

A4000/A3000 HARDWARE DEVELOPER NOTES

PART I: THE ZORRO III BUS SPECIFICATION

PART II: THE AMIGA LOCAL BUS SLOT

PART III: THE AMIGA VIDEO SLOT

Document Revision 1.11

December 1992

by Dave Haynie

with Scott Schaeffer, Scott Hood and Dan Baker

Copyright © 1990, 1991, 1992 Commodore-Amiga, Inc.

IMPORTANT INFORMATION

"A life spent making mistakes is not only more honorable but more useful than a life spent doing nothing."

-George Bernard Shaw

This Document Contains Preliminary Information

The information contained here is preliminary in nature and subject to possible errors and omissions. Few Zorro III cards have yet been designed, so some features described here have not actually been tested in a system, or in some cases, actually implemented as of this writing. That, of course, is one major reason for having a specification in the first place.

Commodore Technology reserves the right to correct any mistake, error, omission (or vicious lie). Corrections will be published as updates to this document, which will be released as necessary. Revisions will be tracked via the revision number that appears on the front cover. New revisions will always list the corrections up front, and developers will be kept up to date on released revisions via the normal CATS channels.

All information herein is Copyright © 1990, 1991, 1992 by Commodore-Amiga, Inc., and may not be reproduced in any form without permission.

ACKNOWLEDGEMENTS

"Art is I; science is we."

-Claude Bernard

I'd like to acknowledge the following people and groups, without whom this new stuff would have been impossible:

- The original Amiga designers, for designing the first microcomputer bus with support for multiple masters, software board configuration, and room to grow.
- The rest of the A3000 Engineers: Greg Berlin, Hedley Davis, Scott Hood, and Scott Schaeffer; PCB master Terry Fisher; and the lab maniacs George Terbush, Brian Fenimore, and Dan Faust. And of course the A3000 boss men, Jeff Porter and Henri Ruben, who let it all happen.
- The folks who helped review the original version of this document, overnight: Joe Augenbraun, Dan Baker, Hedley Davis, Bryce Nesbitt, and Jeff Porter. And to the numerous folks who've helped out with questions, corrections, and other feedback ever since.
- The Commodore-Amiga software group, and the Commodore Semiconductor Group, for excellent support in their respective areas.
- Commodore's Developer Support people from both sides of the Atlantic.
- Gold Disk, for some good and relatively bug free electronic publishing software.
- Iggy; an excellent cat, an excellent foot warmer.

TABLE OF CONTENTS

PART I: THE ZORRO III BUS SPECIFICATION

CHAPTER 1 INTRODUCTION

1.1	Intended Audience.....	1
1.2	Bug Reports.....	2
1.3	Amiga Bus History.....	2
1.4	The Zorro III Rationale.....	3
1.5	Document Revision History.....	4
1.5.1	Changes for Rev 0.90.....	5
1.5.2	Changes for Rev 0.91.....	5
1.5.3	Changes for Rev 1.00.....	5
1.5.4	Changes for Rev 1.01.....	5
1.5.5	Changes for Rev 1.10.....	5
1.5.6	Changes for Rev 1.11.....	6

CHAPTER 2 ZORRO II COMPATIBILITY

2.1	Changes From The A2000 Bus.....	7
2.1.1	6800 Bus Interface.....	8
2.1.2	Bus Memory Mapping and Cache Support.....	8
2.1.3	Bus Synchronization Delays.....	9
2.1.4	Zorro II Master Access to Local Slaves.....	9
2.1.5	Bus Arbitration and Fairness.....	9
2.1.6	Intelligent Cycle Spacing.....	9
2.1.7	Bus Drive and Termination.....	10
2.1.8	DMA Latency and Overlap.....	10
2.1.9	Power Supply Differences.....	10
2.2	Bus Architecture.....	11
2.3	Signal Description.....	11
2.3.1	Power Connections.....	12

2.3.2	Clock Signals.....	12
2.3.3	System Control Signals.....	13
2.3.4	Slot Control Signals.....	15
2.3.5	DMA Control Signals.....	15
2.3.6	Addressing and Control Signals.....	17

CHAPTER 3 BUS ARCHITECTURE

3.1	Basic Zorro III Bus Cycles.....	19
3.1.1	Design Goals.....	20
3.1.2	Simple Bus Cycle Operation.....	20
3.2	Advanced Mode Support Logic.....	22
3.2.1	Bus Locking.....	22
3.2.2	Cache Support.....	22
3.3	Multiple Transfer Cycles.....	23
3.4	Quick Bus Arbitration.....	25
3.5	Quick Interrupts.....	27
3.6	Compatibility with Zorro II Devices.....	28
3.7	Zorro III Implementations.....	29

CHAPTER 4 SIGNAL DESCRIPTION

4.1	Power Connections.....	31
4.2	Clock Signals.....	32
4.3	System Control Signals.....	32
4.4	Slot Control Signals.....	34
4.5	DMA Control Signals.....	35
4.6	Address and Related Control Signals.....	35
4.7	Data and Related Control Signals.....	37

CHAPTER 5 TIMING

5.1	Standard Read Cycle Timing.....	40
5.2	Standard Write Cycle Timing.....	42
5.3	Multiple Transfer Cycle Timing.....	44
5.4	Quick Interrupt Cycle Timing.....	46

CHAPTER 6 ELECTRICAL SPECIFICATIONS

6.1	Expansion Bus Loading.....	49
6.1.1	Standard Signals.....	50
6.1.2	Clock Signals.....	50
6.1.3	Open Collector Signals.....	51
6.1.4	Non-bussed Signals.....	51
6.2	Slot Power Availability.....	51
6.3	Temperature Range.....	51

CHAPTER 7	MECHANICAL SPECIFICATIONS	
7.1	Basic Zorro III PIC.....	54
7.2	PIC with ISA Option.....	55
7.3	PIC with Video Option.....	56
CHAPTER 8	AUTOCONFIG®	
8.1	The AUTOCONFIG® Mechanism.....	57
8.2	Register Bit Assignments.....	58
CHAPTER 9	ZORRO III SIGNAL NAMES	
9.1	Physical and Logical Signal Names.....	63

PART II: THE AMIGA LOCAL BUS SLOT

CHAPTER 10	INTRODUCTION TO THE LOCAL BUS	
10.1	Intended Audience.....	67
10.2	Why a Local Bus Slot?.....	67
10.3	Why an Expansion Bus Slot?.....	68
CHAPTER 11	FUNCTIONALITY AND DESIGN GUIDELINES	
11.1	Slave Devices.....	69
11.2	Master Devices.....	70
11.2.1	Primary Bus Mastership.....	70
11.2.2	Secondary Bus Mastership.....	71
11.3	Clock Generation.....	72
11.4	Local Bus Design Criteria.....	73
CHAPTER 12	LOCAL BUS SIGNAL DESCRIPTIONS	
12.1	Power Connections	75
12.2	System Initialization.....	75
12.3	68030 Signals.....	76
12.4	Bus Arbitration Signals.....	78
12.5	Other Local Bus Signals.....	78
12.5	Clocks.....	79
12.5	Amiga 3000T Signals.....	79
12.5	Amiga 4000 Signals.....	80
12.6	Local Bus Signal Names and Pin Numbers	81

CHAPTER 13 LOCAL BUS FORM FACTORS

13.1	A4000 Local Bus Form Factor (68040 Coprocessor Board).....	86
13.2	A3000 Local Bus Form Factor.....	87

PART III: THE AMIGA VIDEO SLOT

CHAPTER 14 THE AMIGA VIDEO SLOT

14.1	Evolution of the Amiga Video Slot.....	89
14.2	Video Slot Connector 1.....	90
14.2.1	Power Connections.....	90
14.2.2	Clock Signals.....	91
14.2.3	Video Signals.....	92
14.2.4	Audio Signals.....	93
14.2.5	New A4000 Signals.....	93
14.3	Video Slot Connector 2.....	93
14.3.1	Additional Video Signals.....	93
14.3.2	Additional Audio Signals.....	94
14.3.3	Additional Grounding.....	95
14.3.4	Additional Clock Signals.....	95
14.3.5	Parallel Port Connections.....	95
14.4	Video Slot Pinout.....	97

CHAPTER 15 GENLOCK INTERFACE GUIDELINES

15.1	Hardware Interpretation of HSY* and VSY * Reset Pulses.....	99
15.2	Pulse Duration.....	100
15.3	NTSC and PAL Genlock Interface Guidelines.....	100

CHAPTER 16 VIDEO BOARD FORM FACTORS

16.1	A4000 Video Board Form Factor.....	104
16.2	A3000 Video Board Form Factor.....	105

GLOSSARY.....	107
---------------	-----

TABLES AND FIGURES

Figure 1-1	A3000/A4000 Memory Map.....	4
Figure 2-1	A2000 vs. A3000/A4000 Bus Termination.....	10
Figure 2-2	Expansion Bus Clocks.....	13
Figure 2-3	Zorro II Bus Arbitration.....	16
Figure 3-1	Basic Zorro III Cycles.....	21
Figure 3-2	Multiple Transfer Read Cycles.....	24
Figure 3-3	Zorro III Bus Arbitration.....	26
Figure 3-4	Interrupt Vector Cycle.....	27
Figure 3-5	Zorro II Within Zorro III.....	28
Table 4-1	Memory Space Type Codes.....	36
Table 6-1	Zorro III Drive Types.....	50
Figure 8-1	Configuration Register Mapping.....	58

CHAPTER 1

INTRODUCTION

"Welcome, my son. Welcome to The Machine."

-Pink Floyd

This document describes the complete Zorro III bus, first implemented in the Amiga 3000 Computer. The Zorro III bus is a performance 32-bit expansion bus that is also upward compatible with the Zorro II bus (Amiga 2000 expansion bus). The main intent of the Zorro III bus is to allow fast 32-bit peripherals and memory devices to be added to a high performance Amiga, such as the Amiga 3000, while at the same time allowing standard Zorro II devices to be used wherever they make sense in such a system. This compatibility also insures that the Amiga 3000 will have a number of hardware and software compatible expansion devices available upon introduction, and that Amiga 2000 owners will be able to take their expansion card investment along with them should they migrate to a higher performance Amiga.

1.1 Intended Audience

This document was written primarily for hardware engineers interested in designing Plug In Cards (PICs) for the Zorro III expansion bus. While it may occasionally be of use to software engineers interfacing to such Zorro III PICs, Amiga system software provides an interface layer (*expansion.library* in the Amiga OS) which manages the needs of most card-level software. A reasonable level of microcomputer knowledge is prerequisite to get much meaning out of these pages. A good understanding of the Motorola 680x0 processors will be quite useful, as will be an understanding of the Zorro II expansion bus used on earlier Amiga computers such as the Amiga 2000.

1.2 Bug Reports

This is the second major publication of the *Zorro III Bus Specification*. While every effort has been made to keep it as accurate as possible, there is certainly the possibility that some errors have made it into this document. Anyone finding any error is encouraged to contact Commodore at the address below:

Dave Haynie/Amiga Systems Engineering
1200 Wilson Drive
West Chester, PA 19380

Bugs can also be reported on BIX or via Usenet; on BIX, use the "amiga.com/hardware" conference, or contact Dave Haynie directly as "hazy"; for Usenet users, bug reports can be sent to the address "{uunet,rutgers}!cbmvax.cbm.commodore.com" (use "cbmvax.cbm.commodore.com" if you like domain names); please also copy any such reports to "{uunet,rutgers}!cbmvax!daveh".

1.3 Amiga Bus History

The original Amiga computer, the Amiga 1000, was introduced in 1985. While it had no built-in standard for expandability, the capability for some form of expansion was considered extremely important; personal computer history up to that date had shown several times that an open hardware expansion capability was often critical to a personal computer's success and to its capability to adapt to new or unusual applications. The A1000 was designed with a connector giving access to the internal 68000 bus and a few other system signals. Shortly after introduction, the formal expansion specification for a card chassis that would connect to the A1000 was published. This bus became commonly known as the Zorro bus*. While the backplane specification was very easy to implement with 1985 PAL technology based on the existing 68000 signals, the specification did incorporate a number of advanced features. Far more sophisticated than the IBM-XT/AT and Apple II buses in common use at the time, the Zorro bus allowed any slot to master the bus, and it linked expansion cards with the system software. Addressing jumpers were eliminated, the card's address instead being assigned by software, and cards could easily be identified by software and linked with appropriate driver programs, all with a minimum of user intervention.

With the introduction of the Amiga 2000 system, the Zorro bus was changed slightly. Additional discrete interrupt lines were added, replacing the encoded lines that couldn't easily be used by any bus resident device. As it turns out, these additional encoded lines weren't any more useful, as they couldn't be disabled by software, and as such, they're no longer considered an official part of the Zorro II bus specification (they are supported as part of Zorro III). Finally, the form factor was changed to match that of the IBM PC-AT card, acting as both a cost reduction and allowing the Zorro II bus to offer the PC-AT bus as one optional secondary bus extension. This modified specification became commonly known as the Zorro II bus, and it's the

* The original "Zorro" name comes from the code name of one of the A1000 prototype boards. The "Zorro" board was the one that followed the "Lorraine", and was the board in the works when much of the expansion specifications were worked up. Since everyone uses the "Zorro" name, and no one's suggested a better name, I stick with it throughout this document.

Amiga bus standard that's been in use for most of the Amiga's life. And it's a bus standard that will continue to be important.

1.4 The Zorro III Rationale

With the creation of the Amiga 3000, it became clear that the Zorro II bus would not be adequate to support all of that system's needs. The Zorro II bus would continue to be quite useful, as the current Amiga expansion standard, and so it would have to be supported. A few unused pins on the Zorro II bus and the option of a bus controller custom LSI, gave rise to the Zorro III design, which supports the following features:

- Compatibility with all Zorro II devices.
- Full 32-bit address path for new devices.
- Full 32-bit data path for new devices.
- Bus speed independent of host system CPU speed.
- High speed bus block transfer mode.
- Bus locking for multiprocessor support.
- Cache disable for simple cache support.
- Fair arbitration for all bus masters.
- Cycle by cycle bus arbitration mode.
- High speed interrupt mode.

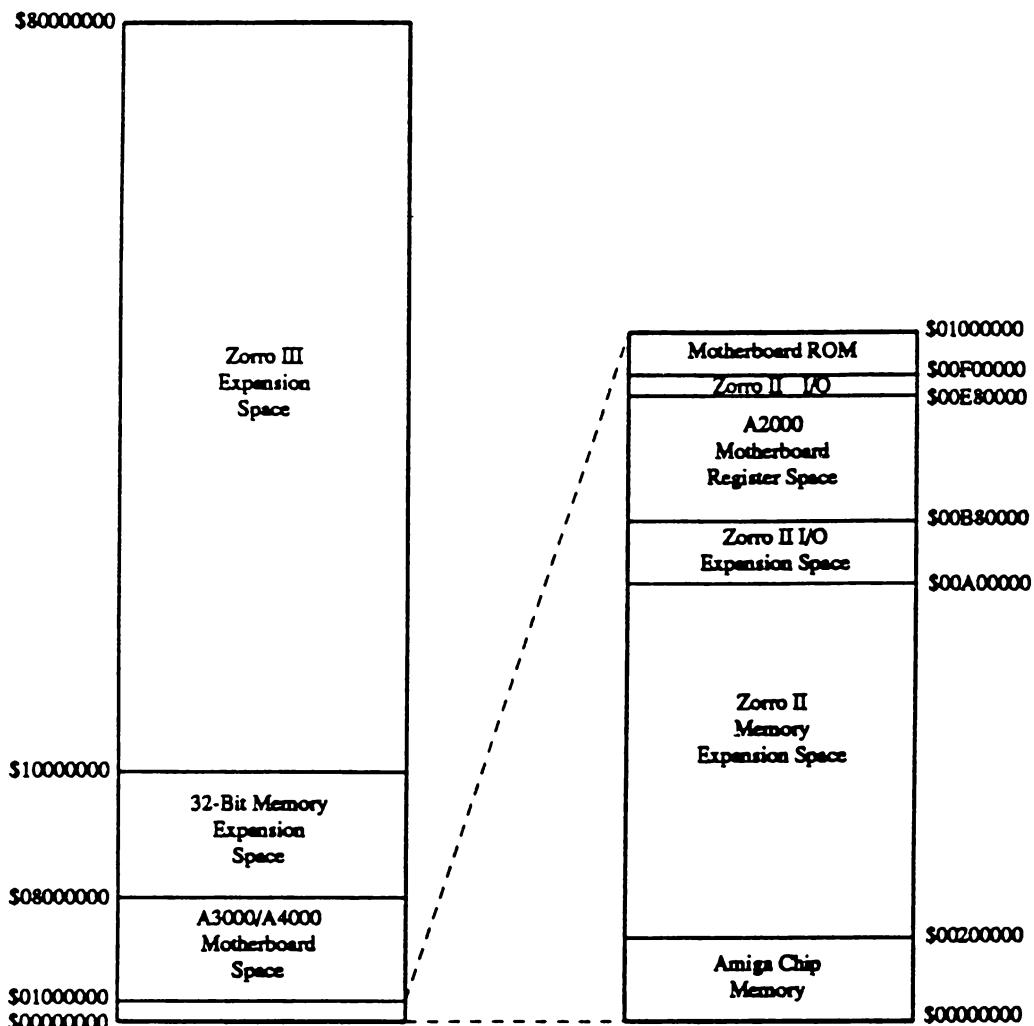
Some of the advanced features, such as burst modes, are designed in such a way as to make them optional; both master and slave arbitrate for them. In addition, it is possible with a bit of extra cleverness, to design a card that automatically configures itself for either Zorro II or Zorro III operation, depending on the status of a sensing pin on the bus.

The Zorro III bus is physically based on the same 100-pin single piece connector as the Zorro II bus. While some bus signals remain unchanged throughout bus operation, other signals change based on the specific bus mode in effect at any time. The bus is geographically mapped into three main sections, *Zorro II Memory Space*, *Zorro II I/O Space*, and *Zorro III Space*. The memory map in *Figure 1-1* shows how these three spaces are mapped in the A3000 and A4000 systems. The Zorro II space is limited to a 16 megabyte region, and since it has DMA access by convention to chip memory, it is in the original 68000 memory map for any bus implementation. The Zorro III space can physically be anywhere in 32-bit memory.

The Zorro III bus functions in one of two different major modes, depending on the memory address on the bus. All bus cycles start with a 32-bit address, since the full 32-bit address is required for proper cycle typing. If the address is determined to be in Zorro II space, a Zorro II compatible cycle is initiated, and all responding slave devices are expected to be Zorro II compatible 16-bit PICs. Should a Zorro III address be detected, the cycle completes when a Zorro III slave responds or the bus times out, as driven by the motherboard logic. It is very important that no Zorro III device respond in Zorro III mode to a Zorro II bus access; as the following chapters will reveal, the two types of cycles make very different use of many of the expansion bus lines, and serious buffer contention can result if the cycle types are somehow

mixed up. The Zorro III bus of course started with the Zorro II bus as its necessary base, but the Zorro III bus mechanisms were designed as much as possible to solve specific needs for high end Amiga systems, rather than extend any particular Zorro II philosophy when that philosophy no longer made any sense. There are actually several variations of the basic Zorro III cycle, though they all work on the same principles. The variations are for optimization of cycle times and for service of interrupt vectors. But all of this in due time.

Figure 1-1: A3000/A4000 Memory Map



1.5 Document Revision History

While there's significantly more real Zorro III hardware actually in existence at the time of this writing than when the first revision of this document was created, various Zorro III issues are still, from time to time, changing. In order to document these changes, this section was created. Although revision histories often discuss revisions in reverse chronological order, it's done here in chronological order to keep the subsection numbers consistent between revisions of this document.

1.5.1 Changes for Rev 0.90

The major changes in Rev 0.90 are actually additions. Specifically, the remaining parts of the Zorro III Timing (Chapter 5) and Mechanical (Chapter 7) specifications have been incorporated into this document. Additionally, the Zorro III design example in Appendix A.4 has been deleted. This simple and somewhat kludgy example has been surplanted by a more useful, straightforward, and thoroughly explained example, available as the separate document *BIGRAM 8/32: A Complete Zorro III Design Example*. In general, we expect both documents to be distributed together, but as always, CATS can assist in the procurement of any missing information.

1.5.2 Changes for Rev 0.91

In the Introduction (Chapter 1), the official revision history has been added as a standard part of this document. The Zorro III Bus Architecture (Chaper 3), section 3.5, has been changed to reflect the revised Quick Interrupt vector allocation mechanism. In the Timing specification (Chapter 5), corrections have been made: timing parameter 6 was left out of the section 5.3 timing, and timing parameter 19 was incorrectly specified in section 5.4. In the AUTOCONFIG® specification (Chapter 8), corrections have been made to the addressing tables for registers 44 and 48. Also the Quick Interrupt Enable bit (register 08:4) and Vector Register (register 50) have been deleted from the specification. Quick Interrupt Vector allocation is now handled via an Exec call, a single configuration unit can have several vectors, and the means of storage on a PIC is up to the designer.

1.5.3 Changes for Rev 1.00

In the AUTOCONFIG® specification (Chapter 8), bit 4 of register 08 has been changed to always read 1 for Zorro III PICs. This change was necessary for compatibility with 1.3, due to a bug in the 1.3 expansion.library. Also, the nybble write configuration mode for the Zorro III configuration block has been eliminated, only byte and word writes are now supported.

1.5.4 Changes for Rev 1.01

The AUTOCONFIG® specification change listed in 1.5.3 was missing from Chapter 8 in Rev 1.00 of the spec, now it's actually there. Additionally, some clarification on the proper action of slave cards and bus error conditions has been added to Chapter 4.

1.5.5 Changes for 1.10

The /INT1, /INT4, /INT5, and /INT7 lines have been eliminated from the Zorro III bus specification. Although the A3000 hardware supports these, some AmigaOS software conventions makes their use impossible under the AmigaOS. These lines are now considered reserved and are not present at all in the A4000. Also, in section 3.5, the vector poll command code was given as 16, where it's actually 15; this has been corrected.

1.5.6 Changes for Rev 1.11

Revision 1.11 of this document includes new information about the A4000 and has additional chapters on the Local Bus (Coprocessor) connector and the video connectors present in all A4000 and A3000 computers.

CHAPTER 2

ZORRO II COMPATIBILITY

"In Jersey anything's legal, as long as ya don't get caught."

- Traveling Wilburys

The Zorro III bus is a rather extensive superset of the Zorro II bus design. The compatibility is based on distinct bus modes, rather than a simple extension to the existing bus mechanisms. Through the use of an integrated bus controller (the Fat Buster chip), the expansion bus configures itself differently for the 16-bit A2000-compatible Zorro II modes than the 32-bit Zorro III modes.

As a result, while there are still only 100 pins on the expansion bus, some pins change function considerably depending on the bus activity that's currently in progress. While the Zorro II modes of the Zorro III bus are as compatible as possible with the Zorro II bus specification (especially the A2000 implementation of this specification), there are some small differences between the two expansion buses.

Aside from these differences, in general, it's important to understand the Zorro II bus in order to understand the Zorro III bus. The general features of Zorro III, like autoconfiguration, the master-slave bus architecture, and the physical attributes come from the Zorro II expansion bus. Other features of the Zorro III bus address shortcomings of the Zorro II architecture, but Zorro II has a hand in how some of these shortcomings are solved under Zorro III. Those with a full understanding of the Zorro II bus will mainly be concerned with the possible bus incompatibilities listed here.

2.1 Changes From The A2000 (Zorro II) Bus

While much effort has been made to assure that the Zorro II mode of the Zorro III bus is as compatible as possible with the A2000, there are a few points to consider here. Primarily, the Zorro II modes of the Zorro III bus are driven with a state machine that emulates the 68000 bus

protocol. This emulation must be based on the published Motorola specifications detailing 68000 bus behavior. While this has the interesting effect of changing the Zorro II bus from CPU-dependent to CPU-independent, there's some margin for error. Zorro II PICs also designed to these specifications should have no trouble in the A3000 bus in most cases. However, anything designed based on observed 68000 behavior rather than documented 68000 operation is at serious risk of failing in an A3000 or A4000, as one might expect. There are also actual documented differences, which are listed below.

2.1.1 6800 Bus Interface

A major difference between the Zorro II mode of Zorro III and the 'real' Zorro II bus are the absence of the signals /VPA and /VMA, which comprise the 6800/6502 peripheral support mechanism that's part of the 68000 bus interface. This mechanism was never a supported part of the Zorro II specification, however, and it should not be used by any PIC. Any Zorro II PIC that depends on /VPA or /VMA will not work in the A3000 bus. It was, in fact, impossible to legally use this on the A2000 bus. The E clock is, however, supported on the Zorro III bus, though its duty cycle may vary in some situations.

2.1.2 Bus Memory Mapping and Cache Support

Another change to the Zorro II implementation is that the bus mapping logic works a little differently. Zorro II address space is broken up into memory and I/O address space. Memory space is the standard 8 megabyte space from \$00200000-\$009FFFFF. The I/O address space is mapped at \$00E80000-\$00EFFFFF, and a new 1.5 megabyte section (previously reserved for motherboard devices) from \$00A00000-\$00B7FFFF. Zorro II cycles are not generated for non-Zorro II address space, even for 68000 space resources on the local bus. So, for example, a CPU access to chip memory would be visible to a Zorro II PIC in an A2000 backplane, but invisible to that same PIC in an A3000 backplane. Since this extra information on the Zorro II backplane can't be legally used by any PIC anyway, it should not be used by any existing A2000 PICs.

The reason for the two distinct mapping regions is for cache support of Zorro II PICs. All access by the local bus* master to Zorro II memory space results in the local bus cache enable signal being driven and a full port read (eg, both bytes) regardless of the actual data transfer size being requested. A local bus access to Zorro II I/O space results in the local bus cache disable signal being driven and the data strobes for reads indicating the requested transfer size. This cache mapping mechanism was first implemented in the A2630 coprocessor card, so it's not an entirely new concept.

* The *local bus*, motherboard bus, and CPU bus are the same thing: the immediate 680x0 bus connected directly to the CPU in an Amiga computer. Current Amiga computers typically support three distinct buses: the expansion bus, local bus, and chip bus. From the point of view of the expansion bus, the local and chip buses appear as a unified device which may be master or slave to the expansion bus.

2.1.3 Bus Synchronization Delays

Due to the asynchronous nature of the local-to-expansion bus interface for Zorro II cycles, extra wait states may occasionally be added for local to expansion or expansion to local cycles. These are generally manifested as delays between consecutive cycles, since the bus controller is not going to require extra waiting during the cycle -- things will have already been synchronized at that point. The synchronization problems get more difficult for Zorro II master access to local bus slaves, and as a result, wait states here are very common. The actual number of wait states generated in any case will be based on the particular implementation.

2.1.4 Zorro II Master Access to Local Slaves

The only supported local bus resource that's guaranteed accessible to a Zorro II expansion bus master as a slave device is chip bus memory. All I/O devices are implementation dependent and not supportable via DMA. Any attempted access to unsupported local bus resources as expansion slaves will result in an error condition being signalled on both the local and the expansion buses. Most other local bus resources, such as local bus fast memory, are located outside of Zorro II space on most systems and obviously not available to Zorro II masters.

2.1.5 Bus Arbitration and Fairness

The Zorro II bus is now arbitrated fairly. The normal slot-based order of precedence is given to requesting devices, just as in the A2000 implementation. As always, once a bus master assumes bus mastership, it has the bus for as long as it wants the bus (of course, trouble can result if a device takes the bus over for too long). Once a master gives up the bus, it will not be granted it back until all subsequent requests have been serviced. Bus arbitration at its best will be slightly slower than in the A2000 implementation, due to the fairness logic, but it is impossible to jam the arbiter with asynchronous bus requests as in the A2000. The new style arbiter also holds off bus grants while hidden local bus cycles are in progress, so there's no guarantee of a minimum time between bus request and bus grant specified.

2.1.6 Intelligent Cycle Spacing

In order to permit a free intermix of Zorro II and Zorro III cycles, the bus control logic is capable of making intelligent decisions when spacing bus cycles. In some cases, a Zorro II cycle has some component that would naturally extend into a following cycle. The cycle spacing logic detects such a condition, and refuses to start a new cycle until the current one is complete, even if this extends beyond the defined bounds of a Zorro II cycle. For Zorro II PICs that really follow the Zorro II specifications, this should have no effect. However, any Zorro II PIC that holds signals much beyond the end of a cycle, especially critical signals like /SLAVE and /DTACK, will likely incur additional wait states on the Zorro III bus. This is not intended as a license for making sloppy expansion card designs, just an acknowledgement that some Zorro II devices may cause a conflict with the faster Zorro III bus timings, and the best thing to do about such cases is to make them work, even with a possible performance penalty.

