Copyright Warning & Restrictions

The copyright law of the United States (Title 17, United States Code) governs the making of photocopies or other reproductions of copyrighted material.

Under certain conditions specified in the law, libraries and archives are authorized to furnish a photocopy or other reproduction. One of these specified conditions is that the photocopy or reproduction is not to be “used for any purpose other than private study, scholarship, or research.” If a, user makes a request for, or later uses, a photocopy or reproduction for purposes in excess of “fair use” that user may be liable for copyright infringement,

This institution reserves the right to refuse to accept a copying order if, in its judgment, fulfillment of the order would involve violation of copyright law.

Please Note: The author retains the copyright while the New Jersey Institute of Technology reserves the right to distribute this thesis or dissertation

Printing note: If you do not wish to print this page, then select “Pages from: first page # to: last page #” on the print dialog screen

NJIT
New Jersey's Science & Technology University

The Van Houten library has removed some of the personal information and all signatures from the approval page and biographical sketches of theses and dissertations in order to protect the identity of NJIT graduates and faculty.

ABSTRACT

DESIGN OF A LAPD INTERFACE USING THE T7130 MULTICHANNEL LAPD CONTROLLER AND THE T7115A SYNCHRONOUS PROTOCOL DATA FORMATTER

by
Hopeton Saint John Walker

As the ISDN telecommunications standard gains acceptance, there is an ever increasing need for intelligent interfaces to implement the L2 protocol known as LAPD. The T7130 MLC, the T7115A SPYDER-T and the MC68020 microprocessor were used to design a LAPD Interface that conforms to the LAPD protocol as specified by CCITT.

The LAPD Interface utilizes a Shared Memory Array which is attached to the Data Link Processors in a single bus configuration. The SMA Arbitration Control allows access to the SMA on a prioritized basis and was implemented as a state machine using PAL's. L2 drivers and L3 management software were written and used to test the operability of the interface via an emulator.

The LAPD Interface was able to terminate 32 HDLC channels configured for LAPD operation with an average throughput of 942.42 messages per second. The LAPD Interface was also able to operate efficiently in an environment in which random errors of the order of 1E-6 were injected.

DESIGN OF A LAPD INTERFACE USING THE T7130 MULTICHANNEL LAPD CONTROLLER AND THE T7115A SYNCHRONOUS PROTOCOL DATA FORMATTER

by
Hopeton Saint John Walker

Robert W. Van Houten Library
New Jersey Institute of Technology

A Thesis
Submitted to the Faculty of
New Jersey Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of
Master of Science in Electrical Engineering

Department of Electrical and Computer Engineering,

January 1995

APPROVAL PAGE

DESIGN OF A LAPD INTERFACE USING THE T7130 MULTICHANNEL LAPD CONTROLLER AND THE T7115A SYNCHRONOUS PROTOCOL DATA FORMATTER

Hopeton Saint John Walker

Dr. Constantine N. Manikopoulos, Thesis Advisor — Date
Associate Professor of Electrical and Computer Engineering, NJIT

Dr. Sotirios Ziavras — Date
Assistant Professor of Electrical and Computer Engineering, NJIT

Dr. Anthony D. Robbi — Date
Associate Professor of Electrical and Computer Engineering, NJIT

BIOGRAPHICAL SKETCH

Author: Hopeton Saint John Walker

Degree: Bachelor of Science in Electrical Engineering

Date: January 1995

Undergraduate and Graduate Education:

- Master of Science in Electrical Engineering,
  New Jersey Institute of Technology,
  Newark, New Jersey, 1995

- Bachelor of Science in Electrical Engineering,
  New Jersey Institute of Technology,
  Newark, New Jersey, 1993

Major: Electrical Engineering

This thesis is dedicated to Herbert Edward Walker II.

Oh how we wondered and wished for you ....
You came too soon and in a flash you were gone.
.... Surely, your memory lives on.

ACKNOWLEDGMENT

The author wishes to thank Dr. Mannikopoulos for his sincere gratitude, guidance friendship and moral support. Throughout my latter years at NJIT, he has provided advice that has lead to my academic and professional growth.

Special thanks to my wife, Shelley Walters-Walker, for her constant and unwavering encouragement, faith, sympathy, good cheer and moral support during the trying periods marked by my determination to complete this thesis.

The author would also like to thank Sheridan Quarless, Juan Segura, William Wong, Jitendra Patel, Steve Christie, Carl Gauntlett, Florencio Martinez, Mariano Lalumia, Errol Drummond, Harold Shichman and Chu Le.

TABLE OF CONTENTS

TABLE OF CONTENTS
(Continued)

TABLE OF CONTENTS
(Continued)

TABLE OF CONTENTS
(Continued)

Chapter Page

TABLE OF CONTENTS
(Continued)

TABLE OF CONTENTS
(Continued)

LIST OF TABLES

ACRONYMS

ACB	Arbitration Control Block
BET	Bus Error Timer
BG	Bus Grant
BR	Bus Request
CB	Control Blocks
CQ	Command Queue
CMOS	Complementary Metal Oxide Semiconductor
CR	Control Register
CRC	Cyclic Redundancy Check
CSA	Common Shared Memory Array
DL	Data Link
DLP	Data Link Processor
DLCI	Data Link Control Identifier
DMA	Direct Memory Access
DSW	Diagnostic Status Word
DTACK	Data Transfer ACKnowledge
EEPROM	Electrically Erasable Programmable Read Only Memory
ETSI	European Telecommunications Standards Institute
EX_CPU	External Central Processing Unit
FCS	Frame Check Sequence
HDLC	High Level Data Link Control
HOST	MC68020 Microprocessor
HSB	HDLC Statistics Block
IACK	Interrupt ACKnowledge
ICB	Interrupt Control Block
IQ	Interrupt Queue
IRR	Interrupt Request Register
ISDN	Integrated Services Didital Network
ISR	Interrupt Status Register
L2	Layer 3
L3	Layer 3
LAP	Link Access Procedure
LAPB	Link Access Procedure on the B-Channel
LAPD	Link Access Procedure on the D-Channel
MLC	Multichannel LAPD Controller (T7130)
NVM	Non-Volatile Memory
PLD	Programmable Logic Device
RFA	Remote Frame Alignment
RG	Request Grant

ACRONYMS
(Continued)

RD	Receive Descriptor
SA	SPYDER-T Attention
SAPI	Service Access Point Identifier
SM	Shared Memory
SMA	Shared Memory Array
SMBR	Shared Memory Bus Request
SPYDER-T	Synchronous Protocol Data Formatter (T7115A)
SRAM	Static Random Access Memory
TD	Transmit Descriptor
TEI	Termination Endpoint Identifier
TQ	Transmit Queue
VLSI	Very Large Scale Integrated

CHAPTER 1

LAPD HARDWARE SPECIFICATIONS

1.1 Data Link Processors - T7130 and T7115A

1.1.1 Function

The Data Link Processors (DLP), provides the termination point for the LAPD protocol and consists of two CMOS VLSI devices. Namely, the T7115 Synchronous Protocol Data Formatter (SPYDER-T) and the T7130 Multichannel LAPD Controller (MLC). These devices are intelligent peripherals to the Motorola MC68020 microprocessor (HOST) and provide the hardware necessary to perform the complete Level 2 (L2) link layer protocol, Link Access Procedure - D Channel (LAPD). The T7130, T7115A and MC68020 exchange all commands and status through common areas located in SRAM within the Shared Memory Array (SMA). These areas include the Control Blocks (CB), Command Queue (CQ) and the Interrupt Queue (CQ). The T7115 and T7130 both have DMA capability. Communication between the SPYDER-T and MLC also occurs through common data structures[1] contained in the LAPD Interface Shared Memory Array. The SPYDER-T and MLC are memory mapped inside the MC68020 I/O space through the Control Register (CR0) for the purpose of reset and initialization.

[1] A more detail description of these data structures can be found in the T7115A and T7130 data sheets.

1.1.2 Synchronous Protocol Data Formatter - SPYDER-T

The LAPD protocol uses two delimiters, called flags, to identify frame boundaries. A Frame Check Sequence (FCS) field is included for the error detection code and is calculated from all the bits in the LAPD frame exclusive of the flags. As each packet of a LAPD data channel is transferred to the T7115A and T7130 the SPYDER-T performs low level formatting functions. This low level formatting corresponds to,

1. insertion and detection of the LAPD header and tail flags, and
2. calculation of the Cyclic Redundancy Check (CRC) code that is used for either
   - comparison with the received FCS (for error reporting), or
   - to append to outgoing message frames as the FCS field on each activated LAPD data channel.

The SPYDER-T does not report that a complete LAPD message has been received until all bytes between the header and tail flags are received without any CRC errors and the address, control and information fields have been transferred to receive buffers in SRAM within the Shared Memory Array (SMA). The SPYDER-T has limited on-chip buffering capability and must periodically performs a two byte DMA transfer[1] on the received data.

In the transmit direction, the SPYDER-T begins transmission of a LAPD frame with the header flag, reads the address, control and information bytes from memory, then appends the CRC bytes and closing flag. Whenever a complete frame is received or transmitted by the SPYDER-T, it updates selected bits in the shared data structure between the SPYDER-T and MLC. This shared data structure is referred to as the T7115

[1] The SPYDER-T must perform the DMA transfer to shared memory within six clock cycles (16.67MHZ clock) after a request has been initiated.

Interrupt Queue (T7115A-IQ) and contains information about the status of each LAPD channel. The Interrupt Queue is polled[2] by the MLC and indicates which incoming or outgoing LAPD channel is ready for further Level-2 (L2) processing by the MLC. Although the SPYDER-T has on-chip DMA capability, it does not have a transfer acknowledge mechanism to ensure that a I/O cycle has been completed, such as the DTACK input signal commonly found on the MC680XX family of microprocessors. Consequently, external logic has been included in the T7130 and T7115A block to detect for any illegal[3] SPYDER-T address outside valid SMA space.

1.1.3 Multichannel LAPD Controller - T7130

The T7130 Multichannel LAPD Controller manages all Level-2 Data Link (DL) functions and processes the LAPD frame header, address and control fields. In the receive direction, these fields are parsed by the MLC whereas in the transmit direction, these fields are built by the T7130 and appended to the outgoing LAPD frame.

The address field is two bytes wide and provides two levels of multiplexing specific for each LAPD frame. The first level of multiplexing discriminates between Terminal End Points when multiple users are sharing the same physical interface. For example, a single LAPD channel allocated between point A and point B, can serve as the Data Link for multiple transceivers. In order to identify a single transceiver at point B, a unique 7-bit number (modulo 128) referred to as a Termination Endpoint Identifier (TEI), is assigned to each transceiver and is part of the address field. The second level of multiplexing is concerned with identifying the type of traffic message that is contained in

[2] The T7115-T7130 polling is interrupt driven.

[3] A SPYDER-T Illegal address interrupt is generated and sent to the LAPD Interface Interrupt Control Block.

the information field for each TEI. Packet data, call control procedure, and message flow control are a few examples. The Service Access Point Identifier (SAPI), a 6-bit field contained in the address bytes, provides the second level of multiplexing. Examples of TEI and SAPI assignments are shown in table 1. The Data Link Control Identifier (DLCI) is used to uniquely identify a logical connection and consists of a TEI/SAPI pair.

Table 1
Examples of TEI and SAPI assignments

TEI VALUE	
0-63	TEI not automatically assigned
127	Used during auto TEI assignment
SAPI VALUE	
0	Call Control Procedure - Signaling
63	L2 Management Procedure

The control field of the LAPD frame consists of either one or two bytes identifying the LAPD message as one of three types: S-frame, U-frame, or I-frame. The first two types provide supervisory and control functions for the Data Link, while Level-3 (L3) data is carried in I-frames. The T7130 administers all message flow and error control through the S and U frames without intervention from the HOST processor.

After processing the message frame headers, the MLC passes any Level-3 or management entity information to the HOST by way of the T7130 Interrupt Queue (IQ) in the LAPD Common SRAM[4] Array (CSA). The IQ permits a handshaking mechanism between the HOST and MLC for the purpose of exchanging message pointers.

All commands necessary to activate selected LAPD channels and initialize a Data Link are provided by the HOST through the MLC Command Queue (CQ). The HOST writes commands to the CQ which subsequently controls all Data Link operations, including the actual establishment and management of a Data Link. All this is done autonomously by the MLC.

1.1.4 T7130, T7115A and HOST Interface

The T7115A, T7130 and HOST exchange all commands and status information through common shared data structures. This information exchange process is interrupt driven primarily to reduce the Shared Memory Array I/O bus occupancy of the T7130. In the polling mode, the T7130 and HOST must constantly request the LAPD Interface shared memory resource area where Command and Interrupt Queues are located. The T7130 is interrupted whenever,

1. the T7115 updates the T7115 Interrupt Queue (T7115-IQ), or
2. the HOST updates the T7130 Command Queue (T7130-CQ).

Upon receiving an interrupt from either of the above two sources, the T7130 reads the T7115-IQ first, performs the required actions, and then reads the T7130-CQ and executes the host commands. The HOST, on the other hand, is interrupted whenever the T7130 updates it's Interrupt Queue (T7130-IQ). This DLP HOST directed interrupt is processed through the Interrupt Control Block (ICB).

[4] The SRAM devices in the SMA do not introduce any wait states for the T7130.

1.1.5 Multichannel LAPD Controller Private I/O Port

A 16-bit address, 24-bit data I/O port on the MLC provides access to the Level-2 LAPD protocol code that is contained in fast SRAM and executed by the T7130. This I/O port is accessible only by the T7130. A read or write to private memory is performed once per MLC clock cycle. The MLC operates with a 50 ns clock period (20MHZ). The MLC executes all the LAPD code from this private memory and this scheme provides for modifications to be made to the LAPD code without complications. The LAPD code resides in EEPROM, located in the SMA and must be downloaded to the T7130 private memory during initialization. As a result, changes to the LAPD code is possible through an update of this LAPD program memory.

1.1.6 Initialization of the Data Link Processor

The initialization of the T7130 and T7115A is performed by the HOST on power-up. At this point, both the SPYDER-T and MLC are held in a reset state until the MC68020 completes any applicable diagnostic routine. On successful completion, the HOST

1. initializes (*formats*) the data structures in shared memory which are used by the DLP.
2. removes the reset to the SPYDER-T and MLC.

After removal of the reset, the SPYDER-T remains in an idle mode while the MLC begins the following initialization procedure,

1. downloads the LAPD code[5] from EEPROM to the private SRAM memory.

[5] The EEPROM containing the LAPD code resides in the Shared Memory Array. Hardware write protection of the LAPD EEPROM is provided by the HOST

2. verify the integrity of the LAPD code in SRAM by computing a 24-bit Cyclic Redundancy Checksum (CRC).
3. begin execution of the LAPD code in private memory.

Since the SPYDER-T is in the idle state and has not been activated, no status information is exchanged between the T7115 and T7130, through the T7115-IQ. To complete initialization of the T7130 and T7115A, the HOST must pulse the SPYDER-T Attention pin (SA) which causes the SPYDER-T to load internal DMA registers with a pointer[6], to access instructions in shared memory and begin Data Link operations.

1.2 HOST - MC68020

1.2.1 Function

The Motorola MC68020 features a 32-bit address and 32-bit data bus and operates at 33.33MHZ[7]. The motivation for using the MC68020 is twofold. The first is the fast memory access[8] and the second is the ability for long word (32-bit) access. Instruction and data fetches can take advantage of long word access. The HOST is supported by EEPROM, SRAM and several peripheral devices summarized in the following list

- 256K bytes of EEPROM for program memory
- 256K bytes of "shadow" SRAM for program memory
- 256K bytes of SRAM for data memory
- 64K bytes of EEPROM[9] for the LAPD code

[6] SPYDER-T Configuration Pointer

[7] 30 ns clock cycle

[8] 4 clock cycles for an I/O access

[9] This memory space is also I/O mapped into the T7130 I/O space for the purpose of down loading the LAPD code into the private memory of the T7130

- Interrupt Control Devices

  MC68153 (BIM - Interrupt Control)

  MC68901 (MFP - Interrupt Control and Timer).

  Interrupt Status Register (ISR)

  Interrupt Request Register (IRR)

- Control Registers

  CR0

  CR1

- HDLC Interface Device

  AT&T T7121 (HIFI-64)

All peripheral devices and memory, including the SMA, are mapped in the HOST I/O space. Control register, CR0, is used for reset and initialization of the DLP and peripheral devices. Write protection[10] of program memory space contained in both EEPROM and SRAM is set through the control register CR1. The status of any write violations[11] are contained in the ISR.

Interrupts from the HOST directed to either the T7130, T7115A or an External Processor (EX_CPU) are possible through the IRR. All program and data memory (contained in fast SRAM) are configured as a 32-bit (*longword*) port which is designed for *zero* wait states. All EEPROM program memory is configured as a word port and the peripherals configured as a byte port.

[10] The EEPROMS are also protected with a software key whereas SRAM is protected only through the write enable bit in control register CR1

[11] The source of the violation

The HOST performs three major functions:

1. LAPD Interface Initialization and any applicable diagnostics
2. Level-2 Management
3. Level-3 Management

1.2.2 HOST Initialization and Diagnostics

The HOST can perform a series of diagnostics and initialization tests that are executed immediately following a power-up or reset. A failure in anyone of the tests could indicate that the Interface requires service. Non-destructive diagnostics can also run during normal operation. These may include memory tests.

1.2.3 Level-2 Management of the LAPD Data Links

The Level-2 programs that manage Data Link operations[12] reside in EEPROM and are executed by the HOST in order to provide a stable link connection. The control and return status of the DLP is accomplished through shared data structures in memory and the HOST administers the format of these structures. The T7130 and T7115A are permitted to activate LAPD data channels after the HOST completes any relevant diagnostics.

1.2.4 Level-3 Management

To assist with Level-3 (L3) management routines, a T7121 device has been added to provide a HDLC communication path which can assist in the processing and/or dispatching of Operation and Link Maintenance.

[12] TEI management, connection management, flow control, etc

1.3 Shared Memory Array

1.3.1 Function

The Shared Memory Array (SMA) serves as the common memory resource for the LAPD Interface. The SMA is a single port (16-bit) memory array consisting of both EEPROM and SRAM devices. Since the SMA bus is limited to only one I/O transfer at a time, access management is performed by the Arbitration Control Function Block. The SMA couples the LAPD protocol devices[13] and HOST along with any other external interface that may be added to complement a system. External devices may include an External CPU (EX_CPU).

The SRAM functions as a shared memory resource for the T7115, T7130 and HOST, and is referred to as the Common Shared SRAM Array (CSA). The CSA provides a total of 512 Kbytes of storage. The CSA is mapped into the I/O space of the HOST, T7130 and T7115A. EEPROM devices included in the SMA are accessible only by the T7130 and HOST. All communication between the HOST and LAPD devices is through data structures contained in the CSA, within the SMA. The SRAM in the CSA does not introduce any wait states for either the T7130 or T7115.

This common SRAM resource functions as the central depository for all L3 messages, associated L2 message flow and error control, and LAPD protocol state variables. Other type data structures include, Receive Descriptors (RD) and Transmit Descriptors (TD). These provide buffers and message pointers for the status of all activated LAPD channels. The instruction primitives required for the DLP are contained

[13] T7115 (SPYDER-T), T7130 (MLC)

in Command Queues (CQ) while the Interrupt Queues (IQ) serve as a handshake mechanism between the T7130, T7115A, HOST and any external processor.

In addition to SRAM, the SMA also contains 64K bytes of EEPROM which is programmed with the LAPD protocol code[14] executed by the T7130. Only the T7130 and HOST have access to the EEPROM devices contained in the SMA. The MLC can only read the EEPROM devices whereas the HOST can write new LAPD code by execution of an update routine.

1.3.2 Shared Memory Array Initialization

The format for all the data structures are defined in the data sheets of the LAPD protocol devices and is administered by the HOST during any initialization of the LAPD Interface. Initialization of the CSA follows immediately after the HOST completes any applicable diagnostics. The HOST then begins to configure (*format*) the data structures and upon completion of the formatting process, the HOST removes the reset from the MLC and issues a SPYDER Attention (SA) instruction which initiates the request for establishment of a LAPD data-link.

1.3.3 Shared Memory Request - DMA Level Assignment

The SPYDER-T (T7115) has limited on-chip buffers and has the most critical DMA requirement for memory access otherwise corruption of activated LAPD channels will occur. The next highest level has been assigned to any External CPU (EX_CPU) which prevents the EX_CPU from excessive waiting times when requesting access into the LAPD Interface Shared Memory Array. The HOST obtains the SMA resource

[14] The LAPD code is transferred at initialization to private memory of the T7130

(EEPROM or SRAM) whenever the SPYDER-T or an EX_CPU are not in the process of requesting access to the SMA. There is no critical DMA requirement for the T7130 which is assigned the lowest priority for access to the SMA.

1.4 Arbitration Control

1.4.1 Function

The Arbitration Control (AC) block of the HOST processes all requests for SMA access from the SPYDER-T, MLC, HOST and any EX_CPU. Communication among these processors is through the SMA. Access to the SMA is through a single 16-bit I/O port and only one device can access the common memory resource at any given time. The AC block provides the mechanism that manage all accesses to either SRAM in the CSA or to the EEPROM devices containing the LAPD code. The selection process that determines which device is allowed access into the SMA is based upon critical DMA times and the order of priority is presented in table 2. The Arbitration Control block is initialized through the power-on reset which clears all Bus Grants (BG).

Table 2
Shared Memory Array DMA Level Assignment

PRIORITY	DEVICE
1	SPYDER-T
2	EX_CPU
3	HOST
4	MLC

1=Highest Priority

1.5 Interrupt Control

1.5.1 Function

There are two interrupt control paths that are designed for the Interface. These include:

1. the interrupt requests for HOST service from on-board peripheral devices, or from an EX_CPU, and
2. the interrupt requests for an EX_CPU service from the HOST.

1.5.2 Interrupt Requests for HOST Service

All interrupt requests to the HOST are handled by the MC68153 Bus Interrupt Module (BIM) and the MC68901 Multi-Function Peripheral (MFP). Interrupt Requests are terminated and prioritized by the BIM and MFP. The BIM provides interrupt request service for four external devices and the MFP provides interrupt request service for eight external and eight internal interrupts. One BIM interrupt channel input, CH0, is dedicated to the MFP. The interrupt request output of the MFP which represents the state of any MFP interrupt source is connected to BIM CH0 allowing all of the MFP interrupt requests to be grouped into a single level before being presented to the HOST. The interrupt requests for the HOST are shown in table 3.

Even though all of the MFP interrupts are grouped at the same level, they are assigned to individual levels within the MFP. Both devices also support programmable level assignment and mask enables for each interrupt request.

Table 3
Interrupt Requests to HOST

Interrupts serviced by the BIM	
Interrupt	BIM Channel
Memory Write Violations	CH3
SPYDER-T Illegal Address	CH2
Task Scheduler (Timer)	CH1
MFP	CH0
Interrupts serviced by the MFP	
Interrupt	MFP Channel
EX_CPU	CH7
MLC Link Interrupt	CH6
HDLC Interrupt	CH5
SPARE	CH4
SPARE	CH3
SPARE	CH2
SPARE	CH1
SPARE	CH0

1.5.3 BIM Interrupts

All HOST memory write violations are assigned to BIM input channel CH3. These interrupts result from the HOST attempting to write to either local program memory (*EEPROM or SRAM*) or to the T7130 LAPD protocol memory (*EEPROM contained in the SMA*) without the proper write enable bit set. These enables are contained in Control Register 1 (CR1). In order to determine which write violation caused the BIM channel CH3 to be activated, the HOST must read the Interrupt Status Register (ISR), which contains all the write violation status flags.

BIM channel CH2 indicates any illegal SPYDER-T DMA address outside valid SMA space and the 10msec Timer or Task Scheduler is assigned to CH1. The MFP is assigned BIM channel CH0.

1.5.4 MFP Interrupts

All data link related interrupts from the T7130 MLC are assigned to channel CH6 of the MFP. Whenever the T7130 updates it's interrupt queue, an interrupt to the MFP is generated. The HDLC interface device (HIFI-64) is assigned to MFP channel CH5 and indicates a full or empty FIFO condition on the HIFI-64 device which provides the private HDLC data link between the LAPD Interface and other external entity[15].

Channel 7 of the MFP can be memory mapped into the EX_CPU I/O space and provides for an interrupt driven message interface between the HOST and an EX_CPU. All EX_CPU to HOST (out-going) messages are contained in an out-going message queue in shared memory and whenever the EX_CPU writes any message for the HOST in this queue, the EX_CPU has the option of interrupting the HOST through this MFP channel.

1.5.5 Exception Vectors

The vectors for BIM channels CH3, CH2 and CH1 are supplied by the BIM while the MFP provides all the interrupt vectors for BIM channel CH3. The Interrupt handlers are in a daisy chain configuration with the BIM first in the chain followed by the MFP. All interrupt vectors must be loaded into the interrupt handlers BIM and MFP by the HOST during initialization. The HOST obtains the exception vectors during executing an Interrupt ACKnowledge (IACK) cycle and individual interrupt requests that caused the exception is cleared during the IACK cycle[16].

[15] eg SS7

[16] (With the exception of the Write Violations (BIM CH0); the HOST must clear these write violation sources with a write command to the Interrupt Status register, (ISR).

1.5.6 Interrupt Requests for External CPU service

The LAPD Interface provides an interrupt path from the HOST to an EX_CPU. The LAPD Interface can generate a level six interrupt request to the EX_CPU. An EX_CPU interrupt acknowledge cycle must clear the interrupt request and obtains the exception vector from the interrupt vector register. The HOST interrupt vector is loaded during initialization.

The LAPD Interface Interrupt Request is part of the interrupt driven message exchange protocol between the HOST and EX_CPU. All HOST to EX_CPU (incoming) messages are contained in an incoming message queue in the CSA and whenever the HOST writes any message for the EX_CPU in this queue, the HOST has the option of interrupting the EX_CPU through this interrupt path.

1.6 LAPD Interface Clocks

1.6.1 Function

The LAPD Interface requires various clock sources, eight are derived from three resident crystal oscillators. All crystal oscillators are enabled from a tri-state control.

Table 4
LAPD Interface Clock Requirements

Device	Frequency	Source
HOST	33.33MHZ	66.66 MHZ Oscillator
T7115 SPYDER-T	16.67 MHZ	
Arbitration Circuit	40.00 MHZ	40.00 MHZ Oscillator
T7130MLC	20.00 MHZ	
MC68901	4.00 MHZ	4.00 MHZ Oscillator
T7115-CHI	8.0 KHZ	
T7121-CHI	2.048 MHZ	

1.7 HOST to HDLC Private Communication Link

1.7.1 Function

The Interface supports a private serial HDLC communication link between the LAPD Interface and any other subsystem that requires such service. The AT&T T7121 HDLC interface for ISDN (HIFI-64), provides this feature. The HIFI-64 is memory mapped inside the HOST I/O space and is controlled through various internal registers accessible by the HOST. The device contains a 64-byte FIFO which significantly reduces the number of interrupts that must be processed by the HOST.

1.8 Registers

1.8.1 HOST Device Interrupt Request Register

The Interrupt Request Register (IRR) is located at address xx3xxxf0. Table 5 shows the bit positions, names and read or write status of the relevant bits. All undefined bit positions are don't cares.

Table 5
Interrupt Request Register

31	30	29	28	27	26	25	24
HOST to EX_CPU INT REQ	HOST to MLC INT REQ	undefined	undefined	undefined	undefined	undefined	undefined
R/W	W						

HOST to EX_CPU INT REQ
This is the INTerrupt REQuest (INT REQ) for an EX_CPU service. Whenever the HOST sets this bit to a logic "0", an interrupt request to the EX_CPU is generated. The interrupt request is enabled through the HOST System Control Register and it is disabled on a reset. Whenever the EX_CPU executes an interrupt acknowledge cycle, the interrupt request bit is cleared.

HOST to MLC INT REQ
This is the INTerrupt REQuest for the T7130 Multi-channel LAPD Controller (MLC INT REQ). Whenever the HOST sets this bit to a logic "0", an interrupt request to the MLC is generated. This is an address and data triggered command ;this register bit position is non-latching. The value read back is not guaranteed to be the same value that was written to bit-30.

1.8.2 HOST Control Register - CR0

The HOST Control Register (CR0) is located at address xx3xxxf2. Table 6 shows the bit positions, names and read or write status of the relevant bits. Bit position 25 in CR0 is undefined.

Table 6
HOST Control Register 0 - CR0

31	30	29	28	27	26	25	24
MFP RST	BIM RST		T7121 RST	T7130 RST	T7115 RST		T7115A SA
R/W	R/W		R/W	R/W	R/W		W

MFP RST
This is the ReSeT control for the MC68901 Multi-function Peripheral MFP.

BIM RST
This is the ReSeT control for the MC68153 Bus Interrupt Module (BIM).

T7121 RST
This is the ReSeT control for the T7121 HDLC (HIFI-64).

T7130 RST
This is the ReSeT control for the T7130 Multi-channel LAPD Controller (MLC).

T7115 RST"
This is the ReSeT control for the T7115 SPYDER-T.

T7115 SA
This is the start-up (initialization) control for the T7115 SPYDER-T. This is an address and data triggered command. As a result, when this bit is set to a logic "0," a pulse is issued to the T7115 SPYDER Attention (SA) pin. No provision for a read back of this bit

is provided. If a read of this bit is attempted, there is no guarantee that this bit is either a logic "0" or "1".

Resets

All resets are a two step process. To reset a peripheral, the processor must first write a zero into the selected bit position (which is the active state for the peripheral device), followed by writing a one which negates the reset control to the peripheral.

1.8.3 HOST Control Register - CR1

The HOST Control Register (CR1) is located at address xx3xxxf4. Table 7 shows the bit positions, names and read or write status of the relevant bits. This register is cleared (all bits set to a logic "0") during a LAPD Interface reset. All spare bits are read and write but do not perform ANY FUNCTION.

Table 7
HOST Control Register 1 - CR1

31	30	29	28	27	26	25	24
EEWEL MP	SRWEL MP	EEWEL LD	UNUSED	UNUSED	UNUSED	UNUSED	UNUSED
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

EEWEN_MP

This is the Write ENable for HOST EEPROM program memory. When the EEWEN_MP bit is set to a logic "1", no write violation interrupt is generated when the HOST attempts to write to this memory space.

SRWEN_MP
This is the Write enable for HOST SRAM for program memory. When the SRWEN_MP bit is set to a logic "1," no interrupt is generated when the HOST attempts to write to this memory space.

EEWEN_LD
This is the Write ENnable for LAPD EEPROM protocol memory contained in the Shared Memory Array. When the EEWEN_LD bit is set to a logic "1", no interrupt is generated when the HOST attempts to write to this memory space.

1.8.4 HOST Interrupt Status Register

The HOST Interrupt Status Register is located at address xx3xxxf6. Table 8 shows the bit positions, names and read or write status of the relevant bits. All undefined bit positions are don't cares. No provision is made to clear individual bits in this register. A clear (all write violations set to a logic "0") is performed by addressing this register while in the write mode.

Table 8
HOST Interrupt Status Register

31	30	29	28	27	26	25	24
EEWVL MP	SRWVL MP	EEWVL LD	undefined	undefined	undefined	undefined	undefined
R/W	R/W	R/W					

EEWV_MP
This is the Write Violation status flag for the HOST EEPROM. When this bit is read as a logic "1", this indicates an attempted write to HOST EEPROM program memory has occurred without the corresponding write enable bit set. The EEWEN_MP bit in CR1

controls the state of the write violation status flag. If EEWEN_MP is set to a logic "1," the HOST can write to EEPROM space and no write violation interrupt is generated.

SRWV_MP
This is the Write Violation status flag for the HOST SRAM. When this bit is read as a logic "1", this indicates an attempted write to HOST SRAM program memory has occurred without the corresponding write enable bit set. The SRWEN_MP bit in CR1 controls the state of the write violation status flag. If SRWEN_MP is set to a logic "1," the HOST can write to SRAM space and no write violation interrupt is generated.

EEWV_LD
Write Violation status flag for the HOST LAPD EEPROM. When this bit is read as a logic "1", this indicates that an attempted write to LAPD protocol memory EEPROM in the SMA has occurred, without the corresponding write enable bit set. The LDWEN_MP bit in CR1 controls the state of the write violation status flag. If LDWEN_MP is set to a logic "1", the HOST can write to LAPD protocol memory and no write violation interrupt is generated.

1.8.5 HOST Exception Vector

Table 9 shows the bit positions, names and read or write status of the relevant bits for the HOST Exception Vector Register. All undefined bit positions are don't cares.

Table 9
HOST Exception Vector Register

			Upper Byte				
31	30	29	28	27	26	25	24
undefined	undefined	undefined	undefined	undefined	undefined	undefined	undefined
			Upper Middle Byte				
23	22	21	20	19	18	17	16
CPFC2	CPFC1	CPFC0	HOST RST	undefined	undefined	undefined	undefined
R	R	R	R				
			Lower Middle Byte				
15	14	13	12	11	10	9	8
undefined	undefined	undefined	undefined	undefined	undefined	undefined	undefined
			Lower Byte				
7	6	5	4	3	2	1	0
undefined	undefined	undefined	undefined	undefined	undefined	undefined	undefined
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

CPFC0-CPFC2

The three bits, CPFC0, CPFC1, and CPFC2 are the function code of the host and identifies the address space of the current bus cycle.

HOST RST

The HOST ReSeT bit indicates the state of the HOST (MC68020) reset pin.

Interrupt Vector

This is the HOST Exception Vector. The exception vector is loaded by the EX_CPU during system initialization.

1.8.6 System Control Register

Table 10 shows the bit positions, names and read or write status of the relevant bits for the System Control Register. All undefined bit positions are don't cares.

Table 10
System Control Register

Upper Byte

31	30	29	28	27	26	25	24
HOST INT	HOST HLT	HOST INTEN	UNUSED	UNUSED	UNUSED	UNUSED	UNUSED
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Upper Middle Byte

23	22	21	20	19	18	17	16
undefined	undefined	undefined	undefined	undefined	undefined	undefined	undefined

Lower Middle Byte

15	14	13	12	11	10	9	8
undefined	undefined	undefined	undefined	undefined	undefined	undefined	undefined

Lower Byte

7	6	5	4	3	2	1	0
undefined	undefined	undefined	undefined	undefined	undefined	undefined	undefined

HOST RST

The HOST ReSeT bit controls the reset function of the MC68020. To reset the HOST, this bit is set to a logic "1". The reset process is s single step process. To reset the HOST, bit 31 is set to a logic "1". Circuitry on the interface will clear all bits in the Control Register (set to a logic "0") after the processor has cycled through reset.

HOST HLT

The HaLT bit controls the halt function of the MC68020. To halt the HOST, this bit is set to a logic "1". To disable the halt request, the bit must be set to a logic "0".

HOST INTEN

The HOST INTerrupt ENable bit allows a HOST to EXT_CPU interrupt request. The HOST to EX_CPU interrupt is described in the HOST Device Interrupt Request. If this bit is set to a logic "1", the HOST interrupt requests are enabled.

1.8.7 External CPU to HOST Interrupt Request Register

Table 11 shows the bit positions, names and read or write status of the relevant bits for the EX_CPU to HOST Interrupt Request Register. All undefined bit positions are don't cares.

Table 11
External CPU to HOST Interrupt Request Register

Upper Byte

31	30	29	28	27	26	25	24
HOST INT	undefined	undefined	undefined	undefined	undefined	undefined	undefined
W							

Upper Middle Byte

23	22	21	20	19	18	17	16
undefined	undefined	undefined	undefined	undefined	undefined	undefined	undefined

Lower Middle Byte

15	14	13	12	11	10	9	8
undefined	undefined	undefined	undefined	undefined	undefined	undefined	undefined

Lower Byte

7	6	5	4	3	2	1	0
undefined	undefined	undefined	undefined	undefined	undefined	undefined	undefined

HOST INT

The HOST INTerrupt triggers an EX_CPU to HOST interrupt request. This is an address and data triggered command. If this bit is set to a logic “1”, an interrupt request is sent to the HOST through the interrupt control block. (channel 7 of the MFP). If a read of this bit is attempted, there is no guarantee that this bit is either a logic “0” or logic “1”.

1.9 Shared Memory Requirements for LAPD Interface

1.9.1 Calculation of Memory Requirement

The following tables, tables 12 through 13, gives an idea of how to calculate the amount of shared memory required for an interface. Refer to the SPYDER-T and Multi-channel LAPD Controller data sheets for a detailed explanation of the shared data structures.

Table 12
TD and RD Memory Requirements

	Total
T7130 Transmit Descriptor	
(19 + 14) = 33 bytes each	
33 x #LL x Window	13K
T7130 Transmit Data Block	
260-Bytes msg. length	
#LL x Window x 260	100K
T7130 Receive Descriptor	
14 bytes each	
14 x #LL x Window size	5K
T7130 Receive Data Block	
260-Byte msg. length	
#LL x Window x 260	100K
Total	218K

Table 13
Other Shared Memory Data Structure Requirements

Other Shared Memory Data Structures	Bytes
Logical Link Control Blk	
128per Logical link (#LL x 128)	39K
T7130 Command Queue	
8 bytes per # of commands (typ 128 commands)	1K
T7130 Interrupt Queue	
10 bytes per Logical Link	3K
T7130 Passthrough Queue	
2 bytes per other messages (typ 2-4K messages)	8K
Service Access Point Ctl Blk	
256 per channel (32 x 256)	8K
Terminal Endpoint Ctl Blk	
512 per active SPI (512 x 3 x 32)	50K
T7130 Device Ctl Blk Ptr	4
T7130 Device Ctl Blk	82
HDLC Statistics Blk Tbl	512
T7130 Channel Ctl Blk Tbl	2K
Protocol Action Tbl	6K
T7115 Configuration Ptr	4
T7115 Attention Reg	8
T7115 Interrupt Queue Ptr	4
T7115 Interrupt Queue	4K
T7115 Channel Confg Reg	64
T7115 Channel Ctl Blks	512
Total	120K

CHAPTER 2

LAPD CIRCUIT DESCRIPTION

2.1 Circuit Description for the LAPD Interface

2.1.1 Interface Overview

This section presents the circuit description for the LAPD Interface. The T7130 Multi-Channel LAPD Controller and T7115A Synchronous Protocol Data Formatter provides the termination point for the LAPD protocol. These devices are intelligent peripherals to the MC68020 (HOST) microprocessor and provide the hardware necessary to perform the complete Level 2 link layer protocol (LAPD).

The T7130, T7115A and MC68020 exchange all commands and status through common areas, namely, the Control Blocks, Command Queues and Interrupt Queues located in the Shared Memory Array (SMA). The T7115 and T7130 both have DMA capability and communication between them occurs via the common data structures maintained in the Shared Memory Array[1].

The SPYDER-T and MLC are memory mapped inside the MC68020 I/O space through the HOST (MC68020) Control Register CR0 for the purpose of reset and initialization.

[1] A detailed description of these data structures can be found in the T7115 (SPYDER-T) and T7130 (Multi-channel LAPD Controller) data sheets

2.2 SPYDER-T Access to Shared Memory Array

2.2.1 Shared Memory Access

The SPYDER-T device has a 24-bit address and 16-bit data I/O bus and is capable of Direct Memory Access into the SMA. A request to access the SRAM devices within the SMA originates from the SPYDER-T Shared Memory Bus Grant (SPYBRL) and is passed to the Arbitration Control Block (ACB) for processing. The SPYDER-T address and data bus remain tri-state until the SPYDER-T Shared Memory Bus Grant (SPYBGL) becomes active. SPYBGL is also routed to PAL50, which enables the R/W strobes, and detects invalid shared memory SPYDER-T addresses.

Access to the SMA originates from the SPYDER-T Shared Memory Bus Request (SPYBRL) signal and is passed to the Arbitration Control Block (ACB) for processing. The SPYDER-T address and data bus remain tri-state until the SPYDER-T Shared Memory Bus Grant (SPYBGL) becomes active. SPYBGL is also routed to PAL50, which enables the R/W strobes, and detects invalid shared memory SPYDER-T addresses.

