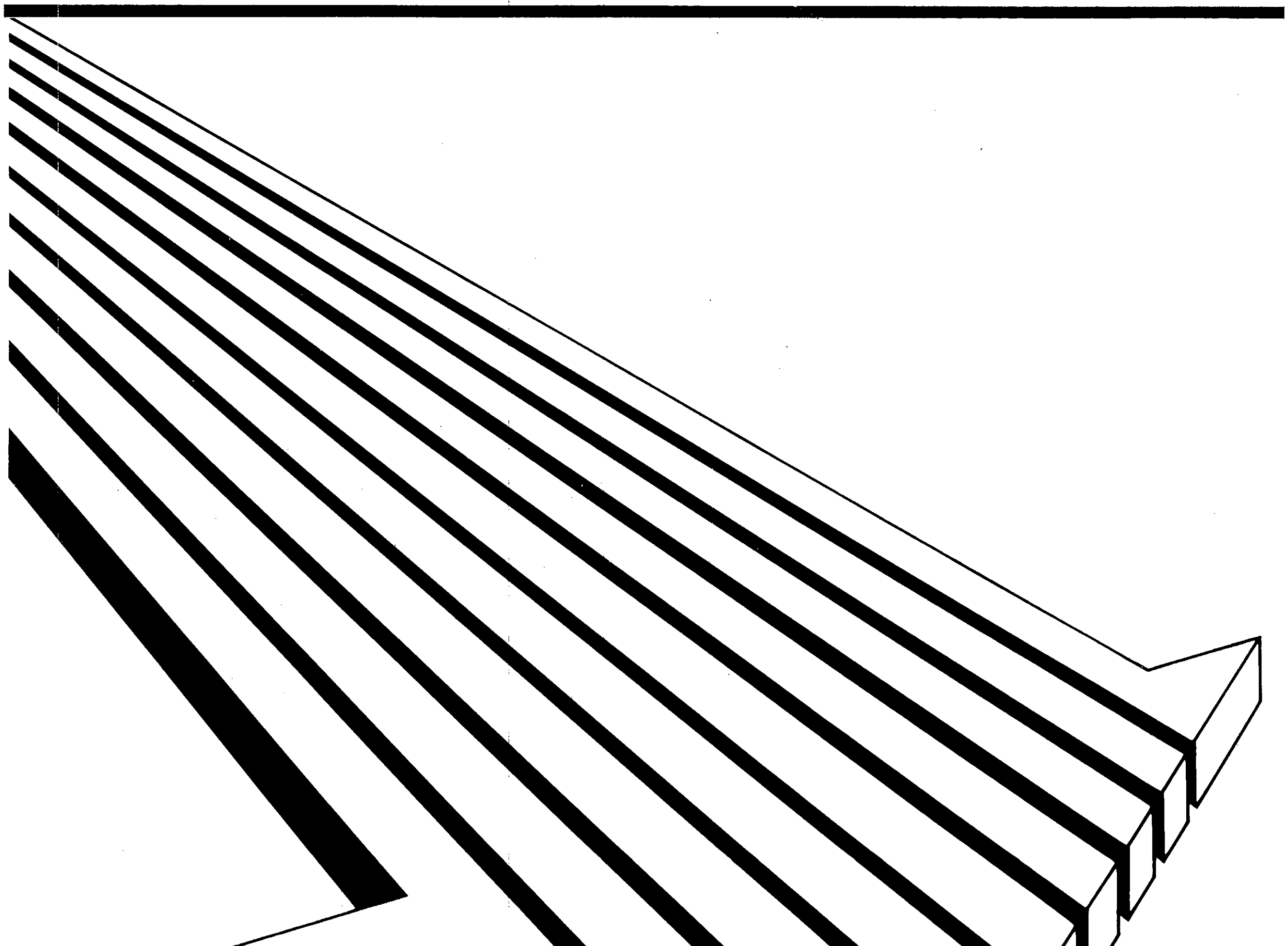
INTEL iLBX™ BUS SPECIFICATION



Intel Corporation, 3065 Bowers Avenue, Santa Clara, California 95051

Order Number: 145695-Rev. A

INTEL iLBX™ BUS SPECIFICATION

Order Number: 145695-Rev. A

Intel Corporation, 3065 Bowers Avenue, Santa Clara, California 95051

REV.	REVISION HISTORY	PRINT DATE
A	Original Issue.	1/83

Additional copies of this manual or other Intel literature may be obtained from:

Literature Department
Intel Corporation
3065 Bowers Avenue
Santa Clara, CA 95051

The information in this document is subject to change without notice.

Intel Corporation makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Intel Corporation assumes no responsibility for any errors that may appear in this document. Intel Corporation makes no commitment to update nor to keep current the information contained in this document.

Intel Corporation assumes no responsibility for the use of any circuitry other than circuitry embodied in an Intel product. No other circuit patent licenses are implied.

Printed in U.S.A./OM-006/3K/0283/AP

PREFACE

The iLBX bus is one of the subsidiary buses within the overall Multibus Interface System. The iLBX bus is stand-alone to the extent that its interface and protocol do not require the existence of the general purpose Multibus interface or any of the other subsidiary buses. The iLBX bus uses the form factor of the Multibus P2 connector and imposes restrictions on board designs implementing both the general purpose Multibus interface and the iLBX bus.

This specification describes the operation protocol of the iLBX bus and defines the electrical and mechanical requirements of the iLBX bus. A section of design guideline examples provide additional insight for implementing the iLBX bus in a system. This specification does not duplicate specification information from the Multibus Interface Specification or any of the subsidiary bus specifications. Information on the Multibus interface or a subsidiary bus is provided in the following specifications.

- *Intel Multibus® Interface Specification,* Order Number: 9800683
- *Intel iSBX™ Bus Specification,* Order Number: 142686
- *Intel Multichannel™ Bus Specification,* Order Number: 144330

This specification follows the general guidelines in the "Recommendations on Terminology for IEEE Computer Society Interface Standards" review draft dated September 9, 1981, and revised November 3, 1981, and June 3, 1982. In compliance with the terminology recommendations, this specification use decimal notation when numbering bus lines with bit 0 as the least significant bit. This specification also uses the trailing asterisk to designate active Low signal lines. Where Multibus interface signal names (or subsidiary bus signal names) are used in this specification, these names are converted to comply with the terminology recommendations. For example, the Multibus address extension line ADR14/ is listed in this specification as ADR20*.

CONTENTS

TITLE SHEET NO.

TITLE SHEET NO.

FIGURES

TABLES

1. Introduction

The Local Bus Extension (iLBX) bus is a specialized electrical and mechanical interfacing protocol operating within the overall Multibus interfacing system. The iLBX bus provides local memory expansion which is physically off-board, without loss of execution speed. A typical iLBX installation would have a master iSBC processor board attached to an additional memory board via the iLBX bus lines. Because of the increased execution speed of the iLBX bus, this off-board memory will be used as local on-board memory by the processor board. Up to five boards may use the iLBX bus in one system.

1.1 Multibus® Interface Overview

Figure 1 illustrates the overall Multibus interfacing system. The foundation of the Multibus interfacing system is the general purpose Multibus interface, the flexible bus structure used to interface the family of Intel's 80/86 products including both 8- and 16-bit products. The Multibus interface supports both 8- and 16-bit data transfers and direct addressability of up to 16 megabytes of memory. In many systems, the Multibus interface provides all of the required interconnect capability for the system.

As systems grow in complexity and performance, the throughput demands on the interconnect architecture increase. The Multibus interfacing system meets these demands by off-loading specific interconnect needs to the following subsidiary buses:

- iSBX™ bus
- Multichannel™ bus
- iLBX™ bus

In a fully expanded Multibus interface system, the Multibus interface is used mainly for system control and low to medium-speed data transfers.

1.1.1 iSBX™ Bus

Increasing the number of functions residing on each system board attached to the Multibus interface increases the system performance. The improved system performance results because the resident functions are accessed without bus arbitration. The trade-off becomes choosing the resident functions when designing the system board. The iSBX bus extension of the Multibus interfacing system helps reduce the need to make design choices. The special functions are designed onto individual small boards and connected to a system board using the iSBX bus interface. When installed on the system board, the special function operates as though it were residing on the system board. Thus a system designer can have resident on the system boards those special functions most advantageous to his system.

1.1.2 Multichannel™ Bus

Reducing the impact of burst-type peripherals (e.g. most disk peripherals) on the Multibus® interface provides a second means of increasing system performance. The actual data transfers from a burst type peripheral can saturăte a general purpose interface such as the Multibus interface. Adding more burst-type peripherals to a system often decreases the computing performance of the system. The Multichannel bus extension to the Multibus interfacing system helps reduce the bus-saturation problem. The Multichannel bus protocol specifically accommodates burst-type data transfers. The full performance improvement requires use of dual port memory accessed over both the Multichannel bus and the Multibus interface.

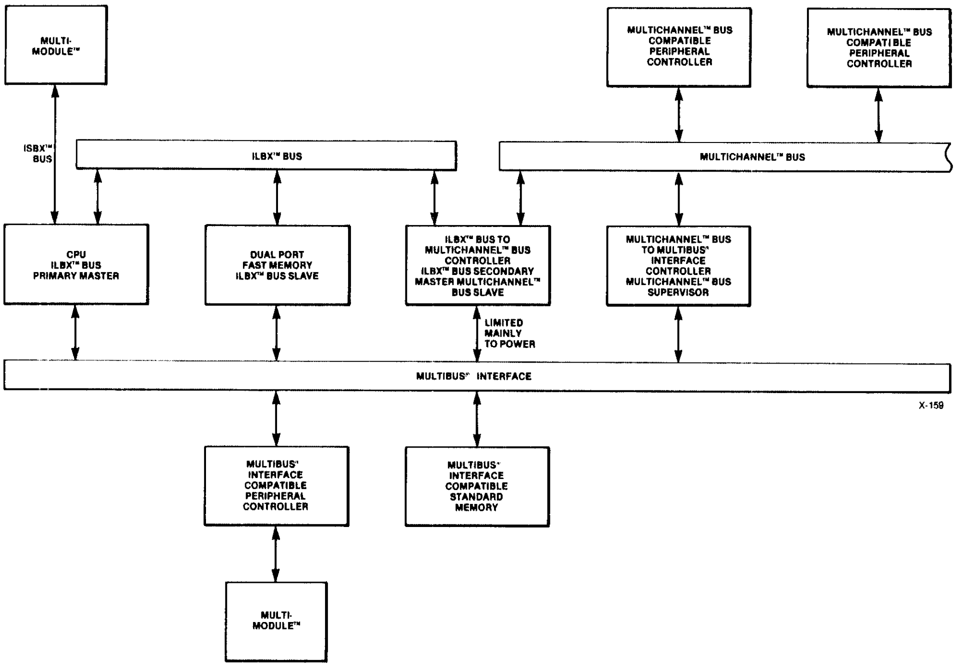


Figure 1 Multibus® Interface System

1.1.3 iLBX™ Bus

Dramatically increasing the local (on-board) memory resources of a high performance processor provides a third means of increasing system performance. As with other special functions, memory residing on the processor board improves system performance because the processor directly addresses the memory without waiting for bus arbitration. However, there is a physical limitation to the amount of memory that can reside on the processor board. The iLBX bus helps to reduce the physical space limitation. Using the iLBX bus, the additional memory no longer needs to be located on the processor board or on a multimodule attached to the processor board. The full 16 megabytes of memory addressable by the processor can be accessed over the iLBX bus and appear to the processor as though it were resident on the processor board. Dual porting the memory between the iLBX bus and the Multibus interface makes the same memory resources available to other system components.

1.2 iLBX™ Bus General Description

The iLBX bus configuration uses the form-factor of the standard 60-pin Multibus P2 connector. It occupies 56 of the P2 connector pins (55 defined signal lines and one reserved signal line). The four Multibus address extension lines (pins 55 through 58) on the Multibus P2 connector retain the standard Multibus interface functions.

The iLBX bus is designed for direct high-speed Master-Slave data transfers and provides the following features.

- A minimum of two and up to five devices can be connected over the iLBX bus.
- Two (maximum) masters can share the bus, limiting the need for bus arbitration.
- Bus arbitration is asynchronous to the data transfers.
- Slave devices are defined as byte-addressed memory resources.
- Slave device functions are directly controlled from iLBX bus signal lines.

2. Functional Description

2.1 Introduction

The Functional Description defines the various elements of the iLBX bus interface. These elements include descriptions of the device categories using the iLBX bus, the signal line grouping and functions, the timing requirements, and the bus communication protocol.

2.2 Notation Conventions

The general notational conventions used in this manual conform to the notational conventions used in the Multibus Specification. The following paragraphs summarize the notational conventions. The iLBX bus lines are assigned unique names and, for brevity, unique mnemonics. The signal line names are shown with initial capital letters when used in text. The corresponding signal line mnemonics are shown in all capital letters. Signal mnemonics for lines that are Active when electrically High (also called positive true) or Active when either electrically High or Low (the data lines for example) do not have a special terminating character as the last character in the mnemonic. Signal mnemonics for lines Active when electrically Low (also called negative true) have an asterisk (*) character as the last (terminating) character in the mnemonic.

The descriptions of electrical signal characteristics use the terms High and Low (in initial capital letters) corresponding to the relative voltage level of the signal. The terms true, false, 1, and 0 are avoided to reduce misinterpretation. Thus an Active Low signal is asserted when its relative voltage level is Low. Table 1 relates the electrical signal characteristics to the corresponding logical and state notations. The Example mnemonic, XMPL, illustrates the notational conventions.

The address and data bit numbering scheme used with the iLBX bus has bit 0 as the least significant bit, and decimal numbers are used to identify the lines. Thus, Data line 0 (DB0) carries the least significant bit and Data line 15 (DB15) carries the most significant bit.

Table 1 Notational Summary

SIGNAL NAME	NOTATION		
	ELECTRICAL	LOGICAL	STATE
XMPL	H, High	1, True	Active, Asserted
	L, Low	0, False	Inactive
XMPL*	H, High	0, False	Inactive
	L, Low	1, True	Active, Asserted

2.3 iLBX™ Bus Devices

Three device categories interface to the iLBX bus. The device categories are the following:

- Primary Master
- Secondary Master
- Slave

At most, five devices can be simultaneously attached to the iLBX bus. The set of devices must include a *Primary Master* and one *Slave* device. The remaining three devices are optional and may include additional Slave devices and one *Secondary Master*. The Slave device(s) contain the memory resources used by the Primary Master and the Secondary Master. The combined, directly accessible, memory total for the Slave devices is 16 megabytes.

2.3.1 Primary Master

The Primary Master controls the iLBX bus and manages the Secondary Master access to the Slave memory resources. Every implementation of the iLBX bus must have a device performing all, or an allowed subset, of the Primary Master functions. A maximum of one Primary Master can be attached to the iLBX bus. Typically the system processor board includes the Primary Master function as an on-board function with the iLBX bus as an extension of the Primary Master's on-board local bus.

The Primary Master must perform three specific functions in addition to general iLBX-bus control. The Primary Master must actively drive all iLBX bus signal lines (except Slave Acknowledge and Secondary Master Request) unless it relinquishes signal line control to the Secondary Master. It must provide a +5VDC pull-up termination for those iLBX bus lines requiring termination. It must monitor the Secondary Master Request signal line and release control of the iLBX bus to a Secondary Master for data transfer.

Single-master implementations are allowed where a limited Primary Master does not monitor the Secondary Master Request signal line. When the Primary Master lacks the ability to monitor the Secondary Master Request signal line, a Secondary Master cannot be attached to the iLBX bus.

2.3.2 Secondary Master

The optional Secondary Master provides alternate access over the iLBX bus to the Slave resources. The Secondary Master transfers data over the iLBX bus in the same manner as a Primary Master; however, the Secondary Master must first request control and the Primary Master must acknowledge the request to pass iLBX bus control to the Secondary Master. The specified maximum of two masters (one Primary Master and one Secondary Master) reduces bus arbitration to a simple request and acknowledgement process. Bus control arbitration occurs asynchronously to the data transfers.

When the Secondary Master controls the iLBX bus, the Secondary Master must actively drive all signal lines (except Slave Acknowledge and Secondary Master Acknowledge) until it returns signal line control to the Primary Master. The Secondary Master must not provide additional signal line termination.

The Secondary Master must provide a means for varying the timing of its response to the Slave Acknowledge. The iLBX bus data-transfer-timing allows closely coupling (optimizing) the Slave device's data-transfer rate to the Primary Master's data-transfer rate. When the close coupling is implemented, part of the Slave device's access overhead occurs concurrently with the Primary Master's acknowledge acceptance overhead. When implementing the iLBX bus in a system, the Secondary Master's acknowledge acceptance timing must be adjusted to match the Primary Master's acknowledge acceptance timing to assure reliable data transfers.

2.3.3 Slave

The Slave device(s) contain the memory resources used by the Primary Master and the Secondary Master. The combined, directly accessible, memory total for the Slave devices is 16 megabytes. Any given iLBX bus implementation is limited to a maximum of four Slave devices.

The Slave devices must continually monitor the iLBX bus Address lines and the Address Strobe line. The Slave device detecting an assigned memory address on the Address lines assumes selection, and data transfer initiation occurs when the Slave device detects the Address Strobe signal. The iLBX bus protocol requires a positive response to complete any selection, and Slave devices must be able to actively drive the Slave Acknowledge line.

The iLBX-bus data-transfer-timing allows closely coupling (optimizing) the Slave device's data-transfer rate to the Primary Master's data-transfer rate. When the close coupling is implemented, part of the Slave device's access overhead occurs concurrently with the Primary Master's acknowledge acceptance overhead. Use of the optimized operation is optional, and a Slave device designed to implement optimized operation must provide a means for varying the timing of its Slave Acknowledge. Refer to the timing specifications and timing adjustment requirements in the hardware reference manual for iLBX bus compatible devices.

2.4 Signal Line Descriptions

The following four signal line categories make up the physical structure of the iLBX bus interface:

- address and data lines
- control lines
- command lines
- bus access lines

2.4.1 Address and Data Lines

The bus has 16 data lines and 24 address lines.

