ERROR.LOG, the adapter generates a COMMAND.STATUS interrupt and sets the SSB_CMD field to >000A. The SSB_PARM_0 field is updated with COMMAND_STATUS as shown in Table 4–31.

Table 4–31. READ.ERROR.LOG Return Codes

Value	Explanation
>8000	Good completion

Command READ.ADAPTER

Description The READ.ADAPTER command is used to transfer adapter memory contents across the system interface to host system memory.

Initiation

	Byte + 0	Byte + 1
SCB_CMD	>000B	
SCB_PARM_0	DATA_AREA_ADDR (High)	
SCB_PARM_1	DATA_AREA_ADDR (Low)	

Completion

SSB_CMD	>000B
SSB_PARM_0	COMMAND_STATUS
SSB_PARM_1	>0000
SSB_PARM_2	>0000

Initiation The SCB_PARM_0 and SCB_PARM_1 fields contain a 32-bit pointer to a buffer in host system memory, where the adapter stores the contents of the specified adapter memory locations. The host system data area specified by the host system should be written to in the format shown in Figure 4–30.

Figure 4–30. Host System Data Area Format

+ 0	DATA_COUNT	Data area
+ 2	DATA_ADDRESS	
+ 4		
+ N-2		

This 16-bit DATA_COUNT field contains the number of bytes to be read from the adapter.

DATA_ADDRESS is a 16-bit field containing the address of the data in Chapter 1 of adapter memory to be read. Bit 15 is reset to zero by the adapter. The READ.ADAPTER command results in an ADAPTER.CHECK interrupt if reference is made to an undefined storage area.

Note:

Any chapter of TMS380C16 memory may be inspected by using the DIO registers, with the following exceptions:

1) Address range >01.0200 through >01.02FE. If this is read, an adapter check occurs.
2) Address range >01.0100 through >01.01FE. If this is read, erroneous operation and deinsertion may occur.

Table 4–8 illustrates some of the internal adapter pointers accessible via the READ.ADAPTER command or DIO. These pointers reside beginning at location >01.0A00 in adapter memory; they must be read following initialization but before the OPEN command is issued.

Completion Upon completion of READ.ADAPTER, the adapter generates a COMMAND.STATUS interrupt and sets the SSB_CMD field to >000B. The SSB_PARM_0 field is updated with COMMAND_STATUS as shown in Table 4–32.

Table 4–32. READ.ADAPTER Return Codes

Value	Explanation
>8000	Good completion

MAC 000D MODIFY.OPEN.PARAMETERS Command

Command MODIFY.OPEN.PARAMETERS

Description MODIFY.OPEN.PARAMETERS may be used to modify certain adapter operational parameters that were set by the OPEN command. All OPEN options, with the exception of WRAP_INTERFACE, that were specified in the OPEN_OPTIONS field of the OPEN parameter list may be modified. The bit corresponding to WRAP_INTERFACE (bit 0) is ignored by this command.

A COMMAND.REJECT is issued if the adapter has not been opened.

Initiation

	Byte + 0 Byte + 1
SCB_CMD	>000D
SCB_PARM_0	OPEN_OPTIONS
SCB_PARM_1	>0000

Completion

SSB_CMD	>000D
SSB_PARM_0	COMMAND_STATUS
SSB_PARM_1	>0000
SSB_PARM_2	>0000

Initiation The SCB_PARM_0 of the SCB contains a 16-bit OPEN_OPTION field. The bit descriptions for this field are the same as those specified for the OPEN_OPTIONS field of the OPEN command. SCB_PARM_1 is ignored but should be set to zero.

Completion Upon completion of MODIFY.OPEN.PARAMETERS, the adapter generates a COMMAND.STATUS interrupt, sets SSB_CMD to >000D and updates SSB_PARM_0 with COMMAND_STATUS. The COMMAND_STATUS takes on values according to Table 4–33.

Table 4–33. MODIFY.OPEN.PARAMETERS Return Codes

Value	Explanation
>8000	Good completion
>0100	Invalid open option

MAC 000E RESTORE.OPEN.PARAMETERS Command

Command RESTORE.OPEN.PARAMETERS

Description RESTORE.OPEN.PARAMETERS may be used to modify certain adapter operational parameters that were set by the OPEN command. All Open options, with the exception of WRAP_INTERFACE, that were specified in the OPEN_OPTIONS field of the open parameter list may be modified. The bit corresponding to WRAP_INTERFACE (bit 0) is ignored by this command.

A COMMAND.REJECT is issued if the adapter has not been opened.

Initiation

	Byte + 0	Byte + 1
SCB_CMD	>000E	
SCB_PARM_0	OPEN_OPTIONS	
SCB_PARM_1	>0000	

Completion

SSB_CMD	>000E
SSB_PARM_0	COMMAND_STATUS
SSB_PARM_1	>0000
SSB_PARM_2	>0000

Initiation The SCB_CMD field of the SCB must be set to 000E. The SCB_PARM_0 field of the SCB contains a 16-bit OPEN_OPTIONS field. The bit descriptions for this field are the same as those specified for the OPEN_OPTIONS field of the OPEN command. SCB_PARM_1 must be set to zero.

Completion Upon completion of RESTORE.OPEN.PARAMETERS command, the adapter generates a COMMAND.STATUS interrupt, sets SSB_CMD to >000E, and updates SSB_PARM_0 with COMMAND_STATUS. The COMMAND_STATUS takes on values according to Table 4–34.

Table 4–34. RESTORE.OPEN.PARAMETERS Return Codes

Value	Explanation
>8000	Good completion
>0100	Invalid open option

MAC 000F SET.FIRST.16.GROUP.ADDRESS Command

Command SET.FIRST.16.GROUP.ADDRESS

Description The SET.FIRST.16.GROUP.ADDRESS command is used to alter the first two bytes in the adapter group address. After the OPEN command has been issued, these first two bytes are >C000.

A COMMAND.REJECT is issued if the adapter has not been opened.

Initiation

	Byte + 0	Byte + 1
SCB_CMD	>000F	
SCB_PARM_0	FIRST.16.GROUP.ADDRESS	
SCB_PARM_1	>0000	

Completion

SSB_CMD	>000F	
SSB_PARM_0	COMMAND_STATUS	
SSB_PARM_1	>0000	
SSB_PARM_2	>0000	

Initiation The SCB_PARM_0 field contains the two high-order bytes in the group address to be stored in the adapter group address register. Bit 0 is forced to one by the adapter.

Completion Upon completion of SET.FIRST.16.GROUP.ADDRESS, the adapter generates a COMMAND.STATUS interrupt with SSB_CMD field set to >000F and COMMAND_STATUS as shown in Table 4–35.

Table 4–35. SET.FIRST.16.GROUP.ADDRESS Return Codes

Value	Explanation
>8000	Good completion

MAC 0010 SET.BRIDGE.PARMS Command

Command SET.BRIDGE.PARMS

Description The SET.BRIDGE.PARMS command is used with adapters that implement an external source routing checker device. This command provides values and conditions for the adapter hardware to use when frames are copied for forwarding.

This command must be issued after the adapter has been opened. A COMMAND.REJECT is issued if the adapter has not been opened.

Initiation

	Byte + 0	Byte + 1
SCB_CMD	>0010	
SCB_PARM_0	BRIDGE_PARM_BLOCK (High)	
SCB_PARM_1	BRIDGE_PARM_BLOCK (Low)	

Completion

SSB_CMD	>0010
SSB_PARM_0	COMMAND_STATUS
SSB_PARM_1	>0000
SSB_PARM_2	>0000

Initiation The SCB_PARM_0 and SCB_PARM_1 fields contain a 32-bit pointer to the BRIDGE_PARM_BLOCK, which is shown in Figure 4–31. The detailed descriptions in the BRIDGE_PARM_BLOCK are shown in Table 4–36.

Figure 4–31. BRIDGE_PARM_BLOCK

	Byte + 0	Byte + 1
+ 0	OPTIONS	
+ 2	SOURCE_RING	
+ 4	TARGET_RING	
+ 6	BRIDGE_NUMBER	

Table 4–36. BRIDGE_PARM_BLOCK Field Definitions

Field	Definition	
OPTIONS	Bit 0	SINGLE_ROUTE_BROADCAST: This bit is used to limit the routing of broadcast frames. When this bit is zero, single-route broadcast frames are rejected; all-routes broadcast frames are considered for forwarding to the host. If this bit is a one, then both single-route and all-routes broadcast frames are forwarded.
	Bits 1—5	RESERVED: These bits must be set to zero.
	Bits 6—11	MAXIMUM_ROUTING_FIELD_LENGTH: Maximum number of bytes (up to 30) that the routing field may contain, including the routing control field. Broadcast frames are rejected if they equal the maximum length, and all frames are rejected if the maximum length is exceeded. To operate to the IBM specification, a value of 18 should be used.
	Bits 12—15	PARTITION_LENGTH: The value in this field is used to determine what portion of each 2-byte segment in the routing information field contains the bridge number. A value of 4 means that the low-order 4 bits of the segment indicate the bridge number. The remaining 12 bits contain the ring number. There is no default value for this field. The host system software is responsible for maintaining the validity check on the value used. All bridges in the network must use the same value for this field or its equivalent. A partition length of zero, or more than 14, results in an invalid bridge option.
SOURCE_RING	When determining whether to forward a frame, the adapter compares the value in this field with the routing information source ring field in frames received from the ring. This value must be the number of the ring to which this adapter is connected. The acceptable range of values for this field depends on the setting of the partition length in the option field. For example, the valid range of values is >001 to >FFF if the partition length parameter value is 4. The SOURCE_RING value must be different from the TARGET_RING value.	

Table 4–36. BRIDGE_PARM_BLOCK Field Definitions (Continued)

Field	Definition
TARGET_RING	When determining whether to forward a frame, the adapter compares the value in this field with the routing information target ring field in frames received from the ring. This value must be the number of the ring to which the other adapter in this host is connected. The acceptable range of values for this field depends on the setting of the partition length in the option field. For example, the valid range of values is >001 to >FFF if the partition length parameter value is 4. The TARGET_RING value must be different from the SOURCE_RING value.
BRIDGE_NUMBER	When determining whether to forward a frame, the adapter compares the value in this field with the routing information bridge number field in the frames received from the ring. This value can be from zero up to the number allowed by the partition length.

Completion Upon completion of the SET.BRIDGE.PARMS command, the adapter generates a COMMAND.STATUS interrupt and sets the SSB_CMD field to >0010. The SSB_PARM_0 field is updated with COMMAND_STATUS. If there is a SET_BRIDGE_PARMS error, the TMS380SRA is disabled. The COMMAND_STATUS takes on values according to Table 4–37.

Table 4–37. SET.BRIDGE.PARMS Return Codes

Bit	Definition
0	Good completion
1	Invalid maximum routing field length. This length given was odd, less than 6, or larger than supported by the hardware. (Maximum for TMS380SRA is 30.)
2	Invalid SOURCE_RING number. The number given was zero or was larger than allowed by the setting of the PARTITION_LENGTH in the options field. This bit is set if the TARGET and SOURCE_RING numbers are equal.
3	Invalid TARGET_RING number. The number given was zero or was larger than allowed by the setting of the PARTITION_LENGTH in the options field. This bit is set if the TARGET and SOURCE_RING numbers are equal.
4	Invalid BRIDGE_NUMBER. The number given was larger than allowed by the setting of the PARTITION_LENGTH in the options field.
5	Invalid bridge options.
6	The TMS380SRA failed diagnostics.
7	The TMS380SRA does not exist in the present hardware configuration.

Command CONFIG.BRIDGE.PARMS

Description The CONFIG.BRIDGE.PARMS command forces the bridge functional address bit (bit 23) to one in the OPEN.ADAPTER and SET.FUNCTIONAL.ADDRESS commands. This comand must be issued after the adapter is opened. A COMMAND.REJECT is issued if the adapter is not open.

Initiation

	Byte + 0	Byte + 1
SCB_CMD	>0011	
SCB_PARM_0	CONTROL	
SCB_PARM_1	>0000	

Completion

SSB_CMD	>0011
SSB_PARM_0	COMMAND_STATUS
SSB_PARM_1	>0000
SSB_PARM_2	>0000

Initiation The SCB_PARM_0 field contains the CONTROL parameter for this command. If the CONTROL parameter is nonzero, then the bridge functional address (>C000 0000 0100) is set. A CONTROL of zero turns off the bridge functional address. The SCB_PARM_1 field is ignored but should be set to zero.

Completion Upon completion of the CONFIG.BRIDGE.PARMS, the adapter generates a COMMAND.STATUS interrupt, sets SSB_CMD to >0011, and updates SSB_PARM_0 with COMMAND_STATUS. The COMMAND_STATUS takes on values according to Table 4–38.

Table 4–38. CONFIG.BRIDGE.PARMS Return Codes

Value	Explanation
>8000	Good completion

4.14 Adapter LLC Commands

This section discusses the commands that are available when the LLC adapter software is installed and the LLC_ENABLE bit is set (1). When the LLC_ENABLE bit is clear (0) please refer to the adapter MAC-only commands. This adapter software is referred to as the LLC interface. Command codes indicated as LLC interface are valid only when the LLC_ENABLE bit of the initialization options is set to one and the proper adapter software is loaded. Figure 4–32 lists these commands and corresponding command codes.

Figure 4–32. TMS380C16 COMMProcessor Command Set

Command	Code
OPEN	>0003
TRANSMIT	>0004
TRANSMIT.HALT	>0005
RECEIVE	>0006
CLOSE	>0007
SET.GROUP.ADDRESS	>0008
SET.FUNCTIONAL.ADDRESS	>0009
READ.ERROR.LOG	>000A
READ.ADAPTER	>000B
MODIFY.OPEN.PARAMETERS	>000D
RESTORE.OPEN.PARAMETERS	>000E
SET.FIRST.16.GROUP.ADDRESS	>000F
SET.BRIDGE.PARMS	>0010
CONFIG.BRIDGE.PARMS	>0011
LLC.RESET	>0014
OPEN.SAP	>0015
CLOSE.SAP	>0016
OPEN.STATION	>0019
CLOSE.STATION	>001A
CONNECT.STATION	>001B
MODIFY.LLC.PARMS	>001C
FLOW.CONTROL	>001D
LLC_STATISTICS	>001E
DIR.INTERRUPT	>001F
LLC.REALLOCATE	>0021
TIMER.SET	>0022
TRANSMIT.I.FRAME.REQUEST (See note below.)	>0023

Note:

The TRANSMIT.I.FRAME.REQUEST command is valid only if bit 12 in the OPEN_OPTIONS of the OPEN_PARM_BLOCK has been selected during the adapter OPEN command.

Command OPEN

Description Before the adapter can be used for data communications, the attached system must first open the adapter by issuing an OPEN command. The OPEN command sets the adapter's addresses and enables the receipt of frames from the ring. A RECEIVE command should be the first command issued after a successful OPEN completion. During the OPEN process, the adapter suspends processing of all interrupt requests except reset.

Since the LLC interface is enabled, the OPEN command also enables the global (>FF) and null (>00) Service Access Points (SAPs) of LLC. The options that are set by this command can be modified by using the MODIFY.OPEN.PARAMETERS command or by closing the adapter via the CLOSE command and then reopening the adapter with the desired options.

Initiation

	Byte + 0	Byte + 1
SCB_CMD	>0003	
SCB_PARM_0	OPEN_PARM_BLOCK (High)	
SCB_PARM_1	OPEN_PARM_BLOCK (Low)	

Completion

SSB_CMD	>0003	
SSB_PARM_0	Status	Error

Initiation The SCB_PARM_0 and SCB_PARM_1 fields of the SCB point to an OPEN parameter block, which is used by the adapter to configure itself for operation. The size of this block is dependent on whether the LLC interface was enabled during initialization of the adapter (LLC_ENABLE = 1). If the LLC_ENABLE bit was set to one, the adapter reads a 44-byte block of system memory via DMA; otherwise, the block is 32 bytes. Table 4–39 gives the OPEN parameter block field descriptions, and Table 4–40 gives a detailed description of each field. The SCB_PARM_0 and SCB_PARM_1 contain a 32-bit address that points to the OPEN parameter list.

Table 4–39. OPEN Parameter Block

		Byte + 0	Byte + 1
OPEN_PARM_BLOCK	+ 0	OPEN_OPTIONS	
	+ 2	NODE_ADDRESS (High)	
	+ 4	NODE_ADDRESS	
	+ 6	NODE_ADDRESS (Low)	
	+ 8	GROUP_ADDRESS (High)	
	+ 10	GROUP_ADDRESS (Low)	
	+ 12	FUNCTIONAL_ADDRESS (High)	
	+ 14	FUNCTIONAL_ADDRESS (Low)	
	+ 16	RECEIVE_LIST_SIZE	
	+ 18	TRANSMIT_LIST_SIZE	
	+ 20	BUFFER_SIZE	
	+ 22	RESERVED	
	+ 24	RESERVED	
	+ 26	XMIT_BUF_MIN	XMIT_BUF_MAX
	+ 28	PRODUCT_ID_ADDRESS	
	+ 30	PRODUCT_ID_ADDRESS	
	+ 32	SAP_MAX	STATION_MAX
	+ 34	GSAP_MAX	GSAP_MEMBER_MAX
	+ 36	TIMER_T1_1	TIMER_T2_1
	+ 38	TIMER_TI_1	TIMER_T1_2
	+ 40	TIMER_T2_2	TIMER_TI_2
	+ 42	MAX_FRAME_SIZE	

Table 4–40. OPEN Parameter Block Field Descriptions

Field	Definition	
OPEN_OPTIONS	Bit 0	WRAP_INTERFACE: If this bit is set, ring insertion process is omitted, and all user transmit data appears as user receive data. The data is transmitted on the lobe from the attached product to the wiring concentrator. This option can be used for system interface debug, system interface DMA testing, or lobe media testing. MODIFY.OPEN.PARA-METERS cannot be used to alter the value of this bit; a CLOSE command must be issued to terminate WRAP mode.
	Bit 1	DISABLE_HARD_ERROR: If this bit is set, a RING.STATUS interrupt for HARD_ERROR and TRANSMIT_BEACON is not generated.
	Bit 2	DISABLE_SOFT_ERROR: If this bit is set, a RING.STATUS interrupt for SOFT_ERROR is not generated.
	Bit 3	PASS_ADAPTER_MAC_FRAMES: The value of this bit determines the action to be taken by the adapter when unsupported adapter MAC frames are received. If this bit is set to one, these MAC frames are passed to the attached system as normal receive data. If this bit is reset to zero, the adapter purges these frames and transmits a negative response MAC frame to the originating station.
	Bit 4	PASS_ATTENTION_MAC_FRAMES: If this bit is set to one, all attention MAC frames that are not equal to the last attention MAC frame received are passed to the system as receive data, following normal processing by the adapter.

Table 4–40. OPEN Parameter Block Field Descriptions (Continued)

Field	Definition	
OPEN_OPTIONS (Cont.)	Bit 5	PAD_ROUTING_FIELD: If this bit is set to one, the adapter pads the routing field to 18 bytes. If no routing field is present in the received frame, the entire field is padded to 18 bytes. This option is voided if the current list's data count is not at least 32 bytes. In this case, the frame is transferred as if the bit were set to zero.
	Bit 6	FRAME_HOLD: If this bit is set to one, the adapter waits for an entire frame to be read from the ring before initiating the DMA transfer of the frame to the system. This bit is forced to one if the LLC interface is enabled.
	Bit 7	CONTENDER: If this bit is set to one, the adapter participates in the monitor contention process when another adapter detects the need for contention and initiates the process. This bit has no effect if this adapter detects the need for contention and initiates the monitor contention process.
	Bit 8	PASS_BEACON_MAC_FRAMES: If this bit is set, the adapter passes beacon MAC frames to the attached system after processing them. After passing the beacon MAC frame, the next beacon MAC frame is passed only if the source address or the beacon-type subvector changes.
	Bits 9—10	RESERVED: Must be reset to zero.
	Bit 11	EARLY_TOKEN_RELEASE: This bit is valid only in 16-Mbps operation. When this bit is set to zero, the TMS380C16 operates in the early token release mode. This bit has no effect in 4-Mbps operation.

Table 4–40. OPEN Parameter Block Field Descriptions (Continued)

Field	Definition
OPEN_OPTIONS (Cont.)	Bit 12 HOST-BASED_I_FRAME_TRANSMIT: Setting this bit to a one causes the adapter to use the host-based I-frame transmit method instead of the original adapter-based I-frame transmit method. A description of these methods is given beginning with "Transmit Operation" on page 4-135. Bits 13—15 RESERVED. Must be set to zero.
NODE_ADDRESS	This 6-byte field specifies the node address for the adapter. If this address is all zeros, the adapter uses the BIA read from the BIA PROM. Byte 0, bit 0 must be set to zero.
GROUP_ADDRESS	This 32-bit field specifies the adapter group address and causes the adapter to receive messages that are sent to this address. GROUP_ADDRESS can be any value, but bit 0 is forced to one by the adapter. The two high-order bytes of the group address are set to >C000.
FUNCTIONAL_ADDRESS	This 32-bit field specifies the functional address and causes the adapter to receive messages that are sent to the functional address. Bits 0 and 31 are ignored by the adapter. The two high-order bytes of the functional address are set to >C000.
RECEIVE_LIST_SIZE	This 16-bit field indicates the number of bytes that must be read by the adapter when obtaining a receive list from the attached system. A decimal value of 0, 14, 20, or 26 is required. If zero, the default value of 26 is used.
TRANSMIT_LIST_SIZE	This 16-bit field indicates the number of bytes that must be read by the adapter when obtaining a transmit list from the attached system. A decimal value of 0, 14, 20, or 26 is required. If zero, the default value of 26 is used. If the LLC interface is enabled, the adapter automatically adds four bytes to the value of the parameter for start-of-frame lists.
BUFFER_SIZE	The TMS380C16 adapter software rounds the BUFFER_SIZE parameter up to the nearest 1K-byte boundary (such as 1024, 2048, etc.) because internal adapter buffers must reside on 1K-byte boundaries.
RESERVED	These fields are reserved and ignored by the adapter.

Table 4–40. OPEN Parameter Block Field Descriptions (Continued)

Field	Definition
TRANSMIT_BUFFER _MINIMUM_COUNT	This byte parameter contains the number of adapter buffers that are to be reserved as transmit buffers. These buffers are reserved for transmit only and are never used for receive. If zero is specified, no buffers are reserved for transmit. The minimum transmit buffer count must be less than or equal to the TRANSMIT_BUFFER_MAXIMUM_COUNT.
TRANSMIT_BUFFER _MAXIMUM_COUNT	This byte parameter contains the maximum number of adapter buffers that are to be used for transmit. A minimum of two buffers must be available for receive. If zero, a default value of six is used. The product of TRANSMIT_BUFFER_MAXIMUM_COUNT and BUFFER_SIZE–8 determines the largest frame that the adapter can transmit.
PRODUCT_ID_ADDRESS	This 32-bit field contains a 32-bit address of the attached system product ID. Eighteen bytes are read, starting from the location specified during the OPEN command processing. After the OPEN command is complete, these 18 bytes in attached system memory may be released for other purposes. The bytes read are included in the product ID subvector of the attachment MAC frame. This frame is transmitted in response to the request station attachment MAC frame.
SAP_MAX	This parameter specifies the maximum number of SAPs that may be open at any one time. If set to zero, a default of 2 is used.
STATION_MAX	This parameter specifies the maximum number of link stations that can be open at any one time. This number reflects link stations opened for all SAPs. If set to zero, a default of 6 is used.
GROUP_SAP_MAX	This parameter specifies the maximum number of group service access points that can be open at any one time. This parameter must be less than or equal to the SAP_MAX parameter value.
GSAP_MEMBER_MAX	This parameter specifies the maximum number of service access points that can be assigned to a group. The value specified in this field must be less than or equal to the value of SAP_MAX.

Table 4–40. OPEN Parameter Block Field Descriptions (Continued)

Field	Definition
TIMER_T1_1	This parameter specifies the number of 40-ms clock periods that make up a group 1 (short) response timer period. If the value input is zero, then the default value of five is used.
TIMER_T2_1	This parameter specifies the number of 40-ms clock periods that make up a group 1 (short) receive acknowledge timer period. If the value input is zero, then the default value of one is used.
TIMER_TI_1	This parameter specifies the number of 40–ms clock periods that make up a group 1 (short) inactive period. If the value is zero, then the default value of 25 is used.
TIMER_T1_2	This parameter specifies the number of 40-ms clock periods that make up a group 2 (long) response timer period. If the value input is zero, then the default value of 25 is used.
TIMER_T2_2	This parameter specifies the number of 40-ms clock periods that make up a group 2 (long) receive acknowledge timer period. If the value input is zero, then the default value of 10 is used.
TIMER_TI_2	This parameter specifies the number of 40-ms clock periods that make up a group 2 (long) inactivity timer period. If the value input is zero, then the default value of 125 is used.
MAX_FRAME_SIZE	Maximum number of bytes in a frame (AC through information) that can be transmitted by the adapter. The value must be at least 15.

Completion Upon completion of the OPEN command, a COMMAND.STATUS interrupt is posted and the SSB updated with command completion information. The OPEN command code is loaded into SSB_CMD, and the OPEN_COMPLETION code is loaded into SSB_PARM_0. The bit definitions of the OPEN_COMPLETION field are listed in Table 4–41.

Note:

Refer to the MAC OPEN command for a discussion of adapter buffer management and buffer allocation.

Table 4–41. OPEN_COMPLETION Bit Field Definitions

Bit	Definition
0	OPEN_SUCCESSFUL: Bit 0 is set to one if the OPEN command completed successfully. All other bits are set to zero.
1	NODE_ADDRESS_ERROR: Bit 1 is set to one if an error was detected in the NODE_ADDRESS field of the open parameters, if the BIA was read as all zeros, or if there is no BIA, and the node address was read as all zeros.
2	LIST_SIZE_ERROR: Bit 2 is set to one if the RECEIVE_LIST_SIZE and/or the TRANSMIT_LIST_SIZE are not equal to 0, 14, 20, or 26.
3	BUFFER_SIZE_ERROR: Bit 3 is set to one if fewer than two buffers of the specified size can be created from available adapter memory.
4	RESERVED.
5	TRANSMIT_BUFFER_COUNT_ERROR: Bit 5 is set to one if the number of receive buffers (total number of buffers minus the TRANSMIT_BUFFER_MAXIMUM_COUNT) is less than two.
6	OPEN_ERROR: Bit 6 is set to one if an error is detected during insertion onto the ring. Bits 8—15 in the SSB_PARM_0 field can be read to determine the cause of an error. Bits 8—15 are effectively divided into 4-bit entities. The first 4-bit field, open phase, is set to the OPEN command processing phase in which the error defined in the second 4-bit field occurred. The second 4-bit field, open error code, is set to an error code that reflects the ring-related error that occurred during OPEN command processing. The open phases and open error codes are listed in Table 4–33 and Table 4–34 describes the error codes.
7	LLC_OPEN_ERROR: Bit 7 is set to one if an invalid open option is selected or an error is detected during the LLC processing portion of the OPEN command. Bits 8—15 determine the error, as shown in Table 4–44.

Table 4–42. OPEN Phases and OPEN Error Codes

Bits				
8	**9**	**10**	**11**	**OPEN PHASES**
0	0	0	1	Lobe media test
0	0	1	0	Physical insertion
0	0	1	1	Address verification
0	1	0	0	Participation in ring poll
0	1	0	1	Request initialization
Bits				
12	**13**	**14**	**15**	**OPEN ERROR CODES**
0	0	0	1	Function failure
0	0	1	0	Signal loss
0	1	0	1	Timeout
0	1	1	0	Ring failure
0	1	1	1	Ring beaconing
1	0	0	0	Duplicate node address
1	0	0	1	Request initialization
1	0	1	0	Remove received

Table 4–43. OPEN Error Code Descriptions

Error Code	Definition
FUNCTION_FAILURE	This code is returned when the adapter is unable to transmit to itself while wrapped through its lobe at the wiring concentrator, or if any MAC frames are received before physical insertion.
SIGNAL_LOSS	This code is returned if a signal loss condition is detected at the adapter receiver input during the open process (when either wrapped or inserted onto the ring).
TIMEOUT	This code is returned if the adapter fails to logically insert onto the ring before the insertion timer expires. Each phase of the insertion process must complete before expiration of the 18-second insertion timer.
RING_FAILURE	This code is returned if the adapter times out when attempting a ring purge after becoming the active monitor: that is, when the adapter is unable to receive its own ring purge MAC frames.
RING_BEACONING	This code is returned if the adapter receives a beacon MAC frame after physically inserting into the ring.
DUPLICATE_NODE _ADDRESS	This code is returned if the adapter finds that another station on the ring already has the address that the adapter wishes to use.
REQUEST _PARAMETERS	This code is returned if the adapter determines that a Ring Parameter Server (RPS) is present on the ring but does not respond to a request initialization MAC frame. (If no RPS is present, the adapter will not return this code.)
REMOVE_RECEIVED	This code is returned if the adapter receives a remove adapter MAC frame during the insertion process.

Table 4–44. LLC OPEN Error Codes

Bits 8—15	LLC OPEN Error
>06	Invalid LLC OPEN options. Bit 13—15 must be clear (0).
>30	Inadequate receive buffers to open. The MAX_SAP, MAX_STATION, and TRANSMIT_BUFFER_MINIMUM_COUNT parameters are too large for two buffers to be left for receive.
>00	Invalid OPEN options.

TRANSMIT Operation (LLC)

The TMS380 adapter with LLC supports the transmission of seven different types of frames. These are listed below, each with a brief description.

Direct Frame. The direct frame is used to transmit data from the direct station. All STATION_IDs other than a direct station are invalid for this frame. For this frame, the entire frame must be prepared by the attached system. MAC frames or LLC frames may be transmitted with this type of frame. This is the only way that MAC frames may be transmitted.

UI Frame. The UI frame is used to transmit an LLC unnumbered information frame for a SAP. This frame is an unacknowledged data frame sent by LLC. The attached system must build the MAC header and data portions of the frame. The adapter builds the LLC header from information provided in the STATION_ID and REMOTE_SAP fields of the transmit list. There is no need to account for or save space for the LLC header.

XID Command. This frame is an LLC XID command frame with the poll bit set to one. The attached system provides the MAC header. If the XID_HANDLER bit (bit 4) of the SAP_OPTIONS on the OPEN.SAP for this SAP was set to one, then the attached system must also provide the data field. The adapter provides the LLC header, and, if the XID_HANDLER bit was set to zero, the data field. If the attached system wishes to add a data field, it must leave the first three bytes of that field empty for the adapter to fill in. The attached system must always have at least three bytes in the data field. The user does not account for or save space for the LLC header. The adapter builds the LLC header from information provided in the STATION_ID and REMOTE_SAP fields of the transmit list.

XID Response Final. This is an LLC XID response frame with the final bit set to one. This frame should be transmitted only by those SAPs that were opened with the XID_HANDLER bit on. Otherwise, the adapter does not prevent this frame from being transmitted. The attached system provides the MAC header and data field. The adapter provides the LLC header. The adapter builds the LLC header from information provided in the STATION_ID and REMOTE_SAP fields of the transmit list. There is no need to account for or save space for the LLC header.

XID Response Not Final. This is an LLC XID response frame with the final bit set to zero. This frame should be transmitted only by those SAPs that were opened with the XID_HANDLER bit on. The attached system provides the MAC header and data field. The adapter provides the LLC header. The adapter builds the LLC header from information provided in the STATION_ID and REMOTE_SAP fields of the transmit list. There is no need to account for or save space for the LLC header.

Test Command. This frame is an LLC test command frame with the poll bit set to one. The attached system provides the MAC header and optional data field. The adapter builds the LLC header from information provided in the STATION_ID and REMOTE_SAP fields of the transmit list. There is no need to account for or save space for the LLC header.

I-Frame. This frame is used to transmit data for an LLC type 2 connection. This frame is transmitted only by a link station. The attached system provides only the data field. The adapter provides the MAC and LLC headers. The adapter builds the LLC header from information provided in the STATION_ID field of the transmit list. There is no need to account for or save space for the MAC or LLC headers. This frame is an acknowledged information frame. The adapter handles waiting for a response frame and will automatically retry or request retries (based on the setting of the host-based I-frame transmit bit in the OPEN adapter options) to guarantee the delivery of data in an I-frame. On reception of an I-frame, the adapter also handles the transmission of the acknowledgement frame.

The LLC interface supports two separate types of I-frame transmission: host-based I-frames and adapter-based I-frames. Under adapter-based I-frame transmission, the TMS380 LLC software accepts I-frames from the host system and holds these frames internally in the adapter until they are acknowledged by the remote system. However, there are cases in which, under heavy loading and unreliable data transfer ability over the network, this transmit and hold technique can cause the adapter to congest and potentially lose links. To remedy this situation, the host-based transmit I-frame option forces the adapter to request frames on demand from the host system whenever it needs to transmit them onto the network. It can be seen that, for cases where there is unreliable data transfer, the adapter could request an I-frame from the host system several times before the frame is actually acknowledged by the remote link station. This transmit I-frame mechanism frees the previously dedicated adapter memory and removes the risk of receive congestion and link errors. To implement the host-based I-frame transmit option, you must set the host-based transmit I-frame bit (bit 12) in the open adapter options.

Command TRANSMIT (Host-Based I-Frame)

Description This TRANSMIT command is used to transmit all frame types and I-frames using the host-based technique. Some frames are built entirely in the attached system using the logical frame format described in Chapter 2 and are moved from attached system memory to adapter internal buffers, via DMA, upon initiation of this command. Other frames are built partially by the attached system and completed by the adapter after DMA completes. For I-frame transmission, each frame needs to be initiated by issuing a TRANSMIT.I.FRAME.REQUEST command.

A COMMAND.REJECT interrupt is posted if the adapter has not been opened, if there is already an executing TRANSMIT command, or if the address passed in the SCB_PARM_0 and SCB_PARM_1 is not aligned on a word boundary.

