Manual P/N 454-43500

BVM Limited,
Hobb Lane,
Hedge End,
Southampton,
SO30 0GH, UK.
TEL: +44 (0)1489 780144
FAX: +44 (0)1489 783589
E-MAIL: sales@bvmltd.co.uk
WEB: http://www.bvmltd.co.uk

User's Manual

BVME3000
&
RP3000

6U MC68360
CPU Module

Board Revision **B**
Manual Revision **B** August 19, 1997

DISCLAIMER

The information in this document has been checked and is believed to be entirely reliable, however no responsibility is assumed for inaccuracies. BVM Ltd. reserves the right to make changes and/or improvements in both the product and the product documentation without notice. BVM Ltd. does not assume any liability arising out of the application or use of any product described herein; neither does it convey any licence under its patent rights or the rights of others.

USE OF PRODUCT

This product has been designed to operate in a VMEbus and IndustryPack™ compatible electrical environment. Insertion of the board into any slot which is not VMEbus compatible is likely to cause serious damage. Insertion and removal of the board from the backplane or IndustryPack(s)™ or cable(s) from the board must not be done whilst in a powered condition.

Do not lever out any devices from the product, which uses surface-mounted devices extensively, as these can be fractured by excessive force.

This product uses devices sensitive to static electricity. Ensure adequate static electricity precautions are observed when handling the product and associated devices.

RF. INTERFERENCE

This product complies with European Council Directive 89/336/EEC (EMC directive), and conforms to EN55022:1995 Class B (Limits and methods of measurement of radio interference characteristics of information technology equipment) and EN50082-1:1992 (Electromagnetic compatibility - Generic immunity standard, Part 1: residential, commercial and light industry) when used in accordance with the BVM EMC Guidelines Manual part number 454-77000 (available on request from BVM Ltd.).

GENERAL NOTICES

UNPACKING AND INSPECTION

This product contains components which are susceptible to static discharge, and should be handled with appropriate caution.

Upon receipt of this product, visually inspect the board for missing, broken or damaged components and for physical damage to the printed circuit board or connectors. This product was shipped in perfect physical condition. Any physical damage to the product is the responsibility of the shipping carrier and should be reported to the carrier's agent immediately.

RETURN OF GOODS

Before returning a product for repair, verify as well as possible that the suspected unit is at fault. Then call BVM Ltd. for a Customer Return (CR) number. Carefully package the unit, in the original shipping carton if this is available, and ship prepaid and insured, preferably by courier, with the CR number written on the outside of the package.

Include a return address and the telephone number of a technical contact, and a detailed description of the observed fault. For out-of-warranty repairs, a purchase order for repair charges must accompany the return. BVM Ltd. will not be responsible for damage due to improper packaging of returned items. Out of warranty repairs can be arranged, and will be charged on a material and labour basis, subject to a minimum repair charge. Return transportation and insurance will be charged as part of the repair and is in addition to the minimum charge.

SOFTWARE LICENCE NOTICE

Any software that is provided as Copyright BVM Ltd. is proprietary and confidential property of BVM Ltd., and each single copy is given on the agreed understanding that it is licensed for use on product combinations supplied by BVM Ltd. or their appointed distributors only. The software product may not be copied (except for backup purposes), given away, rented, loaned, reproduced, distributed or transmitted in any way or form, in whole or in part, without written permission of BVM Ltd.

This applies to any merged, modified or derivative version of the software including, but not limited to, versions produced by customising, translating, reverse engineering, decompiling or disassembly.

This licence may be automatically terminated without notice if any of its provisions are breached. Reasonable legal costs may be awarded to the prevailing party in connection with this licence agreement. Use of, or accepted delivery of these products shall constitute your acceptance of the provisions of this licence agreement.

WARRANTY

A) BVM Ltd. warrants that the articles furnished hereunder are free from defects in material and workmanship for one year after the date of shipment.

B) All warranties and conditions, express and implied, statutory and otherwise, as to the quality of the goods or their fitness for any purpose are hereby excluded and with the exception of liability for death or personal injury caused by negligence as defined in the Unfair Contract Terms Act 1977 the seller shall not be liable for any loss, injury or damage arising directly or indirectly from the use, application or storage of such goods.

C) Subclause (B) above shall not apply where the buyer deals as a consumer as this expression is defined in the Unfair Contract Terms Act 1977.

D) The liability of BVM Ltd. hereunder shall be limited to repair or replacement at the manufacturers discretion of any defective unit. Equipment or parts which have been subject to abuse, misuse, accident, alteration, neglect, unauthorised repair or installation are not covered by this warranty. BVM Ltd. shall have the right of determination as to the existence and cause of any defect.

E) The warranty period of the replacement or a repaired product or part shall terminate with the termination of the warranty period with respect to the original product or part for all replacement parts supplied or repairs made during the warranty period.

F) Although BVM Ltd. offer a high level of technical support and advice, due to the complex nature and wide application of product configurations it is the responsibility of the purchaser to be satisfied at the time of purchase that the products are suitable for the final application.

G) The term 'Software' used herein is defined as 'any program data or code in source or binary format recorded in or on any readable device or media'.

H) BVM Ltd. will effect all reasonable effort to resolve accepted reproducible software errors reported within 12 months of purchase. Acceptance of an error shall solely be based on conformance to supporting specifications. Proper operation of earlier releases is not guaranteed.

NOTICES

OS-9 is a registered trademark of Microware Systems Corporation.

IndustryPack is a registered trademark of Greenspring Computers.

Table Of Contents

Contents

Page

1. Introduction

1.1 Scope

This manual provides:-

A getting started guide.
Configuration details.
A user reference guide.
A memory map.
A map of all register locations.
A detailed description of all dedicated registers.
Details of implementation specific considerations for major devices.
General Hardware Description.

This user manual does not provide:-

Detailed data on the operation of the MC68EN360.
Details of the IndustryPack™ Specification.

Information is provided to allow the module to be integrated into a system and configured by the system engineer. The User manual is intended for use by system integrators, service personnel, software engineers and end users.

Unless otherwise stated, address information is in hexadecimal notation.

The term “IP” is used as an abbreviation for “IndustryPack™” throughout this manual.

This document has been developed as a User manual for the BVME3000 and RP3000. For this reason, the module will be identified as the BVME3000 in universal sections of text, the BVME3000 in sections of texts relating to the VME module and the RP3000 in sections of text relating to the Remote processor.

1.2 BVME3000/RP3000 Series Part Numbers

The BVME3000 series is available in two different types, the BVME3000 which is a 6U VME master/slave card and an RP3000 which is intended to be used as a Remote Processor. The BVME3000 and RP3000 are available in various memory options, the more popular ones are listed below. Please contact your supplier for details of other configurations.

452-43511 6U VME Module with 2Mbytes FLASH, 4Mbytes DRAM and 512Kbytes SRAM.
452-43512 6U VME Module with 2Mbytes FLASH, 8Mbytes DRAM and 512Kbytes SRAM.
452-43514 6U VME Module with 2Mbytes FLASH, 16Mbytes DRAM and 512Kbytes SRAM.

453-43511 Remote Processor with 2Mbytes FLASH, 4Mbytes DRAM and 512Kbytes SRAM.

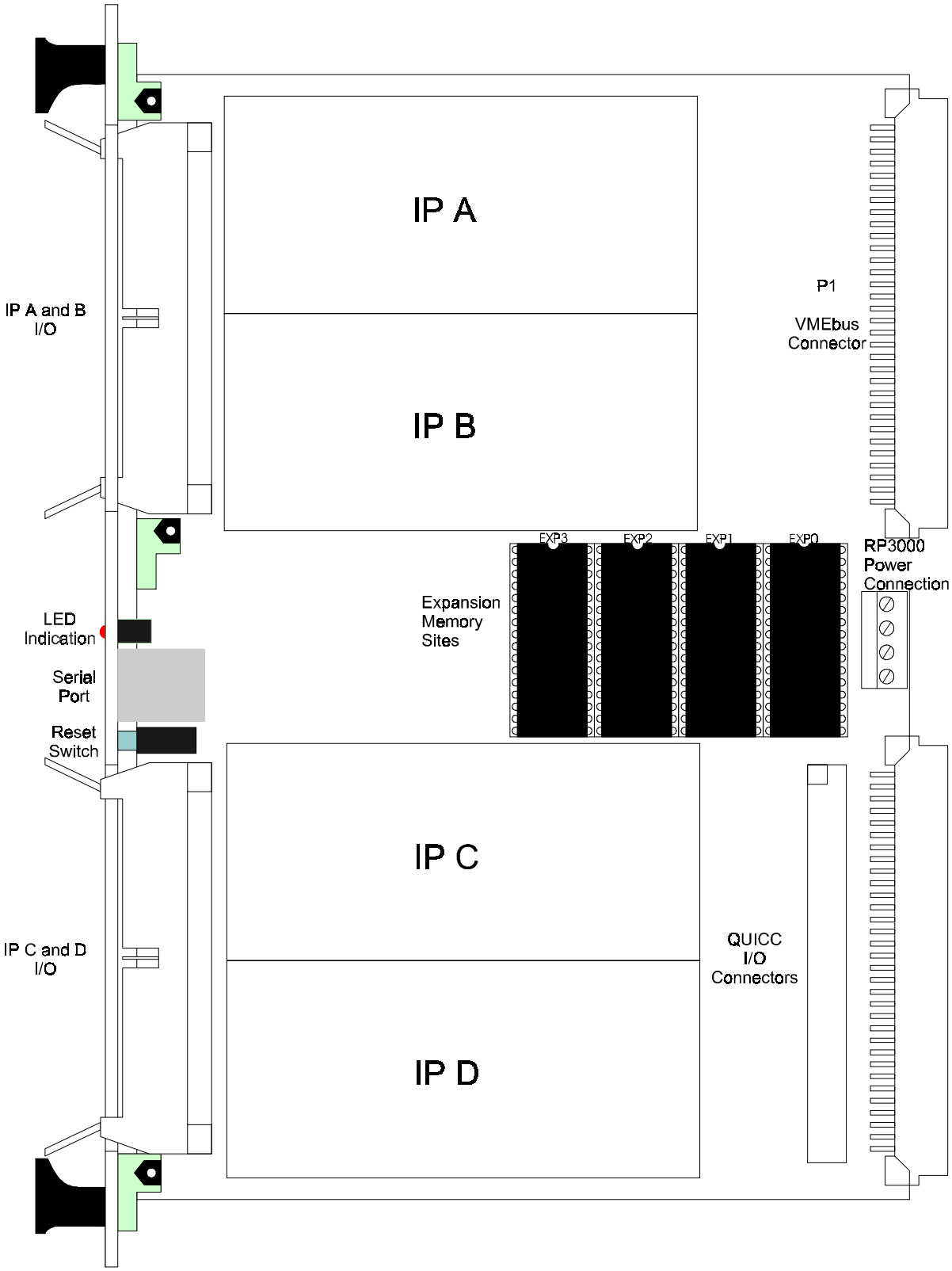
An engineering kit is available which is designed as a module development kit, and includes a BDM interface cable and software, the MC68360 user manual and an IndustryPack™ Specification.

453-43600 Engineering Kit.

453-43601 BVME3000 CPU32 ICD Cable (background Debug Mode Interface)

Other variants of the BVME3000/RP3000 are available, contact your supplier for details.

2. Overview



2.1 Features

- Quad Integrated Communication Controller Ethernet Variant (QUICC) MC68EN360 operating at 32MHz.

- The 68EN360 provides general purpose I/O pin's which can be configured to provide a combination from some of the following features:-

  4 SCC can provide with external buffering on each channel :-
  - 10Mbps Ethernet,
  - Synchronous / Asynchronous RS232,
  - Synchronous / Asynchronous RS422/485,
  - X25,
  - ISDN.

  2 SMC can provide extra transparent serial interface's.
  4 General Purpose Counter Timers.
  SPI Serial Peripheral Interface.
  PIP 18 bit Parallel Interface Port.

- 4/8/12/16 Mbytes 32-bit wide local DRAM.

- 1 or 2 Mbytes of Boot sector FLASH.

- 512 Kbytes 32-bit wide Non Volatile Dual Ported SRAM.

- 256 byte EEPROM for configuration settings.

- 4 x 32pin sockets for SRAM, EPROM or FLASH Expansion (32bit).

- Reset/Configuration Switch.

- Indication LED's.

- Real Time Clock (battery backed).

- RS232 (422/485 option) Serial port shared with the 68360's Transition Connector via an RJ45 Connector.

- Optimised A24/D16 master/slave VMEbus interface.
  - VMEbus Interrupter.
  - VMEbus Interrupt handler.
  - Location monitor - Mailbox Interrupt.

- System Controller Functions:-
  - Single level Arbiter.
  - RESET, SYSCLK generator, Bus timeout BERR generator.

- 68360 I/O connection via a 60 Way IDC or P2 connector, with fused supply for Transition Modules.

- Single slot 6U form factor. Can be used as a non VMEbus stand alone variant - the RP3000.

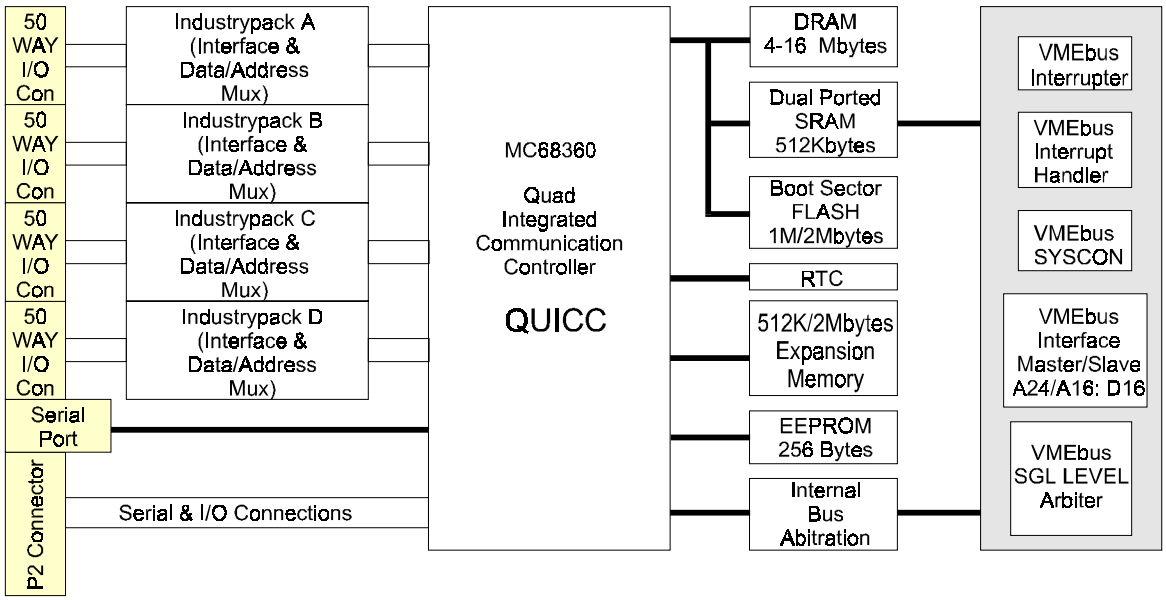
- +5V only operation, (excluding IndustryPack™ requirements).
- Fully compatible to VMEbus specification revision C.1.
- Four 16-bit IndustryPack™ Logic Interface Specification Rev 0.7.1. compatible sites:-
  8/32MHz Option,
  Double height 32-bit access supported,
  DMA support on sites A and C.
- Motorola BDM (Background Debug Mode) interface support.
- Full OS9 Software support.

2.2 Applications

- VMEbus Main System processor.
- VMEbus Intelligent I/O processor.
- Comms Engine.
- High Performance Remote Processor.

3. Operation

3.1 Block Diagram



3.2 Processor

The BVME3000 is based on the MC68EN360FE33 32-bit Quad Integrated Communications Processor from Motorola. The part can operate up to speeds of 33MHz, however in this application the processor operates at 32MHz.

The 68EN360 provides a number of I/O functions including 4 Serial Communication Controllers, 2 SMC Transparent Serial ports, counter timers, general purpose I/O and an SPI port. The 68EN360 is fitted with the Ethernet protocol which allows Ethernet to run on any of the SCC's but not necessarily at the same time. The 68EN360 provides a BDM and JTAG test facility for module debug.

3.3 Memory

FLASH The BVME3000 is factory fitted with either 1Mbyte or 2Mbyte of FLASH memory on a 16-bit word bus. The FLASH is the main boot memory and must be programmed via the BDM interface before the card can boot. The FLASH is programmed in 128Kbyte banks, bank 0 can be write protected using a jumper link to give a virtual boot block.

DRAM The BVME3000 can be factory fitted with between 0 and 16 Mbytes of DRAM, selected in 4Mbyte banks. The DRAM supports full 32-bit accesses.

SRAM The BVME3000 can be fitted with 512Kbytes of SRAM and supports 32-bit accesses. The SRAM is Dual Ported to the VMEbus as a 512Kbyte block. The block can be programmed in the Control Register to locate anywhere in the VMEbus A24 space on 512Kbyte boundaries. The memory is non-volatile providing 7 day continuous backup.

