Acorn expansion card specification

(formerly *Podule Electrical Specification*)

Acorn expansion card specification

Published by Acorn Computers Technical Publications Department

Within this publication, the term 'BBC' is used as an abbreviation for 'British Broadcasting Corporation'.

Published by Acorn Computers Limited
Part number 0472,200
Issue 3, January 1992

Acorn expansion card specification

Acorn expansion card specification

Introduction

This document outlines the I/O system implemented on current Acorn Archimedes, R-series and BBC A3000 computer systems. For more information on the RISC OS software interface with expansion cards, and how to write loaders for them, see *Expansion Cards* in the *RISC OS Programmer's Reference Manual* (*Expansion cards and Extension ROMs* in the *RISC OS 3 PRM*).

Expansion cards for Archimedes computers were formerly known within Acorn as *podules*, and some relics of this nomenclature persist in software.

It is important to realise that future systems may have a different implementation of the I/O system, and in particular the addresses (and number) of expansion card locations may change. For this reason, and to ensure that any device may be plugged into any slot, all driver code for expansion cards must be relocatable. References to the direct expansion card addresses should never be used. It is up to the machine operating system, in conjunction with the expansion card id, to determine the address at which an expansion card should be accessed. To this extent, the sections on *System architecture*, *System memory map* and *I/O space memory map* are for background information only.

Some models of computer, such as the A400/1 series, also support a co-processor card on expansion card slot 2. The co-processor interface is a superset of the expansion card interface. The co-processor interface is for Acorn use only, and is not described in this document.

It is assumed that the reader is familiar with the basic concepts of computer systems built around the ARM micro-processor unit (MPU) and its associated support devices, MEMC, VIDC and IOC.

In this document logic low active signals are indicated by a bar over the signal name, e.g. $\overline{BL}$.

Detailed specifications on the ARM chip set can be found in the Prentice Hall publication:

Acorn RISC Machine Family Data Book
ISBN 0-13-781618-9.

Physical dimensions

The expansion card printed circuit board (PCB) mechanics follow the Eurocard format. Either single (100mm) or double (233.4mm) width cards of standard length (160mm) may be used. All cards have a 25.4mm high (5 HP) metal back panel, for mounting externally-accessible connectors.

Each expansion card has a single 64-way DIN 41612ac connector, with the a and c rows loaded. A double-width expansion card, when viewed from above, with the metal back panel towards you, should have the DIN 41612ac connector fitted in the lefthand position.

The A3000 internal expansion card is 6.37 inches wide and 2.675 inches long. It has a 25.6 mm high metal back panel for mounting externally accessible connectors.

Further details are given in the section entitled *Mechanical Specification* on page 25.

Types of expansion card

All I/O is memory-mapped. MEMC decodes the MPU address space and generates an $\overline{IORQ}$ signal to indicate I/O space is being accessed. I/O cycles are terminated when MEMC receives an $\overline{IOGT}$ signal. There are two types of expansion card access cycle which are distinguished by the way $\overline{IOGT}$ is generated:

- The simple access cycle, controlled by IOC
- The MEMC access cycle, where the expansion card logic returns $\overline{IOGT}$.

Simple access expansion cards

For simple expansion cards, IOC controls the I/O cycle by returning $\overline{IOGT}$ to MEMC. IOC offers four different access timings, selected by address value. All expansion cards must have simple expansion card logic, for the expansion card id system.

MEMC access expansion cards

With MEMC access cycles the expansion card logic works directly with the MEMC $\overline{IORQ}$ and $\overline{IOGT}$ signals. This means the expansion card designer can create optimised cycle timings for his application. System software should not access MEMC space unless it is certain that an expansion card is present to return $\overline{IOGT}$. If $\overline{IOGT}$ is not returned, the I/O system remains hung up waiting for it. While in the hung state the MPU clocks are stopped and only the video display process can continue. After 10 micro seconds – for ARM2 – the MPU register state cannot be relied upon. The computer must be restarted with a system reset.

System architecture

The I/O system (which includes expansion card devices) consists of a 16-bit data bus (BD[0:15]), a buffered address bus (LA[2:21]) and various control and timing signals. The I/O data bus is independent from the main 32-bit system data bus, being separated from it by bidirectional latches and buffers (see Figure 1: *System architecture* on page 2). In this way the I/O data bus can run at much slower speeds than the main system bus to cater for slower peripheral devices. The latches between the two buses are controlled by the I/O controller, IOC. The IOC caters for four different cycle speeds (slow, medium, fast and synchronous) and the programmer can choose the most suitable cycle for a particular device.

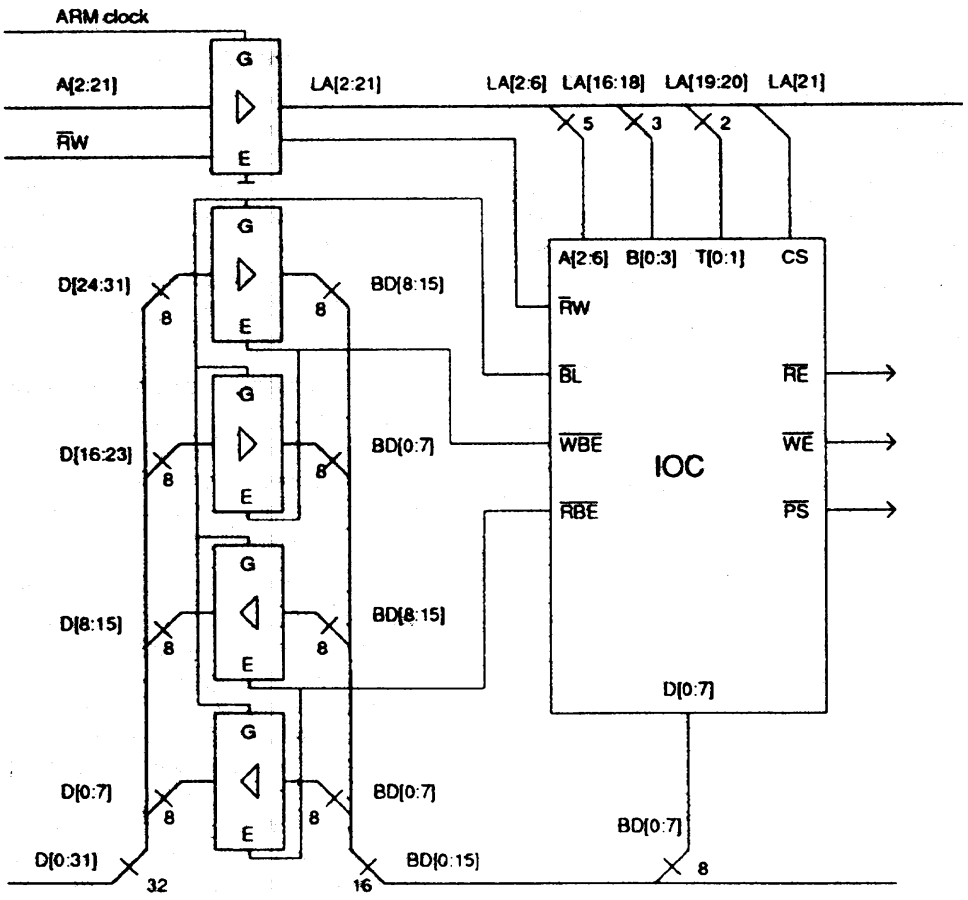


Figure 1: System architecture

A typical I/O system address decoder is shown in Figure 2: *Typical I/O system with expansion cards fitted* on page 3. For clarity, the main data and address buses are omitted from this diagram.

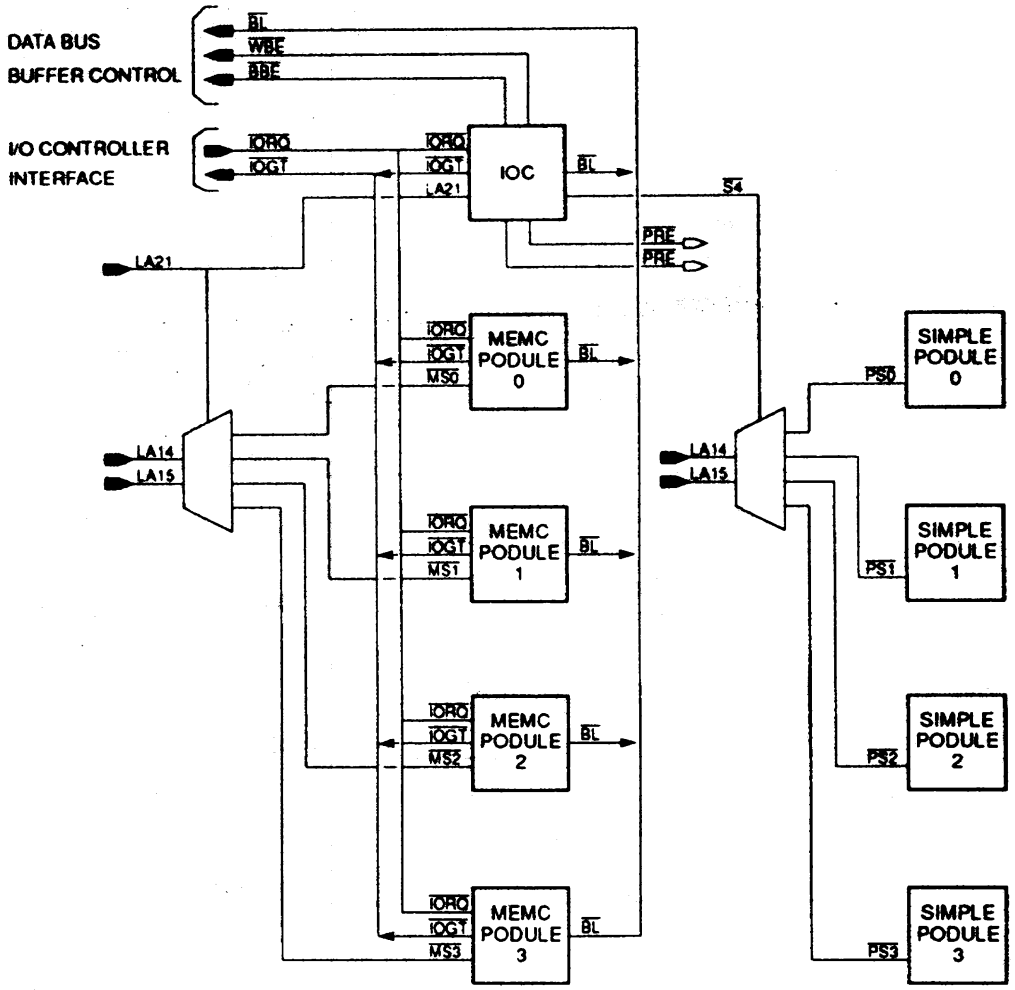


Figure 2: Typical I/O system with expansion cards fitted

A single expansion card may have just simple expansion card logic on it, or both simple and MEMC logic, but not MEMC logic without simple logic.

