SCHEMATICS
AND
EXPANSION
SPECIFICATIONS

June 9, 1986

commodore
COMPUTERS

SCHEMATICS
AND
EXPANSION
SPECIFICATIONS

June 9, 1986

DISCLAIMER

THE INFORMATION IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED. THE ENTIRE RISK AS TO THE RESULTS AND PERFORMANCE OF THE PROGRAM IS ASSUMED BY YOU. YOU (AND NOT THE DEVELOPER OR COMMODORE-AMIGA, INC. OR ITS DEALERS) ASSUME THE ENTIRE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION. FURTHER, COMMODORE-AMIGA DOES NOT WARRANT, GUARANTEE, OR MAKE ANY REPRESENTATIONS REGARDING THE USE OF, OR THE RESULTS OF THE USE OF, THE INFORMATION IN TERMS OF CORRECTNESS, ACCURACY, RELIABILITY, CURRENTNESS, OR OTHERWISE; AND YOU RELY ON THE PROGRAM AND THE RESULTS SOLELY AT YOUR OWN RISK. IN NO EVENT WILL COMMODORE-AMIGA, INC. BE LIABLE FOR DIRECT, INDIRECT, INCIDENTAL OR CONSEQUENTIAL DAMAGES RESULTING FROM ANY DEFECT IN THE INFORMATION EVEN IF IT HAS BEEN ADVISED OF THE POSSIBLITY OF SUCH DAMAGES. SOME LAWS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF IMPLIED WARRANTIES OR LIABLITIES FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THE ABOVE LIMITATION OR EXCLUSION MAY NOT APPLY.

Warning:

When designing hardware to interface to the Amiga, please use Timing Guidelines given in expansion documentation.

Schematics represent current machine which is subject to change wihout notice.

PAL EQUATIONS

PAL16L8
DPALCAS 252128-02 REV 3
15-MAY-85
Copyright 1985 Commodore-Amiga Inc.,

/ARW A20 A19 /PRW /UDS /LDS /ROME /RE /RGAE GND /DAE /ROM01
/C1 /RRW LCEN UCEN /CDR /CDW /NC1 VCC

```
ROMO1   =ROME*A20*A19*/PWR + ROME*/A20*/A19*/PRW
CDR     =RE*/PWR*/C1 +  RGAE*/PRW*/C1 + CDR*LDS + CDR*UDS
CDW     =RE*PRW + RGAE*PRW + CDW*/C1 + /DAE*/UDS*/UCEN
/UCEN   =/DAE*/RE*/UCEN + /DAE*/RE*C1 +/DAE*UDS*C1 + /DAE*/UDS*/UCEN
/LCEN   =/DAE*/RE*/LCEN + /DAE*/RE*C1 +/DAE*/LDS*C1 + /DAE*/LDS*/LCEN
RRW     =RE*PRW + DAE*ARW*C1 + RRW*DAE
```

Description

In RRW equation, may not need RRW*/C1 hold term, but doesn't hurt

PRELIMINARY

PAL16L8
DPALEN FOR RAMROM BOARD 252128-01 REV 2
15-MAY-85
Copyright 1985 Commodore-Amiga Inc.

```
A23 A22 A21 /AS /DBR OVL /OVR  XRDY /C3 GND / C1 /VPA
/MYRAME /DAE /RGAE /RE /DTACK / BLS NC VCC

IF(/OVR) VPA        =AS* A23*/A22* A21
MYRAME =  AS*/DTACK*A23*A22*A21*/OVR*C1*/C3 +
           AS*/DTACK*/A23*/A22*/A21*/OVR*OVL*C1*/C3 +
          MYRAME*C1 +
          MYRAME*C3 +
RE       =  /DBR*AS/DTACK*/A23*/A22*/A21*/OVL* C1*/C3*/OVR +
                                    RE* C1 + RE* C3

IF(/OVR) DTACK =
         AS*/A23*/A22* A21* XRDY +
         AS*/A23* A22* XRDY +
         AS* A23*/A22*/A21* XRDY +
         MYRAME*XRDY*C3 +
         RE*C3 +
         RGAE*C3 + DTACK*AS*XRDY
RGAE     =/DBR*AS*/DTACK* A23* A22*/A21* C1*/C3*/OVR +
                              RGAE*C1 + RGAE* C3
DAE      =DBR*                                  C1*/C3 +
                               DAE* C1 + DAE * C3
BLS      = AS*/DTACK*/A23*/A22*/A21*/OVL* C1*/C3*/OVR +
           AS*/DTACK* A23* A22*/A21        C1*/C3*/OVR +
           BLS*C1 +BLS*C3
```

Description
This is the RAMROM board version of the PALEN PAL of the main board. When it returns to the main board, rethink the MYRAME and DTACK stuff.

PRELIMINARY

PAL 16L8
DAUGCAS ram/rom CAS 252128-04 REV 1
16-APR-85
Copyright 1985 Commodore-Amiga Inc.

/SPROM A18 A17 /PRW /UDS /LDS /RE /RES /ROME GND /C1 /BERR
/WPRO /RRW /LCEN /UCEN /CDR /CDW /ROMO1 VCC

```
ROMO1   =ROME*/A18*/WPRO*/SROM*/PRW
CDR     =CDR*LDS + CDR*UDS + RE*/PRW*/C1*A18 + RE*/PRW*/C1*/A18*WPRO
         + RE*/PRW*/C1/A18*SROM
CDW     = RE*PRW + CDW*/C1
UCEN    =UCEN*/C1 + RE*UDS*A18  RE*UDS*/A18*WPRO +
         RE*UDS*/A18*SROM
LCEN    =LCEN*/C1 + RE*LDS*A18 + RE*LDS*/A18*WPRO +
          RE*LDS*/A18*SROM
RRW     =RE*PRW*A18/WPRO*/SROM
BERR    =WPRO*PRW*RE
WPRO    =WPRO*/RES + PRW*RE/A18
```

Description

This is the CAS PAL for the RAMROM board

```
PAL16L8
DAUGEN PAL FOR RAMROM BOARD 252128-03 REV 2
15-MAY 85
Copyright 1985 Commodore-Amiga Inc.

A23 A22 A21 /AS /DBR OVL /OVR XRDY /C3 GND /C1 /CNT
/RE DAE /NC1 A19 /DTACK A20 /ROME VCC

/CNT      =DAE*C1*C3

RE        =AS*/DTACK*A23*A22*A21*A20*A19*/OVR*C1*/C3  +
            AS*/DTACK*/A23*/A22*/A21*/A20*/A19*/OVR*OVL*C1*/C3  +
           RE*C1 + RE*C3

/DAE      =/C1*/C3 + RE +
            AS*/DTACK*A23*A22*A21*A20*A19*/OVR*OVL*C1*/C3 +
            AS*/DTACK*/A23*/A22*/A21*/A20*/A19*/OVR*OVR*C1*/C3

ROME      =AS*A23*A22*A21*A20*A19*/OVR +
           AS*/A23*/A22*/A21*/A20*/A19*/OVR*OVL
```

Description

This is the daughter RAM/ROM DAUGEN PAL

PORT DESCRIPTORS

Parallel Interface Connector Specification

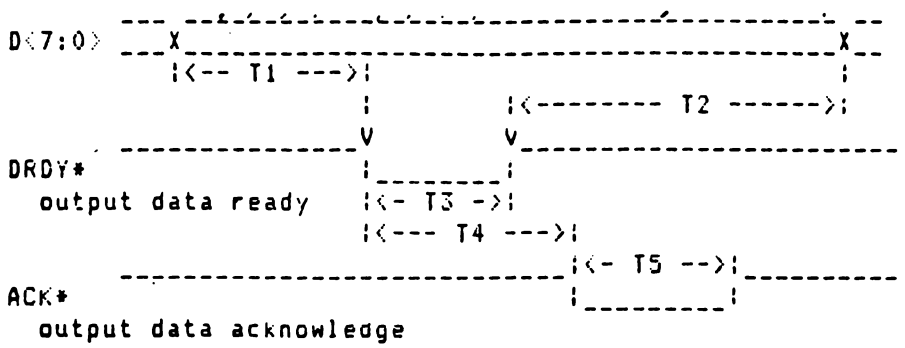
The 25 pin D type connector with pins (DB25P) at the rear of the Amiga is nominally used to interface to parallel printers. In this capacity, data flows from the Amiga to the printer. This interface may also be used for input or bidirectional data transfers. The implementation is similar to Centronics, but the pin assignment and drive characteristics vary significantly from the specification.

Parallel Connector Pin Assignment (J8)

Pin	Name	dir	Notes
1	DRDY*	O	Output data ready signal to parallel devices in output mode, used in conjunction with ACK* (pin 10) for a 2 line asynchronous handshake. Functions as input data accepted from Amiga in input mode (similar to ACK* in output mode). See timing diagrams in following section.
2	D0	I/O	
3	D1	I/O	
4	D2	I/O	
5	D3	I/O	D0–D7 comprise an 8 bit bidirections bus for communication with parallel devices, nominally, a printer.
6	D4	I/O	
7	D5	I/O	
8	D6	I/O	
9	D7	I/O	
10	ACK*	I	Output data acknowledge from parallel device in output mode, used in conjunction with DRDY* (pin 1) for a 2 line asynchronous handshake. Functions as input data ready from parallel device in input mode (similar to DRDY* in output mode). See timing diagram in following section. The 8520 can be programmed to conditionally generate a level 2 interrupt to the 68000 whenever the ACK* input goes active.
11	BUSY	I/O	This is a general purpose I/O pin shorted to a serial data I/O pin (serial clock on pin 12). Note: Nominally used to indicate printer buffer full.
12	POUT	I/O	This is a general purpose I/O pin shored to a serial clock I/O pin (serial data on pin 11). Note: Nominally used to indicate printer paper out.

Pin	Name	Dir	Notes
13	SEL	I/O	This a general purpose I/O pin. Note: Nominally a select output from the parallel device to the Amiga.
14	GND		
15	GND		
16	GND		
17	GND		
18	GND		
19	GND		
20	GND		
21	GND		
22	GND		
23	+5V		100 ma maximum. ***WARNING +5V.***
24	---		
25	RESET*	0	Amiga system reset

Parallel Connector Interface Timing, Output Cycle

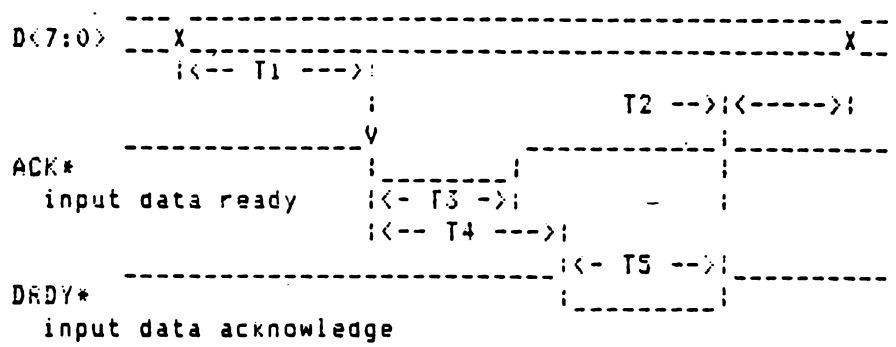


	microseconds min	typ	max	
T1:	4.3		5.3	Output Data setup to ready delay
T2:	nsp		upc	Output Data hold time.
T3:	nsp	1.4	nsp	Output Data ready width
T4:	0		upc	Ready to acknowledge delay
T5:	nsp		upc	Acknowledge width

nsp = not specified
upc = under program control

PRELIMINARY

Parallel Connector Interrface Timing, Input Cycle



	microseconds min	type	max	
T1:	0		upc	Input data setup time.
T2:	nsp		upc	Input data hold time.
T3:	nsp		upc	Input data ready width
T4:	upc		ups	Input data ready to data acknowledge delay
T5:	nsp	1.4	nsp	Input data acknowledge width

nsp = not specified
upc = under program control

PRELIMINARY

The 25 pin D type connector with sockets (DB25S) is used to interface to RS232C standard signals.

WARNING: Pins 14, 21 and 23 carry power. Do not connect to these pins inadvertantly as they can permanently damage external equipment. Also, pins 15–18, 23–25 carry non-standard signals and should not be inadvertantly connected.

NEVER use a fully wired 25 line cable!

Serial Interface Connector Pin Assignment (J6)

Pin	Name	Dir	Std	Notes
1	FGND		y	frame ground
2	TXD	0	y	transmit data
3	RXD	I	y	receive data
4	RTS	0	y	request to send
5	CTS	I	y	clear to send
6	DSR	I	y	data set ready
7	GND		y	signal ground
8	CD	I	y	carrier detect
9	---		n	
10	---		n	
11	---		y	
12	---		n	
13	---		n	
14	-5V		n*	50 ma maximum
15	AUDO	0	n*	audio output
16	AUDI	I	n*	audio input
17	EB	0	n*	716 KHz
18	INT2*	I	y	OPEN COLLECTOR Amiga Interrupt level 2
19	---		n	
20	DTR	0	y	data terminal ready
21	+5V		n*	100 ma maximum
22	---		n	
23	+12V		n*	50 ma maximum
24	C2*	0	n*	3.58 MHz
25	RESB*	0	n*	Amiga system reset

n*:
WARNING: Pins 14, 21 and 23 carry power. Do not connect to these pins inadvertantly as they can permanently damage external equipment. Also, pins 15–18, 23–25 carry non-standard signals and should not be inadvertantly connected.
NEVER use a fully wired 25 line cable!

Serial Interface Connetor Timing

Maximum operating frequency is 19.2 KHz. Refer to EIA standard RS232C for operating and installation specification. A rate of 31.25 KHz will be supported through the use of a MIDI adapter.

Modem control signals (CTS, RTS, DTR, DSR, CD) are completely under software control. The modem control lines have no hardware affect on and are completely asynchronous to TXD and RXD.

Serial Interface Connector Electrical Characteristics

Outputs	
Vol: -2.5 to -5.5V	Negative output voltage range
Voh: 8 to 13.2V	Positive output voltage range
Io: 10 ma max	Output current
Inputs	
Vih: 2.5 to 25V	Positive input voltage range
Vil: -25 to .5V	Negative input voltage range
Vhy: 1V typical	Input hysteresis voltage
Ii: 10 ma max	Input current

Unconnected inputs are interpreted the same as negative input voltages.

External Disk Interface Connector Specification

The 23 pins D type connector with sockets (DB23S) at the rear of the Amiga is nominally used to interface to MFM devices.