2.4.1.1 DATA LINES (DB15 - DB0). All 8-bit and 16-bit data transfers between the active bus master and the selected Slave device use the 16 bi-directional data lines exclusively. The 16-bit data transfers use all 16 data lines. The 8-bit data transfers within the 16-bit data frame use the appropriate low-order (DB7 through DB0) or high-order (DB15 through DB8) data lines.

Byte data is transferred between 8-bit devices using data lines DB7 through DB0. The state of unused data lines during 8-bit data transfers is undefined.

The general data signal line implementation specifications are as follows.

- The data lines require tri-state drivers and any iLBX bus device can drive the data lines.
- The data lines are positive true lines and only the *transmitting* device (master or slave) drives the lines. The *receiving* device and all inactive devices must hold their data line drivers in the high-impedance state during the data transfer.

2.4.1.2 Address Lines (AB0 - AB23). The active bus master uses 24 address lines to select a Slave device and to specify a location in memory. The use of 24 address lines provides the ability to address 16-megabytes of memory space.

Only the active bus master drives the address lines in the iLBX bus. The condition of the address lines during the data time is undefined, and the Slave device must store (latch-in) the address information when the Slave device detects the Address Strobe signal (see Section 2.4.3.1).

The general address signal line implementation specifications are as follows.

- The address lines are positive true lines.
- Primary Masters and Secondary Masters must provide tri-state drivers for the address lines. Limited Primary Masters that do not share the iLBX bus with a Secondary Master can use standard TTL drivers with drive characteristics comparable to the specified tri-state driver.

2.4.1.3 Transfer Parity (TPAR*). The Transfer Parity signal is an optional line and is available to improve data-transfer integrity. The Transfer Parity operates as an additional data line with identical timing requirements. The iLBX bus uses odd parity defined as follows: when there is an even number of one bits in the transfer element (byte, 16-bit word), the transmitting device drives the Transfer Parity line Low. Because the state of unused data lines is undefined, parity generation and checking is limited to the active data lines for the transfer element used. The iLBX bus does not provide a means for reporting a detected transfer parity error.

The general Transfer Parity signal line implementation specifications are as follows.

- The option must be available on all devices, both masters and slaves, if a transfer parity option is to be used on the iLBX bus.
- Any device designed with the parity option must provide a means for disabling recognition of a transfer parity error.

- All masters and Slave devices with the parity option must provide a tri-state driver for the Parity line.

- Only the transmitting device (master or slave) drives the Parity line. The receiving device must sample the Parity line and all inactive devices must hold their parity line driver in the high-impedance state during the data transfer.

2.4.2 Control Lines

The active bus master specifies the data transfer parameters to the selected Slave device by using the three control lines.

2.4.2.1 Read-Not-Write ($R/\overline{W}$). The active bus master controls the direction of data transfer with the Read-Not-Write line. When driven Low, the active bus master transmits the data and the selected slave device receives the data. Driving the Read-Not-Write line High reverses the transfer direction.

The Read-Not-Write signal need not remain valid throughout the data transfer operation. Therefore, the Slave device must store (latch-in) the state of the Read-Not-Write signal line when the Slave device detects the leading (falling) edge of the Address Strobe (see Section 2.4.3.1).

The general Read-Not-Write signal line implementation specifications are as follows.

- Primary Masters and Secondary Masters must provide a tri-state driver for the Read-Not-Write line. Limited Primary Masters that do not share the iLBX bus with a Secondary Master can use a standard TTL driver with drive characteristics comparable to the specified tri-state driver.

- Only the active bus master drives the Read-Not-Write line. The inactive master must hold the read-not-write line-driver in the high-impedance state.

Table 2 16-Bit Data Frame

BITS 15 - 8	BITS 7 - 0
HIGH BYTE	LOW BYTE
WORD	

Table 3 Element Selection

ELEMENT	SIGNAL & LEVEL	
	BHEN	AB0
HIGH BYTE	HIGH	HIGH
LOW BYTE	LOW	LOW
RESERVED	LOW	HIGH
WORD	HIGH	LOW

2.4.2.2 Data Element Select Control Line (BHEN). The iLBX bus data transfers take place within an overall data frame that is limited by the iLBX bus configuration. Within the data frame limits, transfer of an 8-bit (byte) and 16-bit (word) data element is allowed. The particular size data element and its location within a data frame must be specified to the slave device by the active bus master. Refer to Table 2 for element size and location within a sixteen bit data frame.

The active bus master controls the type of data transfer (8-bit or 16-bit) using the Byte High Enable (BHEN) element select line and the low-order address bit (AB0). The four signal-level combinations of these two lines specify both the element size and the element location within the data frame. The signal-level combinations are shown in Table 3.

The BHEN signal is not required to remain valid throughout the data transfer operation. The Slave device must store (latch-in) the configuration of the signal line when the Slave device detects the Address Strobe signal.

The general BHEN signal line implementation specifications are as follows.

- Primary Masters and Secondary Masters must provide tri-state drivers. Limited Primary Masters that do not share the iLBX bus with a Secondary Master can use a standard TTL driver with drive characteristics comparable to the specified tri-state driver.

- The Byte High Enable line must be implemented on all masters and Slave devices.

- Only the active bus master drives BHEN. The inactive master must hold the BHEN line driver in the high-impedance state.

2.4.3 Command Lines

The active bus master and the selected Slave device use three command lines to initiate, control, and terminate a data transfer.

2.4.3.1 Address Strobe (ASTB*). The active bus master drives the Address Strobe line Low to initiate a data transfer cycle. The control line and address line signal levels must be valid prior to the bus master driving the Address Strobe line Low. The Slave device(s) decodes the address to determine if it is selected. The selected Slave device must store the control and address information at the leading (falling) edge of the Address Strobe signal. The selected Slave device then proceeds with the data transfer.

The general Address Strobe signal line implementation specifications are as follows.

- Primary Masters and Secondary Masters must provide a tri-state driver for the Address Strobe line. Limited Primary Masters that do not share the iLBX bus with a Secondary Master can use a standard TTL driver with drive characteristics comparable to the specified tri-state driver.

- Only the active bus master drives the Address Strobe line. The inactive master must hold its address strobe line driver in the high-impedance state.

2.4.3.2 Data Strobe (DSTB*). The active bus master drives the Data Strobe line Low to set-up the actual transfer of data. The active bus master drives the Data Strobe line High after the data transfer is completed to terminate the data transfer cycle. The meaning of the Data Strobe signal varies depending on the direction of data transfer, from master to slave (write) or from slave to master (read).

During a read operation, the active bus master indicates when it is ready to accept data from the selected slave device by driving the Data Strobe line Low. The active bus master must put its data line (DB15 - DB0) tri-state drivers in the high-impedance state before driving the Data Strobe line Low. The selected slave device starts driving the required lines after detecting the leading (falling) edge of the Data Strobe signal.

During a write operation, the active bus master indicates the availability of data for the selected slave device by driving the Data Strobe line Low. The active bus master is allowed a set-up time for the data lines after it drives the Data Strobe line Low. The selected slave device samples the data after first detecting the leading (falling) edge of the Data Strobe signal plus the specified data set-up time.

The general Data Strobe signal line implementation specifications are as follows.

- Primary Masters and Secondary Masters must provide a tri-state driver for the Data Strobe line. Limited Primary Masters that do not share the iLBX bus with a Secondary Master can use standard a TTL driver with drive characteristics comparable to the specified tri-state driver.

- Only the active bus master drives the Data Strobe line. The inactive master must hold its Data Strobe line driver in the high-impedance state.

- The active bus master must hold the Data Strobe line Low for the specified minimum time after it receives the Acknowledge signal from the selected slave device.

2.4.3.3 Acknowledge (ACK*). The selected slave device responds to the active bus master by driving the Acknowledge line low.

The Acknowledge signal timing requirements allow flexibility. The flexibility allows overlapping the internal operation overhead of the selected master with the bus set-up time of the selected slave device. Thus during a read operation, the selected slave device can drive the Acknowledge line Low prior to completing bus set-up of the data lines. The slave device must allow sufficient time to complete bus set-up before the active bus master internally recognizes and responds to the Acknowledge signal.

During a write operation, the Slave device can drive the Acknowledge line Low any time after the leading (falling) edge of the Address strobe provided it can accept the data from the active bus master within the specified time and the active bus master meets the optimized write timing requirements. Overlapping bus set-up time with the Primary Master's internal overhead improves the data transfer performance on the iLBX bus.

The Slave device should provide a means for varying its Acknowledge response time for read operations. The variable Acknowledge timing allows optimizing the data-transfer timing. A Slave device that does not provide a means for adjusting the Acknowledge timing can not drive the Acknowledge line Low before it drives the data lines.

The Slave device should provide a means for including or excluding the Data Strobe signal state as a prerequisite for driving the Acknowledge line Low to allow for optimized or non-optimized write operation. The exclusion of the Data Strobe going Low as a prerequisite allows the close coupling and overhead overlap to optimize the write data transfer operation. A Slave device that does not provide the choice of including the Data Strobe qualification must wait for the leading edge of the Data Strobe before driving the Acknowledge line Low.

The general Acknowledge signal line implementation specifications are as follows.

- Each Slave device must provide an open collector driver for the Acknowledge line.
- When a master on the iLBX bus cannot meet the optimized timing requirements, the Slave device(s) must be configured for non-optimized operation.
- Both Primary and Secondary Masters should provide a timer to force an Acknowledge locally and avoid a system lock-up if the addressed Slave device fails to respond. The iLBX bus does not provide a means for reporting a slave response failure.

2.4.4 Bus Access Lines

The Primary Master and the Secondary Master use the three bus access lines to transfer bus control between the master devices and to control alternate access to dual ported memory on a Slave device.

2.4.4.1 Lock (LOCK*). The active bus master restricts access through the alternate port to dual port RAM memory on a Slave device by driving the Lock line Low. By locking the alternate access, the active bus master assures that shared data stored in the Slave device is not disturbed until the active bus master completes its use of the data.

The Slave device accepts the Lock signal in conjunction with a data transfer and the Slave device remains locked until Lock signal goes inactive. The Slave device must be selected by the active bus master before the Slave device accepts the state of the Lock signal. To lock a given data transfer cycle to the following data transfer cycle, the active bus master must drive the Lock signal Low before the end of the first data transfer cycle (before the trailing edge of the Data Strobe signal) and keep the Lock signal active until after the start of the last locked data transfer cycle (after the leading edge of the Address Strobe).

Dual ported iLBX Slave devices that recognize both the iLBX bus Lock and the Multibus interface present special design considerations. When alternate access to dual port memory can be locked from either port, the system can become deadlocked through access contention to the dual port memory. By limiting the lock application to the selected Slave device, the possibility of an access contention deadlock can be eliminated. However, the lock limitation allows the possibility of corrupting data if the logical memory in use crosses the physical boundary between two Slave devices. Because the logical memory space crosses the physical boundary, only part of the logical memory space is actually locked at any one time. A Slave device designed to recognize the iLBX bus Lock signal any time Lock is active avoids any risk of corrupting memory but risks system deadlock. A Slave device designed with the type of recognition optional allows the system environment to dictate which form of Lock recognition to use.

The general Lock signal line implementation specifications are as follows.

- Primary Masters and Secondary Masters must provide a tri-state driver for the Lock line. Limited Primary Masters that do not share the iLBX bus with a Secondary Master can use a standard TTL driver with drive characteristics comparable to the specified tri-state driver.
- Only the active bus master drives the Lock line. The inactive master must hold the Lock line driver in the high-impedance state.
- Slave devices with single ported memory (iLBX bus only) and Slave devices with ROM memory need not recognize Lock.
- The selected dual ported slave device must recognize the Lock signal. The selected slave device must exclude alternate access until the Lock signal is driven High.

2.4.4.2 Secondary Master Request (SMRQ*). The Secondary Master requests use of the iLBX bus from the Primary Master by driving the Secondary Master Request line low. Once the Secondary Master has control of the iLBX bus and completes its bus operation, it returns control of the iLBX bus to the Primary Master by driving the Secondary Master Request line High.

The Secondary Master must provide a TTL driver for the Secondary Master Request line and the Primary Master must provide a TTL receiver.

2.4.4.3 Secondary Master Acknowledge (SMACK*). The Primary Master allows use of the iLBX bus by the Secondary Master by driving the Secondary Master Acknowledge line Low after the Secondary Master drives the Secondary Master Request line Low. The Primary Master must continue to drive the Secondary Master Acknowledge line Low until after the Secondary Master drives the Secondary Master Request line High. When making the bus control transfer from the Primary Master to the Secondary Master, the Primary Master first grants bus use to the Secondary Master and then must put all tri-state drivers in the high-impedance state. When making the bus control transfer from the Secondary Master to the Primary Master, the Secondary Master first puts all tri-state drivers in the high-impedance state and then returns bus use to the Primary Master. The Primary Master must provide a TTL driver for the Secondary Master Acknowledge line and the Secondary Master must provide a TTL receiver.

2.5 iLBX™ Bus Pin Assignments

The iLBX-bus configuration uses the form-factor of the standard 60-pin Multibus P2 connector and occupies 56 of the P2 connector pins. Table 4 lists the iLBX bus pin assignments for the 60-pin P2 edge connector. The four Multibus address extension lines (pins 55 through 58 on the P2 connector) retain the standard Multibus interface functions.

2.6 iLBX™ Bus Operation Protocol

The operation protocol for the iLBX bus includes the following three main operations:

- bus control access
- write data to memory
- read data from memory

The iLBX bus operations use asynchronous protocol with positive responses. Thus, specified signal level interactions must occur during an operation for the operation to proceed. Most iLBX bus timing parameters list only a minimum time or a maximum time for efficient use of the asynchronous protocol. The iLBX bus timing specifies a minimum abort time of 1 millisecond, within which a given bus transaction should be completed.

The following sections describe the different operations and the iLBX bus signal lines involved in each type of bus operation. The data-transfer operation descriptions cover the differences between 8-bit and 16-bit data transfers.

2.6.1 Bus Access

The iLBX bus uses a request and acknowledgement process to pass control between the two masters. A maximum of two masters can share the iLBX bus access to memory.

The specified maximum of two masters (one Primary Master and one Secondary Master) reduces bus arbitration to a simple request and acknowledgement process. Bus control arbitration occurs asynchronously to the data transfers.

The Primary Master controls bus access on the iLBX bus. A Primary Master must monitor the Secondary Master Request line and drive the Secondary Master Acknowledge line. The Secondary Master must drive the Secondary Master Request line and monitor the Secondary Master Acknowledge line. An iLBX bus master device designed to operate as either a Primary or a Secondary Master must be exclusively configured as one or the other when implementing the iLBX bus.

The Primary Master assumes control of the iLBX bus as the default configuration for iLBX bus control. The Secondary Master must drive the Secondary Master Request line High and the Primary Master must drive the Secondary Master Acknowledge line High during system initialization to set the default control configuration.

Following initialization, the Secondary Master requests control of the iLBX bus by initiating the transfer process. The bus access timing is illustrated in Figure 2. The Secondary Master drives the Secondary Master Request line Low indicating the need to control the bus. The Secondary Master can drive the request line Low anytime. The time to surrender control of the iLBX bus depends on the design implemented for the Primary Master. The time the Primary Master can retain control of the iLBX bus is not specified. Typically, the Primary Master releases control of the iLBX bus immediately if a data transfer is not in progress. When the Primary Master is actively transferring data, it retains control of the iLBX bus until completing the data transfer(s). The Primary Master relinquishes control of the iLBX bus by driving the Secondary Master Acknowledge line low. The Primary Master is allowed a maximum of 35 ns after driving the Acknowledge line Low to put its tri-state drivers in the high-impedance state. The Secondary Master must wait a minimum of 35 ns after receipt of the Acknowledge signal before enabling its tri-state drivers out of the high-impedance state.

The Secondary Master retains iLBX bus control until it completes the data transfer operation(s). The Secondary Master retains control by continuing to drive the Secondary Master Request line low. The time the Secondary Master can retain control of the iLBX bus is not specified. This allows the Secondary Master the option of making a series of data transfers without returning control of the bus to the Primary Master; however, the Secondary Master typically surrenders control of the iLBX bus after completing the data transfer(s). The Secondary Master surrenders control of the iLBX bus by driving the Secondary Master Request line High. The Secondary Master must put its tri-state drivers in the high-impedance

Table 4 iLBX™ Bus Pin Assignments, P2 Edge Connector

COMPONENT SIDE			SOLDER SIDE		
PIN	**SIGNAL**	**SIGNAL NAME**	**PIN**	**SIGNAL**	**SIGNAL NAME**
1	DB0	DATA LINE 0	2	DB1	DATA LINE 1
3	DB2	DATA LINE 2	4	DB3	DATA LINE 3
5	DB4	DATA LINE 4	6	DB5	DATA LINE 5
7	DB6	DATA LINE 6	8	DB7	DATA LINE 7
9	GND	GROUND	10	DB8	DATA LINE 8
11	DB9	DATA LINE 9	12	DB10	DATA LINE 10
13	DB11	DATA LINE 11	14	DB12	DATA LINE 12
15	DB13	DATA LINE 13	16	DB14	DATA LINE 14
17	DB15	DATA LINE 15	18	GND	GROUND
19	AB0	ADDRESS LINE 0	20	AB1	ADDRESS LINE 1
21	AB2	ADDRESS LINE 2	22	AB3	ADDRESS LINE 3
23	AB4	ADDRESS LINE 4	24	AB5	ADDRESS LINE 5
25	AB6	ADDRESS LINE 6	26	AB7	ADDRESS LINE 7
27	GND	GROUND	28	AB8	ADDRESS LINE 8
29	AB9	ADDRESS LINE 9	30	AB10	ADDRESS LINE 10
31	AB11	ADDRESS LINE 11	32	AB12	ADDRESS LINE 12
33	AB13	ADDRESS LINE 13	34	AB14	ADDRESS LINE 14
35	AB15	ADDRESS LINE 15	36	GND	GROUND
37	AB16	ADDRESS LINE 16	38	AB17	ADDRESS LINE 17
39	AB18	ADDRESS LINE 18	40	AB19	ADDRESS LINE 19
41	AB20	ADDRESS LINE 20	42	AB21	ADDRESS LINE 21
43	AB22	ADDRESS LINE 22	44	AB23	ADDRESS LINE 23
45	GND	GROUND	46	ACK*	SLAVE ACKNOWLEDGE
47	BHEN	BYTE HIGH ENABLE	48	R/$\overline{W}$	READ NOT WRITE
49	ASTB*	ADDRESS STROBE	50	DSTB*	DATA STROBE
51	SMRQ*	SECONDARY MASTER REQUEST	52	SMACK*	SECONDARY MASTER ACKNOWLEDGE
53	LOCK*	ACCESS LOCK	54	GND	GROUND
55	ADR22*	MULTIBUS ADDRESS EXTENSION LINE 22	56	ADR23*	MULTIBUS ADDRESS EXTENSION LINE 23
57	ADR20*	MULTIBUS ADDRESS EXTENSION LINE 20	58	ADR21*	MULTIBUS ADDRESS EXTENSION LINE 21
59		RESERVED	60	TPAR*	TRANSFER PARITY

state before driving the request line High. The Primary Master can enable its tri-state drivers out of the high-impedance state when the request signal goes High. Concurrently, the Primary Master should drive the Secondary Master Acknowledge line High. After releasing control of the iLBX bus, the Secondary 'Master must detect the Secondary Master Acknowledge going High before it can again drive the Secondary Master Request line Low.