The number of expansion card slots varies with computer model:

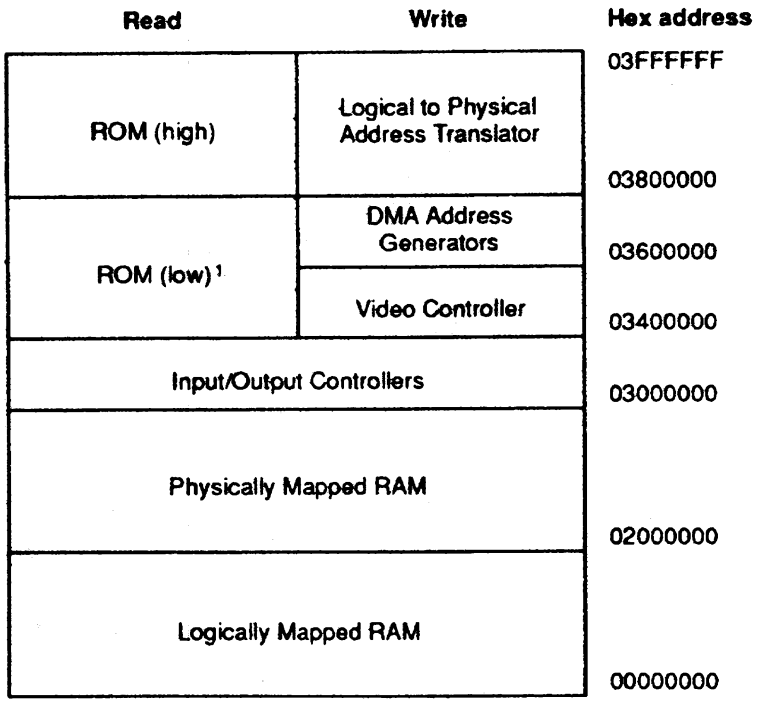
- A305/310 have an optional 2 slot backplane
- A440 and A400/1 series have a 4 slot backplane
- A3000 has one internal and one external expansion card connector.
- A540 has a 4 slot backplane, one occupied by a SCSI card.
- A5000 has a 4 slot backplane, optional on some models.

Non-IOC devices

The IOC controls devices in the upper half of the I/O space. The lower half of the I/O space may be used by other devices which are not timed by the IOC. Such devices (e.g. MEMC expansion cards) share the same handshaking control lines to the MEMC as the IOC. The advantage of devices in the MEMC is that they are not tied to one of the four IOC I/O cycle timings.

System memory map

The system memory map is defined by the MEMC, and is shown in Figure 3: *System memory map* on page 4. Note that all system components, including I/O devices, are memory mapped.



The shaded areas are accessible only to processes running in supervisor mode.
[1]Not used in current designs.

Figure 3: System memory map

I/O space memory map

The I/O space is split into two, with the upper half being controlled by IOC, and the lower half allocated to non-IOC mapped devices (e.g. MEMC expansion cards). The IOC-controlled space has allocations for simple expansion cards and external expansion cards. Note that external expansion cards are not supported by the RISC OS operating system.

The MEMC expansion card and simple expansion card spaces are divided into four equal parts, with one part being allocated to each physical expansion card slot (see Figure 4: *I/O space memory map* on page 5).

Present systems implement up to four expansion card slots, designated slots 0 to 3. Each physical slot is identical, but each appears separately in the address space. The way in which the existence of an expansion card is identified is explained in the section entitled *Expansion card identity* on page 7.

Note that in some designs LA16:LA20 are not used by the expansion card decoder logic, so expansion card slots may repeat in the I/O space. The operating system uses the lowest occurrence of each slot in the address range.

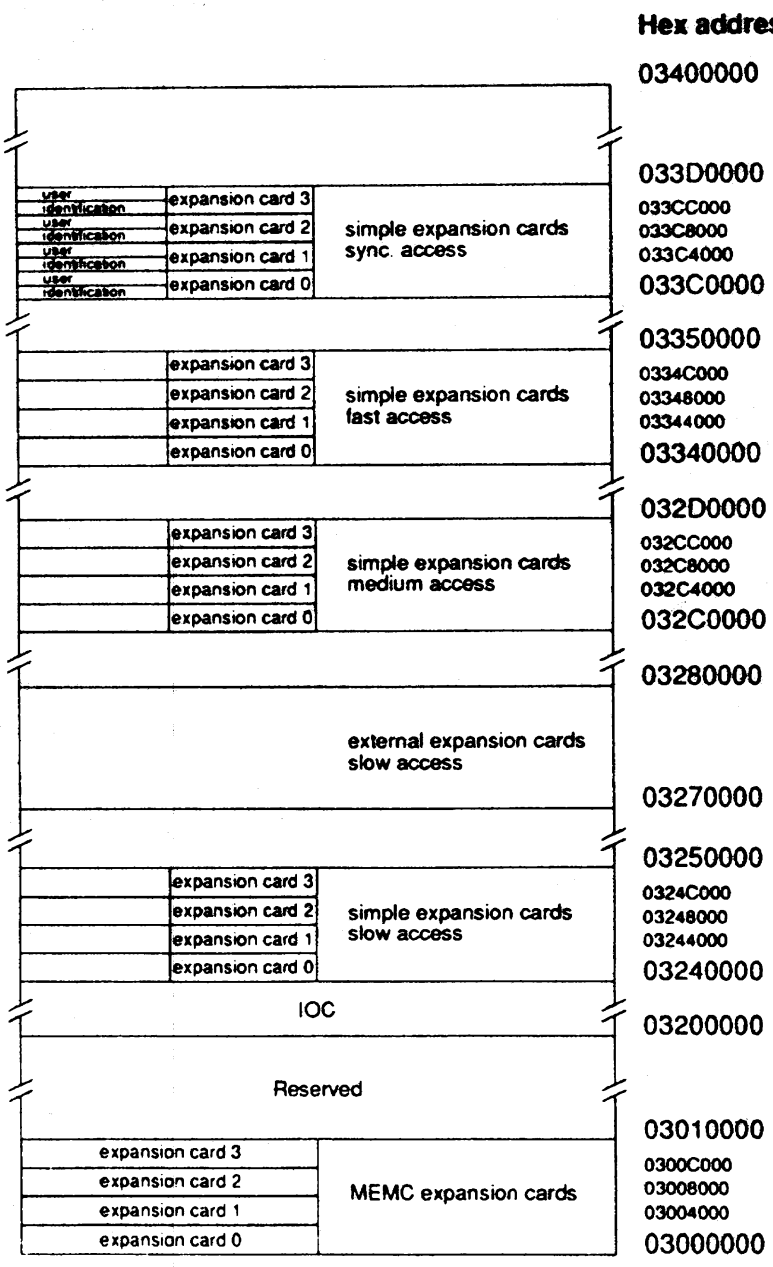


Figure 4: I/O space memory map

Expansion card access speed

Simple expansion cards

The simple expansion cards are mapped through the IOC, and may be accessed at one of four different cycle speeds, as determined by the address at which they are selected. The four cycle types are designated slow, medium, fast and synchronous. Their timings are detailed in the section entitled *Simple expansion cards* on page 13. Their address mapping is shown in Figure 4: *I/O space memory map* on page 5.

MEMC expansion cards

The cycle timing of a MEMC expansion card access must be controlled by the expansion card itself. A simple state machine clocked by the 8 MHz reference signal (REF8M) can be used to control these cycles. Refer to the section entitled *MEMC expansion cards* on page 20.

Expansion card size

The expansion card data bus can be either eight or sixteen bits wide, allowing both byte and half-word access. Each expansion card slot has 4096 word addresses (for example, 03xx4000, 03xx4004, 03xx4008 etc). See Figure 4: *I/O space memory map* on page 5 for the exact address range occupied by each expansion card in each of its five address modes (synchronous, fast, medium, slow and MEMC). At each word address, because of the data bus width restriction, it may only be possible to access either a byte or a half-word. Therefore, the size of an expansion card which is only byte-wide is 4K, and the size of a half-word expansion card is 8K. Some of this space will be used by the expansion card identification, the exact amount being chosen by the designer. Refer to the section entitled *Expansion card identity* on page 7.

Note that access to expansion card space must always take place at word addresses.

Data bus mapping

The I/O data bus (BD[0:15]) connects to the main system data bus (D[0:31]) via a set of bidirectional data latches.

The mapping of the BD[0:15] bus onto the D[0:31] bus is as follows:

- During a WRITE (i.e. ARM to peripheral) BD[0:15] is mapped to D[16:31]
- During a READ (i.e. peripheral to ARM) BD[0:15] is mapped to D[0:15].

Byte accesses

To access bytewide expansion cards, byte instructions should be used. When a byte store instruction is executed, the MPU will place the written byte on all four bytes of the word, and will therefore correctly place the desired value on the lowest byte of the I/O bus. A byte or word load may be used to read a bytewide expansion card into the lowest byte of an ARM register. For example:

```
...
...
LoadByte
LDRB Rdata, [Raddress]
...
...
StoreByte
STRB Rdata, [Raddress]
...
...
```

Half-word accesses

To access a 16-bit wide expansion card, word instructions should be used. When storing, the half-word must be placed on the upper 16 bits, D[16:31]. To maintain upwards compatibility with future machines, half-word stores should replicate the written data on the lower half-word, D[0:15]. When reading, the upper 16 bits are undefined.

For example:

```
...
...
LoadHalfWord
LDR Rdata, [Raddress]
MOV Rdata, Rdata, ASL#16
MOV Rdata, Rdata, ASR#16
...
...
StoreHalfWord
MOV Rdata, Rdata, ASL#16
ORR Rdata, Rdata, Rdata, LSR#16
STR  Rdata, [Raddress]
...
...
```

Word accesses

32-bit (word-wide) expansion cards are not currently supported by the expansion card bus.

Expansion card interrupt handling

There are two interrupt lines on the expansion card bus, $\overline{\text{PIRQ}}$ and $\overline{\text{PFIQ}}$. Both lines are vectored through the IOC and generate ARM $\overline{\text{IRQ}}$ and $\overline{\text{FIQ}}$ signals respectively. $\overline{\text{PIRQ}}$ is the normal interrupt request line, and appears as bit 5 in the IOC IRQ status B register (Hex address 03200020). $\overline{\text{PFIQ}}$ is the fast interrupt request line, and appears as bit 6 in the IOC FIQ status register (Hex address 03200030). For further details on interrupt handling, refer to the IOC data sheet. Note that future systems may have a different implementation of the I/O system, and in particular the addresses (and number) of expansion card locations may change.

An expansion card generating an $\overline{\text{IRQ}}$ interrupt must drive the $\overline{\text{PIRQ}}$ line low. Both interrupt lines have a resistive pullup of 1k2Ω. In order that the ARM can determine which expansion card is generating the interrupt, an expansion card which is driving the $\overline{\text{PIRQ}}$ line low must also set its IRQ status bit high.

An expansion card generating a $\overline{\text{FIQ}}$ interrupt must drive the $\overline{\text{PFIQ}}$ line low. In order that the operating system can

determine which expansion card is generating the interrupt, a card which is driving the PFIQ line low must also set its FIQ status bit high.

Some variants of the computer (Archimedes 400/1, 540, A5000 and R-Series, for example) have extra logic on the backplane PCB, for expansion card interrupt management. The default/power on state of the logic leaves expansion card interrupts enabled, i.e. the logic can be ignored and the system will behave identically to the A300 and early A440 models. Two functions are added by the extra logic, a mask register and a status register. The logic is fitted to support RISC iX.

Expansion card interrupt mask register

This register allows individual expansion card IRQ interrupts to be masked off, and provides a means of implementing an interrupt priority scheme for expansion cards.

Writing a '0' to a bit in the expansion card interrupt mask register disables interrupts from the corresponding slot on the backplane.

Writing a '1' to a bit in the expansion card interrupt mask register enables interrupts from the corresponding slot on the backplane.

The mechanism for identifying which slot is generating the interrupt is described in the section entitled *Expansion card interrupt status register* on page 7.

The mechanism for clearing the interrupt from a particular slot will depend on the device installed in that slot.