External Disk Connector Pin Assignment (J7)

Pin	Name	Dir	Notes
1	RDY*	I/O	If motor on, indicates disk installed and up to speed If motor not on, Identification mode. See below.
2	DKRD*	I	MFM input data to Amiga
3	GND		
4	GND		
5	GND		
6	GND		
7	GND		
8	MTRXD*	OC	Motor on data, clocked into drive's motor on flip flops by the active transistion of SELxB*. Guaranteed setup time is 1.4 usec. Guaranteed hold time is 1.4 used.
9	SEL2B*	OC	Select drive 2
10	DRESB*	OC	Amiga system reset. Drive should reset their motor on flip flops and set their write protect flip flops.
11	CHNG*	I/O	Note: Nomially used as an open collector input. Drives change flop is set at powe up or when no disk is not installed. Flop is reset when drive is selected and the head stepped, but only is a disk is installed.
12	+5V		270 ma maximum; 410 ma surge When below 3.75V, drives are required to reset their motor on flops, and set their write protect on flops.
13	SIDEB*	O	Side 1 if active, side 0 if inactive
14	WPRO*	I/O	Asserted by selected, write protected disk
15	TKO*	I/O	Asserted by selected drive when read/write head is positioned over track 0
16	DKWDB*	OC	Write gate (enable) to drive
17	DKWDB*	OC	MFM output data from Amiga
18	STEPB*	OC	Selected drive steps one cylinder in the direction indicated by DIRB.
19	DIRB	OC	Direction to step the head. Inactive to step towards center of disk (higher humbered tracks).
20	SEL3B*	OC	Select drive 3
21	SEL1B	OC	Select drive 1
22	INDEX*	I/O	Index is a pulse generated once per disk revolution, between the end and geginning of cyliners. The 8520 can be programmed to conditinally generate a level 6 interrupt to the 68000 whenever the INDEX* input goes active
23	+12V		160 ma maximum; 540 ma surge

External Disk Connector Identification Mode

An identification mode is provided for reading a 32 bit serial identification data stream from an external device. To initialize this mode, the motor must be turned on then off. See pin 8, MTRXD* for a disucssion of how to turn the motor on and off. The transition from motor on to motor off reinitializes the serial shift register.

After initialization, the EELxB* signal should be left in the inactive state.

Now enter a loop where SELxB* is driven active, read serial input data on RDY* (pin 1), and drive SELxB*inactive. Repeat this loop a total of 32 times to read in 32 bits of data. The most significant bit is received first.

External Disk Connector Defined Indetifications

$0000 0000 - no drive present
$FFFF FFFF - Amiga standard 3.25 diskette
$5555 5555 - 48 TPI double density double sided

As with other peripheral ID's, users should contact Commodore-Amiga for ID Assignment.

The serial input data is active low and must therefore be inverted to be consistant with the above table.

External Disk Connector Limitations

1. The total cable length including diasy chaining must not exceed 1 meter.

2. A maximum of 3 external devices may reside on this interface.

3. Each device must provide a 1000 Ohm pull up resistor on every open collector input.

********** MEMO **********

Commodore-Amiga

TO: Hardware Developers
FM: H. Stolz - C=AMIGA
DT: 4/17/86

Dear developer;

In this package are the latest drawings of the Expansion Boards for the Amiga. We are releasing these drawings to you with the warning that the dimensions may change and are not final until FCC testing is completed. We feel confident about the 24.0mm (.945 in) center to center on the PCB's and we hope to increase the maximum component height allowed if the current rear panel scheme works out.

The Zorro Backplane Bd. Control Drawing # 327310-01 is included for refrence only. It is a board that was intended for lab use only and is not indicative of the Production backplane PCB.

4-24-86

Dear Hardware Developers,

This memo is to inform you of two updates to the expansion architecture.

The first is that Amiga will put -12Volts @ 250ma on pin 20 of the hundred pin connector. This voltage is optional on 3rd party expansion boxes. Boards should be designed to not use this current, because Amiga has aleady assigned it's use. We recommend that you do include this voltage unless the cost is prohibitive (for instance if you have already ordered power supplies that do not include -12V.)

The other addition is that the RES* line needs a 10K pullup resistor. You must not put more than one load on RES*, and you should make this load as small as possible.

I apologize that these minor additions keep trickling in. However, we are developing this standard as fast as we can, and are trying to give you third party developers the earliest possible notice of our work. At this time, the changes are all minor and only those that are absolutely necessary are made.

Amiga Engineering

Designing Hardware for the Amiga Expansion Architecture

Introduction

This document gives guidelines for designing hardware to reside on the Amiga expansion bus. The Amiga expansion bus is a relatively straightforward extension of the 68000 bus.

Hardware for the bus can be viewed as two categories: backplanes and PICs. Backplanes interface to the 86-pin connector of either another backplane or the Amiga itself. Backplanes buffer the bus and provide 100-pin connectors for PICs to plug into.

PIC is an acronym for plug-in card. A PIC is usually a card that plugs into the standard 100 pin Amiga backplane (as defined in this document).

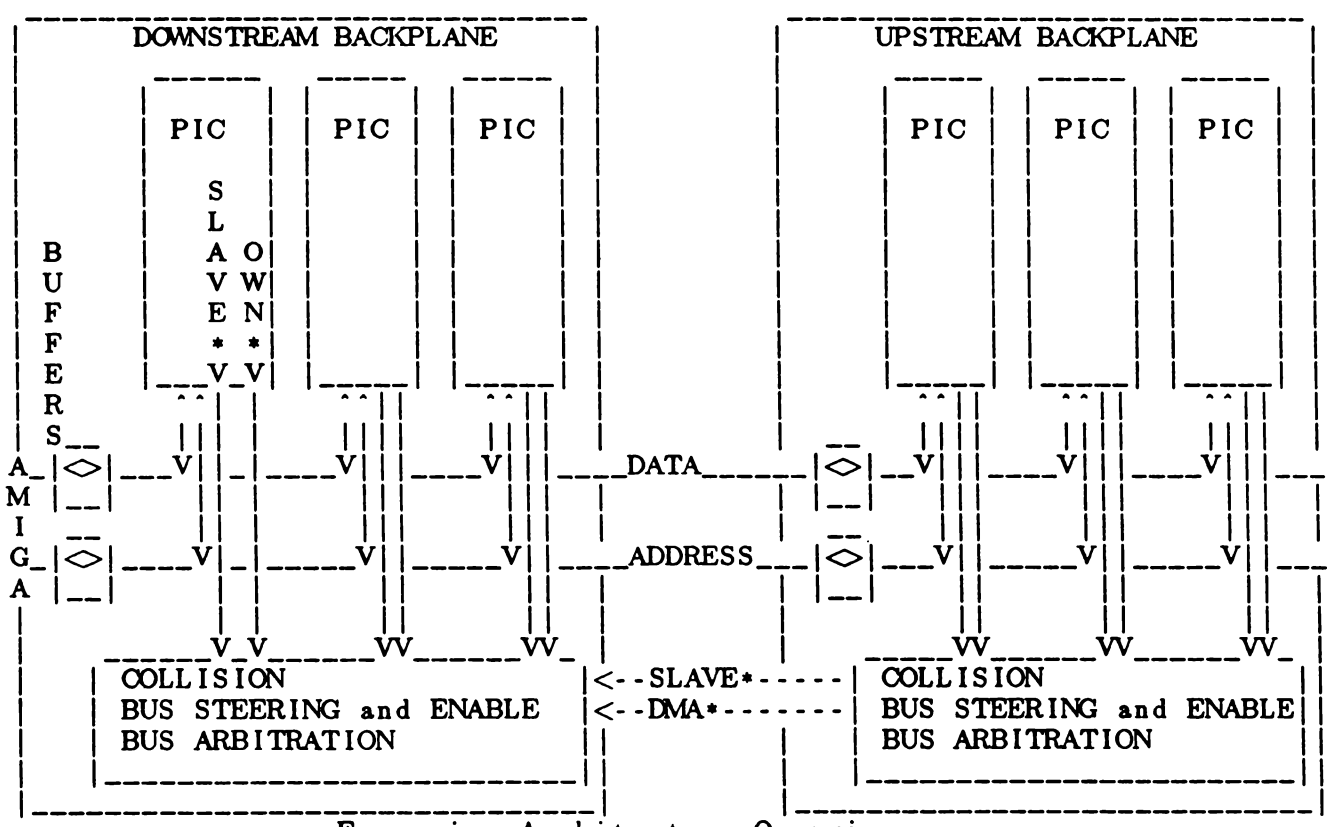
A sub-type of PIC is a combination of backplane and PIC integrated into one package. These combination products should follow all of the applicable backplane and PIC rules, especially auto-configuration.

Software never sees backplanes, all expansion hardware appears to the software as PICs.

1. Expansion Architecture Overview

As shown in the figure "Expansion Architecture Overview," the expansion bus is implemented as daisy-chained backplanes (expansion boxes) which accept PICs (boards). For timing and FCC reasons, it is probably not feasible to configure a system with more than two external backplanes attached. Even with only two, the timing is very tight, so the designer must be careful.

This document attempts to address the worst case, which is designing PICs and backplanes that are fast enough to use two such backplanes daisy-chained. It should be noted that a similar box that is designed to be the only backplane on the system could make some reductions in speed requirements and logic complexity.



Expansion Architecture Overview

Due to timing considerations it is not possible to daisy chain more than two buffered backplanes without inserting wait states.

WARNING

As of this preliminary writing, we have only tested a single backplane by itself, so it should be born in mind that some of the guidelines given here are not yet tested. We are especially concerned that address and data may not make it to PICs in the second expansion box in time, unless capacitance and noise are extremely well controlled. We will update these guidelines as our development work progresses, but you must be responsible for your own worst-case analysis. This document is very preliminary and will be amended in the future.

You should also take extreme care in controlling signal radiation from your product, in order to pass FCC class B regulations.

2. Design Guidelines for Backplanes

2.1. Backplane Block Diagram

2.1.1. Collision Detection Circuit

In this context, collisions are defined as any instance of two slaves attempting to respond to the same bus cycle.

All backplanes must have a collision detect circuit. The reason is that the PICs are auto-configurable and can be accidently instructed by software to respond to overlapping address spaces. Without collision detection, erroneous software can damage the hardware by causing severe tri-state fights.

Collision detect works in the following way. As soon as a PIC knows that it has been selected as the slave for this bus cycle, it asserts SLAVE* low and holds SLAVE* low until the end of the bus cycle (AS* going high).

The collision detect circuit (usually part of a PAL) detects whether more than one slave is responding, and if so asserts BERR*. All data drivers on the expansion bus must be designed to be tri-state disabled whenever BERR* is active. Because data drivers are not turned on until S4 (ASDELAYED* active), BERR* will have disabled the drivers before the tri-state fight can begin.

Note that in order to detect all cases of multiple slave response, the circuit must watch A23 - A19 for Amiga address spaces and also watch SLAVEIN* from the next box out. See discussion of the example schematic for specific PAL equations that implement collision detect.

Because BERR* is listened to by all PICs it will in some systems be heavily loaded, so it should be driven with a hefty open collector or tri-state driver. Each backplane should provide a 1 kiloohm pull-up resistor on BERR*.

2.1.2. Bus Arbitration Logic

The bus arbitration logic is based on the 68000 BR*, BG*, BGACK* protocol as described in the 68000 manual. In order to avoid metastable states in the backplane latches, all changes in state of the BR* lines from the PICs must be clocked by the rising edge of 7M.

The example design gives our current recommended bus arbitration logic.

2.1.3. Buffer Control Logic

The buffer control logic controls output enable and direction of the bidirectional tri-state bus drivers.

2.1.3.1. Data Driver Timing

It should be noted that the backplane drivers must not turn on until the rise of S4 during a read. This is okay because data from the Amiga internal RAMs is not valid during S4 anyway, so nothing is to be gained by turning the data buffers on earlier.

2.1.4. Clock Buffers, 7M, and ASDELAYED*

There are three clocks coming from the Amiga. These are CDAC, C1*, and C3*. The backplane must generate 7M (equivalent to the Processor clock) by the following equation: 7M = C1* XNOR C3*.

2.1.5. List of Signals on 86-Pin and 100-Pin Connectors

Appendix X lists all signal names on the 86 and 100 pin connectors. Note: If appendix X is not in this document yet, see accompanying schematic of backplane for pin lists.

List of signals that are unique to each connector.

List of signals that are on 100 pin conn but not on 86 pin

Describe function of each Amiga specific signal, eg SLAVE*, DMAOUT*, ASDELAYED*, 7M, OWN*, others?

2.2. The Protocols

The bus protocols are basically the same as standard 68000 protocols, however the timing margins are tighter due to the potentially long paths of Amiga and PICs talking to each other across two buffered backplanes.

One unusual feature is that when you are doing a DMA transfer into or out of the Amiga display RAM (the half megabyte starting at address 000000), the DTACK* circuit will synch the master up with C1. Because C1 is twice as slow as 7M, there are two possible phase relationships between C1 and the beginning of the DMA bus cycle. If AS* is asserted during the last quartile of C1 (C1 low and C3 low, see clock timing diagram), we call this an "in sync" bus cycle, and DTACK* will be given in time to do a normal 4-clock (7M) bus cycle (Note: occasionally DTACK* will be delayed due to contention with the graphics chips also, but that does not matter in this discussion).

However DTACK works differently if the DMA controller asserts AS* in the other phase. In the second quartile (C1 high and C3 high) the DTACK* circuit will hold off DTACK* long enough to insert one wait state, thus syncing up the "out of sync" bus cycle.

2.2.1. Read or Write Cycle with Amiga as Master

Since the Amiga bus master is a 68000, the bus cycle is a 68000 cycle. However, the responding slave does not pull DTACK*. Our internal circuitry will pull DTACK* unless the slave pulls XRDY low.

Also, the slave (PIC) must pull its SLAVE* output low as soon as it is selected, and at the end of the cycle disassert SLAVE* when AS* goes away.

2.2.2. Read or Write Cycle with a PIC as Master

A PIC as master must drive the bus using the same protocol as the 68000. Some of the timing margins must be better than those from the 68000, because the PIC is driving through several levels of buffers, and the Amiga logic is designed to the 68000 (8 megahertz part) specs. Specific timing requirements can be found in the tables in the timing appendix (Appendix A).

2.2.3. Bus Arbitration

The bus arbitration scheme is based on the 68000 BR*,BG*,BGACK* protocol. PICs are required to assert BR* clocked by the rising edge of 7M. This makes it less expensive to design bus arbitration logic that will be reliable. Specifically, synchronous arbitration logic can be clocked on 7M without danger of going metastable.

2.3. Timing General Discussion

See appendix A for timing specifications.

There are two main problems to be dealt with in the expansion architecture timing: propagation delays and skews in the clock, address, data, and control paths. The timing is tight, thus we recommend using FAST and AS parts to buffer these lines. To guarantee meeting the timing requirements, you must be careful to not exceed the recommended operating conditions of the parts you chose, for example the capacitive loading. In calculating your loading, note that all PICs are specified to present no more than two "F" loads plus minimal trace capacitance to each connector pin. Backplanes are specified to present no more than one "F" load plus trace capacitance to the Amiga. Do not use "typical" numbers, reliable systems can be built by using "worst case" numbers.

Appendix A gives the timing required in order to meet the timing requirements of two buffered backplanes with PICs.

3. Design Guidelines for PICs

3.1. Block Diagram of Typical PIC

3.2. The Protocols

Same as discussed above.

3.3. Auto Configuration

All PICs implement the auto-configuration protocol. The auto config protocol is designed so that system auto-config software can interrogate the PICs ID locations, build a system table of the installed PICs, and place the PICs in the 68000 memory space.

3.3.1. General Description of Auto Configuration

If it is difficult to imagine how to implement this protocol while it is being described, don't worry. The design requires one PAL, one latch, and one address match circuit. Complete details are given in the example design.

Upon reset, all PICs come up in the unconfigured state. In the unconfigured state, the PIC responds to the 64 kilobyte address space starting at location E80000 if CONFIGIN* is active to the PIC. If CONFIGIN* is not active, the PIC does not respond to any bus cycles.

