

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

	Page
FOREWORD.....	7
Clause	
1 Scope.....	8
2 Introduction to the ISO/IEC 11458 VICbus standard	9
2.1 Objectives	9
2.2 Standard terminology	9
2.2.1 Rule <N.n>	9
2.2.2 Recommendation <N.n>	9
2.2.3 Permission <N.n>	10
2.2.4 Observation <N.n>	10
2.3 Other terminology	10
2.4 Timing diagrams	10
2.5 Tables	10
2.6 Data representation	10
3 Data transfer bus	11
3.1 Introduction	11
3.2 DTB cycle types	11
3.2.1 Direct cycles	12
3.2.2 Transparent cycles	12
3.3 Use of the DTB information lines	13
3.3.1 The address phase	13
3.3.2 The data phase	13
3.3.3 Address / data lines AD31-AD00	13
3.3.4 Control lines CL3-CL0	13
3.3.5 Identification lines ID4-ID0	14
3.3.6 Device number signals DN4-DN0	14
3.3.7 Address signals A31-A02	14
3.3.8 Address extension signals AE5-AE0	14
3.3.9 Register select signals RS4-RS0	14
3.3.10 Block transfer signal BLT	16
3.3.11 Write signal WRITE	16
3.3.12 Byte selection signals LWORD, A01, ASEL0, ASEL1, DSEL0, DSEL1	16
3.3.13 Interrupter number signals IN4-IN0	18
3.3.14 Slave response signal SERR	18
3.3.15 Data signals D31-D00	18
3.4 Transparent VME-A64 cycle	18
3.5 Data transfer cycle - bus protocols and timing	18
3.5.1 Block transfer cycles	19
3.5.2 Read-modify-write cycles	19
3.6 Compelled protocol	21
3.6.1 The address phase	21
3.6.2 The data phase	21
3.7 Non-compelled protocols	23
3.7.1 Non-compelled 1 (NC1)	25
3.7.2 Non-compelled 2 (NC2)	27
3.8 Slave participation in DTB cycles	28

© ISO/IEC 1993

All rights reserved. No part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from the publisher.

ISO/IEC Copyright Office • Case Postale 131 • CH-1211 Genève 20 • Switzerland

Printed in Switzerland

Full document details: ISO IEC 11458-1993



Documents

Document Sharing Hub

3.9	DTB timing rules	29
4	Arbitration	41
4.1	Introduction	41
4.2	Lines	41
4.3	Arbitration protocol	41
4.4	Arbiter	42
4.5	Requester	43
4.6	Transfer of DTB mastership	43
4.7	Loss of the arbiter	44
4.8	Arbitration timing rules	48
5	Interrupts	50
5.1	Introduction	50
5.2	Lines and signals	50
5.3	Interrupt request signal selection	50
5.4	Interrupt protocol	51
5.5	Interrupter	52
5.6	Interrupt handler	53
5.7	Timing regulations	55
6	Utilities	57
6.1	Introduction	57
6.1.1	Arbitration lock line ALOCK	57
6.1.2	Device failure line DEVFAIL	57
6.1.3	Interrupt request select lines INTSEL0 and INTSEL1	57
6.1.4	VICbus reset line VICRESET	58
6.2	INTSEL generator selection	58
6.3	Reset	61
6.3.1	Global reset - VICRESET	61
6.3.2	Selective reset	61
6.4	Online and offline states	62
6.4.1	Regulations	63
6.4.2	Power-up condition	64
6.5	Fault tolerance	64
6.6	Cable connection and disconnection in robust systems	65
7	Electrical specifications	66
7.1	Introduction	66
7.2	Bus drivers and receivers	66
7.3	Cables	67
7.3.1	Cable characteristics	68
7.4	Connectors	68
7.5	Terminators	70
7.5.1	Arbitration daisy-chain (BG line) termination	70
7.5.2	Terminator power	70
7.5.3	Terminators and BGLOOP	70
7.6	Cable continuity for offline devices	72
8	VICbus registers	74
8.1	Introduction	74
8.2	Register summary	74
8.3	Control and status register - CSR	75
8.4	Online register - OLR	76
8.5	Device operational register - DOR	77
8.6	Reset register - RR	78