2.6.2 Data Transfer Operations

The data transfer operations all take place between the active bus master and the selected slave device. Because bus control-access occurs asynchronously to data transfers, it is not mentioned in the data transfer operation descriptions. Both types of data transfer operations, write and read, are similar with the main difference being the device that places the data on the iLBX bus data lines. Both write and read data transfers allow for both optimized and non-optimized operation. The optimized operation imposes additional timing requirements and considerations.

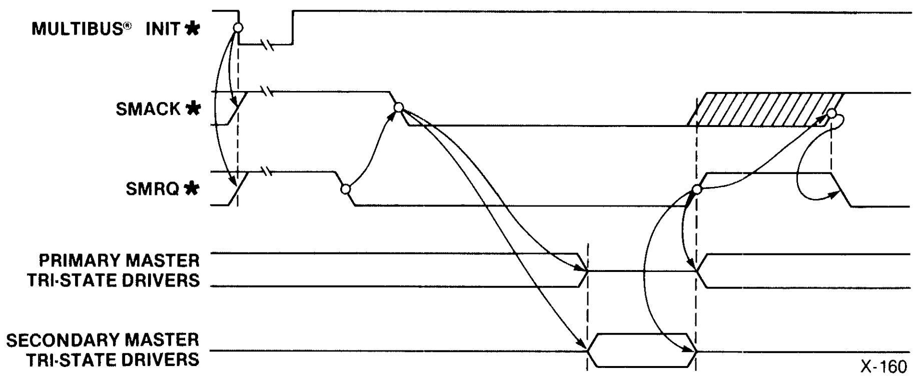

MULTIBUS® INIT*
SMACK*
SMRQ*
PRIMARY MASTER TRI-STATE DRIVERS
SECONDARY MASTER TRI-STATE DRIVERS
X-160

Figure 2 iLBX™ Bus Granting Timing Relationship

2.6.2.1 Write Data-To-Memory. The description of the write data-to-memory operation assumes full completion of any previous data transfer operation before the start of the write operation. The description also assumes the same master device is making another data transfer immediately following the operation described. The subsequent operation could be either read or write and is included here to show the operation to operation timing relationships. The active bus master could be either the Primary Master or the Secondary Master with the same resulting operation. Figure 3 illustrates the optimized 16-bit write data-to-memory timing relationships. The non-optimized write data-to-memory timing uses fixed signal sequences, described in the text, to assure a valid data transfer.

The active bus master initiates the write data-to-memory operation by placing the memory address on the address lines and a control configuration on the control lines. The active bus master must drive the various lines for the specified minimum set-up times before driving the Address Strobe line Low. The selected slave device stores the address information, including the data element selection information, when it detects the leading (falling) edge of the Address Strobe.

2.6.2.1.1 Optimized Operation. All devices attached to the iLBX bus must meet the additional timing requirements for optimized operation to implement optimized operation. The optimized write data-to-memory operation depends the master(s) meeting the specific maximum timing requirements from the leading edge of the Address Strobe signal. The active bus master must provide valid data a maximum of 80 ns from the leading edge of the Address Strobe signal and it must drive the Data Strobe line Low a maximum of 95 ns after the leading edge of the Address Strobe to meet the optimized operation timing requirements.

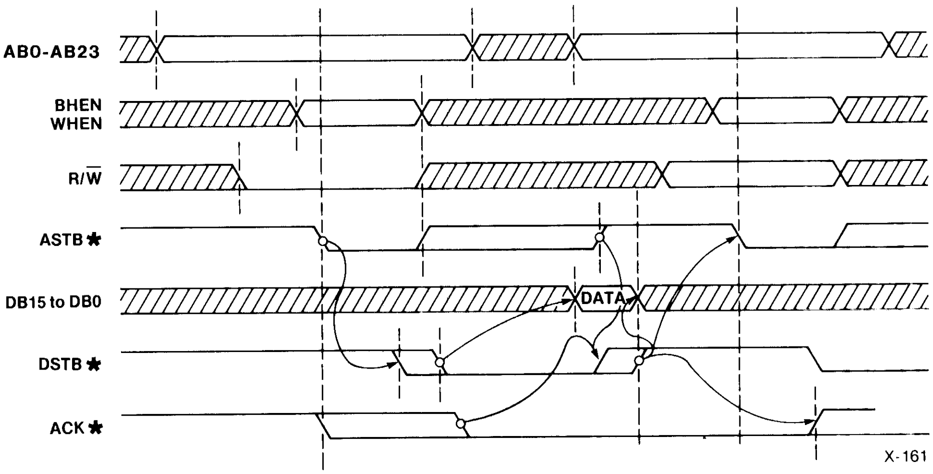


Figure 3 Write Data-To-Memory, 16-Bit Transfer Timing Relationship

Because the optimized operation timing specifies a data valid time relative to the leading edge of the Address Strobe, the Slave device can drive the Acknowledge line Low anytime after the leading edge of the Address Strobe, provided the Slave device can accept the data from the active bus master within the next 80 ns.

The master device must wait for the Acknowledge before completing the data transfer. The time coupling between the slave device and the master requires the master wait until the selected slave drives the Acknowledge line Low, and then the master must wait an additional 80 ns before driving the Data Strobe line High to complete the data transfer. If the Acknowledge is not received within 1 millisecond, the master can abort the operation by driving the Data Strobe line high.

2.6.2.1.2 Non-Optimized Operation. The non-optimized write data-to-memory operation is used when the master(s) can not meet the specific maximum timing requirements from the leading edge of the Address Strobe signal. For the non-optimized write data-to-memory operation, the leading (falling) edge of the Data Strobe becomes the critical timing element. The active bus master must now delay driving the Data Strobe line Low until it can provide valid data within 45 ns after the leading edge of the Data Strobe. The selected Slave device is restricted from pre-acknowledging the Data Strobe, and the Slave device must wait until the leading edge of the Data Strobe before driving the Acknowledge line, ACK*, Low. It can drive the Acknowledge line Low anytime after the leading edge of the Data Strobe provided it can accept the data from the active bus master within the next 80 ns.

The master device must wait for the Acknowledge before completing the data transfer. The time coupling between the slave device and the master requires the master wait until the selected slave drives the Acknowledge line Low, and then the master must wait an additional 80 ns before driving the Data Strobe line High to complete the data transfer. If the Acknowledge is not received within 1 millisecond, the master can abort the operation by driving the Data Strobe line high.

2.6.2.1.3 Operation Completion. The active master device completes the write data-to-memory operation by driving the Data Strobe line High.

When the write data-to-memory transfer is a single event transfer, the active master stops driving the data and address lines. The selected Slave stops driving the Acknowledge line and internally goes to the deselected state.

For sequential cycles of write data-to-memory operation, the active bus master can start address set-up for the next data transfer before driving the Data Strobe line High. The amount of data transfer overlap is limited by the minimum time of 25 ns from the rising edge of the Data Strobe to the falling edge of the next Address Strobe.

The overlap provides a potential cycle time of 105 ns per data transfer or a transfer rate of 9.5 MHz. Assuming 16-bit transfers, the resulting throughput is approximately 19 megabytes per second.

2.6.2.2 Read Data-From-Memory. The description of the read data-from-memory operation assumes full completion of any previous data transfer operation before the start of the read operation. The description also assumes successive data transfers by a single master device. The active bus master could be either the Primary Master or the Secondary Master with the same resulting operation. The read data-from-memory operation uses roughly the same timing relationships as the write data-to-memory. The level of the Read-Not-Write signal and the device driving the data lines constitute the main differences between the two operations.

The description of the read data-from-memory operation includes the special considerations for optimizing the data transfer rate between the active bus master and the selected slave. Because the active bus master retains control of the data lines in the write data-to-memory operation, the bus master is the key device in determining the level of data transfer performance. However, during the read data-from-memory operation, the active bus master surrenders control of the data lines to the Slave device for part of the operation. Thus, the level the cooperation between the active bus master and the selected Slave is the key element in determining the level of data transfer performance. Figure 4 illustrates the optimized 16-bit read data-from-memory timing relationships. The non-optimized read data-from-memory timing uses fixed signal sequences, described in the text, to assure a valid data transfer.

The active bus master initiates the read data-from-memory operation by placing the memory address on the address lines and a control configuration on the control lines. The active bus master must drive the various lines for the specified minimum set-up times before driving the Address Strobe line Low. The selected slave device stores the address information, including the data element selection information, when it detects the leading (falling) edge of the Address Strobe.

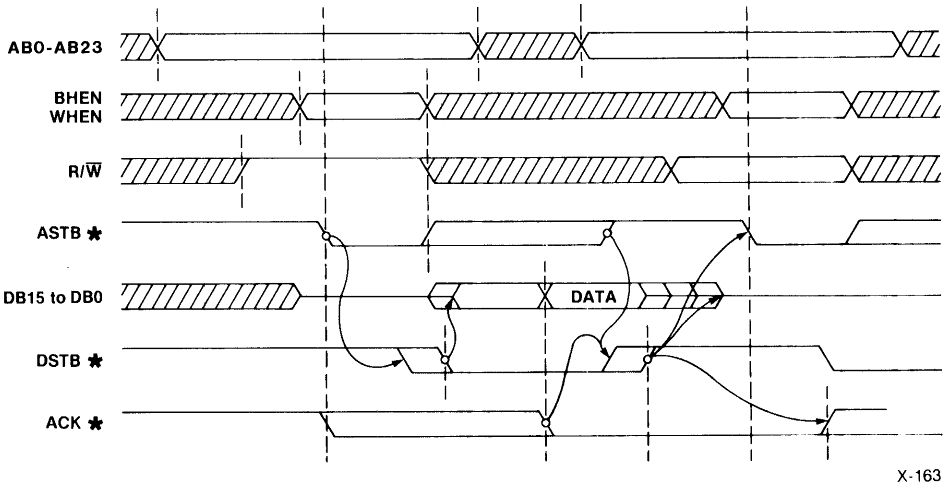


Figure 4 Read Data-From-Memory, 16-Bit Transfer Timing

The selected slave then drives the Acknowledge line, ACK*, Low anytime after the leading edge of the Address Strobe signal. When the selected slave device drives the Acknowledge line Low, it must meet the timing coupling requirements of the active master device and present valid data on the data lines before the active master device samples the data lines.

2.6.2.2.1 Non-Optimized Operation. The non-optimized read data-from-memory operation is used when a Slave device does not have variable Acknowledge to read data valid timing. For the non-optimized read data-from-memory operation, the selected Slave device must have data valid before driving the Acknowledge line Low. Because the leading edge of Data Strobe signals the selected Slave device that it can start driving the data lines, in non-optimized mode the selected Slave device must also wait for the Data Strobe before driving the Acknowledge line low.

A minimum of 80 ns after detecting the Acknowledge line Low, the active bus master completes the non-optimized read data-from-memory operation by driving the Data Strobe line High. If the Acknowledge is not received within 1 millisecond, the master can abort the operation by driving the Data Strobe line high.

2.6.2.2.2 Optimized Operation. The optimized read data-from-memory operation uses the signal overlap prohibited in the non-optimized operation. A Slave device designed for optimized operation must provide an adjustment for the length of time between when it drives the Acknowledge line Low and when it presents valid data on the data lines. The Primary Master determines the maximum allowed amount of signal overlap based on its acknowledge acceptance overhead time. The Slave device Acknowledge timing is then set to an overlap value equal to or less than the Primary Master's acknowledge acceptance overhead time. For any given Slave device, the length of time Acknowledge can precede valid data ranges from coincident (data valid at the same time the Slave device drives the Acknowledge line Low) to the maximum internal memory access time of the Slave device (immediate upon detecting the leading edge of the Address Strobe signal). Regardless of the allowed signal overlap with the optimized operation, the selected slave device must wait until the leading edge of Data Strobe before driving the data lines.

A minimum of 80 ns after detecting the Acknowledge line Low, the active bus master completes the optimized read data-from-memory operation by driving the Data Strobe line High. If the Acknowledge is not received within 1 millisecond, the master can abort the operation by driving the Data Strobe line high.

2.6.2.2.3 Operation Completion. The active bus master completes the read data-from-memory operation by driving the Data Strobe line High. When the read data-from-memory transfer is a single event transfer, the selected slave stops driving the data lines and the Acknowledge line and internally goes to the deselected state.

For sequential cycles of read data-from-memory, the active bus master can start address set-up for the next data transfer before driving the Data Strobe line High. The the amount of data transfer overlap is limited by the minimum time of 25 ns from the rising edge of the data strobe to the falling edge of the next Address Strobe. The signal overlapping provides a total potential cycle time of 105 ns per data transfer or a transfer rate of 9.5 MHz. Assuming 16-bit data transfers, the resulting throughput is approximately 19 megabytes per second.

3. Electrical Specification

3.1 Introduction

This section defines the electrical requirements of the iLBX bus. The descriptions include the types of drivers and receivers required, the method, type, and location of line termination, general signal characteristics, and electrical timing.

3.2 Electrical State Relationships

The electrical state relationships used in this manual conform to the conventions used in the Multibus Specification. The iLBX bus uses commercial grade TTL components for all drivers and receivers. Table 5 relates the general industry voltage level standards for TTL components to the signal line notation conventions used in this manual. The specifications in Table 5 assume a power source of +5 Vdc, +5 percent, referenced to logic ground. The iLBX bus does not include provision for system power and the electrical specification assumes that all power is drawn from the Multibus P1 connector power lines.

Table 5 Notational Summary

TTL ELECTRICAL LEVEL	TTL VOLTAGE RANGE	
	AT RECEIVER	AT DRIVER
H, High	+2.0 TO +5.25 Vdc	+2.4 TO +5.25 Vdc
L, Low	−0.5 TO +0.8 Vdc	0 TO +0.5 Vdc

3.3 Environmental Requirements

The electrical specifications for the iLBX bus must be met under the following environmental conditions. The specifications list the ambient temperature requirements and the non-condensing requirements for humidity.

OPERATING	
Temperature	0 to 55 degrees C
Relative humidity	0 to 85 percent

3.4 DC Specifications

Table 6 lists the iLBX bus DC specifications for the signal line drivers and the receiver loads presented to the signal lines. The DC specifications listed assume the use of devices typically associated with the standard 16-bit implementation of the iLBX bus. Refer to Section 6, Levels of Compliance, for the driver types required with the various allowed subsets of the iLBX bus. The drive and load requirements presented in Table 6 apply regardless of the iLBX bus subset implemented.

The drive requirements include the load capacitance an output driver must drive and for tristate drivers the requirements assume four slave device loads. The load requirements include the maximum allowable input capacitance that any device can present to the signal line. The specifications assume the High drive signals are measured at +2.4 Vdc and the Low drive signals at +0.5 Vdc.

3.5 Termination

DC and AC termination requirements are listed in Table 6. The DC termination for a particular line consists of a resistor connecting the line to +5VDC. The location of each resistor depends on the applicable signal line. Signal lines driven by a tristate or open collector driver (that is, DB15 - DB0, TPAR*, ASTB*, DSTB*, and ACK*. has the termination resistors located at the Primary Master. If a Primary Master is not implemented on the iLBX bus a Secondary Master, operating as a limited Primary Master, must have the capability to provide the resistors. For the signal line, SMRQ*, the resistor is located at the Primary Master, and never at the Secondary Master.

An additional AC termination is required for lines ASTB* and DSTB* and is located at each slave device. Each slave device must provide a series RC network connecting the signal line to logic ground. The network should be placed in close proximity to the receiver component on the slave device.

3.6 AC Specifications

Table 7 lists the iLBX bus timing parameters for the signal lines. The table provides a reference designator for each timing parameter, a description of the timing parameter, the minimum and maximum timing requirements, and the source device where the timing parameter must be implemented. Figures 5 and 6 are timing charts that illustrate the timing relationships for the iLBX bus and the timing specifications on the timing charts use the reference designators from Table 7. Table 7 does not specify the typical transition rise and fall times for the iLBX bus drivers, however, the bus drivers should have slew rates less than 1 volt/nanosecond with less than 24 milliamps drive. Use of drivers having higher slew rates may degrade signal characteristics to an undesirable waveform. Figure 7 depicts a typical acceptable signal waveform during high-to-low and low-to-high transitions.