The SPYDER-T read/write strobes are always active but are held off the common SMA read/write control bus through logic contained in PAL50. Any DMA device requesting access into the SMA is not connected to the common address, data or read/write lines in the SMA without a valid Bus Grant (BG).

During a read cycle, the SPYDER-T latches data on the rising edge of the read strobe (SPYRDL) indicating the end of the read cycle. Shared Memory chip select signals are consequently negated - the data hold-time for SPYRDL is thus 0 ns. During a write cycle, the SMA write strobes (SMWEL and SMWOL) are controlled directly by the

SPYDER-T write enable, SPYWEL. PAL50 also provides the signal SPYDER-T Shared Memory Space (SPYSMSPL) which is decoded using address bits 23 through 19 and a valid T7115 shared memory bus grant. SPYSMSPL and SPYASL are routed to another PLD, PAL36, where SPYASL is used as the clock for a latch whose input is SPYSMSPL. If an invalid SPYDER-T address has been decoded the latch output signal, SPYDER-T Illegal Address, SPYIAL, will be a logic "0", otherwise it will be a logic "1". SPYIAL is directly connected to channel 1 (CH1) on the Bus Interrupt Module (BIM) in the Interrupt Control Block, which generates a SPYDER-T Illegal Address interrupt to the HOST, whenever SPYIAL is active (equal to "0"). These devices keep the T7115A from either "picking-up or dumping" bits. The T7115 device performs only 16-bit word access in the Common SRAM Array (CSA).

The SPYDER-T operates at a clock frequency of 16.67 MHZ. The T7115 clock is derived from the 66.66 MHZ oscillator. A divide by four circuit is used to generate SPYCLK.

2.3 Multi-Channel LAPD Controller - T7130

2.3.1 Function

The Multi-channel LAPD Controller T7130 (MLC) manages all Level 2 Data Link functions for the interface. The T7130 has the ability for Direct Memory Access (DMA) through two external I/O interfaces and operates at 20MHZ. The MLC can access SRAM or EEPROM devices located in the Shared Memory Array. The LAPD protocol code which is executed by the T7130, resides in the EEPROM devices, whereas all the LAPD message and related data structures reside in the SRAM devices.

2.3.2 Multi-Channel LAPD Controller External I/O Port - Access to Shared Memory Array

Access to the SMA is through the 24-bit address, 16-bit data primary I/O port. The address and data lines of the MLC are directly connected to the common SMA address (SMABxx) and data (SMDBxx) bus. All address/data lines are tri-state when the T7130 is not accessing any memory devices, namely SRAM or EEPROM, in the SMA. The MLC operates in the *slave* mode and has the lowest priority for SMA access. The T7130 signals MLCSMBRL[1] and MLCSMBGL[2] are used in the slave mode configuration, to indicate a MLC SMA Bus Request (BR) and Bus-Grant Acknowledgment from the ACB. The MLC can begin external I/O transfers into the SMA when it receives the valid bus-grant signal (MLCSMBGL) from the ACB.

2.3.3 Multi-Channel LAPD Controller External I/O Port - Access to SRAM Devices in the Shared Memory Array

The Common Shared SRAM Array (CSA) inside the SMA is fast enough to accommodate the two cycle external I/O access requirement of the MLC. The MLC requires that memory provide valid data within 10 ns from the MLC output enable signal, OEL. These fast SRAM devices do not introduce any wait-states for the T7130 when an I/O access is allowed to proceed. The T7130 data sheet gives a complete description of the external I/O timing.

[1] MLC Shared Memory Bus Request

[2] MLC Shared Memory Bus Grant

The MLC read and write strobes are also directly connected to the SMA R/W control bus (SMROL, SMREL, SMWOL, SMWEL) since these signal remain tri-state until a valid MLC Bus Grant.

2.3.4 External I/O Port - Access to EEPROM Devices in the Shared Memory Array

The LAPD protocol code that is executed by the MLC out of private memory resides in the SMA in two 32K x 8 EEPROM devices. These EEPROM devices are accessible only by the MLC and HOST. PAL18 decodes chip selects for the EEPROM devices from the HOST or MLC address lines. Both processors must issue a Shared Memory Bus Request (SMBR) before being granted access into EEPROM space. At initialization, the MLC will transfer the LAPD protocol code stored in EEPROM to the T7130's private memory. The MLC has only *read* permission of EEPROM, whereas the HOST has both *write* and *read* permission. As a result, changes to the LAPD code is possible through a HOST update of the EEPROM devices.

A write violation is generated whenever the HOST attempts to write to the EEPROM without the proper write enable bit set. HOST control register CR1, contains the EEPROM write enable bit (LDEWEH). Logic in PAL30 decodes the valid HOST write condition for the LAPD EEPROM and provides the write violation signal CPEEWVL. CPEEWVL is routed to the HOST Interrupt Status Register (ISR) which contain write violation status flags of all HOST write violations.

2.3.5 External I/O Port - Bus Error, SMA Chip Selects and T7130 DTACK

The MLC device is supported by a Bus Error Timer (BET) and three PLD's, PAL10, PAL18, and PAL54, which provides support for a variety of control signals. PAL18 decodes all SMA Chip Select signals, both SRAM and EEPROM, from MLC address bits 23 through 19 qualified with T7130 Address Strobe (MLCASL) and a valid Shared Memory Array Bus Grant (MLCBGL). The CSA address bit (SMAB18) provides the actual upper/lower chip selection but must be enabled by a valid address and SMA Bus Grant.

The MLC Data Transfer Acknowledgment signal (MLCDTACKL) is required to terminate the MLC I/O bus cycle. For access into the CSA, MLCDTACKL is decoded from address bits MLCA23 through 19 qualified with MLCASL and the valid MLC Shared Memory Bus Grant in PAL18 which provides for the no wait-state condition. In order to accommodate the slower EEPROM devices in the SRAM containing the LAPD protocol code, PAL54 provides a minimum of 250 ns for the MLC. The signal MLCESPL (MLC EEPROM Space) from PAL18 is routed to PAL10 which enables the MLC DTACK bus (MLCDTACKL) and is combined with MLCDTACK1L from PAL54.

Two chips, LS590 and ALS10, in the BET circuitry provides the MLC with a bus error condition in the event of a false address. The counter, LS590, begins counting when the MLCASL is active; a valid MLCDTACKL will reset the counter. In the event that there is no active MLCDTACKL, the bus error will become active approximately 128 MLC clock cycles later. Negating the MLCASL signal will pre-set the latch ALS74 and negate the bus error[3].

[3] Bus-error is active low

2.3.6 Multi-Channel LAPD Controller Private I/O Port

A 16-bit address, 24-bit data I/O port on the MLC provides access to the Level 2 LAPD protocol code executed by the T7130 and which is contained in T7130's private SRAM. This I/O port is accessible only by the T7130. All instruction and data fetches pertaining to execution of the LAPD code is through this private memory. Three fast 32K x 8 SRAM devices, function as the private memory port and are directly connected to the T7130. The memory devices have a maximum access time which is less than or equal to 20 ns and meets the MLC single cycle private I/O memory transfer requirement. The MLC operates with a 50 ns clock period (20MHZ) and a read or write to private memory is performed once per MLC clock cycle. The MLC executes all LAPD code from private memory and this scheme provides for modifications to be made to the LAPD code without complications.

2.3.7 Multi-Channel LAPD Controller Clock Source

A 40 MHZ crystal oscillator provides the MLC clock source. The 40 MHZ oscillator output is divided by two in PAL12, to yield the 20MHZ clock for the T7130. The ACB operates with the same timing source as the T7130. The ACB clock source is the 40 MHZ oscillator output.

2.4 T7130, T7115A and MC68020 Interface

2.4.1 Overview

The T7115, T7130 and HOST exchange all commands and status information through common shared data structures. This information exchange process is interrupt driven primarily to reduce the Shared Memory Array I/O bus occupancy[4] for the T7130.

In the polling mode, the T7130 and HOST must constantly request the Shared Memory resource area where the command and interrupt queues are located. These bus request cycles increase shared memory I/O bus occupancy.

2.4.2 HOST or SPYDER-T to Multi-Channel LAPD Controller

The T7130 is interrupted whenever :

- *case 1* - the T7115 updates the T7115 Interrupt Queue (T7115A-IQ) or
- *case 2* - the HOST updates the T7130 Command Queue (T7130-CQ).

Both the T7115A-IQ and the T7130-CQ are located in the SMA. Whenever the SPYDER-T updates it's IQ, an external interrupt pulse is generated whereas the HOST must execute a write to the Interrupt Request Register (IRR). Upon receiving an interrupt from either *case 1* or *case 2*, the T7130 reads the T7115-IQ first, performs the required actions, and then reads the T7130-CQ and executes the HOST commands. PAL48 provides the logic for the HOST and SPYDER-T to send an interrupt to the MLC device. PAL48 is memory mapped into the HOST space as part of the IRR and decodes the address triggered command into an interrupt pulse to the T7130. The HOST then writes

[4] See T7130 Bus Occupancy Consideration Data Sheet

to location 0xx3xxxf0 to generate an interrupt to the MLC. PAL48 is used to OR both the HOST and SPYDER-T Interrupt Requests to the MLC.

2.4.3 Multi-Channel LAPD Controller to HOST

The HOST is interrupted whenever the T7130 updates it's Interrupt Queue (T7130-IQ). This T7130 and T7115A to HOST directed interrupt is processed through the Interrupt Control Block (ICB). The MLC Interrupt Out Signal, MLCIOUTL, is connected to MFP channel CH6 which sends this interrupt request to the HOST, via the Bus Interrupt Module (BIM).

2.4.4 Initialization of the T7130 and T7115A

The initialization of the T7130 and T7115A is performed by the MC68020 microprocessor through the HOST Control Register, CR0. During power-up, the SPYDER-T and MLC are held in a reset state until the MC68020 completes all power up routines and diagnostics, if any. If everything is successful , the MC68020,

1. initializes the data structures in Shared Memory which are used by the T7130 and T7115A.
2. removes the reset to the SPYDER-T and MLC.

After removal of the reset, the SPYDER-T remains in an idle mode whereas the MLC begins the following initialization procedure,

1. downloads the LAPD code from EEPROM in Shared Memory to the private SRAM memory.

2. verify the integrity of the LAPD code in SRAM by computing a 24-bit Cyclic Redundancy Checksum (CRC).
3. begin execution of the LAPD code in private memory.

Since the SPYDER-T is in the idle state and has not been activated, no status information is exchanged between the T7115 and T7130, through the T7115-IQ. To complete initialization of the T7130 and T7115A, the HOST must pulse the SPYDER-T Attention pin (SA) which causes the SPYDER-T to load internal DMA registers with a pointer[5], and begin DL operations[6]. PAL26 contains the logic that decodes a HOST command to pulse the SPYDER-T. The SA-pulse is an address-triggered command similar to the T7130 interrupt request from the HOST. The HOST writes a zero on data bit 24 in location 0xx3xxxf2, Control Register (CR0) which generates a pulse to the SPYDER-T on the SA pin.

2.5 Shared Memory Array

2.5.1 Structure and Function

The SMA serves as the common memory resource for the LAPD Interface. The SMA is a single port (16-bit) memory array consisting of both EEPROM and SRAM devices. Since the SMA bus is limited to only one I/O transfer at a time, access management is performed by the Arbitration Control Function Block. The SMA couples the LAPD protocol devices (T7115 SPYDER-T, T7130 MLC) and HOST along with any other external interface that may be added to complement a system.

[5] SPYDER-T Configuration Pointer

[6] See the T7130 and T7115A data sheets

The SRAM functions as a shared memory resource for the T7115, T7130 and HOST, and is referred to as the Common Shared SRAM Array, (CSA). The CSA provides a total of 512 kbytes of storage. The CSA is mapped into the I/O space of the HOST, T7130 and T7115A. EEPROM devices included in the SMA are accessible only by the T7130 and HOST. All communication between the HOST and LAPD devices is through data structures contained in the CSA, within the Shared Memory Array. SRAM in the CSA does not introduce any wait states for either the T7130 or T7115.

All level-3 messages, associated L2 message flow and error control, and LAPD protocol state variables reside in the CSA. Other types of data structures include, Receive Descriptors (RD) and Transmit Descriptors (TD), which provide buffer, message pointers and status to all of the LAPD activated channels. The instruction primitives[7] required for the T7130 and T7115A are contained in Command Queues (CQ), while the Interrupt Queues serve as a handshake mechanism between the T7130 and T7115A, HOST and any External CPU.

2.5.2 Common Shared SRAM Array

Four 128K x 8-bit fast SRAM devices are arranged in a 265K x 8-bit *odd* bank and a 256K x 8-bit *even* bank. The odd and even banks create the 16-bit I/O port. Each odd and even shared memory bank is further divided into an upper and lower bank. A common chip select is used for each upper and lower word. Four separate read and write strobes (SMREL, SMWEL, SMROL, and SMWOL) form the CSA R/W bus. This arrangement provides for any combination of byte or word access. When there is no device requesting

[7] A more detail description of these data structures can be found in the *LAPD Software for the T7130 Controller* and *Common Software for the T7130 Controller* data sheets

shared memory resource, all chip selects are actively driven high but the common read/write bus is pulled high through 3.3K resistors connected to +5 volts.

A requesting device can drive the R/W line only after a valid Bus Grant is issued from the arbitration circuit. The shared memory write strobes (SMWEL and SMWOL) are delayed from the chip select signals (SMUCSL, SMLCSL) through a delayed bus grant. The SMA Bus Grant remains active one arbitration clock cycle after a device has negated it's (SMA) request signal which enables the common R/W bus to be driven high before being tri-stated.

2.5.3 Shared Memory Array LAPD Protocol Code EEPROM

In addition to the 512 Kbytes of SRAM, the SMA also contains 64 Kbytes of EEPROM containing the LAPD code for the T7130. Two 32K x 8-bit EEPROM devices, one ODD and one EVEN, share the same 16-bit I/O port as the SRAM devices. A common chip select is provided to both EEPROM devices from PAL18 and separate odd-even read/write strobes enable the odd or even I/O memory ports. These devices are mapped only in the HOST and MLC I/O space.

The MLC has READ only permission of the LAPD code whereas the HOST has both WRITE and READ permission. If any changes are to be made to the LAPD code in EEPROM, it can be downloaded through the HOST. The EEPROM read enable is shared between the HOST and MLC but the EEPROM write enable is controlled directly by the HOST. PAL40 decodes the write enables to the EEPROM from the HOST and generates the write violation when the HOST attempts a write without the enable bit set.

HOST Control Register, CR1, contains the LAPD EEPROM write enable bit LDEWEH. When LDEWEH is a logic 0, PAL40 outputs the write violation signal (LDEWVL) that is routed to PAL36. The Interrupt Status Register (ISR) in PAL36, records the status of all HOST write violations.

2.5.4 Shared Memory Array Chip Select

One PLD, PAL18, decodes both the EEPROM and SRAM chip select signals. A SMA chip select depends on a valid Bus Grant from the ACB and the SMA address bit, SMAB18, provides the actual upper and lower chip selection. PAL18 also provides the Data Transfer Acknowledge (MLCDTACK) signal for the T7130. The common shared SRAM array does not introduce any wait states for the T7130 or T7115 while an I/O transfer is in progress.

2.5.5 Shared Memory Array Address/Data Bus

The SMA address and data bus is accessible by the SPYDER-T, T7130 HOST and EX_CPU. Separate Address/Data buffers for the HOST isolate these I/O ports from the SPYDER-T and T7130. All buffers are enabled through a valid bus grant from the Arbitration Control circuit. External I/O buffers for the T7130 and SPYDER-T are not required because each contain internal tri-state control for both address and data ports. Both EEPROM and SRAM devices are connected to the address/data signal lines.

2.5.6 Shared Memory Array R/W Bus

PLD's for the SPYDER-T (PAL50), HOST (PAL38), isolate these devices from the common shared R/W enables (SMREL, SMWEL, SMROL, SMWOL). The R/W signals of the T7130 are directly connected to the SMA R/W lines because of internal tri-state control. The LAPD EEPROM write enables (LDEWEL, LDEWOL) are separate from the write enables for the SRAM devices and are controlled exclusively by HOST (PAL40).

All SMA write strobes from the HOST are delayed from SMA address and chip selects. Bus Grants for SMA address and chip selects are enabled from the rising edge of the Arbitration Control Clock (ARBCLK) while delayed Bus Grants for the write strobes are referenced to the falling edge through ARBCLKI in PAL16.

2.5.7 Initialization

Initialization of the CSA follows immediately after the HOST completes the startup routines and diagnostic routines, if any. The HOST then begins to configure the data structures. Upon completion of the formatting process, the HOST removes the reset from the MLC and issues a SPYDER Attention instruction, SA pulse from PAL26, which begins the request for establishment of a LAPD data-link. The format for all the data structures are defined in the data sheets of the LAPD protocol devices[8] which are administered by the HOST during initialization.

2.6 HOST - Main Processor and Support Logic

2.6.1 Elements and Function

The Motorola MC68020 functions as the main processor (HOST) for the LAPD interface. The HOST is also referred to as the T7130 and T7115A host processor. The HOST features a 32-bit address and 32-bit data bus and operates at a clock rate of 33.33MHZ

[8] See the *LAPD Software for the T7130 Controller* and the *Common Software for the T7130 Controller* data sheets

(30 ns clock cycle). The HOST is supported by EEPROM, SRAM and several peripheral devices summarized in the following list:

- Memory

  256K bytes of EEPROM for program memory

  256K bytes of SRAM for program memory

  256K bytes of SRAM for data memory

  64K bytes of EEPROM for the LAPD code

This memory space is also I/O mapped into the T7130 I/O space for the purpose of down loading the LAPD code into the private SRAM of the T7130.

- Interrupt Control Devices

  MC68153 (BIM - Interrupt Control)

  MC68901 (MFP - Interrupt Control and Timer)

  Interrupt Status Register (ISR, PAL36)

  Interrupt Request Register (IRR, PAL48)

- Control Registers

  CR0

  CR1

All HOST peripheral devices and memory, including all devices in the SMA, are mapped in the HOST I/O space. Control Register, CR0, is used for reset and initialization of the T7130 and T7115A and peripheral devices. Write protection of program memory space contained in both EEPROM and SRAM is set through the Control Register CR1. The EEPROM's are also protected with a lock which can be opened by a software key, whereas SRAM is protected only through the write enable bit in control register CR1. The source of any program write violations reported to the HOST through CH1 of the BIM are contained in the ISR (PAL36). The HOST has the capability to send an interrupt to either the T7130 or T7115A through the IRR (PAL48).

All program and data memory, contained in SRAM only, is configured as a 32-bit *longword* port and is designed for *zero* wait states. All EEPROM program memory is configured as a word port and the peripherals configured as a byte port.

2.6.2 HOST Support Logic

The HOST is supported by various PLD's. These are shown in table 14.

Table 14
HOST Programmable Logic Devices

Reset/Halt Control	16R4	PAL22
Data Transfer and Size Acknowledge (DSACK)	22V10	PAL24
Interrupt Status and Control Register R\W strobes	16L8	PAL26 including the SPYDER-T SA Pulse Generator
Program Data (SRAM) Memory Chip Select	16L8	PAL28 (includes DSACK for local memory)
R/W Strobes for Program Data (SRAM and EEPROM)	22V10	PAL30 and enable signals for the I\/O Transceivers (including generating the write violations)
Peripheral Chip Select	26V12	PAL32
Program Memory (EEPROM) Chip Select	16L8	PAL34
Interrupt Status Register (contains write violations)	20RA10	PAL36 and SPYDER-T Illegal Address Interrupt.
Shared Memory R/W Strobes	16L8	PAL38
LAPD (SMA) EEPROM R\/W Strobes	16L8	PAL40 (including generating the write violation)
IACK control for Interrupt Control Block	16L8	PAL44
Interrupt Request Register and Transceiver I/O Control	16L8	PAL48 HOST to MLC
LAPD EEPROM Delay	20RA10	PAL54

2.6.3 HOST EEPROM and SRAM

Two types of memory devices support the HOST. These include EEPROM and SRAM. Boot and any relevant diagnostics code reside in EEPROM. On powering up LAPD Interface, the HOST executes the boot code and any relevant diagnostics routine, then indicates a pass or fail condition of the through the Diagnostics Status Word (DSW). The HOST updates the DSW in the CSA. Four 128K x 8 SRAM devices function as the HOST program and data memory port and are configured as a 32-bit long word port. One half of this memory (256 Kbytes) contains the system directives, and this area is write protected. The other half is used for data space. Both the EEPROM and SRAM are connected directly to the data and address lines of the HOST creating a local I/O bus. The address/data outputs of the HOST can drive up to 130pf.

All HOST peripherals, including the SMA are isolated from this local I/O bus by BCT245 type transceivers. These transceivers separate all external load capacitance of the peripherals, etc., from the fast HOST memory bus.

2.6.4 Local SRAM Program Space

The 512 Kbytes of local SRAM for the HOST is arranged as 128 K longwords and is divided into two blocks of 64 K long words. HOST address bit 18 (MPAB18) is used to decode which space the processor is accessing. Program space occupying the lower 64 K block, is write protected and requires the write enable bit (CPSRWEH in CR1) to be set to a logic "1" prior to the write cycle. If CPSRWEH is not set, a write violation, through PAL30 and PAL36, is passed to the Interrupt Control Block, CH3 of the BIM. PAL28 provides four separate SRAM chip selects[9], by decoding both HOST address and SIZE[10] bits.

DSACK is also generated in this PLD in order to meet a ZERO wait-state condition for the HOST. External timing information for the no wait state requirement of the HOST can be found in reference manual.

2.6.5 Local EEPROM Program Space

Four 64 K x 8 EEPROM devices provide the HOST with Non-Volatile Memory (NVM) storage capability. The EEPROM devices are connected directly to the HOST address/data bus as a 16-bit word port. Chip selects for the EEPROM space is provided by PAL34. HOST address bit A0 (MPAB0) and the two SIZE bits MPSIZE0 and MPSIZE1, decode whether a byte or word is selected with address bit A17 decoding the upper or lower device. PAL24 provides the DSACK from a wait state generator to accommodate the slow EEPROM.

2.6.6 Access to the Shared Memory Array - SRAM and EEPROM

The HOST gains access to the SMA through the SMA bus request signal (CPSMBRL) which is routed to the ACB and indicates a request for either the EEPROM or SRAM array. The EEPROM contains the LAPD protocol code for the T7130 whereas the SRAM contains the data structures shared among the HOST, T7130, T7115 and EX_CPU. PAL30 decodes the SMA bus request and controls the SMA data transceivers. The data transceivers and address buffers are not enabled until a SMA Bus Grant from the

[9] one for each SRAM device

[10] The SIZE bits indicate a byte, word, or longword access

Arbitration Control Block is issued to the HOST. One wait-state is required for SRAM access in the SMA while twelve wait-states (six clock cycles) are incurred for EEPROM access.

The R/W strobes for both the EEPROM and SRAM devices are processed through PAL38 and PAL40 from MPAB0, the two SIZE bits (MPSIZE0, MPSIZE1) and signals CPLDSML[11] and CLDESPL[12]. Read strobes for shared memory access are not enabled until a HOST Shared Memory Bus Grant (CPSMBGL) is valid. WRITE strobes for the shared memory depend upon a delayed HOST SMA Bus Grant[13] (CPSM2DBGL, PAL16).

The read enables for the LAPD EEPROM devices are directly connected to the common SMA read bus (SMREL, SMROL), but the write enables for the LAPD EEPROM devices (LDEWEL, LDEWOL) require the LAPD write enable bit (LDEWEH) to be set (to a logic "1") in CR1 in before the HOST attempts a write to these devices. PAL40 decodes a LAPD EEPROM write violation if LDEWEH is not set and passes the write violation to the ISR (ISR PAL36[14]).

2.7 HOST Peripherals

2.7.1 Interrupt Control Devices

The remaining HOST peripherals fall into three categories, and are described in their respective sections.

[11] SRAM space in SMA

[12] LAPD EEPROM space in SMA

[13] This allows the address to stabilize prior to a write enable

[14] PAL36 latches all HOST write violations

Interrupt Control Devices

- MC68153
- MC68901 (MFP - Interrupt Control and Timer)
- Interrupt Status Register (ISR, PAL36)
- Interrupt Request Register (IRR, PAL48)

2.7.2 Interrupt Control Block

There are two interrupt control paths that are designed for the LAPD Interface. These are the interrupt requests for HOST service from on-board peripheral devices and the interrupt requests for EX_CPU service from the HOST.

2.7.3 Interrupt Requests for HOST service

All interrupt requests to the HOST are handled by the MC68153 Bus Interrupt Module (BIM) and the MC68901 Multi-Function Peripheral (MFP). Interrupt requests, are terminated and prioritized by the BIM and MFP. The BIM provides interrupt request service for four external devices and the MFP provides interrupt request service for eight external and eight internal interrupts. One BIM interrupt channel input (CH0) is dedicated to the MFP. The interrupt request output of the MFP which represents the state of any MFP interrupt source is connected to BIM CH0 allowing all of the MFP interrupt requests to be grouped into a single level before being presented to the HOST processor.

Although all of the MFP interrupts are grouped at the same level, they are assigned to individual levels within the MFP. The BIM supports programmable level assignment for each interrupt, whereas the MFP level assignment is fixed. Both devices also support mask enables for each interrupt request. A summary of the type of interrupts processed by the interrupt control block are shown in table 3.

2.7.4 BIM Interrupts

All memory write violations are assigned to BIM input channel CH3. These interrupts result from the HOST attempting a write to either local program memory (EEPROM or SRAM) or to the T7130 AND T7115A LAPD protocol memory (EEPROM) without the proper enable[15] bit set. The HOST must read the ISR (PAL36) containing the write violation status flags in order to determine which write violation caused the BIM channel CH3 to be activated.

BIM channel CH2 indicates detection of an illegal SPYDER-T address outside valid SMA space, and the 10msec Timer or Task Scheduler is assigned to CH1. BIM channel CH0 is assigned to the MFP.

2.7.5 MFP Interrupts

All data link related interrupts from the T7130 are assigned to channel CH6 of the MFP and whenever the T7130 updates it's IQ, an interrupt to the MFP is generated.

2.7.6 Exception Vectors

The exception vectors for BIM channels CH3, CH2 and CH1 are supplied by the BIM while the MFP provides all the interrupt vectors for BIM channel CH3. The interrupt handlers are configured in a daisy chain fashion with the BIM first in the chain followed by the MFP.

[15] These enables are contained in the HOST control register CR1

All interrupt vectors are loaded into the interrupt handlers (BIM and MFP) by the HOST during initialization. The HOST obtains the exception vectors during execution of an IACK cycle and the interrupt request that caused the exception is cleared by the BIM during the IACK cycle16 .

2.7.7 HOST Interrupt Acknowledge

The BIM Interrupt Request outputs, BIRQ1 through BIRQ7, are encoded into one of seven interrupt levels by an LS148 chip before being sent to the HOST. A HOST interrupt acknowledgment is decoded from the function code signals CPFC1-CPFC3, qualified with the address strobe CP0ASL. PAL 44 handles the HOST interrupt acknowledge processing for the Interrupt Control Block. The signals INTAEL, INTAL0, and INTAL1 from the BIM are decoded in PAL44 indicating which BIM channel is being serviced and clears the respective interrupt source, CH2-CLRSPIAL and CH1-CLRTM1L or passes control to the MFP through MFPIEL.

2.8 HOST Functions

2.8.1 Overview

The HOST performs three main functions which include,

1. LAPD Interface Initialization and Diagnostics
2. Level-2 Management
3. Level-3 Management

[16] With the exception of the Write Violation in BIM CH0, the HOST must clear these write violation sources with a write command to the Interrupt Status Register, ISR

2.8.2 Initialization and Diagnostics

The HOST performs a series of initialization tests that are executed immediately following a power-up or reset. A failure in anyone of the initialization and any diagnostic routines, indicates that the LAPD Interface requires service.

2.8.3 Level-2 Management of the LAPD Data Links

Level-2 programs are required to manage DL operations, including, TEI management, connection management and flow control. These programs must reside in local EEPROM and are executed by the HOST to provide link connections that are stable. The control and return status of the T7130 and T7115A is accomplished through shared data structures in SRAM contained in the SMA and the HOST administers the format of these structures. The T7130 and T7115A is permitted to activate LAPD data channels after the HOST completes any applicable diagnostics.

2.8.4 Level-3 Management

An AT&T T7121 device has been added to provide a HDLC communication path to assist with level-3 (L3) management routines which are directed toward processing and/or dispatching Operation and Link Maintenance.

2.9 Arbitration Control Block

2.9.1 Function

The Arbitration Control Block (ACB) of the LAPD Interface provides the mechanism that manages all requests for SMA access from the SPYDER-T, HOST and MLC. These devices communicate with one another through common data structures contained in SRAM[17] within the SMA.

Included in the SMA are EEPROM devices accessible only by the HOST and T7130. The ACB provides DMA control for all memory within the SMA. Access to the SMA is through a single 16-bit I/O port and only one device can access the common memory resource at any given time. Both SRAM and EEPROM devices are connected to the single SMA I/O port.

2.9.2 Shared Memory Array Request - DMA Level Assignment

The SPYDER-T, Multi-channel LAPD Controller and the HOST are each assigned a priority level into the SRAM resource based upon critical DMA times. The DMA level assignment is shown in table 2.

The assignment of levels is based upon critical DMA transfer requirements. The T7115 (SPYDER-T) has limited on-chip buffers[18] and hence has the highest priority otherwise corruption of activated LAPD channels will occur. The next highest level has been assigned to an EX_CPU which prevents it from excessive waiting times when requesting access into the SMA, for the SRAM devices.

The host obtains the resource, EEPROM or SRAM, whenever the SPYDER-T is not in the process of requesting access to the SMA. There is no critical DMA requirement for the T7130 which is assigned the lowest priority for access to the SMA.

[17] The SRAM devices are referred to as the Common Shared SRAM Array (CSA)

[18] The T7115 must gain access into the SRAM array within six (6) T7115 clock cycles after a SMA bus request

2.9.3 Arbitration Control - Shared Memory Array Request/Grant Process

The Arbitration Control block consists of four PLD's. Table 15 presents a functional description of each PLD.

Table 15
Arbitration Control Block, Programmable Logic Devices

Function	Type	Identification
Shared Memory Array Bus-Request	16R4	PAL12
Arbitration State Machine	20R8	PAL14
Shared Memory Array Bus Grant	16R4	PAL16
Shared Memory Bus Request Synchronization	16R4	PAL20

A 40 MHZ crystal oscillator[19] functions as the clock source for ACB. The arbitration control is based on a finite state machine contained in PAL14 that operates at 40 MHZ[20]. The state diagram is illustrated below. Only seven states are allowed.

There are no invalid states.

- S0 - IDLE no bus requests from the SPYDER-T, EX_CPU, HOST, or T7130
- S1 - Bus grant to SPYDER-T
- S2 - Bus Grant to EX_CPU
- S3 - Bus Grant to HOST
- S4, S5 - Bus Grant to T7130

[19] The T7130 20MHZ clock, MLCLK, is also derived from this clock source

[20] 40MHZ was determined to be the maximum clock frequency possible without state transition errors due to metasability from asynchronous inputs for PAL12

- S6 - Bus Grant to the T7130 removed, T7130 still accessing the SMA but one of the other devices is requesting a SMA access

The T7130 does not require that a Bus Grant (BG) be active for the duration of the SMA access. Removal of the BG from the T7130 guarantees that at the termination of the T7130 I/O cycle the Bus Request (BR) will be negated. All requests to the SMA, with the exception of the SMA request from the T7130 are asynchronous inputs to the AC state machine and must be synchronized to the 40 MHZ arbitration clock. SMA requests from the SPYDER-T (SPYBRL) and HOST (CPSMBRL) are latched in PAL12 with the rising edge of ARBCLK (40 MHZ Clock). All BG are issued on the next ACB clock cycle after a valid bus-request, hence, any bus-request to bus grant takes a minimum of one and maximum of two ACB clock cycles.

Synchronization of the T7130 SMA BR signal, MLCSMBRL, is treated slightly different. A separate T7130 bus request synchronization latch[21] is provided in PAL16 for MLCSMBRL which guarantees a single cycle RG for the T7130 and more importantly, allows the T7130 to initiate an external I/O transfer into the SMA one full MLCLK earlier.

The SMA Bus Grant signals, SPYBGL, SYSMBGL and CPSMBGL are primary state machine outputs S1, S2 and S3, and remain active as long as SMA Bus Request is valid[22]. However, the Bus Grant for the T7130 is a combination of states S4, S5, and S6. When the T7130 is granted access into the SMA, the ACB state machine issues a BG to

[21] The clock for this latch is an inverted MLCLK derived in PAL12

[22] A valid bus grant enables tri-state buffers for both data and address, and also enable the R/W signals for the SMA. Delayed bus-grant signals, CPSM2DBGL and SYSMDBGL, are used to separate the SMA write strobes (SMWEL, SMWOL) for the SMA chip selects due to a HOST SMA access

the MLC in S4; a transition to S5 results on the next edge of ARBCLK. If no other device is requesting access to the SMA, the AC state machine remains in state S5 and these two states provide the MLC with acknowledgment for a SMA access through the signal MLCSMBGL[23] which is output from PAL16. As soon as a valid SMA bus request is latched by the ACB in PAL14, from either SPYBRL, SYSBRL or CPSMBRL, the state machine proceeds to S6. PAL16 decodes the state transition to S6 and negates MLSMBGL to the MLC. Removal of the BG to the T7130 indicates that it must terminate the current I/O cycle and negate the shared memory bus request MLCSMBRL. Control for the T7130 R/W outputs remain valid throughout the I/O cycle from the signal MLCBGL which is decoded from S4, S5, and S6 in PAL16. MLCBGL is negated when there is a transition out of S6.

2.9.4 Initialization

The Bus Request synchronization PLD's (PAL12 and PAL16) have internal power-on reset control which cause all SMA BR to be inactive and result in an IDLE condition. The ACB state machine also provides for a path to the IDLE state whenever a power-up condition of no active states results.

The Status Register is *read only* and contains information about the MC68020 processor. The state of both the HOST RESET pin and function code which identifies the address space of the current HOST bus cycle is available through the status register

[23] MLC Shared Memory Array Bus Grant

2.10 LAPD Interface Clocks

2.10.1 Function

The LAPD Interface has various clock sources., some of which are derived from crystal oscillators. All crystal oscillators have a tri-state control.

2.11 HDLC Private Communication Link

2.11.1 Function

The Interface supports a private serial HDLC communication link between the LAPD Interface and any other subsystem that may require such service. The AT&T T7121 HDLC interface for ISDN (HIFI-64) provides this feature. The HIFI-64 is memory mapped inside the HOST I/O space and is controlled through various internal registers accessible by the HOST. The device contains a 64-byte FIFO which significantly reduces the number of interrupts that must be processed by the HOST.

The T7121 interrupt output is routed to channel 5 of the MFP in the interrupt control block and provides the interrupt service path for the HIFI-64 to the HOST.

2.12 T7130 and T7115A Memory Map

2.12.1 Memory Maps

The following tables, tables 16 through 19, show the MLC and SPYDER-T memory maps.

Table 16
T7130 I/O Memory Map

FROM	TO	SIZE	DEVICE TYPE	FUNCTION
f80000	ffffff	512 Kbytes	SRAM	R/W CSA
200000	20ffff	512 Kbytes	EEPROM	LAPD Protocol Code (ROM)

Table 17
T7115A I/O Memory Map

FROM	TO	SIZE	DEVICE TYPE	FUNCTION
f80000	ffffff	512Kbytes	SRAM	R/W CSA

Table 18
HOST Memory Map Memory Devices

FROM	TO	SIZE	DEVICE TYPE	FUNCTION
XX-000000	XX-03FFFF	256kbytes	EEPROM	Prog. Mem. Write Protected
XX-100000	XX-13FFFF	256kbytes	SRAM	Prog. Mem. Write Protected
XX-140000	XX-17FFFF	256kbytes	SRAM	Data Mem. R/W
XX-200000	XX-20FFFF	64 Kbytes	EEPROM	LAPD Code SMA Wr. Prot
XX-F80000	XX-FFFFFF	512 Kbytes	SRAM	CSA R/W

Table 19
MC68020 Memory Map for Peripheral Devices

FROM	TO	SIZE	DEVICE TYPE	FUNCTION
XX-3XXX00	XX-03FFFF	24 Bytes	MFP	24 Internal Registers
XX-3XXX30	XX-13FFFF	16 Bytes	BIM	8 Internal Registers
XX-3XXX80	XX-17FFFF	16 Bytes	T7121	16 Internal Registers
XX-3XXXF2		1 Byte	CR0	Control Register 0
XX-3XXXF4		1 Byte	CR1	Control Register 1

CHAPTER 3

BACKGROUND, CONCLUSIONS AND SUGGESTIONS

3.1 LAPD Interface

3.1.1 Importance

After CCITT adopted the HDLC protocol for its Link Access Procedure (LAP), it modified the HDLC protocol for X.25 application. As a result, Link Access Procedure - B Channel (LAPB) became the L2 protocol for X.25 applications. However, LAPB had its limitations, and the CCITT found it necessary to add multiplexing capabilities to its LAP. This resulted in the LAPB protocol being modified to create the LAPD protocol. LAPD was later adopted as the L2 protocol of the Integrated Services Digital Network (ISDN).

The purpose of LAPD is to convey user information between L3 entities across an ISDN facility using the D channel. LAPD supports point-to-point operation, multipoint operation and broadcast operation. Whereas LAPB carries information between a specific pair of users, LAPD carries both signaling information for the control of circuit switched connections as well as data. LAPD procedures are also independent of the specific information flow and the particular channel that carries the information. As a result, the European Telecommunications Standards Institute (ETSI) is currently adapting LAPD for use on the B channel.

One of the major applications of LAPD is in telelphony. As the number of telephone subscribers grows, the old analog telephone system can no longer handle the added growth. ISDN, with LAPD as its L2 protocol, provides a perfect solution to overcome this problem, since it has added multiplexing capabilities.

3.1.2 LAPD Devices

The T7130 Multichannel LAPD Controller and the T7115A Synchronous Protocol Data Formatter are excellent choices for the implementation of the LAPD protocol as specified by CCITT recommendation Q.921. The T7130 and the T7115A combination can provide an efficient and complete link layer solution for any system that requires LAPD protocol termination on multiple HDLC channels, using minimal hardware.

On the transmit side, a system HOST can build Level-3 frames in shared memory then issue a request interrupt to the T7130. The T7130 can then append the appropriate Level-2 header and pass the resulting frame to the T7115A for transmission over a multiplexed facility. On the receive side, the T7115A can read LAPD frames from any facility that has multiple HDLC channels, perform low level formatting functions on the frames then deposit the frames in shared memory. The T7130 can then terminate the LAPD frames and pass all Level-3 data to a system HOST via shared memory.

Currently, there are no other VLSI devices on the market that has the capability of processing 32 HDLC channels. Siemens has a chip that has the capability of processing only 4 HDLC channels. This means that 8 of these chips along with supporting logic would have to be combined in parallel in order to terminate all 32 channels of a CEPT facility. Search on several databases have revealed no other LAPD interface, although it is believed that proprietary interfaces do exist. Also, because of the complexity of the protocol, it is believed that vendors have modified the protocol to suit their particular needs. AT&T, for example, has its own version of the LAPD protocol that does not conform to LAPD specifications as set forth by CCITT Q.921.

3.1.3 Testing and Test Results

The LAPD Interface was thoroughly tested to ensure that it operated as expected. All 32 channels of the T7115A were configured to operate in loopback mode, with the T7115A being the physical loopback point. The LAPD protocol code was downloaded to the T7130 and the device was configured for LAPD protocol termination on all 32 HDLC channels. A MC68020 emulator was then used to run L2 driver software and L3 management software. The HOST software was configured to transmit a data string which consisted of 147 bytes and spanned multiple TD's indefinitely. The software showed the activation of all the channels and the DATA_INDICATION and DATA_RELEASE primitives for the appropriate channels as data was transmitted on that channel. The number of messages transmitted was displayed in increments of 64 K.

