INTERNATIONAL STANDARD

ISO/IEC 11458

First edition
1993-12-01

Information technology – Microprocessor systems – VICbus – Inter-crate cable bus

Technologies de l'information – Systèmes à microprocesseurs – VICbus – Bus à câbles inter-châssis

Reference number
ISO/IEC 11458: 1993(E)

CONTENTS

Annexes

Page

Tables

Page

Figures

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialised system for world-wide standardisation. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organisation to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organisations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC1. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

International Standard ISO/IEC 11458 was prepared by Joint technical committee ISO/IEC JTC 1, *Information technology*, SC 26: *Microprocessor Systems*.

Annex A forms an integral part of ISO/IEC 11458. Annexes B to F are for information only.

Information technology - Microprocessor systems - VICbus - Inter-crate cable bus

1 Scope

The widespread use of high-performance, multi-processor systems based on backplane buses such as the IEC 821 bus (VMEbus), has inevitably led to the requirement to create multi-crate (-subrack, -chassis, etc.) systems. The VICbus inter-crate cable bus is designed to achieve such assemblies in a standard way.

VICbus, a multiplexed, multi-master, multi-slave cable bus, connects multiple backplane buses or stand-alone devices, providing transparent, softwareless interconnection for low latency short data transactions and fast transmission of data blocks over cables of up to 100 m in length. Address and data signals, each of 32 bits, together with those necessary for the control of the bus protocols, signal multiplexing, reset and error reporting are transmitted on twisted-wire pairs using differential line drivers and receivers. Up to 31 devices are permitted on a single VICbus cable.

VICbus data transfer protocols include both a compelled mode with end-to-end acknowledgement as well as two, high speed, non-compelled modes for high rate data transfers. The compelled protocols allow both broadcast (master write) and broadcall (master read) data transfers. One of the non-compelled protocols allows broadcast transfers, whereas neither permit broadcall operation.

Inter-master arbitration uses an efficient, modified single-level, daisy-chained mechanism. The interrupt mechanism allows 32 interrupt requests, multiplexed on eight physical lines. The specification includes system failure reporting, reset and live connection and disconnection, as well as the specification of control and status registers. Particular attention has been paid to redundancy of operation.

Whilst VICbus has been derived with multi-crate backplane bus systems in mind, this specification does not preclude the design of stand-alone VICbus devices. A normative annex giving rules and recommendations for a VMEbus to VICbus interface has been included, and further, similar annexes for other backplane bus standards will be added as the need arises.

2 Introduction to the ISO/IEC 11458 VICbus standard

2.1 Objectives

VICbus is a cable bus intended to be used to connect together multiple devices, particularly backplane bus systems, efficiently and with the possibility of software transparent operation, to allow large multi-crate, multi-processor systems to be constructed.

The objectives of this standard are to:

a) provide a standard cable bus for the interconnection of multiple devices, both backplane bus systems, such as the IEC821 VMEbus, and stand-alone apparatus;

b) specify the electrical characteristics of the cable bus;

c) specify the protocols that precisely define the interaction between devices connected to the VICbus;

d) specify the mechanisms necessary to construct fault-tolerant, multi-device systems;

e) provide the necessary definitions, terminology and background information to fully describe the VICbus protocols and other mechanisms.

2.2 Standard terminology

To avoid ambiguity, and to ensure that it is clear what the requirements for compliance are, many of the paragraphs in this standard are prefixed with sequentially numbered keywords by clause (N), known collectively as **regulations**, thus:

RULE <N.n>

RECOMMENDATION <N.n>

PERMISSION <N.n>

OBSERVATION <N.n>

2.2.1 Rule <N.n>

Rules form the basis of the VICbus standard, and may be expressed in textual, diagrammatic or tabular form. They use the imperative form and include the upper case words SHALL or SHALL NOT, which are reserved for this purpose only. Rules are printed in *italics*, thus:

RULE 2.1
Example: rules contain the upper case words SHALL or SHALL NOT.

2.2.2 Recommendation <N.n>

Recommendations contain information which will be very useful, if not vital, when designing to the VICbus standard and designers are encouraged to heed the advice given to ensure the best possible interpretation of the specification's requirements.

RECOMMENDATION 2.1
Example: recommendations contain very useful information.

2.2.3 Permission <N.n>

Permissions clarify aspects of the standard where multiple choices might be possible, and indicate acceptable lines of approach. The upper case word MAY is reserved for this purpose only.

> ***PERMISSION 2.1***
> Example: permissions help you choose and include the upper case word MAY.

2.2.4 Observation <N.n>

Observations serve to explain the rationale behind rules and other requirements.

> ***OBSERVATION 2.1***
> Example: observations explain the rationale of certain requirements.

2.3 Other terminology

All normative terms (that is those having a specified meaning within the context of this document) are typed in bold font, in general the first time they appear, thus: **arbiter**. Such terms are explained in the informative glossary, included as annex B, in addition to any references within the body of the text.

2.4 Timing diagrams

Timing diagrams are drawn such that a high level represents the asserted state (logical 1).

2.5 Tables

The **asserted** (logical 1) and **deasserted** (logical 0) states are indicated in tables as "1" and "0" respectively.

2.6 Data representation

Table 1 shows the labelling and significance of the bytes (groups of eight bits) within the four-byte location selected by a VICbus address or by the register select signals.

Table 1 - VICbus data representation

Most significant bit ▼ … Least significant bit ▼

	Byte (0)	Byte (1)	Byte (2)	Byte (3)
Data bits	31..........................24	23..........................16	15..........................8	7..........................0

3 Data transfer bus

3.1 Introduction

The Data Transfer Bus (DTB) is the means by which data, address, control and status information is transferred between VICbus **devices**. The device containing the **current master** uses the address information to select one or more **participating slaves** and the addresses within them with which it wishes to exchange data. The control information specifies the direction of data transfer and the number and position of bytes within the four-byte data bus which are to be transferred.

VICbus devices can be self-contained units, interfaces between VICbus and a backplane bus such as VMEbus, or both. Thus the DTB also carries control information which indicates whether the required slave is within a participating slave device or on a backplane associated with it.

Two types of data transfer protocol are specified for use on the DTB: **compelled** and **non-compelled**. The compelled protocol uses a full handshake which permits the cycle timing to be controlled by both the master and the participating slave(s) and, if the slave is on a backplane bus, permits the VICbus timing to be interlocked with that of the backplane. The two non-compelled protocols provide a faster means of transmitting data, by eliminating the round-trip propagation time of the cable which is inherent in the compelled protocol, however they can only be used to access slaves within VICbus devices.

The VICbus arbitration mechanism (specified in clause 4) ensures that only one master may use the DTB at a time. However, the use of bus drivers with a "wired-OR" capability permits a master to transmit data to several participating slaves simultaneously (**broadcast** operation) or to receive data from several participating slaves simultaneously (**broadcall** operation) when using the compelled protocol, or when using one of the non-compelled protocols in the case of broadcast operation. Additionally, whichever protocol is in use, data transfers may be spied upon (that is read on-the-fly with no active participation in the transfer protocol by the device concerned).

The DTB consists of two groups of physical **lines**: the 42 **information lines** (AD31-AD00, CL3-CL0, ID4-ID0 and SERR) onto which logical **signals** are multiplexed, and the three **timing lines** (AS, DS and WAIT) which are not multiplexed and for which the terms "lines" and "signals" can therefore be used interchangeably.

3.2 DTB cycle types

When initiating a DTB cycle, the master transmits control information indicating the cycle type to be performed, so defining the following:

a) the data transfer protocol to be used;

b) whether the required slave is within the addressed device (**direct cycles**) or on an associated backplane bus, if the device is an interface (**transparent cycles**);

c) whether a data transfer cycle or an **interrupt acknowledge** cycle is to be performed (the latter is specified in clause 5);

d) the address width used (32 or 64 bits) on associated VMEbus backplanes;

e) the byte alignment used (VMEbus or **byte lane aligned** (BLA), specified in 3.3.12).

Table 4 details the cycle types in terms of the lines and signals used.

3.2.1 *Direct cycles*

Direct cycles access VICbus slaves within devices rather than those connected to an associated backplane bus. The RS4-RS0 signals (see 3.3.9, below), are used to select one of 32 register locations (some of which may use the A31-A02 signals for sub-addressing). The allocation of mandatory and user definable register locations is specified in clause 8.

The three direct cycle types described in this specification are listed in table 2.

RULE 3.1
A device SHALL be capable of responding to direct compelled cycles as a slave.

Table 2 - Direct cycles

Cycle type	Protocol	Data alignment	Application
Direct compelled	Compelled	Byte lane	Transfers to slaves within devices (that is not on an associated backplane bus) with handshake control
Direct non-compelled 1	Non-compelled	Byte lane	Transfers to slaves within devices with no handshake control
Direct non-compelled 2	Non-compelled	Byte lane	Transfers to slaves within devices with no immediate handshake control; the slave's responses being pipelined. The initial transfer is the exception, since it is fully handshaken in order to confirm the existence and capability of the slave

3.2.2 *Transparent cycles*

The compelled protocol is used to access slaves connected to the backplane bus associated with an interface device, and the timing is interlocked with that of the backplane bus. Note that the non-compelled protocols cannot be used to implement transparent cycles, as this interlock is not possible.

The three types of transparent cycle described in this specification are listed in table 3.

Table 3 - Transparent cycles

Cycle type	Protocol	Data alignment	Application
Transparent VME-A64 [1)]	Compelled	VMEbus	VMEbus A64 cycles are transmitted over VICbus and a corresponding A64 cycle is generated on an associated VMEbus backplane
Transparent VME	Compelled	VMEbus	VMEbus A32, A24 or A16 cycles are transmitted over VICbus and corresponding cycles are generated on an associated VMEbus backplane
Transparent BLA	Compelled	Byte lane	Cycles are generated on an associated non-VMEbus backplane (intended for future applications)

1) A64 transfers are described in the proposed revision of the VMEbus specification document known as "VME64".

3.3 *Use of the DTB information lines*

In order to keep the number of physical lines to a practical number for a cable bus, the DTB information lines are multiplexed, and a DTB cycle therefore consists of an **address phase** and a **data phase** the latter consisting of one or more data transfers. The structure of a DTB cycle is illustrated in figure 1.

3.3.1 *The address phase*

The address phase is used by the master to:

a) select the slave device or devices which it requires to participate in the cycle;

b) specify the cycle type;

c) transmit an address internal to the slave or slaves.

3.3.2 *The data phase*

The data phase is used to transmit data to and / or from the selected address within the slave or slaves. In the case of a block transfer, it transfers data to or from successive addresses.

The multiplexed signals carried by the DTB information lines during the address and data phases are shown in tables 4, 5 and 6, and defined in subclauses 3.3.3 to 3.3.15, inclusive. In some cases the signal carried by a particular line depends on the cycle type, as well as on the phase of the cycle (address or data).

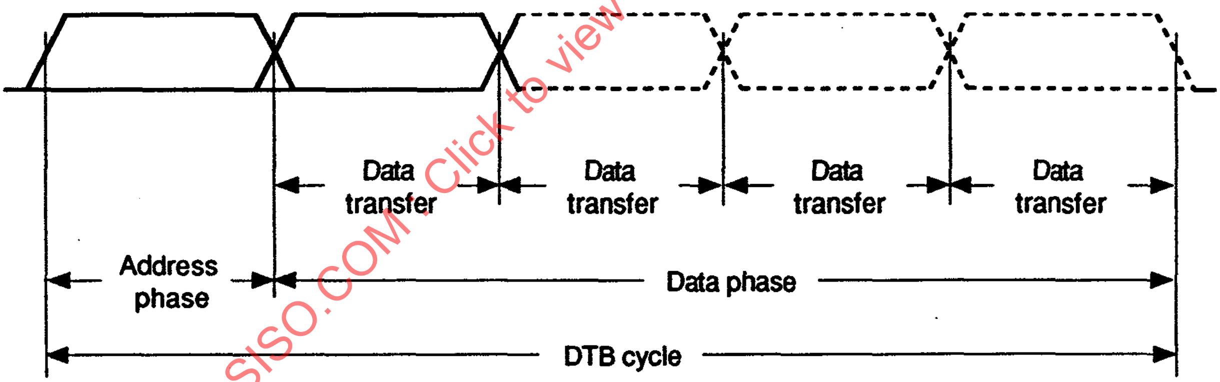

Figure 1 - DTB cycle

3.3.3 *Address / data lines AD31-AD00*

The 32 address / data lines (AD31-AD00) carry address and byte selection signals during the address phase, and data signals during the data phase.

3.3.4 *Control lines CL3-CL0*

The four control lines (CL3-CL0) carry control signals during both the address and data phases. In addition, they carry some addressing signals during the address phase of the transparent VME and VME-A64 cycle types. During the address phase, CL3 and CL2 carry cycle type definition signals, whilst CL1 and CL0 either carry further cycle type definition signals or addressing signals. During the data phase, all four control lines carry additional protocol and byte selection signals.

3.3.5 Identification lines ID4-ID0

The five identification lines (ID4-ID0) carry device selection and other addressing signals. During the address phase they carry the **device number** (or, in the case of an IACK cycle, the **interrupter number**) signals. During the data phase they carry address extension signals for transparent cycles, and register select signals for direct cycles.

3.3.6 Device number signals DN4-DN0

The device number signals carry the highest level of addressing information. They are used by the master to select the VICbus slave device or devices which it wishes to participate in the cycle. Device numbers in the range 31-1 select individual devices, whilst device number 0 (zero) selects all on-line devices capable and willing to participate in broadcast or broadcall operations. Every slave device is normally equipped with a means (for example a switch or switches) to allow the user to set the device number by which it will be selected.

> ***RECOMMENDATION 3.1***
> Equip a slave device with a means whereby a user can set the device number by which it will be selected. Design the device such that the device number can be set and is subsequently visible at its front panel.

> ***OBSERVATION 3.1***
> It is the responsibility of the user to ensure that no more than one slave has a particular device number, if this is undesirable.

> ***RULE 3.2***
> *A cycle carrying device number 0 (zero) SHALL indicate a broadcast or broadcall cycle.*

3.3.7 Address signals A31-A02

The address signals A31-A02 are used to select a four-byte (32 bit) location within selected slave or slaves, except during certain direct cycles when only the register select signals are used for this purpose. The selection of bytes within a four-byte location is described in 3.3.12.

> ***PERMISSION 3.1***
> An interface to a backplane bus MAY provide address mapping facilities whereby the user can program the correspondence between blocks of addresses on VICbus and blocks of addresses on the backplane bus.

3.3.8 Address extension signals AE5-AE0

The address extension signals AE5-AE0 are used during transparent cycles to pass extra addressing information over the VICbus (for example the VMEbus address modifier signals - see annex A).

3.3.9 Register select signals RS4-RS0

The register select signals are used during direct cycles to select one of 32, four-byte register locations within selected slave or slaves, as specified in clause 8. For most register locations, the address signals (A31-A02) are not used and, therefore, do not need to be decoded by the slave or slaves, however, access to facilities within the device, such as banks of memory, may require their use.

Table 4 - Use of the address / data, control and identification lines

NOTE - RSVD = Reserved for future allocation. All other mnemonics are explained in the accompanying text.

Address Phase	Address / data	Control lines				Identification lines				
Cycle type	AD31-AD02[1)]	CL3	CL2	CL1	CL0	ID4	ID3	ID2	ID1	ID0
Transparent VME-A64	A31-A02	0	0	AE5	AE4	DN4	DN3	DN2	DN1	DN0
Transparent BLA	A31-A02	0	1	0	0	DN4	DN3	DN2	DN1	DN0
Reserved	A31-A02	0	1	0	1	RSVD	RSVD	RSVD	RSVD	RSVD
Reserved	A31-A02	0	1	1	0	RSVD	RSVD	RSVD	RSVD	RSVD
IACK	A31-A02	0	1	1	1	IN4	IN3	IN2	IN1	IN0
Direct compelled	A31-A02	1	0	0	0	DN4	DN3	DN2	DN1	DN0
Direct non-compelled 1	A31-A02	1	0	0	1	DN4	DN3	DN2	DN1	DN0
Direct non-compelled 2	A31-A02	1	0	1	0	DN4	DN3	DN2	DN1	DN0
Reserved	A31-A02	1	0	1	1	RSVD	RSVD	RSVD	RSVD	RSVD
Transparent VME	A31-A02	1	1	AE5	AE4	DN4	DN3	DN2	DN1	DN0

1) For details of the use of AD01 and AD00 in the address phase, see tables 5 and 6.

Data Phase	Address / data	Control lines				Identification lines				
Cycle type	AD31-AD00	CL3	CL2	CL1	CL0	ID4	ID3	ID2	ID1	ID0
Transparent VME-A64	D31-D00[2)]	BLT	WRITE	DSEL1	DSEL0	RSVD	AE3	AE2	AE1	AE0
Transparent BLA	D31-D00	BLT	WRITE	DSEL1	DSEL0	RSVD	AE3	AE2	AE1	AE0
Reserved	D31-D00	RSVD	WRITE	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD
Reserved	D31-D00	RSVD	WRITE	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD
IACK	D31-D00	RSVD	0	DSEL1	DSEL0	RSVD	RSVD	RSVD	RSVD	RSVD
Direct compelled	D31-D00	BLT	WRITE	DSEL1	DSEL0	RS4	RS3	RS2	RS1	RS0
Direct non-compelled 1	D31-D00	BLT	1	DSEL1	DSEL0	RS4	RS3	RS2	RS1	RS0
Direct non-compelled 2	D31-D00	BLT	WRITE	DSEL1	DSEL0	RS4	RS3	RS2	RS1	RS0
Reserved	D31-D00	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD
Transparent VME	D31-D00	BLT	WRITE	DSEL1	DSEL0	RSVD	AE3	AE2	AE1	AE0

2) For an explanation of the use of the AD lines in transparent VME-A64 cycles, see 3.4.

3.3.10 *Block transfer signal BLT*

The BLT signal is asserted by the master to indicate that a block transfer is taking place (block transfers are described in 3.5.1).

RULE 3.3
During a block transfer cycle, the master SHALL assert the BLT signal.

OBSERVATION 3.2
During transparent VME and transparent VME-64 block transfer cycles, the BLT signal duplicates information carried by the address extension signals, but is retained for compatibility with other transfer types.

3.3.11 *Write signal WRITE*

The write signal is used by the master to indicate whether it is writing data (WRITE asserted) or reading data (WRITE deasserted). Its value may change for every data transfer within a cycle using the compelled protocol.

RULE 3.4
If during a data transfer the master asserts WRITE, then slaves SHALL NOT drive any of the data signals D31-D00, and if the master deasserts WRITE then it SHALL NOT drive any of the data signals D31-D00.

3.3.12 *Byte selection signals LWORD, A01, ASEL0, ASEL1, DSEL0, DSEL1*

The byte selection signals are used to select which of the bytes within a four-byte location (as defined in 2.6) are to be read or written. Any number from one to four bytes within a four-byte location may be read or written concurrently.

There are two ways in which the selected bytes may be carried by the 32 VICbus data signals:

a) the **VMEbus byte alignment** detailed in table 5;

b) the **byte lane alignment** (BLA) detailed in table 6.

The cycle type indicates which alignment is in use during a cycle. Note that signals carried by the same four lines (AD01-AD00 during the address phase and CL1-CL0 during the data phase) are used for byte selection in both cases, but that the two signals carried by the AD01-AD00 lines have different names in the two cases.

RULE 3.5
All VICbus devices participating in transparent VME and transparent VME-64 cycles SHALL use the byte alignments specified by table 5.
All VICbus devices participating in other cycles SHALL use the byte alignments specified by table 6.

Table 5 - VMEbus byte alignment

	Address phase		Data phase					
Lines ⇒	AD01	AD00	CL1	CL0	AD31-AD24	AD23-AD16	AD15-AD08	AD07-AD00
Signals ⇒	A01	LWORD	DSEL1	DSEL0	D31-D24	D23-D16	D15-D08	D07-D00
byte(0)	0	1	1	0	-	-	byte(0)	-
byte(1)	0	1	0	1	-	-	-	byte(1)
byte(2)	1	1	1	0	-	-	byte(2)	-
byte(3)	1	1	0	1	-	-	-	byte(3)
byte(0-1)	0	1	1	1	-	-	byte(0)	byte(1)
byte(1-2)	1	0	1	1	-	byte(1)	byte(2)	-
byte(2-3)	1	1	1	1	-	-	byte(2)	byte(3)
byte(0-2)	0	0	1	0	byte(0)	byte(1)	byte(2)	-
byte(1-3)	0	0	0	1	-	byte(1)	byte(2)	byte(3)
byte(0-3)	0	0	1	1	byte(0)	byte(1)	byte(2)	byte(3)
byte (4-7) [1]	0	0	1	1	byte(4)	byte(5)	byte(6)	byte(7)

1) VMEbus D64 transfers (see annex A.)

