EK-369AA-OD-007A

TURBOchannel

Hardware Specification

On-line version.

Previous versions of this document are obsolete and should be discarded. This document supersedes all previous versions.

digital™

September 1991

The following are trademarks of Digital Equipment Corporation:

DECstation DECsystem TURBOchannel digital™

The following are trademarks of MIPS Computer Systems, Inc.:

MIPS R3000 R4000

Digital's TRI/ADD Program provides technical and marketing support worldwide to third-party vendors using the SCSI, TURBOchannel, VME, and Futurebus+ interconnects to develop add-on products for open systems.

To receive free technical support and notice of new TURBOchannel documentation, contact Digital's TRI/ADD Program about free membership at the numbers below.

Digital Equipment Corporation
TRI/ADD Program
100 Hamilton Avenue
Palo Alto, CA, U.S.A. 94301

FAX 1.415.853.0155
Internet address: triadd@decwrl.dec.com

U.S.A/Canada	**Australia**	**France**	**Italy**
1.800.678.OPEN	0014.800.125.388	05.90.2874	1678.19087
Japan	**U.K.**	**Germany**	
0031.12.2363	0800.89.2610	0130.81.1974	

TURBOchannel Technology Transfer Agreement

Grant of Right to Use TURBOchannel Technology

In exchange for your agreeing to the warranty disclaimer and liability limitation stipulated in this Technology Transfer Agreement, Digital Equipment Corporation (Digital) grants at no cost to you a royalty-free nonexclusive license to use TURBOchannel technology (as specified in the *TURBOchannel Specifications*) to design and develop any kind of option board, computer system, or application-specific integrated circuit (ASIC). This Agreement does not grant you any other rights in Digital's patents, copyrights, trade secrets, trademarks, or licenses to TURBOchannel technology. The purchase cost of the TURBOchannel kit is basically the cost to reproduce the materials.

Warranty Disclaimer

The TURBOchannel technology is transferred "as is." Digital expressly disclaims all implied warranties including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Digital does not warrant, guarantee, or make any representations regarding the use of or the results of the use of the specification and related documents in terms of correctness, accuracy, or reliability. Digital believes the documentation is accurate; however, you must assume the risk as to the results and performance of any product you develop that is based on the TURBOchannel technology.

Limits of Liability

You agree that Digital shall not be liable to you under this Agreement for any damages, including without limitation any lost profits or lost savings, or any consequential, incidental, or punitive damages arising out of the use or inability to use the TURBOchannel Hardware Specification and related documents, or for any claim by another party. Your exclusive remedy under this Agreement shall be the furnishing by Digital of the technical support provided herein. You agree to hold Digital harmless for all claims and damages arising from any third party as a result of their use of or inability to use any product you develop based on TURBOchannel technology.

Important Changes to the TURBOchannel Hardware Specification

This version of the *TURBOchannel Hardware Specification* includes all changes made since the original specification was released. Revision bars mark technical changes added since Version 2C of the *TURBOchannel Specifications*.

Changes in Version 2C

- **I/O Read of an Option by the System**: This section has been added and contains a detailed explanation of "I/O read of an option by the system", Figure 1-1.
- **I/O Write to an Option by the System**: This section has been added and contains a detailed explanation of "I/O write to the system by an option", Figure 1-2.
- **DMA**: An explanation of DMA burst has been added to this section.

  The explanation of the use of *~err* has been expanded.
- **DMA Read of the System by an Option**: This section has been added and contains a detailed explanation of Figure 1-6, "One-word DMA read of the system by an option", and Figure 1-7, "Two-word DMA read of the system by an option", and
- **DMA Write to the System by an Option**: This section has been added and contains a detailed explanation of Figure 1-8, "One-word DMA write to the system by an option", and Figure 1-9, "Two-word DMA write to the system by an option".
- **Electrical**: Metric units have been added.
- **Timing**: This section has been expanded.
- **Implementation Notes**: Fahrenheit temperatures have been added.
- **Connector**: An explanation of pin assignment has been added.
- **Figure 1-11**: Metric units and notation that units are in inches (and centimeters) have been added.

Changes in Version 2B

- **Table 1-3, Timing Requirements for TURBOchannel Signals**: Minimum times for *~rdy*, *~conflict*, *~int*, *~rReq*, *~wReq* have been revised to one nanosecond.
- **Implementation Notes**: This section has been updated to clarify that an *~err* signal is valid for an option only when the option has asserted the *~ack* signal.

Changes in Version 2A

- **DMA**: This section has been revised to show that deassertion of *~rReq* and *~wReq* may not occur until *~ack* is asserted by the system.
- **Interrupts**: This section now explains that when the *~int* signal has been asserted, the option may not deassert the signal until software dismisses the interrupt condition.
- **Implementation Notes**: This section has been changed to explain signal deassertion/assertion during reset and when the option is not selected.

Changes in Version 2.0

- **General Description**: A section has been added to explain that systems can time-multiplex multiple option slots onto a single memory port, or dedicate a memory port to each option slot. Note that systems dedicating a memory port to each slot may have clock frequency, clock phase, and slot address space discontinuities between slots.
- **Table 1-1, TURBOchannel Signals**: This table has been revised to show that the *~err* signal is used for all forms of DMA error. The previous description did not list parity errors.
- **I/O Transactions**: This section has been revised to specify a timeout period of at least 10 microseconds. The 5- to 10-microsecond range must be eliminated for compatibility with early TURBOchannel option designs.
- **I/O Transactions**: This section now shows that there may be intervening I/O transactions before a conflicting I/O transaction is reissued. This may occur in systems that do not preserve ordering of processor read and write transactions. For example, if an I/O write from a write buffer is conflicting, a higher priority I/O read, posted after the write, may be issued before the write is reissued. A similar situation is possible when I/O writes are issued before conflicting I/O reads are reissued.
- **I/O Addressing**: This section has been corrected to show 27 address bits.
- **I/O Addressing**: This section now explains that ordering of physical slots in the address space is system-specific, and that there may be address-space gaps between slots. This is necessary for implementations with TURBOchannel slots connected through intermediate adapter logic that may itself require some address space.
- **Figure 1-6, One-word DMA Read of an Option by the System**: The *~ack* signal cannot reassert until t6 to provide at least one deassertion cycle after data burst.
- **DMA**: This section now explains that systems with parity checking enabled assert *~err* if incorrect parity is detected on *ad*.
- **Parity**: A section has been deleted to correct an editorial error.
- **Interrupts**: This section was added to explain that interrupts are level-sensitive and that slot priority is system-specific.
- **Clock**: A section has been added to explain that clock phase and frequency variations may exist between different option slots. This may be the case for systems that connect TURBOchannel slots to an asynchronous memory subsystem through repeater logic, or to a memory subsystem through a crossbar interconnect.
- **Power**: This section now shows the metric values for the airflow specification and that airflow obstructions must be uniformly distributed to avoid creation of airflow dead zones.
- **ROM**: ROM information has been reduced in this specification, and a reference to the *TURBOchannel Firmware Specification* has been added.
- **Mechanical Overview**: This section now explains that physical slot numbering is system-specific.

- **Figure 1-11, Space available for TURBOchannel modules inside a DECstation/DECsystem 5000 Model 200**: Airflow information has been changed to show that it may be from either left-to-right or right-to-left. The bulkhead specification has been changed to show that its dimensions are the maximum usable connector area.

Conventions Used in This Specification

The Hardware Specification uses these conventions:

- *Terms in italic type like this* show TURBOchannel signals.
- Square brackets ([ ]) surround address ranges (expressed in bits).
- A tilde (~) precedes signals that are active-low.
- Signals not preceded by a tilde are active-high.

Contents

1 TURBOchannel Hardware Specification

Glossary

Figures

Tables

1

TURBOchannel Hardware Specification

General Description

The TURBOchannel is a synchronous, asymmetrical I/O channel. The beginning of a TURBOchannel cycle is defined by the rising edge of the channel clock signal (*clk*); all signals are specified with respect to that clock edge. A TURBOchannel can be operated at any fixed frequency in the range of 12.5 to 25 MHz.

The TURBOchannel is asymmetrical: One system module and some number of option modules connect to the TURBOchannel. Typically, the system module contains the main memory system and the processor, and the option modules contain controllers for peripheral devices.

Two kinds of transactions are permitted on the TURBOchannel:

- The system module can read or write to an option module; this is an an I/O transaction.
- An option module can read or write to the system module; this is a DMA transaction.

An option module cannot address another option module on the TURBOchannel.

Systems can time-multiplex multiple option slots onto a single memory port to share its bandwidth, or dedicate a memory port to each option slot.

Signals

Table 1-1 lists the source and function of each signal on the TURBOchannel. Signals preceded by a tilde (~) are active-low. Signals not preceded by a tilde are active-high.

The meaning of the *ad* lines depends on the type of transaction and the transaction phase (either address or data).