The table below shows the correspondence between bit position and slot number.

BD7	BD6	BD5	BD4	BD3	BD2	BD1	BD0
RSVD	RSVD	RSVD	RSVD	slot 3	slot 2	slot 1	slot 0

(RSVD = Reserved.)

Expansion card interrupt status register

This read only register allows the processor to identify which expansion card is generating the interrupt without scanning the IRQ flag on each expansion card.

A logic one read in a bit position indicates that the slot is enabled and interrupting. The status register uses the same bit allocation as the mask register.

Layout and Drive

The drive capability of the expansion card interface is limited, so expansion cards should offer the lowest practical load. Expansion card PCB track layout should minimise track length to the DIN edge connector, to avoid ringing or excessive capacitive loading of the interface signals.

Track lengths from the DIN41612 connector should always be less than 50mm with the REF8M tracking to be less than 25mm.

Signals driving the expansion card interface should include series damping resistors if possible, to reduce ringing and ground bounce problems in the card and the computer. About 68 ohms is usually sufficient.

Data signals BD[0:15] should have a maximum logic low input current of -0.4mA, e.g. one LS TTL gate input load. The recommended drive capability of drivers in the data bus is 6mA (e.g., an HC series bus driver). The use of HC or AC family logic is recommended.

Address lines should have a maximum logic low input current of -1.2mA; again the use of HC or AC logic is recommended.

Control signals: the logic low input current of control signals, e.g. CLK2, CLK8 and REF8M should be less than -0.4mA (e.g. one LS TTL gate input load, input load = 20 pf max). Again the use of HC or AC logic is recommended.

Open drain/open collector drivers should be able to sink at least 6mA (1k2 pull up, plus four LS TTL gate input loads) and still achieve a logic low voltage of less than 0.5 Volts.

All output signals from the computer to the expansion card interface are CMOS logic compatible. Expansion cards may drive the interface with TTL levels, but CMOS logic levels are recommended. The signals RST, PIRQ, PFIQ, IOGT and BL are open drain/collector with a resistive pull-up.

In the case of the A3000 internal expansion card interface, it is recommended that the load on each signal does not exceed 3 HCT gates. It is also recommended that the drivers provided on the data bus be capable of driving at least 7 HCT and 3 TTL loads.

Heating

The power allocated for an external expansion card (for example, the A3000) should not be dissipated within the case even if no external card is fitted. This extra heating may create excessive temperatures within the case.

If an Archimedes is fitted with a 4-way backplane, a fan must be installed to improve the ventilation.

Expansion card identity

An expansion card must identifying itself to the host operating system. This is done with the expansion card identity (ECId). It consists of at least one byte (the low byte) of which bits 3 to 7 carry ECId information, and is usually followed by several more bytes. The ECId is read by a synchronous read of address 0 of an expansion card slot.

Expansion card identity space

The expansion card identity space starts at expansion card address 0 and extends into the expansion card space as required. The minimum ECId, which all expansion cards must support, is a single readable byte at address 0, called the ECId low byte. Most expansion cards will support an extended ECId which consists of eight bytes starting from address 0. The ECId (whether extended or not) must appear at the bottom of the expansion card space, from address zero upwards after reset. It does not however have to remain readable at all times, so it can be in a paged address space so long as the expansion card is set to page zero on reset. Refer to the section entitled *Expansion card interrupts* on page 8 and the section entitled *FIQ and IRQ status* on page 9. This has the effect that the ECId, including the expansion card present bit, is only valid from reset until the expansion card driver is installed.

Code space

In addition to the expansion card identity, which all expansion cards must support, an expansion card can have code or data in ROM. RISC OS usually uses this for expansion card driver code, which is downloaded into system memory, by the operating system, before it is used. Often this code will be in a paged address space. The manner in which this code is accessed is variable and so it is accessed via a loader. The format of the loader is defined for each operating system, and gives access to a paged address space. The loader must live in the expansion card space above the ECId after reset, the position and size of it being defined by a chunk directory entry. Note that the ECId and the loader may themselves be in the paged address space as long as they appear at address zero after reset. Refer to the section entitled *Chunk directory structure* on page 10.

Expansion card identity low byte

The low byte of the ECId is as follows:

0,IRQ:	= 0 : not requesting IRQ } see text = 1 : requesting IRQ
1,P:	= 0 : expansion card is present
2,FIQ:	= 0 : not requesting FIQ } see text = 1 : requesting FIQ
3,ID0: 4,ID1: 5,ID2: 6,ID3:	= 0 : extended ECId <> 0 : id field
7,A:	= 0 : Acorn conformant expansion card = 1 : non-conformant expansion card

Expansion card presence

The host operating system has to know if there are any expansion cards present. Normally BD[1] is pulled high by a weak pullup. Reading the low byte of the ECId will therefore read a 1 on this bit if an expansion card is absent. All expansion cards must have bit 1 low in the low byte of the ECId.

Expansion card interrupts

An expansion card which is capable of generating a $\overline{\text{PIRQ}}$ or a $\overline{\text{PFIQ}}$ MUST carry a status bit for each of these interrupt sources. These two status bits must be in the low byte of the ECId unless the expansion card contains an extended ECId in which case the status bits may be relocated in the expansion card address space. An expansion card which is holding $\overline{\text{PIRQ}}$ low must set bit 0 high in the low byte of the ECId. An expansion card which is holding $\overline{\text{PFIQ}}$ low must set bit 2 high in the low byte of the ECId. In this way the operating system can quickly find which expansion card is generating the interrupt.

If the expansion card contains paged ROM, these status bits may be located elsewhere in the expansion card address space, in which case the two bits in the ECId low byte should be zeros. The location of these status bits must appear in the ROM space above the extended ECId.

If an expansion card is not capable of generating either a $\overline{\text{PIRQ}}$ or a $\overline{\text{PFIQ}}$, then bits 0 and/or 2 in the low byte ECId must be zero. If the interrupt status bits have been relocated, then the respective position mask should be set to zero. Refer to section entitled *Interrupt status pointers* on page 9.

ID field

There are four bits in the low byte of the ECId (BD[3:6]) which may be used for expansion card identification. These should only be used for the very simplest of expansion cards. Most expansion cards should implement the extended ECId which eliminates the possibility of expansion card ids clashing. When an extended ECId is used, all four bits in the id field of the low byte ECId must be zero.

Conformance bit

The most significant bit in the low byte of the ECId must be zero for expansion cards that conform to this Acorn specification.

Identification extension

If the id field of the low byte of the ECId is zero, then the ECId is extended. This means that the next seven bytes of the ECId will be read by the operating system. The

extended ECId starts at the bottom of page 0 of the paged identity space, and consists of eight bytes as defined below. If bit 0 of byte 1 is not set then the extended ECId is just eight bytes long. If bit 0 of byte 1 (CD) is set, then a chunk directory follows the interrupt status pointers.

7	6	5	4	3	2	1	0	
C[7]	C[6]	C[5]	C[4]	C[3]	C[2]	C[1]	C[0]	0x1C
M[15]	M[14]	M[13]	M[12]	M[11]	M[10]	M[9]	M[8]	0x18
M[7]	M[6]	M[5]	M[4]	M[3]	M[2]	M[1]	M[0]	0x14
P[15]	P[14]	P[13]	P[12]	P[11]	P[10]	P[9]	P[8]	0x10
P[7]	P[6]	P[5]	P[4]	P[3]	P[2]	P[1]	P[0]	0x0C
R	R	R	R	R	R	R	R	0x08
R	R	R	R	W[1]	W[0]	IS	CD	0x04
A	0	0	0	0	F	0	I	0x00

A	= 0 : Acorn conformant card = 1 : non-conformant card
F	= 0 : not requesting FIQ - see text = 1 : requesting FIQ
I	= 0 : not requesting IRQ - see text = 1 : requesting IRQ
R	= 0 : mandatory at present = 1 : reserved for future use
CD	= 0 : no chunk directory follows = 1 : chunk directory follows interrupt status pointers
IS	= 0 : interrupt status appears in low byte ECId = 1 : interrupt status has been relocated
W[1:0]	= 0 : 8 bit code follows after byte 15 of id = 1 : 16 bit code follows after byte 15 of id = 2 : 32 bit code follows after byte 15 of id = 3 : reserved
C[7:0] M[15:0] P[15:0]	= Reserved = Manufacturer = Product Type

FIQ and IRQ status

If bit 1 in byte 1 of the extended ECId (IS) is not set, then the interrupt status bits have not been relocated within the expansion card address space. In this case the FIQ and IRQ status bits must appear as bits 2 and 0 respectively in the low byte of the ECId. Expansion cards which cannot generate interrupts must drive these bits to zero. If bit 0 in byte 1 of the extended ECId (CD) is not set either, then the interrupt status pointers do not need to be defined, as the operating system will not read them. If CD is set, then the interrupt status pointers should be defined to point to the respective bits in the ECId low byte.

If bit 1 of byte 1 of the extended ECId (IS) is set, then the interrupt status bits have been relocated within the expansion card space. In this case the interrupt status pointers must be defined as described in the section Interrupt status pointers below. Note that if both IRQ and FIQ sources are provided by an expansion card, then a separate status bit must exist for each type of interrupt source, though the two status bits may appear at the same address if convenient. Refer to *Interrupt status pointers* below.

Country code

Previously C[7:0] was the country code. This is no longer used; the UK code 00 should now be used instead.

Manufacturer's code

Every expansion card should have a manufacturers code. The following are some examples of these:

Manufacturer	Code Value
Acorn UK	0
Olivetti	2
Watford	3
Computer Concepts	4
Wild Vision	9

Consult Acorn for the allocation of codes.

Product type code

Every expansion card type must have a unique number allocated to it. These are a few examples:

Product Type	Code Value
SCSI	2
Ethernet	3
RAM/ROM	5
BBC IO	6
MIDI	19

Consult Acorn for the allocation of codes.

Interrupt status pointers

If bit 1 of byte 1 of the extended ECId (IS) is set, then the address of the FIQ and IRQ status bits must be provided in the eight bytes which follow the extended ECId, even if the two status bits are at address 0. There are two sets of four byte numbers as detailed below, each consisting of a three byte address field and a one byte position mask field. The position mask defines which bit within the status byte refers to the status bit. It should consist of single 'one' and seven zeros. The other bits within the status byte may be 'don't cares'. If the expansion card does not provide one of these interrupt sources, then the respective position mask should consist of eight zeros.

Chunk directory structure

If bit 0 of byte 1 of the extended ECId (CD) is set, then bit 1 of byte 1 of the extended ECId (IS) must be set and hence the addresses must be present.

Note that these eight bytes are always assumed to be bytewide. Only the lowest byte in each word should be used. After byte 15 (hex address 40 upwards), wider words may be used, according to the setting of W[1] and W[0] in the extended id. See the section entitled *Identification extension* on page 8.

The 24-bit address field allows for an absolute byte address with an offset from hex 03240000 to be defined. Hence the cycle speed to access the status register can be included in the address (encoded by bits 19 and 20). Bits 14 and 15 should be zero.

	Hex address
IRQ Status Bit	xxxxxx40
Address (24 bits)	xxxxxx34
IRQ Status Bit	
Position mask	xxxxxx30
FIQ Status Bit	
Address (24 bits)	xxxxxx24
FIQ Status Bit	
Position mask	xxxxxx20

If bit 0 of byte 1 of the extended ECId (CD) is set, then following the interrupt status pointers is a directory of chunks of data and/or code stored in the ROM. The lengths and types of these chunks and the manner in which they are loaded is variable, so after the eight bytes of interrupt status pointers there follow a number of entries in the chunk directory.