Table 6 - Byte lane alignment

	Address phase		Data phase					
Lines ⇒	AD01	AD00	CL1	CL0	AD31-AD24	AD23-AD16	AD15-AD08	AD07-AD00
Signals ⇒	ASEL1	ASEL0	DSEL1	DSEL0	D31-D24	D23-D16	D15-D08	D07-D00
byte(0)	0	1	1	0	byte(0)	-	-	-
byte(1)	0	1	0	1	-	byte(1)	-	-
byte(2)	1	1	1	0	-	-	byte(2)	-
byte(3)	1	1	0	1	-	-	-	byte(3)
byte(0-1)	0	1	1	1	byte(0)	byte(1)	-	-
byte(1-2)	1	0	1	1	-	byte(1)	byte(2)	-
byte(2-3)	1	1	1	1	-	-	byte(2)	byte(3)
byte(0-2)	0	0	1	0	byte(0)	byte(1)	byte(2)	-
byte(1-3)	0	0	0	1	-	byte(1)	byte(2)	byte(3)
byte(0-3)	0	0	1	1	byte(0)	byte(1)	byte(2)	byte(3)

3.3.13 Interrupter number signals IN4-IN0

The use of the interrupter number signals is specified in clause 5. They are included in table 4 for completeness.

3.3.14 Slave response signal SERR

The SERR signal carries no timing information, but is asserted by a slave to indicate that it has been unable to execute the requested data transfer for device-specific reasons. For example, it may be used to indicate that the slave has insufficient storage space for the data being written by the master (write data overflow), that it has insufficient data to satisfy the number of read transfers requested by the master (read data underflow), or to signal a read parity error.

3.3.15 Data signals D31-D00

The data signals D31-D00 provide a 32-bit data path between VICbus devices. The way that they are used to carry from one to four bytes of information is specified in 3.3.12.

3.4 Transparent VME-A64 cycle

The transparent VME-A64 cycle is used to transmit VMEbus cycles using 64 bit addressing over VICbus. The 64 bit address is transmitted in two parts as shown in table 7.

RULE 3.6
In a transparent VME-A64 cycle, the 64 bit VMEbus address SHALL be transmitted in two parts as shown in table 7.

Table 7 - Transparent VME-A64 signal assignment

	Address phase		First data phase		
VICbus lines	AD31-AD01	AD00	AD31-AD00	CL1	CL0
VMEbus signals	A31-A01	LWORD*	A63-A32	DS1[1)]	DS0[1)]

[1)] NOTE - the VICbus CL1 and CL0 signals carry the VMEbus DS1* and DS0* signals in inverted (positive true) form.

3.5 Data transfer cycle - bus protocols and timing

This subclause and those following specify the use of the three lines used to implement the bus protocols and timings:

AS	Address Strobe, driven by the master;
DS	Data Strobe, driven by the master;
WAIT	Synchronisation signal, driven by the slave or slaves.

A DTB cycle consists of an address phase followed by a data phase which itself contains one or more data transfers, as illustrated in figure 1. The address phase is non-compelled, that is its duration is controlled by the master alone, and slave devices have a minimum specified time in which to determine whether or not they have been selected and should participate in the cycle. The data phase uses one of two types of protocol:

a) a compelled protocol, described in 3.6, which uses a handshake to determine the duration of each data transfer, such that it proceeds at a rate determined by the master and the slowest **participating slave**;

b) two non-compelled protocols, described in 3.7, with either no handshake (NC1) or a pipelined handshake (NC2) to eliminate the round-trip propagation time of the cable inherent in the compelled protocol.

DTB cycles using the compelled protocol may consist of any number of read and write transfers intermixed in any order, for example block transfers and read-modify-write transfers. Cycles using one of the non-compelled protocols cannot intermix read and write transfers, and can therefore only implement single-word or block transfers.

3.5.1 Block transfer cycles

Block transfer cycles consist of data transfers all in the same direction, read or write. They can be used to pass blocks of data to or from several contiguously addressed locations within slaves. The address of the first location is specified, and thereafter each participating slave increments an internal address by an appropriate amount after every data transfer.

VICbus block transfer cycles will not cause the slave's internal address counter to cross a 2 048 byte boundary. The restriction of a block transfer to 2 K bytes is made to simplify the design of slaves, for example by limiting the size of address counter required, as well as placing a limit on bus latency if bus mastership is relinquished between blocks. This restriction need not significantly increase the time taken to transfer large blocks of data, since a master can, if required, initiate successive block transfer cycles to the same slave without relinquishing control of the DTB between cycles, and without the need for software intervention.

It should be noted that transparent VME cycles carrying VMEbus D08(EO), D16 and D32 block transfers are restricted to 256 bytes by the VMEbus specification. Similarly, VMEbus D64 block transfers are limited to 2 048 bytes.

> ***RULE 3.7***
> *VICbus block transfer cycles SHALL NOT cause a slave's internal address counter to cross a 2 048 byte boundary.*

3.5.2 Read-modify-write cycles

Read-modify-write cycles are used when a master wishes to inspect a location and alter its value, without there being any possibility of another master accessing that location during the operation. Such an indivisible operation is often required as a mechanism for the reservation of a resource, when several masters need to use it at different times. The DTB cycle used in such a case consists of an address phase, a read transfer and a write transfer.

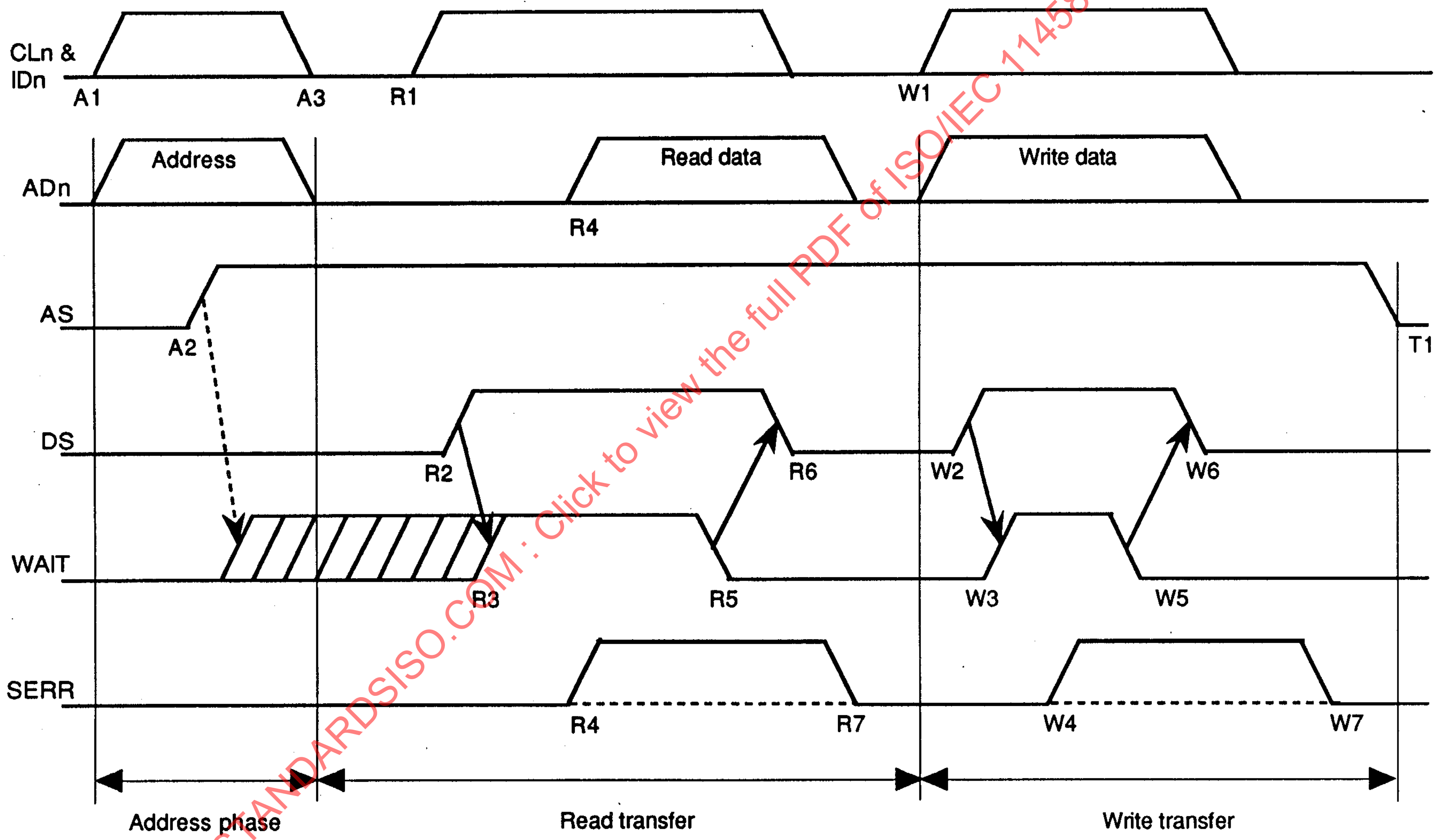

Figure 2 - Compelled protocol

3.6 *Compelled protocol*

Figure 2 illustrates a cycle using the compelled protocol, in which the data phase consists of a read transfer followed by a write transfer. The figure is further explained in the following clauses, in which paragraphs are labelled to correspond to events in the figure.

3.6.1 *The address phase*

A1 The master places the required address phase signals, as defined in tables 4, 5 and 6, onto the CL, ID and AD lines.

A2 The master asserts AS after a specified set-up time. Every on-line slave device then examines the device number and cycle type to determine whether it should participate in the cycle. If this is the case, it may, at this time, assert WAIT.

A3 The master removes the address phase signals from the CL, ID and AD lines after a specified hold time, and proceeds directly to the data phase without waiting for the assertion of WAIT.

3.6.2 *The data phase*

The data phase can consist of any number of data transfers to or from the selected slave(s). Read and write transfers can be intermixed in any order. However, these transfers will all be directed to the same address within the slave, except when the slave is incrementing an internal address counter during a block transfer, as described in 3.5.1.

3.6.2.1 *Read data transfer*

R1 The master places the data phase signals defined in tables 4, 5 and 6 onto the CL and ID lines, and deasserts the AD lines. Note that CL2 carries the WRITE signal, and will therefore be deasserted.

R2 The master asserts DS after a specified set-up time.

R3 All slaves participating in the cycle assert WAIT, no later than a specified time after receiving the assertion of DS.

(Hereafter, the descriptions of slave actions refer to participating slaves only.)

R4 The slave accesses the data requested by the master, and places it on the appropriate AD lines. (Note that in the case of a transparent cycle, this will involve an operation on the backplane associated with the selected interface device.) The slave also asserts or deasserts SERR to indicate the status of the transfer.

R5 The slave deasserts WAIT after the data and SERR signals have been valid for a specified set-up time. This specification does not place an upper limit on the time taken for a slave to respond in this way, but a master should implement a time-out to prevent a hang-up of the VICbus.

When the master receives the deassertion of WAIT it knows that all participating slaves have placed data and SERR (if appropriate) on the bus and that it can store this information. Note: for a broadcall cycle, the master ensures that WAIT has been continuously deasserted for a time equal to twice the cable propagation time to allow for Wired-OR Glitches (described in annex E) which may occur on the WAIT line.

R6 The master deasserts DS to terminate the transfer.

R7 The slave deasserts the AD and SERR lines within a specified time of receiving the deassertion of DS. If, as in this case, the master wishes to perform another data transfer within the same cycle, it ensures that the DS line is deasserted for a time sufficient to ensure that its deassertion and reassertion can be detected at the far end of a cable of maximum permitted length.

3.6.2.2 Write data transfer

W1 The master places the data phase signals defined in tables 4, 5 and 6 onto the CL and ID lines, and the data onto the AD lines. Note that CL2 carries the WRITE signal, and will therefore be asserted.

W2 The master asserts DS after it has been deasserted for the appropriate time (described in R7, above) and after the data phase signals have been asserted for a specified set-up time.

W3 The slave asserts WAIT within a specified time of receiving the assertion of DS.

W4 If it is able to do so, the slave accepts the data, and asserts or deasserts SERR as appropriate. Note that in the case of a transparent cycle this will involve an operation on the backplane associated with the selected interface device.

W5 The slave deasserts WAIT after it has been asserted for a specified minimum time, and after SERR has been stable for a specified set-up time. This specification does not place an upper limit on the time a slave takes to respond in this way, but a master should implement a time-out to prevent a hang-up of the bus.

When the master receives the deassertion of WAIT it knows that all participating slaves have accepted the data and asserted SERR (if appropriate), and that it can therefore sample SERR and terminate the transfer. Note that, for a broadcast cycle, the master actually ensures that WAIT has been continuously deasserted for a time equal to twice the cable propagation time, for the reasons already explained in 3.6.2.1.

W6 The master deasserts DS to terminate the transfer.

W7 The slave deasserts the SERR line within a specified time of receiving the deassertion of DS.

T1 To terminate the cycle, the master deasserts AS, but only after DS has been deasserted for a specified hold time, which depends on the circumstances, as follows:

a) *following a write transfer in which SERR is deasserted:*
the hold time is sufficient to ensure that at the far end of a cable of maximum length, AS is seen to be deasserted after DS;

b) *following a read transfer, or any transfer in which SERR is asserted:*
the hold time equals twice the cable propagation time plus the time allowed for a slave to deassert the SERR and AD lines in response to the deassertion of DS. Thus all the DTB lines are deasserted by the time that AS is deasserted.

The removal of AS indicates to a **pending master** that the DTB is clear, and that it can take control, as described in 4.3.

3.7 *Non-compelled protocols*

Unlike the compelled protocol, described in 3.6, the non-compelled protocols do not use a handshake to control the progress of data transfers. Since the round-trip cable delay is thereby eliminated, they provide faster transfers. Two such protocols are specified, Non-Compelled 1 (NC1) and Non-Compelled 2 (NC2).

The NC1 protocol provides the fastest means of transferring data over the VICbus, being entirely synchronous, with no handshake at all between the master and the slave, but is thereby limited to write transfers only. The master simply assumes that the slave exists, and after the initial address phase, will toggle the DS line and data signals at a pre-determined rate. NC1 can be used for the transmission of single words or blocks of data.

Unlike the NC1 protocol, the NC2 protocol includes slave responses, so both read and write transfers are possible and the slave has the means to report data errors and over- or under-runs. An initial compelled transfer allows a master to confirm the existence of a slave with the specified device number which implements the NC2 protocol. The slave may then, if necessary, delay the start of the non-compelled part of the protocol (for example to ensure access to a multi-ported memory) before allowing the cycle to proceed. After this first compelled data transfer the protocol is non-compelled, and is capable of transferring data at the same rate as the NC1 protocol. Because of this structure, the NC2 protocol is intended for (but is not limited to) the transfer of blocks of data, since for single cycles there is no speed advantage over the compelled protocol.

During the non-compelled part of the NC2 protocol, the slave asserts and deasserts the WAIT line in response to each assertion of DS by the master. The rate at which it toggles the WAIT line may, if necessary, be slower than the master toggles DS. The master does not use WAIT to regulate its DS timing, thereby eliminating the round-trip cable delay inherent in the compelled protocol.

Both NC1 and NC2 are limited to data transfers in one direction only during a cycle (read or write for the NC2 protocol, write only for the NC1 protocol).

NC1 cannot support broadcast transfers in the normal sense, but transfers with DN=0 are permitted, to indicate that multiple slaves may receive data simultaneously. NC2 can support neither broadcast nor broadcall transfers owing to the use of pipelined slave responses. Note, however, that devices can spy on the data passing on the bus during a cycle, whatever protocol is being used.

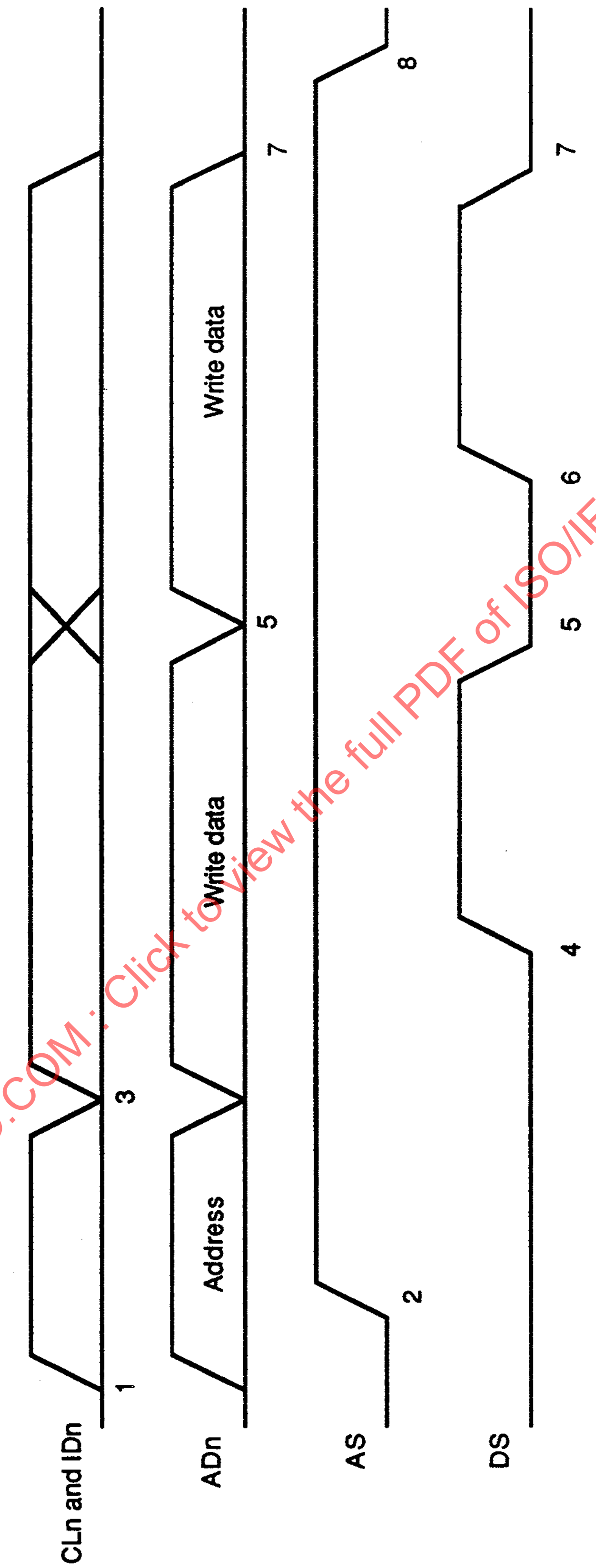

Figure 3 - Non-compelled 1 protocol

3.7.1 Non-compelled 1 (NC1)

The NC1 protocol is illustrated in figure 3, and described in the following paragraphs, which are numbered to correspond with events in the figure.

1 The master places the required address phase signals defined in tables 4, 5 and 6 onto the CL, ID and AD lines.

2 The master asserts AS after a specified set-up time.

3 After a specified hold time, the master replaces the address phase signals on the CL, ID and AD lines with data phase signals and with the first data word to be transferred.

4 The master asserts DS after a specified set-up time.

If no more data is to be transferred, the cycle continues with event 7.

5 The master deasserts DS after it has been asserted for a specified time.

It changes the data on the AD lines a specified hold time after it deasserts DS. It may also change the CL and ID lines at this time, but must restore them to their previous values before event 6.

6 The master asserts DS after it has been deasserted for a specified time, and after the new data has been established for a specified set-up time.

If more data is to be transferred, the cycle continues with event 5.

7 The master deasserts DS after it has been asserted for a specified time, and deasserts the CL, ID and AD lines a specified hold time after the deassertion of DS.