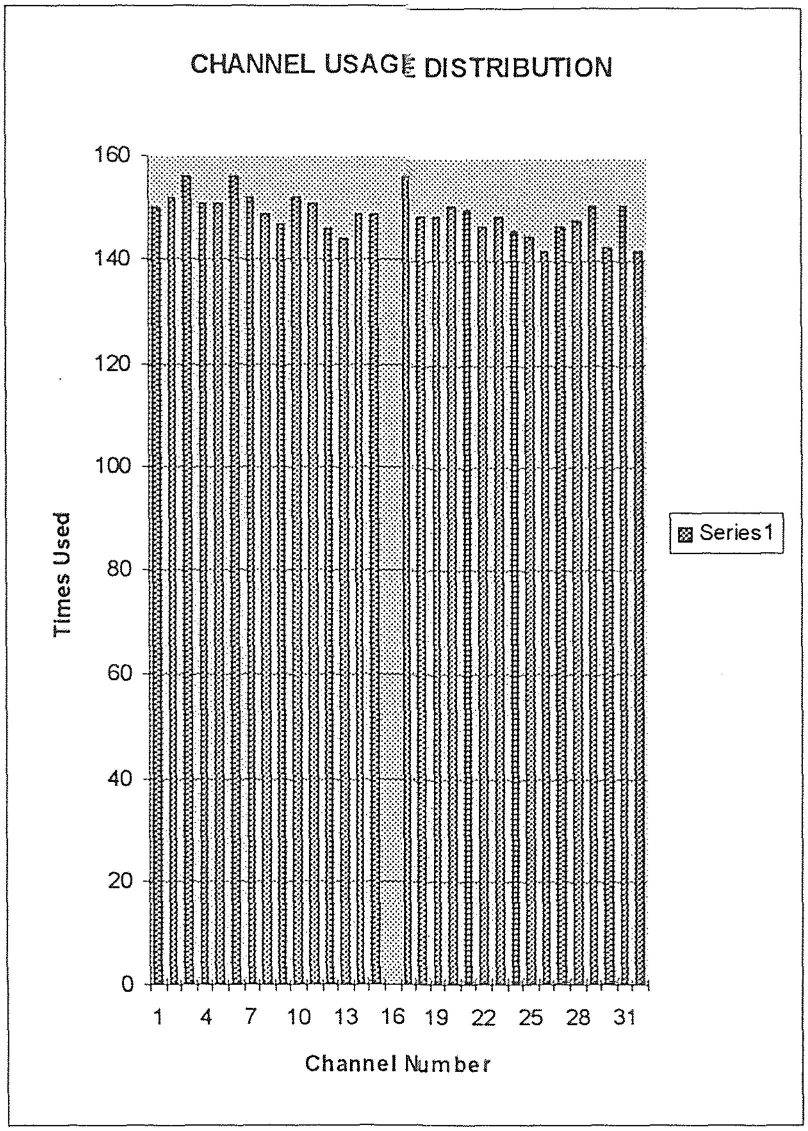
The T7130 has a HDLC Statistics Block (HSB) which keeps track of certain statistics such as good frames received, good frames transmitted and bad CRC's received. A timer was setup to display the HDLC Statistic Block of the last activated channel once every five minutes. Since the T7115A was the physical loopback point, one did not expect to see any bad frames, bad CRC's or aborts. From the data collected, there were no aborts and either the good frames transmitted were equal to the good frames received, or the good frames transmitted plus any outstanding frame were equal to the good frames received. For the latter case, the number of outstanding frames was always less than or equal two, since the window size used was two.

To see how the LAPD Interface would operate in a noisy environment, a noise generator was used to introduce random bit errors at rates of the order of 5E-5 to 1E-6 on a physical link over a single channel. Each time an error occurred it was detected and a retransmission requested. This was seen by looking at the command codes that resided on the command queues. Most of the bit errors showed up as Remote Frame Alignment (RFA) errors. A bit error rate of 10E-3 was also was also introduced on the physical LAPD link to see how the interface would react. This error rate was enough the cause the link to drop and link recovery actions were taken to reestablish the link.

Inspection of the test logs revealed that the T7130 device transmitted data over channels as soon as they channels became available. As soon as the first channel was established, data was transmitted without waiting for other channels to be established. This is one of the good features of the LAPD Interface since it gives a boost to the data throughput. This also ensures that data is transmitted evenly over all 32 channels. To verify that data was being transmitted evenly across all channels, a sample of the channel activation summary containing 4,621 channel activation entries was extracted from the logs and plotted. Table 20 shows the channel number and the number of times that particular channel was utilized along with the channel usage distribution plot. From the distribution, it is evident that all channels were utilized fairly evenly. It should be noted that channel 15 was used solely for signaling and as a result, no data was transmitted across it.

Table 20
Channel Utilization and Distribution

Channel	Usage	Channel	Usage	Channel	Usage	Channel	Usage
0	150	8	147	16	156	24	145
1	152	9	152	17	149	25	142
2	156	10	151	18	149	26	147
3	151	11	146	19	151	27	148
4	151	12	144	20	150	28	151
5	156	13	149	21	147	29	143
6	152	14	149	22	149	30	151
7	149	15	0	23	146	31	142



3.1.4 Throughput

With 32 HDLC channels running the LAPD protocol simultaneously, the T7115A is required to make numerous memory accesses. The HOST must also access the Shared Memory in an effort to ensure that the data buffers are sufficiently full. These two conditions can result in bus access problems which can limit the packet throughput. There are two solutions to this problem. The first is to use a dual bus configuration which requires a dual port memory. This design method was not utilized because dual port memory is expensive and fairly slow. The second is to use a single bus configuration. This is the design that was used and it required fast static RAM as well as an Arbitration Control circuit that provided access to the SM on a prioritized basis.

The T7130, T7115A and HOST exchange commands and status information through an interrupt driven mechanism. This reduces the SM I/O bus occupancy which results in increased throughput. The fact that the T7130 uses channels as soon as they are available also results in increased throughput. The actual throughput of the LAPD Interface was measured and found to be 942.42 messages per second since it took an average of 69.54 seconds to transmit 64 K messages.

Whenever the end of the T7115A-TQ is reached, the T7130 is blocked from adding any new frames. Consequently, the throughput of the LAPD Interface is affected by the length of the T7115A-TQ and also the number of logical channels. To minimize the effects of this problem, the T7115A-IQ is always read before the T7130-CQ. Also, the use of a pending information frame queue and a pending unnumbered frame queue will help to alleviate this problem.

APPENDIX A
PLD EQUATIONS

This section contains the PLD equations for the interface.

PALASM Design Description - PAL10

```
;---------------------------------- Declaration Segment ------------
TITLE    T7130 MLC DTACK PAL - PAL10
PATTERN  LAPD INTERFACE
REVISION 1.00
AUTHOR   HOPETON WALKER
COMPANY
DATE     08/17/94

CHIP     PAL10  PAL16V8
;---------------------------------- PIN Declarations ---------------
PIN 1  ARBCLK
PIN 2  SPYBRL
PIN 3  SYSMBRL
PIN 4  CPSMBRL
PIN 5  ARBCLKX      ;INPUT FOR INVERSE CLOCK
PIN 6  MLCLK
PIN 7  OEL1         ; OEL FOR LOGIC
PIN 8  NC
PIN 9  NC
PIN 10 GND
PIN 11 OEL2         ;OEL FOR LATCH
PIN 12 NC
PIN 13 MLCLKI
PIN 14 NC
PIN 15 CPSMBR1L
PIN 16 SYSMBR1L
PIN 17 SPYBR1L
PIN 19 ARBCLKI
PIN 20 VCC
;----------------------------------- Boolean Equation Segment ------
EQUATIONS

/SPYBR1L   := /SPYBRL
/SYSMBR1L  := /SYSMBRL
/CPSMBR1L  := /CPSMBRL

;SPYBR1L   =/OEL2
;SYSMBR1L  =/OEL2
;CPSMBR1L  =/OEL2

/ARBCLKI = ARBCLKX
/MLCLKI  = MLCLK
```

```
ARBCLKI.TRST  = /OEL1
MLCLKI.TRST   = /OEL1
;------------------------------------ Simulation Segment ------------

SIMULATION
TRACE_ON
OEL1 OEL2 SPYBRL SYSMBRL CPSMBRL MLCLK ARBCLKX ARBCLK

SETF /OEL1 /OEL2 SPYBRL SYSMBRL CPSMBRL /MLCLK /ARBCLKX /ARBCLK
CLOCKF
CHECK MLCLKI ARBCLKI SPYBR1L SYSMBR1L CPSMBR1L
SETF /SPYBRL /SYSMBRL /CPSMBRL /ARBCLK
CLOCKF
CHECK /SPYBR1L /SYSMBR1L /CPSMBR1L
CLOCKF
SETF SPYBRL SYSMBRL CPSMBRL
CLOCKF
SETF ARBCLKX MLCLK
CHECK /ARBCLKI /MLCLKI
CLOCKF
CHECK /ARBCLKI  /MLCLKI
TRACE_OFF
;-----------------------------------------------------------------
```

PALASM Design Description - PAL12

```
;---------------------------------- Declaration Segment ------------
TITLE    SHARED MEMORY BUS REQUEST SYNC PAL - PAL12
PATTERN  LAPD INTERFACE
REVISION 1.09
AUTHOR   HOPETON WALKER
COMPANY
DATE     09/15/94

CHIP     PAL12  PAL16R4

;---------------------------------- PIN Declarations ---------------
PIN 1  ARBCLK
PIN 2  SPYBRL
PIN 3  SYSMBRL
PIN 4  CPSMBRL
PIN 5  ARBCLKX     ;INPUT FOR INVERSE CLOCK
PIN 6  MLCLK
PIN 7  OEL1        ; OEL FOR LOGIC
PIN 8  NC
PIN 9  NC
PIN 10 GND
PIN 11 OEL2        ;OEL FOR LATCH
PIN 12 NC
PIN 13 MLCLKI
PIN 14 NC
PIN 15 CPSMBR1L
PIN 16 SYSMBR1L
PIN 17 SPYBR1L
PIN 19 ARBCLKI
PIN 20 VCC
;----------------------------------- Boolean Equation Segment ------
EQUATIONS

/SPYBR1L   := /SPYBRL
/SYSMBR1L  := /SYSMBRL
/CPSMBR1L  := /CPSMBRL

;SPYBR1L   =/OEL2
;SYSMBR1L  =/OEL2
;CPSMBR1L  =/OEL2

/ARBCLKI = ARBCLKX
/MLCLKI  = MLCLK
```

```
ARBCLKI.TRST  = /OEL1
MLCLKI.TRST   = /OEL1
;------------------------------------- Simulation Segment ------------

SIMULATION
TRACE_ON
OEL1 OEL2 SPYBRL SYSMBRL CPSMBRL MLCLK ARBCLKX ARBCLK

SETF /OEL1 /OEL2 SPYBRL SYSMBRL CPSMBRL /MLCLK /ARBCLKX /ARBCLK
CLOCKF
CHECK MLCLKI ARBCLKI SPYBR1L SYSMBR1L CPSMBR1L
SETF /SPYBRL /SYSMBRL /CPSMBRL /ARBCLK
CLOCKF
CHECK /SPYBR1L /SYSMBR1L /CPSMBR1L
CLOCKF
SETF SPYBRL SYSMBRL CPSMBRL
CLOCKF
SETF ARBCLKX MLCLK
CHECK /ARBCLKI /MLCLKI
CLOCKF
CHECK /ARBCLKI  /MLCLKI
TRACE_OFF
;-------------------------------------------------------------------
```

PALASM Design Description - PAL14

```
;---------------------------------- Declaration Segment ------------
TITLE    SHARED MEMORY ARBITRATION PAL - PAL14
PATTERN  LAPD
REVISION 1.21
AUTHOR   HOPETON WALKER
COMPANY
DATE     09/22/94

CHIP     PAL14  PAL20R8

;---------------------------------- PIN Declarations ---------------
PIN  1       ARBCLK                    ;2
PIN  2       SPYBR1L         ;3
PIN  3       SYSMBR1L                ;4
PIN  4       CPSMBR1L        ;5
PIN  5       MLCSMBR1L             ;6
PIN  12       GND                    ;14
PIN  13       OEL                    ;16
PIN  15       MLCLK                  ;18
PIN  16       S6             ;19
PIN  17       S5             ;MLCSMBGL  ;20
PIN  18       S4             ;MLCSMBGL  ;21
PIN  19       S3             ;CPSMBGL   ;23
PIN  20       S2             ;SYSMBGL   ;24
PIN  21       S1             ;SPYBGL    ;25
PIN  22       S0             ;/IDLE     ;26
PIN  24       VCC                    ;28

;---------------------------------- Boolean Equation Segment ------
EQUATIONS

;STATE /S0      = IDLE
;STATE /S1      = SPYBGH
;STATE /S2      = SYSMBGH
;STATE /S3      = CPSMBGH
;STATE /S4 + /S5 = MLCSMBGH, routed to  PAL20 - MLCSMBGL

/S0 := /S0 * SPYBR1L * SYSMBR1L * CPSMBR1L * MLCSMBR1L   +
     /S1 * SPYBR1L * SYSMBR1L * CPSMBR1L * MLCSMBR1L   +
     /S2 * SPYBR1L * SYSMBR1L * CPSMBR1L * MLCSMBR1L   +
     /S3 * SPYBR1L * SYSMBR1L * CPSMBR1L * MLCSMBR1L   +
     /S5 * SPYBR1L * SYSMBR1L * CPSMBR1L * MLCSMBR1L   +
     S0 * S1 * S2 * S3 * S4 * S5 * S6
```

```
/S1 := /S0 * /SPYBR1L          +
     /S1 * /SPYBR1L          +
     /S2 * /SPYBR1L * SYSMBR1L +
     /S3 * /SPYBR1L * CPSMBR1L +
     /S6 * /SPYBR1L * MLCSMBR1L

/S2 := /S0 * SPYBR1L * /SYSMBR1L +
     /S1 * SPYBR1L * /SYSMBR1L +
     /S2 * /SYSMBR1L          +
     /S3 * SPYBR1L * CPSMBR1L * /SYSMBR1L +
     /S6 * SPYBR1L * CPSMBR1L * MLCSMBR1L * /SYSMBR1L

/S3 := /S0 * SPYBR1L * SYSMBR1L * /CPSMBR1L +
     /S1 * SPYBR1L * SYSMBR1L * /CPSMBR1L +
     /S2 * SPYBR1L * SYSMBR1L * /CPSMBR1L +
     /S3 * /CPSMBR1L                 +
     /S6 * SPYBR1L * SYSMBR1L * /CPSMBR1L * MLCSMBR1L

/S4 := /S0 * SPYBR1L * SYSMBR1L * CPSMBR1L * /MLCSMBR1L +
     /S1 * SPYBR1L * SYSMBR1L * CPSMBR1L * /MLCSMBR1L +
     /S2 * SPYBR1L * SYSMBR1L * CPSMBR1L * /MLCSMBR1L +
     /S3 * SPYBR1L * SYSMBR1L * CPSMBR1L * /MLCSMBR1L

/S5 := /S4 +
     /S5 * SPYBR1L * SYSMBR1L * CPSMBR1L * /MLCSMBR1L

/S6 := /S5 * /SPYBR1L +
     /S5 * /SYSMBR1L +
     /S5 * /CPSMBR1L +
     /S6 * /MLCSMBR1L

/MLCLK := MLCLK

SIMULATION
TRACE_ON
ARBCLK SPYBR1L SYSMBR1L CPSMBR1L MLCSMBR1L OEL  MLCLK
S6 S5 S4 S3 S2 S1 S0
;;all inputs on
SETF /ARBCLK /OEL SPYBR1L SYSMBR1L CPSMBR1L MLCSMBR1L
CLOCKF
CLOCKF
CLOCKF
CHECK S6 S5 S4 S3 S2 S1 /S0
;;all devices request shared memory at the same time
```

```
;;remain in state s0 (=0)
SETF  /SPYBR1L /SYSMBR1L /CPSMBR1L /MLCSMBR1L
CHECK S6 S5 S4 S3 S2 S1 /S0
;
CLOCKF
CHECK S6 S5 S4 S3 S2 /S1 S0 ;SPYDER-T gets bus
CLOCKF
CHECK S6 S5 S4 S3 S2 /S1 S0
CLOCKF
CHECK S6 S5 S4 S3 S2 /S1 S0
SETF SPYBR1L              ;SPYDER-T request goes away
;
CLOCKF
CHECK S6 S5 S4 S3 /S2 S1 S0 ;SYSTEM-bus gets bus
CLOCKF
CHECK S6 S5 S4 S3 /S2 S1 S0
CLOCKF
CHECK S6 S5 S4 S3 /S2 S1 S0
SETF SYSMBR1L             ;EX_CPU request goes away
;
CLOCKF
CHECK S6 S5 S4 /S3 S2 S1 S0 ;HOST gets bus
CLOCKF
CHECK S6 S5 S4 /S3 S2 S1 S0
CLOCKF
CHECK S6 S5 S4 /S3 S2 S1 S0
SETF CPSMBR1L             ;HOST request goes away
;
CLOCKF
CHECK S6 S5 /S4 S3 S2 S1 S0 ;T7130 gets bus
CLOCKF
CHECK S6 /S5 S4 S3 S2 S1 S0 ;GO to S5
CLOCKF
CHECK S6 /S5 S4 S3 S2 S1 S0 ;STAY IN S5 - No Requests
CLOCKF
CHECK S6 /S5 S4 S3 S2 S1 S0 ;STAY IN S5 - No Requests
CLOCKF
CHECK S6 /S5 S4 S3 S2 S1 S0 ;STAY IN S5 - No Requests
;
SETF /SPYBR1L             ;SPYDER-T Request - push to S6
CLOCKF
CHECK /S6 S5 S4 S3 S2 S1 S0 ;S6 active
CLOCKF
CHECK /S6 S5 S4 S3 S2 S1 S0 ;Remain in S6 until T7130 BR goes away
CLOCKF
```

```
SETF SPYBR1L              ;SPYDER-T request goes away
CLOCKF
CHECK /S6 S5 S4 S3 S2 S1 S0 ;Remain in S6 until T7130 BR goes away
CLOCKF
CHECK /S6 S5 S4 S3 S2 S1 S0 ;Remain in S6 until T7130 BR goes away
SETF MLCSMBR1L              ;T7130 request goes away.
CLOCKF
CHECK S6 S5 S4 S3 S2 S1 S0  ;Illegal State - but go to IDLE next CLK
CLOCKF
CHECK S6 S5 S4 S3 S2 S1 /S0 ;Go to IDLE
;
SETF /MLCSMBR1L             ;Again to S6
CLOCKF
CHECK S6 S5 /S4 S3 S2 S1 S0 ;T7130 gets bus
CLOCKF
CHECK S6 /S5 S4 S3 S2 S1 S0 ;GO to S5
CLOCKF
CHECK S6 /S5 S4 S3 S2 S1 S0 ;STAY IN S5 - No Requests
SETF /SPYBR1L             ;SPYDER-T Request - push to S6
CLOCKF
CHECK /S6 S5 S4 S3 S2 S1 S0 ;S6 active
CLOCKF
CHECK /S6 S5 S4 S3 S2 S1 S0 ;Remain in S6 until T7130 BR goes away
CLOCKF
SETF MLCSMBR1L              ;Next State S1
CLOCKF
CHECK S6 S5 S4 S3 S2 /S1 S0 ;SPYDER-T gets bus, S1 active
SETF SPYBR1L /CPSMBR1L
CLOCKF
CHECK S6 S5 S4 /S3 S2 S1 S0 ;HOST gets bus, S3 active
SETF /SPYBR1L             ;SPYDER-T comes in
CLOCKF
CHECK S6 S5 S4 /S3 S2 S1 S0 ;S3 remains active
SETF CPSMBR1L
CLOCKF
CHECK S6 S5 S4 S3 S2 /S1 S0 ;SPYDER-T gets bus, S1 active
CLOCKF
CHECK S6 S5 S4 S3 S2 /S1 S0 ;SPYDER-T gets bus, S1 active

TRACE_OFF
```

PALASM Design Description - PAL16

```
;---------------------------------- Declaration Segment ------------
TITLE    MLC Shared Memory Bus Grant - PAL16
PATTERN  LAPD
REVISION 1.0O
AUTHOR   HOPETON WALKER
COMPANY
DATE     08/13/94

CHIP     PAL16  PAL16R4

;---------------------------------- PIN Declarations ---------------
PIN  1      ARBCLKI
PIN  2      S4
PIN  3      S5
PIN  4      S6
PIN  5      CPSMBGL
PIN  6      SYSMBGL
PIN  7      NC
PIN  8      NC
PIN  9      OEL1                 ;OEL FOR COMBINATORIAL
PIN  10      GND
PIN  11      OEL2                ;OEL FOR LATCH
PIN  12      NC
PIN  13      NC
PIN  14      SYSMDBGL   ;LATCH
PIN  15      CPSM2DBGL           ;LATCH
PIN  16      NC
PIN  17      S45                 ;LATCH
PIN  18      MLCBGL
PIN  19      MLCSMBGL
PIN  20      VCC

;---------------------------------- Boolean Equation Segment ------
EQUATIONS

/S45 := /S4 + /S5

/CPSM2DBGL := /CPSMBGL

/SYSMDBGL := /SYSMBGL

/MLCSMBGL = /S45 + /S4
```

```
/MLCBGL = /S45 + /S4 + /S6

MLCSMBGL.TRST = /OEL1
MLCBGL.TRST = /OEL1

;---------------------------------- Simulation Segment ------------
SIMULATION
TRACE_ON
OEL1 OEL2 S4 S5 S6 MLCBGL MLCSMBGL S45

SETF /OEL1 /OEL2 S4 S5 S6 /ARBCLKI SYSMBGL CPSMBGL
CLOCKF
CHECK MLCBGL MLCSMBGL
CLOCKF
SETF /S4
CHECK /MLCBGL /MLCSMBGL
CLOCKF
CHECK /MLCBGL /MLCSMBGL
SETF S4 /S5
CHECK /MLCBGL /MLCSMBGL
CLOCKF
CHECK /MLCBGL /MLCSMBGL
SETF S5 /S6
CHECK /MLCBGL /MLCSMBGL
CLOCKF
CHECK /MLCBGL MLCSMBGL
CLOCKF
SETF S6
CHECK MLCBGL MLCSMBGL
CLOCKF
CHECK SYSMDBGL CPSM2DBGL
SETF /SYSMBGL /CPSMBGL
CLOCKF
CHECK /SYSMDBGL /CPSM2DBGL
SETF SYSMBGL CPSMBGL
CLOCKF
CHECK SYSMDBGL CPSM2DBGL

TRACE_OFF
;------------------------------------------------------------------
```

PALASM Design Description - PAL18

```
;------------------------Declaration Segment--------------
TITLE    SHARED MEMORY CHIP SELECT - PAL18
PATTERN  LAPD
REVISION 1.00
AUTHOR   HOPETON WALKER
COMPANY
DATE     08/15/94

CHIP   HOSTPAL     PAL22V10

;------------------------PIN Declarations-----------------
;        PINS
PIN     1       SPYBGL                  ;CK/I
PIN     2       SYSMBGL                 ;I1
PIN     3       CPSMBGL                 ;I2
PIN     4       MLCA23                  ;I3
PIN     5       MLCA22                  ;I4
PIN     6       MLCA21                  ;I5
PIN     7       MLCA20                  ;I6
PIN     8       MLCA19                  I7
PIN     9       MLCASL                  ;I8
PIN     10      SMAB18                  ;I9
PIN     11      CPSMSL          ;I10
PIN     12      GND             ;GND
PIN     13      CLDESPL                 ;I11
PIN     14      MPAB20                  ;I/O0
PIN     15      OEL             ;I/O1
PIN     16      MLCDTACKL               ;I/O2
PIN     17      MLCESPL                 ;I/O3
PIN     18      MLCBGL                  ;I/O4
PIN     19      SPYSMCSL          ;I/O5
PIN     20      SBASL           ;I/O6
PIN     21      LDECSL                  ;I/O7
PIN     22      SMLCSL                  ;I/O8
PIN     23      SMUCSL                  ;I/O9
PIN     24      VCC             ;VCC


STRING MLCSMSPH ' /MLCASL * MLCA23 * MLCA22 * MLCA21 * MLCA20 *
MLCA19 '

STRING MLCEESPH ' /MLCA23 * /MLCA22 * MLCA21 * /MLCA20 * /MLCASL '
```

```
EQUATIONS

        SMUCSL.TRST        =       /OEL
        SMLCSL.TRST        =       /OEL

        LDECSL.TRST        =       /OEL

        MLCESPL.TRST       =       /OEL

        MLCDTACKL.TRST=            MLCSMSPH * /MLCBGL * /OEL

        ;Shared Memory - SRAM

        /SMUCSL            =       SMAB18 * MLCSMSPH * /MLCBGL +
                           SMAB18 * /SPYSMCSL * /SPYBGL +
                           SMAB18 * /SBASL * /SYSMBGL +
                           SMAB18 * /CPSMSL * /CPSMBGL


        /SMLCSL     =      /SMAB18 * MLCSMSPH * /MLCBGL +
                           /SMAB18 * /SPYSMCSL * /SPYBGL +
                           /SMAB18 * /SBASL * /SYSMBGL +
                           /SMAB18 * /CPSMSL * /CPSMBGL


        ;Shared Memory - EEPROM (LAPD)

        /LDECSL            =       MLCEESPH * /MLCBGL  +
                     /CPSMBGL * /CLDESPL * /MPAB20

        /MLCESPL    =      MLCEESPH * /MLCBGL

        /MLCDTACKL         =       MLCSMSPH * /MLCBGL

SIMULATION

TRACE_ON   SPYBGL SYSMBGL SPYSMCSL CPSMBGL CPSMSL MLCASL
MLCA23
           MLCA22 MLCA21 MLCA20 MLCA19
           MLCASL SMAB18 OEL SMLCSL SMUCSL
           CLDESPL MPAB20 MLCESPL LDECSL
```

```
SETF  /OEL SPYBGL SYSMBGL SBASL SPYSMCSL CPSMBGL CPSMSL
MLCBGL
SETF   MLCASL MLCA23 MLCA22 MLCA21 MLCA20 MLCA19 SMAB18
CLDESPL MPAB20
CHECK          SMUCSL SMLCSL

;SPYDER
SETF  /SPYBGL SYSMBGL CPSMBGL SMAB18
CHECK          SMUCSL SMLCSL LDECSL
SETF  /SPYSMCSL
CHECK          /SMUCSL SMLCSL
SETF  /SMAB18
CHECK          SMUCSL /SMLCSL
SETF   SPYSMCSL
CHECK          SMUCSL SMLCSL
SETF  SPYBGL /CPSMSL
CHECK          SMUCSL SMLCSL

;CP  -  CPSMSL set low previous
SETF  SPYBGL SYSMBGL /CPSMBGL SMAB18
CHECK          /SMUCSL SMLCSL
SETF  /SMAB18
CHECK          SMUCSL /SMLCSL
SETF  CPSMBGL CPSMSL
CHECK          SMUCSL SMLCSL
SETF  /CLDESPL /MPAB20
CHECK          SMUCSL SMLCSL LDECSL
SETF  /CPSMBGL
CHECK          SMUCSL SMLCSL /LDECSL
SETF  /CPSMBGL CLDESPL /MPAB20
CHECK          SMUCSL SMLCSL LDECSL
SETF  /CPSMBGL /CLDESPL /MPAB20
CHECK          SMUCSL SMLCSL /LDECSL
SETF  CPSMBGL /CLDESPL /MPAB20
CHECK          SMUCSL SMLCSL LDECSL
SETF  /CPSMBGL /CLDESPL /MPAB20
CHECK          SMUCSL SMLCSL /LDECSL
SETF  CPSMBGL CLDESPL /MPAB20
CHECK          SMUCSL SMLCSL LDECSL
;

;MLC bus grant -SRAM
SETF  /MLCBGL
CHECK          SMUCSL SMLCSL LDECSL
;
```

```
SETF  /MLCASL /MLCA23 MLCA22 MLCA21 MLCA20 MLCA19 SMAB18
SETF  /MLCASL /MLCA23 MLCA22 MLCA21 MLCA20 MLCA19 SMAB18
CHECK        SMUCSL SMLCSL ^MLCDTACKL
;
SETF  MLCA23
CHECK        /SMUCSL  /MLCDTACKL
;
SETF  /MLCA22
CHECK        SMUCSL  ^MLCDTACKL
;
SETF  MLCA22
CHECK        /SMUCSL  /MLCDTACKL
;
SETF  /MLCA21
CHECK        SMUCSL  ^MLCDTACKL
;
SETF  MLCA21
CHECK        /SMUCSL  /MLCDTACKL
;
SETF  /MLCA20
CHECK        SMUCSL  ^MLCDTACKL
;
SETF  MLCA20
CHECK        /SMUCSL  /MLCDTACKL
;
SETF  /MLCA19
CHECK        SMUCSL  ^MLCDTACKL
;
SETF  MLCA19
CHECK        /SMUCSL /MLCDTACKL
;
SETF  /SMAB18
CHECK        SMUCSL /SMLCSL LDECSL
;
SETF  MLCBGL
CHECK        SMUCSL SMLCSL ^MLCDTACKL
;
SETF  MLCASL MLCA23 MLCA22 MLCA21 MLCA20 MLCA19 SMAB18
CHECK        SMUCSL SMLCSL ^MLCDTACKL
SETF  /MLCBGL
CHECK        SMUCSL SMLCSL LDECSL ^MLCDTACKL

;MLC - LAPD EEPROM
SETF  /MLCASL /MLCA23 /MLCA22 MLCA21 /MLCA20 /MLCA19 /SMAB18
CHECK        /MLCESPL /LDECSL
```

```
;
SETF  MLCA23
CHECK        MLCESPL LDECSL

SETF  /MLCA23
CHECK        /MLCESPL /LDECSL

SETF  MLCA22
CHECK        MLCESPL LDECSL

SETF  /MLCA22
CHECK        /MLCESPL /LDECSL

SETF  /MLCA21
CHECK        MLCESPL LDECSL

SETF  MLCA21
CHECK        /MLCESPL /LDECSL

TRACE_OFF
```

PALASM Design Description - PAL20

```
;---------------------------------- Declaration Segment ------------
TITLE    SHARED MEMORY BUS REQUEST SYNC PAL - PAL20
PATTERN  1.0
REVISION 1.00
AUTHOR   HOPETON WALKER
COMPANY
DATE     08/15/94

CHIP     PAL20  PAL16R4

;---------------------------------- PIN Declarations ---------------
PIN 1  MLCLKI
PIN 2  MLCSMBRL
PIN 3  NC
PIN 4  NC
PIN 5  NC
PIN 6  NC
PIN 7  NC
PIN 8  NC
PIN 9  NC
PIN 10 GND
PIN 11 OEL2        ;OEL FOR LATCH
PIN 12 NC
PIN 13 NC
PIN 14 NC
PIN 15 NC
PIN 16 NC
PIN 17 MLCSMBR1L
PIN 19 NC
PIN 20 VCC
;----------------------------------- Boolean Equation Segment ------
EQUATIONS

/MLCSMBR1L  := /MLCSMBRL

;----------------------------------- Simulation Segment ------------

SIMULATION
TRACE_ON
OEL2 MLCSMBRL
SETF /OEL2 MLCSMBRL /MLCLKI
CLOCKF
CHECK MLCSMBR1L
```

```
CLOCKF
CHECK MLCSMBR1L
SETF /MLCSMBRL
CLOCKF
CHECK /MLCSMBR1L
CLOCKF
CHECK /MLCSMBR1L
SETF MLCSMBRL
CLOCKF
CHECK MLCSMBR1L
TRACE_OFF
;----------------------------------------------------------------------
```

PALASM Design Description - PAL22

```
;---------------------------------- Declaration Segment ------------
TITLE    HOST Reset - PAL22
PATTERN  LAPD
REVISION 1.00
AUTHOR   HOPETON WALKER
COMPANY
DATE     08/15/94

CHIP     PAL22  PAL16V8

;---------------------------------- PIN Declarations ---------------
PIN  1       BFRM                            ;
PIN  2       CPHORENH                        ;
PIN  3       CUPRSTH                         ;
PIN  4       CUPHLTH                         ;
PIN  5       PRSTL                             ;
PIN  6       OEN1L                             ;
PIN  7       NC                               ;
PIN  8       NC                               ;
PIN  9       GPRSTH                             ;
PIN  10       GND                             ;
PIN  11       OEN2L                            ;
PIN  12       CPHORH                            ;
PIN  13       CPRSTH                            ;
PIN  14       CPHLTH                            ;
PIN  15       GPRST1L                           ;
PIN  16       SYCRGCL                            ;
PIN  17       SYCRGC1L                           ;
PIN  18       SYGPRST1H                          ;
PIN  19       SYGPRSTH                       ;
PIN  20       VCC                            ;

;---------------------------------- Boolean Equation Segment ------
EQUATIONS

/CPHORH = /(CPHORENH * CUPRSTH + CPHORENH * CUPHLTH)
CPHORH.TRST = /OEN1L

/CPRSTH = /(/PRSTL + CUPRSTH + /GPRST1L)
CPRSTH.TRST = /OEN1L

/CPHLTH = /CUPHLTH
```

```
CPHLTH.TRST = /OEN1L

/SYCRGC1L := CUPRSTH

/SYCRGCL := /SYCRGC1L

/GPRST1L = GPRSTH + /GPRST1L * /SYGPRSTH
GPRST1L.TRST = /OEN1L

/SYGPRST1H := GPRST1L

/SYGPRSTH := /SYGPRST1H

;------------------------------------ SIMULATION SEGMENT ------------
SIMULATION

;INITIALIZE
SETF /BFRM /CPHORENH /CUPRSTH /CUPHLTH PRSTL /OEN1L /GPRSTH
/OEN2L

;TEST TRI-STATE
SETF OEN1L
SETF /OEN1L
SETF OEN2L
SETF /OEN2L

;TEST CORE INITIATED RESET
SETF CUPRSTH
WHILE (SYCRGCL) DO
 BEGIN
  CLOCKF BFRM
 END
SETF /CUPRSTH

;TEST GPITS INITIATED RESET
SETF GPRSTH
SETF /GPRSTH
CLOCKF BFRM
CLOCKF BFRM
CLOCKF BFRM
CLOCKF BFRM

;TEST POWER ON RESET
SETF /PRSTL
SETF PRSTL
```

```
;TEST CORE INITIATED HALT
SETF CUPHLTH
SETF /CUPHLTH

;TEST RESET/HALT SIGNALS TO ANGEL
SETF CPHORENH
SETF CUPRSTH
SETF /CUPRSTH
SETF CUPHLTH
SETF /CUPHLTH

;----------------------------------------------------------------------
```