Table 6 DC Specifications

Signal Name	Driver Type	DC Termination (To +5 VDC)	Minimum Driver Requirements			Maximum Receiver Requirements		
			High	Low	Load Cap	High	Low	Load Cap
DB15-0	TRI-STATE	10K OHMS	0.6 ma	9 ma	75 pf	0.15 ma	2 ma	18 pf
TPAR*	TRI-STATE	10K OHMS	0.6 ma	9 ma	75 pf	0.15 ma	2 ma	18 pf
ABD23-0	TRI-STATE	NONE	0.4 ma	20 ma	120 pf	0.10 ma	5 ma	30 pf
R/$\overline{W}$	TRI-STATE	NONE	0.2 ma	8 ma	75 pf	0.05 ma	2 ma	18 pf
BHEN	TRI-STATE	NONE	0.2 ma	8 ma	75 pf	0.05 ma	2 ma	18 pf
LOCK*	TRI-STATE	NONE	0.2 ma	8 ma	75 pf	0.05 ma	2 ma	18 pf
SMRQ*	TTL	10K OHMS	0.05 ma	2 ma	20 pf	0.05 ma	2 ma	18 pf
SMACK*	TTL	NONE	0.05 ma	2 ma	20 pf	0.05 ma	2 ma	18 pf
ASTB*	TRI-STATE	10K OHMS ¶	0.2 ma	9 ma	75 pf	0.05 ma	2 ma	18 pf
DSTB*	TRI-STATE	10K OHMS ¶	0.2 ma	9 ma	75 pf	0.05 ma	2 ma	18 pf
ACK*	OPEN COLL	330 OHMS	N.A.	20 ma	45 pf	0.05 ma	2 ma	18 pf

Note: ¶ = Additional AC termination for both ASTB* and DSTB* lines are required on each slave device. Each terminator is a series RC (100 ohm, 10 picofarad) network between the signal line and ground. The location of the termination network should be as close as possible to the receiver component input.

The following are the general and specific notes for Table 7.

General Notes:

- All times listed are nanoseconds unless otherwise noted
- TPAR* timing is the same as DB15 - DB0
- M refers to the current active bus master
- S refers to the currently selected slave device
- PM refers to the Primary Master
- SM refers to the Secondary Master

Specific Notes:

1. Board designs can implement either of two transfer rates, optimized and non-optimized, based on the degree of close coupling desired between the master and slave devices. Two factors determine the coupling and the degree of optimization realized when implementing the iLBX bus: the acknowledge acceptance time of the master device and the range of variability in the slave device to pre-acknowledge the data transfer. A master device designed for optimized operation must meet the t17 maximum time for write operation and the t11 maximum time for read operations. When the master devices meet the required times, the Slave device is allowed to drive the Acknowledge line Low any time after the leading edge of the Address Strobe. A master device that does not meet the maximum write time requirements, by default, transfers data using non-optimized timing, and the Slave device must wait for the leading edge of the Data Strobe before driving the Acknowledge line Low. See Note 3 for the Slave device timing restrictions.

Table 7 iLBX^{TM} Bus Timing Parameters

REF	PARAMETER DESCRIPTION	TIMING		SOURCE	NOTE
		MIN	MAX		
t1	ASTB* DURATION (WIDTH)	25		M	
t2	ADDRESS SETUP TO LEADING EDGE OF ASTB*	40		M	
t3	ADDRESS HOLD AFTER LEADING EDGE OF ASTB*	40		M	
t4	BHEN SETUP TO LEADING EDGE OF ASTB*	30		M	
t5	BHEN HOLD AFTER LEADING EDGE OF ASTB*	30		M	
t6	R/$\overline{W}$ SETUP TO LEADING EDGE OF ASTB*	20		M	
t7	R/$\overline{W}$ HOLD AFTER LEADING EDGE OF ASTB*	25		M	
t8	TRAILING EDGE OF ASTB* TO TRAILING EDGE OF DSTB*	10		M	
t9	TRAILING EDGE OF DSTB* TO LEADING EDGE OF ASTB*	25		M	
t10	DSTB* DURATION (WIDTH)	50		M	
t11	LEADING EDGE OF ASTB* TO LEADING EDGE OF DSTB*	0	95	M	1
t12	ACK* HOLD AFTER TRAILING EDGE OF DSTB*	0	45	S	2
t13	LEADING EDGE OF ACK* TO READ DATA VALID	0	tacc	S	3
t14	READ DATA HOLD TIME AFTER TRAILING EDGE OF DSTB*	0	45	S	
t15	LEADING EDGE OF ACK* TO TRAILING EDGE OF DSTB*	80		M	
t16	LEADING EDGE OF DSTB* TO READ DATA VALID	0		S	
t17	LEADING EDGE OF ASTB* TO WRITE DATA VALID		80	M	1
t18	LEADING EDGE OF DSTB* TO WRITE DATA VALID		45	M	
t19	WRITE DATA HOLD TIME AFTER TRAILING EDGE OF DSTB*	20		M	
t20	LEADING EDGE OF ASTB* TO FIRST SAMPLE OF ACK* LINE	45 to t9		M	6
t21	LOCK* SETUP TO TRAILING EDGE OF DSTB*	15		M	
t22	LOCK* HOLD AFTER TRAILING EDGE OF DSTB*	15		M	
t23	SMACK* LOW TO TRI-STATE DRIVERS IN HIGH IMPEDANCE STATE		35	PM	
t24	SMACK* LOW TO TRI-STATE DRIVERS OUT OF HIGH IMPEDANCE STATE	35		SM	
t25	SMRQ* HIGH TO TRI-STATE DRIVERS IN HIGH IMPEDANCE STATE		0	SM	
t26	SMRQ* HIGH TO TRI-STATE DRIVERS OUT OF HIGH IMPEDANCE STATE	0		PM	5
t27	SMRQ* HIGH TO SMACK* HIGH	0		PM	
t28	SMRQ* LOW TO SMACK* LOW	0		PM	
t29	SMACK* HIGH TO SMRQ* LOW	0		SM	
t30	LEADING EDGE OF ASTB* TO TRAILING EDGE OF DSTB* (ABORT)	1ms		M	7
t31	WRITE DATA ACTIVE AFTER TRAILING EDGE OF DSTB*	45		M	4

2. The selected slave device must stop driving the Acknowledge line Low immediately upon detection of the trailing edge of the Data Strobe. The 45 ns maximum hold over time listed for the Acknowledge signal allows for the assumed input-to-output delay for the Acknowledge driver of 15 ns and the typical pull-up charge time through a 330 ohm resistor required to bring the Acknowledge signal from 0.2 Vdc to 2.4 Vdc assuming a worst case capacitive load of 45 pf.

3. The slave device should be provided with variable (typically discrete) timing capabilities for driving the Acknowledge line Low. For write operations, the slave device can drive the Acknowledge line Low anytime after the leading edge of the Address Strobe signal subject to the limitations listed in Note 1. For read operations, the slave device can pre-acknowledge the data transfer by driving the Acknowledge line Low before it provides valid data on the data lines. Pre-acknowledgement is subject to the limitations listed in Note 1. The amount of variability provided should range from 0.0 ns (data valid when the slave drives the Acknowledge line Low) to the maximum access time of the slave's memory resources (t_{acc}). If the board designer chooses not to provide variable timing, the slave device must have data valid at the time it drives the Acknowledge line Low.

4. The minimum t31 guarantees that a master does not start to drive the data bus (write cycle) until the slave has stopped driving the data bus (previous read cycle).

5. The t26 timing does not apply during system initialization (for example, when the Primary Master receives the Multibus interface initialization).

6. The t9 time used for computing t20 is the actual t9 time of the master. The t20 time can range from 0 to 20 nanoseconds.

7. The minimum operation abort time is 1 millisecond.

4. Mechanical Specification

4.1 Introduction

This section defines the physical and mechanical requirements that must be considered when designing iLBX bus compatible printed circuit boards or when implementing the iLBX bus in a system. The descriptions include the form factor requirements specific to the iLBX bus, the method, type, and location of connectors, and connector keying. Implementation of the iLBX bus on Multibus-compatible printed circuit boards is assumed. The iLBX bus Mechanical Specifications generally are limited to those specifications different from or in addition to the Multibus interface mechanical specifications.

4.2 iLBX™ Bus Form Factor

Because of cable length restrictions, any board implementing the iLBX bus typically is installed in a Multibus chassis. A partial interface of the iLBX bus compatible device to the Multibus P1 connector is assumed because the iLBX bus does not provide any power connections, initialization signals, or interrupt capabilities. The Multibus specification requirements for board-to-board spacing, board thickness, component lead length, and component height above the board remain the same as in the Multibus specification. Refer to the INTEL MULTIBUS SPECIFICATION for details on the general Multibus interface mechanical specifications.

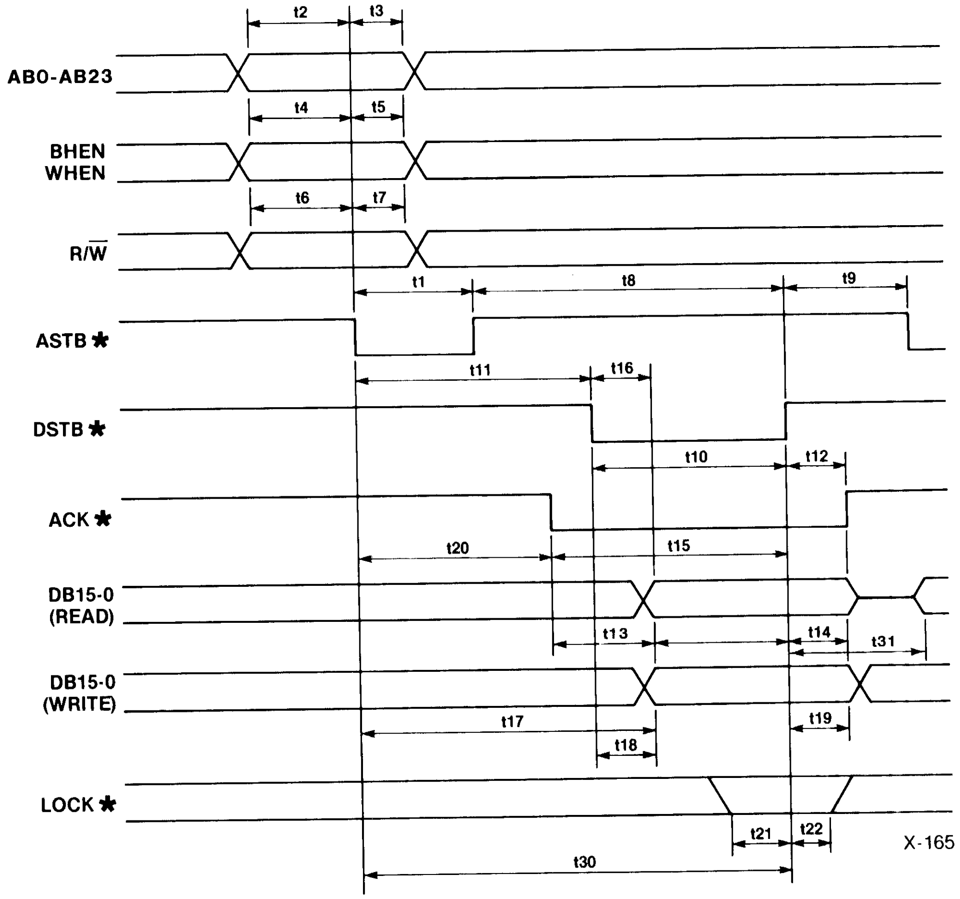


Figure 5 iLBX^{TM} Bus Data Transfer Timing Chart

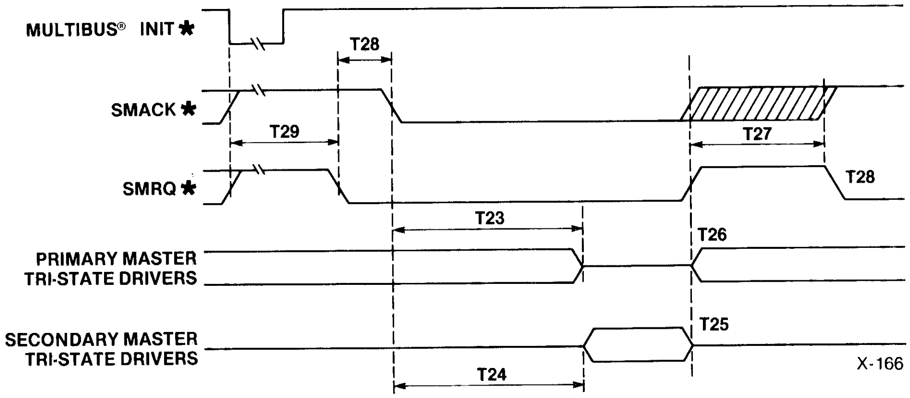
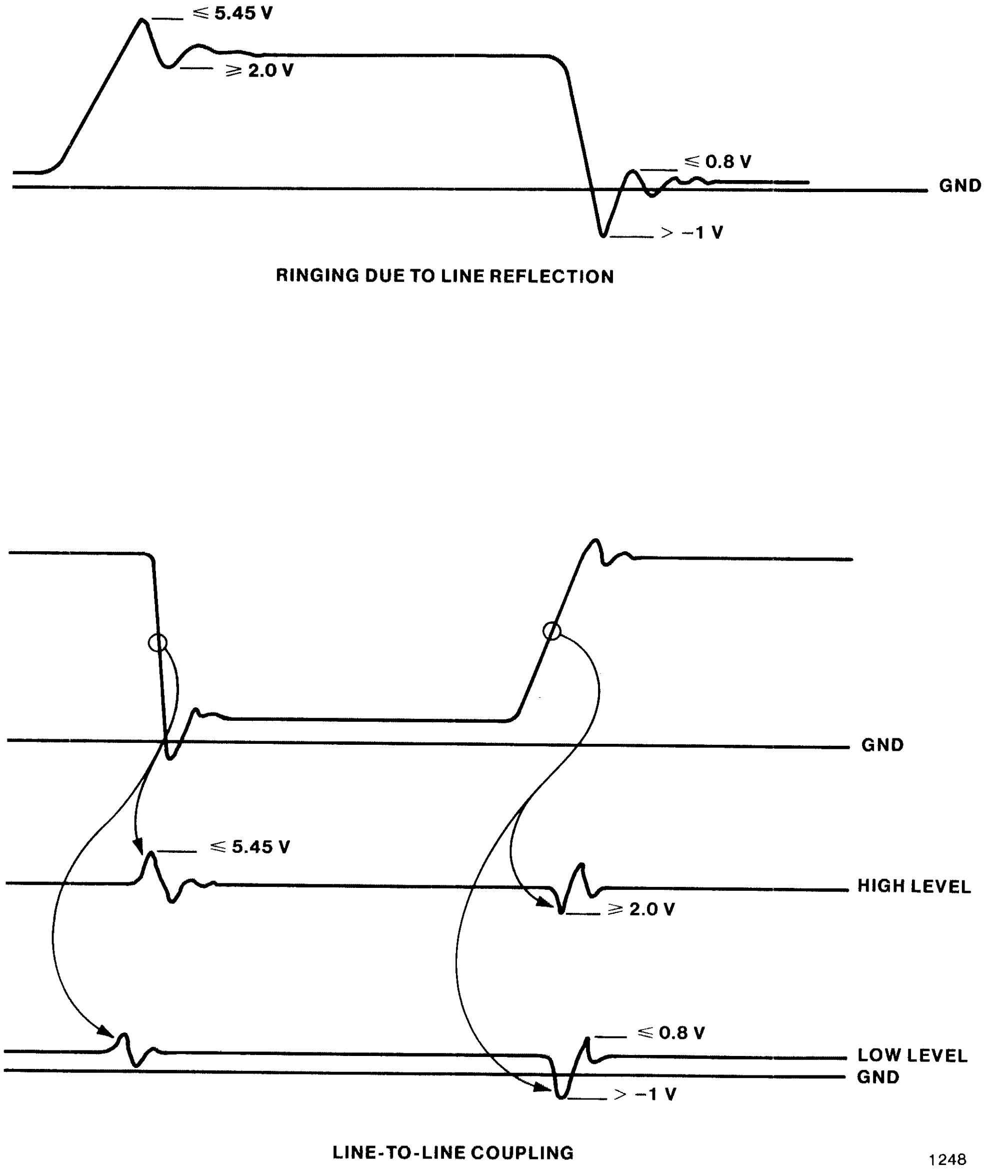


Figure 6 iLBX^{TM} Bus Control Transfer Timing Chart



RINGING DUE TO LINE REFLECTION

LINE-TO-LINE COUPLING

Figure 7 Typical Waveform

4.2.1 Connector Locations and Board Outline

The Multibus interface P1 connector is unchanged for implementation of the iLBX bus. The iLBX bus resides on the Multibus form factor P2 connector and supercedes the Multibus interface definitions for the P2 signals. The 8- and 16-bit iLBX bus implementations use the same, 0.1 inch centers, 60-pin P2 connector as the standard Multibus interface. Figure 8 illustrates the standard printed circuit board outline for the Multibus interface modified to accommodate the iLBX bus.

Figure 8 is limited to the basic form factor information for the iLBX bus implemented on a Multibus board. Refer to the Multibus Specification for additional information on the Multibus board form factor.

4.2.2 Pin Numbering Convention

The iLBX bus specification uses the same connector pin number convention as the Multibus specification. The numbering convention specifies locating pin 1 on the component side of the board so that it is at the left end of the connector when you face the connector with the component side of the board up. Pin 2 is located immediately under pin one on the solder side of the board. The pins are then numbered in ascending order from left to right with the odd numbered pins located on the component side of the board and the even numbered pins located on the solder side of the board. Figure 9 illustrates the iLBX bus P2 connector pin numbering convention.

4.2.3 Component Layout Considerations

To maintain the electrical signal quality of the iLBX bus signals, care must be taken when the iLBX bus drivers, receivers, and transceivers are positioned, relative to the P2 connector, on a board implementing the iLBX bus. Any device, driver or receiver, directly connected to an iLBX bus signal line should be located close to the P2 connector. The printed circuit board trace connecting a driver or receiver pin to the corresponding P2 connector pin should not exceed 5 cm (2 in) in length. On a Primary Master, include any additional trace required to connect the terminating resistor to the signal line trace when calculating the maximum trace lengths.

The iLBX bus interface components (integrated circuits containing drivers, receiver, or transceivers directly connected to an iLBX bus signal line) must have adequate connection to signal ground.

On boards with a ground interlayer, the interlayer should be solid under the iLBX bus interface components with the ground connections made directly to the interlayer. On boards without a ground interlayer, the ground trace to the iLBX bus interface components should be 1.27 mm (0.05 in) wide (assuming 1 oz copper plate) and be directly connected to the main signal ground for the board.