8 The master deasserts AS after a further specified time.

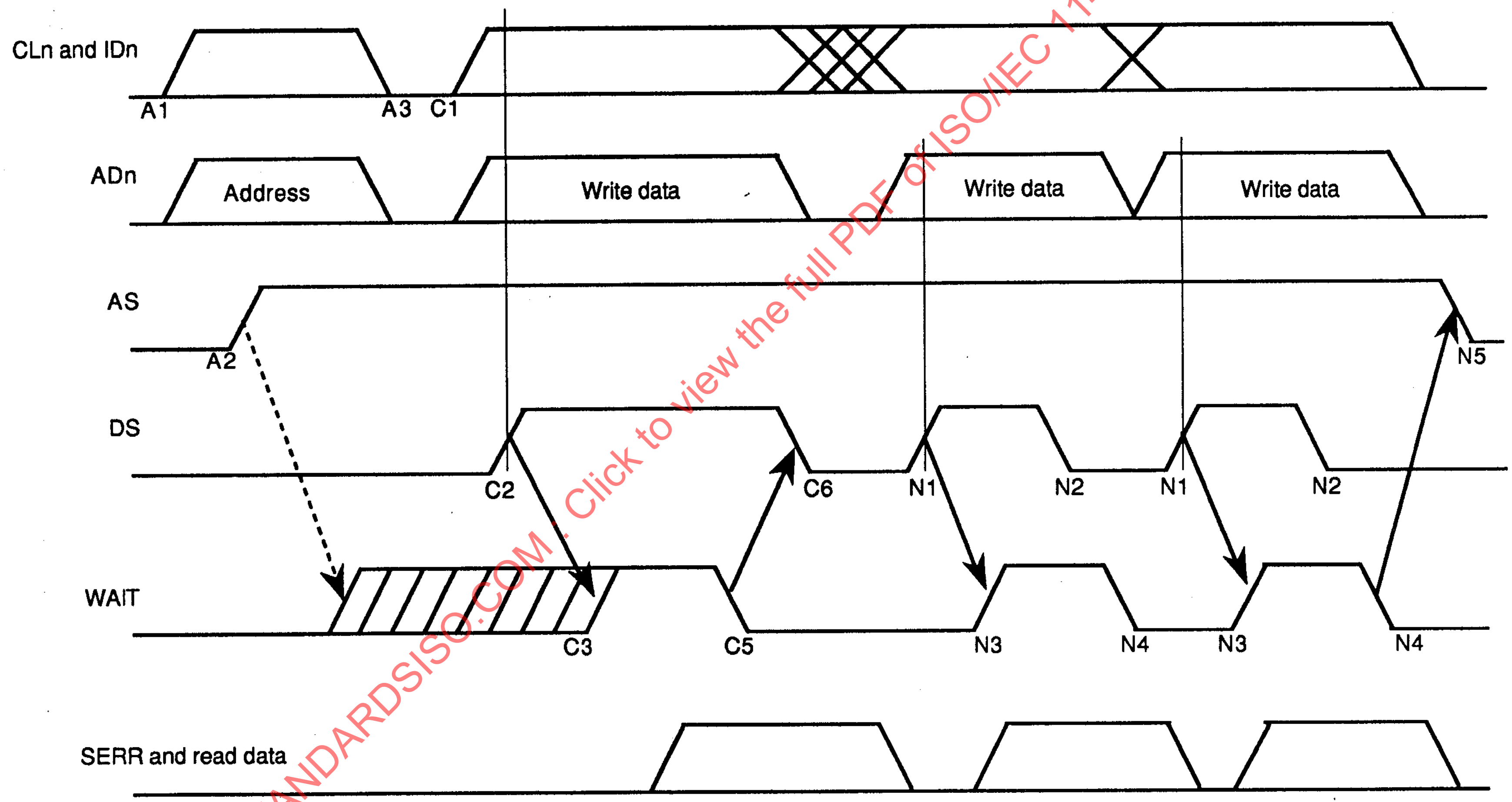

Figure 4 - Non-compelled 2 protocol

3.7.2 Non-compelled 2 (NC2)

Figure 4 illustrates both NC2 read and write transfers, since the protocol sequence (controlled by AS, DS and WAIT) is identical in the two cases, and only the use of the AD lines is different. The sequence of events seen by the master is shown with the slave's responses being delayed by the round-trip cable delay between master and slave.

During the address phase (events labelled A1 to A3 in the figure) and first data phase (C1 to C7), the NC2 protocol is identical to the compelled protocol, as described in 3.6, with the exception that the hold time for SERR and read data (C7) is relative to the deassertion of WAIT rather than DS. Therefore only the non-compelled part of the cycle is described in the following paragraphs, which are labelled to correspond with events in the figure.

N1 The master asserts DS after it has been deasserted for a specified time and after the CL and ID lines, and AD lines in the case of a write transfer, have been established for a specified set-up time.

N2 The master deasserts DS after it has been asserted for a specified time. A specified hold time after this it may change the data on the CL and ID lines, and on the AD lines in the case of a write transfer (but note that the CL and ID lines must be re-established with the *same* values in time for the next assertion of DS).

The master repeats actions N1 and N2 as many times as is necessary to send or request the number of data words it wishes to transfer during the cycle and then terminates the cycle as described at N5.

The slave responds to each assertion of DS by asserting and deasserting WAIT, but if necessary at a slower rate. For example, during a read cycle the master will toggle DS to indicate both the maximum rate at which it can accept data and the number of data words it requires. If the slave cannot transmit data at this rate it responds more slowly, and thus the delay from each assertion of DS to the corresponding WAIT response will increase as the cycle progresses. The slave, then, must count the assertions of DS and also its own responses. However, during a write cycle, since the slave must accept data at the rate transmitted by the master, it will most likely issue WAIT responses at the same rate.

N3 Whenever the slave receives an assertion of DS during a write cycle it captures the data from the AD lines. For both read and write cycles it responds (eventually) by asserting WAIT, and by asserting the SERR and AD lines if appropriate.

N4 The slave deasserts WAIT when:

a) it has been asserted for a specified time, and

b) the SERR line, and AD lines during a read cycle, have been asserted for specified set-up times.

It then deasserts the SERR and AD lines, after a specified hold time.

When the master receives the deassertion of WAIT it captures the information on the SERR line and, for a read cycle, on the AD lines.

N5 To terminate the transfer, the master deasserts AS a specified time after receiving the final deassertion of WAIT.

OBSERVATION 3.3
In order to detect the final deassertion of WAIT in an NC2 cycle, the master can either count WAIT responses or can use a time-out to detect that WAIT has been deasserted for longer

than would be normal. The latter is inefficient since the master's VICbus interface circuitry cannot distinguish between a cycle which ends normally and one which times-out due to an error condition.

3.8 Slave participation in DTB cycles

A slave will participate in a DTB cycle if it implements the cycle type indicated by the master and if a number of other criteria are met. Table 8 summarises the action the slave will make, with references to the regulations governing its behaviour.

RULE 3.8
A slave SHALL participate in a cycle if it implements the cycle type indicated by the master and if the conditions given in table 8 are met.

PERMISSION 3.3
A slave MAY choose not to participate in broadcast or broadcall cycles.

Table 8 - Summary of slave participation in DTB cycles

Remark- The actions given are dependent on whether or not the slave implements the cycle type.

Cycle type	Device status (see 6.4)	DN4-DN0 value (*dn* indicates the device's own number)	Participates (if the cycle type is implemented)	Relevant regulations (in addition to RULE 3.8)
Any	Offline	*dn* or 0	No	RULE 6.14
Transparent	Online-disabled	*dn* or 0	No	RULE 6.12
	Online-enabled	*dn*	Yes (mandatory)	-
		0	Yes (optional)	PERMISSION 3.3
Direct Compelled	Online (either)	*dn*	Yes (mandatory)	RULE 3.1
		0	Yes (optional)	PERMISSION 3.3
Direct NC1	Online (either)	*dn*	Yes (mandatory)	-
		0	Yes (optional)	PERMISSION 3.3
Direct NC2	Online (either)	*dn*	Yes (mandatory)	-
		0	No	1)

1) A master should never attempt to broadcast or broadcall with this cycle type.

OBSERVATION 3.4
An addressed slave which does not implement the requested cycle type will not participate in the cycle and thus the cycle will time-out, unless it is a broadcast or broadcall cycle in which another slave participates or the cycle uses the NC1 protocol (for which there is no slave response defined).
A time-out indicates to the master the absence of a suitable slave at this device number, but if a slave responds, and in doing so asserts SERR, the master knows that there has been some other type of error. In the case of broadcast and broadcall cycles there is no indication as to *which* slaves have responded, except in the data returned for certain broadcall cycles used to elicit just such information (see clause 8).

3.9 DTB timing rules

The timing rules and the associated regulations governing the behaviour of masters and slaves are given in figures 5 to 10 and tables 9 and 10. Timings are specified for both **long** and **short** cables (refer to 8.3).

OBSERVATION 3.5
There is no bus timer function inherent in the compelled or NC2 protocol, so every master needs to implement its own timer to terminate a cycle if a transfer does not complete within a reasonable time.

RECOMMENDATION 3.2
Design masters such that the user can select the value of the time-out period.