Table 1-1. TURBOchannel Signals

Signal Name	Source	Function
ad[P, 31..0]	bussed	Address/data bus
~sel	system	I/O read/write select
~write	system	I/O read/write specifier
~ack	system	DMA read/write acknowledge
~err	system	DMA error
~reset	system	System reset
clk	system	Channel clock
~rdy	option	I/O read/write ready
~conflict	option	I/O read/write conflict
~rReq	option	DMA read request
~wReq	option	DMA write request
~int	option	I/O interrupt

Transactions

The minimum length of a TURBOchannel transaction is two clock cycles. Some system implementations may be able to achieve back-to-back transactions. Other implementations may insert idle cycles between transactions. The only exception is that back-to-back I/O transactions to the same option must have an idle cycle in between, guaranteeing the deassertion of select to that option.

Idle Cycles

During idle channel cycles, the system drives the *ad* lines.

I/O Transactions

Processor load/store instructions to the I/O slot address range generate I/O read/write transactions. As the system drives the address onto the TURBOchannel, the system decodes the address and asserts a select signal (*~sel*) for the specified option slot.

For read transactions, the system waits for the slot to drive data onto the TURBOchannel and assert the ready signal (*~rdy*). Figure 1-1 (page 4) shows signal activity when a system performs an I/O read of an option. The section "I/O Read of an Option by the System" (page 5) contains a detailed explanation of Figure 1-1.

For write transactions, the system drives data onto the TURBOchannel until the option asserts the ready signal. Figure 1-2 (page 6) shows signal activity when a system performs an I/O write to an option. The section "I/O Write to an Option by the System" (page 7) contains a detailed explanation of Figure 1-2.

Options should minimize ready assertion latency to maximize channel utilization. If the option does not respond within the system-specified timeout period, the system aborts the transaction. The timeout period must be listed in the system guide and must be at least 10 microseconds.

The system could select an option for an I/O transaction when the option is already committed to a DMA transaction. The option can

- Assert the ~*conflict* signal.

  Because the ~*conflict* signal affects system performance and interrupt service latency, this signal should be used only when there is no other means of breaking a deadlock. The ~*conflict* signal should be asserted with ~*rdy* and is used only when the option is selected.
- Not respond and let the system time out. The system could retry the I/O transaction at a future time. Other I/O transactions can occur before a conflicting I/O transaction is reissued.

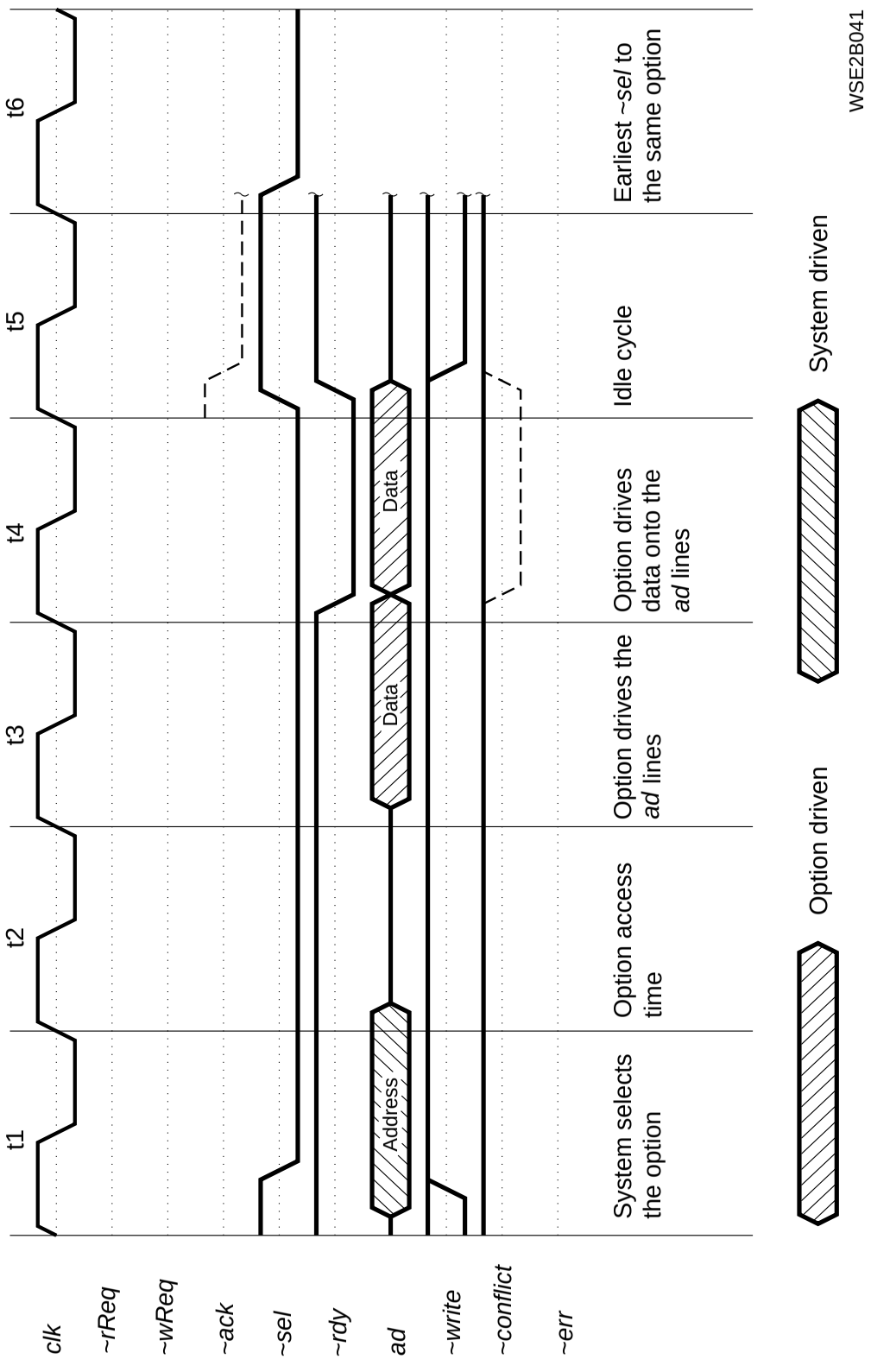

Figure 1-1. I/O read of an option by the system

I/O Read of an Option by the System

This section contains a detailed explanation of Figure 1-1.

- **Cycle t1**

  The system selects the option. The system

  - Drives an address onto the *ad* lines.
  - Decodes the address and asserts *~sel* to the I/O option controller.
  - Deasserts *~write* to signify a read function.

- **Cycle t2**

  Option access time. The option

  - Stores the last snapshot of the address bits to registers and begins to decode.
  - Begins retrieving data at the address that was specified.

- **Cycle t3**

  This cycle does not exist if the option is fast enough to retrieve the data and begin driving it onto the *ad* lines immediately.

  If the cycle exists, it repeats until one of the following conditions occurs:

  - The option is ready with data.
  - A system-defined timeout takes place and the system aborts the transaction (t3 cycles repeat until system timeout).

- **Cycle t4**

  If valid data is available during this cycle, the option

  - Asserts *~rdy*.
  - Drives valid data onto the *ad* lines.

  If the option is already committed to a DMA transaction, the option could assert *~conflict*.

- **Cycle t5**

  This is an idle cycle. The system

  - Recognizes assertion of the *~rdy* signal.
  - Stores the last snapshot of the data from the *ad* lines to registers.
  - Deasserts *~sel*.
  - Asserts *~ack* if a DMA request by the option is pending.

  The option

  - Stops driving the *ad* lines.
  - Deasserts *~rdy*.

- **Cycle t6**

  This is the earliest time a new I/O *~sel* transaction to the same option can begin.