Expansion The BVME3000 is fitted with 4 32-pin expansion sockets. The sockets provide 32-bit access (all sites must be fitted with identical device types). The boards are shipped to support 128K x 8 SRAM, however the sockets can be PLD configured to support SRAM FLASH or EPROM and a number of device sizes up to 1024K x 8. Please contact your supplier for a list of supported devices.

BVME3000 supports up to 70nS device as standard, slower devices are supported (up to 100nS) by software reconfiguration.

All the above memory devices use the 68EN360's on board chip selects, please see section 7 for configuration details.

EEPROM The BVME3000 utilises the SPI mode in the 68360's I/O block to access the 256bytes of EEPROM. The part can be programmed to be block protected in ¼ or ½ device size blocks. A user link can be used to fully protect against accidental writes.

The EEPROM can be removed from the BVME3000 freeing the SPI signals for their alternative functions.

3.4 Real Time Clock

The BVME3000 provides a battery backed real time clock shadowed under the even byte of the lower 1Mbyte of the FLASH memory. The clock provides full date and time functions, and is battery backed using a lithium battery giving typically 10 years of non volatile operation

3.5 Serial Port

The BVME3000 utilises SCC4 on the 68360 as a general purpose RS232 port. Asynchronous baud rates of up to 115.2Kbits/s are supported, this restriction is due to the serial buffer and not the 68360. The port is made available via an RJ45 to reduce panel space. This and all the other SCC's are available via the 68360 I/O connections P2 and JP5. If SCC4 is required externally the RS232 Serial Buffer should be removed.

3.6 Interrupts

3.6.1 VMEbus Interrupt Handler

The BVME3000 will support VMEbus interrupts on all 7 levels. A board control register can be programmed to enable each IRQ level individually.

A user vector VMEbus interrupt causes the CPU to reply with a VMEbus Master Interrupt acknowledge cycle. This cycle uses only IACK that is broadcast in a similar way to the addresses. The A1, A2 and A3 address lines indicate the address level being handled.

The interrupting device returns an ID vector on the odd data byte. This is used as the user vector by the CPU.

3.6.2 Internal Interrupts

Internal CPU interrupts are generated from a variety of sources, as detailed in the table below:

Level	Source	Type
7	VME IRQ7 Abort switch ACFAIL	Vectored Autovectored Autovectored
6	VME IRQ6 IndustryPack™	Vectored Vectored
5	VME IRQ5	Vectored
4	VME IRQ4	Vectored
3	VME IRQ3	Vectored
2	VME IRQ2 Location Monitor	Vectored Autovectored
1	VME IRQ1 IndustryPack™	Vectored Vectored
Internal	Variable (see following paragraph)	

IndustryPack™ interrupts are generated on either level 1 or level 6. A board control register is used to select the interrupt level of each IndustryPack™ individually.

The RP3000 does not support VME Interrupts.

The 68360 can also generate internal interrupts to control the operation of its internal peripherals. Although an external cycle does not occur the interrupts will effect the priority of the external interrupts and thought must be given to the internal interrupt levels. The 68360 has three main sources of Internal Interrupts:-

The CPM which controls the SCC's, parallel I/O, timers, DMA, SPI and the RISC timer table.
The PIT, which is an internal timer derived from the SPI clock.
The SWT which is the software watchdog timer.

The CPM and PIT can be programmed to generate an interrupt on any level while the SWT can only generate an Interrupt on level 7 if selected.

The CPM interrupt sources all share the same interrupt, however these can be prioritised inside the CPM.

3.6.3 VMEbus Interrupter

The BVME3000 can generate VMEbus interrupts on any programmable single level 1- 7 and responds with a software programmable ID to the subsequent interrupt acknowledge cycle. Writing the ID to the vector register causes a VMEbus interrupt to be generated on the selected level. The BVME3000 VMEbus interrupt ID vector may be programmed to suit the application.

3.7 VMEbus Interface

3.7.1 VMEbus Master

Byte or word Master accesses may be made to the standard (A24) and short (A16) address spaces. Read Modify Write (RMW) cycles are supported.

The BVME3000 supports the Release On Request (ROR) VMEbus arbitration method. The method uses FAIR requesting, ensuring each master has an equal chance of obtaining the bus. Digital bus busy filtering and arbitration interleaving is used to ensure premium arbitration performance.

3.7.2 VMEbus Slave

The on-board SRAM on the BVME3000 is dual ported onto the VMEbus. The VMEbus base address is programmed into the board control register which can be set on 512 Kbyte boundaries anywhere in the (A24) address space. The BVME3000 supports byte, word and RMW cycles and is compatible with VMEbus address pipelining.

The BVME3000 provides a Location Monitor Interrupt which is set by an external Bus Master accessing standard A24 space at either 512Kbytes above the SRAM if it is on a 1Mbyte boundary (A19 clear in the board control register), or by accessing the SRAM if it is on a 512Kbyte boundary (A19 set in the board control register).

3.8 VMEbus System Controller Functions

The BVME3000 provides a number of system controller functions that may be enabled as follows:-

RESET

Asserted if the +5V falls below +4.7V when link selected. VMEbus RESET has a minimum asserted period of 200mS.

Arbitration

The BVME3000 is configured to support Single level arbitration when enabled in a board control register.

SYSCLK

The BVME3000 provides a 16MHz VMEbus SYSCLK when link selected.

VMEBERR

The BVME3000 provides a 128µS Bus Timeout BERR signal when link selected.

The processor Bus Timeout BERR is 32µS signal and is not linked to the VMEBus Timeout BERR. Therefore when doing a VMEbus master access, if an internal timeout occurs, the user should not perform another VME master access for 96µS as the cycle cannot be guaranteed.

3.9 IndustryPack™

Four standard IndustryPack™ compatible sites are provided. The IndustryPack® interface complies fully with the IndustryPack™ specification. The four sites may be used individually for single IP's which are accessed as 16-bit wide, or in pairs (IPA and IPB or IPC and IPD) for double 32-bit wide IP's. IndustryPack™ operation is supported for both 8MHz and 32MHz speeds. Two DMA channels on the 68360 are made available for the IndustryPack™ interface. These can be programmed and link selected to operate as both on IPA, or one on IPA and one on IPC. The IP ID and I/O spaces are 256bytes each , and memory spaces are 8Mbytes. IP vectored interrupts are fully supported, and the Interrupt levels may be programmed as level 1 or 6.

3.10 Power Supply Monitor/Reset

A MAX791 provides power up/power down control for the non volatile RAM and Processor Reset. The Processor can generate an SRESET signal if a software reset occurs, which is driven into the MAX791 to generate a full system reset if link selected.

3.11 Local Bus Monitor/Watchdog

Local Bus moitoring on the BVME3000 is controlled by the 68360. A programmable register in the 68360's DPR must be set to enable BERR and to derive the timeout period. This function is necessary to prevent CPU hang-up as a result of invalid addresses being generated. VMEbus errors are monitored by the system controller. If the BVME3000 is a system controller a separate timer in the control logic will timeout if an address is not acknowledged.

The watchdog monitor on the BVME3000 is controlled by the 68360. A programmable register in the 68360's DPR must be set to enable the watchdog. The register can only be programmed once after reset, and initialises when the watchdog is enabled.

3.12 Configuration/Reset Switch

The BVME3000 has no configuration switch, but a combination of multiple RESET depressions and EEPROM storage can be used to allow access to a configuration program. The BVME3000 is fitted with a flush Reset Switch to prevent accidental operation and is Link selectable. Reset is asserted from the MAX791 for at least 200mS following release of the button.

3.13 LED Indicators

RED LED Indicates processor activity.

GREEN LED Indicates IP activity driven from combined buffered IP selection signals.

4. VMEbus Installation

The BVME3000 module is inserted into a vacant VMEbus slot. If it is to function as the system controller, then it should be positioned in the left most slot. It passes through all VMEbus daisy chained arbitration signals.

IACK should be jumpered to IAKIN on the backplane in the left hand slot. All interrupt IAKIN to IAKOUT and BGIN to BGOUT signals should be jumpered across vacant slots to the left of the module.

If it is not the system controller, it may be located in any of the VMEbus slots to the right of the VMEbus system controller.

To install the BVME3000:-

1. Ensure all backplane jumpers associated with the slot for the BVME3000 are removed.

2. Ensure the BVME3000 module is correctly configured for the target system.

3. Connect the 68360 I/O port to any transition modules via JP5, or the P2 connector as required.

4. Insert the BVME3000 module into the rack pushing the VMEbus connector fully home.

5. Secure the BVME3000 into the rack with the two fixing screws top and bottom.

6. Plug in serial cable to JP3 (if not using a Transition module).

7. Connect the IP I/O connections to the four 50 way front panel connectors.

Removal is the reverse of assembly.

If the test or application software fails, ensure that all installation instructions have been correctly carried out. Some typical reasons for incorrect operation are:-

1. Socketed components may become disturbed in transit. Push home all socketed components where suspect.

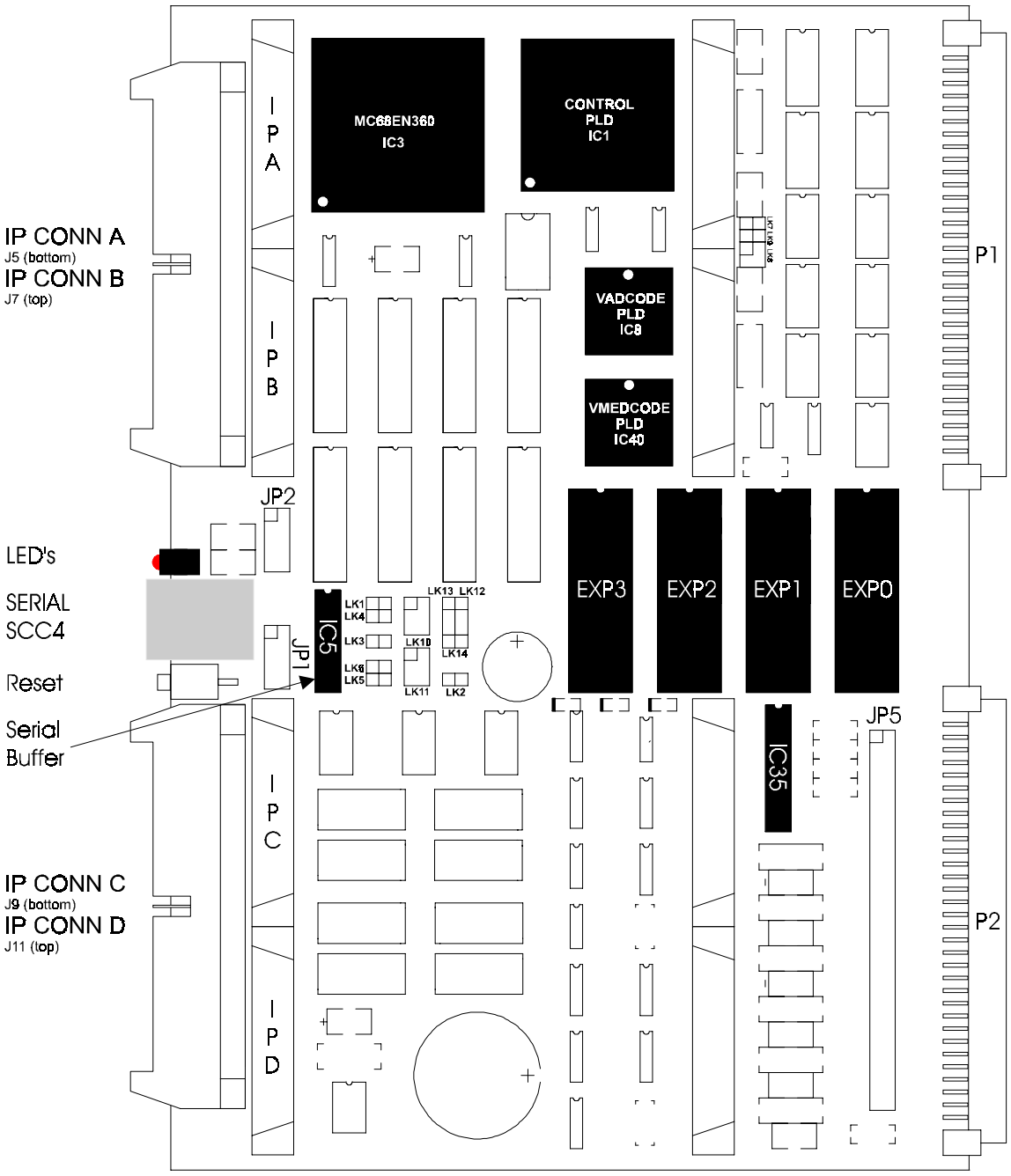
2. The BVME3000 module uses the VMEbus Address modifier codes to determine address significance. Ensure the host CPU module produces the correct address modifier codes.

3. Ensure that all links are configured to the default set-up or that any alterations to the default are correctly configured.

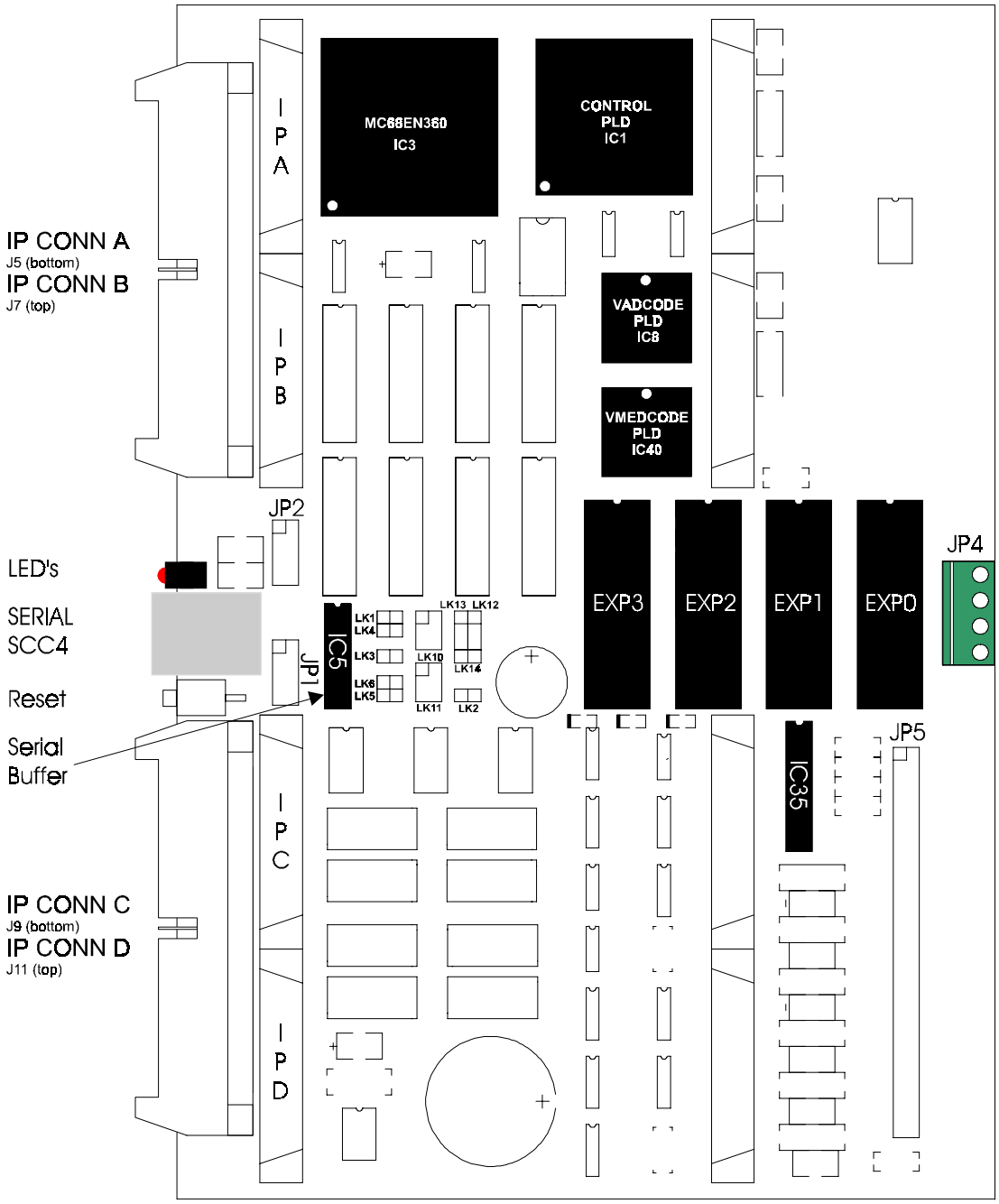
4. Ensure that the VMEbus backplane (if used) is correctly configured with regard to the daisy-chain signal jumpers and the IACK termination jumpers (if any).