8.7	Transparent register - TR.....	79
8.8	Device identification registers - DIR.....	80

Annexes

A	Interfacing between VMEbus and VICbus	82
A.1	Introduction	82
A.2	Data transfer bus	83
A.2.1	Address and data	83
A.2.2	VMEbus AM codes	83
A.2.3	VICbus slave response	83
A.2.4	VMEbus RETRY*	84
A.2.5	VMEbus D64 transfers	84
A.2.6	VMEbus address only cycles	84
A.2.7	Block transfers	84
A.3	Interrupts	86
A.4	Utilities	87
A.4.1	System failure	87
A.4.2	System reset	88
A.5	VMEbus interface functions	89
B	Glossary	90
C	Summary of lines and signals	93
D	Arbitration dead lock	95
E	Wired-OR glitch	96
F	VICbus electrical characteristics	97
F.1	Electrical termination	97
F.2	Practical VICbus implementations	98

	Page
Tables	
1	VICbus data representation 10
2	Direct cycles 12
3	Transparent cycles 12
4	Use of the address / data, control and identification lines 15
5	VMEbus byte alignment 17
6	Byte lane alignment 17
7	Transparent VME-A64 signal assignment 18
8	Summary of slave participation in DTB cycles 28
9	Master - timing regulations 36
10	Slave - timing regulations 39
11	Arbiter - timing regulations 48
12	Requester - timing regulations 49
13	Interrupt request multiplexing 50
14	IACK byte alignment 52
15	Summary of interrupt protocol actions 54
16	Interrupts - timing regulations 56
17	INTSEL generator selection - timing regulations 60
18	Summary of the online / offline state of a device following various actions 62
19	Summary of actions permitted in the three online / offline states 62
20	VICbus lines 67
21	VICbus connector pin assignments 69
22	Register summary 74
23	Command and status register 75
24	Online register 76
25	Device operational register 77
26	Reset register 78
27	Transparent register 79
28	Device identification registers - byte assignments 80
29	Device identification register 2 - bit assignments 81
30	Device identification register 3 - bit assignments 81
A.1	VMEbus interface control and status functions 89

Figures

1	DTB cycle	13
2	Compelled protocol	20
3	Non-compelled 1 protocol	24
4	Non-compelled 2 protocol	26
5	Compelled cycle	30
6	Compelled cycle last data transfer and end of cycle	31
7	NC1 address phase and first data transfer	32
8	NC1 last data transfer and end of cycle	33
9	NC2 address phase and first data transfer	34
10	NC2 last data transfer and end of cycle	35
11	Arbitration-1	45
12	Arbitration-2	46
13	Arbitration-3	47
14	Interrupt request selection timing	55
15	INTSEL generator selection	59
16	Electrical transmission	71
17	Bus grant daisy-chain and BGLOOP	72
18	Bus Grant termination and continuity of lines in robust systems	73
A.1	Inter-crate block transfers	85
A.2	DEVFAIL / SYSFAIL* interconnection for a VMEbus to VICbus interface	87
A.3	Reset circuit for a VMEbus to VICbus interface	88
D.1	Arbitration dead-lock resolution	95
E.1	Wired-OR glitch	96
F.1	VICbus termination	97
F.2	Device / cable length derating	98

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 broadcast (master read) data transfers. One of the non-compelled protocols allows broadcast transfers, whereas neither permit broadcast 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 (**broadcast 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:

- select the slave device or devices which it requires to participate in the cycle;
- specify the cycle type;
- 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).