2.1.7 Bus Drive and Termination

Finally, the Zorro III bus uses different bus termination than that in the A2000. The Zorro II specification didn't specify the termination expected; backplanes were built that didn't even have termination. The A2000 bus used a circuit consisting of a capacitor in series with a resistor to ground for most of the bus signals. This has good reflection cancelling properties without increasing crosstalk (a major concern on the 2-layer A2000 motherboard), but it does slow things down measureably. The main reason for the change on the A3000 backplane is to support the faster Zorro III bus modes. The multi-layer A3000 motherboard permits a

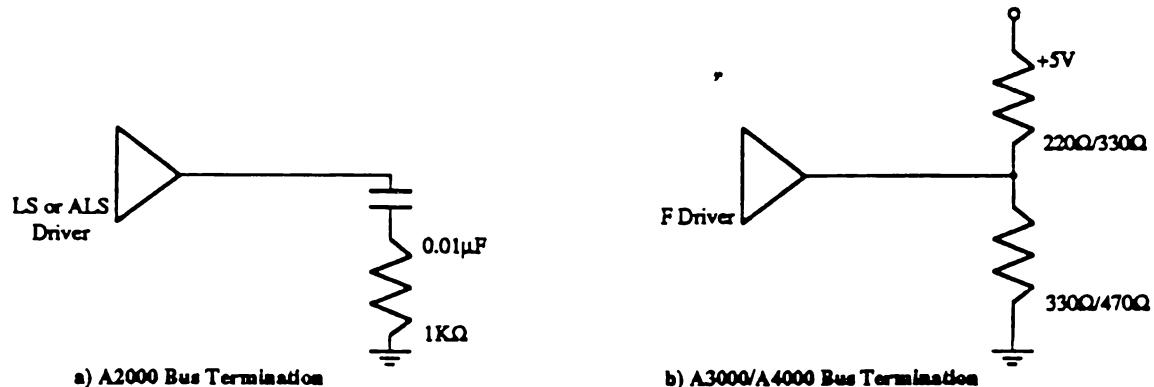


Figure 2-1: A2000 vs. A3000/A4000 Bus Termination

reasonably high current bus without undue crosstalk. The thevenin termination makes switching logic levels start from a midpoint instead of a rail, especially for a bus coming out of tri-state (which, based on the Zorro III design, happens constantly). This should not cause problems with Zorro II cards, but it's conceivable that some cards may need to be adjusted to work in this bus (the Zorro III bus requires somewhat higher current capability than the Zorro II bus does. The A3000 and A4000 do not support enough slots for loading to be a likely problem, but future Zorro III backplanes will have more slots and make this an important consideration).

2.1.8 DMA Latency and Overlap

Zorro II bus masters in a Zorro III backplane will, in many cases, receive a bus grant much sooner than they would in a standard Zorro II backplane. Additionally, in some cases, expansion bus cycles will overlap local bus cycles. The latency incurred on the Zorro II bus during heavy custom chip activity has been greatly reduced for any Zorro III bus master. This should be transparent to the card in question, though it's a good thing to be aware of.

2.1.9 Power Supply Differences

The Zorro II bus is defined as supplying +5VDC @ 2 Amps to each slot, with one slot per backplane supplying 5.0VDC @ 4.0 Amps. The Zorro III bus only provides the 5.0VDC @ 2.0 Amps for each slot.

2.2 Bus Architecture

The Zorro II bus is a simple extention of the 68000 processor bus. Those without a good knowledge of the 68000 local bus will find *The 68000 User's Manual* from Motorola an excellent reference for many Zorro II issues. The *A500/A2000 Technical Reference Manual* from Commodore-Amiga is also required reading for any Zorro II design issues, as it includes a complete description of all the Commodore-Amiga details that aren't part of the 68000 specification.

The basic Zorro II bus is a buffered version of the 68000 processor bus, physically provided on a 100-pin one-piece connector. The bus is 16 bits wide, and provides 24 bits of addressing information. A bus cycle looks exactly like a 68000 bus cycle. The cycle is defined by an address strobe, terminated by a data transfer strobe, and qualified by a read/write strobe, some memory space qualifiers, and one or two byte selection strobes. The basic bus cycle runs for a total of four cycles of a 7.16 MHz clock, though it can be extended to add wait states when required.

The Zorro II bus adds a number of features to the basic 68000 CPU bus. It supplies some Amiga system signals that are useful for expansion card designs, such as many of the **Amiga system clocks**. The bus provides a default data transfer signal, which expansion cards can easily use and modify rather than go to the trouble of creating their own. It provides a number of discrete interrupt lines which are mixed to provide the 68000 with its standard encoded interrupts. The 68000 bus arbitration protocol is used to allow multiple bus masters; arbitration of the bus requests are managed by the Zorro II bus controller to avoid contention between multiple masters. And of course the bus supplies a number of supply voltages for powering cards.

A powerful aspect of the Zorro II bus is its convention for automatically configuring expansion cards, AUTOCONFIG®. On system powerup, the system software interrogates each board to determine what kind of board is installed and how much memory space it needs on the bus. The software then tells each board where to reside in memory. The bus provides hardware lines to allow the boards to be configured in a daisy chained fashion regardless of which slots they occupy and to prevent damage to boards if accidentally configured to reside at the same memory location. Firmware standards also permit software to autoboot or autoinitilize any board, to match soft-loaded device drivers with individual boards, and to link memory boards into the appropriate system memory lists.

2.3 Signal Description

The Zorro II bus can be broken down into various logical signal groups. Some of these groups are unchanged in the Zorro III bus modes, others are drastically different. This section makes note of the original Zorro II name for each signal and the current Zorro III physical pin name for each signal, where different. Some of this information will be repeated in the Zorro III chapters, where appropriate; nothing in this chapter is considered critical to understanding the Zorro III bus, but it is useful. As previously mentioned, the A2000 bus signals unsupported by

the Zorro II specification have been deleted from both the Zorro III specification and its implementation in the A3000 and A4000; this section will, however, document those signals for reference purposes. Please see Chapter 9 for a complete list with pin numbers of the various logical signals that appear on the physical bus during the different phases of the Zorro II and Zorro III bus cycles.

2.3.1 Power Connections

The Zorro III expansion bus provides several different voltages designed to supply expansion devices. There are no changes here that affect Zorro II cards.

Digital Ground (Ground)

This is the digital supply ground used by all expansion cards as the return path for all expansion supplies.

Main Supply (+5VDC)

This is the main power supply for all expansion cards, and it is capable of sourcing large currents; each expansion slot can draw up to 2.0 Amps @ +5VDC. The extra power for one card in any backplane drawing up to 4.0 Amps @ +5VDC is no longer supported.

Negative Supply (-5VDC)

This is a negative version of the main supply, for small current loads only. There is no maximum load specified for the Zorro II bus on a per-slot basis; the A2000 implementation specifies 0.3 Amps @ -5VDC for the entire system.

High Voltage Supply (+12VDC)

This is a higher voltage supply, useful for communications cards and other devices requiring greater than digital voltage levels. This is intended for relatively small current loads only. There is no maximum load specified for the Zorro II bus on a per-slot basis; the A2000 implementation specifies 8.0 Amps @ +12VDC for the entire system, most of which is normally devoted to floppy and hard disk drive motors, not slots.

Negative High Supply (-12VDC)

Negative version of the high voltage supply, also commonly used in communications applications, and similarly intended for small loads only. There is no maximum load specified for the Zorro II bus on a per-slot basis; the A2000 implementation specifies 0.3 Amps @ -12VDC for the entire.

2.3.2 Clock Signals

The Zorro III expansion bus provides clock signals for expansion boards. These clocks are for synchronous Zorro II designs and for other synchronous activity such as bus arbitration. While originally based on Amiga local bus clocks, these have no guaranteed relationship to any local bus activity in newer Amiga computers, but are maintained in Amiga computers as part of the expansion bus specification. The relationship between these clocks is illustrated in *Figure 2-2*.

/C1 Clock

This is a 3.58 MHz clock (3.55 MHz on PAL systems) that's synched to the falling edge of the 7M system clock.

/C3 Clock

This is a 3.58 MHz clock (3.55 MHz on PAL systems) that's synched to the rising edge of the 7M system clock.

CDAC Clock

This is a 7.16 MHz system clock (7.09 MHz on PAL systems) which trails the 7M clock by 90° (approximately 35ns).

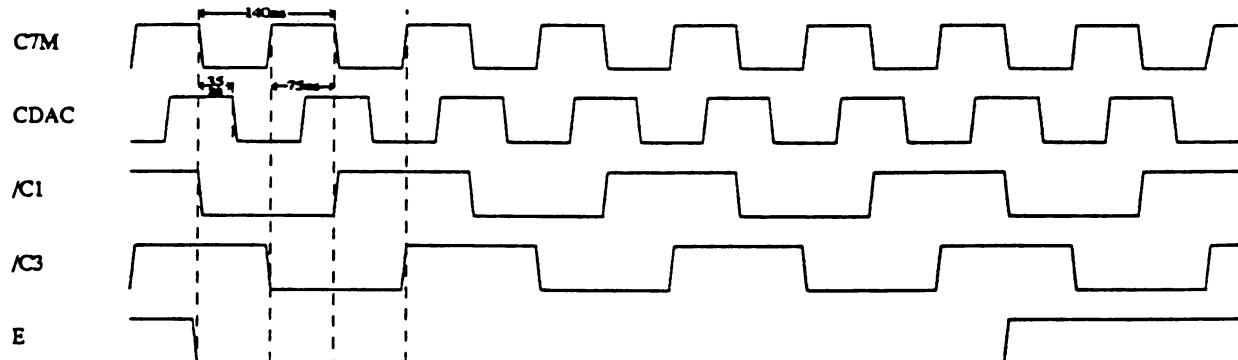


Figure 2-2: Expansion Bus Clocks

E Clock

This is the 68000 generated "E" clock, used for 6800 family peripherals driven by "E" and 6502 peripherals driven by Φ_2 . This clock is four 7M clocks high, six clocks low, as per the 68000 spec. Note that the bus does not support the rest of the 68000's 6800/6502 compatible interface; there may be better ways to clock such devices.

7M Clock

This is the 7.16 MHz system clock (7.09 MHz on PAL systems). This clock forms the basis for all Zorro II/68000 compatible activity, and for various other system functions, such as bus arbitration.

2.3.3 System Control Signals

The signals in this group are available for various types of system control; most of these have an immediate or near immediate effect on expansion cards and/or the system CPU itself.

Bus Error (/BERR)

This is a general indicator of a bus fault condition. Any expansion card capable of detecting a hardware error relating directly to that card can assert /BERR when that bus error condition is detected, especially any sort of harmful hardware error condition. This

signal is the strongest possible indicator of a bad situation, as it causes all PICs to get off the bus, and will usually generate a level 2 exception on the host CPU. For any condition that can be handled in software and doesn't pose an immediate threat to hardware, notification via a standard processor interrupt is the better choice. The bus controller will drive /BERR in the event of a detected bus collision or DMA error (an attempt by a bus master to access local bus resources it doesn't have valid access permission for).

All cards must monitor /BERR and be prepared to tri-state all of their on-bus output buffers whenever this signal is asserted. The current bus master should, if possible, retry the bus cycle after /BERR is negated unless conditions warrant otherwise. Since any number of devices may assert /BERR, and all bus cards must monitor it, any device that drives /BERR must drive with an open collector or similar device capable of sinking at least 12ma, and any device that monitors /BERR should place a minimal load on it (1 "F" type load or less). This signal is pulled high by a passive backplane resistor.

System Reset (/RST, /BUSRST) ≡ (/RESET, /IORST) for Zorro III

The bus supplies two versions of the system reset signal. The /RST signal is bidirectional and unbuffered, allowing an expansion card to hard reset the system. It should only be used by boards that need this reset capability, and is driven only by an open collector or similar device. The /BUSRST signal is a buffered output-only version of the reset signal that should be used as the normal reset input to boards not concerned with resetting the system on their own. All expansion devices are required to reset their autoconfiguration logic when /BUSRST is asserted. This signal is pulled high by a passive backplane resistor.

System Halt (/HLT)

This signal is similar to the 68000 processor halt signal, and is driven by a PIC with an open-collector or similar gate only. Its main use is to indicate a full-system reset. Based on the 68000 conventions, an I/O-only reset, such as initiated by the 680x0 RESET instruction, will drive only /RST and /BUSRST on the bus. A full-system reset, such as a powerup reset or a keyboard reset, drives /HLT low as well. PICs that wish to reset the system CPU as well as the bus and I/O devices drive /RST and /HLT, some bus devices such as processor cards may internally reset only on full-system resets. This signal is pulled high by a passive backplane resistor.

System Interrupts

Six of the decoded, level sensitive 680x0 interrupt inputs were originally available on the expansion bus, and these are labelled as /INT₂, /INT₆, /EINT₁, /EINT₄, /EINT₅, /EINT₇ on the Zorro II bus. Only the /INT₂ and /INT₆ interrupt inputs are actually supported by Commodore-Amiga as part of the Zorro II specification; the A2000 hardware did not provide the to software the required support mechanisms for the safe use of these lines. Each of these interrupt lines are shared by wired ORing, thus each line must be driven by an open-collector or equivalent output type, and all are pulled high by passive backplane resistors.

2.3.4 Slot Control Signals

This group of signals is responsible for the control of things that happen between expansion slots.

Slave (/SLAVEn)

Each slot has its own /SLAVE output, driven actively, all of which go into the collision detect circuitry. The "N" refers to the expansion slot number of the particular /SLAVE signal. Whenever a Zorro II PIC is responding to an address on the bus, it must assert its /SLAVE output within 35ns of /AS asserted. The /SLAVE output must be negated at the end of a cycle within 50ns of /AS negated. Late /SLAVE assertion on a Zorro II bus can result in loss of data setup times and other problems. A late /SLAVE negation for Zorro II cards can cause a collision to be detected on the following cycle. While the Zorro III sloppy cycle logic eliminates this fatal condition, late /SLAVE negation can nonetheless slow system performance unnecessarily. If more than one /SLAVE output occurs for the same address, or if a PIC asserts its /SLAVE output for an address reserved by the local bus, a collision is registered and results in /BERR being asserted.

Configuration Chain (/CFGIN_N, /CFGOUT_N)

The slot configuration mechanism uses the bus signals /CFGOUT_N and /CFGIN_N, where "N" refers to the expansion slot number. Each slot has its own version of each, which make up the configuration chain between slots. Each subsequent /CFGIN is a result of all previous /CFGOUTs, going from slot 0 to the last slot on the expansion bus. During the AUTOCONFIG® process, an unconfigured Zorro PIC responds to the 64K address space starting at \$00E80000 if its /CFGIN signal is asserted. All unconfigured PICs start up with /CFGOUT negated. When configured, or told to "shut up", a PIC will assert its /CFGOUT, which results in the /CFGIN of the next slot being asserted. The backplane passes on the state of the previous /CFGOUT to the next /CFGIN for any slot not occupied by a PIC, so there's no need to sequentially populate the expansion bus slots.

Data Output Enable (DOE)

This signal is used by an expansion card to enable the buffers on the data bus. The main Zorro II use of this line is to keep PICs from driving data on the bus until any other device is completely off the bus and the bus buffers are pointing in the correct direction. This prevents any contention on the data bus.

2.3.5 DMA Control Signals

There are various signals on the expansion bus that coordinate the arbitration of bus masters. Native Zorro III bus masters use some of the same logical signals, but their arbitration protocol is considerably different.

PIC is DMA Owner (/OWN)

This signal is asserted by an expansion bus DMA device when it becomes bus master. This output is to be treated as a wired-OR output between all expansion slots, any of

which may have a PIC signalling bus mastership. Thus, this should be driven with an open-collector or similar output by any PIC using it. This signal is the main basis for data direction calculations between the local and expansion busses, and is pulled up by a backplane resistor.

Slot Specific Bus Arbitration (/BR_N, /BG_N)

These are the slot-specific /BR_N and /BG_N signals, where "N" refers to the expansion slot number. The bus request from each board is taken in by the bus controller and ultimately used to take over the system from 680x0 on the local bus. The bus controller eventually returns one bus grant to the winner among all requesting PICs. From the point of view of

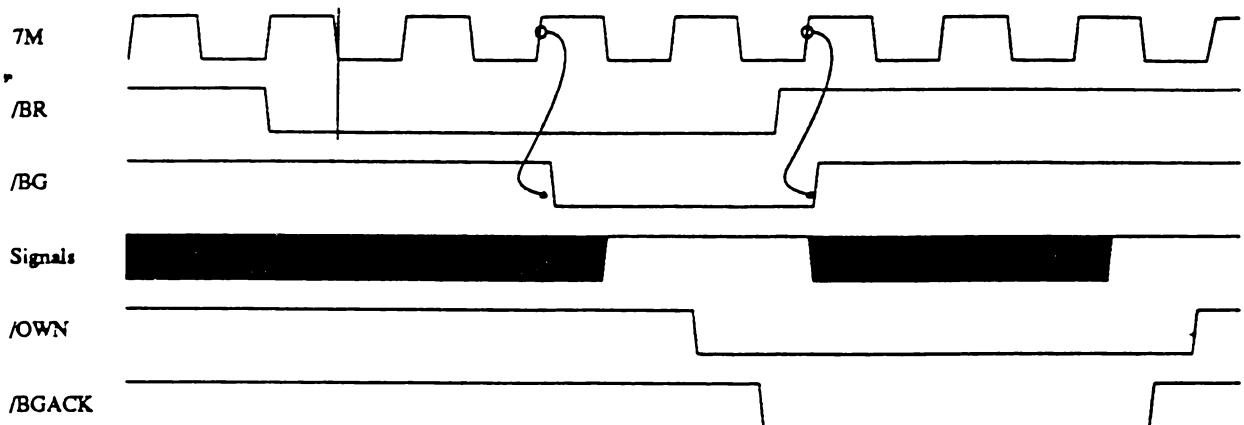


Figure 2-3: Zorro II Bus Arbitration

the individual PIC, the protocol is very similar to that of the 68000 arbitration mechanism. The PIC asserts /BR_N on the rising edge of 7M; some time later, /BG_N is returned on the falling edge of 7M. The PIC waits for all bus activity to finish, asserts /OWN followed by /BGACK, then negates /BR_N, assuming bus mastership. It retains mastership until it negates /BGACK followed by /OWN.

Bus Grant Acknowledge (/BGACK)

Any Zorro II PIC that receives a bus grant asserts this signal as long as it maintains bus mastership. This signal may never be asserted until the bus grant has been received, /AS is negated, /DTACK is negated, and /BGACK itself is negated, indicating that all other potential bus masters have relinquished the bus. This output is driven as a wired-OR output, so all PICs must drive it with an open collector or equivalent device, and a passive pullup is supplied by the backplane.

Bus Want/Clear (/GBG) ≡ (/BCLR) for Zorro III

This signal is asserted by the bus controller to indicate that a PIC wants to master the bus. A bus master assumes that the host CPU wants the bus, and that any time wasted as master is stealing time from the CPU. To avoid such waste, a master should use cache or FIFO to grab slow-coming data, and then transfer it all at once. /BCLR is asserted to indicate that additionally, another PIC wants the bus, and the current bus master should get off as soon as possible. This signal is equivalent to /GBG on the A2000 bus.

2.3.6 Addressing and Control Signals

These signals are various items used for the addressing of devices in Zorro II mode by the local bus and any expansion DMA devices. Most of these signals are very much like 68000 generated bus signals bi-directionally buffered to allow any DMA device on the bus to drive the local bus when such a device is the bus master.

Read Enable (READ)

This is the read enable for the bus, which is equivalent to the 68000's R/W output. READ asserted during a bus cycle indicates a read cycle, READ negated indicates a write cycle. Note that this signal may become valid in a cycle earlier than a 68000 R/W line would, but it remains valid at least as long as long at the cycle's end.

Address Bus (A₁-A₂₃)

This is logically equivalent to the 68000's address bus, providing 16 megabytes of address space, although much of that space is not assigned to the expansion bus (see the memory map in *Figure 1-1*).

Address Strobe (/AS) ≡ (/CCS) for Zorro III

This is equivalent to the 68000 /AS, called /CCS, for Compatibility Cycle Strobe, in the Zorro III nomenclature. The falling edge of this strobe indicates that addresses are valid, the READ line is valid, and a Zorro II cycle is starting. The rising edge signals the end of a Zorro II bus cycle, signaling the current slave to negate all slave-driven signals as quickly as possible. Note that /CCS, like /AS, can stay asserted during a read-modify-write access over multiple cycle boundaries. To correctly support such cycles, a device must consider both the state of /CCS and the state of the data strobes. Many current Zorro II cards don't correctly support this 680x0 style bus lock.

Data Bus (D₀-D₁₅)

This is a buffered version of the 680x0 data bus, providing 16 bits of data accessible by word or either byte. A PIC uses the DOE signal to determine when the bus is to be driven on reads, and the data strobes to determine when data is valid on writes.

Data Strobes (/UDS, /LDS) ≡ (/DS₃, /DS₂) for Zorro III

These strobes fall on data valid during writes, and indicate byte select for both reads and writes. The lower strobe is used for the lower byte (even byte), the upper strobe is used for the upper byte (odd byte). There is one slight difference between these lines and the 68000 data strobes. On reads of Zorro II memory space, both /DS₃ and /DS₂ will be asserted, no matter what the actual size of the requested transfer is. This is required to support caching of the Zorro II memory space. For Zorro II I/O space, these strobes indicate the actual, requested byte enables, just as would a 68000 bus master.

Data Transfer Acknowledge (/DTACK)

This signal is used to normally terminate both Zorro bus cycles. For Zorro II modes, it is equivalent to the 68000's Data Transfer Acknowledge input. It can be asserted by the

bus slave during a Zorro II cycle at any time, but won't be sampled by the bus master until the falling edge of the S₄ state on the bus. Data will subsequently be latched on the S₆ falling edge after this, and the cycle terminated with /AS negated during S₇. If a Zorro II slave does nothing, this /DTACK will be driven by the bus controller with no wait states, making the bus essentially a 4 cycle synchronous bus. Any slow device on the bus that needs wait states has two options. It can modify the automatic /DTACK negating XRDY to hold off /DTACK. Alternately, it may assert /OVR to inhibit the bus controller's generation of /DTACK, allowing the slave to create its own /DTACK. Any /DTACK supplied by a slave must be driven with an open-collector or similar type output; the backplane provides a passive pullup.

Processor Status (FC₀-FC₂)

These signals are the cycle type or memory space bits, equivalent for the most part with the 68000 Processor Status outputs. They function mainly as extensions to the bus address, indicating which type of access is taking place. For Zorro II devices, any use of these lines must be gated with /BGACK, since they are not driven valid by Zorro II bus masters. However, when operating on the Zorro III backplane, Zorro II masters that don't drive the function codes will be seen generating an FC₁ = 0, which results in a valid memory access. Zorro II cycles are not generated for invalid memory spaces when the CPU is the bus master.

/DTACK Override (/OVR)

This signal is driven by a Zorro II slave to allow that slave to prevent the bus controller's /DTACK generation. This allows the slave to generate its own /DTACK. The previous use of this line to disable motherboard memory mapping, which was unsupported on the A2000 expansion bus, has now been completely removed. The use of XRDY or /OVR in combination with /DTACK is completely up to the board designer -- both methods are equally valid ways for a slave to delay /DTACK. In Zorro III mode, this pin is used for something completely different.

External Ready (XRDY)

This active high signal allows a slave to delay the bus controller's assertion of /DTACK, in order to add wait states. XRDY must be negated within 60ns of the bus master's assertion of /AS, and it will remain negated until the slave wants /DTACK. The /DTACK signal will be asserted by the bus controller shortly following the assertion of XRDY, providing the bus cycle is a S₄ or later. XRDY is a wired-OR from all PICs, and as such, must be driven by an open collector or equivalent output. In Zorro III mode, this pin is used for something completely different.

CHAPTER 3

BUS ARCHITECTURE

"We follow in the steps of our ancestry, and that cannot be broken."

-Midnight Oil

While the Zorro II bus design was based in a large part on an already existing bus cycle, the 68000 cycle, the Zorro III bus design had a much different set of preconditions. It is not modeled after any particular CPU specific bus protocol, but instead it's a logical outgrowth of both the need to support Zorro II cards on the same bus and the need to achieve various modern feature and performance goals. These goals were summarized in Chapter 1, now they'll be covered in greater detail here.

3.1 Basic Zorro III Bus Cycles

The basic Zorro III bus cycle is a multiplexed address/data cycle which supplies a full 32 bits worth of address and data per simple cycle. The cycle is a fully asynchronous cycle. The bus master for a given cycle supplies strobes to indicate when address is valid, write data is valid, and read data may be driven. In return, the bus slave for a cycle supplies a strobe to indicate that it is responding to a bus address, and a strobe to indicate that it is done with the bus data for a write cycle, or has supplied valid bus data for a read cycle. The minimum theoretical bus speed is governed only by setup and hold time requirements for the various bus signals. Actual bus speeds are always a function of the bus master and bus slave active for a given cycle. This is considerably different than the way things work under the Zorro II bus, and for several good reasons, which are explained below.

3.1.1 Design Goals

For any computer bus, there are two basic possibilities concerning the fundamental operation of the bus; it's either synchronous or asynchronous. The difference is simple -- the synchronous bus is ultimately tied to a clock of some sort, while the asynchronous bus has no defined relationship to any clock signal. While Motorola specifies the 68000 bus cycle as an asynchronous cycle, they're really referring to the fact that most 68000 inputs are internally synchronized with the bus clock, and therefore, synchronous setup times on the bus do not have to be met to avoid metastability. But the 68000 bus, and the Zorro II bus by extension, are synchronous buses, based on a single bus clock (called E7M on the Zorro II bus). Most Zorro II signals are asserted relative to an edge of the bus clock, and most Zorro II inputs are sampled on an edge of the bus clock. The minimum Zorro II cycle is four bus clocks long, and every wait state added, regardless of the method, will result in a single additional bus clock wait, regardless of the asynchronous appearance of the termination and wait signals on the Zorro II bus.

The Zorro III bus is a fully asynchronous bus, in that all bus events are driven by strobes, and there is no reference clock. The choice of an asynchronous versus a synchronous bus design is governed by the intended application of the bus. Synchronous designs are preferred when a CPU and a memory system (e.g., master and slave) can be very tightly coupled to each other. Such designs generally require a tight adherence to timing based on the specific CPU. This is optimal for tightly coupled systems, such as fast memory on the A3000 local bus. Synchronous designs can also be easier to do accurately, as the designer can use clock edges for scheduling events, and there's never any need to waste time in synchronizers to achieve a reliable design.

The design goals for an expansion bus are considerably different. While a fast memory circuit on a system motherboard can change for every new and better design, it's not feasible to require redesign of any significant number of expansion cards every time an improved motherboard design is created. And while a synchronous transfer can be optimal for matched clocks, it can be very inefficient for mismatched CPU and expansion clocks, as synchronizer delays must be introduced for any reliable operation. The A3000 project started with the need to support CPU systems at 16MHz and at 25MHz, and it's obvious that the growth of CPU clock speed will be here for some time to come. Zorro III cards are based on asynchronous handshaking between master and slave in both directions. This means that, as long as masters and slaves manage their own needs, any slave can work with any master. But as masters and slaves improve with technology, bus transfer speeds can automatically increase, without rendering any slower cards obsolete. The Zorro III bus attempts to address the needs of device expansion as much as the needs of memory expansion.

3.1.2 Simple Bus Cycle Operation

The normal Zorro III bus cycle is quite different than the Zorro II bus in many respects. *Figure 3-1* shows the basic cycle. There is no bus clock visible on the expansion bus; the standard Zorro II clocks are still active during Zorro III cycles, but they have no relationship to the Zorro II bus cycle. Every bus event is based on a relationship to a particular bus strobe, and strobes are alternately supplied by master and slave.

A Zorro III cycle begins when the bus master simultaneously drives addressing information on the address bus and memory space codes on the FC_n lines, quickly following that with the assertion of the Full Cycle Strobe, /FCS; this is called the *address phase* of the bus. Any active slaves will latch the bus address on the falling edge of /FCS, and the bus master will tri-state the addressing information very shortly after /FCS is asserted. It's necessary only to latch A₃₁-A₈; the low order A₇-A₂ addresses and FC_n codes are non-multiplexed.