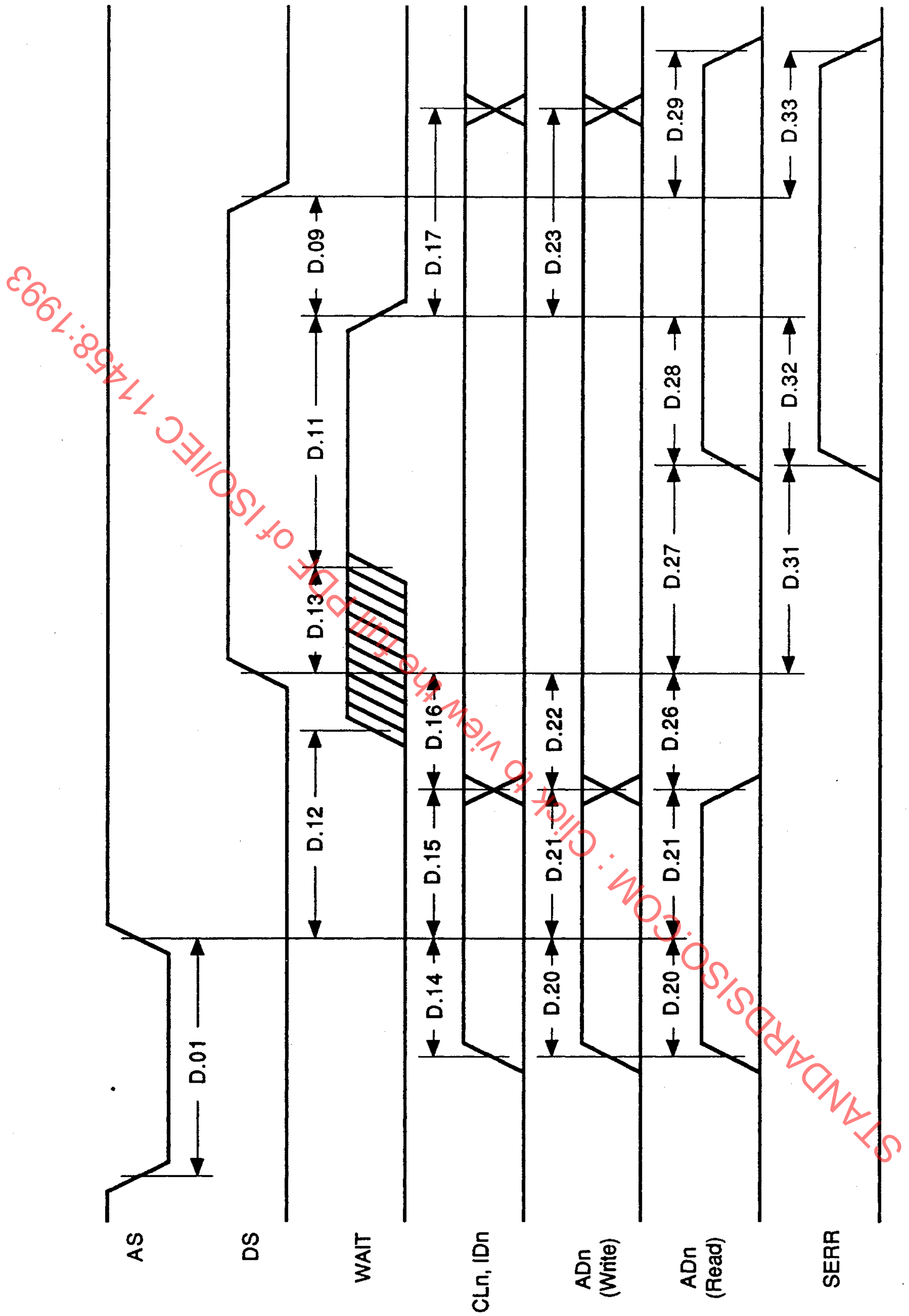

Figure 5 - Compelled cycle: address phase and first data transfer

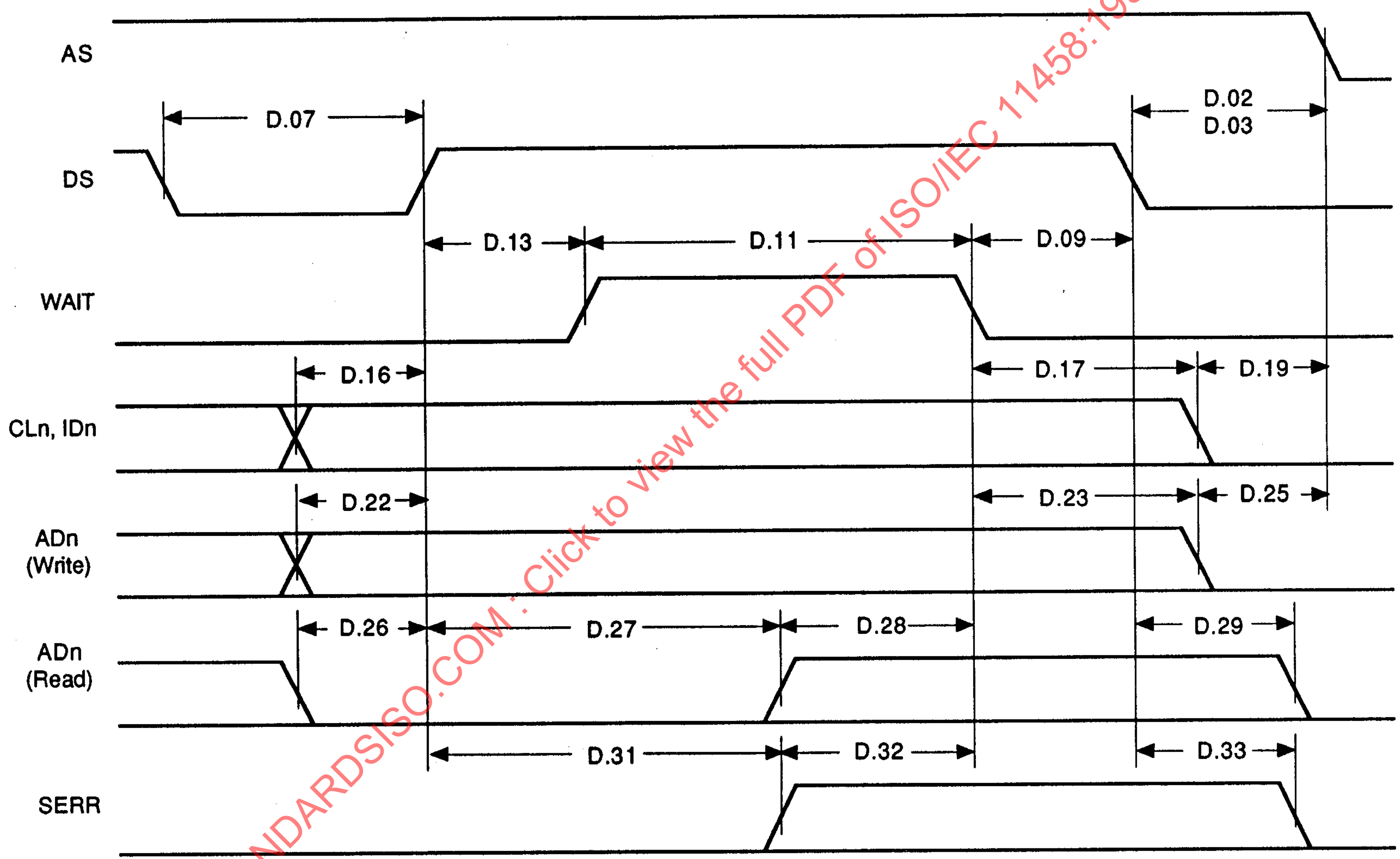

Figure 6 - Compelled cycle - last data transfer and end of cycle

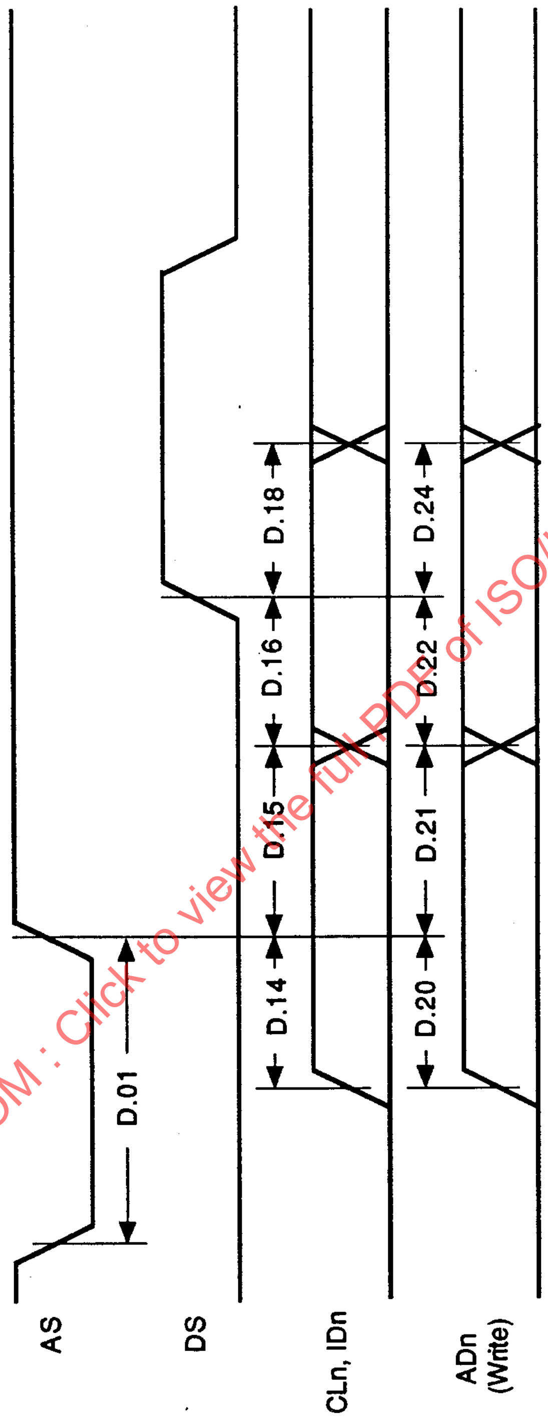

Figure 7 - NC1 - address phase and first data transfer

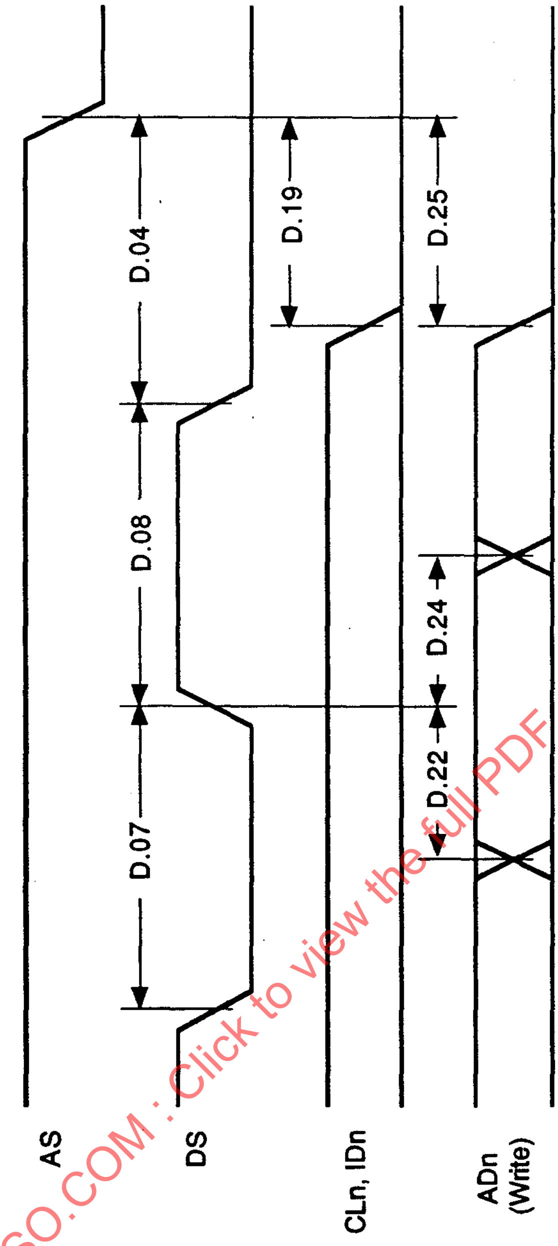

Figure 8 - NC1 - last data transfer and end of cycle

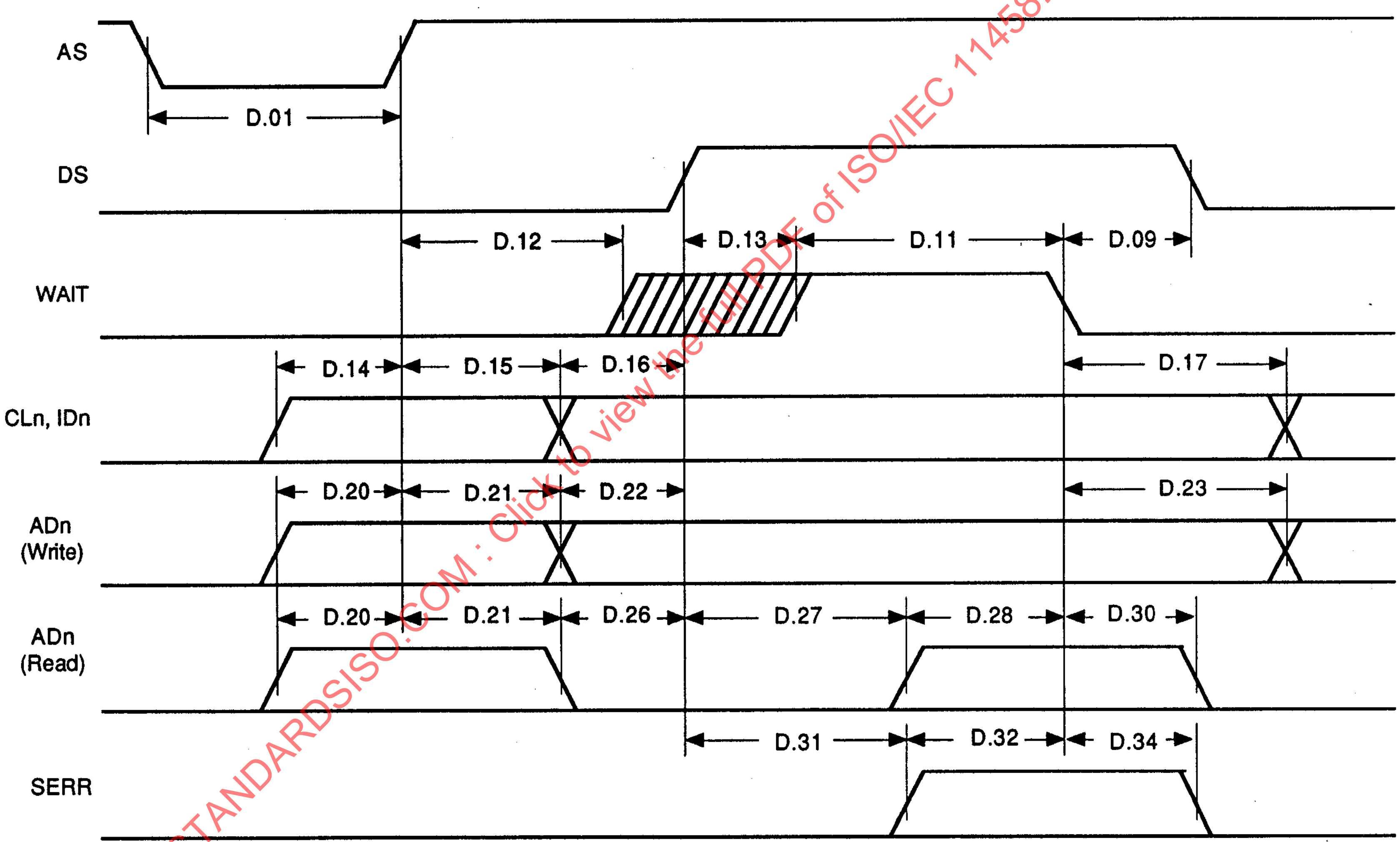

Figure 9 - NC2 - address phase and first data transfer

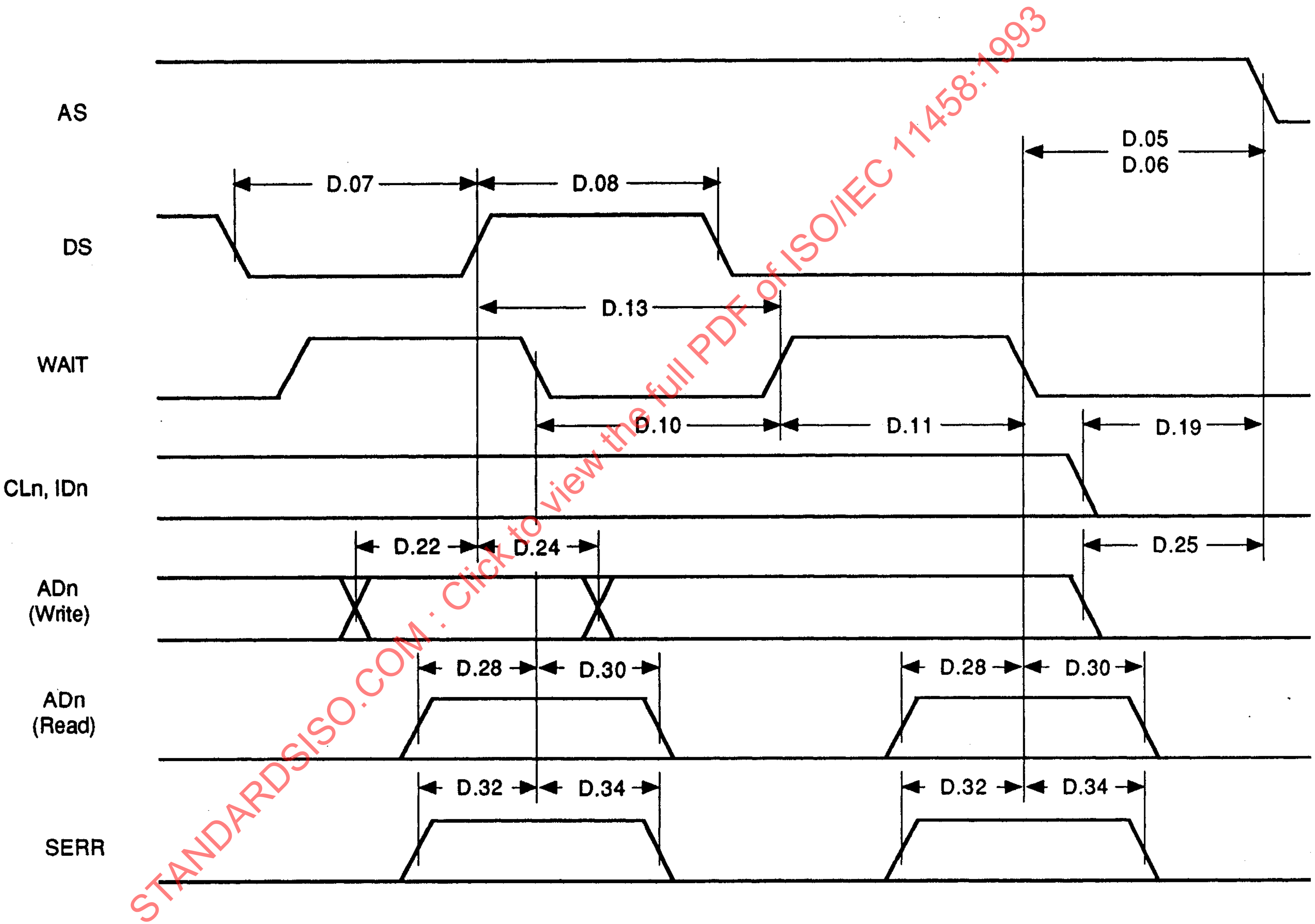

Figure 10 - NC2 - last data transfer and end of cycle

Table 9 - Master - timing regulations

(Refer to figures 5, 6, 7, 8, 9 and 10)

Regulation		Para meter	Long cable min. ns	Long cable max. ns	Short cable min. ns	Short cable max. ns
RULE 3.9	*The master SHALL ensure that AS is deasserted prior to asserting AS*	D.01	50	-	50	-
RULE 3.10	*If the last data transfer in a compelled cycle is a write and SERR is not asserted, the master SHALL maintain AS asserted after deasserting DS*	D.02	30	-	25	-
RULE 3.11	*If the last data transfer in a compelled cycle is a read, if SERR is asserted or if the cycle has timed-out, the master SHALL maintain AS asserted after deasserting DS*	D.03	1 230	-	1 230	-
RULE 3.12	*In a NC1 cycle, the master SHALL maintain AS asserted after deasserting DS*	D.04	30	-	25	-
RULE 3.13	*In a NC2 cycle, if the data transfer is a write and SERR is not asserted, the master SHALL maintain AS asserted until the final WAIT is deasserted*	D.05	0	-	0	-
RULE 3.14	*In a NC2 cycle, if the data transfer is a read or SERR is asserted, the master SHALL maintain AS asserted after the final WAIT is deasserted*	D.06	30	-	25	-
RULE 3.15	*The master SHALL ensure that DS is deasserted between successive data transfers:*					-
	a) Compelled cycles	D.07	50	-	50	-
	b) NC1 and NC2 cycles	D.07	50	-	50	-
RULE 3.16	*In a NC1 cycle and a NC2 cycle after the first (compelled) transfer, the master SHALL assert DS for a minimum time*	D.08	160	-	70	-
RULE 3.17	*In a compelled cycle and during the first (compelled) transfer in a NC2 cycle, the master SHALL maintain DS asserted until WAIT is deasserted as follows:*					
	a) Non-broadcast and non-broadcall cycles	D.09	0	-	0	-
	b) Broadcast and broadcall cycles	D.09	1 200	-	360	-
OBSERVATION 3.6	In a NC2 cycle, the master is assured that WAIT will be deasserted for a minimum time	D.10	50	-	50	-
OBSERVATION 3.7	The master is assured that WAIT will be asserted for a minimum time:					
	a) Compelled cycles	D.11	50	-	50	-
	b) NC2 cycles	D.11	50	-	50	-

Table 9 - (continued)

Regulation		Para meter	Long cable min. ns	Long cable max. ns	Short cable min. ns	Short cable max. ns
OBSERVATION 3.8	In the first data transfer in a compelled and a NC2 cycle, the master is assured that WAIT will not be asserted before it asserts AS	D.12	0	-	0	-
RULE 3.18	*In the address phase, the master SHALL maintain the CLn and IDn lines after asserting AS*	D.15	90	-	30	-
RULE 3.19	*The master SHALL establish the CL and ID lines prior to asserting DS*	D.16	90	-	30	-
RULE 3.20	*In all compelled data transfers, and in the first data transfer of a NC2 cycle, the master SHALL maintain the CLn and IDn lines until WAIT is deasserted as follows:*					
	a) non-broadcast and non-broadcall cycles	D.17	0	-	0	-
	b) broadcast and broadcall cycles	D.17	1 200	-	360	-
RULE 3.21	*In a NC1 cycle, and in data transfers subsequent to the first in a NC2 cycle, the master SHALL maintain the CLn and IDn lines after asserting DS*	D.18	0	-	0	-
RULE 3.22	*The master SHALL deassert the CLn and IDn lines prior to deasserting AS*	D.19	30	-	25	-
RULE 3.23	*In the address phase, the master SHALL establish the ADn, CLn and IDn lines prior to asserting AS*	D.14 D.20	90	-	30	-
RULE 3.24	*In the address phase the master SHALL maintain the ADn lines after asserting AS*	D.21	90	-	30	-
RULE 3.25	*In a write transfer, the master SHALL establish the ADn lines prior to asserting DS*	D.22	90	-	30	-
RULE 3.26	*In any write transfer in a compelled cycle and in the first write transfer in NC2 cycles, the master SHALL maintain the ADn lines until WAIT is deasserted as follows:*					
	a) non-broadcast and non-broadcall cycles	D.23	0	-	0	-
	b) broadcast and broadcall cycles	D.23	1 200	-	360	-
RULE 3.27	*In any transfer in a NC1 cycle and in a write transfer subsequent to the first transfer in a NC2 cycle, the master SHALL maintain the ADn lines after asserting DS*	D.24	90	-	30	-

Table 9 - (concluded)

Regulation		Para meter	Long cable min. ns	Long cable max. ns	Short cable min. ns	Short cable max. ns
RULE 3.28	*If the last data transfer is a write, the master SHALL deassert the AD lines prior to deasserting AS*	D.25	30	-	25	-
RULE 3.29	*In a read transfer, the master SHALL deassert the ADn lines prior to asserting DS*	D.26	130	-	25	-
OBSERVATION 3.9	The master is assured that the ADn lines will not be asserted before it asserts DS	D.27	0	-	0	-
OBSERVATION 3.10	In a read transfer the master is assured that the ADn lines will be established before WAIT is deasserted	D.28	10	-	10	-
OBSERVATION 3.11	In a read transfer in a compelled cycle, the master is assured that the ADn lines will be maintained until it deasserts DS	D.29	0	-	0	-
OBSERVATION 3.12	In a read transfer in a NC2 cycle, the master is assured that the ADn lines will be maintained a minimum time after WAIT is deasserted	D.30	10	-	10	-
OBSERVATION 3.13	The master is assured that SERR will not be asserted before it asserts DS	D.31	0	-	0	-
OBSERVATION 3.14	In case of an error, the master is assured that SERR will be asserted a minimum time before WAIT is deasserted	D.32	10	-	10	-
OBSERVATION 3.15	In case of an error in a compelled cycle, the master is assured that SERR will be maintained asserted until it deasserts DS	D.33	0	-	0	-
OBSERVATION 3.16	In case of an error in a NC2 cycle, the master is assured that SERR will be maintained after WAIT is deasserted	D.34	10	-	10	-

Table 10 - Slave - timing regulations

(Refer to figures 5, 6, 7, 8, 9 and 10)

Regulation		Parameter	Long cable min. ns	Long cable max. ns	Short cable min. ns	Short cable max. ns
RULE 3.30	*In a NC2 cycle, a slave SHALL ensure that WAIT is deasserted for a minimum time between successive data transfers*	D.10	50	-	50	-
RULE 3.31	*In compelled and NC2 cycles, a slave SHALL assert WAIT for a minimum time*	D.11	160	-	70	-
RULE 3.32	*A slave SHALL not assert WAIT before AS is asserted*	D.12	0	-	0	-
RULE 3.33	*In compelled and NC2 cycles, in all data transfers subsequent to the first data transfer, a slave SHALL NOT assert WAIT before the corresponding DS is asserted*	D.13	0	-	0	-
OBSERVATION 3.17	In the address phase, a slave is assured that the CLn and IDn lines will be established before AS is asserted	D.14	10	-	10	-
OBSERVATION 3.18	In the address phase, a slave is assured that the CLn and IDn lines will be maintained after AS is asserted	D.15	10	-	10	-
OBSERVATION 3.19	The slave is assured that the CLn and IDn lines will be established before DS is asserted	D.16	10	-	10	-
OBSERVATION 3.20	In any data transfer in a compelled cycle and during the first data transfer in a NC2 cycle, a slave is assured that the CLn and IDn lines will be maintained until it deasserts WAIT	D.17	0	-	0	-
OBSERVATION 3.21	In a NC1 cycle and a NC2 cycle subsequent to the first, the slave is assured that the CLn and IDn lines will be maintained a minimum time after DS is asserted	D.18	10	-	10	-
OBSERVATION 3.22	In the address phase, the slave is assured that the AD lines will be established a minimum time before AS is asserted	D.20	10	-	10	-
OBSERVATION 3.23	In the address phase a slave is assured that the AD lines will be established a minimum time before DS is asserted	D.21	10	-	10	-
OBSERVATION 3.24	In a write transfer, a slave is assured that the ADn lines will be established before DS is asserted	D.22	10	-	10	-

Table 10 - (concluded)

Regulation		Para meter	Long cable min. ns	Long cable max. ns	Short cable min. ns	Short cable max. ns
OBSERVATION 3.25	In any write transfer in a compelled cycle and in the first data transfer in a NC2 cycle, a slave is assured that the ADn lines will be maintained until it deasserts WAIT	D.23	0	-	0	-
OBSERVATION 3.26	In a NC1 cycle and a write transfer subsequent to the first in a NC2 cycle, a slave is assured that the ADn lines will be maintained after DS is asserted	D.24	10	-	10	-
OBSERVATION 3.27	If the last transfer is a write, a slave is assured that the AS line will be maintained a minimum time after the ADn lines are de-established	D.25	10	-	10	-
OBSERVATION 3.28	In a read transfer, a slave is assured that the ADn lines will be deasserted before DS is asserted	D.26	0	-	0	-
RULE 3.34	*A slave SHALL not assert the ADn lines before DS is asserted*	D.27	0	-	0	-
RULE 3.35	*In a read transfer, a slave SHALL establish the ADn lines prior to deasserting WAIT*	D.28	30	-	25	-
RULE 3.36	*In a read transfer in a compelled cycle, a slave SHALL maintain the ADn lines until DS is deasserted, and it SHALL deassert the ADn lines after DS is deasserted*	D.29	0	50	0	50
RULE 3.37	*In a read transfer in a NC2 cycle, a slave SHALL maintain the ADn lines after it deasserts WAIT, and it SHALL deassert the ADn lines after it deasserts WAIT*	D.30	30	70	25	65
RULE 3.38	*A slave SHALL NOT assert SERR before DS is asserted*	D.31	0	-	0	-
RULE 3.39	*In case of an error, a slave SHALL assert SERR prior to deasserting WAIT*	D.32	10	-	10	-
RULE 3.40	*In case of an error in a compelled cycle, the slave SHALL maintain SERR asserted until DS is deasserted*	D.33	0	50	0	50
RULE 3.41	*In case of an error in a NC2 cycle, the slave SHALL maintain SERR asserted after it deasserts WAIT*	D.34	30	70	25	65

4 Arbitration

4.1 Introduction

The purpose of the VICbus arbitration mechanism is to allocate the use of the DTB to potential masters and interrupt handlers in response to their requests. A bus request / bus grant mechanism is used with daisy-chain prioritisation.

A master or **interrupt handler** will use an associated **requester** to send **bus requests** to the **arbiter**.

A **rotating arbiter** scheme is used, whereby a device granted bus mastership also becomes the arbiter for the next arbitration cycle. This scheme in effect produces a "fair" arbitration scheme, and reduces arbitration latency. It also contributes to the strategy for creating "robust" VICbus systems (see 6.5) by eliminating the arbiter as a possible single point of failure in the system.

4.2 Lines

The arbitration system uses the following lines:

ALOCK	Arbitration Lock: used to inhibit the arbitration process during the connection or disconnection of a device to or from VICbus (see 6.6)
AP	Arbiter Present: asserted by the current arbiter
BBSY	Bus Busy: asserted by a requester to indicate that its associated master is becoming bus master
BG	Bus Grant: a line daisy-chained through all requesters which is asserted by the arbiter to indicate allocation of DTB mastership
BGIN	Bus Grant In: the BG line is called BGIN when it enters a device
BGOUT	Bus Grant Out: the BG line is called BGOUT when it leaves a device
BGLOOP	Bus Grant Loop: the BGOUT of the last device on the VICbus is connected to the BGIN of the first device
BR	Bus Request: asserted by a requester to request mastership of the DTB from the arbiter

4.3 Arbitration protocol

A requester asks the arbiter for DTB mastership by asserting the BR line, and then waits until it receives the assertion of the bus grant on BGIN before asserting BBSY, at which time it also deasserts BR. It will not propagate the bus grant signal from BGIN to BGOUT. (Non-requesting devices will pass the BG signal on to the next device **downstream**, in the sense of the BG daisy-chain, by asserting BGOUT.) On receipt of the assertion of BBSY from the requester, the arbiter will deassert BG.

If several devices request the use of the DTB simultaneously, their priority is decided by their position on the bus grant daisy-chain relative to the arbiter in the downstream direction; those physically nearer have the higher priority.

Since arbitration can occur concurrently with a DTB cycle, and the previous master may not have completed its cycle when a new master is selected, the new master cannot necessarily take control of the DTB immediately. In this situation, the device newly granted bus mastership is termed the **pending master**, and that which is still executing a DTB cycle, the **current master**. A pending master detects when the current master completes its cycle, and thus when it can start to use the

DTB, by monitoring AS.

When the current arbiter grants the DTB to a requesting device, it also passes the arbiter function to that device by deasserting AP when it asserts BGOUT. The device gaining mastership will take over the function of arbiter, and assert AP. In this way, the arbiter function moves or "rotates" between devices. In the case that the AP signal remains deasserted for longer than a specified time, the arbiter is considered "lost", and the current INTSEL generator (see clause 6) will assume the function and assert AP.

Note that one of the functions of the VICbus electrical terminator (see 7.5) is to connect the BGIN/BGOUT daisy chain to the BGLOOP line, to ensure that devices upstream of the arbiter (in the sense of the propagation of the BG signal) will receive bus grants.

4.4 Arbiter

The arbiter is a logical module which performs a dialogue with the bus requesters associated with potential masters and interrupt handlers in order to allocate the use of the DTB in an orderly fashion.

As described above, the VICbus uses a rotating arbiter whereby after successful arbitration, the device gaining mastership will assume the function of arbiter for the next arbitration cycle. However, for the purposes of fault-finding and system integration, it is permitted to fix the arbiter function onto a particular master. In such a case, the device wishing to retain the arbiter function simply does not deassert AP during the arbitration process.

> ***RULE 4.1***
> *All devices which can assume bus mastership SHALL also be equipped to assume the function of arbiter.*

> ***PERMISSION 4.1***
> For purposes of fault-finding, the arbiter function MAY remain associated with one particular device.

> ***RECOMMENDATION 4.1***
> Design devices to have a user-selectable ability to retain the arbiter function (by not deasserting AP during the arbitration process).

> ***RULE 4.2***
> *When a device is granted bus mastership:*
> *a) if the AP signal is in the deasserted state then it SHALL assume the function of arbiter, and SHALL assert AP;*
> *b) if the AP signal is in the asserted state then it SHALL NOT assume the function of arbiter, and SHALL NOT assert AP.*

During cable connection or disconnection, ALOCK will be asserted in robust systems to inhibit arbitration whilst the BG daisy-chain continuity relay contacts settle (described in detail in 6.6). An arbiter, recognising the assertion of ALOCK, will relinquish its function and deassert AP, and, if necessary, BGOUT.