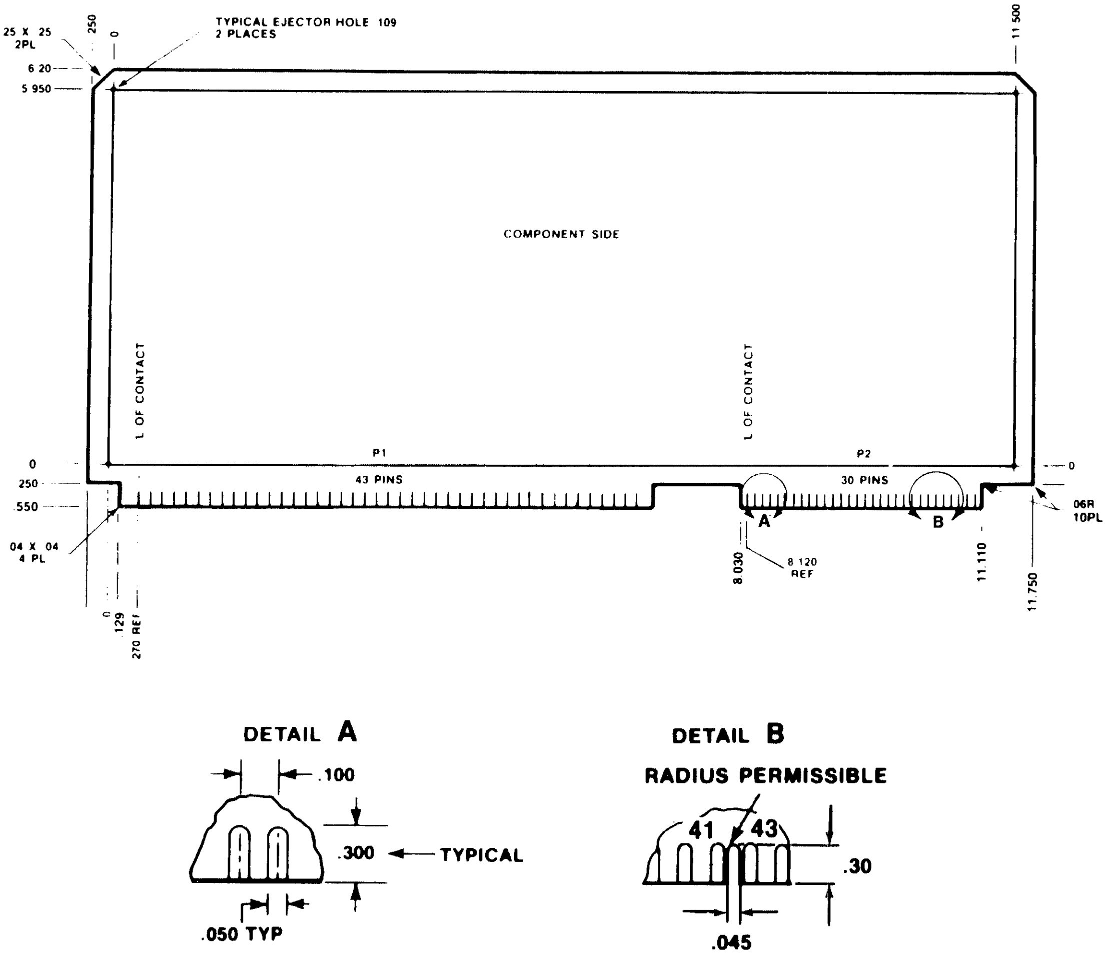
4.2.4 iLBX™ Bus Pin Assignments

The iLBX-bus configuration uses the form-factor of the standard 60-pin Multibus P2 connector and occupies 56 of the P2 connector pins.

Table 8 lists the iLBX bus pin assignments for the 60-pin P2 edge connector. The four Multibus address extension lines (pins 55 through 58 on the Multibus P2 connector) retain the standard Multibus interface functions. Information on designing a P2 layout with iLBX bus or Multibus P2 compatibility (limited to two front panel lines) is located in Section 5.

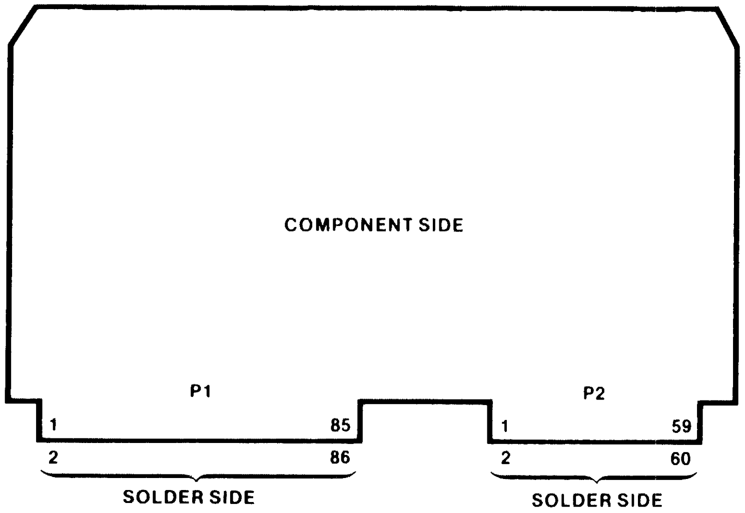
4.2.5 Connector Key Slot

The P2 keyslot for the iLBX bus is located between P2 pins 41 and 43 for 8- and 16-bit compatible boards. Figure 8 includes the routing specifications for the iLBX bus P2 connector key slot. All iLBX bus compatible boards must be key slotted to assure the board is not plugged into a P2 connector with Multibus P2 connector compatible battery back-up signals.



1249

Figure 8 iLBX™ Bus Standard Printed Circuit Board Outline



1250

Figure 9 iLBX™ Bus Connector and Pin Numbering Conventions

4.3 Battery Back-Up and Front Panel Interface

Implementation of the iLBX bus on a printed circuit board supercedes the full interface of the Multibus specification of the battery back-up and front panel interface signals to the P2 connector. Information on designing a P2 layout with iLBX bus or Multibus P2 compatibility (limited to two front panel lines) is located in Section 5. The iLBX bus specification introduces the use of an auxiliary right-angle connector (JX) mounted on top of the board to connect the battery back-up and front panel signals to the board. There are 14 signals assigned to the JX connector divided into two subset groups: the battery back-up signals (pins 1 through 6); and the front panel interface signals (pins 7 through 14). A full implementation of the JX connector can be made or either of the two subsets if board space is limited or when the additional signals are not used on the board. For example, the Slave devices typically do not require use of the front panel interface signals.

Table 8 iLBX™ Bus Pin Assignments, P2 Edge Connector

COMPONENT SIDE			SOLDER SIDE		
PIN	SIGNAL	SIGNAL NAME	PIN	SIGNAL	SIGNAL NAME
1	DB0	DATA LINE 0	2	DB1	DATA LINE 1
3	DB2	DATA LINE 2	4	DB3	DATA LINE 3
5	DB4	DATA LINE 4	6	DB5	DATA LINE 5
7	DB6	DATA LINE 6	8	DB7	DATA LINE 7
9	GND	GROUND	10	DB8	DATA LINE 8
11	DB9	DATA LINE 9	12	DB10	DATA LINE 10
13	DB11	DATA LINE 11	14	DB12	DATA LINE 12
15	DB13	DATA LINE 13	16	DB14	DATA LINE 14
17	DB15	DATA LINE 15	18	GND	GROUND
19	AB0	ADDRESS LINE 0	20	AB1	ADDRESS LINE 1
21	AB2	ADDRESS LINE 2	22	AB3	ADDRESS LINE 3
23	AB4	ADDRESS LINE 4	24	AB5	ADDRESS LINE 5
25	AB6	ADDRESS LINE 6	26	AB7	ADDRESS LINE 7
27	GND	GROUND	28	AB8	ADDRESS LINE 8
29	AB9	ADDRESS LINE 9	30	AB10	ADDRESS LINE 10
31	AB11	ADDRESS LINE 11	32	AB12	ADDRESS LINE 12
33	AB13	ADDRESS LINE 13	34	AB14	ADDRESS LINE 14
35	AB15	ADDRESS LINE 15	36	GND	GROUND
37	AB16	ADDRESS LINE 16	38	AB17	ADDRESS LINE 17
39	AB18	ADDRESS LINE 18	40	AB19	ADDRESS LINE 19
41	AB20	ADDRESS LINE 20	42	AB21	ADDRESS LINE 21
43	AB22	ADDRESS LINE 22	44	AB23	ADDRESS LINE 23
45	GND	GROUND	46	ACK*	SLAVE ACKNOWLEDGE
47	BHEN	BYTE HIGH ENABLE	48	R/$\overline{W}$	READ NOT WRITE
49	ASTB*	ADDRESS STROBE	50	DSTB*	DATA STROBE
51	SMRQ*	SECONDARY MASTER REQUEST	52	SMACK*	SECONDARY MASTER ACKNOWLEDGE
53	LOCK*	ACCESS LOCK	54	GND	GROUND
55	ADR22*	MULTIBUS ADDRESS EXTENSION LINE 22	56	ADR23*	MULTIBUS ADDRESS EXTENSION LINE 23
57	ADR20*	MULTIBUS ADDRESS EXTENSION LINE 20	58	ADR21*	MULTIBUS ADDRESS EXTENSION LINE 21
59		RESERVED	60	TPAR*	TRANSFER PARITY

Figure 10 shows the iLBX bus board outline with the JX location area shaded. The JX connector must be located within the specified area to keep the mating cable lengths to a minimum. Figure 11 illustrates the height, pin spacing, and pin location requirements for the JX connector and Table 9 lists the pin assignments for the JX connector. The signal lines assigned to the JX connector are standard Multibus interface lines, and the descriptions and timing specifications are located in the Multibus Specification.

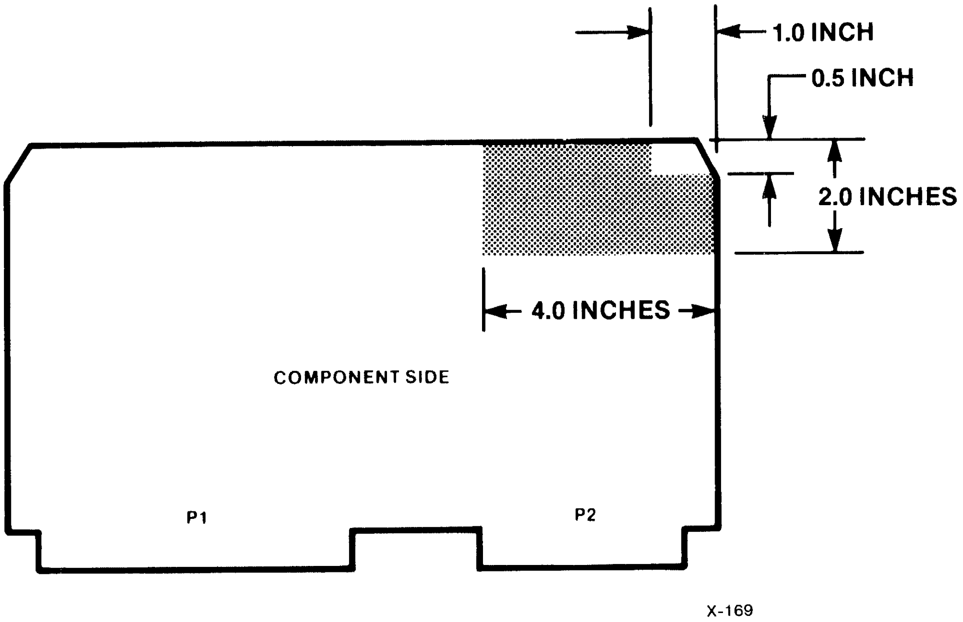


Figure 10 Auxiliary (JX) Connector Location Area

4.4 iLBX™ Bus Connectors and Cabling

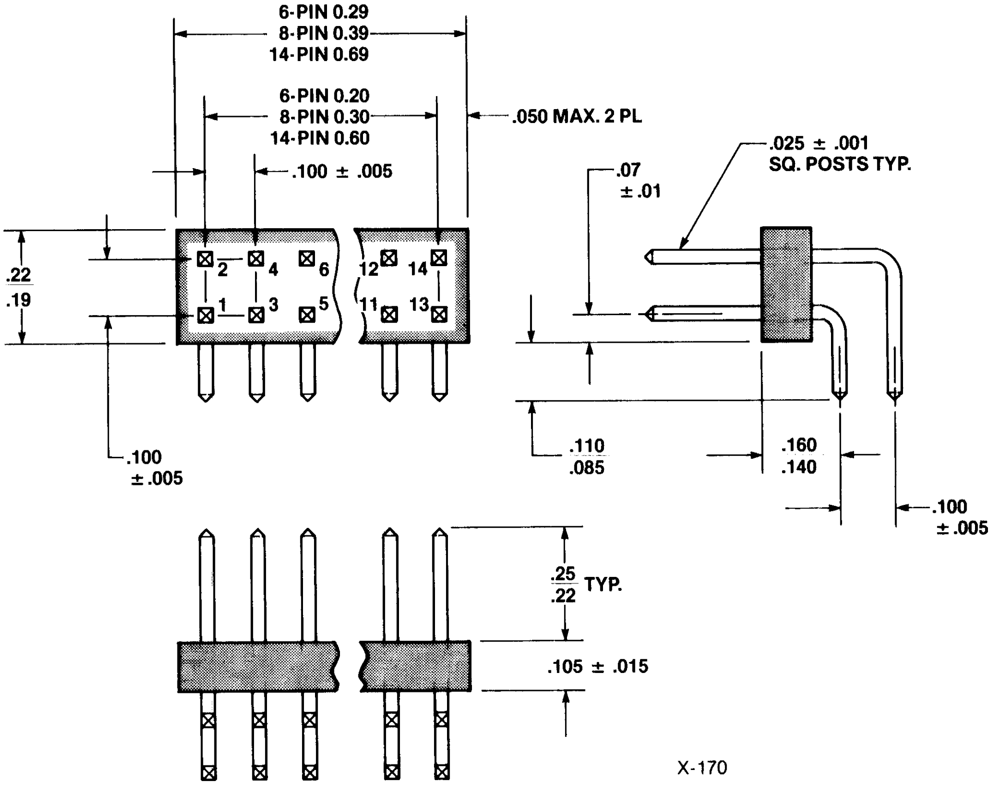
The iLBX bus does not use a rigid backplane to interconnect the iLBX bus compatible boards but rather an interconnect cable assembly. This method of interconnection allows the system designer more freedom because the iLBX bus compatible boards are not required to be in adjacent board slots. The specification further simplifies system implementation by using ribbon cable and mass terminated connectors to make up the required interconnect cables. Table 10 lists suppliers that produce ribbon cable and connectors compatible with the iLBX bus. The table also lists the required installation hardware.

4.4.1 iLBX™ Bus Cable

The iLBX bus interconnect cable uses 28 AWG, 60 conductor, flat ribbon cable for interconnecting 8- and 16-bit compatible boards. The maximum length for the interconnect cable is 10 centimeters (approximately 4 inches) for the interconnect cable. For best system performance and good electrical design practice, the interconnect cable should be kept as short as possible. The following are the general electrical and insulation specifications for iLBX bus compatible cable:

ELECTRICAL PROPERTIES	
Impedance	100 ohms ±10%
Propagation velocity (maximum)	2.0 nx/ft
Capacitance (maximum)	15 pf/ft

INSULATION REQUIREMENTS	
Voltage rating (minimum)	100 Vdc
Insulation resistance (minimum)	$1x10^{10}$ ohms



MATERIALS AND FINISHES

INSULATOR: GLASS FILLED POLYESTER OR EQUIVALENT.

CONTACT: PHOSPHOR BRONZE.

FINISH: .000020 IN. MIN. GOLD OVER .000050 IN. MIN. NICKEL PLATE.

Figure 11 Auxiliary Connector Mechanical Specifications

4.4.2 iLBX™ Bus Connectors

The iLBX bus requires use of 60 pin, insulation displacement type female receptacles to attach the interconnecting cable to 8- and 16-bit compatible boards at the P2 edge connector. The female receptacle must have a key block compatible with the key slot specifications for the iLBX bus P2 connector.

Table 9 Auxiliary (JX) Connector Pin Assignments

LOWER ROW			UPPER ROW		
PIN	MNEMONIC	SIGNAL NAME	PIN	MNEMONIC	SIGNAL NAME
1	+5	+5 VDC BATTERY	2	GND	GROUND
3	+5	+5 VDC BATTERY	4	GND	GROUND
5	MPRO*	MEMORY PROTECT	6	NVE*	NON-VOL. ENABLE
7	ALE	ADDR LATCH ENABLE	8	GND	GROUND
9	ARES*	RESET SWITCH	10	GND	GROUND
11	INT	FRONT PANEL INT	12	RES	RESERVED
13	PFSN*	POWER FAIL SENSE	14	PFIN*	POWER FAIL INT.

Table 10 Cable And Receptacle Vendors

iLBX™ BUS COMPATIBLE CABLE

VENDOR	VENDOR PART NUMBER	CONDUCTORS
T & B Ansley	171-60	60
T & B Ansley	173-60	60
3M	3365/60	60
3M	3306/60	60
Berg	76164-060	60
Belden	9L28060	60
Spectrastrip	455-240-60	60

iLBX™ BUS COMPATIBLE RECEPTACLES

VENDOR	VENDOR PART NUMBER	PINS
KELAM	RF30-2803-5	60
KELAM*	110-10-001-37 (polarizing key)	
T & B Ansley**	A3020 (609-6026 modified)	60

Notes: * = Mounting hardware for KELAM consists of 2 sets of 5/8 inch 4-40 Philips round head screw, 1/8 inch 4-40 spacer, 4-40 internal tooth lock washer, 4-40 hex nut.
** = Mounting Hardware for T & B consists of 2 sets of 0.5 inch 4-40 Philips, Fillister head screw, 4-40 lock washer, and 4-40 hex nut.