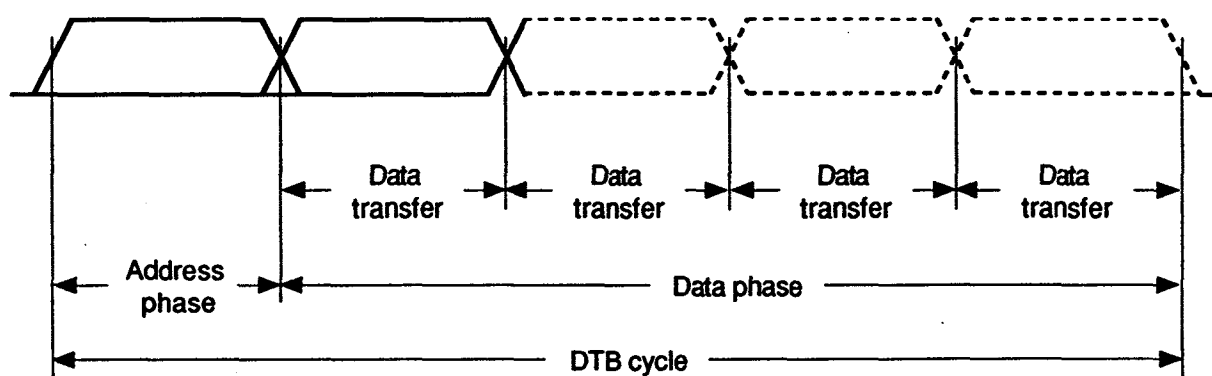


Figure 1 - DTB cycle

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 broadcast 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 broadcast 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

Lines ⇒	Address phase		Data phase					
	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

Lines ⇒	Address phase		Data phase					
	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 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(E0), 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.