Note that from here on the definition is in terms of bytes. If the expansion card supports a 16 bit wide (or a future 32 bit) interface then the driver code must take this into account.

The chunk directory entries are eight bytes long and all follow the same format. There may be any number of these entries. This list of entries is terminated by a block of four bytes of zeros.

Start Address	n + 8
4 Bytes (32 bits)	n + 4
Size in bytes	
3 Bytes (24 bits)	n + 1
Operating System	
Identity Byte	n

One of these blocks must be the code loader. It contains the code to load bytes from the (optionally paged) ROM into main memory, and as such is capable of updating the page register as required. There may be more than one loader present, to cater for different operating systems. All the loader code must be accessible after reset. After the loader is transferred to main memory, all further chunks are transferred via the loader. The chunks are again referenced by the chunk directory as above, starting at virtual address zero. Note that after the loader has been loaded, the main expansion card id area may be mapped out. An example of a typical use of the chunk directory is shown in Figure 7: *Chunk directory structure* on page 11. The shaded areas refer to chunks which are transferred via the loader.

Operating system identity byte

The operating system identity byte forms the first byte of the chunk directory entry, and determines the type of data which appears in the chunk to which the chunk directory refers.

OS[3]	= 0	reserved
OS[3]	= 1	mandatory at present
OS[2:0]	= 0	Acorn Operating System #0 (RISC OS) D[3:0] = 0 loader = 1-15 OS dependent
	= 1	Acorn Operating System #1 D[3:0] = 0 loader = 1-15 reserved
	= 2	Acorn Operating System #2 D[3:0] = 0 loader = 1-15 reserved
	= 3-5	reserved D[3:0] = 0-15 reserved
	= 6	manufacturer defined D[3:0] = 0-15 manufacturer specific
	= 7	device data D[3:0] = 0 link (for 0, the object pointed to is another directory) = 1 serial number = 2 date of manufacture = 3 modification status = 4 place of manufacture = 5 description = 6 part number (for 1-6, the data in the pointed-to location contains the ASCII string of the information.) = 7-15 reserved

Examples of use

The previous paragraphs explained the system of expansion card identification. You do not need to use all of these features on all expansion cards, and the implementation depends on the needs and complexity of the expansion card in question. All expansion cards must implement at least the simplest form of expansion card identification. Synchronous cycles are used by the operating system to read and write any locations within the ECId space (to simplify the design of synchronous expansion cards).

Non-extended expansion card identity

This is the simplest possible expansion card identity mechanism. It may be used for temporary expansion cards or where expansion cards are used in a localised, closed environment. It should not be used for expansion cards for general sale. Non-extended ECIds do not need PR/$\overline{W}$ factored into their enable, as the operating system will only read the ECId space.

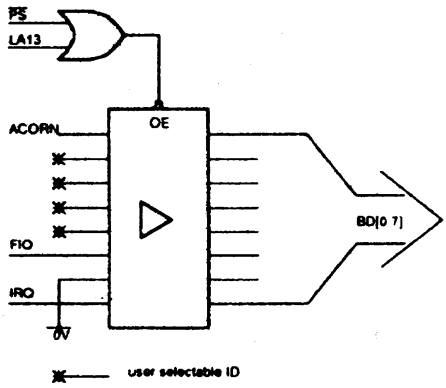


Figure 5: Non-extended ECId

Extended expansion card identity

The next simplest case which most expansion cards should implement as a minimum is the case of an extended ECId but no code in ROM. This can be achieved by a 32 x 8 bit PROM. An example is in Figure 6: *Extended ECId* on page 11.

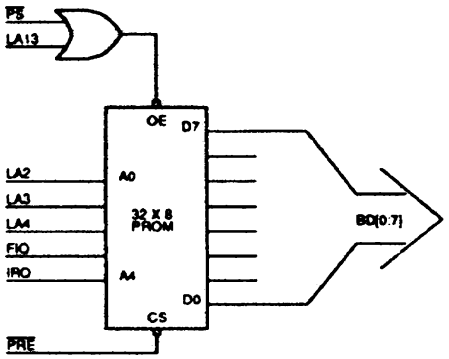


Figure 6: Extended ECId

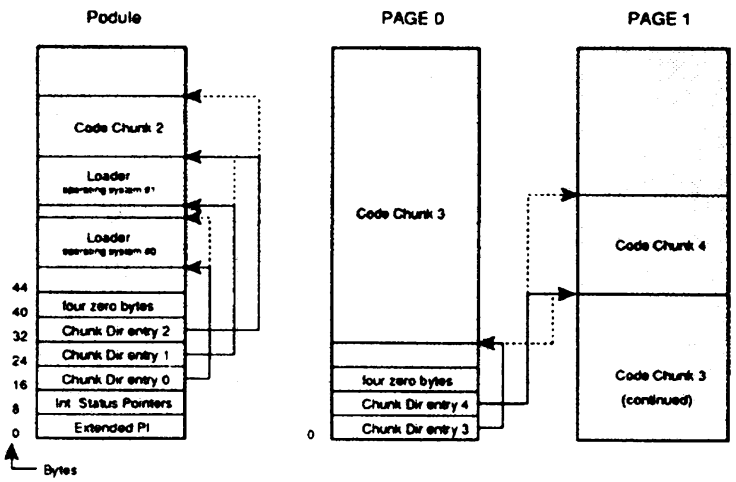


Figure 7: Chunk directory structure

Extended expansion card identity with paged ROM

When the expansion card includes driver code in ROM, there are several possibilities for implementing the ECId. One example showing an EPROM with a paging register is shown in Figure 8: *Extended ECId with paged ROM* on page 12. Simplifications can be made where there is only one page, or where a larger EPROM allows the inclusion of the low byte of the ECId. (FIQ and/or IRQ can be factored into the address space as in the previous example).

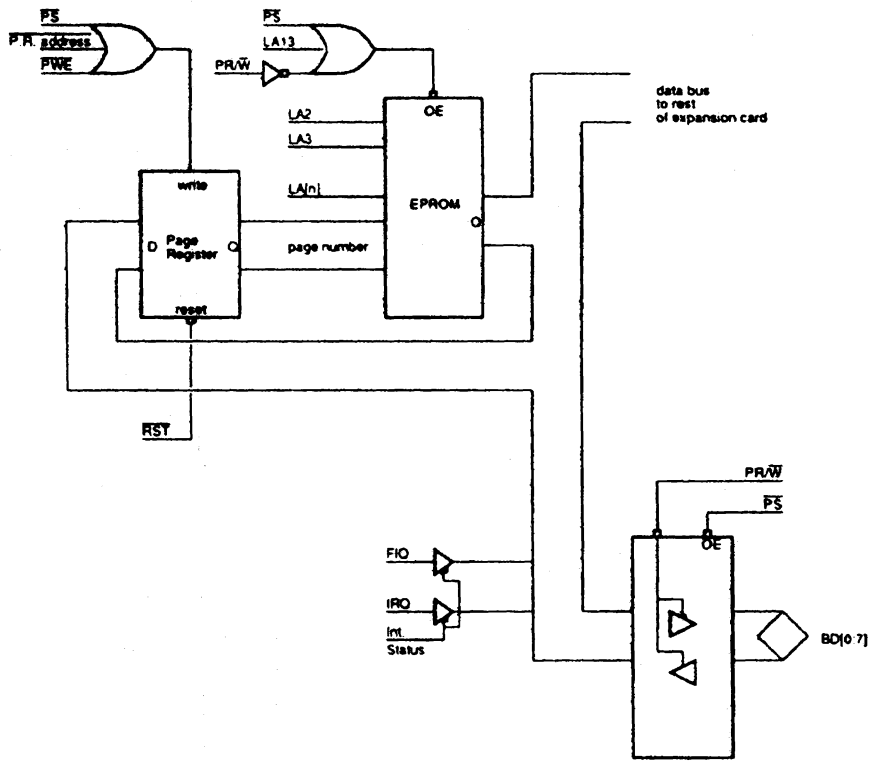


Figure 8: Extended ECId with paged ROM

Simple expansion cards

Simple expansion cards are controlled by the IOC. These expansion cards may be accessed by one of four types of cycle, designated slow, medium, fast, and synchronous. The cycles are mapped at different addresses.

Once the cycle has started, MEMC may de-assert $\overline{\text{IORQ}}$ (and hence IOC will de-assert $\overline{\text{RBE}}$) in order to carry out memory refresh or DMA operations. This is indicated by the shaded area in Figure 9: *IOC driving an expansion card read cycle* on page 13. If, when the IOC has finished the cycle, the MEMC has not reasserted $\overline{\text{IORQ}}$ then the I/O cycle will be stretched until the MEMC is ready to complete the cycle. This does not however alter the cycle the expansion card sees, because the cycle is in effect finished before the stretching takes place. In the case of a write the $\overline{\text{WBE}}$ and $\overline{\text{PS}}$ have already been de-asserted, in the case of a read the data from the expansion card has already been latched into the data buffers by $\overline{\text{BL}}$.

Figure 9: *IOC driving an expansion card read cycle* on page 13 shows how the IOC generates a fast expansion card read: the IOC generates the expansion card read, write and select strobes, and also controls the $\overline{\text{IOGT}}$ and $\overline{\text{BL}}$ signals. The expansion card read, write and select signals are timed with respect to the signal CLK8 and NO relationship between REF8M and CLK8 can be assumed.

Figure 10: *Stretched expansion card read cycle* on page 14 shows the same cycle, but this time the MEMC has de-asserted $\overline{\text{IORQ}}$ at the time when the IOC is about to finish the cycle. Accordingly, the cycle is stretched by one REF8M (the shaded region), but note that the access to the expansion card has not been stretched.