PALASM Design Description - PAL24

```
;------------------------------Declaration Segment---------
TITLE    HOST D/T/SACK PAL - PAL24
PATTERN  LAPD
REVISION 1.08
AUTHOR   HOPETON WALKER
COMPANY
DATE     08/31/94

CHIP     PAL24  PAL22V10

;------------------------------PIN Declarations------------
;     PINS
PIN  1        SPYCLK
PIN  2        CPEESL
PIN  3        CPSMDBGL
PIN  4        CLDSMSPL
PIN  5        CLDESPL
PIN  6        CPUIRSL
PIN  7        PERIFDKL
PIN  8        HIFICSL
PIN  9        MFPDK0L
PIN  10       BDK0L
PIN  11       UDK0L
PIN  12       GND
PIN  13       PERIFCSL                ;DELAYED Perifcsl with Data-strobe
PIN  14       CP0ASL
PIN  15       OEL
PIN  16       RESEN0L
PIN  17       CPDSACK1L
PIN  18       CPDSACK0L
PIN  19       BIACKINL
PIN  20       W8
PIN  21       W6
PIN  22       W4
PIN  23       W2
PIN  24       VCC

NODE 1        SRPIN

;---------------------------------- Boolean Equation Segment ------
EQUATIONS

; RESET CONTROL
```

```
/RESEN0L = CPEESL * CPSMDBGL * PERIFCSL * BIACKINL
SRPIN.RSTF = /RESEN0L

;SHIFT REGISTER
/W2 := VCC
/W4 := /W2
/W6 := /W4
/W8 := /W6

;DATA-TRANSFER-SIZE ACKNOWOLEDGE SIGNALS

/CPDSACK0L = /W6 * /CPUIRSL * /PERIFCSL * /CP0ASL +
          /W6 * /HIFICSL * /PERIFCSL * /CP0ASL +
          CPUIRSL * HIFICSL * /W2 * /PERIFCSL * /PERIFDKL * /CP0ASL +
          /MFPDK0L * /PERIFCSL * /CP0ASL +
          /BDK0L * /PERIFCSL * /CP0ASL +
          /UDK0L * /PERIFCSL * /CP0ASL +
          /MFPDK0L * /BIACKINL * /CP0ASL +
          /BDK0L * /BIACKINL * /CP0ASL +
          /UDK0L * /BIACKINL * /CP0ASL

/CPDSACK1L = /W8 * /CPEESL * /CP0ASL +
          /W6 * /CLDESPL * /CPSMDBGL * /CP0ASL +
              /CLDSMSPL * /CPSMDBGL * /CP0ASL

;OEL CONTROL FOR DSACK0-1L
CPDSACK0L.TRST = RESEN0L * /OEL
CPDSACK1L.TRST = RESEN0L * /OEL
RESEN0L.TRST = /OEL
W2.TRST = /OEL
W4.TRST = /OEL
W6.TRST = /OEL
W8.TRST = /OEL
;------------------------------------ Simulation Segment ------------
SIMULATION
TRACE_ON
SPYCLK CPEESL CPSMDBGL CLDSMSPL CLDESPL CPUIRSL
PERIFDKL PERIFCSL HIFICSL MFPDK0L BDK0L UDK0L CP0ASL
OEL RESEN0L CPDSACK1L CPDSACK0L
W8 W6 W4 W2

SETF /OEL /SPYCLK CPEESL CPSMDBGL CLDSMSPL CLDESPL CPUIRSL
   PERIFDKL PERIFCSL HIFICSL MFPDK0L BDK0L UDK0L CP0ASL BIACKINL
CLOCKF
```

```
CHECK /RESEN0L ^CPDSACK1L ^CPDSACK0L
    W8 W6 W4 W2
;
;HOST EEPROM PROGRAM MEMORY
SETF /SPYCLK /CP0ASL /CPEESL
CLOCKF
CHECK RESEN0L CPDSACK1L CPDSACK0L
    W8 W6 W4 /W2
CLOCKF
CHECK W8 W6 /W4 /W2
SETF /SPYCLK
CLOCKF
CHECK W8 /W6 /W4 /W2
SETF /SPYCLK
CLOCKF
CHECK RESEN0L /CPDSACK1L CPDSACK0L /W8 /W6 /W4 /W2
SETF /SPYCLK CP0ASL CPEESL
CLOCKF
CHECK /RESEN0L ^CPDSACK1L ^CPDSACK0L
    W8 W6 W4 W2
CLOCKF
CHECK W8 W6 W4 W2
;
;SHARED MEMORY - SRAM
SETF /SPYCLK /CP0ASL /CPSMDBGL /CLDSMSPL
CHECK RESEN0L /CPDSACK1L CPDSACK0L
CLOCKF
CHECK W8 W6 W4 /W2
SETF /SPYCLK CP0ASL CLDSMSPL CPSMDBGL
CHECK /RESEN0L ^CPDSACK1L ^CPDSACK0L
CLOCKF
CHECK /RESEN0L ^CPDSACK1L ^CPDSACK0L
    W8 W6 W4 W2
;
;HIFI SIX WAIT STATES
SETF /SPYCLK /CP0ASL /HIFICSL /PERIFCSL
CHECK RESEN0L CPDSACK1L CPDSACK0L
CLOCKF
CHECK W8 W6 W4 /W2
CLOCKF
CHECK W8 W6 /W4 /W2
SETF /SPYCLK
CLOCKF
CHECK W8 /W6 /W4 /W2
CHECK RESEN0L CPDSACK1L /CPDSACK0L
```

```
SETF /SPYCLK
CLOCKF
CHECK RESEN0L CPDSACK1L /CPDSACK0L /W8 /W6 /W4 /W2
SETF /SPYCLK CP0ASL HIFICSL PERIFCSL
CHECK /RESEN0L ^CPDSACK1L ^CPDSACK0L
CLOCKF
CHECK /RESEN0L ^CPDSACK1L ^CPDSACK0L
    W8 W6 W4 W2
;
;EX_CPUIRSL SIX WAIT STATES (ASSUMES 68020)
SETF /SPYCLK /CP0ASL /CPUIRSL /PERIFDKL /PERIFCSL
CHECK RESEN0L CPDSACK1L CPDSACK0L
CLOCKF
CHECK W8 W6 W4 /W2
CLOCKF
CHECK W8 W6 /W4 /W2
SETF /SPYCLK
CLOCKF
CHECK W8 /W6 /W4 /W2
CHECK RESEN0L CPDSACK1L /CPDSACK0L
SETF /SPYCLK
CLOCKF
CHECK RESEN0L CPDSACK1L /CPDSACK0L /W8 /W6 /W4 /W2
SETF /SPYCLK CP0ASL CPUIRSL PERIFCSL PERIFDKL
CHECK /RESEN0L ^CPDSACK1L ^CPDSACK0L
CLOCKF
CHECK /RESEN0L ^CPDSACK1L ^CPDSACK0L
    W8 W6 W4 W2
;
;CLDESPL SIX WAIT STATES
SETF /SPYCLK /CP0ASL /CLDESPL /CPSMDBGL
CHECK RESEN0L CPDSACK1L CPDSACK0L
CLOCKF
CHECK W8 W6 W4 /W2
CLOCKF
CHECK W8 W6 /W4 /W2
SETF /SPYCLK
CLOCKF
CHECK W8 /W6 /W4 /W2
CHECK RESEN0L /CPDSACK1L CPDSACK0L
SETF /SPYCLK
CLOCKF
CHECK RESEN0L /CPDSACK1L CPDSACK0L /W8 /W6 /W4 /W2
SETF /SPYCLK CP0ASL CLDESPL CPSMDBGL
CHECK /RESEN0L ^CPDSACK1L ^CPDSACK0L
```

```
CLOCKF
CHECK /RESEN0L ^CPDSACK1L ^CPDSACK0L
    W8 W6 W4 W2
;
;PERIFDKL (CR0,CR1,ISR,) TWO WAIT STATES
;
SETF /SPYCLK /CP0ASL /PERIFCSL /PERIFDKL
CHECK RESEN0L CPDSACK1L CPDSACK0L
CLOCKF
CHECK W8 W6 W4 /W2
CHECK RESEN0L CPDSACK1L /CPDSACK0L
SETF /SPYCLK CP0ASL PERIFCSL PERIFDKL
CHECK /RESEN0L ^CPDSACK1L ^CPDSACK0L
CLOCKF
CHECK /RESEN0L ^CPDSACK1L ^CPDSACK0L
    W8 W6 W4 W2
;
;MFP-BIM-  with PERIFCSL only
;MFP - NO WAIT STATES
SETF /SPYCLK /CP0ASL /MFPDK0L /PERIFCSL
CHECK RESEN0L CPDSACK1L /CPDSACK0L
CLOCKF
CHECK RESEN0L CPDSACK1L /CPDSACK0L
    W8 W6 W4 /W2
SETF /SPYCLK CP0ASL MFPDK0L PERIFCSL
CHECK /RESEN0L ^CPDSACK1L ^CPDSACK0L
CLOCKF
CHECK /RESEN0L ^CPDSACK1L ^CPDSACK0L
    W8 W6 W4 W2
;
;BIM - NO WAIT STATES
SETF /SPYCLK /CP0ASL /BDK0L /PERIFCSL
CHECK RESEN0L CPDSACK1L /CPDSACK0L
CLOCKF
CHECK W8 W6 W4 /W2
SETF /SPYCLK CP0ASL BDK0L PERIFCSL
CHECK /RESEN0L ^CPDSACK1L ^CPDSACK0L
CLOCKF
CHECK /RESEN0L ^CPDSACK1L ^CPDSACK0L
    W8 W6 W4 W2
;
 ;
;HOST Interrupt Acknowledge Cycle MFP-BIM
;MFP - NO WAIT STATES
SETF /SPYCLK /CP0ASL /MFPDK0L /BIACKINL
```

```
CHECK RESEN0L CPDSACK1L /CPDSACK0L
CLOCKF
CHECK RESEN0L CPDSACK1L /CPDSACK0L
    W8 W6 W4 /W2
SETF /SPYCLK CP0ASL MFPDK0L BIACKINL
CHECK /RESEN0L ^CPDSACK1L ^CPDSACK0L
CLOCKF
CHECK /RESEN0L ^CPDSACK1L ^CPDSACK0L
    W8 W6 W4 W2
;
;BIM - NO WAIT STATES
SETF /SPYCLK /CP0ASL /BDK0L /BIACKINL
CHECK RESEN0L CPDSACK1L /CPDSACK0L
CLOCKF
CHECK W8 W6 W4 /W2
SETF /SPYCLK CP0ASL BDK0L BIACKINL
CHECK /RESEN0L ^CPDSACK1L ^CPDSACK0L
CLOCKF
CHECK /RESEN0L ^CPDSACK1L ^CPDSACK0L
    W8 W6 W4 W2
;
 TRACE_OFF
```

PALASM Design Description - PAL26

```
;---------------------------------- Declaration Segment ------------
TITLE   CONTROL/INTERRUPT REGS  - PAL26
;       R/W STROBES AND SPYDER ATTENTION PALS.,
;       INCLUDES SAP GENERATION
PATTERN  LAPD
REVISION 1.09
AUTHOR   HOPETON WALKER
COMPANY
DATE     08/15/94

CHIP     PAL26  PAL16V8

;---------------------------------- PIN Declarations ---------------
PIN  1      CPDB24                          ;
PIN  2      CR0SL                           ;
PIN  3      CR1SL                           ;
PIN  4      CP0RWL                          ;
PIN  5      CP0DSL                          ;
PIN  6      IRSL                            ;
PIN  7      NC                              ;
PIN  8      NC                              ;
PIN  9      NC                              ;
PIN  10      GND                            ;
PIN  11      OENL                           ;
PIN  12      ISRCL                          ;
PIN  13      ISRDL                          ;
PIN  14      CR1CKL                         ;
PIN  15      CR1RDL                         ;
PIN  16      CR0CKL                         ;
PIN  17      CR0RDL                         ;
PIN  18      SPYSA1                         ;
PIN  19      SPYSAP                         ;
PIN  20      VCC                            ;

;---------------------------------- Boolean Equation Segment ------
EQUATIONS

/SPYSA1 = /CPDB24
SPYSA1.TRST = /OENL * /CP0RWL * /CP0DSL * /CR0SL

/SPYSAP = SPYSA1
SPYSAP.TRST = /OENL
```

```
/CR0RDL = /CR0SL * /CP0DSL * CP0RWL
CR0RDL.TRST = /OENL

/CR0CKL = /CR0SL * /CP0DSL * /CP0RWL
CR0CKL.TRST = /OENL

/CR1RDL = /CR1SL * /CP0DSL * CP0RWL
CR1RDL.TRST = /OENL
/CR1CKL = /CR1SL * /CP0DSL * /CP0RWL
CR1CKL.TRST = /OENL

/ISRDL = /IRSL * /CP0DSL * CP0RWL
ISRDL.TRST = /OENL

/ISRCL = /IRSL * /CP0DSL * /CP0RWL
ISRCL.TRST = /OENL

;----------------------------------- Simulation Segment ------------
SIMULATION

;INITIALIZE
SETF /CPDB24 CR0SL CR1SL CP0RWL CP0DSL IRSL /OENL

;TEST TRISTATE CAPABILITY
SETF OENL
SETF /OENL

;TEST FOR TRISTATE, PULL UP IF TRUE, AND TEST CR0 WRITE
WHILE (CP0RWL + CP0DSL + OENL + CR0SL) DO
 BEGIN
  SETF SPYSA1
  SETF /CP0RWL /CP0DSL /CR0SL
 END
SETF CP0RWL CP0DSL CR0SL

;TEST CR0 READ
SETF /CP0DSL
SETF /CR0SL
SETF CP0DSL CR0SL

;TEST CR1 READ/WRITE
SETF /CP0DSL
SETF /CR1SL
SETF CP0DSL
SETF CR1SL
SETF /CP0DSL /CP0RWL
SETF /CR1SL
SETF CP0DSL CP0RWL
SETF CR1SL

;TEST ISR READ/WRITE
SETF /CP0DSL
SETF /IRSL
SETF CP0DSL
SETF IRSL
SETF /CP0DSL /CP0RWL
SETF /IRSL
SETF CP0DSL CP0RWL
SETF IRSL

;----------------------------------------------------------------------
```

PALASM Design Description - PAL28

```
;---------------------------------- Declaration Segment ------------
TITLE    PROGRAM DATA MEMORY CHIP SELECT PAL - PAL28
PATTERN  LAPD
REVISION 1.04
AUTHOR   HOPETON WALKER
COMPANY
DATE     09/05/94
CHIP     PAL28  PAL16L8

;---------------------------------- PIN Declarations ---------------
PIN  1      MPAB0                          ;
PIN  2      MPAB1                          ;
PIN  3      MPSIZE0                        ;
PIN  4      MPSIZE1                        ;
PIN  5      MPAB22                         ;
PIN  6      MPAB21                         ;
PIN  7      MPAB20                         ;
PIN  8      MPAB19                         ;
PIN  9      CP0ASL                         ;
PIN  10      GND                           ;
PIN  11      OENL                          ;
PIN  12      GPITSEL                       ;
PIN  13      PERIFSEL                      ;
PIN  14      CPDSACK1L                              ;
PIN  15      CPDSACK0L                              ;
PIN  16      PDLLCSL                       ;
PIN  17      PDLMCSL                                ;
PIN  18      PDUMCSL                                ;
PIN  19      PDUUCSL                       ;
PIN  20      VCC                           ;

STRING PDSP '/MPAB22 * /MPAB21 * MPAB20 * /MPAB19 * /CP0ASL'

;---------------------------------- Boolean Equation Segment ------
EQUATIONS

/PDUUCSL = PDSP * /MPAB1 * /MPAB0
PDUUCSL.TRST = /OENL

/PDUMCSL = PDSP * /MPAB1 * /MPSIZE0
        + PDSP * MPAB0 * /MPAB1
        + PDSP * /MPAB1 * MPSIZE1
PDUMCSL.TRST = /OENL
```

```
/PDLMCSL = PDSP * /MPAB0 * MPAB1
        + PDSP * /MPAB1 * /MPSIZE0 * /MPSIZE1
        + PDSP * /MPAB1 * MPSIZE0 * MPSIZE1
        + PDSP * MPAB0 * /MPAB1 * /MPSIZE0
PDLMCSL.TRST = /OENL

/PDLLCSL = PDSP * MPAB0 * MPSIZE0 * MPSIZE1
        + PDSP * /MPSIZE0 * /MPSIZE1
        + PDSP * MPAB0 * MPAB1
        + PDSP * MPAB1 * MPSIZE1
PDLLCSL.TRST = /OENL

/CPDSACK0L = MPAB20 * /CP0ASL
CPDSACK0L.TRST = /OENL * PDSP

/CPDSACK1L = MPAB20 * /CP0ASL
CPDSACK1L.TRST = /OENL * PDSP

/PERIFSEL = /MPAB22 * MPAB21 * MPAB20 * /CP0ASL
PERIFSEL.TRST = /OENL

/GPITSEL = MPAB22 * /MPAB21 * /MPAB20 * /CP0ASL
GPITSEL.TRST = /OENL

;---------------------------------- Simulation Segment ------------
SIMULATION

;INITIALIZE
SETF /MPAB0 /MPAB1 /MPSIZE0 /MPSIZE1 /MPAB22 /MPAB21 /MPAB20
/MPAB19 CP0ASL /OENL

;TEST TRISTATE CAPABILITY
SETF OENL
SETF /OENL

;TEST ALL 16 SIZE/ADDR COMBINATIONS WITH "PDSP" TERM TRUE
;THE "PDSP" TERMS WILL BE TESTED LATER
SETF MPAB20 /CP0ASL
SETF MPAB0
SETF MPAB1 /MPAB0
SETF MPAB0
SETF MPSIZE0 /MPAB1 /MPAB0
SETF MPAB0
SETF MPAB1 /MPAB0
SETF MPAB0
SETF MPSIZE1 /MPSIZE0 /MPAB1 /MPAB0
SETF MPAB0
SETF MPAB1 /MPAB0
SETF MPAB0
SETF MPSIZE0 /MPAB1 /MPAB0
SETF MPAB0
SETF MPAB1 /MPAB0
SETF MPAB0

;TEST TERMS OF "PDSP" AND PERIFSEL
SETF MPAB22
SETF /MPAB22
SETF MPAB21
SETF /MPAB21
SETF /MPAB20
SETF MPAB20
SETF MPAB19
SETF /MPAB19
SETF CP0ASL
SETF /CP0ASL

;-----------------------------------------------------------------
```

PALASM Design Description - PAL30

```
;---------------------------------Declaration Segment-------------
TITLE    HOST MEMORY R/W, SM REQUEST SIGNAL,
;          I/O BUFFER CONTROL - PAL30
PATTERN  LAPD
REVISION 1.23
AUTHOR   HOPETON WALKER
COMPANY
DATE     09/5/94

CHIP  PAL30  PAL22V10

;---------------------------------- PIN Declarations ---------------
PIN  1        CP0RWL                          ;
PIN  2        CP0DSL                          ;
PIN  3        CPSRWEH                         ;
PIN  4        CPEEWEH                         ;
PIN  5        MPAB18                          ;
PIN  6        MPAB19                          ;
PIN  7        MPAB20                          ;
PIN  8        MPAB21                          ;
PIN  9        MPAB22                          ;
PIN  10        CP0ASL                         ;
PIN  11        CPSMBGL                                ;
PIN  12        GND                            ;
PIN  13        MPAB24 ; (WAS OEL)             ;
PIN  14        CPSMBEL                        ;
PIN  15        CPDBSEL                        ;
PIN  16        CPSMBRL                                ;
PIN  17        CLDESPL                        ;
PIN  18        CLDSMSL                                ;
PIN  19        CPEEWVL                                ;
PIN  20        CPSRPWVL                        ;
PIN  21        CPEEWL                         ;
PIN  22        CPPDWL                         ;
PIN  23        CPMEMRL                                ;
PIN  24        VCC                            ;

;---------------------------------- Boolean Equation Segment ------

STRING EESPH '/MPAB22 * /MPAB21 * /MPAB20 * /MPAB19 * /MPAB18 *
/CP0ASL'
STRING PDSPH '/MPAB22 * /MPAB21 *  MPAB20 * /MPAB19 * /CP0ASL'
```

```
EQUATIONS

; Program, Data R/W signals

/CPPDWL = PDSPH * MPAB18 * /CP0ASL * /CP0RWL * /CP0DSL +
       CPSRWEH * PDSPH * /MPAB18 * /CP0ASL * /CP0RWL * /CP0DSL
/CPSRPWVL = /CPSRWEH * PDSPH * /MPAB18 * /CP0ASL * /CP0RWL * /CP0DSL

/CPEEWL = CPEEWEH * EESPH * /CP0RWL */CP0DSL

/CPEEWVL = /CPEEWEH * EESPH * /CP0RWL */CP0DSL

/CPMEMRL = CP0RWL

;Shared Memory Request and Strobes

/CPSMBRL = /MPAB24 * MPAB22 * MPAB21 * MPAB20 * MPAB19 * /CP0ASL
      + /MPAB22 * MPAB21 * /MPAB20 * /MPAB19 * /MPAB18 * /CP0ASL
/CLDSMSL = /MPAB24 * MPAB22 * MPAB21 * MPAB20 * MPAB19 * /CP0ASL
/CLDESPL = /MPAB22 * MPAB21 * /MPAB20 * /MPAB19 * /MPAB18 * /CP0ASL

; I/O Transceievr Control
; Transceivers for Shared Memory Array
/CPSMBEL =  /MPAB24 * MPAB22 * MPAB21 * MPAB20 * MPAB19 * /CP0ASL *
/CPSMBGL
  + /MPAB22 * MPAB21 * /MPAB20 * /MPAB19 * /MPAB18 * /CP0ASL *
/CPSMBGL

/CPDBSEL = /MPAB22 * MPAB21 * MPAB20 * /CP0ASL ;"3" Periferials
                              ;See PAL48 for CPIACK control
                              ;of these I/O buffers
;Tri-State GONE ....
;CPPDWL.TRST = /OEL
;CPSRPWVL.TRST = /OEL
;CPEEWL.TRST = /OEL
;CPEEWVL.TRST = /OEL
;CPMEMRL.TRST = /OEL
;CPSMBRL.TRST = /OEL
;CLDSMSL.TRST = /OEL
;CLDESPL.TRST = /OEL
;CPSMBEL.TRST = /OEL
;CPDBSEL.TRST = /OEL

SIMULATION
```

```
TRACE_ON
 MPAB18 MPAB19 MPAB20 MPAB21 MPAB22  CPSMBGL
   CPEEWEH CPSRWEH CP0RWL CP0DSL
CPPDWL CPSRPWVL CPEEWEH CPEEWVL CPSMBRL CLDSMSL CLDESPL
    CPSMBEL CPDBSEL

SETF  MPAB18 MPAB19 MPAB20 MPAB21 MPAB22  CPSMBGL
   /CPEEWEH /CPSRWEH CP0RWL CP0ASL CP0DSL
CHECK CPPDWL CPSRPWVL CPEEWVL CPSMBRL CLDSMSL CLDESPL
    CPSMBEL CPDBSEL

;I/O buffers - Peripherals
SETF  MPAB20 MPAB21 /MPAB22  CPSMBGL
    /CP0RWL /CP0ASL /CP0DSL
CHECK CPPDWL CPSRPWVL CPEEWVL CPSMBRL CLDSMSL CLDESPL
    CPSMBEL /CPDBSEL
SETF CP0ASL CP0RWL CP0DSL

;SM Buffers + CPSMBRL
SETF  MPAB24 MPAB18 MPAB19 MPAB20 MPAB21 MPAB22 CPSMBGL
    CP0ASL CP0RWL CP0DSL
CHECK CPPDWL CPSRPWVL CPEEWVL CPSMBRL CLDSMSL CLDESPL
    CPSMBEL CPDBSEL
;SRAM in SMA
SETF  /MPAB24 MPAB19 MPAB20 MPAB21 MPAB22 CPSMBGL
SETF  /CP0ASL /CP0RWL /CP0DSL
CHECK CPPDWL CPSRPWVL CPEEWVL /CPSMBRL /CLDSMSL CLDESPL
    CPSMBEL CPDBSEL
SETF   /CPSMBGL
CHECK CPPDWL CPSRPWVL CPEEWVL /CPSMBRL /CLDSMSL CLDESPL
    /CPSMBEL CPDBSEL
SETF  /MPAB18 ;No effect with MPAB18
CHECK CPPDWL CPSRPWVL CPEEWVL /CPSMBRL /CLDSMSL CLDESPL
    /CPSMBEL CPDBSEL
;Check each bit
SETF  MPAB18 MPAB19 /MPAB20 MPAB21 MPAB22 /CPSMBGL
CHECK CPPDWL CPSRPWVL CPEEWVL CPSMBRL CLDSMSL CLDESPL
    CPSMBEL CPDBSEL
SETF  MPAB18 MPAB19 MPAB20 /MPAB21 MPAB22 /CPSMBGL
CHECK CPPDWL CPSRPWVL CPEEWVL CPSMBRL CLDSMSL CLDESPL
    CPSMBEL CPDBSEL
SETF  MPAB18 MPAB19 MPAB20 MPAB21 /MPAB22 /CPSMBGL
CHECK CPPDWL CPSRPWVL CPEEWVL CPSMBRL CLDSMSL CLDESPL
    CPSMBEL /CPDBSEL
SETF  MPAB18 MPAB19 MPAB20 MPAB21 MPAB22 CP0ASL /CPSMBGL
```

```
CHECK CPPDWL CPSRPWVL CPEEWVL CPSMBRL CLDSMSL CLDESPL
    CPSMBEL CPDBSEL
SETF  MPAB18 MPAB19 MPAB20 MPAB21 MPAB22 /CP0ASL CPSMBGL
CHECK CPPDWL CPSRPWVL CPEEWVL /CPSMBRL /CLDSMSL CLDESPL
    CPSMBEL CPDBSEL
SETF  MPAB24
CHECK CPPDWL CPSRPWVL CPEEWVL CPSMBRL CLDSMSL CLDESPL
    CPSMBEL CPDBSEL

;LAPD EEPROM
SETF  /MPAB18 /MPAB19 /MPAB20 MPAB21 /MPAB22 CPSMBGL /CP0ASL
/CP0RWL /CP0DSL
CHECK CPPDWL CPSRPWVL CPEEWVL
CHECK  /CPSMBRL CLDSMSL /CLDESPL CPSMBEL CPDBSEL
SETF  /CPSMBGL
CHECK CPPDWL CPSRPWVL CPEEWVL /CPSMBRL CLDSMSL /CLDESPL
/CPSMBEL CPDBSEL

; LAPD EEPROM Shared Memory write violation
SETF  MPAB18 MPAB19 MPAB20 MPAB21 MPAB22 CPSMBGL
    CP0ASL CP0RWL CP0DSL
SETF  MPAB18 MPAB19 MPAB20 MPAB21 MPAB22 CPSMBGL
   CP0ASL CP0RWL CP0DSL
SETF  /MPAB18 /MPAB19 /MPAB20 /MPAB21 /MPAB22 /CP0ASL /CP0RWL
/CP0DSL
CHECK CPEEWL /CPEEWVL
SETF CPEEWEH
CHECK /CPEEWL CPEEWVL
SETF  MPAB18 MPAB19 MPAB20 MPAB21 MPAB22 CPSMBGL
    CP0ASL CP0RWL CP0DSL
CHECK CPPDWL CPSRPWVL CPEEWL CPEEWVL
SETF  MPAB18 MPAB19 MPAB20 MPAB21 MPAB22 CPSMBGL
    CP0ASL CP0RWL CP0DSL
SETF  /MPAB19 MPAB20 /MPAB21 /MPAB22 /CP0ASL /CP0RWL /CP0DSL
CHECK /CPPDWL CPSRPWVL
SETF CPSRWEH /MPAB18
CHECK /CPPDWL CPSRPWVL
SETF CPSRWEH
CHECK /CPPDWL CPSRPWVL
SETF /CPSRWEH
CHECK CPPDWL /CPSRPWVL
SETF /CPSRWEH
CHECK CPPDWL /CPSRPWVL

TRACE_OFF
```

PALASM Design Description - PAL32

```
;---------------------------------- Declaration Segment ------------
TITLE   PERIPHERAL SELECT - PAL32
PATTERN  LAPD
REVISION 1.09
AUTHOR   HOPETON WALKER
COMPANY
DATE     08/01/94

CHIP     PAL32  PAL26V12

;---------------------------------- PIN Declarations ---------------
PIN  1       PERIFCLK     COMBINATORIAL   ; CLOCK
PIN  2       PERIFSEL                     ; GLOBAL DECODED SIGNAL
PIN  3       BIMRSTL
PIN  4       MPAB7
PIN  5       MPAB6
PIN  6       MPAB5
PIN  7       VCC
PIN  8       MPAB4
PIN  9       MPAB3
PIN  10       MPAB2
PIN  11       MPAB1
PIN  12       MPAB0
PIN  13       CP0ASL
PIN  14       CP0DSL
PIN  15       CPDS1L       REGISTERED
PIN  16       PERIFCSL      REGISTERED       ; DELAYED PERIFSEL FOR CHIP
SELECTS
PIN  17       PERIFDKL
PIN  18       GPSTATL
PIN  19       IRSL
PIN  20       CR1SL
PIN  21       GND
PIN  22       CR0SL
PIN  23       CPUIRSL
PIN  24       HIFICSL
PIN  25       UARTCSL
PIN  26       BIMCSL
PIN  27       MFPCSL
PIN  28       OENL

NODE 1        RSTPIN
;---------------------------------- Boolean Equation Segment ------
```

```
EQUATIONS
;RESET FOR LATCHES
RSTPIN.RSTF = CP0ASL

/CPDS1L := /CP0DSL * /PERIFSEL
/PERIFCSL := /CPDS1L
CPDS1L.TRST = /OENL
PERIFCSL.TRST = /OENL

/MFPCSL = /MPAB7 * /MPAB6 * /MPAB5 * /MPAB4 * /CP0ASL * /PERIFCSL
       + /MPAB7 * /MPAB6 * /MPAB5 * MPAB4 * /CP0ASL * /MPAB3 * /PERIFCSL
MFPCSL.TRST = /OENL

/BIMCSL = /MPAB7 * /MPAB6 * MPAB5 * MPAB4 * MPAB0 */CP0ASL *
/PERIFCSL
       + /BIMRSTL
BIMCSL.TRST = /OENL

/HIFICSL = MPAB7 * /MPAB6 * /MPAB5 * /MPAB4 * /CP0ASL * /PERIFCSL
HIFICSL.TRST = /OENL

/UARTCSL = MPAB7 * MPAB6 * /MPAB5 * /MPAB4 * /CP0ASL * /PERIFCSL
UARTCSL.TRST = /OENL

/CPUIRSL = MPAB7 * MPAB6 * MPAB5 * MPAB4 * /MPAB3 * /MPAB2 * /MPAB1
* /MPAB0 *
        /CP0ASL * /PERIFCSL
CPUIRSL.TRST = /OENL

/CR0SL = MPAB7 * MPAB6 * MPAB5 * MPAB4 * /MPAB3 * /MPAB2 * MPAB1 *
/MPAB0 *
        /CP0ASL * /PERIFCSL
CR0SL.TRST = /OENL

/CR1SL = MPAB7 * MPAB6 * MPAB5 * MPAB4 * /MPAB3 * MPAB2 * /MPAB1 *
/MPAB0 *
        /CP0ASL * /PERIFCSL
CR1SL.TRST = /OENL

/IRSL = MPAB7 * MPAB6 * MPAB5 * MPAB4 * /MPAB3 * MPAB2 * MPAB1 *
/MPAB0 *
        /CP0ASL * /PERIFCSL
IRSL.TRST = /OENL
```

```
/GPSTATL = MPAB7 * MPAB6 * MPAB5 * MPAB4 * MPAB3 * /MPAB2 * /MPAB1
* /MPAB0 *
        /CP0ASL * /PERIFCSL
GPSTATL.TRST = /OENL

/PERIFDKL = MPAB7 * MPAB6 * MPAB5 * MPAB4 * /MPAB0 *
        /CP0ASL * /PERIFCSL
PERIFDKL.TRST = /OENL

;------------------------------------ Simulation Segment ------------
SIMULATION
TRACE_ON
     PERIFCLK  PERIFSEL BIMRSTL MPAB7 MPAB6 MPAB5 MPAB4
     MPAB3 MPAB2 MPAB1 MPAB0 CP0ASL CP0DSL CPDS1L
     PERIFCSL PERIFDKL GPSTATL IRSL CR1SL
     CR0SL CPUIRSL HIFICSL UARTCSL
     BIMCSL MFPCSL
     OENL

;INITIALIZE
SETF /PERIFCLK PERIFSEL /BIMRSTL
SETF /MPAB7 /MPAB6 /MPAB5 /MPAB4 /MPAB3 /MPAB2 /MPAB1 /MPAB0
SETF CP0ASL CP0DSL /OENL
CHECK /BIMCSL

;TEST TRISTATE
SETF OENL
SETF /OENL

;TEST MFP CHIP SELECT
SETF /PERIFSEL /CP0ASL /CP0DSL
CLOCKF PERIFCLK
CLOCKF PERIFCLK
SETF MPAB0
SETF MPAB1 /MPAB0
SETF MPAB0
SETF MPAB2 /MPAB1 /MPAB0
SETF MPAB0
SETF MPAB1 /MPAB0
SETF MPAB0
SETF MPAB3 /MPAB2 /MPAB1 /MPAB0
SETF MPAB0
SETF MPAB1 /MPAB0
SETF MPAB0
SETF MPAB2 /MPAB1 /MPAB0
SETF MPAB0
SETF MPAB1 /MPAB0
SETF MPAB0
SETF MPAB4 /MPAB3 /MPAB2 /MPAB1 /MPAB0
SETF MPAB0
SETF MPAB1 /MPAB0
SETF MPAB0
SETF MPAB2 /MPAB1 /MPAB0
SETF MPAB0
SETF MPAB1 /MPAB0
SETF MPAB0
SETF MPAB3 /MPAB2 /MPAB1 /MPAB0

;TEST BIM CHIP SELECT - BIMRSTL A LOW
CHECK /BIMCSL
SETF BIMRSTL
SETF MPAB5 MPAB4 ;DECODE "3"
SETF MPAB0
CHECK /BIMCSL
SETF /MPAB0
CHECK BIMCSL
SETF MPAB0
CHECK /BIMCSL
SETF /MPAB5
CHECK BIMCSL
SETF MPAB5
CHECK /BIMCSL
SETF /MPAB4
CHECK BIMCSL
SETF MPAB4
CHECK /BIMCSL
SETF MPAB7
CHECK BIMCSL
SETF /MPAB7
CHECK /BIMCSL
SETF MPAB6
CHECK BIMCSL

;TEST HIFI CHIP SELECT
SETF MPAB7 /MPAB6 /MPAB5 /MPAB4
SETF MPAB0
SETF MPAB1 /MPAB0
SETF MPAB0
SETF MPAB2 /MPAB1 /MPAB0
SETF MPAB0
SETF MPAB1 /MPAB0
SETF MPAB0
SETF MPAB3 /MPAB2 /MPAB1 /MPAB0
SETF MPAB0
SETF MPAB1 /MPAB0
SETF MPAB0
SETF MPAB2 /MPAB1 /MPAB0
SETF MPAB0
SETF MPAB1 /MPAB0
SETF MPAB0
SETF MPAB4 /MPAB3 /MPAB2 /MPAB1 /MPAB0

;TEST UART CHIP SELECT
SETF MPAB6 /MPAB4
SETF MPAB0
SETF MPAB1 /MPAB0
SETF MPAB0
SETF MPAB2 /MPAB1 /MPAB0
SETF MPAB0
SETF MPAB1 /MPAB0
SETF MPAB0
SETF MPAB3 /MPAB2 /MPAB1 /MPAB0
SETF MPAB0
SETF MPAB1 /MPAB0
SETF MPAB0
SETF MPAB2 /MPAB1 /MPAB0
SETF MPAB0
SETF MPAB1 /MPAB0
SETF MPAB0
SETF MPAB4 /MPAB3 /MPAB2 /MPAB1 /MPAB0

;TEST PERIF DSACK GENERATION VIA FOLLOWING TESTS
;TEST CPU INTERRUPT REQ CHIP SELECT
SETF MPAB5
SETF MPAB0

;TEST CR0 CHIP SELECT
SETF MPAB1
SETF /MPAB0
SETF MPAB0

;TEST CR1 CHIP SELECT
SETF MPAB2 /MPAB1
SETF /MPAB0
SETF MPAB0
```

```
;TEST INTERRUPT STAT CHIP SELECT
SETF MPAB1
SETF /MPAB0
SETF MPAB0

;TEST GPITS STAT CHIP SELECT
SETF MPAB3 /MPAB2 /MPAB1
SETF /MPAB0
SETF MPAB0

;TEST FOR LATCHES RESET
SETF CP0ASL
SETF CP0ASL
SETF CP0ASL
SETF /CP0ASL
SETF /CP0ASL
SETF /CP0ASL
CLOCKF PERIFCLK
CLOCKF PERIFCLK


TRACE_OFF
;------------------------------------------------------------------
```

PALASM Design Description - PAL34

```
;-------------------------------Declaration Degment---------
TITLE   EEPROM PROGRAM MEMORY CHIP SELECT - PAL34
PATTERN  LAPD
REVISION 1.04
AUTHOR   HOPETON WALKER
COMPANY
DATE     08/10/94

CHIP     HOST    PAL16L8

;-----------------------------PIN Declatations-------------
;PINS
PIN     1        MPAB0                          ;I0
PIN     2        MPSIZE0              ;I1
PIN     3        MPSIZE1              ;I2
PIN     4        MPAB22               ;I3
PIN     5        MPAB21               ;I4
PIN     6        MPAB20               ;I5
PIN     7        MPAB19               ;I6
PIN     8        MPAB17               ;I7
PIN     9        CP0ASL               ;I8
PIN     10       GND                  ;GND
PIN     11       CPSMBR1L ;OEL        ;I9
PIN     12       NC                   ;O7
PIN     13       MPAB18               ;I/O6
PIN     14       CPEESL               ;I/O5
PIN     15       CPELECSL             ;I/O4
PIN     16       CPEUECSL             ;I/O3
PIN     17       CPELOCSL             ;I/O2
PIN     18       CPEUOCSL             ;I/O1
PIN     19       CPDKRSTL             ;O0
PIN     20       VCC                  ;VCC

STRING PEESP '/MPAB22 * /MPAB21 * /MPAB20 * /MPAB19 * /MPAB18 *
/CP0ASL'

EQUATIONS

        CPEUECSL.TRST    =       VCC
        CPELECSL.TRST    =       VCC
        CPEUOCSL.TRST    =       VCC
        CPEESL.TRST              =       VCC
        CPELOCSL.TRST    =       VCC
        CPDKRSTL.TRST    =       VCC

        /CPEUECSL  =        PEESP * MPAB17 * /MPAB0
        /CPELECSL  =        PEESP * /MPAB17 * /MPAB0

        /CPEUOCSL  =        PEESP * MPAB17 * MPAB0 +
                            PEESP * MPAB17 * /MPSIZE0 +
                            PEESP * MPAB17 * MPSIZE1

        /CPELOCSL  =        PEESP * /MPAB17 * MPAB0 +
                            PEESP * /MPAB17 * /MPSIZE0 +
                            PEESP * /MPAB17 * MPSIZE1

        /CPEESL          =       /CPEUECSL + /CPELECSL + /CPEUOCSL +
/CPELOCSL

        /CPDKRSTL  =        CP0ASL + CPSMBR1L


SIMULATION

TRACE_ON  MPAB0 MPSIZE0 MPSIZE1 MPAB22 MPAB21 MPAB20
          MPAB19 MPAB17 CP0ASL CPDKRSTL CPEESL CPELECSL
          CPEUECSL CPELOCSL CPEUOCSL CPSMBR1L

SETF  MPAB0 MPSIZE0 MPSIZE1 MPAB22 MPAB21 MPAB20 MPAB19 MPAB18
MPAB17 CP0ASL
     CPSMBR1L
CHECK       /CPDKRSTL CPEESL CPELECSL CPEUECSL CPELOCSL
CPEUOCSL
SETF  /CP0ASL
CHECK       /CPDKRSTL
SETF  /CPSMBR1L
CHECK       CPDKRSTL

SETF  /MPAB18
SETF  MPAB0 MPSIZE0 MPSIZE1 MPAB22 MPAB21 MPAB20 MPAB19 MPAB17
CP0ASL
SETF  /MPAB0 MPSIZE0 MPSIZE1 /MPAB22 /MPAB21 /MPAB20 /MPAB19
MPAB17 /CP0ASL
SETF  /MPAB0 MPSIZE0 MPSIZE1 /MPAB22 /MPAB21 /MPAB20 /MPAB19
MPAB17 /CP0ASL
CHECK       /CPEUECSL /CPEESL
```

```
SETF  MPAB0 MPSIZE0 MPSIZE1 MPAB22 MPAB21 MPAB20 MPAB19 MPAB17
CP0ASL
SETF  MPAB0 MPSIZE0 MPSIZE1 MPAB22 MPAB21 MPAB20 MPAB19 MPAB17
CP0ASL
CHECK        CPEUECSL CPEESL

SETF  /MPAB0 MPSIZE0 MPSIZE1 /MPAB22 /MPAB21 /MPAB20 /MPAB19
/MPAB17 /CP0ASL
SETF  /MPAB0 MPSIZE0 MPSIZE1 /MPAB22 /MPAB21 /MPAB20 /MPAB19
/MPAB17 /CP0ASL
CHECK        CPDKRSTL /CPELECSL /CPEESL

SETF  MPAB0 MPSIZE0 MPSIZE1 MPAB22 MPAB21 MPAB20 MPAB19 MPAB17
CP0ASL
SETF  MPAB0 MPSIZE0 MPSIZE1 MPAB22 MPAB21 MPAB20 MPAB19 MPAB17
CP0ASL
CHECK        CPELECSL CPEESL

SETF  MPAB0 MPSIZE0 MPSIZE1 MPAB22 MPAB21 MPAB20 MPAB19 MPAB17
CP0ASL
CHECK        CPEESL CPELECSL CPEUECSL CPELOCSL CPEUOCSL

TRACE_OFF
```