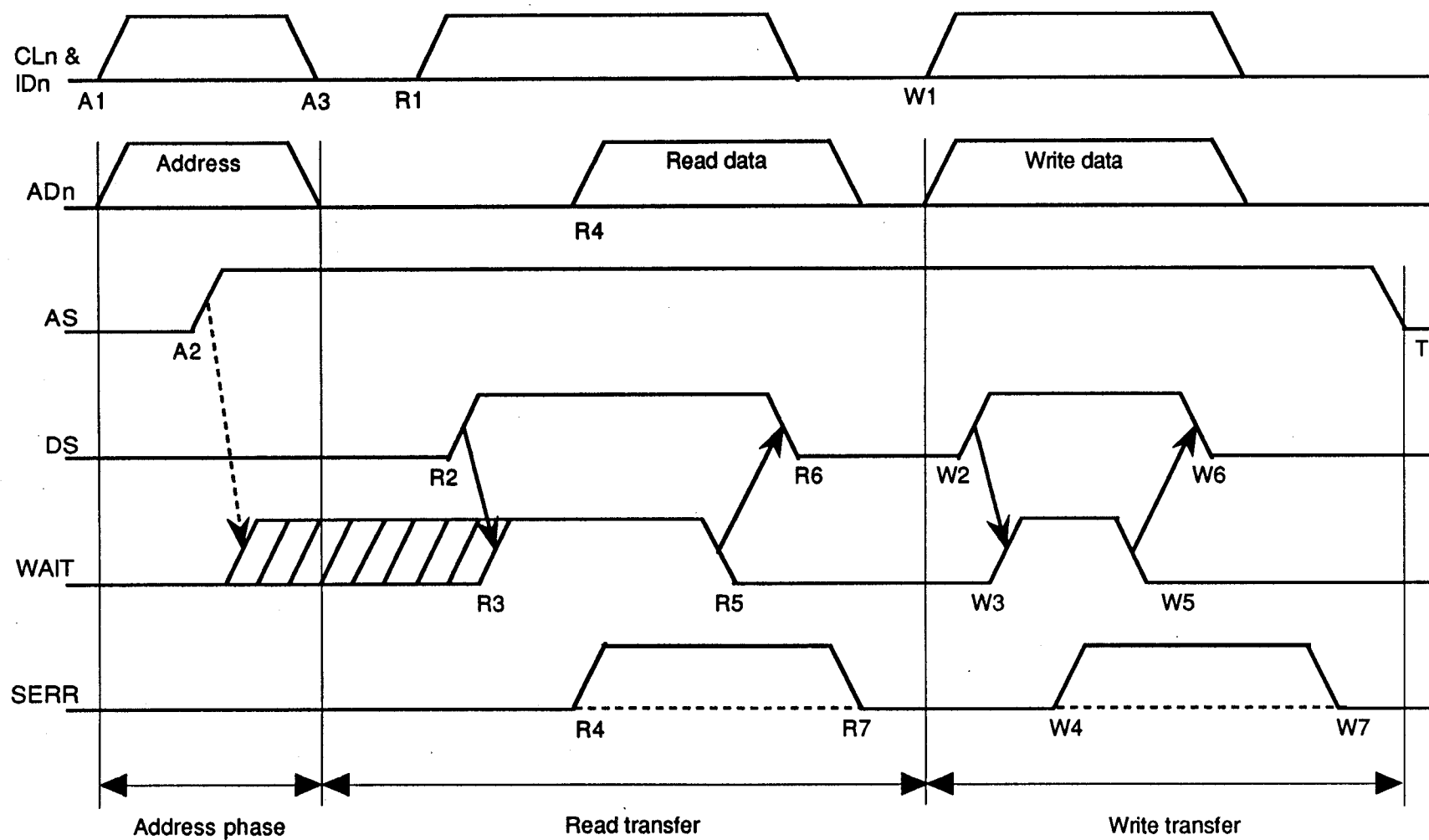


Figure 2 - Compelled protocol

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 broadcast 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 broadcast 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.

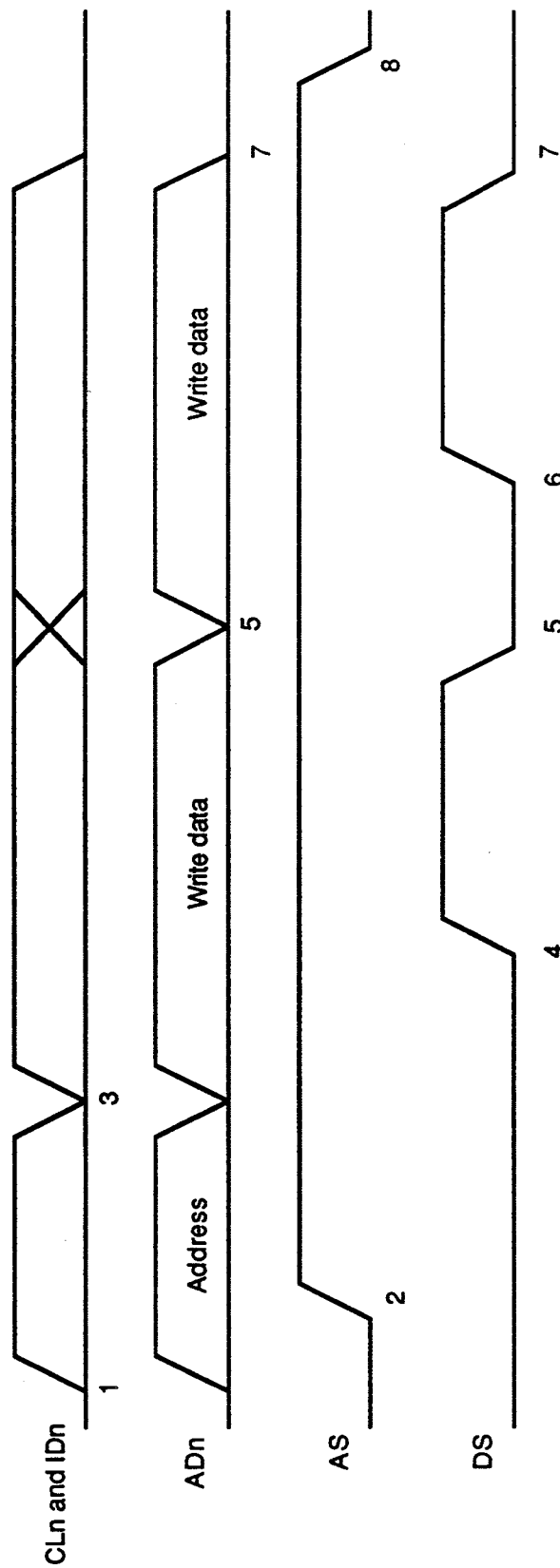


Figure 3 - Non-compelled 1 protocol

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.