> ***RULE 4.3***
> *When ALOCK is asserted an arbiter SHALL deassert AP and BGOUT.*

4.5 Requester

Requesters are logical modules associated with all potential masters and interrupt handlers which dialogue with the arbiter to request and accept DTB mastership when a data transfer or an IACK cycle is required.

When a requester, which has asserted the BR signal, detects the assertion of BGIN, it takes the following actions:

a) asserts BBSY;

b) deasserts BR;

c) informs the associated on-board arbiter that it must assume the arbiter function (if AP is deasserted);

d) informs the master or interrupt handler that it is now the pending master.

For similar reasons to those discussed in 4.4, during cable connection or disconnection a requester, recognising the assertion of ALOCK, will ignore the assertion of BGIN.

> ***RULE 4.4***
> *When ALOCK is asserted a requester SHALL ignore BGIN.*

Requesters may use one of two strategies for relinquishing DTB mastership:

a) a "Release When Done" (RWD) requester will release the DTB by deasserting BBSY as soon as its associated master or interrupt handler has completed its use of the DTB;

b) a "Release On Request" (ROR) requester will retain mastership, even after completing its use of the DTB. It will only deassert BBSY when another requester asserts BR and its associated master or interrupt handler wishes to do so.

A device with a ROR requester can use the DTB immediately with no further arbitration cycle if its requester is still asserting BBSY following a previous arbitration cycle.

A device with a RWD requester must arbitrate for the DTB every time it wishes to use it *even if it is the current arbiter*. This is particularly inefficient in a system with a rotating arbiter since, if the device is the current arbiter, the BG signal will have to travel around the entire daisy-chain (including BGLOOP) before its requester can assert BBSY.

> ***RECOMMENDATION 4.2***
> Design requesters to use the ROR algorithm.

4.6 Transfer of DTB mastership

A current master or interrupt handler will inform its requester when it has completed its use of the DTB, and thus when the requester may deassert BBSY. This can be after the device has finished its use of the DTB, or during the last DTB cycle which the device intends to perform.

> ***RECOMMENDATION 4.3***
> Design masters and interrupt handlers such that their requesters can deassert BBSY at the earliest opportunity during execution of the last DTB cycle which the device intends to perform.

OBSERVATION 4.1
The application of RECOMMENDATION 4.3 leads to the most efficient use of the DTB, by allowing arbitration to proceed in parallel with execution of the last DTB cycle.

Because arbitration can proceed in parallel with a DTB cycle, a device newly granted permission to use the DTB cannot necessarily start to use the DTB immediately.

RULE 4.5
When a device is granted permission to use the DTB, it SHALL NOT assert any of the DTB signals until it detects that AS is deasserted.

OBSERVATION 4.2
A pending master or interrupt handler may assert BBSY while a current master is asserting AS.

PERMISSION 4.2
A device MAY decide not to use the DTB once granted mastership.

4.7 Loss of the arbiter

The deassertion of the AP signal for longer than a time specified by RULE 4.14 (table 11) indicates a failure of the arbitration process. This can occur, for example, when the requester withdraws its bus request as a result of system "hot" reconfiguration. One of the tasks of the current INTSEL generator is to detect this situation, and then to assume the function of arbiter by asserting AP.

RULE 4.6
The current INTSEL generator SHALL monitor AP, and if AP is deasserted for longer than a time-out period (see RECOMMENDATION 4.4*), it SHALL become the current arbiter and assert AP.*

OBSERVATION 4.3
When the current INTSEL generator asserts AP after a loss of arbiter, the previous arbiter will deassert BGOUT (if it has not already done so) in response as specified by RULE 4.16. However, owing to cable and BGIN to BGOUT delays, the deassertion of BGOUT may not reach the INTSEL generator for some time, it must therefore take care that its BGOUT has been deasserted for the period specified in RULE 4.15 before it grants the DTB to a requester.

RULE 4.7
When the current INTSEL generator becomes the current arbiter as specified in RULE 4.6, it SHALL ensure that its BGOUT signal is deasserted for at least the time specified in RULE 4.15 *before it asserts BGOUT in response to BR.*

OBSERVATION 4.4
If a device retains the arbiter function (for the purpose of fault diagnosis) and there is a failure of the arbitration process, the INTSEL generator will not detect this failure since AP is not deasserted. However, the arbiter itself is able to detect the failure by the absence of BBSY for a time longer than that specified in RULE 4.14 after its assertion of BGOUT. In which case, it should take care to ensure that BGOUT is deasserted for the period specified in RULE 4.15 before reasserting BGOUT.

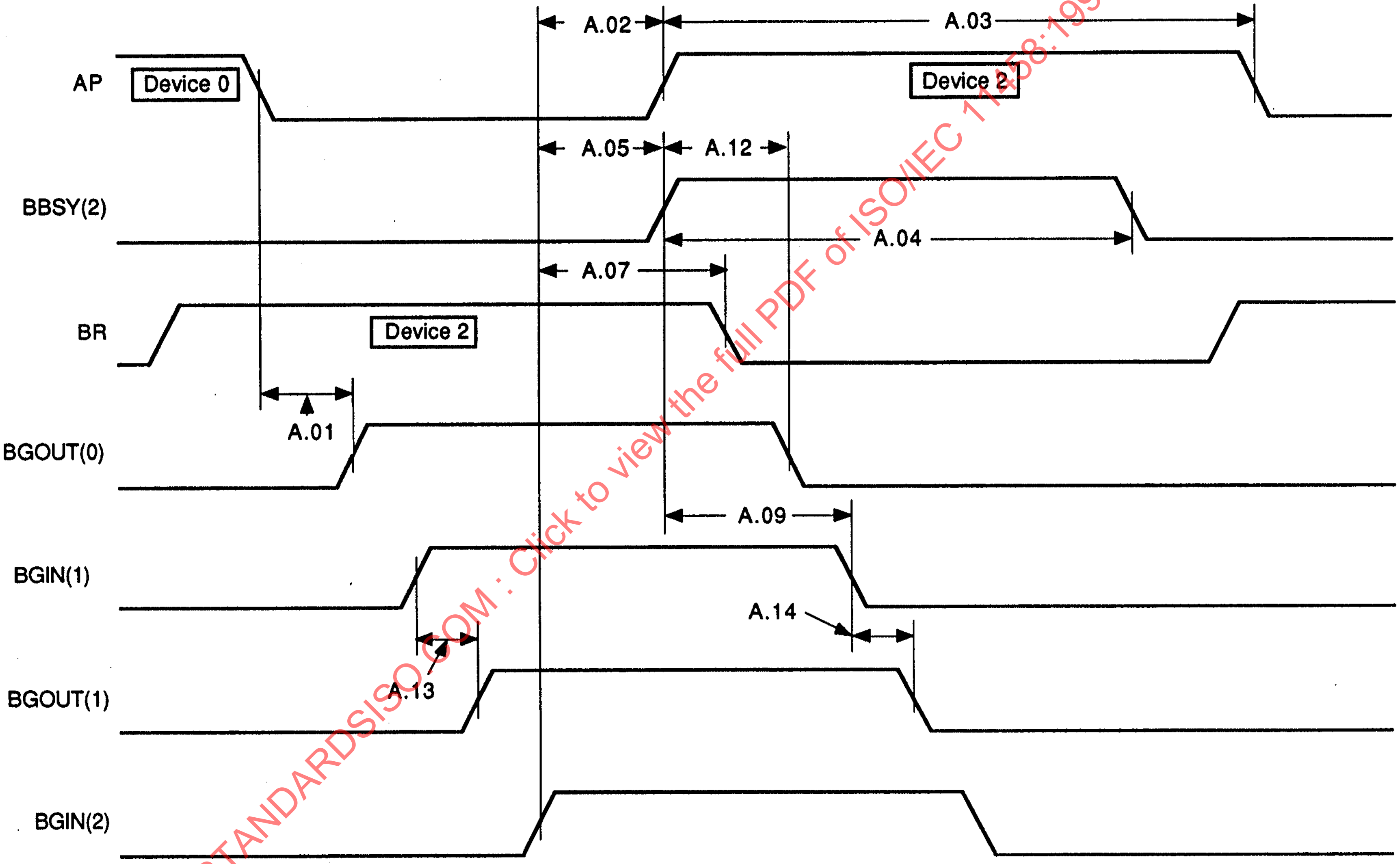

Figure 11 - Arbitration-1

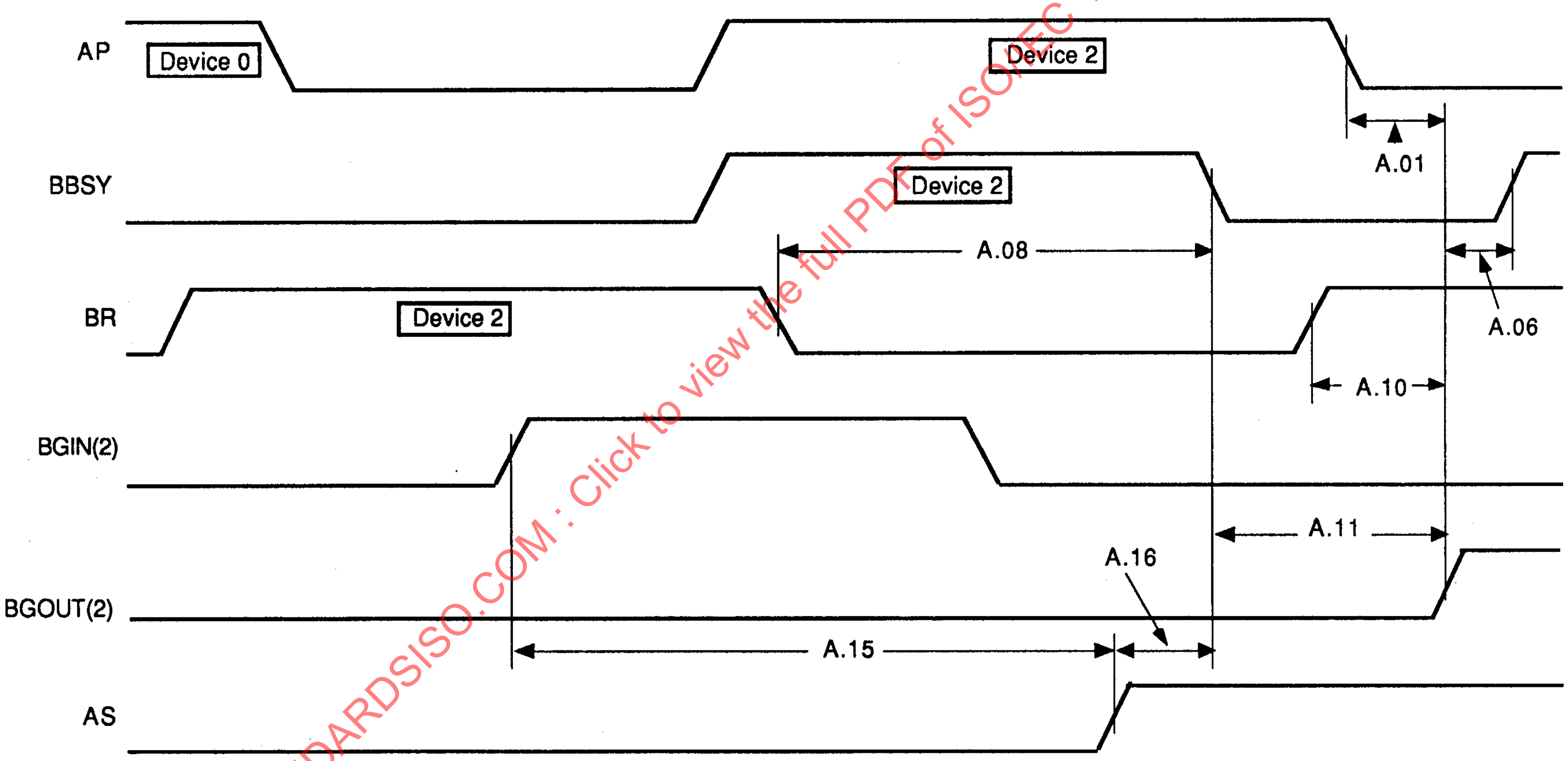

Figure 12 - Arbitration-2

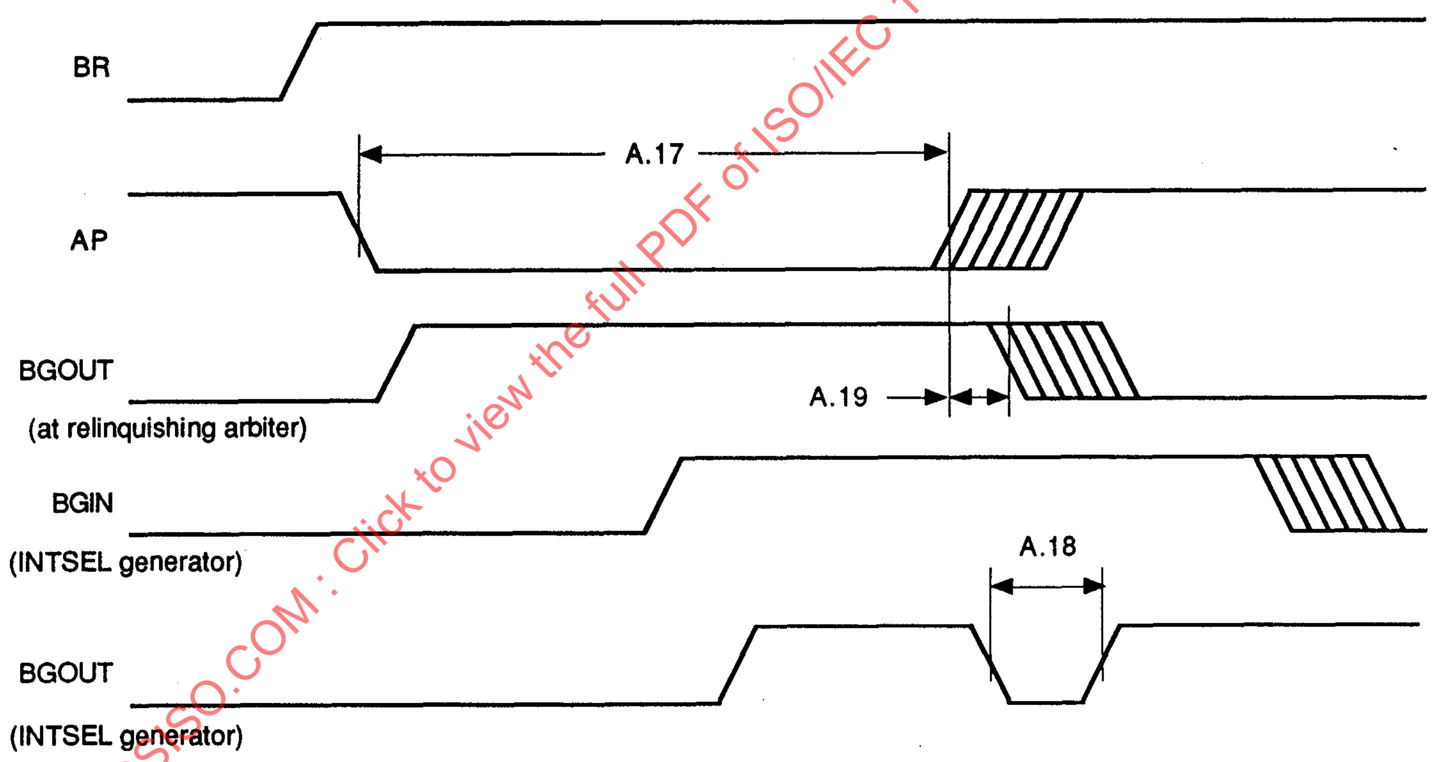

Figure 13 - Arbitration-3

4.8 *Arbitration timing rules*

Table 11 - Arbiter - timing regulations

(Refer to figures 11, 12 and 13)

Regulation		Para meter	min. ns	max. ns
RULE 4.8	*When the arbiter wishes to relinquish its function, it SHALL deassert AP before asserting BGOUT*	A.01	140	230
RULE 4.9	*The arbiter SHALL assert AP within a maximum time after BGIN is asserted, if AP is not already asserted*	A.02	0	250
RULE 4.10	*The arbiter SHALL assert AP for a minimum time:* *a) long cable* *b) short cable*	 A.03 A.03	 160 70	 - -
OBSERVATION 4.5	The arbiter is assured that BBSY will be asserted for a minimum time	A.04	50	-
OBSERVATION 4.6	The arbiter is assured that BBSY will not be asserted before it asserts BGOUT	A.06	0	-
OBSERVATION 4.7	The arbiter is assured that BR will be maintained asserted until it asserts BGOUT	A.07	0	-
RULE 4.11	*The arbiter SHALL NOT assert BGOUT before BR is asserted*	A.10	0	-
RULE 4.12	*The arbiter SHALL NOT assert BGOUT before BBSY is deasserted*	A.11	0	-
RULE 4.13	*The arbiter SHALL maintain BGOUT asserted until BBSY is asserted and it SHALL deassert BGOUT after BBSY is asserted*	A.12	0	50
RECOMMENDATION 4.4 ***OBSERVATION 4.8***	The deassertion of AP for longer than this time SHALL indicate the failure of the arbitration process This is the minimum time which an arbiter will use to detect a failure of the arbitration system	A.17	10 000	-
RULE 4.15	*The arbiter SHALL maintain BGOUT deasserted for a minimum time*	A.18	50	-
RULE 4.16	*When the current INTSEL generator asserts AP after an arbitration timeout, the previous arbiter SHALL deassert BGOUT*	A.19	-	50

Table 12 - Requester - timing regulations

(Refer to figures 11 and 12)

Regulation		Para meter	min. ns	max. ns
RULE 4.17	*The requester SHALL assert BBSY for a minimum time*	A.04	160	-
RULE 4.18	*The requester SHALL NOT assert BBSY before BGIN is asserted and SHALL assert BBSY after BGIN*	A.05	0	250
RULE 4.19	*The requester SHALL maintain BR asserted until BGIN is asserted*	A.07	0	-
RULE 4.20	*The requester SHALL deassert BR before it deasserts BBSY*	A.08	30	-
OBSERVATION 4.9	The requester is assured that BGIN is maintained asserted until it asserts BBSY	A.09	0	-
RULE 4.21	*The requester SHALL NOT assert BGOUT before BGIN is asserted and SHALL assert BGOUT after BGIN*	A.13	0	250
RULE 4.22	*The requester SHALL deassert BGOUT after BGIN is deasserted*	A.14	0	50
RULE 4.23 ***OBSERVATION 4.10***	*The requester SHALL maintain BGOUT deasserted for a minimum time.* A requester can achieve this by ensuring that it deasserts BGOUT for at least the same time as BGIN is deasserted at its input, since RULE 4.15 requires that BGOUT (from an arbiter) is deasserted for at least this time		50	-
RULE 4.24	*The master SHALL NOT assert AS before BGIN is asserted at its associated requester*	A.15	0	-
RULE 4.25	*The requester SHALL maintain BBSY asserted after the assertion of AS*	A.16	10	-

5 Interrupts

5.1 *Introduction*

Given the practical limitations imposed by cable and connector sizes, VICbus provides an interrupt mechanism which will allow simple slave devices to operate with minimal implementation overheads. For more complex devices, the use of the DTB for message passing should be envisaged as an alternative to the "hardware" interrupt mechanism described in this clause.

Interrupters use the interrupt system to send **interrupt requests** to interrupt handlers. The interrupt handlers then may use DTB IACK cycles to fetch **vector** information from the interrupter. The DTB, the arbitration system and the interrupt system may all be involved in the VICbus interrupt process.

5.2 *Lines and signals*

The interrupt mechanism uses:

INT7-INT0	The eight interrupt request lines
INTSEL1, INTSEL0	The 125kHz clocks used to multiplex the MINTn signals onto the INTn lines
MINT31-MINT00	32 multiplexed interrupt request signals

5.3 *Interrupt request signal selection*

The 32 MINTn signals are multiplexed onto the eight physical INTn lines by means of the INTSEL0 and INTSEL1 two phase clock, as shown in table 13. The interrupt handler samples the state of the INTn lines at all transitions of both of the clocks.

Table 13 - Interrupt request multiplexing

MINTn	INTSEL1 state	INTSEL0 state	INTn
00	0	0	0
01	0	0	1
02	0	0	2
03	0	0	3
04	0	0	4
05	0	0	5
06	0	0	6
07	0	0	7
08	0	1	0
09	0	1	1
10[1]	0	1	2
11	0	1	3
12	0	1	4
13	0	1	5
14	0	1	6
15	0	1	7
16	1	1	0
17	1	1	1
18	1	1	2
19	1	1	3
20	1	1	4
21	1	1	5
22	1	1	6
23	1	1	7
24	1	0	0
25	1	0	1
26	1	0	2
27	1	0	3
28	1	0	4
29	1	0	5
30	1	0	6
31	1	0	7

1) Example: when INTSEL1 = 0 and INTSEL0 = 1, the MINT10 signal is carried by the INT2 line.