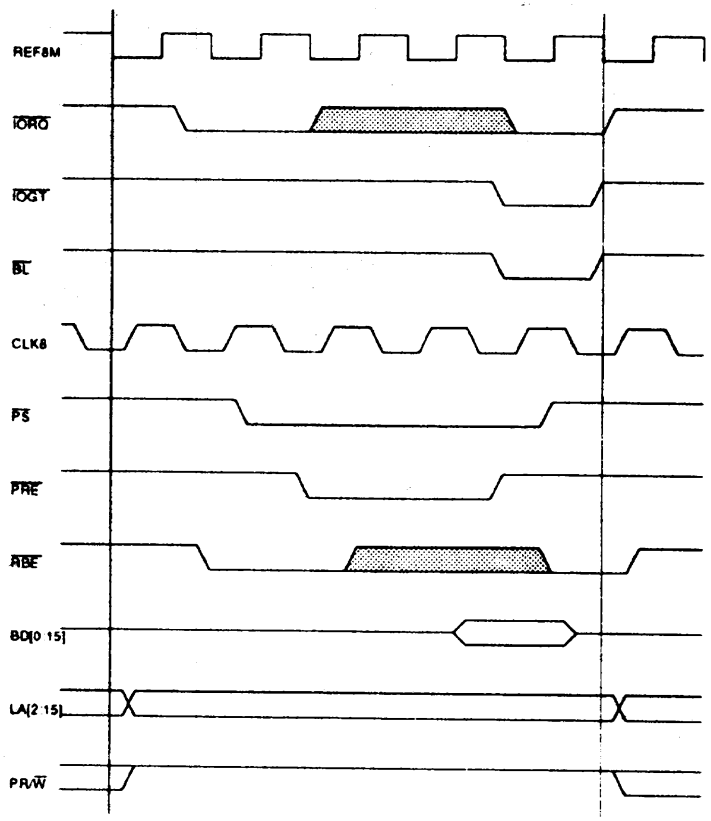


Figure 9: IOC driving an expansion card read cycle

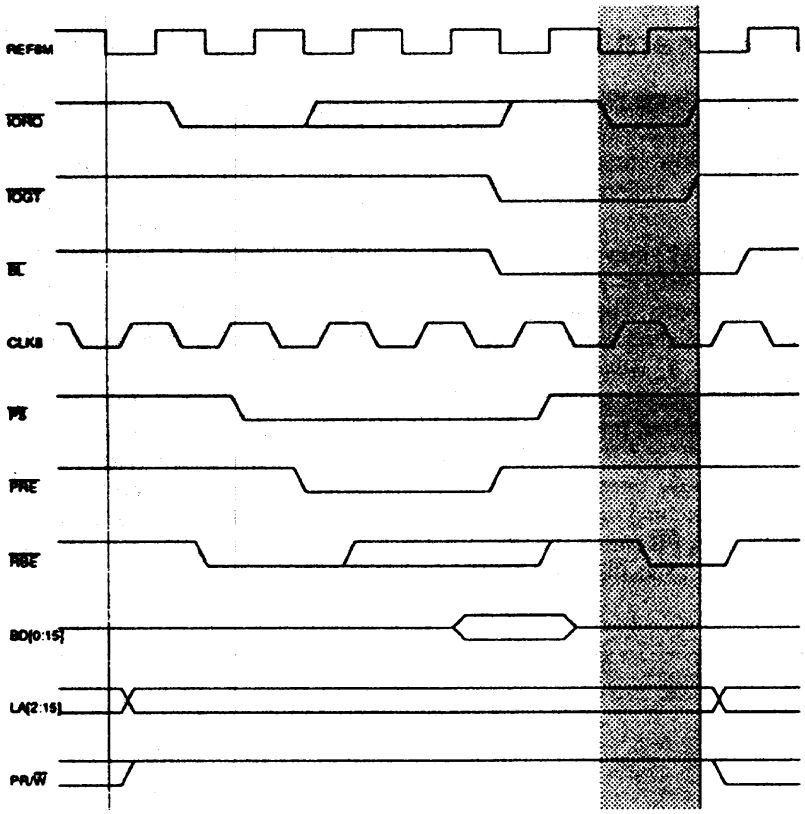


Figure 10: Stretched expansion card read cycle

Expansion card accesses

The following diagrams detail the four possible types of IOC expansion card access. In each diagram REF8M is shown, but this is only for reference. The phase relationship of REF8M and CLK8 is NOT guaranteed.