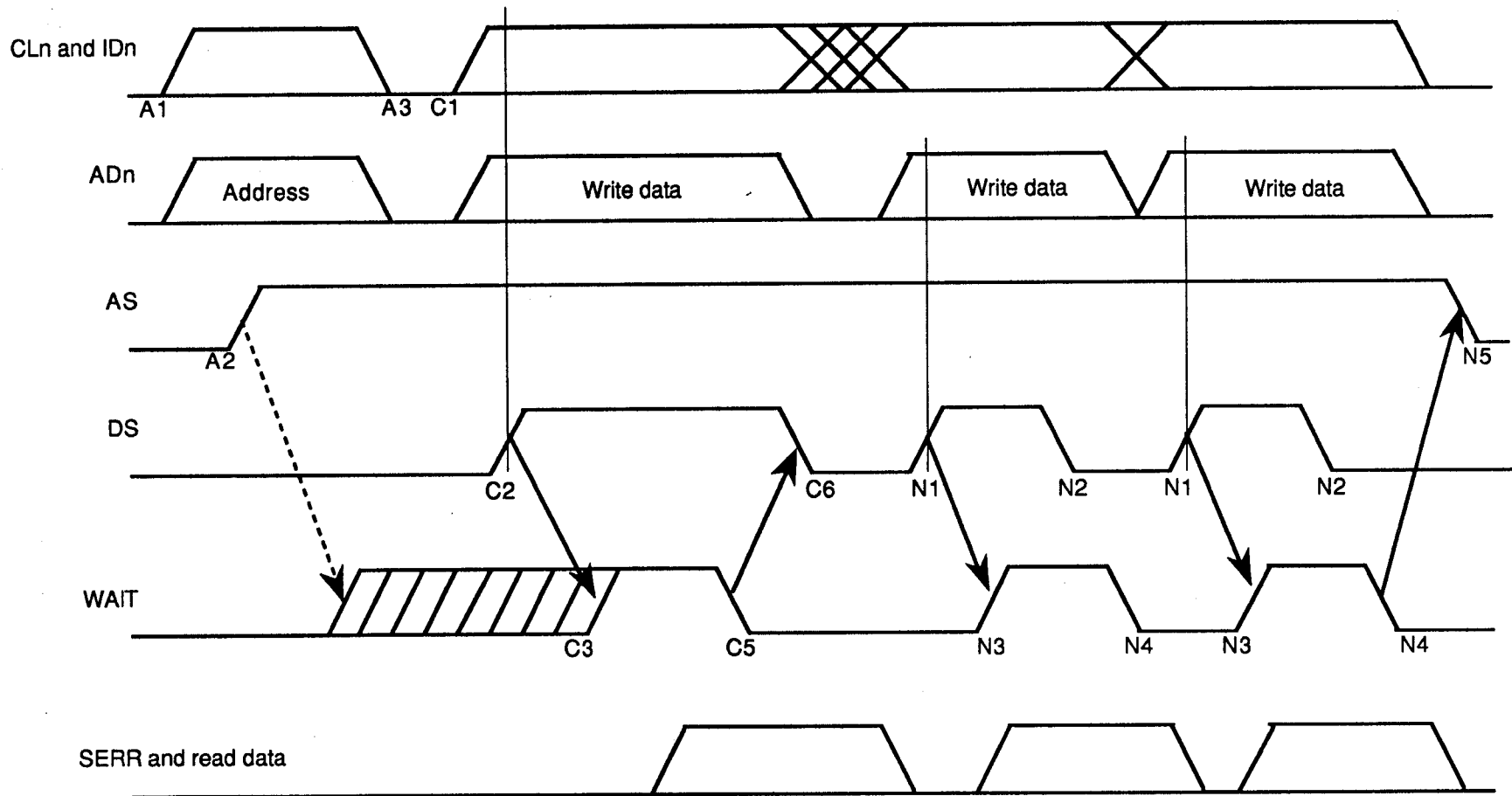


Figure 4 - Non-compelled 2 protocol

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 broadcast 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 (<i>dn</i> indicates the device's own number)	Participates (if the cycle type is implemented)	Relevant regulations (in addition to RULE 3.8)
Any	Offline	<i>dn</i> or 0	No	RULE 6.14
Transparent	Online-disabled	<i>dn</i> or 0	No	RULE 6.12
	Online-enabled	<i>dn</i>	Yes (mandatory)	-
		0	Yes (optional)	PERMISSION 3.3
Direct	Online (either)	<i>dn</i>	Yes (mandatory)	RULE 3.1
Compelled		0	Yes (optional)	PERMISSION 3.3
Direct NC1	Online (either)	<i>dn</i>	Yes (mandatory)	-
		0	Yes (optional)	PERMISSION 3.3
Direct NC2	Online (either)	<i>dn</i>	Yes (mandatory)	-
		0	No	1)

1) A master should never attempt to broadcast or broadcast 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 broadcast 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 broadcast cycles there is no indication as to which slaves have responded, except in the data returned for certain broadcast 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.