5. Configuration

5.1 BVME3000 PCB Layout



5.2 RP3000 PCB Layout

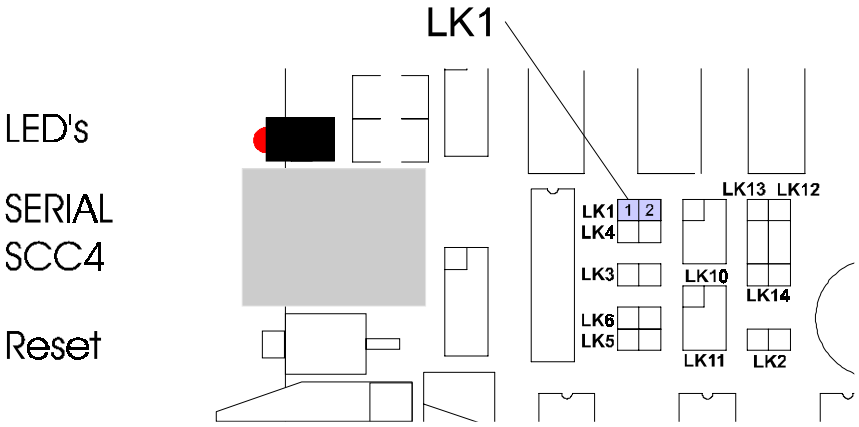


5.3 Link Definitions

The following link definitions show the links grouped in the same orientation as the layout drawing on the previous pages, i.e. the IndustryPack™ I/O connectors to the left. Link positions marked with a ✪ show the BVME3000 default configuration while link positions marked with a ⌖ show the RP3000 default configuration.

The following features on the BVME3000 are all link selectable:-

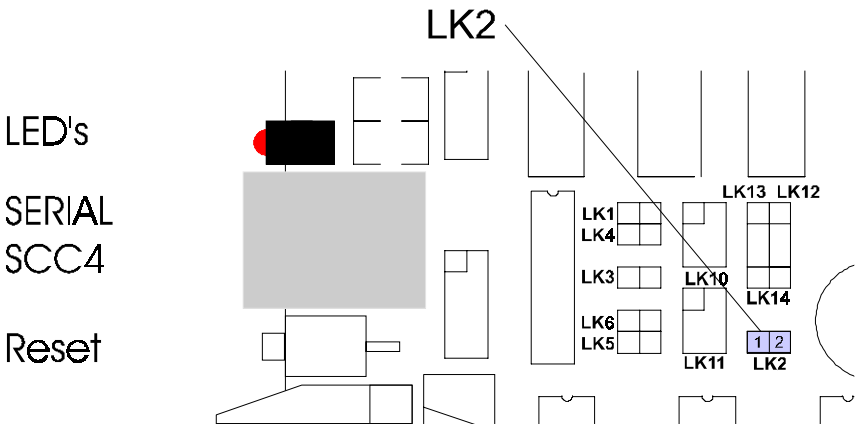
5.3.1 LK1 EEPROM Write Enable



LK1 when fitted fully disables SPI writes to the EEPROM.

1 & 2 Fitted	EEPROM writes are disabled
1 & 2 Omitted ✪ ⌖	EEPROM writes are enabled

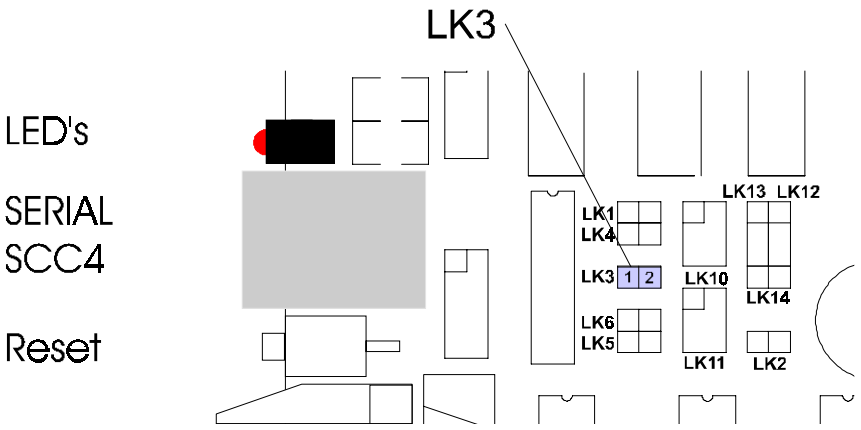
5.3.2 LK2 68360 TRI-STATE Enable



LK2 when fitted tells the 68360 to put its OUTPUT pins into tri-state. This pin can be used in conjunction with the JTAG interface for boundary scan operations.

1 & 2 Fitted	Tri-states all 68360 I/O pins
1 & 2 Omitted ✪ ⌖	Output pins set to defined functions

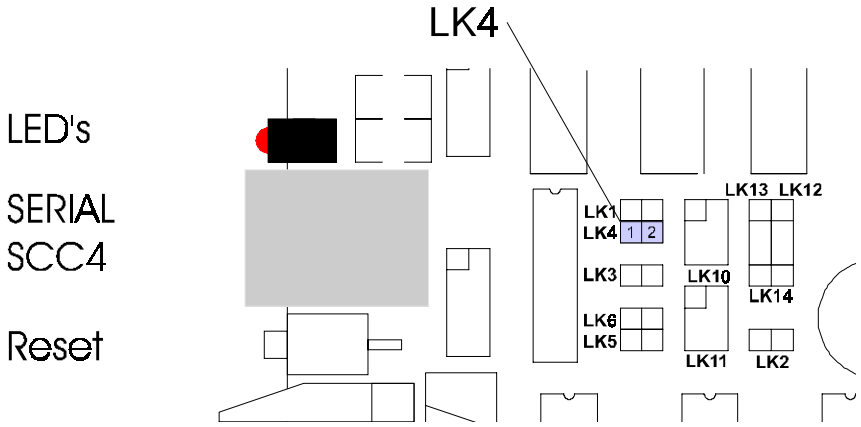
5.3.3 LK3 Abort



This link can be used as a header for an abort switch. When pin 1 is connected to pin 2, an autovectored level 7 interrupt is generated and the abort bit will be set in the Autovectored Level 7 source register.

1 & 2 Closed	Autovectored level 7 interrupt generated
1 & 2 Open	Autovectored level 7 interrupt not generated

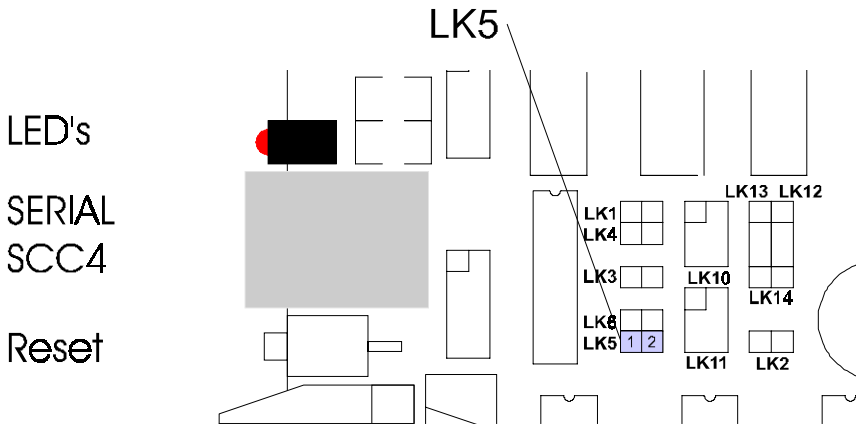
5.3.4 LK4 Boot Block programming Enable



LK4 when fitted allows the bottom 128Kbytes of FLASH to be programmed. This allows FLASH block 0 to become a virtual boot block.

1 & 2 Fitted	Programming to FLASH block 0 enabled
1 & 2 Omitted ✪ ⌖	Programming to FLASH block 0 disabled

5.3.5 LK5 Software Reset Enable

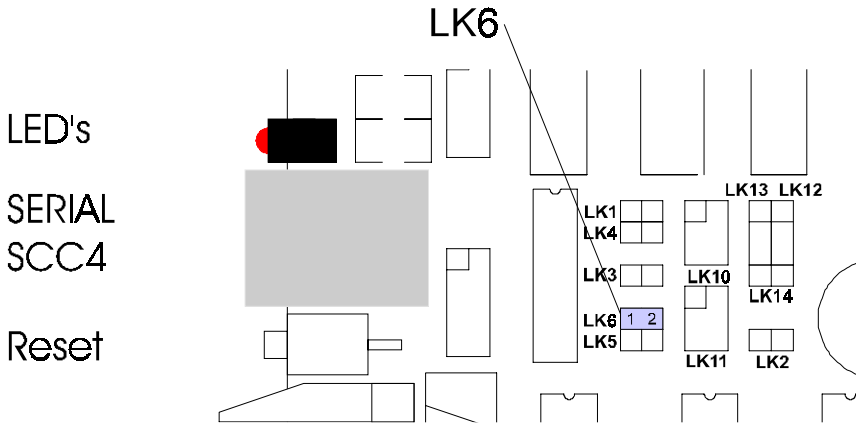


LK5 When fitted allows Reset generated in software to cause a BVME3000 reset.

1 & 2 Fitted ✪ ⌖	Software reset source enabled
1 & 2 Omitted	Software reset source disabled

Note: When using an ICD BDM Interface cable for either programming or debug purposes, LK5 must be removed.

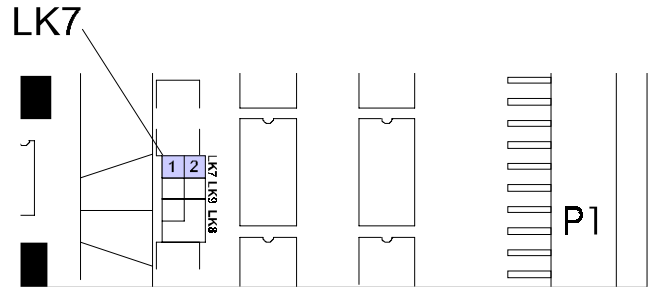
5.3.6 LK6 Reset Switch Enable



LK6 when fitted allows the Reset switch to generate a BVME3000 reset.

1 & 2 Fitted ✪ ⌖	Reset switch source enabled
1 & 2 Omitted	Reset switch source disabled

5.3.7 LK7 VMEbus SYSCLOCK Enable

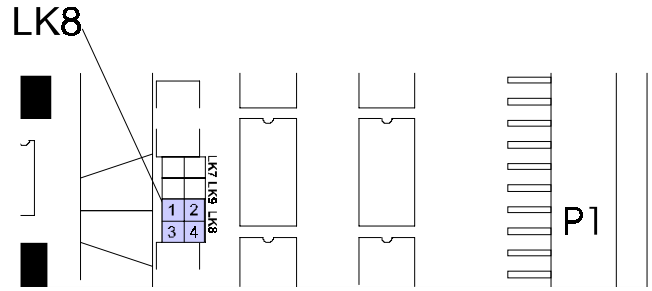


LK7 when fitted causes the BVME3000 to provide the VMEbus 16MHz SYSCLK signal for the system. This link should only be fitted if the BVME3000 is the system controller.

1 & 2 Fitted ✪	VMEbus SYSCLOCK generation enabled
1 & 2 Omitted	VMEbus SYSCLOCK generation disabled

Note:- this link block is not fitted to the RP3000.

5.3.8 LK8 VMEbus Reset Operation



When the link between pins 1 and 2 is fitted, the VMEbus RESET will be driven by the BVME3000 reset. This link should only be fitted when the BVME3000 is the system controller.

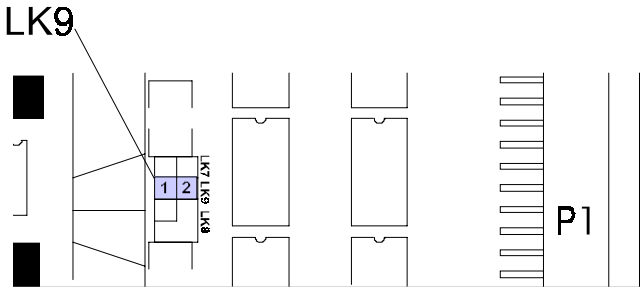
1 & 2 Fitted ✪	BVME3000 Reset to VMEbus disabled
1 & 2 Omitted	BVME3000 Reset to VMEbus disabled

When the Link between pins 3 and 4 is Fitted, the BVME3000 RESET will be driven by the VMEbus reset.

3 & 4 Fitted ✪	VMEbus Reset to BVME3000 enabled
3 & 4 Omitted	VMEbus Reset to BVME3000 disabled

Note:- this link block is not fitted to the RP3000.

5.3.9 LK9 VMEbus BERR Enable

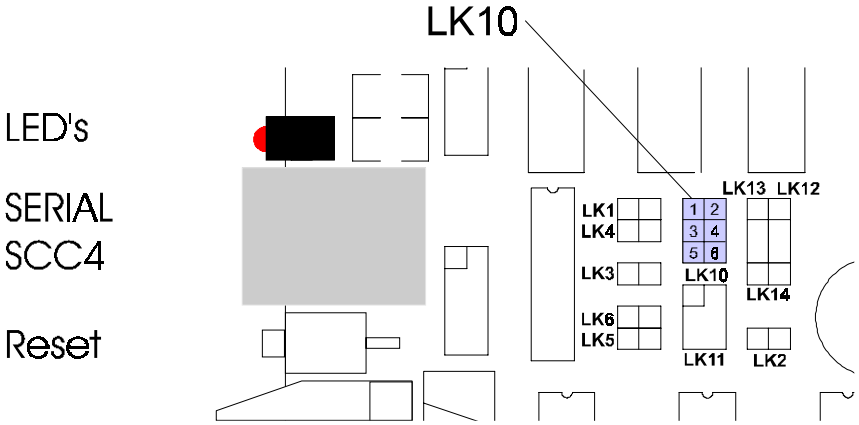


LK9 when fitted causes the BVME3000 to provide the VMEbus BERR timer for the system. This link should only be fitted if the BVME3000 is the system controller.

1 & 2 Fitted ✪	VMEbus BERR Generation enabled
1 & 2 Omitted	VMEbus BERR Generation disabled

Note:- this link block is not fitted to the RP3000**.**

5.3.10 LK10 DMA channel 1 Enable

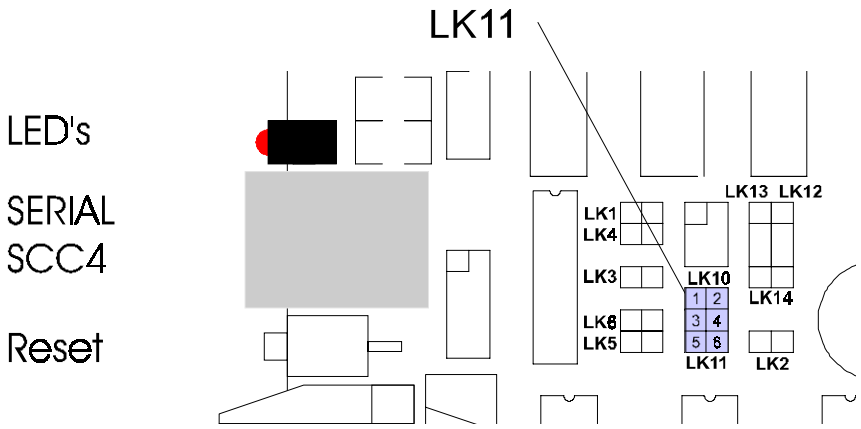


LK10 when fitted connects DMA channel 1 on the 68360 to IPA DMA channel 0. When omitted, the 68360 I/O pins are available to the 68360 I/O connector.

1 & 2 Fitted 3 & 4 Fitted 5 & 6 Fitted	DMA accesses to IPA channel 0 enabled
1 & 2 Omitted ✪ ⌖ 3 & 4 Omitted ✪ ⌖ 5 & 6 Omitted ✪ ⌖	DMA accesses to IPA channel 0 disabled

Note: not available on revision B boards.

LK11 DMA channel 2 Enable



LK11 when fitted connects DMA channel 2 on the 68360 to IPA DMA channel 1 or IPC DMA Channel 0, depending on the status of the IPDMA bit in the IPA Clock/ DMA Control Register. When omitted, the 68360 I/O pins are available to the 68360 I/O connector.

1 & 2 Fitted 3 & 4 Fitted 5 & 6 Fitted	DMA accesses to IPA channel 1or IPC channel 0 enabled (depending on Software control)
1 & 2 Omitted ✪ ⌖ 3 & 4 Omitted ✪ ⌖ 5 & 6 Omitted ✪ ⌖	DMA accesses to IPA channel 1or IPC channel 0 disabled

Note: not available on revision B boards.

5.3.11 Factory Links

The BVME3000 is fitted with three factory links:- LK12, LK13 and LK14. These links should not be changed by users.

5.4 Indicators

RED LED Indicates processor activity.

GREEN LED Indicates IndustryPack™ activity driven from a combined buffered IndustryPack™ selection signals.