PALASM Design Description - PAL36

```
;------------------------------Declaration Segment----------
TITLE    MEMORY WRITE VIOLATION - PAL36
PATTERN  LAPD
REVISION 1.15
AUTHOR   HOPETON WALKER
COMPANY
DATE     1/09/94

;Memory Write Violations PAL20RA10

CHIP   HOSTPAL     PAL20RA10

;----------------------------PIN Declarations--------------
;
;PINS                DIP PINS      PLCC PINS

PL                   ;PL    1            2
CPSRPWVL             ;I0    2            3
CPEEWVL              ;I1    3            4
LPEEWV               ;I2    4            5
ISRDL                ;I3    5            6
IRSRCL                      ;I4    6            7
NC                   ;I5    7            9
SPYSMSPL             ;I6    8            10
SPYIACKL             ;I7    9            11
HGH                  ;I8    10           12
NC                   ;I9    11           13
GND                  ;GND   12           14
OEL                  ;OEL   13           16
NC                   ;IO0   14           17
SPYASL               ;IO1   15           18
EEWVL                       ;IO2   16           19
CPWVL                       ;IO3   17           20
LPWVL                       ;IO4   18           21
SPYIAL                      ;IO5   19           23
CPDB29               ;IO6   20           24
CPDB30               ;I07   21           25
CPDB31               ;I08   22           26
WRTVL                       ;IO9   23           27
VCC                  ;VCC   24           28


EQUATIONS
```

```
    /CPDB31        :=      HGH
    CPDB31.CLKF            =       /HGH
    CPDB31.RSTF            =       /CPEEWVL
    CPDB31.TRST            =       /ISRDL
    CPDB31.SETF            =       /IRSRCL

    /CPWVL         :=      HGH
    CPWVL.TRST             =       HGH
    CPWVL.CLKF             =       /HGH
    CPWVL.RSTF             =       /IRSRCL
    CPWVL.SETF             =       /CPSRPWVL

    /CPDB30        :=      HGH
    CPDB30.CLKF            =       /HGH
    CPDB30.RSTF            =       /CPSRPWVL
    CPDB30.TRST            =       /ISRDL
    CPDB30.SETF            =       /IRSRCL

    /EEWVL         :=      HGH
    EEWVL.TRST             =       HGH
    EEWVL.CLKF             =       /HGH
    EEWVL.RSTF             =       /IRSRCL
    EEWVL.SETF             =       /CPEEWVL

    /CPDB29        :=      HGH
    CPDB29.CLKF            =       /HGH
    CPDB29.RSTF            =       /LPEEWV
    CPDB29.TRST            =       /ISRDL
    CPDB29.SETF            =       /IRSRCL

    LPWVL          :=      HGH
    LPWVL.TRST             =       HGH
    LPWVL.CLKF             =       /HGH
    LPWVL.RSTF             =       /IRSRCL
    LPWVL.SETF             =       /LPEEWV

    WRTVL          =       CPWVL * EEWVL * LPWVL
WRTVL.TRST         =       HGH

    /SPYIAL                :=      SPYSMSPL
    SPYIAL.TRST            =       HGH
    SPYIAL.CLKF            =       SPYASL
    SPYIAL.RSTF            =       /SPYIACKL
```

```
SIMULATION

TRACE_ON  CPSRPWVL CPEEWVL LPEEWV IRSRCL CPDB31
CPDB30 CPDB29 WRTVL ISRDL SPYSMSPL SPYIACKL SPYASL SPYIAL

SETF  PL /OEL HGH CPSRPWVL CPEEWVL LPEEWV /ISRDL /IRSRCL
SPYIACKL SPYSMSPL SPYASL
SETF  CPSRPWVL CPEEWVL LPEEWV /ISRDL /IRSRCL SPYIACKL SPYSMSPL
/SPYASL
CHECK       WRTVL /CPDB31 /CPDB30 /CPDB29 /SPYIAL

SETF  CPSRPWVL CPEEWVL LPEEWV IRSRCL
CHECK       WRTVL /CPDB31 /CPDB30 /CPDB29 /SPYIAL

SETF  CPSRPWVL CPEEWVL LPEEWV
SETF  CPSRPWVL CPEEWVL LPEEWV
CHECK       WRTVL /CPDB31 /CPDB30 /CPDB29 /SPYIAL

SETF  CPSRPWVL /CPEEWVL LPEEWV /SPYSMSPL SPYIACKL /SPYASL
SETF  CPSRPWVL /CPEEWVL LPEEWV /SPYSMSPL SPYIACKL
CHECK       /WRTVL CPDB31 /CPDB30 /CPDB29 /SPYIAL

SETF  CPSRPWVL CPEEWVL LPEEWV SPYASL
SETF  CPSRPWVL CPEEWVL LPEEWV
CHECK       /WRTVL CPDB31 /CPDB30 /CPDB29

SETF  SPYIACKL /SPYASL
CHECK       SPYIAL

SETF  ISRDL /SPYASL /SPYSMSPL SPYIACKL
SETF  /ISRDL /SPYSMSPL SPYIACKL
CHECK       /WRTVL CPDB31 /CPDB30 /CPDB29 SPYIAL

SETF  CPEEWVL /SPYASL
SETF  SPYASL SPYSMSPL
SETF  SPYASL
CHECK       /WRTVL CPDB31 /CPDB30 /CPDB29 /SPYIAL

SETF  /IRSRCL SPYSMSPL /SPYASL
SETF  IRSRCL
CHECK       WRTVL /CPDB31 /CPDB30 /CPDB29 /SPYIAL

SETF  SPYSMSPL
SETF  /SPYIACKL
SETF  /SPYIACKL
SETF  /SPYIACKL
CHECK        SPYIAL

SETF  /SPYIACKL /SPYASL
SETF  /SPYIACKL /SPYASL
CHECK        SPYIAL

SETF  CPEEWVL
SETF  /SPYASL SPYSMSPL SPYIACKL
CHECK        WRTVL /CPDB31 /CPDB30 /CPDB29 SPYIAL

SETF  /CPSRPWVL
SETF  /CPSRPWVL
CHECK        /WRTVL CPDB30 SPYIAL

SETF  CPSRPWVL ISRDL SPYASL SPYIACKL
SETF  /ISRDL
CHECK        /WRTVL CPDB30

SETF  /IRSRCL
CHECK        WRTVL /CPDB31 /CPDB30 /CPDB29

SETF  IRSRCL
CHECK        WRTVL /CPDB31 /CPDB30 /CPDB29

SETF  /LPEEWV
CHECK        /WRTVL /CPDB31 /CPDB30 CPDB29

SETF  LPEEWV ISRDL
SETF  /ISRDL
CHECK        /WRTVL CPDB29

SETF  LPEEWV
CHECK        /WRTVL CPDB29

SETF  /IRSRCL
CHECK        WRTVL /CPDB31 /CPDB30 /CPDB29

SETF  IRSRCL
CHECK        WRTVL /CPDB31 /CPDB30 /CPDB29

SETF  /HGH
CHECK        ^WRTVL ^EEWVL ^CPWVL ^SPYIAL

TRACE_OFF
```

PALASM Design Description - PAL38

```
;------------------------------Declaration Segment--------
TITLE   HOST SHARED MEMORY R/W STROBES - PAL38
PATTERN  LAPD
REVISION 1.13
AUTHOR   HOPETON WALKER
COMPANY
DATE     08/22/94

CHIP   HOST     PAL16L8

;------------------------------PIN Declarations------------
;     PINS
PIN    1       CLDESPL
PIN    2       CPSMBGL
PIN    3       CLDSMSL
PIN    4       CP0RWL
PIN    5       CP0ASL
PIN    6       CP0DSL
PIN    7       MPAB0
PIN    8       MPSIZE0
PIN    9       MPSIZE1
PIN    10      GND
PIN    11      OEL
PIN    12      NC
PIN    13      CPSM2DBGL
PIN    14      NC
PIN    15      NC
PIN    16      SMWEL
PIN    17      SMWOL
PIN    18      SMREL
PIN    19      SMROL
PIN    20      VCC

STRING SMRDH '/CPSMBGL * /CP0DSL * /CP0ASL * (/CLDSMSL + /CLDESPL)'
STRING SMWTH '/CPSMBGL * /CP0DSL * /CP0ASL * /CLDSMSL'

EQUATIONS

        SMWEL.TRST      =      /CPSMBGL * /CPSM2DBGL * /OEL

        SMREL.TRST      =      /CPSMBGL * /OEL

        SMWOL.TRST      =      /CPSMBGL * /CPSM2DBGL * /OEL
        SMROL.TRST          =       /CPSMBGL * /OEL

        /SMWEL              =       SMWTH * /CP0RWL * /MPAB0

        /SMREL              =       SMRDH * CP0RWL * /MPAB0

        /SMWOL              =       SMWTH * MPAB0 * /CP0RWL +
                            SMWTH * /MPSIZE0 * /CP0RWL +
                            SMWTH * MPSIZE1 * /CP0RWL

        /SMROL              =       SMRDH * MPAB0 * CP0RWL +
                            SMRDH * /MPSIZE0 * CP0RWL +
                            SMRDH * MPSIZE1 * CP0RWL

SIMULATION

TRACE_ON   OEL CPSMBGL CLDESPL CLDSMSL CP0RWL CP0ASL CP0DSL
           MPAB0 MPSIZE0 MPSIZE1 SMWEL SMWOL SMREL SMROL

SETF  /OEL /CPSMBGL /CPSM2DBGL
SETF  CLDSMSL CP0RWL /CP0ASL CP0DSL MPAB0 MPSIZE0 MPSIZE1
SETF  /CPSMBGL CLDESPL CLDSMSL CP0RWL /CP0ASL CP0DSL MPAB0
MPSIZE0 MPSIZE1
CHECK      SMWEL SMWOL SMREL SMROL

SETF  /CPSMBGL CLDSMSL CP0RWL /CP0ASL CP0DSL MPAB0 MPSIZE0
MPSIZE1
SETF  /CPSMBGL CLDSMSL CP0RWL /CP0ASL CP0DSL MPAB0 MPSIZE0
MPSIZE1
CHECK      SMWEL SMWOL SMREL SMROL

SETF  /CPSMBGL /CLDSMSL /CP0RWL /CP0ASL /CP0DSL /MPAB0
SETF  /CPSMBGL /CLDSMSL /CP0RWL /CP0ASL /CP0DSL /MPAB0
CHECK      /SMWEL

SETF  /CPSMBGL CLDSMSL CP0RWL /CP0ASL CP0DSL MPAB0 MPSIZE0
MPSIZE1
SETF  /CPSMBGL CLDSMSL CP0RWL /CP0ASL CP0DSL MPAB0 MPSIZE0
MPSIZE1
CHECK      SMWEL SMWOL SMREL SMROL

SETF  /CPSMBGL /CLDSMSL CP0RWL /CP0ASL /CP0DSL /MPAB0
SETF  /CPSMBGL /CLDSMSL CP0RWL /CP0ASL /CP0DSL /MPAB0
CHECK      /SMREL
```

```
SETF  /CPSMBGL CLDSMSL CP0RWL /CP0ASL CP0DSL MPAB0 MPSIZE0
MPSIZE1
SETF  /CPSMBGL CLDSMSL CP0RWL /CP0ASL CP0DSL MPAB0 MPSIZE0
MPSIZE1
CHECK        SMREL

SETF  /CPSMBGL /CLDSMSL /CP0RWL /CP0ASL /CP0DSL /MPAB0 /MPSIZE0
MPSIZE1
SETF  /CPSMBGL /CLDSMSL /CP0RWL /CP0ASL /CP0DSL /MPAB0 /MPSIZE0
MPSIZE1
CHECK        /SMWOL

SETF  /CPSMBGL CLDSMSL CP0RWL /CP0ASL CP0DSL MPAB0 MPSIZE0
MPSIZE1
SETF  /CPSMBGL CLDSMSL CP0RWL /CP0ASL CP0DSL MPAB0 MPSIZE0
MPSIZE1
CHECK        SMWOL

SETF  /CPSMBGL /CLDSMSL CP0RWL /CP0ASL /CP0DSL MPAB0 MPSIZE0
MPSIZE1
SETF  /CPSMBGL /CLDSMSL CP0RWL /CP0ASL /CP0DSL MPAB0 MPSIZE0
MPSIZE1
CHECK        /SMROL

SETF  /CPSMBGL CLDSMSL CP0RWL /CP0ASL CP0DSL MPAB0 MPSIZE0
MPSIZE1
SETF  /CPSMBGL CLDSMSL CP0RWL /CP0ASL CP0DSL MPAB0 MPSIZE0
MPSIZE1
CHECK        SMROL

SETF  /CPSMBGL /CLDSMSL /CP0RWL /CP0ASL /CP0DSL MPAB0 MPSIZE0
MPSIZE1
SETF  /CPSMBGL /CLDSMSL /CP0RWL /CP0ASL /CP0DSL MPAB0 MPSIZE0
MPSIZE1
CHECK        /SMWOL

SETF  /CPSMBGL CLDSMSL CP0RWL /CP0ASL CP0DSL MPAB0 MPSIZE0
MPSIZE1
SETF  /CPSMBGL CLDSMSL CP0RWL /CP0ASL CP0DSL MPAB0 MPSIZE0
MPSIZE1
CHECK        SMWOL

SETF  /CPSMBGL /CLDSMSL CP0RWL /CP0ASL /CP0DSL /MPAB0
SETF  /CPSMBGL /CLDSMSL CP0RWL /CP0ASL /CP0DSL /MPAB0
```

```
CHECK       /SMREL

SETF  /CPSMBGL CLDSMSL CP0RWL /CP0ASL CP0DSL MPAB0 MPSIZE0
MPSIZE1
SETF  /CPSMBGL CLDSMSL CP0RWL /CP0ASL CP0DSL MPAB0 MPSIZE0
MPSIZE1
CHECK       SMREL

SETF  /CPSMBGL /CLDSMSL /CP0RWL /CP0ASL /CP0DSL /MPAB0 /MPSIZE1
SETF  /CPSMBGL /CLDSMSL /CP0RWL /CP0ASL /CP0DSL /MPAB0
CHECK       /SMWEL

SETF  /CPSMBGL CLDSMSL CP0RWL /CP0ASL CP0DSL MPAB0 MPSIZE0
MPSIZE1
SETF  /CPSMBGL CLDSMSL CP0RWL /CP0ASL CP0DSL MPAB0 MPSIZE0
MPSIZE1
CHECK       SMWEL

SETF  /CPSMBGL /CLDSMSL CP0RWL /CP0ASL /CP0DSL /MPAB0  MPSIZE0
/MPSIZE1
SETF  /CPSMBGL /CLDSMSL CP0RWL /CP0ASL /CP0DSL /MPAB0
CHECK       /SMREL SMROL

SETF  /CPSMBGL CLDSMSL CP0RWL /CP0ASL CP0DSL MPAB0 MPSIZE0
MPSIZE1
SETF  /CPSMBGL CLDSMSL CP0RWL /CP0ASL CP0DSL MPAB0 MPSIZE0
MPSIZE1
CHECK       SMREL

SETF  /CPSMBGL /CLDSMSL /CP0RWL /CP0ASL /CP0DSL /MPAB0 /MPSIZE0
/MPSIZE1
SETF  /CPSMBGL /CLDSMSL /CP0RWL /CP0ASL /CP0DSL /MPAB0 /MPSIZE0
/MPSIZE1
CHECK       /SMWOL /SMWEL

SETF  /CPSMBGL CLDSMSL CP0RWL /CP0ASL CP0DSL MPAB0 MPSIZE0
MPSIZE1
SETF  /CPSMBGL CLDSMSL CP0RWL /CP0ASL CP0DSL MPAB0 MPSIZE0
MPSIZE1
CHECK       SMWOL SMWEL

SETF  /CPSMBGL CLDESPL CP0RWL /CP0ASL CP0DSL MPAB0 MPSIZE0
MPSIZE1
CHECK       SMWEL SMWOL SMREL SMROL
```

```
SETF  /CPSMBGL CLDESPL CP0RWL /CP0ASL CP0DSL MPAB0 MPSIZE0
MPSIZE1
SETF  /CPSMBGL CLDESPL CP0RWL /CP0ASL CP0DSL MPAB0 MPSIZE0
MPSIZE1
CHECK       SMWEL SMWOL SMREL SMROL

SETF  /CPSMBGL /CLDESPL /CP0RWL /CP0ASL /CP0DSL /MPAB0
SETF  /CPSMBGL /CLDESPL /CP0RWL /CP0ASL /CP0DSL /MPAB0
CHECK       SMWEL

SETF  /CPSMBGL CLDESPL CP0RWL /CP0ASL CP0DSL MPAB0 MPSIZE0
MPSIZE1
SETF  /CPSMBGL CLDESPL CP0RWL /CP0ASL CP0DSL MPAB0 MPSIZE0
MPSIZE1
CHECK       SMWEL SMWOL SMREL SMROL

SETF  /CPSMBGL /CLDESPL CP0RWL /CP0ASL /CP0DSL /MPAB0
SETF  /CPSMBGL /CLDESPL CP0RWL /CP0ASL /CP0DSL /MPAB0
CHECK       /SMREL

SETF  /CPSMBGL CLDESPL CP0RWL /CP0ASL CP0DSL MPAB0 MPSIZE0
MPSIZE1
SETF  /CPSMBGL CLDESPL CP0RWL /CP0ASL CP0DSL MPAB0 MPSIZE0
MPSIZE1
CHECK       SMREL

SETF  /CPSMBGL /CLDESPL /CP0RWL /CP0ASL /CP0DSL /MPAB0 /MPSIZE0
MPSIZE1
SETF  /CPSMBGL /CLDESPL /CP0RWL /CP0ASL /CP0DSL /MPAB0 /MPSIZE0
MPSIZE1
CHECK       SMWOL

SETF  /CPSMBGL CLDESPL CP0RWL /CP0ASL CP0DSL MPAB0 MPSIZE0
MPSIZE1
SETF  /CPSMBGL CLDESPL CP0RWL /CP0ASL CP0DSL MPAB0 MPSIZE0
MPSIZE1
CHECK       SMWOL

SETF  /CPSMBGL /CLDESPL CP0RWL /CP0ASL /CP0DSL MPAB0 MPSIZE0
MPSIZE1
SETF  /CPSMBGL /CLDESPL CP0RWL /CP0ASL /CP0DSL MPAB0 MPSIZE0
MPSIZE1
CHECK       /SMROL
```

```
SETF  /CPSMBGL CLDESPL CP0RWL /CP0ASL CP0DSL MPAB0 MPSIZE0
MPSIZE1
SETF  /CPSMBGL CLDESPL CP0RWL /CP0ASL CP0DSL MPAB0 MPSIZE0
MPSIZE1
CHECK        SMROL

SETF  /CPSMBGL /CLDESPL /CP0RWL /CP0ASL /CP0DSL MPAB0 MPSIZE0
MPSIZE1
SETF  /CPSMBGL /CLDESPL /CP0RWL /CP0ASL /CP0DSL MPAB0 MPSIZE0
MPSIZE1
CHECK        SMWOL

SETF  /CPSMBGL CLDESPL CP0RWL /CP0ASL CP0DSL MPAB0 MPSIZE0
MPSIZE1
SETF  /CPSMBGL CLDESPL CP0RWL /CP0ASL CP0DSL MPAB0 MPSIZE0
MPSIZE1
CHECK        SMWOL

SETF  /CPSMBGL /CLDESPL CP0RWL /CP0ASL /CP0DSL /MPAB0
SETF  /CPSMBGL /CLDESPL CP0RWL /CP0ASL /CP0DSL /MPAB0
CHECK        /SMREL

SETF  /CPSMBGL CLDESPL CP0RWL /CP0ASL CP0DSL MPAB0 MPSIZE0
MPSIZE1
SETF  /CPSMBGL CLDESPL CP0RWL /CP0ASL CP0DSL MPAB0 MPSIZE0
MPSIZE1
CHECK        SMREL

SETF  /CPSMBGL /CLDESPL /CP0RWL /CP0ASL /CP0DSL /MPAB0 /MPSIZE1
SETF  /CPSMBGL /CLDESPL /CP0RWL /CP0ASL /CP0DSL /MPAB0
CHECK        SMWEL

SETF  /CPSMBGL CLDESPL CP0RWL /CP0ASL CP0DSL MPAB0 MPSIZE0
MPSIZE1
SETF  /CPSMBGL CLDESPL CP0RWL /CP0ASL CP0DSL MPAB0 MPSIZE0
MPSIZE1
CHECK        SMWEL

SETF  /CPSMBGL /CLDESPL CP0RWL /CP0ASL /CP0DSL /MPAB0  MPSIZE0
/MPSIZE1
SETF  /CPSMBGL /CLDESPL CP0RWL /CP0ASL /CP0DSL /MPAB0
CHECK        /SMREL SMROL

SETF  /CPSMBGL CLDESPL CP0RWL /CP0ASL CP0DSL MPAB0 MPSIZE0
MPSIZE1
```

```
SETF  /CPSMBGL CLDESPL CP0RWL /CP0ASL CP0DSL MPAB0 MPSIZE0
MPSIZE1
CHECK        SMREL

SETF  /CPSMBGL /CLDESPL /CP0RWL /CP0ASL /CP0DSL /MPAB0 /MPSIZE0
/MPSIZE1
SETF  /CPSMBGL /CLDESPL /CP0RWL /CP0ASL /CP0DSL /MPAB0 /MPSIZE0
/MPSIZE1
CHECK        SMWOL SMWEL

SETF  /CPSMBGL CLDESPL CP0RWL /CP0ASL CP0DSL MPAB0 MPSIZE0
MPSIZE1
SETF  /CPSMBGL CLDESPL CP0RWL /CP0ASL CP0DSL MPAB0 MPSIZE0
MPSIZE1
CHECK        SMWOL SMWEL

SETF CPSMBGL
CHECK ^SMROL ^SMWOL ^SMREL ^SMWEL
SETF /CPSMBGL CPSM2DBGL
CHECK SMROL ^SMWOL SMREL ^SMWEL
SETF /CPSMBGL CPSM2DBGL
CHECK SMROL ^SMWOL SMREL ^SMWEL
SETF /CPSMBGL /CPSM2DBGL
CHECK SMROL SMWOL SMREL SMWEL

TRACE_OFF
```

PALASM Design Description - PAL40

```
;-------------------------------Declaration Segment--------

TITLE    HOST LAPD EEPROM Memory Write Strobes - PAL40
;        and Write Violation strobe
PATTERN  LAPD
REVISION 1.08
AUTHOR   HOPETON WALKER
COMPANY
DATE     08/08/94

CHIP     HOST_PAL      PAL16V8

;-------------------------------PIN Declarations-----------
;        PINS

PIN   1      OEL
PIN   2      CPSMBGL
PIN   3      CLDESPL
PIN   4      LDEWEH
PIN   5      CP0RWL
PIN   6      CP0ASL
PIN   7      CP0DSL
PIN   8      MPAB0
PIN   9      MPSIZE0
PIN   10     GND
PIN   11     MPSIZE1
PIN   12     MLCEERL
PIN   13     CPSM2DBGL
PIN   14     SMROL
PIN   15     SMREL
PIN   16     NC
PIN   17     LDEEWVL
PIN   18     LDEWEL
PIN   19     LDEWOL
PIN   20     VCC

STRING LDW  '/CPSMBGL * /CPSM2DBGL * /CLDESPL * /CP0DSL *
             /CP0RWL * /CP0ASL * LDEWEH'

STRING LDWV  '/CPSMBGL * /CPSM2DBGL * /CLDESPL * /CP0DSL *
             /CP0RWL * /CP0ASL * /LDEWEH'

EQUATIONS
```

```
         LDEWEL.TRST = /OEL
         LDEWOL.TRST = /OEL
      LDEEWVL.TRST = /OEL

         /LDEWOL      = LDW * MPAB0 +
               LDW * /MPSIZE0 +
                  LDW * MPSIZE1

      /LDEWEL = LDW * /MPAB0

      /LDEEWVL = LDWV * MPAB0  +
               LDWV * /MPSIZE0 +
                  LDWV * MPSIZE1 +
               LDWV * /MPAB0

         /MLCEERL = /SMROL + /SMREL
       MLCEERL.TRST = /OEL
SIMULATION

TRACE_ON   OEL CPSMBGL CPSM2DBGL CLDESPL LDEWEH CP0RWL CP0ASL
CP0DSL
                    MPAB0 MPSIZE0 MPSIZE1
              LDEWOL LDEWEL LDEEWVL
SETF /OEL CPSMBGL /CPSM2DBGL CLDESPL LDEWEH CP0RWL CP0ASL
CP0DSL
          MPAB0 MPSIZE0 MPSIZE1  SMROL SMREL
CHECK LDEWOL LDEWEL LDEEWVL MLCEERL
SETF CPSMBGL /CLDESPL LDEWEH /CP0RWL /CP0ASL /CP0DSL
          MPAB0 MPSIZE0 MPSIZE1
CHECK LDEWOL LDEWEL LDEEWVL
SETF  /CPSMBGL /CLDESPL LDEWEH /CP0RWL /CP0ASL /CP0DSL
          /MPAB0 MPSIZE0 MPSIZE1
CHECK /LDEWOL /LDEWEL LDEEWVL
SETF  /CPSMBGL /CLDESPL /LDEWEH /CP0RWL /CP0ASL /CP0DSL
          /MPAB0 MPSIZE0 MPSIZE1
CHECK LDEWOL LDEWEL /LDEEWVL
SETF LDEWEH MPAB0 MPSIZE0 /MPSIZE1
CHECK /LDEWOL LDEWEL LDEEWVL
SETF  /MPAB0 MPSIZE0 /MPSIZE1
CHECK LDEWOL /LDEWEL LDEEWVL
SETF  /MPAB0 /MPSIZE0 /MPSIZE1
CHECK /LDEWOL /LDEWEL LDEEWVL
SETF  /MPAB0 /MPSIZE0 /MPSIZE1
CHECK /LDEWOL /LDEWEL LDEEWVL
```

```
SETF  /MPAB0 MPSIZE0 MPSIZE1
CHECK /LDEWOL /LDEWEL LDEEWVL
SETF  MPAB0 MPSIZE0 /MPSIZE1
CHECK /LDEWOL LDEWEL LDEEWVL
SETF CPSMBGL
CHECK LDEWOL LDEWEL LDEEWVL
SETF CPSMBGL CP0ASL CP0DSL
CHECK LDEWOL LDEWEL LDEEWVL
SETF CPSMBGL CP0ASL CP0DSL
CHECK LDEWOL LDEWEL LDEEWVL
SETF /CPSMBGL /CP0ASL /CP0DSL CP0RWL
CHECK LDEWOL LDEWEL LDEEWVL
SETF CPSM2DBGL
CHECK LDEWOL LDEWEL LDEEWVL
SETF CPSM2DBGL
CHECK LDEWOL LDEWEL LDEEWVL
SETF /CPSMBGL /CPSM2DBGL /CP0ASL /CP0DSL CP0RWL
CHECK LDEWOL LDEWEL LDEEWVL

SETF SMROL /SMREL
CHECK /MLCEERL
SETF SMREL
CHECK MLCEERL
SETF /SMROL
CHECK /MLCEERL
SETF SMROL
CHECK MLCEERL


TRACE_OFF
```

PALASM Design Description - PAL44

```
;---------------------------------- Declaration Segment ------------
TITLE   PERIPHERAL IACK AND HIFI R/W - PAL44
PATTERN LAPD
REVISION 1.06
AUTHOR  HOPETON WALKER
COMPANY
DATE    08/08/94

CHIP PAL44 PAL16V8

;---------------------------------- PIN Declarations ---------------
PIN 1     CPIACKL                         ;
PIN 2     MFPIE0L                         ;
PIN 3     BIMRSTL                         ;
PIN 4     INTAEL                          ;
PIN 5     INTAL0                          ;
PIN 6     INTAL1                          ;
PIN 7     HHIFICSL                        ;
PIN 8     CPDSL                           ;
PIN 9     CPRWL                           ;
PIN 10     GND                            ;
PIN 11     OENL                           ;
PIN 12     HIFICSL                        ;
PIN 13     CLRSPIAL                       ;
PIN 14     HIFRL                          ;
PIN 15     HIFWL                          ;
PIN 16     CLRTM1L                        ;
PIN 17     MFPIEL                         ;
PIN 18     BIACKL                         ;
PIN 19     UIACKL                         ;
PIN 20     VCC                            ;

;---------------------------------- Boolean Equation Segment ------
EQUATIONS



/BIACKL = /CPIACKL + /BIMRSTL
BIACKL.TRST = /OENL

/MFPIEL = /INTAEL * /INTAL1 * /INTAL0 ;(00) CASE
MFPIEL.TRST = /OENL
```

```
/CLRTM1L = /INTAEL * /INTAL1 * INTAL0 ;(01) CASE
CLRTM1L.TRST = /OENL

/CLRSPIAL = /INTAEL * INTAL1 * /INTAL0 ;(10) CASE
CLRSPIAL.TRST = /OENL

/HIFRL = /HHIFICSL * CPRWL * /CPDSL
HIFRL.TRST = /OENL

/HIFWL = /HHIFICSL * /CPRWL * /CPDSL
HIFWL.TRST = /OENL

/HIFICSL = /HHIFICSL

;----------------------------------- Simulation Segment ------------
SIMULATION

;INITIALIZE
SETF CPIACKL MFPIE0L /BIMRSTL INTAEL /INTAL0 /INTAL1 HHIFICSL CPDSL
CPRWL /OENL

;TEST TRISTATE
SETF OENL
SETF /OENL



;TEST BIACKL
SETF BIMRSTL
SETF /CPIACKL
SETF CPIACKL

;TEST MFPIEL, CLRTM1L, AND CLRSPIAL
SETF /INTAEL
SETF INTAL0
SETF INTAL1 /INTAL0
SETF INTAL0
SETF INTAEL /INTAL1 /INTAL0

;TEST HIFI WRITE
SETF /HHIFICSL /CPDSL /CPRWL
CHECK /HIFICSL /HIFWL HIFRL
SETF CPDSL
CHECK /HIFICSL HIFWL HIFRL
```

```
SETF /CPDSL CPRWL
CHECK /HIFICSL HIFWL /HIFRL
SETF HHIFICSL CPRWL CPDSL
CHECK HIFICSL HIFRL HIFWL

;TEST HIFI READ
SETF /HHIFICSL CPRWL /CPDSL
CHECK /HIFICSL HIFWL /HIFRL
SETF CPDSL
CHECK /HIFICSL HIFWL HIFRL
SETF /CPDSL HHIFICSL
CHECK HIFICSL HIFWL HIFRL

;-----------------------------------------------------------------
```

PALASM Design Description - PAL48

```
;--------------------------------Declaration segment--------
TITLE   HOST INTERUPT REQUEST PAL - PAL48
PATTERN  LAPD
REVISION 1.13
AUTHOR   HOPETON WALKER
COMPANY
DATE     08/26/94

;This PAL is the host interrupt request
;LAPD Interface

CHIP    HOSTPAL     PAL16V8
;------------------------------PIN Declarations------------
;                    PINS

CP0RWL                      ;I0
CP0DSL                      ;I1
CPUIRSL                     ;I2
CPDB30                      ;I3
SPYINT0L                    ;I4
CPDB31                      ;I5
SIRQL                       ;I6
CP0ASL                      ;I7
NC                          ;I8
GND                         ;I9
OEL                         ;O0
MLCIRQL                     ;O7
CPUIRQL                     ;I/O6
CP2DB31                     ;I/O5
MLCIRQL2                    ;I/O4
CPIACKL                     ;I/O3
CPDBSEL                     ;I/O2
CPDBENL                     ;I/O1
NC                          ;I/O0
VCC                         ;VCC


EQUATIONS

        CPUIRQL.TRST     =      /CP0RWL * /CP0DSL * /CPUIRSL

        CP2DB31.TRST     =      CP0RWL * /CP0DSL * /CPUIRSL
        MLCIRQL2.TRST    =      /OEL

        MLCIRQL.TRST     =      /MLCIRQL2

        /CPUIRQL    =     /CPDB31

        /CP2DB31    =     /SIRQL

        /MLCIRQL2   =     /SPYINT0L +
                          /CP0RWL * /CP0DSL * /CPUIRSL

        /MLCIRQL    =     /CPDB30 + /SPYINT0L

      /CPDBENL  = /CPIACKL * /CP0ASL  + /CPDBSEL * /CP0ASL
       CPDBENL.TRST = /OEL

SIMULATION

TRACE_ON  OEL CP0RWL CP0DSL CPUIRSL CPDB31 SIRQL CPDB30 SPYINT0L
CPUIRQL CP2DB31 MLCIRQL MLCIRQL2 CPIACKL CPDBENL CPDBSEL

SETF  /OEL /CP0ASL CP0RWL CP0DSL CPUIRSL /CPDB31 SPYINT0L /SIRQL
/CPDB30
SETF  CP0RWL CP0DSL CPUIRSL /CPDB31 CPIACKL CPDBSEL
CHECK        ^CPUIRQL

SETF  /CP0RWL /CP0DSL /CPUIRSL CPDB31
SETF  /CP0RWL /CP0DSL /CPUIRSL
SETF  /CP0RWL /CP0DSL /CPUIRSL
CHECK        CPUIRQL

SETF  /CP0RWL /CP0DSL /CPUIRSL /CPDB31 CPDB30
SETF  /CP0RWL /CP0DSL /CPUIRSL
SETF  /CP0RWL /CP0DSL /CPUIRSL
CHECK        /CPUIRQL MLCIRQL

SETF  /CP0RWL /CP0DSL /CPUIRSL /CPDB31 /CPDB30
SETF  CP0RWL CP0DSL CPUIRSL
SETF  CP0RWL CP0DSL CPUIRSL
SETF  CP0RWL CP0DSL CPUIRSL
CHECK        ^CPUIRQL

SETF  /CP0RWL /CP0DSL /CPUIRSL /CPDB31
SETF  /CP0RWL /CP0DSL /CPUIRSL
SETF  /CP0RWL /CP0DSL /CPUIRSL
```

```
CHECK        /CPUIRQL

SETF  /CP0RWL /CP0DSL /CPUIRSL CPDB31
SETF  /CP0RWL /CP0DSL /CPUIRSL
SETF  /CP0RWL /CP0DSL /CPUIRSL
CHECK        CPUIRQL

SETF  /CP0RWL /CP0DSL /CPUIRSL /CPDB31
SETF  /CP0RWL /CP0DSL /CPUIRSL
SETF  /CP0RWL /CP0DSL /CPUIRSL
CHECK        /CPUIRQL

SETF  CP0RWL /CP0DSL /CPUIRSL SIRQL
SETF  CP0RWL /CP0DSL /CPUIRSL
SETF  CP0RWL /CP0DSL /CPUIRSL
CHECK        CP2DB31

SETF  CP0RWL /CP0DSL /CPUIRSL /SIRQL
SETF  CP0RWL /CP0DSL /CPUIRSL
SETF  CP0RWL /CP0DSL /CPUIRSL
CHECK        /CP2DB31

SETF  CP0RWL /CP0DSL /CPUIRSL SIRQL
SETF  CP0RWL /CP0DSL /CPUIRSL
SETF  CP0RWL /CP0DSL /CPUIRSL
CHECK        CP2DB31

SETF  /CP0RWL /CP0DSL /CPUIRSL /CPDB31
SETF  /CP0RWL /CP0DSL /CPUIRSL
SETF  /CP0RWL /CP0DSL /CPUIRSL
CHECK        /CPUIRQL

SETF  /CP0RWL /CP0DSL /CPUIRSL CPDB31
SETF  /CP0RWL /CP0DSL /CPUIRSL
SETF  /CP0RWL /CP0DSL /CPUIRSL
CHECK        CPUIRQL

SETF  /CP0RWL /CP0DSL /CPUIRSL /CPDB31
SETF  /CP0RWL /CP0DSL /CPUIRSL
SETF  /CP0RWL /CP0DSL /CPUIRSL
CHECK        /CPUIRQL

SETF  CP0RWL /CP0DSL /CPUIRSL SIRQL
SETF  CP0RWL /CP0DSL /CPUIRSL
SETF  CP0RWL /CP0DSL /CPUIRSL
```

```
CHECK       CP2DB31

SETF  CP0RWL /CP0DSL /CPUIRSL /SIRQL
SETF  CP0RWL /CP0DSL /CPUIRSL
SETF  CP0RWL /CP0DSL /CPUIRSL
CHECK       /CP2DB31

SETF  CP0RWL /CP0DSL /CPUIRSL SIRQL
SETF  CP0RWL /CP0DSL /CPUIRSL
SETF  CP0RWL /CP0DSL /CPUIRSL
CHECK       CP2DB31

SETF  /CP0RWL /CP0DSL /CPUIRSL /CPDB31
SETF  /CP0RWL /CP0DSL /CPUIRSL
SETF  /CP0RWL /CP0DSL /CPUIRSL
CHECK       /CPUIRQL

SETF  /CP0RWL /CP0DSL /CPUIRSL SPYINT0L /CPDB30
SETF  /CP0RWL /CP0DSL /CPUIRSL SPYINT0L /CPDB30
CHECK       /MLCIRQL2 /MLCIRQL

SETF  /CP0RWL /CP0DSL /CPUIRSL SPYINT0L /CPDB30
SETF  /CP0RWL /CP0DSL /CPUIRSL SPYINT0L /CPDB30
CHECK       /MLCIRQL2 /MLCIRQL

SETF  CP0RWL CP0DSL CPUIRSL SPYINT0L /CPDB30
SETF  CP0RWL CP0DSL CPUIRSL SPYINT0L /CPDB30
SETF  CP0RWL CP0DSL CPUIRSL SPYINT0L /CPDB30
SETF  CP0RWL CP0DSL CPUIRSL SPYINT0L /CPDB30
CHECK       MLCIRQL2 ^MLCIRQL

SETF  CP0RWL CP0DSL CPUIRSL SPYINT0L CPDB30
SETF  CP0RWL CP0DSL CPUIRSL SPYINT0L CPDB30
CHECK       MLCIRQL2 ^MLCIRQL

SETF  CP0RWL CP0DSL CPUIRSL SPYINT0L CPDB30
SETF  CP0RWL CP0DSL CPUIRSL SPYINT0L CPDB30
CHECK       MLCIRQL2 ^MLCIRQL

SETF  CP0RWL CP0DSL CPUIRSL /SPYINT0L CPDB30
SETF  CP0RWL CP0DSL CPUIRSL /SPYINT0L CPDB30
SETF  CP0RWL CP0DSL CPUIRSL /SPYINT0L CPDB30
SETF  CP0RWL CP0DSL CPUIRSL /SPYINT0L CPDB30
CHECK       /MLCIRQL2 /MLCIRQL
```

```
SETF   CP0RWL CP0DSL CPUIRSL SPYINT0L CPDB30
SETF   CP0RWL CP0DSL CPUIRSL SPYINT0L CPDB30
CHECK         MLCIRQL2 ^MLCIRQL

SETF   CP0RWL CP0DSL CPUIRSL SPYINT0L CPDB30
SETF   CP0RWL CP0DSL CPUIRSL SPYINT0L CPDB30
CHECK         MLCIRQL2 ^MLCIRQL

SETF   CP0RWL CP0DSL CPUIRSL SPYINT0L /CPDB30
SETF   CP0RWL CP0DSL CPUIRSL SPYINT0L /CPDB30
SETF   CP0RWL CP0DSL CPUIRSL SPYINT0L /CPDB30
SETF   CP0RWL CP0DSL CPUIRSL SPYINT0L /CPDB30
CHECK         MLCIRQL2 ^MLCIRQL

SETF   CP0RWL CP0DSL CPUIRSL SPYINT0L CPDB30
SETF   CP0RWL CP0DSL CPUIRSL SPYINT0L CPDB30
CHECK         MLCIRQL2 ^MLCIRQL

SETF /CPIACKL
CHECK /CPDBENL
SETF CP0ASL
CHECK CPDBENL
SETF CP0ASL
CHECK CPDBENL
SETF /CP0ASL CPIACKL
SETF /CP0ASL CPIACKL
CHECK CPDBENL
SETF /CPDBSEL
CHECK /CPDBENL
SETF CP0ASL
CHECK CPDBENL
SETF /CP0ASL CPDBSEL
CHECK CPDBENL
TRACE_OFF
```

PALASM Design Description - PAL50

```
;--------------------------------Declaration Segment-------
TITLE   SPYDER-T T7115A SHARED MEMORY R/W - PAL50
;       STROBE AND ILLEGAL ADDRESS INT.
PATTERN  LAPD
REVISION 1.05
AUTHOR   HOPETON WALKER
COMPANY
DATE     08/28/94

CHIP   HOSTPAL     PAL16V8
;-------------------------------PIN Declarations-----------
;                          PINS

PIN    1      SPYWEL              ;I0
PIN    2      SPYRDL              ;I1
PIN    3      SPYA23                       ;I2
PIN    4      SPYA22                       ;I3
PIN    5      SPYA21                       ;I4
PIN    6      SPYA20                       ;I5
PIN    7      SPYA19                       ;I6
PIN    8      SPYASL              ;I7
PIN    9      SPYBGL              ;I8
PIN    10     GND                 ;GND
PIN    11     OEL                 ;I9
PIN    12     SPYSMCSL            ;O7
PIN    13     SPYBRL              ;I/O6
PIN    14     NC                  I/O5
PIN    15     SPYSMSPL            ;I/O4
PIN    16     SMROL                        ;I/O3
PIN    17     SMREL                        ;I/O2
PIN    18     SMWOL               ;I/O1
PIN    19     SMWEL               ;O0
PIN    20     VCC                 ;VCC

STRING SPYSMSPH  'SPYA23 * SPYA22 * SPYA21 * SPYA20 * SPYA19 *
/SPYBGL '

EQUATIONS

        SMWEL.TRST      =      /OEL * /SPYBGL
        SMWOL.TRST      =      /OEL * /SPYBGL
        SMREL.TRST      =      /OEL * /SPYBGL
        SMROL.TRST      =      /OEL * /SPYBGL
        SPYSMSPL.TRST      =      /OEL
        SPYSMCSL.TRST      =      /OEL

        /SMWEL             =      /SPYBGL * /SPYWEL
        /SMWOL             =      /SPYBGL * /SPYWEL
        /SMREL             =      /SPYBGL * /SPYRDL
        /SMROL             =      /SPYBGL * /SPYRDL

        /SPYSMSPL  =      SPYSMSPH
        /SPYSMCSL  =      /SPYRDL + SPYRDL * /SPYASL

SIMULATION

TRACE_ON  SPYA23 SPYA22 SPYA21 SPYA20 SPYA19 SPYASL SPYBGL
          SPYWEL SPYRDL SPYBRL SMWEL SMWOL SMREL SMROL
SPYSMSPL

SETF  /OEL SPYA23 SPYA22 SPYA21 SPYA20 SPYA19 SPYASL SPYBGL
SPYRDL
SETF  SPYBRL SPYWEL
CHECK      ^SMWEL ^SMWOL ^SMREL ^SMROL SPYSMSPL SPYSMCSL

SETF  SPYA23 SPYA22 SPYA21 SPYA20 SPYA19 SPYASL SPYRDL
SETF  SPYA23 SPYA22 SPYA21 SPYA20 SPYA19 SPYASL SPYRDL

SETF  /SPYASL /SPYBGL /SPYWEL
CHECK      /SMWEL /SMWOL /SPYSMCSL

SETF  /SPYA23
CHECK      SPYSMSPL
SETF  SPYA23
CHECK      /SPYSMSPL

SETF  /SPYA22
CHECK      SPYSMSPL
SETF  SPYA22
CHECK      /SPYSMSPL

SETF  /SPYA21
CHECK      SPYSMSPL
SETF  SPYA21
CHECK      /SPYSMSPL

SETF  /SPYA20
CHECK      SPYSMSPL
```

```
SETF  SPYA20
CHECK        /SPYSMSPL

SETF  /SPYA19
CHECK         SPYSMSPL
SETF  SPYA19
CHECK        /SPYSMSPL

SETF  SPYASL
CHECK        /SPYSMSPL SPYSMCSL
SETF  /SPYASL
CHECK        /SPYSMSPL /SPYSMCSL

SETF  SPYWEL
CHECK         SMWEL SMWOL
SETF  /SPYWEL
CHECK        /SMWEL /SMWOL
SETF  SPYBGL
CHECK        ^SMWEL ^SMWOL
SETF  /SPYBGL

;READ
SETF  SPYA23 SPYA22 SPYA21 SPYA20 SPYA19 SPYASL SPYRDL SPYWEL
SETF  SPYA23 SPYA22 SPYA21 SPYA20 SPYA19 SPYASL SPYRDL SPYWEL

SETF  /SPYASL /SPYBGL /SPYRDL
CHECK        /SMREL /SMROL /SPYSMCSL

SETF  /SPYA23
CHECK         SPYSMSPL
SETF  SPYA23
CHECK        /SPYSMSPL

SETF  /SPYA22
CHECK          SPYSMSPL
SETF  SPYA22
CHECK        /SPYSMSPL

SETF  /SPYA21
CHECK         SPYSMSPL
SETF  SPYA21
CHECK        /SPYSMSPL

SETF  /SPYA20
CHECK         SPYSMSPL
```

```
SETF  SPYA20
CHECK        /SPYSMSPL

SETF  /SPYA19
CHECK        SPYSMSPL
SETF  SPYA19
CHECK        /SPYSMSPL

SETF  SPYASL
CHECK        /SPYSMSPL /SPYSMCSL
SETF  /SPYASL
CHECK        /SPYSMSPL /SPYSMCSL

SETF  SPYRDL
CHECK        SMREL SMROL /SPYSMCSL
SETF  SPYASL
CHECK        SMREL SMROL SPYSMCSL
SETF  /SPYRDL
CHECK        /SMREL /SMROL /SPYSMCSL
SETF  SPYBGL
CHECK        ^SMREL ^SMROL ^SMWEL ^SMWOL

TRACE_OFF
```