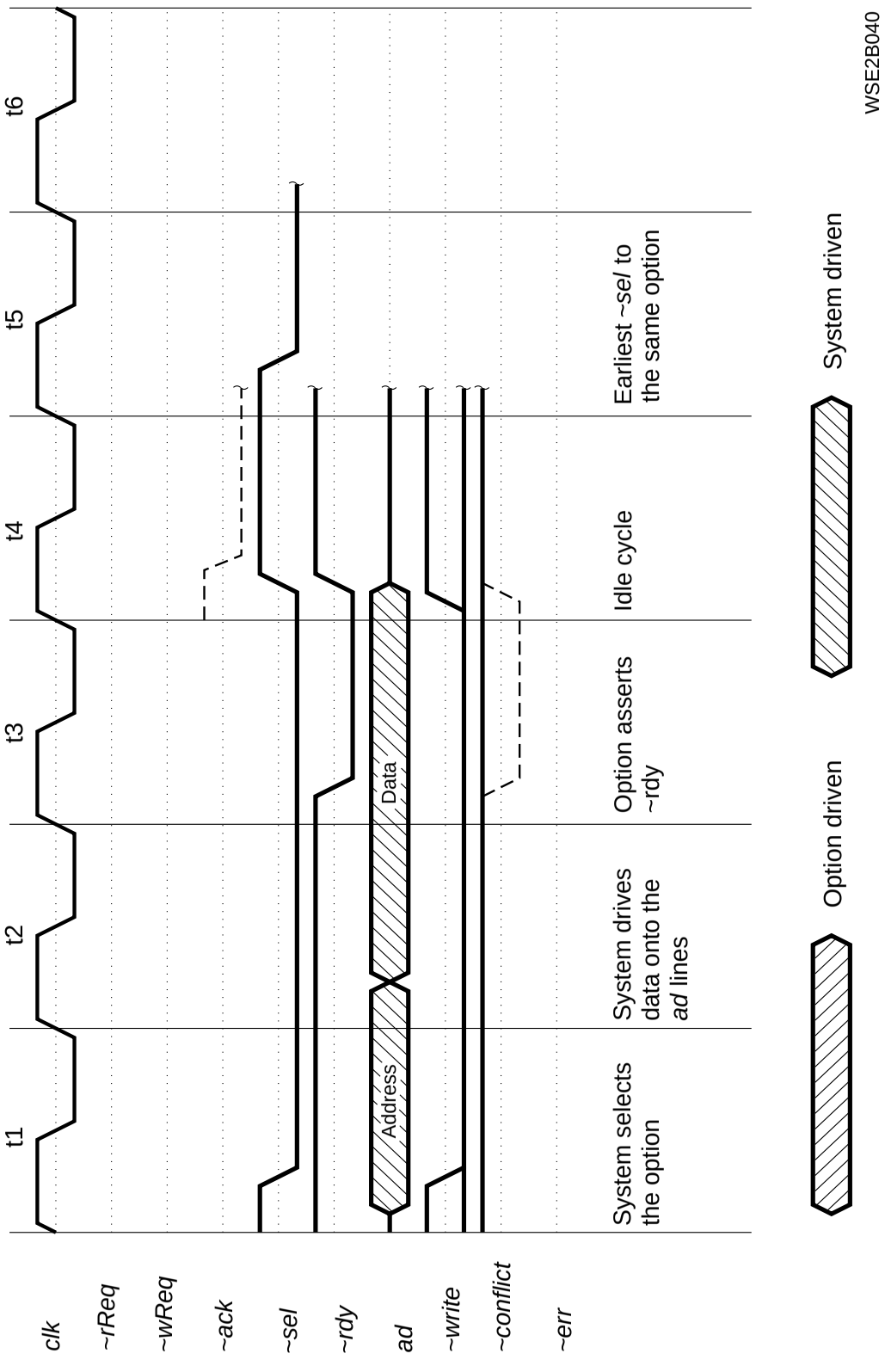

Figure 1-2. I/O write to an option by the system

I/O Write to an Option by the System

This section contains a detailed explanation of Figure 1-2.

- **Cycle t1**

  The system selects the option. The system

  - Drives the address onto the *ad* lines. Bits 1, 2, 3, and 4 of the address form a byte mask; each bit indicates data validity in the corresponding byte lanes of the 32 bit data word to follow.
  - Decodes the address and asserts *~sel* to the option.
  - Asserts *~write* to signal a write function.

- **Cycle t2**

  The system drives data onto the *ad* lines.

  - Because *~sel* was asserted, the option stores the last snapshot of the address to registers and begins to decode.
  - Because *~write* was asserted, the option prepares to receive data and store it at the address that was specified.

  If the option is not fast enough, this cycle continues until the option is ready or until a system-defined timeout, at which point the system aborts the transaction (t2 cycles repeat until system timeout).

  If the option is fast enough to prepare to receive data just before the next rising edge, actions shown for cycle t4 happen in cycle t3.

- **Cycle t3**

  The option

  - Asserts *~rdy*.
  - Stores the last snapshot of the data from the *ad* lines to registers.

  If the option is already committed to a DMA transaction, the option could assert *~conflict*.

- **Cycle t4**

  This is an idle cycle. The system, having recognized the assertion of the *~rdy* signal

  - Stops driving the data onto the *ad* lines.
  - Deasserts *~sel*.
  - Deasserts *~write*.
  - May assert *~ack* for a pending DMA request by this option.

  The option deasserts *~rdy*.

- **Cycle t5**

  This is the earliest time a new I/O *~sel* transaction to the same option can begin.

I/O Addressing

Each slot has a 4- to 512-megabyte address range for I/O transactions. The size of the slot address space is system-specific in power-of-two multiples. For systems with less than 512-megabyte spaces, the high-order address bits are undefined and must be treated as *don't care* values. Options should use the minimal number of low-order address bits to decode internal logic. Addresses are 27-bit word addresses; therefore, the least significant two address bits of the byte address are implicitly 0.

Figure 1-3 shows the interpretation of the *ad* signals during an I/O read address cycle. Bits *ad*[31..5] specify the word address within the slot space. The option responds by driving the addressed word onto *ad*.

Figure 1-4 shows the interpretation of the *ad* signals during an I/O write address cycle. Bits *ad*[31..5] specify the word address within the slot space. Bits *ad*[4..1] specify byte masks for the *ad* signals during the subsequent data cycle of the transaction. If a byte mask bit is 1, the corresponding byte lane is not stored in the addressed word.

The number of slots, size of the slot space, and base address of the slots is system-specific. The order of physical option slots in the address space is also system-specific. The address space between adjacent physical option slots may have system-specific gaps. The address space must be documented in the system guide.

DMA

All words in TURBOchannel DMA transactions are 32 bits long. A DMA transaction can be any length up to an implementation-defined limit, which must be at least 64 words and a power-of-two multiple. The flow of two or more data words, one after the other, is called a DMA burst.

To obtain the best possible performance, data words are transmitted one per cycle. When the first DMA read data word is available, subsequent data words of the block are available (and must be accepted) in every cycle. DMA write data words are accepted (and must be supplied) in every cycle.

Request arbitration for DMA transactions is implementation-specific. Fixed-priority schemes will allow a specified slot to achieve full bandwidth; other schemes (such as fair service) are also allowed. The system guide must describe the arbitration scheme used.

An option requests a DMA transaction by asserting either the *~rReq* signal or the *~wReq* signal; these signals must not be asserted simultaneously. When the *~rReq* or *~wReq* signal has been asserted, the option may not deassert the signal until the system grants service by asserting the *~ack* signal. The option indicates the block length by continuing to assert *~rReq* or *~wReq* for as many cycles after *~ack* as the number of data words.

DMA transaction addresses have a 16-gigabyte address space (34 address bits). Bits *ad*[4..0] are the upper five address bits. Bits *ad*[31..5] contain the rest of the word address. The two low-order address bits are implicitly 0. A system does not always support the entire address space; the consequences of addressing nonexistent memory in a DMA transaction is implementation-dependent. Figure 1-5 (page 9) shows a DMA transaction *ad* signal.