6. Connector Pinouts.

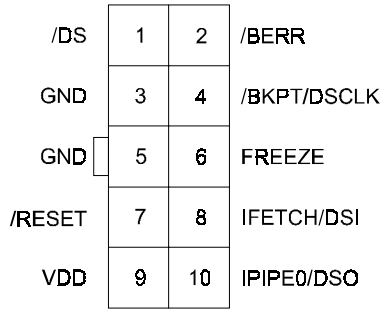
6.1 JP1 BDM (Background Debug Mode)

The BDM port on the BVME3000 is utilised for programming the FLASH memory and as an online debugging tool.

By toggling the BKPT pin during a CPU Reset, the 68360 drops into the BDM mode. Once in BDM mode the user can poke registers and run simple code, perform debug functions to the 68360 while the chip is running, stopped or even before it has performed its first cycle. These functions enable the boot code to be programmed into the FLASH memory before the 68360 is started.

The interface is generally connected via an BDM interface circuit to the centronics port on a PC. The interface circuit is the public domain, but more sophisticated interfaces are on the market. There are also many debuggers available, again there is a public domain version but many vendors offer software which is more suitable.

The BVME3000 engineering kit includes an BDM interface connector and debug software. Please contact your supplier for more information.

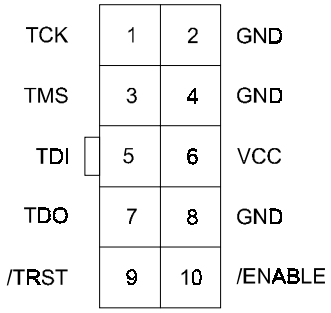


The BDM connection is via a 2 x 5 way header. The normal convention allows for 8 pins but the 68360 interface includes /DS and BERR to allow hardware breakpoints to be implemented.

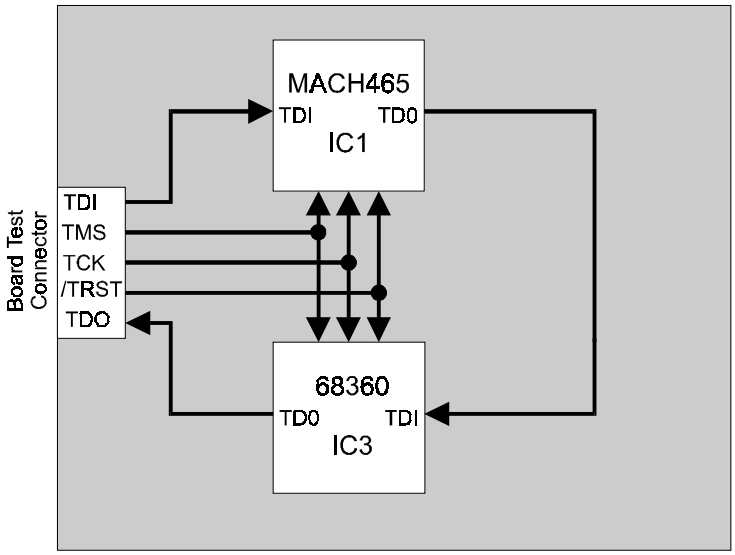
Note: When using an ICD BDM Interface cable for either programming or debug purposes, LK5 the Software Reset Enable link must be removed

6.2 JP2 JTAG

The JTAG connection is via a 2 x 5 way header, the pinout matching AMD's MACH programming lead.



Both the 68360 and the logic cell support the JTAG interface, the 68360 has a 3 bit instruction register while the MACH465 has a 6 bit instruction register. At present BVM only supports JTAG access to the logic cell.



It is possible that future developments may allow some boundary scan testing.

The Boundary scan interface includes the ability to place all the 68360 outputs into tri-state condition, this is controlled by LK2.

6.3 SCC4 Serial Connection

The serial port signals for the BVME3000 are derived from SCC4 in the 68360. The RS232 level signals are available from an RJ45 socket. There is not a standard pinout for RS232 using RJ45 connectors therefore the pinout has been designed for 1 to 1 connection to a 25 way D connector. TX DATA and RTS are OUTPUT signals, RX DATA and CTS are INPUT signals. If SCC4 is required for an external operation ensure the MAX202 buffer is removed.

6.4 68360 I/O Connections

The 68360 I/O connections are available via two connectors. On the BVME3000 variant the I/O can be accessed either via the P2 connector on rows a and c or via JP5, the 60 way boxed header mounted behind P2. On the RP3000 the P2 connector is removed leaving I/O to be taken from JP5, the 60 way boxed header. JP5 can also be used for any top mounting transition modules. Pin 1 is shown on JP5 by a triangle on the connector body.

6.4.1 P2 Connections

P2 Connection	Row a	Row c
1	GND	GND
2	/RESET	/ABORT
3	PA0	PA1
4	PA2	PA3
5	PA4	PA5
6	PA6	PA7
7	PA8	PA9
8	PA10	PA11
9	PA12	PA13
10	PA14	PA15
11	PB0	PB1
12	PB2	PB3
13	PB4	PB5
14	PB6	PB7
15	PB8	PB9
16	PB10	PB11
17	PB12	PB13
18	PB14	PB15
19	PB16	PB17
20	GND	GND
21	PC0	PC1
22	PC2	PC3
23	PC4	PC5
24	PC6	PC7
25	PC8	PC9
26	PC10	PC11
27	GND	GND
28	+5V	+12V
29	GND	GND
30	+12V	+5V
31	N/C	N/C
32	N/C	N/C

The centre row (row b) should not be used, as the pins have specific VME functions.

Power is made available on the 68360 I/O connector to allow for signal conditioning. Up to 2A can be drawn from the +5V rail or the ±12V rails, all power rails are fused. Note the baseboard will operate from +5V only, therefore it is not always necessary to apply +12V to the system

6.4.2 JP5 Connections

GND	1	2	GND
/ABORT	3	4	/RESET
PA1	5	6	PA0
PA3	7	8	PA2
PA5	9	10	PA4
PA7	11	12	PA6
PA9	13	14	PA8
PA11	15	16	PA10
PA13	17	18	PA12
PA15	19	20	PA14
PB1	21	22	PB0
PB3	23	24	PB2
PB5	25	26	PB4
PB7	27	28	PB6
PB9	29	30	PB8
PB11	31	32	PB10
PB13	33	34	PB12
PB15	35	36	PB14
PB17	37	38	PB16
GND	39	40	GND
PC1	41	42	PC0
PC3	43	44	PC2
PC5	45	46	PC4
PC7	47	48	PC6
PC9	49	50	PC8
PC11	51	52	PC10
GND	53	54	GND
+12V	55	56	+5V
GND	57	58	GND
+5V	59	60	+12V

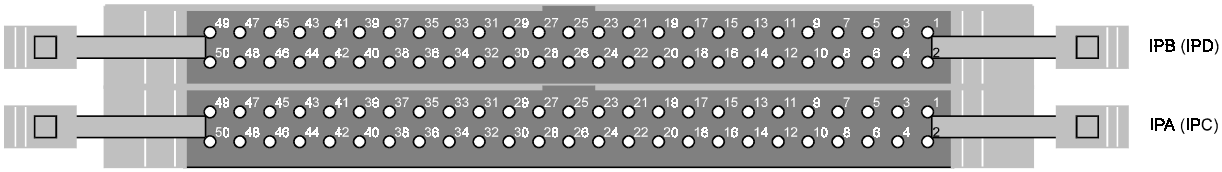
Note:- Odd number pins on JP5 connect to P2 row c, and even numbered pins on JP5 connect to P2 row a. Thus a ribbon cable connected to P2 can result in the JP5 pinout on an IDC connector on the ribbon cable.

6.5 J5/J7 and J9/J11 IndustryPack™ I/O

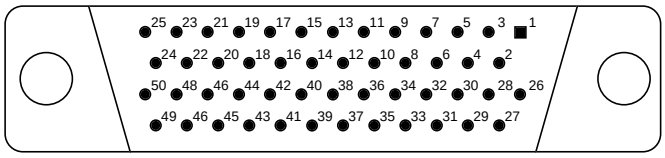
Each of the 50 pins on each I/O connector for the four IndustryPack™ sites, connects to a like-numbered pin on the four corresponding flat cable connectors. IPA connects to J5, IPB connects to J7, IPC connects to J9 and IPD connects to J11. J5 and J7 are a double stacked 50 way IDC fitted through the front panel towards the top of the module, while J9 and J11 are a double stacked 50 way IDC fitted through the front panel towards the bottom of the module. The IndustryPack™ I/O connector, the BVME3000 flat cable connectors, and the wires on the ribbon cables are all numbered identically from 1 to 50.

Pin 1 on IP and BVME3000 connectors is marked with a square pad, observable from the solder side of the respective board. Pin 1 is shown on J5/7/9/11 by a triangle on the connector body. Pin 1 is typically marked on ribbon cable with a red stripe and on ribbon cable connectors with a manufacturer's mark, often a moulded textured triangle.

Caution: This consistent pin numbering system is not maintained with many mass-terminated connectors. Each type of connector has its own intrinsic pin numbering system. Systems integrators or users making their own cables must be certain which pin corresponds to which signal.

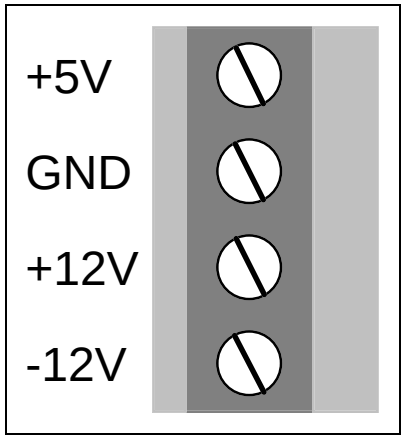


The pin assignment of the IP I/O interface connector is fixed by the connector manufacturer and repeated in the IndustryPack™ Specification. This assignment is shown below.



IndustryPack™ Connector Pin Numbering
Viewed from solder side of BVME3000

6.6 JP4 RP3000 Power Connections



The RP3000 requires 5 volts DC for it's operation. The IP interface also requires 5 volts, and some IPs use ±12 volts DC, however the RP3000 itself does not require the ±12 volts DC supply for it's own operation. The power supply is connected to the four way terminal block with screw terminals suitable for wire up to 1.5mm^2.

7. Programming

The following section describes the programming information required to run the basic functionality of the BVME3000. This section includes specific programming information for the BVME3000's Board control Registers, as well as a description of the 68360's register settings when relevant to the BVME3000's hardware operation. For a more detailed description of the 68360's operation and programming, for example Ethernet and SCC operation, please see the 68360 User Manual available in the BVME3000's Engineering Kit (contact your supplier for more details).

7.1 Address Map

The recommended address map for the BVME3000 is shown below. The board utilises the 68360's chip select lines for memory control, so these locations may be adjusted as required to the users requirements. The VME and IP locations however are set in a PLD and cannot be changed without use of a special PLD. Accessing a non mapped area will generate a Bus Error provided the Bus Monitor is enabled in the 68360. Accessing a peripheral which has not been fitted will generate a Bus Error providing the Bus Monitor is enabled in the 68360 and the peripherals chip select register has not been configured.

Address Range	Function	Size	CS	Width
00000000 - 001FFFFF	FLASH PROM	1 or 2 Mbytes (Variant)	0	D16
00200000 - 0027FFFF	SRAM	512Kbytes	5	D32
00400000 - 007FFFFF	Expansion Memory	4 Mbytes User Option	6	D32
01000000 - 01FFFFFF	DRAM	4/8/12/16 Mbytes (Variant)	1/2/3/4	D32
F0000000 - F07FFFFF	IP0A Memory Space	8 Mbytes	EXT	D16
F0800000 - F0FFFFFF	IP0B Memory Space	8 Mbytes	EXT	D16
F1000000 - F17FFFFF	IP0C Memory Space	8 Mbytes	EXT	D16
F1800000 - F1FFFFFF	IP0D Memory Space	8 Mbytes	EXT	D16
F4000000 - F4FFFFFF	IP0A/B Memory Space	16 Mbytes	EXT	D32
F5000000 - F5FFFFFF	IP0C/D Memory Space	16 Mbytes	EXT	D32
FE000000 - FEFFFFFF	VME Standard (A24)	16 Mbytes	EXT	D16
FF000000 - FFFFFFFF	I/O map	16 Mbytes	EXT/7	D32

7.2 I/O Map

The above map shows a function called the I/O map. This map contains the settings for all the board's I/O configurables including VME Short I/O, IP I/O and ID space and the various Board Control Registers. The 68360's Dual Port Ram is also located in this area, by programming the 68360's MBAR register to FF000000.

Address Range	Function	Size	CS	Width
FF000000 - FF001FFF	68360 Dual Port RAM	8 Kbytes	INT	D32
FF800000 - FF80007F	IP0A I/O	128 Bytes	EXT	D16
FF800080 - FF8000FF	IP0A ID	128 Bytes	EXT	D16
FF800100 - FF80017F	IP0B I/O	128 Bytes	EXT	D16
FF800180 - FF8001FF	IP0B ID	128 Bytes	EXT	D16
FF800200 - FF80027F	IP0C I/O	128 Bytes	EXT	D16
FF800280 - FF8002FF	IP0C ID	128 Bytes	EXT	D16
FF800300 - FF80037F	IP0D I/O	128 Bytes	EXT	D16
FF800380 - FF8003FF	IP0D ID	128 Bytes	EXT	D16
FF800800 - FF8008FF	IP0A/B I/O	256 Bytes	EXT	D32
FF800900 - FF8009FF	IP0C/D I/O	256 Bytes	EXT	D32
FF800F00 - FF800FFF	Board Control Registers	256 Bytes	EXT	D8
FFFF0000 - FFFFFFFF	VMEbus Short I/O A16	64 Kbytes	7	D16

7.3 BVME3000 Board Control Registers

Location	Function					Size	Read/Write
FF800F01	IPA clock speed / DMA1 select					Byte	Write
7	6	5	4	3	2	1	0
						IPDMA	IPACLK

FF800F03	IPB clock speed					Byte	Write
7	6	5	4	3	2	1	0
							IPBCLK

FF800F05	IPC clock speed					Byte	Write
7	6	5	4	3	2	1	0
							IPCCLK

FF800F07	IPD clock speed					Byte	Write
7	6	5	4	3	2	1	0
							IPDCLK

FF800F09	IndustryPack™ Interrupt Level					Byte	Write
7	6	5	4	3	2	1	0
INTREQD1	INTREQD0	INTREQC1	INTREQC0	INTREQB1	INTREQB0	INTREQA1	INTREQA0

FF800F0B	VMEbus Arbitration					Byte	Write
7	6	5	4	3	2	1	0
							ARBIT

FF800F0D	Expansion Memory Wait State					Byte	Write
7	6	5	4	3	2	1	0
							EXP

FF800F0F	EEPROM Chip Select					Byte	Write
7	6	5	4	3	2	1	0
							SPICS

FF800F11	Reserved						
7	6	5	4	3	2	1	0

FF800F13	Autovectored Level 7 Source					Byte	Read
7	6	5	4	3	2	1	0
						ABORT	ACFAIL

FF800F15	Location Monitor Interrupt Control					Byte	Write
7	6	5	4	3	2	1	0
							LOCINT

FF800F17	VMEbus Interrupt Enable					Byte	Write
7	6	5	4	3	2	1	0
VRIQ7	VIRQ6	VIRQ5	VIRQ4	VIRQ3	VIRQ2	VIRQ1	

FF800F19	VMEbus Interrupt Vector					Byte	Write
7	6	5	4	3	2	1	0

VEC7	VEC6	VEC5	VEC4	VEC3	VEC2	VEC1	VEC0

FF800F1B	VMEbus Interrupt Level					Byte	Write
7	6	5	4	3	2	1	0
					VIPL2	VIPL1	VIPL0

FF800F1D	VMEbus Standard Slave Address					Byte	Write
7	6	5	4	3	2	1	0
			VMEA23	VMEA22	VMEA21	VMEA20	VMEA19

FF800F1F	VMEbus Slave Address/Location Monitor Enable					Byte	Write
7	6	5	4	3	2	1	0
						LOCEN	SRAMEN

7.4 Board Configuration.

The 68360 automatically boots from CS0, which in the case of the BVME3000 selects the FLASH memory. The first task of the boot code is to configure the 68360 to the required hardware setting of the BVME3000, including address decoding and system function dedicated pin selection.

7.4.1 Module Base Register

The first task is to set the 68360,s Dual Port RAM base address, This is achieved by writing the address into the MBAR register.

MBAR Module Base Address Register: 0003FF00
Size: 32 Bit
Access: Read/Write

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BA31	BA30	BA29	BA28	BA27	BA26	BA25	BA24	BA23	BA22	BA21	BA20	BA19	BA18	BA17	BA16
1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BA15	BA14	BA13	NU	NU	NU	AS8	AS7	AS6	AS5	AS4	AS3	AS2	AS1	AS0	V
0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1

BA31 - BA13 Base Address

The Base Address bits for the MBAR, incremental in 8Kbytes block. This value should be set to $FF000000 to match the BVME3000's standard address map. These bits are cleared on Reset

AS8 - AS0 Address Space

The Address Space field bits allow particular address spaces to be masked, depending on the access type. An access to the 68360 with the wrong address space will cause an external cycle to occur and the block will not be accessed.

