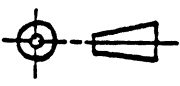


Spec set 1

<b>Honeywell</b> HONEYWELL INFORMATION SYSTEMS		SPEC. NO. 60149766	SHEET 1/101	REV. H
LOC. Billerica, MA U.S.A.		PROJECTION 	CODE C61C	
PREPARED BY A.C. Hirtle	DATE 5/20/83	TITLE Engineering Product Specification, Part 1 Local Area Controller Subsystem (LACS)		
APPROVED BY See Approval Sheet	DATE			

REVISION RECORD

REV.	AUTHORITY	DATE	SIGNATURE	SHEETS AFFECTED
A	BLCDD7588	5/20/83	<i>ACH</i>	All
B	-----	8/11/83	<i>ACH</i>	All
C	-----	11/4/83	<i>ACH</i>	As marked
D	-----	1/12/84	<i>ACH</i>	As marked
E	-----	4/18/84	<i>ACH</i>	As marked
F	BLCDE7775	10/4/84	<i>ACH</i>	As marked
G	BLCDF7256	3/31/85	<i>ACH</i>	As marked
H	BLCDG7319	4/30/86	<i>ACH</i>	As marked

NOTE

This EPS-1, if not revised within one year, should be considered obsolete and therefore reference should be made to the appropriate product manual.

This document and the information contained herein are confidential to and the property of Honeywell Information Systems Inc. and are made available only to Honeywell employees for the sole purpose of conducting Honeywell's business. This document, any copy thereof and the information contained herein shall be maintained in strictest confidence; shall not be copied in whole or in part except as authorized by the employee's manager; and shall not be disclosed or distributed (a) to persons who are not Honeywell employees, or (b) to Honeywell employees for whom such information is not necessary in connection with their assigned responsibilities. Upon request, or when the employee in possession of this document no longer has need for the document for the authorized Honeywell purpose, this document and any copies thereof shall be returned to the employee's manager. There shall be no exceptions to the terms and conditions set forth herein except as authorized in writing by the responsible Honeywell Vice President.

1951  
1952  
1953

Code

1954

EMPLOYERS INFORMATION  
STATE OF CALIFORNIA

1955

1956

1957

1958

1959

1960  
1961  
1962  
1963  
1964  
1965  
1966  
1967  
1968  
1969  
1970  
1971  
1972  
1973  
1974  
1975  
1976  
1977  
1978  
1979  
1980  
1981  
1982  
1983  
1984  
1985  
1986  
1987  
1988  
1989  
1990  
1991  
1992  
1993  
1994  
1995  
1996  
1997  
1998  
1999  
2000  
2001  
2002  
2003  
2004  
2005  
2006  
2007  
2008  
2009  
2010  
2011  
2012  
2013  
2014  
2015  
2016  
2017  
2018  
2019  
2020  
2021  
2022  
2023  
2024  
2025

HONEYWELL		SPEC. NO. 60149766	SHEET 1.1	REV. H
Author: A. C. Hirtle First Draft Date 5/20/83			Code  C61C	

<b>HONEYWELL INFORMATION SYSTEMS SYSTEMS ENGINEERING - BOSTON</b>	
ENGINEERING	
PRODUCT	LOCAL AREA CONTROLLER TITLE: <u>SUBSYSTEM (LACS)</u>
SPECIFICATION	VERSION: <u>APPROVED</u> DATE: <u>4/30/86</u>
( EPS - 1 )	PRODUCT CALENDAR REFERENCE _____
SUBSYSTEM LEVEL	_____

NOTE

This EPS-1, if not revised within one year, should be considered obsolete and therefore reference should be made to the appropriate product manual.

This document and the information contained herein are confidential to and the property of Honeywell Information Systems Inc., and are made available only to Honeywell employees for the sole purpose of conducting Honeywell's business. This document, any copy thereof and the information contained herein shall be maintained in strictest confidence; shall not be copied in whole or in part except as authorized by the employee's manager; and shall not be disclosed or distributed (a) to persons who are not Honeywell employees, or (b) to Honeywell employees for whom such information is not necessary in connection with their assigned responsibilities. Upon request, or when the employee in possession of this document no longer has need for the document for the authorized Honeywell purpose, this document and any copies thereof shall be returned to the employee's manager. There shall be no exceptions to the terms and conditions set forth herein except as authorized in writing by the responsible Honeywell Vice President.



CONTENTS

	<u>PAGE</u>
SECTION 1 INTRODUCTION.....	3
1.1 Document Definition.....	3
1.2 Reference Documents.....	3
1.3 Abbreviations/Definition.....	4
SECTION 2 ARCHITECTURE.....	6
2.1 Overview.....	6
2.2 Security.....	10
2.3 Multiprocessor Support.....	10
2.4 Local Area Controller (LAC).....	11
2.5 LAC Block Diagram.....	12
2.6 Memory Layout.....	14
2.7 LAC Memory Management.....	16
2.8 LAC Software-Firmware-Hardware Structure.....	17
2.8.1 General	17
2.8.2 Interprocess Communications	18
2.8.3 Relative Process Priorities	18
2.9 Mailboxes and Mailbox Message Queues.....	20
2.9.1 Mailboxes Associated with IF Software	20
2.9.2 Mailbox Message Priorities.....	21
2.10 LAN Control Block.....	21
2.11 LAC Software Interface to the Megabus.....	21
2.12 LAC Software Interface with Adapters.....	21
2.13 Software Interface Between Level 6 and LACS....	22
2.13.1 Transmit Data Operation Flow.....	22
2.13.2 Receive Data Operation Flow.....	27
2.13.3 Control Operations.....	31
2.14 LL Software & IF Software Functional Diagrams..	33
2.15 Interrupts to Level 6.....	37
2.16 Statistics.....	38
2.17 LACS Startup.....	39
2.18 Timers.....	41
2.19 Handling of Software Error Conditions.....	41
SECTION 3 FUNCTIONAL DESCRIPTION (LAC).....	44
3.1 Basic Functions.....	44
3.2 I/O Orders.....	45
3.2.1 List of I/O Orders.....	45
3.2.2 Output Order Definitions.....	45
3.2.3 Input Order Definition.....	47
3.2.4 WAIT, NAK and ACK Responses.....	47
3.2.5 Initialization, Start/Stop and Load/Dump	48

CONTENTS (CONTINUED)

	<u>PAGE</u>
3.3 Channel Numbers.....	52
3.4 Device Identification Number.....	52
3.5 Data Transfers within the LAC.....	53
3.6 Instructions.....	53
3.7 PROM Firmware.....	53
3.8 IF Software.....	54
3.8.1 IF Software Processes	54
3.8.1.1 Memory DMA Request (MEMDMA)	54
3.8.1.2 MAC	64
3.8.1.3 I/O Order Dispatch (IODISP)	66
3.8.1.4 Megabus Layer Management	72
3.8.1.5 MAC Layer Management	72
3.8.2 IF Software Interrupt Routines	74
3.8.2.1 Interrupt Retry	74
3.8.2.2 I/O Dispatch	75
3.8.2.3 DMA Controller	76
3.8.2.4 Megabus DMA Register Exhaust	77
3.8.2.5 Adapter Interrupt	77
3.8.2.6 RAM Parity Error	77
3.9 Handling of Hardware Errors	80
SECTION 4 INTERFACES	81
4.1 Level 6 Interface with LACS	81
4.2 LAC Interface with the Megabus	81
4.3 LAC Interface with Adapters	82
4.4 Test Connector Interface	83
SECTION 5 PERFORMANCE	84
5.1 LAN Data Rate	84
5.2 Data Throughput	84
5.3 LAC Instruction Timing	84
5.4 Priority Position of LAC	84
5.5 Priority Position of Adapters	84
SECTION 6 PHYSICAL STRUCTURE	85
6.1 Packaging	85
6.2 Power	85
6.3 Environment	85
6.4 Safety	85
6.5 Standards	85
6.6 Testability	85
6.7 RFI Emanation	85

CONTENTS (CONTINUED)		<u>PAGE</u>
SECTION 7	RELIABILITY AND MAINTAINABILITY	86
7.1	General Requirements	86
7.2	Maintenance Features	86
7.2.1	Quality Logic Test (QLT)	86
7.2.2	Error Checking and Reporting	86
7.2.3	Internal Loopback	86
7.2.4	Statistic Gathering	87
7.2.5	LAC RAM Load/Dump	87
7.2.6	FCS Checking	87
7.2.7	Adapter RAM Access	87
7.2.8	Remote Loopback	87
7.2.9	MAC Internal Status	87
7.2.10	Remote Station Inquiry (XID)	87
7.2.11	System Area in RAM	88
7.3	Maintainability	88
7.4	Detailed Address Space Assignments	89
7.4.1	Overall Maps	89
7.4.2	Dedicated RAM Area Layouts	91
APPENDIX A	TOKEN BUS ADAPTER	93
A.1	General Description	96
APPENDIX B	CSMA/CD (ETHERNET) ADAPTER	96
B.1	General Description	98
APPENDIX C	TOKEN RING ADAPTER	98
C.1	General Description	98

ILLUSTRATIONS

FIGURE		PAGE
2-1	LACS Applications to Local Area Networks	8
2-2	LACS Applications	9
2-3	LAC Block Diagram	12
2-4	LAC RAM Memory Map	14
2-5	LAC Operating Structure	19
2-6	LLC Transmit Flow	26
2-7	LLC Receive Flow	28
2-8	I/O Dispatch Process	33
2-9	Memory DMA Process	34
2-10	Adapter Interrupt	35
2-11	Adapter-Specific MAC Processes	36
3-1	LCB Format for Load/Dump	50
3-2	LCB Format for Start I/O	51
3-3	DMA Request/Confirm	58
3-4	DMA Request-Confirm	59
3-5	DMA Request/Confirm	60
3-6	DMA Request/Confirm	61
3-7	Retry Dispatch Table Request/Confirm	62
3-8	L6 Buffer List	63
3-9	Data Transmit/Fault Indicate	66
3-10	Fault Indicate Status From Data Transmit	67
3-11	Event Indicate Message Block	68
3-12	IODISP Dispatch Table Setup	71
3-13	IOLD Information Message Block	76
3-14	Data Indicate Message Block	78
3-15	LMI Request/Confirm	79
4-1	LAC-Adapter Bus	82
4-2	Test Connector	83
7-1	Address Space Layout During QLT	89
7-2	Address Space Layout During Normal Operation	90
7-3	I/O Order Temporary Queue	91
7-4	Configuration Information Area	92
7-5	LAC Error Status	92
A-1	Token Bus Block Diagram	94
B-1	CSMA/CD (ETHERNET) Block Diagram	97
C-1	Token Ring Block Diagram	99

TABLES

2-1	Software Event Notification Codes	43
-----	-----------------------------------	----



HONEYWELL		SPEC. NO.	SHEET	REV
		60149766	3	H

## SECTION 1 INTRODUCTION

### 1.1 DOCUMENT DEFINITION

This document defines the functional characteristics and program-visible architecture of the Local Area Controller Subsystem (LACS).

### 1.2 REFERENCE DOCUMENTS

- |     |                 |   |
|-----|-----------------|---|
| 1.  | 60149824        | DPS6 Local Area Network Controller PFS  |
| 2.  | 58088015        | Local Area Networks Global Functional Spec.   |
| 3.  | CC71            | Level 6 Minicomputer Handbook   |
| 4.  | DSA-41          | Local Area Networks   |
| 5.  | IEEE 802.1      | Local Area Network Overview, Rev. B, June 1983  |
| 6.  | ISO 7498        | Open System Interconnection Ref. Model  |
| 7.  | IEEE 802.2      | Local Area Network Logical Link Control, Draft E, March 1984  |
| 8.  | IEEE 802.3      | Local Area Network CSMA/CD Access, July 1983  |
| 9.  | IEEE 802.4      | Local Area Network Token Bus Access, Draft E, July 1983   |
| 10. | IEEE 802.5      | Local Area Network Token Ring Access, Draft 9/23/83   |
| 11. | -----           | Ethernet Specification, Version 2.0 (Digital Equipment Corp., Intel Corp., Xerox Corp.)                     |
| 12. | B01.08          | HIS Standard, Operating Environment   |
| 13. | B01.09          | HIS Standard, Equipment Safety  |
| 14. | D.002.01        | PWB/PWA Testability Design Rules  |
| 15. | 60149832        | MRX Megabus EPS-1   |
| 16. | MTG2            | PWA Test Documentation Requirements   |
| 17. | MTG3            | PWA Microdiagnostic Creation  |
| 18. | MTG5            | QLT Creation  |
| 19. | MTG6            | T&V Creation  |
| 20. | 58035052        | Worldwide Maintenance Requirements  |
| 21. | MC68000UM (AD2) | MC68000 User's Manual (Motorola Semi-conductor Products, Inc.)  |
| 22. | 60126298        | Level 6 Bus (Megabus) EPS-1   |
| 23. | 03860507        | Purchase Specification for RF Modem   |
| 24. | 210891-001      | 82586 Reference Manual (INTEL Corp.)  |
| 25. | 09-0016-00      | ESPL Software Technical Reference Manual, Vol. 1, Kernel and Support Software (Bridge Communications, Inc.) |
| 26. | 60149817        | LAN Software EPS-1  |

### 1.3 ABBREVIATIONS/DEFINITIONS

ACK	-	Positive Acknowledgement (as defined by the L6 Bus EPS-1)
CM	-	Controller Management (Software)
CRC	-	Cyclic Redundancy Check
CPU	-	Central Processing Unit
CS	-	Communication Service (Software)
CSMA/CD	-	Carrier Sense Multiple Access/Collision Detect
DMA	-	Direct Memory Access
DA	-	Destination Address
DSAP	-	Destination Service Access Point
FC	-	Function Code/Frame Control
FIFO	-	First-In-First-Out
GA	-	Group Address
ICW	-	Interrupt Control Word
IORB	-	Input/Output Request Block
ID	-	Identification
IF	-	Interface (Software)
I/O	-	Input/Output
IOLD	-	Input/Output Load
LAC	-	Local Area Controller
LACS	-	Local Area Controller Subsystem
LAN	-	Local Area Network
LCB	-	LAN Control Block
LCBI	-	LAN Control Block Image
LLC	-	Link Layer Control
LME	-	Layer Management Entity
LMI	-	Layer Management Interface
LSAP	-	Link Service Access Point
LSI	-	Large Scale Integration
MAC	-	Media Access Controller
MBZ	-	Must be Zero
MSB	-	Most Significant Byte
MSb	-	Most Significant bit
MTBF	-	Mean Time Between Failures
MTTR	-	Mean Time To Repair
NAK	-	Negative Acknowledgement (defined by the L6 Bus EPS-1)
ORU	-	Optimum Replaceable Unit
OS	-	Operating System
OSI	-	Open Systems Interconnection

HONEYWELL		SPEC. NO.	SHEET	REV
		60149766	5	H

1.3 ABBREVIATIONS/DEFINITIONS (Continued)

PC	-	Personal Computer
PIO	-	Physical Input/Output
PROM	-	Programmable Read-Only Memory
PDU	-	Protocol Data Unit
QLT	-	Quality Logic Test
RAM	-	Random Access Memory
RFU	-	Reserved for Future Use
RHU	-	Reserved for Hardware Use
RSU	-	Reserved for Software Use
SA	-	Source Address/Station Address
SC	-	Service Call
SMDSI	-	Systems Management Data Service Interface
SSAP	-	Source Service Access Point
TBD	-	To Be Defined
TC	-	Trunk Coupler
T&V	-	Test and Verification
WS	-	Work Station



HONEYWELL		SPEC. NO.	SHEET	REV
		60149766	6	H

## SECTION 2 ARCHITECTURE

### 2.1 OVERVIEW

The Local Area Controller Subsystem (LACS) is a programmable communications subsystem that connects to the Level 6 Megabus. LACS comprises the following set of communications components:

- o Local Area Controller (LAC) Motherboard
- o Media Access Controller (MAC) & Physical Layer Adapters
- o Trunk Couplers (TCs)
- o RF Modems

This specification is concerned with the definition and description of the first two items above (i.e., the LAC and the adapters).

With suitable software loads and hardware/ adapter selection, the LACS is intended to be capable of supporting any of the IEEE 802 Local Area Network Standards.

The design of the LACS minimizes the interactions required over the Level 6/LACS interface and isolates the LACS on-board communications software from the specific hardware characteristics of the Level 6 and LAN adapter interfaces via supporting interface software.

A communications kernel based on one by Bridge Communications Inc., is used as the O.S. within the LAC. In this specification "CS Software" (Communication Service) refers to LAC-resident software which implements the OSI Link, Network, and Transport layers; "SM Software" (System Management layer instance) refers to LAC-resident software which supports IEEE802 System Management functions. CS and SM software will be written in "C" language by Honeywell Software Development and is the subject of a separate EPS-1.

"IF Software" (InterFace) refers to LAC-resident software which provides interfacing support between the CS and SM software and the LACS hardware. IF software will be written in "C" and assembly language by Honeywell Hardware Development and its functionality and interfaces are described in this specification. The Term "LAC Software" refers to all LAC-resident software (O.S. Kernel, CS SM and IF).

HONEYWELL		SPEC. NO.	SHEET	REV
		60149766	7	H

Although the IEEE 802 Standard goes no higher than a standard data link control interface (Layer 3/Layer 2), the Level 6-to-LACS interface that is provided is so heavily software-defined and flexible as to be readily adaptable to support of higher (e.g. Session/Transport) layer interfaces (with suitable Level 6 and CS software).

The LACS, used for all Local Area Network (LAN) applications, is mounted in a standard Level 6 chassis and requires one slot on the Megabus; it will support the 32-bit address bus of the larger Level 6 systems. The LAN adapters provide an interface from the LAC to the LAN. The adapter (a daughterboard) includes a Media Access Controller (MAC). The LAC provides for the attachment of up to four adapter daughterboards.

The adapters are of several types (e.g., Token Bus MAC, CSMA/CD MAC, etc.).

The TCs are of several types (for example, broadband directional coupler, token ring TCU, Ethernet transceiver) and are packaged as separate units. The RF modem, used for broadband applications, is also separately packaged.

Because of its ability to support adapters of similar or dissimilar types, the LACS (with appropriate LAC software) can be used not only for 802 LAN connection with a Level 6 but also in future as a gateway between 802 LANS, or, in the case of broadband LANs, as a bridge between broadband channels. Other applications for the LACS could be as LAN traffic monitor/journalizer and network control. The CS and SM software would, of course be tailored for each application; it is expected that the IF software and O.S. Kernel will be used in all applications.

Figure 2-1 shows a Local Area Network with LACS providing connections for Level 6 systems, for workstation LAN access, and for Gateway between LANs; Figure 2-2 shows various configurations of the LACS. A maximum of two ports are shown in use.

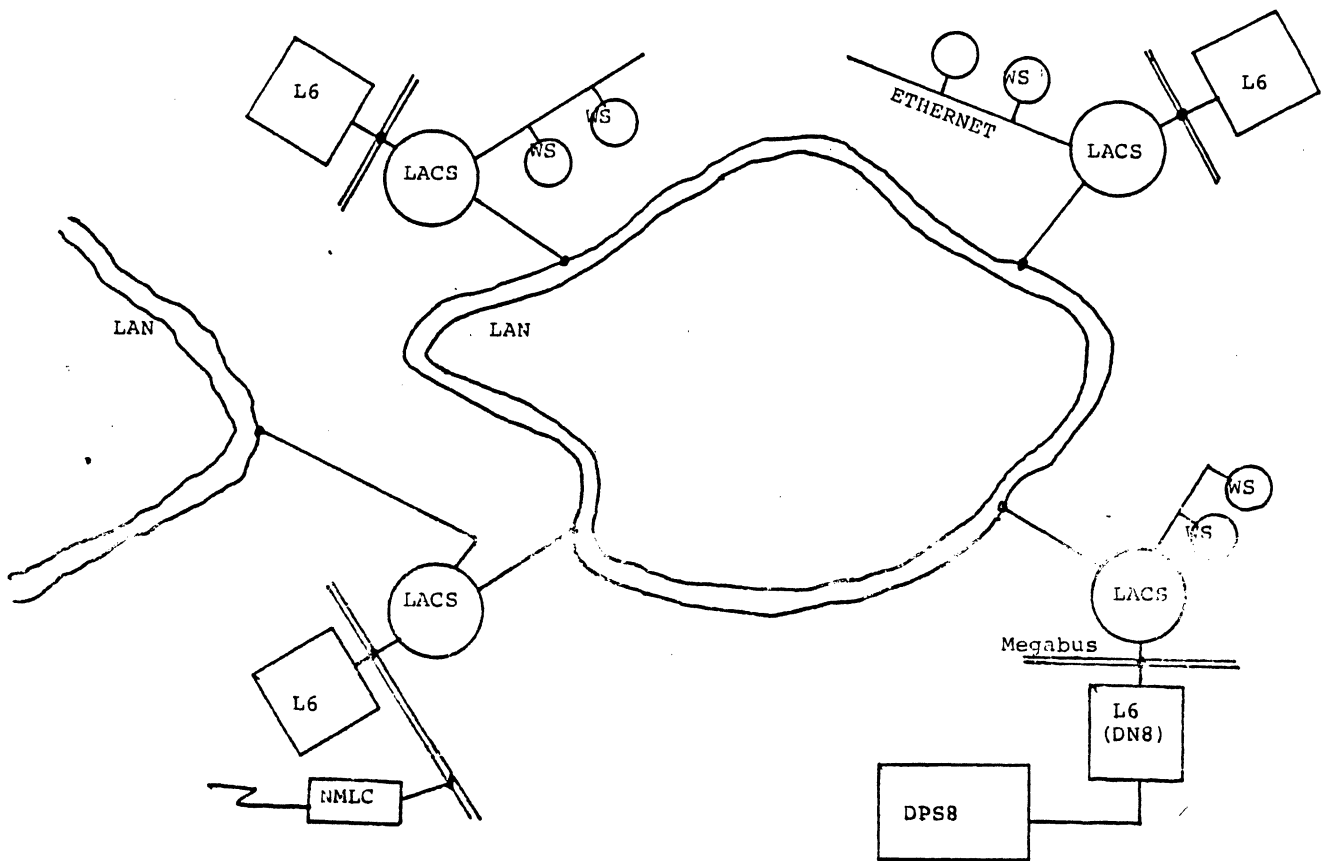
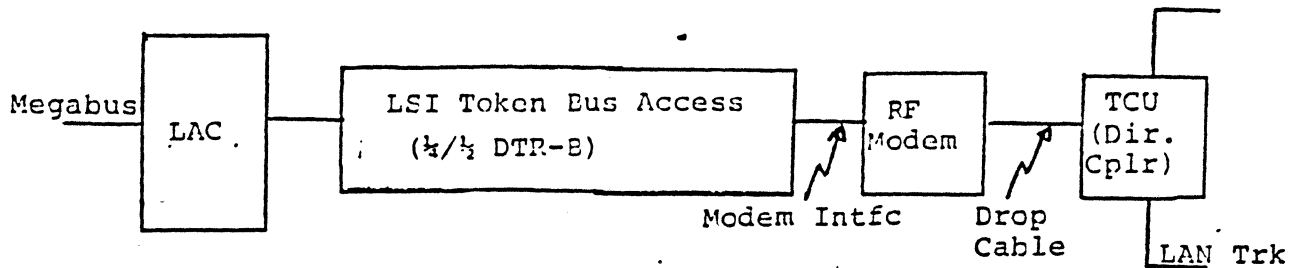
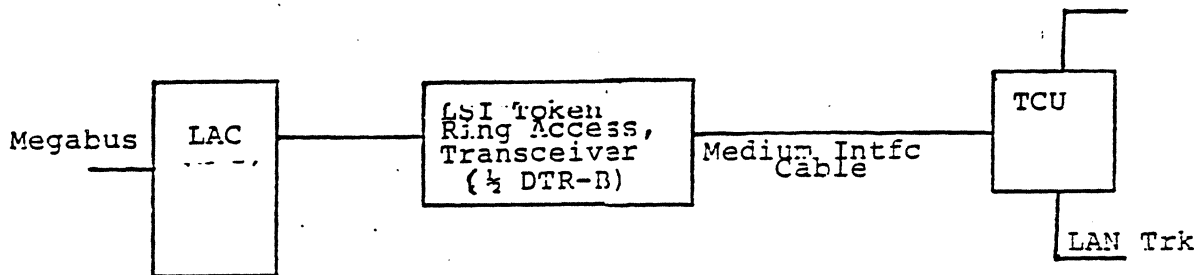


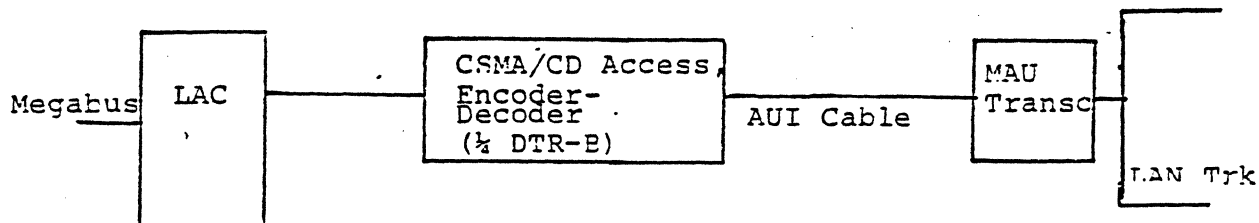
Figure 2-1 LACS APPLICATIONS TO LOCAL AREA NETWORKS



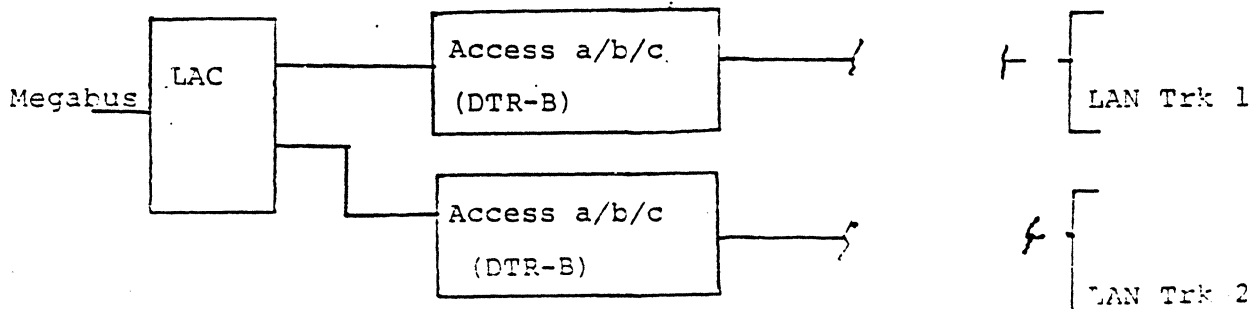
a. Broadband Token Bus Application



b. Token Ring Application



c. CSMA/CD Application



d. Gateway Application

Figure 2-2 LACS APPLICATIONS



HONEYWELL		SPEC. NO.	SHEET	REV
		60149766	10	H

## 2.2 SECURITY

No encryption/decryption functionality is defined for the LACS for the following reasons:

1. The IEEE 802 Standards do not specify any link layer or station encryption technique.
2. For a LAN, the use of the National Bureau of Standards Data Encryption Standard (DES) would be better applied at the Session or Presentation layer.
3. Key management on a LAN at the link layer presents many logistical problems with a DES technique. It would seem there is a good possibility that if an encryption technique is defined for LAN use, it will be based on some kind of Public Key scheme applied by the station.
4. Standards for Public Key systems do not exist at present.