As quickly as possible after /FCS is asserted, a slave device will respond to the bus address by asserting its /SLAVEn line, and possibly other special-purpose signals. The autoconfiguration process assigns a unique address range to each PIC base on its needs, just as on the Zorro II bus. Only one slave may respond to any given bus address; the bus controller will generate a /BERR signal if more than one slave responds to an address, or if a single slave

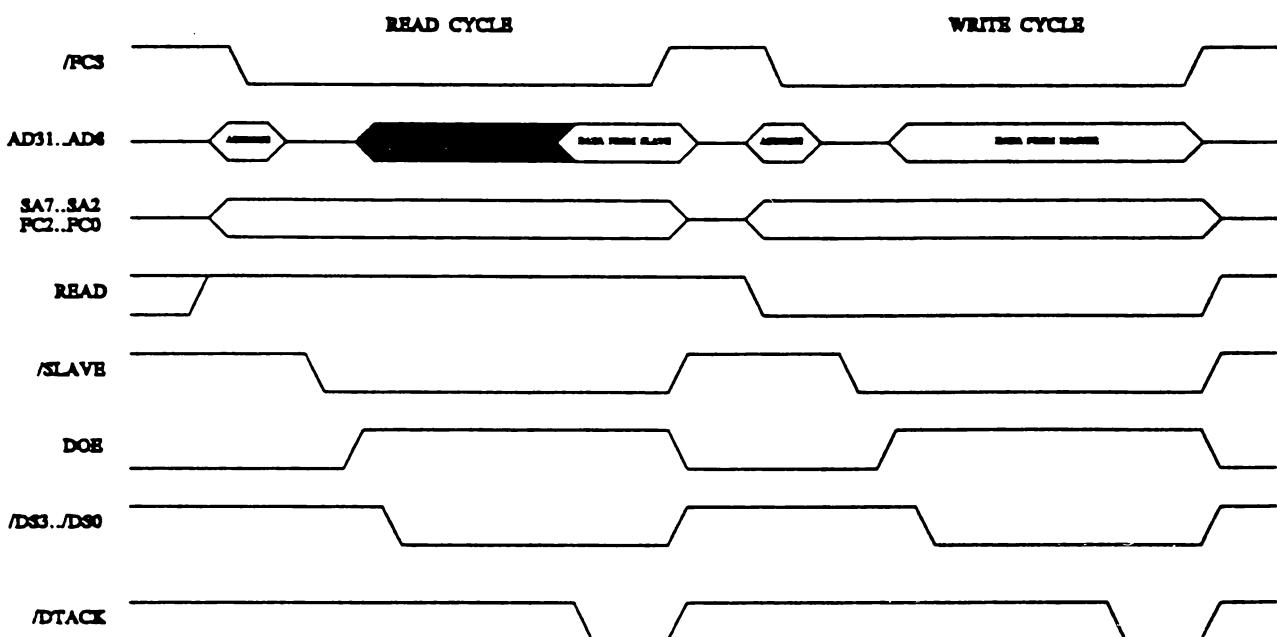


Figure 3-1: Basic Zorro III Cycles

responds to an address reserved for the local bus (this is called a bus collision, and should never happen in normal operation). Slaves don't usually respond to CPU memory space or other reserved memory space types, as indicated by the memory space code on the FC_n lines (see Chapter 4 for details)!

The *data phase* is the next part of the cycle, and it's started when the bus master asserts DOE onto the bus, indicating that data operations can be started. The strobes are the same for both read and write cycles, but of course the data transfer direction is different.

For a read cycle, the bus master drives at least one of the data strobes /DS_n, indicating the physical transfer size requested (however, cachable slaves must always supply all 32 bits of data). The slave responds by driving data onto the bus, and then asserting /DTACK. The bus master then terminates the cycle by negating /FCS, at which point the slave will negate its

`/SLAVEn` line and tri-state its data. The cycle is done at this point. There are a few actions that modify a cycle termination, those will be covered in later sections.

The write cycle starts out the same way, up until `DOE` is asserted. At this point, it's the master that must drive data onto the bus, and then assert at least one `/DSN` line to indicate to the slave that data is valid and which data bytes are being written. The slave has the data for its use until it terminates the cycle by asserting `/DTACK`, at which point the master can negate `/FCS` and tri-state its data at any point. For maximum bus bandwidth, the slave can latch data on the falling edge of the logically ORed data strobes; the bus master doesn't sample `/DTACK` until after the data strobes are asserted, so a slave can actually assert `/DTACK` any time after `/FCS`.

3.2 Advanced Mode Support Logic

The Zorro III bus provides support for some more advanced things that weren't generally handled correctly on the Zorro II bus. Amiga computers have traditionally been supporting things that the more mainstream personal computers haven't. High speed DMA transfers and expansion coprocessors such as the Bridge Cards have been with the Amiga since the early days, and high performance main system CPUs with cache memory are now becoming common. The Zorro II bus never properly or easily supported such devices; the Zorro III bus attempts to make support of cache and coprocessor both possible and relatively straightforward. Other new features are covered in later sections.

3.2.1 Bus Locking

The first advanced modification of the basic bus cycle is bus locking, via the `/LOCK` signal. Bus locking is a hardware convention that allows a bus master to guarantee several cycles will be atomic on the bus. This is necessary to support the sharing of special "mail-box" memory between a bus master and an alternate PIC-based processor; Bridge Cards are an example of this kind of device. The Zorro II bus itself supports bus locking via the 68000 convention. However, the 68000 style of bus locking is often difficult to implement, and support for it was often ignored in Zorro II designs, especially those not directly concerned with multiprocessor support.

The Zorro III mechanism involves no change to the basic bus cycle, other than the monitoring of this `/LOCK` signal, and as such is much more reasonable to support. The `/LOCK` signal is asserted by a bus master at address time and maintained across cycles to lock out shared memory coprocessors, allowing hardware backed semaphores to easily be used between such coprocessors. We expect multiprocessing will be a greater concern on the Zorro III bus than it is at present; video coprocessors, RISC devices, and special purpose processors for image processing or mathematics should find a comfortable home on the Zorro III bus.

3.2.2 Cache Support

The other advanced cycle modifier on the Zorro III bus is the cache inhibit line, `/CINH`. On the Zorro II bus, there was originally no caching envisioned, and therefore no real support for

caching of Zorro II PICs. First in the A2630 and later in the Zorro III bus's emulation of Zorro II, conventions were adopted to permit caching of Zorro II cards. These conventions aren't perfect; MMU tables will sometimes have to supplant this geographic mapping. While Zorro III doesn't have any cache consistency mechanisms for managing caches between several caching bus masters, it does allow cards that absolutely must not be cached to assert a cache inhibit line, /CINH, on a per-cycle basis (asserted at slave time by a responding slave). This cache management is basically the lowest level of a cache management system, mainly useful for support of I/O and other devices that shouldn't be cached. Software will be required for the higher levels of cache management.

3.3 Multiple Transfer Cycles

The multiplexed address/data design of the Zorro III bus has some definite advantages. It allows Zorro III cards to use the same 100-pin connector as the Zorro II cards, which results in every bus slot being a 32-bit slot, even if there's an alternate connector in-line with any or all of the system slots; current alternate connectors include Amiga Video and PC-AT (now sometimes called ISA, for *Industry Standard Architecture*, now that it's basically beyond the control of IBM) compatible connectors. This design also makes implementation of the bus controller for a system such as the A3000 simpler. And it can result in lower cost for Zorro III PICs in many cases.

The main disadvantage of the multiplexed bus is that the multiplexing can waste time. The address access time is the same for multiplexed and non-multiplexed buses, but because of the multiplexing time, Zorro III PICs must wait until *data time* to assert data, which places a fixed limit on how soon data can be valid. The Zorro III Multiple Transfer Cycle is a special mode designed to allow the bus to approach the speed of a non-multiplexed design. This mode is especially effective for high speed transfers between memory and I/O cards.

As the name implies, the Multiple Transfer Cycle is an extension of the basic full cycle that results in multiple 32-bit transfers. It starts with a normal full cycle *address phase* transaction, where the bus master drives the 32-bit address and asserts the /FCS signal. A master capable of supporting a Multiple Transfer Cycle will also assert /MTCR at the same time as /FCS. The slave latches the address and responds by asserting its /SLAVEn line. If the slave is capable of multiple transfers, it'll also assert /MTACK, indicating to the bus master that it's capable of this extended cycle form. If either /MTCR or /MTACK is negated for a cycle, that cycle will be a basic full cycle.

Assuming the multiple transfer handshake goes through, the multiple cycle continues to look similar to the basic cycle into the data phase. The bus master asserts DOE (possibly with write data) and the appropriate /DSN, then the slave responds with /DTACK (possibly with read data at the same time), just as usual. Following this, however, the cycle's character changes. Instead of terminating the cycle by negating /FCS, /DSN, and DOE, the master negates /DSN and /MTCR, but maintains /FCS and DOE. The slave continues to assert /SLAVEn, and the bus goes into what's called a *short cycle*.

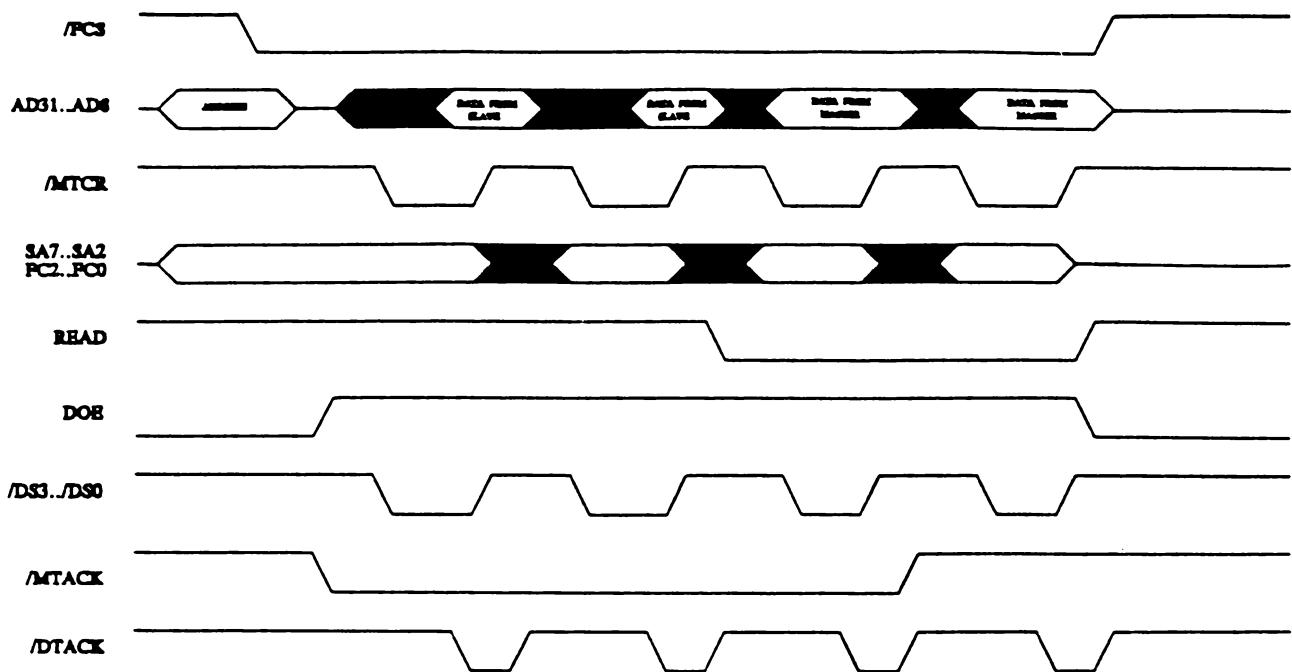


Figure 3-2: Multiple TransferCycles

The short cycle begins with the bus master driving the low order address lines A₇-A₂; these are the non-multiplexed addresses and can change without a new *address phase* being required (this is essentially a page mode, fully random accesses on this 256 byte page). The READ line may also change at this time. The master will then assert /MTCR to indicate to the slave that the short cycle is starting. For reads, the appropriate /DSN are asserted simultaneously with /MTCR, for writes, data and /DSN are asserted slightly after /MTCR. The slave will supply data for reads, then assert /DTACK, and the bus will terminate the short cycle and start into either another short cycle or a full cycle, depending on the multiple cycle handshaking that has taken place.

The question of whether a subsequent cycle will be a full cycle or a short cycle is answered by multiple cycle arbitration. If the master can't sustain another short cycle, it will negate /FCS and DOE along with /MTCR at the end of the current short cycle, terminating the full cycle as well. The master always samples the state of /MTACK on the falling edge of /MTCR. If a slave can't support additional short cycles, it negates /MTACK one short cycle ahead of time. On the following short cycle, the bus master will see that no more short cycles can be handled by the slave, and fully terminate the multiple transfer cycle once this last short cycle is done.

PICs aren't absolutely required to support Multiple Transfer Cycles, though it is a highly recommended feature, especially for memory boards. And of course, all PICs must act intelligently about such cycles on the bus; a card doesn't request or acknowledge any Multiple Transfer Cycle it can't support.

3.4 Quick Bus Arbitration

The Zorro II bus does an adequate job of supporting multiple bus masters, and the Zorro III bus extends this somewhat by introducing fair arbitration to Zorro II cards. However, some desirable features cannot be added directly to the Zorro II arbitration protocol. Specifically, Zorro III bus arbitration is much faster than the Zorro II style, it prohibits bus hogging that's possible under the Zorro II protocol, and it supports intelligent bus load balancing.

Load balancing requires a bit of explanation. A good analogy is to that of software multitasking; there, an operating system attempts to slice up CPU time between all tasks that need such time; here, a bus controller attempts to slice up bus time between all masters that need such time. With preemptive multitasking such as in the Amiga and UNIX OSs, equal CPU time can be granted to every task (possibly modified by priority levels), and such scheduling is completely under control of the OS; no task can hog the CPU time at the expense of all others. An alternate multitasking scheme is a popular add-on to some originally non-multitasking operating systems lately. In this scheme, each task has the CPU until it decides to give up the CPU, basically making the effectiveness of the CPU sharing at the mercy of each task. This is exactly the same situation with masters on the Zorro II bus. The Zorro III arbitration mechanism attempts to make bus scheduling under the control of the bus controller, with masters each being scheduled on a cycle-by-cycle basis.

When a Zorro III PIC wants to master the bus, it *registers* with the bus controller. This tells the bus controller to include that PIC in its scheduling of the expansion bus. There may be any number of other PICs registered with the bus controller at any given time. The CPU is always scheduled expansion bus time, and other local bus devices, such as a hard disk controller, may be registered from time to time.

Once registered, a PIC sits idle until it receives a *grant* from the bus controller. A grant is permission from the bus controller that allows the PIC to master the Zorro III bus for one full cycle. A PIC always gets one full cycle of bus time when given a grant, and assuming it stays registered, it may receive additional full cycles. Within the full cycle, the PIC may run any number of Multiple Transfer Cycles, assuming of course the responding slave supports such cycles. For multiprocessor support, a PIC will be granted multiple atomic full cycles if it locks the bus. This feature is *only* for support of hardware semaphores and other such multiprocessor needs; it is not intended as a means of bus hogging!

Figure 3-3 shows the basics of Zorro III bus arbitration, which is pretty simple. While it uses some of the same signals as the 680x0 inspired Zorro II bus arbitration mechanism, it has nothing to do with 680x0 bus arbitration; the /BRN and /BGN signals should be thought of as completely new signals. In order to register with the bus controller as a bus master, a PIC asserts its private /BRN strobe on the rising edge of the 7M clock, and negates it on the next rising edge. The bus controller will indicate mastership to a registered bus master by asserting its /BGN. Once granted the bus, the PIC drives only the standard cycle signals: addresses, /FCS, /EDSN, data, etc. in a full cycle. The bus controller manages the assertion of /OWN and /BGACK, which are important only for bus management and Zorro II support. While a scheduling scheme

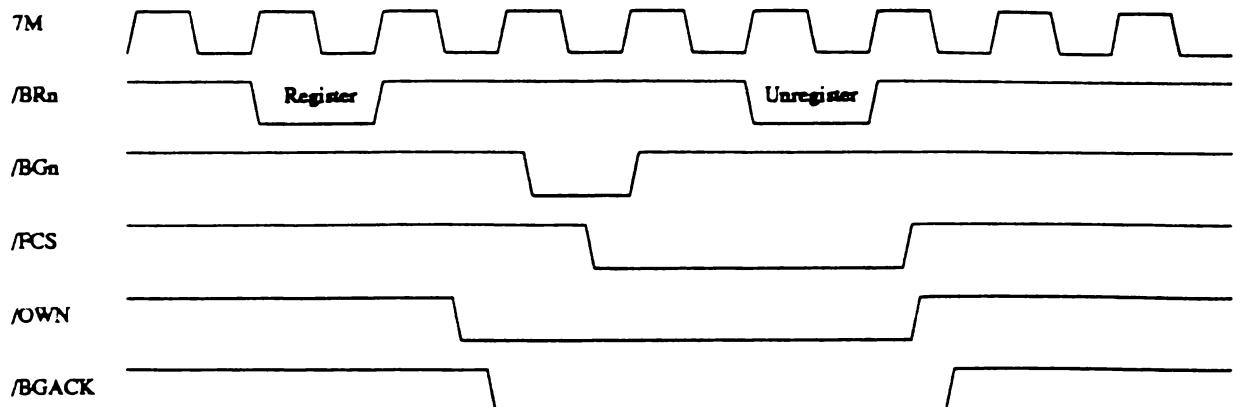


Figure 3-3: Zorro III Bus Arbitration

isn't part of this bus specification, the bus master will only be guaranteed one bus cycle at a time. The /BGn line is negated shortly after the master asserts /FCS unless the bus controller is planning to grant multiple full cycles to the master. The only thing that'll force the controller to grant multiple full cycles is a locked bus. Any master that works better with multiple cycles, such as devices with buffers to empty into memory, should run a Multiple Transfer Cycle to transfer several longwords during the same full cycle. For this reason, slave cards are encouraged to support Multiple Transfer Cycles, even if they don't necessarily run any faster during them.

Once a registered bus master has no more work to do, it unregisters with the bus controller. This works just like registering -- the PIC asserts /BRN on the rise of 7M, then negates it on next rising 7M. This is best done during the last cycle the bus master requires on the bus. If a registered master gets a grant before unregistering and has no work to do, it can unregister without asserting /FCS, to give back the bus without running a cycle. It's always far better to make sure that the master unregisters as quickly as possible. Bus timeout causes an automatic unregistering of the registered master that was granted that timed-out cycle; this guarantees that an inactive registered master can't drag down the system. If a master sees a /BERR during a cycle, it should terminate that cycle immediately and re-try the same cycle. If the retried cycle results in a /BERR as well, nothing more can be done in hardware; notification of the driver program is the usual recourse.

The bus controller may have to mix Zorro II style bus arbitration in with Zorro III arbitration, as Zorro II and Zorro III cards can be freely mixed in a backplane. Because of this, Multiple Transfer Cycles, and the self-timed nature of Zorro III cards, there's no way to guarantee the latency between bus grants for a Zorro III card. The bus controller does, however, make sure that all masters are fairly scheduled so that no starvation occurs, if at all possible. Zorro III cards must use Zorro III style bus arbitration; although current Zorro III backplanes can't differentiate between Zorro II and Zorro III cards when they request (other than by the request mechanism), it can't be assumed that a backplane will support Zorro III cycles with Zorro II mastering, or vice-versa.

3.5 Quick Interrupts

While the Zorro II bus has always supported shared interrupts, the Zorro III bus supports a mechanism wherein the interrupting PIC can supply its own vector. This has the potential to make such vectored interrupts much faster than conventional Zorro II chained interrupts, arbitrating the interrupting device in hardware instead of software.

A PIC supporting quick interrupts has on-board registers to store one or more vector numbers; the numbers are obtained from the OS by the device driver for the PIC, and the PIC/driver combination must be able to handle the situation in which no additional vectors are available. During system operation, this PIC will interrupt the system in the normal manner, by

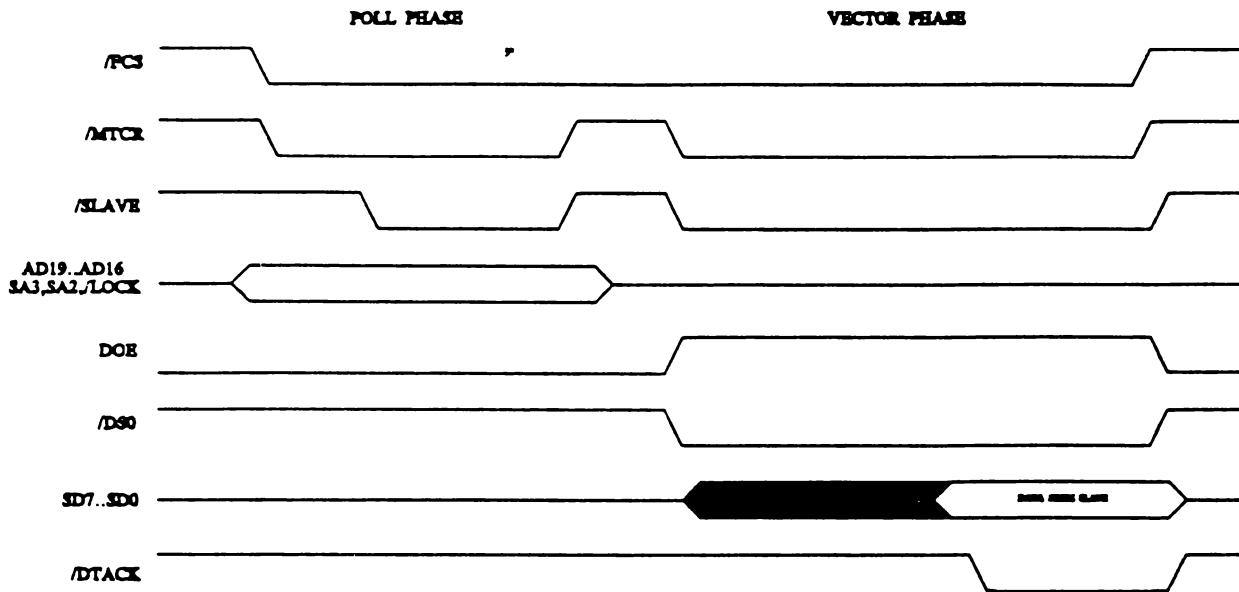


Figure 3-4: Interrupt Vector Cycle

asserting one of the bus interrupt lines. This interrupt will cause an interrupt vector cycle to take place on the bus. This cycle arbitrates in hardware between all PICs asserting that interrupt, and it's a completely different type of Zorro III cycle, as illustrated in *Figure 3-4*.

The bus controller will start an interrupt vector cycle in response to an interrupt asserted by any PIC. This cycle starts with /FCS and /MTCR asserted, a FC code of 7 (CPU space), a CPU space cycle type, given by address lines A16-A19, of 15, and the interrupt number, which is on A1-A3 (A1 is on the /LOCK line, as in Zorro II cycles). The interrupt numbers 2 and 6 are currently defined, corresponding to /INT2 and /INT6 respectively; all others are reserved for future use. At this point, called the *polling phase*, any PIC that has asserted an interrupt and wants to supply a vector will decode the FC lines, the cycle type, match its interrupt number against the one on the bus, and assert /SLAVEn if a match occurs. Shortly thereafter, the /MTCR line is negated, and the slaves all negate /SLAVEn. But the cycle doesn't end.

The next step is called the *vector phase*. The bus controller asserts one /SLAVEn back to one of the interrupting PICs, along with /MTCR and /DS0, but no addresses are supplied. That

PIC will then assert its 8-bit vector onto the logical D0-D7 (physically AD₁₅-AD₈) of the 32-bit data bus and /DTACK, as quickly as possible, thus terminating the cycle. The speed here is very critical; an automatic autovector timeout will occur very quickly, as any actual waiting that's required for the quick interrupt vector is potentially delaying the autovector response for Zorro II style interrupts. A PIC stops driving its interrupt when it gets the response cycle; it must also be possible for this interrupt to be cleared in software (e.g., the PIC must make choice of vectoring vs. autovectoring a software issue).

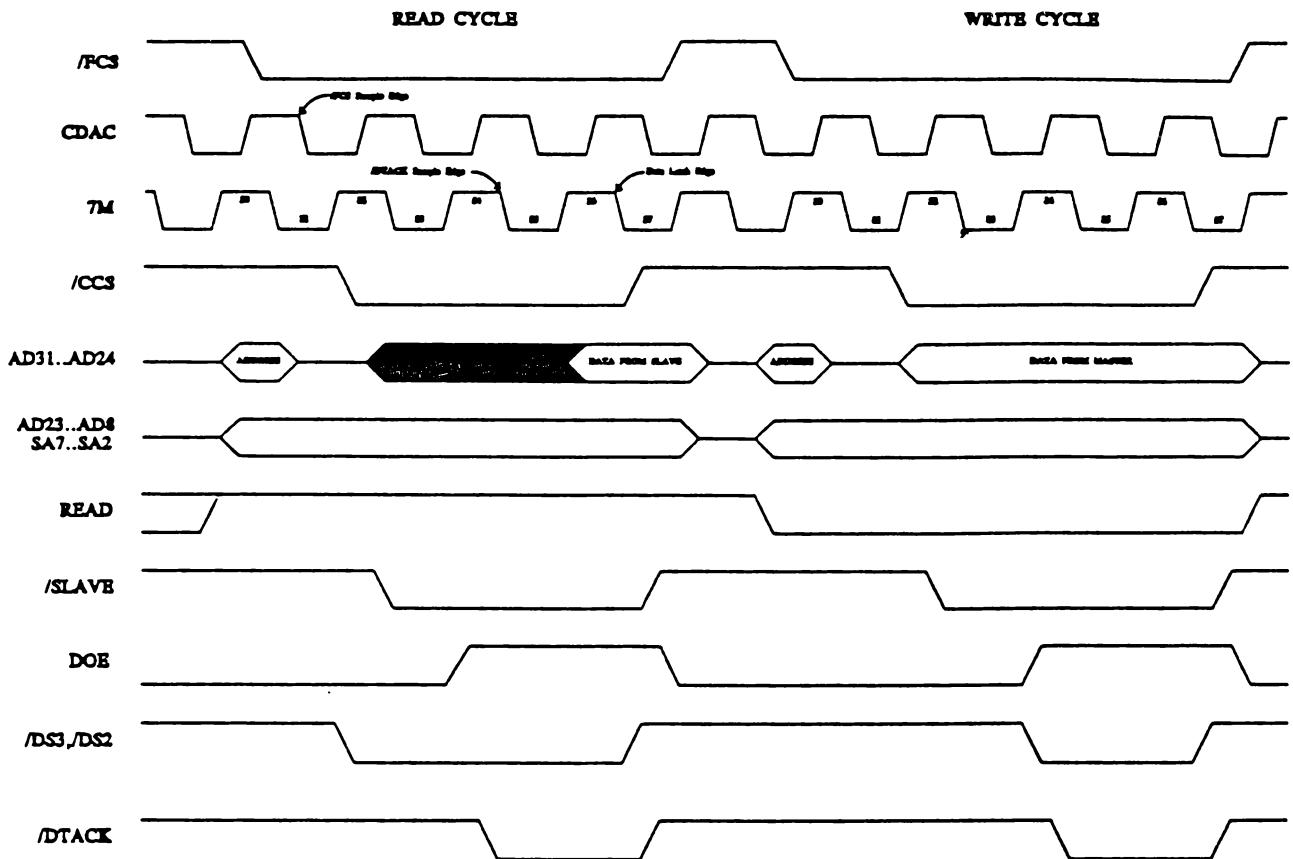


Figure 3-5: Zorro II Within Zorro III

3.6 Compatibility with Zorro II Devices

As detailed in Chapter 2, the Zorro III bus supports a bus cycle mode very similar to the 68000-based Zorro II bus, and is expected to be compatible with all properly designed Zorro II PICs. As shown in *Figure 1-1*, Zorro II and Zorro III expansion spaces are geographically mapped on the Zorro III bus. The mapping logic resides on the bus, and operates on the bus address presented for any cycle. Every cycle starts out assuming a Zorro III cycle, but the mapping logic will inscribe a Zorro II cycle within the Zorro III cycle if the address range is right. *Figure 3-5* details the bus action for this mode.

The cycle starts out with the usual address phase activity; the bus master asserts /FCS after asserting the full 32-bit address onto the address bus. The bus decoder maps the bus

address asynchronously and quickly, so that by the time /FCS is asserted, the memory space is determined. A Zorro II space access will cause A₈-A₂₃ to remain asserted, rather than being tri-stated along with A₂₄-A₃₁, as the Zorro III cycle normally does. The bus controller synchs the asynchronous /FCS on the falling edge of CDAC, then drives /CCS (the /AS equivalent) out on the rising edge of 7M, based on that synched /FCS. For a read cycle, /DS₃ and/or /DS₂ (the /UDS and /LDS replacements, respectively) would be asserted along with /CCS; write cycles see those lines asserted on the next rising edge of 7M, at S₄ time. The DOE line is also asserted at the start of S₄.

The bus controller starts to sample /DTACK on the falling edge of 7M between S₄ and S₅, adding wait states until /DTACK is encountered. As per Zorro II specs, the PIC need not create a /DTACK unless it needs that level of control; there are Zorro II signals to delay the controller-generated /DTACK, or take it over when necessary. The controller will drive its automatic /DTACK at the start of S₄, leaving plenty of time for the sampling to come at S₅. Once a /DTACK is encountered, cycle termination begins. The controller latches data on the falling 7M edge between S₆ and S₇, and also negates /CCS and the /DS_N at this time. Shortly thereafter, the controller negates /DTACK (when controlling it), DOE, and tri-states the data bus, getting ready for the next cycle.

3.7 Zorro III Implementations

Functionally, there are two possible implementation levels in existence for the Zorro III bus. All of the features described in this document are required for a full compliance Zorro III bus. However, the original Amiga 3000 computers were shipped with a bus controller that supported only a subset of the Zorro III specification published here. This is, however, upgradable.