4.4.3 iLBX™ Bus Cable Assembly

An iLBX bus cable assembly can have from two to five female receptacle connectors mass terminated to the flat ribbon cable. The spacing between the female receptacles assembled to the cable varies with the Multibus backplane used and any intervening, non-iLBX bus compatible, boards. Refer to the applicable chassis hardware reference manual for information on board spacing. Figure 12 illustrates an iLBX bus interface cable assembly. Figure 13 and 14 illustrate mounting of the insulation displacement type receptacle to the Multibus backplane, for the KELAM and T & B Ansley connectors.

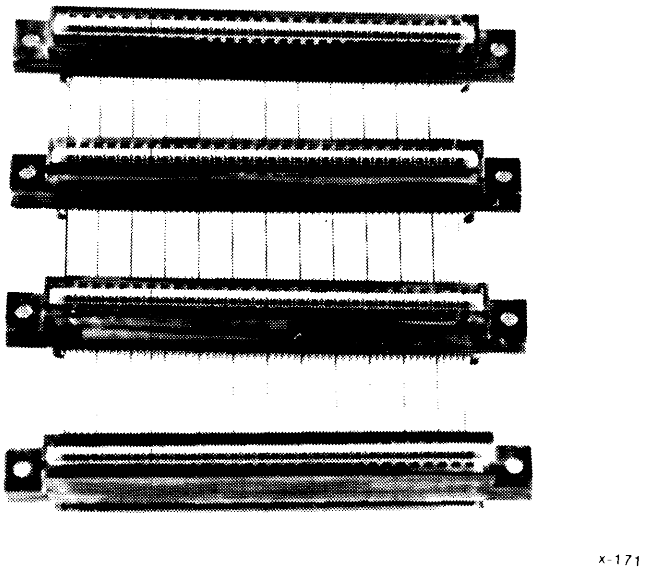


Figure 12 Typical iLBX™ Bus Interface Cable Assembly

5. Design Guidelines and System Applications

5.1 Introduction

This section provides examples of the typical circuits required to implement the interface between a given device and the iLBX bus. The circuit examples used in this chapter illustrate the main interface circuits. The examples do not show most of the on-board circuits that generate the timing of the control signals. Two full interface examples, a 16-bit Primary Master and a 16-bit Slave device, illustrate the general interfacing requirements for the iLBX bus. Additional partial examples show the special considerations for a Secondary Master, and an 8-bit device (master or slave). Wherever possible, the examples represent proven interface circuits.

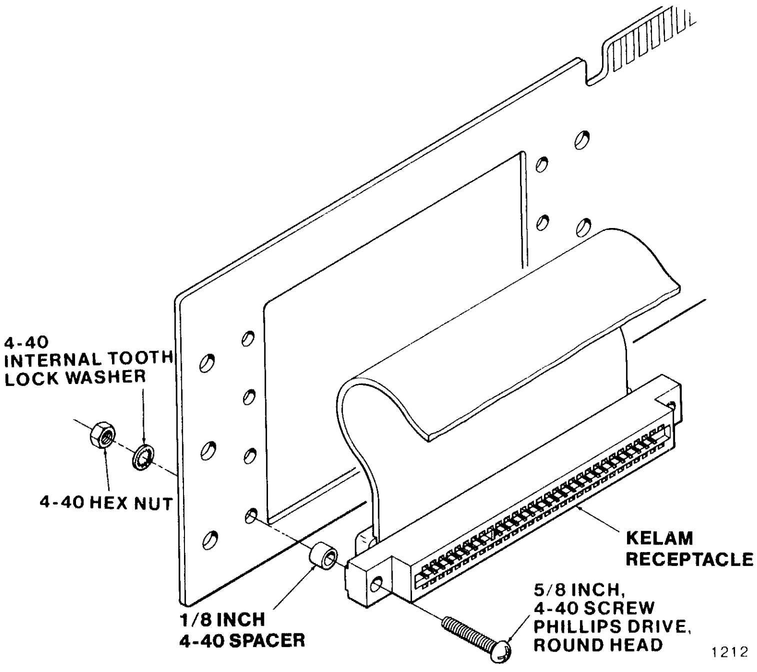


Figure 13 Kelam Connector Installation

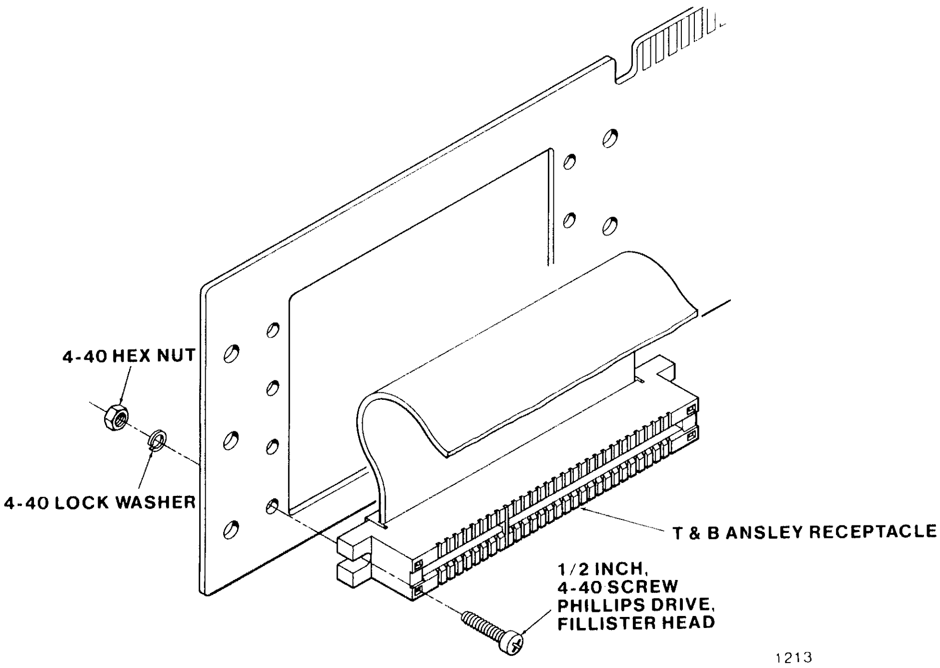


Figure 14 T & B Ansley Connector Installation

The examples do not attempt to show fully optimized circuits. In most of the examples, the iLBX bus interface derives signals directly from the microprocessor or support component signals. For example, the Primary Master circuit example shows the Secondary Master Request signal applied, through an inverting buffer, to the HOLD signal input to the microprocessor. In turn, the microprocessor HOLDA signal, through an inverting buffer, drives the Secondary Master Acknowledge line. Thus in this example, when the Primary Master transfers control to the Secondary Master, the microprocessor waits in the standby mode until the Secondary Master returns control to the Primary Master.

5.2 Primary Master Design Example — 16-Bit

The 16-bit Primary Master interface design example is for a full rather than limited Primary Master. Because the full Primary Master must be able to transfer control to a Secondary Master, the design example uses tri-state drivers for the address lines. The interface example also shows the jumpers required to allow the Primary Master to access the front panel control signals over the standard Multibus interface P2 connector. (The option is mutually exclusive. The Primary Master P2 connector must be interfaced to the iLBX bus or the Multibus interface. It cannot interface to both at the same time.) Figure 15 illustrates the 16-bit Primary Master interface example.

5.2.1 Address Decode

The 16-bit circuit example uses a programmed array logic (PAL) component (U7) to decode the memory address from the processor and direct the address to the appropriate memory array. The PAL in the example shows four outputs: Multibus memory access (MBACESS*); on-board RAM access (OBRAM*); on-board ROM access (OBROM*); and iLBX bus access (LBXEN*). The PAL inputs shown are the address lines AE-A17, the memory I/O signal ($\overline{M}$/IO), and three select lines with programming jumpers. The memory IO input limits PAL address decoding to memory accesses only. The address line inputs provide memory address decoding in 16k byte increments for the on-board RAM and Multibus memory access. The programmable select inputs specify the iLBX bus memory size in 512k byte increments. Table 11 lists the jumper configuration used to set the upper limit for the iLBX bus memory. The select jumpers in the circuit example relate directly to the address lines: E5-E6 to A13, E3-E4 to A14, and E1-E2 to A15. Installation of a jumper sets the select value to zero. The first two configurations in the Table 11 are reserved to avoid overlap with the standard Multibus memory address range.

For example, assume the following system memory configuration: 16k bytes of on-board RAM (base address at 0H); 32k bytes of on-board ROM (base address at FF8000H); and jumper E1-E3 only installed corresponding to 1M byte of iLBX bus memory (base address at 100000H). The PAL drives the OBRAM* output active for memory addresses within the address range of 0H to 03FFFH. The PAL drives the OBROM* output active for memory addresses within the address range of FF8000H to FFFFFFH. The PAL drives the MBACCESS* output active for memory addresses within the address range of 04000H to 0FFFFFH. The PAL drives the LBXEN* output active for memory addresses within the address range of 100000H to 1FFFFFH.

5.2.2 Data Drivers

The 16-bit circuit example uses two 74LS245 octal tri-state transceivers (U4 and U5) to drive and receive the iLBX bus data lines DB15-DB0. The ANDed combination of the iLBX Bus Enable (LBXEN*) signal active from the address decode and Data Enable (DEN*) signal active from the processor circuit drives the chip select enable ($\overline{CS}$) to the data transceivers. The processor's Data Transmit Not Receive (DT/$\overline{R}$) signal directly drives the transceiver direction control input (DIR).

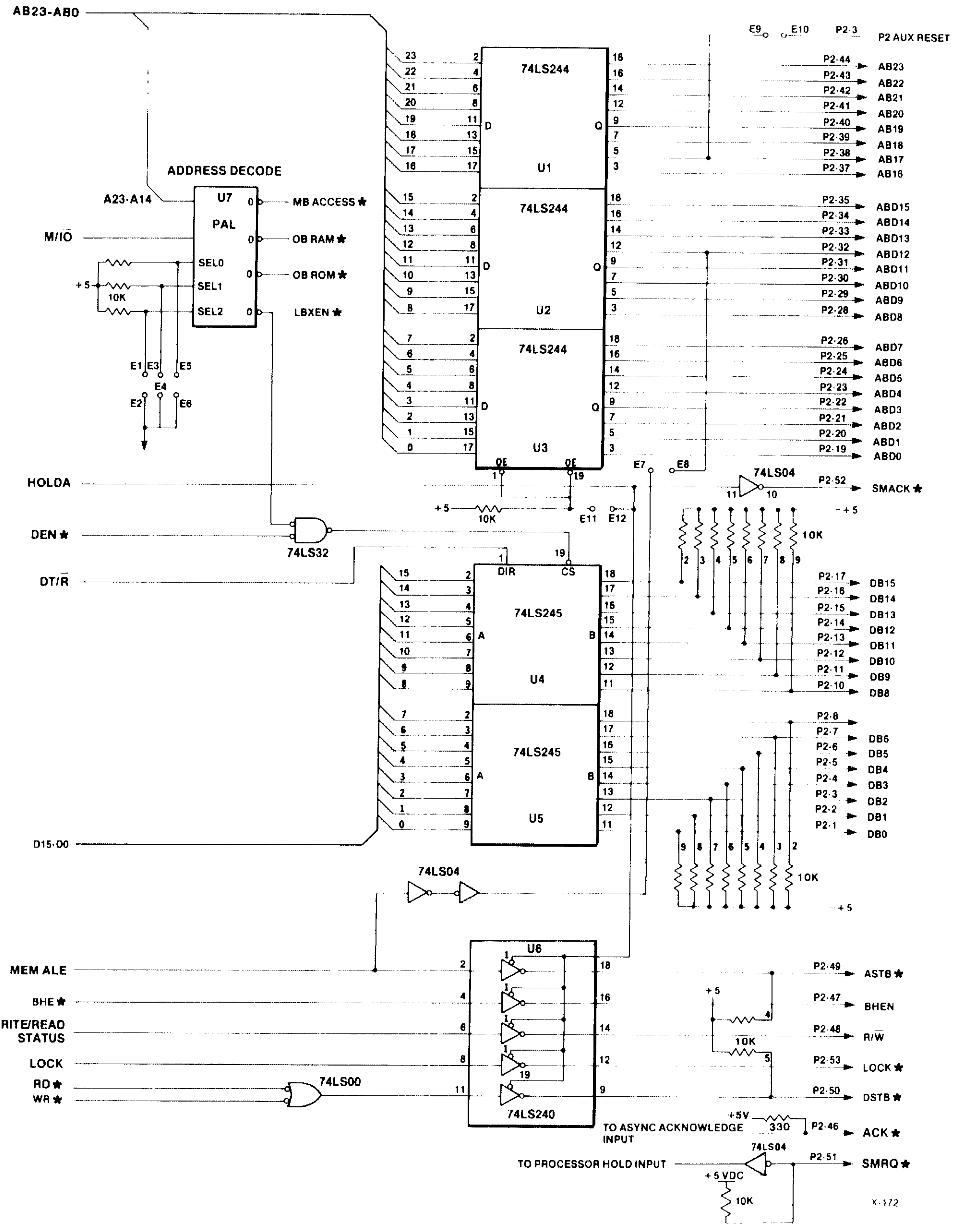


Figure 15 Interface Circuit Example — 16-Bit Primary Master

5.2.3 Address Drivers

The 16-bit circuit example uses three 74LS244 octal tri-state buffers (U1, U2, and U3) to drive the iLBX bus address lines. Typically, the on-board processor keeps the address lines enabled to the iLBX bus Slave devices unless the iLBX bus interface is specifically disabled.

Table 11 iLBX™ Bus Address Range Select Jumpers

iLBX™ BUS UPPER LIMIT	SELECT JUMPERS		
	E1-E2	E3-E4	E5-E6
RESERVED	IN	IN	IN
RESERVED	IN	IN	OUT
17FFFFH	IN	OUT	IN
1FFFFFH	IN	OUT	OUT
27FFFFH	OUT	IN	IN
2FFFFFH	OUT	IN	OUT
37FFFFH	OUT	OUT	IN
DISABLE iLBX™ ACCESS	OUT	OUT	OUT

5.2.4 Control and Command Drivers

The 16-bit circuit example uses an 74LS240 octal tri-state inverting buffer (U6) to drive the iLBX bus control and command lines. The interface circuit example uses processor signal lines or direct derivatives from processor signal lines as inputs to the tri-state buffer. With this design, the processor uses the iLBX bus as though it were an extension of the processor's local bus and drives the iLBX bus control and command signals each time the processor accesses memory. Whenever the processor directs the memory access to non-iLBX bus memory, none of the iLBX Slave devices respond to the control signals and the processor aborts the iLBX bus portion of the operation when the non-iLBX bus memory responds.

In the example, the processor's Address Latch Enable signal (or a direct derivative) drives the iLBX bus Address Strobe line (ASTB*). The processor's Byte High Enable (BHE*) and Lock (LOCK*) signal directly drive the corresponding iLBX bus lines (BHEN and LOCK*). The ORed combination of the processor's Read strobe (RD*) and Write strobe (WR*) signals drives the iLBX bus Data Strobe line (DSTB*). The iLBX bus Read-Not-Write line does not have a directly equivalent processor signal and must be derived from the processor's status signals. To implement the circuit example with an 8086 family processor, derive the Read-Not-Write drive signal from the S1* status output.

Typically, the on-board processor keeps the control and command lines enabled to the iLBX bus Slave devices unless the iLBX bus interface is specifically disabled.

5.2.5 Interface Disabling

The circuit example depicts two conditions when the address, control, and command tri-state drivers would be placed in the high impedance state. The first condition occurs when the on-board processor drives the Secondary Master Acknowledge line (processor HOLDA or equivalent signal, iLBX bus SMACK*) Low in response to the Secondary Master's driving the Secondary Master Request line (SMRQ*) Low. The circuit example shows the Secondary Master Request signal connected through an inverting buffer to the processor's Hold request input. The second condition occurs as a result of configuring the board for non-iLBX bus operation. The configuration change requires three jumper changes. The removal of the Jumper E11-E12 disconnects the HOLDA signal from pins 1 and 19 (output enable) on the address line drivers, U1-U3. Pins 1 and 19 are pulled High and the buffers placed in the high-impedance state. The installation of Jumper E7-E8 connects the Address Latch Enable signal to P2 connector pin 32 and the installation of Jumper E9-E10 connects the Auxiliary Reset signal to P2 connector pin 38.

Neither disabling option is required under the following conditions.

- Limited Primary Masters that do not share the iLBX bus with a Secondary Master can use standard TTL drivers with drive characteristics comparable to the specified tri-state buffers for the line drivers. Limited Primary Masters are not required to monitor the Secondary Master Request line.
- An iLBX bus compatible DMA controller would not need the partial Multibus interface P2 compatibility.

5.2.6 Address Strobe Timing

Implementing the iLBX bus interface circuit example shown in Figure 16 on a board with an 8086/8088 processor requires the use of an additional timing support circuit. The 8086/8088 processor address line set-up time relative to its driving the Address Latch Enable signal active is shorter than the required address set-up for the iLBX bus. Thus, the processor's Address Latch Enable signal cannot directly drive the iLBX bus Address Strobe line. Figure 17 illustrates a T-state generator initialized by the processor's Address Latch Enable signal. The example uses the output of the T-state generator to develop a delayed Address Strobe drive signal from the Address Latch Enable signal.

Immediately following the Address Latch Enable signal going active, the T-state generator outputs a sequence of four valid signals directly related to the processor T-states. During the processor T1 time, the active Address Latch Enable (inverted) from the processor resets the 74S175 (D-type flip-flops) Q outputs Low. The processor deactivates the Address Latch Enable signal before the end of the T1 time. At the next falling edge of the processor clock, the D1 flip-flop sets and the Q1 output goes High signaling the start of the processor T2 time.

Because the T-state generator has the Q1 output connected to the D2 input, at the next fall of the processor clock, the Q2 output goes High signaling the start of the processor T3 time. Additional T-state signals can be generated by attaching the Q2 output to the D3 input and the Q3 output to the D4 input. Once the T-state generator completes the sequence, all outputs remain High until the next time the processor activates the Address Latch Enable signal.