The processor comes out and reads nibbles of ID data on D15 - D12 from the PIC. The table of ID data and the locations of control latches is detailed later in this section. This data includes such things as size of address space required, manufacturer's product number, and whether to add the PIC to the free memory pool (if it is a memory PIC.)

Under normal conditions, the processor determines how much address space the PIC requires and then loads the PIC's address latch with an appropriate base address. This permanently (until Reset) relocates the PIC at its new address, and passes CONFIGOUT* out to the next PIC's CONFIGIN*, whereupon the process is enacted again until all PICs are configured.

The smallest unit of memory that a PIC can ask for is 64 kilobytes. The largest is 8 megabytes. All PICs should be designed to be based on boundaries that match their space requirements, for example 1 megabyte PICs should be designed to reside on 1 megabyte boundaries (match circuit matches A23-A20.) There are two exceptions to this rule, however. Four megabyte PICs must be capable of being placed on 4 megabyte boundaries, as well as at hex 200000 and at hex 600000. Eight megabyte PICs should be capable of being placed on eight meg boundaries and at hex 200000. This strange requirement is because the 8 megabyte space reserved for expansion in the current machine begins at hex 200000; however, future models will probably place 4 and 8 megabyte PICs on their natural boundaries.

Address Specification Table. Nybble address is shown in ()

(15/14/13/12)	(15/14/13/12)	(hardware bits)	
(00/02)	7 6 5 4	3 2 1 0	Board type and size
(04/06)	7 6 5 4	3 2 1 0	Product number
(08/0A)	7 6 5 4	3 2 1 0	Preference for 8 Meg Space (Bits 5-0 must)
(0C/0E)	7 6 5 4	3 2 1 0	Reserved, must be 0
(10/12)	7 6 5 4	3 2 1 0	Mfg# high byte
(14/16)	7 6 5 4	3 2 1 0	Mfg# low byte
(18/1A)	7 6 5 4	3 2 1 0	Optional serial number, byte 0 (msb)
(1C/1E)	7 6 5 4	3 2 1 0	Optional serial number, byte 1

(20/22)	7 6 5 4 3 2 1 0	Optional serial number, byte 2
(24/26)	7 6 5 4 3 2 1 0	Optional serial number, byte 3 (lsb)
(28/2A)	7 6 5 4 3 2 1 0	Optional init/diagnostic vector high byte
(2C/2E)	7 6 5 4 3 2 1 0	Optional init/diagnostic vector low byte
(30/32)	7 6 5 4 3 2 1 0	Reserved, must be 0
(34/36)	7 6 5 4 3 2 1 0	Reserved, must be 0
(38/3A)	7 6 5 4 3 2 1 0	Reserved, must be 0
(3C/3E)	7 6 5 4 3 2 1 0	Reserved, must be 0
(40/42)	7 6 5 4 3 2 1 0	Optional control status register
(44/46)	7 6 5 4 3 2 1 0	Reserved, must be 0
(48/4A)	7 6 5 4 3 2 1 0	Base address register
(4C/4E)	x x x x x x x x	Optional ''shut up'' address
(50/52)	7 6 5 4 3 2 1 0	Reserved, must be 0
(54/56)	7 6 5 4 3 2 1 0	Reserved, must be 0
(58/5A)	7 6 5 4 3 2 1 0	Reserved, must be 0
(5C/5E)	7 6 5 4 3 2 1 0	Reserved, must be 0
(60/62)	7 6 5 4 3 2 1 0	Reserved, must be 0
(64/66)	7 6 5 4 3 2 1 0	Reserved, must be 0
(68/6A)	7 6 5 4 3 2 1 0	Reserved, must be 0
(6C/6E)	7 6 5 4 3 2 1 0	Reserved, must be 0
(70/72)	7 6 5 4 3 2 1 0	Reserved, must be 0
(74/76)	7 6 5 4 3 2 1 0	Reserved, must be 0
(78/7A)	7 6 5 4 3 2 1 0	Reserved, must be 0
(7C/7E)	7 6 5 4 3 2 1 0	Reserved, must be 0

Descriptions:

(00/02) 7 6 5 4 3 2 1 0 — Board type and size

- bits 7-6: Memory size
  - 000 = 8 megabytes
  - 001 = 64 kilobytes
  - 010 = 128 kilobytes
  - 011 = 256 kilobytes
  - 100 = 512 kilobytes
  - 101 = 1 megabytes
  - 110 = 2 megabytes
  - 111 = 4 megabytes
- bit 5: Chained config request, indicates that the next auto-config device in the daisy chain is physically tied to this device.
- bit 4: Diagnostic address valid
- bit 3: Link into memory free list

```
                |
                |___________________Board type
                                    00 = Reserved
                                    01 = Reserved
                                    10 = Reserved
                                    11 = New style board
```

(04/06)	7 6 5 4 3 2 1 0	Product number, this number is defined by the manufacturer of the board and is used by auto-config software to initialize drivers for the board.
(08/0A)	7 6 5 4 3 2 1 0	Reserved, must be as specified

```
                | | |_________|
                | |     |_____________bits are currently zero
                | |___________________0 means this board can be shut up
                |                     1 means this board cannot be shut up
                |_____________________0 means any space okay
                                      1 means I prefer to be put in the 8 Meg spa
```

(0C/0E)	7 6 5 4 3 2 1 0	Reserved, must be 0
(10/12)	7 6 5 4 3 2 1 0	Mfg# high byte
(14/16)	7 6 5 4 3 2 1 0	Mfg# low byte , these 2 bytes are assigned by Commodore-Amiga. They are used by the auto-config software to initialize drivers for boards.
(18/1A)	7 6 5 4 3 2 1 0	Optional serial number, byte 0 (msb)
(1C/1E)	7 6 5 4 3 2 1 0	Optional serial number, byte 1
(20/22)	7 6 5 4 3 2 1 0	Optional serial number, byte 2
(24/26)	7 6 5 4 3 2 1 0	Optional serial number, byte 3 (lsb)
(28/2A)	7 6 5 4 3 2 1 0	Optional init/diagnostic vector high byte
(2C/2E)	7 6 5 4 3 2 1 0	Optional init/diagnostic vector low byte, If the 'diag addr valid' bit (4 of nibble 0) is set, then these 2 bytes are the offset from the boards base address at which the start of the diagnostic information is located. If the bit is not set, then these 2 bytes have no meaning. The rules as to how this vector is to be used have not been defined yet. If you have a need for this feature, contact Commodore Amiga third party support

% NOTE: we need to define the contents of this information

(30/32)	7 6 5 4 3 2 1 0	Reserved, must be 0
(34/36)	7 6 5 4 3 2 1 0	Reserved, must be 0
(38/3A)	7 6 5 4 3 2 1 0	Reserved, must be 0
(3C/3E)	7 6 5 4 3 2 1 0	Reserved, must be 0
(40/42)	7 6 5 4 3 2 1 0	Optional control status register

```
| | | | | | | |
| | | | | | | |        Write                 Read
| | | | | | | |        ----------------      ----------------
| | | | | | | |_____Interrupt enable       Interrupt enable
| | | | | | |_______must be 0              don't care
| | | | | |_________Local reset            must be 0
| | | | |___________must be 0                don't care
| | | |_____________must be 0                INT2 pending
| | |_______________must be 0                INT6 pending
| |_________________must be 0                INT7 pending
|___________________must be 0                I am pulling INT
```

(44/46) 7 6 5 4 3 2 1 0 Reserved

Write	Read
Not defined	must be 00

% Neil - what is this?

(48/4A) 7 6 5 4 3 2 1 0 Base address register, write-only

```
|_____________|
      |_____________These bits are compared with A23 through A8
                    (or fewer) to determine the base address
                    of this board.
```

(4C/4E) x x x x x x x x Optional ''shut up'' address, a write to this address will cause the board to pass its config out and then never again respond to any address. RESET will re-enable the board. The actual address that has this effect is 4C. A write to 4E is ignored. This is write only.

(50/52)	7 6 5 4 3 2 1 0	Reserved, must be 00
(54/56)	7 6 5 4 3 2 1 0	Reserved, must be 00
(58/5A)	7 6 5 4 3 2 1 0	Reserved, must be 00
(5C/5E)	7 6 5 4 3 2 1 0	Reserved, must be 00
(60/62)	7 6 5 4 3 2 1 0	Reserved, must be 00
(64/66)	7 6 5 4 3 2 1 0	Reserved, must be 00
(68/6A)	7 6 5 4 3 2 1 0	Reserved, must be 00
(6C/6E)	7 6 5 4 3 2 1 0	Reserved, must be 00
(70/72)	7 6 5 4 3 2 1 0	Reserved, must be 00
(74/76)	7 6 5 4 3 2 1 0	Reserved, must be 00
(78/7A)	7 6 5 4 3 2 1 0	Reserved, must be 00
(7C/7E)	7 6 5 4 3 2 1 0	Reserved, must be 00

3.4. Mechanical

Two mechanical drawings are included in this package. The first gives the physical dimensions and pin locations for a PIC board.

The second gives detailed dimensions of the 86-pin expansion connector on the right side of the Amiga.

Board center to top of highest component should not exceed .60 inches on PICs. The backplane should provide a minimum of .80 inches center to center for the boards.

Note that due to differing market and system requirements, we have not provided a form factor for the expansion box itself. Our current laboratory prototype backplane comes straight out horizontally from the Amiga connector. The PICs plug into this backplane vertically. Several other configurations make sense.

3.5. RES* and RESB*

Note that there are two reset lines going to every PIC, RES* on pin 53 and RESB* (reset buffered) on pin 94. The RESB* line is intended to be the normal Reset input to the PIC. All normal PICs will use this line as an input, so it is buffered.

RES* is intended only to be used by those PICs which are designed to have the capability of resetting the system. Normal PICs will not drive nor load this line. Note that because RES* is not buffered, it can reset the Amiga, as well as all PICs (via RESB*.)

4. Example Backplane Design

We have designed a backplane as an example implementation of our expansion architecture. This section is a detailed description of the schematic of that backplane.

4.1. Backplane Schematic Overview

While reading this section, refer to the backplane schematic to see what is being described.

This implementation of the backplane consists of:

J1-J5 five 100 pin edge connectors for PICs
J6 & J7 are not used
J8 the power supply connector
J9 edge fingers passing the bus out to the next device
J10 edge fingers connecting to the Amiga (or the "next in" device)
thirteen integrated circuits for buffering, control, and timing
termination resistors

The bus comes in on the left from the processor via J10. Note that both the data bus and address bus are buffered through bi-directional buffers. The buffers are bi-directional in order to allow external DMA controllers.

4.2. The Bus Buffers and Their Control Logic

This subsection describes the bus buffers, their timing and control logic. In this discussion "upstream" means away from the processor, and "downstream" means toward the processor. For instance, if you daisy chain two devices on the bus, the further away of the two is "upstream" from the closer (downstream) device.

Throughout this document, there are references to signals going active. Active is defined in the glossary.

4.2.1. The Address and Control Buffers

The address lines, function codes, UDS*, LDS*, R/W, and AS* are all buffered in the same manner by 74F245s. Their buffer direction is determined by DMAOUT. They are enabled by ADDR_OE* (address output enable bar).

4.2.1.1. Generating DMAOUT

This section explains the PAL equation for DMAOUT found in the STEERING PAL appendix. Refer to those PAL equations during this discussion.

DMAOUT active means that the current bus master is upstream of the buffers. Since the buffers are at the extreme downstream end of this backplane, the master is either on this backplane or upstream from this backplane. Thus when DMAOUT is high, the drivers drive the address and control lines downstream (toward the Amiga.)

The PAL equation for DMAOUT is very straightforward:

DMAOUT = DMAIN + OWN

DMAIN is active when the bus master is upstream from this backplane. So when DMAIN is active, DMAOUT must go active.

OWN* is the wire OR'ed signal which means that this backplane has the current bus master. Thus because all PICs on this backplane are upstream from the address (and data) buffers, DMAOUT must be active when OWN (or OWN*) is active.

4.2.1.2. Generating ADDR_OE*

This section explains the PAL equation for ADDR_OE*. Refer to the "STEERING PAL" appendix to see the equation (AOE).

ADDR_OE* is active (enabling the address drivers) most of the time. It only disables the drivers when ownership of the bus is changing (for example a new master takes control). At these transition times ADDR_OE* is inactive so that the tri-state drivers will not fight the drivers on the next backplane while they are changing direction.

Refer to the equation for AOE in the "STEERING PAL" appendix. AOE = ADDR_OE* inverted. The inverter is in the output stage of the PAL.

BGACK is asserted (BGACK* pulled low) by all bus masters (except the 68000) when they are the current master, so ADDR_OE* is active when BGACK is active.

The term (BG* * DMAOUT*) is true most of the time that the 68000 owns the bus. However, when the 68000 is about to give up the bus, BG* will go active and thus (BG* * DMAOUT*) will go inactive. It is important that the address drivers remain on until the end of the final 68000 bus cycle when the 68000 is giving up the bus, so the term AS holds AOE active when BG goes active during the bus cycle.

AS does not last quite long enough, so ASQ90 (which is a slightly delayed AS) holds AOE active long enough to finish the cycle.

4.2.2. The Data Buffers

This section describes when and why the data drivers are turned on and off. It also describes control of data direction.

4.2.2.1. Generating DBOE*

Refer to the STEERING PAL equation for DBOE.

Note that all the bus drivers are enabled for every bus cycle unless BERR* is asserted. This allows for easier use of bus-monitoring tools such as state analyzers.

It is fairly difficult to avoid tri-state fights on the data buffers. In order to get data out to dynamic RAM PICs at an early enough time, we do not use the data strobes to enable the data drivers, because these strobes can go active very late in a write cycle.

On a read cycle we use the data strobes, so that in case the cycle turns out to be a Read-Modify-Write cycle, the drivers will be turned off (to avoid tri-state fight) while the R/W line is changing state.

Refer to the PAL equation for DBOE in the STEERING PAL appendix. The term (AS * RD*) turns on the drivers for all write cycles, including the write portion of Read-Modify-Write cycles. Note that since AS turns off the data drivers, the data hold time is not guaranteed beyond AS going inactive, so it is poor design practice to try to use the rising edge of AS*, UDS*, or LDS* to latch data.

The terms (UDS * RD * ASQ) and (LDS * RD * ASQ) turn on the drivers for all read cycles. The UDS and LDS turn off the drivers in the middle of a read-modify-write cycle. Note that since AS turns off the data drivers, the data-hold time is not guaranteed beyond AS going inactive. Therefore it is poor design practice to try to use the rising edge of AS*, UDS*, or LDS* to latch data.

The terms (UDS * RD * ASQ) and (LDS * RD * ASQ) turn on the drivers for all read cycles. The ASQ (ASDELAYED equivalent) keeps the data buffers from turning on until after there has been enough time for the collision

detect circuit to assert BERR* low and thus disable the data drivers before they fight (see collision detection.)

4.2.3. Generating D_TO_PROC*

The inverse of the D_TO_PROC* signal is called D2P in the PAL equation.

4.3. Collision Detection

Each backplane or device that passes the bus, or allows more than one slave device, must have a collision detect circuit. This circuit will usually be implemented in a PAL. This circuit must detect any instance of two slaves responding to the same bus cycle and assert BERR* immediately upon detecting such an error.