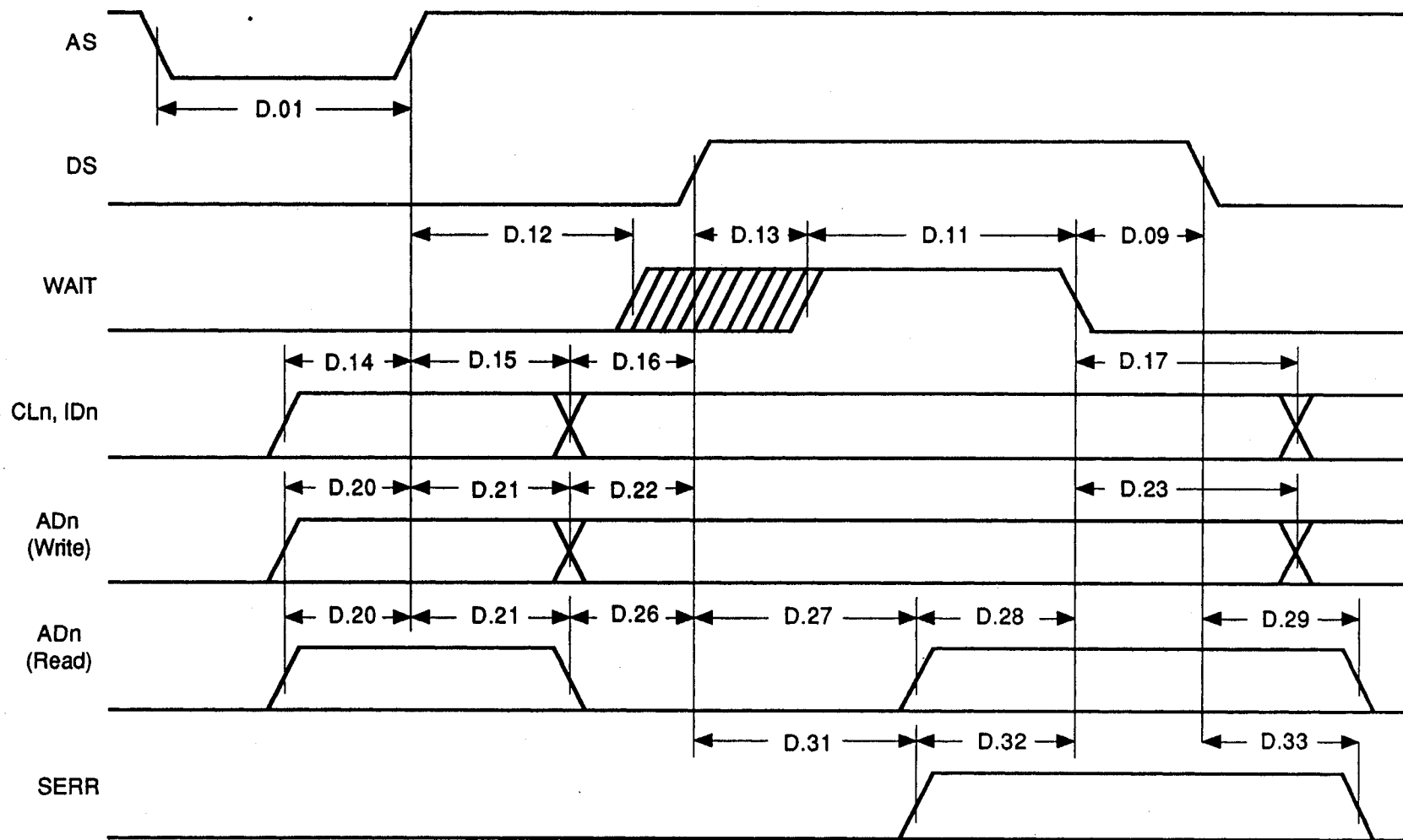


Figure 5 - Compelled cycle: address phase and first data transfer