The circuit example produces the iLBX bus Address Strobe drive signal by ANDing the Q1 output with the inverted Q2 output. At the start of the T2 time when the Q1 output goes High, the AND gate output goes High driving the iLBX bus Address Strobe line Low. At the start of the T3 time when the inverted Q2 output goes Low, the AND gate output goes Low driving the iLBX bus Address Strobe line High. Thus the circuit example generates the Address Strobe signal starting at the processor T2 time with a width of one processor clock cycle.

5.3 Primary Master Design Example — 8-Bit

An 8-bit device is a master or a slave designed to perform all data transfers as 8-bit bytes. An iLBX compatible device capable of making 16-bit data transfers over the iLBX bus is classed as a 16-bit device, regardless of its internal architecture. When designing an 8-bit Primary Master, the level of 16-bit compatibility required must be determined. The data transceiver portion of the interface must accommodate the desired 8-bit data transfer format.

Designing an 8-bit Primary Master that operates with 8-bit Slave devices only, an iLBX bus interface circuit similar to the 16-bit Primary Master interface circuit can be used. Because the 8-bit interface does not use the high-order eight data lines, the 74LS245 tri-state transceiver, U4, can be eliminated and the connection from the processor Byte High Enable (BHE*) signal to the control and command driver can be eliminated. The remainder of the interface circuit example can be implemented as shown.

Designing an 8-bit Primary Master that operates with both 8-bit or 16-bit Slave devices requires a more extensive modification to the data transceiver circuit. Figure 17 shows a data transceiver circuit example that allows the 8-bit Primary Master to operate with 16-bit slave devices. An added feature of the circuit example, the jumper option for the A0 address bit, allows the interface to work with either 8-bit or 16-bit Slave devices.

When configured for compatibility with 16-bit Slave devices (E1-E2 connected), the A0 address bit from the processor controls transceiver selection and the iLBX bus Byte High Enable line. When the processor addresses an even numbered byte (A0 Low), the byte is placed on the low-order data lines DB7-DB0.

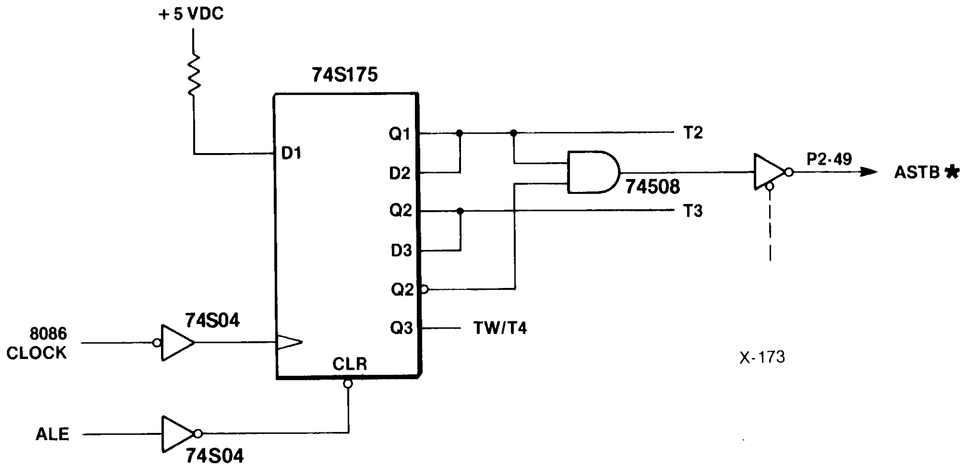


Figure 16 T-State Generator Circuit

When the processor addresses an odd numbered byte (A0 High), the byte is placed on the high-order data lines DB15-DB8. In this implementation, the unused data transceiver holds the corresponding data lines in the high-impedance state.

When configured for compatibility with 8-bit Slave devices (E2-E3 connected), the A0 address bit line is opened and the selection circuit input connected to ground, forcing all data transfers to take place over the low-order data lines DB7-DB0.

5.4 Secondary Master Design Example

Because most of the iLBX bus interface circuits are the same as those used for the Primary Master, the Secondary Master interface design example concentrates on the following three circuits used specifically on the Secondary Master:

- the Secondary Master Request and Secondary Master Acknowledge circuit
- the adjustment circuit for the Secondary Master acknowledge acceptance time
- the Secondary Master to limited Primary Master conversion circuit requirements

Figure 18 illustrates the Secondary Master interface design example. Figure 18 does not repeat in detail the circuits shown in the Primary Master design example. These circuits are generalized in block diagram form to emphasize the circuits being described.

5.4.1 Bus Request Circuit

The Secondary Master circuit example uses a cross coupling latch, along with timing derived from the on-board clock, to meet the bus request and release requirements. In the example, the cross coupling latch is wired-up from a pair of two input NAND gates. Any time the Primary Master has control of the bus, the Secondary Master Acknowledge (SMACK*) line is High. Inverted, the Secondary Master Acknowledge preconditions the cross coupling latch to accept an iLBX bus request from the on-board iLBX controller. With the cross coupling latch preconditioned, a valid iLBX request (LBXRQ) from the iLBX controller removes the reset clamp from the synchronizing flip-flop FF2 and resets FF3, driving the Secondary Master Request (SMRQ*) line Low. When the Primary Master responds by driving SMACK* low, the synchronizing flip-flops, FF1 and FF2, delays the input for two clock pulses (50 to 100 ns) before the enabling the iLBX bus drivers.

The same circuit times the release of the iLBX bus back to the Primary Master. The Secondary Master initiates the release by driving the internal LBXRQ signal Low. This resets FF2 and immediately disables the iLBX bus drivers. At the next clock pulse, FF3 is reset to drive the Secondary Master request line High. At this time, the cross coupling latch is only partially restored to its preconditioned wait state. An iLBX request by the iLBX controller is blocked by the cross coupling latch and cannot be recognized until the Primary Master drives SMACK* High.

5.4.2 Programmable Acknowledge Delay Circuit

The Secondary Master can optionally provide an Acknowledge delay circuit for matching its acknowledge acceptance overhead time to that of the Primary Master. The Secondary Master example uses a 74175 hex D-type flip-flop wired as a delay line for the Acknowledge delay. Immediately following the Acknowledge (ACK*) going active, the delay line outputs a sequence of six Qualified Acknowledge (QACK) signals each delayed by 50 ns from the preceding signal. By connecting the output for the desired delay to the iLBX controller input, any levels of delay can be programmed for the Secondary Master.

5.4.3 Master Level Conversion

The Secondary Master can provide a configuration option to convert it from a Secondary Master to a limited Primary Master. In the Secondary Master example, two changes are required to make the conversion. The jumper from E1 to E2 connects the iLBX bus request from the cross coupling latch to the SMACK* input. Because the Secondary Master (now limited Primary Master) must provide termination for the bus lines, sockets are provided to accept the applicable plug-in resistor packs.

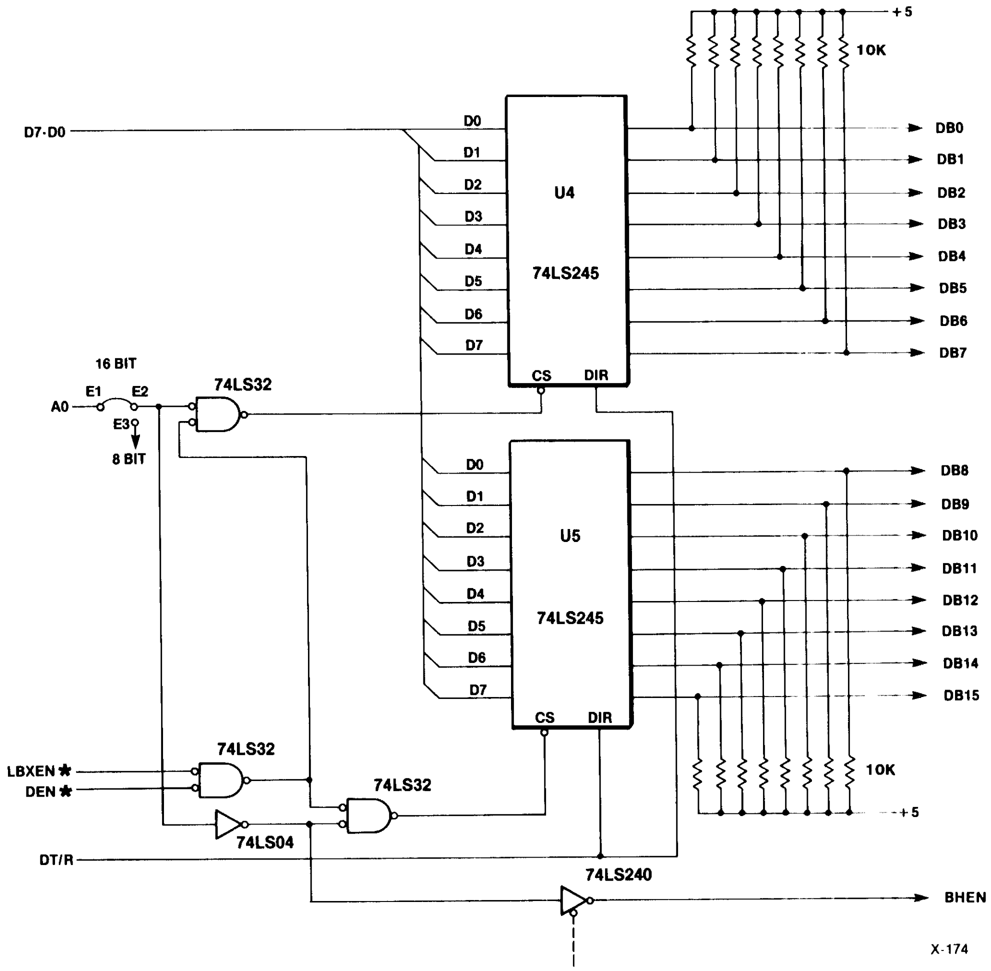


Figure 17 The 8-Bit Data Transceiver Circuit

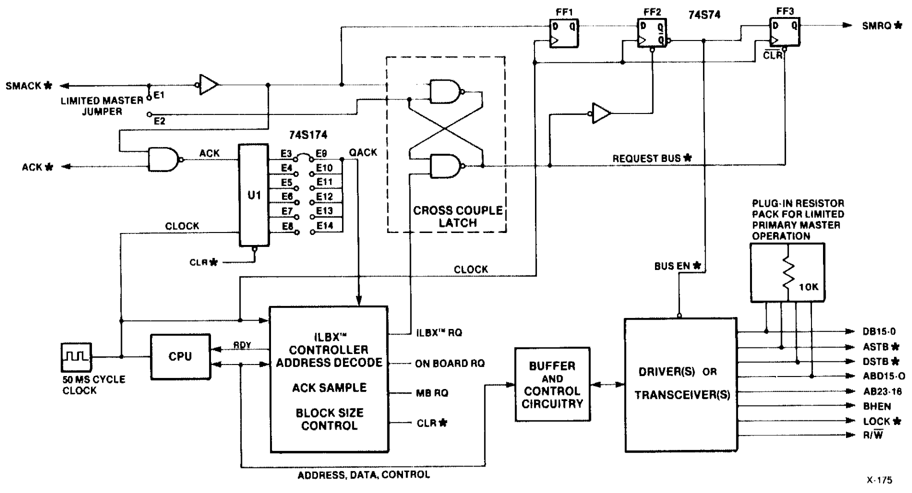


Figure 18 Interface Circuit Example — Secondary Master

5.5 Slave Design Example

Because most of the iLBX bus interface circuits are similar to those used for the Primary Master, the Slave interface design example concentrates on the following circuits used specifically on the Secondary Master:

- the data element decode circuit
- the Acknowledge set up and pre-acknowledge adjustment circuit

Figure 19 illustrates the Slave interface design example. Figure 19 does not repeat in detail the circuits shown in the Primary Master design example. These circuits are generalized in block diagram form to emphasize the circuits being described.

5.5.1 Data Element Circuit

The Slave circuit example uses a pair of three input NAND gates to select the data element being transferred and enable the interface transceivers. The circuit example allows the Slave to accept or transmit a 16-bit word, or either the high-order or low-order byte. In the circuit example, the data line transceivers are disabled (in the high impedance state) except for the Data Strobe time when the Slave is selected.

5.5.2 Programmable Acknowledge Delay Circuit

The Slave can provide the choice of timing the driving the Acknowledge signal from Address Strobe or Data Strobe. The interface timing of the master(s) determines if Address Strobe can be used or if Data Strobe must be used. Connecting a jumper from E1 to E2 selects the Address Strobe as the clock to set the start cycle flip-flop, FF2. Connecting the jumper from E2 to E3 selects the Data Strobe as the start cycle clock. The clock and acknowledge timer circuit implements a delay line comparable to the delay line in the Secondary Master circuit example. Immediately following the Start Cycle going active, the delay line outputs a sequence of six signals each delayed by 50 ns from the preceding signal. By connecting the output for the desired delay to the Acknowledge driver, any level of delay can be programmed for the Acknowledge signal.

5.6 System Timing Considerations

Because the iLBX bus allows close coupling between the operation of the master and slave devices, the level of close coupling (optimization) used must be determined and set at system implementation time. The following descriptions examine the various read and write timing considerations for system implementation.

5.6.1 General Considerations

When the Primary and, if installed, the Secondary Master meet the optimized timing requirements for a master device, then the Slave device(s) can be configured to drive the Acknowledge line Low before the leading (falling) edge of the Data Strobe. Otherwise, the Slave device(s) must wait for the leading edge of the Data Strobe. A Slave device without the Data Strobe pre-acknowledgement capability can operate with the optimized timing from the Primary Master. Also, a mix of optimized and non-optimized Slave devices can operate successfully with an optimized master. If either the Primary Master or the Secondary Master cannot meet the optimized timing, all the Slave devices must be configured for non-optimized operation. The optimized master continues to use the optimized timing, but it does not realize the full optimum performance.

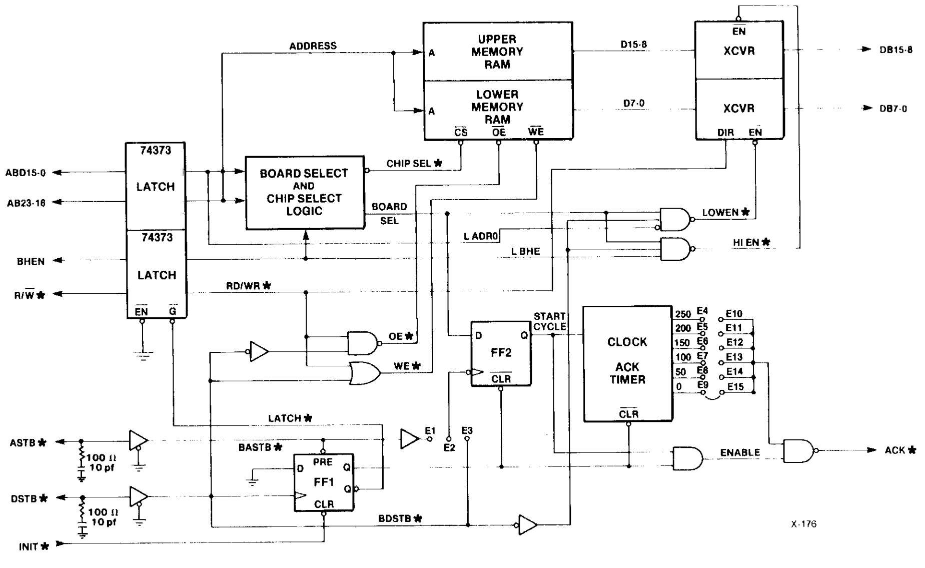


Figure 19 Interface Circuit Example — Slave

When a Slave device has variable Acknowledge timing for the read operation, it can pre-acknowledge the read data transfer by driving the Acknowledge line Low before it provides valid data on the data lines. The amount of variability provided can range from 0.0 ns (data valid when the slave drives the Acknowledge line Low) to the maximum access time of the slave's memory resources (driving Acknowledge Line Low immediately after detecting the leading of the Address Strobe). The amount of pre-acknowledgement implemented must be less than the acknowledge acceptance overhead of the fastest master.

5.6.2 Read Timing Examples

The following examples are provided to better illustrate the timing considerations.

Assume an iLBX bus implementation with a Primary Master and a single Slave device. The Slave device memory access time is 150 ns, and the Slave has pre-acknowledge capabilities (variable Acknowledge timing in 10 ns increments). The Primary Master has a 100 ns internal acknowledge acceptance overhead time (from receipt of Acknowledge to the sampling of the data lines). The Figure 20 timing chart presents the non-optimized and optimized timing for a read data transfer from the leading edge of the Address Strobe until the time when the Primary Master samples the data lines (SAMPLE* on the timing chart). In the optimized configuration, there is a 90 ns improvement in the total data transfer time.