Initiation

	Byte + 0 Byte + 1
SCB_CMD	>0004
SCB_PARM_0	XMIT_PARM_BLOCK (High)
SCB_PARM_1	XMIT_PARM_BLOCK (Low)

Completion

SSB_CMD	>0004
SSB_PARM_0	XMIT_STATUS
SSB_PARM_1	XMIT_PARM_LIST (High)
SSB_PARM_2	XMIT_PARM_LIST (Low)

Initiation The SCB_PARM_0 and SCB_PARM_1 contain a 32-bit word-aligned address that points to a transmit parameter list. The format of the transmit parameter list is shown in Figure 4–33, and the definitions of each field are given in Table 4–45. Transmit parameter lists must be aligned on even word boundaries. Note that one transmit parameter list cannot be used to transmit more than one frame, but several transmit parameter lists can be used to transmit a single frame.

Figure 4–33. TRANSMIT Parameter List

		Byte + 0	Byte + 1
XMIT_PARM_LIST	+ 0	FORWARD_POINTER (High)	
	+ 2	FORWARD_POINTER (Low)	
	+ 4	TRANSMIT_CSTAT	
	+ 6	FRAME_SIZE	
	+ 8	DATA_COUNT	
	+ 10	DATA_ADDRESS (High)	
	+ 12	DATA_ADDRESS (Low)	
	+ 14	DATA_COUNT	
	+ 16	DATA_ADDRESS (High)	
	+ 18	DATA_ADDRESS (Low)	
	+ 20	DATA_COUNT	
	+ 22	DATA_ADDRESS (High)	
	+ 24	DATA_ADDRESS (Low)	
	+ 26	STATION_ID	
	+ 28	REMOTE_SAP	HEADER_LENGTH or FRAME_CORRELATOR

Table 4–45. TRANSMIT Parameter List Fields

Field	Definition
FORWARD_POINTER	This 32-bit field contains a 32-bit pointer to the next transmit parameter list in the chain. When the pointer is odd, the current transmit parameter list is the last in the chain. The adapter continues to process transmit parameter lists until it reads an odd FORWARD_POINTER, at which time the adapter waits for the last frame (list with ODD address) to be transmitted before exiting the command. If the system updates the FORWARD_POINTER before the transmission of the last frame is completed, the adapter continues to process transmit parameter lists. The system must update the FORWARD_POINTER from the most significant byte to the least significant byte to ensure that the address is valid before changing to an even address. The FORWARD_POINTER should not be initialized to point to itself; this could cause problems because of the pipelined nature of the adapter's list processing. The adapter does not alter this parameter.
TRANSMIT_CSTAT	This 16-bit parameter is set by the attached system when the transmit parameter list is created. It is overwritten by the adapter to report frame completion status. When initially set by the attached system, this parameter field is referred to as the TRANSMIT_CSTAT_REQUEST field. After a frame completes transmission, the adapter writes TRANSMIT_CSTAT_COMPLETE to the list that has START_FRAME set to one. These bits indicate the completion status of the frame, not the TRANSMIT command itself. **Note that the system must update the least significant byte prior to the most significant byte to ensure the frame type is set before the valid bit is set.**

Table 4–45. TRANSMIT Parameter List Fields (Continued)

Field	Definition	
TRANSMIT_ CSTAT_ REQUEST	Bit 0	VALID: The adapter waits for bit 0 to be set to a one before processing the current transmit parameter list. The attached system must issue a TRANSMIT.VALID interrupt request when changing bit 0 from zero to one. This bit is ignored unless the list is an anticipated start of frame (follows the end of frame or is first list of command).
	Bit 1	FRAME_COMPLETE: Must be reset to zero.
	Bit 2	START_FRAME: Must be set to one for a list that defines the start of a frame.
	Bit 3	END_FRAME: Must be set to one for a list that defines the end of a frame.
	Bit 4	FRAME_INTERRUPT: If this bit is set to one, an adapter-to-attached-system interrupt is generated when the frame has been transmitted, rather than after all frames on the chain have been transmitted. This bit is ignored unless START_ FRAME (bit 2) is a one.
	Bit 5	RESERVED: This bit must be reset to zero.
	Bit 6	PASS_CRC: When this bit is set to one, the adapter assumes the CRC to be transmitted with the frame contained in the last four bytes of the frame data. In this case, the adapter does not generate the CRC on transmit but uses the CRC passed with frame data. Note that the CRC value is not checked. This option can be used by adapters, such as MAC-layer bridges, that must forward frames without altering the CRC. This bit is ignored unless the START_FRAME bit (bit 2) is set to one.
	Bit 7	PASS_SOURCE_ADDRESS: When this bit is set to one, the adapter transmits the frame with the source address as given by the host. The adapter does not overwrite the source address with the adapter node address. This option can be used by adapters, such as MAC-layer bridges, that must forward frames without altering the source address. This bit is ignored unless the START_FRAME bit (bit 2) is set to one.
	Bits 8—10	FRAME_TYPE: This field is used to indicate the type of frame to be transmitted. The frame types that may be transmitted are listed in Table 4–46.
	Bits 11—15	RESERVED: These bits must be reset to zero.

Table 4–45. TRANSMIT Parameter List Fields (Continued)

Field	Definition
TRANSMIT_ CSTAT_ COMPLETE	The completion code for the transmitted frame is written to this field in the list that has START_FRAME bit equal to one when the adapter has completed transmission of the frame. CSTATs that are not in a list defining the start of a frame are not altered by the adapter. The TRANSMIT_CSTAT_COMPLETE bit definitions are shown below. Bit 0 VALID: Reset to zero. Bit 1 COMPLETE: Set to one. Bit 2 FRAME_START: Bit 2 has the same value as specified by the attached system in TRANSMIT_CSTAT_REQUEST. Bit 3 FRAME_END: Bit 3 has the same value specified by the attached system in TRANSMIT_ CSTAT_REQUEST. Bit 4 INTERRUPT_FRAME: Bit 4 has the same value specified by the attached system in TRANSMIT_CSTAT_REQUEST. Bit 5 TRANSMIT_ERROR: Bit 5 is set to one if the frame transmit or strip process was in error, or if a nonfatal error occurred in processing the transmit list. A nonfatal error does not halt list processing. If this bit is set to one, then bits 8—15 contain an error reason code instead of the stripped FS byte. Bit 6 PASS_CRC: This bit reflects the state of bit 6 in the TRANSMIT_CSTAT_REQUEST. This bit is valid only if the START_FRAME bit (bit 2) is set to one. Bit 7 PASS_SOURCE_ADDRESS: This bit reflects the state of bit 7 in the TRANSMIT_CSTAT_REQUEST. This bit is valid only if the START_ FRAME bit (bit 2) is set to one. Bits 8—15 STRIP_FS: If TRANSMIT_ERROR (bit 5) is set to zero, then this field contains a copy of the FS byte returned when the transmitted frame is stripped off the ring. For LLC enabled only, if the frame should be copied by this adapter and the received FS byte was >00, the STRIP_FS byte reported in this field will be >CC. If TRANSMIT _ERROR is set to one, this field contains an error reason code as shown in Table 4–47, with applicability to I frames as described.

Table 4–45. TRANSMIT Parameter List Fields (Continued)

Field	Definition
FRAME_SIZE	This 16-bit field contains the number of bytes to be transferred from the attached system for this frame. The FRAME_SIZE value includes all data provided by the attached system. FRAME_SIZE does not include headers and/or trailers appended by the adapter. This parameter is valid only in the transmit parameter list that has the START_FRAME bit set to one. FRAME_SIZE must, however, be nonzero in all lists. The adapter does not alter this parameter.
DATA_COUNT	This 16-bit field contains the number of bytes to be read at the address defined by DATA_ADDRESS. There can be a maximum of three DATA_COUNT/DATA_ADDRESS parameters per transmit parameter list. If bit 0 is zero, then this DATA_COUNT is the last DATA_COUNT in the transmit parameter list. Bit 0 of the third DATA_COUNT is ignored. A DATA_COUNT of 0 is permitted (with or without bit 0 set). The sum of the used DATA_COUNT parameters must equal the FRAME_SIZE specified in the list that has start_frame set to one. The DATA_COUNT can be even or odd. The adapter does not alter this parameter.
DATA_ADDRESS	This 32-bit field contains a 32-bit pointer to a portion of (or the entire) logical frame residing in attached system memory. DATA_ADDRESS may be even or odd.
STATION_ID	This field contains the STATION_ID of the SAP or link station from which the frame is to be transmitted. This value is valid only in start-of-frame lists.
REMOTE_SAP	This parameter defines the SAP value at the destination station to which the frame is to be sent. This parameter is not used for I-frame transmission. This value is valid only in start-of-frame lists.
HEADER_LENGTH or FRAME_CORRELATOR	This field is ignored for direct frame requests. This parameter defines the length of the MAC header, including any routing information, for all type-1 frame requests. The source routing bit in the source address field is automatically set if this value is greater than 14. For transmit I-frame requests, this parameter defines the frame correlator assigned to this I-frame by the TRANSMIT.I.FRAME.REQUEST command.

Table 4–46. TRANSMIT Frame Types

TRANSMIT_CSTAT Request			**Frame Type**
Bit 8	Bit 9	Bit 10	
0	0	0	Direct Frame
0	0	1	Ui Frame
0	1	0	XID Command
0	1	1	XID Response Final
1	0	0	XID Response Not Final
1	0	1	Test Command
1	1	0	I-Frame
1	1	1	Invalid Frame Type

Table 4–47. Transmit Error/Link Status Codes

Error Code	**Explanation**
>00	No error. This code is returned only for I-frame requests. It is returned if there is no error in the request, but the link state for the requested link station does not permit the frame to be sent. This error code requires no action (when necessary, the adapter requests the frame from the host again).
>08	Unauthorized access priority.
>23	Error in frame transmit or strip. For I-frames, the frame is automatically rerequested, and no action should be taken on this error.
>24	Unauthorized MAC frame.
>26	Invalid frame correlator. This is not necessarily a fatal error code. If link activity causes the link to set back its send sequence numbers since it last requested an I-frame from the host, this error code results. Under normal circumstances, the error code should be ignored.
>28	Invalid transmit frame length. If this error occurs on an I-frame, the adapter automatically places the link in a disconnected state.
>40	Invalid STATION_ID for the frame type specified.
>41	Protocol error. The link station is not in the proper state for transmitting I-frames.
>44	Invalid routing information length.
>F0	Invalid frame type (111).

Execution The attached system can create a circular chain of transmit parameter lists by setting the FORWARD_POINTER of the last transmit parameter list to the address of the first list. The valid bit of TRANSMIT_CSTAT_REQUEST is manipulated to initiate and suspend frame transmission. When the adapter reads a list with the FRAME_START bit set to one and the valid bit reset to zero, it suspends processing until a TRANSMIT.VALID interrupt request is issued by the attached system. The attached system is not notified of a transmit suspended condition by the adapter.

If a fixed transmit chain technique is utilized and more than one list is used to transmit a single frame, lists that do not have the FRAME_START bit set should have the valid bit reset to zero because the adapter does not alter the TRANSMIT_CSTAT field for lists that do not have the FRAME_START bit set. Revalidating of the start of frame list also releases the remaining frame lists if the valid bits were initially set.

A circular chain of one or two frames should not be used, because of the pipelined nature of transmit parameter list processing. An implementation of this nature can cause the adapter to send the same frame twice. A circular chain should contain enough lists to hold two entire frames plus an extra list.

Since transmit lists may be added dynamically to the transmit parameter list chain, a test should be made, following a TRANSMIT.STATUS interrupt, to determine if the adapter has processed all frames that the attached system has placed in the chain. If frames have been added to the chain subsequent to the TRANSMIT command completion, another TRANSMIT command should be executed with the SCB_PARM_0 and SCB_PARM_1 set to the address of the first valid list.

A chain of transmit lists can contain multiple frames. These frames can be of any type and can be of multiple types.

When the last list in a chain contains an odd FORWARD POINTER, the adapter performs the following operations in order:

1) Updates the CSTAT field of the last start of frame list.
2) Posts the FRAME COMPLETE status if requested.
3) Rereads the FORWARD POINTER field of the list containing the odd FORWARD POINTER.
4) Posts a COMMAND COMPLETE status if FORWARD POINTER is still odd.

Because the adapter rereads the FORWARD POINTER after posting the FRAME COMPLETE status, the host software should not modify the

FORWARD POINTER of the last list except to add another transmit list to the chain. Therefore, this list containing the odd FORWARD POINTER should not be considered free until one of the following occurs:

❏ A COMMAND COMPLETE status is received for this list, or

❏ Any transmit status (such as FRAME COMPLETE, COMMAND COMPLETE, or LIST ERROR) occurs, specifying a list further down the chain.

Completion A TRANSMIT.STATUS interrupt is generated when one of the following conditions occurs during processing of the TRANSMIT command:

❏ All the frames specified by the transmit parameter list chain have been transmitted.

❏ A TRANSMIT.HALT command has been issued and completed.

❏ A frame, other than an I-frame, that had the FRAME_INTERRUPT bit set in TRANSMIT_CSTAT has been transmitted.

❏ The FRAME_INTERRUPT bit is set, and an I-frame has been DMAed into the adapter and is being held.

❏ An I-frame has been acknowledged by the remote station.

❏ A fatal list error is detected.

❏ The adapter is ready to DMA another I-frame.

Prior to issuing the TRANSMIT.STATUS interrupt, the adapter updates SSB_PARM_0 with the XMIT.STATUS completion code and the SSB_PARM_1 and SSB_PARM_2 with the address of the last transmit parameter list processed by the adapter if bits 0, 1, or 2 are set. If bits 3 or 4 of XMIT_STATUS are set, the SSB_PARM_1 is updated with the related STATION_ID, and SSB_PARM_2 has no meaning. The bit definitions of XMIT_STATUS are listed in Table 4–48.

Table 4–48. XMIT_STATUS Bit Definitions

Bit	Definition
0	COMMAND_COMPLETE: Set to one to indicate that the TRANSMIT command has completed. The system must issue another TRANSMIT command to transmit additional frames. SSB_PARM_1 and SSB_PARM_2 contain a pointer to the transmit parameter list of the last frame transmitted. This bit is also set as a result of a TRANSMIT_HALT command. The COMMAND_COMPLETE and FRAME_COMPLETE bits are not set at the same time.
1	FRAME_COMPLETE: When set to one, this bit indicates that a frame has been transmitted and the INTERRUPT_FRAME bit was set in TRANSMIT_CSTAT_REQUEST. Since frames in the transmit parameter list chain may be transmitted faster than the system can respond to the interrupts and/or faster than the adapter can cause the interrupts, this bit can report the completion of more than one frame at a time. SSB_PARM_1 and SSB_PARM_2 contain a pointer to the transmit parameter list of the last frame transmitted. If lists with the FRAME_INTERRUPT bit set are intermixed with lists that do not have the FRAME_INTERRUPT bit set, FRAME_COMPLETE can include frames that did not have FRAME_INTERRUPT set. The attached system should check the TRANSMIT_ERROR bit (bit 5) of the CSTAT in the returned list for possible non-fatal errors.
2	LIST_ERROR: Bit 2 is set to one if there is an error in one of the lists that compose the frame. Bits 8—13 define the error. The TRANSMIT command is terminated and the system must issue another TRANSMIT command to continue transmission. SSB_PARM_1 and SSB_PARM_2 contain a pointer to the list that starts the frame with errors. The TRANSMIT.STATUS interrupt for LIST_ERROR is not generated until the status of every other transmit has been posted. The CSTATs of lists found to be in error are not altered by the adapter. Neither the FRAME_COMPLETE nor COMMAND_COMPLETE bit is set with LIST_ERROR.
3	I_FRAME_ACK: Bit 3 is set to one to indicate that an I-frame has been acknowledged by the remote station. When this bit is set, bits 8—15 indicate the number of I-frames that were acknowledged. When this bit is set, the SSB_PARM_1 indicates the STATION_ID of the link station associated with this SSB, and SSB_PARM_2 has no meaning.

Table 4–48. XMIT_STATUS Bit Definitions (Continued)

Bit	Definition
4	READY: Bit 4 is used to indicate at what time a length station is ready to accept another I-frame from the attached system. If this bit is set to 1, then the length station is requesting that the attached system build a transmit list chain for the I-frame whose correlator matches the correlator in the most significant byte of SSB_PARM_2. When this bit is set, SSB_PARM_1 indicates the STATION_ID of the length station associated with this SSB, the most significant byte of SSB_PARM_2 is set to the FRAME_CORRELATOR of the next I-frame to be sent by this length station in the least significant byte of SSB_PARM_2 is reserved.
5—7	RESERVED: Reset to zero.
8	FRAME_SIZE_ERROR: Bit 8 is set to one if FRAME_SIZE does not equal the sum of the valid DATA_COUNT fields, if FRAME_SIZE is less than the required header plus one byte of information field (15 bytes plus routing field), or if FRAME_SIZE was specified as zero in any list, except in lists that define I-frames.
9	TRANSMIT_THRESHOLD: Bit 9 is set to one if FRAME_SIZE is greater than the product of (BUFFER_SIZE – 8) and the TRANSMIT_BUFFER_MAXIMUM_COUNT parameters from the OPEN command.
10	ODD_ADDRESS: Bit 10 is set to one if an odd FORWARD_POINTER value is read on a list that does not have END_FRAME set to 1.
11	FRAME_ERROR: Bit 11 is set to one if the START_FRAME bit is set to one on a list that is not an anticipated start of frame or if the START_FRAME bit is zero on an anticipated start of frame.
12—13	RESERVED: Reset to zero.
14	ILLEGAL_FRAME_FORMAT: Bit 14 is set to one if bit 0 of the FC field was set to one. This bit is set only if the LLC Interface is not enabled. If the LLC interface has been enabled, this error is a nonfatal list error and is indicated in the CSTAT of the associated list.
15	RESERVED: Reset to zero.

LLC 0004 TRANSMIT (Adapter-Based I-Frame) Command

Command TRANSMIT (Adapter-Based I-Frame)

Description This TRANSMIT command is used to transmit all frame types and I-frames using the adapter-based technique. Some frames are built entirely in the attached system using the logical frame format described in Chapter 2 and are moved from attached system memory to adapter internal buffers, via DMA, upon initiation of this command. Other frames are built partially by the attached system and completed by the adapter after DMA completes.

A COMMAND.REJECT interrupt is posted if the adapter has not been opened, if there is already an executing TRANSMIT command, or if the address passed in the SCB_PARM_0 and SCB_PARM_1 is not aligned on a word boundary.

Initiation

	Byte + 0	Byte + 1
SCB_CMD	>0004	
SCB_PARM_0	XMIT_PARM_BLOCK (High)	
SCB_PARM_1	XMIT_PARM_BLOCK (Low)	

Exit

SSB_CMD	>0004
SSB_PARM_0	XMIT_STATUS
SSB_PARM_1	XMIT_PARM_LIST (High)
SSB_PARM_2	XMIT_PARM_LIST (Low)

Initiation The SCB_PARM_0 and SCB_PARM_1 contain a 32-bit word-aligned address that points to a transmit parameter list. The format of the transmit parameter list is shown in Figure 4–34, and the definitions of each field are given in Table 4–49. Transmit parameter lists must be aligned on even word boundaries.

Figure 4–34. TRANSMIT Parameter List

		Byte + 0	Byte + 1
XMIT_PARM_LIST	+ 0	FORWARD_POINTER (High)	
	+ 2	FORWARD_POINTER (Low)	
	+ 4	TRANSMIT_CSTAT	
	+ 6	FRAME_SIZE	
	+ 8	DATA_COUNT	
	+ 10	DATA_ADDRESS (High)	
	+ 12	DATA_ADDRESS (Low)	
	+ 14	DATA_COUNT	
	+ 16	DATA_ADDRESS (High)	
	+ 18	DATA_ADDRESS (Low)	
	+ 20	DATA_COUNT	
	+ 22	DATA_ADDRESS (High)	
	+ 24	DATA_ADDRESS (Low)	
	+ 26	STATION_ID	
	+ 28	REMOTE_SAP	HEADER_LENGTH

Table 4–49. TRANSMIT Parameter List Fields

Field	Definition
FORWARD_POINTER	This 32-bit field contains a 32-bit pointer to the next transmit parameter list in the chain. When the pointer is odd, the current transmit parameter list is the last in the chain. The adapter continues to process transmit parameter lists until it reads an odd FORWARD_POINTER, at which time the adapter waits for the last frame (list with ODD address) to be transmitted before exiting the command. If the system updates the FORWARD_POINTER before the transmission of the last frame is completed, the adapter continues to process transmit parameter lists. The system must update the FORWARD_POINTER from the most significant byte to the least significant byte to ensure that the address is valid before changing to an even address. The FORWARD_POINTER should not be initialized to point to itself; this could cause problems because of the pipelined nature of the adapter's list processing. The adapter does not alter this parameter.
TRANSMIT_CSTAT	This 16-bit parameter is set by the attached system when the transmit parameter list is created. It is overwritten by the adapter to report frame completion status. When initially set by the attached system, this parameter field is referred to as the TRANSMIT_CSTAT_REQUEST field. After a frame completes transmission, the adapter writes TRANSMIT_CSTAT_COMPLETE to the list that has START_FRAME set to one. These bits indicate the completion status of the frame, not the TRANSMIT command itself. **Note that the system must update the least significant byte prior to the most significant byte to ensure the frame type is set before the valid bit is set.**

Table 4–49. TRANSMIT Parameter List Fields (Continued)

Field	Definition	
TRANSMIT_ CSTAT _ REQUEST	Bit 0	VALID: The adapter waits for bit 0 to be set to a one before processing the current transmit parameter list. The attached system must issue a TRANSMIT.VALID interrupt request when changing bit 0 from zero to one. This bit is ignored unless the list is an anticipated start of frame (follows the end of frame or is first list of command).
	Bit 1	FRAME_COMPLETE: Must be reset to zero.
	Bit 2	START_FRAME: Must be set to one for a list that defines the start of a frame.
	Bit 3	END_FRAME: Must be set to one for a list that defines the end of a frame.
	Bit 4	FRAME_INTERRUPT: If this bit is set to one, an adapter-to-attached-system interrupt is generated when the frame has been transmitted, rather than after all frames on the chain have been transmitted. This bit is ignored unless START_ FRAME (bit 2) is a one.
	Bit 5	RESERVED: This bit must be reset to zero.
	Bit 6	PASS_CRC: When this bit is set to one, the adapter assumes the CRC to be transmitted with the frame is contained in the last four bytes of the frame data. In this case, the adapter does not generate the CRC on transmit but uses the CRC passed with frame data. Note that the CRC value is not checked. This option can be used by adapters, such as MAC-layer bridges, that must forward frames without altering the CRC. This bit is ignored unless the START_FRAME bit (bit 2) is set to one.
	Bit 7	PASS_SOURCE_ADDRESS: When this bit is set to one, the adapter transmits the frame with the source address as given by the host. The adapter does not overwrite the source address with the adapter node address. This option can be used by adapters, such as MAC-layer bridges, that must forward frames without altering the source address. This bit is ignored unless the START_FRAME bit (bit 2) is set (1).
	Bits 8—10	FRAME_TYPE: This field is used to indicate the type of frame to be transmitted. These bits must be set to zero unless the LLC interface has been enabled. The frame types that may be transmitted are listed in Table 4–50.
	Bits 11—15	RESERVED: These bits must be reset to zero.

Table 4–49. TRANSMIT Parameter List Fields (Continued)

Field	Definition
TRANSMIT_ CSTAT_ COMPLETE	The completion code for the transmitted frame is written to this field in the list that has START_FRAME bit equal to one when the adapter has completed transmission of the frame. CSTATs that are not in a list defining the start of a frame are not altered by the adapter. The TRANSMIT_CSTAT_COMPLETE bit definitions are shown below. Bit 0 VALID: Reset to zero. Bit 1 COMPLETE: Set to one. Bit 2 FRAME_START: Bit 2 has the same value as specified by the attached system in TRANSMIT_CSTAT_REQUEST. Bit 3 FRAME_END: Bit 3 has the same value as specified by the attached system in TRANSMIT_ CSTAT_REQUEST. Bit 4 INTERRUPT_FRAME: Bit 4 has the same value as specified by the attached system in TRANSMIT_CSTAT_REQUEST. Bit 5 TRANSMIT_ERROR: Bit 5 is set to one if the frame transmit or strip process was in error, or if a nonfatal error occurred in processing the transmit list. A nonfatal error does not halt list processing. If this bit is set to one, then bits 8—15 contain an error reason code instead of the stripped FS byte. List processing continues. Bit 6 PASS_CRC: This bit reflects the state of bit 6 in the TRANSMIT_CSTAT_REQUEST. This bit is valid only if the START_FRAME bit (bit 2) is set to one. Bit 7 PASS_SOURCE_ADDRESS: This bit reflects the state of bit 7 in the TRANSMIT_CSTAT_REQUEST. This bit is valid only if the START_FRAME bit (bit 2) is set to one. Bits 8—15 STRIP_FS: If TRANSMIT_ERROR (bit 5) is set to zero, then this field contains a copy of the FS byte returned when the transmitted frame is stripped off the ring. For LLC enabled only, if the frame should be copied by this adapter and the received FS byte was >00, the STRIP_FS byte reported in this field will be >CC. If TRANSMIT_ERROR is set to one, this field contains an error reason code as shown in Table 4–51, with applicability to I frames as described.

Table 4–49. TRANSMIT Parameter List Fields (Continued)

Field	Definition
FRAME_SIZE	This 16-bit field contains the number of bytes to be transferred from the attached system for this frame. The FRAME_SIZE value includes all data provided by the attached system. FRAME_SIZE does not include headers and/or trailers appended by the adapter. This parameter is valid only in the transmit parameter list that has the START_FRAME bit set to one. FRAME_SIZE must, however, be nonzero in all lists. The adapter does not alter this parameter.
DATA_COUNT	This 16-bit field contains the number of bytes to be read at the address defined by DATA_ADDRESS. There can be a maximum of three DATA_COUNT/DATA_ADDRESS parameters per transmit parameter list. If bit 0 is zero, then this DATA_COUNT is the last DATA_COUNT in the transmit parameter list. Bit 0 of the third DATA_COUNT is ignored. A DATA_COUNT of 0 is permitted (with or without bit 0 set). The sum of the used DATA_COUNT parameters must equal the FRAME_SIZE specified in the list that has START_FRAME set to one. The DATA_COUNT can be even or odd. The adapter does not alter this parameter.
DATA_ADDRESS	This 32-bit field contains a 32-bit pointer to a portion of (or the entire) logical frame residing in attached system memory. DATA_ADDRESS may be even or odd.
STATION_ID	This field contains the STATION_ID of the SAP or link station from which the frame is to be transmitted. This value is valid only in start-of-frame lists.
REMOTE_SAP	This parameter defines the SAP value at the destination station to which the frame is to be sent. This parameter is not used for I-frame transmission. This value is only valid in start-of-frame lists.
HEADER_LENGTH	This parameter defines the length of the MAC header, including any routing information. The source routing bit in the source address field is automatically set if this value is greater than 14. This value is not valid for I-frames.

Table 4–50. TRANSMIT Frame Types

Code			Frame Type
Bit 8	Bit 9	Bit 10	
0	0	0	Direct Frame
0	0	1	UI Frame
0	1	0	XID Command
0	1	1	XID Response Final
1	0	0	XID Response Not Final
1	0	1	Test Command
1	1	0	I-Frame
1	1	1	Invalid Frame Type

Table 4–51. Transmit Error/Link Status Codes

Error Code	Definition
>08	Unauthorized access priority.
>23	Error in frame transmit or strip. For I-frames, the frame is automatically rerequested and no action should be taken on this error.
>24	Unauthorized MAC frame.
>27	Link not transmitting I-frames. This error code is returned if an I-frame is issued before the DMA is ready.
>28	Invalid transmit frame length.
>40	Invalid STATION_ID.
>41	Protocol error.
>44	Invalid routing information length.
>F0	Invalid frame type (111).
>FE	I-frame DMA into adapter. Link now in not-ready state.
>FF	I-frame DMA. Link now ready to DMA another frame.

Execution The attached system can create a circular chain of transmit parameter lists by setting the FORWARD_POINTER of the last transmit parameter list to the address of the first list. The valid bit of TRANSMIT_CSTAT_REQUEST is manipulated to initiate and suspend frame transmission. When the adapter reads a list with the FRAME_START bit set to one and the valid bit reset to zero, it suspends processing until a TRANSMIT.VALID interrupt request is issued by the attached system. The attached system is not notified of a transmit suspended condition by the adapter.

If a fixed transmit chain technique is utilized and more than one list is used to transmit a single frame, lists that do not have the FRAME_START bit set should have the valid bit reset to zero because the adapter does not alter the TRANSMIT_CSTAT field for lists that do not have the FRAME_START bit set. Revalidating of the start of frame list also releases the remaining frame lists if the valid bits were initially set.

A circular chain of one or two frames should not be used, because of the pipelined nature of transmit parameter list processing. An implementation of this nature can cause the adapter to send the same frame twice. A circular chain should contain enough lists to hold two entire frames plus an extra list.

Since transmit lists may be added dynamically to the transmit parameter list chain, a test should be made, following a TRANSMIT.STATUS interrupt, to determine if the adapter has processed all frames that the attached system has placed in the chain. If frames have been added to the chain subsequent to the TRANSMIT command exit, another TRANSMIT command should be executed with the SCB_PARM_0 and SCB_PARM_1 set to the address of the first valid list.

A chain of transmit lists may contain multiple frames. These frames can be of any type and can be of multiple types.

When the last list in a chain contains an odd FORWARD POINTER, the adapter performs the following operations in order:

1) Updates the CSTAT field of the last start of frame list.

2) Posts the FRAME COMPLETE status if requested.

3) Rereads the FORWARD POINTER field of the list containing the odd FORWARD POINTER.

4) Posts a COMMAND COMPLETE status if FORWARD POINTER is still odd.

Because the adapter rereads the FORWARD POINTER after posting the FRAME COMPLETE status, the host software should not modify the FORWARD POINTER of the last list except to add another transmit list to the chain. Therefore, this list containing the odd FORWARD POINTER should not be considered free until one of the following occurs:

❑ A COMMAND COMPLETE status is received for this list, or

❑ Any transmit status (such as FRAME COMPLETE, COMMAND COMPLETE, or LIST ERROR) occurs, specifying a list further down the chain.

Completion A TRANSMIT.STATUS interrupt is generated when one of the following conditions occurs during processing of the TRANSMIT command:

❑ All the frames specified by the transmit parameter list chain have been transmitted.

❑ A TRANSMIT.HALT command has been issued and completed.

❑ A frame, other than an I-frame, that had the FRAME_INTERRUPT bit set in TRANSMIT_CSTAT has been transmitted.

❑ The FRAME_INTERRUPT bit is set, and an I-frame has been DMAed into the adapter and is being held.

❑ An I-frame has been acknowledged by the remote station.

❑ A fatal list error is detected.

❑ The adapter is ready to DMA another I-frame.

Prior to issuing the TRANSMIT.STATUS interrupt, the adapter updates SSB_PARM_0 with the XMIT.STATUS completion code and SSB_PARM_1/SSB_PARM_2 with the address of the last transmit parameter list processed by the adapter if bits 0, 1, or 2 are set. If bits 3 or 4 of XMIT_STATUS are set, the SSB_PARM_1 is updated with the related STATION_ID, and SSB_PARM_2 has no meaning. The bit definitions of XMIT_STATUS are listed in Table 4–52.

Table 4–52. XMIT_STATUS Bit Definitions

Bit	Definition
0	COMMAND_COMPLETE: Set to one to indicate that the TRANSMIT command has completed. The system must issue another TRANSMIT command to transmit additional frames. SSB_PARM_1 and SSB_PARM_2 contain a pointer to the transmit parameter list of the last frame transmitted. This bit is also set as a result of a TRANSMIT_HALT command. The COMMAND_COMPLETE and FRAME_COMPLETE bits are not set at the same time.
1	FRAME_COMPLETE: When set to one, this bit indicates that a frame has been transmitted and the INTERRUPT_FRAME bit was set in TRANSMIT_CSTAT_REQUEST. Since frames in the transmit parameter list chain may be transmitted faster than the system can respond to the interrupts and/or faster than the adapter can cause the interrupts, this bit can report the completion of more than one frame at a time. SSB_PARM_1 and SSB_PARM_2 contain a pointer to the transmit parameter list of the last frame transmitted. If lists with the FRAME_INTERRUPT bit set are intermixed with lists that do not have the FRAME_INTERRUPT bit set, FRAME_COMPLETE can include frames that did not have FRAME_INTERRUPT set. The attached system should check the TRANSMIT_ERROR bit (bit 5) of the CSTAT in the returned list for possible non-fatal errors.
2	LIST_ERROR: Bit 2 is set to one if there is an error in one of the lists that compose the frame. Bits 8—13 define the error. The TRANSMIT command is terminated and the system must issue another TRANSMIT command to continue transmission. SSB_PARM_1 and SSB_PARM_2 contain a pointer to the list that starts the frame with errors. The TRANSMIT.STATUS interrupt for LIST_ERROR is not generated until the status of every other transmit has been posted. The CSTATs of lists found to be in error are not altered by the adapter. Neither the FRAME_COMPLETE nor COMMAND_COMPLETE bit is set with LIST_ERROR.
3	I_FRAME_ACK: Bit 3 is set to one to indicate that an I-frame has been acknowledged by the remote station. When this bit is set, bits 8—15 indicate the number of I-frames that were acknowledged. When this bit is set, the SSB_PARM_1 indicates the STATION_ID of the link station associated with this SSB, and SSB_PARM2 has no meaning.