RULE 5.1
When making an interrupt request, interrupters SHALL assert their assigned INTn line only during the combination of states of the INTSEL0 and INTSEL1 lines specified by table 13.

OBSERVATION 5.1
Whilst an interrupter's MINTn signal is asserted, it will, in turn, assert the specified INTn line whenever the appropriate combination of INTSEL1 and INTSEL0 is present. Interrupt handlers can thereby reconstruct the MINTn signal with a latency of between one and two INTSEL periods (8-16µs).

5.4 Interrupt protocol

When one device wishes to interrupt another, its associated interrupter will assert a previously assigned MINTn signal. The interrupt handler assigned to this MINTn will receive this assertion, and perform implementation dependent actions within its device (for example set a flag, assert an interrupt request on the local backplane bus, etc.).

The interrupt handler may then request bus mastership and perform a DTB IACK cycle. This cycle will carry an interrupter number (IN) equal to the number of the asserted MINTn in order to read a vector value from the interrupter. The vector may be 8, 16 or 32 bits wide, right aligned (refer to table 14). The interrupt request may be reset directly by the IACK cycle, or a master associated with the interrupt handler may make a specific action within the requesting device via a standard DTB cycle, which will indirectly reset the request.

In most simple cases, it will be convenient to set IN equal to DN, when the source of the interrupt will be directly associated with a device.

Instead of the two types of interlocked actions, described above, a requester may simply assert and deassert its MINTn in a synchronous, protocol-less manner requiring no action by an interrupt handler. The three types of interrupter are more fully described in 5.5.

The actions taken by the interrupt handler and its associated VICbus master depend on the type of interrupter involved, and are summarised in table 15.

RULE 5.2
IACK cycles SHALL obey the same timing rules as compelled DTB read cycles and SHALL use the byte alignment specified in table 14.

OBSERVATION 5.2
The IACK cycle is essentially a compelled DTB read cycle, with the interrupt handler acting in a similar manner to a master, except that the identification lines are used to specify an interrupter number (IN) rather than a device number (DN).

Table 14 - IACK byte alignment

	Address phase		Data phase							
Lines ⇒	AD01	AD00	CL1	CL0	AD31-AD24	AD23-AD16	AD15-AD08	AD07-AD00
Signals ⇒	A01	LWORD	DSEL1	DSEL0	D31-D24	D23-D16	D15-D08	D07-D00
8 bit vector	X[1)]	1	0	1	-	-	-	byte(1)
16 bit vector	X	1	1	1	-	-	byte(0)	byte(1)
32 bit vector	X	0	1	1	byte(0)	byte(1)	byte(2)	byte(3)

1) X = don't care.

5.5 *Interrupter*

An interrupter can be assigned to only one MINTn signal at a time, but a device may have multiple interrupters.

Three types of interrupter are described:

a) **ROACK - Release-On-ACKnowledge**
The interrupter will assert its assigned MINTn and will reply to a DTB IACK cycle addressed to it (in which the IN value corresponds to the number of the MINTn signal asserted) by supplying a vector value and deasserting the MINTn.

b) **RORA - Release-On-Register-Access**
The interrupter will assert its assigned MINTn which it will only deassert after local action by its associated device resulting from some VICbus DTB activity originating in the interrupted device. It will, however, respond to a DTB IACK cycle by supplying a vector value.

c) **ROSA - Release-On-Synchronous-Action**
The interrupter simply asserts and deasserts its MINTn signal, respecting the appropriate timing requirements, and does not wait for the return of an IACK cycle or for any other DTB activity to reset it, directly or indirectly.

RULE 5.3
If both of the following conditions are fulfilled, a ROACK or a RORA interrupter SHALL respond to a DTB IACK cycle:
a) it has a pending interrupt request (that is it has a MINTn signal asserted);
b) its MINTn number corresponds to the value of the IN field of the IACK cycle (ID4-ID0).

RULE 5.4
A ROACK interrupter SHALL maintain its MINTn signal asserted until it responds to a DTB IACK cycle.

RULE 5.5
A RORA interrupter SHALL maintain its MINTn signal asserted until reset by explicit action originating in its associated device, following a VICbus DTB cycle.

PERMISSION 5.2
A ROSA interrupter MAY deassert its MINTn signal at any time provided that the timing requirements for INTn are met (see RULE 5.8).

OBSERVATION 5.3
A ROSA interrupter MAY assert its assigned INTn line for one or more consecutive occurrences of the appropriate combinations of INTSEL1 and INTSEL0.

PERMISSION 5.3
Since IACK vector fetches are always DTB read cycles and the necessary address information is carried in the IN field, the interrupter MAY choose not to monitor WRITE or A31-A01.

PERMISSION 5.4
A device MAY use more than one interrupter, and consequently more than one MINTn signal.

5.6 Interrupt handler

The interrupt handler has the following tasks:

a) to prioritise the MINTn signals assigned to it;
b) to signal the interrupt to its associated device;
c) to generate a DTB IACK cycle if necessary.

Interrupt handlers may monitor one or more MINTn signals, and more than one interrupt handler may be associated with each MINTn signal. However, in the latter case only one (the **active interrupt handler**) may reply to the assertion of the interrupt request with a DTB IACK cycle, if this is appropriate.

RECOMMENDATION 5.1
Prioritise the MINTn signals received by an interrupt handler in ascending order of their number, MINT00 having the lowest priority and MINT31 the highest.

PERMISSION 5.5
Since IACK vector fetches are always read cycles and the necessary address information is carried in the IN field, the interrupt handler MAY choose not to drive WRITE or A31-A01.

PERMISSION 5.6
More than one interrupt handler MAY react on the assertion of a MINTn signal by making some internal action within its device.

OBSERVATION 5.4
It is the responsibility of the user to ensure that only one interrupt handler will acknowledge an interrupt by performing a DTB IACK cycle.

Table 15 - Summary of interrupt protocol actions

Step	Interrupter type	Interrupt handler action
1	All	An interrupt handler recognises and latches the assertion of any MINTn signals to which it has been assigned on the transition of INTSEL0 or INTSEL1 following its recognition of an INTn assertion
2	All	The interrupt handler prioritises the MINTn assertions it has recognised
3	ROACK and RORA	The active interrupt handler generates an Interrupt Acknowledge (IACK) DTB cycle to fetch a vector from the interrupting device. This action will either be automatic or under the control of a processor in the VICbus device or on the associated backplane bus
	ROSA	The interrupt handler informs its associated device of the assertion of the MINTn line associated with it. The device may then take actions such as the setting of a status flag and / or the generation of a local interrupt
4	ROACK	The IACK cycle causes the original MINTn signal to be deasserted
	RORA	No action: the MINTn signal will remain asserted until the VICbus master takes explicit action by means of a DTB cycle to reset the interrupt
	ROSA	No action: the MINTn signal will remain asserted until deasserted by the interrupter

5.7 *Timing regulations*

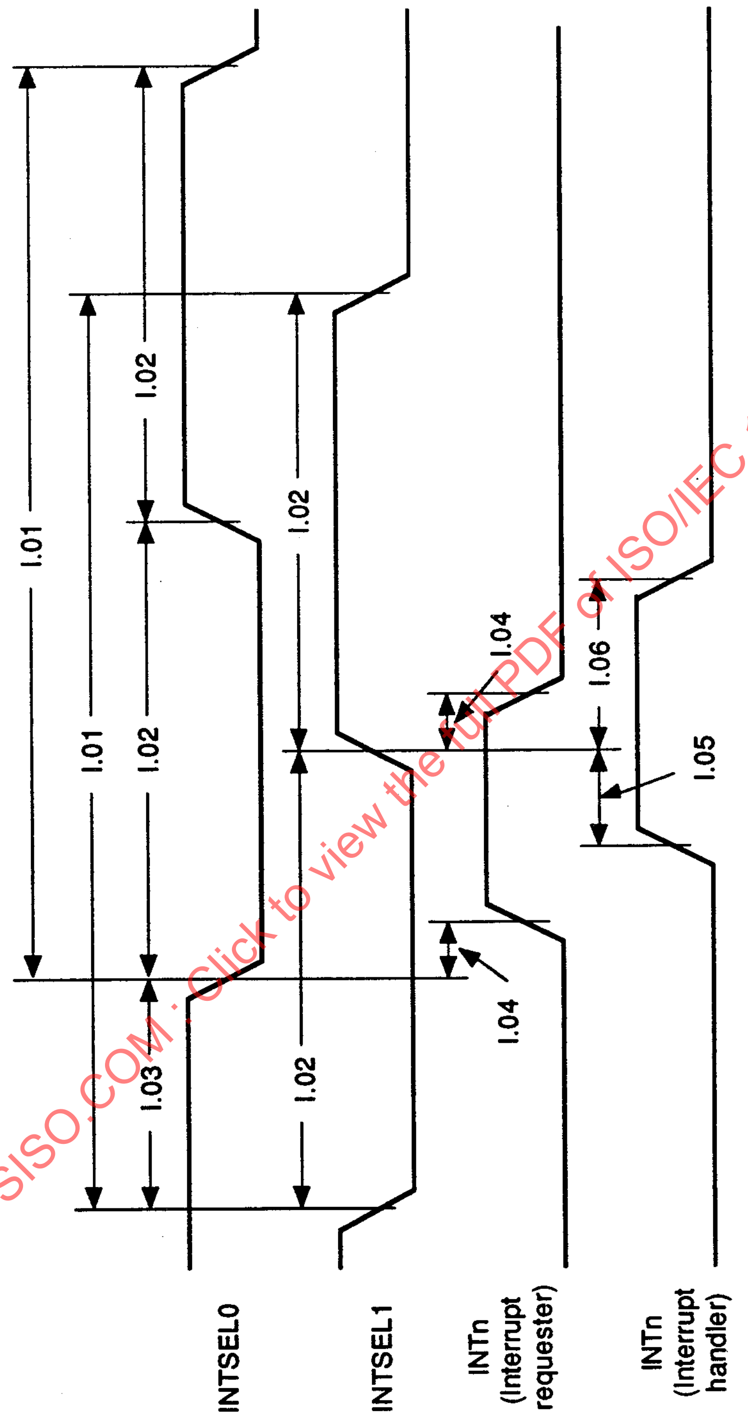

Figure 14 - Interrupt request selection timing

Table 16 - Interrupts - timing regulations

(Refer to figure 14)

Regulation		Para meter	min. ns	max. ns
RULE 5.7	*INTSEL0 and INTSEL1 SHALL be driven by 125 kHz, 50 % nominal duty cycle clocks as follows:*			
	a) period	I.01	7 920	8 080
	b) asserted and deasserted time	I.02	3 960	4 040
	c) phase relationship	I.03	1 980	2 020
RULE 5.8	*An interrupter SHALL only assert or deassert the INTn line after the appropriate INTSEL edge*	I.04	100	500
OBSERVATION 5.5	The interrupt handler is assured the following set-up and hold times of the INTn line relative to the next INTSEL edge:			
	a) set-up time, short cable	I.05	1 000	-
	b) set-up time, long cable	I.05	200	-
	c) hold time	I.06	50	-

6 Utilities

6.1 Introduction

The utility lines and signals are listed below, and discussed in the subsequent subclauses:

ALOCK	Arbitration Lock
DEVFAIL	Device Failure
INTSEL0	Interrupt Request Select 0
INTSEL1	Interrupt Request Select 1
VICRESET	System Reset

This clause also includes descriptions of the **INTSEL generator** selection mechanism, system reset, online and offline states and the cable connection and disconnection procedure.

6.1.1 Arbitration lock line ALOCK

The ALOCK signal has two functions:

a) arbitration inhibit: during the connection or disconnection of a device to or from the VICbus cable, the arbitration mechanism will be subject to perturbations. The ALOCK line may be asserted by the device to inhibit the arbitration process during this time. Details are given in 6.6;

b) INTSEL generator selection: the process leading to the selection of the INTSEL generator is triggered by the deassertion of ALOCK, as described in 6.2.

6.1.2 Device failure line DEVFAIL

The DEVFAIL signal may be asserted in response to the assertion of a global or selective reset to indicate that the device is performing its internal initialisation and self-tests. In addition, it may be asserted to indicate a major fault in the device or in an associated backplane bus system at any time during operation. DEVFAIL will normally be used in conjunction with the reset mechanism and the OPER bit in the command and status register (as well as the device operational register, if appropriate).

RECOMMENDATION 6.1

Use the DEVFAIL signal to indicate one or more of the following:

a) device failure: the device is unable to function correctly and requires attention from a system control process;

b) internal initialisation and self-test: the device is temporarily unable to function correctly until it has completed its internal initialisation, self-test, etc., and is ready to participate in bus activity;

c) failure of an associated backplane bus: a device on an associated backplane bus is signalling a failure condition.

6.1.3 Interrupt request select lines INTSEL0 and INTSEL1

The interrupt request select signals (INTSEL0 and INTSEL1) are independent, non-gated, fixed frequency, 125kHz clocks, where INTSEL0 leads INTSEL1 by 90° in phase, and are produced by an INTSEL generator. They are used to multiplex the MINT31-MINT00 interrupt request signals onto the INT7-INT0 lines. See 5.7 for the timing regulations concerning INTSEL0 and INTSEL1.

RULE 6.1
INTSEL0 and INTSEL1 SHALL only be driven by the current INTSEL generator.

6.1.4 VICbus reset line VICRESET

The assertion of the VICRESET signal will simultaneously reset all devices connected to the VICbus, as described in 6.3.

6.2 INTSEL generator selection

Robust operation (see 6.5) requires that more than one potential INTSEL generator is present on a VICbus, and that, in case of failure or of deliberate removal of the device carrying the current INTSEL generator, its function is automatically taken up by another.

RECOMMENDATION 6.2
Equip all potential master devices to be potential INTSEL generators.

The absence of the INTSEL generator is detected by the lack of a transition on either of the INTSEL0 or INTSEL1 lines during a specified time. All potential INTSEL generators will monitor the signals, and if a failure is sensed will assert ALOCK for the minimum specified time (unless ALOCK has already been asserted, as described in 6.6, for the purposes of cable connection or disconnection, when it need not do so). The subsequent deassertion of ALOCK is used to synchronise the INTSEL generator selection procedure.

On detection of the deassertion of ALOCK, all potential INTSEL generators will count periods of 64 µs (the **INTSEL selection counter**). If such a device counts a number of these periods equal to the value of its own device number without recognising INTSEL0 and INTSEL1, it will assume the function of INTSEL generator and start to drive the INTSEL0 and INTSEL1 signals. Refer to figure 15 and table 17 for the detailed timing.

PERMISSION 6.1
When a potential INTSEL generator recognises the loss of the current INTSEL generator, it MAY assert ALOCK without regard to the current state of ALOCK.

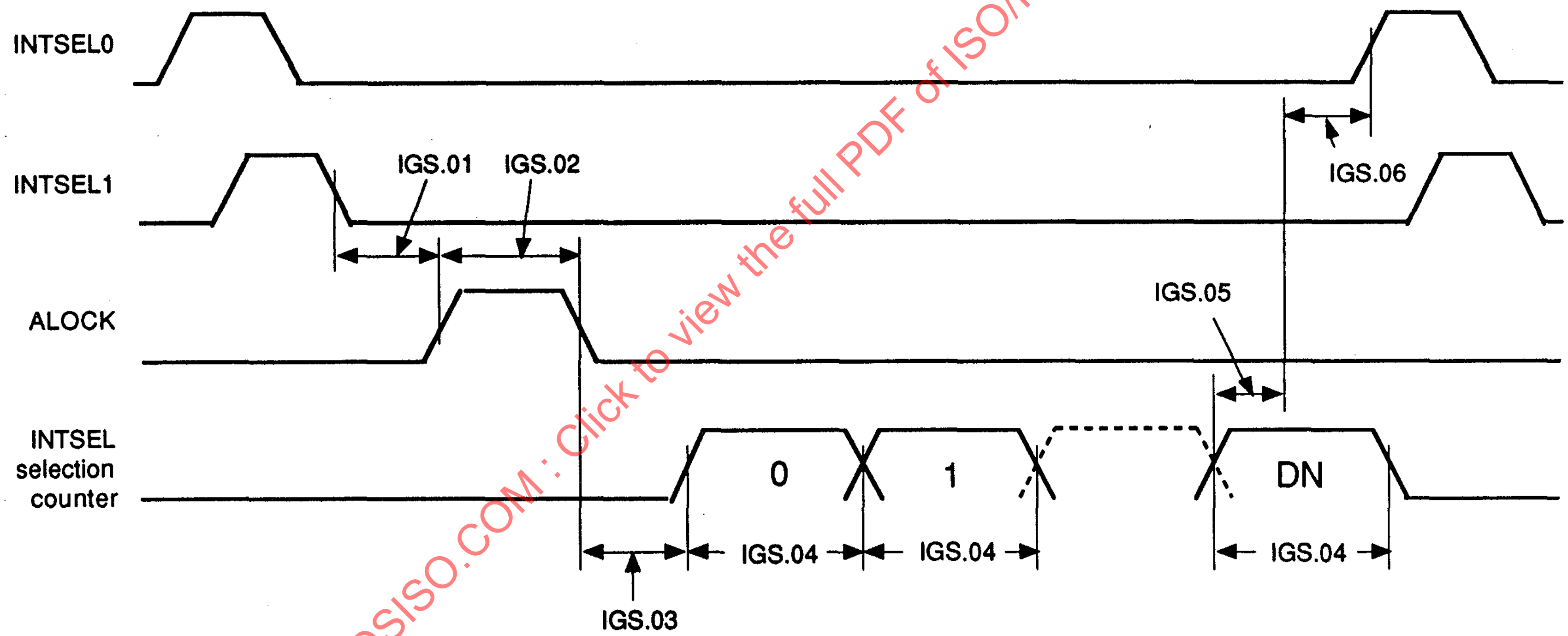

Figure 15 - INTSEL generator selection

Table 17 - INTSEL generator selection - timing regulations

(Refer to figure 15)

Regulation		Para meter	min. µs	max. µs
RULE 6.2	*The absence of a transition of either INTSEL0 or INTSEL1 SHALL indicate the loss of the INTSEL generator*	IGS.01	2,1	16,0
RULE 6.3	*ALOCK SHALL be asserted for a minimum of two cable propagation times:*			
	a) long cable	IGS.02	1,2	-
	b) short cable	IGS.02	0,36	-
RULE 6.4	*When the loss of the INTSEL generator is recognised according to RULE 6.2, the INTSEL selection counter SHALL start when the deassertion of ALOCK is detected*	IGS.03	0	3,0
OBSERVATION 6.1	The INTSEL selection counter will be restarted if ALOCK is subsequently asserted by another device			
OBSERVATION 6.2	ALOCK is also asserted during "hot" connection and disconnection in robust systems, in which case the INTSEL selection process will only take place if the INTSEL generator is "lost"			
RULE 6.5	*During the INTSEL generator selection process, all competing devices SHALL count with this period starting with the value 0 (zero)*	IGS.04	63,6	64,4
RULE 6.6	*If no assertion of INTSEL0 and INTSEL1 is detected after this time, a potential INTSEL generator whose device number corresponds to its counter value SHALL become the current the INTSEL generator*	IGS.05	30,0	34,0
RULE 6.7	*A newly selected current INTSEL generator SHALL produce the first transition of INTSEL0 or INTSEL1 within this time*	IGS.06	0	12,0

6.3 Reset

VICbus devices may be reset by global or selective action as described below. Following a reset action, a device may carry out an internal initialisation and self-test procedure, or may require external intervention over VICbus, or from some intelligence connected to an associated backplane bus. During its initialisation, the device may assert DEVFAIL to warn the rest of the system, see 6.1.2.

RECOMMENDATION 6.3
Design VICbus devices such that during reset and internal initialisation they will cause no spurious activity on the VICbus.

6.3.1 Global reset - VICRESET

All VICbus devices are simultaneously reset by the assertion of the VICRESET line.

RULE 6.8
All devices SHALL monitor VICRESET.

RULE 6.9
VICRESET SHALL be asserted for a minimum of 200 ms.

RULE 6.10
Following the assertion of VICRESET, all devices SHALL immediately deassert all lines and SHALL return to the online-disabled state.

OBSERVATION 6.3
RULE 6.10 implies that the device will stop all activity, as a master, a slave, interrupter, interrupt handler, arbiter, INTSEL generator or requester, without waiting for any protocols to terminate.

PERMISSION 6.2
VICRESET MAY be asserted by any device to initialise the system from a manual push-button or under program control.

6.3.2 Selective reset

Individual VICbus devices may be reset by writing 1 into the SRESET bit of its control and status register (CSR), or, if the reset register (RR) is implemented, by writing 1 into the RR bit corresponding to the device number.

RULE 6.11
Devices which detect the assertion of the SRESET bit in their CSR, or the bit corresponding to their device number in the RR SHALL deassert all lines after waiting for the completion of the cycle during which these bits were asserted, and SHALL return to the online-disabled state.

OBSERVATION 6.4
The selective reset is received via a normal DTB cycle, the completion of which must be allowed before the reset action is taken.