The A3000 implementation of the Zorro III bus is driven by a custom controller chip called Fat Buster. The specification of this chip and the A3000 hardware are fully capable of supporting the complete Zorro III bus, but the initial silicon on Fat Buster, called the Level 1 Fat Buster, omits some features. Missing are:

- Support of Multiple Transfer Cycles.
- Support for Zorro III style bus arbitration.
- Support for Quick Interrupts.

The Level 2 version of Fat Buster has been in testing for some time at Commodore in West Chester, PA. Any developers who immediately intend to design PICs supporting these features are urged to contact Commodore Amiga Technical Support/Amiga Developer Support Europe for more information on obtaining samples of this part for use in A3000 systems. These parts are likely to be introduced into production, and available as part of an A3000 upgrade, very soon. All Buster chip revisions "13G" and earlier support the Level 1 features. Buster chip revisions "13H" and later support Level 2 features and improved Level 1 features as well.

CHAPTER 4

SIGNAL DESCRIPTION

*"Pushing back the limits of human achievement, reaching for the stars,
that's not something we do. It's what we are."*

-Michael Swaine

The signals detailed here are the Zorro III mode signals. While some of this information is the same in as the Zorro II signal description of Chapter 2, many like-seeming bus signals behave differently in Zorro III mode than Zorro II mode. These can be a very important differences; thus the complete set of signals is detailed here.

4.1 Power Connections

The expansion bus provides several different voltages designed to supply expansion devices. These are basically the same for the Zorro III bus as they were for the Zorro II bus, with the exception of one pin, and that the specification has been clarified a bit. Note that all Zorro III PICs must list their power consumption specifications.

Digital Ground (Ground)

This is the digital supply ground used by all expansion cards as the return path for all expansion supplies.

Main Supply (+5VDC)

This is the main power supply for all expansion cards, and it is capable of sourcing large currents; each PIC can draw up to 2.0 Amps @ +5VDC.

Negative Supply (-5VDC)

This is a negative version of the main supply, for small current loads only; each PIC can draw up to 60 mA @ -5VDC.

High Voltage Supply (+12VDC)

This is a higher voltage supply, useful for communications cards and other devices requiring greater than digital voltage levels. This is intended for relatively small current loads only; each PIC can draw up to 500mA @ +12VDC.

Negative High Supply (-12VDC)

Negative version of the high voltage supply, also used in communications applications, and similarly intended for small loads only; each PIC can draw up to 60 mA @ -12VDC.

4.2 Clock Signals

The expansion bus provides clock signals for expansion boards. The main use for these clocks on Zorro III cards is bus arbitration clocking. There is no relationship between any of these clocks and normal Zorro III bus activity. The relationship between these clocks is illustrated in *Figure 2-2*.

/C1 Clock

This is a 3.58 MHz clock (3.55 MHz on PAL systems) that's synched to the falling edge of the 7M system clock.

/C3 Clock

This is a 3.58 MHz clock (3.55 MHz on PAL systems) that's synched to the rising edge of the 7M system clock.

CDAC Clock

This is a 7.16 MHz system clock (7.09 MHz on PAL systems) which trails the 7M clock by 90° (approximately 35ns).

E Clock

This is the 68000 generated "E" clock, used for 6800 family peripherals driven by "E" and 6502 peripherals driven by Φ_2 . This clock is four 7M clocks high, six clocks low, as per the 68000 spec.

7M Clock

This is the 7.16 MHz system clock (7.09 MHz on PAL systems). This clock drives the bus master registration mechanism for Zorro III bus masters.

4.3 System Control Signals

The signals in this group are available for various types of system control; most of these have an immediate or near immediate effect on expansion cards and/or the system CPU itself.

Hardware Bus Error/Interrupt (/BERR)

This is a general indicator of a bus fault or special condition of some kind. Any expansion card capable of detecting a hardware error relating directly to that card can assert /BERR when that bus error condition is detected, especially any sort of harmful hardware error condition. This signal is the strongest possible indicator of a bad situation, as it causes all PICs to get off the bus, and will usually generate a level 2 exception on the host CPU. For any condition that can be handled in software and doesn't pose an immediate threat to hardware, notification via a standard processor interrupt is the better choice. The bus controller will drive /BERR in the event of a detected bus collision or DMA error (an attempt by a bus master to access local bus resources it doesn't have valid access permission for). All cards must monitor /BERR and be prepared to tri-state all of their on-bus output buffers whenever this signal is asserted. An expansion bus master will attempt to retry a cycle aborted by a single /BERR and notify system software in the case of two subsequent /BERR results. Since any number of devices may assert /BERR, and all bus cards must monitor it, any device that drives /BERR must drive with an open collector or similar device, and any device that monitors /BERR should place a minimal load on it. This signal is pulled high by a passive backplane resistor.

Note that, especially for the slave device being addressed, that /BERR alone is not always necessarily an indication of a bus failure in the pure sense, but may indicate some other kind of unusual condition. Therefore, a device should still respond to the bus address, if otherwise appropriate, when a /BERR condition is indicated. It simply tri-states its bus buffers and other outputs, and waits for a change in the bus state. If the /BERR signal is negated with the cycle unterminated, the special condition has been resolved and the slave responds to the rest of the cycle as it normally would have. If the cycle is terminated by the bus master, the resolution of the special condition has indicated that the addressed slave is not needed, and so the cycle terminates without the slave being used.

System Reset (/RESET, /IORST)

The bus supplies two versions of the system reset signal. The /RESET signal is bidirectional and unbuffered, allowing an expansion card to hard reset the system. It should only be used by boards that need this reset capability, and is driven only by an open collector or similar device. The /IORST signal is a buffered output-only version of the reset signal that should be used as the normal reset input to boards not concerned with resetting the system on their own. All expansion devices are required to reset their autoconfiguration logic when /IORST is asserted. These signals are pulled high by passive backplane resistors.

System Halt (/HLT)

This signal is driven, along with /RESET, to assert a full-system reset. A full-system reset is asserted on a powerup reset or a keyboard reset; any PIC that needs to differentiate between full system and I/O reset should monitor /HLT and /IORST unless it also needs to drive a reset condition. This is driven with an open-collector output, or the equivalent, and pulled up by a backplane resistor.

System Interrupts

Two of the decoded, level sensitive 680x0 interrupt inputs are available on the expansion bus, and these are labelled as /INT₂ and /INT₆. Each of these interrupt lines is shared by wired ORing, thus each line must be driven by an open-collector or equivalent output type. Zorro III interrupts can be handled Zorro II style, via autovectors and daisy-chained polling, or they can be vectored using the quick interrupt protocol described in Chapter 3. Zorro II and Zorro III systems originally provided /INT₁, /INT₄, /INT₅, and /INT₇ lines as well, but as these were never properly supportable by system software, they have been eliminated, those lines now considered reserved for future use in a Zorro III system.

4.4 Slot Control Signals

This group of signals is responsible for the control of things that happen between expansion slots.

Slave (/SLAVEn)

Each slot has its own /SLAVEn output, driven actively, all of which go into the collision detect circuitry. The "n" refers to the expansion slot number of the particular /SLAVE signal. Whenever a Zorro III PIC is responding to an address on the bus, it must assert its /SLAVEn output very quickly. If more than one /SLAVEn output occurs for the same address, or if a PIC asserts its /SLAVEn output for an address reserved by the local bus, a collision is registered and the bus controller asserts /BERR. The bus controller will assert /SLAVEn back to the interrupting device selected during a Quick Interrupt cycle, so any device supporting Quick Interrupts must be capable of tri-stating its /SLAVEn; all others can drive SLAVEn with a normal active output.

Configuration Chain (/CFGINN, /CFGOUTN)

The slot configuration mechanism uses the bus signals /CFGOUTN and /CFGINN, where "n" refers to the slot number. Each slot has its own version of both signals, which make up the *configuration chain* between slots. Each subsequent /CFGINN is a result of all previous /CFGOUTs, going from slot 0 to the last slot on the expansion bus. During the autoconfiguration process, an unconfigured Zorro III PIC responds to the 64K address space starting at either \$00E80000 or \$FF000000 if its /CFGINN signal is asserted. All unconfigured PICs start up with /CFGOUTN negated. When configured, or told to "shut up", a PIC will assert its /CFGOUTN, which results in the /CFGINN of the next slot being asserted. Backplane logic automatically passes on the state of the previous /CFGOUTN to the next /CFGINN for any slot not occupied by a PIC, so there's no need to sequentially populate the expansion bus slots.

Backplane Type Sense (SenseZ3)

This line can be used by the PIC to determine the backplane type. It is grounded on a Zorro II backplane, but floating on a Zorro III backplane. The Zorro III PIC connects this signal to a 1K pullup resistor to generate a real logic level for this line. It's possible, though more complicated, to build a Zorro III PIC that can actually run in Zorro II mode when in a Zorro II backplane. It's hardly necessary or required to support this backward

Multiplexed Address Bus (A8-A31)

These signals are driven by the bus master during address time, prior to the assertion of /FCS. Any responding slave must latch as many of these lines as it needs on the falling edge of /FCS, as they're tri-stated very shortly after /FCS goes low. These addresses always include all configuration address bits for normal cycles, and the cycle type information for Quick Interrupt cycles.

Short Address Bus (A2-A7)

These signals are driven by the bus master during address time, prior to the assertion of /FCS, for full cycles, and prior to the assertion of /MTCR for short cycles. They stay valid for the entire full or short cycle, and as such do not need to be latched by responding slaves.

Table 4-1: Memory Space Type Codes

FC0	FC1	FC2	Address Space Type	Z3 Response
0	0	0	Reserved	None
0	0	1	User Data Space	Memory
0	1	0	User Program Space	Memory
0	1	1	Reserved	None
1	0	0	Reserved	None
1	0	1	Supervisor Data Space	Memory
1	1	0	Supervisor Program Space	Memory
1	1	1	CPU Space	Interrupts

Memory Space (FC0-FC2)

The memory space bits are an extension to the bus address, indicating which type of access is taking place. Zorro III PICs must pay close attention to valid memory space types, as the space type can change the type of the cycle driven by the current bus master. The encoding is the same as the valid Motorola function codes for normal accesses. These are driven at address time, and like the low short address, are valid for an entire short or full cycle.

Compatibility Cycle Strobe (/CCS)

This is equivalent to the Zorro II address strobe, /AS. A Zorro III PIC doesn't use this for normal operation, but may use it during the autoconfiguration process if configuring at the Zorro II address. AUTOCONFIG® cycles at \$00E8xxxx always look like Zorro II cycles, though of course /FCS and the full Zorro III address is available, so a card can use either Zorro II or Zorro III addressing to start the cycle. However, using the /CCS strobe can save the designer the need to compare the upper 8 bits of address. Data must be driven Zorro II style, though if the /DSN lines are respected for reads, /CINH is asserted, and /MTACK is negated, the resulting Zorro III cycle will fit within the expected Zorro II cycle generated by the bus controller.

compatibility mechanism, and in many cases it'll be impractical. The Zorro III specification does require that this signal be used, at least, to shut the card down and pass /CFGIN to /CFGOUT when in a Zorro II backplane.

4.5 DMA Control Signals

There are various signals on the expansion bus that coordinate the arbitration of bus masters. Zorro II bus masters use some of the same logical signals, but their arbitration protocol is considerably different.

PIC is DMA Owner (/OWN)

This is asserted by the bus controller when a master is about to go on the bus and indicates that some master owns the bus. Zorro II bus masters drive this, and some Zorro III slaves may find a need to monitor it, or /BGACK, to determine who's the bus master. This is ordinarily not important to Zorro III PICs, and they may not drive this line.

Slot Specific Bus Arbitration (/BR_N, /BG_N)

These are the slot-specific /BR_N and /BG_N signals, where "N" refers to the expansion slot number. The bus request from each board is taken in by the bus controller and ultimately used to take over the system from the primary bus master, which is always the local master. Zorro III PICs toggle /BR_N to register or unregister as a master with the bus controller. /BG_N is asserted to one registered PIC at a time, on a cycle by cycle basis, to indicate to the PIC that it gets the bus for one full cycle.

Bus Grant Acknowledge (/BGACK)

Asserted by the bus controller when a master is about to go on the bus. As with /OWN, most Zorro III PICs ignore this signal, and none may drive it.

Bus Want/Clear (/BCLR)

This signal is asserted by the bus controller to indicate that a PIC wants to master the bus; Zorro III cards can use this to determine if any Zorro II bus requests are pending; Zorro III bus requests don't affect /BCLR.

4.6 Address and Related Control Signals

These signals are various items used for the addressing of devices in Zorro III mode by bus masters either on the bus or from the local bus. The bus controller translates local bus signals (68030 protocol on the A3000 and A4000) into Zorro III signals; masters are responsible for creating the appropriate signals via their own bus control logic.

Read Enable (READ)

Read enable for the bus; READ is asserted by the bus master during a bus cycle to indicate a read cycle, READ is negated to indicate a write cycle. READ is asserted at address time, prior to /FCS, for a full cycle, and prior to /MTCR for a short cycle. READ stays valid throughout the cycle; no latching required.

Yes, that should sound weird; it's based on the mapping of Zorro II vs. Zorro III signals, and of course the fact that /FCS always starts any cycle. Also note that a bus cycle with /CCS asserted and /FCS negated is always a Zorro II PIC-as-master cycle. Many Zorro III cards will instead configure at the alternate \$FF00xxxx base address, fully in Zorro III mode, and thus completely ignore this signal.

Full Cycle Strobe (/FCS)

This is the standard Zorro III full cycle strobe. This is asserted by the bus master shortly after addresses are valid on the bus, and signals the start of any kind of Zorro III bus cycle. Shortly after this line is asserted, all the multiplexed addresses will go invalid, so in general, all slaves latch the bus address on the falling edge of /FCS. Also, /BGN line is negated for a Zorro III mastered cycle shortly after /FCS is asserted by the master.

4.7 Data and Related Control Signals

The data time signals here manage the actual transfer of data between master and slave for both full and short cycle types. The burst mode signals are here too, as they're basically data phase signals even though they don't only concern the transfer of data.

Data Output Enable (DOE)

This signal is used by an expansion card to enable the buffers on the data bus. The bus master drives this line to keep slave PICs from driving data on the bus until *data time*.

Data Bus (Do-D₃₁)

This is the Zorro III data bus, which is driven by either the master or the slave when DOE is asserted by the master (based on READ). It's valid for reads when /DTACK is asserted by the slave; on writes when at least one of /DS_N is asserted by the master, for all cycle types.

Data Strobes (/DS_N)

These strobes fall during *data time*; /DS₃ strobos D₂₄-D₃₁, while /DS₀ strobos Do-D₇. For write cycles, these lines signal data valid on the bus. At all times, they indicate which bytes in the 32 bit data word the bus master is actually interested in. For cachable reads, all four bytes must be returned, regardless of the value of the sizing strobes. For writes, only those bytes corresponding to asserted /DS_N are written. Only contiguous byte cycles are supported; e.g. /DS₃₋₀ = 2, 4, 5, 6, or 10 is invalid.

Data Transfer Acknowledge (/DTACK)

This signal is used to normally terminate a Zorro III cycle. The slave is always responsible for driving this signal. For a read cycle, it asserts /DTACK as soon as it has driven valid data onto the data bus. For a write cycle, it asserts /DTACK as soon as it's done with the data. Latching the data on writes may be a good idea; that can allow a slave to end the cycle before it has actually finished writing the data to its local memory.

Cache Inhibit (/CINH)

This line is asserted at the same time as /SLAVEn to indicate to the bus master that the cycle must not be cached. If a device doesn't support caching, it must assert /CINH and actually obey the /DSn byte strobes for read cycles. Conversely, if the device supports caching, /CINH is negated and the device returns all four bytes valid on reads, regardless of the actual supplied /DSn strobes.

Multiple Cycle Transfers (/MTCR,/MTACK)

These lines comprise the Multiple Transfer Cycle handshake signals. The bus master asserts /MTCR at the start of *data time* if it's capable of supporting Multiple Transfer Cycles, and the slave asserts /MTACK with /SLAVEn if it's capable of supporting Multiple Transfer Cycles. If the handshake goes through, /MTCR strobes in the short address and write data as long as the full cycle continues.

CHAPTER 5

TIMING

*"When dealing with the insane, the best method is
to pretend to be sane."*

-Hermann Hesse

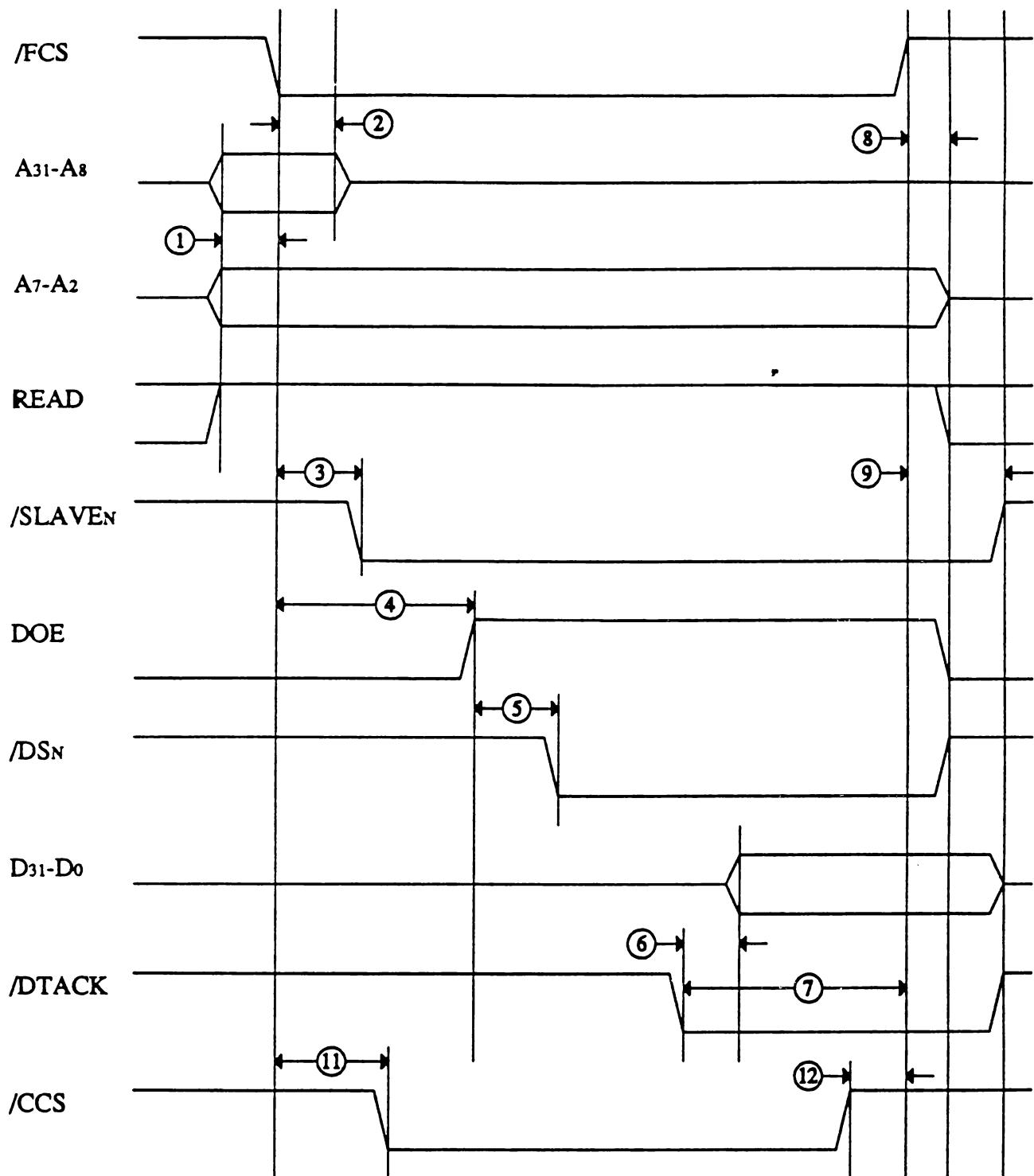
Some of this information is considered preliminary. Nothing is expected to get any more speed critical, but as mentioned previously, the testing of Zorro III designs has just started at the time of this writing, final bus controllers are not yet available, and only a few PIC designs have even been conceived.

This section covers the various timing specifications in detail for different Zorro III operations. It's important to realize that this timing information is a specification. Actual Zorro III systems may offer much more relaxed timings. Today. The whole point of the specification is that as long as all Zorro III PICs and all Zorro III backplanes base things on the timings given here, they'll always work together nicely. Any design based on the actual characteristics of any particular backplane will very likely wind up working only on that particular backplane.

The philosophy of timing on the Zorro III bus is to keep things as simple as possible without compromising the performance goals of the bus. Zorro III PICs are expected to be based on F-Series or ACT-series TTL logic, fast PALs, and possibly full custom chip designs. It's very unlikely the designer will meet any of these specifications with the LS parts left over from old Zorro II card designs.

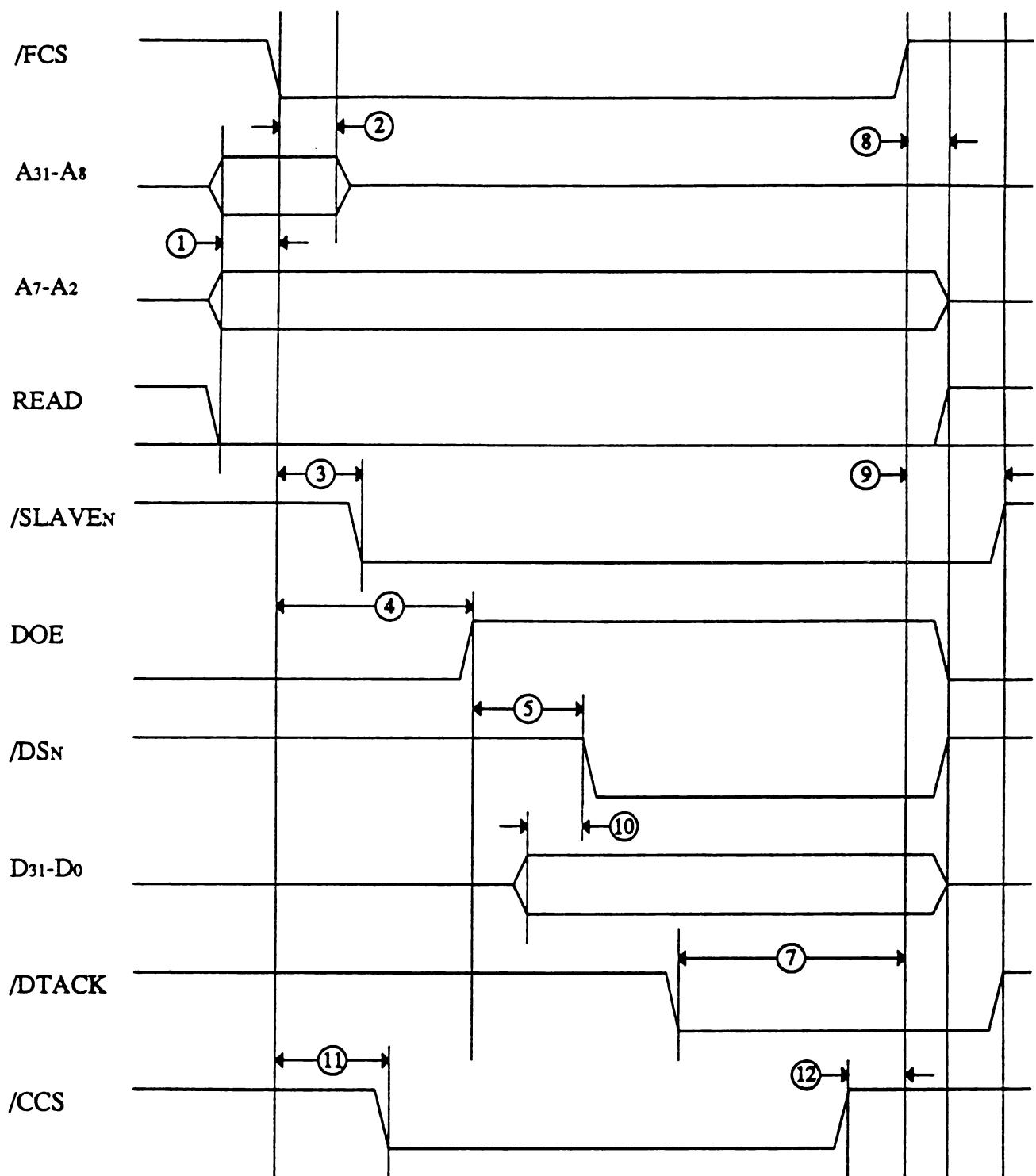
5.1 Standard Read Cycle Timing

No.	Name	Symbol	Min	Max
1	Address setup to /FCS	T _{AFS}	15ns	----
2	Address hold from /FCS	T _{HAF}	10ns	----
3	/FCS to /SLAVE _N delay	T _{SLV}	----	25ns
4	/FCS to DOE delay	T _{DOE}	30ns	----
5	DOE to /DS _N delay	T _{DS}	10ns	----
6	Data setup to /DTACK	T _{RD}	0ns	----
7	/DTACK to /FCS off	T _{OFF}	10ns	----
8	Master signal hold from /FCS off	T _{HMC}	0ns	5ns
9	Slave signal hold from /FCS off	T _{HSC}	0ns	15ns
11	/FCS to /CCS delay	T _{CCS}	35ns	175ns
12	/CCS off to /FCS off	T _{OVL}	40ns	----



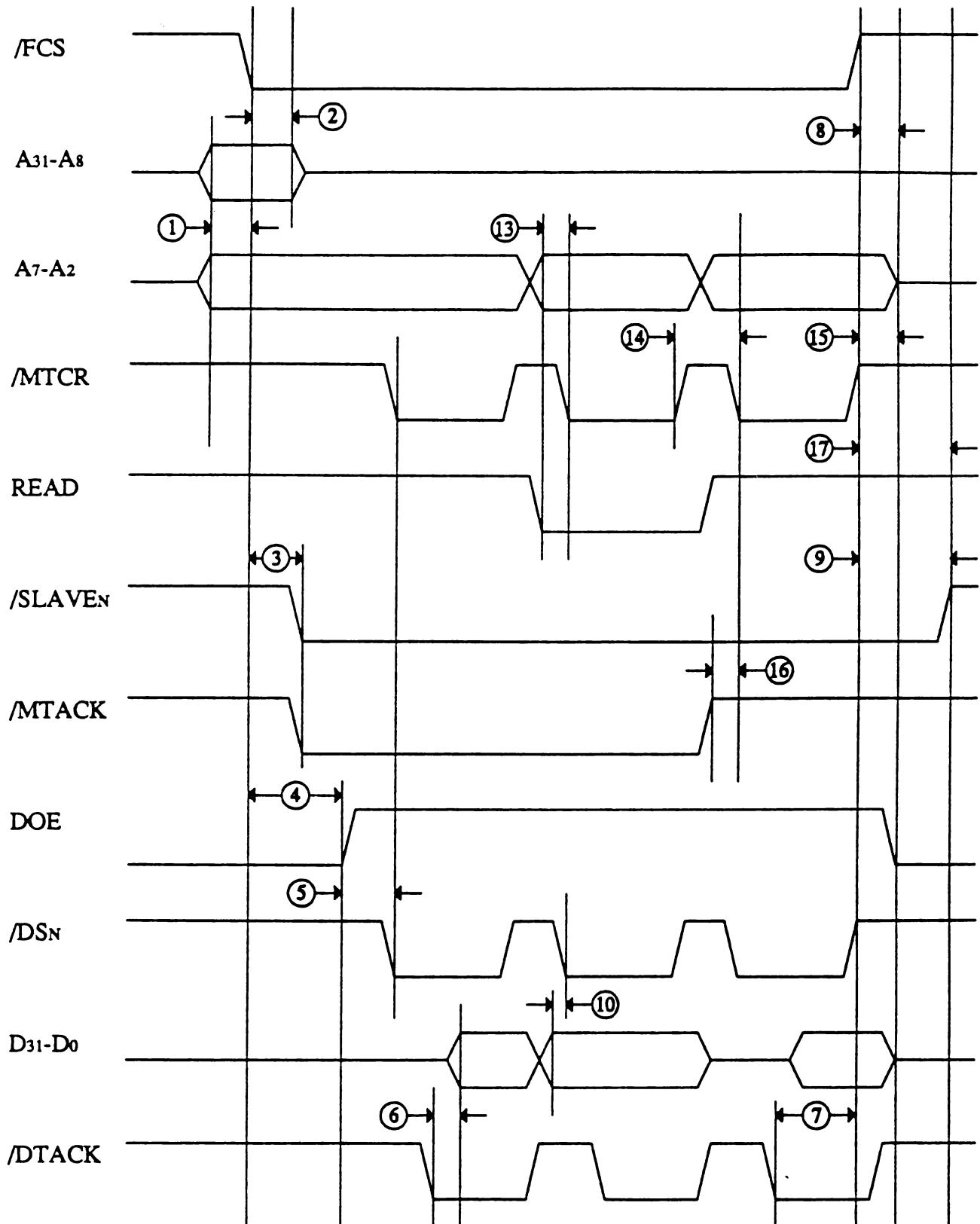
5.2 Standard Write Cycle Timing

No.	Name	Symbol	Min	Max
1	Address setup to /FCS	T _{AFS}	15ns	-----
2	Address hold from /FCS	T _{HAF}	10ns	-----
3	/FCS to /SLAVE _N delay	T _{SLV}	-----	25ns
4	/FCS to DOE delay	T _{DOE}	30ns	-----
5	DOE to /DS _N delay	T _{DS}	10ns	-----
7	/DTACK to /FCS off	T _{OFF}	10ns	-----
8	Master signal hold from /FCS off	T _{HMC}	0ns	5ns
9	Slave signal hold from /FCS off	T _{HSC}	0ns	15ns
10	Write data setup to /DS _N	T _{WDS}	5ns	-----
11	/FCS to /CCS delay	T _{CCS}	35ns	175ns
12	/CCS off to /FCS off	T _{OVL}	40ns	-----