Table 4–52. XMIT_STATUS Bit Definitions (Continued)

Bit	Definition
4	READY: Bit 4 is used to indicate when a link station is ready to accept I-frames from the attached system. When this bit is set, the SSB_PARM1 indicates the STATION_ID of the link station associated with this SSB, and SSB_PARM_2 has no meaning. This bit is set under either of two conditions. First, when a CONNECT.STATION command completes, the attached system assumes a not ready state and waits for a READY transmit status interrupt. This interrupt occurs when the link station is ready for I-frame transmission, whether or not a TRANSMIT command has been issued. This interrupt may occur prior to reception of a command complete interrupt for the CONNECT.STATION command. Second, after an I-frame has been DMAed into the adapter and a CSTAT completion of FE has been received, the link station is in a not-ready state. When this condition clears, a READY transmit interrupt is issued. If, after a FE completion has been issued, another I-frame for that link station is passed to the adapter, the adapter rejects the I-frame with the error code 27 (link not transmitting I-frames). Processing of the transmit lists continues, and frames for other link stations are not affected. For performance reasons, it is suggested that the adapter have a minimum of three transmit and three receive lists, plus associated buffers for each.
5—7	RESERVED: Reset to zero.
8	FRAME_SIZE_ERROR: Bit 8 is set to one if FRAME_SIZE does not equal the sum of the valid DATA_COUNT fields, if FRAME_SIZE is less than the required header plus one byte of information field (15 bytes plus routing field), or if FRAME_SIZE was specified as zero in any list, except in lists that define I-frames.
9	TRANSMIT_THRESHOLD: Bit 9 is set to one if FRAME_SIZE is greater than the product of (BUFFER_SIZE – 8) and the TRANSMIT_BUFFER_MAXIMUM_COUNT parameters from the OPEN command.
10	ODD_ADDRESS: Bit 10 is set to one if an odd FORWARD_POINTER value is read on a list that does not have END_FRAME set to 1.
11	FRAME_ERROR: Bit 11 is set to one if the START_FRAME bit is set to one on a list that is not an anticipated start of frame or if the START_FRAME bit is zero on an anticipated start of frame.
12—13	RESERVED: Reset to zero.
14	ILLEGAL_FRAME_FORMAT: Bit 14 is set to one if bit 0 of the FC field was set to one. This bit is set only if the LLC Interface is not enabled. If the LLC interface has been enabled, this error is a nonfatal list error and is indicated in the CSTAT of the associated list.
15	RESERVED: Reset to zero.

Note:

Under heavy traffic conditions, it is possible for the TRANSMIT_STATUS interrupt to be generated with the READY bit set BEFORE the corresponding CSTAT completion of >FE has been received. It is recommended that the host software assume that a link is not READY when the software queues an I-frame list onto the transmit list chain and that the software wait for the CSTAT completion. If the CSTAT completes with >FE, the host software should not queue another I-frame until a TRANSMIT.STATUS interrupt occurs with the ready bit set.

If the CSTAT completes with >FF or CSTAT completes with >FE and is followed by a TRANSMIT_STATUS with the READY bit set, then the host can assume the link is ready and it can queue the next I-frame list when available. This procedure takes into account the ability of the adapter to generate the early TRANSMIT_STATUS interrupts with the READY bit set.

LLC 0005 TRANSMIT.HALT Command

Command TRANSMIT.HALT

Description The TRANSMIT.HALT command is used to interrupt the transmit list chain. Following recognition of this command, the adapter terminates the transmit chain as soon as possible. Any frames queued in the adapter are purged, and the TRANSMIT command is terminated by posting a TRANSMIT.STATUS interrupt and updating SSB_PARM_0 with XMIT_STATUS, and SSB_PARM_1 and SSB_PARM_2 with a pointer to the last transmit parameter list processed by the adapter. If TRANSMIT.HALT is issued and no frames have been transmitted, SSB_PARM_1 and SSB _PARM_2 are cleared.

This command is ignored by the adapter if there is not an executing TRANSMIT command or if the adapter has not been opened. A COMMAND.REJECT is issued if the LLC Interface has been enabled.

Initiation

	Byte + 0	Byte + 1
SCB_CMD	>0005	
SCB_PARM_0	>0000	
SCB_PARM_1	>0000	

Completion

SSB_CMD	>0004
SSB_PARM_0	XMIT_STATUS
SSB_PARM_1	XMIT_PARM_LIST (High)
SSB_PARM_2	XMIT_PARM_LIST (Low)

Initiation TRANSMIT.HALT does not require a command parameter block. Both the SCB_PARM_0 and SCB_PARM_1 fields are ignored but should be set to zero.

Completion Upon completion of TRANSMIT.HALT, the adapter generates a COMMAND.STATUS interrupt, sets SSB_CMD to >0004, updates SSB_CMD_0 with COMMAND_STATUS of the TRANSMIT command, and updates SSB_PARM_1 and SSB_PARM_2 with a pointer to the last transmit parameter list processed by the adapter.

Command RECEIVE

Description The RECEIVE command is used to receive frames from other stations on the ring. This command is normally issued only once (after OPEN) because receive data may be dynamically added to a RECEIVE parameter list chain. The RECEIVE command can be terminated because of a list error or by issuing a CLOSE command.

The routing field is passed to the attached system for all frames, if received. If the PAD ROUTING FIELD option is specified during OPEN, the routing field is padded to 18 bytes. If the frame does not contain a routing field, the field is still padded to 18 bytes. The padding does not alter the contents of the system's data buffer.

The logical format of received frames passed from the adapter to the attached system is identical to the logical format shown in Figure 4–16. The access control, frame control, destination address, source address, any routing information, and LLC fields are transferred to the attached system as they were received from the ring.

The RECEIVE command is rejected with adapter COMMAND REJECT STATUS under the following conditions:

□ The adapter has not been opened,
□ A RECEIVE command has already been issued, or
□ The address passed in the SCB is not word-aligned.

Initiation

	Byte + 0	Byte+ 1
SCB_CMD	>0006	
SCB_PARM_0	RCV_PARM_LIST (High)	
SCB_PARM_1	RCV_PARM_LIST (Low)	

RECEIVE.STATUS Interrupt

SSB_CMD	>0006
SSB_PARM_0	RCV_STATUS
SSB_PARM_1	RCV_PARM_LIST (High)
SSB_PARM_2	RCV_PARM_LIST (Low)

Initiation The SCB_PARM_0 and SCB_PARM_1 contain a 32-bit word-aligned address, which points to a RECEIVE parameter list. The format of the RECEIVE parameter list is shown in Figure 4–35. RECEIVE parameter lists must be aligned on even word boundaries.

Note:

When issuing this command, wait for the first word of the SCB to be cleared (>0000), before issuing another command.

Figure 4–35. RECEIVE Parameter List

		Byte + 0	Byte + 1
RCV_PARM_LIS	+ 0	FORWARD_POINTER (High)	
	+ 2	FORWARD_POINTER (Low)	
	+ 4	RECEIVE_CSTAT	
	+ 6	FRAME_SIZE	
	+ 8	DATA_COUNT	
	+ 10	DATA_ADDRESS (High)	
	+ 12	DATA_ADDRESS (Low)	
	+ 14	DATA_COUNT	
	+ 16	DATA_ADDRESS (High)	
	+ 18	DATA_ADDRESS (Low)	
	+ 20	DATA_COUNT	
	+ 22	DATA_ADDRESS (High)	
	+ 24	DATA_ADDRESS (Low)	

The definitions of the FORWARD_POINTER, DATA_COUNT, and DATA_ADDRESS fields are the same as for the transmit parameter list fields in Table 4–49. The RECEIVE_CSTAT field definition is given in Table 4–53.

Table 4–53. RECEIVE Parameter Field Bit Definitions

Field	Definition	
RECEIVE_CSTAT	This 16-bit parameter is set by the attached system when the receive parameter list is created. It is overwritten by the adapter to report frame completion status. When initially set by the attached system, this parameter field is referred to as the RECEIVE_CSTAT_REQUEST field. After a frame has been received, the adapter writes RECEIVE_CSTAT_COMPLETE to lists that start and end a frame. These bits indicate the completion status of the frame, not the RECEIVE command itself. However, if the forward pointer is odd, this field is not updated until a new list is given to the adapter.	
RECEIVE_CSTAT_REQUEST	Bit 0	VALID: The adapter waits for bit 0 to be set to a one before processing the current receive parameter list. The attached system must issue a RECEIVE.VALID interrupt request when changing bit 0 from zero to one. This bit is examined for every receive parameter list.
	Bit 1	FRAME_COMPLETE: Must be reset to zero.
	Bit 2	START_FRAME: Must be reset to zero.
	Bit 3	END_FRAME: Must be reset to zero.
	Bit 4	FRAME_INTERRUPT: If this bit is set to one, an adapter-to-attached system interrupt is generated when a frame has been received. This bit is ignored unless a list starts a frame.
	Bit 5	INTERFRAME_WAIT: If this bit is set to one, the adapter interrupts the attached system when a frame has been received. The adapter then assumes a receive-suspended state, waiting for the attached system to issue a RECEIVE.VALID interrupt. This bit is ignored unless a list starts or ends a frame. (Note: Valid or cancel is still required on a receive, pending interrupt.).
	Bit 6	PASS_CRC: When this bit is set, the adapter includes the frame's CRC in the information passed to the attached system. The CRC is the last four bytes passed to the attached system. The addi-tional four bytes are reflected in the FRAME_ SIZE field of the receive list. This bit is ignored in lists that do not start a frame. The CRC is passed only for externally matched frames.
	Bits 7—15	RESERVED: Must be reset to zero.

Table 4–53. RECEIVE Parameter Field Bit Definitions (Continued)

Field	Definition	
RECEIVE_ CSTAT_ COMPLETE	The completion code for the received frame is written to this field in lists that start or end a frame. RECEIVE_CSTATs that are not in lists that define the start or end of a frame are not altered by the adapter. The RECEIVE_CSTAT_COMPLETE bit definitions are shown below.	
	Bit 0	VALID: Reset to zero.
	Bit 1	FRAME_COMPLETE: Set to one.
	Bit 2	FRAME_START: Set to one on the list that starts the frame.
	Bit 3	FRAME_END: Set to one on the list that ends the frame.
	Bit 4	FRAME_INTERRUPT: Bit 4 has the same value as specified by the attached system in RECEIVE_CSTAT_REQUEST.
	Bit 5	INTERFRAME_WAIT: Bit 5 has the same value as specified by the attached system in RECEIVE_CSTAT_REQUEST.
	Bit 6	CRC_PASSED: This bit is set only if the PASS_CRC bit of the RECEIVE_CSTAT_REQUEST is set and if the CRC is contained within the data. This bit is set only in lists that have the FRAME_START bit (bit 2) set to one, and for which the frame received is externally matched.
	Bit 7	RESERVED: Reset to zero.
	Bits 8—13	RECEIVE_FS: On lists with START_FRAME set to one, this field contains the frame status field from the received frame.
	Bits 14—15	ADDRESS_MATCH: These bits contain the codes shown in Table 4–54.

Table 4–54. ADDRESS_MATCH Codes

Code		Definition
Bit 14	Bit 15	
0	0	Reserved
0	1	Internally address matched
1	0	Externally matched via XMATCH/XFAIL interface
1	1	Internally and externally matched

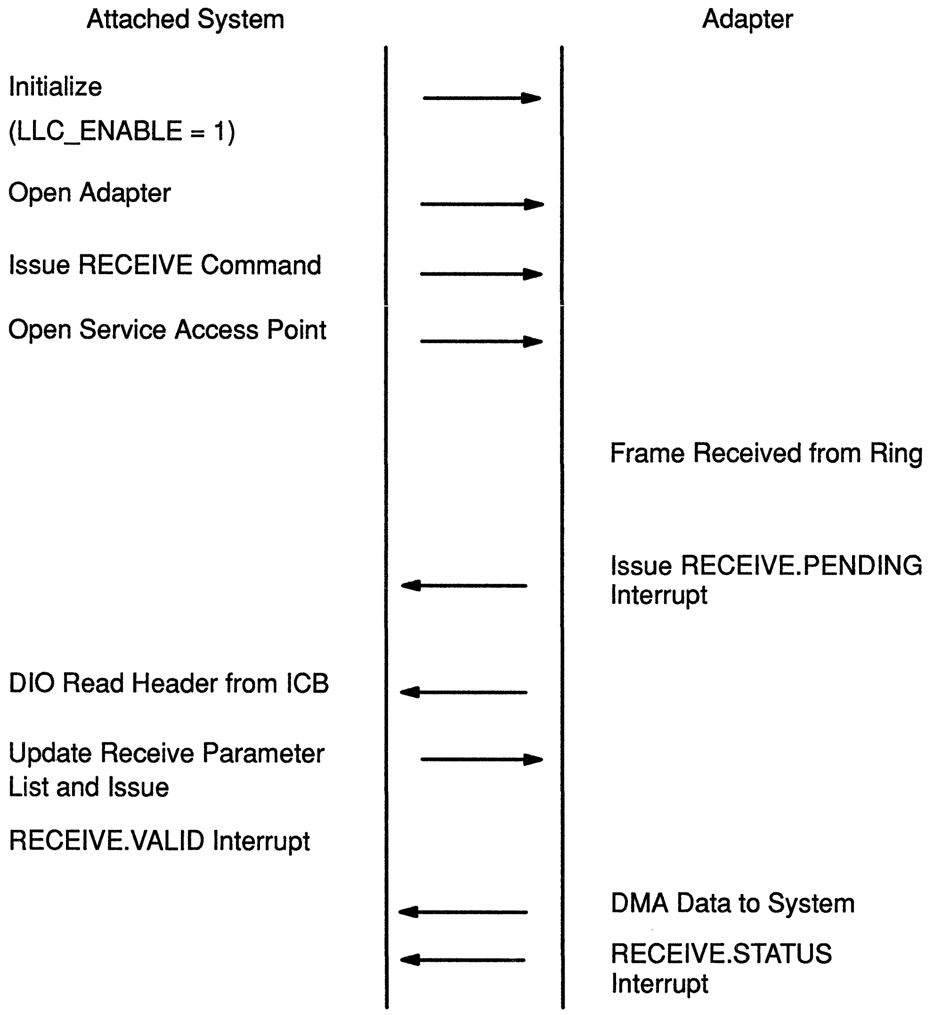
Execution

If the LLC interface is enabled, use the following method for routing frame data. The adapter, upon receiving a frame, updates the interface control block (see Section 4.10) with header information and posts a RECEIVE.PENDING interrupt to the attached system.

The attached system then reads the ICB via DIO to determine what action needs to be taken. The ICB indicates which SAP or link station the frame is destined for, the lengths of the MAC and LLC headers, and the length of the frame. It also indicates from whom the frame was sent. The attached system may transfer the frame data to its memory by updating the current receive parameter list and issuing a RECEIVE.VALID interrupt. If the attached system does not want the frame, a RECEIVE.CANCEL interrupt may be issued, and the frame is purged from the adapter's internal buffers. The RECEIVE.PENDING interrupt must be answered by the attached system.

Once the adapter has begun to DMA the frame data to the attached system's buffers, if it reaches an invalid receive list, it suspends operation but does not notify the attached system. If an odd forward pointer is reached, then the adapter suspends the receive and notifies the attached system. At this point, the attached system may continue the receive process by adding to the receive list chain and issuing a RECEIVE.VALID interrupt. The attached system may also purge the remainder of the frame by issuing a RECEIVE.CANCEL interrupt. Note that the RECEIVE.VALID starts the receive list processing, no matter what caused it to stop (valid bit or odd forward pointer). A typical receive flow for an adapter with LLC enabled is shown in Figure 4–36.

Figure 4–36. LLC RECEIVE Command Flow



Interrupt RECEIVE.STATUS

A RECEIVE.STATUS interrupt is generated when the RECEIVE parameter list chain has ended (odd address in FORWARD_POINTER) or when a frame is copied into a list that has the FRAME_INTERRUPT bit set in the RECEIVE_CSTAT field. The adapter updates SSB_PARM_0 with the RECEIVE_COMPLETE code and SSB_PARM_1 and SSB_PARM_2 with a 32-bit pointer to the last receive parameter list processed by the adapter. The RECEIVE_COMPLETE bit definitions are listed in Table 4–54.

Table 4–55. RECEIVE_COMPLETE Field Bit Definitions

Bit	Definition
0	FRAME_COMPLETE: Bit 0 is set to one when a frame has been received and the FRAME_INTERRUPT bit is set in RECEIVE_CSTAT. Since frames may be received and transferred faster than the attached system can respond to the interrupts and/or faster than the adapter can cause the interrupts, the RECEIVE.STATUS interrupt may report the arrival of more than one frame at a time. The SSB_PARM_1 and SSB_PARM_2 contain the receive parameter list address of the last frame transferred to the system. If lists with the FRAME_INTERRUPT bit set are intermixed with lists that do not have the FRAME_INTERRUPT bit set, RECEIVE.STATUS can include frames that do not have the FRAME_INTERRUPT bit set. The FRAME_COMPLETE bit will not be set with the RECEIVE_SUSPENDED bit also set.
1	RECEIVE_SUSPENDED: Bit 1 is set to one when the adapter detects an odd address in the FORWARD_POINTER field of a receive parameter list. SSB_PARM_1 and SSB_PARM_2 contain the address of the list that has an odd FORWARD_POINTER. The attached system must update the FORWARD_POINTER and issue a RECEIVE.VALID interrupt (RECEIVE.CONTINUE if the LLC interface is not enabled) or a RECEIVE.CANCEL to resume the RECEIVE command processing. The RECEIVE_SUSPENDED bit is not set with the FRAME_COMPLETE bit also set.
2—15	RESERVED: Reset to zero.

LLC 0007 CLOSE Command

Command CLOSE

Description The CLOSE command is used to terminate transmission on the ring or to terminate OPEN with the wrap option. Any frames residing in adapter internal buffers are purged. All SAPs and link stations are closed. After this command has completed, the adapter must be reset and then reinitialized before issuing an OPEN command.

A COMMAND.REJECT interrupt is issued if the adapter has not been opened.

Initiation

	Byte + 0	Byte + 1
SCB_CMD	>0007	
SCB_PARM_0	>0000	
SCB_PARM_1	>0000	

Completion

SSB_CMD	>0007	
SSB_PARM_0	COMMAND_STATUS	
SSB_PARM_1	>0000	
SSB_PARM_2	>0000	

Initiation CLOSE does not require a command parameter block. Both the SCB_PARM_0 and SCB_PARM_1 fields are ignored but should be set to zero.

Completion Upon completion of CLOSE, the adapter generates a COMMAND.STATUS interrupt, sets SSB_CMD to >0007, and updates SSB_PARM_0 with COMMAND_STATUS, as shown in Table 4–56.

Table 4–56. CLOSE Return Codes

Value	Explanation
>8000	Good completion

Command SET.GROUP.ADDRESS

Description The SET.GROUP.ADDRESS command is used to alter the adapter group address after an OPEN command has been issued.

A COMMAND.REJECT interrupt is issued if the adapter has not been opened.

Initiation

	Byte + 0	Byte + 1
SCB_CMD	>0008	
SCB_PARM_0	GROUP_ADDRESS_HIGH	
SCB_PARM_1	GROUP_ADDRESS_LOW	

Completion

SSB_CMD	>0008
SSB_PARM_0	COMMAND_STATUS
SSB_PARM_1	>0000
SSB_PARM_2	>0000

Initiation The SCB_PARM_0 and SCB_PARM_1 fields contain a 32-bit address, which is the group address to be stored in the adapter group address register. Bit 0 is forced to one by the adapter. The most significant two bytes of the 48-bit group address is >C000.

Completion Upon completion of SET.GROUP.ADDRESS, the adapter generates a COMMAND.STATUS interrupt. The SSB_CMD field is set to >0008, and the SSB_PARM_0 field is updated with COMMAND_STATUS, as shown in Table 4–57.

Table 4–57. SET.GROUP.ADDRESS Return Codes

Value	Explanation
>8000	Good completion

Command SET.FUNCTIONAL.ADDRESS

Description The SET.FUNCTIONAL.ADDRESS command is used to alter the adapter functional address after an OPEN command has been issued.

A COMMAND.REJECT interrupt is issued if the adapter has not been opened.

Initiation

	Byte + 0	Byte + 1
SCB_CMD	>0009	
SCB_PARM_0	FUNCTIONAL_ADDRESS_HIGH	
SCB_PARM_1	FUNCTIONAL_ADDRESS_LOW	

Completion

SSB_CMD	>0009
SSB_PARM_0	COMMAND_STATUS
SSB_PARM_1	>0000
SSB_PARM_2	>0000

Initiation The SCB_PARM_0 and SCB_PARM_1 fields contain a 32-bit address, which is the functional address to be stored in the adapter's internal functional address register. Bits 0 (most significant bit) and 31 (least significant bit) of the functional address are ignored. The most significant two bytes of the 48-bit functional address is >C000.

Completion Upon completion of SET.FUNCTIONAL.ADDRESS, the adapter generates a COMMAND.STATUS interrupt and sets the SSB_CMD field to >0009. The SSB_PARM_0 field is updated with COMMAND_STATUS, as shown in Table 4–58.

Table 4–58. SET.FUNCTIONAL.ADDRESS Return Codes

Value	**Explanation**
>8000	Good completion

Command READ.ERROR.LOG

Description The READ.ERROR.LOG command is used to read the adapter's error counters. These error counters are reset after this command has completed.

This command is ignored if the adapter has not been opened.

Initiation

	Byte + 0	Byte + 1
SCB_CMD	>000A	
SCB_PARM_0	ERROR_LOG_ADDR (High)	
SCB_PARM_1	ERROR_LOG_ADDR (Low)	

Completion

SSB_CMD	>000A	
SSB_PARM_0	COMMAND_STATUS	
SSB_PARM_1	>0000	
SSB_PARM_2	>0000	

Initiation The SCB_PARM_0 and SCB_PARM_1 fields contain a 32-bit pointer to a 14-byte area in attached system memory where the error log table is to be written. Data is transferred to the host system using the RECEIVE command DMA transfer method and burst size.The 14-byte error log table format is shown in Figure 4–37. The error counters are explained in Table 4–59.

Figure 4–37. Error Log Table

	Byte + 0	Byte + 1
+ 0	LINE_ERROR	RESERVED
+ 2	BURST_ERROR	ARI/FCI_ERROR
+ 4	RESERVED	RESERVED
+ 6	LOST_FRAME _ERROR	RECEIVE_CONGESTION _ERROR
+ 8	FRAME_COPIED_ERROR	RESERVED
+ 10	TOKEN_ERROR	RESERVED
+ 12	DMA_BUS_ERRORS	DMA_PARITY_ERRORS

Table 4–59. Adapter Error Counters

Counter	Definition
LINE_ERROR	The line error counter is incremented whenever 1) A frame is repeated or copied, and 2) The Error Detected Indicator (EDI) is zero in the incoming frame, and 3) At least one of the following conditions exists: a) A code violation between the starting delimiter and the ending delimiter of the frame. b) A code violation in a token. c) A Frame Check Sequence (FCS) error. When the line error is incremented, the EDI of the frame is set to one so that no further adapters count the error.
BURST_ERROR	The burst error counter is contained in all adapter configurations and is incremented when the adapter detects the absence of transitions for five half-bit times between SDEL and EDEL, or EDEL and SDEL. Only one adapter detects the burst five condition because the adapter that detects a burst four condition (four half-bit times without transition) directs its transmitter to transmit idles (zeros), if the burst five condition is detected.
ARI/FCI_ERROR	The ARI/FCI error counter is incremented when an adapter receives an Active Monitor Present (AMP) MAC frame with the ARI/FCI bits equal to zero and a Standby Monitor Present (SMP) MAC frame with the ARI/FCI bits equal to zero, or more than one SMP MAC frame with the ARI/FCI bits equal to zero, without receiving an intervening AMP MAC frame. This condition indicates that the upstream neighbor is unable to set the ARI/FCI bits in a frame that it has copied.
LOST_FRAME _ERROR	The lost frame error counter is incremented when an adapter is in transmit (stripping) mode and fails to receive the end of the frame it transmitted.
RECEIVE_ CONGESTION_ERROR	The receive congestion error counter is incremented when an adapter in the repeat mode recognizes a frame addressed to it but has no buffer space available to copy the frame.
FRAME_COPIED _ERROR	The frame copied error counter is incremented when an adapter in the receive/repeat mode recognizes a frame addressed to its specific address but finds the ARI bits not equal to zero. This indicates a possible line hit or duplicate address.

Table 4–59. Adapter Error Counters (Continued)

Counter	Definition
TOKEN_ERROR	The token error counter is active only in the active monitor station. It is incremented when the active monitor detects an error with the token protocol as follows: 1) The MONITOR_COUNT bit of a token with nonzero priority equals one. 2) The MONITOR_COUNT bit of a frame equals one. 3) No token or frame is received within a 10-ms window. 4) The starting delimiter/token sequence has a code violation in an area where code violations must not exist.
DMA_BUS_ERRORS	The DMA bus error counter counts the occurrences of DMA bus errors that do not exceed the abort thresholds as specified in the initialization parameters.
DMA_PARITY _ERRORS	The DMA parity error counter counts the occurrences of DMA parity errors that do not exceed the abort thresholds as specified in the initialization parameters.

Completion Upon completion of READ.ERROR.LOG, the adapter generates a COMMAND.STATUS interrupt and sets the SSB_CMD field to >000A. The SSB_PARM_0 field is updated with COMMAND_STATUS as shown in Table 4–60.

Table 4–60. READ.ERROR.LOG Return Codes

Value	Explanation
>8000	Good completion

Command READ.ADAPTER

Description The READ.ADAPTER command is used to transfer adapter memory contents across the system interface to host system memory.

Initiation

	Byte + 0	Byte + 1
SCB_CMD	>000B	
SCB_PARM_0	DATA_AREA_ADDR (High)	
SCB_PARM_1	DATA_AREA_ADDR (Low)	

Completion

SSB_CMD	>000B
SSB_PARM_0	COMMAND_STATUS
SSB_PARM_1	>0000
SSB_PARM_2	>0000

Initiation The SCB_PARM_0 and SCB_PARM_1 fields contain a 32-bit pointer to a buffer in host system memory, where the adapter stores the contents of the specified adapter memory locations. Data is transferred to the host system using the RECEIVE command, DMA transfer method and burst size. The host system data area specified by the host system should be written to in the format shown in Figure 4–38.

Figure 4–38. Host System Data Area Format

+ 0	DATA_COUNT	
+ 2	DATA_ADDRESS	
+ 4		Data area
+ N-2		

This 16-bit DATA_COUNT field contains the number of bytes to be read from the adapter.

DATA_ADDRESS is a 16-bit field containing the address of the data in Chapter 1 of adapter memory to be read. Bit 15 is reset to zero by the adapter. The READ.ADAPTER command results in an ADAPTER.CHECK interrupt if reference is made to an undefined storage area.

Note:

Any chapter of TMS380C16 memory may be inspected by using the DIO registers, with the following exceptions:

1) Address range >01.0200 through >01.02FE. If this is read, an adapter check occurs.
2) Address range >01.0100 through >01.01FE. If this is read, erroneous operation and deinsertion may occur.

Table 4–8 illustrates the internal adapter pointers accessible via the READ.ADAPTER command or DIO. These pointers reside beginning at location >01.0A00 in adapter memory and **must** be read following initialization but **before** issuing the OPEN command.

Completion Upon completion of READ.ADAPTER, the adapter generates a COMMAND.STATUS interrupt and sets the SSB_CMD field to >000B. The SSB_PARM_0 field is updated with COMMAND_STATUS as shown in Table 4–61.

Table 4–61. READ.ADAPTER Return Codes

Value	Explanation
>8000	Good completion

LLC 000D MODIFY.OPEN.PARAMETERS Command

Command MODIFY.OPEN.PARAMETERS

Description MODIFY.OPEN.PARAMETERS may be used to modify certain adapter operational parameters that were set by the OPEN command. All OPEN options, with the exception of WRAP_INTERFACE, that were specified in the OPEN_OPTIONS field of the OPEN parameter list may be modified. The bit corresponding to WRAP_INTERFACE (bit 0) is ignored by this command. HOST_BASED_I_FRAME_TRANSMIT (bit 12) should not be modified by this command.

A COMMAND.REJECT is issued if the adapter has not been opened. This command is functionally equivalent to the RESTORE.OPEN.PARAMETERS.

Initiation

	Byte + 0	Byte + 1
SCB_CMD	>000D	
SCB_PARM_0	OPEN_OPTIONS	
SCB_PARM_1	>0000	

Completion

SSB_CMD	>000D	
SSB_PARM_0	COMMAND_STATUS	
SSB_PARM_1	>0000	
SSB_PARM_2	>0000	

Initiation The SCB_PARM_0 of the SCB contains a 16-bit OPEN_OPTION field. The bit descriptions for this field are the same as those specified for the OPEN_OPTIONS field of the OPEN command. SCB_PARM_1 is ignored but should be set to zero.

Completion Upon completion of MODIFY.OPEN.PARAMETERS, the adapter generates a COMMAND.STATUS interrupt, sets SSB_CMD to >000D and updates SSB_PARM_0 with COMMAND_STATUS. The COMMAND_STATUS takes on values according to Table 4–62.

Table 4–62. MODIFY.OPEN.PARAMETERS Return Codes

Value	Explanation
>8000	Good completion
>0100	Invalid OPEN option

LLC 000E RESTORE.OPEN.PARAMETERS Command

Command RESTORE.OPEN.PARAMETERS

Description RESTORE.OPEN.PARAMETERS may be used to modify certain adapter operational parameters that were set by the OPEN command. All Open options, with the exception of WRAP_INTERFACE, that were specified in the OPEN_OPTIONS field of the open parameter list may be modified. The bit corresponding to WRAP_INTERFACE (bit 0) is ignored by this command. HOST_BASED_I_FRAME_TRANSMIT (bit 12) should not be modified by this command.

A COMMAND.REJECT is issued if the adapter has not been opened. This command is functionally equivalent to the MODIFY.OPEN.PARAMETERS.

Initiation

	Byte + 0	Byte + 1
SCB_CMD	>000E	
SCB_PARM_0	OPEN_OPTIONS	
SCB_PARM_1	>0000	

Completion

SSB_CMD	>000E
SSB_PARM_0	COMMAND_STATUS
SSB_PARM_1	>0000
SSB_PARM_2	>0000

Initiation The SCB_PARM_0 field of the SCB contains a 16-bit OPEN_OPTIONS field. The bit descriptions for this field are the same as those specified for the OPEN_OPTIONS field of the OPEN command. SCB_PARM_1 must be set to zero.

Completion Upon completion of RESTORE.OPEN.PARAMETERS command, the adapter generates a COMMAND.STATUS interrupt, sets SSB_CMD to >000E, and updates SSB_PARM_0 with COMMAND_STATUS. The COMMAND_STATUS takes on values according to Table 4–63.

Table 4–63. RESTORE.OPEN.PARAMETERS Return Codes

Value	Explanation
>8000	Good completion
>0100	Invalid OPEN option

LLC 000F SET.FIRST.16.GROUP.ADDRESS Command

Command SET.FIRST.16.GROUP.ADDRESS

Description The SET.FIRST.16.GROUP.ADDRESS command is used to alter the first two bytes in the adapter group address. After the OPEN command has been issued, these first two bytes are >C000.

A COMMAND.REJECT is issued if the adapter has not been opened.

Initiation

	Byte + 0	Byte + 1
SCB_CMD	>000F	
SCB_PARM_0	FIRST.16.GROUP.ADDRESS	
SCB_PARM_1	>0000	

Completion

SSB_CMD	>000F
SSB_PARM_0	COMMAND_STATUS
SSB_PARM_1	>0000
SSB_PARM_2	>0000

Initiation The SCB_PARM_0 field contains the two high-order bytes in the group address to be stored in the adapter group address register. Bit 0 is forced to one by the adapter.

Completion Upon completion of SET.FIRST.16.GROUP.ADDRESS, the adapter generates a COMMAND.STATUS interrupt with SSB_CMD field set to >000F and COMMAND_STATUS as shown in Table 4–64.

Table 4–64. SET.FIRST.16.GROUP.ADDRESS Return Codes

Value	**Explanation**
>8000	Good completion

Command SET.BRIDGE.PARMS

Description The SET.BRIDGE.PARMS command is used with adapters that implement an external source routing checker device. This command provides values and conditions for the adapter hardware to use when frames are copied for forwarding.

This command must be issued after the adapter has been opened. A COMMAND.REJECT is issued if the adapter has not been opened.

Initiation

	Byte + 0	Byte + 1
SCB_CMD	>0010	
SCB_PARM_0	BRIDGE_PARM_BLOCK (High)	
SCB_PARM_1	BRIDGE_PARM_BLOCK (Low)	

Completion

SSB_CMD	>0010
SSB_PARM_0	COMMAND_STATUS
SSB_PARM_1	>0000
SSB_PARM_2	>0000

Initiation The SCB_PARM_0 and SCB_PARM_1 fields contain a 32-bit pointer to the BRIDGE_PARM_BLOCK, which is shown in Figure 4–39. Descriptions of the BRIDGE_PARM_BLOCK fields are shown in Table 4–65.