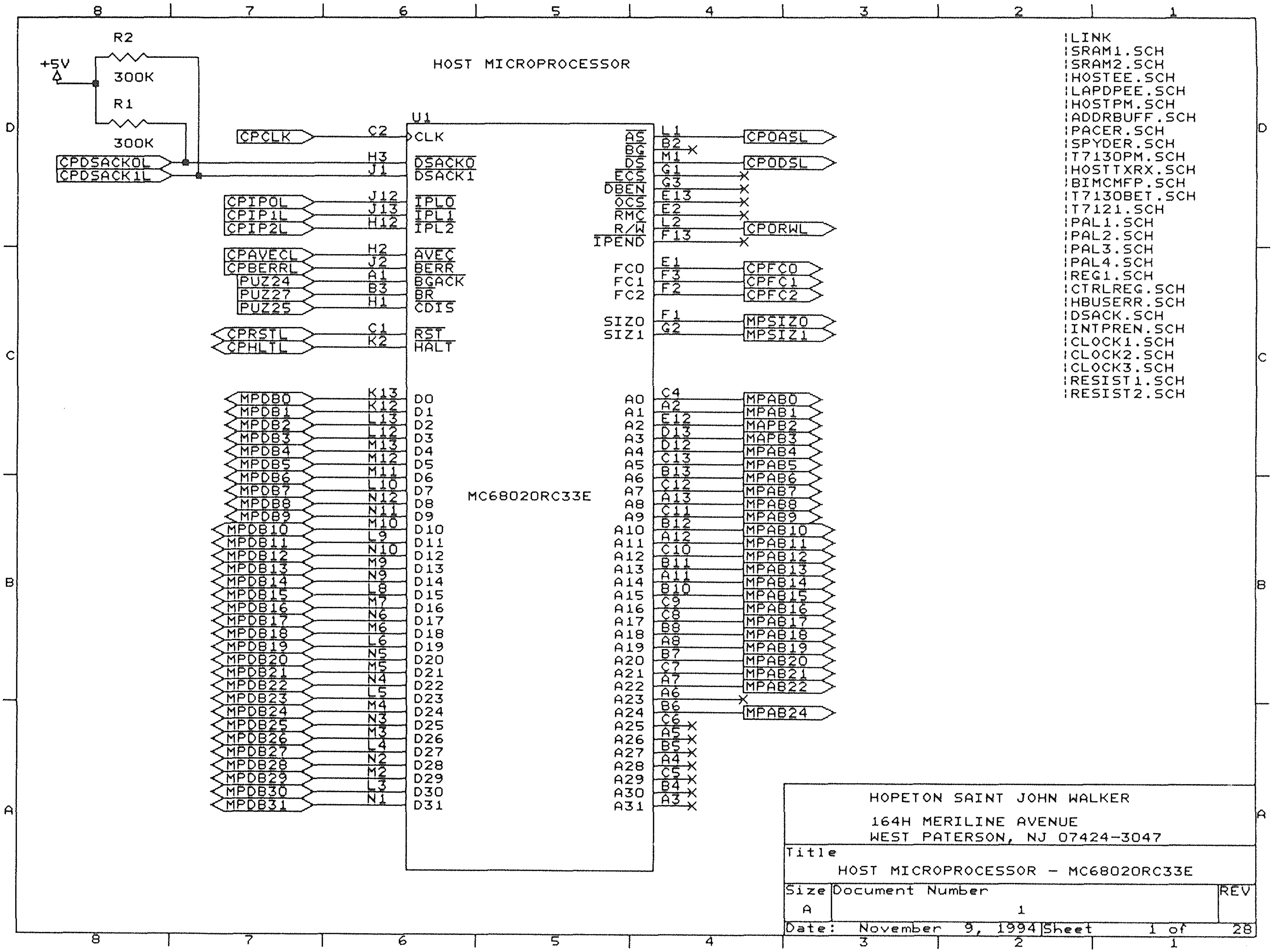
PALASM Design Description

```
;--------------------------------Declaration Segment-------
TITLE  MLC DTACK  - PAL54
PATTERN LAPD
REVISION 1.08
AUTHOR HOPETON WALKER
COMPANY
DATE 9/23/94

;This PAL generates wait states for the T7130 - PAL20RA10

CHIP   T7130WAITSTATE   PAL20RA10

;PINS                              DIP PINS      PLCC PINS

PIN 1   PL                 ;PL      1             2
PIN 2   MFPCLK             ;I0      2             3
PIN 3   MLCEERL            ;I1      3             4
PIN 4   GPITSEL            ;I2      4             5
PIN 5   CPDSL                       ;I3      5             6
PIN 12 GND                 ;GND    12            14
PIN 13 OEL                 ;OEL    13            16
PIN 14 MLCDTK0L            ;IO0    15            17
PIN 15 MLCDTK1L            ;IO1    14            18
PIN 17 GPDSL                        ;IO3    17            20
PIN 18 GPITSEL0L           ;IO4    18            21
PIN 19 GPITSEL1L           ;IO5    19            23
PIN 24 VCC                 ;VCC    24            28


EQUATIONS

          /MLCDTK0L :=       VCC
          MLCDTK0L.CLKF   =       MFPCLK
          MLCDTK0L.RSTF   =       MLCEERL
          MLCDTK0L.TRST   =       VCC

          /MLCDTK1L :=       /MLCDTK0L
          MLCDTK1L.CLKF   =       MFPCLK
          MLCDTK1L.RSTF   =       MLCEERL
          MLCDTK1L.TRST   =       VCC

 SIMULATION
```

```
TRACE_ON  OEL PL MFPCLK MLCDTK0L MLCDTK1L MLCEERL
               CPDSL    SETF /OEL PL /MFPCLK  MLCEERL CPDSL
SETF MFPCLK
CHECK MLCDTK0L MLCDTK1L GPDSL
SETF /MFPCLK
SETF MFPCLK
CHECK MLCDTK0L MLCDTK1L
SETF /MFPCLK
SETF /MFPCLK  /MLCEERL
SETF MFPCLK
CHECK /MLCDTK0L MLCDTK1L
SETF /MFPCLK
SETF MFPCLK
CHECK /MLCDTK0L /MLCDTK1L
SETF /MFPCLK
SETF MLCEERL
CHECK MLCDTK0L MLCDTK1L

TRACE_OFF
```

APPENDIX B
ORCAD SCHEMATICS

This section contains the schematics for the LAPD Interface. All the schematics were generated by ORCAD.

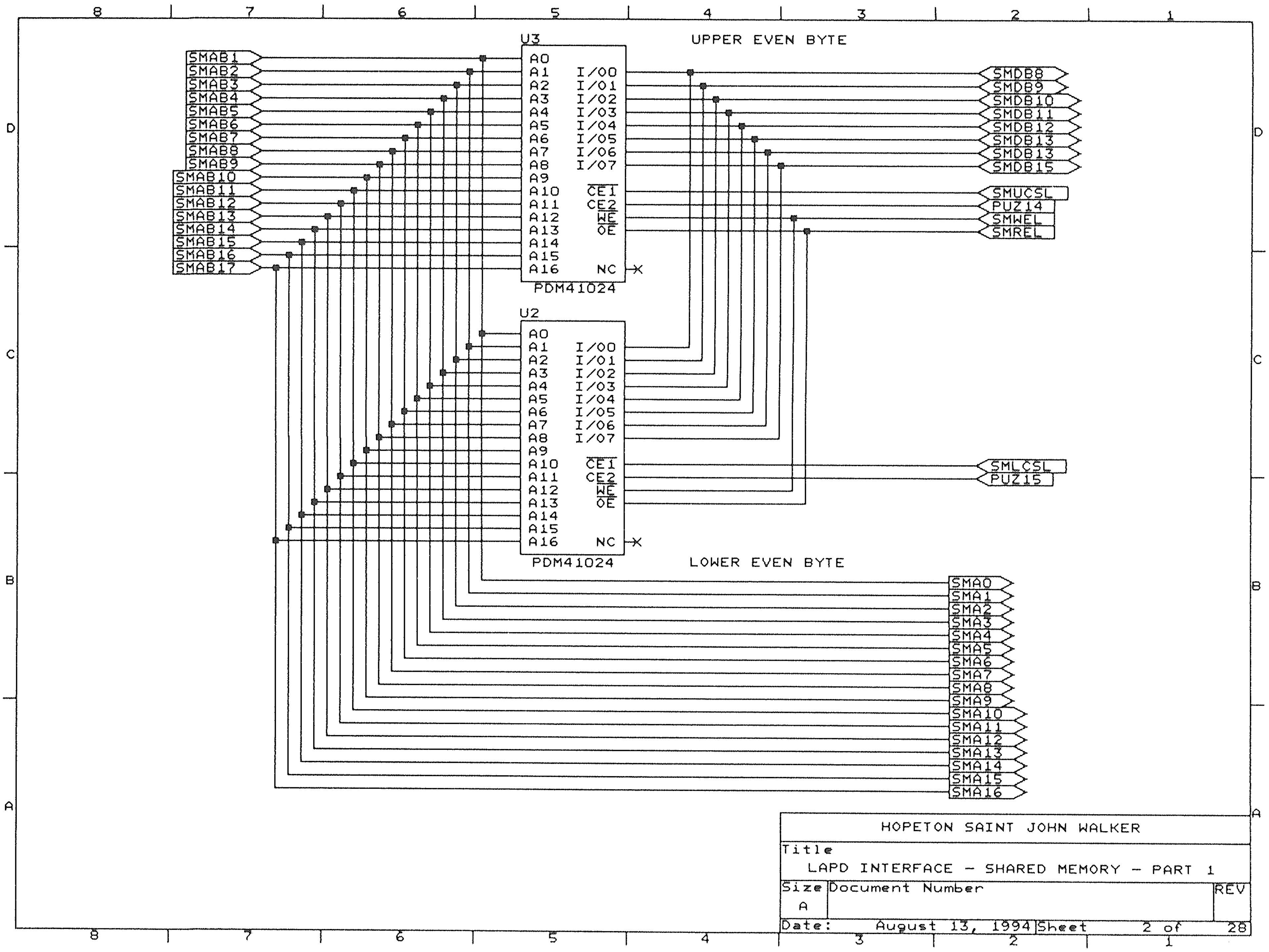


HOPETON SAINT JOHN WALKER

164H MERILINE AVENUE
WEST PATERSON, NJ 07424-3047

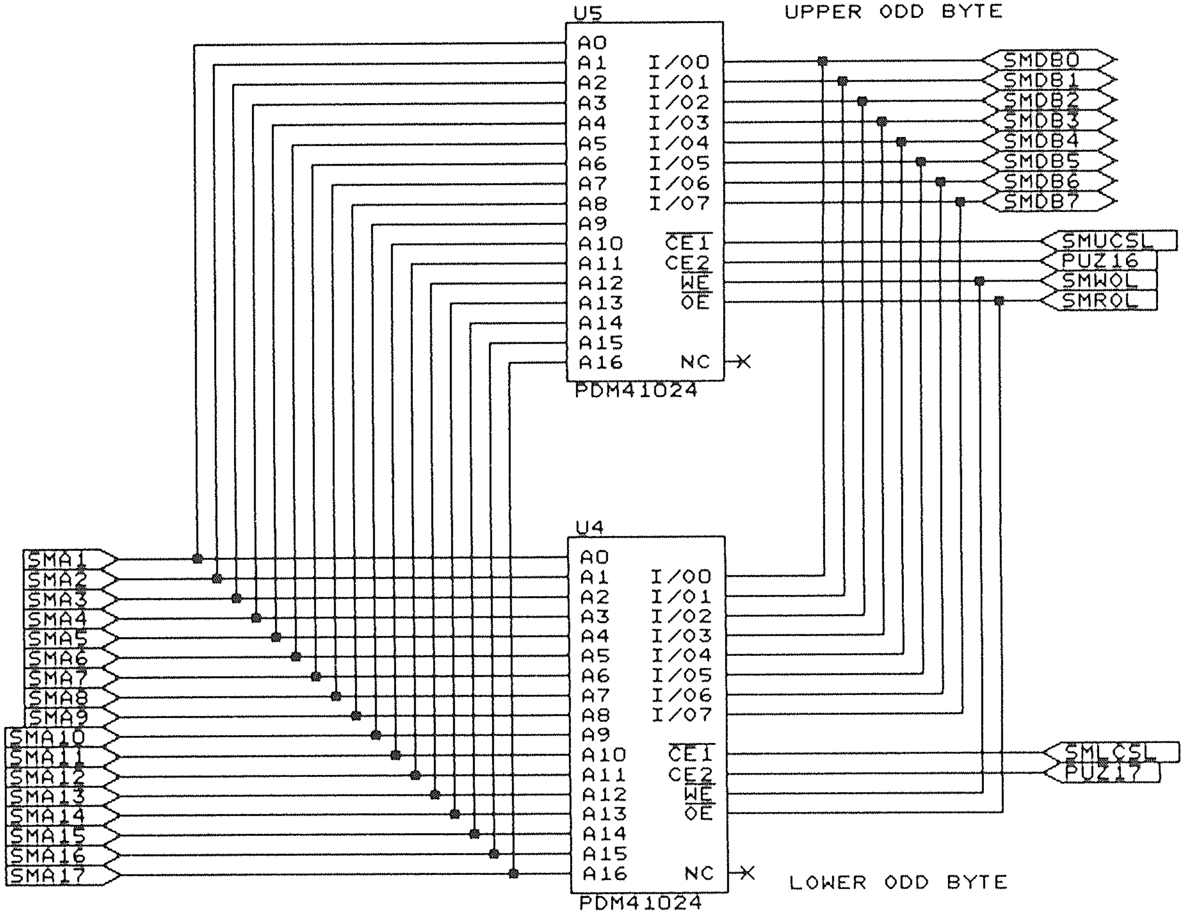
Title
HOST MICROPROCESSOR - MC68020RC33E

Size	Document Number	REV
A	1	
Date: November 9, 1994	Sheet 1 of 28	



HOPETON SAINT JOHN WALKER

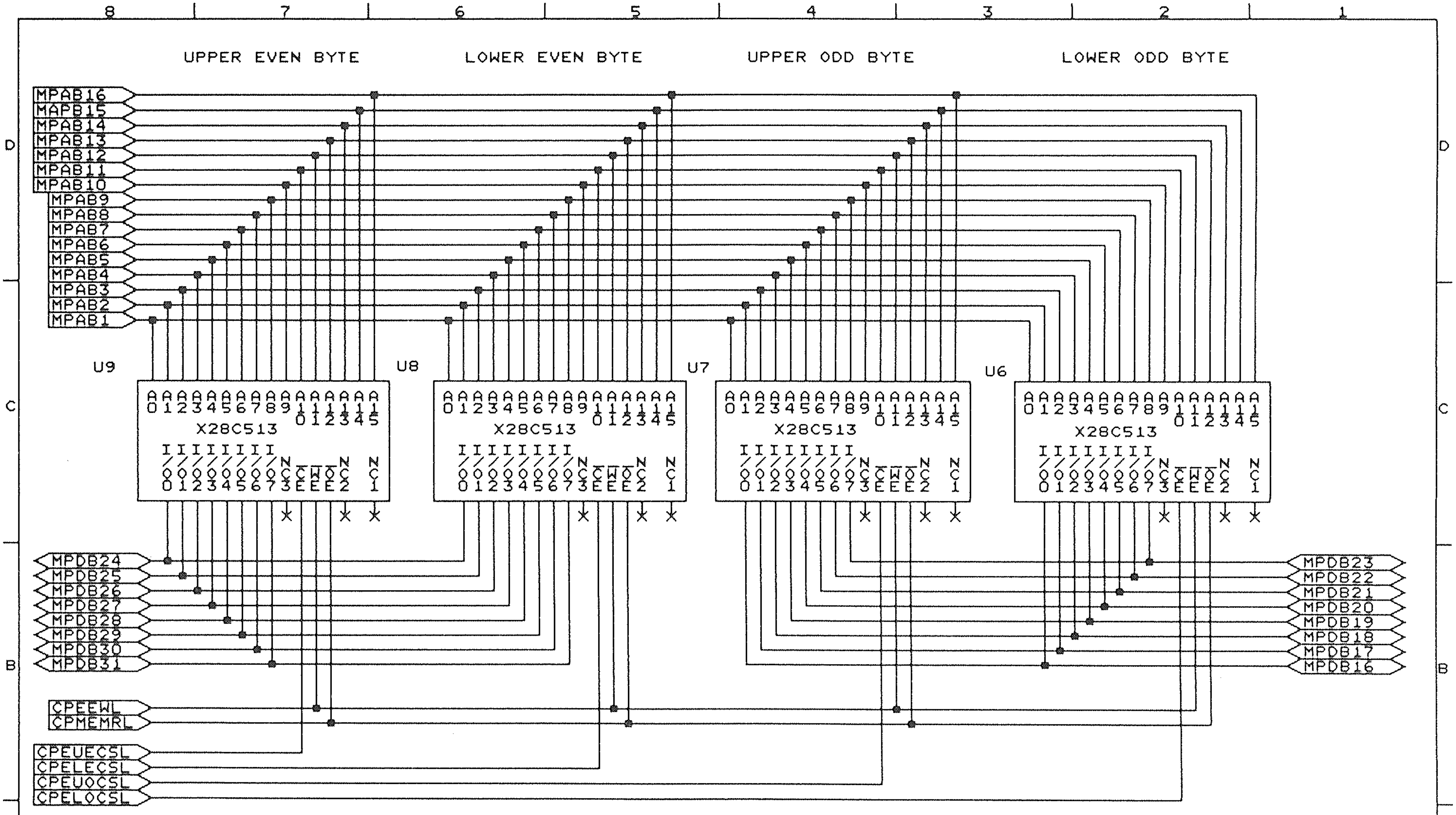
Title LAPD INTERFACE - SHARED MEMORY - PART 1		
Size A	Document Number	REV
Date: August 13, 1994	Sheet 2 of 28	



HOPETON SAINT JOHN WALKER

164H MERILINE AVENUE
WEST PATERSON, NJ 07424-3047

Title LAPD INTERFACE - SHARED MEMORY - PART 2		
Size A	Document Number	REV
Date: August 13, 1994	Sheet 3 of 28	



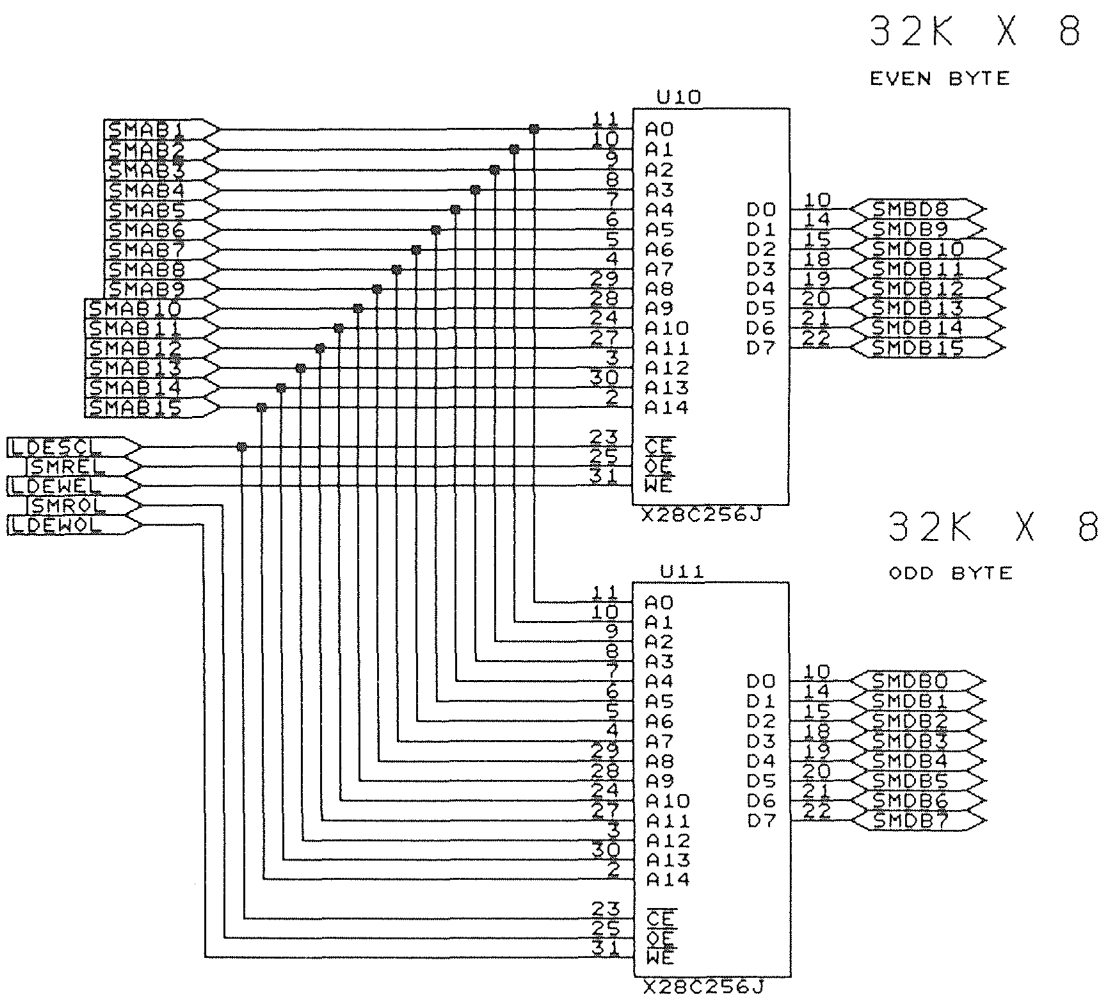
HOPETON SAINT JOHN WALKER

164H MERILINE AVENUE
WEST PATERSON, NJ 07424-3047

Title
HOST - PROGRAM MEMORY (TEXT SPACE)

Size	Document Number	REV
A		

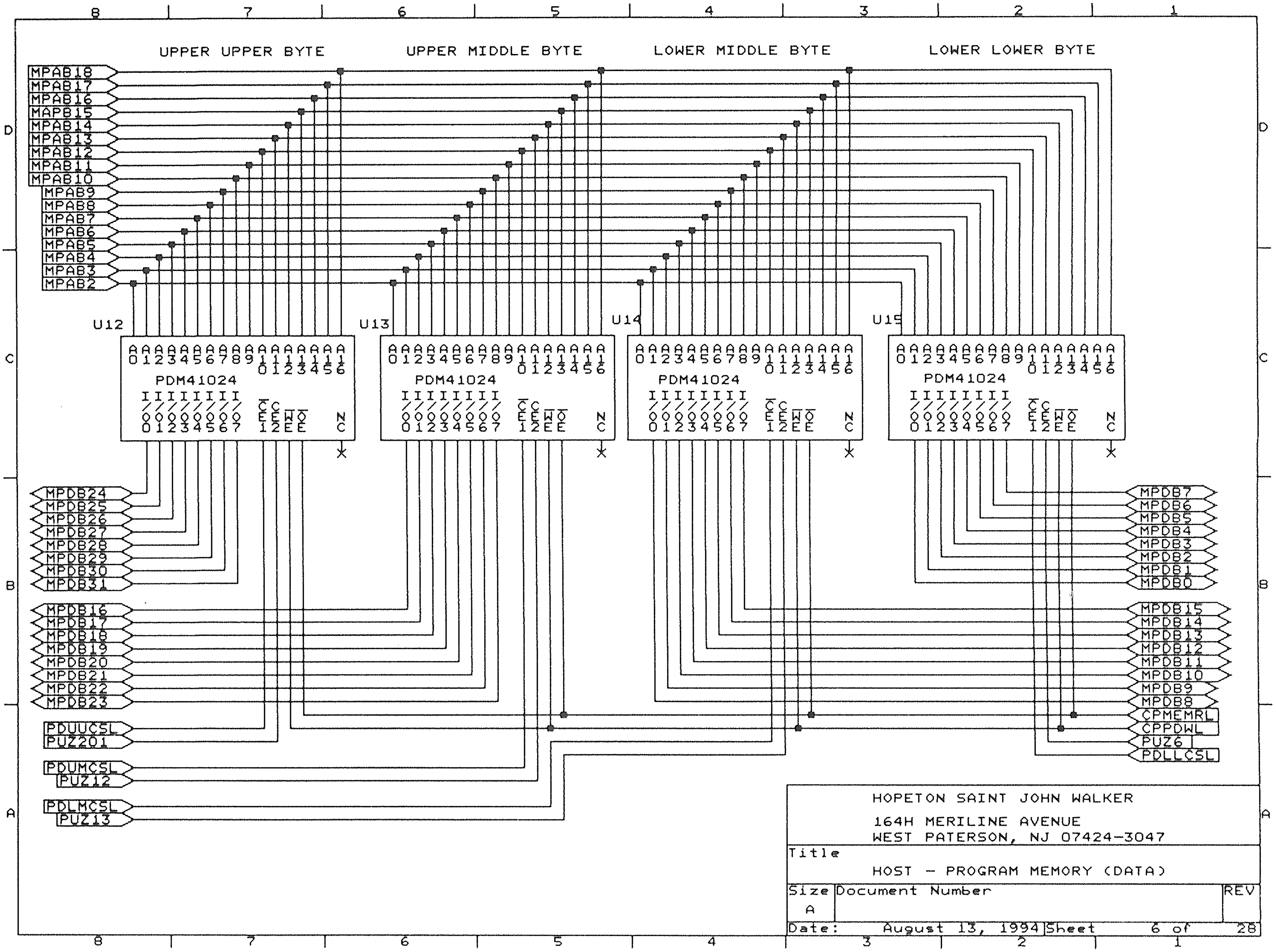
Date: August 13, 1994 Sheet 4 of 28



HOPETON SAINT JOHN WALKER

164H MERILINE AVENUE
WEST PATERSON, NJ 07424-3047

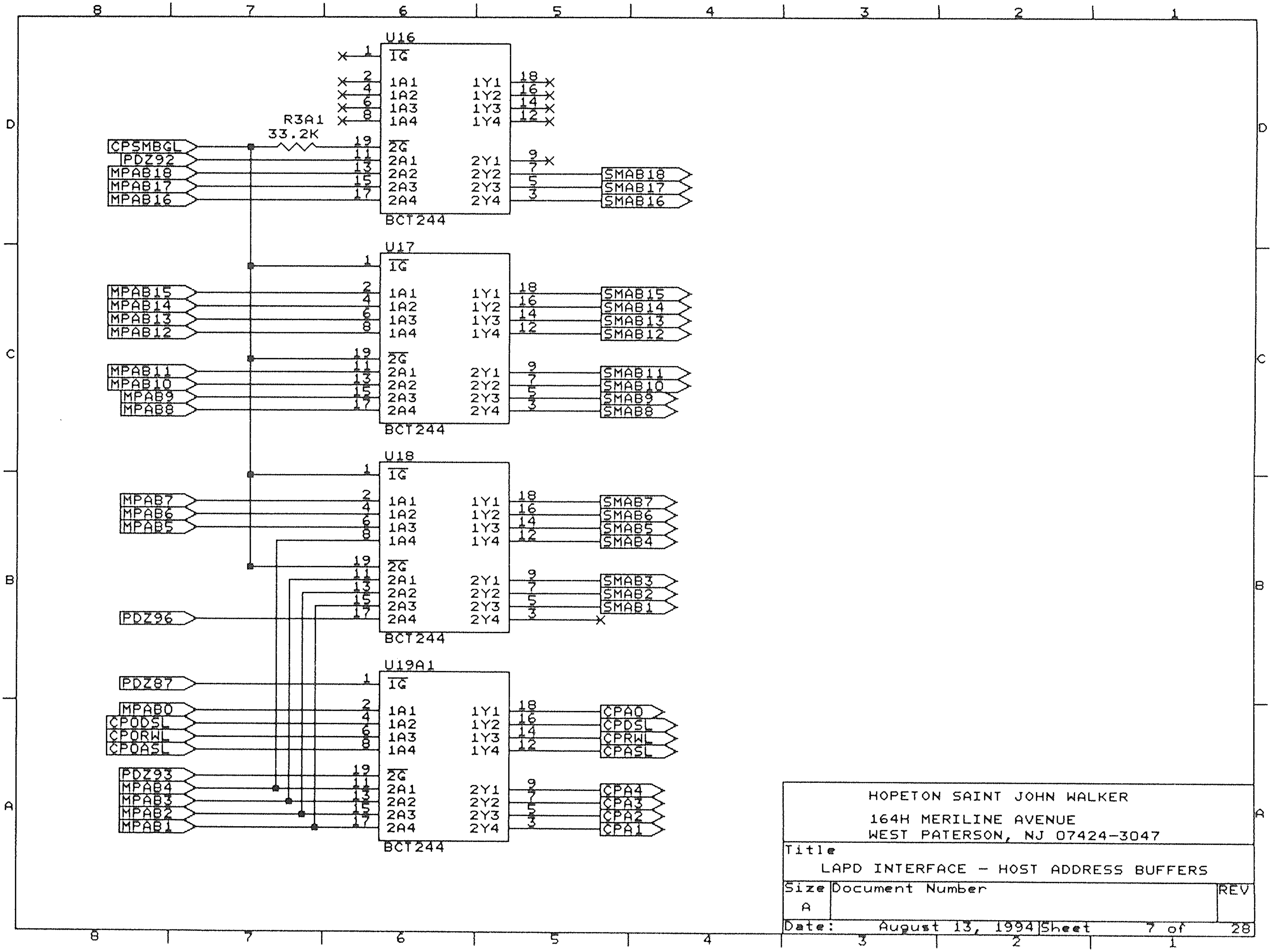
Title LAPD PROGRAM EEPROM - IN SHARED MEMORY		
Size A	Document Number	REV
Date: August 13, 1994	Sheet 5 of 28	



HOPETON SAINT JOHN WALKER
164H MERILINE AVENUE
WEST PATERSON, NJ 07424-3047

Title
HOST - PROGRAM MEMORY (DATA)

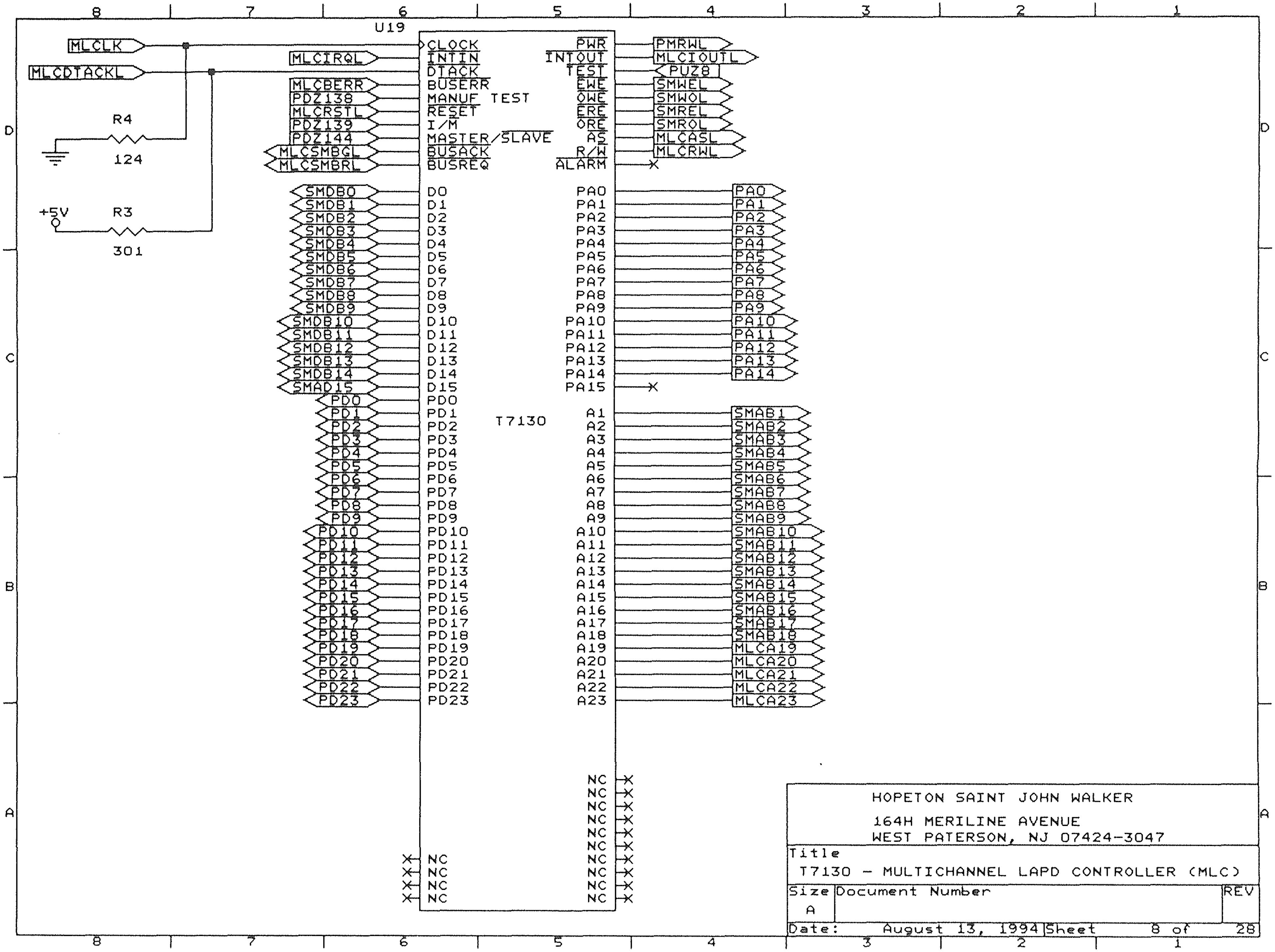
Size	Document Number	REV
A		
Date: August 13, 1994	Sheet 6 of 28	



HOPETON SAINT JOHN WALKER

164H MERILINE AVENUE
WEST PATERSON, NJ 07424-3047

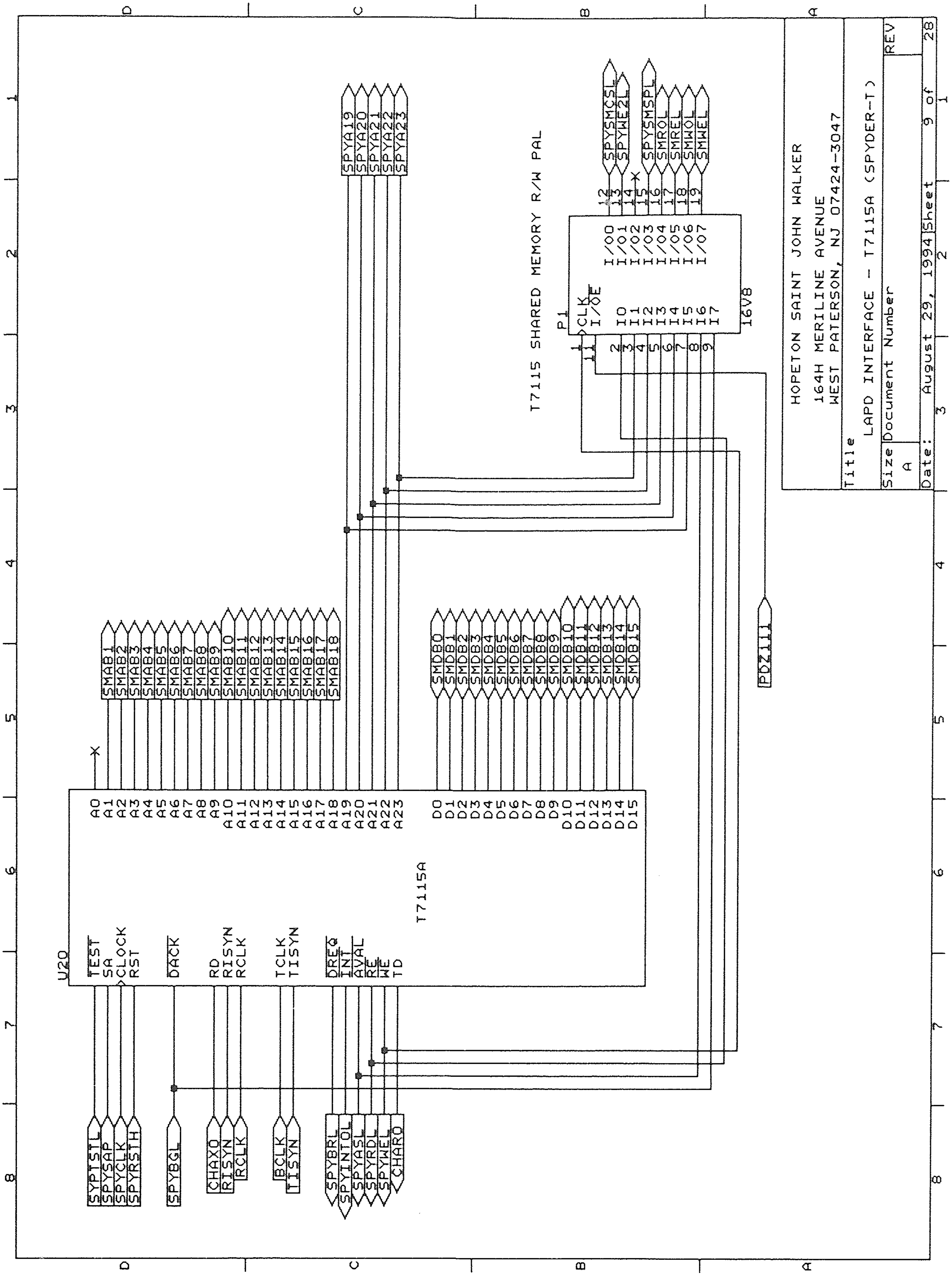
Title LAPD INTERFACE - HOST ADDRESS BUFFERS		
Size A	Document Number	REV
Date: August 13, 1994	Sheet 7 of 28	