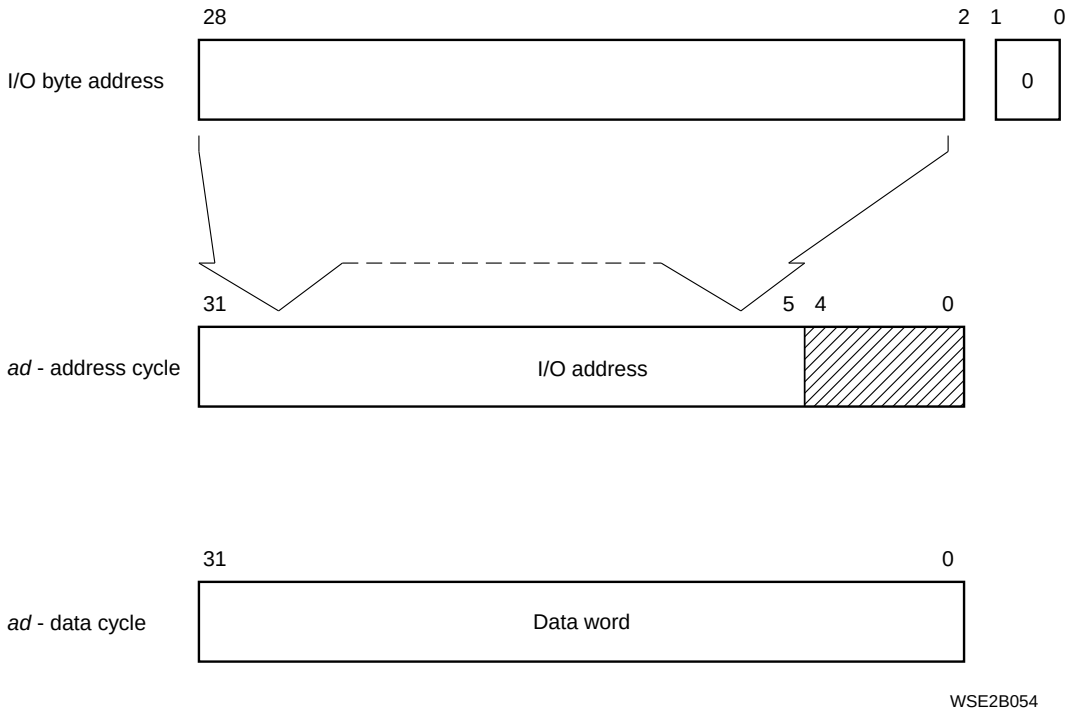

Figure 1-3. Interpretation of the *ad* signal during an I/O read

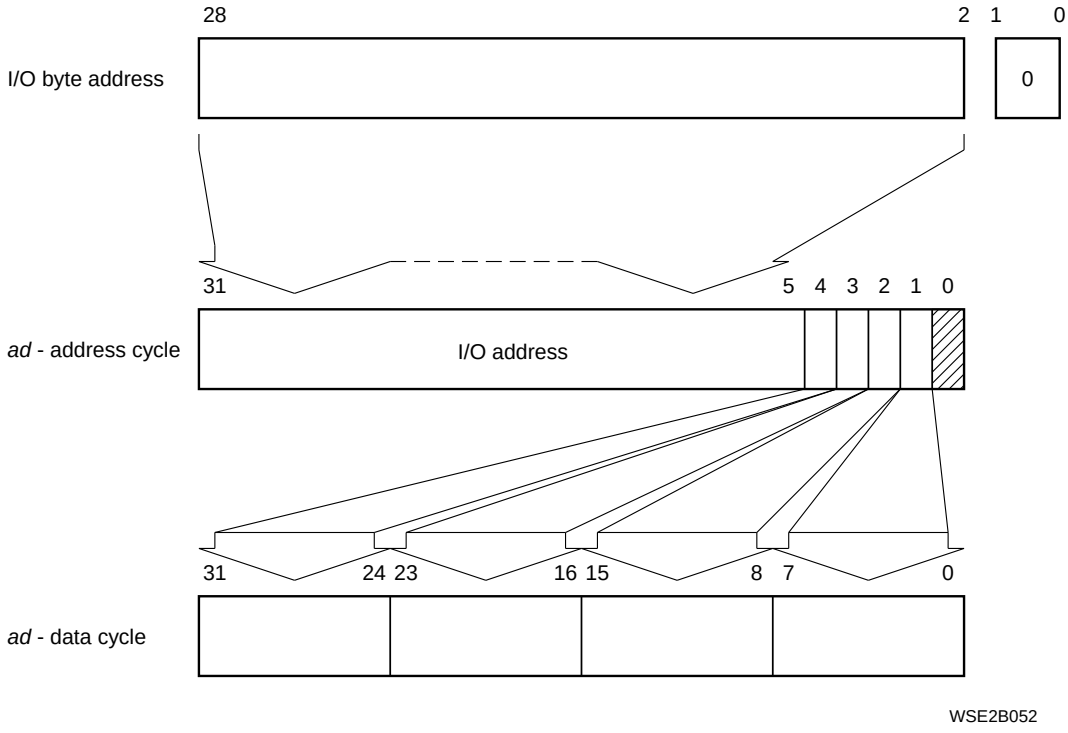

Figure 1-4. Interpretation of the *ad* signal during an I/O write

During the data cycle of a DMA transaction TURBOchannel transmits a word of data in every channel cycle; this high level of performance may require some implementations to use page-mode accesses to the system memory dynamic RAMs. DMA transactions cannot cross 2048-byte address boundaries.

The system asserts the *~err* signal when the system cannot successfully complete the requested DMA transaction. For example

- When an uncorrectable memory error occurs during a DMA read transaction; the system also asserts the uncorrectable data.
- When a memory error occurs during a DMA write transaction.
- When an option requests more than the maximum block size for a DMA read or write transaction.
- When parity checking is enabled and the system detects incorrect parity on the *ad* signals.

The option must terminate its DMA transaction on the cycle immediately after the *~err* signal is asserted.

The *~ack* signal can remain asserted for several cycles when DMA read errors occur (for example, when memory is pipelined). Option logic must be designed so that an *~ack* signal from an aborted transfer cannot be interpreted as an *~ack* signal for a subsequent DMA request.

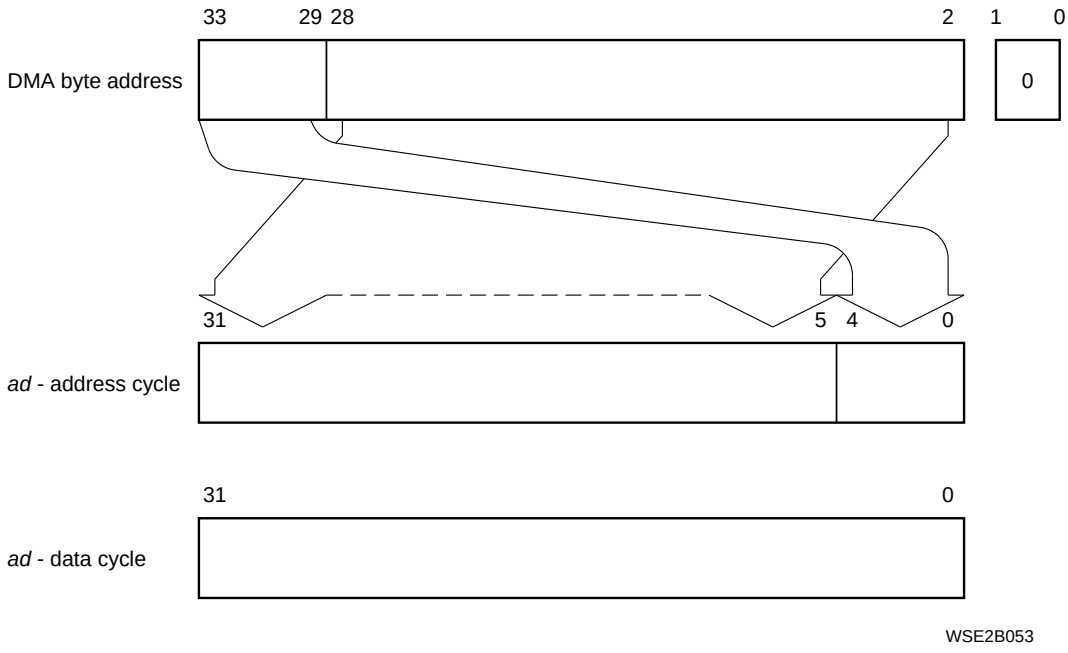

Figure 1-5. Interpretation of the *ad* signal during a DMA transaction

Figures 1-6 and 1-7 (pages 12 and 13) show signal activity when an option performs a one- or two-word DMA read of the system. The section "DMA Read to an Option by the System" (page 16) contains a detailed explanation of Figures 1-6 and 1-7. Figures 1-8 and 1-9 (pages 14 and 15) show signal activity when an option performs a one- or two-word DMA write to the system. The section "DMA Write to an Option by the System" (page 17) contains a detailed explanation of Figures 1-8 and 1-9.