Figure 4–39. BRIDGE_PARM_BLOCK

	Byte + 0	Byte + 1
+ 0	OPTIONS	
+ 2	SOURCE_RING	
+ 4	TARGET_RING	
+ 6	BRIDGE_NUMBER	

Table 4–65. BRIDGE_PARM_BLOCK Field Definitions

Field	Definition	
OPTIONS	Bit 0	SINGLE_ROUTE_BROADCAST: This bit is used to limit the routing of broadcast frames. When this bit is zero, single-route broadcast frames are rejected; all-routes broadcast frames are considered for forwarding to the host. If this bit is a one, then both single-route and all-routes broadcast frames are forwarded.
	Bits 1—5	RESERVED: These bits must be set to zero.
	Bits 6—11	MAXIMUM_ROUTING_FIELD_LENGTH: Maximum number of bytes (up to 30) that the routing field may contain, including the routing control field. Broadcast frames are rejected if they equal the maximum length, and all frames are rejected if the maximum length is exceeded. To operate to the IBM specification, a value of 18 should be used. This value must be even at least six.
	Bits 12—15	PARTITION_LENGTH: The value in this field is used to determine what portion of each 2-byte segment in the routing information field contains the bridge number. A value of 4 means that the low-order 4 bits of the segment indicate the bridge number. The remaining 12 bits contain the ring number. There is no default value for this field. The host system software is responsible for maintaining the validity check on the value used. All bridges in the network must use the same value for this field or its equivalent. A partition length of zero, or more than 14, results in an invalid bridge option. To operate to the IBM specification, a value of four should be used.
SOURCE_RING	When determining whether to forward a frame, the adapter compares the value in this field with the routing information source ring field in frames received from the ring. This value must be the number of the ring to which this adapter is connected. The acceptable range of values for this field depends on the setting of the partition length in the option field. For example, the valid range of values is >001 to >FFF if the partition length parameter value is 4. The SOURCE_RING value must be different from the TARGET_RING value.	

Table 4–65. BRIDGE_PARM_BLOCK Field Definitions (Continued)

Field	Definition
TARGET_RING	When determining whether to forward a frame, the adapter compares the value in this field with the routing information target ring field in frames received from the ring. This value must be the number of the ring to which the other adapter in this host is connected. The acceptable range of values for this field depends on the setting of the partition length in the option field. For example, the valid range of values is >001 to >FFF if the partition length parameter value is 4. The TARGET_RING value must be different from the SOURCE_RING value.
BRIDGE_ NUMBER	When determining whether to forward a frame, the adapter compares the value in this field with the routing information bridge number field in the frames received from the ring. This value can be from zero up to the number allowed by the partition length.

Completion Upon completion of the SET.BRIDGE.PARMS command, the adapter generates a COMMAND.STATUS interrupt and sets the SSB_CMD field to >0010. The SSB_PARM_0 field is updated with COMMAND_STATUS. If there is a SET_BRIDGE_PARMS error, the TMS380SRA is disabled. The COMMAND_STATUS takes on values according to Table 4–66.

Table 4–66. SET.BRIDGE.PARMS Return Codes

Bit	Definition
0	Good completion
1	Invalid maximum routing field length. This length given was odd, less than 6, or larger than supported by the hardware. (Maximum for TMS380SRA is 30.)
2	Invalid SOURCE_RING number. The number given was zero or was larger than allowed by the setting of the PARTITION_LENGTH in the options field. This bit is set if the TARGET and SOURCE_RING numbers are equal.
3	Invalid TARGET_RING number. The number given was zero or was larger than allowed by the setting of the PARTITION_LENGTH in the options field. This bit is set if the TARGET and SOURCE_RING numbers are equal.
4	Invalid BRIDGE_NUMBER. The number given was larger than allowed by the setting of the PARTITION_LENGTH in the options field.
5	Invalid bridge options.
6	The TMS380SRA failed diagnostics.
7	The TMS380SRA does not exist in the present hardware configuration.

LLC 0011 CONFIG.BRIDGE.PARMS Command

Command CONFIG.BRIDGE.PARMS

Description The CONFIG.BRIDGE.PARMS command forces the bridge functional address bit (bit 23) to one in the OPEN.ADAPTER and SET.FUNCTIONAL.ADDRESS commands. This comand must be issued after the adapter is opened. A COMMAND.REJECT is issued if the adapter is not open.

Initiation

	Byte + 0	Byte + 1
SCB_CMD	>0011	
SCB_PARM_0	CONTROL	
SCB_PARM_1	>0000	

Completion

SSB_CMD	>0011	
SSB_PARM_0	COMMAND_STATUS	
SSB_PARM_1	>0000	
SSB_PARM_2	>0000	

Initiation The SCB_PARM_0 field contains the CONTROL parameter for this command. If the CONTROL parameter is nonzero, then the bridge functional address (>C00000000100) is set. A CONTROL of zero turns off the bridge functional address. The SCB_PARM_1 field is ignored but should be set to zero.

Completion Upon completion of the CONFIG.BRIDGE.PARMS, the adapter generates a COMMAND.STATUS interrupt, sets SSB_CMD to >0011, and updates SSB_PARM_0 with COMMAND_STATUS. The COMMAND_STATUS takes on values according to Table 4–67.

Table 4–67. CONFIG.BRIDGE.PARMS Return Codes

Value	Explanation
>8000	Good completion

Command LLC.RESET

Description LLC.RESET is used to terminate all outstanding commands for all link stations within a specified SAP, including commands to the SAP itself. The link station(s) and SAP that are affected by LLC.RESET must be reopened for further use. LLC.RESET may also be used as a global reset for all SAPs and all link stations. If all SAPs and link stations are reset, the adapter is placed in a state identical to the state immediately following the OPEN.ADAPTER command.

A COMMAND.REJECT interrupt is issued if the adapter has not been opened or if the LLC interface has not been enabled.

Initiation

	Byte + 0	Byte + 1
SCB_CMD	>0014	
SCB_PARM_0	STATION_ID	
SCB_PARM_1	>0000	

Completion

SSB_CMD	>0014
SSB_PARM_0	COMMAND_STATUS
SSB_PARM_1	STATION_ID
SSB_PARM_2	>0000

Initiation The STATION_ID associated with the SAP(s) to be reset is placed in SCB_PARM_0. STATION_ID >XX00 resets all link stations in SAP >XX, including the SAP itself, and STATION_ID >0000 resets all SAPs and all link stations. SCB_PARM_1 is ignored but should be set to zero.

Completion Upon completion of LLC.RESET, the adapter generates a COMMAND.STATUS interrupt, sets SSB_CMD to >0014 and updates SSB_PARM_0 with COMMAND_STATUS. The COMMAND_STATUS takes on values according to Table 4–68.

Table 4–68. LLC.RESET Return Codes

Value	Explanation
>0000	Good completion
>0040	Invalid station ID

Command OPEN.SAP

Description OPEN.SAP can be used to activate an individual SAP or a group SAP. Parameters for SAP operation and default link station parameters are passed to the adapter by this command. The parameter values set by this command may be modified with the MODIFY.LLC.PARMS command.

This command is rejected if the adapter has not been opened or if the LLC interface has not been enabled.

Initiation

	Byte + 0	Byte + 1
SCB_CMD	>0015	
SCB_PARM_0	SAP_PARM_BLOCK (High)	
SCB_PARM_1	SAP_PARM_BLOCK (Low)	

Completion

SSB_CMD	>0015
SSB_PARM_0	COMMAND_STATUS
SSB_PARM_1	STATION_ID
SSB_PARM_2	FAILING_GROUP_SAP

Initiation The SCB_PARM_0 and SCB_PARM_1 fields contain a 32-bit pointer to the SAP_PARM_BLOCK, which is illustrated in Figure 4–40. The detailed field definitions are given in Table 4–69.

Figure 4–40. SAP_ Parm_ Block

	Byte + 0	Byte + 1
+ 0	RESERVED	
+ 2	RESERVED	
+ 4	TIMER_T1	TIMER_T2
+ 6	TIMER_TI	MAXOUT
+ 8	MAXIN	MAXOUT_INCR
+ 10	MAX_RETRY_COUNT	GSAP_MAX_MEMBER
+ 12	MAX_I_FIELD	
+ 14	SAP_VALUE	SAP_OPTIONS
+ 16	STATION_CNT	RESERVED
+ 18	RESERVED	GROUP_MEMBER_CNT
+ 20	SAP_GROUP_LIST_PTR (High)	
+ 22	SAP_GROUP_LIST_PTR (Low)	

+ 0	GSAP_0	GSAP_1
+ 2	GSAP_2	GSAP_3
+ 4	GSAP_4	GSAP_5
+ 6	GSAP_6	GSAP_7

Table 4–69. SAP Parameter Block Field Definitions

Field	Definition
TIMER_T1	This parameter sets the response timer value and is specified as an integer between 1 and 10. Timer values between 1 and 5 use the group 1 timer period (as a multiplier), and values between 6 and 10 use the group 2 timer period (as a multiplier). An I-frame is retried if an acknowledgement for a transmitted frame is not received within the time period specified. If the value is zero, a default of 5 is used.
TIMER_T2	This parameter sets the receive acknowledge timer value and is specified as an integer between 1 and 10. Timer values between 1 and 5 use the group 1 timer period (as a multiplier), and values between 6 and 10 use the group 2 timer period (as a multiplier). If the receive window (defined by the MAXIN parameter) is not met, the link station acknowledges a received frame(s) as soon as possible after this timer expires. The value specified for this parameter must be less than that specified for TIMER_T1. If the value is zero, a default of 2 is used. If the value is greater than 10, the timer is not used.
TIMER_TI	This parameter sets the Inactivity Timer (TI) value and is specified as an integer between 1 and 10. Timer values between 1 and 5 use the group 1 timer period (as a multiplier), and values between 6 and 10 use the group 2 timer period (as a multiplier). If no link activity has been detected within this time period, a link station SAP performs a test to determine whether its link with the remote station is still intact . If the value is zero, a default of 3 is used.
MAXOUT	This parameter specifies the maximum number of unacknowledged transmitted I-frames that may be outstanding at any one time. If this parameter is set to zero, a default value of two is used. The maximum value for this parameter is 127.
MAXIN	This parameter specifies the maximum number of I-frames that may be received before an acknowledgement is sent. If this parameter is set to zero, a default value of 1 is used. The maximum value for this parameter is 127.
MAXOUT_INCR	This parameter is used to specify the dynamic windowing algorithm increment value. If set to zero, a default value of one is used. This is the number of frames that must be acknowledged before the dynamic windowing algorithm increments MAXOUT.

Table 4–69. SAP Parameter Block Field Definitions (Continued)

Field	Definition
MAX_RETRY_COUNT	This parameter specifies the default IEEE value for link stations associated with SAP_VALUE. IEEE defines the maximum number of times that an I-format or S-format frame can be retransmitted by the link station. The maximum value of this parameter is 255. If specified as zero, this parameter is defaulted to 8.
GSAP_MAX_MEMBER	If the service access point being opened is to be a group SAP (GSAP), this parameter specifies the maximum number of SAPs that may be opened as members of the GSAP. This parameter is ignored if bit 6 of the SAP_OPTIONS is zero. This parameter must be less than or equal to the similar parameter in the OPEN command.
MAX_I_FIELD	This parameter specifies the maximum information field length in a frame received for a link station associated with this SAP. If this field is zero, a default value of 600 is used. This parameter is ignored if the STATION_COUNT parameter is zero.
SAP_VALUE	This parameter specifies the value of the SAP to be activated. SAP value >00, the null SAP, is reserved and must not be used in this field. The least significant bit of SAP_VALUE must be zero. The value >FE must not be used for a group SAP.

Table 4–69. SAP Parameter Block Field Definitions (Continued)

Field	Definition
SAP_OPTIONS	The bit functions of the SAP options field are defined below: Bits 0—2 ACCESS_PRIORITY: The value specified in this field is placed in the AC field of all frames transmitted from this SAP. The maximum allowed access priority is 6. Bit 3 RESERVED: This bit must be set to zero. Bit 4 XID_HANDLER: When this bit is set, responses to XID frames are managed by the attached system, and the adapter passes all received XID frames to the attached system. If this bit is 0, the adapter automatically responds to XID frames. If Bit 7 of the SAP_OPTIONS is set, this bit must be set exactly as it is in the group SAP, or the OPEN_SAP command returns an error. Bit 5 INDIVIDUAL_SAP: When this bit is set, the SAP is activated as an individual SAP. Bit 6 GROUP_SAP: When this bit is set, the SAP is activated as a group SAP. Bit 7 GROUP_MEMBER: When this bit is set, the SAP is activated as a member of a group SAP. **Note**: At least one of the bits 5, 6, or 7 must be set. Bit 7 can be set only if bit 5 is set.
STATION_ COUNT	This parameter specifies the maximum number of link stations that may be associated with this SAP. This command returns an error if this parameter is nonzero and bit 5 of the SAP_OPTIONS is reset to zero. If this parameter is equal to zero, then this SAP can transmit only type 1 LLC frames.
GROUP_ MEMBER_ COUNT	This parameter specifies the number of GSAPs of which the SAP to be opened wishes to become a member. The number specified in this field signals to the adapter how many of the GSAP_0—GSAP_7 fields are valid. This parameter is ignored if bit 5 or bit 7 of the SAP_OPTIONS is zero.
SAP_GROUP_ LIST_PTR	These fields point to a list of group SAPs, GSAP_0—GSAP_7, to which the opening SAP belongs.
GSAP_0 —GSAP_7	These fields contain the GSAP values for the group service access points of which the SAP to be opened wishes to become a member. None of these fields may be skipped (if membership in 3 GSAPs is desired, the GSAP values must be placed in GSAP_0, GSAP_1, and GSAP_2).

Completion Upon completion of OPEN.SAP, the adapter generates a COMMAND.STATUS interrupt and sets the SSB_CMD field to >0015. The SSB_PARM_0 field is updated with COMMAND_STATUS. The COMMAND_STATUS takes on values according to Table 4–70. The STATION_ID associated with the opened SAP is returned in SSB_PARM_1. If the command fails because of a problem with a group SAP (error codes >0045 or >0049), the SSB_PARM_2 field contains the FAILING_GROUP_SAP value.

Table 4–70. OPEN.SAP Return Codes

Value	Explanation
>0000	Good completion.
>0006	Invalid SAP_OPTIONS.
>0008	Unauthorized access priority. The maximum access priority allowed is 6, unless altered by net management.
>0042	Any parameter exceeds the maximum allowed.
>0043	Invalid SAP value.
>0045	Requesting group membership in nonexistent group.
>0046	Resources not available. MAX_SAP number of SAPs are already opened, or STATION_COUNT is greater than the remaining free internal link station resources..
>0049	Group SAP has max members and attempted to open another member.

LLC 0016 CLOSE.SAP Command

Command CLOSE.SAP

Description This command is used to deactivate a SAP. A SAP may not be closed if any link stations associated with the SAP are open. A group SAP may not be closed until all member SAPs have been closed. Any frames received for the specified SAP that are not in the process of being transferred across the system interface are purged.

A COMMAND.REJECT interrupt is issued if the adapter has not been opened or if the LLC interface has not been enabled.

Initiation

	Byte + 0	Byte + 1
SCB_CMD	>0016	
SCB_PARM_0	STATION_ID	
SCB_PARM_1	>0000	

Completion

SSB_CMD	>0016	
SSB_PARM_0	COMMAND_STATUS	
SSB_PARM_1	STATION_ID	
SSB_PARM_2	>0000	

Initiation The STATION_ID of the SAP to be closed is placed in SCB_PARM_0. SCB_PARM_1 is ignored by the adapter but should be set to zero.

Completion Upon completion of CLOSE.SAP, the adapter generates a COMMAND.STATUS interrupt, sets the SSB_CMD field to >0016, and updates SSB_PARM_0 with COMMAND_STATUS. The COMMAND_STATUS takes on the values shown in Table 4–71. The STATION_ID of the closed SAP is returned in SSB_PARM_1.

Table 4–71. CLOSE.SAP Return Codes

Value	Explanation
>0000	Good completion.
>0040	Invalid STATION_ID.
>0047	Unclosed link stations on SAP. Must close all link stations first.
>0048	Group SAP cannot close. All member SAPs not closed.
>004C	Close sequence error. Must close in proper order.

Command OPEN.STATION

Description OPEN.STATION allocates resources within the adapter to support a connection between a local and a remote link station. Specify the parameters needed by the adapter for allocation of these resources in a station parameter block, which is read by the adapter. A COMMAND.REJECT interrupt is issued if the adapter has not been opened or if the LLC interface has not been enabled.

Initiation

	Byte + 0	Byte + 1
SCB_CMD	>0019	
SCB_PARM_0	STATION_PARM_BLOCK (High)	
SCB_PARM_1	STATION_PARM_BLOCK (Low)	

Completion

SSB_CMD	>0019
SSB_PARM_0	COMMAND_STATUS
SSB_PARM_1	STATION_ID
SSB_PARM_2	>0000

Initiation The SCB_PARM_0 and SCB_PARM_1 fields contain a 32-bit pointer to the STATION_PARM_BLOCK, which is shown in Figure 4–41. The detailed definitions of these fields are given in Table 4–72.

Figure 4–41. STATION_PARAMETER_BLOCK

		Byte + 0	Byte + 1
STATION_PARM_BLOCK	+ 0	STATION_ID	
	+ 2	RESERVED	
	+ 4	TIMER_T1	TIMER_T2
	+ 6	TIMER_TI	MAXOUT
	+ 8	MAXIN	MAXOUT_INCR
	+ 10	MAX_RETRY_COUNT	RSAP_VALUE
	+ 12	MAX_I_FIELD	
	+ 14	STATION_OPTIONS	
	+ 16	REMOTE_NODE_ADDRESS_PTR (High)	
	+ 18	REMOTE_NODE_ADDRESS_PTR (Low)	

Table 4–72. STATION_PARAMETER_BLOCK Field Descriptions

Field	Definition
STATION_ID	This field contains the STATION_ID of the SAP under which the new link station is to be established.
TIMER_T1	As defined for OPEN.SAP, if this field is set to a value other than zero, then that value is used for this station. If zero, then the value defaults to that value set in the OPEN.SAP command.
TIMER_T2	As defined for OPEN.SAP if this field is set to a value other than zero, then that value is used for this station. If zero, then the value defaults to that value set in the OPEN.SAP command.
TIMER_TI	As defined for OPEN.SAP, if this field is set to a value other than zero, then that value is used for this station. If zero, then the value defaults to that value set in the OPEN.SAP command.
MAXOUT	As defined for OPEN.SAP, if this field is set to a value other than zero, then that value is used for this station. If zero, then the value defaults to that value set in the OPEN.SAP command.
MAXIN	As defined for OPEN.SAP, if this field is set to a value other than zero, then that value is used for this station. If zero, then the value defaults to that value set in the OPEN.SAP command.
MAXOUT_INCR	As defined for OPEN.SAP, if this field is set to a value other than zero, then that value is used for this station. If zero, then the value defaults to that value set in the OPEN.SAP command.
MAX_RETRY_COUNT	As defined for OPEN.SAP, if this field is set to a value other than zero, then that value is used for this station. If zero, then the value defaults to that value set in the OPEN.SAP command.
RSAP_VALUE	This parameter defines the SAP value of the remote station. The SAP value must be for an individual SAP (least significant bit=0). RSAP_VALUE is used in the DSAP field of transmitted frames and compared to the SSAP field of received frames.
MAX_I_FIELD	As defined for OPEN.SAP, if this field is set to a value other than zero, then that value is used for this station. If zero, then the value defaults to that value set in the OPEN.SAP command.
STATION_OPTIONS	The bit functions of the SAP options field are provided below: Bits 0—2 PRIORITY: This bit field defines the transmission priority for frames transmitted by this station. Bits 3—15 RESERVED: These bits must be set to zero.
REMOTE_NODE _ADDRESS_PTR	This field contains a 32-bit pointer to the location in system memory where the 6-byte ring address of the remote station is stored.

Completion Upon completion of OPEN.STATION, the adapter generates a COM MAND.STATUS interrupt, sets the SSB_CMD to >0019, and updates SSB_PARM_0 with COMMAND_STATUS. The COMMAND_STATUS takes on a value according to Table 4–73. The STATION_ID of the opened station is returned in SSB_PARM_1.

Table 4–73. OPEN.STATION Return Codes

Value	Explanation
>0000	Good completion.
>0008	Unauthorized access priority. The maximum access priority allowed is 6, unless altered by net management.
>0040	Invalid STATION_ID.
>0042	Parameter exceeds maximum.
>0043	Invalid remote SAP value or remote SAP value already in use.
>0046	Resources not available. MAX_STATION number of link station are already opened.
>004F	Invalid remote node address.

LLC 001A CLOSE.STATION Command

Command CLOSE.STATION

Description CLOSE.STATION is used to signal the adapter to initiate the sequence of events necessary to place the specified link station in a closed state.

A COMMAND.REJECT interrupt is issued if the adapter has not been opened or if the LLC interface has not been enabled.

Initiation

	Byte + 0	Byte + 1
SCB_CMD	>001A	
SCB_PARM_0	STATION_ID	
SCB_PARM_1	>0000	

Completion

SSB_CMD	>001A
SSB_PARM_0	COMMAND_STATUS
SSB_PARM_1	STATION_ID
SSB_PARM_2	>0000

Initiation The STATION_ID of the link station to be closed is placed in SCB_PARM_0. SCB_PARM_1 is ignored by the adapter.

Completion Upon completion of CLOSE.STATION, the adapter generates a COMMAND.STATUS interrupt, sets the SSB_CMD to >001A, and updates SSB_PARM_0 with COMMAND_STATUS. The COMMAND_STATUS takes on the values in Table 4–74. The STATION_ID of the closed station is returned in SSB_PARM_1.

Table 4–74. CLOSE.STATION Return Codes

Value	**Explanation**
>0000	Good completion.
>0002	Duplicate command outstanding. Have already issued a CLOSE.STATION command to this link.
>0040	Invalid STATION_ID.
>004B	Station closed without remote acknowledgement.
>004C	Sequence error. A CONNECT.STATION command is outstanding. Wait for connect command completion and reissue CLOSE.STATION command.

Command CONNECT.STATION

Description The CONNECT.STATION command is used to initiate the protocol necessary to place the local and remote link stations in a data transfer state. Specify the information the adapter needs to establish a logical link in the connect parameter block, which is read by the adapter.

A COMMAND.REJECT interrupt is issued if the adapter has not been opened or if the LLC interface has not been enabled.

Note: For ADAPTER_BASED_I_FRAME transmission only.

After a CONNECT.STATION command completes successfully, the attached system must assume a DMA-not-ready state. The adapter generates a TRANSMIT.STATUS interrupt to inform the attached system when the DMA is ready to accept I-frames for transmission. This interrupt occurs whether or not a TRANSMIT command has been issued. For more information, refer to the section on the TRANSMIT command for adapter-based I-frames.

Initiation

	Byte + 0	Byte + 1
SCB_CMD	>001B	
SCB_PARM_0	CONNECT_PARM_BLOCK (High)	
SCB_PARM_1	CONNECT_PARM_BLOCK (Low)	

Completion

SSB_CMD	>001B
SSB_PARM_0	COMMAND_STATUS
SSB_PARM_1	STATION_ID
SSB_PARM_2	>0000

Initiation The SCB_PARM_0 and SCB_PARM_1 fields contain a 32-bit pointer to the CONNECT_PARM_BLOCK, which is shown in Figure 4–42. The fields in this block are described in Table 4–75.

Figure 4–42. CONNECT_PARM_BLOCK

	Byte + 0	Byte + 1
+ 0	STATION_ID	
+ 2	RESERVED	
+ 4	ROUTING_INFO_PTR (High)	
+ 6	ROUTING_INFO_PTR (Low)	

Table 4–75. CONNECT_PARAMETER_BLOCK Field Bit Descriptions

Field	Definition
STATION_ID	This parameter specifies the link station with which a logical connection is desired.
RESERVED	This field is reserved and should be set to zero.
ROUTING_INFO_PTR	This field contains a 32-bit pointer to the system memory location of the routing information. The routing information is necessary for establishing a link between two stations that are not on the same ring. The routing information is ignored if the CONNECT.STATION is issued to a link that has received a SABME. For more information on the routing information field, see Chapter 2 of this document or refer to the *IBM Token-Ring Network Architecture Reference Guide* (IBM literature number 6165877). The adapter DMAs 18 bytes from the address pointed to by this field. If the value of the pointer is zero, no routing information is used.

Completion Upon completion of CONNECT.STATION, the adapter generates a COMMAND.STATUS interrupt, sets SSB_CMD to >001B, and updates SSB_PARM_0 with COMMAND_STATUS. The COMMAND_STATUS takes on values according to Table 4–76. The STATION_ID of the connected station is returned in SSB_PARM_1.

Table 4–76. CONNECT.STATION Return Codes

Value	Explanation
>0000	Good completion.
>0002	Duplicate command outstanding. Have already issued a CONNECT.STATION command to this link.
>0040	Invalid STATION_ID.
>0041	Protocol error. Link in invalid state for command.
>0044	Invalid routing information length.
>004D	Unsuccessful connection attempt. The remote station did not accept the connection request.
>004A	Sequence error. A CLOSE.STATION is outstanding. Wait for command completion and reissue a CONNECT.STATION command.

LLC 001C MODIFY.LLC.PARMS Command

Command MODIFY.LLC.PARMS

Description MODIFY.LLC.PARMS is used to modify open link station parameters or default SAP parameters. Parameter values to be modified are passed to the adapter via a modify parameter block.

A COMMAND.REJECT interrupt is issued if the adapter has not been opened or if the LLC interface has not been enabled.

Initiation

	Byte + 0	Byte + 1
SCB_CMD	>001C	
SCB_PARM_0	MODIFY_PARM_BLOCK (High)	
SCB_PARM_1	MODIFY_PARM_BLOCK (Low)	

Completion

SSB_CMD	>001C
SSB_PARM_0	COMMAND_STATUS
SSB_PARM_1	STATION_ID
SSB_PARM_2	FAILING_GROUP_SAP

Initiation The SCB_PARM_0 and SCB_PARM_1 fields contain a 32-bit pointer to the CONNECT_PARM_BLOCK, which is shown in Figure 4–43. The descriptions of the CONNECT_PARM_BLOCK fields are given in Table 4–77.

Figure 4–43. MODIFY_PARM_BLOCK

	Byte + 0	Byte + 1
+ 0	RESERVED	
+ 2	STATION_ID	
+ 4	TIMER_T1	TIMER_T2
+ 6	TIMER_TI	MAXOUT
+ 8	MAXIN	MAXOUT_INCR
+ 10	MAX_RETRY_COUNT	RESERVED
+ 12	RESERVED	
+ 14	SAP_OPTIONS	RESERVED
+ 16	RESERVED	
+ 18	RESERVED	GROUP_MEMBER_CNT
+ 20	SAP_GROUP_LIST_PTR (High)	
+ 22	SAP_GROUP_LIST_PTR (Low)	

	GSAP_0	GSAP_1
+ 2	GSAP_2	GSAP_3
+ 4	GSAP_4	GSAP_5
+ 6	GSAP_6	GSAP_7
+ 8	GSAP_8	GSAP_9
+ 10	GSAP_10	GSAP_11
+ 12	GSAP_12	RESERVED

Table 4–77. MODIFY_PARM_BLOCK Field Definitions

Field	Definition
STATION_ID	This field contains the STATION_ID of the SAP for which parameters are to be modified.
TIMER_T1	This parameter sets the response timer value and is specified as an integer between 1 and 10. Timer values are as defined in the OPEN.SAP command. If the value is zero, the current value remains set.
TIMER_T2	This parameter sets the receive acknowledge timer value and is specified as an integer between 1 and 10. Timer values are as defined in the OPEN.SAP command. If the value is zero, the current value remains set. If the value is greater than 10, the timer is not used.
TIMER_TI	This parameter sets the inactivity timer value and is specified as an integer between 1 and 10. Timer values are as defined in the OPEN.SAP command. If the value is zero, the current value remains set.
MAXOUT	This parameter specifies the maximum number of unacknowledged transmitted I-frames that may be outstanding at any one time. If this parameter is set to zero, the current value remains set. The maximum value for this parameter is 127.
MAXIN	This parameter specifies the maximum number of I-frames that may be received before an acknowledgement is sent. If this parameter is set to zero, the current value remains set. The maximum value for this parameter is 127.
MAXOUT_INCR	This parameter is used to specify the dynamic window increment value. If this value is zero, then the current value remains set.
MAX_RETRY_COUNT	This parameter specifies the default N2 value for link stations associated with the SAP_VALUE referenced by the STATION_ID. The maximum value of this parameter is 255. If this value is zero, then the current value remains set.
SAP_OPTIONS	The bit functions of the SAP options field are defined below: Bits 0—2 ACCESS_PRIORITY: The value specified in this bit field is placed in the AC field of all frames transmitted from this SAP. This parameter must be filled in. Bits 3—7 RESERVED: These bits must be set to zero.
GROUP_MEMBER _COUNT	This parameter specifies the number of valid fields in the SAP_GROUP_LIST (GSAP_0 — GSAP_12).

Table 4–77. MODIFY_PARM_BLOCK Field Definitions (Continued)

Field	Definition
SAP_GROUP_LIST_PTR	These fields point to a list of group SAPs as defined below. These fields (GSAP_0—GSAP_12) are ignored if the GSAP_MEMBER_COUNT parameter is zero.
GSAP_0—GSAP_12	These fields may be used either to request membership in additional group SAPs or to request that a membership be deleted. If the low-order bit of the SAP value is zero, then the addition of this membership is being requested. If the low-order bit is one, then the cancellation of this membership is being requested. These fields are ignored if the GSAP_MEMBER_COUNT parameter is zero.

Completion Upon completion of MODIFY.LLC.PARMS, the adapter generates a COMMAND.STATUS interrupt, sets SSB.CMD to >001C, and updates SSB_PARM_0 with COMMAND_STATUS. The COMMAND_STATUS takes on values according to Table 4–78. The STATION_ID associated with the command is returned in SSB_PARM_1. In the event of errors >0045 or >0049, the adapter updates SSB_PARM_2 with the SAP_VALUE of the FAILING_GROUP_ SAP.

Table 4–78. MODIFY.LLC.PARMS Return Codes

Value	Explanation
>0000	Good completion.
>0002	Duplicate command outstanding. Have already issued a MODIFY.LLC.PARMS command to this link.
>0008	Unauthorized access priority (bits 0–2) in the SAP_OPTIONS command.
>0040	Invalid STATION_ID.
>0042	Parameter exceeds maximum.
>0045	Requesting group membership in nonexistent group SAP.
>0049	Group SAP has max members.
>004E	Member not found in group SAP.

LLC 001D FLOW.CONTROL Command

Command FLOW.CONTROL

Description FLOW.CONTROL can be used to control the local busy state of a specific link station or of all link stations within a specified SAP. FLOW_OPTIONS are passed to the adapter; they specify the state of the local busy status and the local busy status reset condition. A FLOW.CONTROL to a STATION_ID of >0000 has no effect.

A COMMAND.REJECT interrupt is issued if the adapter has not been opened or if the LLC interface has not been enabled.

Initiation

	Byte + 0 Byte + 1
SCB_CMD	>001D
SCB_PARM_0	STATION_ID
SCB_PARM_1	FLOW_OPTIONS

Completion

SSB_CMD	>001D
SSB_PARM_0	COMMAND_STATUS
SSB_PARM_1	STATION_ID
SSB_PARM_2	>0000

Initiation A STATION_ID is placed in SCB_PARM_0 to specify the link station(s) that require data flow modification. A STATION_ID of >xx00 changes the local busy status of all link stations in SAP >xx. The FLOW_OPTIONS are placed in SCB_PARM_1. The FLOW_OPTIONS field bit positions are defined in Table 4–79.

Table 4–79. FLOW_OPTIONS Field Bit Positions

Bit	Definition
0	LOCAL_BUSY_STATUS: When this bit is cleared (0), the specified station(s) enter the local busy state. Bit 1 is ignored. When the bit is set (1), the specified link stations are reset from a local busy state entered by the condition specified in LOCAL_BUSY_CONDITION (Bit 1).
1	LOCAL_BUSY_CONDITION: When this bit is cleared (0), a user-set local busy condition is reset. When the bit is set (1), an out-of-receive buffer, a no-receive-command outstanding, or a receive-canceled state is reset.
2—15	RESERVED: These bits must be cleared to zero.

Completion Upon completion of FLOW.CONTROL, the adapter generates a COMMAND.STATUS interrupt, sets SSB_CMD to >001D, and updates SSB_PARM_0 with COMMAND_STATUS. The COMMAND.STATUS takes on values as shown in Table 4–80. The STATION_ID associated with the command is returned in SSB _PARM_1.

Table 4–80. FLOW.CONTROL Return Codes

Value	**Explanation**
>0000	Good completion
>0040	Invalid STATION_ID

Command LLC.STATISTICS

Description LLC.STATISTICS is used to read the counters and statistics for a link station. The counters may also be reset with this command. The counters and the MAC header are written to attached system memory at locations pointed to in the statistics parameter block.

A COMMAND.REJECT interrupt is issued if the adapter has not been opened or if the LLC interface has not been enabled.

Initiation

	Byte + 0	Byte + 1
SCB_CMD	>001E	
SCB_PARM_0	STAT_PARM_BLOCK (High)	
SCB_PARM_1	STAT_PARM_BLOCK (Low)	

Completion

SSB_CMD	>001E
SSB_PARM_0	COMMAND_STATUS
SSB_PARM_1	STATION_ID
SSB_PARM_2	>0000

Initiation The SCB_PARM_0 and SCB_PARM_1 fields contain a 32-bit pointer to the STAT_PARM_BLOCK, which is shown in Figure 4–44. Table 4–81 gives the detailed field definitions. Table 4–82 illustrates the format of the data that is transferred to the attached system upon execution of the LLC.STATISTICS command.

Figure 4–44. STAT_PARM_BLOCK

	Byte + 0	Byte + 1
+ 0	STATION_ID	
+ 2	BUFFER_LENGTH	
+ 4	BUFF_ADDRESS (High)	
+ 6	BUFF_ADDRESS (Low)	
+ 8	ACTUAL_BUFF_LENGTH	
+ 10	OPTIONS	

Table 4–81. STATISTICS_PARAMETER_BLOCK Field Descriptions

Field	Definition
STATION_ID	This parameter specifies the link station whose statistics are to be written to attached system memory.
BUFFER_LENGTH	This field contains the length of the buffer pointed to by BUFF_ADDRESS.
BUFF_ADDRESS	This field contains a 32-bit pointer to a buffer in attached system memory where the statistics data is to be written.
ACTUAL_BUFF_LENGTH	The actual length of the statistics data that was requested. This value is returned by the adapter. If the value is greater than the value of BUFFER_LENGTH, then not all of the statistics have been transferred.
OPTIONS	When the statistics parameter block is created, this 16-bit parameter is set by the attached system to indicate actions to be taken by the adapter. The OPTIONS bits to be set by the attached system are defined as follows: Bit 0 RESET: If this bit is set to 1, bytes 0—7 are reset to zero after this command is executed. Bits 1—15 RESERVED: These bits are ignored by the adapter but must be set to zero.