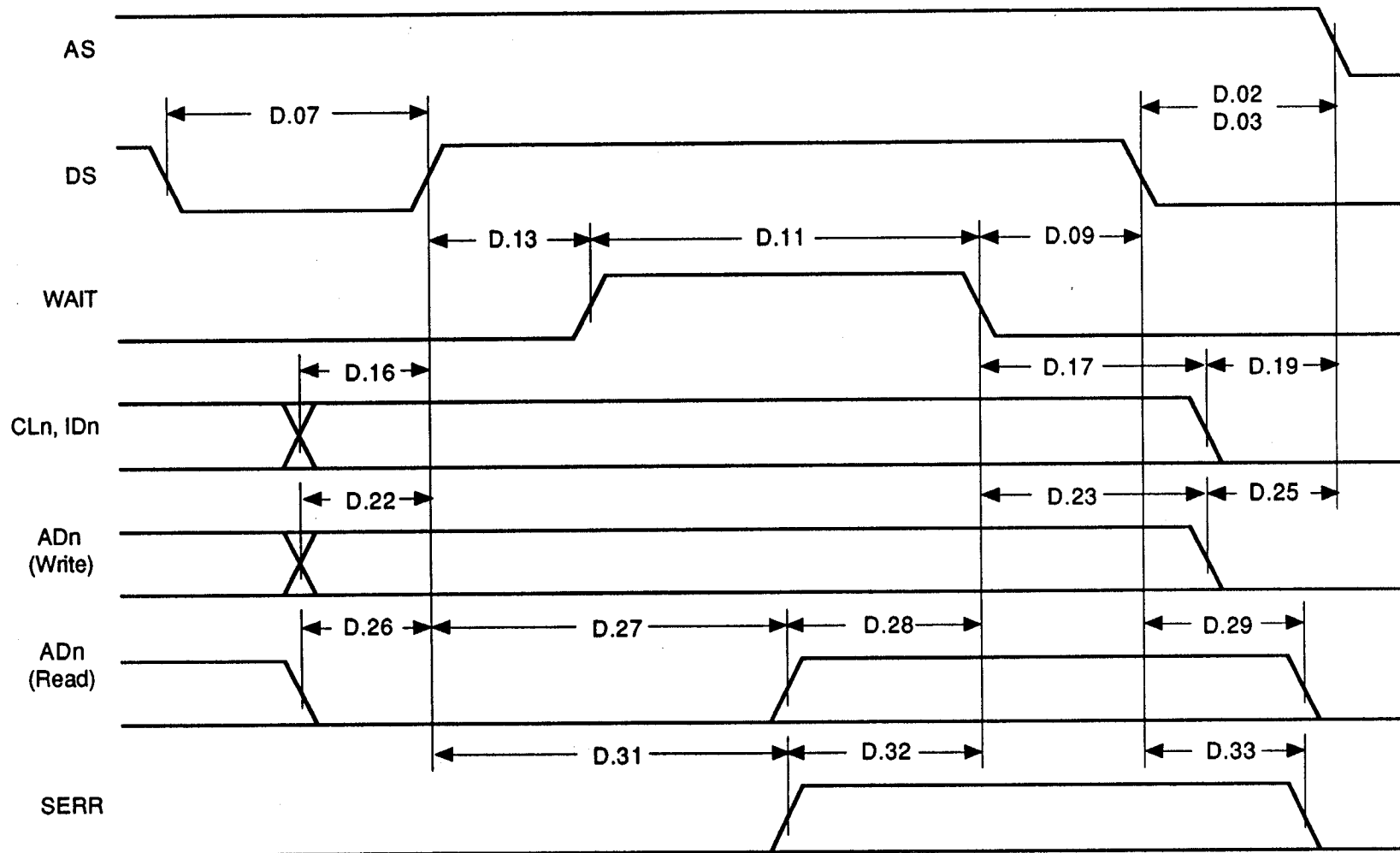


Figure 6 - Compelled cycle - last data transfer and end of cycle