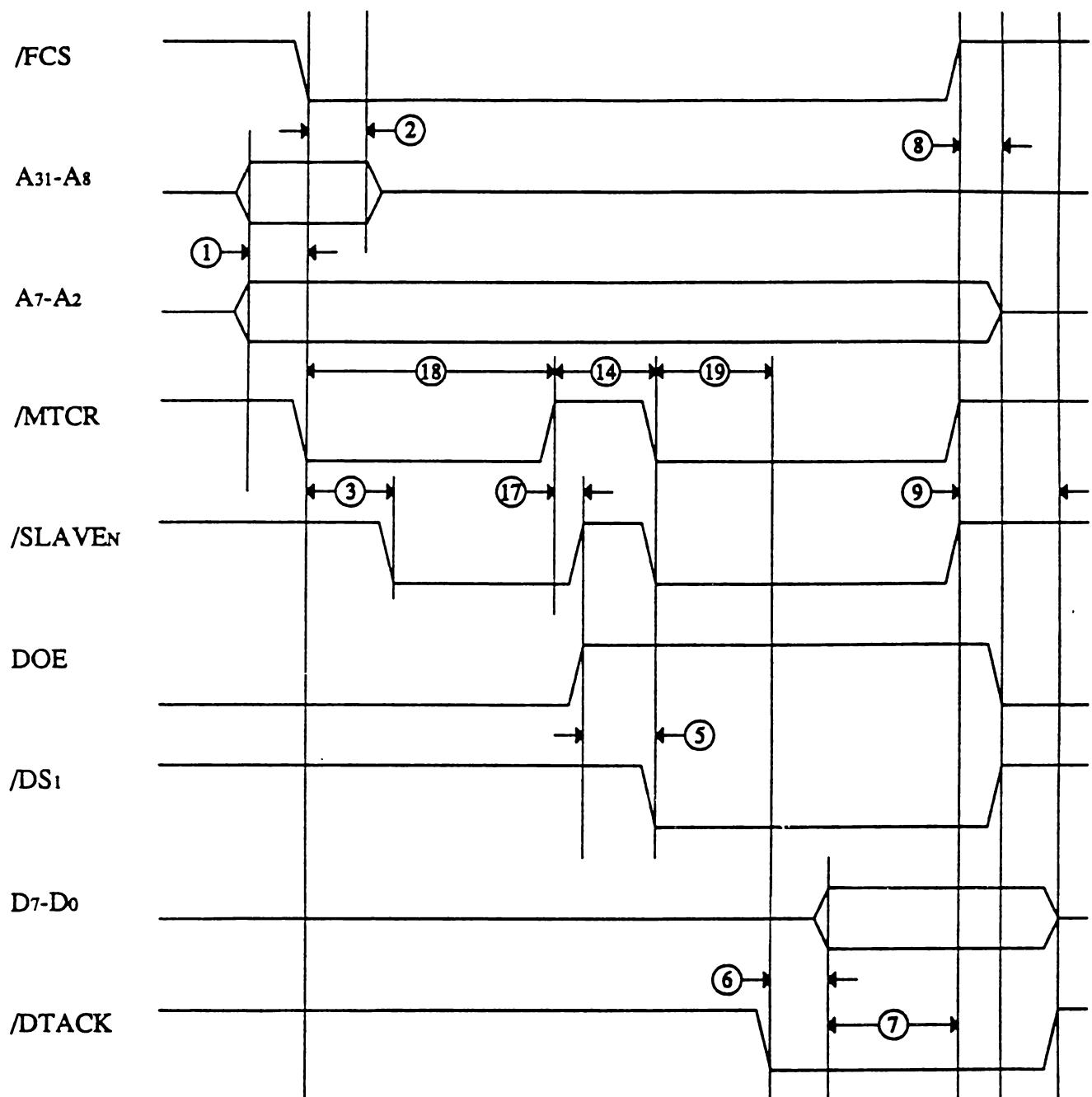
5.3 Multiple Transfer Cycle Timing

No.	Name	Symbol	Min	Max
1	Address setup to /FCS	T _{APS}	15ns	----
2	Address hold from /FCS	T _{HAF}	10ns	----
3	/FCS to /SLAVEn, /MTACK delay	T _{SLV}	----	25ns
4	/FCS to DOE delay	T _{DOE}	30ns	----
5	DOE to /DSN, /MTCR delay	T _{DS}	10ns	----
6	Data setup to /DTACK	T _{RDS}	0ns	----
7	/DTACK to /FCS, /MTCR off	T _{OFF}	10ns	----
8	Master signal hold from /FCS off	T _{HMC}	0ns	5ns
9	Slave signal hold from /FCS off	T _{HSC}	0ns	15ns
10	Write data setup to /DSN	T _{WDS}	5ns	----
13	Address, READ setup to /MTCR	T _{AMS}	5ns	----
14	/MTCR off to /MTCR on	T _{REF}	10ns	----
15	Address, READ hold from /MTCR	T _{HAM}	0ns	----
16	/MTACK off to /MTCR	T _{BCD}	10ns	----
17	Slave signal hold from /MTCR off	T _{HSM}	0ns	5ns



5.4 Quick Interrupt Cycle Timing

No.	Name	Symbol	Min	Max
1	Address setup to /FCS	T _{AFS}	15ns	-----
2	Address hold from /FCS	T _{HAF}	10ns	-----
3	/FCS to /SLAVEn delay	T _{SLV}	-----	25ns
5	DOE to /DS _N delay	T _{DS}	10ns	-----
6	, Data setup to /DTACK	T _{RD}	0ns	-----
7	/DTACK to /FCS off	T _{OFF}	10ns	-----
8	Master signal hold from /FCS off	T _{HMC}	0ns	5ns
9	Slave signal hold from /FCS off	T _{HSC}	0ns	15ns
14	/MTCR off to /MTCR on	T _{REF}	10ns	-----
17	Slave signal hold from /MTCR off	T _{HSM}	0ns	5ns
18	Poll Phase time	T _{POL}	30ns	100ns
19	Vector Phase start to /DTACK time	T _{VEC}	-----	100ns



CHAPTER 6

ELECTRICAL SPECIFICATIONS

"...I collected the instruments of life around me, that I might infuse a spark of being into the lifeless thing that lay at my feet"

-Victor Frankenstein

The Zorro III bus has a number of electrical specifications that are very important for PIC designers to consider, along with the timing parameters of course. It's extremely important to base designs on the specification of the backplane, rather than the actual behavior of the backplane. New backplanes for new machines are designed to conform to the specification, they are not necessarily based on previous designs. This is especially important with the Zorro III bus, since timing is far more critical than in the past, and the bus controller is designed from this specification, rather than the reverse, as in the Amiga 2000.

6.1 Expansion Bus Loading

The Zorro III bus loading is specified based on typical TTL family "F" series buffer devices, though in reality, compatible CMOS devices are likely to be used in some bus controllers or PICs. Thus, it's important to accept the TTL levels as a minimum voltage level, and make sure that all inputs are the appropriate TTL levels, while outputs can be at TTL or CMOS voltage levels as long as they provide the required source and sink.

While some A2000 designs used "LS" or "ALS" buffers instead of "F", the bus will generally work with these older cards, at least with current backplane designs such as the A3000 backplane. However, Zorro III designs must exactly obey these loading rules; it's very probable that some future Zorro III machines will have a large number of slots. In such machines, PICs built on the Zorro II specification will still work in a lightly loaded bus, but may not function in a fully loaded bus. All Zorro III PICs built to spec will work in any Zorro III backplane, without any loading problems, if all loading and timing rules are followed by the PIC designer. The bus

Table 6-1: Zorro III Drive Types

Signal	Direction	High Level	Low Level
Standard	Loading Driven	+140µA @ +2.7VDC +2.5VDC @ -3.0mA	-3.2mA @ +0.4VDC +0.4VDC @ +64mA
Clock	Loading	+20µA @ +2.7VDC	-1.6mA @ +0.4VDC
O.C.	Loading Driven	+80µA @ +2.7VDC Not Driven	-3.2mA @ +0.4VDC +0.4VDC @ +20mA
Non-bussed	Loading Driven	+80µA @ +2.7VDC +2.5VDC @ -0.4mA	-1.0mA @ +0.4VDC +0.4VDC @ +4.0mA

signals are divided up into the four groups shown in Table 6-1, based on the loading characteristics of the particular signal. The signals in each group are given here.

6.1.1 Standard Signals

The majority of signals on the bus are in this group. These are bussed signals, driven actively on the bus by F-series (or compatible) drivers such as 74F245, usually tri-stated when ownership of the signal changed for master and slave, and generally terminated with a $220\Omega/330\Omega$ thevenin terminator. PICs can apply two standard loads to each of these signals when necessary.

/FCS	/CCS	/DS0-/DS3	/LOCK
A2-A7	AD8-AD31	SD0-SD7	READ
FC0-FC2	DOE	/IORST	/BCLR
/MTCR	/MTACK		

6.1.2 Clock Signals

All clock signals on the bus are in this group. Many designs are very sensitive to clock delay, skew, and rise/fall times, so loading on the clock lines must be kept to a minimum. These are bussed signals, actively driven by the backplane, and source terminated with a low value

series resistor. PICs can apply one standard load to each of these signals when necessary. Zorro II cards have the same clock rules, so there should never be clocking problems when using either card type in a backplane.

/C3 E Clock	CDAC	/C1	7M
----------------	------	-----	----

6.1.3 Open Collector Signals

Many of the bus signals are shared via open collector or open drain outputs rather than via tri-stated signals; this is of course required for some asynchronous things like the shared interrupt lines, and it works well for other types of signals as well. Of course, a backplane resistor pulls these lines high, PICs only drive the line low.

/OWN	/BGACK	/CINH	/BERR
/DTACK	/RESET	/INT₂	/INT₆
/HLT			

6.1.4 Non-bussed Signals

The non-bussed, or slot specific, signals are involved with only one slot on the bus (eg, each slot has its own copy). As a result, the drive requirements are much less for these signals. The backplane provides pullups or pulldowns, as required by the specific signal.

/CFGIN_N SenseZ₃	/CFGOUT_N /SLAVEN	/BR_N	/BG_N
--	--	------------------------	------------------------

6.2 Slot Power Availability

The system power for the Zorro III bus is totally based on the slot configurations. A backplane is always free to supply extra power, but it must meet the minimum requirements specified here. All PICs must be designed with the minimum specifications in mind, especially the tolerances.

Pin	Supply
5,6	+5 VDC ± 5% @ 2 Amps
8	-5 VDC ± 5% @ 60 mA
10	+12 VDC ± 5% @ 500mA
20	-12 VDC ± 5% @ 60mA

6.3 Temperature Range

The Zorro III bus is specified for operation over a temperature range of 0° C to 70° C.

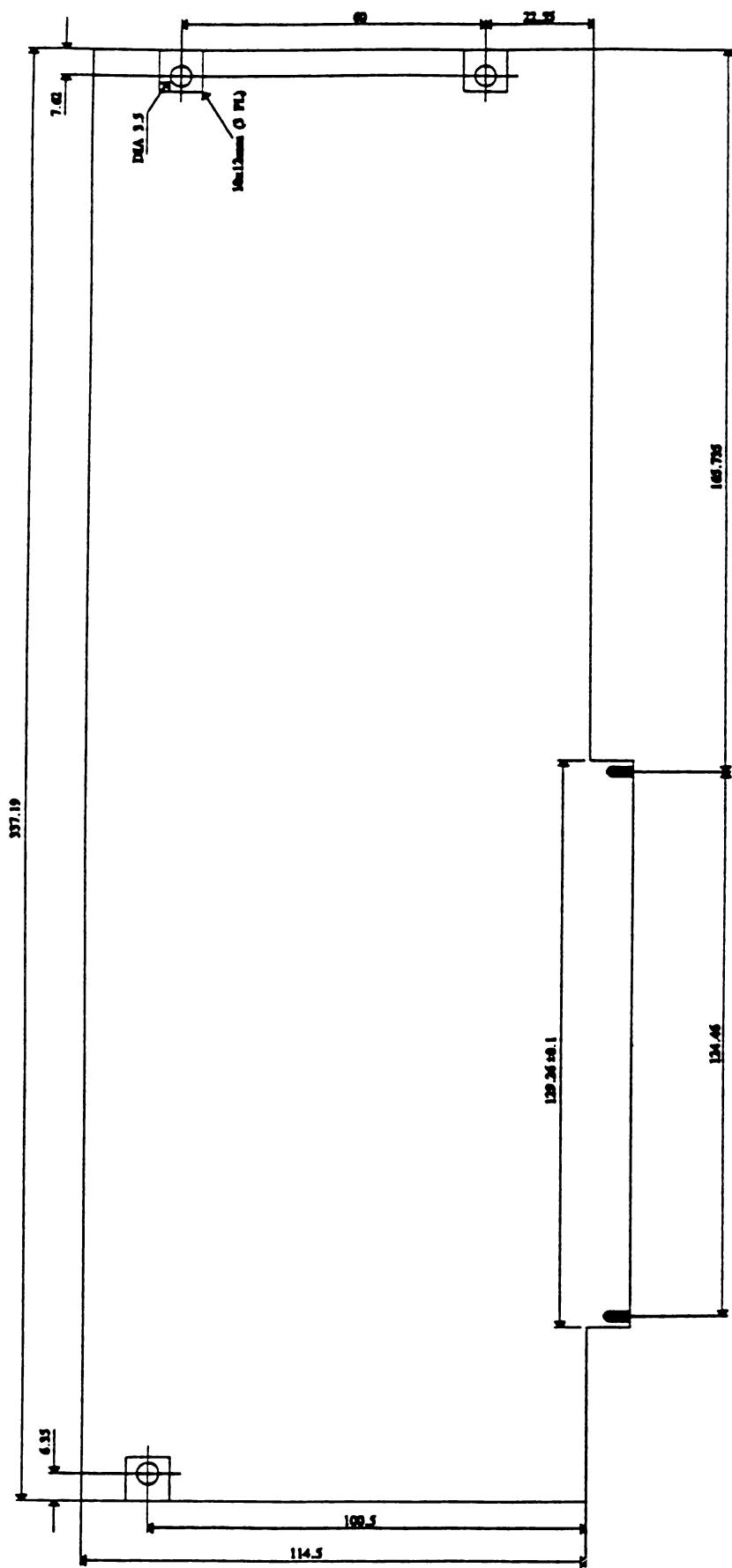
CHAPTER 7

MECHANICAL SPECIFICATIONS

"Never speak more clearly than you think."

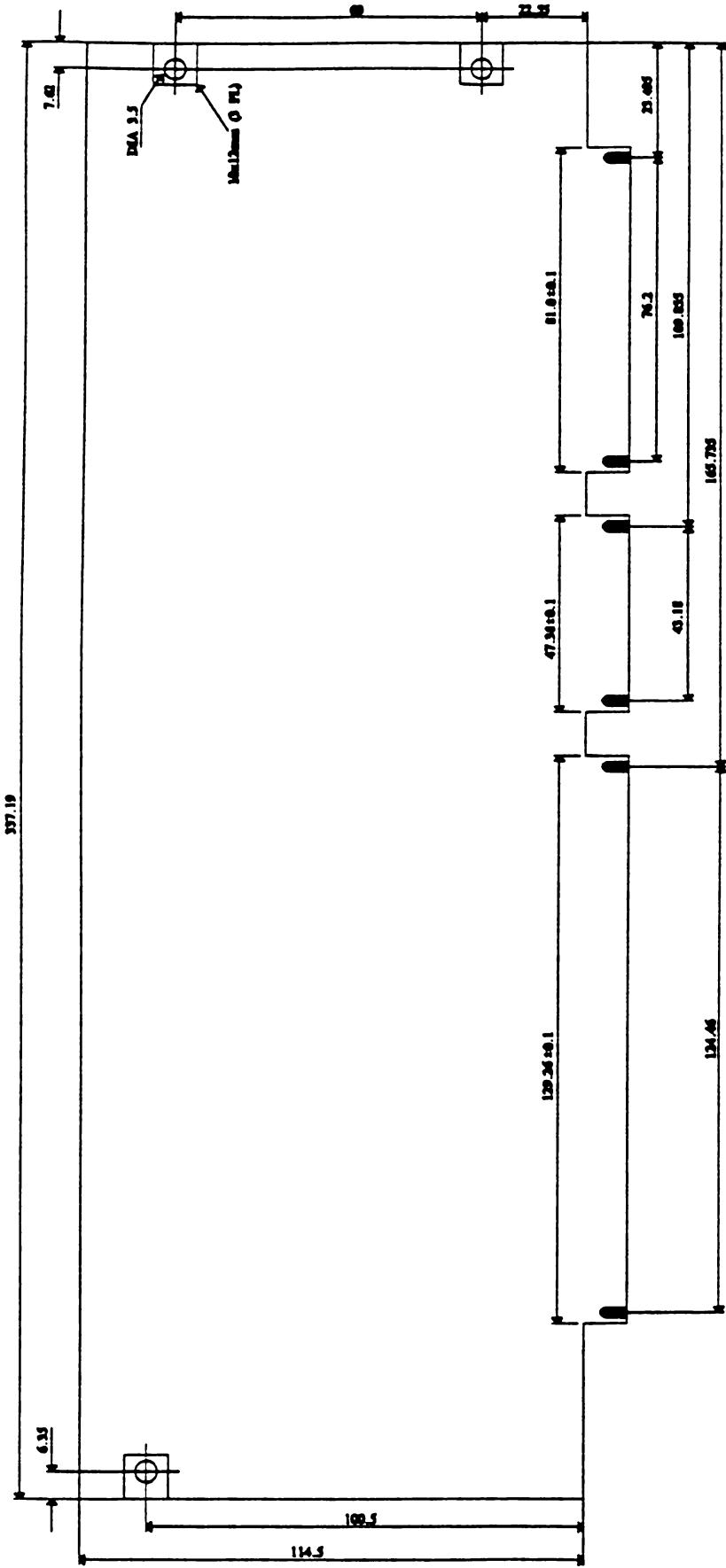
-Jeremy Bernstein

This section covers the various mechanical details of Zorro III cards. Note that these specifications are considered preliminary.



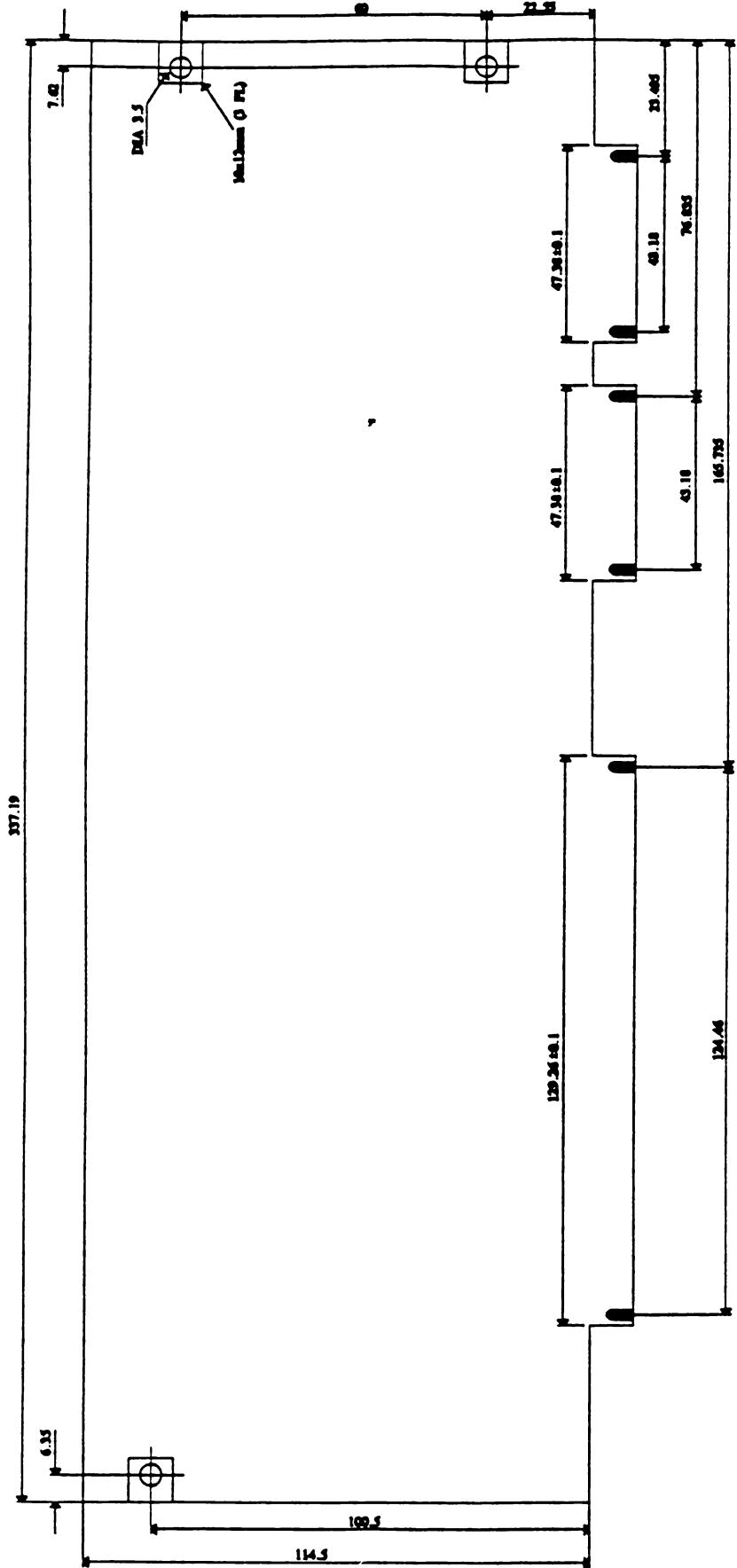
7.1 Basic Zorro III PIC

This drawing shows the basic Zorro III PIC. All of the dimensions are in millimeters.



7.2 PIC with ISA Option

This drawing shows the basic Zorro III PIC, with both Zorro III and the ISA Bus fingers specified. All of the dimensions are in millimeters.



7.3 PIC with Video Option

This drawing shows the basic Zorro III PIC, with both Zorro III and the Amiga Video Slot fingers specified. All of the dimensions are in millimeters.

This form factor is for the A3000 only. On the A4000, the video slot has been extended. See chapter 16 for a form factor specification of an A4000 PIC with video option.

CHAPTER 8

AUTOCONFIG®

"The goal of all inanimate objects is to resist man and ultimately defeat him."

-Russell Baker

8.1 The AUTOCONFIG® Mechanism

The AUTOCONFIG® mechanism used for the Zorro III bus is an extension of the original Zorro II configuration mechanism. The main reason for this is that the Zorro II mechanism works so well, there was little need to change anything. The changes are simply support for new hardware features on the Zorro III bus.

Amiga autoconfiguration is surprisingly simple. When an Amiga powers up or resets, every card in the system goes to its unconfigured state. At this point, the most important signals in the system are /CFGINN and /CFGOUTN. As long as a card's /CFGINN line is negated, that card sits quietly and does nothing on the bus (though memory cards should continue to refresh even through reset, and any local board activities that don't concern the bus may take place after /RESET is negated). As part of the unconfigured state, /CFGOUTN is negated by the PIC immediately on reset.

The configuration process begins when a card's /CFGINN line is asserted, either by the backplane, if it's the first slot, or via the configuration chain, if it's a later card. The configuration chain simply ensures that only one unconfigured card will see an asserted /CFGINN at one time. An unconfigured card that sees its /CFGINN line asserted will respond to a block of memory called *configuration space*. In this block, the PIC will assert a set of read-only

registers, followed by a set of write-only registers (the read-only registers are also known as AUTOCONFIG® ROM). Starting at the base of this block, the read registers describe the device's size, type, and other requirements. The operating system reads these, and based on them, decides what should be written to the board. Some write information is optional, but a board will always be assigned a base address or be told to shut up. The act of writing the final bit of base address, or writing anything to a shutup address, will cause the PIC to assert its /CFGOUT_N, enabling the next board in the configuration chain.

The Zorro II configuration space is the 64K memory block \$00E8xxxx, which of course is driven with 16 bit Zorro II cycles; all Zorro II cards configure there. The Zorro III configuration space is the 64K memory block beginning at \$FF00xxxx, which is always driven with 32 bit Zorro III cycles (PICs need only decode A₃₁-A₂₄ during configuration). A Zorro III PIC can configure in Zorro II or Zorro III configuration space, at the designer's discretion, but not both at once. All read registers physically return only the top 4 bits of data, on D₃₁-D₂₈ for either bus mode. Write registers are written to support nybble, byte, and word registers for the same register, again based on what works best in hardware. This design attempts to map into real hardware as simply as possible. Every AUTOCONFIG® register is logically considered to be 8 bits wide; the 8 bits actually being nybbles from two paired addresses.

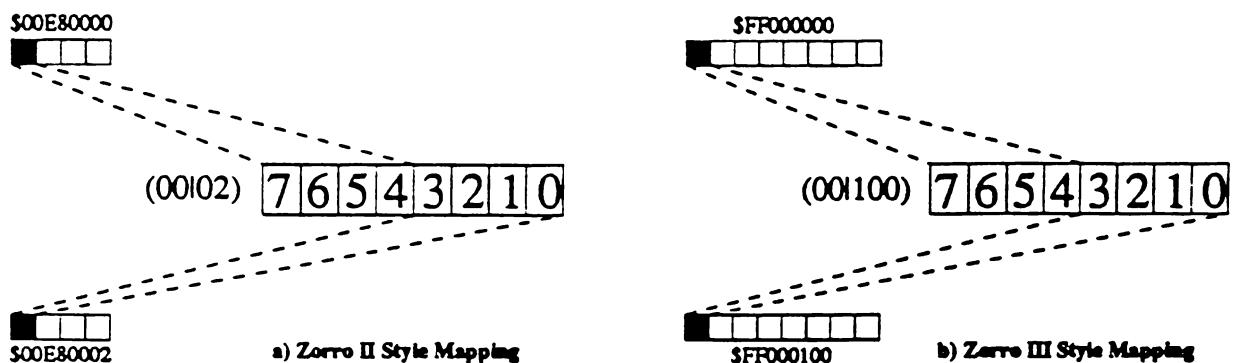


Figure 8-1: Configuration Register Mapping

The register mappings for the two different blocks are shown in *Figure 8-1*. All the bit patterns mentioned in the following sections are logical values. To avoid ambiguity, all registers are referred to by the number of the first register in the pair, since the first pair member is the same for both mapping schemes. In the actual implementation of these registers, all read registers except for the 00 register are physically complemented; eg, the logical value of register 3C is always 0, which means in hardware, the upper nybbles of locations \$00E8003C and \$00E8003E, or \$FF00003C and \$FF00013C, both return all 1's.

8.2 Register Bit Assignments

The actual register assignments are below. Most of the registers are the same as for the Zorro II bus, but are included here anyway for completeness. The Amiga OS software names for these registers in the ExpansionRom or ExpansionControl structures are included.