Table 4–82. LLC Statistics Data

Offset	Definition
Bytes 0—1	Number of I-frames transmitted.
Bytes 2—3	Number of I-frames received.
Byte 4	Number of I-frames received with errors.
Byte 5	Number of I-frames transmitted ending in error.
Bytes 6—7	Number of times T1 timer expired when not transferring data.
Byte 8	Last command/response received. LLC control byte 0.
Byte 9	Last command/response sent.

Table 4–82. LLC Statistics Data (Continued)

Offset	Definition
Byte 10	Link primary state. Bit values are shown below. Bit 0: Closed Bit 1: Disconnected Bit 2: Disconnecting Bit 3: Opening Bit 4: Resetting Bit 5: FRMR Sent Bit 6: FRMR Received Bit 7: Opened
Byte 11	Link secondary state. Bit values are shown below. Bit 0: Checkpointing Bit 1: Local Busy (user set) Bit 2: Local Busy (buffer set) Bit 3: Remote Busy Bit 4: Rejection Bit 5: Clearing Bit 6: Dynamic Windowing Algorithm Running Bit 7: Reserved
Byte 12	Send state variable (V(S)).
Byte 13	Receive state variable (V(R)).
Byte 14	Last received N(R).
Byte 15	Length of MAC_HEADER used in transmitting I-frames. This length includes the AC through the RI fields.
Bytes 16—47	MAC_HEADER used in transmitting I-frames.

Completion Upon completion of LLC.STATISTICS, the adapter generates a COMMAND.STATUS interrupt, sets SSB_CMD to >001E, and updates SSB_PARM_0 with COMMAND_STATUS. The COMMAND_STATUS takes on the values according to Table 4–83. The STATION_ID associated with the command is returned in SSB_PARM_1.

Table 4–83. LLC.STATISTICS Return Codes

Value	Explanation
>0000	Good completion.
>0015	Log data lost due to inadequate space; log reset. If the length is less than eight, then the counters are not reset.
>0040	Invalid STATION_ID.

Command DIR.INTERRUPT

Description DIR.INTERRUPT can be used to cause the adapter to interrupt the attached system. The adapter reads the SCB, writes a response SSB, and then interrupts the attached system with a COMMAND.STATUS interrupt.

A COMMAND.REJECT is issued if the LLC interface has not been enabled.

Initiation

	Byte + 0	Byte + 1
SCB_CMD	>001F	
SCB_PARM_0	>0000	
SCB_PARM_1	>0000	

Completion

SSB_CMD	>001F	
SSB_PARM_0	COMMAND_STATUS	
SSB_PARM_1	>0000	
SSB_PARM_2	>0000	

Initiation The SCB_PARM_0 and SCB_PARM_1 fields are ignored but must be set to zero.

Completion Upon completion of DIR.INTERRUPT, the adapter generates a COMMAND.STATUS interrupt, sets SSB_CMD to >001F, and updates SSB_PARM_0 with COMMAND_STATUS. The COMMAND_STATUS takes on values according to Table 4–84.

Table 4–84. DIR.INTERRUPT Return Codes

Value	Explanation
>0000	Good completion

Command LLC.REALLOCATE

Description The LLC.REALLOCATE command reallocates memory for link control blocks. Additional link control blocks can be allocated for a SAP. Alternately, link control blocks previously allocated to a SAP can be returned to the adapter pool. This effectively alters the STATION_COUNT parameter of the OPEN.SAP command.

A COMMAND.REJECT is issued if the LLC interface has not been enabled.

Initiation

	Byte + 0	Byte + 1
SCB_CMD	>0021	
SCB_PARM_0	STATION_ID	
SCB_PARM_1	OPTIONS	STATION_COUNT

Completion

SSB_CMD	>0021	
SSB_PARM_0	COMMAND_STATUS	
SSB_PARM_1	STATION_ID	
SSB_PARM_2	ADAPTER_COUNT	SAP_COUNT

Initiation The SCB_PARM_0 field contains the STATION_ID to or from which the control blocks are to be allocated. The SCB_PARM_1 field contains two one-byte parameters described in Table 4–85.

Table 4–85. SCB_PARM_1 Field Description

Field	Description	
OPTIONS	Bit 0	If this bit is equal to zero, then the link control blocks are to be taken from the adapter pool and added to the SAP pool. If this bit is equal to one, then the link control blocks are to be taken from the SAP pool and added to the adapter pool.
	Bits 1—7	RESERVED: Must be set to zero.
STATION_COUNT	This byte parameter is the number of link control blocks to be reallocated.	

Completion Upon completion of LLC.REALLOCATE, the adapter generates a COMMAND.STATUS interrupt, sets SSB_CMD to >0021, and updates SSB_PARM_0 with COMMAND_STATUS (See Table 4–86). SSB_PARM_1 is updated with the STATION_ID of the station or SAP in question. SSB_PARM_2 is updated with the ADAPTER_COUNT and the SAP_COUNT parameters described in Table 4–87.

Table 4–86. COMMAND_STATUS Return Codes

Value	Explanation
>0000	Good completion.
>0040	Invalid STATION_ID.

Table 4–87. LLC.REALLOCATE Return Parameters

Field	Definition
ADAPTER_COUNT	The number of link control blocks available in the adapter pool (not assigned to any SAP) when the command completes.
SAP_COUNT	The number of link control blocks assigned to the SAP designated by the STATION_ID when the command completes. This byte is valid only if the COMMAND_STATUS was >0000.

LLC 0022 TIMER.SET Command

Command TIMER.SET

Description The TIMER SET command is used to enable and disable timer interrupts generated by the adapter to the attached system. The timer duration is passed as a parameter of this command and specifies the number of 10-ms intervals to be used for the timer period. If the TIMER.SET command is issued to enable a timer that has already been started, the old request is canceled and the timer restarted with the new timer period.

A COMMAND.REJECT is issued if the LLC interface has not been enabled.

Initiation

	Byte + 0	Byte + 1
SCB_CMD	>0022	
SCB_PARM_0	>00	TIMER_COUNT
SCB_PARM_1	>0000	

Completion

SSB_CMD	>0022
SSB_PARM_0	COMMAND_STATUS
SSB_PARM_1	>0000
SSB_PARM_2	>0000

Initiation The LSB of the SCB_PARM_0 field contains the timer count described below. The MSB of the SCB_PARM_0 and the SCB_PARM_1 fields is ignored and should be set to zero.

TIMER.SET Parameter Description

The TIMER_COUNT parameter specifies the number of 10-ms timer intervals that make up the requested timer period. A value from >01 to >FF starts or restarts the timer. A value of >00 stops the timer.

Completion Upon completion of TIMER.SET, the adapter generates a COMMAND.STATUS interrupt, sets the SSB_CMD to >0022, and updates SSB_PARM_0 with COMMAND.STATUS. The SSB_PARM_1 and SSB_PARM_2 fields should be ignored. The COMMAND_STATUS takes on values according to Table 4–88.

Table 4–88. TIMER.SET Return Codes

Value	Explanation
>0000	Good completion

Command TRANSMIT.I.FRAME.REQUEST (For Host-Based I-Frames)

Description The TRANSMIT.I.FRAME.REQUEST command is designed to be used by the host system anytime a host-based I-frame must be sent through an existing link station. Before building a transmit list chain for the host-based I-frame, the host system must first issue this command to the adapter. If the state of the link is valid, the adapter responds by assigning to this particular I-frame request a one- byte frame correlator. At this point, the host system should queue the I-frame request and the associated frame correlator to a temporary queue.

Once the adapter determines that the link state is appropriate for sending I-frames, it requests that the host system build a transmit list chain for an I-frame. The host system can determine the proper I-frame to build by matching the frame correlator given by the adapter in its request to the frame correlators in the host system's temporary queue.

Note that the adapter may request a particular I-frame several times before the request is acknowledged, and the frame correlator may be dequeued from the host system's temporary queue. This is due to a variety of link error conditions and recovery that the link station must perform as part of the IEEE 802.2 protocols.

A COMMAND.REJECT interrupt is issued if the adapter has not been opened, if the LLC interface has not been enabled, or if the host-based I-frame transmit bit has not been set in the open adapter options.

Initiation

	Byte + 0	Byte + 1
SCB_CMD	>0023	
SCB_PARM_0	STATION_ID	
SCB_PARM_1	>0000	

Completion

SSB_CMD	>0023	
SSB_PARM_0	COMMAND_STATUS	
SSB_PARM_1	STATION_ID	
SSB_PARM_2	FRAME_CORRELATOR	RESERVED

Initiation The STATION_ID of the link station that the I-frame will be sent from is placed in SCB_PARM_0. SCB_PARM_1 is ignored by the adapter.

Completion Upon completion of the TRANSMIT.I.FRAME.REQUEST command, the adapter generates a COMMAND.STATUS interrupt, sets the SSB_CMD to >0023, and updates the SSB_PARM_0 with COMMAND_STATUS. The COMMAND_STATUS values and their meanings are given in Table 4–89.

The STATION_ID of the requested link station is returned in SSB_PARM_1. The FRAME_CORRELATOR of the requested frame is returned in the most significant byte of SSB_PARM_2.

Table 4–89. TRANSMIT.I.FRAME.REQUEST Return Codes

Value	Explanation
>0000	Good completion/build I-frame transmit list chain. The request was successful, and the host system should build the transmit list for the requested I-frame, queue it to the adapter transmit list chain, and issue a new transmit command if necessary. The host system should save the FRAME_CORRELATOR in case the adapter must later request that the I-frame's list be reissued to the adapter transmit list chain.
>0025	Maximum commands exceeded. There are currently a maximum of 127 outstanding I-frame requests (unacknowledged I-frames) for this link station. Wait for I-frames to be acknowledged before issuing any more requests.
>0040	Invalid STATION_ID.
>0041	Protocol error. The link station is not in the proper state for transmitting I-frames.
>00FF	Good completion. The adapter has assigned a FRAME_CORRELATOR to the request and will inform the host at a later time, using the TRANSMIT_COMMAND interrupt. when it is ready for the transmit list chain to be built for this I-frame. The host system should not issue another I.FRAME.REQUEST command for the same link station after receiving this status. The host should wait until the adapter signals the host with a Transmit Status interrupt, with the ready bit set, to proceed with the next I.FRAME.REQUEST, and to build the transmit list for I-frame.

Chapter 5

Customer Information

This chapter includes general information on TMS380 physical characteristics, and parts ordering. Topics covered in this chapter include:

Section **Page**

5.1 TMS380 Adapter Software Licensing Procedures 5-2
5.2 TMS380 Software Ordering Information 5-6
5.3 Mechanical Package Information 5-8
5.4 TMS380 Family Numbering and Symbol Conventions 5-14
5.5 TMS380 Pricing and Delivery 5-19

5.1 TMS380 Adapter Software Licensing Procedures

All TMS380 software packages with the exception of the development kit, require a software license. Depending on the software package, royalty payments may be required. This section describes the steps that are taken for a customer to receive his requested software.

5.1.1 Second Generation Software

This section specifically describes the steps that are taken for the second-generation software that requires a license.

5.1.1.1 MAC-Only License

This section describes the sequence of steps that are taken to obtain a second generation MAC-only software license agreement.

1) Customer contacts Texas Instruments salesperson requesting the required product (TMS380C16MAC).
2) TI salesperson initiates the license by filling out the mask on the M/380 file.
3) The TI Software License Coordinator fulfills the license request and sends it to the client.
4) The client signs the license agreement and returns it to TI.
5) The client places an order for the integration master. This is the original software that copies are made from. The integration master contains the MAC object code and download procedures. The integration master part number is TMS380C16MAC.
6) TI ships the integration master to the client, after verification of a signed license on file, and invoices them.

Note:

If a client has a second generation LLC license, they do not need to obtain a MAC-only license unless they have some products that use MAC software only.

5.1.1.2 LLC License

This section describes the sequence of steps that are taken to obtain a second generation LLC software license agreement. Users of this software receive the MAC software included in the LLC software. They do not need to purchase or license the TMS380C16MAC software separately. This software requires royalty payments.

1) Customer contacts TI salesperson requesting the required product (TMS380C16LLC1, TMS380C16LLC2, or TMS380C16MU).
2) TI salesperson initiates the license, or first generation amendment, by filling out the mask on the M/380 file, and selecting either price option1 or price option 2.
3) The TI Software License Coordinator fulfills the license request and sends it to to the client.
4) The client signs the license agreement, or amendment, for either option 1 (high volume) or option 2 (low volume) specifying their projected volumes and returns it to TI.
5) The client places an order for the integration master. This is the original software that copies are made from. The integration master contains the source code for the adapter handler. The integration master has three part numbers based on the option selected.
   - TMS380C16LLCMU (Second generation amendment to first generation license.)
   - TMS380C16LLCM1 (LLC license option 1)
   - TMS380C16LLCM2 (LLC license option 2)
6) TI ships the integration master to the client, after verification of a signed license on file, and invoices them.
7) The client has the right to make copies of the software in accordance with the pricing option they selected. On a quarterly basis, the client is responsible to report to TI the number of copies of software made that quarter.

Note:

If a client is a first generation licensee, and wants a second-generation LLC license, keeping their pricing option and yearly volumes the same, then they only need to sign the license ammendment for first to second generation conversion product.

5.1.1.3 Copy All Frames (CAF) License

This section describes the sequence of steps that are taken to obtain a Copy All Frames (CAF) software license agreement. The CAF software is an extension of the MAC-only software. A separate MAC-only software license is not required.

1) Customer contacts TI salesperson requesting the required product (TMS380C16CAF).
2) TI salesperson initiates the license by filling out the mask on the M/380 file.
3) The TI Software License Coordinator fulfills the license request and sends it to the client.
4) The client signs the license agreement, and returns it to TI.
5) The client places an order for the integration master. This is the original software that copies are made from. The integration master contains the CAF object code and download procedures.The integration master part number is TMS380C16CAF.
6) TI ships the integration master to the client, after verification of a signed license on file, and invoices them.

5.1.2 First Generation Software

This section specifically describes the steps that are taken for the first-generation software that requires a license.

5.1.2.1 LLC License

This section describes the sequence of steps that are taken to obtain a first generation LLC software license agreement. This software requires royalty payments.

1) Customer contacts TI salesperson requesting the required product (TMS380C16LLC–1 or TMS380C16LLC–2).
2) TI salesperson initiates the license by filling out the mask on the M/380 file, and selects either price option 1 or price option 2.
3) The TI Software License Coordinator fulfills the license request and sends it to the client.
4) Client signs the license agreement for either option 1 (high volume) or option 2 (low volume), specifying their projected volumes. and returns it to TI.
5) The client places an order for the integration master. This is the original software that copies are made from. The integration master contains the source code for the adapter handler. The integration master has two part numbers based on the option selected.
   - TMS380LLCM–1 (LLC license option 1)
   - TMS380LLCM–2 (LLC license option 2)
6) TI ships the integration master to the client, after verification of a signed license on file, and invoices them.
7) The client has the right to make copies of the software in accordance with the pricing option they selected. On a quarterly basis, the client is responsible to report to TI the number of copies of the software made that quarter.

5.2 TMS380 Software Ordering Information

All TMS380 software requires that a license agreement be signed.

5.2.1 Second Generation Software Options

5.2.1.1 MAC-Only Software

This software is for customers who only need the high-level IEEE 802.5 medium access control protocol services. A one-time reproduction license fee is required.

Part Number	Description
TMS380C16MAC	Medium access control software

5.2.1.2 LLC Software

This software is for customers who need the high-level IEEE 802.5 MAC and IEEE802.2 LLC protocol services to interface their token ring hardware to industry-standard communications software. Users of this software receive the MAC software included in the LLC software. They do not need to purchase or license the TMS380C16MAC software separately.

Part Number	Description
TMS380C16LLCM1	Second generation MAC/LLC high volume software
TMS380C16LLCM2	Second generation MAC/LLC low volume software

5.2.1.3 Copy All Frames Software

Copy All Frames (CAF) software is used primarily in promiscuous protocol analyzers. A one-time reproduction license fee is required. The CAF software is an extension of the MAC-only software. A separate MAC-only software package is not required.

Part Number	Description
TMS380C16CAF	Copy all frames software

5.2.1.4 LAN Development Kit

This kit contains one TMX380C16 16-Mbps Token Ring COMMprocessor and two separate, stand alone software packages for development purposes. One package contains MAC-only software, and the other contains LLC/MAC software. This kit is for developmental purposes only and the software is not to be used for production. **The customer must acquire the correct license and software for mass production purposes.**

Part Number	Description
TMDX380C16LDK	LAN development kit

5.2.2 First Generation to Second Generation Conversion Software

This part number allows first-generation LLC licensees to convert to a second-generation license, providing the pricing option and yearly volume remain the same. For the sequence of steps to obtain this ammendment, see section 5.1.1.2

Part Number	Description
TMS380C16LLCMU	Conversion software from first generation to second generation

5.2.3 First Generation Software Options

5.2.3.1 LLC Software

This software is for customers who need the high-level IEEE 802.5 MAC and IEEE802.2 LLC protocol services to interface their token ring hardware to industry-standard communications software. Users of this software receive the MAC software included in the LLC software. They do not need to purchase or license the TMS380C16MAC software separately.

Part Number	Description
TMS380LLCM–1	First generation LLC high volume software
TMS380LLCM–2	First generation LLC low volume software

5.2.3.2 MAC-Only Software

This software is for customers who only need the high level IEEE 802.5 Medium Access Control protocol services. This program is incorporated into the protocol handler device (TMS38021). The purchase of additional software is not required. This version of the MAC code does not support copy all frames.

5.3 Mechanical Package Information

The TMS380 Token Ring devices are assembled in four package types according to the type of material and outline used for the package. These package types are:

- ❑ Plastic dual-inline package (DIP)
- ❑ Plastic leaded chip carrier (PLCC)
- ❑ Quad Flat Package (QFP)
- ❑ Grid Array Package (GBP)

Package types are designated in the device symbol by the suffix of the standard device number. Table 5–1 indicates the package type, suffix indicator, and devices supported on that package type.

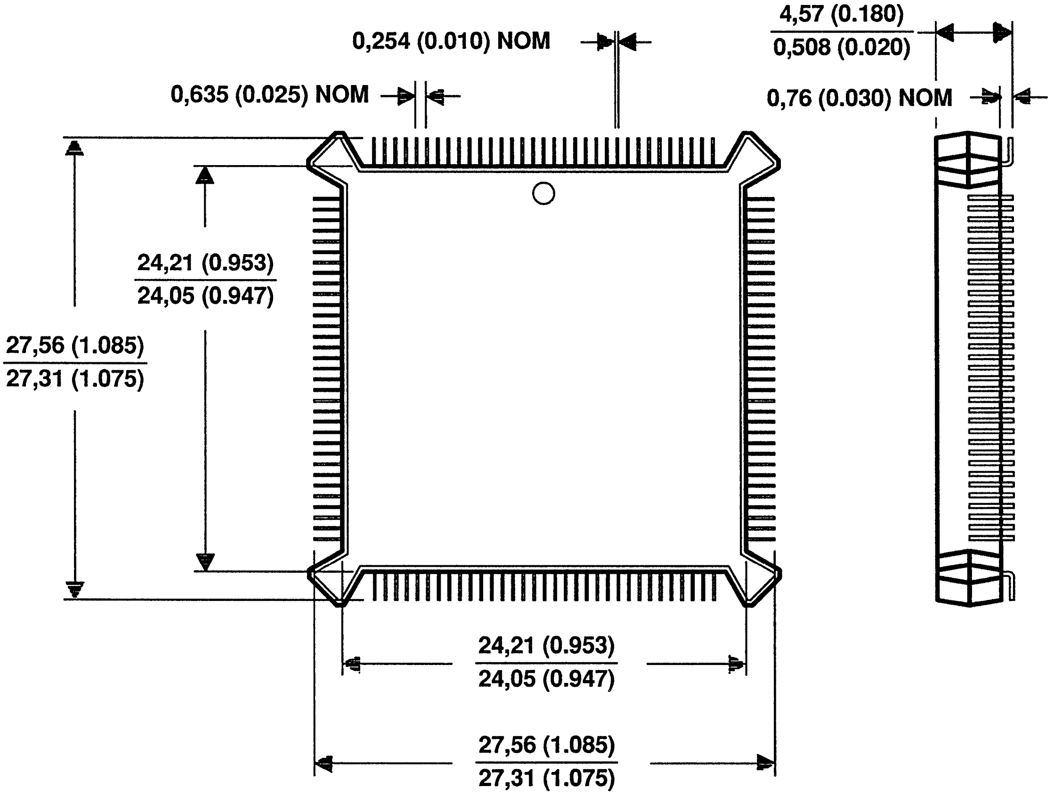
Table 5–1. Package Types

Package Type	Suffix Indicator	Family Members	
132-pin plastic QFP (25-mil pin spacing)	PQ	TMS380C16	Second Generation
44-pin PLCC (50-mil pin spacing)	FN	TMS38053 and TMS380SRA	
48-pin plastic DIP (100-mil pin spacing)	N	TMS38010 and TMS38021	First Generation
100-pin GBP (50-mil pin spacing)	GB	TMS38030	
132-pin QFP (25-mil pin spacing)	PQ	TMS38030	
22-pin plastic DIP (100-mil pin spacing)	N	TMS38051	
20-pin plastic DIP (100-mil pin spacing)	N	TMS38052	

5.3.1 Second Generation Packages

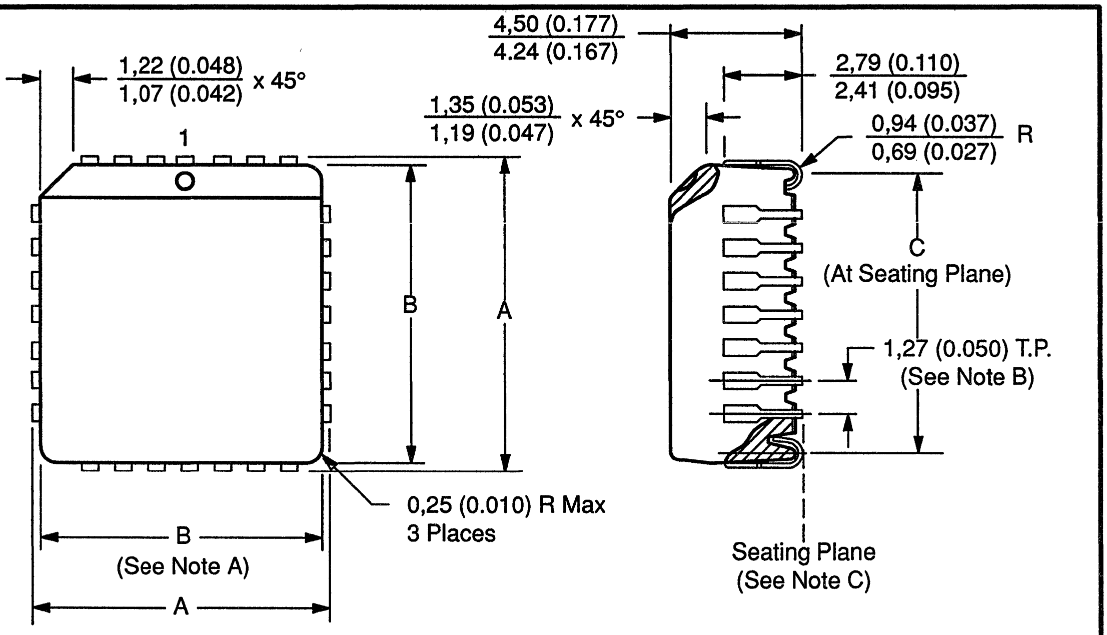
The following figures in this section show the mechanical package information for the second generation chipset.

Figure 5–1. TMS380C16 132-Pin Plastic Quad Flat Package, 25-Mil Pin Spacing (Type PQ Package Suffix)

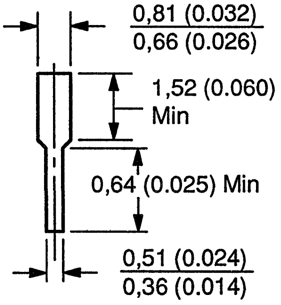


ALL LINEAR DIMENSIONS ARE IN MILLIMETERS AND PARENTHETICALLY IN INCHES

Figure 5–2. TMS38053 and TMS380SRA Plastic-Leaded Chip Carrier Package, 50-Mil Pin Spacing (Type FN Package Suffix)



JEDEC OUTLINE	NO. OF PINS	A		B		C	
		MIN	MAX	MIN	MAX	MIN	MAX
MO-047AA	20	9,78 (0.385)	10,03 (0.395)	8,89 (0.350)	9,04 (0.356)	7,87 (0.310)	8,38 (0.330)
MO-047AB	28	12,32 (0.485)	12,57 (0.495)	11,43 (0.450)	11,58 (0.456)	10,41 (0.410)	10,92 (0.430)
MO-047AC	44	17,40 (0.685)	17,65 (0.695)	16,51 (0.650)	16,66 (0.656)	15,49 (0.610)	16,00 (0.630)
MO-047AE	68	25,02 (0.985)	25,27 (0.995)	24,13 (0.950)	24,33 (0.956)	23,11 (0.910)	23,62 (0.930)
MO-047AF	84	30,10 (1.185)	30,35 (1.195)	29,21 (1.150)	29,41 (1.158)	27,69 (1.090)	28,70 (1.130)



Lead Detail

All dimensions and notes to the specified JEDEC outline apply

ALL LINEAR DIMENSIONS ARE IN MILLIMETERS AND PARENTHETICALLY IN INCHES

Notes: A) Centerline of center pin each side is within 0,10 (0.004) of package centerline as determined by dimention B.

B) Location of each pin is within 0,127 (0.005) of true position with respect to center pin on each side.

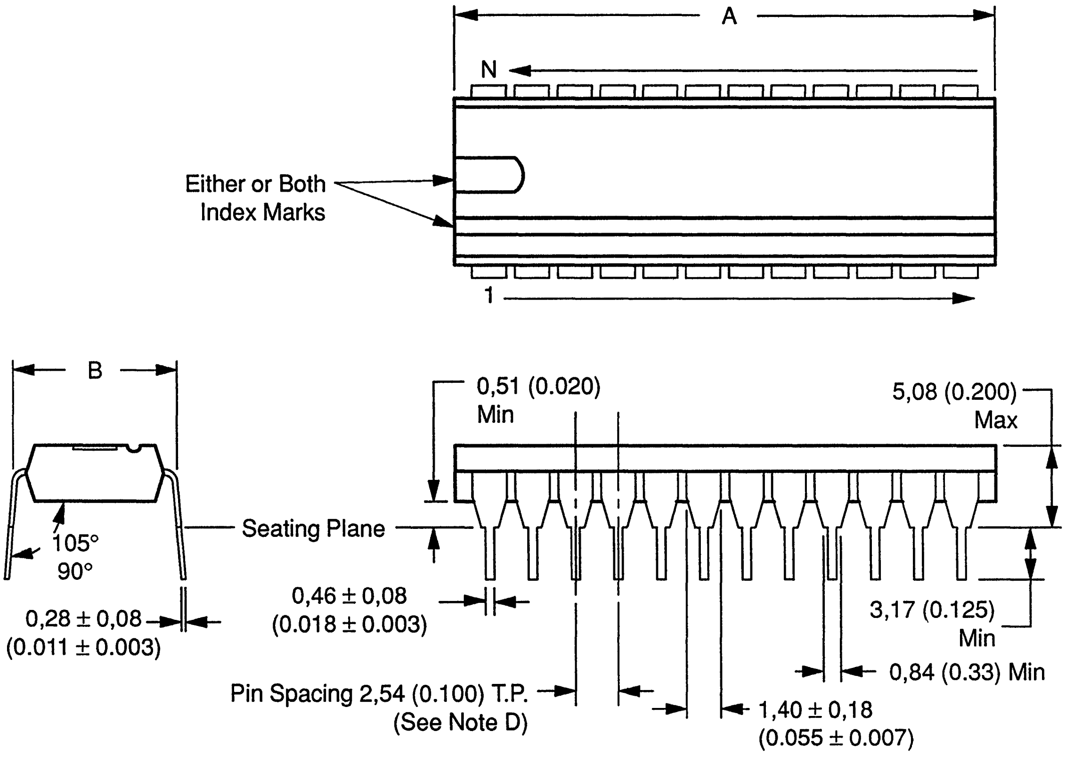
C) The lead contact points are planar within 0,10 (0.004).

D) Only the 44-pin package is offered in the TMS380 family.

5.3.2 First Generation Packages

The following figures in this section show the mechanical package information for the first generation chipset.

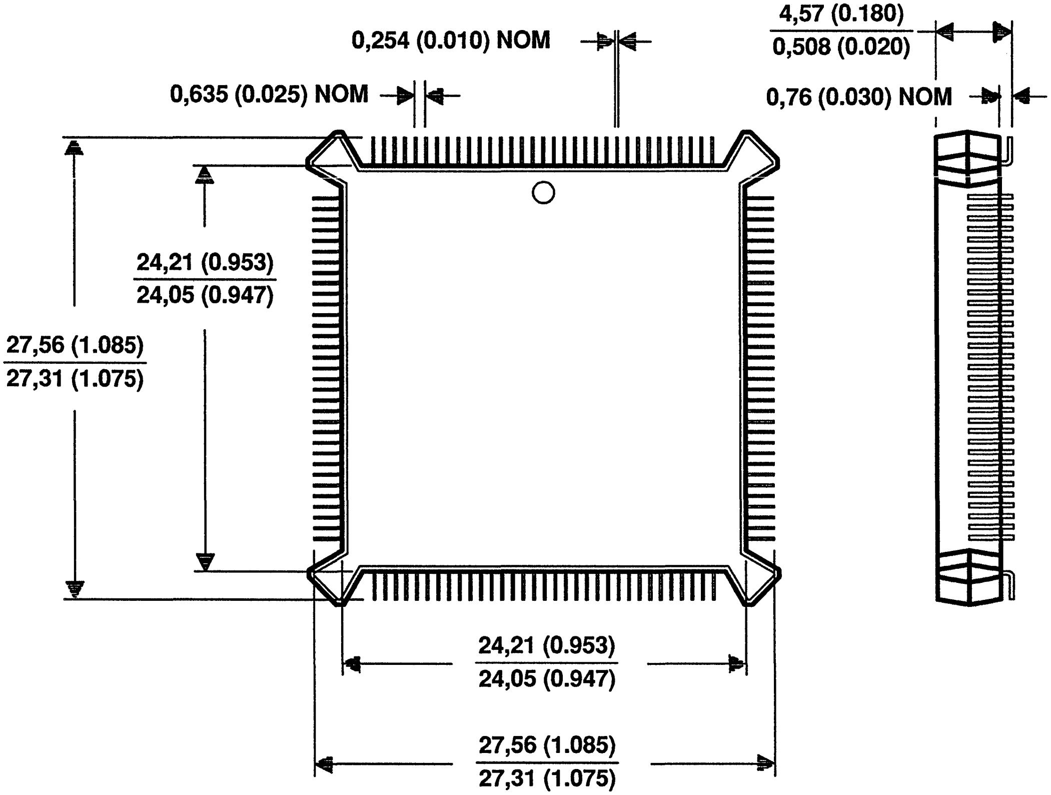
Figure 5–3. TMS38010, TMS38021, TMS38051, TMS38052 Plastic Dual-Inline Packages, 100-Mil Pin Spacing (Type N Package Suffix)



N	A (MAX)	B
20	24, 77 (0.975)	7, 62 ± 0,25 (0.300 ± 0.10)
22	27, 94 (1.100)	10, 16 ± 0,25 (0.400 ± 0.10)
48	62, 2 (2.45)	15, 24 ± 0,25 (0.600 ± 0.10)

ALL LINEAR DIMENSIONS ARE IN MILLIMETERS AND PARENTHETICALLY IN INCHES

Figure 5–4. TMS38030 132-Pin Quad Flat Package, 25-Mil Pin Spacing (Type PQ Package Suffix)



ALL LINEAR DIMENSIONS ARE IN MILLIMETERS AND PARENTHETICALLY IN INCHES

Figure 5–5. TMS38030 100-Pin Grid Array Package, 50-Mil Pin Spacing (Type GB Package Suffix)

Not available at time of printing.

5.4 TMS380 Family Numbering and Symbol Conventions

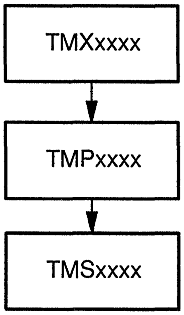
All TMS380 devices are marked with information as to the type, package, copyright date (if applicable), place of manufacture, and manufacturing data.

5.4.1 Device Prefix Designators

To provide expeditious system evaluations by customers during the product development cycle, TI assigns a prefix designator with three options: TMX, TMP, and TMS.

TMX, TMP, and TMS are representative of the evolutionary stages of product development from engineering prototypes through fully qualified production devices. Figure 5–6 depicts this evolutionary development flowchart. Production devices shipped by TI have the TMS designator signifying that they have demonstrated the high standards of TI quality and reliability.

Figure 5–6. Development Flowchart



Experimental devices that may not be representative of the final device's electrical specifications and have not completed reliability verification.

Devices that conform to the electrical specifications but have not completed quality and reliability verification.

Fully qualified production devices.

TMX devices are shipped against the following disclaimer:

1) Experimental product and its reliability has not been characterized.
2) Product is sold "as is".
3) Product is not warranted to be exemplary of final production version if or when released by TI.