OBSERVATION 6.5
Since selective resets are generated by means of normal DTB write cycles, they may also be broadcast.

6.4 Online and offline states

VICbus devices may be in one of three states: **offline**, **online-disabled** or **online-enabled** (the latter implies the presence of an associated backplane bus). The permitted actions in each of the three states are summarised in table 19, and the applicable regulations are given in 6.4.1.

A device in the offline state has been logically removed from the VICbus cable and consequently makes neither any response to, nor will cause any activity on the VICbus. However, it must ensure the continuity of the BG daisy-chain line if it remains attached to the VICbus cable. The offline state may be achieved by local action, either by means of a demand initiated from a manual switch (the **online switch**) or under local software control, for instance by setting a bit in a local command register, or remotely through the action of the ONLINE bit in the CSR (see 8.3). Once set offline by remote action, a device can only be returned to the online state by local action, and then only if the online switch is in the online position.

Table 18 - Summary of the online / offline state of a device following various actions

Action	Next state	Relevant regulation
Online switch moved to offline	Offline	RULE 6.14
Following power-up	Any	PERMISSION 6.3
Following the assertion of VICRESET	Online-disabled	RULE 6.10
Following selective reset	Online-disabled	RULE 6.11
CSR ONLINE bit written with 0	Offline	RULE 8.4
CSR ONLINE bit written with 1	No action	RULE 8.4
Local action to go online	Online	RECOMMENDATION A.4
Local action to go offline	Offline	RECOMMENDATION A.4

Table 19 - Summary of actions permitted in the three online / offline states

<table>
<tr><th></th><th colspan="3">Permitted device actions if the device state is:</th></tr>
<tr><th>Function</th><th>Offline</th><th>On-line disabled</th><th>On-line enabled</th></tr>
<tr><td>DTB master</td><td>None</td><td colspan="2">— May generate direct cycles
— May generate transparent cycles
— May forward cycles from an associated backplane bus</td></tr>
<tr><td rowspan="2">DTB slave</td><td rowspan="2">None</td><td colspan="2">— May respond to direct cycles</td></tr>
<tr><td>— May not respond to transparent cycles</td><td>— May respond to transparent cycles by generating corresponding cycles on an associated backplane bus</td></tr>
<tr><td>Requester</td><td>None</td><td colspan="2">— May generate bus requests</td></tr>
<tr><td rowspan="2">Interrupter</td><td rowspan="2">None</td><td colspan="2">— May generate interrupt requests</td></tr>
<tr><td>— May not forward interrupts generated by a device connected to an associated backplane bus, as it cannot handle IACK cycles</td><td>— May forward interrupts generated by a device connected to an associated backplane bus</td></tr>
<tr><td>Interrupt handler</td><td>None</td><td colspan="2">— May handle interrupts</td></tr>
</table>

6.4.1 Regulations

The following regulations govern how a device's online / offline state affects its operation. Table 8 summarises the conditions under which a slave will participate in a DTB cycle, and thus refers to these regulations.

6.4.1.1 Online-enabled device

PERMISSION 6.3
An online-enabled device MAY generate, and participate in, all VICbus activity which it is designed to support.

OBSERVATION 6.6
The online-enabled state is not applicable to devices not having an associated backplane bus system.

6.4.1.2 Online-disabled device

RULE 6.12
An online-disabled slave device SHALL NOT participate in transparent DTB cycles.

OBSERVATION 6.7
An online-disabled device may generate (as DTB master) any type of DTB cycle, including transparent cycles; and may do so whether the originating master is within the device, or is on its associated backplane bus (if the device is an interface).

RULE 6.13
An online-disabled interface device SHALL NOT generate VICbus interrupts in response to interrupts on its associated backplane, and SHALL NOT respond to IACK cycles which relate to such interrupts.

OBSERVATION 6.8
An online-disabled interface device may generate interrupts on its associated backplane bus in response to VICbus interrupts, and may respond (as interrupt handler) to any VICbus interrupt whether the interrupt handler is within the device or is on its associated backplane bus.

6.4.1.3 Offline device

RULE 6.14
An offline device SHALL NOT respond to or generate any activity on the VICbus.

OBSERVATION 6.9
An offline device will not participate in any DTB cycles. However, in a broadcall read of the (optional) online register, it will appear as offline due to the action of the VICbus terminators.

RULE 6.15
An offline device which is physically connected to the VICbus and which is powered SHALL maintain the continuity of the BG daisy-chain line (that is SHALL pass BGIN to BGOUT).

OBSERVATION 6.10
A robust device (described in 6.5) in the offline state will also maintain the continuity of the BG daisy-chain line when powered down, by means of its BG continuity relay.

6.4.1.4 Online switch

RULE 6.16
Devices SHALL be equipped with a manual, two-position switch on their front panels (the online switch) for selection of the online (-enabled, -disabled) or offline states.

OBSERVATION 6.11
The online position of the online switch allows the device to be in either the online-enabled or online-disabled state, depending on the state of the ONLINE and TRANSP bits in the control and status register. Refer to 8.3.

RULE 6.17
Setting the online switch to the "offline" position SHALL place the device in the offline state regardless of any other condition.

RECOMMENDATION 6.4
Use a switch equipped with a mechanical lock to implement the online switch in order to prevent accidental actuation.

RECOMMENDATION 6.5
Provide a visual indication (LED or other) of the interface online state (offline, online-enabled, online-disabled) on the front panel of the device.

6.4.2 Power-up condition

PERMISSION 6.4
Devices MAY power-up in the online-enabled (if appropriate), online-disabled or offline states.

6.5 Fault tolerance

VICbus systems will be subject to the failure of critical components, to operational exigencies requiring system reconfiguration, or the addition or removal of devices for maintenance or other purposes. Two approaches to the potential failure of the system can be taken: **non-robust** or **robust**.

In non-robust systems, the VICbus will fail because of the failure of any one component or during any "hot" reconfiguration.

Robust systems are covered in this specification with the description of the measures taken to avoid single points of failure in the operation of the basic elements of the bus, and of mechanisms which allow the addition or removal of devices during VICbus operation ("hot" reconfiguration):

a) Arbitration

— arbiter: multiple arbiters, with automatic reallocation (see 4.3)

— bus grant daisy-chain: automatic assurance of continuity in case of device power failure or of deliberate removal (see 7.6);

b) Interrupts

multiple INTSEL generators, with automatic reallocation (see 6.2);

c) Bus drivers and receivers

glitch-free connection and disconnection of bussed signals is guaranteed by the characteristics of the bus drivers and receivers.

6.6 Cable connection and disconnection in robust systems

In order to remove a device from a robust VICbus in a safe manner, the operator will set it into the offline state as described in 6.4, and when given permission (for example by a visual indicator - LED or other) will remove the VICbus **cable transition module** described in 7.6. This should only happen when all activity on the VICbus originating in the device in question has ceased, and the BG line continuity has been assured by means of the **continuity relays**, also described in 7.6.

When reconnecting a device to a working, robust VICbus, the operator will physically connect the transition module carrying the VICbus cables to the device, and will request logical reconnection (for example by throwing the online switch to the "online" position).

RECOMMENDATION 6.6
Provide the operator with a visual indicator on the front panel of a robust device to indicate when it is permitted to remove the cable transition module when going offline in the powered state.

In both cases the device will assert the ALOCK line to prevent arbitration transactions whilst the bus grant lines (BG and BGLOOP) are subject to perturbations as the contacts of the continuity relays settle. During the assertion of ALOCK, the current bus master will continue its data transfers, but all active bus requesters (those with BR asserted) will ignore any activity on the bus grant daisy chain and will wait for arbitration with BR asserted. ALOCK may be asserted with no subsequent device connection or disconnection, but for a minimum of two cable propagation times to ensure its recognition by all devices.

After asserting ALOCK for the minimum time required, the device's BG continuity relay contacts will be opened or closed. ALOCK will remain asserted until the relay contacts are stable, when it may be deasserted.

RULE 6.18
If a device forms part of a robust system, then it SHALL NOT open or close its BG continuity relays sooner than 1,2 µs after the assertion of ALOCK.

RULE 6.19
A device SHALL maintain ALOCK asserted until its BG continuity relay contacts are stable (device dependent time).

OBSERVATION 6.12
RULE 6.3 specifies the minimum assertion time permitted for ALOCK.

RECOMMENDATION 6.7
When a device carrying the current INTSEL generator is to be removed from the VICbus, continue to drive INTSEL0 and INTSEL1 as long as possible before deasserting ALOCK in order to minimise the interrupt latency.

OBSERVATION 6.13
During cable connection or disconnection in robust systems, time-outs may occur on associated backplane buses since no arbitration is possible and bus mastership cannot be exchanged for perhaps several milliseconds whilst ALOCK is asserted.

7 Electrical specifications

7.1 Introduction

VICbus uses differential signal transmission on twisted-pair cables with open collector / emitter line drivers, giving a wired-OR capability which permits simultaneous access to more than one device (broadcast and broadcall). The cable between devices consists of 64 twisted-pairs, of which 63 carry active signals, and one carries a ground reference (see tables 20 and 21).

Every pair is electrically terminated at each extremity in order to reduce signal reflections as well as to restore the line to logic 0 when it is not being driven. To ensure the integrity of signal transmission in robust systems (see 6.5), the termination networks or terminators must be powered at all times, and must, therefore, be powered separately from any device. If robust operation is not required, a terminator may physically form part of a device.

The terminators are named according to their position relative to the BG daisy-chain (see clause 4). The terminator at the upstream end of the VICbus (in the sense of the BG propagation) is the **head-end terminator** and that at the downstream end, the **tail-end terminator** (see figure 17).

The VICbus is defined only as far as the cable connectors. Robust systems will make the connection into the device from these connectors by means of a transition module carrying the BG continuity relays, such that the VICbus cables may be removed from the device whilst assuring the integrity of the VICbus including the BG daisy-chain.

7.2 Bus drivers and receivers

The VICbus drivers will have an output stage consisting of an open emitter and open collector having two operational states: active, with the driver outputs "on" and passive, with the driver outputs "off". In the passive state, with the drivers "off", the lines assume a known polarity and voltage set by the pre-biasing termination network, in the active state, with the drivers "on" a differential voltage is impressed on the lines. In order to define wired-OR logic, the active state is considered to be logical TRUE, and the passive state, FALSE.
VICbus receivers will incorporate hysteresis for differential noise immunity.

RULE 7.1
VICbus drivers SHALL have the following characteristics:

PARAMETER	***VALUE***
Minimum output differential voltage	*0,5 V*
Minimum differential output current	*±50 mA*

RULE 7.2
VICbus receivers SHALL have the following characteristics:

PARAMETER	***VALUE***
Minimum input differential voltage	*50 mV*
Minimum input resistance to ground	*4 kΩ*
Minimum input differential resistance	*8 kΩ*
Propagation delay spread, differential inputs to TTL outputs	*±5 nS*

RULE 7.3
The logic levels on the VICbus differential lines SHALL be defined thus:

LOGIC LEVEL	***CONDITION***
0	*Line+ is at a lower potential than Line –*
1	*Line+ is at a higher potential than Line –*

OBSERVATION 7.0
The driver and receiver characteristics given above allow for interoperability between VICbus devices. The precise performance of the chosen transceiver parts and the operating environment, will determine in practice the number of devices and the length of cable which can be used with safety. Annex F gives empirical data on observed results from a number of implementations.

7.3 Cables

VICbus uses twisted pair cables to transmit signals on differential lines. The timing parameters given in this specification are based on certain assumptions of absolute cable propagation time and the spread of this parameter between individual pairs, as well as differences in the propagation and response times of transmitters and receivers (skew). In addition, the maximum allowable length of the cable is specified in order to define certain timing parameters, as well as to ensure signal integrity.

Table 20 lists the VICbus lines.

Table 20 - VICbus lines

Mnemonic	Function	Number of pairs	Type
AD31-AD00	Multiplexed address / data	32	Bussed
ALOCK	Arbitration lock	1	Bussed
AP	Arbiter present	1	Bussed
AS	Address strobe	1	Bussed
BBSY	Bus busy	1	Bussed
BG (BGIN/BGOUT)	Bus grant	1	Daisy-chained
BGLOOP	Bus grant loop	1	End-to-end
BR	Bus request	1	Bussed
CL3-CL0	Control lines	4	Bussed
DEVFAIL	System error	1	Bussed
DS	Data strobe	1	Bussed
GROUND	Ground reference	1	Bussed
ID4-ID0	Identification lines	5	Bussed
INT7-INT0	Interrupt requests	8	Bussed
INTSEL0	Interrupt request select 0	1	Bussed
INTSEL1	Interrupt request select 1	1	Bussed
SERR	Slave error	1	Bussed
VICRESET	Global reset	1	Bussed
WAIT	DTB protocol synchronisation	1	Bussed
	Total	64 pairs	

7.3.1 Cable characteristics

VICbus cables may be manufactured using cable in any form, but which must have the characteristics specified in RULE 7.4, and which must be compatible with the specified connector (see 7.4).

RULE 7.4
The VICbus cables SHALL consist of 28AWG, stranded, twisted pair flat cable with 32 pairs. The electrical characteristics and physical dimensions of the cable SHALL conform to the MIL C-49055/14 specification, with the exclusion of that standard's environmental requirements.

OBSERVATION 7.1
The VICbus cables will have the following typical electrical characteristics measured between the signal conductors within one twisted pair:

Capacitance:	45 pF/m
Characteristic impedance:	120 Ω
Inductance:	0,65 µH/m
Propagation speed:	5,3 ns/m

OBSERVATION 7.2
In this specification, the calculation of cable propagation time T_{CP} assumes the signal propagation speed of twisted-pair cables to be 6 ns/m.

RULE 7.5
The total length of a VICbus SHALL NOT exceed 100 m.

OBSERVATION 7.3
The maximum permitted length of a VICbus is limited to 100 m for reasons of bandwidth limitation and noise immunity, as well as to allow the definition of certain timing parameters.

If a connection of more than 100 m is required, multiple VICbus connections may be made, with an intermediate backplane bus system to provide the necessary connectivity.

OBSERVATION 7.4
The value of the CSR CABLEL bit (bit 03) may be used to optimise the timing of transactions on the VICbus (see 8.3).

RULE 7.6
The stubs connecting the VICbus cables to the drivers and receivers in the devices SHALL NOT exceed 100 mm per line per device. See figure 16.

7.4 Connectors

Four connectors according to DIN41651 are required for one VICbus connection: two connectors for the incoming connection (relative to the BG daisy-chain) and two for the out-going connection, named VC1-i, VC2-i, VC1-o and VC2-o respectively.

RULE 7.7
Connectors according to specification DIN41651 SHALL be used on the VICbus cables. Pin headers SHALL be of type A064MS-C1xx and receptacles of type A064FI-C1xx.

RULE 7.8
The VICbus cable SHALL be connected to the VICbus connectors as shown in table 21.

Table 21 - VICbus connector pin assignments

Connectors VC1-i and VC1-o			
Pin	Line	Pin	Line
1	AP+	2	AP–
3	ALOCK+	4	ALOCK–
5	WAIT+	6	WAIT–
7	SERR+	8	SERR–
9	ID0+	10	ID0–
11	ID1+	12	ID1–
13	ID2+	14	ID2–
15	ID3+	16	ID3–
17	CL3+	18	CL3–
19	CL2+	20	CL2–
21	CL0+	22	CL0–
23	CL1+	24	CL1–
25	INT0+	26	INT0–
27	INT1+	28	INT1–
29	INT2+	30	INT2–
31	INT3+	32	INT3–
33	AD00+	34	AD00–
35	AD01+	36	AD01–
37	AD02+	38	AD02–
39	AD03+	40	AD03–
41	AD04+	42	AD04–
43	AD05+	44	AD05–
45	AD06+	46	AD06–
47	AD07+	48	AD07–
49	AD08+	50	AD08–
51	AD09+	52	AD09–
53	AD10+	54	AD10–
55	AD11+	56	AD11–
57	AD12+	58	AD12–
59	AD13+	60	AD13–
61	AD14+	62	AD14–
63	AD15+	64	AD15–

Connectors VC2-i and VC2-o			
Pin	Line	Pin	Line
1	ID4+	2	ID4–
3	BGLOOP+	4	BGLOOP–
5	AS+	6	AS–
7	DS+	8	DS–
9	GROUND	10	GROUND
11	BBSY+	12	BBSY–
13	INTSEL0+	14	INTSEL0–
15	VICRESET+	16	VICRESET–
17	BR+	18	BR–
19	DEVFAIL+	20	DEVFAIL–
21	BGIN+ (VC2-i) BGOUT+ (VC2-o)	22	BGIN– (VC2-i) BGOUT– (VC2-o)
23	INTSEL1+	24	INTSEL1–
25	INT4+	26	INT4–
27	INT5+	28	INT5–
29	INT6+	30	INT6–
31	INT7+	32	INT7–
33	AD16+	34	AD16–
35	AD17+	36	AD17–
37	AD18+	38	AD18–
39	AD19+	40	AD19–
41	AD20+	42	AD20–
43	AD21+	44	AD21–
45	AD22+	46	AD22–
47	AD23+	48	AD23–
49	AD24+	50	AD24–
51	AD25+	52	AD25–
53	AD26+	54	AD26–
55	AD27+	56	AD27–
57	AD28+	58	AD28–
59	AD29+	60	AD29–
61	AD30+	62	AD30–
63	AD31+	64	AD31–

7.5 Terminators

Common mode reflections and therefore tolerance to common mode noise can be reduced significantly by terminating the differential lines in a symmetrical fashion. By centering the termination network, all induced noise and reflections will appear as common mode signals. Refer also to annex F.

RULE 7.9
Termination networks SHALL be connected to each extremity of every line with the exception of the BG and BGLOOP lines, as shown in figure 16 with:
$R_1 = R_3 = 250\ \Omega$, R_2=125 Ω, (all ±5 %) and Vcc = 5 V (±5 %).

OBSERVATION 7.5
The values of the termination resistors given in RULE 7.9 represent the nearest practical values to those calculated.

7.5.1 Arbitration daisy-chain (BG line) termination

The BG bus grant daisy-chain is connected to every bus requester where it enters a differential line receiver and is called BGIN. After treatment by the requester logic, it is driven on to the next requester downstream of the arbiter by a differential line driver and is called BGOUT. See figure 18.

RULE 7.10
Termination networks SHALL be connected to BGIN and BGOUT as shown in figure 17 with:
$R_1 = R_3 = 250\ \Omega$, R_2=125 Ω (all ±5 %) and Vcc = 5 V (±5 %).

7.5.2 Terminator power

The termination networks are essential to the proper electrical operation of the VICbus. If the requirement is for robust operation (see 6.5) the terminators must be powered from a source separate from that of any device (and in itself uninterruptible). However, if robust operation is not required, then the termination networks can be implemented within a device and share its power supply.

RULE 7.11
Termination networks SHALL be powered at all times when the VICbus is operational.

OBSERVATION 7.6
RULE 7.11 applies to all lines except the BG terminations in offline devices.

RECOMMENDATION 7.1
Power the termination networks from a source separate from that of any device.

7.5.3 Terminators and BGLOOP

Apart from their electrical function, the terminators have the logical function of providing the connection of the BGIN/BGOUT daisy chain to the BGLOOP line, so allowing the arbiter to be placed at any physical location on the VICbus with respect to the ends of the cable (that is the terminators). See clause 4 and figure 17 for more details.

RULE 7.12
The head-end terminator SHALL connect BGLOOP to BGIN and the tail-end terminator BGOUT to BGLOOP.