## 2.3 MULTIPROCESSOR SUPPORT

The LAC provides for multiprocessor central systems (having up to 16 processors) in its I/O interface. Each I/O command block (LCB) issued to the LACS will contain fields which indicate the channel number of the CPU which is to be interrupted on completion of the function called for and the interrupt LEVEL to use. The LAC Software retains this information while carrying out the operation.

LAC hardware provides an interlock mechanism to handle the problem of two or more CPUs attempting to issue IOLDs (with Function Code 09/OD) simultaneously to the LAC. Further details are in subsection 3.2.2.

LAC hardware provides the specific capabilities required for connection to the multiprocessor type Megabus as well as to a monoprocessor type Megabus. Further details are in subsection 4.1.

HONEYWELL		SPEC. NO.	SHEET	REV
		60149766	11	H

#### 2.4 LOCAL AREA CONTROLLER (LAC)

The LAC, via hardware, software-generated parameters and CS software, provides OSI layer functions such as sequencing of outgoing frames, error control and retransmission, sequence checking of incoming frames and statistics gathering.

There are three software-visible interfaces associated with the LACS which are described in this specification: the CS/SM Software - IF Software interface in operations with the CPU and main memory; the CS/SM Software- IF Software interface in operations with the attached adapter(s) and the LACS I/O Order interface as seen by the LACS Driver in the CPU.

The operations of the three interfaces listed above and a brief description of the physical organization of the LAC are covered in the remainder of this Section. Further details of the physical organization of the LACS and its physical interfaces are covered in Section 4, 6, and 7 and in the appendices.

2.5 LAC BLOCK DIAGRAM

A block diagram of the LAC is shown in Figure 2-3.

The LAC uses a commercially popular microprocessor (MC68000) and presents that microprocessor's bus to the adapters.

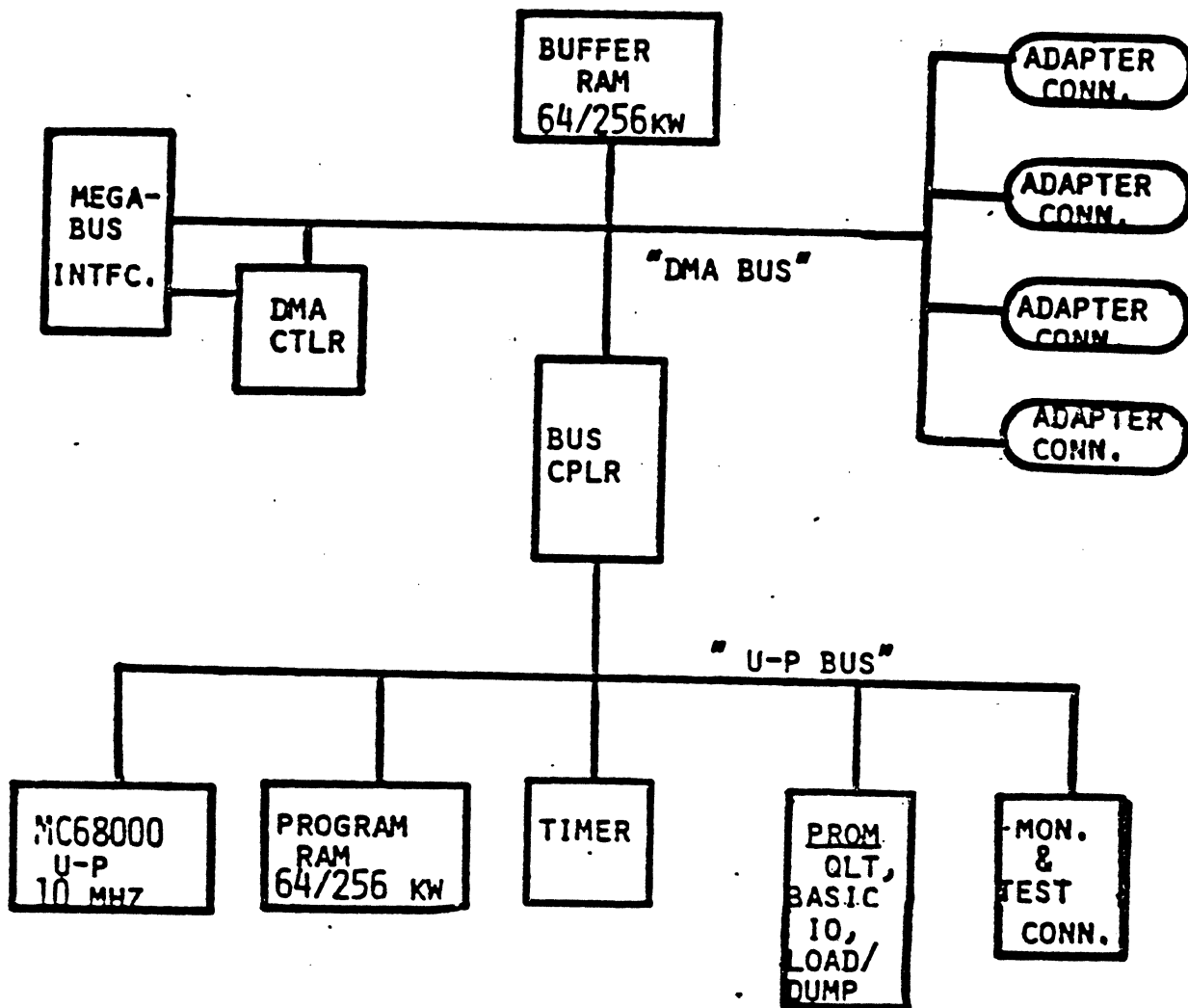


Figure 2-3 LAC BLOCK DIAGRAM

HONEYWELL		SPEC. NO.	SHEET	REV
		60149766	13	H

The RAM is physically separated into two sections: a data buffer RAM and program RAM. The intent of the separation is to allow for simultaneous DMA of data in the data buffer RAM with the Level 6 memory or with the LAN adapters along with software execution in the program RAM. The Bus Coupler allows for simultaneous independent operation of the MC68000 busses on each side, yet permits the microprocessor to perform accesses to any location in the total RAM.

The DMA controller is a two-channel device; one channel is used by the microprocessor to perform the DMA movement of data between main memory and the data buffer RAM. The other channel is used to accept I/O order information from the Megabus and deliver it to a temporary queue in the data buffer RAM for further analysis and disposition by firmware or IF software.

The timer device provides the basic clock tick for the LAC operating system to use in providing timer functions for LAC software .

DMA functionality for the adapters is provided by hardware located on the adapters themselves. Adapter DMA is always into or out of the data buffer RAM.

Data movement between the program RAM and the data buffer RAM is performed directly by the MC68000 microprocessor; data movement between the program RAM and main memory (as in Load/Dump operations) is performed in two steps: a movement between program RAM and data buffer RAM under control of the microprocessor, and a movement between data buffer RAM and main memory performed by the DMA controller.

## 2.6 MEMORY LAYOUT

The Software-visible layout of the LAC RAM as it exists during normal running is shown in Figure 2-4.

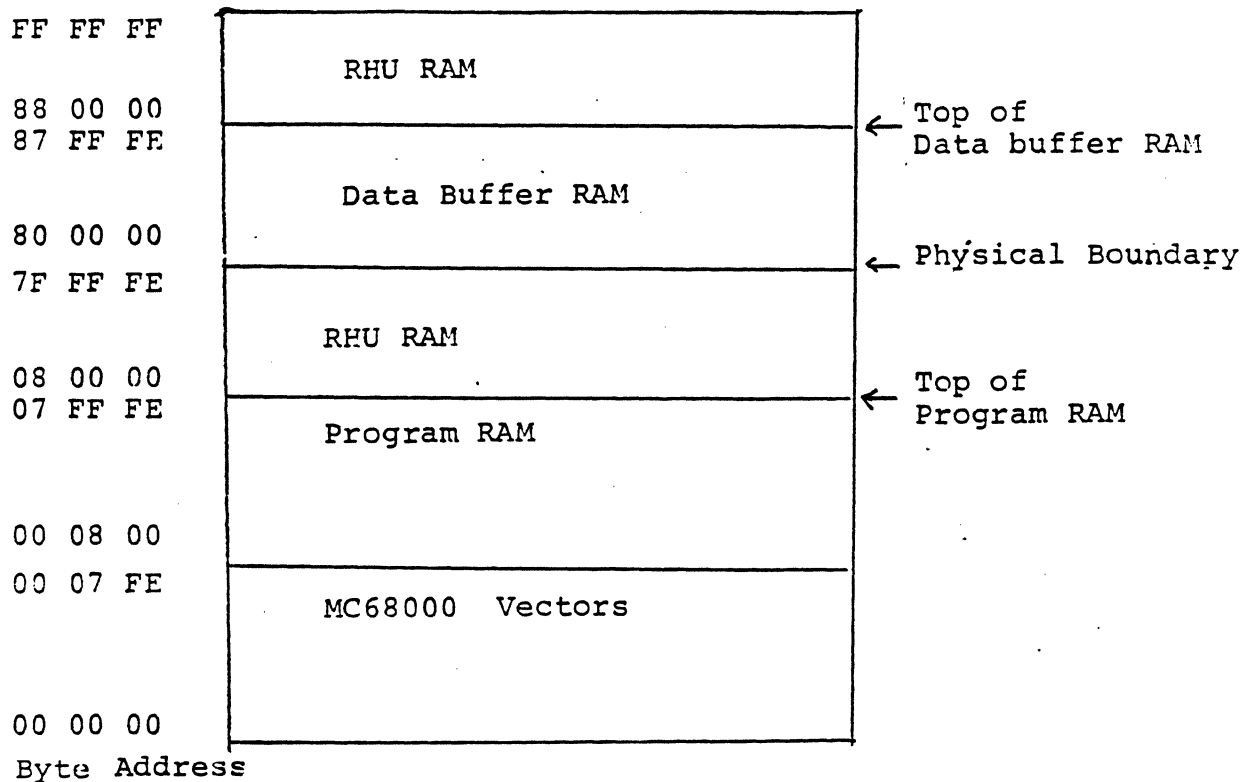


Figure 2-4. LAC RAM MEMORY MAP

The first 1K words contain the MC68000 Exception and Trap vectors. The address of the top of software-useable Program memory may vary depending upon how much Program RAM is physically installed (64K or 256K words). The PROM-resident QLT routines will locate this address and save its value in RAM (see Figure 7-4).

The physical base of the Data Buffer RAM occurs at the location where the Most Significant Bit of the word address becomes a One.

HONEYWELL		SPEC. NO.	SHEET	REV
		60149766	15	H

The address of the top of software usable Data Buffer memory may vary depending upon how much Data Buffer RAM is physically installed (64K or 256K words). This address is ascertained by PROM-resident QLT routines and is saved in RAM (see Figure 7-4).

Software should not perform Test and Set (TAS) instructions on locations in Data Buffer RAM as this may cause a collision with a DMA operation and result in a Bus Error exception. This is a fatal error and its handling is described in Subsection 3.9.

Movement of information to or from main memory may be performed with any LAC RAM locations as an aid to troubleshooting, status/statistic gathering, or other control purposes. However, it should be noted that DMA to or from the Program RAM requires active support from the microprocessor; thus, routines must be executed by the microprocessor in performing such a DMA operation; the Load/Dump Operation and the Memory DMA process provide this function as needed.

If a microprocessor program attempts an access to non-existent program/buffer RAM, a Bus Error exception will occur. This is a Fatal error and its handling is described in Subsection 3.9. If a DMA controller channel or an adapter attempts to access non-existent data buffer RAM, the fact will be indicated in status returned by the IF software on completion of the DMA operation.

The RHU (Reserved for Hardware Use) areas of RAM should not be addressed by CS, SM or O.S. software as they may contain pointers and status required by the hardware and IF software. Thus CS, SM and O.S. software should restrict itself to use of the software-visible areas described in this subsection.

It should be noted that although hardware-specific details are included in this specification, they are not sufficiently complete to permit writing of CS/SM software which does not make use of the HIS-provided IF software support which is described in this specification.

Parity is provided in both LACS RAMs. If a DMA Controller channel or an adapter access to data buffer RAM causes a Parity Error the fact will be indicated in status returned by the IF software on completion of the DMA operation. If a RAM Parity Error Occurs on an Access by the microprocessor the condition is handled as described in Subsection 3.9.

## 2.7 LAC MEMORY MANAGEMENT

In order to utilize the LAC RAM efficiently, memory management functionality is provided through procedure calls to the Operating System (O.S.) O.S. memory management applies only to the software-visible RAM described in subsection 2.6. O.S. memory management uses both a memory block and memory buffer allocation scheme; it maintains a list of free blocks and free buffers of various sizes from which it allocates blocks and buffers on request (ALLOCATE call or GETBUF call). When no longer needed, blocks and buffers may be returned to the free list by another call (MFREE or FREEBUF).

Ownership of buffers may be passed between LAC processes via mailbox messages. Ownership of the block containing a mailbox message is also passed when a process sends the message to another process.

To guard against exceeding the memory space availability of the LAC or of having one user or function monopolize the available space, the CS software in the LAC and the LAC Driver software in the CPU jointly implement a flow control mechanism as described in the LAN software EPS-1.

HONEYWELL		SPEC. NO.	SHEET	REV
		60149766	17	H

## 2.8 LAC SOFTWARE-FIRMWARE-HARDWARE STRUCTURE

### 2.8.1 General

Figure 2-5 represents the structural relationships of the O.S., the CS software, the SM software, the IF software and the hardware.

The Figure reflects the thrust of the functionality described throughout this specification in which CS and SM software does not directly control the LAC hardware but instead interfaces with it through IF Software processes and routines. This IF Software isolates the CS and SM software from the particular characteristics of the hardware so that future reimplementations of hardware (e.g, with larger-scale LSI parts) need not affect that software. All LAC Software is loaded into the LAC RAM.

In this specification IF software is described as consisting of processes or interrupt routines according to whether the particular piece of software is invoked by a mailbox message being sent to it or is normally invoked by the occurrence of an interrupt from the LAC hardware. From the viewpoint of the O.S., these IF "interrupt routines" are either associated with an IF mailbox-invoked process or are a process which consists essentially of only an interrupt agent.

The IF software MEMDMA and IODISP processes have associated with them a Megabus Layer Management Entity (MBLME) to which those processes report various unusual events or faults. MBLME may in turn report certain of these events to SM software; it also serves generally as the intermediary between SM and those processes.

The IF software MAC processes consist of a MAC Transmit, Receive, and Layer Management process for each physically attached adapter. The MAC layer Management process(es) perform functions analogous to those of MBLME.

CS software provides Transport, Network, and Link layer functions for the LAN connection(s). Each of these layers and layer instances has a layer management entity associated with it which performs functions analogous to MBLME.

SM software provides overall control and system status reporting for the LACS layer management entities and with System Management software in the CPU.

O.S. Kernel software provides service functions such as timers and controls the dispatching of processes and passing of mailbox messages. The handling of error responses from the Kernel for the various procedure calls sent to it by CS and IF software is given in Subsection 2.19.

The LAC also contains some PROM-resident firmware which provides for QLTs, RAM load/dump and basic I/O orders. Figure 2-5 does not show the PROM-resident firmware.



HONEYWELL		SPEC. NO.	SHEET	REV
		60149766	18	H

### 2.8.2 Interprocess Communications

Mailbox messages, sent via the SENDMSG Procedure Call, are the means whereby one process may send a message or request service of another process. They are also the means whereby the occurrence of asynchronous events or the completion of asynchronous services are made visible to the software so that software processing can proceed to its next step.

The called process will retrieve messages sent to its mailboxes via the RECEIVE or BRECEIVE Procedure call.

Software processes may obtain the ID of their own mailboxes via the MYID macro; they may obtain the ID of another process' well-known registered mailbox via the RESOLVE procedure call.

### 2.8.3 Relative Process Priorities (Normal Running)

Figure 2-5 generically names the CS and SM Software processes, and gives the names of the IF software processes. IF software also includes associated interrupt and exception-handling routines.

During normal running after start-up, processes associated with data Receive operations must have higher priority than those associated with data Transmit operations so as to minimize loss of messages and consequent retransmissions.

Among these Receive processes, the MAC Receive process(es) must be the highest priority so that as Receive buffers are used by the hardware adapter(s) they may be quickly replaced by the MAC Receive Process(es) so that other messages may be accepted. Simulations have shown that the remaining data Receive processes (Link, Network, and Transport) work well at an equal priority among themselves. The MEMDMA IF software process seems to give best performance at the same priority as these three Receive processes.

Among the data Transmit processes, simulations have shown that the MAC Transmit process(es) should be highest followed in decreasing order by LLC Transmit, Network Transmit and Transport Transmit.

In summary, the data handling process priorities in decreasing order of priority should be:

1. MAC Receive
2. LLC Receive, Network Receive, Transport Receive, MEMDMA
3. MAC Transmit
4. LLC Transmit
5. Network Transmit
6. Transport Transmit

For the remaining LAC processes, IODISP, the Layer Management processes, and System Management, a fairly high priority seems reasonable.

There may be a different priority relationship among the processes during the initial running of the CS and IF processes. As the processes complete their initial run, they may set themselves to their normal running priority via PROPRIORITY procedure calls.

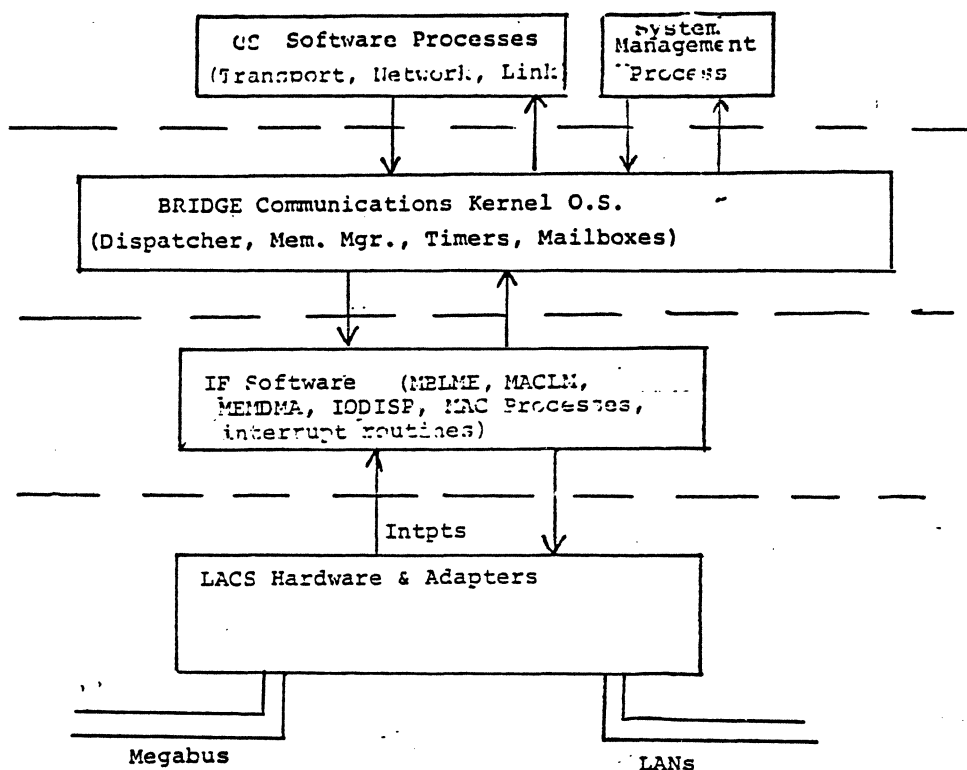


Figure 2-5. LAC OPERATING STRUCTURE

HONEYWELL		SPEC. NO.	SHEET	REV
		60149766	20	H

## 2.9 MAILBOXES AND MAILBOX MESSAGE QUEUES

Mailbox messages are the means of inter-process communication.

In order that RAM blocks used for mailbox messages may be reused freely, a common block size should generally be used for mailbox message blocks.

Queues of mailbox messages may be built up on any mailbox using OS facilities. The process which owns the mailbox picks up each message for processing; the act of picking up a message deletes it from the queue. Each message consists of an OS-defined header and a message body. The format and content of messages sent between CS or SM software, and IF Software is given in Section 3.

### 2.9.1 Mailboxes Associated with IF Software

During the running of the Kernel's "init" process at startup (Subsection 2.17), mailboxes are set up for IF software processes MBLME, IODISP, and MEMDMA, in the well-known mailbox list. The string name of these mailboxes is the same as the process name.

The SM and CS processes will obtain the mailbox ID of MBLME, IODISP, and MEMDMA, processes by means of RESOLVE procedure calls; they will send mailbox messages to IODISP to provide mailbox IDs for IODISP's Dispatch table.

HONEYWELL		SPEC. NO.	SHEET	REV
		60149766	21	H

## 2.9.2 Mailbox Message Priorities

The Bridge O.S. provides a number of priorities for mailbox messages which affect the relative position of messages in a mailbox queue. The available message priorities are URGENT, NORMAL, MUST DELIVER, and FAST.

Simulations have shown that CS processes should use an URGENT Priority for receive-related requests to the Memory DMA process whereas they should use a NORMAL priority for transmit-related Memory DMA requests.

Requests for control operations on adapters should probably use URGENT priority; data transmit requests addressed to MAC processes may all use NORMAL priority since the process will queue data messages by Service Class.

## 2.10 LAN CONTROL BLOCK

The LAN Control Block (LCB) is the prime vehicle of intercommunication between the Level 6 CPU and the LACS. One particular form of LCB is used in connection with the PROM-resident firmware-supported Load/Dump operations as shown in Figure 3-1; other forms are used for communication between software in the CPU and CS software in the LAC and these are described in the LAN Software EPS-1. LCBs are constructed in main memory by the LAC Driver software and are interpreted and executed in the LAC.

## 2.11 LAC SOFTWARE INTERFACE TO THE MEGABUS

The CS/SM software interface with the Megabus is supported through mailbox messages received from the IF software I/O Dispatch process and through mailbox messages sent to the IF Software Memory DMA process. The mailbox messages received consist essentially of pointers to LCBs (subsection 2.9) in main memory. The mailbox messages to the Memory DMA process are used to cause movement of data between main memory and the LAC Data Buffer RAM, or to read in LCBs, or to write status type information into LCBs in memory and interrupt the CPU.