The collision circuit has an input (see schematic) SLAVEIN* which is passed from the upstream backplane or device (if any is present). If no upstream device is present, the pull-up resistor will hold SLAVEIN* inactive (high). SLAVEIN* tells the circuit whether or not an upstream PIC is responding to the current bus cycle as a slave.

The circuit also has one input for each slot on this backplane. If any PIC on this backplane is responding as a slave, the corresponding SLAVEn* will be active.

The collision circuit also monitors A23 through A19 and OVR* on the bus, so that the internal reserved address spaces of the Amiga can be checked. An access to any of the internal address spaces will make the Amiga respond as the slave unless OVR* (override) is asserted.

Any two slave responses on the same cycle constitute a collision.

Refer to the "COLLISION PAL" appendix.

4.3.1. Generating the PROC Term

Before generating the collision detection equation, we must make the equation that detects whether the Amiga processor board is responding to this cycle as a slave. This signal is call PROC internally to the PAL. While it comes out on pin 18, it is not used external to the PAL.

The term BAS * /A23 * /A22 * /A21 * /RESET * /OVR will be true when the processor board memory is responding to the 2 megabyte space starting at hex 000000.

Similarly, the next term will be true when the processor board is responding to the 2 megabyte space that starts at hex A00000.

The next term detects the processor board responding to the 2 megabyte space starting at C00000.

The next term detects the processor board responding to the 1/2 megabyte space starting at E00000.

And the last term detects the proc board responding to the 1/2 megabyte space starting at F80000. This takes care of all the spaces used by the processor board.

4.3.2. Generating NOTCOLIS

Why the inverted name? We would have preferred to call this signal /COLLISION but our PAL assembler does not allow a NOT sign in the name on the left side of the equal sign. NOTCOLIS goes out through the output inverter and becomes /NOTCOLIS which is logically equivalent to NOTNOTCOLIS = COLLISION, so NOTCOLIS being true inside the PAL will make COLLISION false outside the PAL.

Now that PROC will tell us when the responding slave is inside the Amiga, we are ready to do collision detection.

In our example we have seven possible slaves to keep track of. They are the Amiga board (PROC), five PICs on this backplane, and SLAVEIN* from the upstream backplane or device. If six of the seven are inactive at all times, we know that no two are active at the same time.

Because the slave lines go inactive between bus cycles, there should not be a case of one slave going active before the previous one went inactive.

By the way, don't worry about two slaves colliding on the upstream of the backplane; that backplane has a collision detect circuit of its own.

Thus, each of the seven product terms indicates that a collision is not happening at this time. Only one of them needs to be true to know that a collision is not happening at this time.

4.4. Bus Arbitration Circuit

The bus arbitration circuit's main job is to determine which PIC will receive BG* active (Bus Grant) when the 68000 asserts BG*. The circuit we recommend does this based on priority, where the closest PIC to the 68000 is the highest priority. You could implement something fancier as long as only one PIC owns the bus at a time.

PICs are only allowed to assert BR* off the rising edge of 7M. This allows the bus arbitration circuit to operate synchronously, clocked by the rising edge of 7M.

The output of the bus arbitration circuit only changes when the 68000 changes the state of BG*. If the 68000 is asserting BG*, the arbitration circuit passes BG* active to the highest priority active requester. When the 68000 disasserts BG*, the arbitration disasserts BG* also. Therefore no PIC has a grant.

4.5. RES* and RESB*

Note that there are two reset lines going to every PIC, RES* on pin 53 and RESB* on pin 94. The RESB* line is intended to be the normal reset input to the PIC. All normal PICs will use this line as an input, so it is buffered.

RES* is intended only to be used by those PICs which are designed to have the capability of reseting the system. Normal PICs will not drive nor load this line. Note that because RES* is not buffered, it can reset the Amiga, as well as resetting all PICs (via RESB*).

4.6. CONFIG_IN* CONFIG_OUT* Daisy Chain

The CONFIG_IN* signal will be passed to CONFIG_OUT* at the appropriate time if there is a PIC plugged in the slot. On this backplane, we have used 74LS32s to pass CONFIG_OUT* to the next slot if there is no PIC. The pull down resistor allows the CONFIG_IN* signal to pass directly through the gate to CONFIG_IN* of the next slot if there is no PIC installed, thus bypassing the empty slot. If a PIC is installed, the PIC's CONFIG_OUT* driver will override the pull down resistor.

Another method that would work is to use special pins on the connector at pins 11 and 12, such that 11 and 12 short to each other when there is no PIC inserted in the connector. This would eliminate the need for the 74LS32 gates.

4.7. Backplane Timing Generation

4.7.1. Clock Buffers

The clock buffers for C1*, C3*, and CDAC were chosen for minimum propagation delay and minimum skew. Notice that buffered clocks are passed to the 100 pin edge connectors, but that the unbuffered clocks are passed to the 86 pin connector that goes on to the next box in order to minimize propagation delay to the next backplane.

4.7.2. Generating 7M

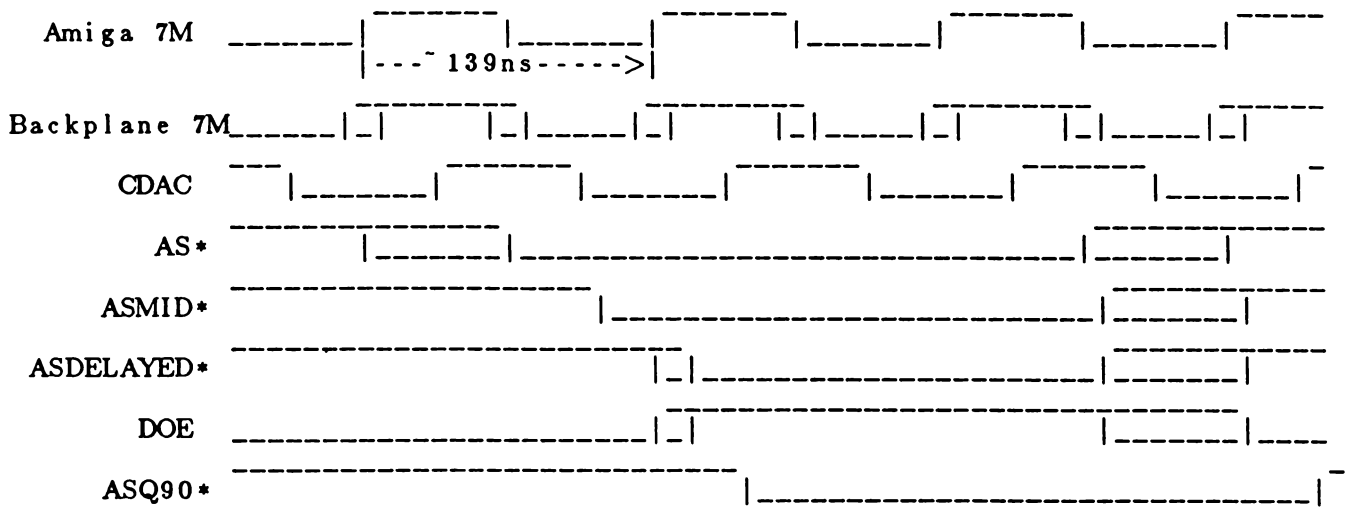
We generate 7M (equivalent to the processor clock) by:

$$7M = C1* \text{ XNOR } C3*$$

This yields a 7.16Mhz clock which is used to generate ASDELAYED*, DOE, and ASQ90*. 7M is also passed to the PICs on pin 92 of the edge connectors, so they will have a cheap clock for accessing the bus.

4.7.3. DOE, ASDELAYED*, ASQ90*

DOE (Data output enable) and ASDELAYED* are the compliment of each other. ASDELAYED*is used in the steering PAL (ASQ = ASDELAYED in the PAL equations) to time turning on of the data drivers during a read cycle. DOE is passed to the PICs on pin 93 of the edge connectors, to tell the PICs when to turn on data drivers during a read cycle.



Backplane Timing Signals

4.8. Power Budget

We recommend that you provide 2.5 Amps of +5 volts per slot position on your backplane, plus 1.5 Amps additional. In other words, an 8-slot backplane should provide 21.5 Amps of +5 ((8x2.5)+1.5 = 21.5). This will allow one of the boards to be an 8 megabyte RAM board (4 Amps) without limiting the normal 2.5 Amps to the other boards.

The backplane should also provide 1 Amp of +12 volts total,

The backplane should also provide 1/2 Amp of -5 volts total.

5. Example PIC Design

This section is a description of the schematic for a small 16 kilobyte RAM board that we designed as our first test PIC for the expansion architecture. It is valuable as an example because it implements all of the basic features of a slave PIC.

5.1. The PIC at System Startup

The heart of auto-config is in U1 (address register), U2 (address comparator), and U3 (ID PAL and control PAL).

When the board comes out of Reset, CONFIG_OUT* is inactive, and does not pass the config token on to the next PIC. CONFIG_IN* may or may not be active at first. If it is not active, the board will not respond to any bus cycles. For instance, we can see at U11 that SLAVE* is disabled when CONFIG_IN* is inactive (high), because this does not allow BOARD_SEL* to go active.

In turn, BOARD_SEL* is an input to U3, the control PAL. Without BOARD_SEL*, all ten of the PAL outputs are held inactive (see PAL equations for test ram).

5.2. Reading The ID Locations

Eventually, during execution of the auto config code, CONFIG_IN* will be asserted to this PIC between bus cycles (AS* inactive). Notice that the address latch is tri-stated off so that the pull-up and pull-down resistors are inputing a pattern of E8 to the address comparator. When the backplane addresses E8xxxx, this board will now respond because CONFIG_IN* is active but CONFIG_OUT* is not yet active. In other words, CONFIG_IN* is enabling board select, and CONFIG_OUT* has not yet allowed the address latch to move the board to a different address space.

Notice that whenever BOARD_SEL* goes active, SLAVE* will go active unless SHUT_UP_FOREVER is latched active. SHUT_UP_FOREVER* is a feedback latch in the PAL. It is only set by the software if the board cannot be configured into the system (for instance if the user has plugged in too many large address space PICs and there is no room left for this one).

If you analyze the PAL equations for BD15 through BD12, you will see that their data drivers turn on for all reads ANDed with BOARD_SEL active, until CONFIG_OUT* is set active (or some exception happens such as reset, bus error, or shutup).

By the way, if you're not used to PALs, it's normal old Boolean: * means AND, / is bar, + is OR, IF(term) means "If the term evaluates to TRUE then turn on the tri-state driver".

Further analysis of the BD15-BD12 equations will show that almost all addresses put out ones; however, remember that most of the nibbles are inverted because the spec says they have to be (not all of course, noooo that would be too simple.) The inversion makes it possible to implement the codes in active low PALs; it is just a cost reduction.

Analysis of the equations shows that the only nibbles (we don't care about above HEX 80) that are outputting any zeros are:

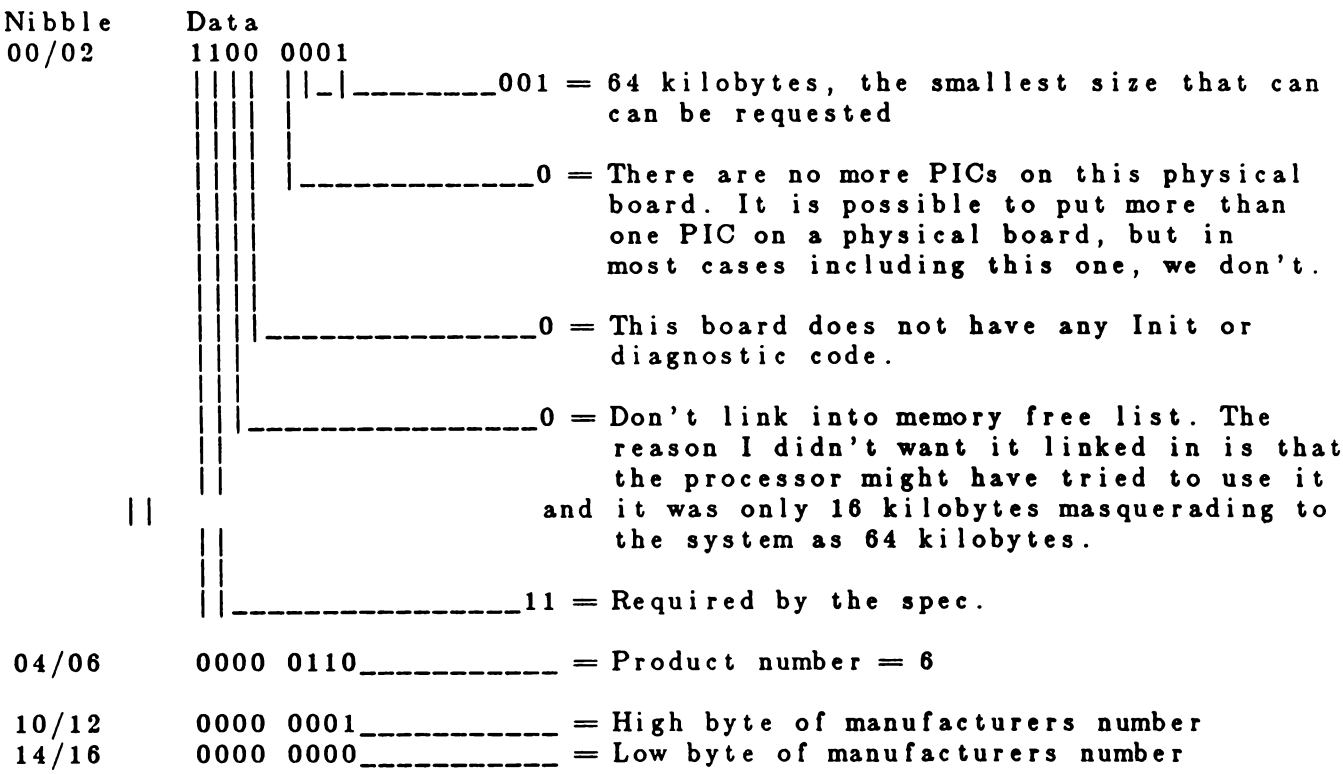
```
00/02  1100 0001
04/06  1111 1001
10/12  1111 1110
40/42  0000 0000
```

To interpret this code, we need to remember that the spec says that all nibbles get inverted except 00, 02, 40, and 42. So our new table looks like this:

```
00/02  1100 0001
04/06  0000 0110
10/12  0000 0001
40/42  0000 0000
```

And all the other nibbles that were ones are now inverted to zeros.

What do these codes mean?



40/42 0000 0000_____ = Because this PIC does not generate INTs

When you want to program your own ID PAL, just work back to the equations. First determine what ID pattern you need by reading about the nibbles in the spec. Write down a table of ones and zeros. Invert all of these except nibbles 00, 02, 40, and 42. Then, doing one data line at a time, write a product term for each binary zero that you want to output from the ID PAL.

5.3. Passing CONFIG_OUT*

The equations for CONFIG_OUT* in this implementation make two feedback latches in the PAL. The first latch PRE_CONFIG_OUT* is set during the bus cycle in which the processor does a write to the address register. In fact, in this design the rising edge of PRE_CONFIG_OUT latches the final Address value into the address latch.

The second latch outputs CONFIG_OUT*. This latch goes active after AS* goes inactive at the end of the bus cycle in which the new address was written. Notice that CONFIG_OUT* enables the address latch U1, so it now provides the new address range to the comparator.