TMP devices are shipped against the following disclaimer:

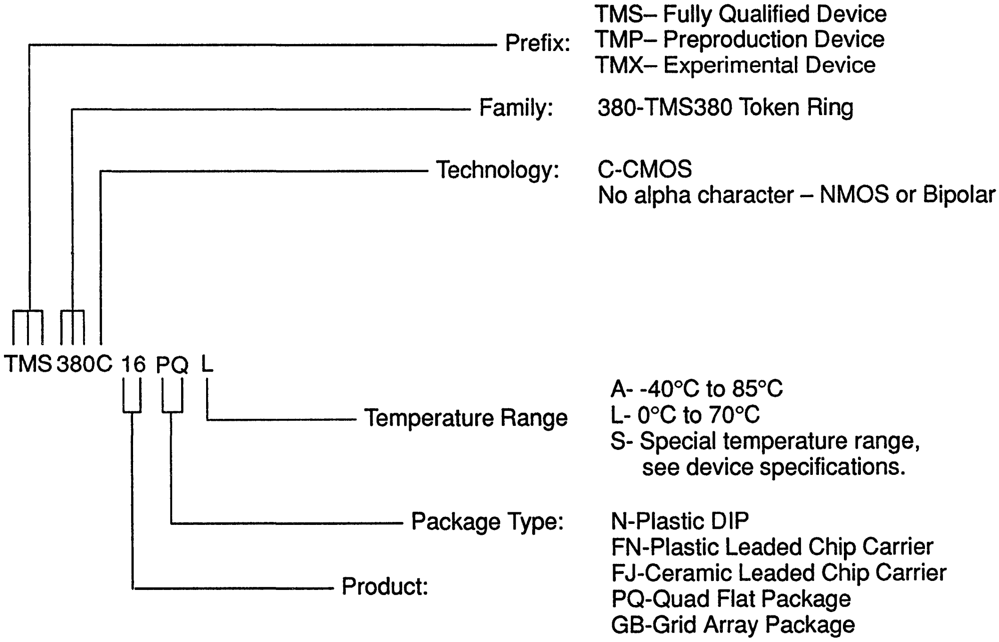
1) Customer understands that the product purchased hereunder has not been fully characterized and the expectation of reliability cannot be defined; therefore, TI standard warranty refers only to the device's specifications.
2) No warranty of merchantability or fitness is expressed or implied.

TMS devices have been fully characterized and the quality and reliability of the device has been fully demonstrated. TI's standard warranty applies.

5.4.2 Device Numbering Convention

Figure 5–7 illustrates the numbering and symbol nomenclature for the TMS380 family.

Figure 5–7. TMS380 Family Nomenclature



Note: The TMS38030 also appends a –8 or –10 to indicate the clock speed of the device.

5.4.3 Device Symbols

The device symbolization of the TMS380 family members can be shown as follows:

5.4.3.1 TMS380 Second Generation Devices

Figure 5–8. TI Standard Symbolization for TMS380C16 Device in 132-Pin PQ-Type Package

Line 1:	(a) TMS380C16PQL	
Line 2:	(b)	(c) 12345678
Line 3:		(d) FRSYYWW
Line 4:	(e) INDY	(f) USA

Key:
(a) Standard device part number
(b) Texas Instruments Trademark
(c) Lot code
(d) Tracking mark and date code
(e) Assembly site
(f) Country of origin

Figure 5–9. TI Standard Symbolization for TMS38053 Device in 44-Pin FN Type Package

Line 1:	(a) TMS38053FNL	
Line 2:	(b)	(c) 12345678
Line 3:		(d) FRSYYWW
(Backside)		(e) Philippines

Key:
(a) Standard device part number
(b) Texas Instruments Trademark
(c) Lot code
(d) Tracking mark and date code
(e) Assembly site (bottom of package)

Figure 5–10. TI Standard Symbolization for TMS380SRA Device in 44-Pin FN Type Package

Not available at time of printing.

5.4.3.2 TMS380 First Generation Devices

Figure 5–11. TI Standard Symbolization for TMS38010 Device in 48-Pin N-Type Package

Line 1:	(a)	(b) TMS38010NL
Line 2:		(c) FRSYYWW
Line 3:	(d) 12345678	(e) Philippines

Key:
(a) Texas Instruments Trademark
(b) Standard device part number
(c) Tracking mark and date code
(d) Lot code
(e) Assembly site

Figure 5–12. TI Standard Symbolization for TMS38021 Device in 48-Pin N-Type Package

Line 1:	(a)	(b) TMS38021NL	
Line 2:		(c) RCO127	(d) FRSYYWW
Line 3:		(e) ©1986TI/IBM	
Line 4:	(f) 12345678	(g) Philippines	

Key:
(a) Texas Instruments Trademark
(b) Standard device part number
(c) ROM code revision
(d) Tracking mark and date code
(e) TI/IBM microcode copyright
(f) Lot code
(g) Assembly site

Figure 5–13. TI Standard Symbolization for TMS38030 Device in 132-Pin PQ-Type Package

Line 1:	(a)	
Line 2:	(b) TMS38030PQL–10	
Line 3:	(c) FRSYYWW	
Line 4:	(d) 12345678	
Line 5:	(e) INDY	(f) USA

Key:
(a) Texas Instruments Trademark
(b) Standard device part number
(c) Tracking mark and date code
(d) Lot code
(e) Assembly site
(f) Country of origin

Figure 5–14. TI Standard Symbolization for TMS38030 Device in 100-Pin GB-Type Package

Line 1:	(a)
Line 2:	TMS
Line 3:	(b) 38030GBL–10
Line 4:	(c) FRSYYWW
Line 5:	(d) 12345678
Line 6:	(e) Phillipines

Key:
(a) Texas Instruments Trademark
(b) Standard device part number
(c) Tracking mark and date code
(d) Lot code
(e) Assembly site

Figure 5–15. TI Standard Symbolization for TMS38051 Device in 22-Pin N-Type Package

Line 1:	(a)	(b) TMS38051NL
Line 2:		(c) FRSYYWW
Line 3:	(d) 12345678	(e) Philippines

Key:
(a) Texas Instruments Trademark
(b) Standard device part number
(c) Tracking mark and date code
(d) Lot code
(e) Assembly site

Figure 5–16. TI Standard Symbolization for TMS38052 Device in 20-Pin N-Type Package

Line 1:	(a)	(b) TMS38052NL
Line 2:		(c) FRSYYWW
Line 3:	(d) 12345678	(e) Philippines

Key:
(a) Texas Instruments Trademark
(b) Standard device part number
(c) Tracking mark and date code
(d) Lot code
(e) Assembly site

5.5 TMS380 Pricing and Delivery

To obtain pricing and delivery information regarding TMS380 Token Ring products or any TI semi-conductor product, please contact your local TI sales office, or authorized distributor. A list of these is included at the back of this document.

Appendix A

TMS380C16 Token-Ring COMMprocessor Data Sheet

The TMS380C16 Token-Ring COMMprocessor data sheet was not available at the time this book was printed. For information or assistance, call the TMS380 Customer Support Line at (713) 274–4027.

Appendix B

TMS38053 Ring Interface Data Sheet

The TMS38053 Ring Interface data sheet was not available at the time this book was printed. For information or assistance, call the TMS380 Customer Support Line at (713) 274–4027.

Appendix C

TMS380SRA Source Routing Accelerator Data Sheet

The TMS380SRA Source Routing Accelerator data sheet was not available at the time this book was printed. For information or assistance, call the TMS380 Customer Support Line at (713) 274–4027.

Appendix D

MAC Frame Summary

This appendix provides a summary of the MAC frames discussed in this user's guide. Explanations of the letter symbols used in the tables throughout this appendix are listed below.

RS	=	ring station
REM	=	ring error monitor
RPS	=	ring parameter server
NM	=	network manager
F(REM)	=	ring error monitor functional address
F(RPS)	=	ring parameter server functional address
F(NM)	=	network manager functional address
Target	=	a specific destination address
A	=	subvector always transmitted
O	=	optional subvector
R	=	required subvector
All stations	=	all stations address (broadcast)
N	=	subvector not syntax-checked
ACK	=	frame elicits response MAC frame
(NACK)	=	frame sends responses only if error found in syntax checking
UNA	=	upstream neighbor's address

Table D–1. MAC Frame Major Vectors

M–V CMD	M–V (Frame) Name and XMIT Type	Destination Class	Source Class	Destination Address	Subvectors		
					R/O		
					XMT	RCV	Name
>00	Response (Type = Resp.)	Source Class of received frame	>0	Source Address of received frame	A A	N N	Correlator Response Code
>02	Beacon (Type = Orig.) FC = >02	>0	>0	All Stations	A A A	R O R	Beacon Type Physical Drop Number UNA
>03	Claim Token (Type = Orig.) FC = >03	>0	>0	All Stations	A A	O R	Physical Drop Number UNA
>04	Ring Purge (Type = Orig.) FC = >04	>0	>0	All Stations	A A	O R	Physical Drop Number UNA
>05	Active Monitor Present (AMP) FC = >05	>0	>0	All Stations	A A	O R	Physical Drop Number UNA
>06	Standby Monitor Present (SMP) (Type = Resp.) FC = >06	>0	>0	All Stations	A A	O R	Physical Drop Number UNA
>07	Duplicate Address Test (Type = Orig.) FC = >01	>0	>0	Target			None
>08	Lobe Media Test (Type = Orig.)	>0	>0	Zero	A	N	Wrap Data
>09	Transmit Forward	>0	Any	Target		R	Frame Forward
>0B	Remove Ring Station FC = >01	>0	>4	Target			None

Table D–1. MAC Frame Major Vectors (Continued)

M–V CMD	M–V (Frame) Name and XMIT Type	Destination Class	Source Class	Destination Address	Subvectors		
					R/O		
					XMT	RCV	Name
>0C	Change Parameters (ACK)	>0	>4	Target		O O O O O O	Correlator Local Ring Number Assigned Physical Drop Number Soft Error Report Timer Value Enabled Function Classes Allowed Access Priority
>0D	Initialize Ring Station (ACK)	>0	>5	Target		O O O O	Correlator Local Ring Number Assigned Physical Drop Number Soft Error Report Timer Value
>0E	Request Station Address	>0	>4	Target		O	Correlator
>0F	Request Station State	>0	>4	Target		O	Correlator
>10	Request Station Attachment	>0	>4	Target		O	Correlator
>20	Request Initialization (Type = Orig.)	>5	>0	F(RPS)	A A A		Adapter Software Level UNA Product ID

Table D–1. MAC Frame Major Vectors (Continued)

M–V CMD	M–V (Frame) Name and XMIT Type	Destination Class	Source Class	Destination Address	Subvectors		
					R/O		
					XMT	RCV	Name
>22	Report Station Address (Type = Resp.)	>4	>0	Source Address of Received Request Frame	A A A A A		Correlator Physical Drop Number UNA Group Address Functional Address
>23	Report Station State (Type = Resp.)	>4	>0	Source Address of Received Request Frame	A A A		Correlator Adapter Software Level Adapter Status Vector
>24	Report Station Attachment (Type = Resp.)	>4	>0	Source Address of Received Request Frame	A A A A A		Correlator Product ID Functional Address Authorized Function Class Authorized Access Priority
>25	Report New Monitor (Type = Orig.)	>4	>0	F(NM)	A A A		Physical Drop Number UNA Product ID
>26	Report SUA Change (Type = Resp.)	>4	>0	F(NM)	A A		Physical Drop Number UNA
>27	Report Ring Poll Failure (Type = Orig.) FC = >01	>6	>0	F(REM)	A		Address of Last Ring Poll

Table D–1. MAC Frame Major Vectors (Continued)

M–V CMD	M–V (Frame) Name and XMIT Type	Destination Class	Source Class	Destination Address	Subvectors		
					R/O		
					XMT	RCV	Name
>28	Report Monitor Error (Type = Orig.)	>6	>0	F(REM)	A A A		Physical Drop Number UNA Error Code: –Monitor Error –Duplicate Monitor –Duplicate Address
>29	Report Error (Type = Orig.)	>6	>0	F(REM)	A A A A		Physical Drop Number UNA Isolating Error Counts Nonisolating Error Counts
>2A	Report Transmit Forward	>4	>0	F(NM)	A		Transmit Status Code

Table D–2. MAC Frame Subvectors

Type	Length†	Subvector Name	Value
>01	>04	Beacon Type	Reason for beaconing: >0001 Set recovery mode. >0002 Ring signal loss detected. >0003 Monitor contention failed; no contention frames received. >0004 Monitor contention failed; contention frame(s) received.
>02	>08	UNA	Station's UNA-specific address (all zeroes if unknown).
>03	>04	Local Ring Number	Local ring number of sending adapter.
>04	>06	Assign Physical Drop Number	Physical drop number to be set in the receiving adapter.

Note: † Length in bytes of subvector value.

Table D–2. MAC Frame Subvectors (Continued)

Type	Length†	Subvector Name	Value
>05	>04	Soft Error Report Timer Value	Timeout value (in units of 10 milliseconds) for the adapter's soft error report timer. Valid range is >0000 to >FFFF. The value >0000 sets the time-out value to >10000 units of 10ms.
>06	>04	Enabled Function Classes	Source classes for which the attached product is enabled to transmit. Valid range is b"0000 0000 0000 0000" to b"1111 1111 1111 1111". Each bit 0 to 15 corresponds to function class b"0000" to b"1111" (0 to 15). Bit values of b"1" means function class is enabled. Classes 0, 1, 4, and 5 cannot be enabled.
>07	>04	Allowed Access Priority	Maximum allowed token priority with which the attached product is allowed to transmit is b"xxxx xxxx xxxx xxvv". The x values are ignored, and the vv value is the maximum access priority (0 to 3).
>09	>04	Correlator	Correlator used to relate frames.
>0A	>08	Address of Last Ring Poll	Source address of last AMP or SMP MAC frame before the poll cycle failed.
>0B	>06	Physical Drop Number	Physical drop number of the sending adapter (zero if not set).
>20	>06	Response Code	See Table 2–7.
>21	>04	Reserved	This subvector type is reserved and is equal to zero.
>22	>14	Product ID	This value is not examined by the adapter.
>23	>OC	Adapter Software Level	Level of the sending adapter's software. –Bytes 0—4: Feature code. –Bytes 5—9: EC Level.
>26	>xx	Wrap Data	Wrap data (xx = variable length).
>27	>xx	Frame Forward	Frame to be forwarded. Includes access control through last information byte.
>29	>08	Adapter Status Vector	Indication of the current state of the sending adapter's software. See Table D–3.
>2A	>04	Transmit Status Code	Code indicating the strip status of a transmitted frame.
>2B	>06	Group Address	Group address set in sending station (zero if not set).
>2C	>06	Functional Address	Functional address field set in the sending station.

Note: † Length in bytes of subvector value.

Table D–2. MAC Frame Subvectors (Continued)

Type	Length†	Subvector Name	Value
>2D	>08	Isolating Errors	Byte 0 – Line error 1 – Reserved 2 – Burst error 3 – ARI/FCI error 4 – Reserved 5 – Reserved
>2E	>08	Nonisolating Errors	Byte 0 – Last frame 1 – Receive congestion 2 – Frame copied error 3 – Reserved 4 – Token error 5 – Reserved
>30	>04	Error Code	Error code defining the error condition. >0001 – Monitor error >0002 – Duplicate monitor >0003 – Duplicate address

Note: † Length in bytes of subvector value.

Table D–3. Adapter Status Vector Decode

Byte	Bit	Definition
0		SIF task status
	0	SIF initialized
	1	SSB busy
	2	SSB queued
	3	Adapter open
	5	Receive command
	6	Transmit command
1	0	SSB wait SCB clear
	1	SSB wait ring status
	2	SSB wait command completion
	3	SSB wait receive
	4	SSB wait transmit
2		SIF transmit status
	0	Subtask not ready
	1	Wait Xmit complete
	2	Wait validation
	4	Xmit halt enabled
	5	Expecting SOF list
	6	List error
3	0	CSTAT valid bit
	2	CSTAT SOF bit
	3	CSTAT EOF bit
4		SIF receive status
	0	Subtask not ready
	1	Waiting for continue
	2	Waiting for validation
5	0	Expecting SOF list
	1	RI-pad escape

Appendix E

Test, Quality, and Reliability

This appendix discusses the quality and reliability of network operation as determined by the individual semiconductor quality and reliability characteristics, as well as the built-in network architecture.

Topics in this appendix include

Section Page
E.1 Network Management and Reliability E-2
E.2 Network Management Services E-3
E.3 Semiconductor Quality and Reliability E-4

E.1 Network Management and Reliability

An effective LAN strategy must comprehend the cabling system that interconnects products, the operation of each product, and the ability to manage the collective group of products on the network. Extensive diagnostic and network management features are implemented in both the first- and second-generation TMS380 chipsets. These services assure reliable and manageable network operation, regardless of the equipment attached to the network. Management functions are distributed to each station throughout the token-ring network via the TMS380 family of adapter chipsets. These functions include

- ❑ Self-test diagnostics
- ❑ Fault isolation and recovery
- ❑ Accounting and security
- ❑ Performance monitoring and tuning
- ❑ Preventive maintenance scheduling

E.1.1 Self-Test Diagnostics

TMS380C16 contains 3K bytes of self-test diagnostic software that automatically executes every time the chipset is powered up. This test includes an internal loop-back test of the chipset and a lobe media test that checks the entire length of cable between the product and the wiring concentrator. TMS380C16 must successfully complete all of these tests before inserting into the ring.

E.1.2 Fault Isolation and Recovery

TMS380C16 isolates faults automatically, allowing the LAN to remove bad nodes from the ring with minimal disruption to service. This fault isolation capability, combined with the star-wired configuration of the token ring, allows errors to be quickly isolated and serviced. In addition, TMS380C16 detects and reports shorts, opens, and degraded signaling characteristics of the cable connecting the attached product to the wiring concentrator. Fault conditions can be further isolated to either the product side or the wiring concentrator.

E.2 Network Management Services

TMS380C16 has a wealth of network management services built into the media access control (MAC) frames. These MAC frames are specialized network messages that all adapters use to exchange network management information. The services provided by MAC frames can be used to build and manage dependable networks that satisfy the ever increasing demands of users for many years to come.

Numerous hardware error checkers and background software are built into the network management of the TMS380C16 to detect faults as they occur. In addition to testing the adapter on which they are resident, the error checkers test the entire ring as well. In this way, the adapters are able to locate faults on the ring and remove any faulty adapters.

E.3 Semiconductor Quality and Reliability

The quality and reliability of Texas Instruments microprocessor and microcontroller products, which includes the two generations of the TMS380, relies on feedback from

- ❑ Customers,
- ❑ Total manufacturing operation from front-end wafer fabrication to final shipping inspection,
- ❑ Product quality and reliability monitoring.

Our customer's perception of quality must be the governing criterion for judging performance. This concept is the basis for Texas Instruments corporate quality policy, which is as follows:

> For every product or service we offer, we shall define the requirements that solve the customer's problems, and we shall conform to those requirements without exception.

Texas Instruments offers a leadership reliability qualification system based on years of experience with leading-edge memory technology as well as years of research into customer requirements. Quality and reliability programs at TI are therefore based on customer input and internal information to achieve constant improvement in quality and reliability.

E.3.1 Reliability Stress Tests

Accelerated stress tests are performed on new semiconductor products and process changes to ensure product reliability excellence. These are typical test environments used to qualify new products or major changes in processing:

- ❑ High-temperature operating life
- ❑ Storage life
- ❑ Temperature cycling
- ❑ Biased humidity
- ❑ Autoclave
- ❑ Electrostatic discharge
- ❑ Package integrity
- ❑ Electromigration
- ❑ Channel-hot electrons (performed on geometries less than 2.0 mm).

Typical events or changes that require internal requalification of product include

- □ New die design, shrink, or layout
- □ Wafer process
- □ Packaging assembly
- □ Component parts
- □ Manufacturing site

TI reliability control systems extend beyond qualification. Total reliability controls and management include product reliability monitoring, as well as final product release controls. MOS memories, utilizing high-density active elements, serve as the leading indicator in wafer-process integrity at TI MOS fabrication sites, enhancing all MOS logic device yields and reliability. TI places more than several thousand MOS devices per month on reliability test to ensure and sustain built-in product excellence.

Table E–1 lists the microprocessor and microcontroller reliability tests, the duration of the test, and sample size. The following definitions describe those tests in the table.

AOQ (average outgoing quality) Amount of defective product in a population, usually expressed in terms of parts per million (PPM).

FIT (failure in time) Estimated field failure rate in number of failures per billion power-on device hours; 1000 FITS equals 0.1 percent failure per 1000 device hours.

Operating lifetest Device dynamically exercised at high ambient temperature (usually 125°C) to simulate field usage that would expose the device to a much lower ambient temperature (such as 55°C). Using a derived high temperature, a 55°C ambient failure rate can be calculated.

High-temperature storage Device exposed to 150°C unbiased condition. Bond integrity is stressed in this environment.

Biased humidity Moisture and bias used to accelerate corrosion-type failures in plastic packages. Conditions include 85°C ambient temperature with 85 percent relative humidity (RH). Typical bias voltage is +5 V and grounded on alternating pins.

Autoclave (pressure cooker)

Plastic-packaged devices exposed to moisture at 121°C using a pressure of one atmosphere above normal pressure. The pressure forces moisture permeation of the package and accelerates corrosion mechanisms (if present) on the device. External package contaminants can also be activated and caused to generate inter-pin current leakage paths.

Temperature cycle

Device exposed to severe temperature extremes in an alternating fashion(65°C for 15 minutes and 150°C for 15 minutes per cycle) for at least 1000 cycles. Package strength, bond quality, and consistency of assembly process are stressed in this environment.

Thermal shock

Test similar to the temperature cycle test, but involving a liquid-to-liquid transfer, per MIL-STD-883C, method 1011.

PIND

Particle-impact noise-detection test. A nondestructive test to detect loose particles inside a device cavity.

Mechanical sequence:

Fine and gross leak	Per MIL–STD–883C, method 1014.5
Mechanical shock	Per MIL–STD–883C, method 2002.3, 1500g, 0.5 ms, condition B
PIND (optional)	Per MIL–STD–883C, method 2020.4
Vibration, variable frequency	Per MIL–STD–883C, method 2007.1, 20 g, condition A
Constant acceleration	Per MIL–STD–883C, method 2001.2, 20 kg, condition D, Y1 plane min
Fine and gross leak	Per MIL–STD–883C, method 1014.5
Electrical test	To data sheet limits

Thermal sequence:

Fine and gross leak	Per MIL–STD–883C, method 1014.5
Solder heat (optional)	Per MIL–STD–750C, method 1014.5
Temperature cycle (10 cycles minimum)	Per MIL–STD–883C, method 1010.5, 65 to +150C, condition C
Thermal shock (10 cycles minimum)	Per MIL–STD–883C, method 1011.4, 55 to +125C, condition B
Moisture resistance	Per MIL–STD–883C, method 1004.4
Fine and gross leak	Per MIL–STD–883C, method 1014.5
Electrical test	To data sheet limits

Thermal/mechanical sequence:

Fine and gross leak	Per MIL–STD–883C, method 1014.5
Temperature cycle (10 cycles minimum)	Per MIL–STD–883C, method 1010.5 65 to +150C, condition C
Constant acceleration	Per MIL–STD–883C, method 2001.2, 30 kg, Y1 plane
Fine and gross leak	Per MIL–STD–883C, method 1014.5
Electrical test	To data sheet limits
Electrostatic discharge	Per MIL–STD–883C, method 3015
Solderability	Per MIL–STD–883C, method 2003.3
Solder heat	Per MIL–STD–750C, method 2031,10 sec
Salt atmosphere	Per MIL–STD–883C, method 1009.4, condition A, 24 hrs min
Lead pull	Per MIL–STD–883C, method 2004.4, condition A
Lead integrity	Per MIL–STD–883C, method 2004.4, condition B1
Electromigration	Accelerated stress testing of conductor patterns to ensure acceptable lifetime of power-on operation.
Resistance to solvents	Per MIL–STD–883C, method 2015.4

Table E–1. Microprocessor and Microcontroller Tests

TEST	DURATION	SAMPLE SIZE PLASTIC	CERAMIC
Operating life, 125°C, 5.0 V	1000 hrs	129	129
Operating life, 150°C, 5.0 V	1000 hrs	77[1]	77
Storage life, 150°C	1000 hrs	77	77
Biased 85°C/85 percent RH, 5.0 V	1000 hrs	129	–
Autoclave, 121°C, 1 ATM	240 hrs	77	–
Temperature cycle, 65 to 150°C	1000 cyc[2]	129	129
Temperature cycle, 0 to 125°C	3000 cyc	129	129
Thermal shock, 65 to 150°C	200 cyc	129	129
Electrostatic discharge, 2 kV		12	12
Latch-up (CMOS devices only)		5	5
Mechanical sequence		–	38
Thermal sequence		–	38
Thermal/mechanical sequence		–	38
PIND		–	45
Internal water vapor		–	3
Solderability		22	22
Solder heat		22	22
Resistance to solvents		15	15
Lead integrity		15	15
Lead pull		22	–
Lead finish adhesion		15	15
Salt atmosphere		15	15
Flammability (UL94–V0)		3	–
Thermal impedance		5	5

Notes: 1) If junction temperature does not exceed plasticity of package.
2) For severe environments; reduced cycles for office environments.

Table E–2 lists the approximate number of transistors for the TMS380C16 device.

Table E–2. TMS380C16 Transistors

DEVICE	# TRANSISTORS
CMOS: TMS380C16	220,000

Note:

Texas Instruments reserves the right to make changes in MOS semiconductor test limits, procedures, or processing without notice. Unless prior arrangements for notification have been made, TI advises all customers to reverify current test and manufacturing conditions prior to relying on published data.

Appendix F

Ring Interface Design Notes

The circuit and board layout of the TMS380 Second-Generation LAN Adapter Ring Interface requires special attention be given to the selection of components and their placement to reduce induced noise. This Appendix describes a set of guidelines that can be used for the layout of the ring interface circuit.

The following sections are contained in this appendix.

Section **Page**

F.1 Circuit Diagrams F-2
F.2 Circuit Diagram Layout Notes F-8
F.3 System Layout F-9

F.1 Circuit Diagrams

Figure F–1 shows the circuit diagram for a single-speed (16 Mbps or 4 Mbps) TMS38053 ring interface, and Figure F–2 shows the circuit diagram for a software switchable (16 Mbps and 4 Mbps) TMS38053 ring interface. Because it is important to provide adequate AC filtering of the V_{CC} supply voltage, TI recommends a generous number of V_{CC}-to-ground decoupling capacitors, ferrrite beads, and power/ground plane isolation. The signals in the ring interface area can be grouped into five types: digital control, digital data, analog, differential data, and differential drive. Digital data and digital control signals have voltage swings that are compatible with standard TTL logic. Analog signals are associated with the critical PLL signal waveforms. Differential data is the signal that is transmitted or received on the transmission media, with a typical transmission signal amplitude of 4 V peak-to-peak and a minimum receiver amplitude of 50 mV peak-to-peak. Differential drive signals provide the differential drive input data for the TMS38053 transmitter output. Table F–1 shows the relationship between these signals and actual signal names on the TMS38053.

Table F–1. TMS38053 Signal Classification

Signal	Type
Digital Control	$\overline{\text{WRAP}}$(36), $\overline{\text{WFLT}}$(26), $\overline{\text{NRST}}$(24), $\overline{\text{REDY}}$(18), ENABLE(5), SPSW(7), FRAQ.(22)
Digital Data	RCLK(3), RCVR(2), XTAL(23)
Analog	EQUALA(44), EQUALB(43), RCVHYS(42), PHOUTA(29), PHOUTB(27), WDTCAP(25), NGRCAP(19), STERES(16), FILTER(12), NGRCAP(19)
Differential Data	RCVINA(38), RCVINB(37), DROUTA(32), DROUTB(31)
Differential Drive	DRVR(35), $\overline{\text{DRVR}}$(34)

Figure F–1 shows two jumpers for selecting between glue circuit values for a 4 Mbps ring and for a 16 Mbps ring. Table F–2 summarizes the functions of these jumpers. In designs that are for a single fixed frequency, these jumpers are not required, and the glue circuit values can be chosen for one option.

Table F–2. Circuit Diagram Jumpers

Jumper	Definition
J1	Ring speed is controlled by the setting of J1, which is tied to a TTL input on the TMS38053. This input is labeled SPSW. If J1 is tied to ground, a 16-Mbps mode is selected, and if J1 is tied to V_{CC} a 4-Mbps mode is selected. This signal controls a divide-by-four circuit on the VCO in the TMS38053. The VCO capacitor used for the first-generation chipset is no longer required.
J2	This jumper sets the PLL filter bandwidth for either 4-Mbps or 16-Mbps operation.

In Figure F–2 , the jumpers are replaced by FET switches to allow software switchability between 16 Mbps and 4 Mbps. V_{DD} in Figure F–2 is connected to +12 volts. Table F–3 shows the part selection and tolerances for the circuits described in this appendix.

Figure F–1. TMS38053 Ring Interface Circuit Diagram (Jumper Switchable)

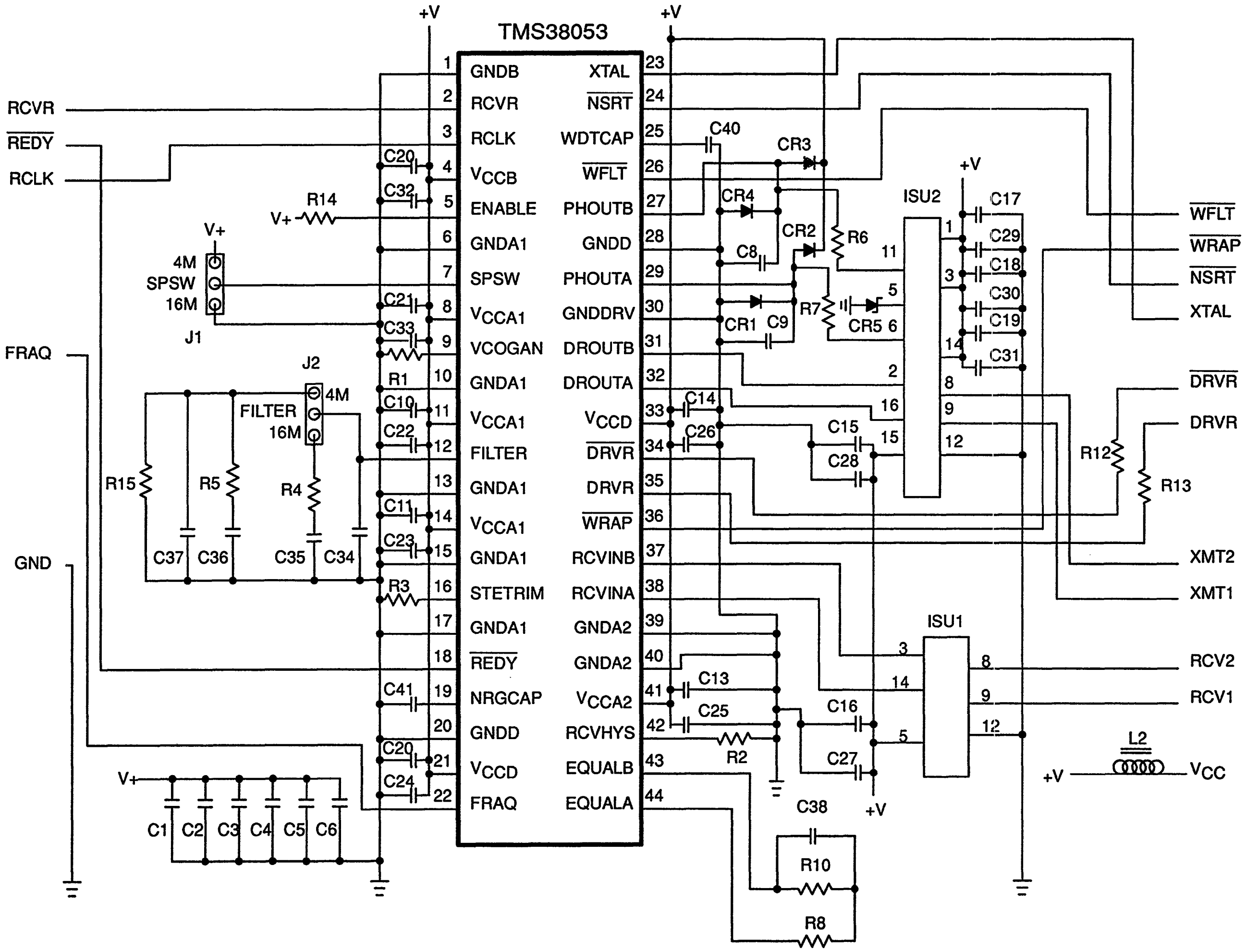


Figure F–2. TMS38053 Ring Interface Circuit Diagram (Software Switchable)

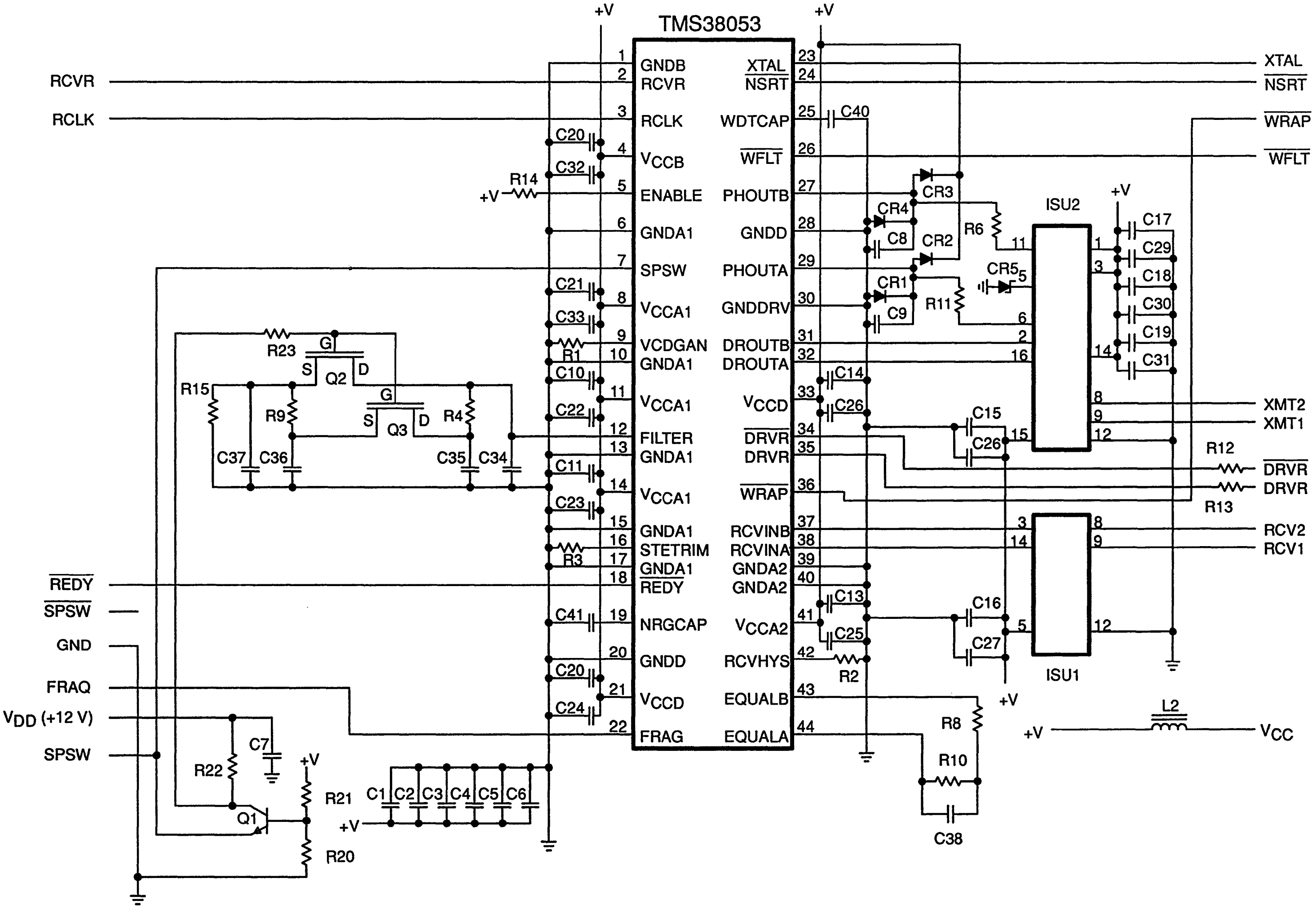


Table F–3. Part Selections and Tolerances

Part	Value/Type	Tolerance
C1—C9	10 µFD, 25V Tantulum	20%
C10—C21	0.22 µFD, 25V X7R	20%
C22—C33, C43	0.0047 µFD, 25V NPO	20%
C34	47 pFD, 25V NPO	10%
C35	0.47 µFD, 25V Tantulum	10%
C36	10 µFD, 25V Tantulum	10%
C37	680 pFD, 25V NPO	10%
C38	390 pFD, 25V NPO	10%
C40	1.5 µFD, 25V Tantulum	10%
C41	6,800 pFD, 25V X7R	10%
CR1—CR4	1N4004, 400V	
CR5	IN5347B	
J1—J3	Jumpers	
L1—L6	TDK ZBF503D-00TA or equivalent ferrite bead	
R1—R2	2.49K Ω, 1/4W	1%
R3	1.1K Ω, 1/4W	1%
R4	1.5K Ω, 1/4W	
R5	604 Ω, 1/4W	1%
R6, R7	49.9 Ω, 2W	5%
R8	82.5 Ω, 1/4W	1%
R9	1.0K Ω, 1/4W	1%
R10	332 Ω, 1/4W	1%
R12, R13	274 Ω, 1/4W	5%
R14	10K Ω, 1/4W	5%
R15	510K Ω, 1/4W	5%
R17, R18	680 Ω, 1/4W	5%
R19	10K Ω, 1/4W	
Common Mode Filter	SPRAGUE 23Z87SM or PE 64680	
R20	5K Ω, 1/4W	5%
R21	10K Ω, 1/4W	5%

Note: For C1—C9, 8.2 µFD is acceptable.