## 2.12 CS SOFTWARE INTERFACE WITH ADAPTERS

The CS/SM software interface with adapters is supported through mailbox messages generated by the IF Software MAC processes (i.e. Data Indicate and Control Indicate) and through mailbox messages sent to IF Software MAC processes.

HONEYWELL		SPEC. NO.	SHEET	REV
		60149766	22	H

### 2.13 SOFTWARE INTERFACE BETWEEN LEVEL 6 AND LACS

The software interface between Level 6 and the LACS during normal running uses IOLD orders addressed to the LACS and return status information delivered to main memory by the LAC accompanied by interrupts to the Level 6.

All of the data message and Administrative and Management operations are based on the use of LAN Control Blocks (LCBs) located in main memory and which are pointed to by information given in IOLD orders. The appropriate software process in the LAC will cause the LCB to be copied into the RAM as an LCB Image (LCBI), and after completing the requested operation, will cause final status to be delivered to the LCB. In carrying out the operation, the process will make use of various other processes.

Details of the software interface are given in the LAN software EPS-1.

As an aid to a general understanding of the intended operation of the LAC, the next two subsections describe the transmission and reception of a data message. Control functions related to setup of data transmissions and to Systems Management are then briefly mentioned. The functionality of IF Software operations and processes is described in Section 3.

#### 2.13.1 Transmit Data Operation Flow

Figure 2-6 is a representation of the anticipated flow for a transmit data operation.

For purposes of this description, the starting point is assumed to be the IORB of the MOD 400 PIO software interface with higher level system software having effectively isolated the application from any LAN-specific characteristics.

In the following description, CS software is treated as if it consists of a single process in order to simplify the narrative.

CS software may provide support of Link, Network and Transport layer functionality. The IEEE 802 Standard includes only up to Link Layer definition; in IEEE 802 terms the operation described here is that which involves an L\_Data.request or L\_Data\_CONNECT.request primitive.

In Step 1, the LACS Driver software in the CPU sets up the LCB in memory from information in the IORB. The LCB will contain information defining the processing and function required and parameters; it also contains physical address(es) and range(s) defining the buffer(s) in memory which contain the data to be sent. The LCB also includes space for return status from the LAC.

In Step 2, the LACS Driver issues an IOLD to the LACS. The address given with the order points to the LCB and the "Range" parameter contains two fields: the high order 8 bits are a Function Code field and the low order 8 bits define the size of the LCB. The IOLD information is taken off the Megabus and placed in a temporary queue by the LAC hardware DMA controller. This causes an interrupt which invokes the I/O Dispatch process (IODISP) which inspects the order; having determined that the IOLD is valid, it uses the channel number in the order to reference a Dispatch table and determine where to route the order for further processing. In this case, the routine obtains a block of RAM (via an ALLOCATE call), places the LCB Pointer IOLD information in the block, and sends it (via a SENDMSG call) to a CS process mailbox. The format of the LCB Pointer IOLD information message block is shown in Section 3. If there are additional I/O orders in the queue, the I/O Dispatch process will handle these also. All message blocks obtained by the I/O Dispatch process must be returned to free memory by some other process (e.g. in Step 12).

In Step 3, a CS process is scheduled for execution by the O.S. (because of the mailbox message addressed to it); the process retrieves the mailbox message and, after securing a block of RAM for an LCB image (LCBI) sends a message to the mailbox of the Memory DMA Request process requesting DMA of the LCB into this LCBI. The CS process may then suspend itself if it has nothing else to do for the moment. The format of the message block sent to the Memory DMA process is shown in Section 3.

In Step 4, the Memory DMA Request process causes the DMA controller to copy the LCB into the LCBI. On completion of the operation, the controller interrupts the microprocessor and this causes the Memory DMA process to be re-invoked. This process places status information, shown in Section 3, in the message block which was sent by the CS process and then returns the block (via a SENDMSG call) to the specified return mailbox. Information originally placed in the RSU field of the block by the CS process in Step 3 allows it to identify the particular DMA operation which has been completed.

HONEYWELL		SPEC. NO.	SHEET	REV
		60149766	24	H

In Step 5, the CS process responds to the mailbox message of Step 4. After inspecting the LCBI and calculating the total of the L6 buffer ranges, it performs a GETBUF call to obtain a RAM buffer big enough to hold the data message; then it sends a mailbox message to the Memory DMA process to cause the movement of data from main memory to this buffer in RAM. The format of the message block is shown in Figure 3-4; the L6 buffer list is obtained from the LCBI and the LEVEL field should be zero.

In Step 6, the Memory DMA process causes the DMA controller to copy the data from main memory to the RAM buffer. The process will support a "gather" type DMA with respect to main memory, if required; with respect to the LAC RAM, DMA is always done on a logically single buffer. On completion of the DMA, the Memory DMA process is reinvoked and places status in the message block and returns it to the specified return mailbox (of the CS process).

In Step 7, the CS process responds to the mailbox message of Step 6. It sends a mailbox message to the Memory DMA process to cause it to set status complete in the LCB in memory and to interrupt the CPU. At some later time the LACS driver will post the completion into the IORB. If a message is to be sent over an IEEE 802 LAN, CS processes must create header fields and add this as prefix information to the RAM buffer. CS processes must also have left additional space at the beginning of the buffer for the MAC process to prefix its headers. A CS LLC process assembles a mailbox message and sends it to the appropriate MAC process. The format of the mailbox message block is shown in Figure 3-9. This constitutes the MA DATA.request of the IEEE 802 Standard.

In Step 8, the MAC Transmit process may queue the request if there are higher priority requests to be handled. As soon as it can, the process delivers the request to the adapter. The adapter completes prefixing of the message frame (SA and FC), and when media access rules permit, delivers a correctly formatted frame (including preamble, delimiters and FCS) to the LAN via the adapter's PHYS layer facilities. When transmission is complete, the adapter's DMA controller sends an interrupt to the microprocessor of the LAC.

In Step 9, the Adapter Interrupt routine invokes the MAC transmit Process which fetches final status from the adapter. The MAC Transmit process releases the RAM buffer (FREEBUF call). If there are other transmit requests pending the process will deliver one to the adapter.

HONEYWELL		SPEC. NO.	SHEET	REV
		60149766	25	H

Figure 2-6 shows the transmit flow just described. It should be noted that although the description and diagram present a single thread of flow for clarity, there are actually multiple threads being processed at various stages at any instant of time. Since each software process is written to try to complete all of its outstanding tasks, if possible, before voluntarily relinquishing the microprocessor, the number of context switches performed per message transmitted will tend to be less under typical load than when considering only a single message thread.



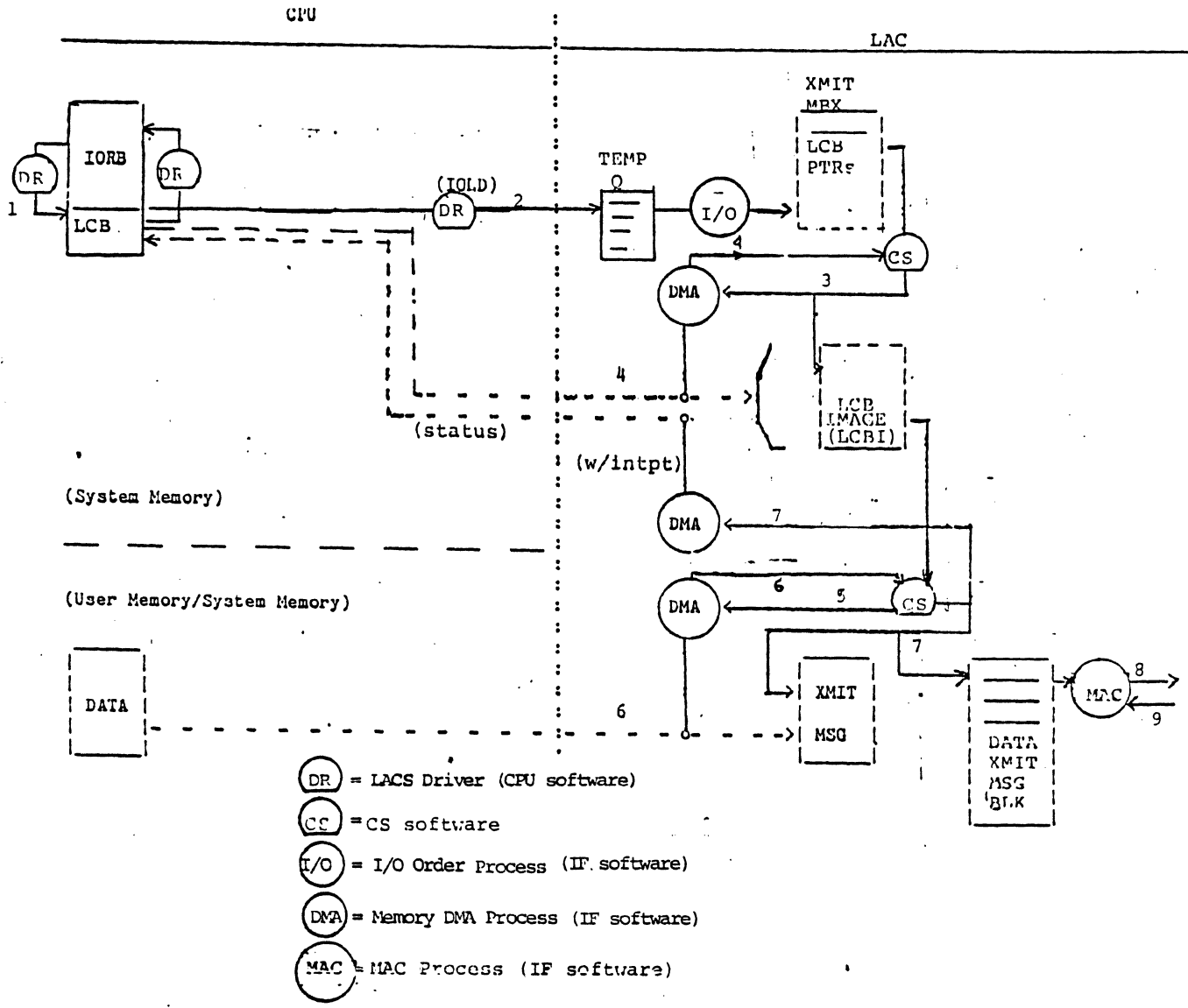


Figure 2-6 LACS TRANSMIT FLOW

HONEYWELL		SPEC. NO.	SHEET	REV
		60149766	27	H

### 2.13.2 Receive Data Operation Flow

The operation described here is that which involves an IEEE 802.2 L\_DATA.indication or L\_DATA-CONNECT.indication primitive of type 1 or type 2 operation.

For handling received messages, one of two schemes can be used depending on whether the application wishes to allocate a buffer only if and when a message is received from the LAN or whether it wishes to allocate a buffer in anticipation of a possible incoming message. In the first or Read-Notify case, two IOLDs must be issued and two interrupts must be sent to the CPU for each message and the time required in the CPU to obtain the buffer may cause performance problems. In the second case, main memory space requirements tend to be greater because of the buffers which are tied up waiting for a message.

The description of receive flow will not be given in as much detail as in the transmit case since the interactions of CS software processes, IF software processes, hardware interrupts, and interrupt firmware are similar.

For receive operation it is not necessary for CS software to request data buffers from memory management, as is the case for transmit operation. Instead, the IF Software MAC processes will automatically make available enough buffers for each adapter. After a valid message is received, the Data Indicate routine of the MAC process will pass the buffer to the proper CS process.

In the Read-Notify case, shown in Figure 2-7, CPU software issues a series of LCBs, which are called "Read-Notify" LCBs, to the LAC via Output LCB Pointer IOLD orders. These serve to provide LCBs which the CS software may use for notifying CPU software of the arrival of messages. When the arrival of a message has been indicated by this means, the CPU software will issue a Read LCB to specify where, in main memory, the message is to be placed and will also, in general, issue another Read-Notify LCB to replace the one which was used. This scheme allows the data to be input directly to the application's buffer. Read LCBs are differentiated from Read-Notify LCBs by some software-defined indication in the LCB itself.

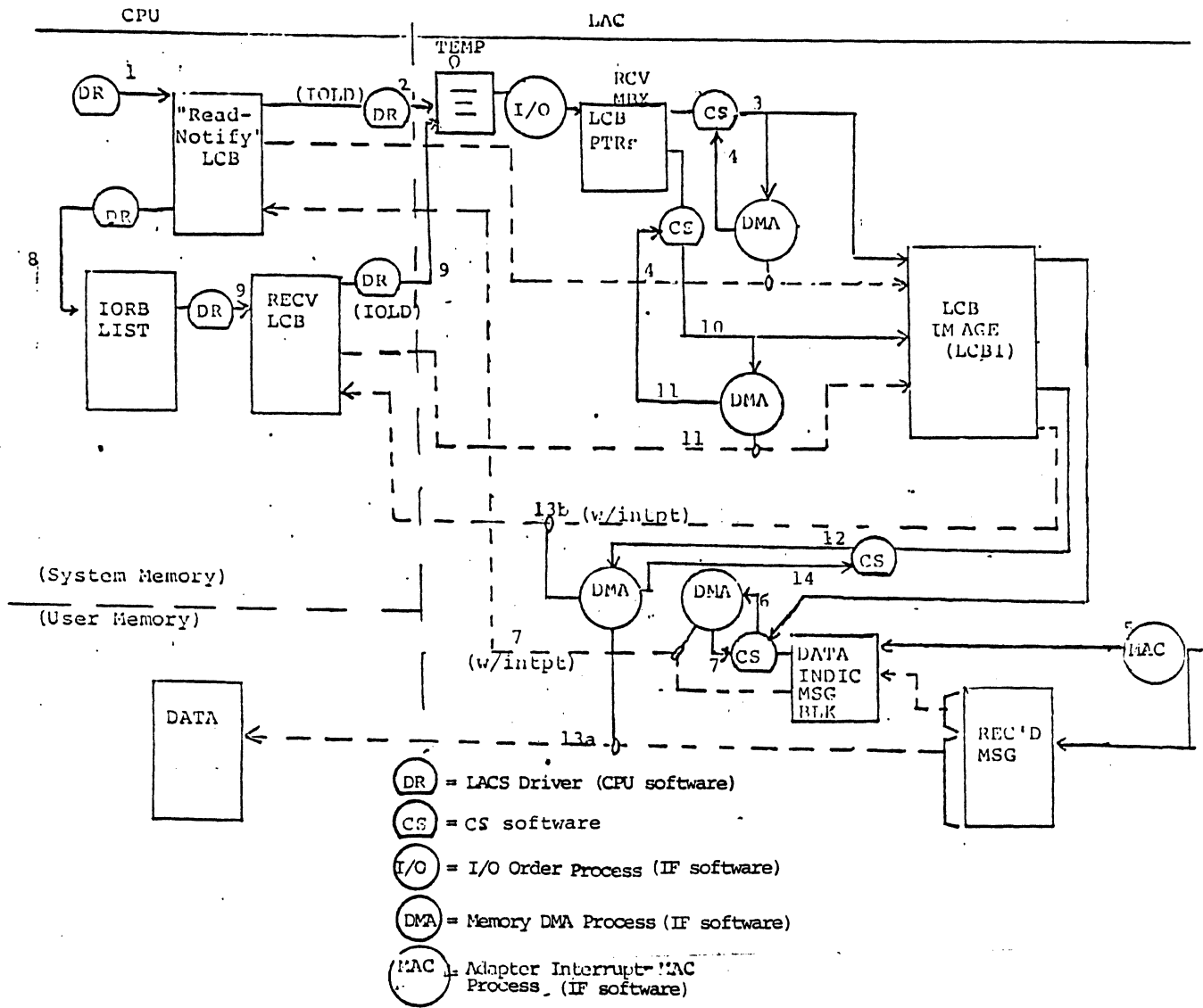


Figure 2-7 LACS RECEIVE FLOW (READ-NOTIFY CASE)

HONEYWELL		SPEC. NO.	SHEET	REV
		60149766	29	H

The sequence of operations is as follows:

In Step 1, the LACS Driver software in the CPU sets up the Read-Notify LCB in memory.

In Step 2, the LACS Driver issues an IOLD to the LACS pointing to the LCB. The DMA Controller in the LAC takes the IOLD information from the Megabus and places it in a temporary working queue and interrupts the microprocessor. In a manner similar to the transmit case, the I/O Dispatch process delivers the Read-Notify LCB Pointer IOLD information to a CS process.

In Step 3, the CS process issues a request for the Memory DMA process to copy the LCB into an LCBI in RAM, similar to the transmit case.

In Step 4, the DMA Controller copies the LCB into the LCBI and the CS process is notified of completion.

In Figure 2-7, the arrival of a message is labeled Step 5 for convenience. This should not be construed to imply that a message only arrives after the operations of Steps 1 through 4; message arrival is an independent occurrence.

In Step 5, a message arrives and is placed in the predefined buffer by DMA facilities of the adapter. On completion of this operation, the adapter interrupts the microprocessor, causing the Adapter Interrupt routine to invoke the MAC Receive process which performs an ALLOCATE call, to obtain a block of RAM for a mailbox message, places Data Indicate information in the block and sends it to the mailbox of the appropriate CS process. The ID of this mailbox has been previously made known to the MAC process. The format of the Data Indicate message block is shown in Section 3. The MAC Receive process provides the adapter DMA facility with replacement buffer(s) by performing GETBUF call(s).

HONEYWELL		SPEC. NO.	SHEET	REV
		60149766	30	H

In Step 6, a CS process consults its list of Read-Notify LCBs to see if there is one which pertains to the particular message just received. If there is none, the message is retained in RAM (however if some reasonable time passes without an appropriate LCB the process may be forced to discard the message). In the usual case, a CS process assembles information from the message header to be delivered to the LCB in memory, assembles a mailbox message block, and sends it to the Memory DMA process requesting DMA of this information into the Read-Notify LCB. In the message block, the CPU Chan and Intpt LEVEL fields reflect information given in the original IOLD and LCB as does the Chan No. (Also included is an identifier which the LACS Driver in the CPU can place in the Read LCB when it issues it).

In Step 7, the DMA controller delivers the information to the Read-Notify LCB and interrupts the microprocessor, causing it to re-invoke the Memory DMA process. This process now sends the requested interrupt to the CPU and when this has been accomplished returns the message block of Step 6 to the return mailbox (the CS process).

In Step 8, CPU software responds to the interrupt and, by consulting a list of outstanding IORBs or by other means, determines where in main memory the data message should be placed. The LACS Driver then sets up a Read LCB in memory. This LCB will contain the identifier of Step 6 (so that the CS process in the LAC can identify which data message is to be delivered) and specifies the main memory area(s) into which it is to be placed.

In Step 9, the LACS Driver issues an IOLD to the LACS pointing to the LCB. In the usual manner, the IF Software delivers the LCB Pointer information to the CS process.

In Step 10, the CS process issues a request for the Memory DMA process to copy the LCB into an LCBI in RAM.

In Step 11, the Memory DMA process copies the LCB into the LCBI and the CS process is notified of completion.

In Step 12, the CS process inspects the LCBI and determines that a Read operation is involved. The process calculates the total size of the L6 buffer and calculates a Range Residue value for LCB status and places final status in the LCBI; it then issues a request to the Memory DMA process to move the data message from RAM to main memory and to deliver final status from the LCBI to the LCB and to interrupt the CPU.

In Step 13, the DMA Controller copies the data from buffer RAM to main memory, performing a "scatter" DMA if required, under control of the DMA process. On successful completion of the data transfer the DMA process performs a Block transfer which copies the LCBI status into the LCB and interrupts the CPU. On completion of this the Memory DMA process return the mailbox message block to the return mailbox (the CS process).

In Step 14, the CS process may release the data buffer, the LCBI block and the mailbox message block.

Although the description and Figure present a single thread of flow for clarity, there are actually multiple threads being processed at various stages at any instant of time. Because each software process is written so as to try to complete all its outstanding tasks before relinquishing the microprocessor, the number of context switches performed per message received will be less under typical load than when considering only a single message thread.

In the Read LCB case, not shown in a Figure, the CPU issues IOLDS, which point to Read LCBs, each Read LCB includes pointer(s) to buffer(s) in system memory large enough to hold the largest possible message. Only one interrupt need be sent to the CPU, i.e, after the data and final status have been delivered.

### 2.13.3 Control Operations (during normal running)

Control operations may be divided into two general categories: functions which are involved with setting up, controlling flow, or terminating a connection to a remote user and System Management functions which are related to setting up, maintaining, and monitoring operation of the LAN itself.

HONEYWELL		SPEC. NO.	SHEET	REV
		60149766	32	H

The first category of functions is visible to the user or to higher-level software directly supporting the user. At the link-level interface, these functions consist of the L\_CONNECT, L\_DISCONNECT, L\_CONNECT\_FLOWCONTROL, and L\_RESET primitives. Although these functions do not involve any data transfer with higher level entities, they are handled essentially the same as in data transfer operations, using primitive codes which distinguish them from data operations. These functions generally involve link-layer messages between stations (e.g. SABM, DISC, etc.).

The second category of functions are not visible to a user or to higher-level software in support of the user. These encompass various management and administrative operations. These operations use an interface with the Level 6 which is based on the use of LCBs. The System Management functions are given in the LAN Software EPS-1). System Management performs the following operations:

- o Systems Management Data Service Interface (SMDSI) to a remote station via a LAN.
- o Layer Management Interface (LMI) to Layer Management Entities (LMEs) in the LACS.

2.14 IF SOFTWARE FUNCTIONAL DIAGRAMS

Figures 2-8 through 2-11 are a set of flow charts whose purpose is to indicate in more detail the functional responsibilities of the various IF software processes and interrupt routines. The charts are not complete in all details and are not intended to necessarily represent how the detailed implementation of a process must be done. The flow of the CS and CM processes is given in the software EPS.

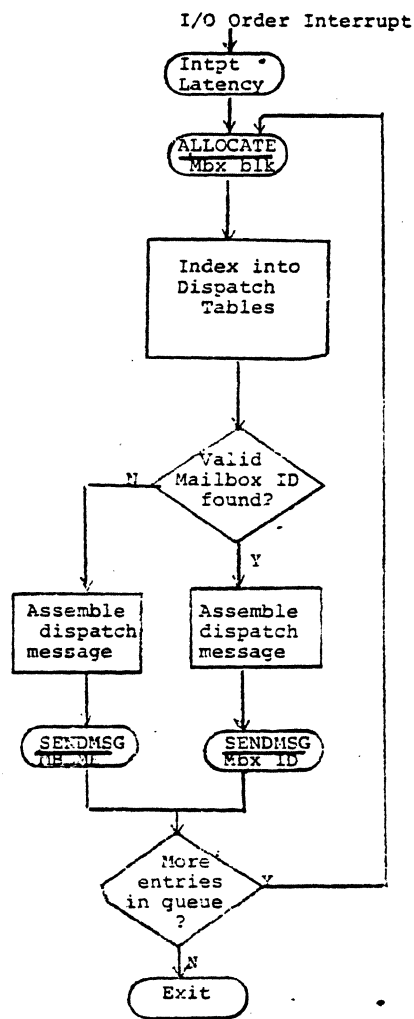
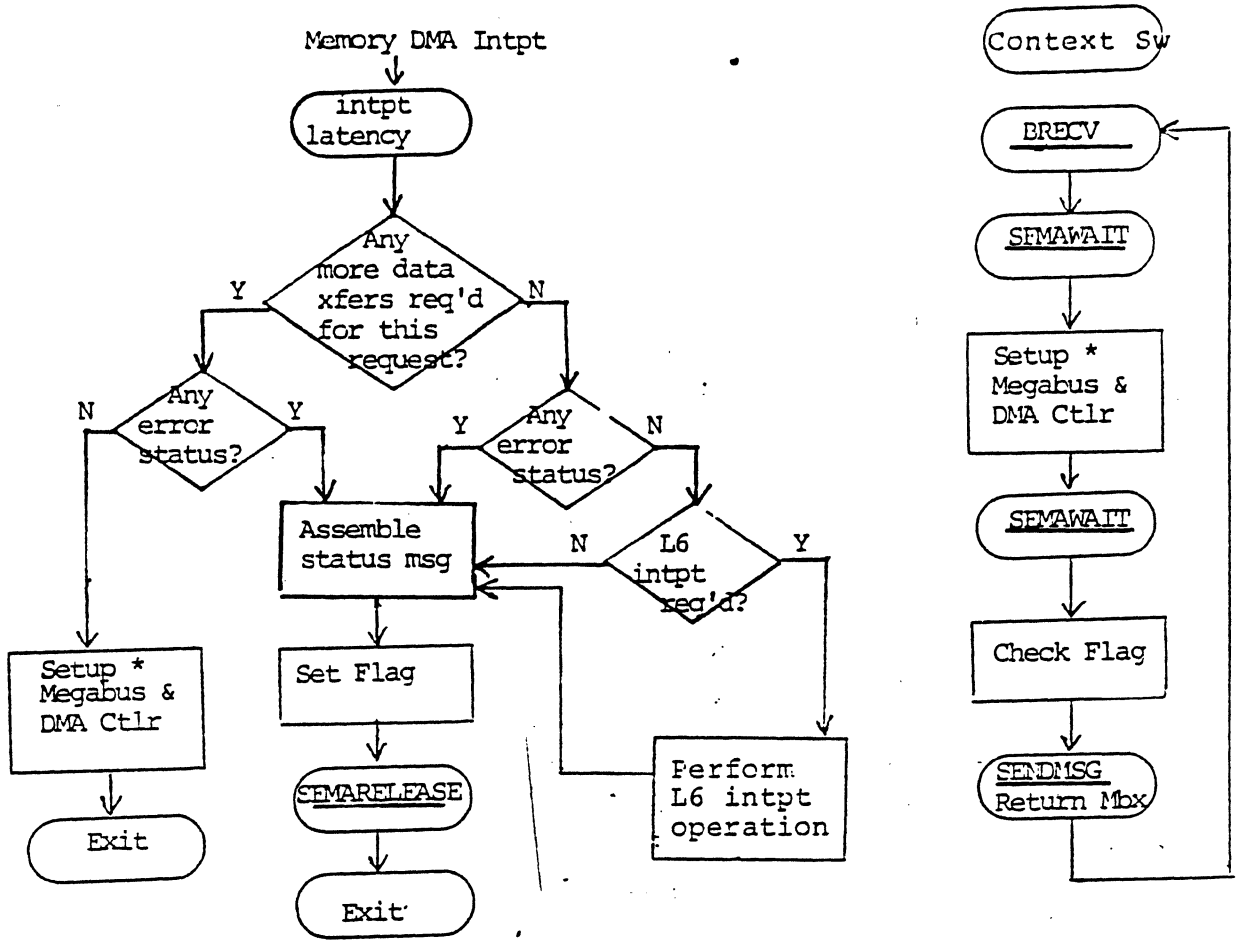


Figure 2-8 I/O DISPATCH PROCESS (IF SOFTWARE)





\* If the DMA request includes both a Buffer Descriptor and a RAM Ptr, all the buffer xfers are done before the RAM block xfer.