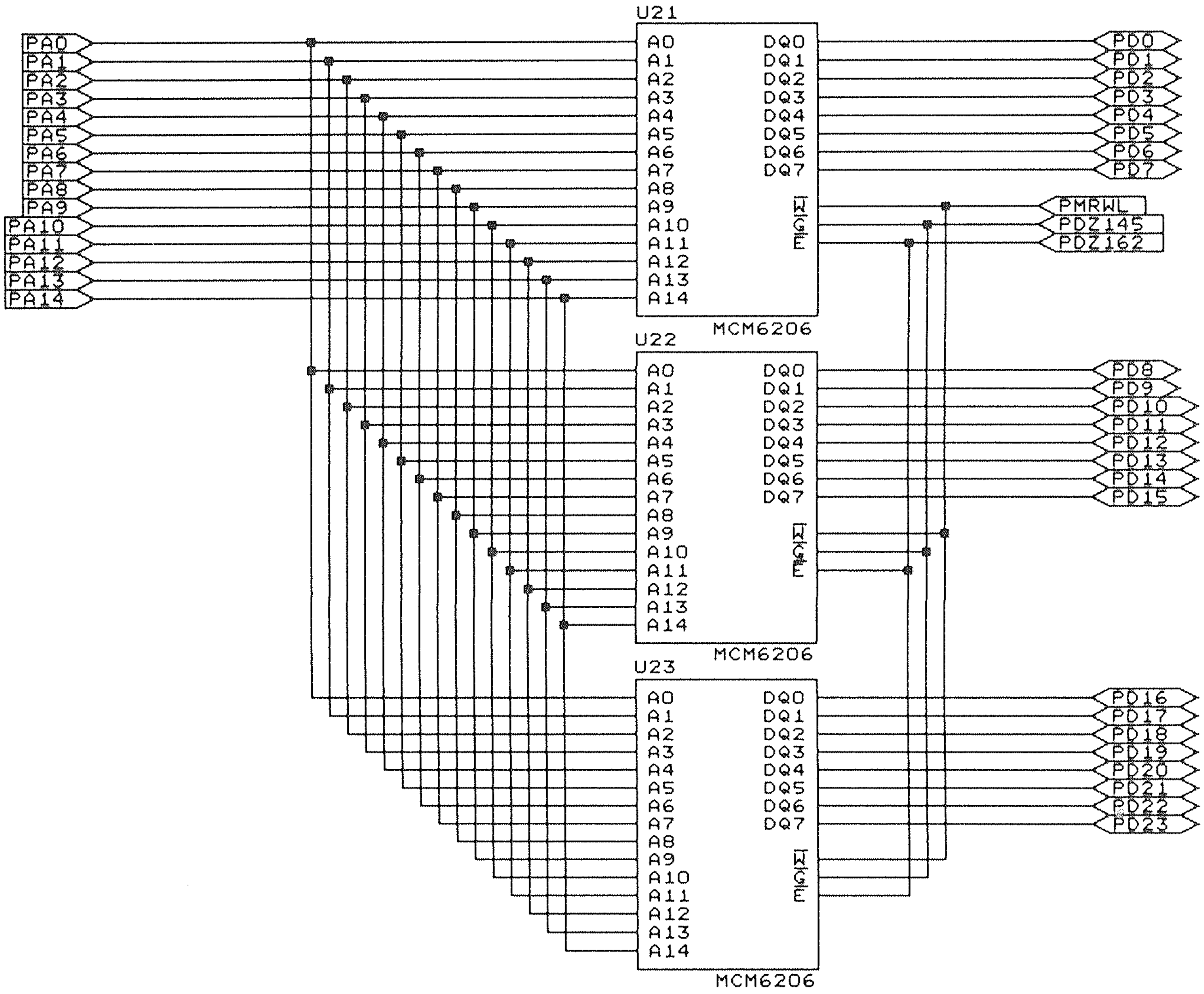
HOPETON SAINT JOHN WALKER

164H MERILINE AVENUE
WEST PATERSON, NJ 07424-3047

Title		
T7130 - MULTICHANNEL LAPD CONTROLLER (MLC)		
Size A	Document Number	REV
Date: August 13, 1994	Sheet 8 of 28	

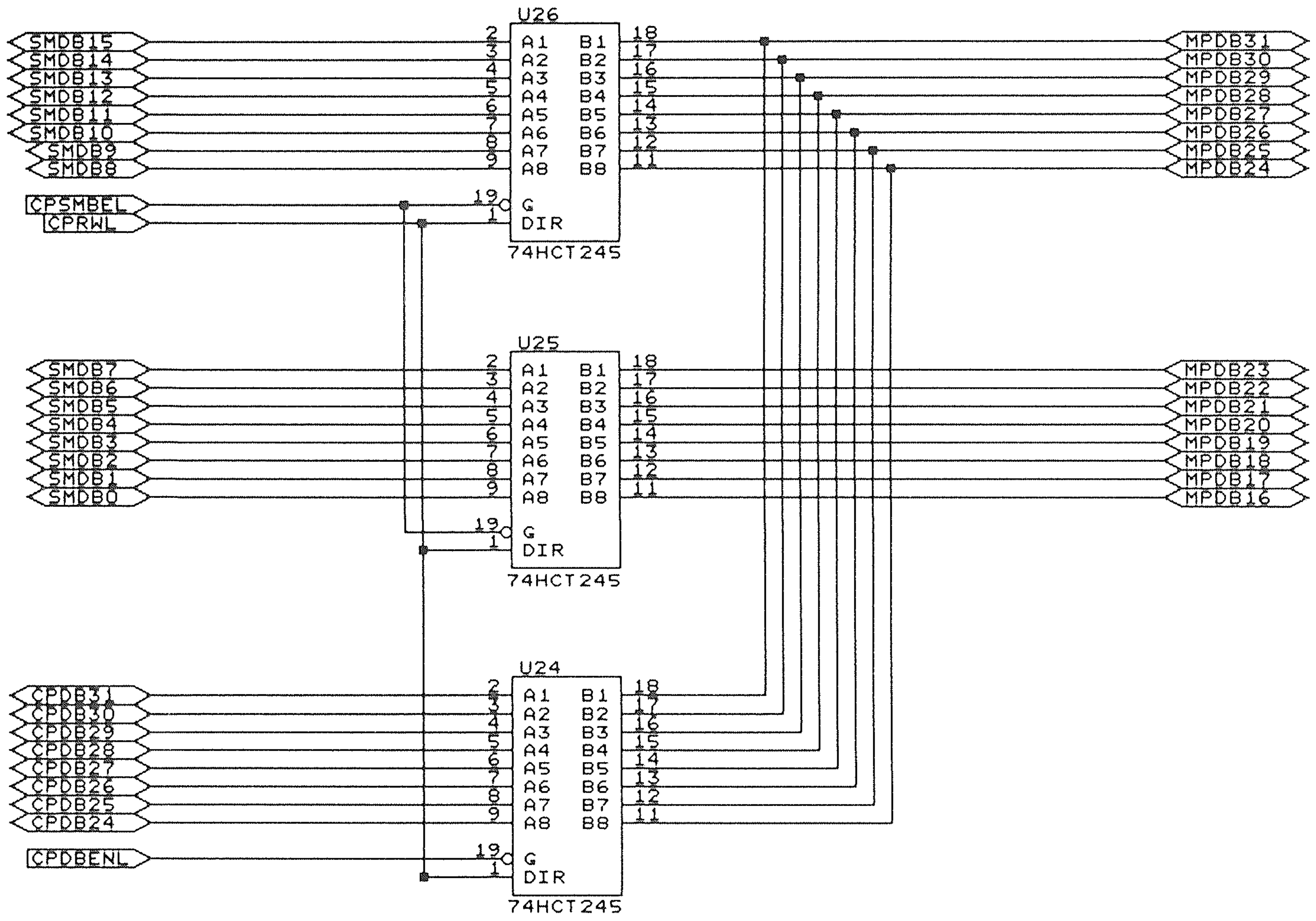


T7115 SHARED MEMORY R/W PAL
P1
CLK
I/OE
I0
I1
I2
I3
I4
I5
I6
I7
I/O0
I/O1
I/O2
I/O3
I/O4
I/O5
I/O6
I/O7
16V8
1 11 2 3 4 5 6 7 8 9
12 13 14 15 16 17 18 19
SPYSMCSL
SPYWE2L
SPYSMSPL
SMROL
SMREL
SMWOL
SMWEL
SPYA19
SPYA20
SPYA21
SPYA22
SPYA23
PDZ111
U20
T7115A
TEST
SA
CLOCK
RST
DACK
RD
RISYN
RCLK
TCLK
TISYN
DREQ
INT
AVAL
RE
WE
TD
A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 A10 A11 A12 A13 A14 A15 A16 A17 A18 A19 A20 A21 A22 A23
D0 D1 D2 D3 D4 D5 D6 D7 D8 D9 D10 D11 D12 D13 D14 D15
SMAB1 SMAB2 SMAB3 SMAB4 SMAB5 SMAB6 SMAB7 SMAB8 SMAB9 SMAB10 SMAB11 SMAB12 SMAB13 SMAB14 SMAB15 SMAB16 SMAB17 SMAB18
SMDB0 SMDB1 SMDB2 SMDB3 SMDB4 SMDB5 SMDB6 SMDB7 SMDB8 SMDB9 SMDB10 SMDB11 SMDB12 SMDB13 SMDB14 SMDB15
SYPTSTL
SPYSAP
SPYCLK
SPYRSTH
SPYBGL
CHAXD
RISYN
RCLK
BCLK
TISYN
SPYBRL
SPYINTOL
SPYASL
SPYRDL
SPYWEL
CHARD
HOPETON SAINT JOHN WALKER
164H MERILINE AVENUE
WEST PATERSON, NJ 07424-3047
Title
LAPD INTERFACE - T7115A (SPYDER-T)
Size
A
Document Number
REV
Date: August 29, 1994
Sheet 9 of 28
A B C D
1 2 3 4 5 6 7 8



HOPETON SAINT JOHN WALKER
164H MERILINE AVENUE
WEST PATERSON, NJ 07424-3047

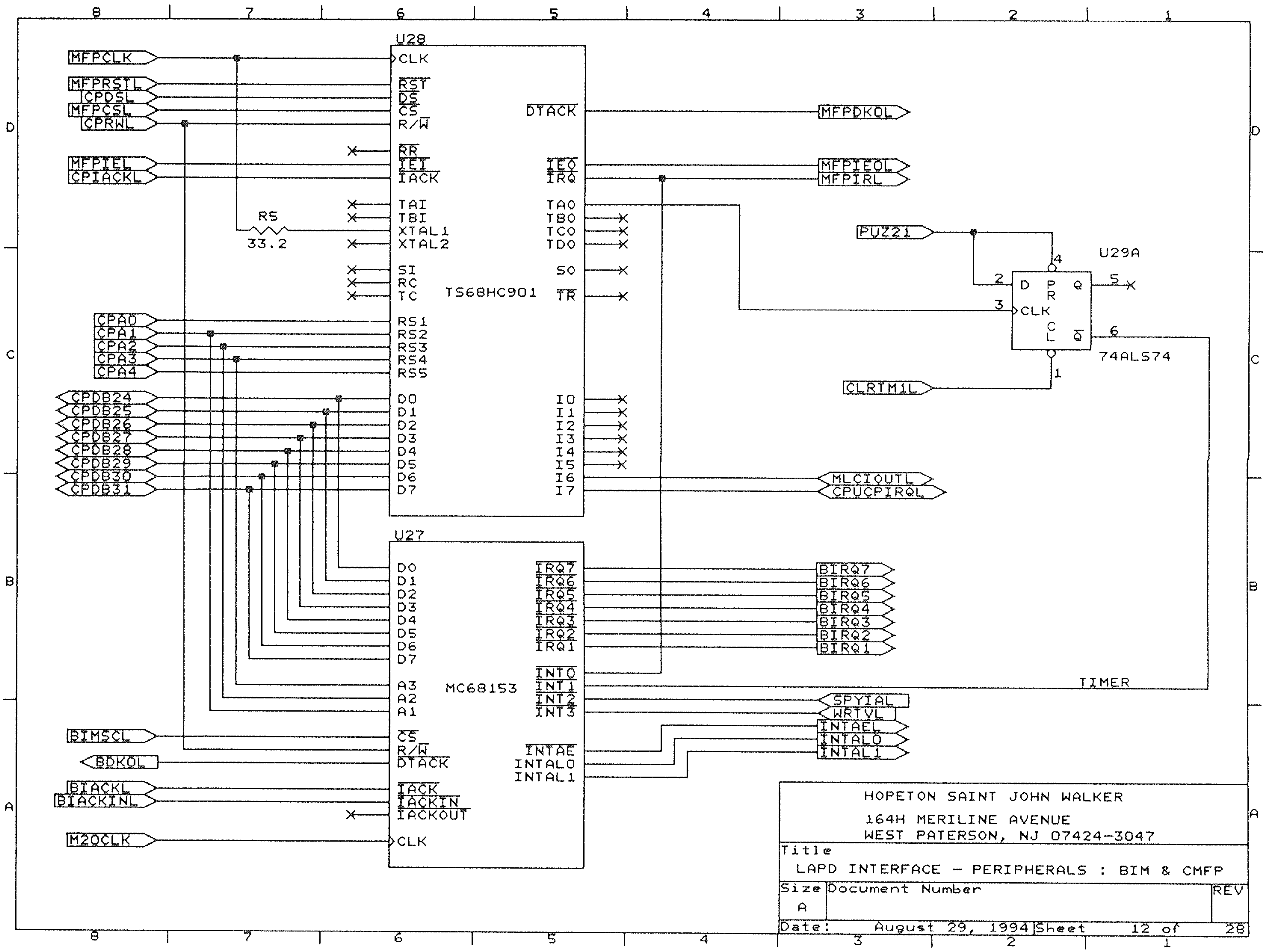
Title		
LAPD INTERFACE – T7130 PRIVATE MEMORY		
Size A	Document Number	REV
Date: August 29, 1994	Sheet 10 of 28	

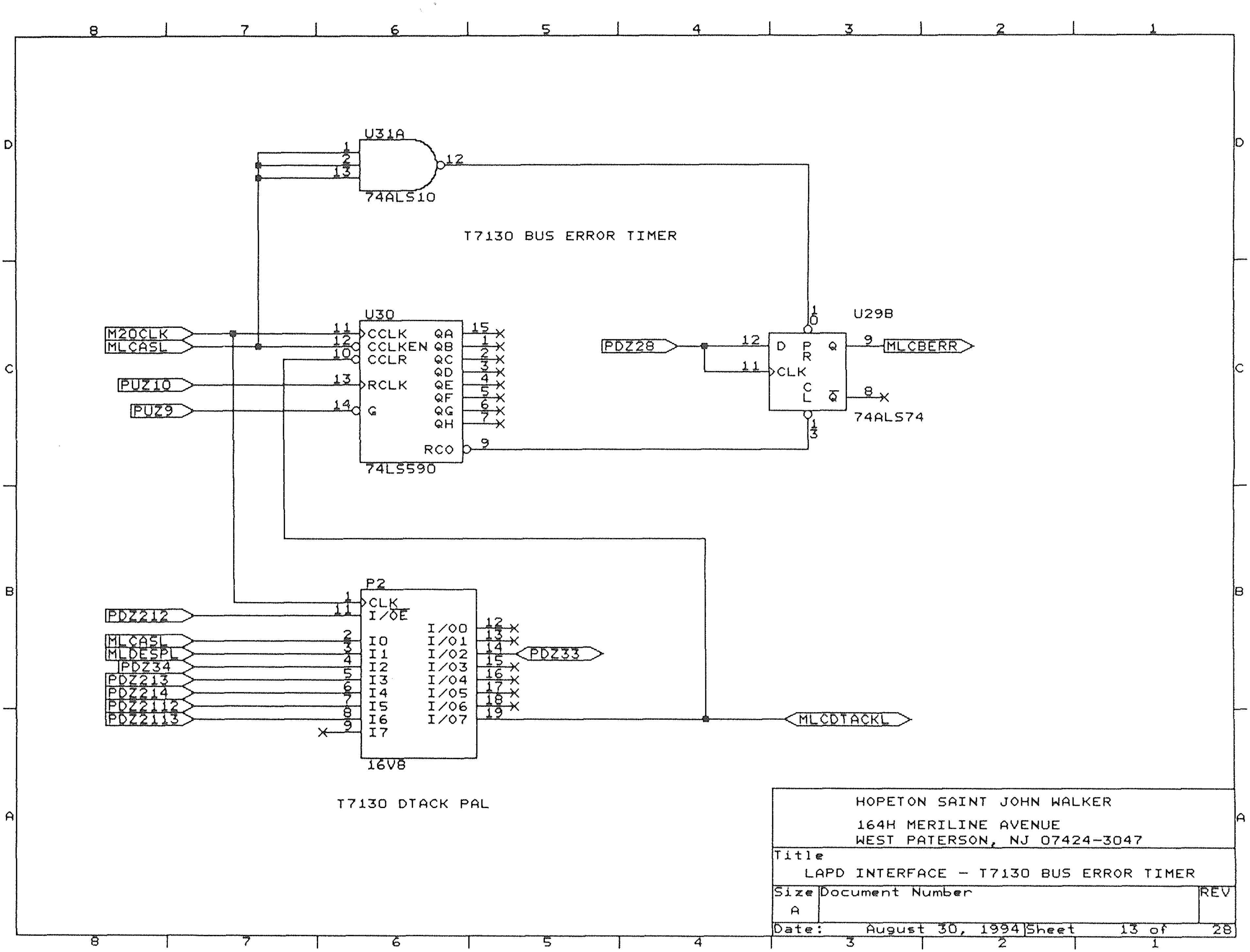


HOPETON SAINT JOHN WALKER

164H MERILINE AVENUE
WEST PATERSON, NJ 07424-3047

Title		
LAPD INTERFACE - HOST DATA TRANSCEIVERS		
Size A	Document Number	REV
Date: August 13, 1994	Sheet 11 of 28	


U28
TS68HC901
CLK
RST
DS
CS
R/W
RR
IEI
IACK
TAI
TBI
XTAL1
XTAL2
SI
RC
TC
RS1
RS2
RS3
RS4
RS5
D0
D1
D2
D3
D4
D5
D6
D7
DTACK
IEO
IRQ
TAO
TBO
TCO
TDO
SO
TR
I0
I1
I2
I3
I4
I5
I6
I7
MFPCLK
MFPRSTL
CPDSL
MFPCSL
CPRWL
MFPIEL
CPIACKL
R5
33.2
CPA0
CPA1
CPA2
CPA3
CPA4
CPDB24
CPDB25
CPDB26
CPDB27
CPDB28
CPDB29
CPDB30
CPDB31
MFPDKOL
MFPIEOL
MFPIRL
PUZ21
U29A
D
PR
CLK
CL
Q
Q
74ALS74
2
3
4
5
6
1
CLRTM1L
MLCIOUTL
CPUCPIRQL
U27
MC68153
D0
D1
D2
D3
D4
D5
D6
D7
A3
A2
A1
CS
R/W
DTACK
IACK
IACKIN
IACKOUT
CLK
IRQ7
IRQ6
IRQ5
IRQ4
IRQ3
IRQ2
IRQ1
INT0
INT1
INT2
INT3
INTAE
INTAL0
INTAL1
BIRQ7
BIRQ6
BIRQ5
BIRQ4
BIRQ3
BIRQ2
BIRQ1
TIMER
SPYIAL
WRTVL
INTAEL
INTAL0
INTAL1
BIMSCL
BDKOL
BIACKL
BIACKINL
M20CLK
8
7
6
5
4
3
2
1
D
C
B
A
HOPETON SAINT JOHN WALKER
164H MERILINE AVENUE
WEST PATERSON, NJ 07424-3047
Title
LAPD INTERFACE - PERIPHERALS : BIM & CMFP
Size
A
Document Number
REV
Date: August 29, 1994
Sheet 12 of 28



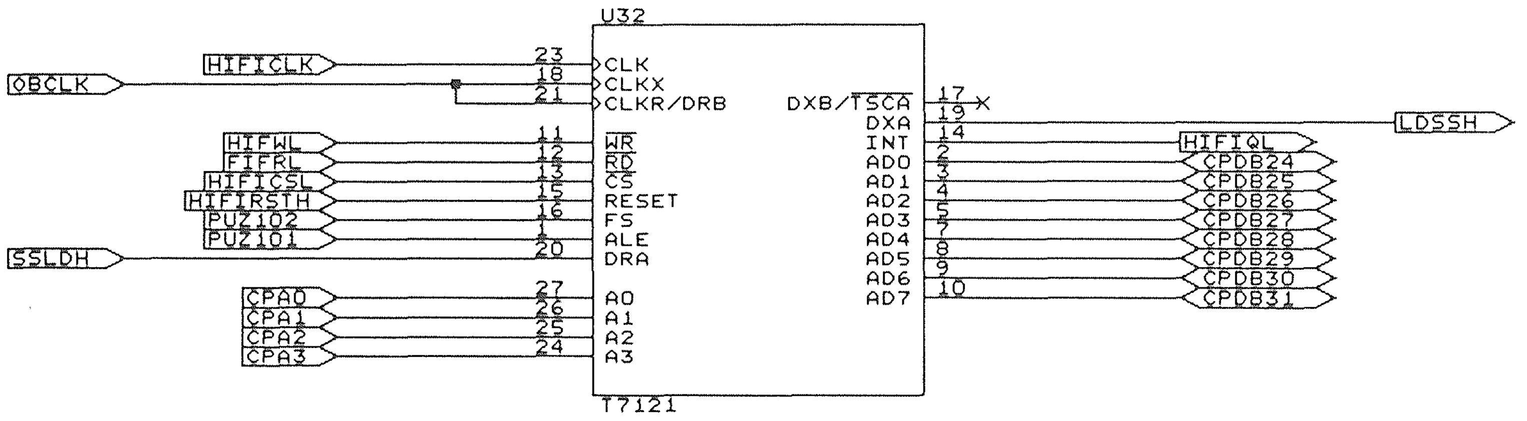
HOPETON SAINT JOHN WALKER

164H MERILINE AVENUE
WEST PATERSON, NJ 07424-3047

Title
LAPD INTERFACE - T7130 BUS ERROR TIMER

Size A | Document Number | REV

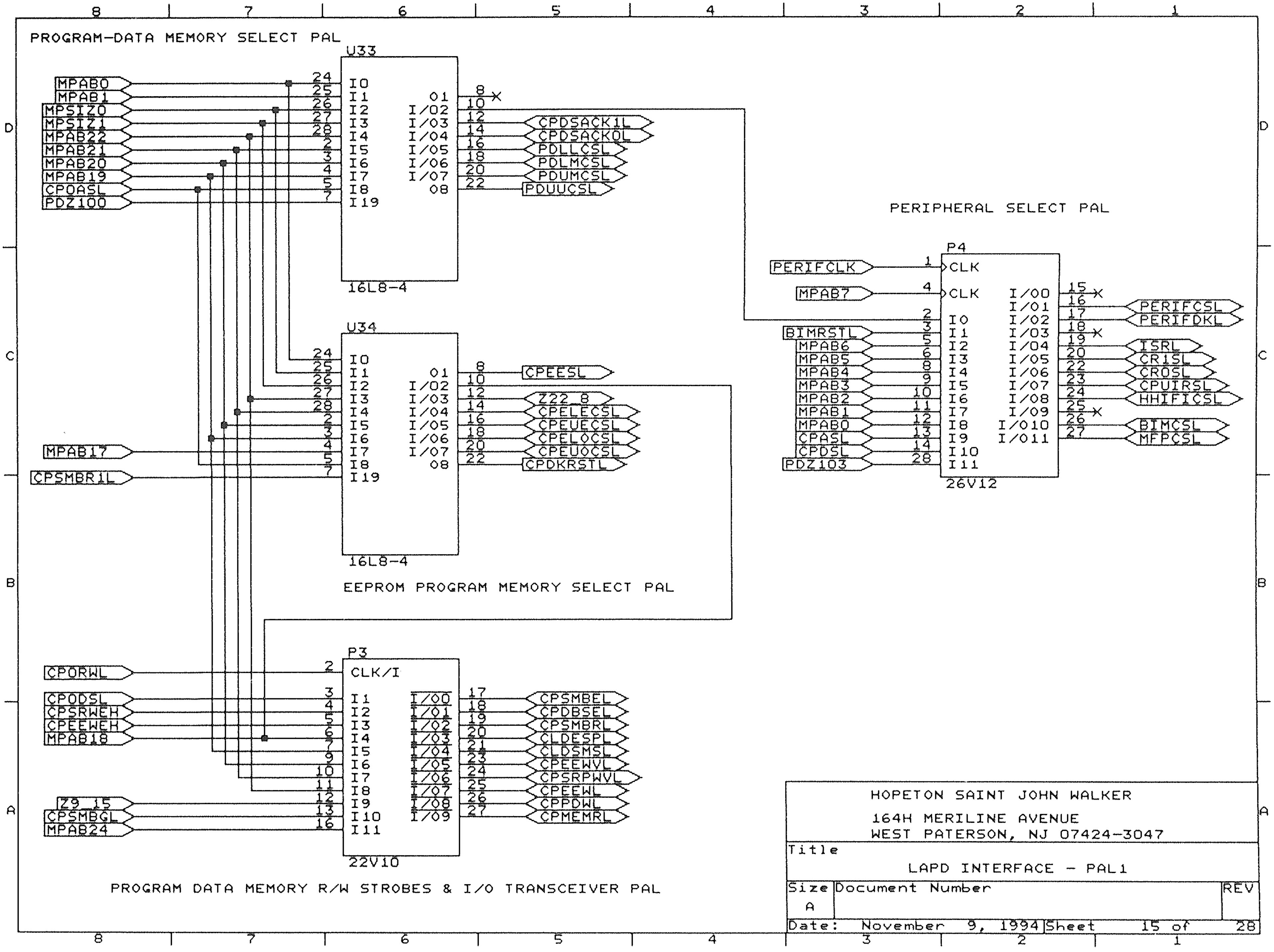
Date: August 30, 1994 | Sheet 13 of 28



HOPETON SAINT JOHN WALKER

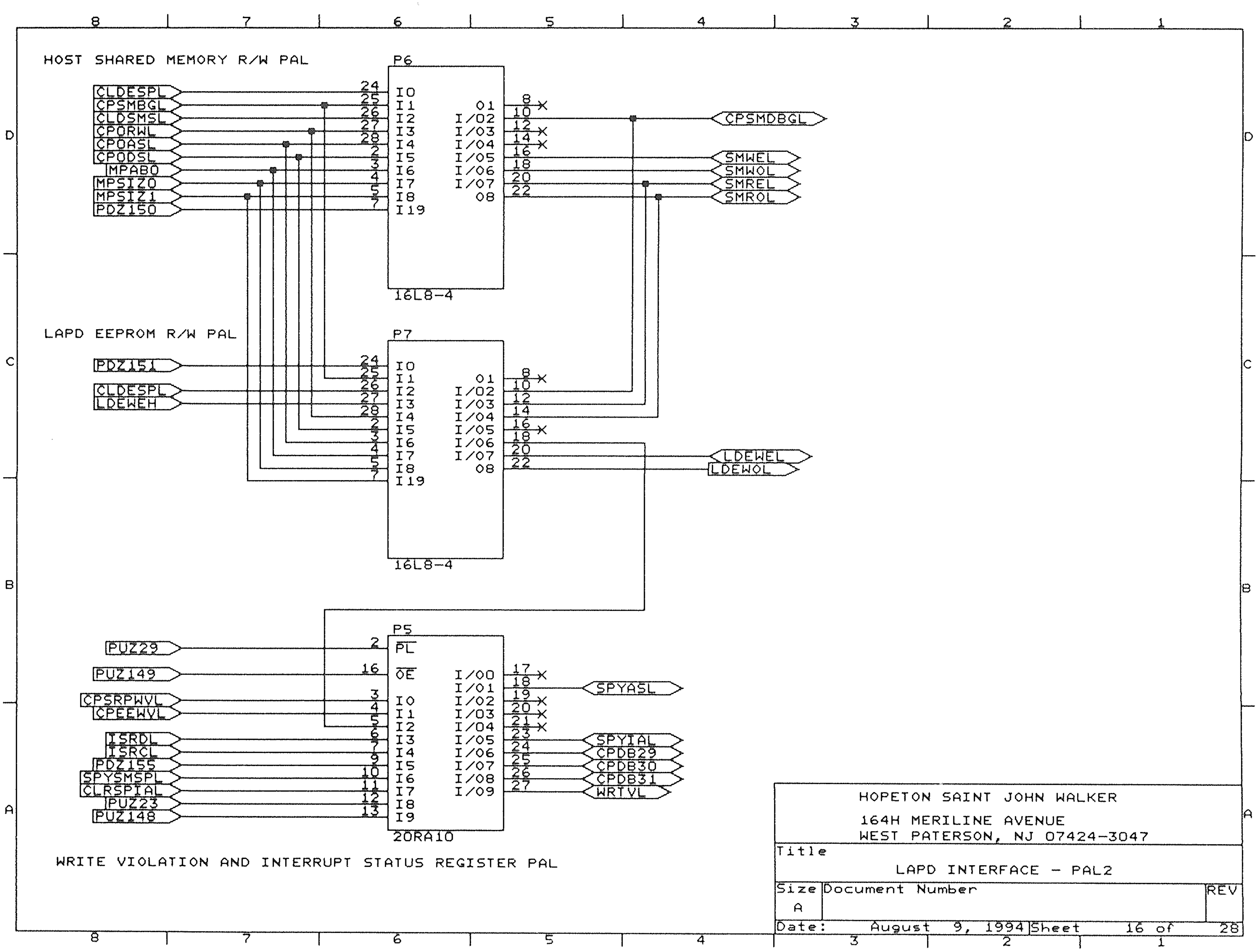
164H MERILINE AVENUE
WEST PATERSON, NJ 07424-3047

Title LAPD INTERFACE – CEPT LINE INTERFACE		
Size A	Document Number	REV
Date: August 13, 1994	Sheet 14 of 28	



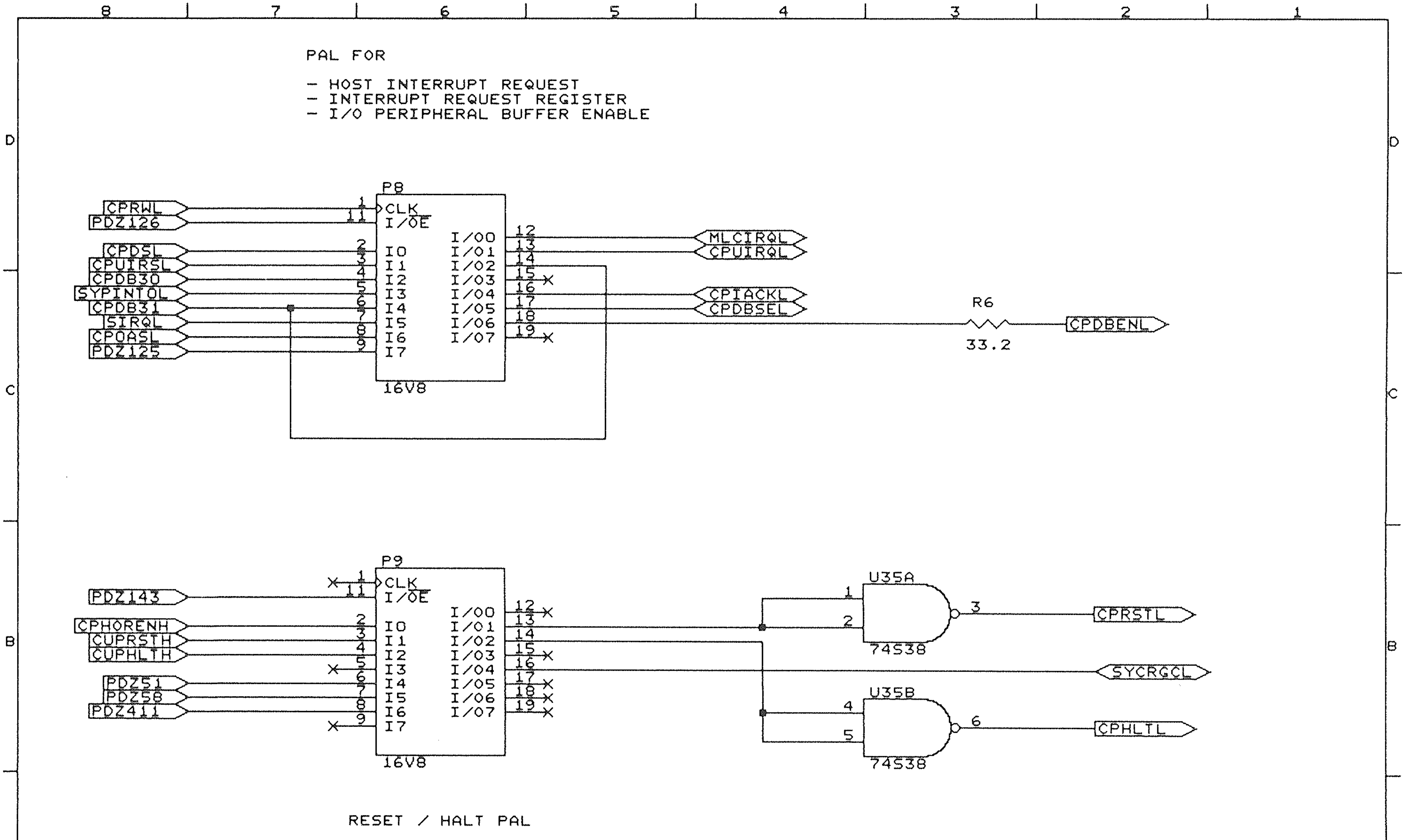
HOPETON SAINT JOHN WALKER
164H MERILINE AVENUE
WEST PATERSON, NJ 07424-3047

Title	LAPD INTERFACE - PAL1	
Size A	Document Number	REV
Date: November 9, 1994	Sheet 15 of 28	



HOPETON SAINT JOHN WALKER
164H MERILINE AVENUE
WEST PATERSON, NJ 07424-3047

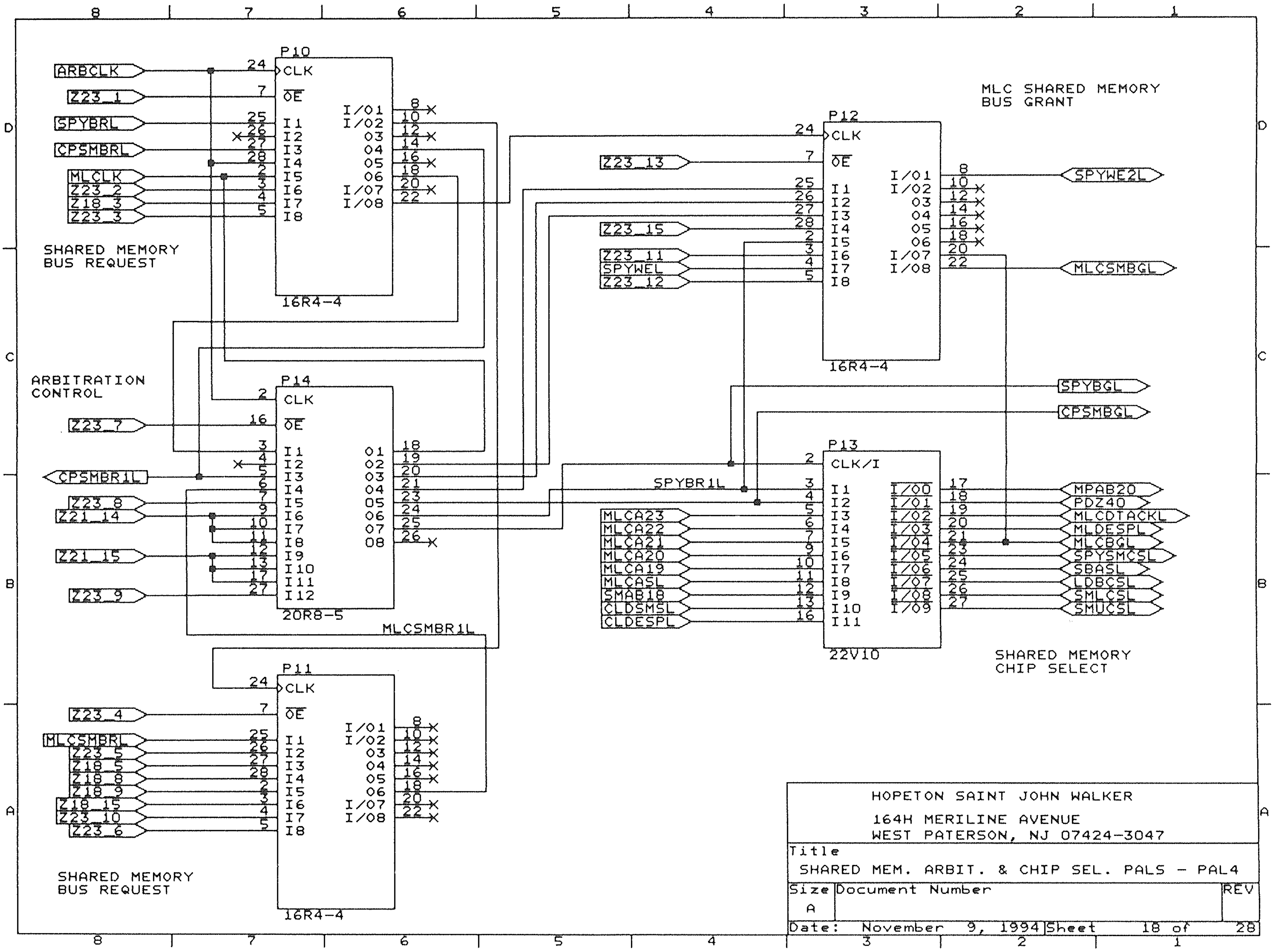
Title		
LAPD INTERFACE - PAL2		
Size A	Document Number	REV
Date: August 9, 1994	Sheet 16 of 28	