Table F–3. Part Selections and Tolerances (Continued)

Part	Value/Type	Tolerance
R22	100K Ω, 1/4W	5%
R23	1M Ω, 1/4W	5%
Q1	2N2222A or similar NPN	
Q2, Q3	B5170	
ISU1	PE 65611	
ISU2	PE 65621	

F.2 Circuit Diagram Layout Notes

F.2.1 Digital Data and Control

The path lengths for RCVR and RCLK should be kept similar because the timing between the data and its recovered clock should be maintained. The total trace length between the TMS38053 and 74BCT244 for RCLK should be minimized. The XTAL signal is gated by FRAQ, as shown in Figure F–3. This gating must be done outside the critical ring interface circuit area. The digital signals should be routed away from the analog and differential data signals.

F.2.2 Differential Data Signals

The transmit pair, DROUTA/DROUTB or XMT1/XMT2, should be run as close as possible, with separation not exceeding 0.3 inches. The receive pair, RCVINA/RCVINB or RCV1/RCV2, should be run as close as possible, with separation not exceeding 0.3 inches. The transmit and receive pair should NOT be routed close to the other signals–particularly, the analog signals. Where possible, the transmit pair and its components should be separated from the receive pair and its components by at least one inch. The ring isolation circuits, ISU1 and ISU2, should be separated by one inch. Routing of the phantom drive pair (PHOUTA, PHOUTB) is not critical.

F.2.3 Analog Signals

The analog signal placement is most critical to low noise operation. Here are the placement guidelines:

- ❑ Place the PLL filter components (R15, R16, C37, C36, R5, R4, C35, C34) as close as possible to the filter signal (pin 12). No other signals should be routed in this area.
- ❑ Place the VCO gain resistor (R1), RCVHYS resistor (R2), and STETRIM resistor (R3) as close as possible to their respective pins. No other signals should be routed in this area.
- ❑ Place the equalizer components (C38, R10, R8) as close as possible to the EQUALA and EQUALB signals (pins 43, 44). No other signals should be routed in this area.

F.2.4 Differential Drive Signals

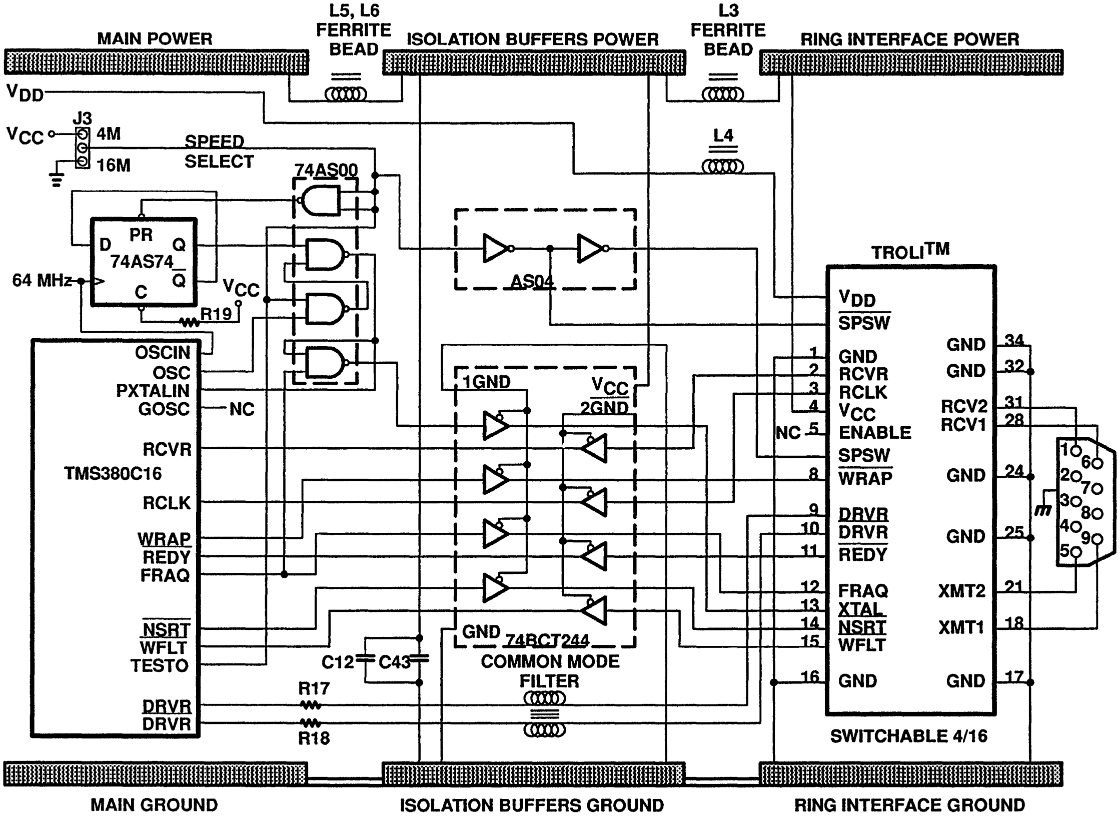
Run the differential drive pair, DRVR/$\overline{\text{DRVR}}$, as close as possible, with separation not exceeding 0.3 inches. These signals should be routed apart from the analog and differential data signals.

F.3 System Layout

F.3.1 Isolation Buffer

Figure F–3 shows the Isolation Buffer interconnecting the TROLI ring interface and the TMS380C16. The TROLI is a ring interface module manufactured by Pulse Engineering and contains the TMS38053 and other circuits. More details on this TROLI module are given in Pulse Engineering Bulletin, 857-5. The Isolation Buffers consist of a 74BCT244 Driver, an S04 Driver, and decoupling capacitors.

Figure F–3. Isolation Buffer Interconnection



The Jumper J3 (or, optionally, a host system register bit that is software writeable) selects the ring speed for the TMS380C16, and also provides the XTAL signal at 32 MHz or 8 MHz to the ring interface. In Figure F–3, this signal is applied to pin TEST0 of the TMS380C16 for speed select; the low level ring speed is 16 Mbps and the high level is 4 Mbps. The READ ADAPTER COMMAND (address 01.0A0C) makes the value of the TEST0 pin available to the host software.

A common mode filter connects the DRVR/$\overline{\text{DRVR}}$ signals and the FEDRVR/$\overline{\text{FEDRVR}}$ signals. The filter is placed over the gap in the ground and V_{CC} planes as shown.

Ferrite filter beads connect the V_{CC} power planes at a single point as shown in Figure F–5, and also filter any +12-V power required in the ring interface module.

The DB9 ring connector is located in an area without either V_{CC} or ground planes. The DB9 case should be connected to chassis ground.

The Isolation Buffers are located on a “necked down” peninsula ground area sitting between the ring interface and remaining board areas, as shown in Figure F–3. Power is supplied through a ferrite bead, and two decoupling capacitors are utilized.

F.3.2 Power and Ground Layout

Figure F–4 shows a recommended grounding layout diagram, and Figure F–5 shows a recommended power layout diagram. No unbuffered or unfiltered signals should be routed from the logic power area to the ring interface area of the board. The +12V V_{DD} is connected through a ferrite bead and is decoupled on both sides of the ferrite bead with suitable capacitors.

Figure F–4. Recommended Grounding Layout Diagram

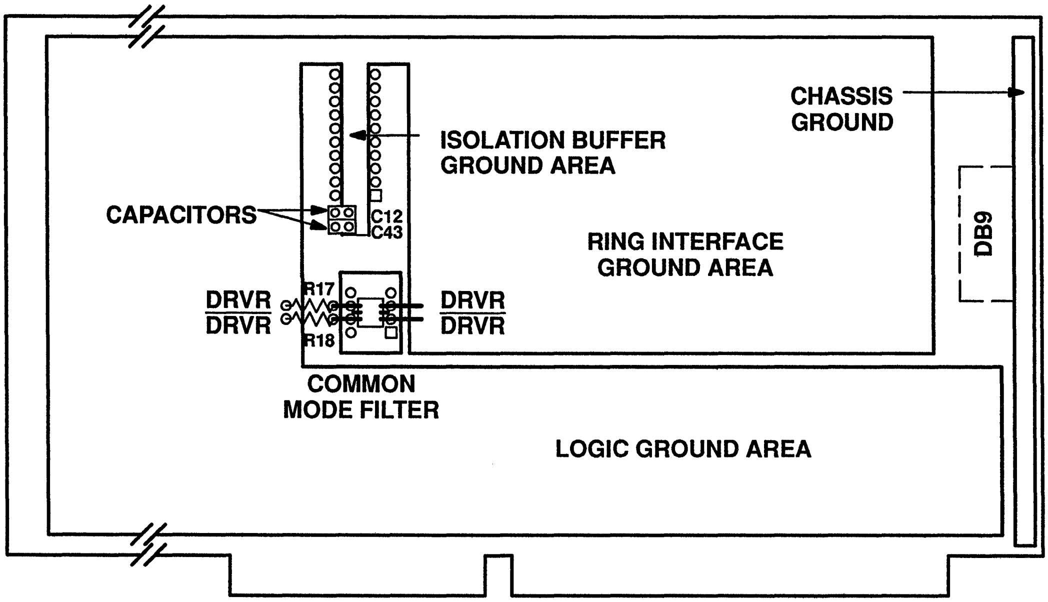
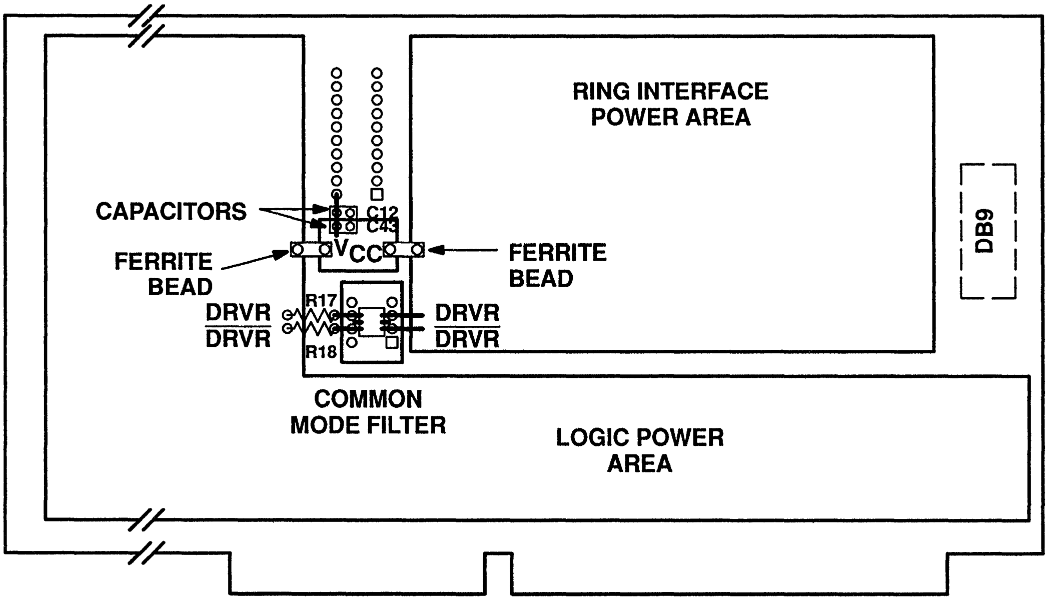


Figure F–5. Recommended Power Layout Diagram



The board design should use at least four levels, with ground and power being the two internal levels. The ground level should be closest to the component side with a single "necked down" quarter-inch wide connection into the ring interface area at a single point. The power levels should be separate between the ring interface area, the isolation buffer, and the main board, with a ferrite bead filtered connection between these power areas. The ground and power layouts in Figure F–4 and Figure F–5 are not to scale.

Appendix G

Host-Mapped EPROM for Download of TMS380 Adapter Software

This appendix describes the implementation of a host-mapped EPROM for a boot-time download of the adapter software to a TMS380C16- based adapter card. These procedures are applicable to any environment; however, for simplicity, the details are given for the IBM-PC/ AT environment.

The following sections are contained in this appendix:

Section **Page**

G.1 Host System Features G-2
G.2 Theory of Operation G-3
G.3 Memory Map G-4
G.4 Hardware and PAL Equations G-6
G.5 EPROM Software Program G-9

G.1 Host System Features

During the Power-On Self Test (POST) of a PC/AT, the existence of any boot-ROMs are determined. If boot-ROMs are found, they are called to initialize the respective hardware they represent (such as disk controllers, video controllers, and network controllers). See the *IBM/AT Technical Reference Guide* for more information.

G.2 Theory of Operation

An EPROM containing a downloader for the TMS380C16 and the adapter software in binary image format is placed on the host address/data buses. The EPROM is mapped into the memory of the PC by using a PAL to generate the chip select and output enable for the EPROM. To conserve space in the memory map, the EPROM is shadow-mapped so that only half of the EPROM is active at any time. This enables the EPROM to fit into the memory map of a broader range of PC-AT class machines.

A minimum chip solution is achieved by using the octal bus transceiver, which drives the TMS380C16's data onto the PC data bus, and also drives the EPROM data onto the PC data bus.

The EPROM code resets the adapter, halts the adapter CPU, and downloads the adapter software. The code also changes to the second half of the EPROM (if necessary, to complete the download) and starts execution of the adapter software.

G.3 Memory Map

The ROM BIOS provides address space from absolute C0000h to EFFFFh for adapter boot-ROM modules. Therefore, it is necessary for an EPROM to appear in the PC memory map at one of these locations. For flexibility of the design, a jumper is used to select the address of the EPROM in the PC memory map.

The MAC software requires approximately 20 Kbytes of space in addition to the space required by the host downloader, initialization routines, and other device-dependent routines. The LLC software requires approximately 40 Kbytes of space in addition to the downloader. Thus, a 64-Kbyte EPROM is needed to load the MAC/LLC software.

The EPROM is mapped so that the first 32 Kbytes of the EPROM appear in the memory of the PC. When downloading from the first half is complete, a write to the EPROM changes the memory map so that the second half of the EPROM appears in the memory map. The download continues from the second half. The first half of the EPROM can be accessed only after an adapter hardware reset, which causes the mapping to change to its original condition. The EPROM has been mapped this way to allow the EPROM to fit into the memory map of a broader range of machines. The same functionality could be achieved and the design simplified if the EPROM were mapped into a 64-Kbyte space.

For this design example, we have chosen to shadow-map a 64-Kbyte EPROM into a 32-Kbyte block of host memory. The base address of the EPROM can be selected by using an EPROM address-select jumper, as shown in Table G–1. The address decode logic for the EPROM is contained in the PAL U22, defined by the equations in Table G–1. The address-select-jumper outputs connect to the PAL as shown in Figure G–2.

Table G–1. EPROM Jumper Settings

Jumper Settings ES0	ES1	ES2	EPROM Address in PC Memory Map
–	–	–	C0000 H
X	–	–	C8000 H
–	X	–	D0000 H
X	X	–	D8000 H
–	–	X	E0000 H
X	–	X	E8000 H
–	X	X	F0000 H
X	X	X	Disabled

Note: X indicates a jumper.
– indicates no jumper.

G.4 Hardware and PAL Equations

The EPROM Chip Select (ECS) signal produced by U22 is used to chip-select the EPROM. The EPROM Output Enable ($\overline{\text{EOE}}$) enables the outputs of the EPROM. The Data Low Byte Enable ($\overline{\text{DLBEN}}$) is used to enable the 74ALS245 octal bus transceiver when the EPROM is being accessed, driving the EPROM data onto the PC data bus. The most significant address line to the EPROM is generated by the PAL U22. This is done to select between the first and second 32-Kbyte maps in the EPROM.

LSA15 is activated by writing to the EPROM. It can be reset only when $\overline{\text{MRESET}}$ is set to its active level by a hardware reset of the adapter (in other words, setting ARESET (bit 8) of SIFACL to 1, or setting the $\overline{\text{SRESET}}$ pin to its active level).

Figure G–1. PAL Equations for U022:

```
MODULE EPROM_SELECT
TITLE 'PŒPROM DECODE LOGIC FOR 20L8: TI,HOUSTON TX MAR 10,1989'

U022 DEVICE 'P20L8';

NMRESET,AEN,NSDBEN,SA19,SA18,SA17,SA16,SA15   PIN1,2,3,4,5,6,7,8;
NSMEMR,NSMEMW,ES2,ES1,ES0,NC1,NC2,NC3  PIN9,10,11,13,14,15,16,17;
NC4,LSA15,NECS,NEOE,NDLBEN,NC5         PIN18,19,20,21,22,23;

L,H,X,Z  =   0,1,.x.,.z.;
INPUTS   =   [SA19,SA18,SA17,SA16,SA15,NSMEMR,NSMEMW,NMRESET,
                AEN,NSDBEN,ES2,ES1,ES0,NEOE,NECS,LSA15];
OUTPUTS  =   [NDLBEN,NECS,NEOE,LSA15];

EQUATIONS

ENABLE NEOE        =    1;
!NEOE              =    !NECS & !NSMEMR;

ENABLE NECS        =    1;
!NECS              =       SA19    &  SA18     & !SA17     & !SA16
                         & !SA15    &  ES2      &  ES1      &  ES0
                         & !AEN
                        #   SA19    &  SA18     & !SA17     & !SA16
                         &  SA15    &  ES2      &  ES1      & !ES0
                         & !AEN
                        #   SA19    &  SA18     & !SA17     &  SA16
                         & !SA15    &  ES2      & !ES1      &  ES0
                         & !AEN
                        #   SA19    &  SA18     & !SA17     &  SA16
                         &  SA15    &  ES2      & !ES1      & !ES0
                         & !AEN
                        #   SA19    &  SA18     &  SA17     & !SA16
                         & !SA15    & !ES2      &  ES1      &  ES0
                         & !AEN
                        #   SA19    &  SA18     &  SA17     & !SA16
                         &  SA15    & !ES2      &  ES1      & !ES0
                         & !AEN
                        #   SA19    &  SA18     &  SA17     &  SA16
                         & !SA15    & !ES2      & !ES1      &  ES0
                         & !AEN;

ENABLE NDLBEN      = 1;
!NDLBEN            = (!NECS # !NSDBEN)
                        & (!NSMEMR # !NSDBEN);

ENABLE LSA15       = 1;
!LSA15             = ((NECS # NSMEMW) & !LSA15)
                        # !NMRESET;
END EPROM_SELECT
```

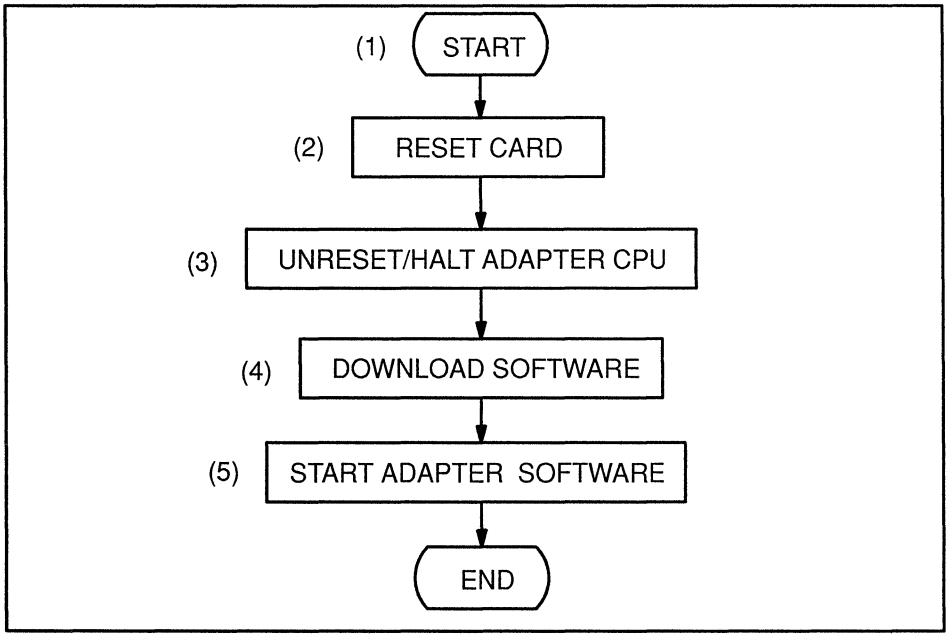
Figure G–2. Schematic of Host-Mapped EPROM

Illustration Is Not Available

G.5 EPROM Software Program

The program stored in the EPROM is written in the host processor's assembly language (Intel 808X assembly code for the example in Figure G–4) and follows the flow chart in Figure G–3. The program should first reset the adapter and halt the adapter CPU. Next, the adapter software is downloaded by using the DIO register interface. The binary image of the adapter software is contained in the code segment of the program in word data format so that it can be accessed by the code in the EPROM.

Figure G–3. Algorithm to Download From EPROM to a TMS380C16-Based Board



The adapter software is obtained in the word data format for Intel 808X processors by using a utility program included on the software package utility disk (the COFF loader program). This utility program may produce the binary image in multiple blocks or in a single block. In this example, the following format was chosen for the output of the utility program.

```
dw   Address Extended Blk 1, Address Blk 1, Length, Word 1 of Blk 1,
dw   Word 2 of Blk 1, etc..., Last word of Blk 1,
dw   Address Extended Blk 2, Address Blk 2, Length, Word 1 of Blk 2,
dw   Word 2 of Blk 2, etc..., Last word of Blk 2,
dw   Address Extended Blk N, Address Blk N, Length, Word 1 of Blk N,
dw   Word 2 of Blk N, etc..., Last word of Blk N,
dw    0001h,0000h,0000h
```

N represents the number of the last block of the binary image for the adapter software.

In the last block, the extended address of 0001h is used to provide compatibility with first-generation host software, and the length of zero indicates that there are no more blocks to download. Thus, when the download routine shown in the code example (Figure G–4) reads a block length of zero, it starts execution of the adapter software. The file containing the adapter software in this data format is imported into the assembly source code for the EPROM.

The host initialize and download code must be at the beginning of both host EPROM maps, and the data must begin in map one and continue in the second map (if required). When the code in the first map has been downloaded, a memory write to the EPROM address must be executed by the program. This causes the external logic in the PAL U22 to change to the second map of the EPROM. Then, the code in the second map is downloaded to the adapter.

In the example of code shown in Figure G–4, the binary image of the adapter software is downloaded to the DRAM of a TMS380C16-based adapter. The DIO interface is used to transfer the data from the EPROM to the adapter memory. This routine supports download of the binary image in any number of blocks. The binary image data begins at the label MAC.

Figure G–4. Assembly Download Code for Adapter Software

```
          MOV     DX,CS:SIFACL   ;Gethe  port addressof the SIFACL register
          IN      AX,DX          ;Get the SIFACL contents
          OR      AX,0080H       ;Set the ARESET bit to reset adapter
          OUT     DX,AX          ;Output to the register to reset
          AND     AX,0108H       ;Save interrupt enable and pseudo DMA bits
          OR      AX,0060H       ;Set CPU halt/DRAM enable bits
          MOV     BX,AX          ;Save for later use
          IN      AX,DX          ;Get hardware strapping modes
          AND     AX,0016H       ;Isolate reserved  and  parity enable bits
          OR      AX,BX          ;Add the rest of the new modes
          CALL    WAIT_10    ;Allow the reset to take effect(10µs)
          OUT     DX,AX          ;Clear ARESET and set new modes

          CALL    WAIT_10    ;Allow the reset to clear(10µs)

          MOV     SI,OFFSET CODE ;Point to beginning of microcode
          PUSH    CS             ;Set up data segment to
          POP     DS             ;point to data

;   Download all records to adapter

LP:     LODSW                    ;Get SIFADX value word
          MOV     DX,CS:SIFADX   ;Set SIFADX register address
          OUT     DX,AX          ;Set SIFADX to value
          LODSW                  ;Get SIFADR value word
          MOV     DX,CS:SIFADR   ;Set SIFADR register address
          OUT     DX,AX          ;Set SIFADR to value
          LODSW                  ;Get data length value
          MOV     DX,CS:SIFINC   ;Set SIFINC register address
          MOV     CX,AX          ;Set data move count
          JCXZ    DONE           ;Exit loader loop if done

DWML:   LODSW                    ;Get   a   word
          OUT     DX,AX          ;Write it to SIFINC register
          LOOP    DWML           ;Loop till count is zero
          JMP     LP             ;Go process next record

;  Loading is complete now, execute code

DONE:      MOV DX,CS:SIFACL    ;Get the port address of the SIFACL register
          IN      AX,DX         ;Get the SIFACL contents
          AND     AX,0F3FH      ;Clear the CPU halt bit
          OUT DX,AX             ;Start execution of Adapter Software.
```

Appendix H

Adapter-Based EPROM for TMS380C16 Adapter Software

This appendix describes the implementation of EPROMs on the TMS380C16 adapter local bus side for the case where DRAMs on the local bus, which require downloading the adapter code from the host, is not possible.

The following sections are included in this appendix.

Section **Page**

H.1 Theory of Operation H-2
H.2 EPROM Access Conditions H-4
H.3 Adapter Timing Considerations H-5
H.4 Hardware Schematic Description H-6
H.5 Timing Requirements H-7
H.6 Bus Loading Considerations H-8

H.1 Theory of Operation

EPROMs can be implemented on the adapter local bus side when adapter code downloading from a host system is not desired or not possible. The schematic for such an EPROM-based adapter is shown in Figure H–1.

After adapter reset, the TMS380C16 begins executing code from EPROM. If execution of bring-up diagnostics, initialization, and the adapter OPEN command is successful, operation may continue using ROM-based code, or overriden by host-software control to download new microcode into DRAM. This adds the benefit of flexibility while keeping the added chip count as low as possible.

Note:

The EPROM size illustrated in this application is 128 Kbytes, implemented in two EPROMS. Due to the fact that the microcode is always check-summed and an auto parity is performed internally, an EPROM for parity is not required.

Figure H–1. TMS380C16 EPROM-Based Interface Schematic

Illustration Is Not Available

H.2 EPROM Access Conditions

The memory bus timings for an EPROM are given in Figure 3–10 and described in section 3.2.2.3. An EPROM access occurs when the following two conditions are true:

- ❑ The boot bit in the SIFACL register is zero. This can be achieved by hardwiring the BTSTRP pin to ground. The logic level of the BTSTRP pin is loaded into the SIFACL register at reset to form a default value. It can also be set/reset by software when the ARESET bit is 1.
- ❑ A read from an address in the range >00.0010 to >00.FFFE (chapter 0) or >1F.0000 to >1F.FFFE (chapter 31) is performed. Note that MAC/LLC microcode requires about 40 Kbytes; thus TMS27C256-type EPROMs (maximum TMS27C512 for future versions) can be used, giving 64 Kbytes. 128 Kbytes is the maximum allowable.

H.3 Adapter Timing Considerations

The MREF signal (DRAM refresh active) is taken low at the start of M8 of the previous cycle to indicate to any external logic attached to the DRAM C inputs that the cycle is not a refresh cycle.

The $\overline{\mathrm{MAL}}$ signal serves as a strobe signal that is typically supplied to a transparent latch. The rising edge of $\overline{\mathrm{MAL}}$ can be used as a starting reference point for EPROM access time calculations (see parameter 33 in the TMS380C16 data sheet). At the falling edge of $\overline{\mathrm{MAL}}$, the latch then retains the full valid 20-bit address on the MADH0—MADH7, MADL0—MADL7, MAXPH, MAXPL, MAX0, and MAX2 buses throughout the cycle.

The $\overline{\mathrm{MROMEN}}$ signal is taken low on M1 during the first 5/16 of the cycle only. It therefore must be latched externally by the falling edge of $\overline{\mathrm{MAL}}$ to maintain an EPROM output or chip-enable signal throughout the cycle. Using it to generate a chip enable helps reduce power consumption when the EPROM is not addressed. The generation of $\overline{\mathrm{MROMEN}}$ during this phase is exclusive to accesses in the defined address range. It is, therefore, not necessary to generate or facilitate any signals other than $\overline{\mathrm{MAL}}$ and $\overline{\mathrm{MROMEN}}$ to interface to EPROMs.

After the MADH, MADL, MAXPH, and MAXPL signals are tri-stated, $\overline{\mathrm{MBEN}}$ is taken low to drive the EPROM data onto these lines by enabling the transceiver outputs. The DIR input is driven by LEN. When low, the 74ALS245 will drive data from its B inputs onto the MAD bus. $\overline{\mathrm{MBEN}}$ is taken high at the start of M7 to three-state the buffer outputs before the next address is output in the following M1. Buffer decode logic is not needed thus reducing chip count to a minimum.

Note:

Figure 3–10 shows that it is possible to use MRLS to generate an EPROM $\overline{\mathrm{OE}}$ signal because its timing coincides with EPROM reads. This is a feasible option, however, when using both DRAMs and EPROMs, the MRLS signal is also connected to the $\overline{\mathrm{MRAS}}$ inputs of five DRAMs. Since EPROMS have significant capacitive loading at the $\overline{\mathrm{OE}}$ inputs (25 pF each for 27C256), $\overline{\mathrm{MRAS}}$ would be used near its AC load limit (85 pF). Buffering or gating can reduce this value to a safe margin, but the delays must be considered in EPROM access time calculation.