Figure 2-9 MEMORY DMA PROCESS (IF SOFTWARE)

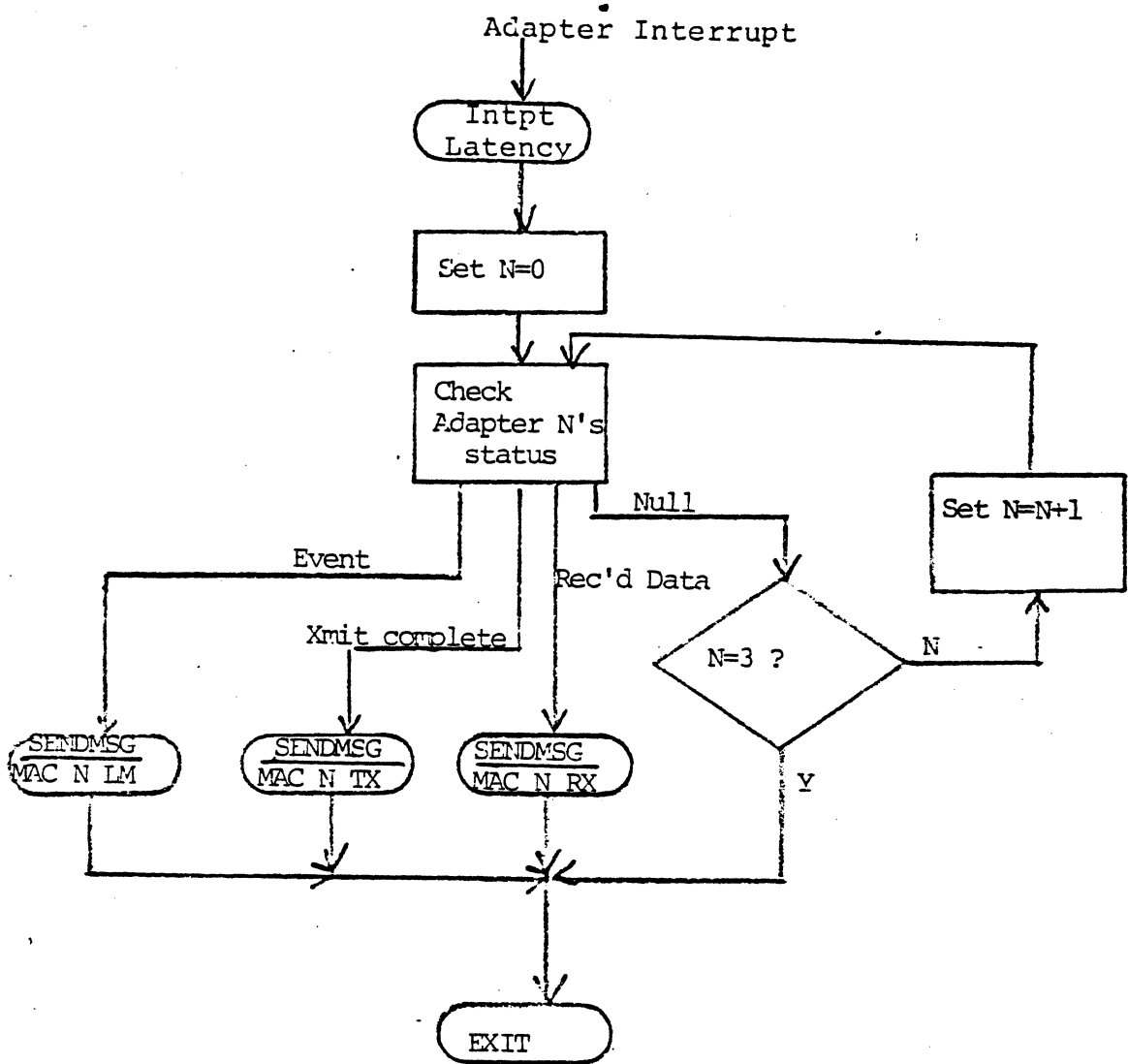


Figure 2-10 ADAPTER INTERRUPT ROUTINE (IF SOFTWARE)

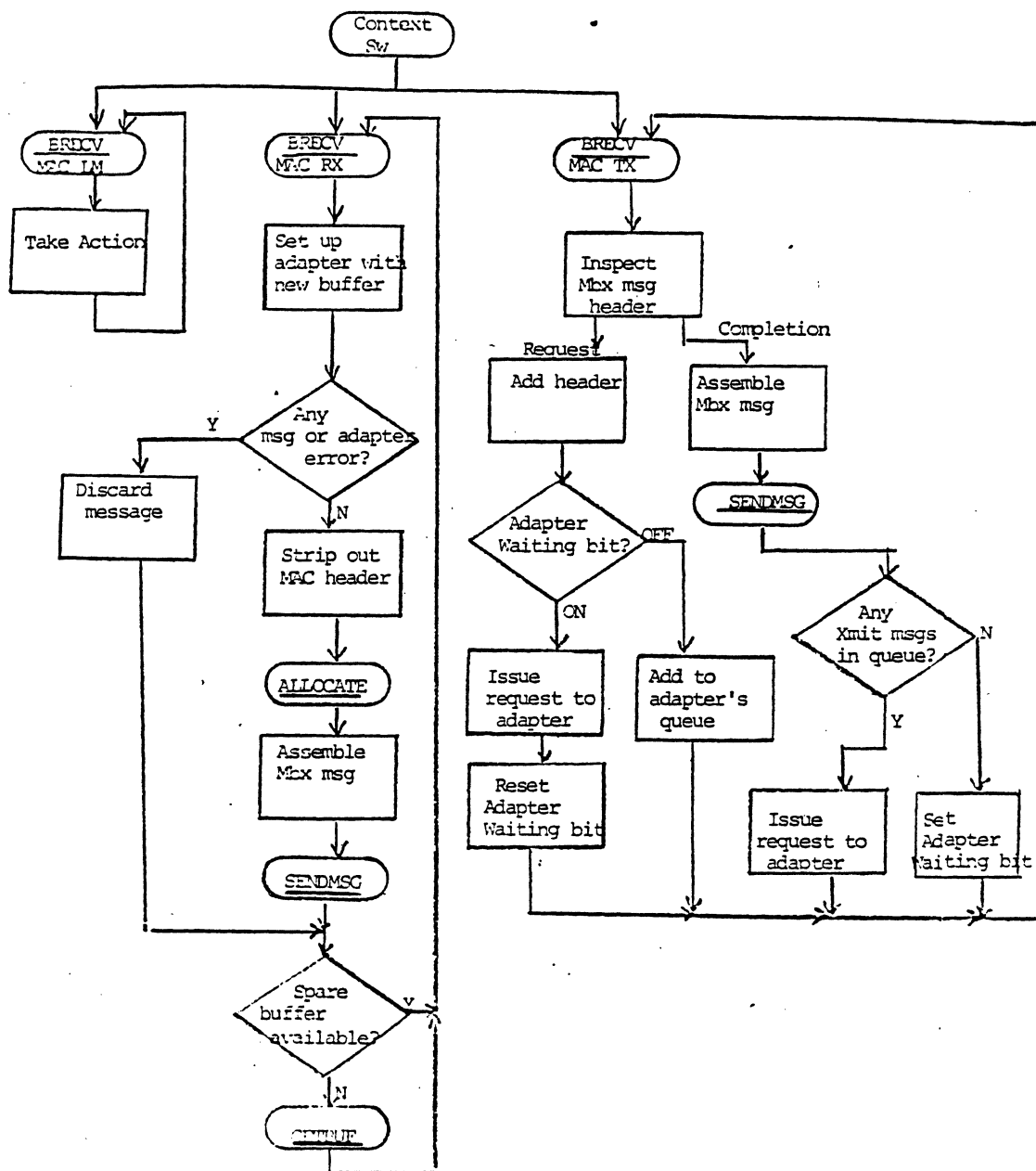


Figure 2-11 ADAPTER-SPECIFIC MAC PROCESSES (IF SOFTWARE)

HONEYWELL		SPEC. NO. 60149766	SHEET 37	REV H
-----------	--	-----------------------	-------------	----------

## 2.15 INTERRUPTS TO LEVEL 6

The Memory DMA process will send an interrupt to a CPU on successful completion of a DMA operation if the mailbox message which initiated the DMA operation so specifies. The mailbox message specifies the CPU number the interrupt level, and the LAC channel number to use; this information is obtained by CS software from the IOLD and LCB.

If the interrupt is rejected by the CPU, the Memory DMA process will put the interrupt in a Deferred Interrupt queue.

When a CPU indicates that it may be able to accept an interrupt (by a Resume Interrupt signal on the Megabus) an IF software interrupt routine, Interrupt Retry, will attempt to send the highest priority deferred interrupt of each CPU. Any interrupts that are not accepted are returned to the Queue.

Because of the way the Queue is operated, when an interrupt is accepted by a CPU, it may be an indication of completion of more than one IOLD-LCB. CPU software then must search through its list of outstanding LCBS; it can determine which have been completed by the completion status placed in them. Subsection 3.8.1.1 gives more detail on the Deferred Interrupt queue.

The CS software and the LAC Driver implement a completion sequence numbering scheme as detailed in the software EPS-1.

## 2.16 STATISTICS

The following physical connection statistics are maintained by the MAC Process on a per-physical-connection (i.e. LAN) basis and may be retrieved by System Management via its LMI interface with the MAC Layer Management Entity:

TX-FRM	:	Number of frames sent correctly
PCEDA	:	Number of Octets sent
RCV_FRM	:	Number of frames received correctly
PCRDA	:	Number of Octets received
PCRG	:	Number of group-addressed PDUs received correctly
PCRER	:	(Calculated) total number of frames received incorrectly
UNR_ERR	:	Number of underruns

A subset of the following physical connection statistics is maintained on a per-physical-connection (i.e. LAN) basis. The particular subset is a function of the adapter type. The statistics may be retrieved by System Management via its LMI interface with the MAC Layer Management Entity.

o OVR_ERR	:	Number of overruns
o FCS_ERR	:	Number of FCS errors
o PCRAB	:	Number of received abort sequences
o ALN_ERR	:	Number of misaligned packets
o BUF_ERR	:	Number of packets lost due to no buffer
o PCRLN	:	Number of packets lost due to length error
o PCEBY	:	Number of transmit deferrals
o PCECO	:	Number of transmit aborts due to collision
o PCRCL	:	Number of collisions spoiling reception
o PCNTR	:	Number of token pass stations
o PCNTP	:	Number of stations observed to pass token
o PCNPR	:	Number of token pass repeats
o PCNJT	:	Number of times use-token state entered
o TOK_ROT	:	Measured token rotation time
o NS	:	Next station/successor address
o PS	:	Previous/predecessor address
o NMB_STN	:	Number of stations in ring
o PCEER	:	Number of frames aborted due to error
o RTRY_ERR	:	Number of frames sent on second retry
o RTRY_RWR	:	Number of retries on request-with-response frame

2.17 LACS STARTUP

After the LACS has been powered up and initialized, it operates under control of firmware resident in the PROM. This firmware provides the QLTs which verify the basic operability of the LACS and supports IO and IOLD orders which permit the LACS to be identified, the LAC RAM to be loaded or dumped and LACS operation under software control to be started.

The CPU software should issue one or more Load IOLDS (Subsection 3.2.5.2) to cause the LAC to load the LAC software from main memory. On completion of the Load IOLD(s), CPU software should then issue a Start I/O (subsection 3.2.5.3) and then should not issue further IOLDS to the LAC until the Start I/O IOLD has been marked as completed (any IOLDS which are issued during this time will either be NAK'd or will have their LCBs marked with Invalid Function Status).

The Start I/O IOLD information is placed in a working area in RAM by the PROM-resident firmware. This firmware then relinquishes control of the microprocessor to LAC software in the Program RAM with software operation starting at the location specified in the Start I/O LCB.

The LAC microprocessor now commences execution of LAC software, specifically the Bridge kernel O.S. "main" routine. This leads to the creation of an O.S. initial process called "init" which creates and then registers a well-known mailbox for, each of the processes named in the init table (i.e. MEMDMA, IODISP, MBLME, SYS\_mgr). The init process then allows each of these processes to run in sequence according to their initial relative priorities.

As each process runs its initial routine it may establish mailbox interconnections with other processes and may spawn various layer instance processes as described in the LAN software EPS and LAN software component specifications. System Management sends each of the layer management processes a mailbox ID which will be retained in reserve by each process for possible need in certain trouble situations requiring an Event Notification as described in subsection 2.19. The IF Software processes, as part of their initial running, may equalize their priorities as mentioned in Subsection 2.8.3

HONEYWELL		SPEC. NO.	SHEET	REV
		60149766	40	H

2.17 Continued

The IODISP process will also set up its Dispatch table from information it finds in its mailbox and then dispatches the Start I/O IOLD to the SM process via the Event Notification mailbox.

The SM process responds to the Start I/O mailbox message by requesting the MEMDMA process to copy the Start I/O LCB into RAM. When this has been done, MEMDMA so notifies SM (by returning the request mailbox message). The SM process then sends another DMA request to MEMDMA to cause completion status to be placed in the Start I/O LCB and an interrupt to be sent to the CPU. On completion of this the request mailbox message is returned to the Return mailbox.

The CPU software may now proceed to issue more IOLDS; some of these first IOLDS will undoubtedly be for the purpose of setting up some current parameters for SM's use.

Note that the Start I/O applies to the LAC as a whole. Startup or shutdown of individual LAN communications is entirely a software function.

HONEYWELL		SPEC. NO.	SHEET	REV
		60149766	41	H

A Stop I/O order is implemented in the LAC; this initializes the LAC and its adapters and returns control to the PROM-resident firmware but does not clear the LAC RAM.

Note that when the LAC is running under software control (after a Start I/O IOLD), any loading or dumping of RAM must be accomplished under CS or SM software control (i.e. not using the Load/Dump IOLD of subsection 3.2.5.2)

## 2.18 TIMERS

Interval timers are provided via O.S. facilities. The timer tick interval is 50 milliseconds. The O.S. also provides means for performing timing functions shorter than one tick interval.

## 2.19 HANDLING OF SOFTWARE ERROR CONDITIONS

This subsection is concerned with the action to be taken by IF and CS software on the occurrence of an error response by the O.S. to a Procedure Call.

The scheme for handling those conditions is as follows:

- a. When an error response occurs, a SETALARM timer call is to be made for a period of (TBD) milliseconds if the type of error is due to a likely temporary condition (i.e. it is a recoverable error).
- b. After the time-out, if applicable, the original call should be retried.
- c. If, the error cannot be cleared by retry, it is classed as a Reportable Catastrophic error and the error is to be reported as an Event to the associated layer management entity using the mailbox Message format given in Figure 3-11 and Event Code given in Table 2-1.
- d. If the error cannot be reported it is an Unreportable Catastrophic error and, it is fatal. In this case, a trap is taken to a special routine which will set the Megabus logic to a state where it will not accept any I/O orders (except an Output LACS Control IO Order) and will mask interrupts. It is expected that the CPU will then eventually issue a Stop I/O order and dump the LAC RAM.



HONEYWELL		SPEC. NO.	SHEET	REV
		60149766	42	H

e. If the Event has been successfully reported the Notifying process may possibly continue to retry.

The list of software events in Table 2-1 is exhaustive; the particular subset which may arise during a process' execution depends upon the particular subset of O.S. Procedure calls made by that process. In each case the subset is defined in the description of the process and the particular action taken by its associated layer management entity is defined in the description of that entity; this information is given in Section 3 of this specification and in the LAN software EPS and software component specifications.

Procedure Call	Event Code	Event Description
PRDCREATE	1000	procreate kernel call reported an error */
PRORUN_1	1101	prorun error, process to be run does not exist */
PRORUN_2	1102	prorun error, process is waiting, can not run */
PROPRORITY_1	1201	priority call error, invalid process control block */
PROPRORITY_2	1202	priority call error, shared stack restriction violation */
PROPRORITY_3	1203	priority call error, invalid priority set */
MBOXCREATE	2000	mailbox create call error, no mail boxes left */
MBOXDELETE_1	2101	mailbox delete call error, no such mailbox exists */
MBOXDELETE_2	2102	mailbox delete call error, requestor is not owner */
SENDMSG_1	2201	send message call error, invalid mailbox used */
SENDMSG_2	2202	send message call error, invalid buffer descriptor used */
SENDMSG_3	2203	send message call error, invalid message priority used */
SENDMSG_4	2204	send message call error, mailbox is full already */
MBOXON_1	2301	turn mailbox on call error, invalid mail box used */
MBOXON_2	2302	turn mailbox on call error, requestor is not owner */
MBOXOFF_1	2401	turn mailbox off call error, invalid mailbox used */
MBOXOFF_2	2402	turn mailbox off call error, requestor is not owner */
BRCEIVE	2600	wait to receive message call err, invalid operation from shared stack process
NOTIFYFULL_1	2701	notifyfull call error, invalid mailbox used */
NOTIFYFULL_2	2702	notifyfull call error, invalid buffer descriptor used */
NOTIFYFULL_3	2703	notifyfull call error, mailbox is not full */
STOPFULL_1	2801	stopfull call error, invalid mailbox used */
STOPFULL_2	2803	stopfull call error, last message not found from notifyfull call */
TESTMBOX_1	2901	testmbox call error, invalid mailbox used */
TESTMBOX_2	2902	testmbox call error, requestor is not owner */
REGMBOX_1	2A01	register a mailbox call error, invalid mail box used */
REGMBOX_2	2A02	register a mailbox call error, no room in registration table */
REGMBOX_3	2A03	register a mailbox call error, mailbox name is too long */
RESOLVE	2B01	resolve mailbox name call error, can't find in well known name list */
SEMCREATE	3001	semcreate call error, no more semaphores available */
SEMWAIT_1	3101	semwait call error, specified semaphore does not exist */
SEMWAIT_2	3102	semwait call error, illegal if called by a shared stack process */
SEMWAIT_3	3103	semwait call error, invalid semaphore identification */
SEMARELEASE_1	3201	semarelease call error, specified semaphore does not exist */
SEMARELEASE_2	3202	semarelease call error, invalid semaphore identification */
ALLOCCATE	4000	allocate memory call error, no more free memory available */
GETBUF	4100	getbuf memory call error, no more buffers or buffer descriptors available */
JOINBUF	4201	joinbuf error call, invalid buffer descriptor supplied */
PREPENDBUF	4300	prepend buffer call error, no buffer descriptor available */
APPENDBUF	4401	appendbuf call error, bad parameters or no buffer descriptor available */
PADBUF	4501	padbuf call error, bad parameters or no buffer descriptors available */
COYBUF	4600	copybuf call error, no buffers or buffer descriptors available */
UNAPPENDBUF	4801	unappendbuf call error, bad parameters supplied, can't do it */
BUFINFO	4901	bufinfo call error, bad parameters supplied, can't do it */
SETALARM_1	5001	setalarm call error, invalid timeout interval */
SETALARM_2	5002	setalarm call error, invalid buffer descriptor in message */
STOPALARM	5103	stopalarm call error, can't find desired alarm message */
UNPREPEND	4700	unprepend call error

TABLE 2-1 SOFTWARE EVENT NOTIFICATION CODES

HONEYWELL		SPEC. NO.	SHEET	REV
		60149766	44	H

### SECTION 3 FUNCTIONAL DESCRIPTION

#### 3.1 BASIC FUNCTIONS

This section describes the orders, instructions, mailbox communications, IF Software, and LAC firmware functionality which support operation of the LACS.

Much of the hardware specifics of the LACS is hidden from the CS and System Management software by IF Software routines and processes. In this Section, subsections 3.2 through 3.7 address the basic built-in functionality of the LAC while subsection 3.8 addresses the functionality provided by these IF software routines and processes. Sections 4, 6, 7, and the appendices provide additional detail of the basic hardware characteristics.

The standard LAN software product in the LAC will include Link layer and System Management functions in accordance with IEEE 802.1 and 802.2 Standards.

### 3.2 I/O ORDERS

The following I/O Orders are provided for control of the LACS by the Level 6 CPU.

#### 3.2.1 List of I/O Orders

##### OUTPUT ORDERS

- |         |            |                     |
|---------|------------|---------------------|
| 1. IO   | (FC=01)    | Output LACS Control |
| 2. IOLD | (FC=09/0D) | Output LCB Pointer  |

##### INPUT ORDERS

- |       |         |                 |
|-------|---------|-----------------|
| 1. IO | (FC=26) | Input Device ID |
|-------|---------|-----------------|

For details concerning the structure of the I/O order and other bus-related information, see the referenced Megabus EPS's. The conditions for the above orders to receive ACK, NAK, or WAIT response are given in Subsection 3.2.4.

#### 3.2.2 Output Order Definitions

Only the following Output I/O Orders are defined. All other Output I/O order Function Codes (whether I/O or IOLD) are considered undefined.

Undefined I/O orders are ignored by the LAC; no response is sent to the CPU and the order is not taken by the LAC.

- Output LACS Control IO (FC=01) - This order transfers a 16-bit control word to the LACS. All adapters and interfaces are affected by this order. The channel number used in the order is immaterial. The bits in the word are defined as follows:

- o Bit 0: Hard Initialize (if a one)
- o Bits 1: Stop I/O (if a one and bit 0 is a zero)
- o Bits 2-15: MBZ

Hard Initialize causes the actions described in Subsection 3.2.5.1. It requires several seconds to accomplish and is intended to be used only at startup or by T&V software.

Stop I/O performs the actions described in Subsection 3.2.5.3.

- Output LCB Pointer IOLD (FC=09/OD) - This order involves two separate bus transfers to the LAC: The first transfer is a 32-bit byte address and the second is a 16-bit "Range" word of which the high order eight bits are interpreted as described below and the low order eight bits define the LCB Size in bytes. The address and LCB Size define the location and size of an LCB in main memory. The LCB must begin on a word boundary; LCB Size must also be word-bound.

Information from the bus is placed in a temporary queue by LAC hardware and an interrupt is then sent to the LAC microprocessor. Ensuing action depends upon the value of the Protocol ID byte and whether the LAC is operating under control of its PROM-resident firmware) or running under control of LAC software after a Start I/O IOLD.

If the LAC is under firmware control the following interpretation of the high order byte of the "Range" word is observed by the PROM-resident firmware:

(FF)16=Load/Dump (Subsection 3.2.5.2)  
 (FE)16=Start I/O (Subsection 3.2.5.3)  
 (OO)16to(FD)16=Invalid

An invalid code in this byte will cause the firmware to set Invalid Function and Status complete status in the LCB (see Figure 3-1 for the general format of LCBs) and send an interrupt to the CPU in accordance with the contents of the LCB.