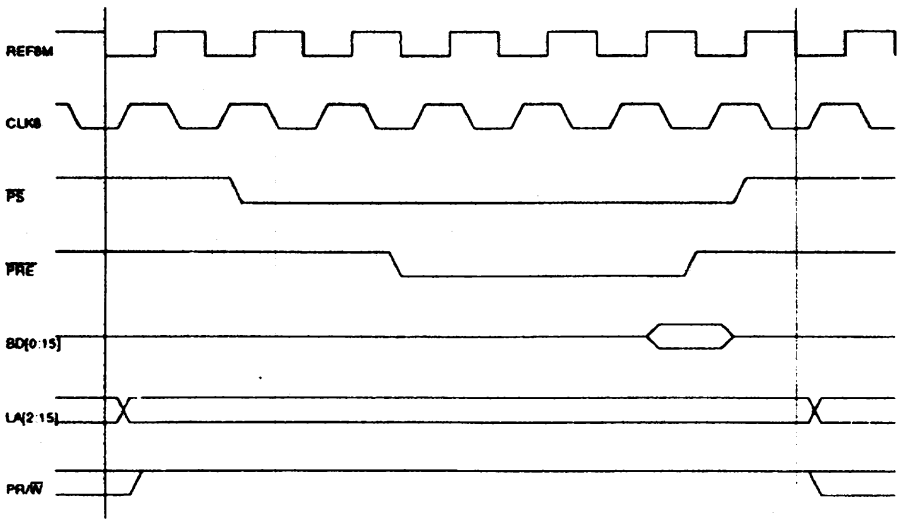


Figure 11: Slow cycle read

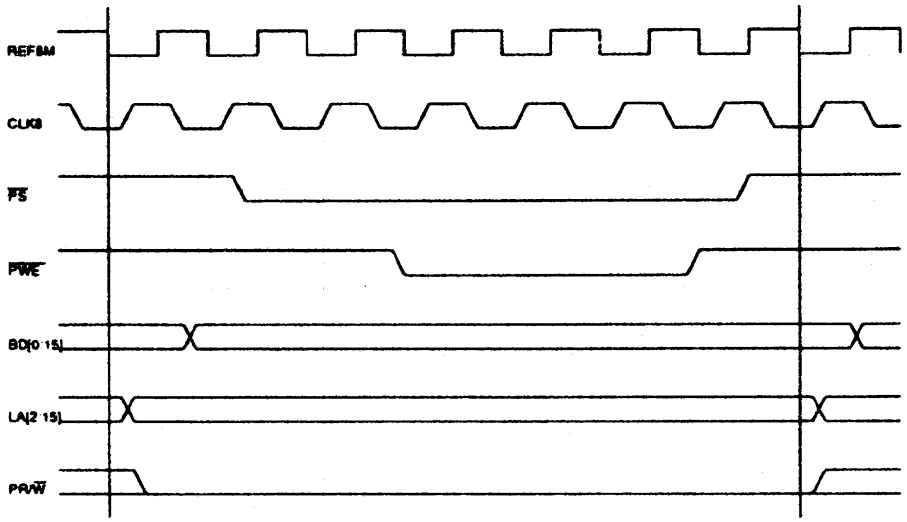


Figure 12: Slow cycle write

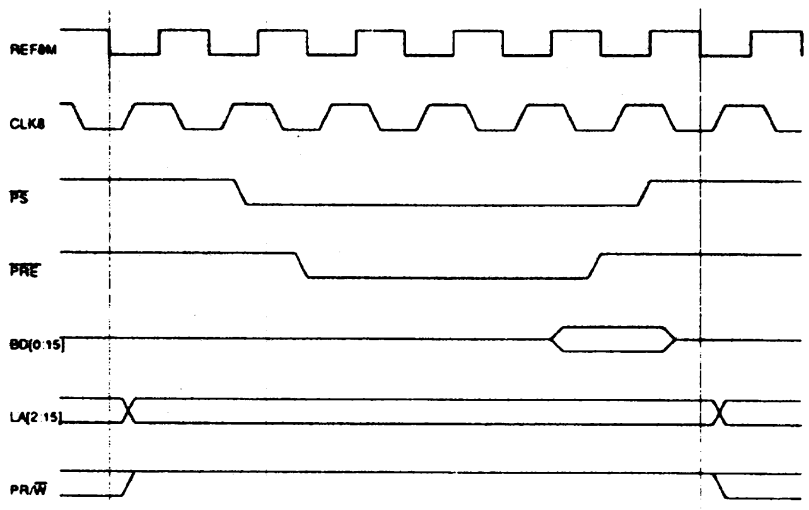


Figure 13: Medium cycle read

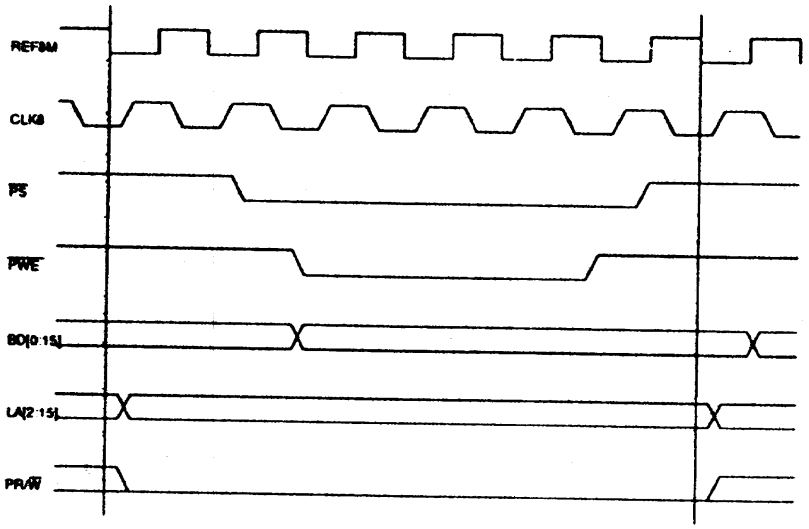


Figure 14: Medium cycle write

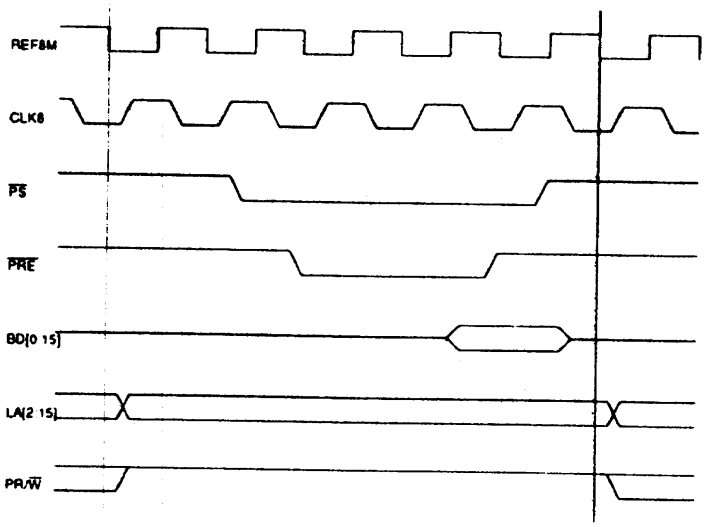


Figure 15: Fast cycle read

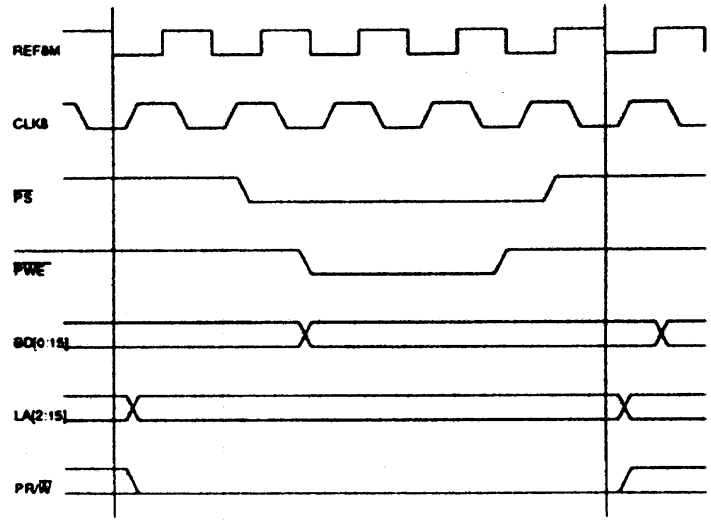


Figure 16: Fast cycle write

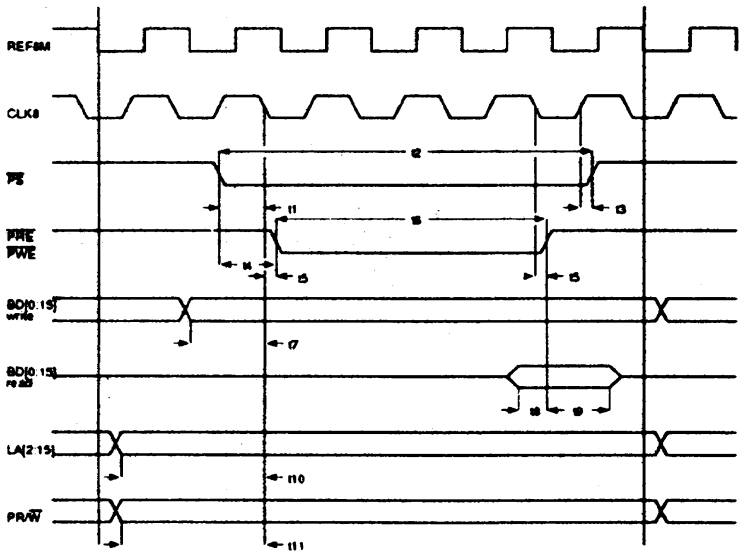


Figure 17: General timing for slow, medium and fast cycle types

Sym	Parameter	Min	(ns) Nom	Max
t1	$\overline{PS}$ setup to CLK8	40		120
t2	$\overline{PS}$ width TYPE slow		625	
t2	$\overline{PS}$ width TYPE med		500	
t2	$\overline{PS}$ width TYPE fast		375	
t3	$\overline{PS}$ hold from CLK8	0		50
t4	$\overline{PS}$ to $\overline{PRE}$ or $\overline{PWE}$ TYPE slow		187	
t4	$\overline{PS}$ to $\overline{PRE}$ or $\overline{PWE}$ TYPE med/fast		62	
t5	$\overline{PRE}$ or $\overline{PWE}$ delay from CLK8	0		15
t6	$\overline{PRE}$ or $\overline{PWE}$ width TYPE slow/med		375	
t6	$\overline{PRE}$ or $\overline{PWE}$ width TYPE fast		250	
t7	write data setup to CLK8	100		
t8	read data setup to $\overline{PRE}$	20		
t9	read data hold from $\overline{PRE}$	15		
t10	address setup to CLK8	150		
t11	PR/$\overline{W}$ setup to CLK8	140		

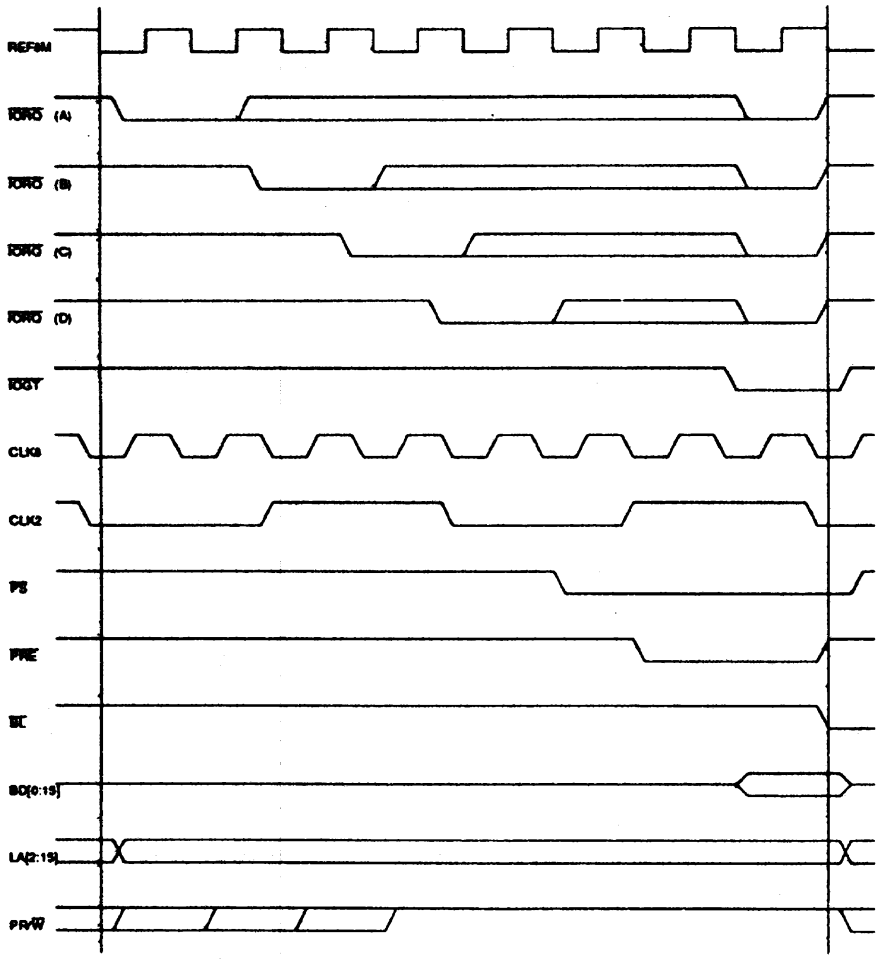


Figure 18: Synchronous cycle read

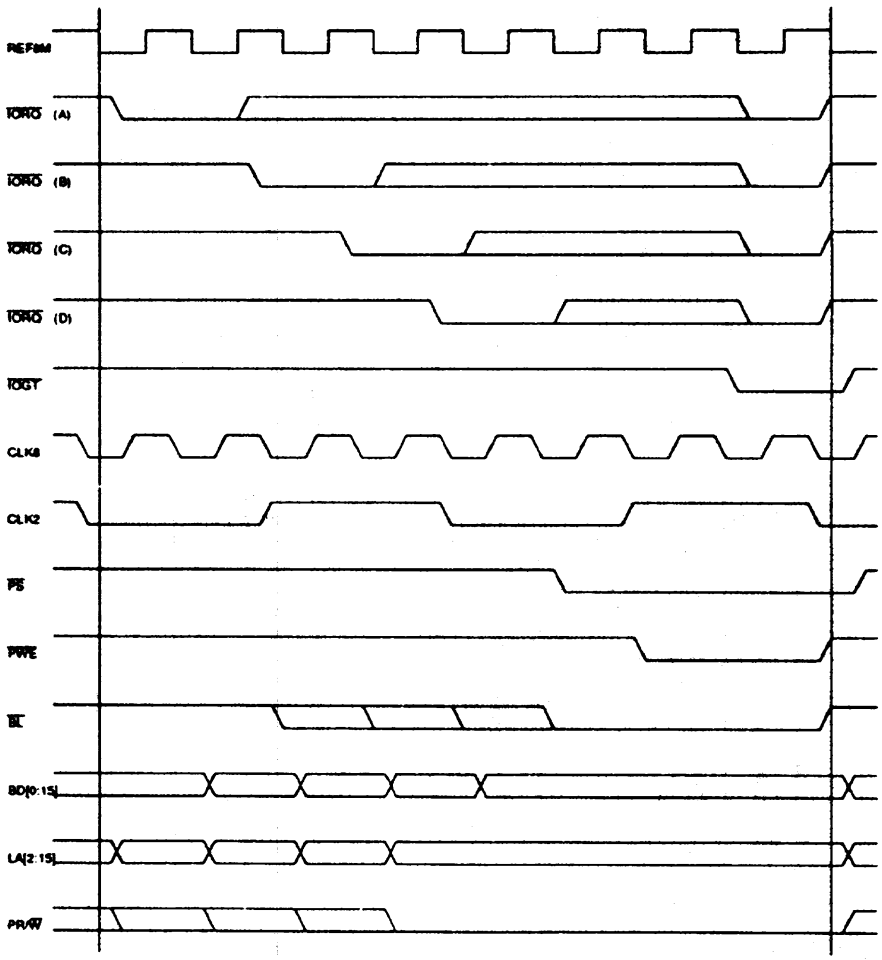


Figure 19: Synchronous cycle write

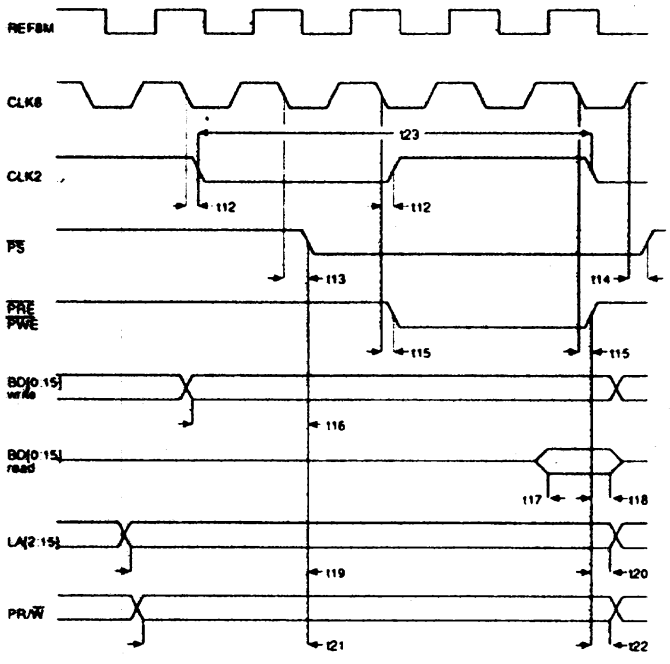


Figure 20: Timings for synchronous cycle types

Sym	Parameter	Min	Max
t12	CLK2 delay from CLK8	0	15
t13	PS delay from CLK8	5	50
t14	PS hold from CLK8	10	
t15	PRE or PWE delay from CLK8	0	15
t16	write data setup to PS	100	
t16a	write data hold from PWE or CLK2	20	
t17	read data setup to PRE or CLK2	50	
t18	read data hold from PRE or CLK2	15	
t19	address setup to PS	150	
t20	add. hold from PRE or PWE or CLK210		
t21	PR/W setup to PS 140		
t22	PR/W hold from PRE or PWE or CLK2	10	
t23	cycle time square wave	500	

MEMC expansion cards

MEMC expansion cards are not controlled by IOC, although they share the same interface, so they have to time their own cycles, with their own I/O control logic. The interface has two control lines, IORQ (driven by MEMC) and IOGT (driven by the expansion card or IOC). IOGT is an open drain signal allowing multiple devices to drive this signal. MEMC expansion cards are decoded by LA[21] low, and IOC is decoded by LA[21] high. But even when IOC is not selected, it continues to control the external buffer enables, RBE and WBE. The latching of the buffer must however be controlled by the expansion card which is controlling the cycle. This is done by pulling BL low. BL is an open drain signal.

I/O controller interface

I/O controllers use a handshaking system to synchronise I/O peripherals with the system data bus. The interface is timed with respect to the REF8M clock, and cycles may be produced in multiples of 8 MHz clock ticks. When the processor accesses the I/O controller address space (while MEMC is in supervisor mode), MEMC starts an I/O cycle by driving IORQ low and holding the processor clocks (stretching the processor cycle when PH2 is high). The I/O controller signals when it is ready tc end the I/O cycle by driving IOGT, low. The I/O cycle ends when both IORQ and IOGT are seen low on the rising edge of REF8M. Then MEMC drives IORQ high and releases the processor clocks. The I/O controller de-asserts IOGT which goes high on the next falling edge of REF8M.

A MEMC I/O cycle is shown in Figure 21: *I/O cycle* on page 21. The cycle starts with $\overline{\text{IORQ}}$ being taken low. There follows a number of 8 MHz clock ticks until the I/O controller is in a position to complete the cycle. The $\overline{\text{IOGT}}$ line is taken low, and both MEMC and the I/O controller see $\overline{\text{IORQ}}$ and $\overline{\text{IOGT}}$ low on the rising edge of REF8M, so the I/O cycle terminates on the next falling edge of REF8M.