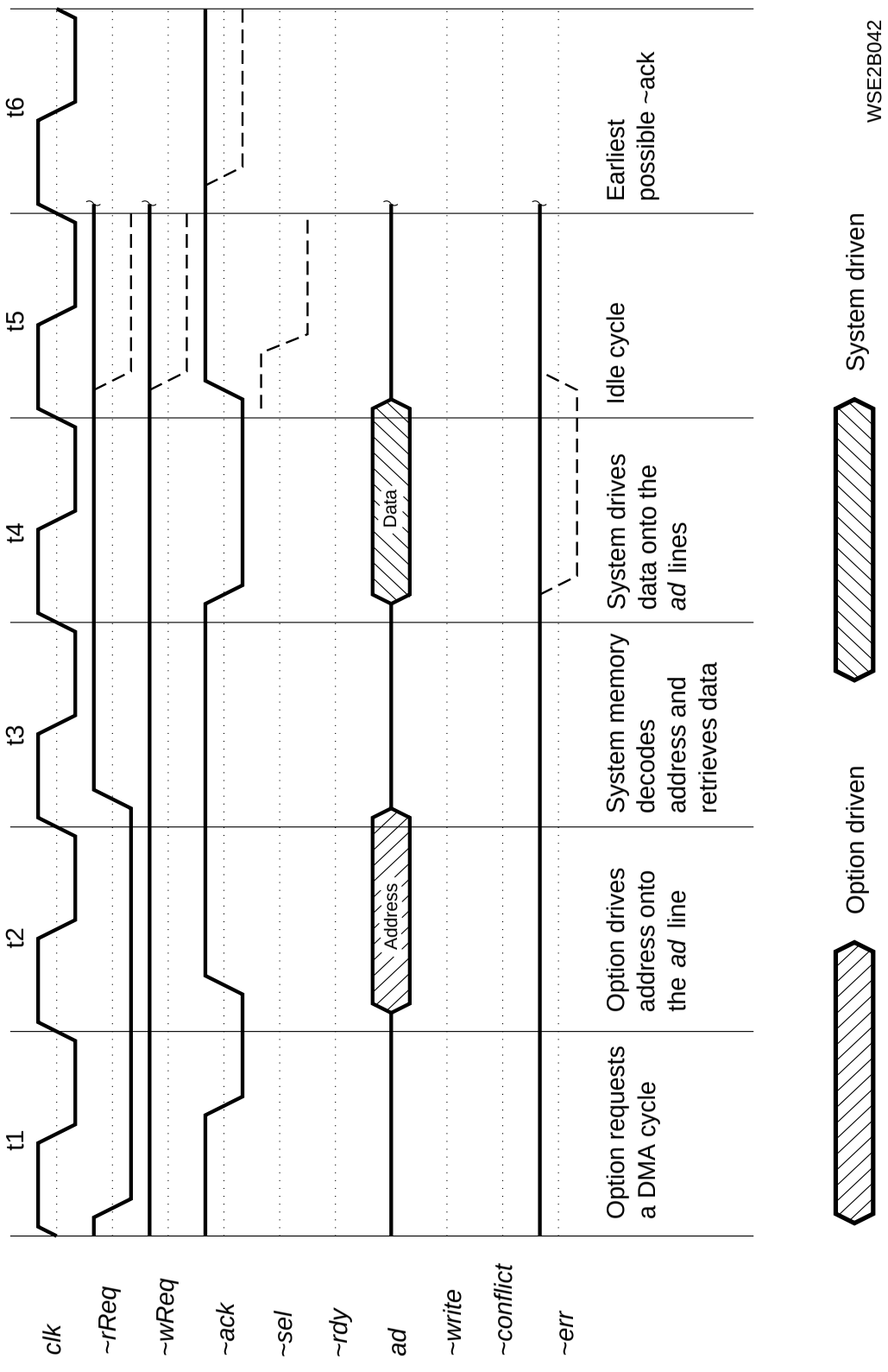

Figure 1-6. One-word DMA read of the system by an option

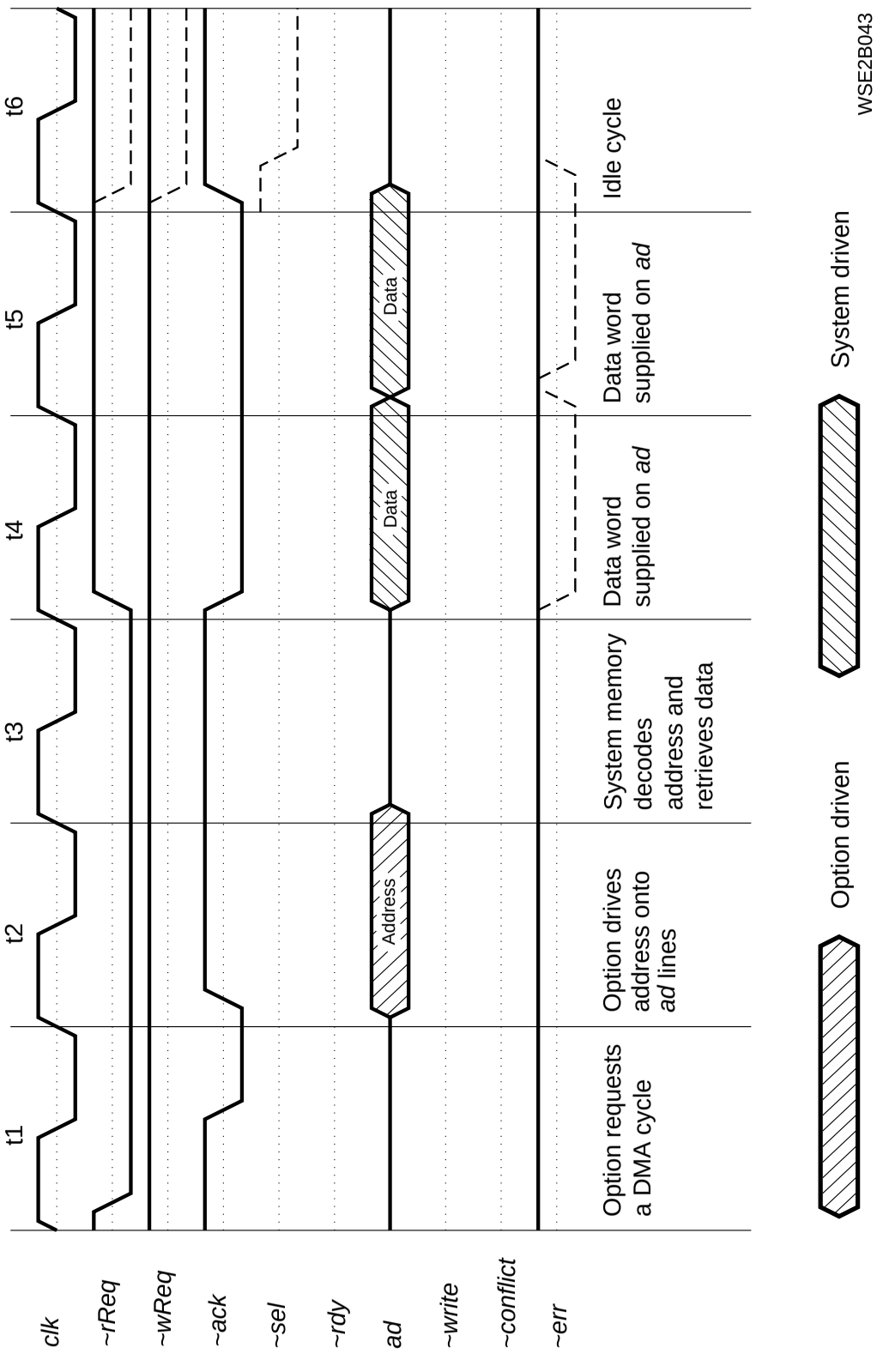

Figure 1-7. Two-word DMA read of the system by an option

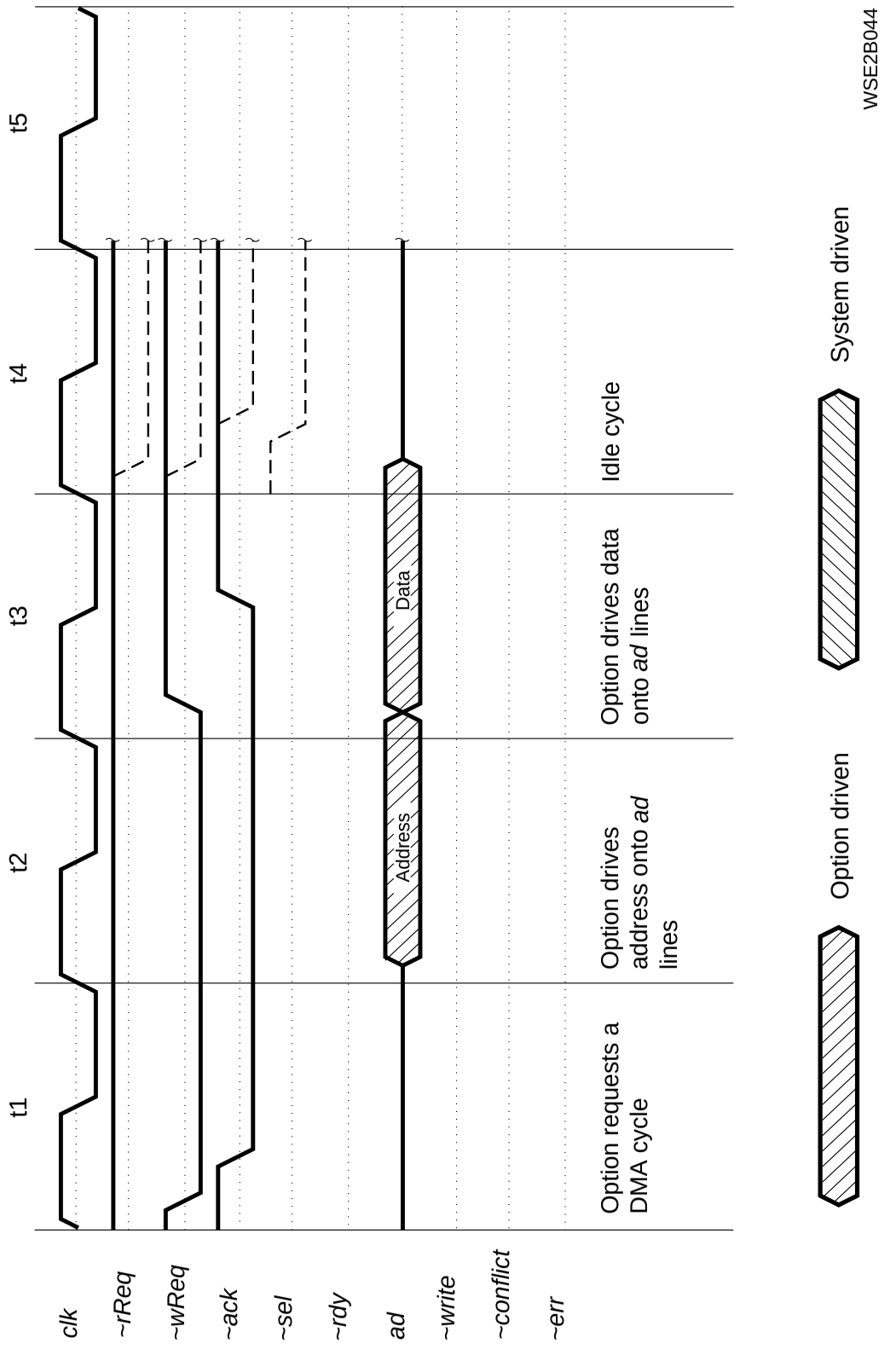

Figure 1-8. One-word DMA write to the system by an option

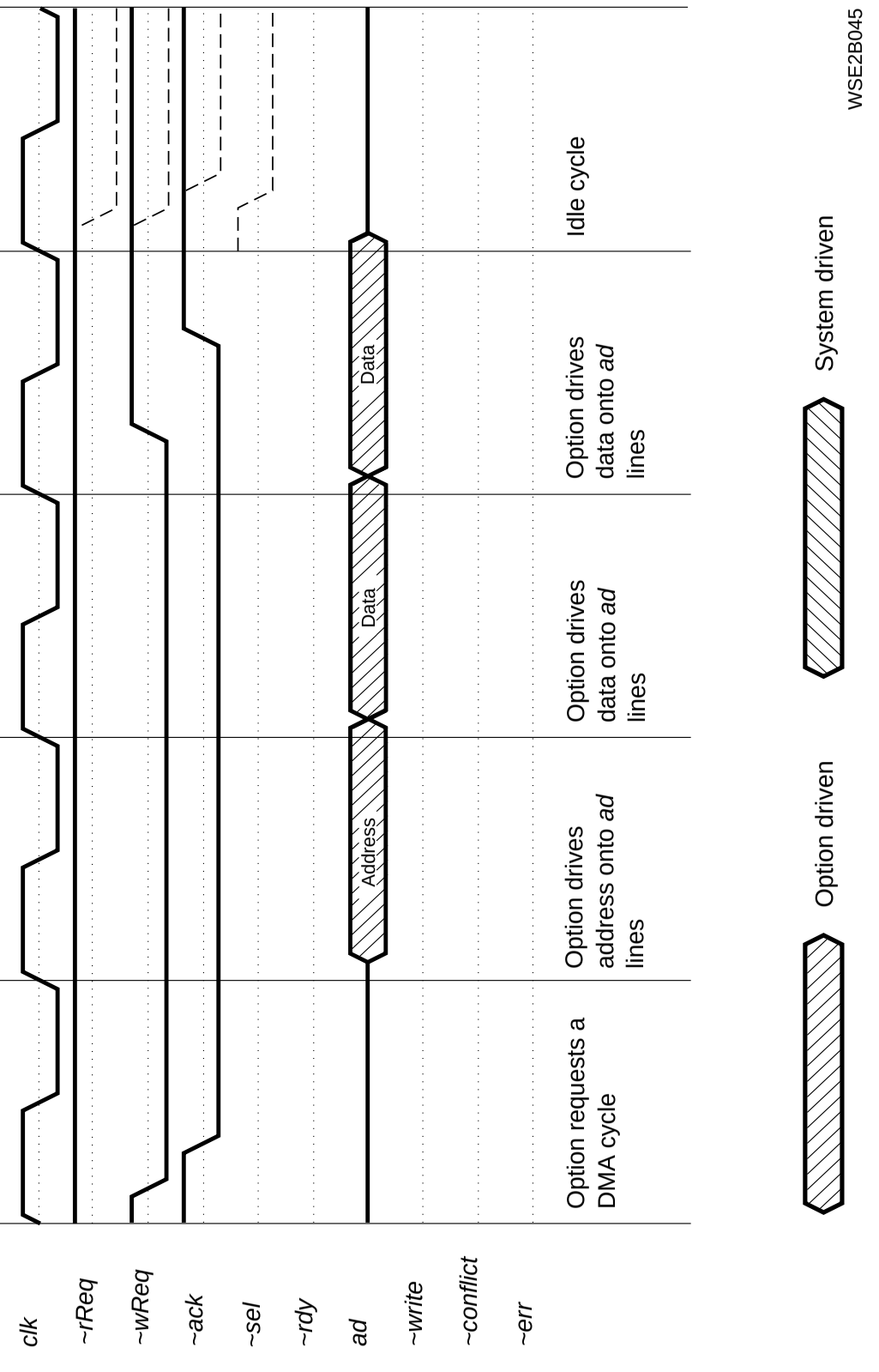

Figure 1-9. Two-word DMA write to the system by an option

DMA Read of the System by an Option

This section contains a detailed explanation of Figures 1-6 and 1-7.

- **Cycle t1**
  - The option asserts ~*rReq* and waits for ~*ack* from the system.
  - The system senses ~*rReq* and could react quickly enough to assert ~*ack* within the same cycle, or could choose to wait many cycles before asserting ~*ack*.

  Once an option asserts a DMA request, either ~*rReq* or ~*wReq*, the signal must remain asserted until the system responds. An option must never assert ~*rReq* and ~*wReq* simultaneously.
- **Cycle t2**
  - The option drives the address onto the ~*ad* lines in response to ~*ack* from the system.
  - The system deasserts ~*ack*.
  - The option continues to assert ~*rReq* for as many cycles after ~*ack* was sensed as the number of words of data that are being requested.
- **Cycle t3**
  - System memory decodes the address and retrieves the data. The number of cycles required for this decode and retrieval of data depends upon the speed of system memory, also know as memory access latency (this may be many t3 cycles).
  - If only one word of data is being requested, the option deasserts ~*rReq*.
- **Cycle t4**

  The system
  - Drives the first data word onto the *ad* lines.
  - Reasserts ~*ack*.
  - Asserts ~*err* if an uncorrectable error occurs.
- **Cycle t5**

  The option
  - Continues to sense ~*ack* and note the bits on the *ad* lines.
  - Stores a snapshot of the data to registers in response to the assertion of ~*ack*.

  If a single word of data was requested, this is an idle cycle:
  - The system deasserts ~*ack*.
  - This is the earliest time that the option could assert ~*rReq* or ~*wReq*.