If the LAC is running under software control the following interpretation of the high order byte of the "Range" word is observed by the IODISP process:

0	3	4	7
Function Code		MBZ	

The Function Code is used by the IODISP Process (Subsection 3.8.1.3) in dispatching the order to the appropriate software process.

It should be noted that it is possible that in a multiprocessor CPU configuration, two (or more) CPUs might attempt to send an IOLD (i.e. Function Code 09/OD) to the LAC simultaneously; the LAC will detect this and reject (NAK) an IOLD which tries to intervene on the bus between the two bus transfers of an IOLD which has already performed its first transfer.

HONEYWELL		SPEC. NO.	SHEET	REV
		60149766	47	H

### 3.2.3 Input Order Definition

Immediately following completion of execution of an Output LACS Control order (FC=01) the following Input I/O Order is provided (all other Input I/O orders are treated as undefined); this condition remains in effect until an Output LCB Pointer IOLD (FC=09/0D) is issued. Thereafter all input I/O orders are treated as undefined.

Input Device ID (FC=26) - This order causes the LAC PROM firmware to deliver a 16-bit Device ID word to the Megabus. This ID reflects both the LAC and the adapter attached to the addressed adapter channel (subsection 3.3). The ID codes are given in subsection 3.4 and in the appendices.

Undefined Input I/O orders are ignored by the LAC (i.e. no response is made to the order and no information is sent).

### 3.2.4 WAIT, NAK, and ACK Responses

Normally, the LAC responds to I/O Orders with an ACK response. It will never respond with a WAIT. It will respond with a NAK in the following cases:

1. An order, other than an Output LACS Control (FC=01) is issued while the LACS is performing all of the operations commanded by an Output LACS control order (FC=01).
2. An IOLD order intervenes on the bus with one already underway (subsection 3.2.2).
3. The LAC is unable to accept another order because its temporary queue is full. This condition should not occur if the LAC Driver observes the flow control mechanism described in the LAC Software EPS-1. If the condition occurs, the order should be retried later.
4. A Fatal error condition exists (see subsection 3.9).
5. An IOLD order is issued while a Start I/O operation is still being carried out (subsection 2.17).

As noted in subsections 3.2.2 and 3.2.3, no response is made to an undefined I/O order.

HONEYWELL		SPEC. NO.	SHEET	REV
		60149766	48	H

### 3.2.5 Initialization, Start/Stop, and Load/Dump

#### 3.2.5.1 Initialization

LACS Hard Initialize. This function is initiated by a Power-On sequence or by the Output LACS Control Order (FC=01) with bit 0 a one. It causes the following actions:

- a. The LAC and adapter RAMs are cleared
- b. Hardware registers in the LAC and adapters are cleared.
- c. The LAC runs its QLT
- d. The LAC enters a stopped condition in which its functionality consists of those functions supported by PROM-resident firmware as described in Subsection 3.2

Note that the QLT routine ascertains the configuration information shown in Figure 7-4. If the motherboard QLT does not run successfully to completion the IOLD and Input I/O orders of subsections 3.2.2 and 3.2.3 will be NAK'd. A Stop I/O order or a depression of the CP clear button on the CPU control panel may be used to put the LAC in a condition to accept a Dump order so that QLT status retrieval can be attempted.

### 3.2.5.2 Load/Dump

Load/Dump operations provide means for loading of LAC Software, from main memory, for dumping various portions of the LAC RAM to main memory, and for retrieving certain configuration information from the LAC. The LAC must be running under firmware control for a Load/Dump to be performed. The operation is commanded via an Output LCB Pointer IOLD in which the high order byte of the "Range" word is (FF) 16. The detailed information needed to perform the operation is contained in the LCB as shown in Figure 3-1. The LAC channel number used for the IOLD is used in the interrupt sent on completion but is otherwise immaterial to the LAC. On completion, the Controller Status word in the LCB is updated and the Status Complete bit is set and an interrupt is sent to the CPU as specified in the LCB.

If a Megabus Parity, Memory Red or Non-existent Memory fault occurs on the retrieval of the LCB from memory, the order will not be performed and appropriate status will be set in the configuration area in RAM (Figure 7-4).

### 3.2.5.3 Start I/O

The Start I/O order is defined in Subsection 3.2.2 and the format of the LCB used with this order is shown in Figure 3-2. The conditions under which this order should be used are described in that Subsection and in Subsection 2.17.

The PROM-resident firmware leaves the order in the temporary queue and relinquishes control of the microprocessor to LAC software commencing execution at this Program RAM location which is defined in the LCB.

The fetching of the Start I/O LCB and the storing of final status and sending an interrupt to the CPU is a responsibility of SM software and is described in subsection 2.17. The Start I/O order should be addressed to channel 0 of the LAC; Channel 0 is the address of the System Management process which is the process which handles start-up of the LACS software system.

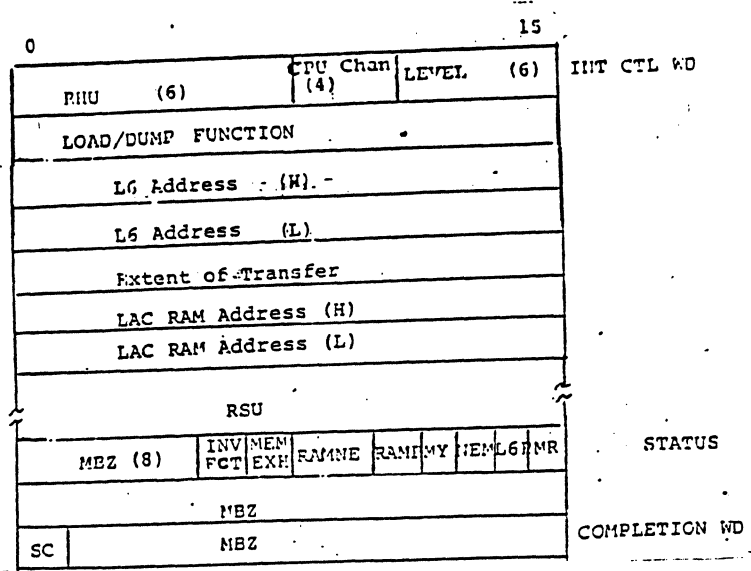
### 3.2.5.4 Stop I/O

The Stop I/O order is defined in Subsection 3.2.2. A Stop I/O is also performed as a result of depressing the CP Clear button on the CPU control panel. A Stop I/O causes the following actions:

- a. Hardware registers in the LAC and adapters are cleared.
- b. The LAC commences (or continues) operation under firmware control in which its functionality consists of those functions supported by PROM-resident firmware as described in Subsection 3.2.

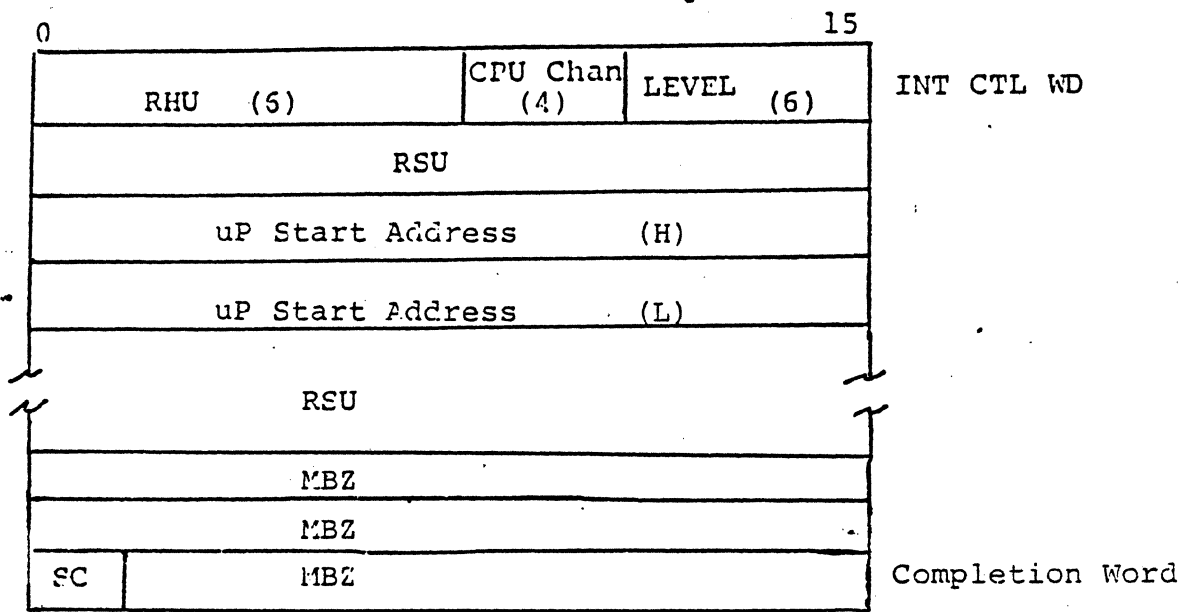


Note that when a Stop I/O is performed, any software-controlled operations currently underway are abruptly terminated, outstanding LCBs are not completed and Pending interrupts are not sent. After a Stop I/O, the LACS should be reloaded and restarted as for an initial start-up (See Subsection 2.10) if further operation under software control is desired.



- Notes:
- In the LCB as set up by the CPU, the Status and Completion words must be all zero. These will be updated by the LAC on completion of the operation.
  - The L6 Address, LAC RAM Address, and Extent are all byte-oriented but all must be word-bound (i.e. with LSB of zero).
  - Status Bits
    - RAMNE = RAM location non-existent
    - RAMP = RAM parity error
    - MY = Memory Yellow
    - NEM = Non-existent L6 memory
    - L6B = L6 bus parity error
    - MR = Memory Red
    - SC = Status Complete
    - MEM EXH = Excessive Number of IOLDs issued
    - INV FCT = Invalid Function/Request
  - LOAD/DUMP FUNCTION
    - (0000) = Store LAC RAM to memory
    - (0001) = Load LAC RAM from memory
    - (0002) = Read Configuration information into memory
  - For the Read Configuration operation the LAC RAM address is ignored. The maximum range should be 96 bytes.

Figure 3-1 LCB FORMAT FOR LOAD/DUMP



Notes:

1. In the LCB as set up by the CPU, the SC bit must be zero. The LAC will set the bit on completion of the Start operation.
2. SC= Status Complete
3. The uP start address is a byte-oriented address but must be word-bound (i.e. with LSb of zero).

FIGURE 3-2 LCB FORMAT FOR START I/O

### 3.3 CHANNEL NUMBERS

The LAC is assigned a set of 64 channel numbers, using the 6 LS bits of the 10-bit channel address on the bus.

For the Input Device ID order (FC=26), the 6 LS bits of the channel address are treated by the LAC as consisting of two fields: the highest two bits specify the adapter's daughterboard position and the lowest four bits specify a subchannel associated with the adapter (to allow for future adapters which may have subchannels). The channel number coding for the Input Device ID order is shown below.

0	3 4	5 6	9
LAC	ADAPT	SUBCH	

LAC = LAC Board Address (Set by DIP switch on board)  
 ADAPT = Adapter Position (0-3)  
 SUBCH = Subchannel on Adapter

For Load/Dump IOLDs and the Output LACS Control IO, the LS 6 bits of the channel address are immaterial to the LAC (however the given channel address will be used in any interrupt which is sent on completion).

### 3.4 DEVICE IDENTIFICATION NUMBER

The device identification number for the LACS is shown below.

0	7	8	9	13	14	15
(2E)	QERR	ADAPTER	SUBCHAN	TYPE	TYPE	

The codes for Adapter Type and Subchannel Type are given in the appropriate adapter specification (see appendices). If the adapter's QLT has not completed without error the QERR bit will be a One. If no adapter is present at this address, Adapter Type will be all zeros. If a Subchannel is not present or the adapter does not have Subchannels, Subchannel Type will be all zeros. The meaning of non-zero Subchannel Types is specific to the adapter type.

### 3.5 DATA TRANSFER WITHIN THE LAC

Multiple data transfers can occur in the LAC concurrently; the adapters and the Megabus interface are half-duplex but DMA transfers with the LAC data buffer RAM can be concurrently under way on all of these facilities.

All data messages are fully buffered in the LAC data buffer RAM. The adapters perform serial-parallel conversions, CRC generation/checking, framing, address recognition, and media access.

Since a message can be received by an adapter at any time and since adapters do not provide message buffering, there must be at least one receive DMA buffer set up for each adapter at all times to avoid loss of a message; it is the responsibility of the IF Software to do this.

The data traffic of the (up to) four adapters and of the Megabus interface must be time-multiplexed over the LAC's internal DMA Bus to the Data Buffer RAM; in addition the microprocessor makes accesses to this RAM via the DMA Bus. These six sources/sinks thus contend for bus access. A fixed priority scheme is used in the LAC to resolve conflicts: the adapter in the channel 0 position is highest in priority, followed by adapters 1 to 3 in order, followed by the microprocessor, followed by the Megabus. The priority is not Preemptive, i.e. it applies only when the current source/sink master releases the DMA bus. While the DMA bus can perform a word transfer in about 550 nanoseconds, the overhead for switching control of the bus from one source/sink to another is about 400 nanoseconds. To obtain maximum throughput over the bus, the time cost for switching is amortized over several effective transfer cycles, by having the sources/sinks perform their transfers in bursts wherever possible. The Megabus interface and 10Mbps adapters use bursts; adapters for speeds under 5Mbps need not operate with bursts. In order to share the bus equitably, all sources/sinks except the Megabus will limit the length of their bursts to a predefined value.

### 3.6 INSTRUCTIONS

The instruction set for the MC68000 microprocessor is given in the MC68000 reference document.

### 3.7 PROM RESIDENT FIRMWARE

Resident firmware in the PROM provides the following functions via microprocessor routines:

- o Support of the applicable I/O order functionality of subsection 3.2.
- o Quality Logic Test routines.
- o Determination of LAC configuration information and storage of it in LAC RAM. Format and location is given in Section 7.
- o Handling of Unreportable Catastrophic errors.

HONEYWELL		SPEC. NO.	SHEET	REV
		60149766	54	H

### 3.8 IF SOFTWARE

IF Software is provided in the form of a set of processes and interrupt routines which support the CS software processes by providing an implementation-independent interface to LACS hardware. These IF Software processes are invoked directly through calls to mailboxes. IF Software interrupt routines are invoked by the occurrence of an interrupt from LACS hardware; several of these are associated with IF software processes. The IF software is able to make various Procedure Calls to the O.S. for services and to call CS software processes; IF software is part of the standard LAN product offering with the LACS.

The IF software processes are invoked via a SENDMSG call to the O.S, specifying the mailbox of the process. Messages (i.e. requests for service) are queued in each mailbox ;the process normally retrieves each request as it is able to handle it via a RECEIVE or BRECEIVE call to the O.S. Further details on mailboxes may be found in the reference Bridge documents.

These processes require parameters each time they are invoked; this is provided in the mailbox message which invokes them. Usually these processes will return a message to the calling process when the operation is complete.

#### 3.8.1 IF SOFTWARE PROCESSES

The IF Software processes are as follows:

- o Memory DMA Request (MEMDMA)
- o Megabus Layer Management (MBLME)
- o I/O Order Dispatch (IODISP)
- o MAC Transmit
- o MAC Receive
- o MAC Layer Management

##### 3.8.1.1 Memory DMA Request (MEMDMA)

This process carries out DMA operations to move information between a buffer and/or block in LAC RAM and one or more buffers (scatter/gather) in Level 6 memory; it carries out the operation via the Megabus interface using the hardware DMA controller and Megabus logic registers.

The format of the mailbox messages which are used in DMA requests to call the Memory DMA process are shown in Figures 3-3 to 3-6. All addresses and ranges shown in these Figures are byte-oriented.

The process mailbox provides the mechanism for queuing requests to the process for DMA operations.

During the handling of DMA operations involving scatter or gather functions the process will be reinvoked via the associated Megabus DMA Register Exhaust interrupt routine as each transfer with a main memory buffer completes and will reload the Megabus hardware with information for the next buffer transfer operation. Although a RAM buffer is logically a single buffer, physically it may consist of many segments. DMA Controller interrupt routines which are part of this process take care of reloading the DMA Controller with these segments. If the mailbox request was not formatted correctly or an error occurs on the Megabus interface (L6 bus parity, Memory Red, Memory Yellow, Non-existent memory) or in accessing the RAM (RAM parity, non-existent RAM), the operation is not completed, any requested interrupt to the CPU is not sent and the fault is indicated in status in the return message sent to the Return Mailbox specified in the request.

If the operation is completed without error, an interrupt will be sent to the CPU if the request parameters so specified (non-zero LEVEL); it is expected that an interrupt will be specified only for a DMA operation which delivers status information to an LCB in memory. On final completion of the total operation (including interrupt) a return message will be sent to the requesting process.

It is possible that an interrupt to a CPU will not be accepted (NAK) because of other operations being performed in the CPU. The process maintains a queues of such Deferred Interrupts (those which have been NAK'd).

The DMA process organizes deferred interrupts into queues on a per-CPU basis. Within a queue pending interrupts are listed first in order of priority (LEVEL) with highest priority at the top; within those of the same LEVEL the ordering is FIFO. The queue members are the actual mailbox message blocks which asked for an interrupt.

HONEYWELL		SPEC. NO.	SHEET	REV
		60149766	56	H

When a DMA request calls for a L6 interrupt and the associated DMA has been successfully completed, the DMA process will first inspect the pending interrupt queue for that CPU (the CPU No. field in the message block is used to select which queue i.e. the Queue No.). If there are any pending interrupts in that queue of equal or higher priority, the interrupt will not be attempted but will be added to the queue; if it is higher than any entry, it will be attempted and if accepted the message block is returned to the requesting process but if rejected it will be added to the queue. If there is a queue member having the identical interrupt specification in the queue, the interrupt (i.e. the message block) will be linked "horizontally" to the one(s) which are like it. Space is available in the DMA request mailbox message header so that the DMA process can link the blocks vertically and horizontally.

When some CPU causes a RINT (Resume Interrupt), the Interrupt Retry routine (Subsection 3.8.2.1) will retry the top entry of each queue as an interrupt; if accepted, the block is removed from the queue and returned to the process which made the request. Note that when there are a number of deferred interrupts with the same interrupt specification, only one interrupt is actually sent even though all of the message blocks will have been retained by the DMA process until that interrupt has been accepted.

In a multiprocessor environment, it may happen that a CPU for which LACS has some deferred interrupts queued will fail and that central system software will then assign some other CPU to service LACS interrupts in place of the failed CPU. Assuming that the SM in the LACS is notified of this, a mechanism is provided whereby the DMA process Interrupt Retry routine can be told (by Megabus Layer Management) to route pending interrupts originally intended for the failed CPU to the new CPU. The DMA process will have a table which relates queue number to CPU number and which is used, during interrupt retry only, to specify the CPU number to use on the interrupt (the CPU number in the queued message block is ignored on retry). This table is initially set up by the DMA process to be one-to-one but any entry in it can be accessed by sending a specific mailbox message to the DMA process. The format of this message is shown in Figure 3-7.

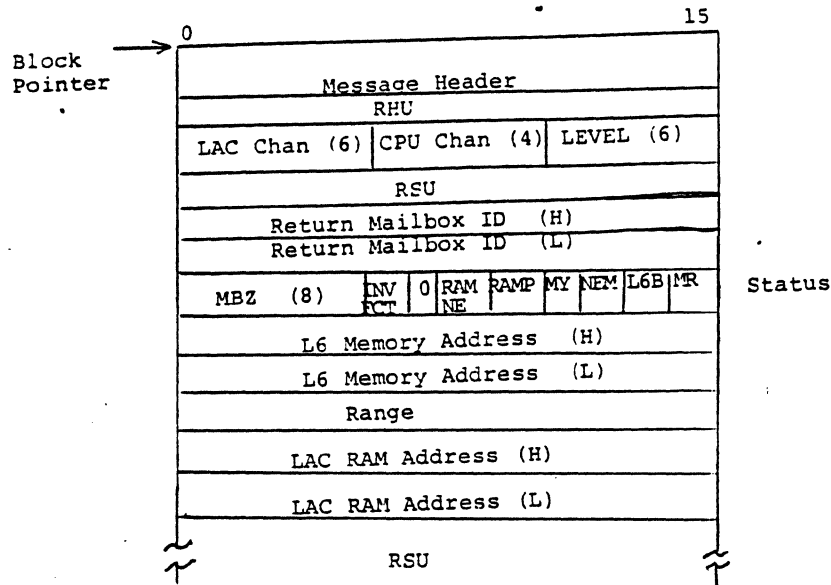
The DMA process may encounter an O.S. error in performing one of the following O.S. Procedure calls:

- o SENDMSG
- o BRECEIVE
- o RESOLVE
- o SEMACREATE
- o SEMAWAIT
- o SEMARELEASE

If such an error occurs, a message in the format shown in Figure 3-11 is sent to System Management using the appropriate Event code from Table 2-1. The MEMDMA process, during initial running, obtains the mailbox ID of SM via a RESOLVE call.

The normal running flow of this process is indicated in Figure 2-9; this figure does not include the initial execution of the process during which the DMA interrupt routines (subsections 3.8.2.3 and 3.8.2.4) are registered the semaphore interlock is set up etc. Additional details may be found in the MEMDMA Component Specification.

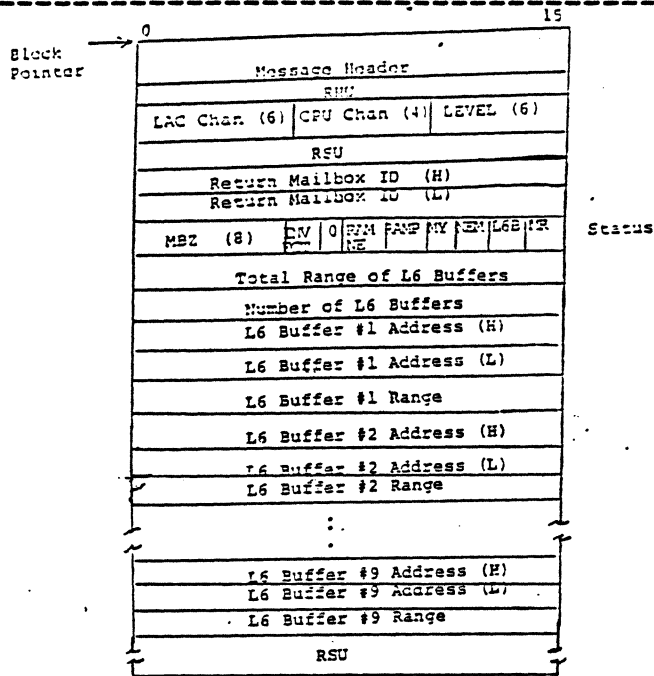




Notes:

- This message is used for requesting a DMA operation to move a block of data between RAM and L6 memory and for confirming completion of the operation.
- A valid Return Mailbox ID must be provided in the request message and the status word must be all zeros.
- In the request message the Message Type for the Header is as follows:  
  
LCB-TO-LAC Transfer from L6 memory to RAM  
LCB-TO-L6 Transfer from RAM to L6 Memory
- If the LEVEL field is non-zero an interrupt will normally be sent to the specified CPU on completion of the operation; however, if an error occurs during the DMA operation, the interrupt will not be sent.
- In the confirm message the Message Type for the Header is the same as in the request message except that the suffix CF is added.
- See Figure 3-1 for definition of Status bits.
- The Buffer Descriptor in the Header is NULL.