AS8 - Mask DMA address space (FC3-FC0 =1xxx) when set.
AS7 - Mask CPU address space (FC3-FC0 =0111) when set.
AS6 - Mask supervisor program address space (FC3-FC0 =0110) when set.
AS5 - Mask supervisor data address space (FC3-FC0 =0101) when set.
AS4 - Mask Motorola address space (FC3-FC0 = 0100) when set.
AS3 - Mask user reserved address space (FC3-FC0 =0011) when set.
AS2 - Mask program address space (FC3-FC0 =0010) when set.
AS1 - Mask user data address space (FC3-FC0 =0001) when set.
AS0 - Mask Motorola address space (FC3-FC0 =1xxx) when set.

The BVME3000 should be set to mask CPU address space, which prevents accesses to the DPR during an interrupt acknowledge cycle.

These bits are all cleared on reset.

V - Valid

This bit shows when the contents of the MBAR register are valid. When set the 68360's Dual Port Ram may be accessed, when clear the Dual Port Ram is not available.

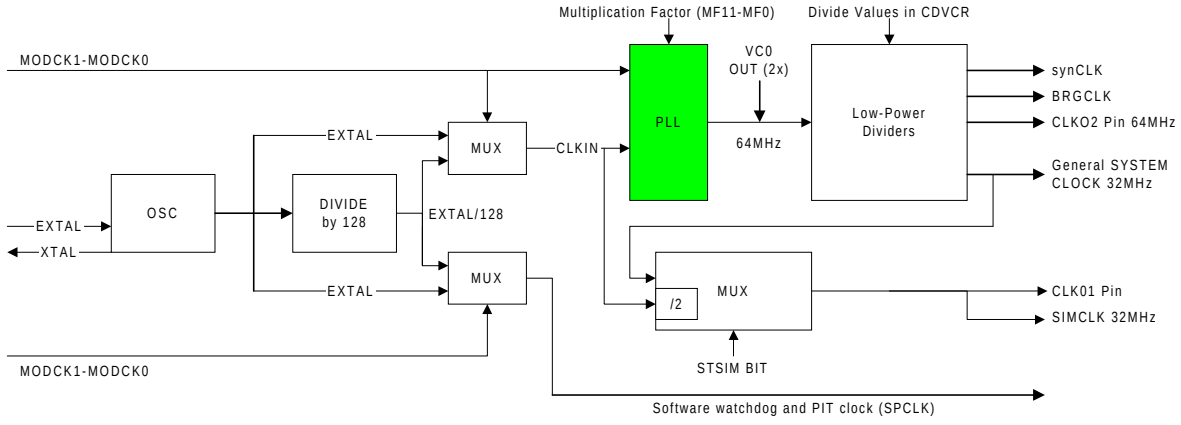
This bit is cleared on Reset.

The MBAR is located in CPU space, therefore before the write can take place, the function codes for the source and destination must be set to CPU space. Typical code is shown below.

```
move.l    #7,d1           ;function code for CPU data space
move.l    #5,d2           ;function code for supervisor data space
move.l    #$FF000101,d0   ;get new value for MBAR
lea       $3FF00,a0       ;get MBAR base address
movec     d1,DFC          ;set DFC to indicate CPU space
movec     d1,SFC          ;set SFC to indicate CPU space
moves.l   d0,(a0)         ;move new contents into MBAR
movec     d2,DFC          ;set DFC back to supervisor data space
movec     d2,SFC          ;set SFC back to supervisor data space
```

7.4.2 Clocking

The 68360 is fitted with a sophisticated PPL and clock system, as shown in the diagram below. The signals derived from this block are used to drive the baud rate generator, the watchdog and PIT clocks and other system clocks. The setting of these values is outside the scope of this manual due to the variations in user application's. For more information please refer to the 68360 User Manual.



The BVME3000 is configured with MODCLK1 = 0 and MODCLK0 = 1,giving the following configuration

PPL	-	ENABLED
128 Prescaler	-	DISABLED
Multi Factor (MF +1)	-	1
External Frequency	-	32MHz
CLKIN to the PLL	-	EXTAL
Internal Frequency	-	EXTAL

The MODCLK pin values can be changed however the above describes the reset value. Care must be taken when changing these values, because external logic blocks are driven from the clocks.

7.4.3 Bus Monitor

The BVME3000 supports BERR and this signal is required for operation with IndustryPacks™ and VME. The Bus Monitor function is set in the System Protection Control Register (SYPCR), and can only be changed with a supervisory access. The SYPCR also controls the watchdog settings, however this register can only be written to once following a reset, therefore care should be taken to ensure the watchdog settings match the system requirements.

SYPRC System Protection Control Register: FF001022
Size: 8 bit
Access: Read/Write

7	6	5	4	3	2	1	0
SWE	SWRI	SWR1	SWT0	DBFE	BME	BTM1	BTM0
X	X	X	X	1	1	0	0

SWE - Software Watchdog Enable

When set, software watchdog is enabled. This bit is **SET** at reset.

SWRI - Software Watchdog Reset/Interrupt Select

When clear, a watchdog timeout causes a level 7 interrupt to the CPU32+ core. When set, a watchdog timeout will cause a system reset. After reset this bit is SET.

SWR1 - SWR0

When used in conjunction with the PITR register, these bits control the determine the watchdog timeout period. These bits mirror MODCLK1 and will therefore be CLEAR after reset.

DBFE - Double Bus Fault Monitor enable

This bit should be set to enable double bus fault monitoring. A double bus error occurs when an address error or bus error occurs during exception processing and a halt signal will be asserted. The 68360 will respond to an internal Double bus error by driving System Reset. Following a reset this bit is SET.

BME - Bus Monitor External Enable

This bit should be set to enable bus error monitoring for external cycles. This bit is CLEAR following a reset.

BMT1 -BMT0 - Bus Monitor Timing

These bits select the timeout period for the bus error.

BMT1	BMT0	Bus Monitor Timeout Period
0	0	1K System Clocks (default)
0	1	512 System Clocks
1	0	256 System Clocks
1	1	128 System Clocks

Note:- system clock = CLKO1 @ 32MHz.

7.4.4 Port E Pin Assignment

A number of the control signals on the 68360 are multiplexed and have different functions depending on the environment they are to operate in. These pins are all assigned to the 68360's Port E. The following table provides the correct settings for the BVME3000.

PEPAR Port E Pin Assignment Register: FF001016
Size: 16 Bit
Access: Read/Write

15	14 13 12	11	10 9	8	7	6	5	4	3	2	1	0
NA	SINTOUT	NA	CF1MODE	IPIPE RASDD	A31-A28 WE0-WE3	OE AMUX	PWW	CAS2,3 IACK3,6	NA	CAS0,1 IACK1,2	CS7 IACK7	AVEC IACK5
70	0 0 0	0	1 0	0	0	0	1	0	0	0	0	0

SINTOUT - Slave Interrupt Out

These bits should never be set as they control the interrupt signals when the 68360 is configured as a slave. At reset all bits are clear.

CF1MODE - Configuration pin 1 mode

These bits control the multiplexed O/P of Config1 pin. The pin can be set as follows:-

00 = CONFIG1 pin function.
01 = CONFIG1 pin function.
10 = BCLRO output function.
11 = RAS2 drive pin function.

The BVME3000 uses the BCLRO function for its VME bus arbitration so the bits should be set accordingly. At reset these bits are clear.

IPIPE/RASDD - IPIPE/RASDD pin mode

When clear this pin is used as the IPIPE1 instruction on the BDM interface, when set acts as a double drive for RAS1. The BVME3000 fully supports BDM and therefore this bit should be left cleared. This bit is cleared on RESET.

A28 - A31/ WE0 - WE3 pin mode

When clear the pins are configured as dedicated address lines, when set the pins at as Memory Write strobes. The BVME3000 supports full 32-bit addressing, therefore this bit should be cleared. After reset these pins are tri-stated until they are enabled.

/OE - /AMUX pin mode

When clear this pin acts as a memory /OE, when set this pin acts as an address multiplexer select pin for DRAM multiplexing. The DRAM on the BVME3000 implements the internal address multiplexing, therefore this bit should be cleared to support the /OE function. After reset this bit is clear.

PWW

This is a read only bit and indicates when set that the A28-A31/WE lines have been set.

CAS2, CAS3 / IACK3, IACK6 pin mode

When clear these pins control the upper CAS lines to the DRAM, when set these pins are used as dedicated interrupt acknowledge lines for levels 3 and 6. The BVME3000 requires the full DRAM interface, therefore these pins should be configured as the upper CAS lines, bit cleared. After reset this bit is clear.

CAS0, CAS1 / IACK1, IACK2 pin mode

When clear these pins control the lower CAS lines to the DRAM, when set these pins are used as dedicated interrupt acknowledge lines for levels 1 and 2. The BVME3000 requires the full DRAM interface, therefore these pins should be configured as the lower CAS lines, bit cleared. After reset this bit is clear.

AVEC / IACK7 pin mode

When clear this pin acts as the AVEC (Autovectored interrupt) signal, when set this pin is used as dedicated interrupt acknowledge lines for levels 7. The BVME3000 uses Autovectored interrupts, therefore this bit should cleared. After reset this bit is clear.

7.4.5 Global Memory Register

The seven chip select lines on the 68360 have a high level of individal setting, however some settings are relevant for all memory locations such as DRAM refresh cycle times, DRAM size and parity settings. The more general settings are controlled in the Global Memory Register.

GMR Global Memory Register: FF001040
Size: 32 Bit
Access: Read/Write

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RCNT 7	RCNT 6	RCNT 5	RCNT 4	RCNT 3	RCNT 2	RCNT 1	RCNT 0	RFEN	RCYC 1	RCYC 0	PGS2	PBS1	PGS0	DPS1	DPS0
x	x	x	x	x	x	x	x	1	0	1	0	1	1	0	0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WBT 40	WBTQ	SYNC	EMWS	ORAR	PBEE	TSS 40	NCS	DWQ	DW4 0	GAMX	NU	NU	NU	NU	NU
0	1	0	1	0	0	0	1	0	1	0	1	0	0	0	0

RCNT7 - RCNT0 Refresh Counter Period

These bits determine the refresh period for the DRAM.

$$\text{Refresh Period} = \frac{\text{RFCNT} + 1}{\text{System clk}/16}$$

The system clock on the BVME3000 is 32Mhz, each bank of DRAM must be refreshed every 15.6µS therefore for one bank

RFCNT = 00011110

The refresh algorithm in the 68360 will refresh a DRAM bank every 15.6µS with the RFCNT value shown above, therefore if two banks are fitted the banks will take it in turns to be refreshed, with 15.6µS between each refresh cycle. Unfortunately this means each bank will get refreshed every 31.2µS, the timer value must therefore be divided further to reflect the number of banks of DRAM fitted.

The table below describes the recommend REFRESH values for the amount of DRAM fitted to the BVME3000:-

DRAM	Refresh Time	RFCNT value
4Mbyte	15.6µS	00011110
8Mbyte	7.8µS	00001111
12Mbyte	5.2µS	00001010
16Mbyte	3.9µS	00000111

At reset these bits are cleared.

RFEN - Refresh Enable

When set refresh cycles to the onboard DRAM are performed. If the BVME3000 is a DRAM variant, this bit should be set. After reset this bit is set.

RCYC1 - RCYC0 - Refresh Cycle Length

These bits determine the refresh cycle length. For the BVME3000, the RCYC bits should be set to 01, this gives a 6 clock refresh with RAS being negated for 5 phases before being asserted. After reset these bits are cleared.

PGS2 - PGS0 - Page Size

The DRAM controller on the BVME3000 can be set for vairous page or row/column sizes. The DRAM on the BVME3000 currently require the PS bits set to 010. This configures the controller for 512K DRAM. At reset these bits are cleared.

DPS1 - DPS0 _DRAM Port Size

These bits determine DRAM port size. The BVME3000 supports 32-bit port sizing.

SPS1	SPS0	Result
0	0	32-bit Port Size
0	1	16-bit Port Size
1	0	Reserved
1	1	External DSACKx Response

After reset these bits are cleared.

WBT40 - Wait between transfers (MC68EC040)

Not supported on the BVME3000.

WBTQ - Wait between transfers

When set this bit allows for a minimum negation time for /RAS of 6 phases, when clear /RAS precharge is reduced to 4. The DRAM used on the BVME3000 requires up to 6 phases (depending on cycle type) so this bit should be set. After reset this bit is clear.

DW40 - Delay write for 040(SRAM Bank only)

Not supported on the BVME3000.

EMWS - External Master wait states

Not supported on the BVME3000.

SYNC - Synchronous External Access MC68360 - Type

Not supported on the BVME3000.

OPAR - Odd Parity

Not supported on the BVME3000.

PBEE - Parity Bus Error Enable

When set a parity error will generate a BERR. The BVME3000 does not monitor parity, therefore this bit should be cleared. After reset this bit is clear.

TSS40 - /TS sample (MC68EC040 only)

Not supported on the BVME3000.

NCS - No CPU Space

When set CPU address space accesses will not be supported by the CS, when clear all accesses will be supported. The BVME3000 does not support CPU address space accesses to its memory peripherals and this should be set. After reset this bit is clear.

GAMUX

When set internal Address Multiplexing to the DRAM is enabled, when clear internal multiplexing is disabled. The BVME3000 requires address multiplexing for DRAM accesses so this bit should be set. After reset this bit is cleared.

7.4.6 Memory Configuration

Memory peripherals on the BVME3000 are controlled by the chip select lines on the 68360. Each chip select is multi-purpose and therefore must be configured to the memory or peripheral it is to control. Configuration is performed in the relevant Option register where device speed, type and size are set and the base register, where the device base address, read/write capabilities and specific CS pin tailoring is performed.

It is recommended that the option register is configure before the base register.

Option register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TCYC 3	TCYC 2	TCYC 1	TCYC 0	AM27	AM26	AM25	AM24	AM23	AM22	AM21	AM20	AM19	AM18	AM17	AM16
1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AM15	AM14	AM13	AM12	AM11	FCM3	FCM2	FCM1	FCM0	BCYC 1	BCYC 0	NU	PGME	SPS1	SPS0	DSEL
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Note the bottom row shows the Reset value.

TCYC3 - TCYC1 - Cycle Length in clocks

This field determines the number of wait states required for a memory cycle:-

	Internal Quicc Master Memory Bus Cycle length				
	Number of Clocks		Number of wait states (SRAM)		Number of wait states (DRAM)
TCYC =	Number	Comments	Number	Comments	Numbers
0	2	Fast Termination	-	Undifined	3
1	3	Normal	0		4
2	4		1		5
3	5		2		6
4	6		3		7
…	…		…		…
15	17		14		18

After reset TCYC defaults to 15.

AM27 - AM11 - Address Mask Field

This field will causes the masking of the corresponding bit in the Base Register. Any clear bits mask the corresponding address bit while any set bits cause the corresponding bit to tbe used in the address comparison.

If two chip selects are set to the same address, the lower chip will be assertd. E.g. CS0 has priority over CS1

At reset all bits are cleared allowing accesses to the full address range.

FCM3 - FCM0 - Function Code mask

This field determines address space type defined to this address line. This allows multiple chip selects at the same address but to different access types.

This register is cleared at reset to allow all access types, however an X111 on the function code pins will be taken as a CPU space access and the CS line will not be asserted.

BCYC1 - BCYC0 - Burst Length Cycle in Clocks

Not supported on the BVME3000.

PGME - Page mode Enable

This bit determines whether the page mode is enabled when the Chip select line is configured as a DRAM bank. The BVME3000 does not support DRAM Page Mode Accesses so this bit should always be CLEAR. PGME is always cleared on reset.

SPS1 - SPS0 SRAM Port Size

These bits determine the port size of the chip select line when it is configured as an SRAM type bank. DRAM banks are always assumed at 32-bit

SPS1	SPS0	Result
0	0	32-bit Port Size
0	1	16-bit Port Size
1	0	8-bit Port Size
1	1	External DSACKx Response

At reset these bits are set depending on the state of the CONFIG2-0 pins. The BVME3000 drives the pins to support a 16-bit port size at startup to support FLASH accesses.

DSSEL

This bit determines the chip select configuration. When set the chip select will perform as a DRAM bank thus CSx will act as a RAS line as the CAS pins will be enabled and address multiplexing will be performed. When clear the chip select will perform as an SRAM bank (SRAM, EPROM FLASH, peripheral etc). At reset this bit is cleared to support FLASH.

Base Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BA31	BA30	BA29	BA28	BA27	BA26	BA25	BA24	BA23	BA22	BA21	BA20	BA19	BA18	BA17	BA16
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BA15	BA14	BA13	BA12	BA11	FC3	FC2	FC1	FC0	TRLXQ	BACK40	CSNT40	CSNTQ	PAREN	WP	V
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BA31- BA11 - Base Address

This field contains the upper 21 bits of the peripherals base address which is compared with the address bus to determine if the peripheral is being accesses. At reset these bits are all cleared.