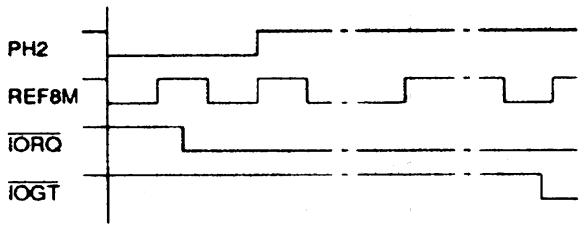


Figure 21: I/O cycle

Some I/O cycles may only take 250 ns as shown in Figure 22: *Fast I/O cycle* on page 21. To give the I/O controller adequate time to recognise such operations, MEMC produces the first $\overline{\text{IORQ}}$ early in the I/O cycle.

The extension of $\overline{\text{IORQ}}$ only happens at the start of an I/O cycle; if the $\overline{\text{IORQ}}$ signal is removed during a DMA or refresh operation, it will be reasserted when REF8M goes low.

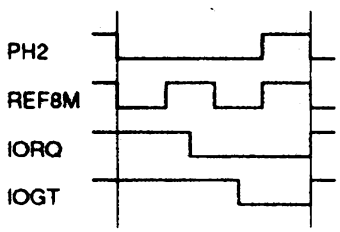


Figure 22: Fast I/O cycle

I/O cycles may be interrupted by DMA and refresh operations, as shown in Figure 23: *I/O cycle interrupted by a DMA or refresh operation* on page 21. If a DMA or refresh operation is pending, the $\overline{\text{IORQ}}$ signal is driven high when REF8M next goes low. The DMA/refresh operation may then begin. When the operation completes, the I/O cycle is resumed by setting $\overline{\text{IORQ}}$ low (provided no more DMA or refresh operations are pending). The DBE line is always driven low by MEMC during DMA/refresh operations to disable the processor data bus drivers. Hence the I/O cycle is stretched, and the write data would become invalid during the cycle. The data must therefore be latched into the data bus buffers by the I/O controller during the first $\overline{\text{IORQ}}$ low period, and be held until the I/O cycle has completed. This is done by the I/O controller driving $\overline{\text{BL}}$ low for this period. The maximum time for which an I/O cycle may be interrupted in this way is 1875 ns (i.e. fifteen REF8M cycles).

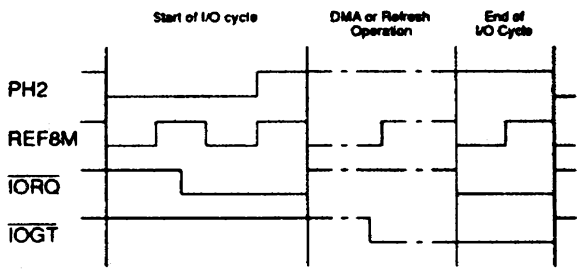


Figure 23: I/O cycle interrupted by a DMA or refresh operation

NOTE: Care must be taken not to address a non-existent I/O controller, as MEMC will hold the processor clocks indefinitely until a low is seen on the $\overline{\text{IOGT}}$ line, or RESET is set high.

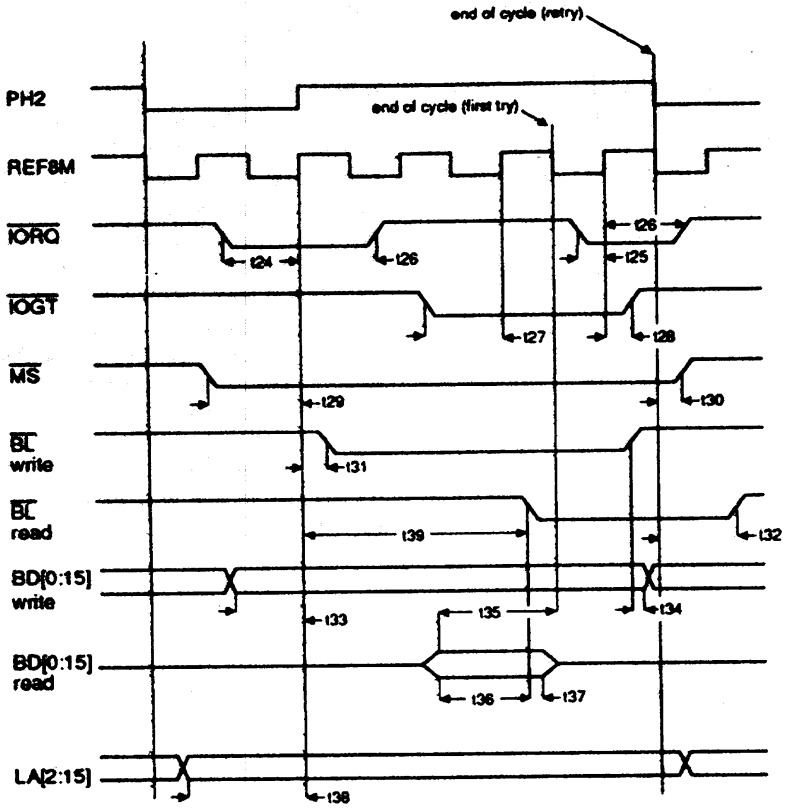


Figure 24: MEMC expansion card timing

Sym	Parameter	Min	Max
t24	$\overline{IORQ}$ setup (first attempt)	70	115
t25	$\overline{IORQ}$ setup (retries)	50	75
t26	$\overline{IORQ}$ hold	50	
t27	$\overline{IOGT}$ setup	25	120
t28	$\overline{IOGT}$ hold	20	100
t29	$\overline{MS}$ setup to REF8M[3]	110	
t30	$\overline{MS}$ hold	5	
t31	$\overline{BL}$ delay write	0	65
t32	$\overline{BL}$ hold read	10	100
t33	BD[0:15] setup write	85[1]	
t34	BD[0:15] hold from $\overline{BL}$ write	5	
t35	BD[0:15] setup to REF8M read	50[2]	
t36	BD[0:15] setup to $\overline{BL}$ read	20	
t37	BD[0:15] hold from $\overline{BL}$ read	15	
t38	LA[2:15], PR/$\overline{W}$ setup to REF8M	140	
t39	$\overline{BL}$ Read delay	60	

[1] with $\overline{BL}$ high i.e. buffers transparent.

[2] setup to the earliest possible end of cycle.

[3] the first rising edge of REF8M after $\overline{IORQ}$ goes low.

MEMC expansion card timing

A typical cycle with timing parameters is shown in Figure 24: *MEMC expansion card timing* on page 22. The sequence consists of a MEMC expansion card access, a DRAM refresh by MEMC, with the end of the expansion card access delayed by one REF8M cycle.

Expansion card bus connector

The expansion card bus is an extension of the I/O data and address buses, and includes necessary control signals.

The expansion card interface consists of a 64-way DIN 41612ac connector. Pin allocations are given below

Pin	a	c	Description
1	0V	0V	ground
2	LA[15]	-5V	supply[1]
3	LA[14]	0V	ground
4	LA[13]	0V	ground
5	LA[12]	reserved	
6	LA[11]	$\overline{MS}$[2]	MEMC exp card select
7	LA[10]	reserved	
8	LA[9]	reserved	
9	LA[8]	reserved	
10	LA[7]	reserved	
11	LA[6]	reserved	
12	LA[5]	$\overline{RST}$	reset (see note)
13	LA[4]	PR/$\overline{W}$	read / not write
14	LA[3]	$\overline{PWE}$	write strobe
15	LA[2]	$\overline{PRE}$	read strobe
16	BD[15]	$\overline{PIRQ}$	normal interrupt
17	BD[14]	$\overline{PFIQ}$	fast interrupt
18	BD[13]	reserved	
19	BD[12]	C1	I²C serial bus clock
20	BD[11]	C0	I²C serial bus data
21	BD[10]	$\overline{EXTPS}$	external card select
22	BD[9]	$\overline{PS}$[3]	simple card select
23	BD[8]	$\overline{IOGT}$	MEMC card handshake
24	BD[7]	$\overline{IORQ}$	MEMC card request
25	BD[6]	$\overline{BL}$	I/O data latch control
26	BD[5]	0V	supply
27	BD[4]	CLK2	2 MHz synchronous clock
28	BD[3]	CLK8	8 MHz synchronous clock
29	BD[2]	REF8M	8 MHz reference clock
30	BD[1]	+5V	supply
31	BD[0]	reserved	

1. The A3000 is 5 volt only. Pins 2c and 32c are not connected.
2. On the A3000 External expansion card bus connector, this signal is $\overline{MS[0]}$.
3. On the A3000 External expansion card bus connector this signal is $\overline{PS[0]}$.

The A3000 Internal expansion card uses a set of signals which are a subset of the full expansion card bus. The connection is via two 17-way 0.1 inch pitch connectors and two 5-way 0.1 inch pitch connectors (the latter fitted as standard to Issue 1 PCBs and later). Expansion cards should use 0.025 inch square pin headers.

Pin	SK3	SK11	SK8	SK9
1	+5V	0V	C[0]	0v
2	$\overline{PWE}$	+5V	C[1]	REF8M
3	$\overline{PS1}$	$\overline{PRE}$	$\overline{BI}$	$\overline{PFIQ}$
4	CLK2	PR/nW	$\overline{IORQ}$	$\overline{Ms[1]}$
5	LA[2]	LA[4]	$\overline{IOGT}$	+5V
6	LA[3]	LA[5]		
7	BD[0]	LA[6]		
8	BD[1]	LA[7]		
9	BD[2]	0V		
10	BD[3]	LA[8]		
11	BD[4]	LA[9]		
12	BD[5]	LA[10]		
13	BD[6]	LA[11]		
14	BD[7]	LA[12]		
15	$\overline{RST}$	LA[13]		
16	0V	$\overline{PIRQ}$		
17	+5V	0V		

Note: Pin 1 is at the righthand end when viewed from the front of the computer.

I²C bus

- this bus is primarily intended for Acorn's use and it should only be used within the computer.
- the maximum recommended load per podule is 20 pf.
- the speed and timings may vary between computer models and operating systems.
- this bus is not a full implementation of the I²C specification (e.g. it does not support other bus masters).
- this bus may not be incorporated in future machines.

The reserved pins must be left unused, as in some machines these pins carry co-processor signals.

Reset

The $\overline{RST}$ signal is the system reset signal. It is driven low at power-on by IOC, or by a user reset, from the keyboard. It is an open-collector signal, and expansion cards may drive it, to generate a system reset. The pulse width should be at least 50ms.

Power consumption

It is strongly recommended that expansion cards work from the 5 volt rail only. Acorn expects future models to continue the trend towards single supply computers, started with A3000.

The maximum current drain allowed from the computer 5 volt supply is: 1.0 amp per expansion card slot, for Archimedes and A3000 computers

Each expansion card may draw a maximum of 50 mA

from the -5 volt rail and 250 mA from the 12 volt rail. These voltages are not available on the A3000 and may not be available on new computers.

Double width cards may use twice the above current values when used with Archimedes.

The A3000 internal expansion card may only draw 600mA from the +5v rail. The maximum dissipation inside the case is 0.5W (100mA).

Backplane circuit description

Expansion card and module selection

S6, LA14, 15 and 21 are decoded by IC1 (74HC139) to provide:

- four expansion card selects $\overline{PS}$[0:3]
- four module selects $\overline{MS}$[0:3]

LA15	LA14	Card selects $\overline{S4}$=0	Module selects LA21=0
0	0	$\overline{PS0}$	$\overline{MS0}$
0	1	$\overline{PS1}$	$\overline{MS1}$
1	0	$\overline{PS2}$	$\overline{MS2}$
1	1	$\overline{PS3}$	$\overline{MS3}$

Interrupt mask (some models only)

The two PALs on the backplane, where fitted, extend IOC's interrupt mask/register structure to include expansion card interrupts (PIRQs). This allows the PIRQ from each expansion card to be individually masked or enabled.