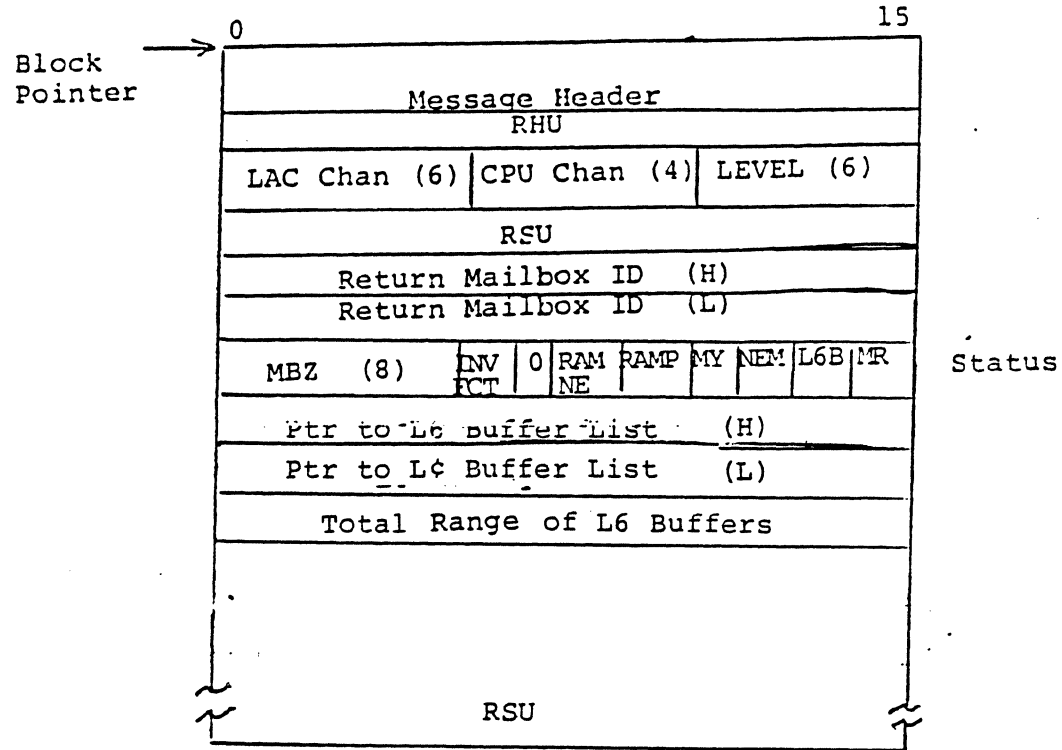
Figure 3-3 DMA Request/Confirm Message for Block Transfer



Notes:

1. This message is used for requesting a DMA operation to move data between a logical buffer in RAM and one in L6 memory and for confirming completion of the operation.
2. A valid Return Mailbox ID must be provided in the request message and the Status word must be all zeros.
3. In the request message the status word should be all zero and the Message Type in the Header is as follows:  
  
 BUF-TO-LAC Transfer from L6 Memory to RAM  
 BUF-TO-L6 Transfer from RAM to L6 Memory
4. If the LEVEL field is non-zero, an interrupt will normally be sent to the specified CPU on completion of the operation; however, if an error occurs during the DMA operation, the interrupt will not be sent.
5. In the confirm message the Message Type for the Header is the same as the request message except that the suffix CF is added.
6. See Figure 3-1 for Status bit definition.
7. The Header contains the RAM Buffer Descriptor.

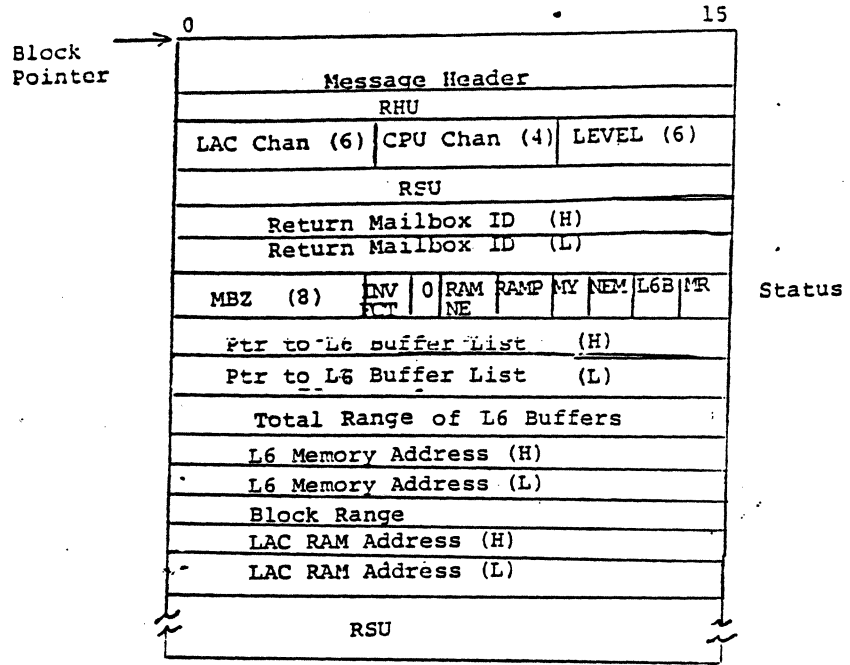
Figure 3-4 DMA Request/Confirm Message for Buffer Transfer



Notes:

- This message is used for requesting an extended DMA operation to move data between a logical buffer in RAM and one in L6 memory and for confirming completion of the operation.
- A valid Return Mailbox ID must be provided in the request message and the Status word must be all zeros.
- In the request message the Message Type for the Header is as follows:  
 BUFX-TO-LAC - Transfer from L6 Memory to RAM  
 BUFX-TO-L6 - Transfer from RAM to L6 Memory
- The Header contains the RAM Buffer Descriptor
- The format of the L6 Buffer list is shown in figure 3-8.
- In the confirm message the Message Type for the Header is the same as in the request message except that the suffix CF is added.
- See Figure 3-1 for Status bit Definition.
- If the LEVEL field is non-zero an interrupt will normally be sent to the specified CPU on completion of the operation; however, if an error occurs during the DMA operation, the interrupt will not be sent.

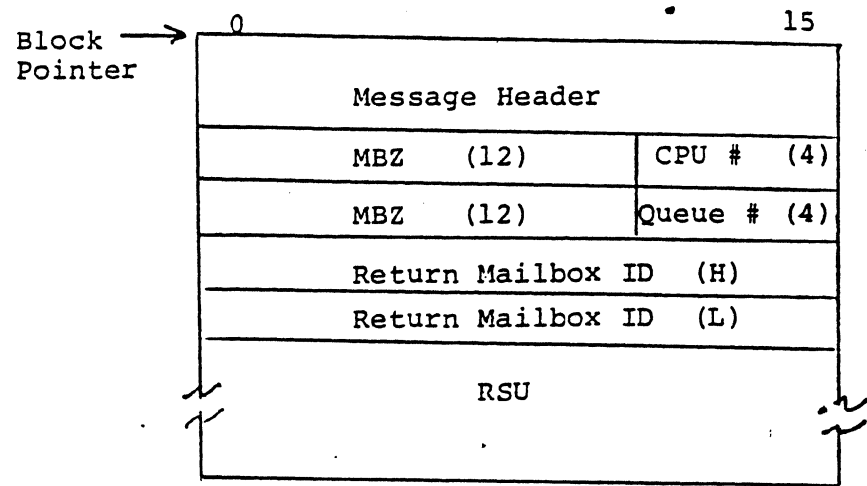
Figure 3-5 DMA Request/Confirm Message for Extended Buffer Transfer



Notes:

1. This message is used for requesting a DMA operation to move data from a logical buffer in RAM to one in L6 memory and to move a block of data from RAM to L6 memory and is used for confirming completion of the operation.
2. A valid Return Mailbox ID must be provided in the request message and the Status word must be all zeros.
3. In the request message the Message Type for the Header is BFLCB-TO-L6. The Header contains the RAM Buffer Descriptor.
4. The format of the L6 Buffer list is shown in Figure 3-8.
5. If an error occurs on the buffer transfer the block transfer and interrupt are not performed.
6. If an error occurs on the block transfer the interrupt is not performed.
7. In the confirm message the Message Type for the Header is the same as in the request message except that the suffix CF is added.
8. See Figure 3-1 for Status bit Definition.

Figure 3-6 DMA Request/Confirm Message for Combined Buffer and Block Transfer



Notes:

1. The Message Type for the Header is as follows for the request:  
 SET-RETRY = Set "CPU#" into Retry table for specified Queue  
 READ-RETRY = Read Retry table for specified queue
2. "CPU#" specifies the CPU to which all deferred interrupts in the specified Queue are henceforth to be sent.
3. "Queue#" refers to one of the possible 16 deferred interrupt queues maintained by the Memory DMA process which are used in accordance with interrupts requested in DMA requests.
4. The result of a "Read table" will be placed in the "CPU#" field of the return message. The Message Type for the Header for the return message is READ-RETRY.CF.

Figure 3-7 Retry Dispatch Table Request/Confirm Message

The L6 Buffer List pointer shown in Figure 3-8 specifies the starting location of a list of descriptors of main memory buffers. The list itself is in LAC RAM; each descriptor in the list consists of a four-byte address and a two-byte range. The list format is similar to that used in LCBs and is shown in Figure 3-8. All L6 buffer addresses and ranges are byte-oriented both in the LCB and in the list given to the Memory DMA process.

The Memory DMA process, without software intervention, performs, by means of its associated interrupt routines, as many loadings and reloadings as is necessary to perform the total requested DMA operation, including scatter/gather.

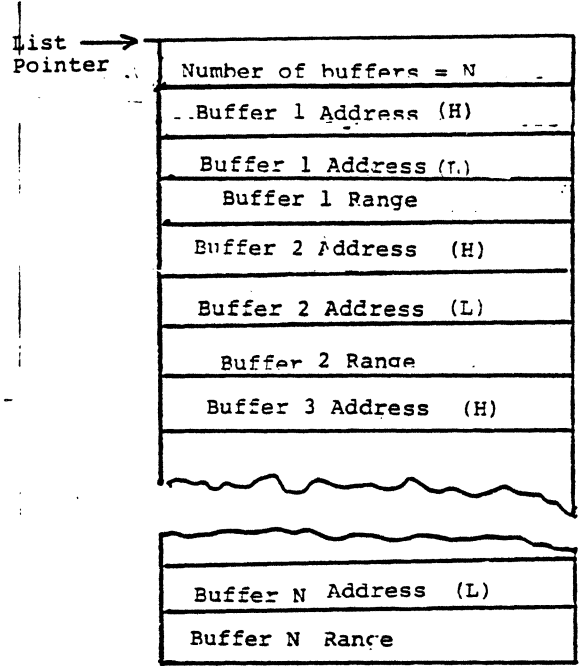


Figure 3-8 L6 BUFFER LIST FORMAT

HONEYWELL		SPEC. NO.	SHEET	REV
		60149766	64	H

### 3.8.1.2 MAC Transmit and Receive Processes

This set of processes performs all functions required to interface the hardware facilities of the adapter(s) to serve the needs of the CS and SM software processes. These functions include:

- o Transmission of data and control messages for LLC.
- o Dispatch of messages received by the adapter(s) to the proper process.

The MAC processes used during normal running consist of a Transmit, a Receive, and a Layer Management process for each attached adapter. During its initial running the System Management process spawns a MAC Layer Management process for each physically present adapter. The Layer Management process, in turn, spawns the Transmit and Receive processes for the adapter and aids the MAC users (LLC layers) and interrupt routines in establishing the mailbox IDS needed to communicate with MAC.

The format of mailbox message used by the Link layer CS processes to call for a Transmit operation is shown in Figure 3-9.

The Transmit process does not leave requests queued in its mailbox; instead it removes them as fast as possible and requeues them internally according to type and Service Class. The process also prepends the MAC header (DA, SA, Type/Length), and adds pads if necessary, to the message in the RAM.

As each Transmit operation is completed the adapter will cause an interrupt to the LAC microprocessor; this invokes the Adapter Interrupt routine which determines that a Transmit completion has occurred and sends a Transmit Interrupt mailbox message to the Transmit process. The Transmit process delivers the topmost Transmit request in its queue to the adapter. If some fault occurred in connection iwth the just-completed Transmit, the Transmit process will send a transmit fault indication message to System Management (Figure 3-9 and 3-10).

When a message is received from the LAN the adapter causes an interrupt to the LAC microprocessor; this invokes the Adapter Interrupt routine which determines that a receive operation has occurred and sends a Receive Interrupt Mailbox message to the Receive process. The Receive process discards the message if there is an error associated with it. The process sets up the adapter with a new receive buffer; it then strips out the MAC header from the mesage in RAM and sends a Data Indicate message to the mailbox of a CS software LLC receive process (the ID of that mailbox was given during startup). Finally, it obtains a new spare buffer from the Kernel O.S. The format of the Data Indicate message is shown in Figure 3-14

HONEYWELL		SPEC. NO.	SHEET	REV
		60149766	65	H

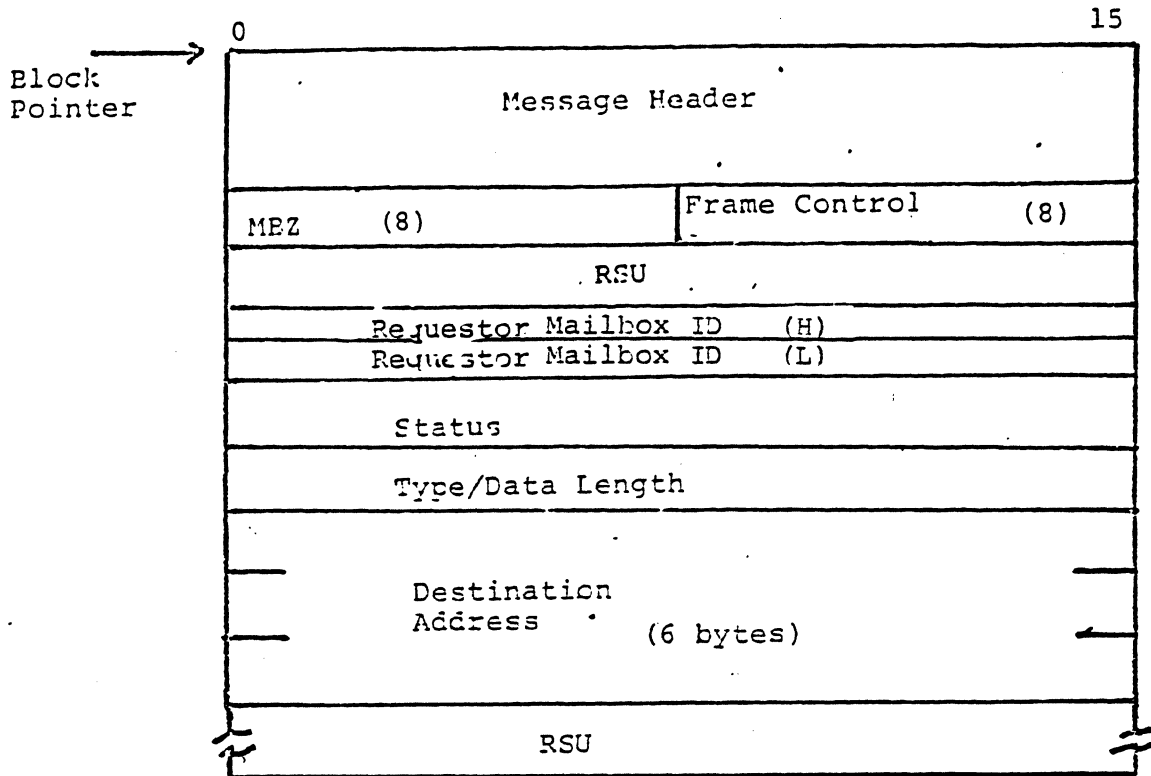
The MAC Transmit and Receive processes may encounter an O.S. error in performing one of the following O.S. Procedure calls:

- o SENDMSG
- o BRECEIVE
- o ALLOCATE
- o GETBUF
- o PREPENDBUF
- o UNAPPENDBUF

If such an error occurs, a message in the format of Figure 3-11 is sent to the System Management process using the appropriate Event code from Table 2-1.

The normal running flow of the MAC processes is shown in Figure 2-11; this figure does not include the initial execution of MAC processes. Additional details may be found in the MAC Component Specification.





Notes:

1. The Message Type for the Header of the request message is MAC\_DATA\_REQ.
2. The header must contain a RAM buffer descriptor.
3. In the request message the Status word must be all zeros.
4. The TYPE/Data Length field is used for IEEE 802.3 Standard CSMA/CD and Ethernet.
5. The Frame Control parameter is used by IEEE 802.4 & 802.5 Standard adapters.
6. The Destination Address field is used for all IEEE 802 Standard adapters.
7. For two-byte IEEE 802 Standard addressing, the first four bytes of the Destination address field must be zero.
8. In the Fault Indicate message the Message Type is TXFLT:
9. See Figure 3-10 for Fault Indicate status.

Figure 3-9. DATA TRANSMIT REQUEST/FAULT INDICATE MESSAGE BLOCK

0	7	8	9	10	11	12	13	14	15
ACCESS CODE	INV FCN	PHY	RAM NE	RAMP	UR	NED	EB	UT	

UT=Unsuccessful transmission  
     Excess collisions (CSMA only)  
     Excess retries  
     Not accepted (Token Ring only)  
 EB=Error bit ON (Token Ring only)  
 NED=Non-exist destination (Token Ring only)  
 UR=Unsuccessful due to underrun  
 PHY=Unsuccessful due to not In-Ring,  
     not On-Bus, PHYS power Off.  
 RAMNE=Non-existent RAM Memory  
 RAMP=RAM parity error  
 INVFCN=Invalid Function (Message Type)

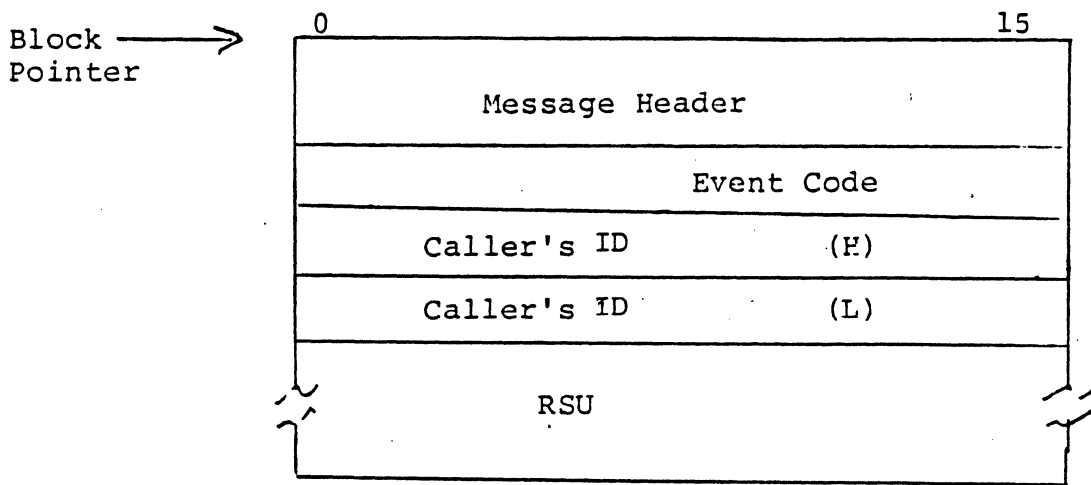
Access Code:

000AAA00

AAA=Access class actually  
     used (Token Bus only)

Figure 3-10 FAULT INDICATE STATUS FOR DATA TRANSMIT

HONEYWELL		SPEC. NO.	SHEET	REV
		60149766	68	H



**Note:**

The Message Type for the Header is as follows:

LMINT - Control Indicate from an adapter (see appendix for Event Codes)

SM\_EVENT\_MSG - Error response received on O.S. call (see subsection 2.19 for Event Codes)

Figure 3-11 Event Indicate Message Block

HONEYWELL		SPEC. NO.	SHEET	REV
		60149766	69	H

### 3.8.1.3 I/O Order Dispatch (IODISP)

This process performs the dispatch of incoming I/O orders to the appropriate CS or SM process by sending a mailbox message containing the I/O order to the mailbox ID corresponding to an entry in its Dispatch tables. The Dispatch tables consist of a 64-entry Mailbox Directory Table plus a number of 16-entry Mailbox ID Tables. Each entry in the Directory table is either NULL or is a pointer to the base of an ID Table. Each entry in an ID Table is either NULL or is a Mailbox ID.

The IODISP process uses the low order 6 bits of the IOLD order channel number to index into the Directory table and if a non-NULL entry is found there, uses the Function code (high order 4 bits of the IOLD "Range" word) to index into the ID table.

The CS and SM processes, in order to have particular IOLD orders dispatched directly to them by this process, must send a mailbox message to this process identifying the mailbox ID to be used. The format of the mailbox message used to set a mailbox ID into the Dispatch table is shown in Figure 3-12. This setup should be done during start-up. Table entries may also be changed or read during normal running. Any table entry which has not been specifically set up with mailbox IDs will contain a "NULL" ID and any orders addressed to its corresponding channel address and Function code will be dispatched to the Megabus Layer Management. The format used for all dispatched IOLD message blocks is shown in Figure 3-13.

The IODISP process may encounter an O.S. error in performing one of the following O.S. Procedure calls:

- o SENDMSG
- o ALLOCATE
- o BRECEIVE

If such an error occurs, a message in the format of Figure 3-11 is sent to the System Management process using the appropriate Event code from Table 2-1 if applicable.

HONEYWELL		SPEC. NO.	SHEET	REV
		60149766	70	H

The normal running flow of the IODISP process is shown in Figure 2-8 ; this figure does not include the initial execution and Table setup. Additional details may be found in the IODISP Component Specification.

The I/O Dispatch interrupt routine (Subsection 3.8.2.2) is associated with this process.

If the IOLD is a Start I/O, IODISP will wait indefinitely for a mailbox ID to be set up for channel 0. It is assumed that the LACS Driver will provide a time-out on Start I/O orders.

0				15
HEADER				
FUNCTION	MBZ	CHAN	(6)	
DISPATCH POINTER (H)				
DISPATCH POINTER (L)				
MAILBOX ID (H)				
MAILBOX ID (L)				
RETURN MAILBOX ID (H)				
RETURN MAILBOX ID (L)				
RSU				

NOTES:

- In the request message, the Message Type for the Header is as follows:

REG-MBPTR = Load Dispatch Pointer and Mailbox ID for specified Channel and Function

READ-MBPTR = Read Dispatch Pointer and Mailbox ID for specified Channel and Function
- In the confirm message the Message Type for the Header is as follows:

REG-MBPTR.CF = Confirming Pointer and Mailbox ID Entry Dispatch Request

READ-MBPTR.CF = Confirming Read of Dispatch Pointer Request.

Figure 3-12 IODISP Dispatch Table Setup Request/Confirm Message Block

#### 3.8.1.4 Megabus Layer Management

This process provides an interface between the IODISP or MEMDMA processes or their associated interrupt routines and the System Management process.

- o For an IOLD which cannot be dispatched due to a NULL Dispatch table entry, Megabus Layer Management will:
  - (1) Send a DMA request to MEMDMA to copy the first word of the LCB (which contains the CPU Number and LEVEL) into RAM.
  - (2) Send a DMA request to MEMDMA to write INVFACT status and Status Complete into the LCB and to interrupt the CPU.

#### 3.8.1.5 MAC Layer Management

This process handles various unexpected events reported by the adapter due to happenings in the adapter or on the LAN. The process also provides an interface between these processes and the System Management process.

HONEYWELL		SPEC. NO.	SHEET	REV
		60149766	73	H

The appendices list the Events and Faults which may be reported by the different adapters. The process reports all these to System Management.

The process also performs LMI requests received from System Management.

The format of mailbox messages used to call for an LMI control function is shown in Figure 3-15. On completion of the operation a return message (confirm) will be sent to the specified Return Mailbox; this message is of the same format except that the Status field will be valid and the Return Value field will contain whatever information was requested (assuming the execution of the request was successful in obtaining it). The coding of the various identifiers and Status is given in the System Management and MAC Layer Management Component Specification.



HONEYWELL		SPEC. NO.	SHEET	REV
		60149766	74	H

### 3.8.2 IF Software Interrupt Routines

The microprocessor interrupts are assigned as follows, listed in order of increasing priority:

1. Interrupt Retry
2. DMA Controller (RAM buffer/I/O order queue)
3. Megabus DMA Register Exhaust
4. I/O Dispatch
5. Clock Interrupt
6. Adapter Interrupt
7. RAM parity error

Of these interrupt routines, the clock Interrupt is handled by a routine furnished by the Bridge O.S. The remainder are handled by IF Software routines and are described in this section.

#### 3.8.2.1 Interrupt Retry

Subsections 2.15 and 3.8.1.1 describe the scheme of sending interrupts to the CPU and of the queuing of those which are rejected, as implemented by the Memory DMA process.

When a CPU switches from one running LEVEL to another (and thus becomes potentially able to accept an interrupt) it sends a Resume Interrupt (RINT) signal over the Megabus. The LAC hardware stores this occurrence and, after any currently underway Memory DMA request has been completed (or immediately, if none was underway), causes an Interrupt Retry interrupt request to be sent to the microprocessor.

The Interrupt Retry routine will attempt to send the interrupt at the top of the queue for each CPU. Any interrupts which are accepted are removed from the queue and their DMA request Message Blocks are returned to the requesting process mailbox in the form of a confirmation. Any interrupts which are rejected are left in the queue. The routine finds the location of the queue from a known location within the MEMDMA process which it is associated with.

The Interrupt Retry routine uses the Retry Dispatch Table to determine which CPU the interrupt from each queue should be addressed to, as described in subsection 3.8.1.1.

HONEYWELL		SPEC. NO.	SHEET	REV
		60149766	75	H

### 3.8.2.2 I/O Dispatch

The I/O Dispatch interrupt routine does the dispatching of IOLD orders, which have been issued to the LACS, to the proper CS Software process.