FC3 - FC0 Function Code
This field specifies that an access to this memory bank is limited to a certain access type. These bits should be programmed in conjunction with the FCM bits in the option register. At reset all bits are cleared to support all address types.

TRLXQ - Timing Relax

This bit when set causes the chip select assertion to be relaxed by one phase from the start of the cycle to aid slower peripherals. In the case of DRAM the assertion of CAS and address multiplexing is delayed by two phases (1 clock cycle). This bit is cleared at reset to give maximum cycle speed.

BACK40 - Burst acknowledge MC68EC040

Not supported on the BVME3000.

CSNT40 - /CS Negate timing MC68EC040 (SRAM Bank only)

Not supported on the BVME3000.

CSNT - /CS Negate timing (SRAM Bank only)

When set /CS is negated one phase early to help the peripheral meet the 68360s address/data hold times. At reset this bit is clear and /CS is negated at the end of the cycle.

PAREN - Parity Check Enable

When set parity checking of the memory is enabled. At reset this bit is clear and Parity checking is disabled.

WP - Write protection

When set write accesses to the memory are disabled and a BERR will occur if it is attempted. At Reset this bit is cleared and both read and write cycles are allowed.

V - Valid

When set the contents of the CS base register and option register is valid and the peripheral can be accessed. At reset this bit is cleared for all chip selects apart from CS0 which controls the boot memory.

7.5 FLASH

FLASH: 00000000
Size: 1M/2Mbytes
Width: 16 Bit
Access: Read/Write

The BVME3000 is factory fitted with either 1Mbyte or 2Mbytes of FLASH memory using 29F040 devices which are organised on a 16-bits wide bus. The memory is quite granular and is divided into 128Kbyte sectors for Programming purposes. The FLASH supports +5V only operation for read and write access and has an internal algorithm for erase and programming functions.

The first sector of FLASH memory must be used as the boot sector, and must be programmed before the BVME3000 will operate. The Boot sector can be programmed instigating the FLASH algorithm over the BDM interface.

7.5.1 FLASH Configuration

The FLASH is controlled by CS0 from the processor which at reset maps the FLASH to entire address range with a maximum internal DTACK. This will allow the BVME3000 to boot and run start up code, however the FLASH configuration must be altered to allow full module operation. FLASH configuration is controlled by the CS0 Base Register and Option Register, it is recommended that the Option Register is altered first.

OR0 Option Register 0: FF001054
Size: 32 Bit
Access: Read/Write

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TCYC 3	TCYC 2	TCYC 1	TCYC 0	AM27	AM26	AM25	AM24	AM23	AM22	AM21	AM20	AM19	AM18	AM17	AM16
0	0	1	0	1	1	1	1	1	1	1	0	0	0	0	0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AM15	AM14	AM13	AM12	AM11	FCM3	FCM2	FCM1	FCM0	BCYC 1	BCYC 0	NU	PGME	SPS1	SPS0	DSEL
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Cycle wait states: 2.
Mask Address: $FFE00000.
Function Code support: 0 All (not CPU Space).
Burst Cycle Length: 0 Not supported.

Page Mode: 0 Disabled.
SRAM Port size: 1 16-bit.
SRAM/DRAM: 0 SRAM.

BR0 Base Register 0: FF001050
Size: 32 Bit
Access: Read/Write

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BA31	BA30	BA29	BA28	BA27	BA26	BA25	BA24	BA23	BA22	BA21	BA20	BA19	BA18	BA17	BA16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BA15	BA14	BA13	BA12	BA11	FC3	FC2	FC1	FC0	TRLXQ	BACK40	CSNT40	CSNTQ	PARENN	WP	V
0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1

Base Address:	$00000000.	CS Negate timing:	1 release early.
Function Code Support:	0 Any.	Parity checking:	0 Disabled.
Timing Relax:	0 No.	Write Protect:	0 Disabled.
Burst Acknowledge 040:	0 Not supported.	Valid:	1 Yes (if fitted).
CS Negate timing 040:	0 Not supported.		

Note CS0 is configured for 2Mbytes, if only 1Mbyte is fitted, the memory will wrap around. For more information refer to “7.4.6 Memory Configuration”.

7.5.2 29F040 Erase/Program Operation

Command Definitions

The FLASH device operations are selected by writing specific commands into the user interface. The following table defines the FLASH memory device commands.

Sequence Command	Bus Write	First Bus Write Cycle		Second Bus Write Cycle		Third Bus Write Cycle		Forth Bus Read/Write Cycle		Fifth Bus Write Cycle		Sixth Bus Write Cycle	
Read/Reset	Cycles	Add	Da	Add	Da	Add	Da	Add	Da	Add	Da	Add	Da
Read/Reset	1	xxxx	F0										
Read/Reset	4	5555	AA	2AAA	55	5555	F0	RA	RD				
Autoselect	4	5555	AA	2AAA	55	5555	90	00	01				
								01	A4				
Byte Program	4	5555	AA	2AAA	55	5555	A0	PA	Data				
Chip Erase	6	5555	AA	2AAA	55	5555	80	5555	AA	2AAA	55	5555	10
Sector Erase	6	5555	AA	2AAA	55	5555	80	5555	AA	2AAA	55	SA	30
Sector Erase Suspend		Erase can be suspended during sector erase with Addr (don’t care), Data (B0)											
Sector Erase Resume		Erase can be resumed after suspend with Addr (don’t care), Data (30)											

Notes:

1. Address bits A15, A16, A17, and A18 = X = Don’t Care for all address commands except for Program Address (PA), Sector Address (SA), Read Address (RA), and autoselect sector protect verify.
2. RA = Address of the memory location to be read. PA = Address of the memory location to be programmed. Addresses are latched on the falling edge of the WE pulse. SA = Address of the sector to be erased. The combination of A18, A17, A16 will uniquely select any sector.
3. RD = Data read from location RA during read operation. PD = Data to be programmed at location PA. Data is latched on the rising edge of WE.
4. Read from non-erasing sectors is allowed in the Erase Suspend mode.
5. All address and data values are in hexadecimal.
6. AutoSelect is not supported on the BVME3000.

Read/Reset Command

The read or reset operation is initiated by writing the read/reset command sequence into the command register. Microprocessor read cycles retrieve array data from the memory. The device remains enabled for reads until the command register contents are altered.

The device will automatically power-up in the read/ reset state. In this case, a command sequence is not required to read data. Standard microprocessor read cycles will retrieve array data. This default value ensures that no spurious alteration of the memory content occurs during the power transition.

Byte Programming

The device is programmed on a byte-by-byte basis. Programming is a four bus cycle operation. There are two “unlock” write cycles. These are followed by the program setup command and data write cycles. Upon executing the Embedded Program Algorithm command sequence the system is not required to provide further controls or timings. The device will automatically provide adequate internally generated program pulses and verify the programmed cell margin.

The automatic programming operation is completed when the data on DQ7 is equivalent to data written to this bit (see Write Operation Status section) at which time the device returns to the read mode and addresses are no longer latched. Therefore, the device requires that a valid address to the device be supplied by the system at this particular instance of time. Hence, Data Polling must be performed at the memory location which is being programmed.

Any commands written to the chip during this period will be ignored.

Programming is allowed in any sequence and across sector boundaries. Beware that a data “0” cannot be programmed back to a “1”. Attempting to do so may cause the device to exceed programming time limits (DQ5 = 1) or result in an apparent success, according to the data polling algorithm, but a read from reset/read mode will show that the data is still “0”. Only erase operations can convert “0”s to “1”s.

Embedded Programming Algorithm using typical command strings and bus operations, see Embedded programming Algorithm flow chart.

Chip Erase

Chip erase is a six bus cycle operation. There are two “unlock” write cycles. These are followed by writing the “set-up” command. Two more “unlock” write cycles are then followed by the chip erase command.

Chip erase does not require the user to program the device prior to erase. Upon executing the Embedded Erase Algorithm command sequence the device automatically will program and verify the entire memory for an all zero data pattern prior to electrical erase. The chip erase is performed sequentially one sector at a time. The system is not required to provide any controls or timings during these operations.

The automatic erase begins on the rising edge of the last WE pulse in the command sequence and terminates when the data on DQ7 is “1” (see Write Operation Status section) at which time the device returns to read the mode.

Embedded Erase Algorithm using typical command strings and bus operations, see Embedded Erase Algorithm flow chart.

Sector Erase

Sector erase is a six bus cycle operation. There are two “unlock” write cycles. These are followed by writing the “set-up” command. Two more “unlock” write cycles are then followed by the sector erase command. The sector address (any address location within the desired sector) is latched on the falling edge of WE, while the command (data) is latched on the rising edge of WE. A time-out of 80 µs from the rising edge of the last sector erase command will initiate the sector erase command(s).

Multiple sectors may be erased concurrently by writing the six bus cycle operations as described above. This sequence is followed with writes of the Sector Erase command to addresses in other sectors desired to be concurrently erased. The time between writes must be less than 80 µs, otherwise that command will not be accepted. It is recommended that processor interrupts be disabled during this time to guarantee this condition. The interrupts can be re-enabled after the last Sector Erase command is written. A time-out of 80 µs from the rising edge of the last WE will initiate the execution of the Sector Erase command(s). If another falling edge of the WE occurs within the 80 µs time-out window the timer is reset. (Monitor DQ3 to determine if the sector erase window is still open, see section DQ3, Sector Erase Timer.) Any command other than Sector Erase or Erase Suspend during this period resets the device to read mode, ignoring the previous command string. In that case, restart the erase on those sectors and allow them to complete. (Refer to the Write Operation Status section for Sector Erase Timer operation.) Loading the sector erase buffer may be done in any sequence and with any number of sectors (1 to 8).

Sector erase does not require the user to program the device prior to erase. The device automatically programs all memory locations in the sector(s) to be erased prior to electrical erase. When erasing a sector or sectors the remaining unselected sectors are not affected. The system is not required to provide any controls or timings during these operations.

The automatic sector erase begins after the 80 µs time out from the rising edge of the WE pulse for the last sector erase command pulse and terminates when the data on DQ7 is “1" (see Write Operation Status section) at which time the device returns to read mode. During the execution of the Sector Erase command, only the Erase Suspend and Erase Resume commands are allowed. All other commands will be ignored. Data polling must be performed at an address within any of the sectors being erased.

Embedded Erase Algorithm uses typical command strings and bus operation, see Embedded Erase Algorithm flow chart.

Erase Suspend

The Erase Suspend command allows the user to interrupt a Sector Erase operation and then perform data reads from a sector not being erased. This command is applicable ONLY during the Sector Erase operation which includes the time-out period for sector erase. The Erase Suspend command will be ignored if written during the Chip Erase operation or Embedded Program Algorithm. Writing the Erase Suspend command during the Sector Erase time-out results in immediate termination of the time-out period and suspension of the erase operation.

Any other command written during the Erase Suspend mode will be ignored except the Erase Resume command. Writing the Erase Resume command resumes the erase operation. The addresses are “don’t-cares” when writing the Erase Suspend or Erase Resume command. When the Erase Suspend command is written during the Sector Erase operation, the device will take a maximum of 15 µs to suspend the erase operation. When the device has entered the erase-suspended mode, DQ7 bit will be at logic “1”, and DQ6 will stop toggling. The user must use the address of the erasing sector for reading DQ6 and DQ7 to determine if the erase operation has been suspended. Further writes of the Erase Suspend command are ignored.

When the erase operation has been suspended, the device defaults to the erase-suspend-read mode. Reading data in this mode is the same as reading from the standard read mode except that the data must be read from sectors that have not been erase-suspended.

To resume the operation of Sector Erase, the Resume command (30H) should be written. Any further writes of the Resume command at this point will be ignored. Another Erase Suspend command can be written after the chip has resumed erasing.

Write Operation Status

	Status		DQ7	DQ6	DQ5	DQ3
In Progress	Byte Programming		/DQ7	Toggle	0	0
	Program/Erase in Auto-Erase		0	Toggle	0	1
	Erase	Erase suspended sector	1	No Tog	0	1
	Suspend Mode	Non-erase suspended sector	Data	Data	Data	Data
Exceeded Time Limits	Byte Programming in Embedded Algorithm		/DQ7	Toggle	1	0
	Embedded Erase Algorithm		0	Toggle	1	1
	Program in erase suspended Mode		/DQ7	Toggle	1	1

DQ7 - Data Polling

The 29F040 device features Data Polling as a method to indicate to the host that the Embedded Algorithms are in progress or completed. During the Embedded Program Algorithm an attempt to read the device produces the compliment of the data last written to DQ7. Upon completion of the Embedded Program Algorithm, reading the device produces the true data last written to DQ7. During the Embedded Erase Algorithm, reading the device produces a “0” at the DQ7 output. Upon completion of the Embedded Erase Algorithm, reading the device produces a “1” at the DQ7 output. Please refer to the flowchart for Data Polling

For chip erase, the Data Polling is valid after the rising edge of the sixth WE pulse in the six write pulse sequence. For sector erase, the Data Polling is valid after the last rising edge of the sector erase WE pulse. Data Polling must be performed at sector address within any of the sectors being erased and not a protected sector. Otherwise, the status may not be valid. Once the Embedded Algorithm operation is close to being completed, the 29F040 data pins (DQ7) may change asynchronously while the output enable (OE) is asserted low. This means that the device is driving status information on DQ7 at one instant of time and then that byte’s valid data at the next instant of time. Depending on when the system samples the DQ7 output, it may read the status or valid data. Even if the device has completed the Embedded Algorithm operation and DQ7 has a valid data, the data outputs on DQ0–DQ6 may be still invalid. The valid data on DQ0–DQ7 will be read on the successive read attempts.

The Data Polling feature is active during the Embedded Programming Algorithm, Embedded Erase Algorithm, Erase Suspend, or sector erase time-out (see the table above).

DQ6 - Toggle Bit

The 29F040 also features the “Toggle Bit” as a method to indicate to the host system that the Embedded Algorithms are in progress or completed.

During an Embedded Program or Erase Algorithm cycle, successive attempts to read (OE toggling) data from the device will result in DQ6 toggling between one and zero. Once the Embedded Program or Erase Algorithm cycle is completed, DQ6 will stop toggling and valid data will be read on the next successive attempts. During programming, the Toggle Bit is valid after the rising edge of the fourth WE pulse in the four write pulse sequence. For chip erase, the Toggle Bit is valid after the rising edge of the sixth WE pulse in the six write pulse sequence. For Sector erase, the Toggle Bit is valid after the last rising edge of the sector erase WE pulse. The Toggle Bit is active during the sector time out.

In programming, if the sector being written to is protected, the toggle bit will toggle for about 2 µs and then stop toggling without the data having changed. In erase, the device will erase all the selected sectors except for the ones that are protected. If all selected sectors are protected, the chip will toggle the toggle bit for about 100 µs and then drop back into read mode, having changed none of the data. Either CE or OE toggling will cause the DQ6 to toggle.

DQ5 - Exceeded Timing Limits

DQ5 will indicate if the program or erase time has exceeded the specified limits (internal pulse count). Under these conditions DQ5 will produce a "1". This is a failure condition which indicates that the program or erase cycle was not successfully completed. Data Polling is the only operating function of the device under this condition. The CE circuit will partially power down the device under these conditions (to approximately 2 mA).

If this failure condition occurs during sector erase operation, it specifies that a particular sector is bad and it may not be reused, however, other sectors are still functional and may be used for the program or erase operation. The device must be reset to use other sectors. Write the Reset command sequence to the device, and then execute program or erase command sequence. This allows the system to continue to use the other active sectors in the device. If this failure condition occurs during the chip erase operation, it specifies that the entire chip is bad or combination of sectors are bad.

If this failure condition occurs during the byte programming operation, it specifies that the entire sector containing that byte is bad and this sector may not be reused, (other sectors are still functional and can be reused).

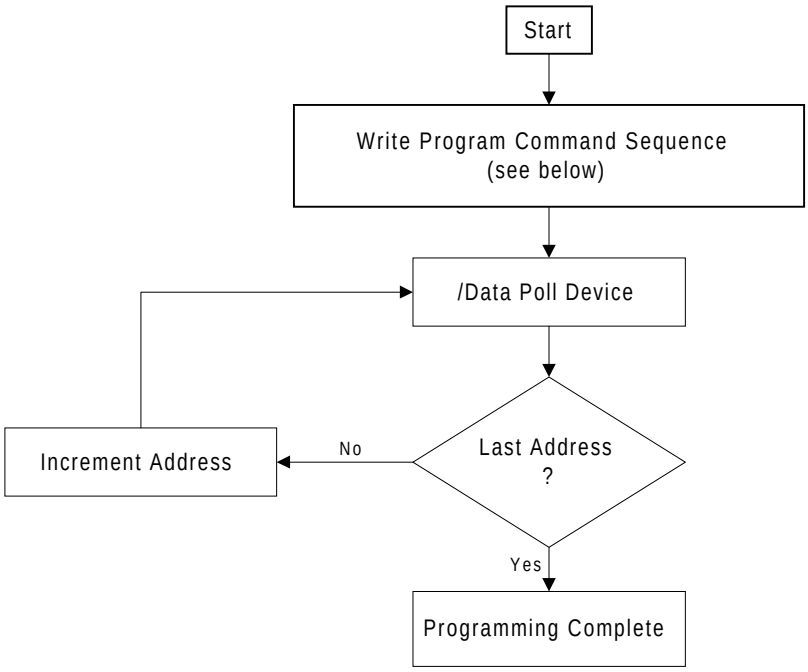
The DQ5 failure condition may also appear if a user tries to program a "1" to a location previously programmed to "0". In this case the device locks out and never completes the Embedded Algorithm operation. Hence, the system never reads a valid data on DQ7 bit and DQ6 never stops toggling. Once the device has exceeded timing limits, the DQ5 bit will indicate a "1". Please note that this is not a device failure condition since the device was incorrectly used.