HOPETON SAINT JOHN WALKER

164H MERILINE AVENUE
WEST PATERSON, NJ 07424-3047

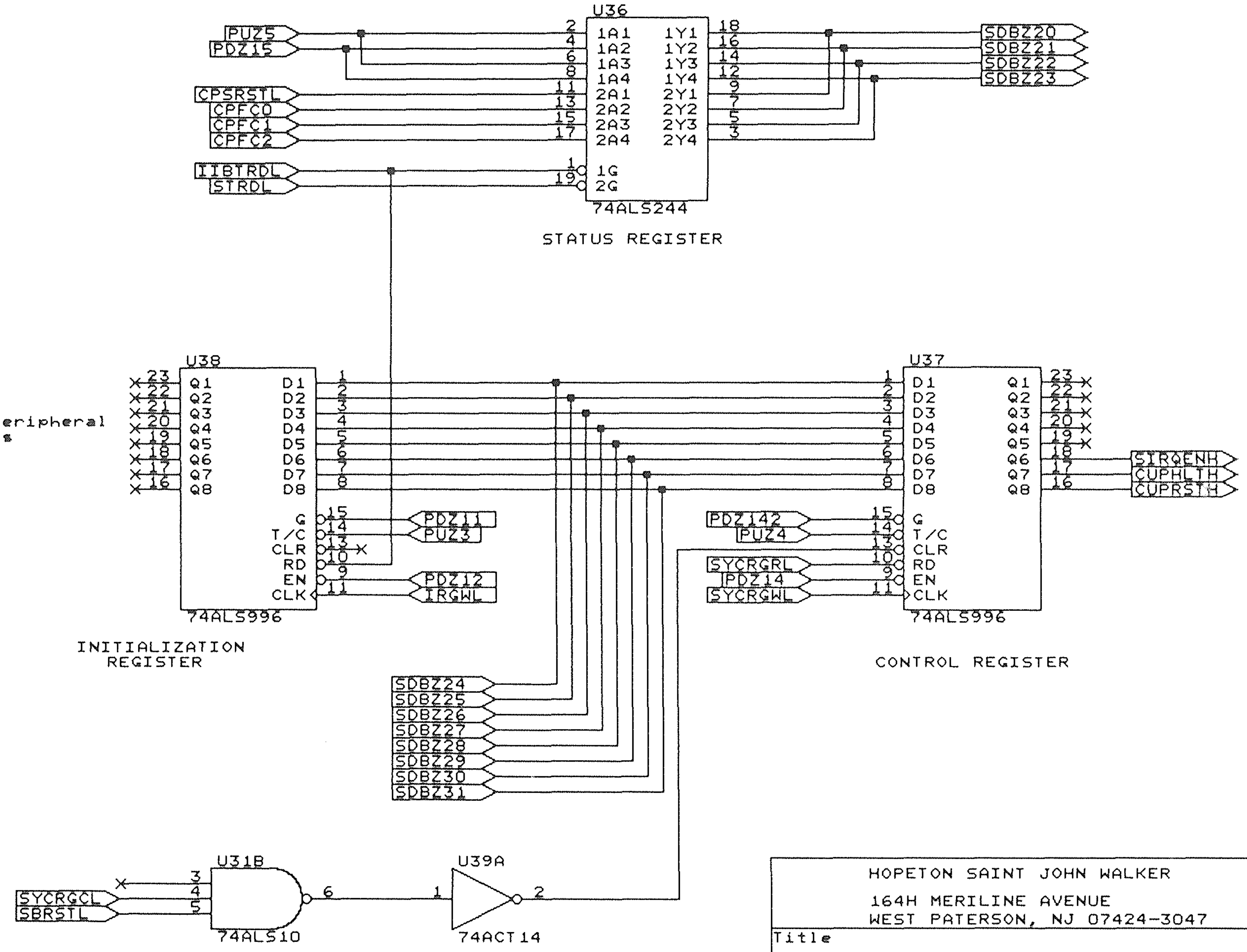
Title LAPD INTERFACE - PAL3		
Size A	Document Number	REV
Date: November 9, 1994	Sheet 17 of 28	



HOPETON SAINT JOHN WALKER

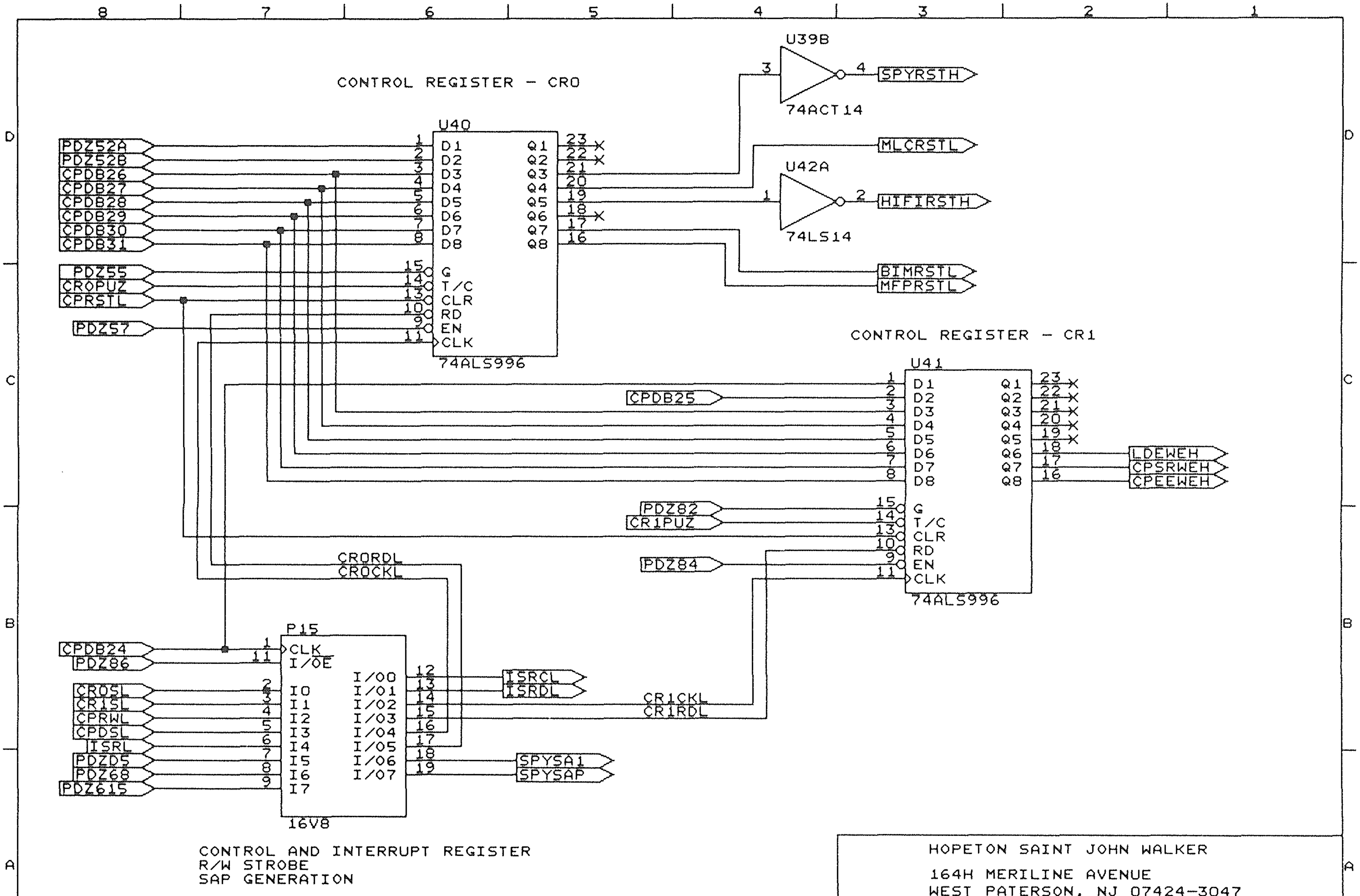
164H MERILINE AVENUE
WEST PATERSON, NJ 07424-3047

Title SHARED MEM. ARBIT. & CHIP SEL. PALS - PAL4		
Size A	Document Number	REV
Date: November 9, 1994	Sheet 18 of 28	



HOPETON SAINT JOHN WALKER
164H MERILINE AVENUE
WEST PATERSON, NJ 07424-3047

Title		
LAPD INTERFACE - REGISTERS (REG1)		
Size A	Document Number	REV
Date: November 9, 1994	Sheet 19 of 28	



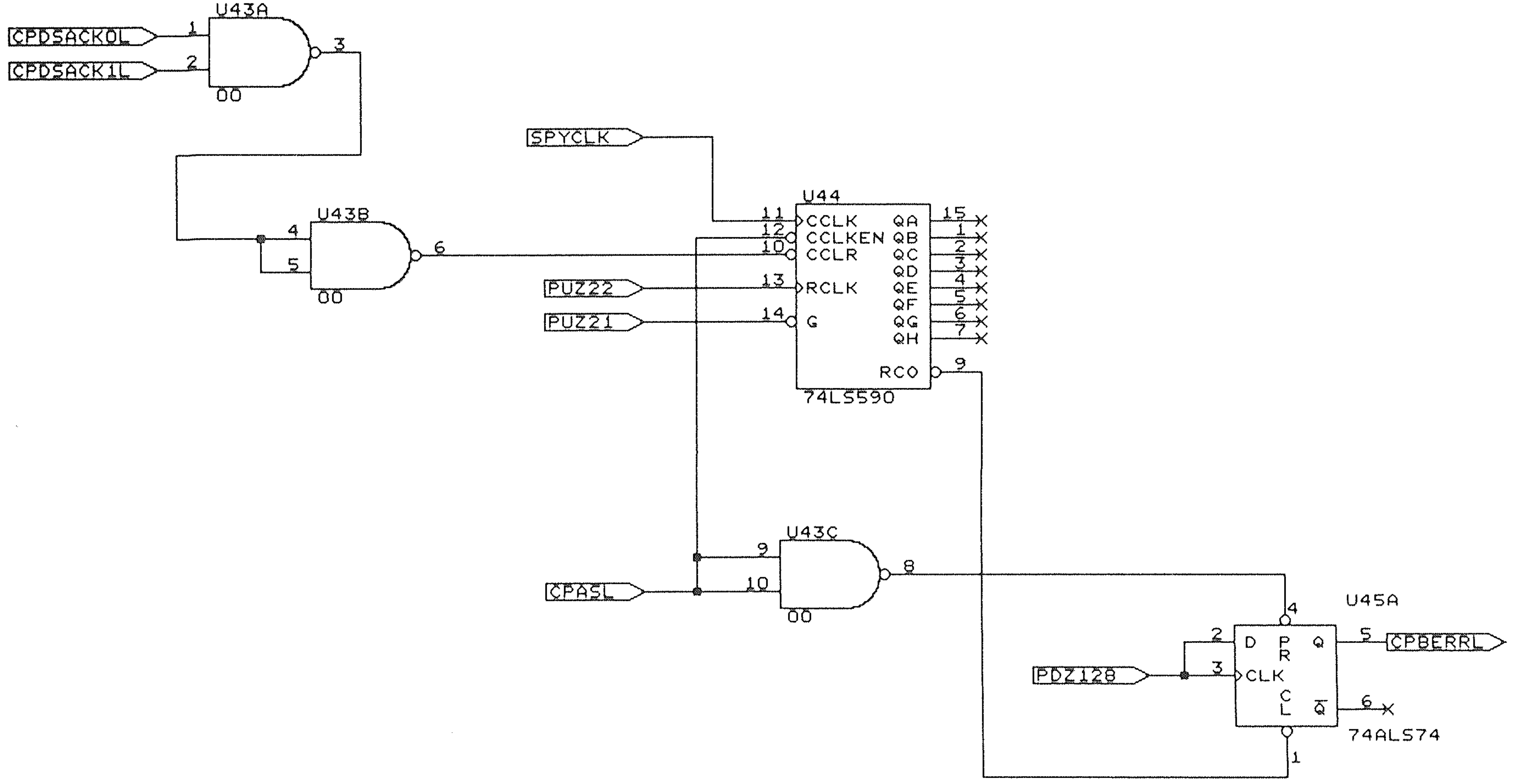
HOPETON SAINT JOHN WALKER

164H MERILINE AVENUE
WEST PATERSON, NJ 07424-3047

Title
LAPD INTERFACE - CONTROL REGISTERS (CR0,1)

Size	Document Number	REV
A		

Date: November 9, 1994 Sheet 20 of 28



HOPETON SAINT JOHN WALKER

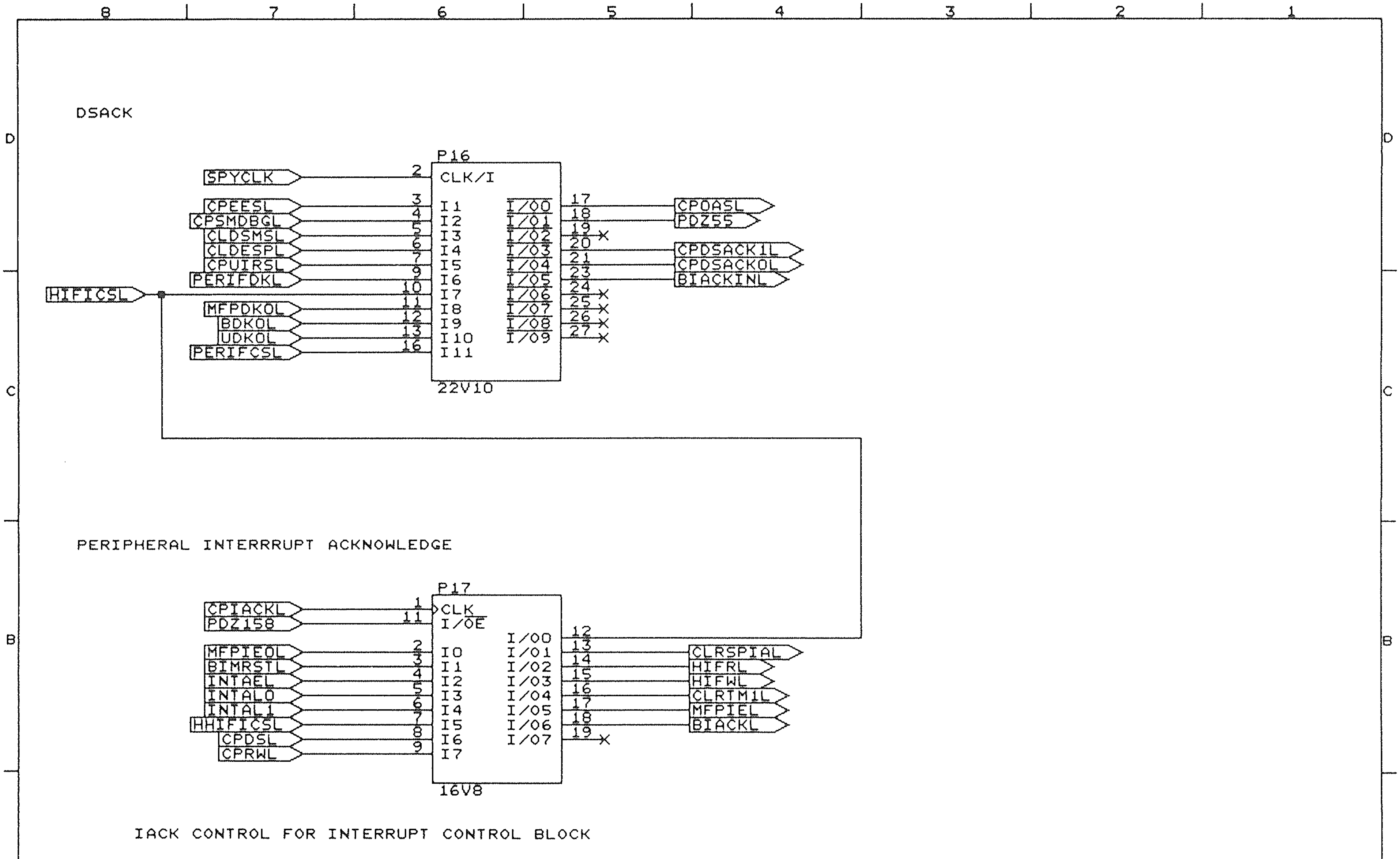
164H MERILINE AVENUE
WEST PATERSON, NJ 07424-3047

Title

LAPD INTERFACE - HOST BUS ERROR

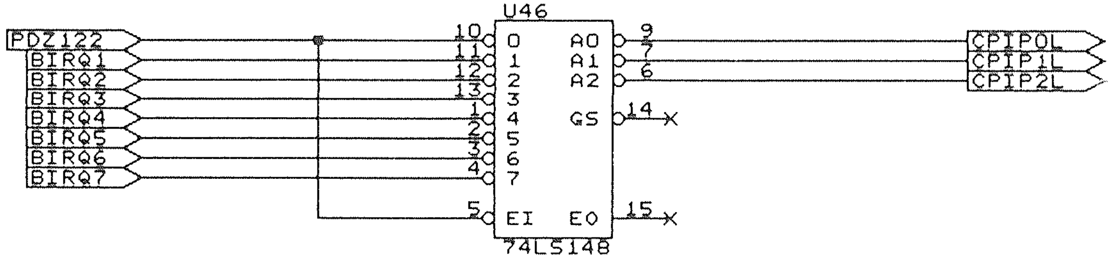
Size	Document Number	REV
A		

Date: August 13, 1994 Sheet 21 of 28



HOPETON SAINT JOHN WALKER
164H MERILINE AVENUE
WEST PATERSON, NJ 07424-3047

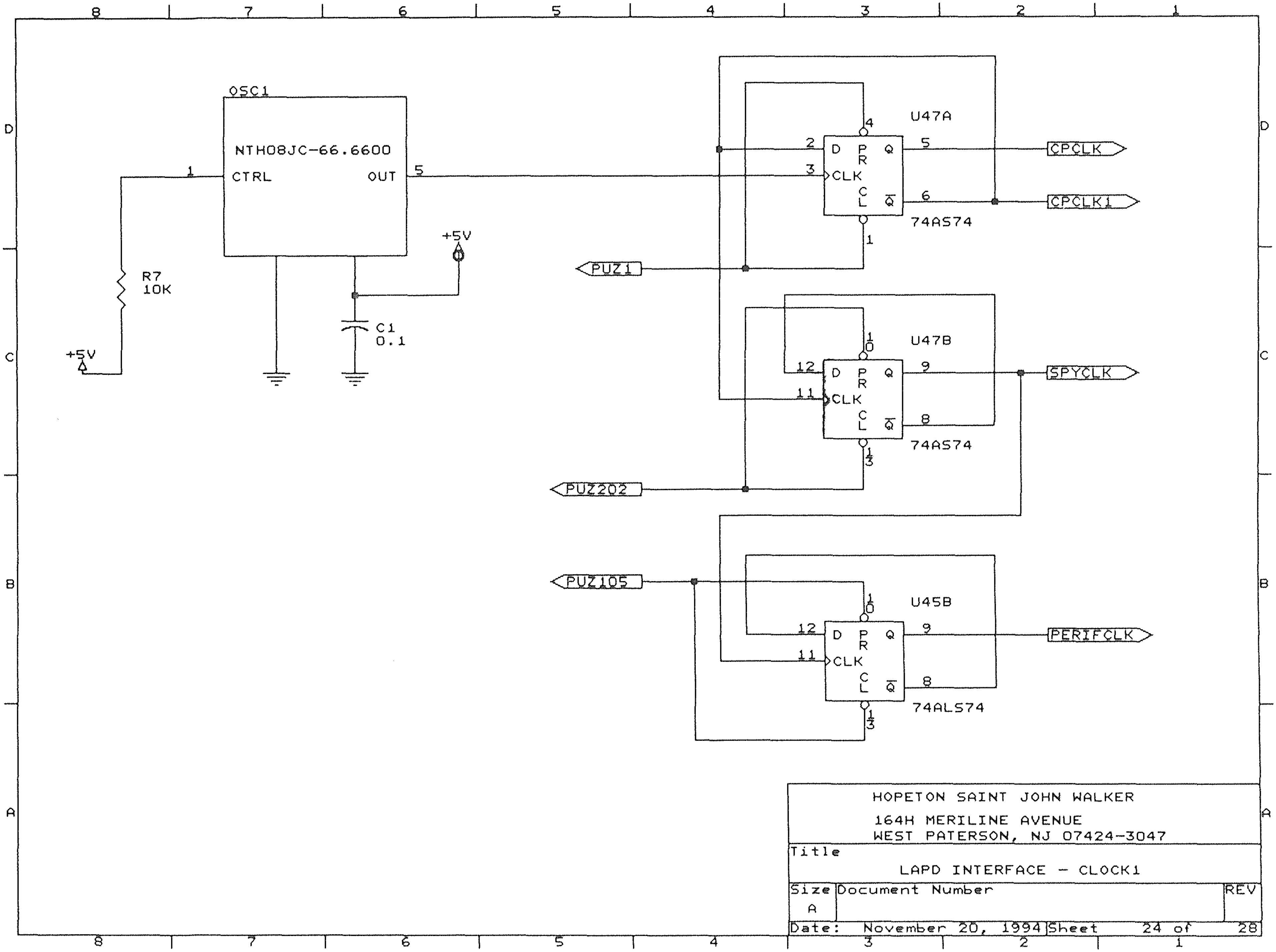
Title		
LAPD INTERFACE - INTERRUPT ACKNOWLEDGE/DSACK		
Size A	Document Number	REV
Date: November 9, 1994	Sheet 22 of	28



HOPETON SAINT JOHN WALKER

164H MERILINE AVENUE
WEST PATERSON, NJ 07424-3047

Title		
LAPD INTERFACE - INTERRUPT PRIORITY ENCODER		
Size A	Document Number	REV
Date: August 9, 1994	Sheet 23 of	28

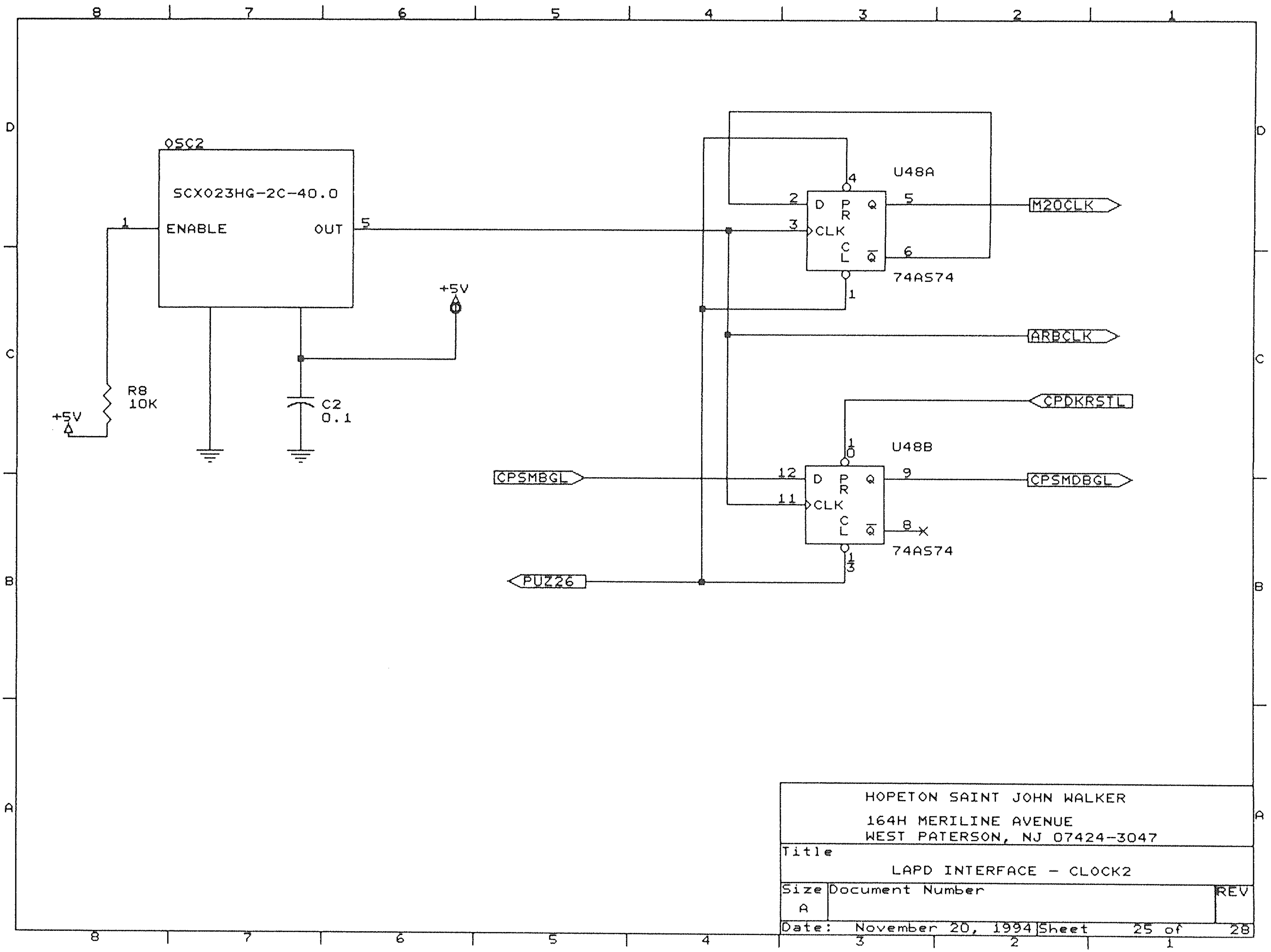


HOPETON SAINT JOHN WALKER
164H MERILINE AVENUE
WEST PATERSON, NJ 07424-3047

Title
LAPD INTERFACE - CLOCK1

Size	Document Number	REV
A		

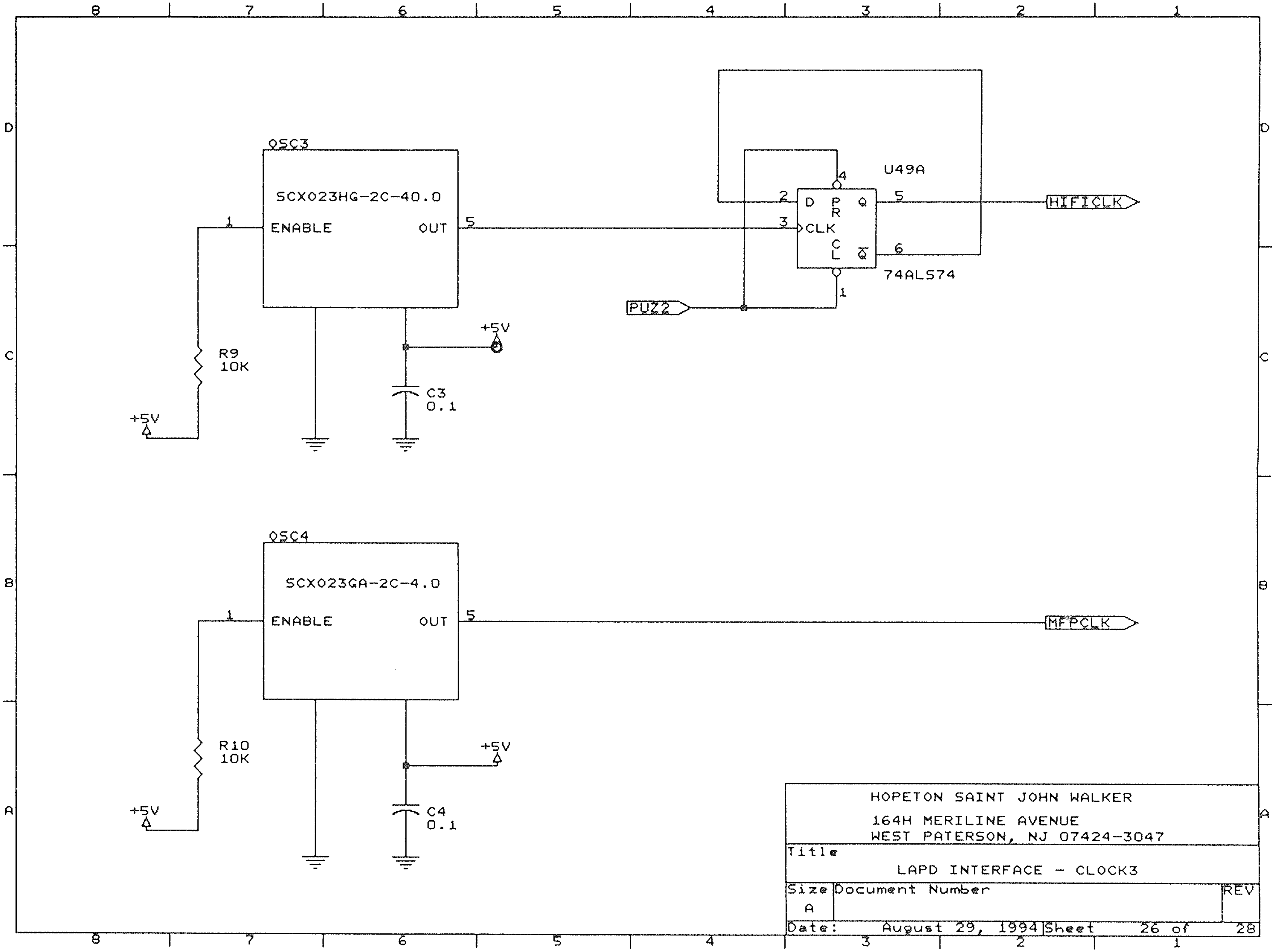
Date: November 20, 1994 Sheet 24 of 28



HOPETON SAINT JOHN WALKER
164H MERILINE AVENUE
WEST PATERSON, NJ 07424-3047

Title
LAPD INTERFACE - CLOCK2

Size	Document Number	REV
A		
Date: November 20, 1994	Sheet 25 of 28	



HOPETON SAINT JOHN WALKER

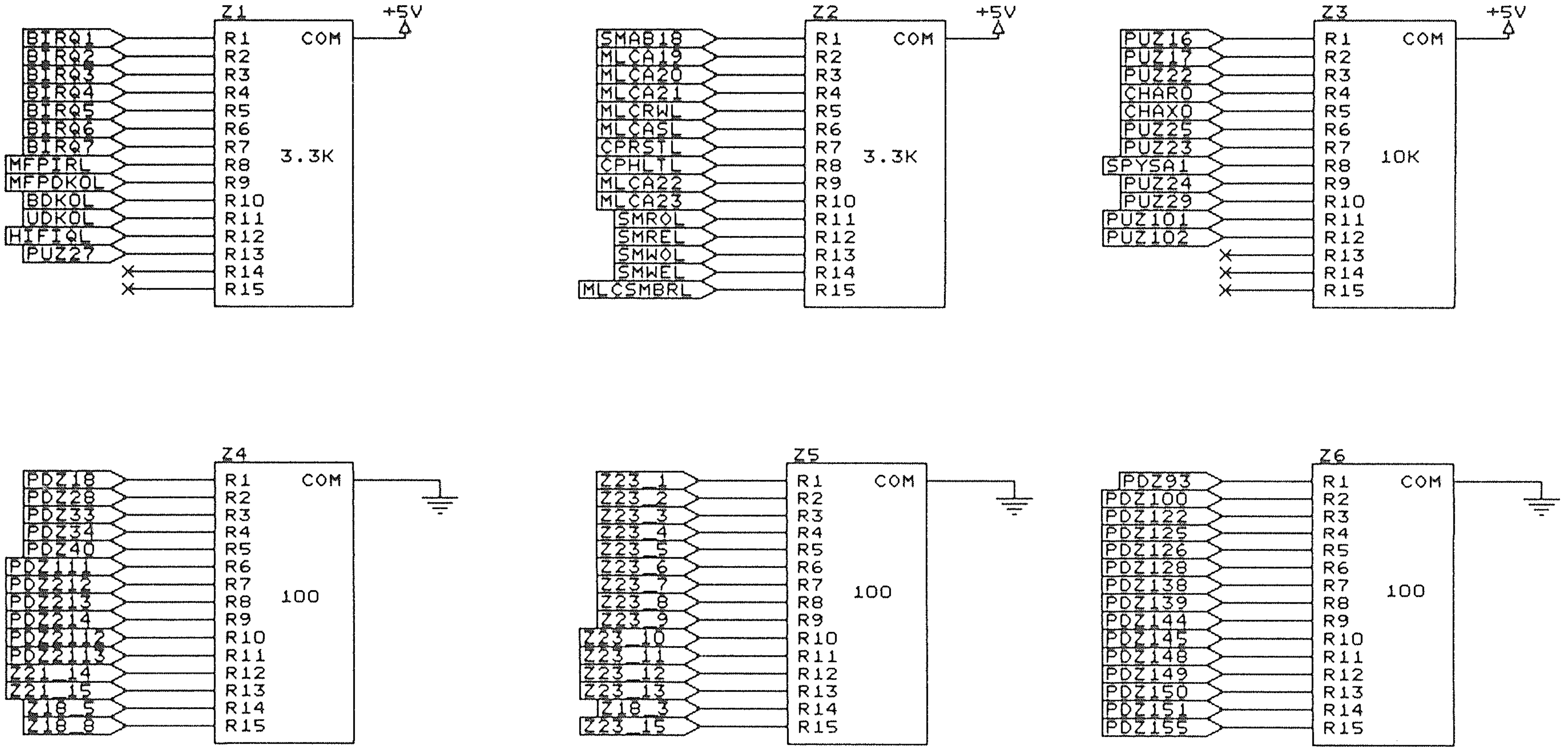
164H MERILINE AVENUE
WEST PATERSON, NJ 07424-3047

Title

LAPD INTERFACE - CLOCK3

Size	Document Number	REV
A		

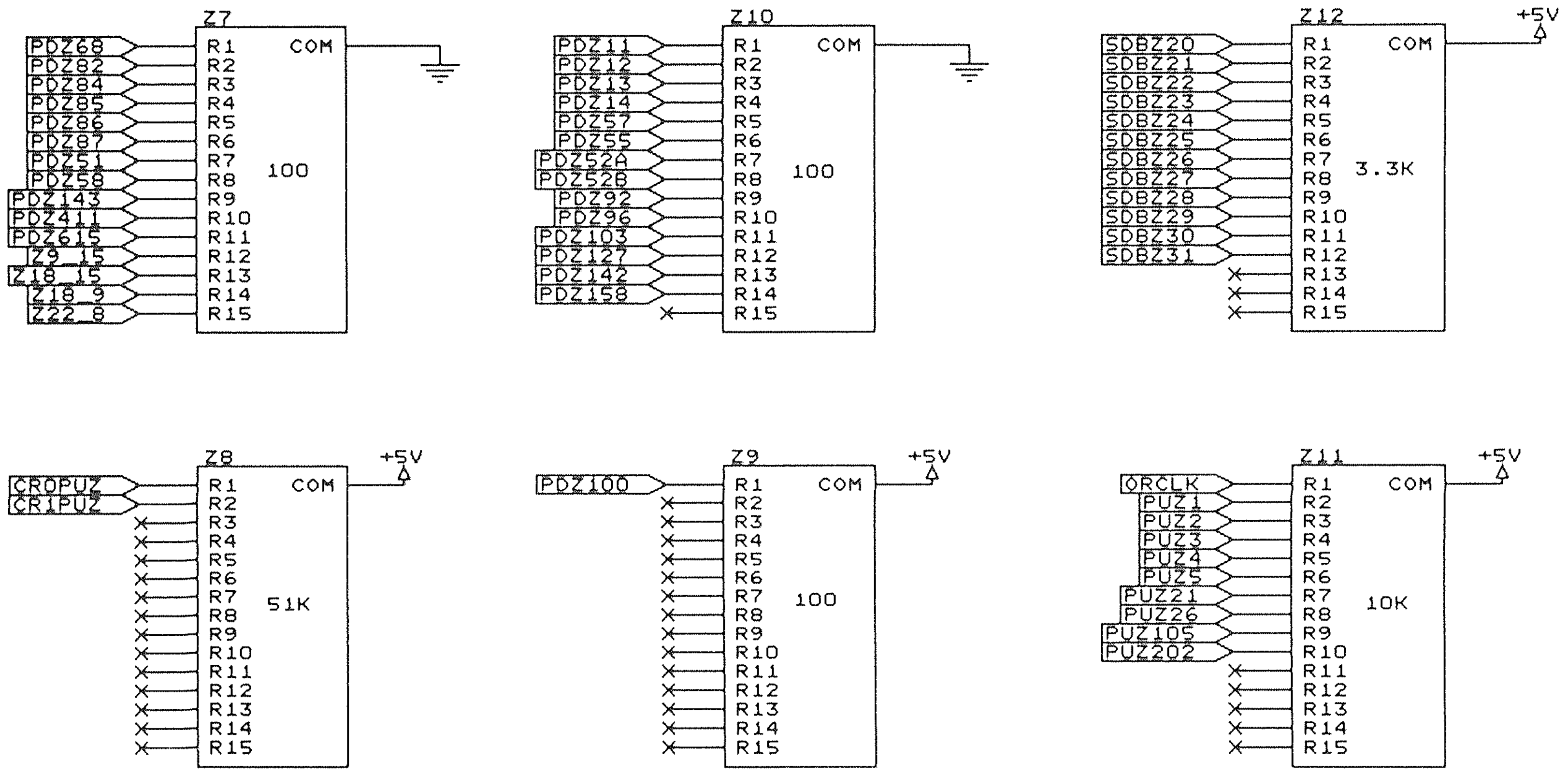
Date: August 29, 1994 Sheet 26 of 28



HOPETON SAINT JOHN WALKER

164H MERILINE AVENUE
WEST PATERSON, NJ 07424-3047

Title LAPD INTERFACE - RESISTOR NETWORK - PART 1		
Size A	Document Number	REV
Date: August 13, 1994	Sheet 27 of	28



HOPETON SAINT JOHN WALKER

164H MERILINE AVENUE
WEST PATERSON, NJ 07424-3047

Title		
LAPD INTERFACE – RESISTOR NETWORK – PART 2		
Size A	Document Number	REV
Date: November 9, 1994	Sheet 28 of 28	

REFERENCES

The following is a list of Data Sheets and Application Notes that were read in order to find out the functionality and limitations of the devices used for the design.

Advanced Micro Devices, *PAL Device Data Book : Bipolar and CMOS*: 1990.

Advanced Micro Devices, *PAL Device Data Book*: 1992.

AT&T Microelectronics. *Data Sheet: T7115A Synchronous Protocol Data Formatter*: April, 1990.

AT&T Microelectronics. *Application Note : T7115A and T7130 Bus Occupancy Considerations*: July, 1992.

AT&T Microelectronics. *Application Note : Memory Requirements for a T7130 Application*: July, 1992.

AT&T Microelectronics. *Application Note : T7115A, T7130 and Host Bus Arbitration Considerations*. July, 1992.

AT&T Microelectronics. *Data Sheet : T7130 Multi-channel LAPD Controller*: February, 1990.

AT&T Microelectronics. *Data Sheet : LAPD Software for T7130 Controller*: February, 1990.

AT&T Microelectronics. *Data Sheet : Common Software for T7130 Controller*: February, 1990.

AT&T Microelectronics. *Data Sheet : T7121 HDLC Interface for ISDN HIFI-64*: April, 1990.

Motorola, Inc. *M68000 Family Reference: M68000FR/AD*: 1989.

Motorola, Inc. *M68000 Programmer's Reference Manual: M68000PM/AD*: 1989.

Motorola, Inc. *M68000 8-/16-/32-Bit Microprocessor User's Manual: M68000UM/AD*: Englewood Cliffs, New Jersey: Prentice-Hall.

REFERENCES
(Continued)

Motorola, Inc. *MC68020 32-Bit Microprocessor User's Manual: 3ed. Mc68020UM/AD*: Englewood Cliffs, New Jersey: Prentice-Hall.

Motorola, Inc. *Motorola Memory Data*: Rev 6, 1990.

Motorola, Inc. *Fast and LS TTL Data*: 4ed. 1990.

Paradigm Technology, Inc. *Memory Data Book*: 1991

Texas Instrument, Inc. Data Transmission Circuits: Line Drivers, Receivers, Transceivers and UART Data Book: 1993.

REFERENCES
(Continued)

The following is a list of recommendations that were read in order to find out the specifications of the LAPD protocol.

European Telecommunications Standards Institute. Recommendation GSM, Sections 8.58,8.59.

European Telecommunications Standards Institute. Recommendation GSM, Sections 11.20.

Recommendations as proposed by Working Group T/WG 11 "Signaling, Protocols and Switching" (SPS). CCITT Recommendation T/S 46-20 (T/GSI 04-02/2 E). *ISDN User-Network Interface Data Link Specification*: Application of CCITT Recommendations Q.920/I.440 and Q.921/I.441.

REFERENCES
(Continued)

The following is a list of books that were read in order to gain more knowledge about the LAPD protocol and microprocessor interface design.

Li, Liu-cheng. *The M68000 Microprocessor Family: Fundamentals Of Assembly Language Programming and Interface Design*: Englewood Cliffs, New Jersey: Prentice-Hall, 1991.

Stallings, William. *ISDN and Broadband ISDN*, New York: Macmillian Publishing Company, 1992.

Stallings, William. *Data and Computer Communications*, 3rd ed. New York: Macmillian Publishing Company, 1991.

Tanenbaum, Andrew. S. *Computer Networks*. 2nd ed. Englewood Cliffs, New Jersey: Prentice-Hall, 1989.

Wilcox, Alan D. 68000 Microcomputer Systems: Englewood Cliffs, New Jersey: Prentice-Hall, 1987.