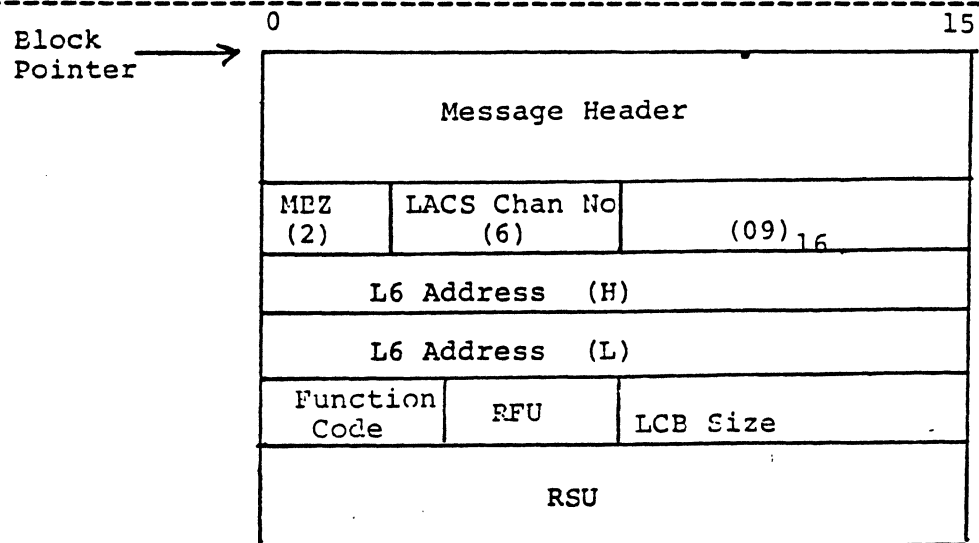
An I/O Order interrupt request is sent to the microprocessor whenever the DMA controller channel which is used for incoming I/O orders takes an IOLD order from the Megabus and places it in a temporary working queue, as described in subsection 2.13.

As shown in Figure 2-8, the routine checks that the I/O order is valid; if it is, the routine obtains a RAM block of memory for a mailbox message, assembles a mailbox message containing the IOLD information (see Figure 3-13) and sends the message to the mailbox specified in its Dispatch table according to the particular Channel number and Function code of the IOLD.

If no mailbox has been specified for the Channel number and Function code used in the IOLD the mailbox message of Figure 3-13 will be sent to the mailbox of the Megabus Layer Management entity (MBLME)

If a software error occurs on an O.S. Procedure call an Event Indicate message (Figure 3-11) is sent to System Management.

This routine is associated with the I/O Dispatch process, IODISP.



Note:

1. The Header has a NULL Buffer Descriptor
2. The L6 Address and the LCB Size are byte-oriented but must be word-bound (i.e the Lsb must be zero),
3. The Message Type for the Header is IOLD.

Figure 3-13 IOLD Information Message Block Format

### 3.8.2.3 DMA Controller

This routine provides reloading of the DMA controller channels. For the channel which is used for Memory DMA it does the reloading of the individual pointers associated with the single logical buffer in RAM which is being used in the DMA operation. For the channel which is used for I/O order input it reloads the channel with a new Temporary queue location for the next I/O order.

The request for this interrupt is generated by the DMA controller when either channel has a range exhaust.

This routine is associated with the Memory DMA process, MEMDMA.

HONEYWELL		SPEC. NO.	SHEET	REV
		60149766	77	H

#### 3.8.2.4 Megabus DMA Register Exhaust

This routine provides reloading of the Megabus address and range registers when performing a scatter-gather Memory DMA operation. If a RAM Block transfer was requested in the Memory DMA Request (Figure 3-6) in addition to one with a logical DMA buffer, this routine will reload the registers and the DMA controller to perform this transfer after the buffer transfer has been completed. If an interrupt was also specified, this routine will attempt the interrupt after completing all transfers. Finally, the routine will fetch the next DMA request (if any) in the Memory DMA process mailbox and set up the register and DMA controller to start this next operation.

This routine is associated with the Memory DMA MEMDMA process. Its flow and functionality are represented in Figure 2-9.

The request for this interrupt is generated by the Megabus range register when it has a range exhaust.

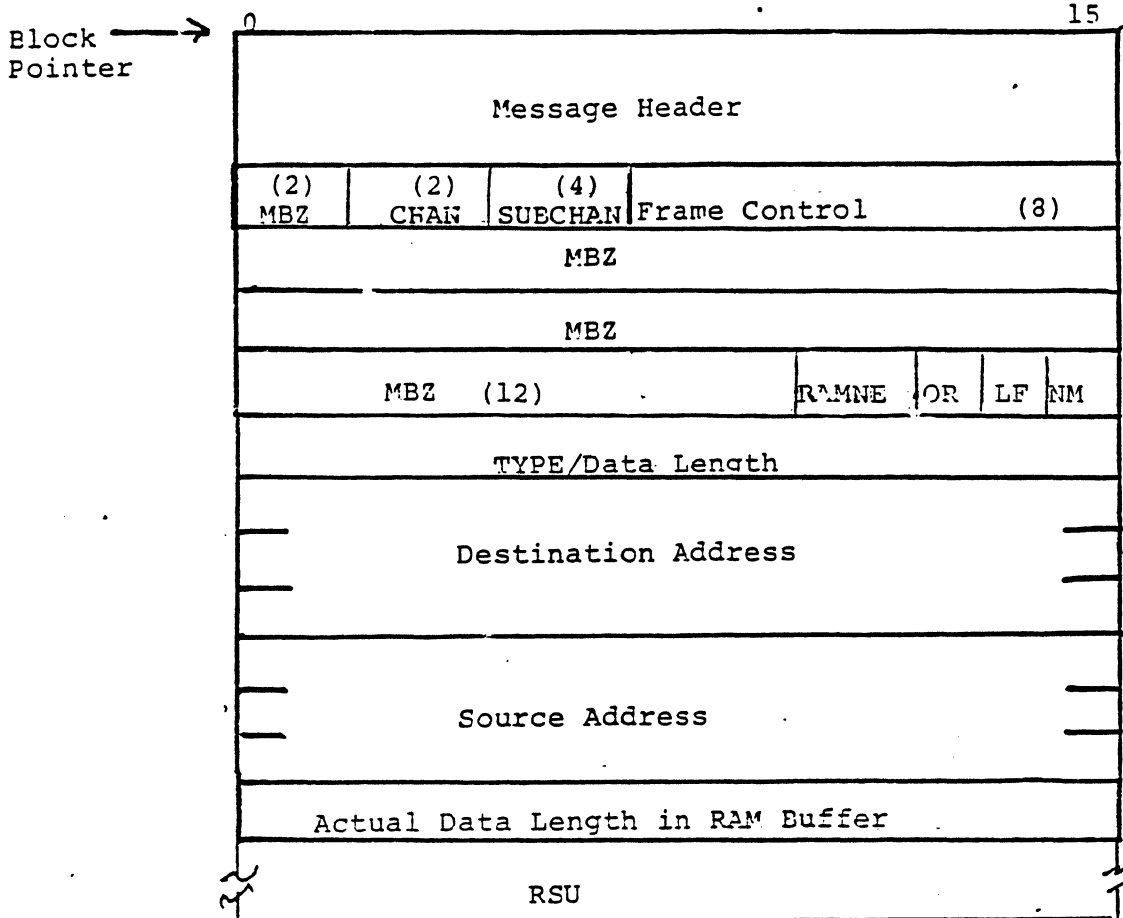
#### 3.8.2.5 Adapter Interrupt

The routine provides handling of interrupts generated by the attached adapters due to completion of requested operations or to the occurrence of an asynchronous event. The routine is associated with the MAC processes. Its flow and functionality is represented in Figure 2-10.

When a message has arrived from the LAN the routine will send a mailbox message to the associated MAC Receive process. For a transmit completion the routine will send a mailbox message to the associated MAC Transmit process. If an unusual event has been detected by the adapter a mailbox message as shown in Figure 3-11 is sent to the associated MAC Layer Management process.

#### 3.8.2.6 RAM Parity Error

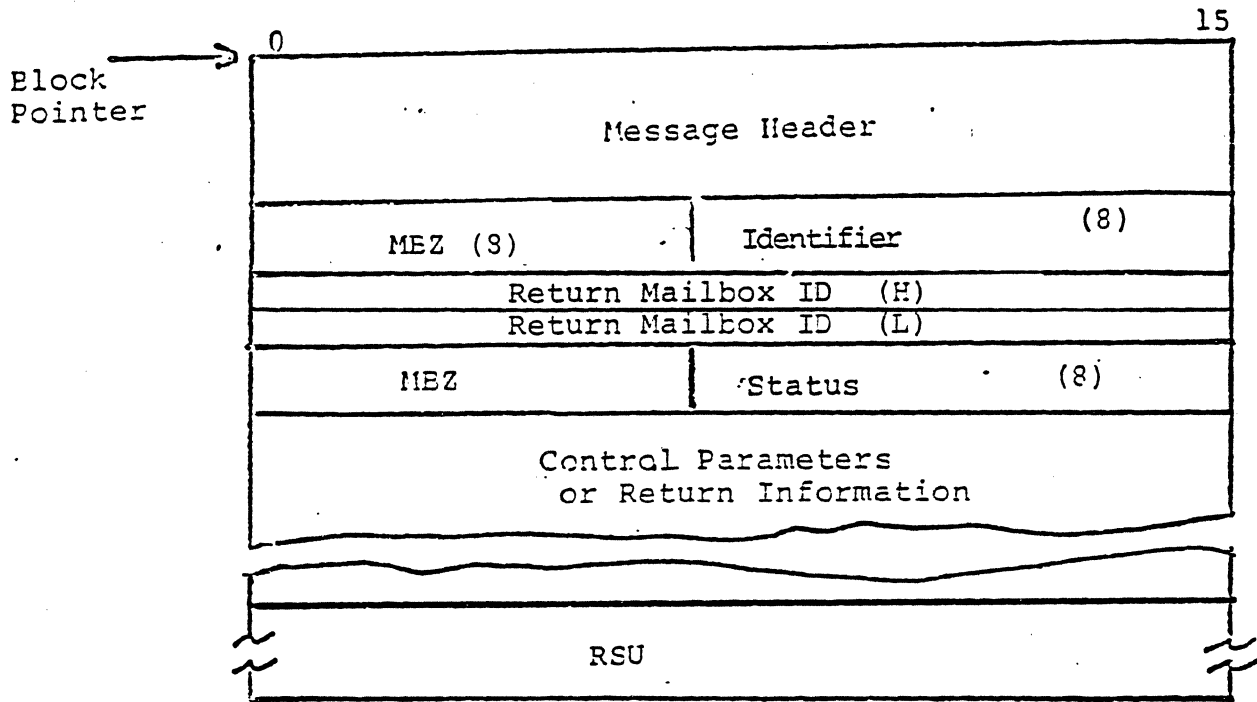
This routine is invoked when a parity error occurs on an access to RAM performed by the microprocessor to fetch an instruction or operand. The functions performed by this routine are specified in Subsection 3.9.



Notes:

- The Header contains the buffer descriptor.
- The Message Type for the Header is MAC\_DATA\_IND.
- LF= Long Frame (longer than buffer)  
OR= Overrun  
RAMNE= Non-existent RAM
- For two byte addresses, the high order four bytes of the address fields are zero.

Figure 3-14 DATA INDICATE MESSAGE BLOCK



Notes:

1. For the request the Message Type for the header is SM\_REQ\_MSG.
2. For the confirm the Message Type for the header is SM\_CNFRM\_MSG.
3. See System Management and MAC Layer Management Component specifications for message detailed definition.

FIGURE 3-15 LMI REQUEST/CONFIRM MESSAGE BLOCK

HONEYWELL		SPEC. NO.	SHEET	REV
		60149766	80	H

### 3.9 HANDLING OF HARDWARE-DETECTED ERRORS

Hardware errors, faults and unusual happenings occurring within an adapter, at its connection with the LAN, or on the LAN itself, are reported to MAC Layer Management which may report it to the SM process via an Event Notification message. In IEEE 802 Standard terminology, these are Control Indicate messages. In general these are Reportable Catastrophic errors as far as the particular LAN port is concerned.

Megabus errors and RAM parity errors occurring during a DMA operation with the Megabus are reported to the process which requested the operation via status in the DMA Confirm Message returned by the Memory DMA process.

A RAM parity error occurring during a DMA operation being performed by an adapter is reported to the process which sent the request to the MAC process via status in the Data Confirm message returned by the MAC process.

On a MOVE instruction being performed by the microprocessor, RAM parity is not regenerated but is simply copied along with the data. No action is taken.

The following errors are Unreportable Catastrophic errors:

A RAM parity error occurring on an instruction fetch or operand fetch by the microprocessor will cause a Trap to a routine in PROM. THE Trap routine will cause all adapters to logically disconnect from their LANs, will store an indication of the fault in the RAM (Figure 7-4). will set the Megabus logic to NAK all I/O orders except an Output LACS Control order (FC=01) and will leave the LAC under PROM-resident firmware control. It is assumed that the LAN Manager in the Level 6 will eventually issue a Stop I/O order and then DUMP the RAM.

Buss Errors illegal MC68000 OP codes, MC68000 parity errors, LAC hardware errors, and errors occurring on a Load/Store or Start I/O are treated the same as an Unreportable Catastrophic RAM Parity error.





HONEYWELL		SPEC. NO.	SHEET	REV
		60149766	81	H

## SECTION 4 INTERFACES

### 4.1 LEVEL 6 INTERFACE WITH LACS

The software-visible interface between Level 6 and the LACS uses LCBs as described in Section 2 and implements it via use of the I/O orders described in Section 3.

### 4.2 LAC INTERFACE WITH THE MEGABUS

The physical interface between Level 6 and the LACS is the Megabus, implemented in accordance with the Level 6 Bus reference specification with extension for support of a 32-bit address bus.

The LAC microprocessor controls LAC operations on the Megabus by controlling hardware Megabus registers and the Megabus DMA controllers. The DMA controllers send interrupts to the microprocessor on completion of a DMA operation.

The LAC board is able to use either the high priority (BSREQH) or low priority (BSREQ) request line so as to satisfy MRX bus, Extended bus, and Standard bus requirements. It may be necessary to provide a DIP switch on the board to select which line to use.

The LAC board will have a DIP switch which must be set in accordance with whether the board is attached to an MRX bus or a Standard/Extended bus. This switch selects proper use of connector Z01 pin 5.

The LAC will not support connection via the Megabus to older 18-bit-data memories.

A program-settable throttling arrangement will be provided on the LAC board so as to achieve equitable sharing of the Megabus DMA capability. Details are TBD.

Since the LAC is a buffered controller, LAC boards should always be located among the lowest priority bus positions in the system.

### 4.3 LAC INTERFACE WITH ADAPTERS

The physical interface between the LAC and the attached adapters uses an MC68000 type bus, the DMA bus of the Data Buffer RAM. Figure 4-1 shows this bus in detail.

The LAC microprocessor controls the adapters by sending them command blocks defining the operation desired. The adapters send interrupts to the microprocessor on completion of a DMA operation or on the occurrence of an asynchronous event. Details are specific to the particular adapter implementation and are covered in the MAC Component Specifications.

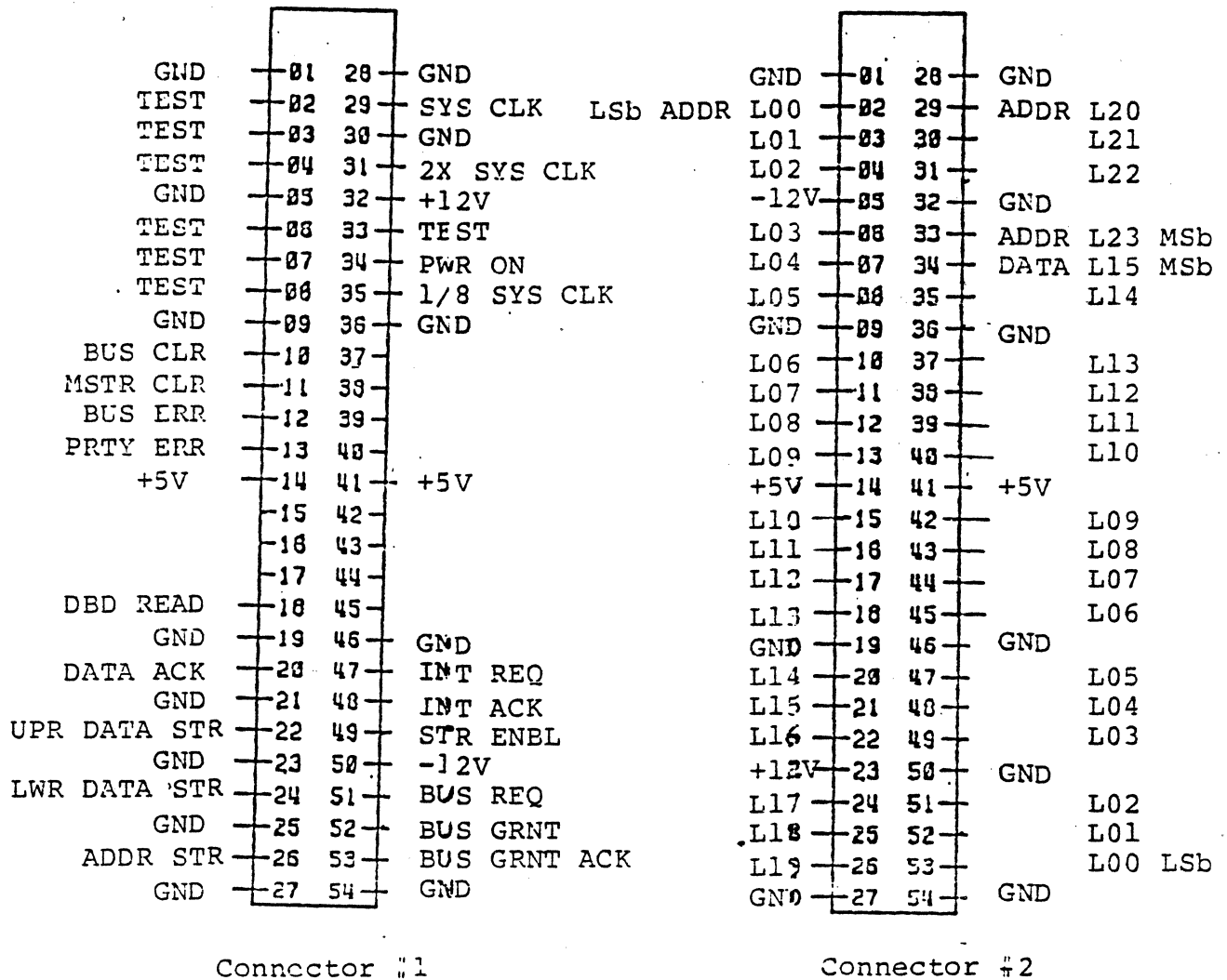


Figure 4-1 LAC-ADAPTER BUS

#### 4.4 TEST CONNECTOR INTERFACE

The LAC provides a test connector interface as shown in Figure 4-2. A separate Test Board may be plugged into the interface.

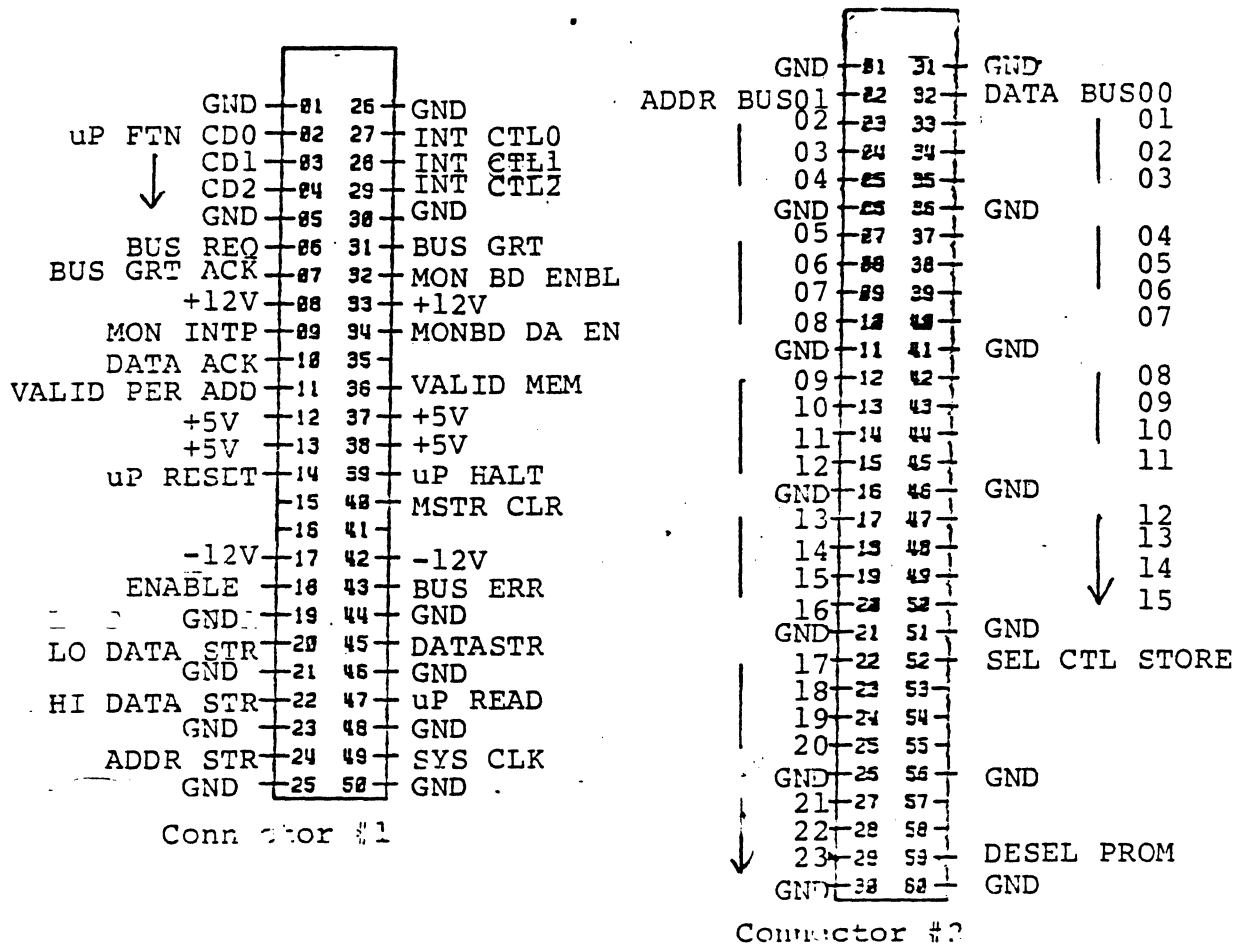


Figure 4-2 TEST CONNECTOR



HONEYWELL		SPEC. NO.	SHEET	REV
		60149766	84	H

## SECTION 5 PERFORMANCE

### 5.1 LAN DATA RATE

The LAC is capable of handling a peak data rate of 2.5 MBytes per second. This permits a 10 Mbps data stream from an adapter concurrent with a similar data rate to the Megabus or 10Mbps data streams concurrently on two adapters.

### 5.2 DATA THROUGHPUT

The packet rate capability of the LACS is heavily dependent upon the LACS Software performance. Measured performance values are not available at this time. However, simulation using timing estimates indicate that the LACS running software up through Transport layer may be capable of about 120 packets per second (total for Receive and Transmit). The packet rate is independent of packet size.

### 5.3 LAC INSTRUCTION TIMING

Instruction times for the MC68000 are found in the MC68000 User's Manual reference document, using 10 MHz as the clock speed except that all memory references include a WAIT state.

### 5.4 PRIORITY POSITION OF THE LAC ON THE MEGABUS

The LAC should be located as the lowest I/O device priority position on its bus (i.e. just above the CPU).

### 5.5 PRIORITY POSITIONING OF ADAPTERS ON THE LAC

Because of the priority scheme on the Data Buffer RAM bus, in assigning positions of adapter daughterboards on the LAC. The adapters which are connected to the highest speed LANS should be assigned to the lowest numbered adapter positions.



HONEYWELL		SPEC. NO.	SHEET	REV
		60149766	85	H

## SECTION 6 PHYSICAL STRUCTURE

### 6.1 PACKAGING

The LAC is a single motherboard that accommodates up to 4 adapter daughterboards (quarter - sized).

### 6.2 POWER

The LACS is powered from the Level 6 bus and has the following requirements:

- o 8.3 amperes @ + 5 Vdc
- o 0.1 amperes @ + 12 Vdc

Adapter power is not included in these figures.