DQ3 - Sector Erase Timer

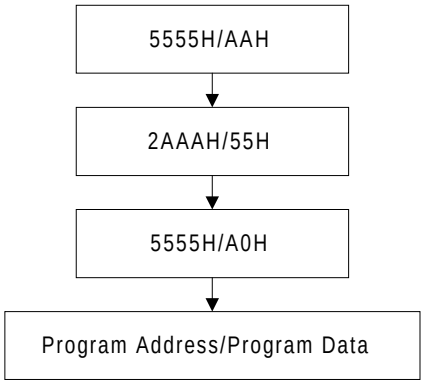
After the completion of the initial sector erase command sequence the sector erase time-out will begin. DQ3 will remain low until the time-out is complete. Data Polling and Toggle Bit are valid after the initial sector erase command sequence.

If Data Polling or the Toggle Bit indicates the device has been written with a valid erase command, DQ3 may be used to determine if the sector erase timer window is still open. If DQ3 is high ("1") the internally controlled erase cycle has begun; attempts to write subsequent commands to the device will be ignored until the erase operation is completed as indicated by Data Polling or Toggle Bit. If DQ3 is low ("0"), the device will accept additional sector erase commands. To insure the command has been accepted, the system software should check the status of DQ3 prior to and following each subsequent sector erase command. If DQ3 is high on the second status check, the command may not have been accepted.

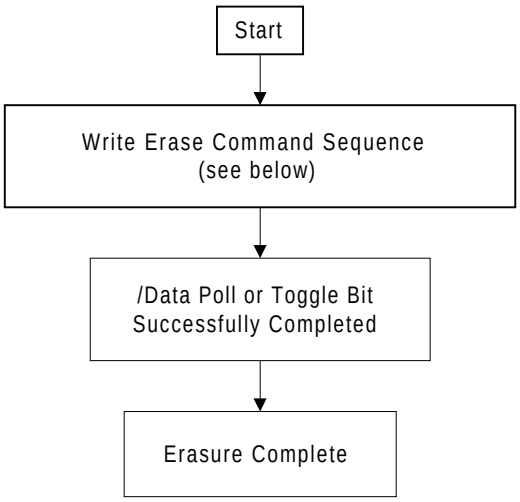
Refer to the Write Operation Status table.



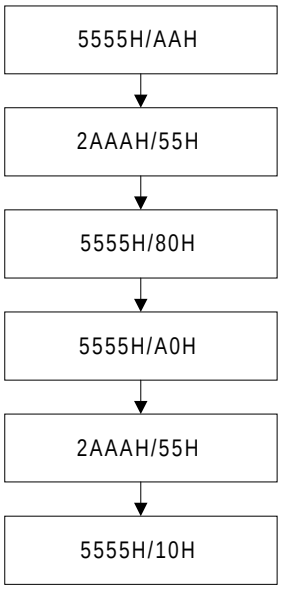
Program Command Sequence (Address/Command):



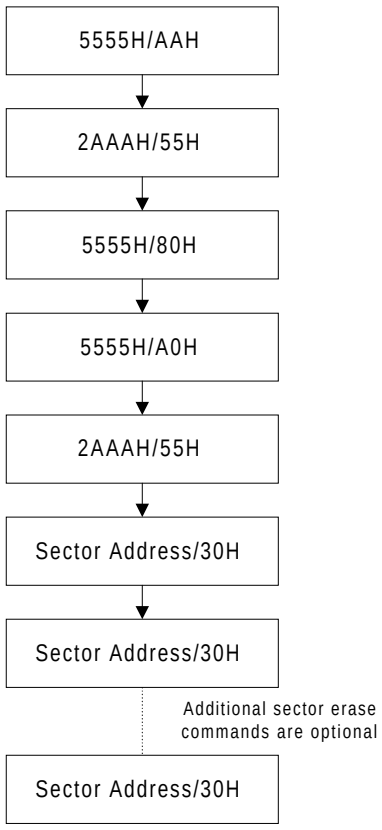
Embedded Programming Algorithm



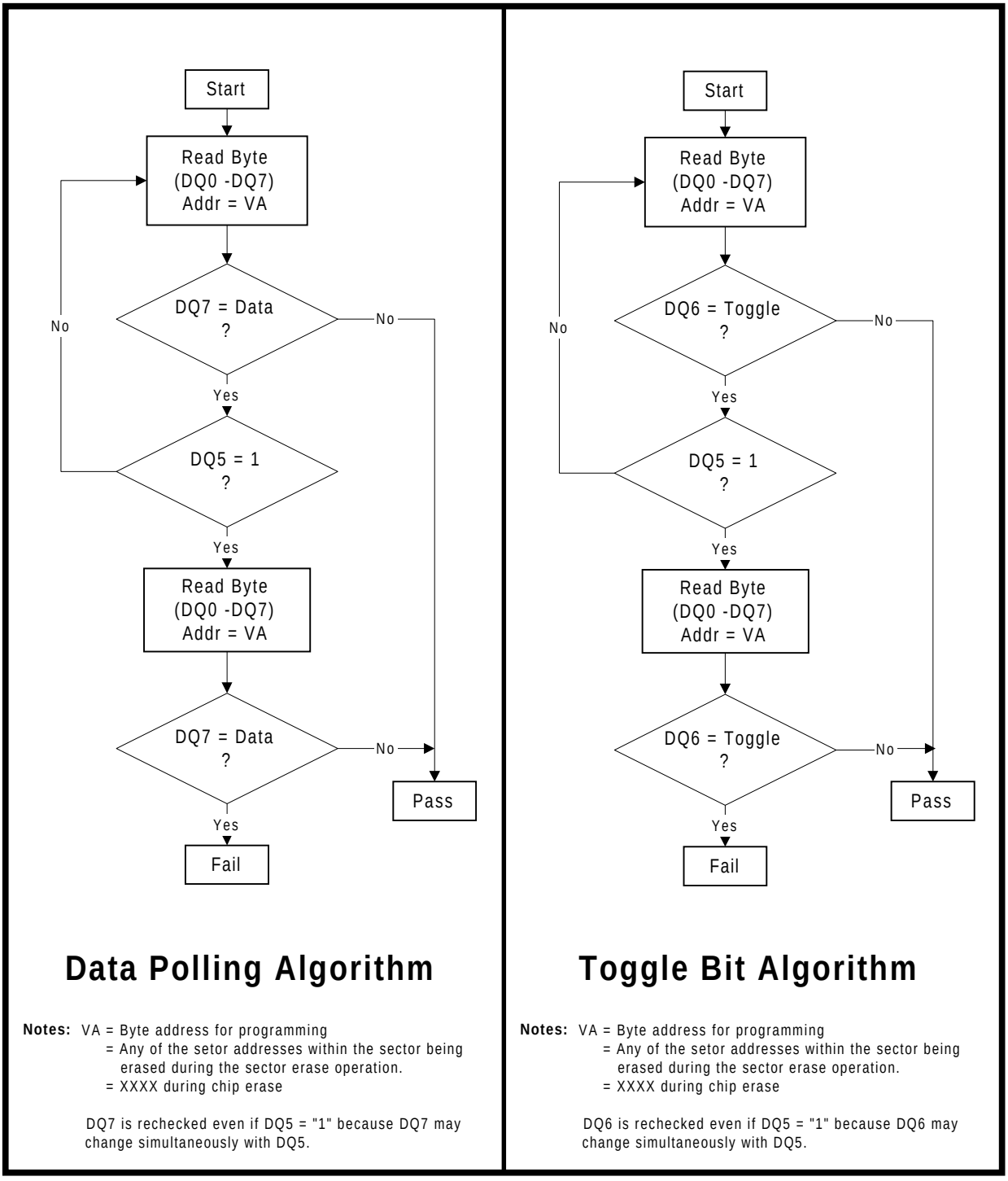
Chip Erase Command Sequence (Address/Command):



Individual Sector/Multiple Sector Erase Command Sequence (Address Command



Embedded Erase Algorithm



For more information on FLASH programming please see the AM29F040 available in the BVME3000 Engineering Kit.

7.6 SRAM

SRAM: 00200000
Size: 512Kbytes
Width: 32 Bit
Access: Read/Write

The BVME3000 can be fitted with 512Kbytes of SRAM, accessed as a 32-bit block. The SRAM is dual ported and can be accessed in the VMEbus standard address space, the base address can be programmed anywhere within that space on 512 Kbyte boundaries. The SRAM is backed up by a memory storage capacitor, and typically is used for non-volatile storage applications such as RAM-disc. The SRAM can retain it's data for up to 7 days

The SRAM is controlled by CS5, its specific configuration is programmed into the CS6 base register and options register as below.

OR5 Option Register 5: FF0010A4
Size: 32 Bit
Access: Read/Write

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TCYC 3	TCYC 2	TCYC 1	TCYC 0	AM27	AM26	AM25	AM24	AM23	AM22	AM21	AM20	AM19	AM18	AM17	AM16
0	0	1	0	1	1	1	1	1	1	1	1	1	0	0	0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AM15	AM14	AM13	AM12	AM11	FCM3	FCM2	FCM1	FCM0	BCYC 1	BCYC 0	NU	PGME	SPS1	SPS0	DSEL
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Cycle wait states: 2.
Mask Address: $FFF80000.
Function Code support: 0 All (not CPU Space).
Burst Cycle Length: 0 Not supported.

Page Mode: 0 Disabled.
SRAM Port size: 0 32-bit.
SRAM/DRAM: 0 SRAM.

BR5 Base Register 5: FF0010A0
Size: 32 Bit
Access: Read/Write

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BA31	BA30	BA29	BA28	BA27	BA26	BA25	BA24	BA23	BA22	BA21	BA20	BA19	BA18	BA17	BA16
0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BA15	BA14	BA13	BA12	BA11	FC3	FC2	FC1	FC0	TRLX Q	BACK 40	CSNT 40	CSNT Q	PARE N	WP	V
0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1

Base Address: $00200000.
Function Code Support: 0 Any.

CS Negate timing: 1 release early.
Parity checking: 0 Disabled.

Timing Relax:	0 No.	Write Protect:	0 Disabled.
Burst Acknowledge 040:	0 Not supported.	Valid:	1 Yes (if fitted).
CS Negate timing 040:	0 Not supported.		

For more information refer to “7.4.6 Memory Configuration”.

7.7 DRAM

DRAM:	01000000
Size:	4/8/12/16Mbytes
Width:	32 Bit
Access:	Read/Write

The BVME3000 can be fitted with 4/8/12/16Mbyte of DRAM accessed in 32-bit blocks. Each bank of 4Mbytes is controlled by a separate chip select line, if a bank is not fitted it should be disabled to prevent DRAM housekeeping tasks from being performed to an unused bank and waste resources.

DRAM is controlled by CS1, CS2 CS3 and CS4 each controlling a separate bank, CS1 controls bank 1, CS2 controls bank 2 and so on. The following tables show the DRAM configuration, the option register are identical for all banks. The base register differs for each bank and the bottom four rows of the table show the settings for CS1, CS2, CS3 and CS4 in order.

OR1 Option Register 1:	FF001064
OR2 Option Register 2:	FF001074
OR3 Option Register 3:	FF001084
OR4 Option Register 4:	FF001094
Size:	32 Bit
Access:	Read/Write

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TCYC 3	TCYC 2	TCYC 1	TCYC 0	AM27	AM26	AM25	AM24	AM23	AM22	AM21	AM20	AM19	AM18	AM17	AM16
0	0	0	1	1	1	1	1	1	1	0	0	0	0	0	0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AM15	AM14	AM13	AM12	AM11	FCM3	FCM2	FCM1	FCM0	BCYC 1	BCYC 0	NU	PGME	SPS1	SPS0	DSEL
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Cycle wait states:	1.	Page Mode:	0 Disabled.
Mask Address:	$FFC00000.	SRAM Port size:	0 32-bit.
Function Code support:	0 All (not CPU Space).	SRAM/DRAM:	1 DRAM.
Burst Cycle Length:	0 Not supported.		

BR1 Base Register 1: FF001060
BR2 Base Register 2: FF001070
BR3 Base Register 3: FF001080
BR4 Base Register 4: FF001090
Size: 32 Bit
Access: Read/Write

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BA31	BA30	BA29	BA28	BA27	BA26	BA25	BA24	BA23	BA22	BA21	BA20	BA19	BA18	BA17	BA16
0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0
0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BA15	BA14	BA13	BA12	BA11	FC3	FC2	FC1	FC0	TRLXQ	BACK40	CSNT40	CSNTQ	PAREN	WP	V
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

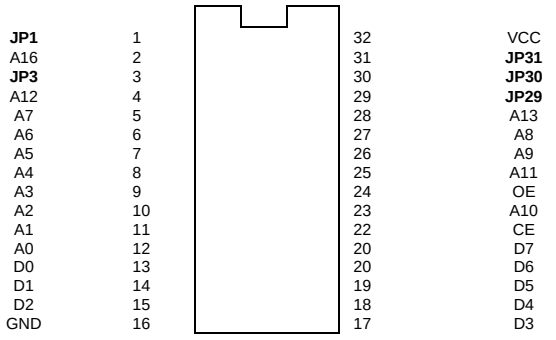
CS1 Base Address: $01000000.
CS2 Base Address: $01400000.
CS3 Base Address: $01800000.
CS4 Base Address: $01C00000.
Function Code Support: 0 Any.
Timing Relax: 0 No.

Burst Acknowledge 040: 0 Not supported.
CS Negate timing 040: 0 Not supported.
CS Negate timing: 0 Not supported.
Parity checking: 0 Disabled.
Write Protect: 0 Disabled.
Valid: 1 Yes (if fitted).

For more information refer to “7.4.6 Memory Configuration”.

7.8 Expansion Memory

The BVME3000 is fitted with four JEDEC compatible sockets. Each socket can be fitted with either SRAM, FLASH or EPROM with size ranging from 128K x 8 to 1024K x 8.



The diagram shows that a number of pins have no dedicated function, these are configured in a PLD as one of the following functions :-

A15 A16 A17 A18 A19 A20 A21 WE

The default PLD is configured to 128K x 8 SRAM, please contact your supplier for details of other supported parts.

Like the other memory, the Expansion memory is controlled by a chip select from the 68360, in this case CS6. The option register and base register settings below enable the entire Expansion memory address space, and assume 70nS devices or faster.

OR6 Option Register 6: FF0010B4
Size: 32 Bit
Access: Read/Write

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TCYC 3	TCYC 2	TCYC 1	TCYC 0	AM27	AM26	AM25	AM24	AM23	AM22	AM21	AM20	AM19	AM18	AM17	AM16
0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AM15	AM14	AM13	AM12	AM11	FCM3	FCM2	FCM1	FCM0	BCYC 1	BCYC 0	NU	PGME	SPS1	SPS0	DSEL
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Cycle wait states: 3.
Mask Address: $FFC00000.
Function Code support: 0 All (not CPU Space).
Burst Cycle Length: 0 Not supported.

Page Mode: 0 Disabled.
SRAM Port size: 0 32-bit.
SRAM/DRAM: 0 SRAM.

BR6 Base Register 6: FF0010B0
Size: 32 Bit
Access: Read/Write

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BA31	BA30	BA29	BA28	BA27	BA26	BA25	BA24	BA23	BA22	BA21	BA20	BA19	BA18	BA17	BA16
0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BA15	BA14	BA13	BA12	BA11	FC3	FC2	FC1	FC0	TRLXQ	BACK 40	CSNT 40	CSNTQ	PAREN	WP	V
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Base Address: $00400000.
Function Code Support: 0 Any.
Timing Relax: 0 No.
Burst Acknowledge 040: 0 Not supported.
CS Negate timing 040: 0 Not supported.

CS Negate timing: 0 at end.
Parity checking: 0 Disabled.
Write Protect: 0 Disabled.
Valid: 1 Yes (if fitted).

For more information refer to “7.4.6 Memory Configuration”.

7.8.1 Slow Expansion memory (<70nS)

Expansion memory control is designed to support 70nS or faster devices, however it is possible to reconfigure the software for slower devices, in this case the TCYC set in the option register must be increased to 4 and the “Expansion Memory wait state board control register” must be configured.

Expansion Memory wait state board Control Register: FF800F0D

Size: 8 bit
Access: Write

7	6	5	4	3	2	1	0
							EXP

EXP - Expansion Memory Wait State

When set the BVME3000 supports Expansion Memory up to 100nS, when clear the BVME3000 supports Expansion memory up to 70nS. After reset this register is clear.