Reg_Z2_Z3_Bit

- 00 02 100 7,6 (*or_Type*) These bits encode the PIC type:
- | | |
|----|-----------|
| 00 | Reserved |
| 01 | Reserved |
| 10 | Zorro III |
| 11 | Zorro II |
- 5 If this bit is set, the PIC's memory will be linked into the system free pool. The Zorro III register 08 may modify the size of the linked memory.
- 4 Setting this bit tells the OS to read an autoboot ROM.
- 3 This bit is set to indicate that the next board is related to this one; often logically separate PICs are physically located on the same card.
- 2-0 These bits indicate the configuration size of the PIC. This size can be modified for the Zorro III cards by the size extension bit, which is the new meaning of bit 5 in register 08.
- | Bits | Unextended | Extended |
|------|---------------|---------------|
| 000 | 8 megabytes | 16 megabytes |
| 001 | 64 kilobytes | 32 megabytes |
| 010 | 128 kilobytes | 64 megabytes |
| 011 | 256 kilobytes | 128 megabytes |
| 100 | 512 kilobytes | 256 megabytes |
| 101 | 1 megabyte | 512 megabytes |
| 110 | 2 megabytes | 1 gigabyte |
| 111 | 4 megabytes | RESERVED |
- 04 06 104 7-0 (*or_Product*) The device's product number, which is completely up to the manufacturer. This is generally unique between different products, to help in identification of system cards, and it must be unique between devices using the automatic driver binding features.
- 08 0A 108 7 (*or_Flags*) This was originally an indicator to place the card in the 8 megabyte Zorro II space, when set, or anywhere it'll fit, if cleared. Under the Zorro III spec, this is set to indicate that the board is basically a memory device, cleared to indicate that the board is basically an I/O device.
- 6 This bit is set to indicate that the board can't be shut up by software, cleared to indicate that the board can be shut up.
- 5 This is the size extension bit. If cleared, the size bits in register 00 mean the same as under Zorro II, if set, the size bits indicate a new size. The

Reg Z2 Z3 Bit

most common new Zorro III sizes are the smaller ones; all new sized cards get aligned on their natural boundaries.

- 4 Reserved, must be 1 for all Zorro III cards.
- 3-0 These bits indicate a board's sub-size; the amount of memory actually required by a PIC. For memory boards that auto-link, this is the actual amount of memory that will be linked into the system free memory pool. A memory card, with memory starting at the base address, can be automatically sized by the Operating System. This sub-size option is intended to support cards with variable setups without requiring variable physical configuration capability on such cards. It also may greatly simplify a Zorro III design, since 16 megabyte cards and up can be designed with a single latch and comparator for base address matching, while 8 megabyte and smaller PICs require large latch/comparator circuits not available in standard TTL packages.

Bits	Encoding
0000	Logical size matches physical size
0001	Automatically sized by the Operating System
0010	64 kilobytes
0011	128 kilobytes
0100	256 kilobytes
0101	512 kilobytes
0110	1 megabyte
0111	2 megabytes
1000	4 megabytes
1001	6 megabytes
1010	8 megabytes
1011	10 megabytes
1100	12 megabytes
1101	14 megabytes
1110	Reserved
1111	Reserved

For boards that wish to be automatically sized by the operating system, a few rules apply. The memory is sized in 512K increments, and grows from the base address upward. Memory wraps are detected, but the design must insure that its data bus doesn't float when the sizing routine addresses memory locations that aren't physically present on the board; data bus pullups or pulldowns are recommended. This feature is designed to allow boards to be easily upgraded with additional or increased density memoried without the need for memory configuration jumpers.

Reg Z2 Z3 Bit

0C	0E 10C 7-0 (or_Reserve03)	Reserved, must be 0.
10	12 110 7-0	Manufacturer's number, high byte.
14	16 114 7-0 (or_Manufacturer)	Manufacturer's number, low bytes. These are unique, and can only be assigned by Commodore.
18	1A 118 7-0	Optional serial number, byte 0 (msb)
1C	1E 11C 7-0	Optional serial number, byte 1
20	22 120 7-0	Optional serial number, byte 2
24	26 124 7-0 (or_SerialNumber)	Optional serial number, byte 3 (lsb) This is for the manufacturer's use and can contain anything at all. The main intent is to allow a manufacturer to uniquely identify individual cards, but it can certainly be used for revision information or other data.
28	2A 128 7-0	Optional ROM vector, high byte.
2C	2E 12C 7-0 (or_InitDiagVec)	Optional ROM vector, low byte. If the ROM address valid bit (bit 4 of register (00102)) is set, these two registers provide the sixteen bit offset from the board's base at which the start of the ROM code is located. If the ROM address valid bit is cleared, these registers are ignored.
30	32 130 7-0 (or_Reserve0c)	Reserved, must be 0. Unsupported base register reset register under Zorro II.
34	36 134 7-0 (or_Reserve0d)	Reserved, must be 0.
38	3A 138 7-0 (or_Reserve0e)	Reserved, must be 0.
3C	3E 13C 7-0 (or_Reserve0f)	Reserved, must be 0.
40	42 140 7-0 (or_Interrupt)	Reserved, must be 0. Unsupported control state register under Zorro II.
44	46 144 7-0	High order base address register, write only.
48	4A 148 7-0 (or_Z3_HighByte) (or_BaseAddress)	Low order base address register, write only. The high order register takes bits 31-24 of the board's configured address, the low ordered register takes bits 23-16. For Zorro III boards configured in the Zorro II space, the configuration address is written both nybble and byte wide, with the ordering:

* The original Zorro specifications called for a few registers, like these, that remained active after configuration. Support for this is impossible, since the configuration registers generally disappear when a board is configured, and absolutely must move out of the \$00E8xxxx space. So since these couldn't really be implemented in hardware, system software has never supported them. They're included here for historical purposes.

Reg Z2 Z3 Bit

Reg	Nybble	Byte
46	A ₂₇ -A ₂₄	N/A
44	A ₃₁ -A ₂₈	A ₃₁ -A ₂₄
4A	A ₁₉ -A ₁₆	N/A
48	A ₂₃ -A ₂₀	A ₂₃ -A ₁₆

Note that writing to register 48 actually configures the board for both Zorro II and Zorro III boards in the Zorro II configuration block. For Zorro III PICs in the Zorro III configuration block, the action is slightly different. The software will actually write the configuration as byte and word wide accesses:

Reg	Byte	Word
48	A ₂₃ -A ₁₆	N/A
44	A ₃₁ -A ₂₄	A ₃₁ -A ₁₆

The actual configuration takes place when register 44 is written, thus supporting any physical size of configuration register.

4C	4E 14C 7-0 (ec_Shutup)	Shut up register, write only. Anything written to 4C will cause a board that supports shut-up to completely disappear until the next reset.
50	52 150 7-0	Reserved, must be 0.
54	56 154 7-0	Reserved, must be 0.
58	5A 158 7-0	Reserved, must be 0.
5C	5E 15C 7-0	Reserved, must be 0.
60	62 160 7-0	Reserved, must be 0.
64	66 164 7-0	Reserved, must be 0.
68	6A 168 7-0	Reserved, must be 0.
6C	6E 16C 7-0	Reserved, must be 0.
70	72 170 7-0	Reserved, must be 0.
74	76 174 7-0	Reserved, must be 0.
78	7A 178 7-0	Reserved, must be 0.
7C	7E 17C 7-0	Reserved, must be 0.

CHAPTER 9

ZORRO III SIGNAL NAMES

"I have been given the freedom to do as I see fit."

-REM

The Zorro III Bus signals vary based on the particular bus mode in effect. This table lists each physical pin by physical name, and then by the logical names for Zorro II mode, Zorro III mode, address phase, and Zorro III data mode, data phase.

Pin No.	Physical Name	Zorro II Name	Zorro III Address Phase	Zorro III Data Phase
1	Ground	Ground	Ground	Ground
2	Ground	Ground	Ground	Ground
3	Ground	Ground	Ground	Ground
4	Ground	Ground	Ground	Ground
5	+5VDC	+5VDC	+5VDC	+5VDC
6	+5VDC	+5VDC	+5VDC	+5VDC
7	/OWN	/OWN	/OWN	/OWN
8	-5VDC	-5VDC	-5VDC	-5VDC
9	/SLAVEN	/SLAVEN	/SLAVEN	/SLAVEN
10	+12VDC	+12VDC	+12VDC	+12VDC
11	/CFGOUT _N	/CFGOUT _N	/CFGOUT _N	/CFGOUT _N
12	/CFGIN _N	/CFGIN _N	/CFGIN _N	/CFGIN _N
13	Ground	Ground	Ground	Ground
14	/C3	/C3 Clock	/C3 Clock	/C3 Clock
15	CDAC	CDAC Clock	CDAC Clock	CDAC Clock
16	/C1	/C1 Clock	/C1 Clock	/C1 Clock

Pin No.	Physical Name	Zorro II Name	Zorro III Address Phase	Zorro III Data Phase
17	/CINH	/OVR	/CINH	/CINH
18	/MTCR	XRDY	/MTCR	/MTCR
19	/INT ₂	/INT ₂	/INT ₂	/INT ₂
20	-12VDC	-12VDC	-12VDC	-12VDC
21	A ₅	A ₅	A ₅	A ₅
22	/INT ₆	/INT ₆	/INT ₆	/INT ₆
23	A ₆	A ₆	A ₆	A ₆
24	A ₄	A ₄	A ₄	A ₄
25	Ground	Ground	Ground	Ground
26	A ₃	A ₃	A ₃	A ₃
27	A ₂	A ₂	A ₂	A ₂
28	A ₇	A ₇	A ₇	A ₇
29	/LOCK	A ₁	/LOCK	/LOCK
30	AD ₈	A ₈	A ₈	D ₀
31	FC ₀	FC ₀	FC ₀	FC ₀
32	AD ₉	A ₉	A ₉	D ₁
33	FC ₁	FC ₁	FC ₁	FC ₁
34	AD ₁₀	A ₁₀	A ₁₀	D ₂
35	FC ₂	FC ₂	FC ₂	FC ₂
36	AD ₁₁	A ₁₁	A ₁₁	D ₃
37	Ground	Ground	Ground	Ground
38	AD ₁₂	A ₁₂	A ₁₂	D ₄
39	AD ₁₃	A ₁₃	A ₁₃	D ₅
40	Reserved	(/EINT ₇)	Reserved	Reserved
41	AD ₁₄	A ₁₄	A ₁₄	D ₆
42	Reserved	(/EINT ₅)	Reserved	Reserved
43	AD ₁₅	A ₁₅	A ₁₅	D ₇
44	Reserved	(/EINT ₄)	Reserved	Reserved
45	AD ₁₆	A ₁₆	A ₁₆	D ₈
46	/BERR	/BERR	/BERR	/BERR
47	AD ₁₇	A ₁₇	A ₁₇	D ₉
48	/MTACK	(/VPA)	/MTACK	/MTACK
49	Ground	Ground	Ground	Ground
50	E Clock	E Clock	E Clock	E Clock
51	/DS ₀	(/VMA)	/DS ₀	/DS ₀
52	AD ₁₈	A ₁₈	A ₁₈	D ₁₀
53	/RESET	/RST	/RESET	/RESET
54	AD ₁₉	A ₁₉	A ₁₉	D ₁₁
55	/HLT	/HLT	/HLT	/HLT
56	AD ₂₀	A ₂₀	A ₂₀	D ₁₂
57	AD ₂₂	A ₂₂	A ₂₂	D ₁₄
58	AD ₂₁	A ₂₁	A ₂₁	D ₁₃
59	AD ₂₃	A ₂₃	A ₂₃	D ₁₅

Pin No.	Physical Name	Zorro II Name	Zorro III Address Phase	Zorro III Data Phase
60	/BR _N	/BR _N	/BR _N	/BR _N
61	Ground	Ground	Ground	Ground
62	/BGACK	/BGACK	/BGACK	/BGACK
63	AD ₃₁	D ₁₅	A ₃₁	D ₃₁
64	/BG _N	/BG _N	/BG _N	/BG _N
65	AD ₃₀	D ₁₄	A ₃₀	D ₃₀
66	/DTACK	/DTACK	/DTACK	/DTACK
67	AD ₂₉	D ₁₃	A ₂₉	D ₂₉
68	READ	READ	READ	READ
69	AD ₂₈	D ₁₂	A ₂₈	D ₂₈
70	/DS ₂	/LDS	/DS ₂	/DS ₂
71	AD ₂₇	D ₁₁	A ₂₇	D ₂₇
72	/DS ₃	/UDS	/DS ₃	/DS ₃
73	Ground	Ground	Ground	Ground
74	/CCS	/AS	/CCS	/CCS
75	SD ₀	D ₀	Reserved	D ₁₆
76	AD ₂₆	D ₁₀	A ₂₆	D ₂₆
77	SD ₁	D ₁	Reserved	D ₁₇
78	AD ₂₅	D ₉	A ₂₅	D ₂₅
79	SD ₂	D ₂	Reserved	D ₁₈
80	AD ₂₄	D ₈	A ₂₄	D ₂₄
81	SD ₃	D ₃	Reserved	D ₁₉
82	SD ₇	D ₇	Reserved	D ₂₃
83	SD ₄	D ₄	Reserved	D ₂₀
84	SD ₆	D ₆	Reserved	D ₂₂
85	Ground	Ground	Ground	Ground
86	SD ₅	D ₅	Reserved	D ₂₁
87	Ground	Ground	Ground	Ground
88	Ground	Ground	Ground	Ground
89	Ground	Ground	Ground	Ground
90	Ground	Ground	Ground	Ground
91	SenseZ ₃	Ground	SenseZ ₃	SenseZ ₃
92	7M	E7M	7M	7M
93	DOE	DOE	DOE	DOE
94	/IORST	/BUSRST	/IORST	/IORST
95	/BCLR	/GBG	/BCLR	/BCLR
96	Reserved	(/EINT ₁)	Reserved	Reserved
97	/FCS	No Connect	/FCS	/FCS
98	/DS ₁	No Connect	/DS ₁	/DS ₁
99	Ground	Ground	Ground	Ground
100	Ground	Ground	Ground	Ground

CHAPTER 10

INTRODUCTION TO THE LOCAL BUS

"What works for me might work for you"

-Jimmy Buffet

This article describes the 200-pin Local Bus Expansion Slot (also known as the Coprocessor Slot) of the A4000 and A3000 Amiga models. This expansion slot is designed to provide high speed access to the Amiga's local, or 68030, bus. This slot is intended for high speed expansion devices that are generally very specific to the 68030 bus or need direct access to the local bus for other reasons. Such devices include alternate 680x0 family processors, cache memory boards, high speed bursting RAM expansion, and similar things.

10.1 Intended Audience

The information presented here is for hardware engineers interested in designing cards for the Local Bus Slot. A good level of microcomputer systems design knowledge is necessary to get much meaning out of these pages. Especially important is familiarity with the 68030 processor hardware conventions, as most of the Local Bus Slot is based directly on the 68030 processor bus. These conventions are described in the *MC68030 User's Manual* by Motorola (3rd Edition, Prentice-Hall, ISBN 0-13-566423-3).

10.2 Why a Local Bus Slot?

The local bus slot was originally introduced on the Amiga 2000 and has served its intended purpose quite well on that system. On the A3000 and A4000 the slot has been expanded from 86 pins to 200 pins but serves the same basic purpose. Its main use is to allow the addition of a single, very tightly coupled expansion device, typically a high speed CPU, cache, or memory board. An alternate 680x0 family device, such as a 68040 processor, needs access to every 68030 signal in order to properly replace the 68030 processor as a host for AmigaOS or UNIX. Cache memories or high-performance Fast memory need direct access to

the 68030 bus to run as tightly coupled as possible to the 68030 processor on the motherboard. Most other types of expansion devices should be designed as Zorro II or Zorro III expansion cards.

In any system design, you can make good arguments for a general purpose expansion bus, and good arguments for an extendable local bus. The Amiga philosophy is that both of these approaches are correct, and serve complementary needs. The kinds of devices that would work well in the Local Bus Slot are by their very nature tightly coupled to the system bus, which is of course based on the 68030 bus. Such devices are not expected to work in future Amiga systems, which could easily have completely different local buses, and therefore, different local bus slots. Also, it's impractical to provide a large number of such slots, since the local bus itself has tight electrical limits on expandability. So a single local bus slot is provided.

10.3 Why an Expansion Bus Slot?

Most add-on cards for an A3000 or A4000 belong in a Zorro III Expansion Bus Slot. First of all, since there is only one Local Bus Slot, it stands to reason that only one such device can be added to any system, while four Zorro III Expansion Bus slots are available on the A3000 and A4000 models. The Zorro III bus doesn't permit the same degree of tight coupling that the Local Bus Slot does, so it is not capable of supporting cache or other zero wait-state memory, and it can't support a direct-replacement 68040. It does permit reasonable speeds, interrupts, bus locking, etc. so it is the place for high performance I/O devices, moderate speed add-on memory boards, processor devices such as DSP, video, or RISC devices that coexist with the main processor, etc.

And of course, devices that are happy as slower 16-bit peripherals can be implemented as Zorro II cards and have the advantage of working in all Amiga computers (including the A500 and A1000 with the proper 3rd party bus adaptor or backplane). Details on the Zorro III bus are available in The Zorro III Expansion Bus Specification in chapters 1-9 of this document and also listed in Appendix K (p. 383) of the *Amiga Hardware Reference Manual*, 3rd Edition (ISBN 0-201-56776-8).

CHAPTER 11

FUNCTIONALITY AND DESIGN GUIDELINES

"Time and distance are out of place here"

-REM

The Local Bus Expansion Slot provides signals to implement both slave and master devices. Memory devices, including cache, are bus slaves, while CPU devices such as 680x0 accelerator boards are bus masters. Any card may, at times, be either slave or master, and in a few cases, both at once.

11.1 Slave Devices

A slave device is a device that responds to the current local bus master. The local connector provides direct access to all local bus signals, which include all 68030 signals, plus a few additional Amiga-specific lines to allow a Local Bus Slot device to control the Amiga's Local Bus.

Local Bus Slot slaves are not autoconfigured like a Zorro III bus device, but instead are fixed in the address range from \$08000000 to \$0FFFFFFF. The Local Bus Slot controller, the Fat Gary chip, provides a decode of this space on the signal /RAMSLOT which is valid at address time. This signal can be ignored and the address decoded by logic on the board if speed is an issue.

Local slave devices should also support the signal /CIIN if they contain uncachable data. The Amiga OS expects anything mapped starting at \$08000000 to be memory, and in fact, the fastest memory in the A3000 and A4000 systems. The OS will automatically size and link in any memory it finds here, and place it in the system as the highest priority memory available. To support control registers or other memory mapped resources on a Local Bus Card, locate them at least 512K above the \$08000000 base, and the OS will ignore them.

A signal named /WAIT is provided for cache support. Asserting this signal will disable address decoding of onboard Fast RAM by the RAMSEY chip and Zorro II/III bus accesses by the BUSTER chip. Constraints imposed by the 68030 allow only 18ns to determine a cache hit. It is often more feasible to assert /STERM before knowing whether the cycle is a cache hit or cache miss and rerunning the cycle via /HALT and /BERR if it is a miss. To achieve this functionality any decoding of the first cycle by RAMSEY or BUSTER must be disabled by asserting /WAIT less than 10ns after address valid. If the cycle is determined to be a cache miss, a rerun is initiated and wait deasserted for the secondary cycle. Assertion of /WAIT will keep /STERM, /CBACK, etc. tristated by BUSTER or RAMSEY and may be controlled by the cache control logic.

11.2 Master Devices

A Master Device is, of course, a device which masters the local bus, replacing the functions of the 68030 during its period of mastership. Bus mastership may be accomplished two ways depending on the desired functionality. The first mode, called primary mastership, totally disables the motherboard 68030 and its arbitration logic, essentially replacing the 68030 with the local slot device. Such a device takes over full arbitration responsibility for the whole system, and must service all interrupts.

The second mode, called secondary mastership, allows the on-board 68030 and the local bus accelerator board to share the bus, permitting multiprocessor capabilities or more traditional DMA from the local bus board. This protocol permits very fast switching between the local slot master and the motherboard 68030. In this mode, the 68030 is still responsible for bus arbitration.

11.2.1 Primary Bus Mastership

The primary bus mastership, or arbitration takeover mode, requires less logic to implement and may be preferred in most implementations. In the absence of multiprocessing software support, this is the mode of choice, and corresponds to the way in which most A2000 local slot cards worked.

The local bus card asserts /CBR at power on to the motherboard which in turn asserts /BR to the motherboard 68030. Upon receiving /BG30 from the motherboard the local card asserts /BOSS. Logic on the motherboard uses /BOSS to force /BGACK30 low to the 68030 only and not the shared local bus /BGACK. In addition the assertion of /BOSS tristates /BG on the motherboard and in turn the local card should untristate and source its /BG.

The local card is now the default bus master and arbiter – it must provide arbitration for the local bus, based on the 68030 bus arbitration rules. The onboard 68030 is bus arbitrated away and never regains the bus. PAL equations to implement this are given in Figure 11-1. All PAL equations are active high and should be inverted in the output stage of the PAL for active low assertion.

```

/* Always assert Coprocessor Bus Request. */
CBR      =      'b'1;

/* The Boss Signal. "poweron_reset" is a signal sourced by the local card and
   is asserted for a few hundred nanoseconds after poweron. This clears the
   feedback path on the pal and may be generated by an RC network which is
   slewrate cleaned by a schmitt trigger device. */
BOSS     =      BG30 # BOSS & !poweron_reset;

/* The Grant Signal. "local_card_bg" is sourced by the local card in
   compliance to the operation of arbitration defined in the 68030 users
   manual. */
BG       =      local_card_bg;

BG.oe    =      BOSS;

```

Figure 11-1: Primary Mode Takeover PAL Equations

Actual timing for takeover mode is not given since all signals are inherently asynchronous. The untristating of /BG should be later than the tristating of /BG on the motherboard to minimize contention on that signal.

11.2.2 Secondary Bus Mastership

The secondary bus acquisition mode uses the 68030 arbiter to provide cycle arbitration between the Local Bus Slot card and the motherboard DMA. Since the 68030 provides only a single bus request input, a scheme referred to as fast arbitration is used between the local card and all other DMA sources, which are controlled via the BUSTER chip. Bus request is an open collector line which is time multiplexed between the local card and BUSTER. On a positive edge of CPUCLK, BUSTER will assert /BR if it requires the bus and /BR is not already asserted by the local bus card. On the negative transition of CPUCLK, the local bus card may assert /BR if it is not already asserted by BUSTER. This scheme allows both masters to share a single bus request and also requires only 20ns to resolve the master arbitration. This mode is very efficient but forces the local card to use high speed logic since the time between clock edges is so short.

It is strongly suggested that the above logic be incorporated in a 7.5ns or faster registered PAL connected to a F38 open collector device. Clock skew between the clock to this PAL and CPUCLK is critical and it is advised that the local card generate and source clocks to the motherboard, especially if CPUCLK is needed elsewhere on the card; load on CPUCLK at the Local Bus Slot must be kept to a minimum to avoid mucking with the local bus timing. Skew between CPUCLK and the clock driving this PAL should be less than 2ns. The PAL equations for /BR are given in Figure 11-2.

After receiving /BG from the motherboard 68030 the local card drives /BGACK (open collector) and assumes mastership of the bus. It may keep the bus for multiple cycles but should not hog the bus for extended periods unless it relinquishes the bus when DMA request is asserted by BUSTER. Hogging the bus may cause adverse operation of the system. Be aware that the

```

/* This is the BR_LOCAL signal, which is an active output fed into a 74f38, or
other open collector equivalent buffer, to produce /BR. /BR is the raw bus
request from the local bus connector (keep this trace short on the local
card). WANTBUS is the request from the local card which deasserts after it
sees BR_LOCAL asserted. BGACK_LOCAL is /BGACK asserted by the local card.
RESET is any reset signal used to prevent poweron latchup of BR_LOCAL. */

BR_LOCAL.d = !BR & !BR_LOCAL & WANTBUS
#     BR_LOCAL & !BGACK_LOCAL & !RESET;

```

Figure 11-2: Secondary Mode Takeover PAL Equations

local bus signals must be tristated (/AS, /DSACK, Address, Data, etc.) prior to deasserting /BGACK. In addition the local card must not drive the bus or /BGACK until /AS, /DSACK, /BERR, /HALT, etc. have deasserted. In other words, standard 680x0 bus arbitration rules apply.

It is generally assumed that any secondary bus master mode Local Bus Slot device will attempt to prevent undue bus hogging at the design level. This implies either a coprocessor device of some kind that makes only periodic requests of the bus, or a CPU subsystem that contains its own local cache or memory. Any device that requires the mastering of the local bus for very long periods of time should be a primary mode bus master.

It is extremely important that the bus master timing from the local card emulate operation of a 25MHz or 16MHz 68030 chip exactly as defined by the *MC68030 User's Manual* (ISBN 0-13-566423-3). The bus control chips currently incorporate burst cycles and cache coherency signals (/CIIN) and future enhancements may incorporate rerun cycles. Signals received by the local card from the motherboard provide only the minimum setup and hold required by a 68030. Do not assume for example that data setup from the motherboard will not significantly change, i.e., data setup from Fast RAM on subsequent cycles of a burst is much less than a typical non-burst cycle.

In addition to the 68030 specifications, the Amiga local bus adds one additional constraint. Slave devices should hold data through the end of cycle, regardless of whether the cycle is terminated by /STERM or the /DSACK lines. In other words, only the current bus master really knows when a cycle has completed. Obviously, burst cycles are an exception to this; they follow the standard 68030 rules in dealing with data hold times.

11.3 Clock generation

The A3000 and A4000 motherboards provide links to disable generation of CPUCLK and CLK90. This allows the Local Bus Slot card to maintain better clock skew relationships between its own logic and that of the motherboard, and is especially important if the local bus card depends upon being synchronous to the motherboard clocks. Just as with the A2000 local bus slot, both synchronous and asynchronous designs are possible, but synchronous designs are much more timing-critical than on the A2000.

If a local bus card drives the clock lines the appropriate jumpers must be moved on the motherboard. CLK90 must be a clock 90° out of phase from CPUCLK. It is typically generated from a 5 tap 25ns delay line where CLK90 is the 10ns tap when running the system at 25Mhz and CLK90 is the 15ns tap when at 16Mhz. Note that these clocks are fed through a 74f08 to provide clocking to the motherboard. If the skew generated by the f08 is unacceptable (as in fast arbitration) the socketed f08 may be removed and replaced by a header which shorts the appropriate inputs to outputs, though it is the responsibility of the local bus card at that point to make sure acceptable versions of CPUCLK and CLK90 exist everywhere on the motherboard. Again careful layout of the clock circuitry is essential for reliable operation.

11.4 Local Bus Design Criteria

Any design which plugs into the Local Bus Slot connector must comply to some basic design rules.

- Any design which plugs into the Local Bus Slot connector must comply to some basic design rules.
- Due to inductance in the 200-pin connector to VCC and GND it is very important to provide ample bypass capacitance in order to maintain good DC VCC and GND levels.
- All signals from this connector are unbuffered and should not be heavily loaded. A good rule is 2 TTL loads. In addition receivers and drivers should be located near the connector and any connector signal should not run over 4 inches in length from the connector before entering or leaving a driver or receiver.
- Clock generation is especially critical. Keep traces short, ECL routing rules should be followed if possible. Fan out multiple clocks from a single die to minimize loading per clock. Light damping resistors minimize radiation but cause clock distortion, so tune the values carefully.
- Keep in mind that current draw in the A3000 and A4000 is tight so use CMOS and powerdown DRAM modes when possible. New FCT devices use significantly lower current than F and run faster. A local bus card should draw no more than 2 Amps @ 5 VDC.
- Be aware of heat dissipation issues especially on very high speed microprocessors.
- Noise test local bus cards and ensure good AC signal quality since a nasty signal will get nastier after passing through an inductive connector to the motherboard. And keep in mind that any noise a local card injects into the system will make the entire system less reliable.

- Local bus card mounting holes are plated through to ground on the A3000 and A4000 motherboard and provide an additional low inductance path to ground. Use this path to minimize ground bounce relative to the motherboard.

CHAPTER 12

LOCAL BUS SIGNAL DESCRIPTIONS

*"There's a name for it
And names make all the difference in the world"*
-David Byrne

The signals on the Local Bus Slot can be broken down into several categories. Some of these are in common with the 68030, some are specific to this slot. The pinout is listed at the end of this chapter.

12.1 Power Connections

These signals provide digital supply levels to the local bus card. There are quite a few of both levels on the physical connector; generally at least one for every two or three signal pins.

Digital Ground (GND)

This is the digital supply ground used by all digital devices in the system. The local bus card gets GND through its mounting posts as well as the connector pins.

Digital Supply (+5VDC)

This is the digital supply. This is specified as +5VDC +/- 5%. The system power budget allocates up to 2 Amps for this slot.

12.2 System Initialization

These signals are driven by the motherboard logic to initialize the system; local bus cards should listen as appropriate.

/RESET

This is an open-collector signal driven by the system reset logic or, indirectly, any CPU device that needs to reset the I/O subsystem. Local bus cards generally don't use this, though it can be used to reset I/O devices or hold the main system in reset if necessary.

/FPURST

This active input is driven by the system reset logic to indicate the full CPU register reset condition.

/CPURST

This open-collector signal is driven by the system reset logic to indicate a full CPU register reset condition to the CPU, and can be driven by the CPU to cause an I/O reset in the rest of the system.

12.3 68030 Signals

All of these signals are directly connected to the 68030, and more information on them is available in the *68030 User's Manual*. Most of these must be driven or sampled by any local slot DMA device.

Address Bus (A31 - A0)

The 32-bit processor address bus, driven by the bus master, tristated by inactive masters.

Data Bus (D31 - D0)

The 32-bit processor data bus, driven by the bus master for writes, the slave for reads.
This is tristated when outside of a bus cycle (/AS is negated).