H.4 Hardware Schematic Description

Figure H–1 shows how the use of three 74AS373-type latches and two 74ALS245-type transceivers are used as data buffers. Since both 32-Kbyte and 64-Kbyte EPROMs may be used, and they are pin-compatible except for pin 1 (address MSB or V_{PP}), the MADH0 (address MSB) line is routed through a separate latch U6. A jumper can then select whether the MADH0 is gated through to the EPROM's address MSB for TMS27C512 types (64-Kbyte), or three-stated by the latch and pulled high for use with TMS27C256 types (32-Kbytes). These types require pin 1 (V_{PP}) to be set to Vcc for normal operation. Note that the pull-up on the TMS380C16's BTSTRP input serves the purpose of reducing crosstalk when EPROMs are not used. To ensure proper DC loading on the BTSTRP and the latch's output enable, separate resistors of 4.7 kΩ are recommended. Setting the jumper to connect pins 2 and 3 will enable the microcode EPROMs upon reset.

To reduce power consumption, the EPROM's $\overline{\text{CE}}$ and $\overline{\text{OE}}$ signals are both driven by the latched $\overline{\text{MROMEN}}$ (LEN in Figure H–1). The EPROMs, data buffers, and transceivers are then shut off for any DRAM or BIA accesses.

When an EPROM read cycle is followed by a DRAM refresh cycle, LEN is kept active into the next $\overline{\text{MAL}}$ high phase because of the latch's own propagation delay in following the address/$\overline{\text{MROMEN}}$ signals. Since $\overline{\text{MBEN}}$ has returned to an inactive level long before the next rising edge of $\overline{\text{MAL}}$, the EPROMs can be enabled during t_{PLH} (74AS373). However, the data buffers have been three-stated; therefore, MADH/L is left undisturbed. The latches will drive the contents of MADH/L onto the EPROM address bus during t_{PLH} (74AS373), but this does not have any effect because $\overline{\text{MROMEN}}$ remains high during the next cycle.

$\overline{\text{MCAS}}$ and $\overline{\text{MBIAEN}}$ are retained at the inactive high level throughout the cycle to prevent accesses to DRAM and BIA PROM respectively. $\overline{\text{MOE}}$ is also retained inactive high to keep DRAM outputs three-stated. $\overline{\text{MRAS}}$ is still issued, however, and performs an additional DRAM refresh of the row address presented on A7—A14.

H.5 Timing Requirements

$\overline{\text{MBEN}}$ is taken low before data valid time; thus, an access time margin of M5 to M6 (see parameter 49 in the Appendix A TMS380C16 data sheet) is reserved for the transceiver gate logic to settle. The compounded delay of external logic (such as PALs and latches) must be less than parameter 49. If this condition is met, the required EPROM access time for local bus speed of 4 MHz can then be calculated as follows:

$$t_{\text{acc(min)}} = t_{33} - t_{\text{PLH(AS373)}} - t_{\text{PZH(ALS245)}}$$

where

t_{33} = time of address valid to data valid

$t_{\text{PLH(AS373)}}$ = AS373 type maximum latch propagation delay

$t_{\text{PZH(ALS245)}}$ = ALS245 type maximum propagation delay

When using data buffers, take care to minimize current drive onto data bus during reads. This keeps noise to a minimum. It is recommended to use ALS-type logic for this purpose. The maximum EPROM access time then results as follows:

144 ns – using all 'AS technology
133 ns – using 'AS latches and ALS buffers
121.5 ns – for an all 'ALS-type design .

H.6 Bus Loading Considerations

As mentioned above, the choice of TTL family has various effects on timing and transmission quality. The following list gives an overview of capacitive and current drive calculations done for the MADH/L bus, which shares the most significant load. Note that these figures refer to worst case values and are approximated in some cases. The choice of parts is based on the schematic in Figure H–1.

❑	Its own output lead capacitance.	**15 pF**
❑	Five (including parity) DRAM address inputs.	**25 pF (TMS4464)**
❑	One DRAM data output.	**5 pF (TMS4464)**
❑	One latch address input.	**~5 pF (74ALS373) or** **6 pF (74AS373)**
❑	One transceiver data output.	**15 pF (74AS245) or** **~13 pF (74ALS245)**

In total, the range of capacitive load then amounts to 63 pF—66 pF.

Even though the drive capability of the TMS380C16 is specified at 100 pF, it is recommended to reserve a safety margin of 30–40% with respect to DC load limitations.

Note:

Using the source routing accelerator (TMS380SRA) chip adds another load to this bus. The SRA's output drive capability is only 50 pF. When the TMS380SRA is used, it therefore becomes necessary to buffer the DRAM inputs to reduce the overall bus load. This, in turn, requires faster DRAM access times (80 ns).

The DC load characteristics for the TMS380C16 are given as follows:

$I_{OL} = 2.0$ mA, $I_{OH} = -400$ uA

The output drive value for each line of the MADH/L bus is then derived from the sum of input currents of the DRAMs, latches, and the EPROM data buffers/transceivers (three-stated during write cycles).

The value range is between 0.17 mA (all ALS) and 1.3 mA (all ALS), well below maximum fanout limitations (2 mA). The choice of the data buffer/transceiver device family is based on a compromise between EPROM access timing and the increased noise on the data bus caused by a relatively high output driven by AS devices.

Appendix I

ASCII Character Sets

Table I–1. ASCII Character Set

ASCII Character Set (7-Bit Code)								
M								
S	0	1	2	3	4	5	6	7
L S N	(0)	(16)	(32)	(48)	(64)	(80)	(96)	(112)
0	NUL	DLE	SP	0	@	P	'	p
1	SOH	DC1	!	1	A	Q	a	q
2	STX	DC2	”	2	B	R	b	r
3	ETX	DC3	#	3	C	S	c	s
4	EOT	DC4	$	4	D	T	d	t
5	ENQ	NAK	%	5	E	U	e	u
6	ACK	SYN	&	6	F	V	f	v
7	BEL	ETB	’	7	G	W	g	w
8	BS	CAN	(	8	H	X	h	x
9	HT	EM	)	9	I	Y	i	y
A	LF	SUB	*	:	J	Z	j	z
B	VT	ESC	+	;	K	[	k	{
C	FF	FS	,	<	L	\	l	|
D	CR	GS	–	=	M	]	m	}
E	SO	RS	.	>	N	∧	n	~
F	SI	US	/	?	O	_	o	DEL

Note: To obtain decimal value add decimal MSN to decimal LSN.

Table I–2. Control Characters

Hex Value	Decimal Value	Character
00	0	NUL
01	1	SOH
02	2	STX
03	3	ETX
04	4	EOT
05	5	ENQ
06	6	ACK
07	7	BEL
08	8	BS
09	9	HT
0A	10	LF
0B	11	VT
0C	12	FF
0D	13	CR
0E	14	SO
0F	15	SI
10	16	DLE
11	17	DC1
12	18	DC2
13	19	DC3
14	20	DC4
15	21	NAK
16	22	SYN
17	23	ETB
18	24	CAN
19	25	EM
1A	26	SUB
1B	27	ESC
1C	28	FS
1D	29	GS
1E	30	RS
1F	31	US
7F	32	DEL

Appendix J

Glossary

A

access control (AC): The access control field of a frame or token is the first octet of the physical control field. The AC contains the priority indicator (PI),token indicator (TI), monitor count (MC) and priority reservation (PR) fields ofthe frame or token.

access priority: The access priority of a frame enqueued for transmission is the maximum priority level of token that the adapter will capture for transmission.

active monitor: The active monitor is the adapter responsible for providing clocking to the ring and other functions such as token error detection and recovery.

adapter: The term adapter refers to the Texas Instruments chipset consisting of the TMS38010 Communications Processor, TMS38020 Protocol Handler, TMS38030 System Interface, and TMS38051 and TMS38052 Ring Interface functions now contained in the TMS380C16 and the TMS38053.

adapter software: Adapter software is software executed by the TMS380C16 COMMprocessor. This software provides IEEE 802.5 compatible medium access control (MAC) services, network management support, and adapter diagnostics.

addr, ADDR: The abbreviation addr stands for address. ADDR is the name of a two-bit field in RCVSTS that indicates the address match condition.

address: The logical location of a terminal, a peripheral device, node or any other unit or component in a network.

ADEL: Abort delimiter. A two-byte sequence on the ring consisting of SDEL followed by EDEL.

AIRQ: Adapter interrupt request. Bit 0 of the SIFCMD register, which when set by the system processor, asserts a level 7 interrupt to the adapter.

application layer: A logical entity of the OSI model; the top of the seven-layer structure, generally regarded as offering an interface to, and largely defined by, the network user.

ARA(0–5): Auxiliary ring address registers. A protocol handler register which contains the address of a ring on which the TMS380C16 is not resident, but for which it may copy frames when used in a tunnel or bridge.

ARI: Address recognized indicator. A bit in the end delimiter of a frame which indicates that at least one of the stations that has received the frame has recognized the destination address as its own.

attached system/attaching product: The attached system is the product which uses the adapter to connect to the ring network; also referred to as the attaching product or the host system.

attention MAC frames: Attention MAC frames are MAC frames received by the adapter in which the attention field of the frame control field (FC) (bits 4—7) is greater than one. This condition causes an attention interrupt within the adapter.

attenuation: The difference (loss) between transmitted and received power, due to the transmission loss through equipment lines or other communications devices.

allowed access priority: The allowed access priority is the highest access priority an attaching product may use when requesting a transmission.

B

bandwidth: The range of frequencies that can pass over a given circuit. Generally, the greater the bandwidth, the more information that can be sent through the circuit in a given amount of time.

baseband: Transmission of signals without modulation. In a baseband local area network, digital signals are inserted directly onto the cable as voltage pulses. The entire bandwidth of the cable is consumed by the signal.

baud: A measure of transmission speed; the reciprocal of the time duration of the shortest signal element in a transmission.

bridge: Equipment which allows the interconnection of LANs, allowing communication between devices on separate networks using similar protocols.

broadcast: Delivery of a transmission to two or more stations at the same time.

buffer: In this specification, the term is used almost exclusively to mean a data structure used to hold data to be transmitted to the ring or to receive data received from it.

burst-4 error: A physical signal consisting of 4 baud without any high-to-low or low-to-high transitions.

burst-5 error: A physical signal consisting of 5 baud without any high-to-low or low-to-high transitions.

C

communications processor: The TMS38010 Communications Processor (CP) is a VLSI component of the first generation TMS380 adapter chipset which contains a dedicated 16-bit CPU. The CP executes the adapter software which controls the functions and processes of the adapter.

contention: A dispute between two or more devices over the use of a common channel at the same time.

CSMA/CD: CARRIER SENSE MULTIPLE ACCESS WITH COLLISION DETECTION. A contention technique which allows multiple stations to successfully share a broadcast channel by avoiding contention via carrier sense and deference, and managing collisions via collision detection and frame retransmission. Defined by IEEE Standard 802.3

cyclic redundancy check (CRC): A characteristic link-level feature of (typically) bit-oriented data communications protocols, wherein data integrity of a received frame is checked using a polynomial algorithm based on the content of the frame, and then matched with the result performed by the sender and included in a field appended to the frame.

D

data link: Any serial data communications transmission path; generally between two adjacent nodes or devices and without any intermediate switching nodes.

data link layer: The logical entity in the OSI reference model concerned with transmission of data between network nodes; the network processing entity that establishes, maintains, and releases data link connections between elements in a network; the second layer processed in the OSI reference model, between the physical and the network layers.

data rate: A measure of the signaling rate of a data link.

destination: Station designated as the intended receiver of data.

direct input output (DIO): Direct I/O refers to direct reading or writing to the TMS380C16 system interface registers by the attached system's processor. This is also commonly called memory-mapped I/O.

direct memory access (DMA): Direct memory access is a mode where a device other than the host processor contends for and receives mastership of the memory bus so that data transfers may take place independent of the host.

E

enabled function classes: The enabled function classes are the source classes a station is allowed to use in the transmission of MAC frames. These are the transmit medium access control frames which may be passed from the attaching product to the adapter for transmission.

error detection: Code in which each data signal conforms to specific rules of construction so that departures from this construction in the received signals can be automatically detected.

F

frame copied indicator (FCI): A bit in the end delimiter of a frame which indicates that at least one of the stations that has received the frame has copied it.

fiber optics: A technology that uses light as an information carrier. Fiber optic cables are a direct replacement for conventional coaxial cable and wire pairs. The glass-based transmission facility occupies less physical volume for an equivalent transmission capacity, and the fibers are immune to electrical interference.

flow control: The capability of network nodes to manage buffering schemes in order to allow devices of differing data transmission speeds to communicate with each other.

frame: A collection of bits that contain both control information and data; the basic unit of transmission on a network. Control information is carried in the frame with the data to provide for such functions as addressing, sequencing, flow control and error control to the respective protocol levels. Can be of fixed or variable length.

frame check sequence (FCS): The frame check sequence is a 32-bit field which follows the information field of a frame. This field contains a CRC value used to verify error-free transmission of the frame.

frame control (FC): The frame control field of a frame or token is the second octet of the physical control field. The FC contains bits which define the frame as a MAC or non-MAC frame and the MAC frame attention code. This field is used by the TMS380C16 for frame processing.

frame format: The exact order and size of the various control and information fields of a frame, including header, address and data fields.

frame overhead: A measure of the ratio of the total frame bits occupied by control information to the number of bits of data, usually expressed as a percent.

frame status (FS): The frame status field of a frame is an octet appended after the ending delimiter of a frame which is used to indicate whether the destination address was recognized and whether the frame had been copied by the destination adapter. This field is neither code-violation nor CRC-protected.

functional address: A form of group address which provides a well known address for network functions such as active monitor, ring error monitor, etc. Up to 31 unique functional addresses can be recognized by the TMS380 LAN adapter.

G

gateway: A special node that interfaces two or more dissimilar networks, providing protocol translation between the networks. A gateway is needed to connect two independent local area networks or to connect a local network to a long-haul network.

group address: An address which may be recognized by more than one node on the ring. A group address may be used for functions such as disk servers.

H

host : A product which uses the TMS380 token-ring LAN adapter to connect to the ring network; also referred to as the attached system or attaching product.

I

IEEE 802: Standards for the interconnection of local area networking computer equipment. The IEEE 802 standard deals with the physical and data link layers of the ISO reference model for open systems interconnection, as well as network management.

ISO: International organization for standardization.

L

layer: In the OSI reference model, referring to a collection of related network-processing functions that comprise one level of a hierarchy of functions.

lobe: The lobe refers to the physical star wiring between the adapter and the wiring concentrator.

local area network (LAN): A type of high-speed data communications arrangement wherein all segments of the transmission medium (typically, coaxial cable, twisted-pair wire, or optical fiber) are under the control of the network operator and operate in a localized geographic area (an office building, complex of buildings, or campus).

logical link control (LLC): A protocol developed by the IEEE 802.2 committee, common to all of its physical and medium access control standards, for data link-level transmission control; the upper sublayer of the IEEE layer 2 (data link layer) protocol that complements the MAC protocol.

M

medium access control (MAC): Medium access control refers to a set of ser vices provided by the adapter which are concerned with proper operation of the ring and the detection of and recovery from error conditions: a medium-specific access control protocol within IEEE 802 specifications and includes variations for the token ring, token bus, and CSMA/CD; the lower sublayer of the IEEE layer 2 (data link layer) which complements the logical link control.

MAC frames: A class of frames on the ring which carry out the processes of the medium access control protocol of the token ring architecture. MAC frames are designated by bits 0 and 1 of the frame control field being set to 0 and 0 respectively. In many cases, the attached system will not be notified of the receipt or transmission of such frames.

major vector: The major vector is the information field of a MAC frame. This field consists of a major vector ID and (optionally) one or more subvectors.

major vector ID: The major vector ID (MV-ID) is defined as the class byte and command byte fields of a MAC frame major vector. It carries information on MAC frame type.

medium: Any material substance that can be, or is, used for the propagation of signals, usually in the form of modulated radio, light, or acoustic waves, from one point to another, such as optical fiber, cable, water, air, or free space.

N

network: An interconnected group of nodes; a series of points, nodes, or stations connected by communications channels; the assembly of equipment through which connections are made between data stations.

network layer: In the ISO OSI reference model, the logical network entity that services the transport layer; responsible for ensuring that data passed to it from the transport layer is routed and delivered through the network.

network management: Administrative services performed in managing a network, such as network topology and software configuration, downloading of software, monitoring network performance, maintaining network operations, and diagnosing and troubleshooting problems.

network manager: A network entity which determines station status, collects configuration information, and measures network performance, among other things.

node: A station; a physical device that allows for the transmission of data within a network.

O

open systems interconnection (OSI): Referring to the international organization for standardization's OSI reference model, a logical structure for network operations standardized within the ISO; a seven-layer network architecture being used for the definition of network protocol standards.

P

physical control field (PCF): The physical control field (PCF) is a field of a frame which follows the starting delimiter. The PCF consists of two 8-bit fields, the access control (AC) field and the frame control (FC) field.

physical drop number: The physical drop number is a facilities-defined number used to assist in performing network management functions. The physical drop number is provided to individual stations by a ring parameter server or network manager if present on the ring.

physical layer: In the OSI reference model the lowest level of network processing, below the data link layer, that is concerned with the electrical, mechanical and handshaking procedures over the interface that connects a device to a transmission medium; referring to an electrical interface, such as RS–232–C.

presentation layer: In the OSI reference model, that layer of processing that provides services to the application layer, allowing it to interpret the data exchanged, and allowing it to structure data messages to be transmitted in a specific display and control format.

protocol: Formal set of rules governing the format, timing, sequencing, and error control of exchanged messages on a data network; may also include facilities for managing a communications link and/or contention resolution. A protocol may be oriented toward data transfer over an interface, between two logical units directly connected, or on an end-to-end basis between two end users over a large and complex network. Both hardware protocols and software protocols can be defined.

protocol handler: The TMS38020/21 protocol handler (PH) is a component of the first generation TMS380 adapter which serializes and de-serializes the ring bit stream and implements the real-time elements of ring protocol, such as CRC generation and checking and address recognition.

R

repeater: For local area networks, a device which increases the signal cover of a single LAN segment by joining it to another, so that frames sent on one segment can be repeated (or copied) onto another, increasing the local environment.

retry: The process of resending the current block of data a prescribed number of times or until it is accepted.

ring: A network topology in which stations are connected to one another in a closed logical circle. In a token-ring LAN, access to the ring passes sequentially from one station to the next by passing an access token from one station to another.

ring error monitor: The ring error monitor is a ring-resident function which maintains statistical records of error conditions on the ring operation.

ring error monitor (REM): The TMS38051 ring interface transceiver and the TMS38052 ring interface controller are bipolar components of the first generation TMS380 adapter. The RI transceiver converts a TTL-level signal from the TMS38020 protocol handler to the appropriate level for transmission on the ring. The RI controller contains the voltage-controlled oscillator for the phase-locked loop.

ring interface module: A module consisting of two bipolar integrated circuits, the TMS38051 and TMS38052 ring interface (RI) chips, transmit and receive transformers, and various passive components, which make up the adapter's electrical interface to the ring twisted pair wires.

ring network: A network topology in which each node is logically connected to two adjacent nodes.

ring parameter server (RPS): The ring parameter server is a host-based function on a ring which manages operating parameters for the ring. This function is supported in the MAC protocol.

ring station: In this document, a ring station consists of an adapter and attached product inserted on the ring.

ring station specific address: A ring station specific address is the address of a ring station which is the unique address known at the physical transmission level for a particular node.

ring status register: The ring status register is a 16-bit word which is used by the adapter to report ring conditions to the attached system.

routing: The process of selecting the correct circuit path for a message across rings or buses.

routing information field: The routing information field is an up to 18-byte field found immediately following the source address field, which is used to hold the necessary information for routing frames among multiple segments (rings, in the case of token ring) in the network.

S

serial interface: An interface which requires serial transmission, or the transfer of information in which bits composing a character are sent sequentially. Implies a single transmission channel.

server: A processor which provides a specific service to the network. Examples of servers are: routing server – connects nodes and networks of like architectures; gateway server – connects nodes and networks of different architectures by performing protocol conversions; terminal server, print server, and file server – provides an interface between compatible peripheral devices on a LAN.

session: A connection between two stations that allows them to communicate; the logical connection between two network addressable units.

session layer: In the OSI reference model, the network-processing layer responsible for binding and unbinding logical links between end users and maintaining an orderly dialogue between them; also responsible for naming of logical entities.

source: Originator of data.

source routing: Method by which stations are addressed across multiple rings.

stand-by monitor: A stand-by monitor is any adapter on the ring which is not currently the active monitor. The stand-by monitor functions are defined in the MAC protocol.

star: A network topology consisting of one central node with point-to-point links to several other nodes. Control of the network is usually located in the central node or switch, with all routing of network message traffic performed by the central node.

station: A network node.

subvector: A subvector is part of the MAC frame major vector. The subvectors are subfields within the major vector used to carry specific information used to process the MAC frame. The subvector is composed of length, type, and value fields.

system command block (SCB): The system command block is a six-byte buffer used to hold the command to be executed by the adapter and a 32-bit address pointer to a parameter block for the command.

system interface (SIF): The TMS38030 system interface is a VLSI component of the first generation adapter chip set which functions as the adapter's interface to the attached system.

system status block (SSB): The system status block is an eight-byte buffer used by the adapter to relay status information to the attached system, such as return codes of completed adapter commands.

throughput: The total useful information processed or communicated during a specified time period. Expressed in bits-per-second or bytes-per-second.

token: A token consists of 24 bits made up of the starting delimiter (SDEL), access control (AC), and ending delimiter (EDEL) fields of a frame. The token indicator (TI) bit of the Access control field is set to 0.

token bus: A token access procedure used with a broadcast topology or network. Defined by the IEEE 802.4 subcommittee.

token indicator: A bit in the access control field used to indicate a free token or a frame.

token passing: A mechanism whereby each device receives and passes the right to use the channel. Tokens are special bit patterns that circulate from node to node. Possession of the token gives a node exclusive access to the network for transmitting its message, thus avoiding conflict with other nodes that wish to transmit. Stations wishing to gain access to the medium must wait for a token to arrive before transmitting data.

token passing: A mechanism whereby each device receives and passes the right to use the channel. Tokens are special bit patterns that circulate from node to node. Possession of the token gives a node exclusive access to the network for transmitting its message, thus avoiding conflict with other nodes that wish to transmit. Stations wishing to gain access to the medium must wait for a token to arrive before transmitting data.

token ring: A local network access mechanism and topology which uses token passing protocol. In a token ring, the next logical station receiving the token is also the next physical station on the ring.

topology: Topology can be physical or logical. Physical topology is the configuration of network nodes and links; a description of the physical geometric arrangement of the links and nodes. Logical topology is a description of the possible logical connections between network nodes indicating which pairs of nodes are able to communicate whether or not they have a direct physical connection.

transceiver: A device required in baseband networks which takes the digital signal from a computer or terminal and imposes it on the baseband medium.

transport layer: In the OSI reference model, the network processing entity responsible, in conjunction with the underlying network, data link and physical layers, for the end-to-end control of transmitted data and the optimized use of network resources.

W

wiring concentrator: A wiring concentrator serves as the electrical interface of the lobe circuit (connecting to the adapter) to the ring circuit. It contains relays which provide for physical insertion to and de-insertion from the ring.

word: A word, as defined in this document, consists of two bytes (16 bits).

Index

A

adapter buffer management, 4-81
adapter hardware design, 3-1
adapter internal buffer, 4-82
adapter memory control, 3-8, 3-10
  DRAM refresh cycle, 3-9
  READ from BIA ROM, 3-8
  READ from DRAM, 3-8
  READ from EPROM, 3-8
  WRITE to DRAM, 3-8
adapter RAM check, 4-49
ARI/FCI Error, 2-53, 2-98, 2-103
ASCII Table, I-2

B

beacon removal functions, 2-102
BIA, 2-25
bring-up diagnostics (BUD), 4-40
buffer allocation, 4-82
burst error (burst 5 error), 2-98

C

character sets, I-2
commands
  LLC
    CLOSE.SAP, 4-190
      SCB, 4-190
      SSB, 4-190
    CLOSE.STATION, 4-194
      SCB, 4-194
      SSB, 4-194
    CONNECT.STATION, 4-195
      parameter block, 4-196
      SCB, 4-195
      SSB, 4-196
    DIR.INTERRUPT, 4-207
      SCB, 4-207
      SSB, 4-207
    FLOW.CONTROL, 4-202
      SCB, 4-202
      SSB, 4-203
    LLC.REALLOCATE, 4-208
      SCB, 4-208
      SSB, 4-209
    LLC.RESET, 4-183
      SCB, 4-183
      SSB, 4-183
    LLC.STATISTICS, 4-204
      parameter block, 4-204
      SCB, 4-204
      SSB, 4-206
    MODIFY.LLC.PARMS, 4-198
      parameter block, 4-199
      SCB, 4-198
      SSB, 4-201
    OPEN.SAP, 4-184
      parameter block, 4-185
      SCB, 4-184
      SSB, 4-189
    OPEN.STATION, 4-191
      parameter block, 4-191
      SCB, 4-191
      SSB, 4-193

TIMER.SET, 4-210
SCB, 4-210
SSB, 4-210
TRANSMIT (Adapter-Based I Frame), 4-148
parameter list, 4-149
SCB, 4-148
SSB, 4-156
TRANSMIT (Host-Based I Frame), 4-137
parameter list, 4-138
SCB, 4-137
SSB, 4-145
TRANSMIT.I.FRAME.REQUEST, 4-211
SCB, 4-211
SSB, 4-212
MAC
CLOSE, 4-109
SCB, 4-109
SSB, 4-109
CONFIG.BRIDGE.PARMS, 4-123, 4-182
SCB, 4-123
SSB, 4-123
MODIFY.OPEN.PARAMETERS, 4-117
SCB, 4-117
SSB, 4-117
OPEN
parameter list, 4-72
SCB, 4-71
SSB, 4-79
READ.ADAPTER, 4-115
buffer, 4-115
SCB, 4-115
SSB, 4-116
READ.ERROR.LOG, 4-112
error log table, 4-112
SCB, 4-112
SSB, 4-114
RECEIVE, 4-100
CSTAT, 4-106
parameter list, 4-101
SCB, 4-101
SSB, 4-107
RESTORE.OPEN.PARAMETERS, 4-118
SCB, 4-118
SSB, 4-118
SET.BRIDGE.PARMS, 4-120, 4-179
BRIDGE_PARM_BLOCK, 4-120, 4-179
SCB, 4-120
SSB, 4-122
SET.FIRST.16.GROUP.ADDRESS, 4-119
SCB, 4-119
SSB, 4-119
SET.FUNCTIONAL.ADDRESS, 4-111
SCB, 4-111
SSB, 4-111
SET.GROUP.ADDRESS, 4-110
SCB, 4-110
SSB, 4-110
TRANSMIT, 4-84, 4-86
CSTAT, 4-93
frame format, 4-84
parameter list, 4-87
SCB, 4-86
SSB, 4-94
TRANSMIT.HALT, 4-99
SCB, 4-99
SSB, 4-99
CPU, 1-12

D

data sheet
COMMprocessor, A-1
ring interface, B-1
development support, ordering information, 5-6, 5-7
differential Manchester code, 2-8
code violation, 2-9
encoded 0 bit, 2-8
encoded 1 bit, 2-8
DIO, 3-39, 3-51
16-bit, 3-43, 3-47, 4-16, 4-28
8-bit, 3-43, 3-47, 4-16, 4-28
DMA, 3-68
16-bit, 3-81
8-bit, 3-80
dynamic windowing, 2-123

E

early token release, 2-5

- elastic buffer, 2-9

F

- frame buffers, 4-57
- frame copied, 2-21, 2-99
- frames, 2-28, 2-117
  - format
    - *AC field, 2-16, 2-18*
      - monitor count, 2-18
      - priority, 2-18
      - reservation, 2-18
      - token indicator, 2-18
    - *destination address (DA), 2-17, 2-19*
      - format of, 2-19
    - *DSAP, 2-118*
    - *end delimiter (EDEL), 2-20*
    - *FC field, 2-16*
      - type bits, 2-19
        - MAC control, 2-19
        - non-MAC control, 2-19
    - *frame check sequence (FCS), 2-17, 2-20*
    - *frame status field (FS), 2-21*
    - *information field (data), 2-29*
      - MAC frame major vector, 2-29, 2-30
      - MAC frame subvectors, 2-29, 2-31
    - *LLC control, 2-118*
    - *PCF attention code, 2-19*
      - active monitor present, 2-19
      - beacon, 2-19
      - claim token, 2-19
      - express buffer, 2-19
      - ring purge, 2-19
      - standby monitor present, 2-19
    - *routing control, 2-26*
      - broadcast bits, 2-26
      - direction, 2-27
      - largest frame, 2-27
      - length, 2-26
    - *routing information field (RI), 2-26*
      - bridge number, 2-27
      - ring number, 2-27
      - segment number, 2-26, 2-27
    - *SDEL, 2-16, 2-18*
    - *source address (SA), 2-17, 2-19*
      - format of, 2-19
      - in bridging, 2-26
    - *SSAP, 2-118*
  - types, 2-16
    - *higher-layer protocol, 2-16*
    - *LLC*
      - DISC, 2-119
      - DM, 2-119
      - format of, 2-117
      - FRMR, 2-119
      - I, 2-119
      - RNR, 2-120
      - RR, 2-120
      - SABME, 2-119
      - test, 2-119
      - UA, 2-119
      - UI, 2-118
      - XID, 2-118
    - *MAC*
      - active monitor, 2-29, 2-33
      - beacon, 2-29, 2-33, 2-44
      - claim token, 2-29, 2-33
      - format of, 2-28
      - ring poll, 2-44
      - ring purge, 2-29, 2-35, 2-44
      - standby monitor, 2-29, 2-35
- frequency error, 2-100
- functional address, 2-23
  - active monitor, 2-23
  - bridge, 2-23
  - network manager, 2-23
  - ring error monitor, 2-23
  - ring parameter server, 2-23

G

- glossary, J-1
- group node addresses, 2-23
  - broadcast, 2-23
  - functional, 2-23
  - group, 2-23

H

- header routing, 4-106

I

- ICB, 4-53

IEEE 802 LAN Standards, 2-4
IEEE 802.1, 2-4
IEEE 802.2, 2-4, 2-5, 2-7, 2-11
IEEE 802.3, 2-4
IEEE 802.4, 2-4
IEEE 802.5, 2-4, 2-5, 2-7, 2-8, 2-11, 2-28, 2-31
initialization, adapter, 4-42
internal error, 2-100

L

LAN header, 4-57
license information, 5-2
line error, 2-98
link station, 1-9
LLC header, 4-58, 4-135
LLC Numbers
  N1, 2-124
  N2, 2-124
  N3, 2-124
  Nw, 2-124, 2-125
  Tw, 2-124, 2-125
  Ww, 2-124, 2-125
LLC Protocol States
  Closed, 2-121
  Disconnected, 2-121
  Disconnecting, 2-121
  Frame Reject Received, 2-122
  Frame Reject Sent, 2-122
  Opened, 2-122
    Checkpointing, 2-122
    Clearing, 2-123
    Local Busy, 2-123
    Rejection, 2-123
    Remote Busy, 2-123
  Opening, 2-122
  Resetting, 2-122
LLC Timers
  T1, 2-123
  T2, 2-123
  TI, 2-123
LLC_ENABLE, 4-44, 4-70
lost frame error, 2-98

M

MAC Frame Class Designators, 2-29
  Hardware Error, 2-44
  LLC Manager Class, 2-29
  Network Manager Class, 2-29
  Ring Error Monitor Class, 2-29
    Phase 1 – Physical Insertion, 2-61, 2-64
    Phase 2 – Address Verification, 2-61, 2-65
    Phase 3 – Participation in Ring Poll, 2-61, 2-66
    Phase 4 – Request Initialization, 2-61, 2-67
  Ring Parameter Server Class, 2-29
  Ring Station Class, 2-29
  Transmit Forwarding, 2-69
MAC frame class designators, soft error counting, 2-96
MAC Processes, 2-44
  Beacon, 2-44, 2-55
  Monitor Contention, 2-44, 2-46
  Ring Insertion, 2-44, 2-60
    Phase 0 – Lobe Media Check, 2-61, 2-63
  Ring Poll, 2-44, 2-51
  Ring Purge, 2-44, 2-50
mechanical data, 5-8
MIF, 1-16

N

Network Management, 2-11, 2-71

O

OSI Model, 2-2
  Application, 2-3
  Data Link, 2-3
  Network, 2-3
  Physical, 2-3
  Presentation, 2-3
  Session, 2-3
  Transport, 2-3

P

packaging, 5-8
parity, 3-8, 3-9
performance, 1-7
PH, 1-15
pseudo-DMA, 3-40

R

receive congestion, 2-99
Ring Clocking, 2-9
Ring Station Specific Address, 2-25

S

SAP, 2-114, 2-115, 4-184, 4-185, 4-190
  global, 4-71, 4-125
  null, 4-71, 4-125
SCB, 4-13
SIF, 1-13
SIF registers
  DMALEN, 4-26
  SDMAADR, 4-26
  SDMAADX, 4-26
  SIFACL, 4-18
  SIFADR, 4-23, 4-25
  SIFADX, 4-23, 4-25
  SIFCMD, SIFSTS, 4-30, 4-31, 4-32, 4-34
  SIFDAT, 4-23, 4-25
  SIFDAT/INC, 4-23, 4-25
Signal Error, 2-110, 2-111
software, XDS, ordering information, 5-6
Source Routing Bridging, 2-26
SSB, 4-13
Streaming Error, 2-100, 2-101
system bus error, 3-75
system HALT, 3-75
system interrupts
  ADAPTER.CHECK, 4-66
    FIELD, 4-68
    SSB, 4-66
  COMMAND.REJECT, 4-64
    FIELD, 4-65
    SSB, 4-64
  RING.STATUS, 4-61
    FIELD, 4-63
    SSB, 4-61
system re-run cycle, 3-75

T

test, quality, and reliability, E-1
Token
  Capture of, 2-73
  Model Access Control (AC), 2-73
  Priority Control Protocol, 2-75
Token Error, 2-99