  - This is the earliest possible time that the system could assert ~*ack*.

  If two or more words of data were requested, the system
  - Drives them onto the *ad* lines one word per cycle and continues to assert ~*ack* for every cycle of data, one 32-bit word per cycle.
  - Asserts ~*err* if an uncorrectable error occurs.

The system must be able to place data words onto the bus, and the option must be able to accept those words every cycle in succession.

- **Cycle t6**

  If a single word of data was requested, this is the earliest time that the system can assert ~*ack*.

  If two words of data were requested, this is an idle cycle

  – The system deasserts ~*ack*.

  – This is the earliest time that the option could assert ~*rReq* or ~*wReq*.

  – This is the earliest possible time that the system could assert ~*ack*.

DMA Write to the System by an Option

This section contains a detailed explanation of Figures 1-8 and 1-9.

- **Cycle t1**

  – The option asserts ~*wReq* and starts the wait for ~*ack* from the system.

  – The system could assert ~*ack* within the same cycle, or could choose to wait many cycles before asserting ~*ack*.

  Once an option asserts a DMA request, either ~*rReq* or ~*wReq*, the signal must remain asserted until the system responds. An option must never assert ~*rReq* and ~*wReq* simultaneously.

- **Cycle t2**

  The option

  – Drives the address onto the ~*ad* lines in response to ~*ack* from the system.

  – Continues to assert ~*wReq* for as many cycles after ~*ack* was sensed as the number of words of data that are being requested.

- **Cycle t3**

  – System memory decodes the address and prepares to accept data.

  – The option drives the first data word onto the *ad* lines.

  If only one word of data is to be written to memory,

  – The option deasserts ~*wReq*.

  – The system senses ~*wReq*; if ~*wReq* is deasserted, the system deasserts ~*ack*.

  If more than one word of data is to be written to memory (a DMA burst),

  – The option continues to assert ~*wReq*.

  – The system continues to assert ~*ack*.

- **Cycle t4**

  The system reads the data on the *ad* lines.