The two PALs are identical: IC2 controls PIRQs from expansion card slots 0 and 1 IC3 controls PIRQs from expansion card slots 2 and 3.

PAL operation

During reset, $\overline{RST}$ input (pin 4) is taken low, setting the mask register to 1's, so enabling interrupts.

To access the registers, $\overline{S6}$ (pin 1) is taken low, LA2 (pin 5) high selects the mask register, low selects the interrupt request register.

Safety

The current industry wide IT safety standard is IEC950 whose European 'harmonized' version is EN60950 / BS7002 but your particular application may also be within the scope of other additional standards.

The main requirements are that the equipment provide protection against:

- the spread of fire
- hazardous voltages or energy.

The spread of fire

The Underwriters Laboratory (UL) of the USA have devised several standard ways of testing plastic materials for their flammability properties. The UL94 test procedure is used within IEC950 to specify the required flame retardant level for materials and components. The ratings start at 94v-0 down to 94v-1, 94v-2 and finally 94HB.

Confirmation of a UL test pass will be the issue of a 'yellow card', a copy of which can be obtained from your supplier, for either the plastic material itself or indeed the component.

IEC950 specifies that PCBs will have a minimum flame-retardancy rating of UL94v-1 and that any components mounted on the PCB meets the lower standard of UL94v-2.

The choice and layout of components should also seek to prevent the spread of fire across the PCB and within the computer.

The computer external enclosure forms a fire barrier and as such its material must meet the higher standard of UL94v-1. As the expansion card rear panel will be part of the fire enclosure the panel itself must meet this standard; metal is acceptable.

If you fit a large plastic connector into this rear panel the connector will also have to be UL94v-1.

Hazardous voltages or energy

The computer PSU is designed to provide only SELV (safety extra low voltage) to the computer. This means that the PCB supply voltages have two independent means of protection against hazardous voltages thus ensuring that even under single fault conditions the voltage on the computer PCB and interfaces will be safe. Within IEC950 hazardous voltages are those greater than 42.4 volts peak or 60 volts DC.

Physical access by a user to any hazardous voltages must be prevented by a physical barrier in which no aperture is greater than 5 mm in any dimension.

Expansion cards MUST be designed in such a way that they do not introduce either a reduction in fire protection or of voltage isolation into the host computer. Expansion card designers should obtain a copy of the standard and if necessary seek further clarification by consulting a reputable test facility such as BSI.

With the EEC Directive on General Product Liability manufacturers and importers within the Community are subject to strict liability. This existing law removes the plaintiff's requirement to prove that a product was 'defective' when seeking damages. The possession of an EN certification may not be an adequate defence and therefore manufacturers need to be aware that there may be safety aspects of their product which are not covered by the standards.

Testing

U.K. legislation now requires that portable electrical equipment be tested regularly, usually annually, for safety. Be aware that this testing may be carried out by a variety of people of varying technical competence and experience.

Class 1, earthed, equipment will be tested by the application of a high current, low voltage, source of 4 - 25 Amps, between any exposed metalwork and the earth pin in the mains plug. The presence of your expansion card in a computer will make it liable for testing.

You should therefore consider whether to provide information in your documentation to either the user or the dealer on how your card should be tested.

EMC Design

When designing for compliance the following points should be borne in mind.

- all external connectors should be of a robust, recognised EMC design.
- all external connectors should have a comprehensive low impedance bond to the rear panel.
- the rear panel must be conductive and fitted to the host computer frame via low impedance fixings.
- when fitted, there should not be any continuous slot longer than 20mm around the expansion card rear panel.
- any external cables must have an EMC performance that does not compromise the host computer system.
- if external cables are required for the card, but not supplied, full details of the recommended cable, connectors and construction methods of the leads should be specified.
- a four layer PCB is preferred together with a good layout.
- connection of the card 0 volts line and the rear panel is not generally recommended, however, on some PCB designs this may be found to give an improvement in static immunity.
- be aware that the rear panel may be subject to high current earth continuity testing; see the section entitled *Testing* on page 25.
- the EMC performance of the card must not deteriorate as a result of insertion and removal during the life of the product.

Mechanical Specification

Expansion cards should be built to the standard DIN Eurocard specification, noting that the thickness of the back panel is 1.6mm. Drawings are attached (0276,099 and 0276,204), giving the relevant dimensions of single- and double-width expansion cards, and their mounting brackets respectively.

A3000 internal expansion cards are described in the attached drawing (Acorn part number 0280,080).

Blanking panels

Single-width expansion cards should be accompanied by a single-width blanking panel and T-piece, to blank off the aperture otherwise left when the full-width Acorn blanking panel is removed to fit the card. Drawings of the T-piece (0276,036) and the blanking panel (0276,035) are attached. Acorn Customer Services may also be able to advise potential sources of these components.

Paint specification

The cream paint colour for back panels is RAL 1013C.

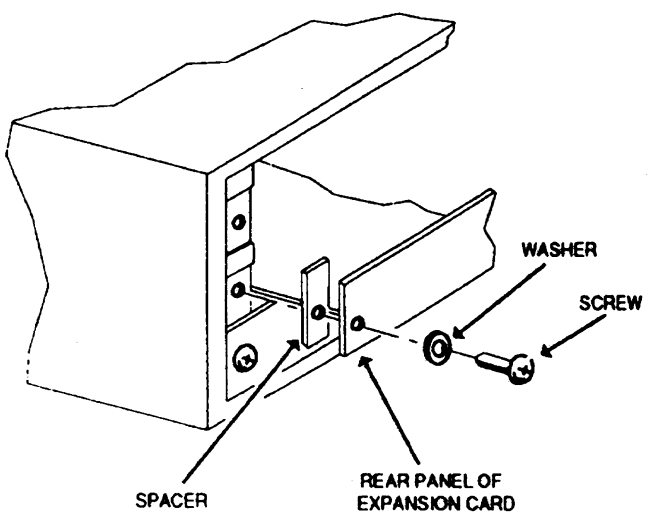
Archimedes 300 series and 440 computers and R140 workstations

Note that the expansion card slot in Archimedes 300-series, 440 and some R140 machines is 2.5mm shorter than it should be (from backplane to rear panel) for a standard Eurocard. This was corrected for Archimedes 400/1 series, R260 and later computers. Expansion cards which may be used by these earlier computers should therefore be supplied with two spacers (see the attached drawing, number 0276,227, for details of the spacer). This will allow the expansion card to be seated securely, without causing the backplane to lean excessively towards the front of the machine.

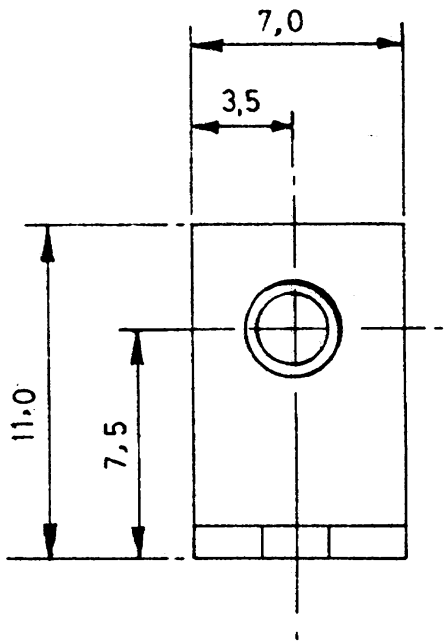
The original screws supplied with the computer will be long enough to accommodate the spacers, provided that the expansion card backplate thickness does not exceed 1.6mm. If your card has a thicker rear panel, you may need to supply longer fixing screws.

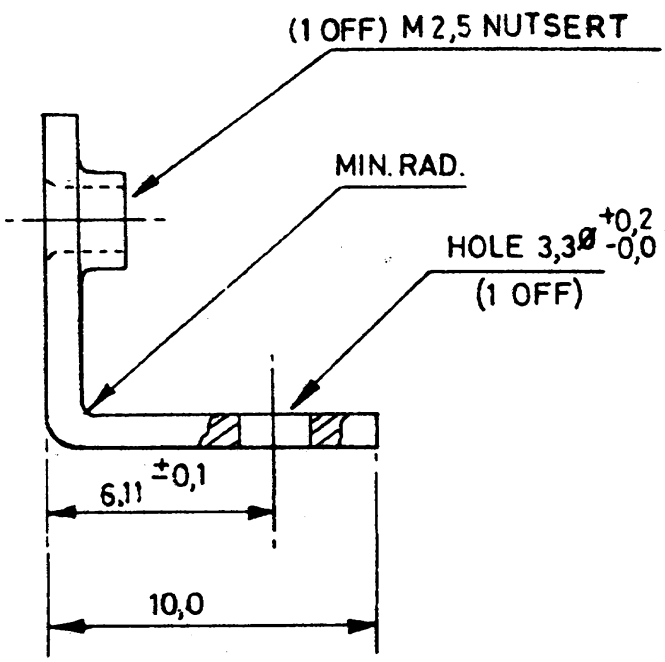
Fitting

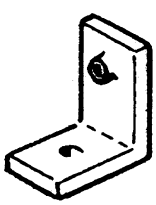
The diagram below shows how the spacers are fitted to earlier computers:



DRG. No.	USED ON	SHT. 1 OF 1
0276,204/	A1	







PROJECTION OF FINISHED ITEM

MATERIAL—1,2 THK. M.S.
ZINC PLT. & CLEAR PASS. OR ZINTEC.
FINISH – CLEAN FREE FROM BURRS.
GEN. TOL.- ±0,1mm

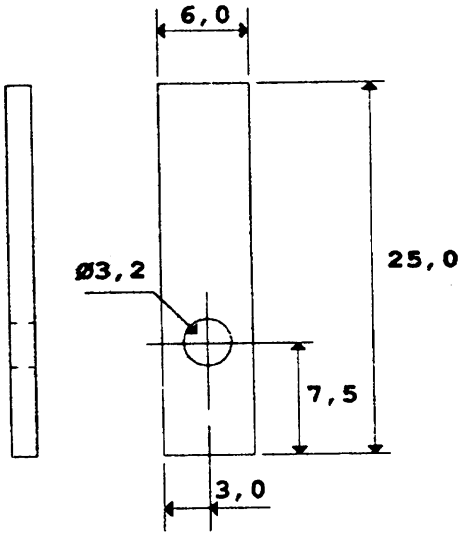
SCALE: 4:1

DRAWN	PMK	PMK	PMK
CHKD			
DATE	5.6.87	17.11.87	22.3.88
CHANGE	AMR E451	ECO E428	ECO 2007
ISSUE	1	2	3

TITLE A1
PODULE PCB MOUNTING BRACKET (STANDARD)

ACORN COMPUTERS Ltd | A4 | DRG. No. 0276,204/

SPACER



All dimensions in mm

Tolerance: ±0,2

Material: 1,6mm ±0,1 Zintec or zinc plate & clear passivate.

Scale: 2:1

ISSUE:	1	2				
DATE:	9.11.87	20.11.90				
DRAWN:	B.Phillips	AMF				
CHANGE:	AMR E604	ECO 2492				

TITLE: SPACER

DRAWING No. A4/ 0276,227 /

SHEET 1 OF 1

ACORN COMPUTERS Ltd.