CONFIG_OUT* enables the next PIC in the chain, and will remain active until a system reset or power down occurs.

6. Appendix A - Timing Specifications

6.1. Timing Requirements

Timing Requirements for Backplane				
Num	Characteristic	Min	Max	Unit
1	AS* UDS* LDS* Delay	2	8	ns
2	Address 23-1 delay	2	8	ns
3	7M(S4 RISE) to Data Enable during Read	0		ns
4	7M (S4 RISE) to Data Valid		35	ns
5	Data 15-0 Delay to Output		8	ns
6	SLAVEIN or SLAVE to SLAVEOUT Delay	0	25	ns

6.2. Timing Requirements for PIC

Timing Requirements for PIC as Slave (Rd & Wr Cycles)				
Num	Characteristic	Min	Max	Unit
1	AS* low to SLAVE* Low	0	35	ns
2	AS* high to SLAVE* high	0	50	ns
3	AS* low to XRDY low (to insert wait)	0	60	ns
4	Read Data Valid to local 7M low (S7)	60		ns

Timing Requirements for PIC as Master (Rd & Wr Cycles)				
Num	Characteristic	Min	Max	Unit
1	7M high(S2) to AS* low	0	67	ns
2	Address 23-1 Valid to AS* low	30		ns
3	7M high (S4) to Data Valid Wr Cycle		0	ns

Timing Requirements for PIC Bus Master (Bus Arbitration)				
Num	Characteristic	Min	Max	Unit
1				ns

6.3. Guaranteed Timing to Backplane

Guaranteed Timing to Backplane				
Num	Characteristic	Min	Max	Unit
1	AS* Low to CDAC low (Setup)	20		ns
2	AS* High to CDAC High (Setup)	20		ns

6.4. Guaranteed Timing to PIC

Guaranteed Timing to PIC (PIC in Slave Mode)				
Num	Characteristic	Min	Max	Unit
1	Valid Address to AS* Low	10		ns
2	Valid Data from 7M High(S4) on Wr to PIC		35	ns

Guaranteed Timing to PIC (PIC in Master Mode)				
Num	Characteristic	Min	Max	Unit
1	Valid Data setup to Local 7M low(S7)	15		ns

7. Appendix B - Backplane Loading Spec

Note: The Amiga is designed to drive into a device that meets the backplane specification.

% need to spec how many pf a box may present to the upstream box

All backplane 86-pin connectors at the processor end of the backplane must present no more than 1 "F" load plus ??pf to the connecting device, with the following exceptions:

Pin 53 RES*

7.1. Appendix C - PIC Loading Spec

All PIC pins must present no more than 2 "F" loads plus trace capacitance.

7.2. Appendix D - Clocks

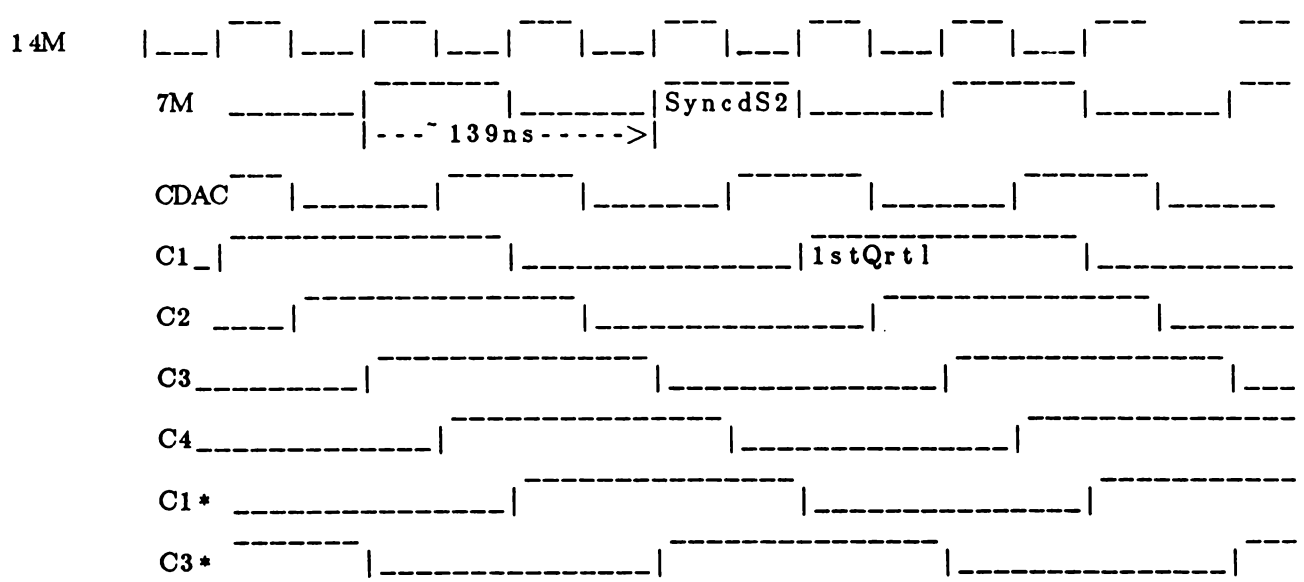


Fig. 1 Amiga System Clocks

8. Glossary

Active Active high signals are considered active when they are in the "one state" or "high state". Active low signals are considered active when they are "low" or in the "zero state". Active high signals do not have barred signal names. Active low signals do have barred signal names. Active means that the signal is

1. is true (non-barred) and is currently in the one state, or
2. is a barred signal name and is currently in the zero state.

An example is AS* (the * = bar). AS* is active when it is equal to zero. A counter example is the signal AS (the inverse of AS*), which is active when it is in the one state.

Auto Configuration The protocol (specified in this document) that Amiga uses to configure expansion cards into the system.

Downstream Downstream means closer to the Amiga. For instance, if two backplanes are daisy chained on the bus, the closer-in backplane is downstream from the further-out backplane. The concepts of upstream and downstream are important in determining which direction the address and data drivers should drive.

Master A PIC which is capable of initiating DMA cycles on the bus.

PIC A PIC is a plug-in card or a product which behaves in the system as a plug in card. That is, it provides a resource that resides on the expansion bus, and follows the rules for auto-config, master protocol, slave protocol, etc.

Slave A slave is a PIC that can only respond to bus cycles. A slave cannot initiate bus cycles: in other words, it does not drive the address lines on the backplane, nor AS*, UDS*, LDS*.

Upstream Upstream means further away from the processor. For instance, all PICs are upstream from the buffers on the backplane that they are plugged into because the buffers are between the PIC and the Amiga.

```
PAL16L8
STEERING150R17 REV3
11-17-85
AMIGA

/SLVOUT RD /ASQ /ASQ90 COLLIS /BG /AS /BGACK /DMAIN GND
/OWN /AOE /UDS /BERR /DMAOUT /LDS /DBOE /RES /D2P VCC


DBOE    = AS   * /RD        * /BERR +          ;DATA DRIVERS DURING WRITE CYCLE
          UDS *  RD * ASQ * /BERR +            ;TURN ON DRIVERS LATE FOR RD
          LDS *  RD * ASQ * /BERR              ;UDS AND LDS PROTECT RD MOD WR
                                               ;TO AVOID TRI_STATE FIGHT
D2P     = /DMAOUT *  SLVOUT *  RD +            ;DOWNSTREAM READS UPSTREAM SLAVE
          DMAOUT * /SLVOUT * /RD +             ;UPSTREAM WRITES DOWNSTREAM SLAVE
          DMAOUT *  SLVOUT                     ;MASTER AND SLAVE ARE UPSTREAM

AOE     = BGACK          +
          /BG * /DMAOUT +
          AS             +                     ;AS KEEPS ADDR WHEN /BG DROPS
          ASQ90                                ;ASQ90 MAINTAINS VALID ADDR ON
                                               ; LAST PROC CYCLE

DMAOUT = DMAIN + OWN

IF (/RES * COLLIS) BERR = VCC

DESCRIPTION

SLVOUT=SLAVEOUT,ASQ=AS DELAYED,ASQ90=AS CLKD ON LOW EDGE OF 7M,
BG=BUS GRANT,OWN=LOCAL OWN
COLLIS=BUS COLLISION,AOE=ADDR OUTPUT EN,DOE=DATA OE
RES=RESET,D2P=DATA TO PROCESSOR
UDS LDS PROTECT AGAINST RDMODIFYWRITE 3STFIGHT & BERR= /DOE
```

```
PAL16R6
ARBITRATE REV1
1-6-86
AMIGA

7M /BRIN /RES /BGIN /BR5 /BR4 /BR3 /BR2 /BR1 GND
GROUND /BGOUT /BGOLD /BG5 /BG4 /BG3 /BG2 /BG1 /BR VCC

BG1 = BGIN * /BGOLD * BR1 *                          /RES +  ;GENERATE BG1
      BGIN * BG1    *                                /RES    ;HOLD UNTIL /BG

BG2 = BGIN * /BGOLD * BR2 * /BR1 *                   /RES +
      BGIN * BG2    *                                /RES

BG3 = BGIN * /BGOLD * BR3 * /BR1 * /BR2 *            /RES +
      BGIN * BG3    *                                /RES

BG4 = BGIN * /BGOLD * BR4 * /BR1 * /BR2 * /BR3 *           /RES +
      BGIN * BG4    *                                      /RES

BG5 = BGIN * /BGOLD * BR5 * /BR1 * /BR2 * /BR3 * /BR4 *  /RES +
      BGIN * BG5    *                                    /RES

BGOLD = BGIN                                              ;STORE OLD STATE OF BG

BR    = BRIN * /RES +                                     ;BR IS RQST TO 68K
        BR1  * /RES +
        BR2  * /RES +
        BR3  * /RES +
        BR4  * /RES +
        BR5  * /RES

BGOUT = BGIN * BGOLD * /BG1 * /BG2 * /BG3 * /BG4 * /BG5   ;SIMPLY MAHVELOUS



DESCRIPTION
BG1 IS HIGHEST PRIORITY
```

PRELIMINARY

```
PAL20L10
TESTRAM
9-11-85
COMMODORE-AMIGA

/ASQ /ASQQ RD /BDSEL /BERR A6 A5 A4 A3 A2
A1 GND /RES BD12 BD13 BD14 BD15 /PRECON /CONOUT /SHUTUP
/RAMOE /WP /DBOE VCC

DBOE = /RES*BDSEL*/BERR*/SHUTUP*/RD +                ;WRITES TURN ON EARLY
       /RES*BDSEL*/BERR*/SHUTUP* RD*ASQ               ;ASQ DELAYS THE READ

WP   = /RES*ASQ*/ASQQ*BDSEL*CONOUT*/SHUTUP*/RD*/BERR

RAMOE = /RES*ASQ*RD*CONOUT*/BERR*BDSEL

SHUTUP = /RES*BDSEL*/RD*ASQ*/CONOUT*A6*/A5*/A4*A3*A2 +
         /RES*SHUTUP

PRECON = /RES*SHUTUP +
         /RES*/RD*BDSEL*ASQQ*A6*/A5*/A4*A3*/A2*/A1 +
         /RES*PRECON

CONOUT = /RES*/ASQ*PRECON +
         /RES*CONOUT

IF(/RES*BDSEL*/CONOUT*RD*/BERR*/SHUTUP) /BD15 =
         /A6*/A5*/A4*/A3*/A2*A1 +
          A6*/A5*/A4*/A3*/A2

IF(/RES*BDSEL*/CONOUT*RD*/BERR*/SHUTUP) /BD14 =
         /A6*/A5*/A4*/A3*A1 +
          A6*/A5*/A4*/A3*/A2

IF(/RES*BDSEL*/CONOUT*RD*/BERR*/SHUTUP) /BD13 =
         /A6*/A5*/A4*/A3*/A2 +
         /A6*/A5*/A4*/A3*A2*A1 +
          A6*/A5*/A4*/A3*/A2

IF(/RES*BDSEL*/CONOUT*RD*/BERR*/SHUTUP) /BD12 =
         /A6*/A5*/A4*/A3*/A2*/A1 +
         /A6*/A5*A4*/A3*/A2*A1 +
          A6*/A5*/A4*/A3*/A2

DESCRIPTION

THING
```

```
PAL16L8
COLLISSION
11-17-85
AMIGA

/BAS /SLV1 /SLV2 /SLV3 /SLV4 /SLV5 /SLVIN A23 A22 GND
A21 /SLVOUT A20 A19 /OVR /RESET P17 /PROC /NOTCOLIS VCC

SLVOUT = SLV1 + SLV2 + SLV3 + SLV4 + SLV5 + SLVIN

NOTCOLIS =          /SLV1 * /SLV2 * /SLV3 * /SLV4 * /SLV5 * /SLVIN +
           /PROC *          /SLV2 * /SLV3 * /SLV4 * /SLV5 * /SLVIN +
           /PROC * /SLV1 *          /SLV3 * /SLV4 * /SLV5 * /SLVIN +
           /PROC * /SLV1 * /SLV2          * /SLV4 * /SLV5 * /SLVIN +
           /PROC * /SLV1 * /SLV2 * /SLV3          * /SLV5 * /SLVIN +
           /PROC * /SLV1 * /SLV2 * /SLV3 * /SLV4          * /SLVIN +
           /PROC * /SLV1 * /SLV2 * /SLV3 * /SLV4 * /SLV5

PROC   = BAS * /A23 * /A22 * /A21 *                  /RESET * /OVR +
         BAS *  A23 * /A22 *  A21 *                  /RESET * /OVR +
         BAS *  A23 *  A22 * /A21 *                  /RESET * /OVR +
         BAS *  A23 *  A22 *  A21 * /A20 * /A19 *    /RESET * /OVR +
         BAS *  A23 *  A22 *  A21 *  A20 *  A19 *    /RESET * /OVR

DESCRIPTION

EMPTY
```

INTERFACING TO THE 68K BUS CONNECTOR ON THE AMIGA

0. Scope

WARNING: While care has been taken to make this document as accurate as possible, it is preliminary and is subject to change without notice.

This document is intended to give the necessary information for interfacing to the 68000 bus connector on the right side of the Amiga computer. It will be updated as people give feedback.

0.1 References

The Amiga Hardware Manual gives the address map and other useful information.

Appendix E of the hardware manual gives the complete pinout for the 86 pin connector.

1. Form Factors

1.1 Form Factors For Standard Box With Cards

We are planning on recommending a standard form factor for motherboards and daughter cards for the Amiga. While this definition is not yet solid, we have some preliminary information. See attached drawing for the proposed daughter card form factor.

1.2 The Connector On The Amiga

The connector is a standard dual row 86 finger (43 on a side) edge connector, spaced on .1" centers. Here are some part numbers of connectors that are compatible:

solder tail	AMP 2-530841-1
wire wrap	AMP 4-530396-7
card extender	AMP 1-530826-2

Mitsumi also makes connectors that fit.

See accompanying drawing for physical dimensions of this connector.

2. Timing

2.1 Clocks

For this discussion see figure 1.