  If the write transaction is a two-word write,

  – The option deasserts ~*wReq*.

- The system senses ~*wReq*; if ~*wReq* is deasserted, the system deasserts ~*ack*.

If the write transaction is a single word or if this is the final write cycle of a DMA burst, this is an idle cycle:

- The option can assert ~*rReq* or ~*wReq* and start a new transaction.
- The system can assert ~*sel* or ~*ack* in response to the ~*rReq* or ~*wReq* from the option.

- **Cycle t5**

  If the write transaction is a two-word write, this is an idle cycle:

  - The option can assert ~*rReq* or ~*wReq* and start a new transaction.
  - The system can assert ~*sel* or ~*ack* in response to the ~*rReq* or ~*wReq* from the option.

Parity

Options may implement odd-word parity for *ad* signals on the *ad[P]* signal. Options that implement parity must provide the ability to enable and disable parity under software control. Parity checking must be disabled by the assertion of the ~*reset* signal.

When parity checking is enabled, the option must check parity when accepting:

- An I/O read address
- An I/O write address
- I/O write data
- DMA read data

If the option detects a parity error, the option should record the error and initiate higher level error notification.

System implementations that support parity should be designed to handle simultaneously options that have parity enabled and options that do not implement parity.

Interrupts

Interrupts are level-sensitive. Slot priority is determined by system hardware and software. When the option has asserted ~*int*, the signal may not be deasserted until software dismisses the interrupt condition.

Electrical

For bussed signals (*ad*[P, 31..0]) on a TURBOchannel the system module and option modules must present a total capacitive load of no more than 180 pF. The total trace length for a bussed signal on the system module cannot exceed 16 inches (40.64 centimeters). An option module must present a capacitive load of no more than 20 pF for any bussed signal. The total trace length for a bussed signal on an option module cannot exceed 2 inches (5.08 centimeters). Capacitance measurements must include all connectors, components, lands, and traces.

The load imposed by an option module for system-generated control signals cannot exceed 50 pF, except for the clock signal, which is 100 pF. This must include the connector, all components, lands, and traces.

For option-generated control signals, the load imposed by the system module cannot exceed 65 pF including the connector, all components, lands, and traces.

An option that is more than one slot wide must connect all signals to one slot. However, the option can draw power from all connectors. This slot is identified in the *TURBOchannel Mechanical Drawings*.

Table 1-2 lists the DC parameters for TURBOchannel signals.

Table 1-2. DC Parameters for TURBOchannel Signals

Value	*ad* (P,31..0)	System-generated	Option-generated
Voh	2.4 V (minimum)	same	same
Vih	2.0 V (minimum)	same	same
Vil	0.8 V (maximum)	same	same
Vol	0.5 V (maximum)	same	same
Iih	+70 uA (maximum)	+0.5 mA (maximum)	+0.5 mA (maximum)
Iil	-70 uA (maximum)	-1.5 mA (maximum)	-1.5 mA (maximum)
Ioh	-1.0 mA (minimum)	same	same
Iol	+2.0 mA (minimum)	same	same

Timing

Setup and hold signal timing specifications for system-generated signals must be met by each option in the system. Signal timing specifications for option-driven signals are measured with full capacitive loading on all signals. All signal timing specifications are measured from the rising edge of the *clk* signal at the option.

Table 1-3 lists timing requirements for each TURBOchannel signal. All times are in nanoseconds.

Table 1-3. Timing Requirements for TURBOchannel Signals

Signal	Source	Minimum	Maximum	Setup	Hold
ad	system			5	2
ad to tristate	system	3	22		
~sel, ~write, ~ack, ~err, ~reset	system			13	2
ad	option	3	34		
ad to tristate	option	3	22		
~rdy, ~conflict, ~int	option	1	12		
~rReq, ~wReq	option	1	7		

System-driven signal voltages are expected to be stable and valid from the setup through the hold time shown in Table 1-3. For system-driven signals, setup is the time before the rising edge of *clk*, hold is the time after the same rising edge of *clk*.

Option-driven signal voltages are expected to be stable and valid within the timing limits shown in Table 1-3. Minimum and maximum times are measured from the rising edge of *clk*.

Clock

The TURBOchannel uses a free-running clock (*clk*) at any fixed frequency from 12.5 to 25 MHz. Clock signal rise and fall times must not exceed 5 nanoseconds measured at the *clk* signal pin on the option connector. The clock signal must be high for at least 15 nanoseconds and low for at least 15 nanoseconds.

Figure 1-10 shows timing requirements for each phase of the *clk* signal.

Clock skew between the system module and option modules must be controlled; the system module must function correctly for option modules that meet timing specifications.

Systems can have option slots with different clock phase and frequency variations.

For an option module's *clk* signal, the module must have a diode terminator connected to ground and a diode terminator connected to the +5 volt supply.

Power

A TURBOchannel system module provides two supply voltages to each TURBOchannel option: +5 volt and +12 volt. Sequencing of the +5 volt and +12 volt supplies is not guaranteed.

Table 1-4 describes the voltage and current available to a TURBOchannel option.

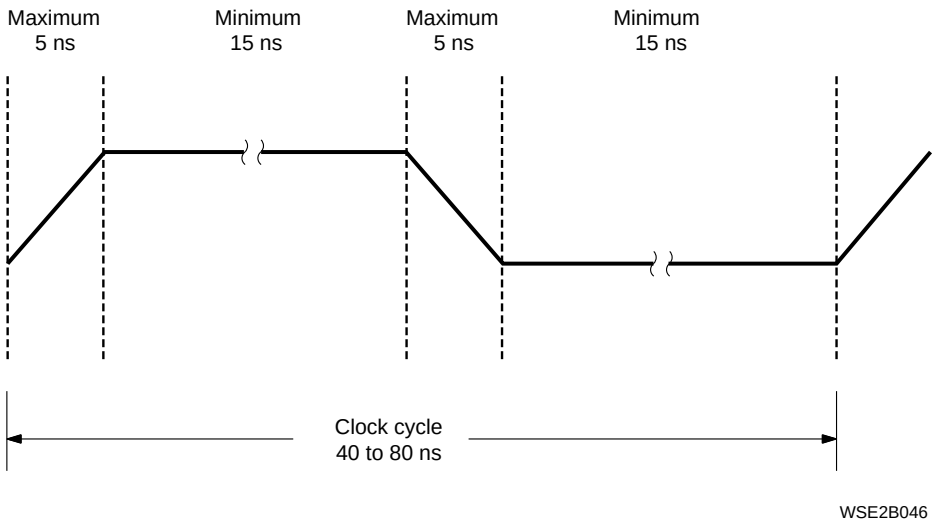

Figure 1-10. Timing requirements for the *clk* signal

Table 1-4. Power Available to TURBOchannel Options

	Maximum Current		
Voltage	**Single-Width Option**	**Double-Width Option**	**Triple-Width Option**
+5 V ±5%	4.0 A	8.0 A	12.0 A
+12 V ±5%	0.5 A	1.0 A	1.5 A

The *~reset* signal is asserted for at least 250 milliseconds after power is switched on and the +5 volt supply has become stable. The *~reset* signal is reasserted at least 500 microseconds before the +5 volt supply drops.

The system will maintain an airflow of at least 50 LFM (linear feet per minute) (25 cm/s (centimeters per second)) below an option module. The system will maintain an airflow of at least 150 LFM (76 cm/s) above an option module. A module's components or daughter cards must not obstruct more than 50 percent of the side-to-side cross-sectional area above the module. Module obstructions must be uniformly distributed so there are no downstream airflow dead zones.

Implementation Notes

During reset, an option must

- Deassert the *~rdy*, *~conflict*, *~rReq*, *~wReq*, and *~int* signals.
- Tristate the *ad* signals.

When an option is not selected, the option must not assert the *~rdy* or *~conflict* signals. Thus, a system need not qualify these signals with the option select line.

Table 1-5 shows which features an option must implement to connect to the corresponding signals:

Table 1-5. Option Implementation Required for Signal Connection

Features	Signal
parity	*ad*[P]
I/O conflicts	*~conflict*
DMA	*~wReq*, *~rReq*, *~ack*, *~err*
interrupts	*~int*

Options that do not have 32-bit internal data paths must still drive all 32 *ad* signals when supplying I/O read data.

Options that cannot address the entire 16-gigabyte DMA address space must drive 0 onto high-order address signals.

Logic families selected to implement the system and option interfaces should use the slowest logic that meets the timing requirements of the TURBOchannel. Logic families should also use internal clamping to control signal reflections. Sufficient capacitance should be used to decouple the supply voltages on system and option modules.

Options should be designed

- To meet all specifications while operating in office environments at 10° to 40°C (50° to 104°F) ambient temperature with a 10°C (18°F) internal rise and 10% to 90% relative humidity.
- To meet appropriate international Class A electromagnetic interference regulations and international safety regulations.
- With appropriate electrostatic discharge protection for their application and for routine shipping and installation handling.