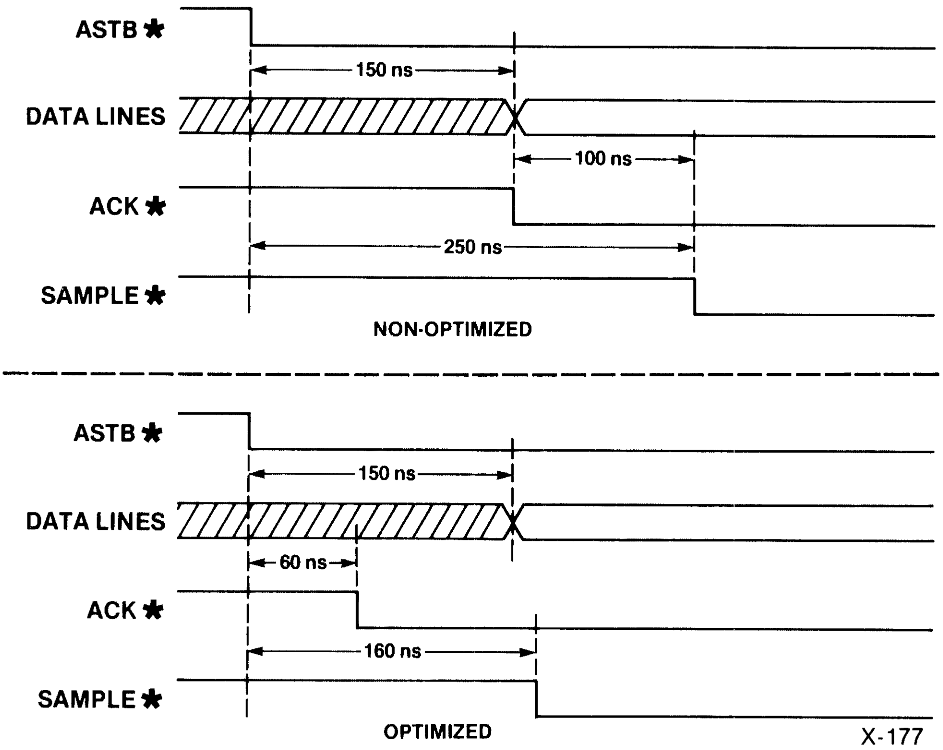


Figure 20 Read Data Optimization Timing — Basic Example

The example presented in Figure 20 is very simple and is used to demonstrate the basic optimized read data transfer considerations. The following example is more complex and assumes a Primary Master, a Secondary Master, and two Slave devices. Assume the Primary Master is relatively slow and has an acknowledge acceptance time of 350 ns. The Secondary Master is relatively fast with an acknowledge acceptance time of 100 ns. The Slave devices also have different memory access times. Slave A's access time is 100 ns with variable Acknowledge timing at 10 ns increments. Slave B's access time is 250 ns with variable Acknowledge timing at 25 ns increments.

Because the devices in this example do not have common timing parameters, the various timing considerations must be considered separately and in a specific order. Assume for now that the Secondary Master's acknowledge acceptance overhead time is fixed (not adjustable).

1. First compare the Secondary Master's overhead time with the Primary Master's overhead time. Because the iLBX bus master cannot indicate to the Slave device which master is accessing the Slave device, the amount of pre-acknowledgement implemented must be less than the acknowledge acceptance overhead of the fastest master.Thus, the overhead time of the Secondary Master (100 ns) becomes the standard master overhead time for this iLBX bus implementation.

2. Set Slave A's pre-acknowledge time to the first increment under 100 ns (90 ns).

3. Set Slave B's pre-acknowledge time to the first increment under 100 ns (75 ns).

The Figure 21 timing chart presents the optimized timing for a read data transfer from both Slave devices to both masters. Note that Slave A is close to fully optimized at 90 ns of pre-acknowledgement. However, Slave B is limited to 75 ns of pre-acknowledgement by the Secondary Master and system throughput is thus limited whenever Slave B is accessed.

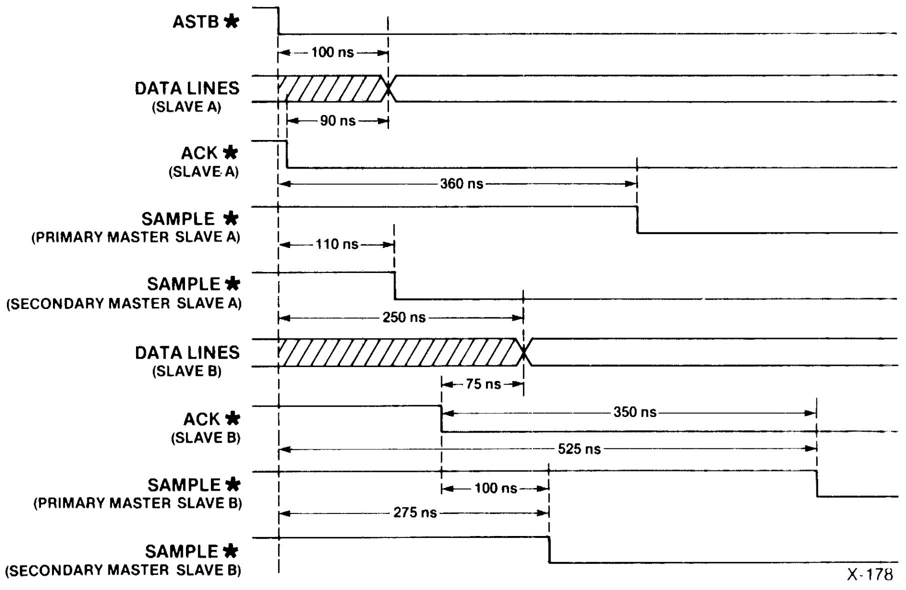


Figure 21 Read Data Optimization Timing — Non-adjustable, Secondary Master Example

The two preceding examples of read operation optimization resulted in direct system performance improvements when optimization is used. The next example adds a timing variable to the Secondary Master. This added level of variability means that overall system operation must be considered when and where to implement the read operation optimization.

Continuing to use the same devices, now assume the Secondary Master provides an adjustment to lengthen the its acknowledge acceptance overhead time in 25 ns increments. Again, the various timing considerations must be considered in a specific order to arrive a the optimum settings.

1. Compare the Secondary Master's overhead time with the Primary Master's overhead time. The Primary Master has the longer overhead time so the Secondary Master's overhead time can be lengthened, possibly to 350 ns equaling the Primary Master's overhead time.

2. Compare the 350 ns overhead time with the Slave devices maximum access time. Because Slave B has a maximum memory access time of 250 ns that is shorter than the Primary Master's overhead time, the Secondary Master's overhead time should be set to a maximum of 275 ns.

3. Set Slave A for its maximum pre-acknowledge time of 100 ns.

4. Set Slave B for its maximum pre-acknowledge time of 250 ns.

The Figure 22 timing chart presents the optimized timing for a read data transfer from both Slave devices to both masters. In this example, both Slave devices are fully optimized. The coupling between the masters and the slaves is stabilized at 350 ns for the Primary Master and 275 ns for the Secondary Master. A comparison between the timing charts in Figure 21 and Figure 22 show that, taken on an individual basis, the throughput for some combinations remain about the same (Primary Master to Slave A and Secondary Master to Slave B), one combination (Primary Master to Slave B) was improved dramatically, and one combination (Secondary Master to Slave A) was much slower.

Thus, the final consideration becomes a system implementation question. For example, if the Primary Master accesses the memory on Slave A and Slave B equally and the Secondary Master almost exclusively accesses the memory on Slave B, then the fully optimized configuration (Figure 22 timing) would improve overall system throughput. Other combinations of use might dictate using the partially optimized timing represented by Figure 21.

6. Levels of Compliance

6.1 Introduction

This section bounds the variability allowed within the iLBX bus specification. The main purpose in bounding variability is to assure the maximum amount of upward compatibility. In most cases, mixing devices designed to different levels of compliance results in the more complex devices in the system operating at the level of the least complex device.

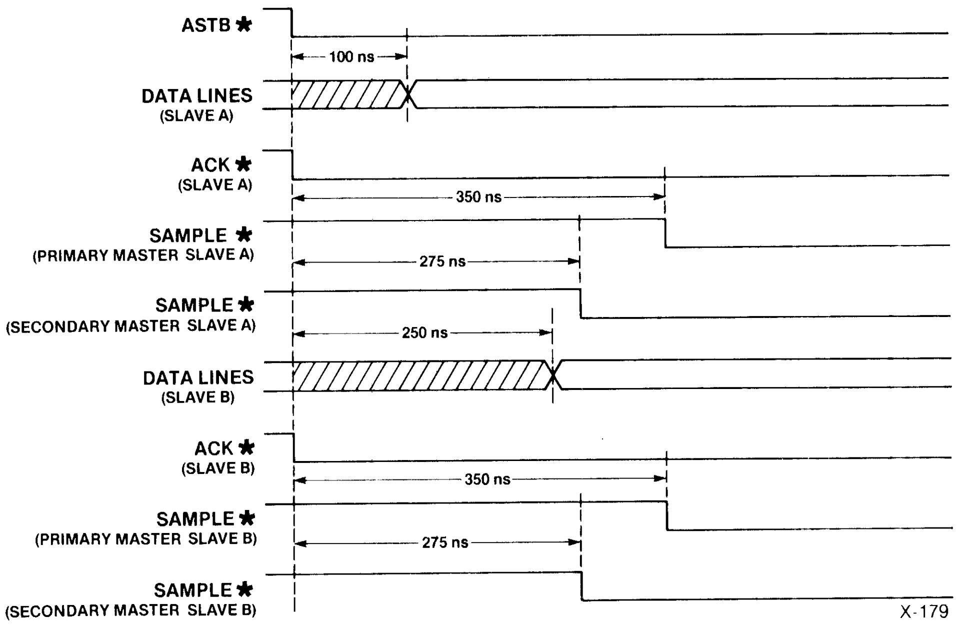


Figure 22 Read Data Optimization Timing, Adjustable Secondary Master Example

6.2 Data Path

The iLBX bus allows 8 and 16-bit data path devices. The 8- and 16-bit data path devices share the same mechanical implementation of the iLBX bus. When a system with a mixture of data path width devices is implemented, the width of a data transfers is limited to the narrower data path width of the master and slave transferring the data. For example, assume that a 16-bit data path Primary Master is connected to a 16-bit data path slave and an 8-bit data path slave. The master would be limited to 8- or 16-bit transfers with the 16-bit data path slave and 8-bit transfers only with the 8-bit data path slave.

6.3 Address Path

There are 24 address lines (AB23 - AB0) defined for the iLBX bus. All devices, masters and slaves, must implement all 24 lines.

6.4 Signal Line Connections

In general, an iLBX bus compatible device needs to connect to most of the iLBX bus lines. The iLBX bus includes a single optional line, Transfer Parity. The following is a list of additional exceptions to full iLBX bus connection:

- 8-bit data path devices need not connect to the Byte High Enable line.
- 8-bit data path devices need not connect to the high-order byte data lines (DB15 - DB8).

- Slave devices should not connect to the Secondary Master Request and Secondary Master Acknowledge lines.
- Single port Slave devices need not connect to the Lock line.
- ROM memory Slave devices need not connect to the Read Not Write line.
- Limited Primary Masters need not connect to the Secondary Master Request or the Secondary Master Acknowledge lines.

6.5 Documentation

The documentation (Hardware Reference Manual or equivalent) for an iLBX bus compatible device should include specifications for the following items:

- type of device - Primary Master (PM), Limited Primary Master (LPM), Secondary Master (SM), or Slave (SL)
- data path width - 8-bit (D8) or 16-bit (D16)
- parity support (P)
- signal lines not used
- acknowledge acceptance overhead time for a Primary Master
- range of pre-acknowledge adjustment and adjustment increments for a Slave device
- minimum data strobe active to the preacknowledge time, if greater than zero, for a slave device.
- maximum leading edge of Address Strobe to leading edge of Data Strobe time for a master
- minimum Data Strobe to read data sample time for a master
- maximum Data Strobe to write data valid on the data lines for a master device
- maximum Data Strobe to read data valid on the data lines for a Slave device
- range of acknowledge acceptance delay adjustment and adjustment increments for a Secondary Master

6.6 Compliance Statement

The compliance level of an iLBX bus compatible board should be clearly stated in the printed specifications. Omission of a capability denotes that the device does not support the capability. For example, a Primary Master that can communicate with 8- or 16-bit data path devices would be marked as follows: PM D8 D16.

A 16-bit data path Slave with parity support would be marked as follows: SL D16 P.

A 16-bit data path Secondary Master that can also operate as a Primary Master would be marked as follows: PM SM D16.

APPENDIX A
iLBX™ SIGNAL DESCRIPTIONS SUMMARY

A.1 Introduction

This appendix provides a summary description of the iLBX bus signal lines. Refer to Section 2 for the full descriptions of the signal lines and the implementation parameters. Refer to Section 3 for the electrical specifications for the signal lines, and to Section 4 for the mechanical specifications. Table 4 lists the iLBX bus pin assignments for the 60-pin P2 edge connector.

A.2 Signal Line Summary

DATA LINES (DB15 - DB0)
The 16 bi-directional data lines used to transfer data between the active bus master and the selected Slave device.

ADDRESS LINES (AB23 - AB0)
The 24 address lines used by the active bus master to select a Slave device and to specify a location in memory. For the 16-bit iLBX bus configuration, the address lines are single-direction lines used exclusively for addressing.

TRANSFER PARITY (TPAR*)
The optional Transfer Parity line operates as an additional data line to improve data-transfer integrity.

READ-NOT-WRITE (R/$\overline{W}$)
The active bus master controls the direction of data transfer with the Read-Not-Write line. When driven Low, the active bus master transmits the data and the selected slave device receives the data. Driving the Read-Not-Write line High reverses the transfer direction.

BYTE HIGH ENABLE (BHEN)
The active bus master in the 16-bit iLBX bus configuration controls the type of data transfer (8-bit or 16-bit) using the Byte High Enable (BHEN) element select line along with the low-order address bit (AB0).

ADDRESS STROBE (ASTB*)
The active bus master drives the Address Strobe line Low to initiate a data transfer cycle.

DATA STROBE (DSTB*)
The active bus master drives the Data Strobe line Low to set-up the actual transfer of data.

ACKNOWLEDGE (ACK*)
The selected slave device responds to selection by the active bus master by driving the Acknowledge line low.

LOCK (LOCK*)
The active bus master restricts access through the alternate port to dual port RAM memory on a Slave device by driving the Lock line Low.

SECONDARY MASTER REQUEST (SMRQ*)

The Secondary Master requests use of the iLBX bus from the Primary Master by driving the Secondary Master Request line low.

SECONDARY MASTER ACKNOWLEDGE (SMACK*)

The Primary Master allows use of the iLBX bus by the Secondary Master by driving the Secondary Master Acknowledge line Low after the Secondary Master drives the Secondary Master Request line Low.

Table 12 iLBX™ Bus Pin Assignments, P2 Edge Connector

COMPONENT SIDE			SOLDER SIDE		
PIN	SIGNAL	SIGNAL NAME	PIN	SIGNAL	SIGNAL NAME
1	DB0	DATA LINE 0	2	DB1	DATA LINE 1
3	DB2	DATA LINE 2	4	DB3	DATA LINE 3
5	DB4	DATA LINE 4	6	DB5	DATA LINE 5
7	DB6	DATA LINE 6	8	DB7	DATA LINE 7
9	GND	GROUND	10	DB8	DATA LINE 8
11	DB9	DATA LINE 9	12	DB10	DATA LINE 10
13	DB11	DATA LINE 11	14	DB12	DATA LINE 12
15	DB13	DATA LINE 13	16	DB14	DATA LINE 14
17	DB15	DATA LINE 15	18	GND	GROUND
19	AB0	ADDRESS LINE 0	20	AB1	ADDRESS LINE 1
21	AB2	ADDRESS LINE 2	22	AB3	ADDRESS LINE 3
23	AB4	ADDRESS LINE 4	24	AB5	ADDRESS LINE 5
25	AB6	ADDRESS LINE 6	26	AB7	ADDRESS LINE 7
27	GND	GROUND	28	AB8	ADDRESS LINE 8
29	AB9	ADDRESS LINE 9	30	AB10	ADDRESS LINE 10
31	AB11	ADDRESS LINE 11	32	AB12	ADDRESS LINE 12
33	AB13	ADDRESS LINE 13	34	AB14	ADDRESS LINE 14
35	AB15	ADDRESS LINE 15	36	GND	GROUND
37	AB16	ADDRESS LINE 16	38	AB17	ADDRESS LINE 17
39	AB18	ADDRESS LINE 18	40	AB19	ADDRESS LINE 19
41	AB20	ADDRESS LINE 20	42	AB21	ADDRESS LINE 21
43	AB22	ADDRESS LINE 22	44	AB23	ADDRESS LINE 23
45	GND	GROUND	46	ACK*	SLAVE ACKNOWLEDGE
47	BHEN	BYTE HIGH ENABLE	48	R/$\overline{W}$	READ NOT WRITE
49	ASTB*	ADDRESS STROBE	50	DSTB*	DATA STROBE
51	SMRQ*	SECONDARY MASTER REQUEST	52	SMACK*	SECONDARY MASTER ACKNOWLEDGE
53	LOCK*	ACCESS LOCK	54	GND	GROUND
55	ADR22*	MULTIBUS ADDRESS EXTENSION LINE 22	56	ADR23*	MULTIBUS ADDRESS EXTENSION LINE 23
57	ADR20*	MULTIBUS ADDRESS EXTENSION LINE 20	58	ADR21*	MULTIBUS ADDRESS EXTENSION LINE 21
59		RESERVED	60	TPAR*	TRANSFER PARITY

REQUEST FOR READER'S COMMENTS

Intel's Technical Publications Departments attempt to provide publications that meet the needs of all Intel product users. This form lets you participate directly in the publication process. Your comments will help us correct and improve our publications. Please take a few minutes to respond.

Please restrict your comments to the usability, accuracy, readability, organization, and completeness of this publication. If you have any comments on the product that this publication describes, please contact your Intel representative. If you wish to order publications, contact the Intel Literature Department (see page ii of this manual).

1. Please describe any errors you found in this publication (include page number).

____
____
____
____
____
____

2. Does the publication cover the information you expected or required? Please make suggestions for improvement.

____
____
____
____
____

3. Is this the right type of publication for your needs? Is it at the right level? What other types of publications are needed?

____
____
____
____
____
____

4. Did you have any difficulty understanding descriptions or wording? Where?

____
____
____
____

5. Please rate this publication on a scale of 1 to 5 (5 being the best rating). ____________

NAME ______________________________ DATE ____________

TITLE ____

COMPANY NAME/DEPARTMENT ______________________________

ADDRESS ____

CITY ____________________ STATE ____________ ZIP CODE ____________
(COUNTRY)

Please check here if you require a written reply. □

WE'D LIKE YOUR COMMENTS . . .

This document is one of a series describing Intel products. Your comments on the back of this form will help us produce better manuals. Each reply will be carefully reviewed by the responsible person. All comments and suggestions become the property of Intel Corporation.

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 79 BEAVERTON, OR

POSTAGE WILL BE PAID BY ADDRESSEE

Intel Corporation
5200 N.E. Elam Young Pkwy.
Hillsboro, Oregon 97123

OMO Technical Publications

intel®

INTEL CORPORATION, 3065 Bowers Avenue, Santa Clara, California 95051 (408) 987-8080

Printed in U.S.A.