The entire computer board is run synchronously to the 3.57954Mhz color clock (C1). This is accomplished by generating a number of sub-multiple frequencies from our master 28.63636Mhz crystal oscillator. The following are the primary clocks on the board:

Name	Description
C1	The 3.579545Mhz Color Clock
C2	C1 shifted 45 degrees later
C3	C1 shifted 90 degrees later
C4	C1 shifted 135 degrees later
7M	C1 XORed with C3* (7.15909Mhz)
DAC	7M shifted 90 degrees later

7M is the processor clock for the 68000 microprocessor. C1 - C4 and DAC are used to clock the custom chips and for determining the timing of signals to the memory arrays.

The following clocks are available at the edge connector:

Name	Pin	Description
C3*	14	C3 inverted
CDAC	15	DAC equivalent
C1*	16	C1 inverted

Note that 7M (the processor clock) is not available at the connector; it can be easily generated by:

C3* XNOR C1* = 7M equivalent

If you need a 14.31818Mhz synchronous clock, you can generate it by:

(7Mequiv) XOR (CDAC) = 14M equivalent

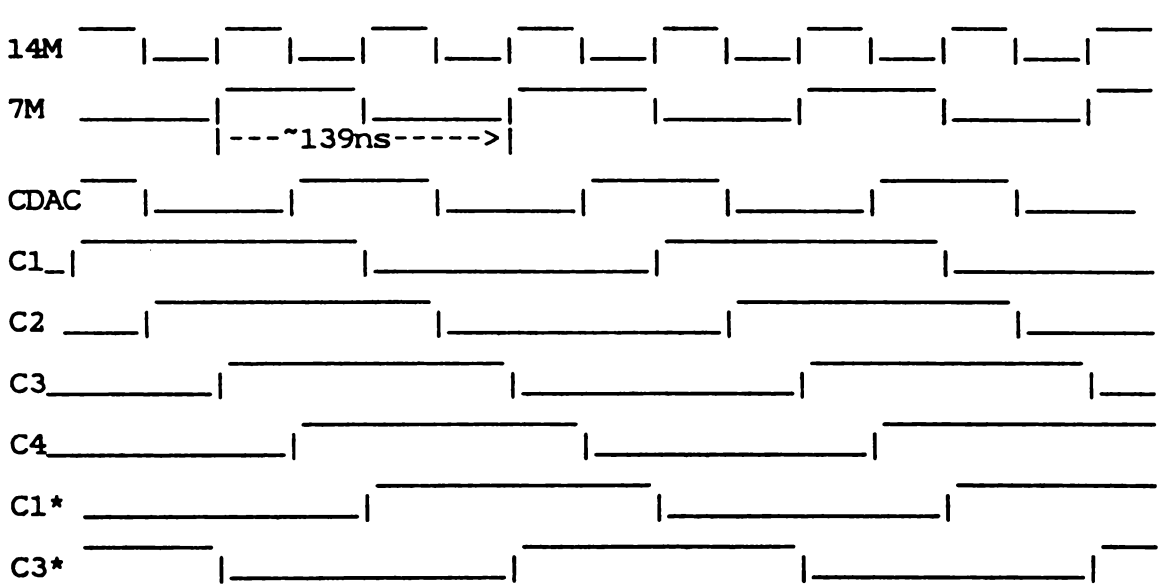


Fig. 1 Amiga System Clocks

2.2 Bus Timing

The 68000 is connected directly to the 86 pin connector, there are no buffers between the 68000 and the connector. Two control inputs, VPA* and DTACK* are driven by logic on the Amiga and should not be driven by your circuitry.

Many boxes are being designed which pass the bus (buffered) out in daisy chain fashion.
In order to allow your device to be the second in the chain, you must take into account an extra level of signal buffers on:

AS*, UDS*, LDS*, Address, Data, Clocks

We will be publishing guidelines on how to design compatible expansion boards, so that boards and boxes from various companies will be able to coexist on the bus. One of the requirements will probably be that data drivers should not be enabled until early in S4 on a Read cycle.

Futhermore, if you are designing a DMA device, the Amiga provides data in response to a Read very late (approx 50ns prior to the fall of S6). If your DMA device is looking at this data through two or three 74F245's (7ns each), this data will not be valid at your DMA controller until approximately 25ns prior to the fall of S6.

Our bus timing is based on an 8Mhz 68000, with only one exception: under normal operation our bus control PAL will assert DTACK* for you. DO NOT ASSERT DTACK*; do not attach any outputs to the DTACK* line.

2.21 Slave Bus Timing

Details of 68000 timing are available in the Motorola 68000 hardware manual. If you are designing a bus slave, most bus timing is per the 68000 spec, except that we will pull DTACK* for you. If you need to delay our assertion of DTACK*, you must pull XRDY (Pin 18) no later than 60ns after the assertion of AS*. You should release XRDY when you are ready to complete the bus cycle.

Also remember that in the expansion architecture, data drivers should not turn on during a Read cycle until S4.

For those of you who have not designed anything on the 68K bus before, this description is intended to make looking at the Motorola timing diagrams easier. For more details and timing specs see Motorola hardware manual (fold out timing diagrams in the back of the book.)

See figure 2 in this document.
Motorola labels the states of the processor clock S0-S7. The processor starts driving the address lines during S1, and asserts AS* (Address Strobe) during S2. If the cycle is a read, the data strobes (UDS*,LDS*) are asserted during S2 also (they are delayed until S4 on a write).

Our board responds to AS* by asserting DTACK* (unless you delay DTACK*

by pulling XRDY low). In order to run a normal 4 clock bus cycle, DTACK* meets the setup time prior to S5. DTACK* is the acknowledge the bus cycle. If DTACK* is not asserted, the 68000 will stay in the middle of the bus cycle until DTACK* (or BERR* or VPA*) is asserted. Once DTACK* is asserted, the processor completes the read (or write) and ends the cycle by disasserting the strobes (AS*,UDS*,LDS*) and tri-stating its bus drivers.

If the slave that you are designing cannot respond fast enough to successfully complete a 4 clock bus cycle, it must pull XRDY low within 60ns after the assertion of AS* (and of course the correct address). Our board then will not assert DTACK* until you release XRDY. You should drive XRDY with an open collector output; we provide a 1K pullup resistor on our board.

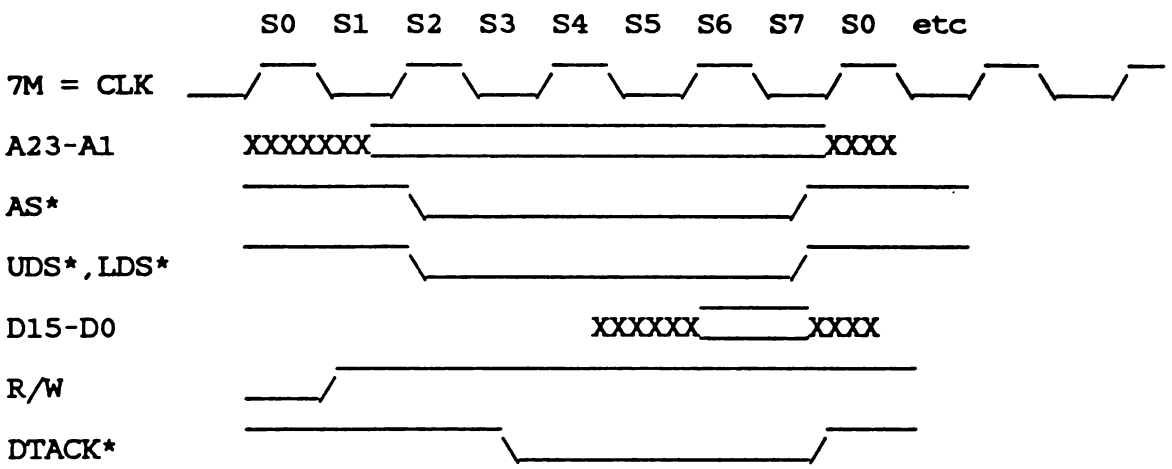


Fig. 2 Standard 4 Clock Read Cycle

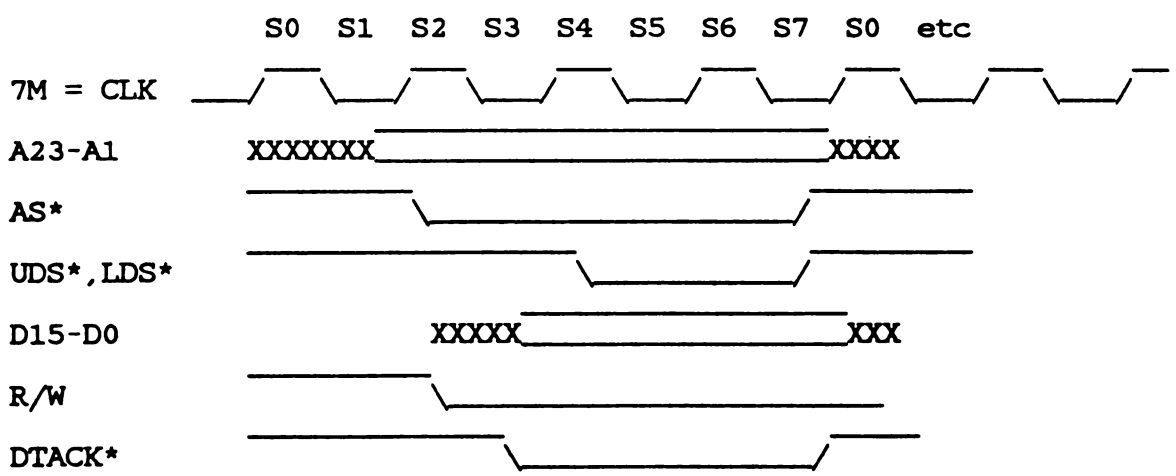


Fig. 3 Standard 4 Clock Write Cycle

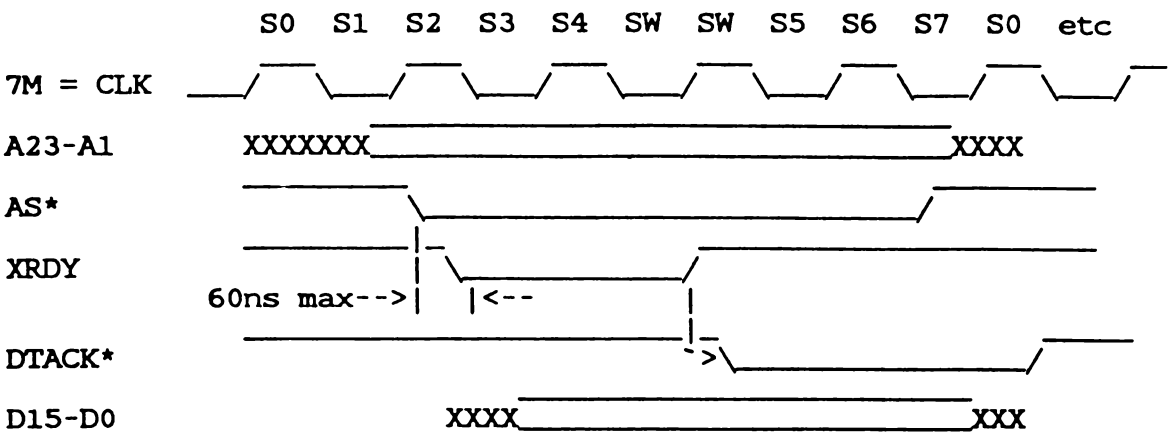


Fig. 4 Using XRDY to Delay DTACK*

2.22 Master Bus Timing

All bus masters must run synchronously to 7M (equivalent), as does the 68000 in the Amiga.

The necessary information for designing a bus master is in the 68000 hardware manual. In our system, a master must meet all of the bus timing specs of an 8Mhz 68000 , for example valid address must precede AS* by at least
30ns just as the spec says it will on an 8Mhz 68000, etc etc.

If you are designing a bus master card that will plug into a box, remember that the address will have to propagate through the address drivers that are built into the box; you should probably allow for the prop delay of three 74F245's in addition to the required 30ns.

The strobes such as AS*,UDS*,LDS* must all function as they would based on the 68000 spec. A master must also respond to DTACK*, HALT*, and BERR* correctly.

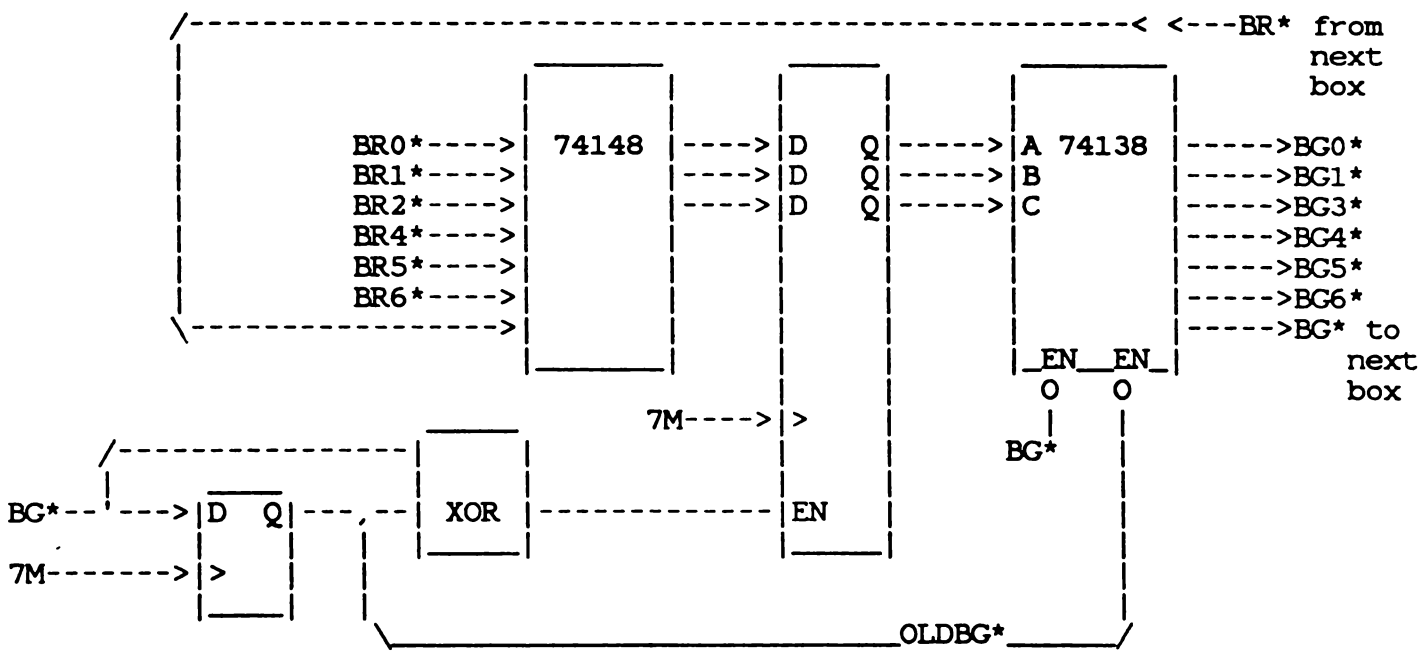
2.23 Bus Arbitration Timing

The bus arbitration logic is based on the BR* BG* OWN* BGACK* protocol as specified by Motorola in the 68000 and 68440 manuals.

All changes in state of BR* BG* and BGACK* should be clocked off the rising edge of 7M (equivalent). This allows the parallel bus arbitration logic to run synchronously to 7M without danger of clocking flip flops while these signals are changing.