Systems that time-multiplex option slots onto a single memory port may distribute common *~write* and *~err* signals to multiple slots. Options can observe transitions of these signals during transactions to other option slots. The *~write* signal is only valid for an option when the option's *~sel* signal is asserted. The *~err* signal is only valid for an option when the option's *~ack* signal is asserted.

Connector

The TURBOchannel uses a 96-pin DIN connector. The system module uses female connectors; option modules use male connectors. Suggested connectors are the AMP 532504-1 (female) and the AMP 532523-1 (male). Options must not connect to the four NC pins. Table 1-6 lists pin assignments for a TURBOchannel connector.

Note that row numbering for TURBOchannel connector pins does not correspond to the row numbering for the suggested connector. Column assignments do correlate. Pins A1 through A32, B1 through B32, and C1 through C32 for the suggested connector correspond to TURBOchannel pins A32 through A1, B32 through B1, and C32 through C1, respectively.

Table 1-6. TURBOchannel Connector Pin Assignments

Row	Column A	Column B	Column C
1	*ad*[31]	GND	*ad*[30]
2	*ad*[29]	GND	*ad*[28]
3	*ad*[27]	GND	*ad*[26]
4	*ad*[25]	GND	*ad*[24]
5	+5 V	GND	+5 V
6	*ad*[23]	GND	*ad*[22]
7	*ad*[21]	GND	*ad*[20]
8	*ad*[19]	GND	*ad*[18]
9	*ad*[17]	GND	*ad*[16]
10	+5 V	GND	+5 V
11	*ad*[15]	GND	*ad*[14]
12	*ad*[13]	GND	*ad*[12]
13	*ad*[11]	GND	*ad*[10]
14	*ad*[9]	GND	*ad*[8]
15	+5 V	GND	+5 V
16	*ad*[7]	GND	*ad*[6]
17	*ad*[5]	GND	*ad*[4]
18	*ad*[3]	GND	*ad*[2]
19	*ad*[1]	GND	*ad*[0]
20	+5 V	GND	+5 V
21	NC	GND	NC
22	NC	GND	NC
23	*~conflict*	GND	*ad*[P]
24	*~ack*	GND	*~err*
25	+5 V	GND	+12 V
26	*~rReq*	GND	*~wReq*
27	†	GND	*~write*
28	*~sel*	GND	*~rdy*
29	*~int*	GND	*~reset*
30	GND	GND	GND
31	GND	*clk*	GND
32	GND	GND	GND

†Row 27 column A is assigned to the *~iaCE* signal in prototype implementations. This signal is similar to *~sel* but is only asserted for the first cycle of the I/O transaction. Option modules should not use this signal.

ROM

A TURBOchannel option module ROM must contain information about itself and the identity of the option. The ROM may also contain additional option firmware. ROM requirements are defined in the *TURBOchannel Firmware Specification*.

Mechanical Overview

Figure 1-11 shows the space available to a TURBOchannel option module inside a DECstation/DECsystem 5000 Model 200 system unit. Refer to *TURBOchannel Mechanical Drawings* for more detailed mechanical drawings. Physical option slot numbering is system-specific and must be documented in the system guide.

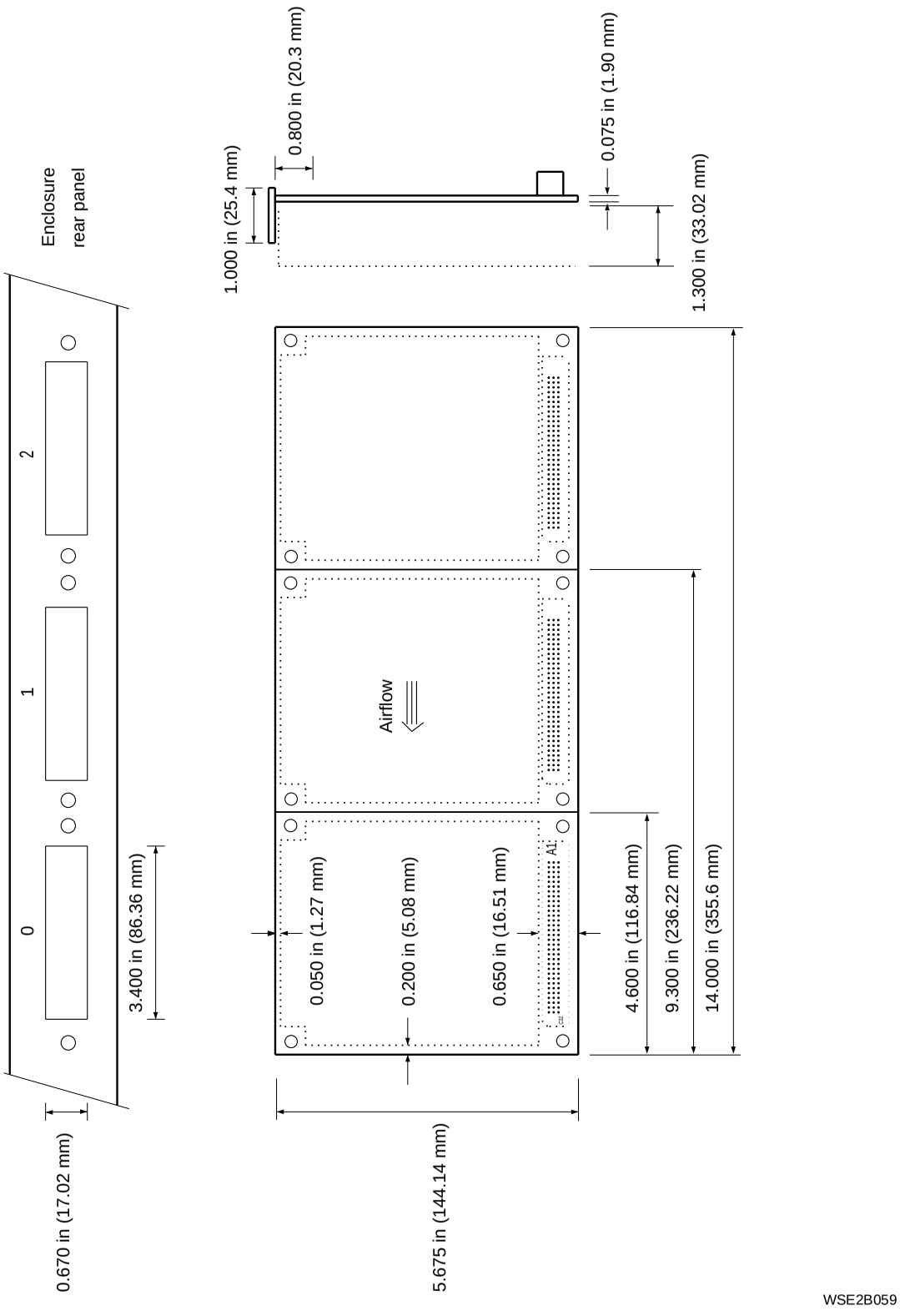

Figure 1-11. Space available for TURBOchannel modules inside a DECstation/DECsystem 5000 Model 200

Glossary

ASIC

Application-Specific Integrated Circuit

Bootstrap

A procedure or device that loads a program into memory from an input device. TURBOchannel bootstraps are run by REX.

Boot Module

A device, such as a disk controller module or an Ethernet controller module, used as a bootstrap device.

Call prototype

The call interface definition for a routine, expressed in ANSI C.

Callback vector

A vector (1-dimensional array) of pointers to routines available to option module ROM firmware.

Dead Zone

An area inside the system module that has restricted airflow.

DMA

Direct Memory Access

DMA burst

The flow of two or more data words one after the other during a DMA transaction.

DMA Transaction

An option module read of or write to system memory.

"Don't care" value

A Boolean value that can be one or zero.

EMC

Electromagnetic Compatibility

EMI

Electromagnetic Interference

Firmware

Software that is stored in ROM.

FRU

Field Replaceable Unit

Integral Options

Components that logically appear to be separate TURBOchannel options but are actually part of the system module.

I/O Transaction

A system module read of or write to an option module.

Land

A metal pad on a PC board where a wire or connector is attached

MER

Memory error register

NC pin

No Connect pin

Option Module

A TURBOchannel option not integral to the system. May contain a controller for or interface to peripheral devices.

REX

ROM Executive

RAM

Random Access Memory

RISC

Reduced Instruction Set Computer

ROM

Read Only Memory

ROM Objects

A collection of named module-specific scripts and firmware routines that are stored in an option module's ROM.

Script

A collection of console commands that run in a set order.

Slot

The physical location of a module or modules.

System Console

A terminal used to display status messages and accept operator commands.

System Module

Contains the main memory system and the processor

TRI/ADD Program

THIRD parties ADDing value to open systems