### 6.3 ENVIRONMENT

- o Temperature: 10° to 50°C ambient
- o Humidity: 5% to 90%

### 6.4 SAFETY

The LACS meets Underwriter Laboratories and Canadian Standards Association requirements.

### 6.5 STANDARDS

The LACS meets the referenced HIS standards to the extent to which they apply.

### 6.6 TESTABILITY

The LAC provides a test connector. The LAC design meets the requirements of the MTG2 and MTG3 reference documents.

### 6.7 RFI EMANATION

The LACS complies with the requirements of Part 15 of Federal Communication Commission Rules for a Class A computing device when mounted in standard Level 6 chassis.





HONEYWELL		SPEC. NO.	SHEET	REV
		60149766	86	H

## SECTION 7 RELIABILITY AND MAINTAINABILITY

### 7.1 GENERAL REQUIRMENTS

- o Mean Time between Failures (MTBF)

28000 hours - LAC Motherboard

- o Mean Time to Repair (MTTR): 1 hour

Mean time to repair includes the time to determine and to replace the defective ORU and validate the repair.

The LAC motherboard is an ORU. Specifics on adapter-related ORUs are given in the Appendices.

### 7.2 MAINTENANCE FEATURES

#### 7.2.1 Quality Logic Test (QLT)

The LAC provides a resident program called QLT that verifies the operability of the LACS. A QLT light is provided on the LAC motherboard and on each adapter. The QLTs run whenever a LACS Hard Initialize is performed. The QLT tests the motherboard first and then the adapters in turn. As it starts, the QLT turns on the QLT light on the motherboard and performs its test. Each adapter is then tested in turn; as each unit's test begins, its QLT light is turned on and when its test is completed successfully, that unit's QLT light is extinguished. A QLT Status word is stored in RAM on completion of each step of each QLT; this word is part of the Configuration information which is shown in subsection 7.4. An adapter's ID is affected by an unsuccessful QLT (subsection 3.4).

#### 7.2.2 Error Checking and Reporting

The Megabus parity checking functions are supported. Failures are reported back to the LAC software as part of the status return for Memory DMA operations.

#### 7.2.3 Internal Loopback

Loopback of the data stream within the adapter may be invoked. Loopback at the Trunk coupler may be possible on some 802 type LANs. Further detail is given in the Appendices.

HONEYWELL		SPEC. NO.	SHEET	REV
		60149766	87	H

#### 7.2.4 Statistic Gathering

The MAC adapter gathers the statistics enumerated in subsection 2.16 using its hardware and firmware. Other statistics may be assembled within the LAC under software control.

#### 7.2.5 LAC RAM Dump/Load

The CPU can issue an I/O order which causes a portion of RAM in the LAC to be dumped into/loaded from main memory.

#### 7.2.6 FCS Checking

FCS generation/checking is provided normally by the adapter. These functions may be independently disabled so that FCS fields can be manipulated by software in the LAC or made visible to CPU software. This ability is adapter implementation specific.

#### 7.2.7 Adapter RAM Access

Software running in the LAC can have read or write access to locations in an adapter's RAM if this function can be supported by the LSI chips in the adapter. This not only allows a type loopback to test the loading of parameters into the adapter but also provides ability to monitor MAC conditions which are not normally visible.

#### 7.2.8 Remote Loopback

Through the TEST command, loopback of data through a remote station may be performed. This is a software function which is further defined in the IEEE 802 Standards.

#### 7.2.9 MAC Internal Status

Read access to a selected set of MAC internal timers and status as defined in the IEEE 802 Standards is provided.

#### 7.2.10 Remote Station Inquiry (XID)

The ability to inquire into the availability of a remote station is provided. This is a software function which is further defined in the IEEE 802 Standards.

7.2.11 System Area in RAM

An area in RAM (see figure 7-5) is provided for saving useful information regarding errors and faults.

7.3 MAINTAINABILITY

Through the use of a complete set of maintainability tools for the LACS that include the QLT, T&V routines, and troubleshooting and repair procedures, the following fault detection and resolution goals apply to the LACS:

<u>ORU</u>	<u>ISOLATION TO ONE ORU</u>	<u>ISOLATION TO TWO ORU'S</u>	<u>DETECTION</u>
CONTROLLER	93%	5%	98%
ADAPTER	93%	5%	98%

In measuring the above goals, shorts on tristate signals on the controller/adapter interface should not be inserted; instead the ground rules developed during NMLC fault insertion tests should be followed.

7.4 DETAILED ADDRESS SPACE ASSIGNMENTS

7.4.1 Overall Maps

Figures 7-1 and 7-2 show the overall map of LAC during QLT execution when the MC68000 is executing code in the PROM and during normal running, respectively.

	Byte Address
Adapter 03 Strobes	FF FF FE
Adapter 02 Strobes	FO 00 00 EF FF FE
Adapter 01 Strobes	E0 00 00 OF FF FE
Adapter 00 Strobes	00 00 00 CF FF FE
DMA Controller Strobes	C0 00 00 BF FF FE
Not Used	B0 00 00 AF FF FE
FIFO Output Strobes	A0 00 00 9F FF FE
Not Used	80 00 00 8F FF FE
Data Buffer RAM expansion to 256K	88 00 00 87 FF FE
Data Buffer RAM basic 64K	82 00 00 81 FF FE
Same as 30 00 00 to 3F FF FE	80 00 00 7F FF FE
Same as 20 00 00 to 2F FF FE	70 00 00 6F FF FE
Same as 10 00 00 to 1F FF FE	60 00 00 5F FF FE
Same as 00 00 00 to 0F FF FE	50 00 00 4F FF FE
Not Used	40 00 00 3F FF FE
uP RAM expansion to 256K	38 00 00 37 FF FE
uP RAM basic 64K	32 00 00 31 FF FE
Not Used	30 00 00 2F FF FE
Monitor Strobes	2D 00 00 2C FF FE
Not Used	2C 00 00 2B FF FE
Control Strobes	29 00 00 28 FF FE
Not Used	20 00 00 1F FF FE
PROM same as 00 00 00 to 00 7F FE	10 00 00 10 7F FE
Not Used	0F FF FE 00 80 00
PROM w/vectors	00 7F FE 00 00 00

Figure 7-1. ADDRESS SPACE LAYOUT DURING QLT

	Byte Address
Adapter 03 Strobes	FF FF FE
Adapter 02 Strobes	F0 00 00 EF FF FE
Adapter 01 Strobes	E0 00 00 DF FF FE
Adapter 00 Strobes	D0 00 00 CF FF FE
DMA Controller Strobes	C0 00 00 BF FF FE
Not Used	B0 00 00 AF FF FE
FIFO Output Strobes	A0 00 00 9F FF FE
Not Used	90 00 00 8F FF FE
Data Buffer RAM expansion to 256K	88 00 00 87 FF FE
Data Buffer RAM basic 64K	82 00 00 81 FF FE
Same as 30 00 00 to 3F FF FE	80 00 00 7F FF FE
Same as 20 00 00 to 2F FF FE	70 00 00 6F FF FE
Same as 10 00 00 to 1F FF FE	60 00 00 5F FF FE
Same as 00 00 00 to 0F FF FE	50 00 00 4F FF FE
Not Used	40 00 00 3F FF FE
uP RAM same as 02 00 00 to 07 FF FE	38 00 00 37 FF FE
uP RAM same as 00 00 00 to 01 FF FE	32 00 00 31 FF FE
Not Used	30 00 00 2F FF FE
Monitor Strobes	2D 00 00 2C FF FE
Not Used	2C 00 00 2B FF FE
Control Strobes	29 00 00 28 FF FE
Not Used	20 00 00 1F FF FE
FROM	10 00 00 10 7F FE
Not Used	10 00 00 0F FF FE
uP RAM expansion to 256K	08 00 00 07 FF FE
uP RAM basic 64K	02 00 00 01 FF FE
	00 00 00

Figure 7-2. ADDRESS SPACE LAYOUT DURING NORMAL OPERATION

7.4.2 Dedicated RAM area Layouts

Figure 7-3 shows the location and layout of the RAM memory area assigned as the I/O order temporary queue.

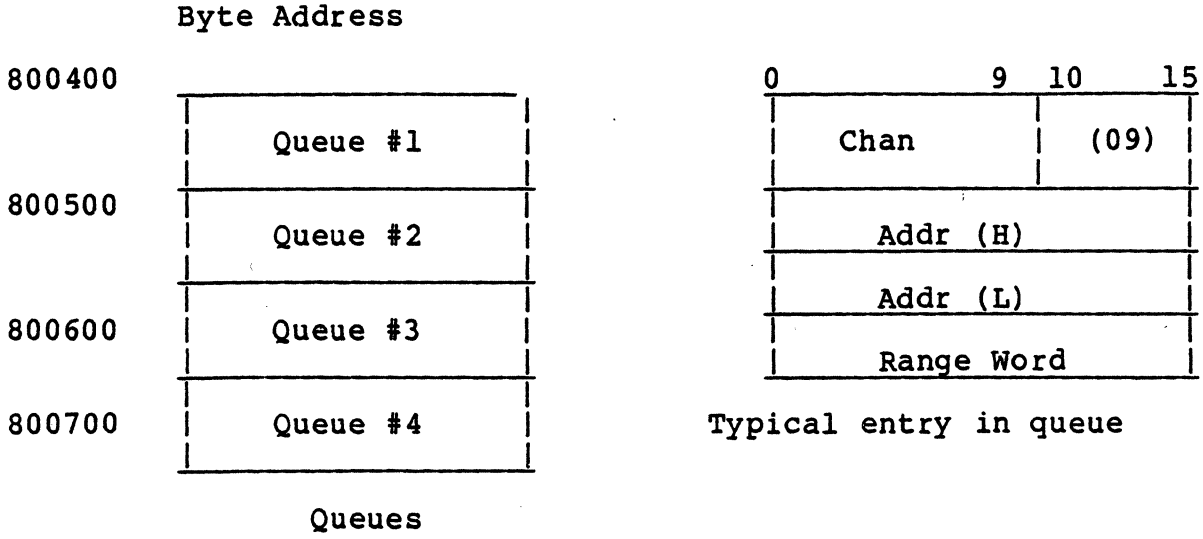
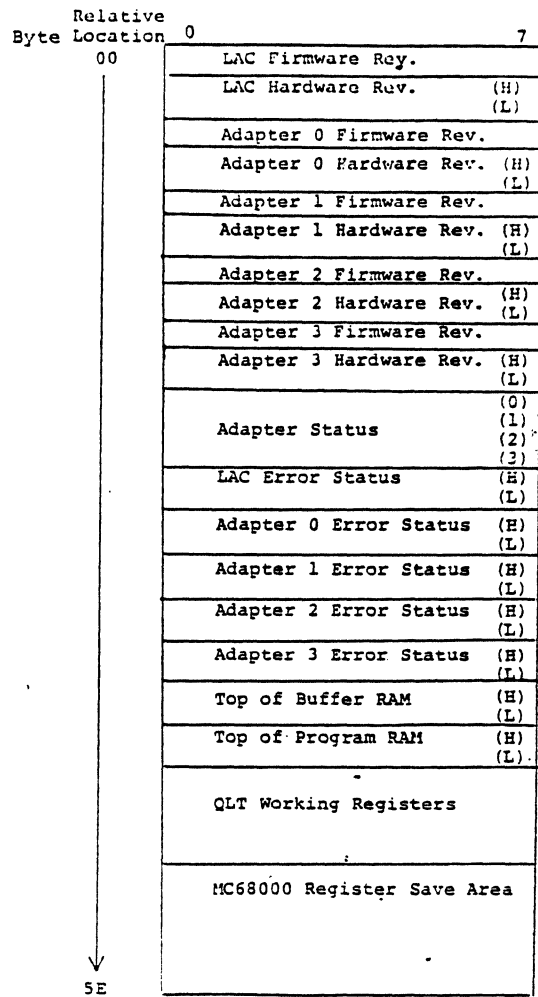


Figure 7-3. I/O TEMPORARY QUEUE

Figure 7-4 shows the layout of the Configuration information (Device IDs, RAM memory sizes, Firmware Revision Number, QLT Status) as it will exist in main memory after performing a Read Configuration operation (section 3.2.5.2).

Figure 7-5 shows the location and layout of the RAM memory area assigned as a System Area for Statistics and records of Event notifications (if maintained by SME software) and Fatal errors.



1. See Figure 7-5 for LAC Error Status

Figure 7-4 Configuration Information

BE	RF	IOC	HE	UPE	(RFU)	NEM	L6B	MR
EVENT CODE								

- BE = 68000 BUS ERROR
- RE = 68000 RAM PARITY ERROR
- IOC = ILLEGAL 68000 INSTRUCTION
- HE = LAC HARDWARE ERROR
- UPE = MICROPROCESSOR PARITY ERROR
- NEM = NON-EXIST MEMORY ON LOAD/STORE OR START I/O LCB
- L6B = MEGABUS PARITY ERROR ON LOAD/STORE OR START I/O LCB
- MR = MEMORY RED ON LOAD/STORE OR START I/O LCB

EVENT CODE = Unreportable Catastrophic error (see Table 2-1)

Figure 7-5 LAC Error Status





HONEYWELL		SPEC. NO.	SHEET	REV
		60149766	93	H

APPENDIX A TOKEN BUS ADAPTER

A.1 GENERAL DESCRIPTION

The Token Bus Adapter consists of a (TBD) sized daughterboard that mounts on the LAC motherboard. It contains logic to interface with the Data Buffer bus, a Media Access controller, digital logic for the Physical layer, and an interface to the R.F. Modem (which is separately packaged) for Broadband application, and a drop cable interface for Baseband applications.

Figure A-1 is a block diagram of the adapter. The interface with the modem is described in the modem purchase specification reference in Subsection 1.2.

The power requirements for the daughterboard are as follows:

TBD amperes @ + 5VDC  
TBD amperes @ ± 12VDC

The maximum Station Delay for this adapter is TBD.

The adapter ID is TBD

Control Indicate Event Codes are as follows:

(0081)<sub>16</sub>=Jabber Inhibit  
(0082)<sub>16</sub>=Faulty Transmitter  
(0083)<sub>16</sub>=Change of Successor  
(0084)<sub>16</sub>=Null Successor

HONEYWELL		SPEC. NO.	SHEET	REV
		60149766	94	H

(TBD)

Figure A-1. TOKEN BUS DAUGHTERBOARD BLOCK DIAGRAM

HONEYWELL CONFIDENTIAL & PROPRIETARY

HONEYWELL		SPEC. NO.	SHEET	REV
		60149766	95	H

The adapter and the referenced modem are Optimum Replaceable Units (ORUs). Reliability and maintainability figures for these units are as follows:

<u>UNIT</u>	<u>MTBF</u>	<u>MTTR</u>
Adapter	TBD hrs	TBD hrs
RF Modem	TBD	TBD

The adapter and the referenced modem have built-in loopback paths which are chosen to utilize as much as possible of the particular unit's circuitry so as to aid in ORU resolution. The specific loopback points are TBD.



HONEYWELL		SPEC. NO.	SHEET	REV
		60149766	96	H

APPENDIX B CSMA/CD (ETHERNET) ADAPTER

B.1 GENERAL DESCRIPTION:

The CSMA/CD Adapter is a quarter-sized daughterboard that mounts on the LLC motherboard. It contains interface logic to interface with the MC68000 bus and two LSI controller chips which provide the Media Access and physical layer functions for interface with the CSMA/CD (Ethernet) transceiver.

The adapter ID returned in Input Device ID operations is as follows:

(2E08)16=Normal Ethernet  
 (2E88)16=QLT Error Ethernet  
 (2E0C)16=Normal 802.3  
 (2E8C)16=QLT Error 802.3

Figure B-1 is a block diagram of the adapter.

The interface with the transceiver is in accordance with IEEE 802 specifications. It does not include the optional Control Out interface circuit.

The Ethernet Controller AMD 7990 chip requires that buffers be at least 100 bytes in size when chained for a packet and 64 bytes minimum when the packet exists in one buffer only. The Honeywell-supplied CS and MAC software observes these requirements.

Control Indicate Event Codes are as follows:

- (0091)16=Heartbeat Fail
- (0092)16=Carrier Sense Failure
- (0093)16=Late Collision Detect
- (0094)16=Jabber Error
- (0095)16=Received Frame Too Long

The power requirements for the daughterboard are as follows:

1.2 amperes @ + 5VDC

The adapter is an ORU.

Maintainability figures for the adapter are as follows:

MTBF 494000 hours MTTR 1 hour

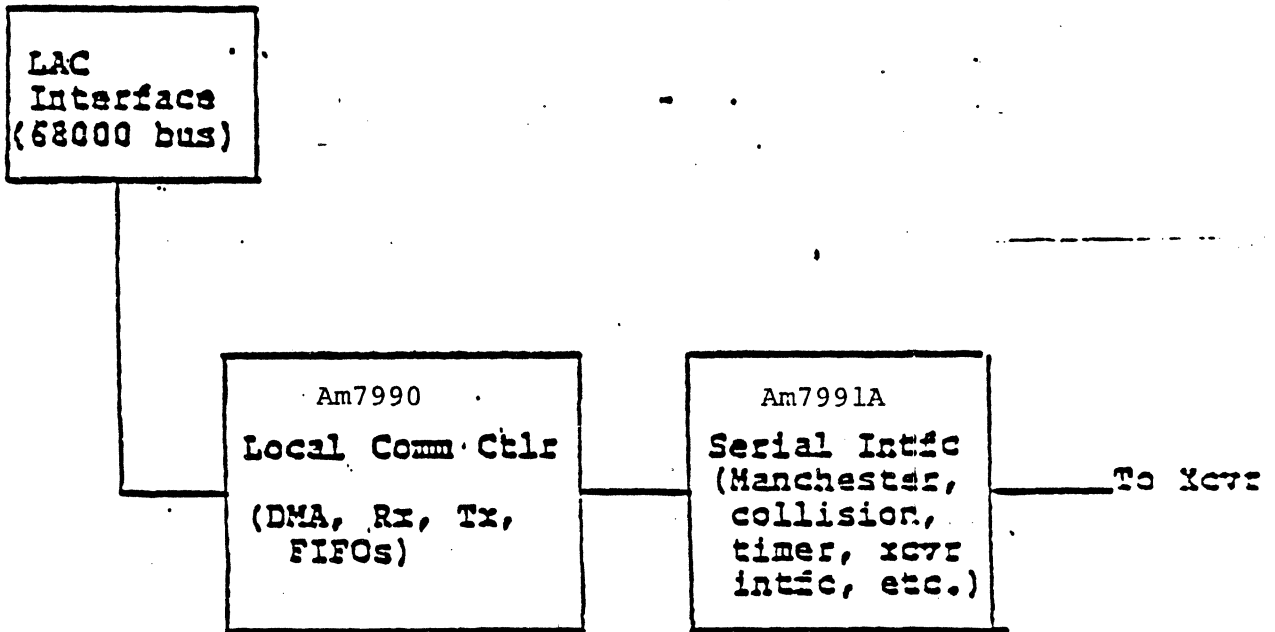


Figure B-1 CSMA/CD (ETHERNET) BLOCK DIAGRAM

HONEYWELL		SPEC. NO. 60149766	SHEET 98	REV H
-----------	--	-----------------------	-------------	----------

## APPENDIX C TOKEN RING ADAPTER

### C.1 GENERAL DESCRIPTION

The Token Ring adapter is a half-size daughterboard that mounts on the LAC motherboard. It contains logic to interface with the MC68000 bus, Media Access Controller, Physical layer circuitry and an interface to a Media Interface Connector (MIC). The MIC provides a connection to the ring Trunk coupling Unit (TCU) which is separately packaged. LSI chips provide the MAC and physical layer functions.

The adapter ID returned in Input Device ID operations is

(2E04) 16= Normal  
(2E84) 16= QLT Error

Power requirements for the adapter are as follows:

TBD amp @ +5VDC

The adapter is an ORU. Maintainability figures for the adapter are as follows:

MTBF: TBD hours MTTR: 1 hour

Figure C-1 is a block diagram of the adapter

Control Indicate Event Codes are as follows:

(0011) 16 = Private Event  
 (0012) 16 = Claim Token State  
 (0013) 16 = Transmit Beacon  
 (0014) 16 = Receive Frame Beacon  
 (0015) 16 = Active Monitor State  
 (0016) 16 = Standby State  
 (0017) 16 = Duplicate Address Detected  
 (0018) 16 = Request Initialization  
 (0019) 16 = Upstream Neighbor Change  
 (0020) 16 = Notification Incomplete  
 (0021) 16 = Error

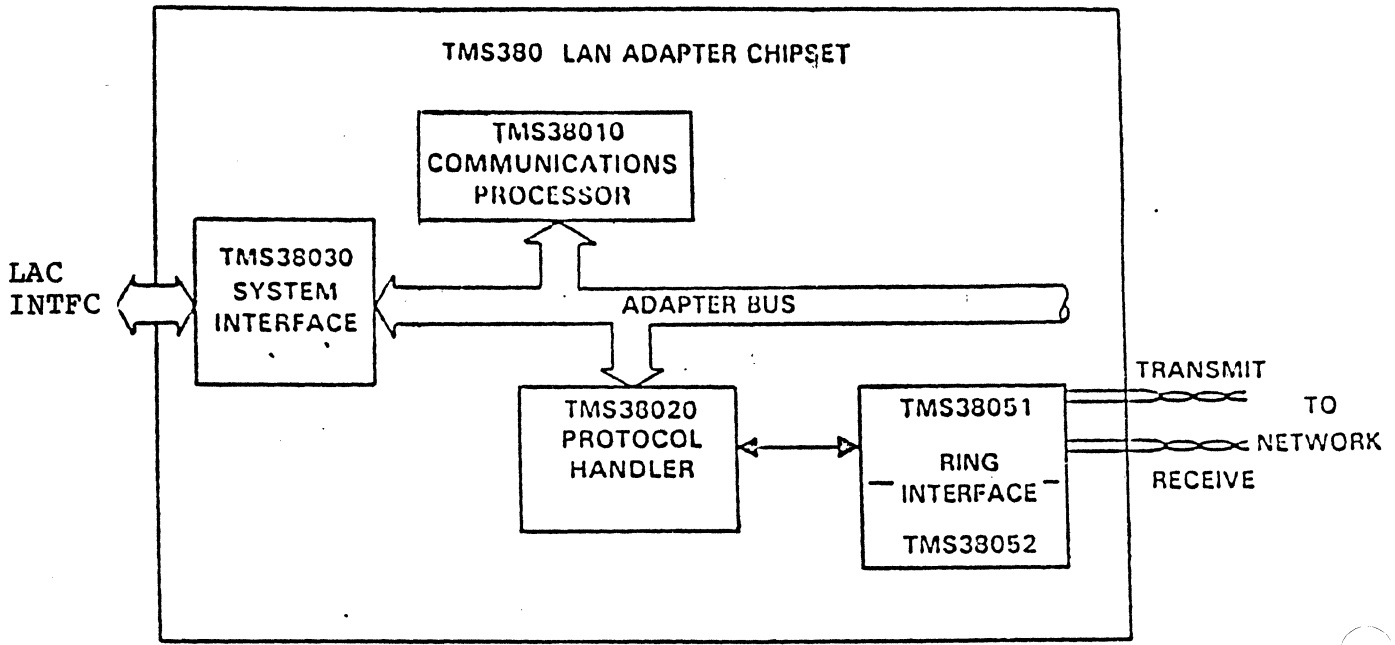


FIGURE C-1 TOKEN RING BLOCK DIAGRAM



HONEYWELL	SPEC. NO. 60149766	SHEET 100	REV E
-----------	-----------------------	--------------	----------

APPROVAL SHEET ( For Revision G)

PREPARED BY: A. Hirtle  
A. Hirtle  
Systems Engineering & Standards

DATE: Jan 14, 1985

APPROVED BY: John W Conway  
J. Conway, Manager  
System Engineering & Standards

DATE: Jan 21, 1985

APPROVED BY: R. Farrell  
R. Farrell, Manager  
Communications Hardware Development

DATE: March 7, 1985

APPROVED BY: W. K. O'Neill  
W. K. O'Neill, Manager  
Communications Software Development

DATE: March 6, 1985

APPROVED BY: D. DeVoy  
D. DeVoy, Director  
MicroSystems, Communications &  
Terminals

DATE: March 20, 1985



HONEYWELL PROPRIETARY

<b>Honeywell</b> HONEYWELL INFORMATION SYSTEMS	SPEC. NO. 60149766	SHEET 101	REV. H
---	-----------------------	--------------	-----------

APPROVALS FOR REVISION H

PREPARED BY *A. Hirtle* DATE *March 26, 1986*  
A. Hirtle  
MSC&T Systems

APPROVED BY *R. Farrell* DATE *April 16, 1986*  
R. Farrell, Manager  
Communications Eng'g

APPROVED BY *D. Devoy* DATE *April 17, 1986*  
D. Devoy, Director  
MSC&T Development