The following design is for illustration, and has not been tested at this time. Notice that all state changes are synchronous to 7M and

that BGn* to the potential masters only changes when BG* from the 68000 has changed.



2.24 BGACK* And OWN* Timing to Avoid Tri-State Fights

The basic timing for bus arbitration conforms very closely to the 68000 and the 68440. When the new master has received BG* and all other signals necessary to take mastership, it must assert OWN* before it asserts BGACK*. This gives the address drivers on the bus time to change direction if necessary before BGACK* turns them

on.

At the end of the DMA cycle, BGACK* must be disasserted before OWN* is disasserted.

BR* should always be asserted off the rising edge of 7M, and should be valid no later than 60ns after that edge.

3. System Level Organization (and Idiosyncrasies)

3.1 The Address Map

Details of the address map are in ~~Appendix 2 of~~ the hardware manual.

3.3 Address Override (OVR*)

Pin 17 OVR* is a non-supported resource. You should not try to use this input without first consulting with Amiga engineering.

3.4 Interrupts

3.41 Use INT2* or INT6* (Dont Pull IPL0*-IPL2*)

There are two interrupt input lines on the Amiga: INT2* and INT6*. INT2* = pin 19, INT6* = pin 22, these lines assert levels 2 and 6 to the processor.

Do not assert the IPL0* thru IPL2* lines, because they are already driven by internal logic.

3.42 Interrupt Latency - blitter, masked ints

Interrupt latency on the Amiga is highly application software dependent, this is because the Blitter can be operated in "nasty mode" at the softwares option. If the blitter is "Nasty" and is given a lot of work to do, the processor receives very few memory cycles, so the interrupt latency will suffer.

The software can also mask out interrupts using on-board interrupt control logic.

3.5 VPA Is Not Recommendd

We recommend that you design your peripherals to run asynchronously on the 68000 bus. That is, a slow peripheral should use pulling XRDY low as a means of making the 68000 run a slower cycle. The use of XRDY to delay DTACK is discussed elsewhere in this document.

We do not recommend using VPA. If you decide to use VPA, you must pull OVR* low 30ns before asserting VPA* low. Pulling OVR* low will tri-state VPA* in the current design PAL, thus allowing your logic to drive VPA*. Pulling OVR* will also prevent DTACK* from being asserted by the PAL.

If your slave uses the VPA VMA protocol to be synchronous with the E clock, you must only use addresses in which A12 and A13 are high. This is because we have synchronous ports on board which are activated by (A12* AND VMA), also (A13* AND VMA).

3.6 Do Not Use Pins Marked EXP

Do not drive or load pins marked EXP or RESERVE.

3.7 CONFIG*

The CONFIG* output is discussed in the Appendix (Autoconfiguration).

Only use CONFIG* as a logical input. On the first version of the Amiga it is tied to ground, but on future versions it may be driven High or Low as appropriate. Do not tie this pin to an output or to VCC or GND.

4. Loading

There is 1 AMP available on the plus 5 volts.

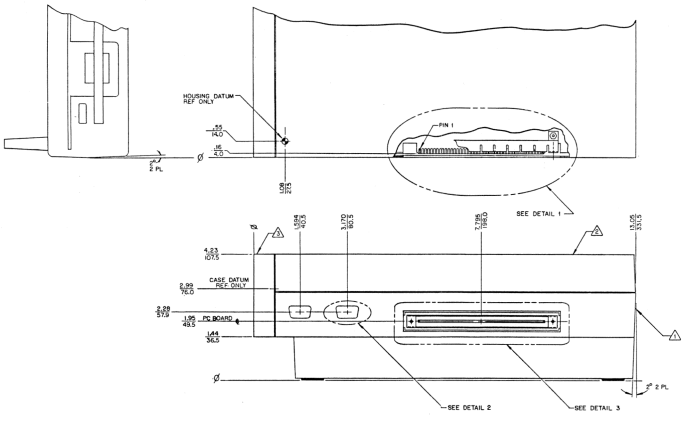
Put your buffers as close to connector as you possibly can.

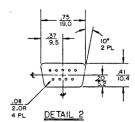
You can put one 74F load on each signal.

5. Warning

The Auto Configuration information in Appendix G of the Amiga Hardware Manual (Rev 1.0 8.27.85) is out of date and should not be implemented. We will soon be releasing information on the supported Auto Configuration Architecture.

REVISIONS				
ZONE	LTR	DESCRIPTION	DATE	APPROVED
	01		9/1/85	JMS
INIT	02	ADDED DIMS TO EJECART FINGERS	2/11/86	JMS





NOTES:

1. 327001-01 HOUSING, BOTTOM, COMPUTER
2. 327002-01 HOUSING, TOP, COMPUTER
3. 327004-01 BEZEL, COMPUTER
4. 327049-01 CONTROL DWG, PCB, COMPUTER
5. 327038-01 RF SHIELD, EXPANSION
6. COLOR—BORG WARNER NO. 33596, LIGHT BEIGE
7. TEXTURE—BEALON BK-1059
8. TO RECEIVE M3 SELF-TAPPING SCREWS FROM EXPANSION CARTRIDGE

STEWART ENGIN'G SERVICES (408) 466-2452

UNLESS OTHERWISE SPECIFIED TOLERANCES ON DECIMALS: X ±.01 .XX ±.02 .XXX ±.005 ANGULAR: ±1°

DRAWN BY: KES 9/85
CHECKED BY: [illegible]
ISSUED: [illegible]
APPR:

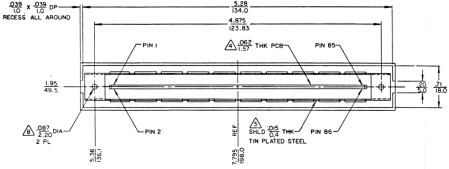
COMMODORE-AMIGA, INC.
983 UNIVERSITY AVE. #D
LOS GATOS, CA 95030

OUTLINE DRAWING
AMIGA EXPANSION

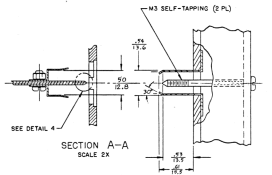
SIZE D | 327275-01 | REV 02
SCALE 1/1 | SHEET 1 OF 2

FINISH: [illegible]
USED ON: [illegible]
NEXT ASSY: [illegible]

REVISIONS				
ZONE	LTR	DESCRIPTION	DATE	APPROVED

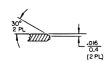


DETAIL 3
SCALE 2X

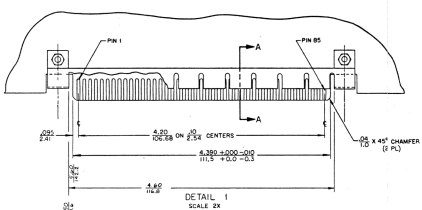


SECTION A–A
SCALE 2X

EXPANSION CARTRIDGE
(REF ONLY)



DETAIL 4
SCALE 4X

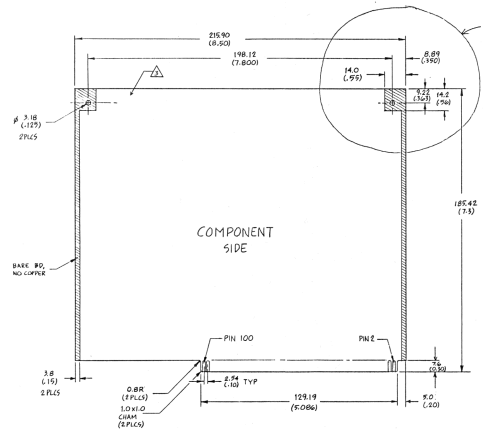


DETAIL 1
SCALE 2X

STEWART ENGINEERING SERVICES (408) 466-2452

UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES MILLIMETERS ARE IN BRACKETS	DRAWN BY: S.E.S.	DATE: 9-88	CAMADOBE AMIGA, INC. 100 UNIVERSITY AVE. #D LOS GATOS, CA 95030
# .XX .XXX ±% ∠	ENGR		
MATERIAL	MFG		OUTLINE DRAWING AMIGA EXPANSION
FINISH			SIZE D 327275-01 REV C1
			SCALE 1/1 SHEET 2 OF 2

REVISIONS				
ZONE	LTR	DESCRIPTION	DATE	APPROVED
	04	REDRAWN	2/11/86	MHS
	05	129.10 WAS PLACED WRONG	3/14/86	MHS
	06	ADDED PANEL MOUNTING HOLES, CHANGED BASE COPPER AREA.		



NOTES:
1. EVEN PIN NO.s ON COMPONENT SIDE.
2. ODD PIN NO.S ON CIRCUIT SIDE.
3. EXTERNAL CONNECTORS ALONG THIS EDGE.

PRELIMINARY !

METRIC
(ENGLISH)

UNLESS OTHERWISE SPECIFIED TOLERANCES ON: DECIMALS .XX ±.4 (.02), .XXX ±.25 (.005), ANGLES ±1°

DRAWN BY: MHS — DATE: 2/11/86
CHKD:
ENGR:
APPR:

MATERIAL:
FINISH:
USED ON
NEXT ASSY

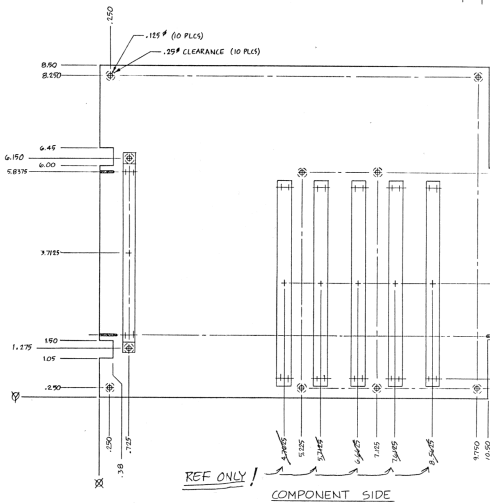
COMMODORE-AMIGA, INC.
983 UNIVERSITY AVE. #D
LOS GATOS, CA 95030

CONTROL DING, EXPANSION BD

SIZE C | 327290-01 | REV 06

SCALE 1:1 | SHEET 1 OF 1

REVISIONS				
ZONE	LTR	DESCRIPTION	DATE	APPROVED



NOTES:

1. MATL: FR4 .090 THK
2. GOLD FINGERS
3. THESE DIMENSIONS ARE NOT CORRECT FOR PRODUCTION. THEY ARE TO BE USED ON THIS PROTOTYPE PCB ONLY.

UNLESS OTHERWISE SPECIFIED TOLERANCES ON: DECIMALS			ANG
.X ±.05	.XX ±.03	.XXX ±.3	±

MATERIAL:

FINISH:

DRAWN BY:	DATE
HKSH	11/3/86
CHKD:	
ENGR:	
APPR:	
USED ON:	NEXT ASSY:

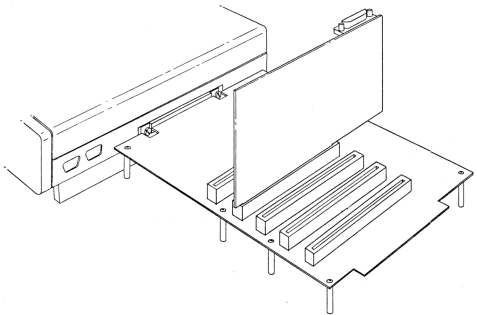
COMMODORE-AMIGA, INC.
983 UNIVERSITY AVE. RD
LOS GATOS, CA 95030

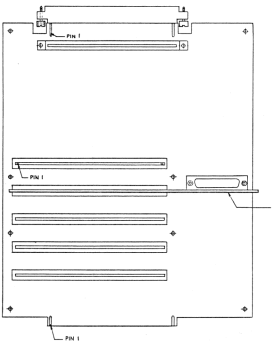
CONTROL DWG.

ZORRO BACKPLANE BD

SIZE		REV
C	327310-01	07
SCALE	SHEET 1 OF 1	

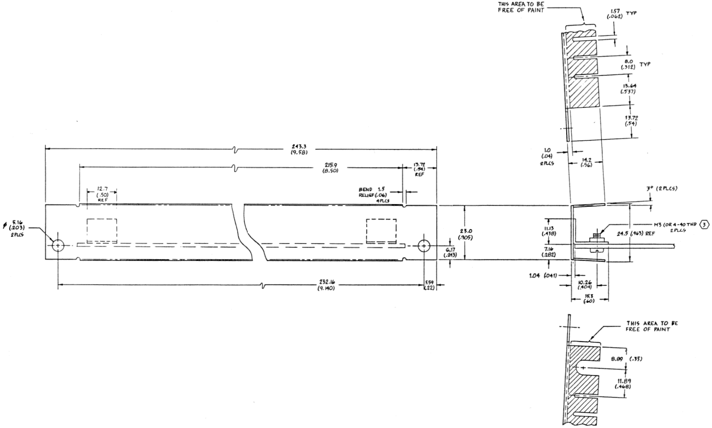
REVISIONS				
ZONE	LTR	DESCRIPTION	DATE	APPROVED

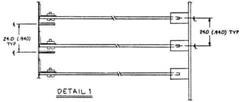




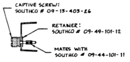
UNLESS OTHERWISE SPECIFIED DIMENSIONS TOLERANCES ON DECIMALS: .X ±.1 .XX ±.05 .XXX ±.010 ANGLES ±1°	DRAWN BY: JAY G.	DATE: 7/14/86	COMMODORE-AMIGA, INC. 983 UNIVERSITY AVE. #D LOS GATOS, CA 95030
	CHKD:		
	ENGR:		LAYOUT, Ex. BOX
	PROJ. ENG:		
MATERIAL:	USED ON:	NEXT ASSY:	SIZE: D 327314-01 REV: 01
FINISH:			SCALE: SHEET 1 OF 1

ZONE	LTR	DESCRIPTION	DATE	APPROVED
	01	PROTO RELEASE	4/6/86	HWS
	02	ADDED PAINT NOTE, BRKT WIDTH	4/11/86	HWS
	03	CHG'D PCB & SCREW HEIGHT	4/16/86	HWS





DETAIL 1
SCALE 1:1



DETAIL 2
2 EA REQ'D
SCALE 1:1

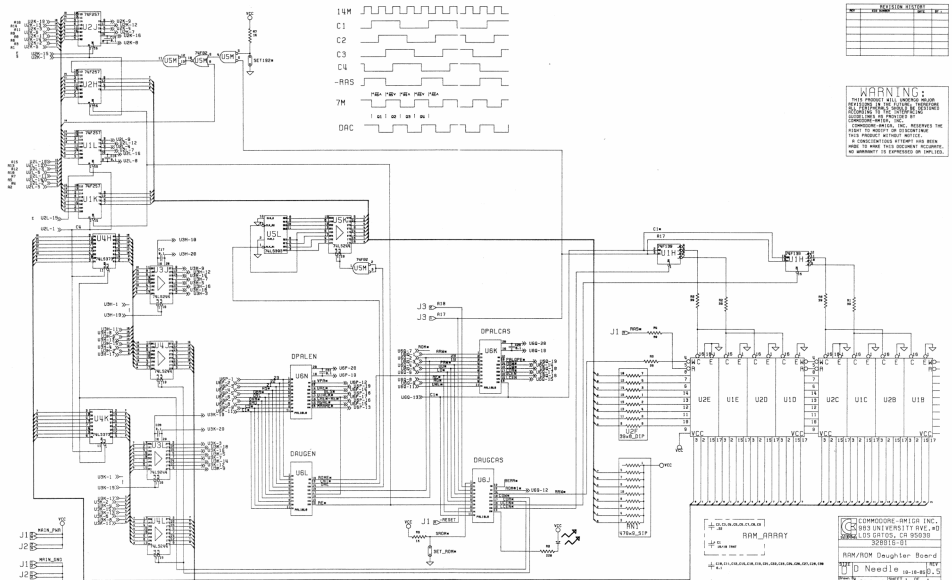
NOTES:

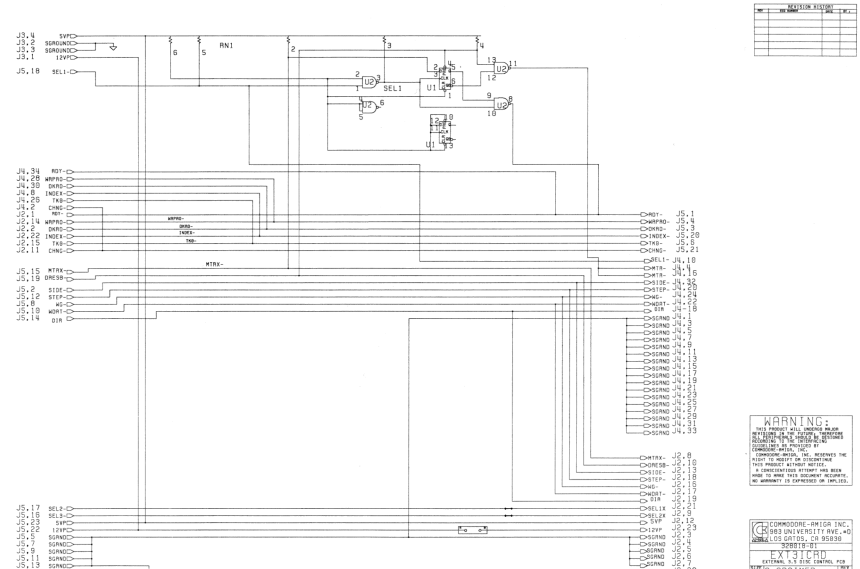
1. MATL: 0.73 (.029) THK CR STEEL, AISI 1010.
2. FINISH: PAINT - HIGH SOLIDS, POLYURETHANE BASE, SEMI GLOSS, SMOOTH COAT, CLASS-A (WHERE INDICATED) OVER CLEAR ZINC.
   COLOR - LT BEIGE: PANTONE PRO.NO. 130905.
3. PANEL SUPPORT USING 2 ANGLE BRACKETS AS SHOWN OR 2 SUB-MINI 'D' CONNRS.
4. SEE EXPANSION PCB CONTROL DWG 327290-01.

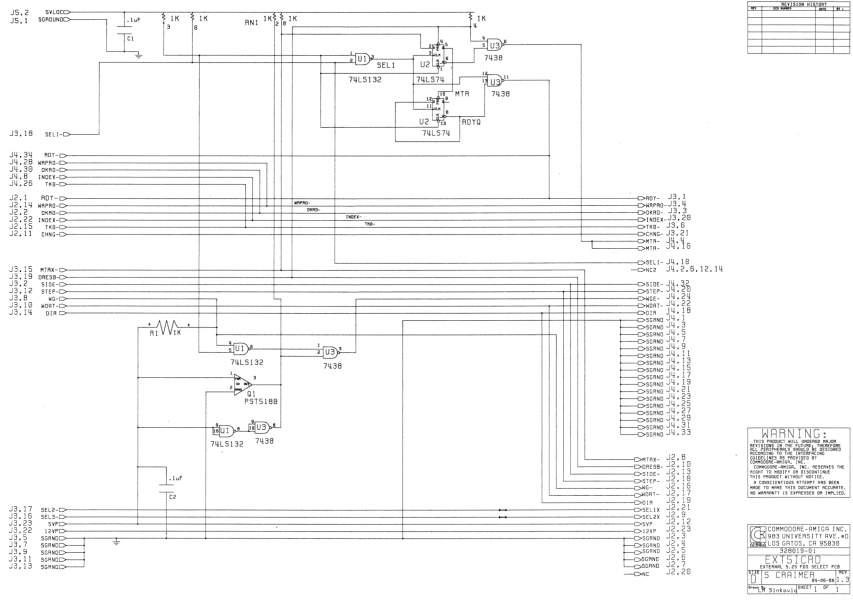
SERIOUSLY
PRELIMINARY!

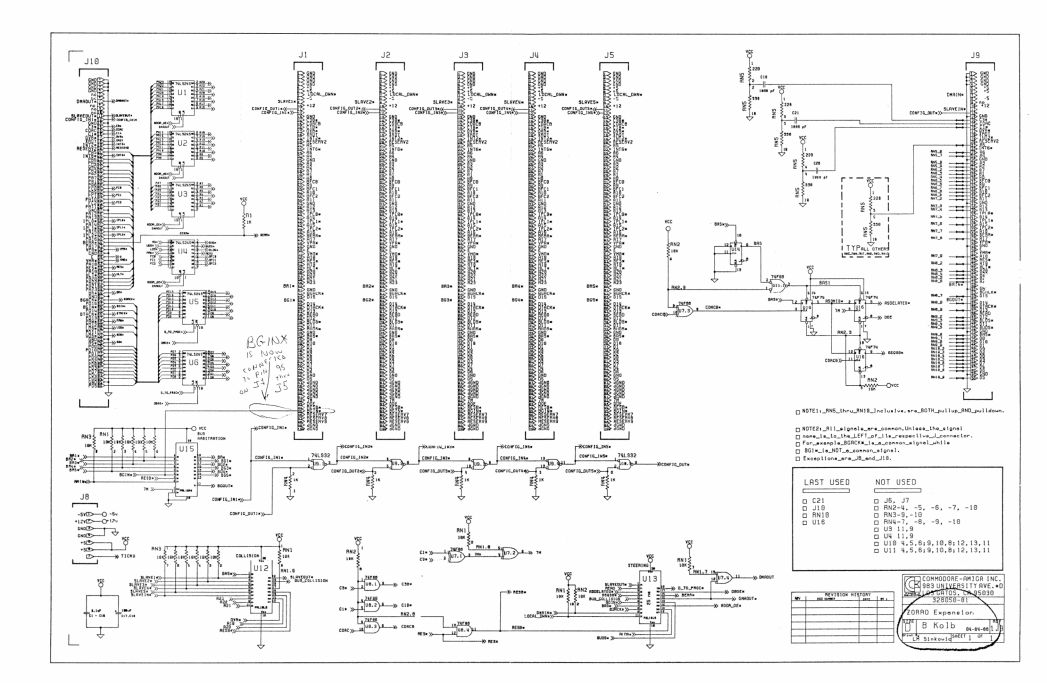
METRIC (INCHES)

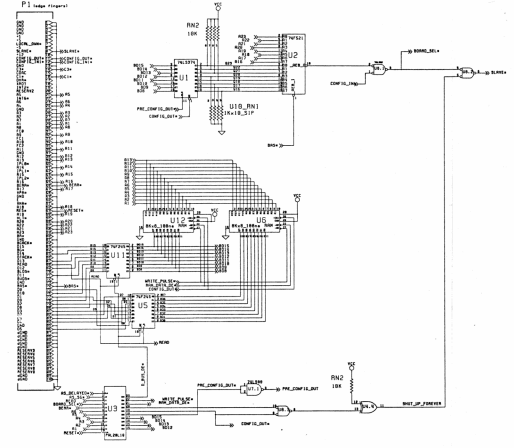
UNLESS OTHERWISE SPECIFIED TOLERANCES ON: DECIMALS .X ±.5 (.02) .XX ±.25 (.01) .XXX ±(.005) ∠S ±1°	DRAWN BY HWS, DATE 4/6/86		COMMODORE-AMIGA, INC. 983 UNIVERSITY AVE. #D LOS GATOS, CA 95030
	CHKD		CONTROL DWG, END PANEL, EX. BD
	ENGR		
	APPR		
MATERIAL: SEE NOTE 1.	USED ON	NEXT ASSY	SIZE D 327335-01 REV 03
FINISH: SEE NOTE 2.			SCALE 2:1 SHEET 1 OF 1


14M
C1
C2
C3
C4
-RAS
7M
DAC
REVISION HISTORY
WARNING:
THIS PRODUCT WILL UNDERGO MAJOR REVISIONS IN ITS FUTURE. TERMINATIONS AND CONNECTIONS MAY BE SUBJECT TO CHANGE. ALL RIGHTS RESERVED TO THE CUSTOM BY COMMODORE-AMIGA, INC.
COMMODORE-AMIGA, INC. RESERVES THE RIGHT TO MODIFY OR DISCONTINUE THIS PRODUCT WITHOUT NOTICE.
NO CONSIDERATIONS HYTGHT HAS BEEN MADE TO MAKE THIS DOCUMENT ACCURATE. NO WARRANTY IS EXPRESSED OR IMPLIED.
DPRLEN
DPRLCAS
DRUGEN
DRUGCAS
RAM_ARRAY
COMMODORE-AMIGA INC.
983 UNIVERSITY AVE, #D
LOS GATOS, CA 95030
328818-01
RAM/ROM Daughter Board
D Needle
10-16-85
Sinkwold
SHEET 1 OF 1


REVISION HISTORY
WARNING:
THIS PRODUCT WILL UNDERGO MAJOR REVISIONS IN THE FUTURE. IT MAY NOT BE MADE TO FUNCTION CORRECTLY BECAUSE ACCORDING TO THE INTERFACING SPECIFICATIONS SUPPLIED BY COMMODORE-AMIGA, INC.
COMMODORE-AMIGA, INC. RESERVES THE RIGHT TO MAKE CHANGES TO THIS PRODUCT WITHOUT NOTICE.
A CONSCIENTIOUS ATTEMPT HAS BEEN MADE TO MAKE THIS DOCUMENT ACCURATE. NO WARRANTY IS EXPRESSED OR IMPLIED.
COMMODORE-AMIGA INC.
1983 UNIVERSITY AVE.,#D
LOS GATOS, CA 95030
320018-01
EXT2ICRO
U S CRAIMER
SHEET 1 OF 1


REVISION HISTORY
REV
DESCRIPTION
ECO #
DATE
APP.
J5.2 SYLGC
J5.1 SGROUND
1uF
C1
1K
RN1
1K
U1
74LS132
SEL1
U2
74LS74
MTR
U3
7438
RDYQ
J3.18 SEL1-
J4.34 RDY-
J4.28 WRPRO-
J4.30 DSKD-
J4.8 INDEX-
J4.26 TKO-
J2.1 RDY-
J2.14 WRPRO-
J2.6 DSKD-
J2.22 INDEX-
J2.15 TK0-
J2.11 CHNG-
J3.15 MTRX-
J3.19 DMESB-
J3.2 SIDE-
J3.12 STEP-
J3.8 WG-
J3.10 WDAT-
J3.14 DIR
R1 1K
Q1
PST51BB
C2 .1uF
J3.17 SEL2-
J3.16 SEL3-
J3.23 5VP
J3.22 12VP
J3.5 SGND
J3.7 SGND
J3.9 SGND
J3.11 SGND
J3.13 SGND
J4.1 SGND
J4.2 SGND
J4.5 SGND
J4.7 SGND
J4.9 SGND
J4.11 SGND
J4.13 SGND
J4.15 SGND
J4.17 SGND
J4.19 SGND
J4.21 SGND
J4.23 SGND
J4.25 SGND
J4.27 SGND
J4.29 SGND
J4.31 SGND
J4.33 SGND
J3.1 RDY-
J3.4 WRPRO-
J3.20 INDEX-
J3.6 TK0-
J3.21 CHNG-
J4.4 MTR-
J4.16 MTR-
J4.10 SEL1-
J4.2,6,12,14 NC
J4.32 SIDE-
J4.20 STEP-
J4.24 WG-
J4.22 WDAT-
J4.18 DIR
J2.8 MTRX-
J2.10 DMESB-
J2.13 SIDE-
J2.17 STEP-
J2.15 WG-
J2.19 WDAT-
J2.21 DIR
J2.9 SELIX
J2.12 5VP
J2.23 12VP
J2.3 GND
J2.4 GND
J2.5 GND
J2.6 GND
J2.7 GND
J2.20 NC
WARNING:
THIS IS A PRELIMINARY DRAWING AND IS SUBJECT TO CHANGE AS OUR REVISIONS OR FOR CORRECTIONS. ALL SPECIFICATIONS AND GUIDELINES AS PROVIDED BY COMMODORE-AMIGA, INC.
COMMODORE-AMIGA, INC. RESERVES THE RIGHT TO MODIFY OR DISCONTINUE THIS PRODUCT AT ANY TIME.
A CONSCIENTIOUS EFFORT HAS BEEN MADE TO MAKE THIS DOCUMENT ACCURATE, NO WARRANTY IS EXPRESSED OR IMPLIED.
COMMODORE-AMIGA INC.
1983 UNIVERSITY AVE.,#D
LOS GATOS, CA 95030
328018-01
EXTSTCAP
EXTERNAL 5.25 FDD SELECT PCB
DRAWN J S CRAIMER
DATE 04-06-84
SHEET 1 OF 1
APPROVED N Stokes


J18
J1
J2
J3
J4
J5
J9
J8
U15
U12
U13
U11
U2
U3
U5
U16
BGMUX IS NOT CONNECTED TO ANY OF THE 35 PINS
NOTE1: RN5_thru_RN18_inclusive are BOTH pullup AND pulldown.
NOTE2: All signals are common Unless the signal name is in the LEFT or the respective J connector.
For example SCREEN is a common signal while BGIn is NOT a common signal.
Exceptions are J8 and J18.
LAST USED
C21
J18
RN18
U16
NOT USED
J6, J7
RN2-4, -5, -6, -7, -10
RN3-9, -10
RN4-7, -8, -9, -10
U3 11,9
U4 11,9
U10 9,5,6,9,10,8;12,13,11
U11 4,5,6,9,10,8;12,13,11
REVISION HISTORY
COMMODORE-AMIGA INC.
1983 UNIVERSITY AVE.
NORWALK, CA 90650
XXXXX-XX
ZORRO Expansion
B Kolb
04-06-85
Sheet 1 of 1



WARNING:
THIS PRODUCT WILL UNDERGO MAJOR REVISIONS AND IMPROVEMENTS. IT IS NOT FULLY TESTED AND MAY BE SPECIFIED DIFFERENTLY IN THE FUTURE. THE GUIDELINES AS PRESCRIBED BY COMMODORE-AMIGA, INC. RESERVES THE RIGHT TO ADJUST OR DISCONTINUE THIS PRODUCT WITHOUT NOTICE.
A CONSTITUTOR INTENT HAS BEEN MADE TO INSURE THE ACCURACY, BUT NO WARRANTY IS EXPRESSED OR IMPLIED.

COMMODORE-AMIGA INC.
1983 UNIVERSITY AVE.#D
LOS GATOS, CA 95030

320050-01

Expansion Test RAM

REV 1.1 | SIZE J | DRAWN B Kolb | 01-15-88 | SHEET 1 OF 1 | APPROVED M Simonsen

This was brought to you

from the archives of

http://retro-commodore.eu