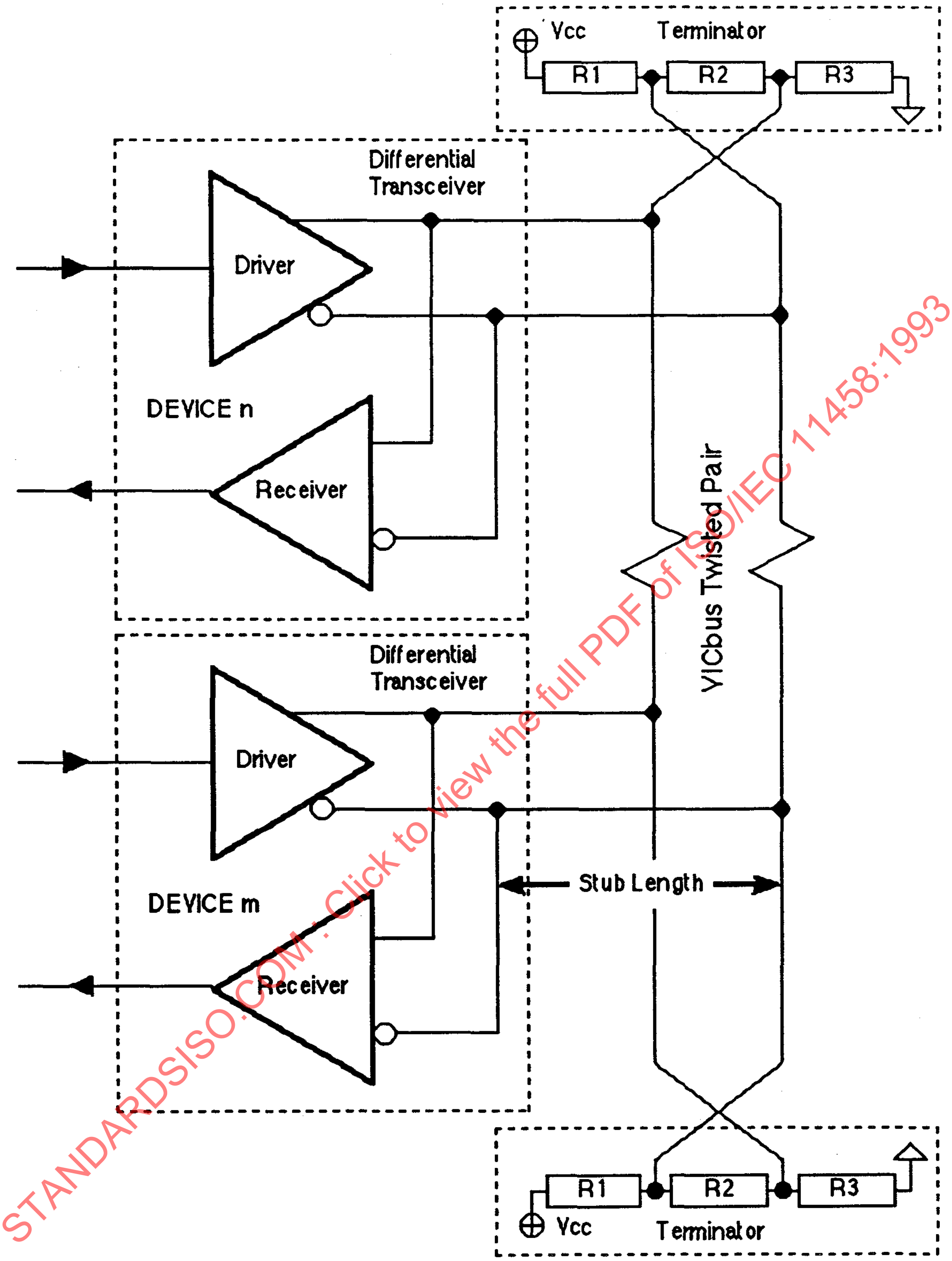

Figure 16 - Electrical transmission

7.6 Cable continuity for offline devices

It is often essential to be able to remove a device from service, either by powering it down or by physically removing it from the VICbus cable. In order to achieve this in robust systems, the electrical continuity of the VICbus lines must be maintained. For most lines continuity is assured by direct connections from the input to the output connectors, possibly implemented on a transition module to which the cables remain permanently connected. The arbitration daisy-chain (BGIN/BGOUT), however, poses a particular problem.

When a device goes offline, the connection of BGIN to BGOUT and removal of the BG terminators can be achieved by the use of the contacts of electro-mechanical relays. In such a case, these relays would again be mounted on a transition module as illustrated in figures 17 and 18.

In systems where robust operation is not required, the VICbus cables may be connected directly to the device, without the intermediate transition module. However, placing a device offline, and / or removing the VICbus cables will cause the system to fail.

RECOMMENDATION 7.2
Provide a mechanism which automatically assures the continuity of all VICbus lines, including the BG line (BGIN/BGOUT) in the case that a device is powered down.

RECOMMENDATION 7.3
In robust systems, provide a transition module to carry the BG continuity relays.

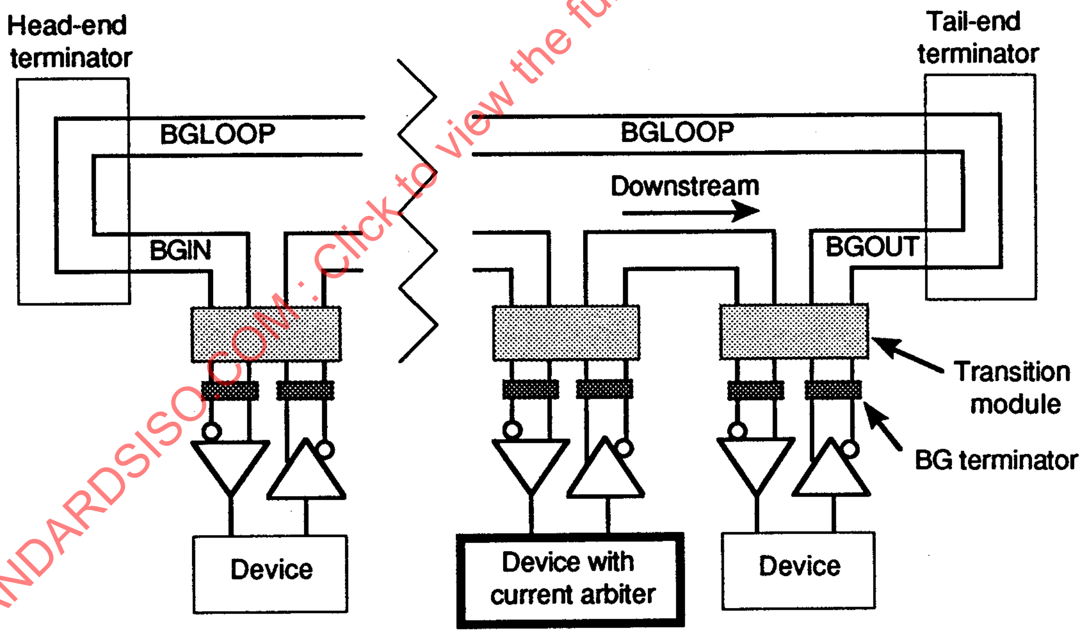

Figure 17 - Bus grant daisy-chain and BGLOOP

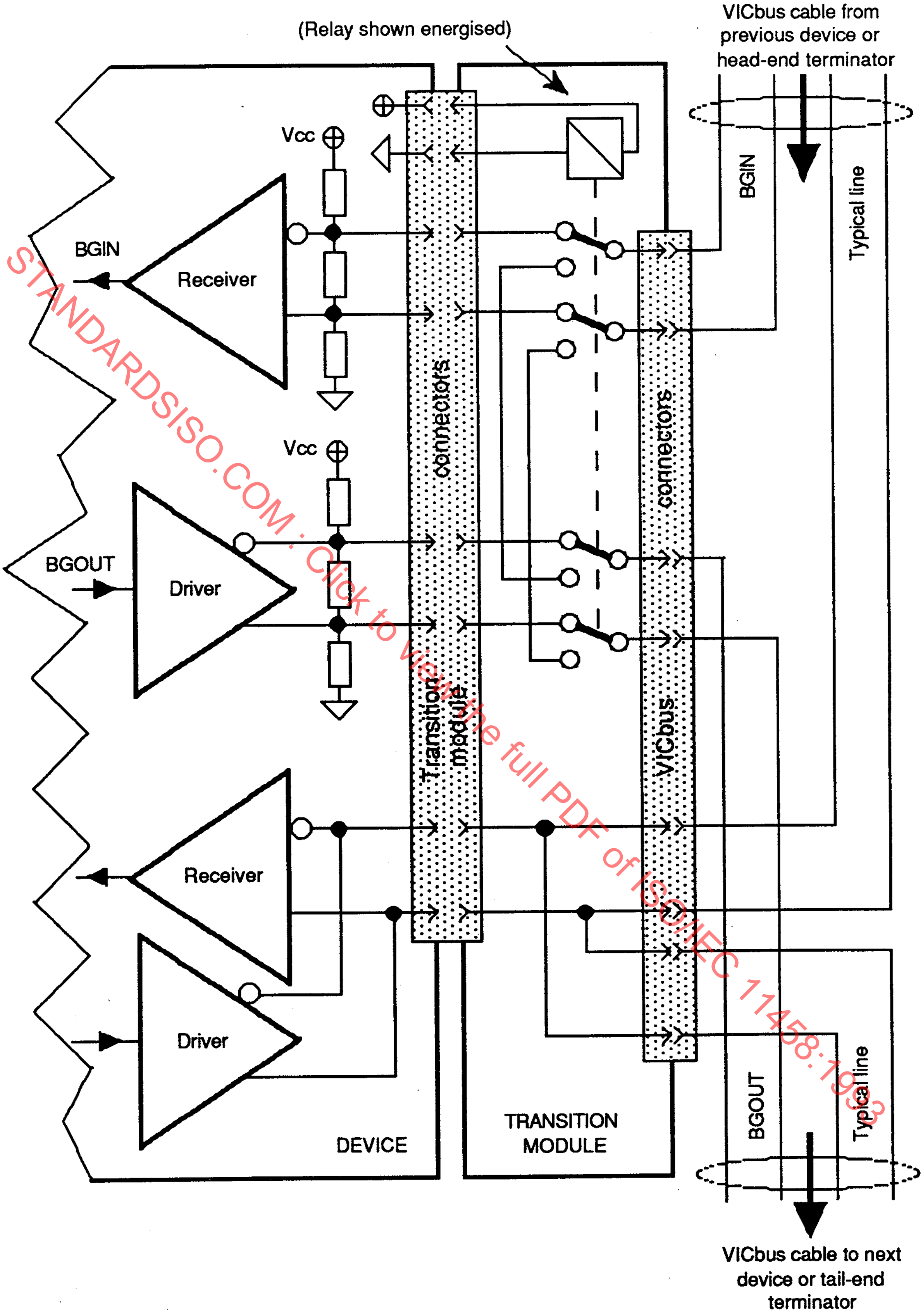

Figure 18 - Bus Grant termination and continuity of lines in robust systems

8 VICbus registers

8.1 Introduction

DTB direct cycles access resources internal to the device (that is not on any associated backplane bus). These resources are collectively known as "registers" and may take the form of one of the optional or mandatory status and control registers described in this clause, or, for example, a block of memory. Individual registers are addressed by means of the register select (RS) signals (one per RS value); in addition, the address signals (A31-A02) may be used to provide secondary address information in order to select locations within memory blocks.

RULE 8.1
Registers summarised in table 22 are marked "mandatory", "optional", or "reserved".
"Mandatory" registers and their bit functions and assignments described in 8.3 et seq., SHALL be implemented in all VICbus devices.
"Optional" registers need not be implemented, but if they are, their bit functions and assignments SHALL be as described in 8.3 et seq.
"Reserved" registers SHALL be reserved for future use, and SHALL reply with value zero if read.

PERMISSION 8.1
The use of registers summarised in table 22 with the mention "user definable" MAY be defined by the user.

8.2 Register summary

Table 22 - Register summary

RS value	Read / Write	Name	Description	Status
0	Read / write	CSR	Control and Status Register	Mandatory
1	Read	OLR	Online Register	Optional
2	Read	DOR	Device Operational Register	Optional
3	-	-	RESERVED	Reserved
4	Write	RR	Reset Register	Optional
5	Read	TR	Transparent Register	Optional
6	-	-	RESERVED	Reserved
7	-	-	RESERVED	Reserved
8-11	Read	DIR	Device Identification Registers	Mandatory
12-15	-	-	RESERVED	Reserved
16-31	-	-	USER DEFINABLE	User definable

8.3 Control and status register - CSR

ADDRESS: RS = 0

TYPE: Read / write - mandatory (refer to RULE 8.1)

FUNCTION: VICbus control and status functions as detailed below:

Table 23 - Command and status register: bit assignments

Regulation		Bit	Read / Write	Mnemonic	Name
RULE 8.2	*When one is written into this bit, the device SHALL be reset. See also RULE 6.11*	00	Write only	SRESET	Selective reset
OBSERVATION 8.1	This is a self-timed, one-shot action and requires no further VICbus cycle to terminate its action				
RULE 8.3	*The device SHALL return its online / offline status:* *<u>Bit 01</u> <u>Status</u>* *1 Online (-enabled or -disabled)* *0 Offline*	01	Read	ONLINE	Online
OBSERVATION 8.2	Bit 01 could be hardwired to 1 for read access, since if the device responds at all to a DTB cycle it is, by definition, online		Read		
RULE 8.4	*When zero is written to this bit, the device SHALL enter the offline state. When one is written, no action SHALL occur*		Write		
RULE 8.5	*The device SHALL return its transparent (-enabled, -disabled) status:* *<u>Bit 02</u> <u>Status</u>* *1 Online-enabled* *0 Online-disabled*	02	Read	TRANSP	Transparent
RULE 8.6	*When zero is written to this bit, the device SHALL remain in, or SHALL switch to the online-disabled state. When one is written to this bit, the device SHALL switch to the online-enabled state, unless forbidden by local action (for example switch or local command bit)*		Write		
RULE 8.7	*The state of this bit SHALL indicate the total length of the VICbus cables in m (l):* *<u>Bit 03</u> <u>Cable Length</u>* *1* $30 < l \le 100$ *0* $0 < l \le 30$	03	Read	CABLEL	Cable length
RULE 8.8	*The master initialising the VICbus SHALL write the appropriate value into this bit at system initialisation time*		Write		
RULE 8.10	*These bits SHALL be reserved for future use, and SHALL be read as zero*	07-04	-	RSVD	Reserved
PERMISSION 8.2	The use of these bits MAY be defined by the user	31-08	-	-	User definable

8.4 *Online register - OLR*

ADDRESS: RS = 1

TYPE: Read only - optional (refer to RULE 8.1)

FUNCTION: When this register is read, every participating device addressed reports its online status in the bit position corresponding to its device number.

Examples:

OLR bit 2 = 1 Device 2 is online (enabled or disabled).

OLR bit 5 = 0 Device 5 is offline.

OBSERVATION 8.3

a) Device 0 does not exist;

b) a broadcall cycle will retrieve the online status of all devices on the VICbus which implement this register;

c) in a broadcall status retrieval, devices not present on the VICbus will be signalled as offline, since the appropriate data line will be held at zero by the action of the terminators;

d) the assertion of VICRESET will set the device to the online-disabled state. See 6.4.

Table 24 - Online register: bit assignments

Regulation (refer to rule 8.1)		Bit	Read / Write	Name
RULE 8.11	*This bit SHALL be reserved for future use, and SHALL be read as zero*	00	Read only	-
RULE 8.12	*The device SHALL return its online / offline status in the bit number corresponding to its own Device number:* *Bit dn* *Status* *1* *Online (-enabled or -disabled)* *0* *Offline*	31-01	Read only	OL31-OL01

8.5 *Device operational register - DOR*

ADDRESS: RS = 2

TYPE: Read only - optional (see RULE 8.1)

FUNCTION: When this register is read, every participating device reports its internal condition (failed or operational) in the bit position corresponding to its device number.

Examples:

DOR bit 2 = 0 Device 2 has failed.

DOR bit 13 = 1 Device 13 is operational.

PERMISSION 8.3
A device MAY deassert the bit corresponding to its device number in its DOR for reasons other than its assertion of the DEVFAIL line.

OBSERVATION 8.4

a) Device 0 does not exist;
b) a broadcall cycle will retrieve the operational status of all devices on the VICbus which implement this register;
c) in such a broadcall status retrieval, non-existent devices will be signalled as failed, since the appropriate data line will be held at zero by the action of the VICbus terminators;
d) the assertion of VICRESET may place all devices in the failed state, until placed operational by some internal action. See 6.1.1.

Table 25 - Device operational register: bit assignments

Regulation (refer to rule 8.1)		Bit	Read / Write	Name
RULE 8.13	*This bit SHALL be reserved for future use, and SHALL be read as zero*	00	Read only	-
RULE 8.14	*The device SHALL return its operational status in the bit number corresponding to its own device number:* *Bit dn* *Status* *1* *Operational* *0* *Failed*	31-01	Read only	DO31-DO01

8.6 Reset register - RR

ADDRESS: RS = 4

TYPE: Write only - optional (refer to RULE 8.1)

FUNCTION: If implemented, this register can be used to selectively reset one or more devices simultaneously.

Writing one into the bit in this register corresponding to the device number of a device will cause the device to be reset as described in 6.3.

This is a self-timed, one-shot action and requires no further VICbus cycle to terminate its action.

Example:

RR bits 2, 5, 15 = 1 Devices 2, 5 and 15 are reset.

OBSERVATION 8.5

a) Device 0 does not exist;
b) a broadcast cycle may be used to perform a selective reset on any number of VICbus devices;
c) this register has an identical function to CSR bit 00;
d) see also RULE 6.11.

Table 26 - Reset register: bit assignments

Regulation (refer to rule 8.1)		Bit	Read / Write	Name
RULE 8.15	*This bit SHALL be reserved for future use*	00	Write only	-
RULE 8.16	*If the bit number corresponding to the device's own device number is written with the value one, the device SHALL be reset*	31-01	Write only	RR31-RR01

8.7 Transparent register - TR

ADDRESS: RS = 5

TYPE: Read only - optional (refer to RULE 8.1)

FUNCTION: When this register is read, every participating device addressed reports its transparent status in the bit position corresponding to its device number.

Examples:

TR bit 2 = 1	Device 2 is transparent.
TR bit 5 = 0	Device 5 is not transparent (it may be online-disabled or offline).

The assertion of VICRESET will set the device to the online-disabled state. See 6.4.

OBSERVATION 8.6

a) Device 0 does not exist;

b) a broadcall cycle will retrieve the online status of all devices on the VICbus which implement this register;

c) stand-alone devices (that is those not associated to a backplane bus) will report "online-disabled" in their TR.

Table 27 - Transparent register: bit assignments

Regulation (refer to rule 8.1)		Bit	Read / Write	Name
RULE 8.17	*This bit SHALL be reserved for future use, and SHALL be read as zero*	00	Read only	-
RULE 8.18	*The device SHALL return its transparent (-enabled, -disabled) status in the bit number corresponding to its own device number:* *Bit dn* — *Status* *1* — *Online-enabled* *0* — *Online-disabled or offline*	31-01	Read only	TR31-TR01

8.8 Device identification registers - DIR

ADDRESS: RS = 8, 9, 10 and 11

TYPE: Read only - mandatory (refer to RULE 8.1)

FUNCTION: Provide a unique identification of the device and its capability.

Table 28 - Device identification registers - byte assignments

Regulation	(refer to rule 8.1)	RS value	Register	Bits	Read / Write	Function
PERMISSION 8.4	The use of DIR0 bits 31-16 MAY be defined by the user	8	DIR0	31-16	Read only	User definable
RULE 8.19	*DIR0 bits 15-08 SHALL be reserved for future use, and SHALL be read as zero*			15-08	Read only	Reserved
RULE 8.20	*DIR0 bits 07-00 SHALL uniquely identify the manufacturer of the device. The codes for this identification SHALL be issued by VITA*[1)]			07-00	Read only	Manufacturer code
PERMISSION 8.5	The use of DIR1 bits 31-16 MAY be defined by the user	9	DIR1	31-16	Read only	User definable
RULE 8.21	*DIR1 bits 15-08 SHALL be reserved for future use, and SHALL be read as zero*			15-08	Read only	Reserved
RECOMMENDATION 8.1	Use DIR1 bits 07-00 to carry a code designated by the manufacturer to uniquely identify the device within his range of VICbus products			07-00	Read only	Device type
PERMISSION 8.6	The use of DIR2 bits 31-16 MAY be defined by the user	10	DIR2	31-16	Read only	User definable
RULE 8.22	*DIR2 bits 15-08 SHALL be reserved for future use, and SHALL be read as zero*			15-08	Read only	Reserved
RULE 8.23	*DIR2 bits 07-00 SHALL designate the capability of a VICbus slave as given in table 29*			07-00	Read only	Capability
PERMISSION 8.7	The use of DIR3 bits 31-16 MAY be defined by the user	11	DIR3	31-16	Read only	User definable
RULE 8.24	*DIR3 bits 15-08 SHALL be reserved for future use, and SHALL be read as zero*			15-08	Read only	Reserved
RULE 8.25	*DIR3 bits 07-00 SHALL specify the speed capability of a slave implementing the NC1 or NC2 protocols according to table 30*			07-00	Read only	NC slave speed

1) VITA, the VFEA International Trade Association of Scottsdale, Arizona, USA, will issue and maintain a register of VICbus manufacturer identification codes.

Table 29 - Device identification register 2 - bit assignments

Relevant regulation	Bit	Read / Write	Name	Capability (a bit in the asserted state indicates that the slave has the given capability)
RULE 8.23	00	Read only	BCC	Slave: broadcast and broadcall transfers
	01		NC1	Slave: non-compelled 1 transfers
	02		NC2	Slave: non-compelled 2 transfers
	03		A64	VMEbus master: A64 transfers
	04		D64	VMEbus master: D64 transfers
	05		BLT	Slave: block transfers
	06		VME	The device is a VMEbus interface
	07		RSVD	Reserved

Table 30 - Device identification register 3 - bit assignments

Relevant regulation	Bit	Read / Write	Name	Function
RULE 8.25	06-00	Read only	NCSPEED	This field SHALL give the minimum time (in tens of ns) between successive assertions of DS which the device is capable of accepting as a slave in a non-compelled transfer. Conditioned by the state of Bit 07
	07		FAST / SLOW	When at zero, the device SHALL be capable of performing non-compelled transactions at the rate given by bits 06-00. When at one, the device SHALL be capable of performing non-compelled transactions at the fastest rate permitted by this specification