Function Codes (FC2 - FC0)

An address bus extension, driven by the bus master, tristated by inactive masters. Most slaves respond only to FC0 + FC1.

Bus Size (SIZ1 SIZ0)

Data bus size request, driven by the bus master, tristated by inactive masters.

Cycle Strobes (/AS, /DS)

/AS indicated the start of a bus cycle and valid addresses, /DS indicates valid data for write cycles. Both are driven by the bus master, tristated by inactive bus masters.

Read Indicator (R/W)

Driven by this bus master, this tristated signal is high to indicate a read, low to indicate a write.

Read-Modify-Write Cycle (/RMC)

This line is asserted by the bus master to effect a bus lock; while it is active, no bus arbitration takes place, and shared memory coprocessors (except Agnus) stay out of the memory involved in the transaction.

Other Strobes (/ECS, /OCS, /DBEN)

These are additional 68030 strobes that aren't used by the Amiga and therefore, don't have to be driven by a local bus master. They are provided here for the possible use of any slave device that wants them; see the *68030 User's Manual* for signal details.

Burst Control (/CBREQ, /CBACK)

The local bus slave drives /CBREQ to indicate that it's capable of supporting a burst cycle. The local bus master responds with /CBACK to indicate that it can run a burst cycle.

Cache Control (/CIIN, /CIOUT)

The bus slave drives the /CIIN line to indicate that the currently addressed location is uncacheable. The bus master drives /CIOUT to indicate that the current location is uncacheable, based on MMU tables as well as /CIIN.

Cycle Termination (/STERM, /DSACK, /DSACK, /AVEC).

The bus slave drives either /STERM or one or more /DTACKs to normally terminate a cycle. The combination of DSACK lines indicates the bus port size; /STERM can only be generated by 32-bit-port slaves. The /AVEC line is driven by local bus logic to terminate an interrupt acknowledge cycle with an autovector rather than a device-supplied vector.

Interrupts (/IPL2 - /IPL0)

Encoded interrupt inputs. These are generally serviced only by the primary bus master, though other schemes are possible with the proper software support. They are inputs; they can't ever be driven by a slave or a master.

Interrupt Pending (/IPEND)

Driven by the 68030 to indicate that there is a pending interrupt to be serviced. A secondary bus master may use this as an indication to let the 68030 back onto the local bus, if the 68030 is handling the interrupts.

Exceptions (/HALT, /BERR)

/HALT driven alone causes the CPU to stop; generally this is used by single-stepping emulators. /HALT driven with /RESET indicates a full 68000 style reset, and is considered archaic on the A3000 and A4000 local bus. /BERR driven alone indicates some kind of bus error, generally a bus collision or timeout. /BERR and /HALT driven together indicate a bus retry.

12.4 Bus Arbitration Signals

These are the signals used to arbitrate the local bus in the ways previously described, supporting both primary and secondary bus masters.

Bus Requests (/BR, /CBR)

The both the /BR and /CBR line cause the bus to be requested from the 68030. The /BR line is for secondary bus masters, and it is a time multiplexed open collector line shared with a similar /BR output from the Buster chip. The /CBR line causes a request to go to the 68030, and once the primary arbitration is completed, the new primary master on the Local Bus Slot must deal with incoming /BR signals from Buster.

SCSI Bus Request (/SBR)

This line allows the local bus card to monitor when the SCSI devices wants the bus; this is primarily used as an indicator to secondary masters to give up the local bus.

Bus Grants (/BG, /BG30)

The /BG line is the main local bus grant signal. It is normally generated by the 68030, but when a primary master takes over, this line will tristate, allowing the primary master to drive /BG in response to an incoming /BR. The /BG30 line is the bus grant line coming directly from the 68030 chip.

Bus Grant Acknowledges (/BGACK, /BOSS)

The /BGACK signal is the main bus grant acknowledge, shared by Buster and the DMAC. Secondary masters drive /BGACK to acquire the local bus. The /BOSS signal is a private /BGACK-equivalent to the 68030, used by a primary bus master to acquire the bus from the 68030.

Bus Clear Request (/EBCLR)

This is a signal from the bus arbiter, indicating that some other bus master wants the local bus. This is generally used by a secondary bus master as an indicator of when to get off the bus.

12.5 Other Local Bus Signals

Local Slot Memory Decode (/RAMSLOT)

This is an address based chip select for the region of memory allocated to the local bus slot, \$08000000-\$0fffffff.

Emulator Mode (/EMUL)

This signal can be driven by local bus slot emulator devices to pull the /CDIS and /MMUDIS lines on the 68030, thereby disabling the cache and MMU for debugging purposes.

Cycle Wait (/WAIT)

This line is asserted by a bus monitoring device, such as a cache, to hold off cycle start by either the memory controller (RAMSEY) or expansion bus controller (BUSTER). This gives the device time to determine if it owns that address, retry the cycle, or anything else necessary to support cache and similar kinds of devices.

FPU Chip Select (/FPUCS)

This is a decode for the Coprocessor Device 1, the FPU, generated by the Gary Chip.

12.6 Clocks

This section details the Amiga system clocks available at the Local Bus Slot, the clocking alternatives available to a local bus device, and various clock control lines to facilitate this control.

System Clocks (CPUCLK, CLK90)

These are the main Amiga system clocks. CPUCLK is a 16MHz or 25MHz clock, depending on the system configuration, and is the main system, CPU, and FPU clock. CLK90 is CPUCLK shifted 90°.

External Clocks (EXTCLK, EXT90)

These clocks can be driven to replace the on-board clocks to the main system. Motherboard jumpers can be arranged to permit the use of these clocks.

12.7 Amiga 3000T Signals

These Local Bus Slot signals appear only in the tower version of the A3000, the A3000T.

System Clock Steal (DIS_CLKS)

This implements an alternate clock replacement method. When the DIS_CLKS line is driven high, the replacement clocks can be driven onto the local bus. This eliminates the need for any jumper adjustments to be made on the motherboard when a clock sourcing board is installed.

Replacement Clocks (ECPUCLK, ECPUCLKB, ECLK90, ECLK90A)

These are replacement main system clocks, and their 90° counterparts, that can be directly driven onto the local bus when DIS_CLKS is asserted.

CPU Clock Steal (DIS_CLK30, ECLK30)

This allows the 68030 clock to be driven from the local bus, again for skew reduction or other such tricks. When DIS_CLK30 is asserted, ECLK30 can be driven by local bus card logic.

12.8 Amiga 4000 Signals

These local bus slot signals appear in the A4000 version of the Amiga only.

Interrupt 6 Out (/INT6)

This is the shared level six interrupt line. It can be driven by a Local Bus Slot device to interrupt the host CPU.

Interrupt 2 Out (/INT2)

This is the shared level two interrupt line. It can be driven by a Local Bus Slot device to interrupt the host CPU.

SCSI (/SCSI)

Address decode from GARY for \$00DD0000 - \$00DD3FFF of user and supervisor space.
Reserved.

DMA Enable (/DMAEN)

Turns on DMA counter in RAMSEY. Reserved.

12.9 Local Bus Connector Pinout

Here is a complete list of the pins and signals on the local bus connector. For a description of the signals see the preceding section. A more complete description of the standard 68030 inputs and outputs is available from the 68030's *User Manual* (ISBN 0-13-566423-3).

The local bus connector is a two-piece, 200-pin, high-density connector. It is made by KEL. Be very careful with connection direction and pinout when doing board layouts.

Pin/ Signal Name	Pin/ Signal Name
1 /DSACK1	35 Ground
2 Ground	36 A [0]
3 Ground	37 A [9]
4 /HALT	38 Ground
5 R/W	39 Ground
6 Ground	40 A [1]
7 Ground	41 A [10]
8 /BGACK	42 Reserved (ECLK90A on A3000T)
9 /SBR	43 /INT6 (Unused in A3000/A3000T)
10 Ground	44 A [2]
11 Ground	45 A [11]
12 /AVEC	46 Reserved (ECLK90 on A3000T)
13 EXT90	47 Ground
14 +5 VDC	48 A [3]
15 +5 VDC	49 A [12]
16 /RAMSLOT	50 Ground
17 /BOSS	51 Ground
18 +5 VDC	52 A [4]
19 +5 VDC	53 A [13]
20 FC [0]	54 Reserved (ECPUCLKB on A3000T)
21 /STERM	55 /WAIT
22 +5 VDC	56 A [5]
23 +5 VDC	57 A [14]
24 FC [1]	58 Reserved (ECPUCLKA on A3000T)
25 /BR	59 Ground
26 +5 VDC	60 A [6]
27 +5 VDC	61 A [15]
28 /CBACK	62 Ground
29 /BERR	63 Ground
30 Reserved (DIS_CLKS on A3000T)	64 A [7]
31 /EMUL	65 A [16]
32 /CBREQ	66 /SCSI (Unused in A3000/A3000T)
33 A [8]	67 /DMAEN (Unused in A3000/A3000T)
34 Reserved (ECLK30 on A3000T)	68 A [24]

Pin/ Signal Name	Pin/ Signal Name
69 A [17]	113 /EBCLR
70 Reserved (DIS_CLK30 in A3000T)	114 Reserved
71 Ground	115 Ground
72 A [25]	116 /IPEND
73 A [18]	117 /RESET
74 Ground	118 Ground
75 Ground	119 Ground
76 A [26]	120 /IPL [0]
77 A [19]	121 SIZ0
78 Reserved	122 Ground
79 Reserved	123 Ground
80 A [27]	124 /IPL [1]
81 A [20]	125 FC [2]
82 /INT2 (Unused in A3000/A3000T)	126 CLK90_EXP
83 Ground	127 Ground
84 A [28]	128 /IPL [2]
85 A [21]	129 SIZ1
86 Ground	130 Ground
87 Ground	131 Ground
88 A [29]	132 /CIIN
89 A [22]	133 /AS
90 Reserved	134 /FPUCS
91 /DSACK0	135 CPUCLK_EXP
92 A [30]	136 /OCS
93 A [23]	137 D [31]
94 +5 VDC	138 Ground
95 +5 VDC	139 Ground
96 A [31]	140 D [15]
97 /DS	141 D [30]
98 +5 VDC	142 Ground
99 +5 VDC	143 Ground
100 /ECS	144 D [14]
	145 D [29]
101 /CIOUT	146 Reserved
102 +5 VDC	147 /CBR
103 +5 VDC	148 D [13]
104 /DBEN	149 D [28]
105 /BG	150 Reserved
106 +5 VDC	151 Ground
107 +5 VDC	152 D [12]
108 /RMC	153 D [27]
109 /CPURST	154 Ground
110 /FPURST	155 Ground
111 Reserved	156 D [11]

Pin / Signal Name

157	D [26]
158	Reserved
159	/BG30
160	D [10]
161	D [25]
162	Reserved
163	Ground
164	D [9]
165	D [24]
166	Ground
167	Ground
168	D [8]
169	D [16]
170	Reserved
171	Reserved
172	D [0]
173	D [17]
174	+5 VDC
175	+5 VDC
176	D [1]
177	D [18]
178	+5 VDC
179	+5 VCD
180	D [2]
181	D [19]
182	+5 VDC
183	+5 VDC
184	D [3]
185	D [20]
186	+5 VDC
187	+5 VDC
188	D [4]
189	D [21]
190	Ground
191	Ground
192	D [5]
193	D [22]
194	Ground
195	Ground
196	D [6]
197	D [23]
198	Ground
199	Ground
200	D [7]

CHAPTER 13

LOCAL BUS FORM FACTORS

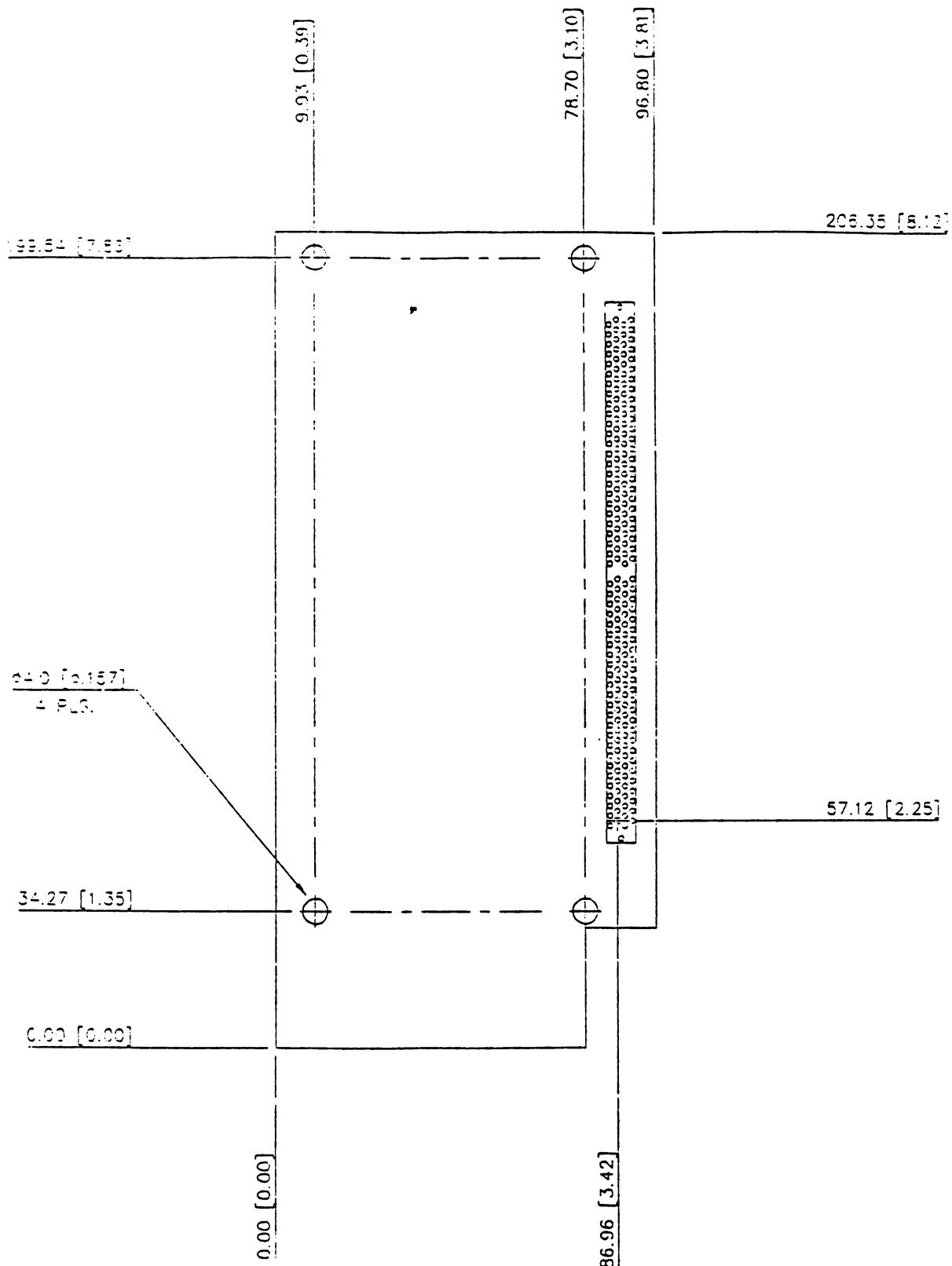
"As you will see from my drawing, everything viewed through them is reversed and appears in mirror image."

-M. C. Escher

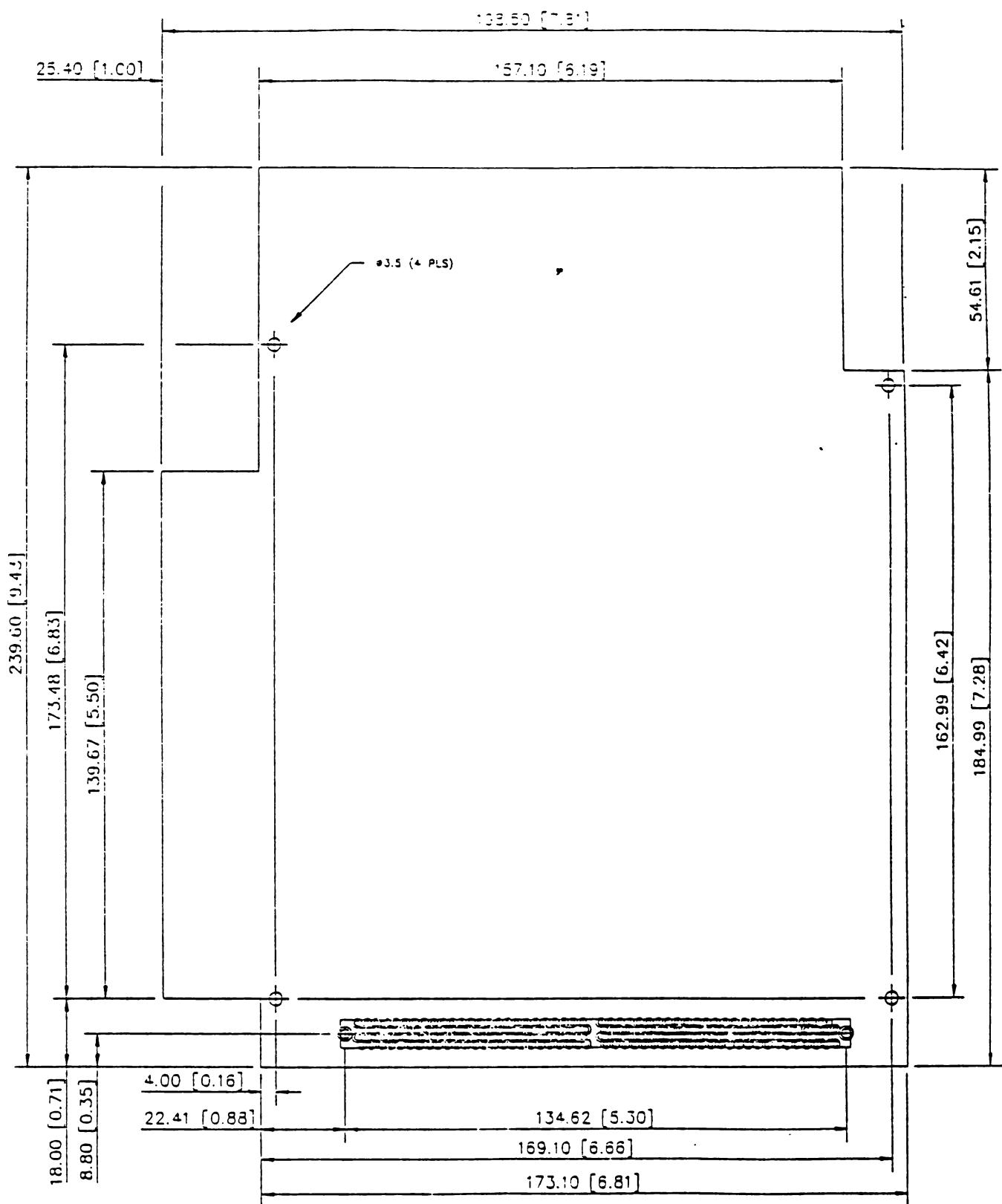
The form factors for a card that plugs into the local bus slot of the A4000 and A3000 are shown on the next two pages. The drawing for the A4000 shows the dimensions of the 68040 coprocessor card supplied by Commodore. Note that these are not the maximum dimensions.

The form factor drawing for the A3000 shows the maximum dimensions for that system. The connector used for the local slot is a KEL 200-pin. Plug in cards use the male edge.

13.1 A4000 Local Bus Slot Form Factor (68040 Coprocessor Board)



13.2 A3000 Local Bus Slot Form Factor



CHAPTER 14

THE AMIGA VIDEO SLOT

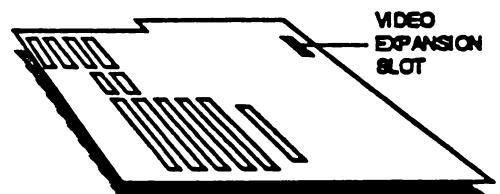
"I like to watch."

-Chauncey Gardener

This section details the signals found on the internal video slot of the Amiga 2000, 3000 and 4000 models. The video slot consists of two in-line, female edge connectors mechanically similar to the slot extension connectors of an IBM PC-AT.

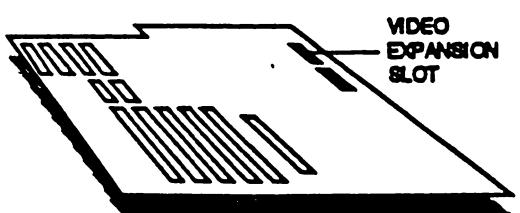
14.1 Evolution of the Amiga Video Slot

Originally the video slot was designed to provide the same functionality as the external 23-pin video connector but in a form that could internally house video boards such as modulators, genlocks and so on. This design required only a single 36-pin connector and the earliest models of the A2000 (4-layer board) do not have a second video connector in-line with the first. Very few of these early models were manufactured.



A2000 video slot with 1 connector (obsolete)

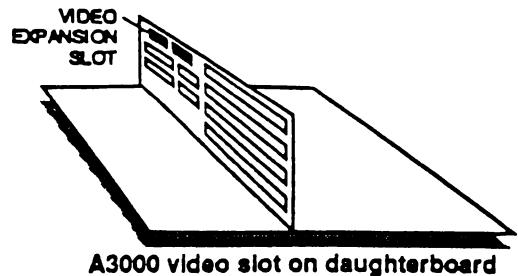
In later versions of the A2000, the video slot was expanded by adding a second 36-pin connector. A similar design was used in the A3000. So, in all A3000s and almost all A2000s, the video slot consists of two in-line, 36-pin female edge connectors. These contain all the signals available on the external video connector, plus all 12 bits of digital color and additional signals.



A2000 video slot with 2 connectors

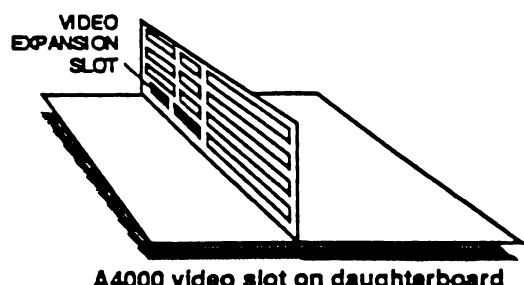
In the A3000, the orientation of the slots was changed because of the addition of the expansion

daughterboard (see figure at right) but they slots work the same way as in the A2000. The arrangement of the expansion daughterboard used in the A3000 allows the video connectors to be in-line with a Zorro III expansion bus connector. Thus there is the potential for creating combination cards or more exotic video applications than previously possible in the Amiga 2000 system.



A3000 video slot on daughterboard

The video slot in the A4000 is very similar to the video slot in the A2000 and A3000 except that the second female edge connector has 54 pins instead of 36. The extra pins on the second connector brings out the additional 12 bits of RGB color available with the AA chip set. As with the A3000, an expansion daughterboard is used allowing the video connectors to be oriented so that they are in line with a Zorro III expansion bus connector. Also note that the video slot is positioned at the bottom of the daughterboard on the A4000 instead of the top as on the A3000.



A4000 video slot on daughterboard

It is possible to create a single plug-in-card that is compatible with all three Amiga models, but there are some mechanical complications. For one thing, the mounting bracket which attaches the video card to the opening at the back of the A4000 has been standardized (Bracket G44 Basic Blank, Globe Manufacturing). The old bracket used in the A2000 and the A3000 is not compatible (it is too large and bumps into the outer casework of the A4000).

The other problem is accessing all the new pins in the second video connector of the A4000. A video card with two sets of fingers designed for the A4000 will not fit into the smaller slots of the A2000 and A3000. The solution is to use three sets of fingers on the video card. That way the card will fit into the A2000 and A3000 while still having access to the new pins in the A4000. See the form factor drawings in chapter 2 of this section for details.

14.2 Video Slot Connector 1

Video slot connector #1 is the slot closest to the rear of the machine (see figures above). It is present on all models of the A2000, A3000 and A4000.

14.2.1 Power Connections

The video slot provides several different voltages for use by video cards. In all Amiga models, there is a single power supply for the main board and all other expansion ports as well as the video slot. The A2000 power supply is rated at 200 watts, the A3000 at 135 watts, the A3000T at 280 watts and the A4000 at 150 watts.

Video Ground

Video supply ground used by all video devices and the internal video circuitry. The Video and Digital grounds are common signals on all Amiga models except for a few very early A2000s (4-layer motherboard). This is available on pins 9, 12, 13, 17, 20, 21, 24 and 32.

Main Supply (+5V).

Main digital level power supply for the video slot. This can supply large currents, on the order of 2 Amps or so. The maximum supply current for the entire A4000 system is 20 Amps for all devices that use +5V including the motherboard. For the A3000, about 0.75 Amps is available for video with 17 Amps available for the whole system. For the A2000, about 2 amps is available for video. Pins 6 and 8 on the A2000 and A3000. Pins 6, 8 and 35 on the A4000.

Negative Supply (-5V).

Negative version of the main supply, for small current loads only; there's a total of 0.2 Amps for the entire A4000 system. (For the A3000, 0.2 Amps; for the A2000, 0.3 Amps.) Pin 31.

High Voltage Supply (+12V).

Higher voltage supply, intended for small loading only; there's total of 4 Amps (8 amp surge) for the entire A4000 system, much of which is normally devoted to floppy and hard disk drive motors. (For the A3000 3 Amps; for the A2000 8 Amps). Pin 10.

14.2.2 Clock Signals

These are various clock signals useful for synchronous timing of video peripherals.

/C1 Clock

For NTSC, this is a 3.58 MHz clock that is synced to the falling edge of the 7.16 MHz system clock. Also known as /CCK in some places. For PAL, these frequencies are 3.55 MHz and 7.09 MHz respectively. Pin 34.

/C4 Clock

For NTSC, this is a 3.58 MHz clock synced to the rising edge of the 7.16 MHz CDAC clock. For PAL, these frequencies are 3.55 MHz and 7.09 MHz respectively. Pin 19.

External Clock (XCLK,/XCLKEN)

The video slot provides for an external system clock, generally used to cause the entire Amiga system to become synchronized to something external. This should be something very close to the 28.64 MHz clock normally used to drive the system; the value used for XCLK can be a somewhat higher frequency, although anything too high will cause memory and other system timings to breakdown. XCLK will only be engaged as the system clock when /XCLKEN is asserted. XCLK is found on pin 33, /XCLKEN is on pin 16. There is no fixed phase relationship between XCLK and internal clocks and video outputs. Video interfaces must synchronize to the output clocks/video.

14.2.3 Video Signals

Access to the video signals are the main point of having the video slot. Most of these are also found on the 23-pin external video connector.

Analog Video

This is the analog RGB output, which consists of Red, Green and Blue signals, each of which generates a 0.7V p-p, 47 ohm terminated analog output. Found, respectively, on pins 7, 11 and 15.

Digital Video

These signals serve as digital output suitable for use with an IBM or Commodore 128-style, 4-bit digital (RGBI), color or monochrome monitor. Each of these outputs is 47 ohm terminated. The pin assignments are Digital Red on pin 29, Digital Green on pin 27, Digital Blue on pin 25, and Digital Intensity on pin 23. The digital Red, Green and Blue are made up of the most significant bits of the 12 or 24 bit video. Intensity is the "middle" bit of the Blue bits.

Blank (BLANK)

Specifies those times during the display when the video should be blanked. A4000 and AA machines only. Pin 26.

Separate Sync (/HSYNC,/VSYNC).

These are the separate, bidirectional, 47 ohm terminated video frame synchronization clocks. The horizontal sync, /HSYNC, is on pin 22; the vertical sync, /VSYNC, is on pin 26. As the names imply, these sync signals are active low.

Composite Sync (/CSYNC)

This is a digital signal that combines /HSYNC and /VSYNC. Pin 14.

Old Composite Sync (COMP SYNC)

Analog sync signal for the composite video. Not present on the A4000 and AA machines. Pin 28.

Burst

NTSC/PAL colorburst. To obtain the correct PAL colorburst signal, the video plug-in card must multiply this signal by 1.25 (i.e., $3.55 \times 1.25 = 4.433$ MHz). Pin 18.

Pixel Switch (/PIXELSW).

Background color indicator (color 0) on a pixel by pixel basis. 47 ohm terminated, /PIXELSW. Pin 30.

14.2.4 Audio Signals

Along with access to video signals, audio signals are also available on video slot #1.

Audio (LINELF, LINERT)

The audio signals are the Left and Right audio channels, on pins 3 and 4, respectively.

This is the audio signal passed through the Amiga's low pass filter. (Raw audio is available on the second video connector, pins 34 and 36.)

14.2.5 New A4000 Signals (Formerly Reserved for Future Expansion)

The video slot implemented in the A2000 and A3000 has pins 1, 2, 5, 35 and 36 reserved for future expansion. In the A4000, all these reserved pins are in use as described below.

PSTROBE

Most of the signals of the parallel port (Centronics connector) are brought out on the second video connector, pins 23-30. The parallel port handshake line, PSTROBE, completes the set. This is available only on pin 36 of the first video connector of the A4000. In the A2000 and A3000, this pin is reserved.

Pixel Synchronous Clock (C280)

This signal on the A4000 provides a clock synchronized with the pixels coming out of Denise allowing digital video boards without the phase-lock loop circuitry required on earlier Amiga models. Pin 5 of the A4000 only.

RGB 16 and RGB 17.

The second video connector provides complete RGB signals on the A2000 and A3000 (4 bits each of R, G and B). But on the A4000, there are 24 bits of RGB color (8 bits each of R, G and B) so there are 12 new signals to bring out. Most of these new signals appear on pins 45-54 of the second video connector but RGB16 (Red bit 0) and RGB 17 (Red bit 1) appear on pins 1 and 2 respectively of the first video connector.

14.3 Video Slot Connector 2 (closest to the front of the machine)

The main purpose of the second video slot is to bring out all the RGB color information from the Denise custom chip. On the A2000 and A3000, there are a total of 12 bits of RGB color information (4 bits each of R, G and B). On the A4000 there are 24 bits of color information (8 bits each of R, G and B).

14.3.1 Additional Video Signals

Composite Video (non-AA)

This is the analog level, monochrome composite video signal also available on the Composite Video jack. Pin 13. Changed to SOG on AA machines.

Sync on Green (SOG)

Sync on green bit programmable via the Copper. A4000 and AA machines only. Pin 13.

Digital Video

Signals on this connector combine with signals on the first connector to provide 12 bits of RGB color information on the A2000 and A3000 or 24 bits of RGB color information on the A4000. The timing of digital video is not tightly specified on the A2000 and A3000. (On those systems, a phase-lock loop circuit is required to synchronize with the digital video signals.) On the A4000, pin 5 of the first video connector (C280) provides a pixel-synchronous clock for digital video boards.

For 12-bit RGB (non-AA), the Red, Green and Blue bits are R0-R3, G0-G3 and B0-B3, respectively. These bits appear on the first connector and the first 36 pins of the second connector. R3, G3 and B3 are the most significant bits of color information.

For 24-bit video (AA), the Red, Green and Blue bits are R0-R7, G0-G7 and B0-B7, respectively. The extra 4 bits per color provide greater color resolution. Therefore, the 4 least significant bits of each color are the new bits and appear on the new pins in the second video connector. Also some unused pins of the original connector space have some of these new bits. Note that the bits that were R0-R3, G0-G3 and B0-B3 on a 12-bit (non-AA) system are now called R4-R7, G4-G7 and B4-B7 on a 24-bit (AA) system.

Light Pen (/LPEN)

This is an input to the Agnus light pen input. This signal should go low in response to the lighting of a pixel on a video display monitor. The Agnus chip latches the raster position that was in effect when the /LPEN signal goes low, so an application can follow the position of a light pen on the screen. Pin 19.

14.3.2 Additional Audio Signals

Raw Audio (RAWLF, RAWRT)

These are the left and right audio channels before they're passed through the low pass filter on output. For many applications, the audio sampling rate is low, and as such requires a low pass filter to be in place at $f_c = 6$ kHz or so, to prevent audio aliasing. However, higher sampling rates are possible, and in such cases a much higher filtering frequency is required for best possible sound. The raw audio, left on pin 33 and right on pin 35, is buffered but unfiltered.

Filter Cutoff (/LED)

This is the /LED port line and also controls the two pole low pass filter on the standard audio channels. When asserted, the filter is in place; when negated the filter is bypassed. This is an input to the connector, useful to allow any Audio/Video card to monitor the audio filtering state. Pin 31.

14.3.3 Additional Grounding

Digital/Video Ground (GROUND).

These pins provide additional grounding for digital or video based devices. Pins 1, 5, 9, 12, 22, and 32.

Audio Ground (AGND)

These pins provide grounding in common with the separate onboard audio ground. Pins 34, 36.

14.3.4 Additional Clock Signals

These are various clock signals useful for synchronous timing of video peripherals.

CDAC Clock.

For NTSC, this is a 7.16 MHz clock that leads the 7.16 MHz system clock by about 70ns (90 degrees). Pin 15. For a PAL system, this is 7.09 MHz.

/C3 Clock

For NTSC, this is a 3.58 MHz clock that is synched to the rising edge of the 7.16 MHz system clock. Also known as /CCKQ in some places. For a PAL system, this is 3.55 MHz. Pin 17.

Timer Time Base (TBASE)

This is the real time clock time-base input, either 50Hz or 60Hz, depending on the country involved and the setting of the Time Base Jumper. The jumper can select either line frequency or vertical synchronization as the clock's time base. Pin 14.

14.3.5 Parallel Port Connections

Most of the signals from the bidirectional parallel port (Centronics connector) are available on the second video slot. (Note that PBUSY is on pin 18 of the first video connector in the A4000 only.)

8 Bit Parallel Port (PDO-PD7)

The 8 bit bidirectional parallel port most commonly used to drive a Centronics interface printer externally is accessible here. It can be used to control various aspects of a complex video interface device. The port lines PDO-PD7 are on pins 23 to 30.

Parallel Port Handshake (/ACK).

This is the acknowledge (/ACK) input, the same as the acknowledge input to the parallel port. Driving this with an output from a Video Card can cause a level 2 interrupt to occur through the 8520 CIA device this is connected to, based on the programming of an 8520 register. On pin 20.

Other Port Lines (BUSY, POUT, SEL).

Connector pins 18 (BUSY) and 16 (POUT) are general purpose I/O signals that together can also function as a synchronous serial data port driven by an 8520 CIA device. In normal printer use, the BUSY signal is used to indicate the printer paper is out. For serial port usage, BUSY is the serial clock, POUT is the serial data line. These should be driven with open collector devices if the Video Card uses them as inputs to the 8520. The SEL signal, on pin 21, is a general purpose I/O port usually used as a device select signal on the parallel port.

14.4 Video Slot Pinout

Video Connector 1 (closest to the rear of the machine)

Amiga 4000	Amiga 3000 and Amiga 2000	Early Amiga 2000 (obsolete)
1 R0	1 Reserved	1 Reserved
2 R1	2 Reserved	2 Reserved
3 Filtered Audio Left	3 Filtered Audio Left	3 Filtered Audio Left
4 Filtered Audio Right	4 Filtered Audio Right	4 Filtered Audio Left
5 C280	5 Reserved	5 Reserved
6 +5 VDC	6 +5 VDC	6 +5 VDC
7 Analog Red	7 Analog Red	7 Analog Red
8 +5 VDC	8 +5 VDC	8 +5 VDC
9 Video Ground	9 Video Ground	9 Video Ground
10 +12 VDC	10 +12 VDC	10 +12 VDC
11 Analog Green	11 Analog Green	11 Analog Green
12 Video Ground	12 Video Ground	12 Video Ground
13 Video Ground	13 Video Ground	13 Video Ground
14 /CSYNC	14 /CSYNC	14 /CSYNC
15 Analog Blue	15 Analog Blue	15 Analog Blue
16 /XCLKEN	16 /XCLKEN	16 /XCLKEN
17 Video Ground	17 Video Ground	17 Video Ground
18 BURST	18 BURST	18 BURST
19 /C4 Clock	19 /C4 Clock	19 /C4 Clock
20 Video Ground	20 Video Ground	20 Video Ground
21 Video Ground	21 Video Ground	21 Video Ground
22 /HSYNC (47 ohm)	22 /Hsync (47 ohm)	22 /Hsync (47 ohm)
23 B4 = DI (47 ohm)	23 B0 = DI (47 ohm)	23 B0 = DI (47 ohm)
24 Video Ground	24 Video Ground	24 Video Ground
25 B7 = DB (47 ohm)	25 B3 = DB (47 ohm)	25 B3 = DB (47 ohm)
26 /VSYNC (47 ohm)	26 /Vsync (47 ohm)	26 /Vsync (47 ohm)
27 G7 = DG (47 ohm)	27 G3 = DG (47 ohm)	27 G3 = DG (47 ohm)
28 BLANK	28 COMP SYNC(Analog)	28 COMP SYNC(Analog)
29 R7 = DR (47 ohm)	29 R3 = DR (47 ohm)	29 R3 = DR (47 ohm)
30 /PIXELSW (47 ohm)	30 /PIXELSW (47 ohm)	30 /PIXELSW (47 ohm)
31 -5 VDC	31 -5 VDC	31 -5 VDC
32 Video Ground	32 Video Ground	32 Video Ground
33 XCLK	33 XCLK	33 XCLK
34 /C1 Clock	34 /C1 Clock	34 /C1 Clock
35 +5 VDC	35 Reserved	35 Reserved
36 PSTROBE	36 PSTROBE	36 Reserved

Video Connector 2 (closest to the front of the machine)

1 Ground	1 Ground
2 R4	2 R0
3 R5	3 R1
4 R6	4 R2
5 Ground	5 Ground
6 G4	6 G0
7 G5	7 G1

Video Connector 2 (closest to the front of the machine)

Amiga 4000	Amiga 3000 and Amiga 2000
8 G6	8 G2
9 Ground	9 Ground
10 B5	10 B1
11 B6	11 B2
12 Ground	12 Ground
13 SOG	13 Composite Video
14 TBASE	14 TBASE
15 CDAC Clock	15 CDAC Clock
16 POUT	16 POUT
17 /C3 Clock	17 /C3 Clock
18 BUSY	18 BUSY
19 /LPEN	19 /LPEN
20 /ACK	20 /ACK
21 SEL	21 SEL
22 Ground	22 Ground
23 PD0	23 PD0
24 PD1	24 PD1
25 PD2	25 PD2
26 PD3	26 PD3
27 PD4	27 PD4
28 PD5	28 PD5
29 PD6	29 PD6
30 PD7	30 PD7
31 /LED	31 /LED
32 Ground	32 Ground
33 Raw Audio Left	33 Raw Audio Left
34 Audio Ground	34 Audio Ground
35 Raw Audio Right	35 Raw Audio Right
36 Audio Ground	36 Audio Ground
37 Reserved	
38 Reserved	
39 Ground	
40 Ground	
41 Reserved	
42 Reserved	
43 Ground	
44 Ground	
45 R2	
46 R3	
47 G1	
48 G2	
49 G3	
50 G4	
51 B0	
52 B1	
53 B2	
54 B3	

CHAPTER 15

GENLOCK INTERFACE GUIDELINES

"One man's 'magic' is another man's engineering."

-Robert Heinlein

This chapter describes the reset mechanism of the Video Beam Counters on the current version of the Agnus chip when an external device (i.e. genlock) applies sync pulses on the HSY* and VSY* pins. The Commodore-recommended method of interfacing genlock devices to the Amiga for current and future compatibility is described here.

15.1 Hardware Interpretation of HSY* and VSY* Reset Pulses

Agnus is configured to accept external syncs, by setting the ERSY bit in the BPLCON0 register. After setting the ERSY bit the horizontal and vertical counters will respond as described below.

Horizontal Counter, NTSC Mode. If the external HSY* reset is not applied at the end of a "short line", the next line will be a "long line", as normal. If no HSY* is applied at the end of a "long line", the horizontal counter will roll-over and is held reset at count 00. The counter will resume counting once the external HSY* pulse is applied.

If the external HSY* reset is applied at the end of a "short line", the following line is forced to be another "short line".

If the counter reaches the end of a "long line" when the external HSY* is asserted, the next line is a "short line" and will start at count 01 (as opposed to count 00). Otherwise, if no reset occurs the counter will roll-over to count 00 and stop.

Horizontal Counter, PAL Mode. In PAL mode the counter will operate with "short lines" at all times ("longlines" are disabled). If the external HSY* is not applied at the end of a line, the horizontal counter will roll-over to the beginning of the next line and is held at count 00, until the next HSY* pulse occurs.

If the external HSY* is applied when the counter reaches the end of a line, the next line will start at count 01, skipping count 00.

Vertical Counter, Interlaced Mode. In interlaced NTSC mode, if an external VSY* pulse is applied, the vertical counter will be reset to count 000 (first line) and the long-field condition is set at the beginning of the next horizontal line (at horizontal count 02).

In interlaced PAL mode, if an external VSY* pulse is applied the long-field condition is also set. Note that for both NTSC and PAL systems, if the external VSY* pulse is not asserted during the end of a short (or long) field, the vertical counter will roll-over and start a long (or short) field sequence.

Vertical Counter, Non-Interlaced Mode. In non-interlace mode the vertical counter is set to operate with either long or short fields, depending on the last value that the software wrote to the FRAME bit. In this mode, whenever an external VSY* pulse is applied, the counter is reset to line count 000 (first line) at the beginning of the next horizontal line.

15.2 Pulse Duration

The width of the active-low vertical (VSY*) pulse should be less than or equal to one line duration (63.556 uS NTSC, 63.999 uS PAL) but greater than one-half line duration (31.77 uS NTSC, 31.999 uS PAL) and should be asserted during the beginning of a horizontal line, at which time the vertical logic is triggered and the first line is started. The width of the active-low horizontal (HSY*) pulse should be greater than or equal to eight hi-res pixels (0.558 uS for NTSC or 0.563 uS for PAL) for proper operation.

15.3 NTSC and PAL Genlock Interface Guidelines

In order to synchronize the Amiga to an external source, the genlock device must provide reset pulses to the Amiga's sync lines which have the effects described in the section above. This section describes the proper method of applying these reset pulses.

For NTSC Amiga models, the genlock device must provide the Amiga with horizontal and vertical reset pulses with the following rates:

- HSY* line:** Active-low pulse of proper duration every two lines
(i.e. 127.11 uS or 7.8671 KHz).
- VSY* line:** Active-low pulse of proper duration every two fields
(i.e. 33.36 mS or 29.97 Hz).

Note that after the VSY* reset pulse the next field will be an even field (long field).

For PAL Amiga models, the genlock device must provide the Amiga with horizontal and vertical reset pulses with the following rates. Both reset pulses must be generated regardless of whether or not source video is being input to the genlock device!

- HSY*** line: Active-low pulse of proper duration every line (i.e. 63.99 uS or 15.625 KHz).
- VSY*** line: Active-low pulse of proper duration every two fields (i.e. 39.99 mS or 25.0 Hz).

Note that after the VSY* reset pulse the next field will be an even field (long field).

In addition to providing the Amiga with the proper horizontal and vertical reset pulses, the genlock device must also provide the Amiga with a system clock that is synchronized with the external video. One method of doing this would be to multiply the video source line duration up to the Amiga's system clock frequency, thus making a line-locked clock that will also be free-running at the correct frequency when no video is present. For correct operation of the Amiga as a multitasking system, the proper system clock frequency appearing on the XCLK pin must be the following for NTSC and PAL Amiga systems:

NTSC XCLK:28.63636 Mhz
PAL XCLK: 28.375156 Mhz

Note that some PAL genlock designers provide the Amiga's XCLK input with a 28.250 Mhz clock frequency. This is not recommended for proper Amiga operation!

CHAPTER 16

VIDEO BOARD FORM FACTORS

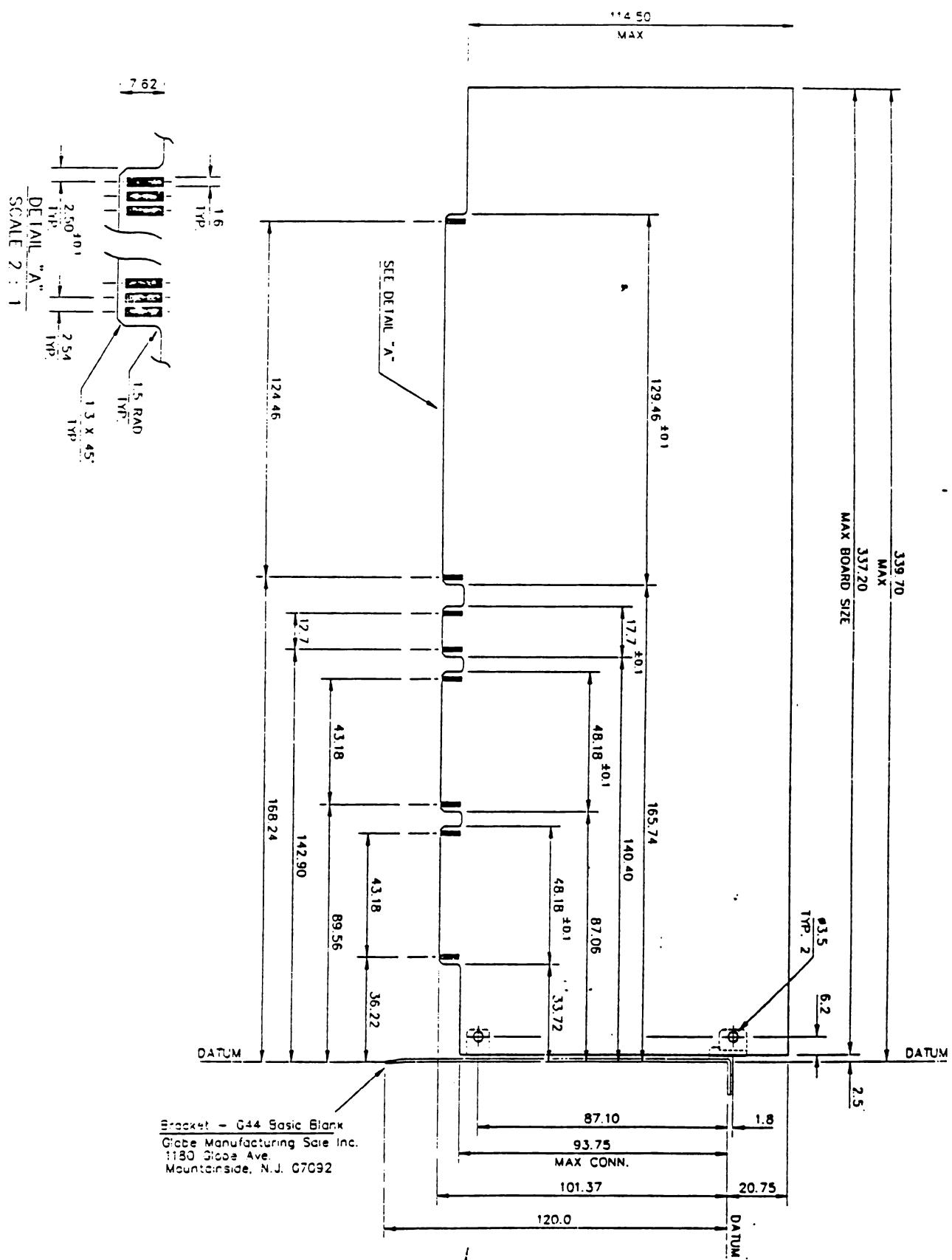
"The occupation of space is the real and final fact"

-W. H. Auden

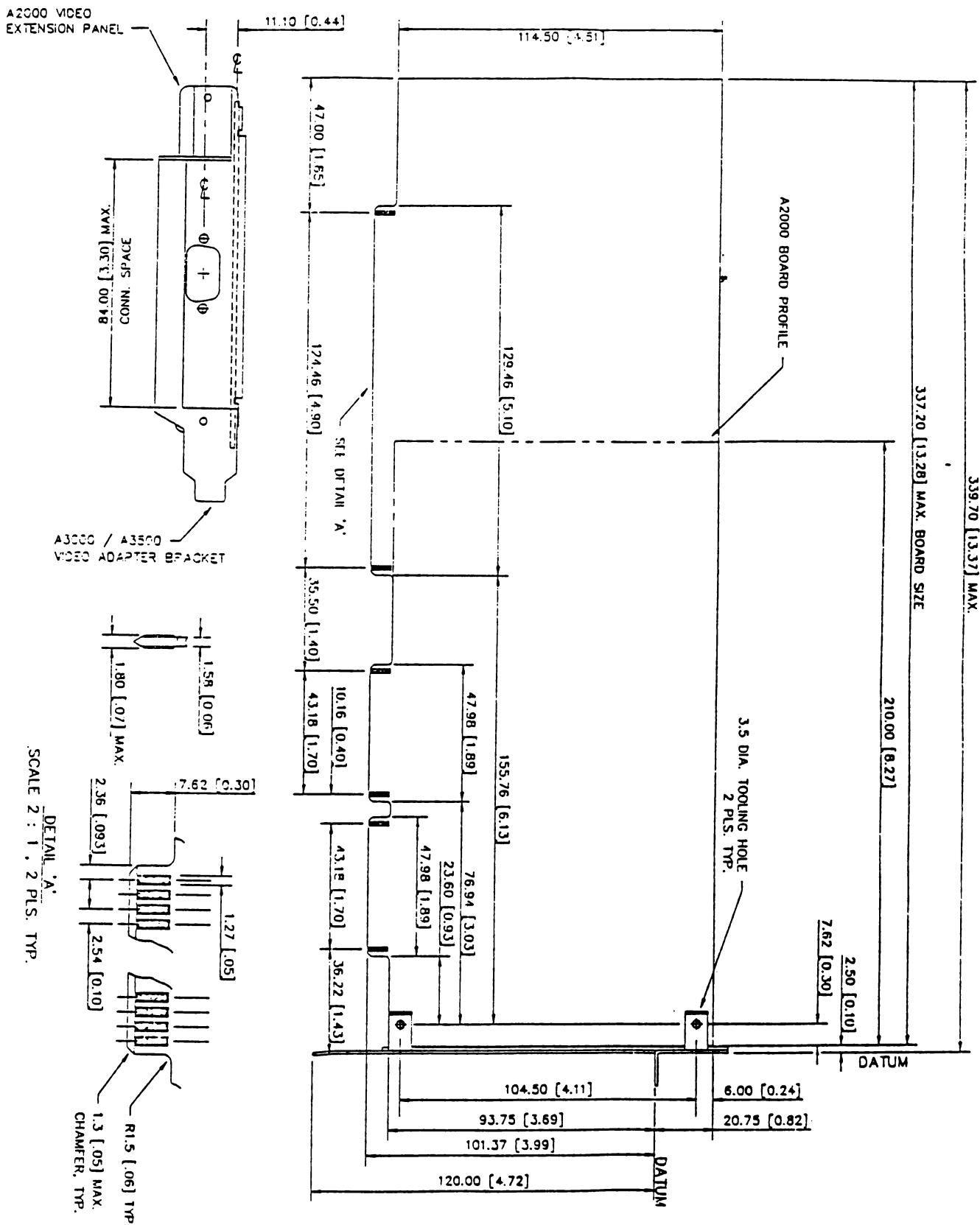
The form factors for a video board for the A3000 and A4000 are shown on the next two pages. The profile for an A2000 video board is also shown in the form factor drawing for the A3000 for those hardware designers who want to create boards that are backward compatible with the A2000.

Keep in mind that there are differences in the way a video board's bracket attaches the board to the opening at the back of the Amiga. Because of these differences, brackets designed for the A2000 or A3000 may be too large to fit within the casework of the A4000. The A4000 video slot uses a standard bracket available from Globe Manufacturing (Bracket G44 Basic Blank, Globe Manufacturing, 1180 Globe Ave., Mountainside NJ 07092).

Amiga 4000 Video Board Form Factor



Amiga 3000 Video Board Form Factor



GLOSSARY

"Thus the American, on his linguistic side, likes to make his language as he goes along"

-H. L. Mencken

G.1 Glossary

The reader may be unfamiliar with a number of terms used in this document. Every effort has been made to include all such terms here.

address	A byte-numbered memory location. The Zorro II bus is based on a 24 bit address, the Zorro III bus on a 32 bit address.
arbitration	The unambiguous selection of one request out of a number of possible simultaneous requests for a resource. There are two kinds of arbitration in a Zorro III system; bus arbitration and quick interrupt arbitration.
asserted	The active state of a state, regardless of its logic sense.
atomic cycle	A cycle or set of cycles that are uninterruptable, and thus treated as a unit; both Multiple Transfer and LOCKed cycles are considered atomic under the Zorro III bus.
AUTOCONFIG®	From "automatic configuration", the Zorro bus specification for how software and hardware cooperate to permit PIC addresses to be set by software and PIC type information to be determined by software. This is explained in Chapter 8, and in the <i>A500/A2000 Technical Reference Manual</i> , available from Commodore-Amiga.

backplane	The cage or motherboard subsection into which PICs are inserted. The Amiga 2000 and Amiga 3000 computers have integral backplanes, the Amiga 500 and Amiga 1000 computers require add-on backplane cages for Zorro II compatibility.
burst	A short name for Multiple Transfer Cycle mode. Essentially, within one full Zorro III cycle there can be any number of Multiple Transfer Cycles. Each full cycle has a complete 32 bit address supplied and a complete 32 bit datum transferred. Each burst cycle supplies only the 8 bit page address, but transfers a complete 32 bit datum faster than the standard full cycle would allow.
bus cycle	One complete bus transaction, indicated by the assertion of least one cycle strobe. For any single bus cycle, there is one address, one data value, one data direction, and one cycle type in effect.
bus hogging	When a bus master takes over the bus for an undue amount of time. The Zorro II bus leaves it completely up to the individual PIC to avoid bus hogging; the Zorro III bus schedules PICs with the bus controller to evenly distribute the bus load.
bus starvation	When a master can't get access to the bus, it is said to be starved. On the Zorro II bus, two busy masters can completely starve a third master. Complete starvation is impossible on the Zorro III bus, though a bus hogging Zorro II card can cause similar symptoms.
byte	A collection of eight signals into a logical group, and the smallest independently addressable quantity on the Zorro bus.
clock	A free running signal driven at a fixed frequency to the bus, used mainly for clocking state machines on Zorro II cards.
cycle strobe	A bus signal that defines the boundary of a bus cycle; the Zorro II and Zorro II modes on a Zorro III bus each have their own cycle strobes. The current bus master always asserts the cycle strobes.
data	The contents of a memory location. The main purpose of a bus cycle is to transfer data between two locations. The Zorro II bus is based on a 16 bit data path, the Zorro III bus is based on a 32 bit data path.
DMA	Direct Memory Access; devices that have direct access to Zorro III slaves are said to have DMA capability. These devices are also called masters.

DMA latency	This is the time between a bus request and a bus grant as seen by a PIC wishing to become bus master.
device	A PIC, e.g., a Zorro bus master or bus slave.
grant	The result of an arbitrated set of requests is a single grant; there are grants given for both the bus and quick interrupts.
Guinness	Attitude adjustment tonic, from Ireland. Said by some to be vital for sanity, if not normal human life.
hidden cycles	Cycles that occur on the local bus of a system, but can't be seen by devices on the expansion bus.
high	A signal driven to a logical +5V state is said to be high.
interrupt	An asynchronous line driven by a PIC to notify the CPU of some event, usually some hardware event governed by that PIC.
local bus	The main system bus of an Amiga computer is called the local bus. In general, the main CPU, video chips, chip memory, and any other built-in resources are on the local bus. The bus controller sits on both the local and expansion buses and manages the communications between them.
longword	Based on the Motorola conventions, a longword is equal to 4 bytes.
low	A signal driven to a logical +0V state is said to be low.
master	The device currently generating addresses for the expansion bus. There is only one master on the bus at a time, this being insured by the bus arbitration logic. The master also drives data on writes, the read, cycle, and data strobes, and several other signals.
motherboard	The main system circuit board for any Amiga computer. Resources on the local bus of a machine are often called motherboard resources.
negated	The inactive state of a signal, regardless of its logic sense.
nybble	A collection of four bits; one half of a byte. AUTOCONFIG® ROMs are physically nybble-wide.
paragraph	A sequence of closely related sentences, generally expressing and supporting one succinct idea. This term has no special computerese meaning in any rational, modern system.
PIC	Plug In Card. Any Amiga expansion card is called a PIC for short.

request	Asking for the use of some resource; the Zorro III bus has two kinds of requests, bus requests and quick interrupt requests.
slave	The device currently responding to the address on the expansion bus. There is only one slave on the bus at a time; an error is signalled by the bus collision detect logic if multiple slaves respond to the same address. The slave also drives data on reads, the transfer acknowledge strobe, and several other signals.
slot	A physical port on a Zorro backplane, which supplies independent /SLAVEN, /BRN, and /BGN lines, chained /CFGIN _N and /CFGOUT _N lines, and is mechanically manifested as a 100 pin single-piece connector.
termination	Circuitry attached to a bus signal in order to minimize annoying analog things like ringing, reflections, crosstalk, and possibly random logic conditions which can arise when a bus is undriven.
timeout	A bus cycle terminated by the bus controller instead of by a responding slave device. If no slave responds to a bus cycle within a reasonable time period, the bus controller will terminate the cycle to prevent lockup of the system.
tri-state	A signal driven to a high impedance condition is said to be tri-stated.
word	Based on the Motorola conventions, a word is equal to 2 bytes.
Zorro	The name given to the Amiga bus specification. "Zorro I" refers to the original design for A1000 backplane boxes, "Zorro II" refers to the modification to this specification used for the A2000 and compatible backplanes, and "Zorro III" refers to the Zorro II compatible bus specification first used in the Amiga 3000 computer and later in the A4000.



**This was brought to you
from the archives of**

<http://retro-commodore.eu>