7.9 EEPROM

The BVME3000 is fitted with an X25020 256 byte EEPROM, which is controlled using the SPI interface on the 68360. The device can be write protected either in software where ¼ or ½ or the entire contents can be protected or by fitting LK1 which protects the entire device.

The following section covers the required register settings for both the 68360 and X25020 to achieve hardware configuration for successful data transfer. For specific information on the actual data transfer please see the MC68360 User Manual and X25020 data sheets available inn the BVME3000 engineering kit.

7.9.1 SPI configuration

The 68360 includes a full master/slave SPI controller, which must be configured to operate with the X25020. A number of registers control the interface, while a separate set of registers must be configured to enable the pins for SPI operation.

Pin configuration

The SPI interface requires 4 dedicated pins SI, SO SCK and /SPICS. The 68360 can provide all 4 however, the /SPICS pin on the BVME3000 is derived in logic to leave the dedicated pin on the 68360 free for any further SPI devices which the user may connected to the 68360 I/O socket. The /SPICS pin is controlled from the “EPPROM Chip Select board control register”.

EEPROM Chip Select board Control Register: FF800F0F
Size: 8 bit
Access: Write

7	6	5	4	3	2	1	0
							SPICS

SPICS - EEPROM Chip Select

When set the EEPROM is chip selected, when clear the EEPROM is deselected. After reset the bit is clear. During EEPROM access this select may be repeatedly switched on and off depending on the cycle type. There is no need to allow settle wait states in software as the select will happen within 15nS of writing to this register.

The SI SO and SCK made available from the 68360’s port B, the pin direction, type and level must all be configured before operation. These functions can be selected in the Port B registers.

BPPAR Port B Pin Assignment register: FF0016BC
Size: 18 Bit
Access: Read/Write

17	16	15	14	13	12	11	10	9	8	5	6	5	4	3	2	1	0
DD 17	DD 16	DD15	DD14	DD13	DD12	DD11	DD10	DD9	DD8	DD7	DD6	DD5	DD4	DD3	DD2	DD1	DD0
X	X	X	X	X	X	X	X	X	X	X	X	X	X	1	1	1	X

When clear the Port B pin is a general purpose I/O pin. When set the port B pin has a dedicated peripheral function. The function is determined in the PBDIR register. After reset all bits are cleared, the BVME3000 requires Bit1 - 3 set for SPI operation.

BPDIR Port B Data Direction Register: FF0016B8
Size: 18 Bit
Access: Read/Write

17	16	15	14	13	12	11	10	9	8	5	6	5	4	3	2	1	0
DR17	DR16	DR15	DR14	DR13	DR12	DR11	DR10	DR9	DR8	DR7	DR6	DR5	DR4	DR3	DR2	DR1	DR0
X	X	X	X	X	X	X	X	X	X	X	X	X	X	1	1	1	X

The operation of this register is dependant on the status of the relevant PBPAR bit if the PBPAR bit is clear this register controls the direction of the general purpose I/O bit. If the PBPAR bit is set the PBDIR register determines which dedicated function is assigned to the Pin. The BVME3000 requires bit1 - 3 to be set to enable the SPI functions to these pins

BPODR Port B Open Drain Register: FF0016C2
Size: 16 Bit
Access: Read/Write

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OD15	OD14	OD13	OD12	OD11	OD10	OD9	OD8	OD7	OD6	OD5	OD4	OD3	OD2	OD1	OD0
X	X	X	X	X	X	X	X	X	X	X	X	0	0	0	X

When set the relevant pin acts as an open drain drive, when clear the pin acts as an Open Drain driven O/P. The SPI function require these bits to be clear to give continuous O/P drive because this register will even effect dedicated output pins. After reset these pins are cleared and should remain so.

It is only necessary to program Bits1 - 3 in the port B registers as these bits contain the dedicated SPI pins:-

Bit 3 SPIMISO when the 68360 is a master pin receives SPI data, when the 68360 is a slave this pin transmits SPI data.

Bit 2 SPIMOSI when the 68360 is a master pin transmits SPI data, when the 68360 is a slave this pin receives SPI data.

Bit 3 SPICLK when the 68360 is a master pin transmits the SPI clock, when the 68360 is a slave this pin receives the SPI clock.

If the SPI interface pins are required for a different task, the X25020 must be removed.

SPI Mode Configuration

Although a common standard the 68360 must be configured to match the EEPROM's interface, issues such as clock speed, clock phase and character length must be configured. The SPI interface is configured in the SPI Mode Register.

SPMODE SPI Mode Register: FF0016A0
Size: 16 Bit
Access: Read/Write

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU	LOOP	CI	CP	DIV16	REV	M/S	EN	LEN				PM3	PM2	PM1	PM0
0	0	0	0	0	1	1	1	0	1	1	1	1	1	1	1

LOOP - Loop Select

When set the SPI will loop back the transmitted data to the receiver, when clear the interface runs normally. The BVME3000 does not use loop mode and this bit should be clear. After reset this bit is clear.

CI - Clock Invert

When set the inactive state of the SPI clock is high, when set the inactive state of the SPI clock is low. The X25020 latches data on the positive edge of the clock so this bit should be clear. After reset this bit is clear.

CP - Clock phase

When set the Clock begins toggling in the middle of the data transfer, when clear the clock is toggled at the beginning of the Data transfer. The X25020 requires the clock to toggled at the beginning of the data transfer so this bit should be clear. After reset this bit is clear.

DIV16 - Divide by 16

When set the 68360 BRGCLK/16 clock is used to derive the clock, when clear the full BRGCLK is used. The value of this bit depends on the users BRGCLK derision, but assuming BVM's standard settings this bit is cleared. At reset this bit is cleared.

REV - Reverse Data

When set the MSB is transmitted or received first, when clear the LSB is transmitted or received first. The X25020 transmits and receives the MSB first so this bit should be set. After Reset bit is clear.

M/S - Master Slave

When clear the SPI interface on the 68360 is configured as a slave, when set the SPI interface on the 68360 is configured as a master. This bit should be set to allow the 68360 to act as the master. After reset this bit is clear

EN -Enable SPI

When set the SPI interface is enabled, when clear this SPI interface is disabled. This bit should be set to enable the SPI interface. After reset this bit is cleared.

LEN - Character Length

The length field is a binary representation of the SPI interface character length, with 0000 = 1 bit and 1111 = 16 bits. The X25020 supports 8 bit characters so this field should be set to 0111. After reset the bits are clear.

PM3 - PM0 Prescale Modulus Select

These bits specify the divide ratio of the prescale divider in the SPI clock generator. The BGRCLK is divided by 4* ([PM3 - PM0] - 1) giving a clock ratio 4 to 64. The value of this bit depends on the users BRGCLK source, but assuming BVM's standard settings these bits should all be set to divide by 64. At reset this bit is cleared.

Operation

Data is passed to and from the X25020 via the 68360's parameter RAM. As the master the 68360 must always transmit an instruction to the X25020 via its transmit register, and await any response from the X25020 in it receive buffers. When the 68360 is transmitting a command the X25020 is in an unknown state and will clock out random data as it receives the command. The random data will enter the receive buffer and should be ignored. As most commands are two cycles long (command + address) there are generally two bytes of random data returned.

Data transmissions commence once the data has been set in the transmit buffer and the start bit is set in the SPI command register. For more information on data transfer please refer to the MC68360 User Manual and X25020 data sheets available inn the BVME3000 engineering kit.

7.9.2 EEPROM Operation

The X25020 contains an 8-bit instruction register. It is accessed via the SI input, with data being clocked in on the rising SCK. CS must be LOW and the HOLD and WP inputs must be HIGH during the entire operation.

The following contains a list of the instructions and their opcodes. All instructions, addresses and data are transferred MSB first.

Instruction Name	Instruction Format*	Operation
WREN	0000 0110	Set the Write Enable Latch (Enable Write Operations)
WRDI	0000 0100	Reset the Write Enable Latch (Disable Write Operations)
RDSR	0000 0101	Read Status Register
WRSR	0000 0001	Write Status Register
READ	0000 0011	Read Data from Memory Array beginning at Selected Address
WRITE	0000 0010	Write Data to Memory Array beginning at Selected Address (1 to 32)

*Instructions are shown MSB in leftmost position. Instructions are transferred MSB first.

Data input is sampled on the first rising edge of SCK after CS goes LOW. SCK is static, allowing the user to stop the clock and then resume operations.

Status Register

The RDSR instruction provides access to the status register. The status register may be read at any time, even during a write cycle. The status register is formatted as follows:

7	6	5	4	3	2	1	0
X	X	X	X	BP1	BP0	WEL	WIP

BP0 and BP1 are set by the WRSR instruction. WEL and WIP are read-only and automatically set by other operations.

The Write-In-Process (WIP) bit indicates whether the X25020 is busy with a write operation. When set to a “1”, a write is in progress, when set to a “0”, no write is in progress. During a write, all other bits are set to “1”.

The Write Enable Latch (WEL) bit indicates the status of the “write enable” latch. When set to a “1”, the latch is set, when set to a “0”, the latch is reset.

The Block Protect (BP0 and BP1) bits are non-volatile and allow the user to select one of four levels of protection. The X25020 is divided into four 512-bit segments. One, two, or all four of the segments may be protected. That is, the user may read the segments but will be unable to alter (write) data within the selected segments. The partitioning is controlled as illustrated below.

Status Register Bits		Array Addresses Protected
BP1	BP0	
0	0	None
0	1	$C0–$FF
1	0	$80–$FF
1	1	$00–$FF

Write Enable Latch

The X25020 contains a “write enable” latch. This latch must be SET before a write operation will be completed internally. The WREN instruction will set the latch and the WRDI instruction will reset the latch. This latch is automatically reset upon a power-up condition and after the completion of a byte, page, or status register write cycle.

Read Sequence

When reading from the EEPROM memory array, CS is first pulled LOW to select the device. The 8-bit READ instruction is transmitted to the X25020, followed by the 8-bit address. After the READ opcode and address are sent, the data stored in the memory at the selected address is shifted out on the SO line. The data stored in memory at the next address can be read sequentially by continuing to provide clock pulses. The address is automatically incremented to the next higher address after each byte of data is shifted out. When the highest address is reached ($FF) the address counter rolls over to address $00 allowing the read cycle to be continued indefinitely. The read operation is terminated by taking CS HIGH.

To read the status register CS line is first pulled LOW to select the device followed by the 8-bit RDSR instruction. After the read status register opcode is sent, the contents of the status register are shifted out on the SO line.

Write Sequence

Prior to any attempt to write data into the X25020 the “write enable” latch must first be set by issuing the WREN instruction. CS is first taken LOW, then the WREN instruction is clocked into the X25020. After all eight bits of the instruction are transmitted, CS must then be taken HIGH. If the user continues the write operation without taking CS HIGH after issuing the WREN instruction, the write operation will be ignored.

To write data to the EEPROM memory array, the user issues the WRITE instruction, followed by the address and then the data to be written. This is minimally a thirty-two clock operation. CS must go LOW and remain LOW for the duration of the operation. The host may continue to write up to 4 bytes of data to the X25020. The only restriction is that the 4 bytes must reside on the same page.

If the address counter reaches the end of the page and the clock continues, the counter will “roll over” to the first address of the page and overwrite any data that may have been written.

For the write operation (byte or page write) to be completed, CS can only be brought HIGH after bit 0 of data byte N is clocked in. If it is brought HIGH at any other time the write operation will not be completed.

To write to the status register, the WRSR instruction is followed by the data to be written. Data bits 0, 1, 4, 5, 6 and 7 must be “0”.

While the write is in progress following a status register or EEPROM write sequence, the status register may be read to check the WIP bit. During this time the WIP bit will be HIGH.

Operational Notes

The X25020 powers-up in the following state:
- The device is in the low power standby state.
- A HIGH to LOW transition on CS is required to enter an active state and receive an instruction.
- SO pin is high impedance.
- The “write enable” latch is reset.

7.10 Real Time Clock

The RTC is 'shadowed' under the FLASH memory replacing the even byte in the lower 1 Mbyte (address $0 - 000FFFFF) FLASH bank . A highly structured sequence of 64 cycles is used to gain access to the time information and temporarily disconnect the FLASH from the system bus. Information transfer is achieved by using address bits CPUA1 and CPUA3 and data bit CPUD24. All RTC operations are then accomplished by read cycles from the FLASH address space. Write and Read operations are determined by the level of CPUA3. When CPUA3 is low, a write cycle is enabled and the data is determined by CPUA1. When CPUA3 is high, a read operation is enabled and data is read on CPUD24.

As the RTC has a relatively slow access time (200nS), it is necessary to extend the DTACK period during RTC accesses; including the cycles used to gain access to the RTC. This is achieved by increasing the value of the TCYC field in the CS0 option register to 6.

As the PROM is disabled when accessing the RTC, the code to access it must first be copied to RAM and executed from there.

For further information refer to the DS1215 data sheet available in the BVME3000 Engineering Kit.

7.11 Interrupt Controller

The interrupt controller on the BVME3000 is responsible for two independent functions:

Processor Interrupter - Takes interrupts from all external sources (including VMEbus IRQ’s IndustryPacks™ IRQ’s Abort ACFAIL and Location monitor) level prioritises them and generate interrupts to the CPU.

VMEbus Interrupter - Generate Interrupts on the VMEbus.

7.11.1 Processor Interrupter

Interrupt Source	Level	Type
ABORT Switch	7	Auto-vectored
VMEbus ACFAIL	7	Auto-vectored
Location Monitor	2	Auto-vectored
VMEbus IRQ 7	7	Vectored. Individually maskable in the VMEbus Interrupt Enable Board Control Register
VMEbus IRQ 6	6	
VMEbus IRQ 5	5	
VMEbus IRQ 4	4	
VMEbus IRQ 3	3	
VMEbus IRQ 2	2	
VMEbus IRQ 1	1	
IP A Int 0	1 or 6	Level individually seleected to 1 or 6 in the IndustryPack™ Interrupt level Board control Register
IP A Int 1	1 or 6	
IP B Int 0	1 or 6	
IP B Int 1	1 or 6	
IP C Int 0	1 or 6	
IP C Int 1	1 or 6	
IP C Int 0	1 or 6	
IP C Int 1	1 or 6	

Where multiple sources are generating interrupts on the same level, the acknowledge cycle is prioritised as follows:

Highest Priority: Auto-vectored
VMEbus Interrupt
Lowest Priority: IP

Autovectored Interrupts

Level 7 interrupts

The BVME3000 has two level 7 sources. ABORT and ACFAIL. As both of these sources share the same service routine, the “Autovectored Level 7 Source board control register” can be read to determine the source.

Autovector Level 7 Source Board Control Register: FF800F13
Size: 8 bit
Access: Read

7	6	5	4	3	2	1	0
						ABORT	ACFAIL

ABORT - level 7 source

When set ABORT is the cause of the Autovectored Interrupt. This bit is cleared at reset until an event occurs. Once the Abort signal is removed the pin will be clear.

ACFAIL - Level 7 source

When set ACFAIL is the cause of the Autovectored Interrupt. This bit is cleared at reset until an event occurs. Once the ACFAIL signal is removed the pin will be clear but by then the system would have powered down.

Level 2 Interrupts

The level 2 Interrupt source is driven by the BVME3000 location Monitor. When a separate VME master accesses the BVME3000 location monitor slave address an Autovectored level 2 interrupt will be generated provided the address has been enabled onto the bus and the "Location Monitor Interrupt Enable board control register" bit (LOCINT) has been set. The interrupt is cleared by clearing the LOCINT bit

Location Monitor Interrupt Enable Board Control Register: FF800F15
Size: 8 bit
Access: Write

7	6	5	4	3	2	1	0
							LOCINT

LOCINT - Location Monitor Interrupt Enable

When set Location Monitor Interrupts are enabled, when clear Location monitor Interrupts are disabled. After an interrupt the bit must be cleared to remove the source and the set to allow a further interrupt. After reset the register is clear.

For a description of setting and enabling the slave address please refer to "7.14.3 VMEbus standard Slave board Control Register".

VMEbus Interrupts

The BVME3000 supports VMEbus Interrupts on all 7 levels. A "VMEbus Interrupt Enable board control register" gives the user the opportunity to mask or enable VMEbus interrupts as required.

VMEbus Interrupt Enable Board Control Register: FF800F17
Size: 8 bit
Access: Write

7	6	5	4	3	2	1	0
VIRQ7	VIRQ6	VIRQ5	VIRQ4	VIRQ3	VIRQ2	VIRQ1	

VIRQX - VMEbus Interrupt level X source Enable

When set a VMEbus interrupt on that level will be generated to the processor, when cleared the interrupt will be masked. On the RP3000 all interrupts must be masked thus all bits should be clear. After a reset all bits are cleared.

IndustryPack™ Interrupts

Each IP can generate interrupts on two separate IRQ lines INT0 and INT1. The "IndustryPack™ Interrupt level board control register" programs each IndustryPack™ interrupt source to either IRQ1 or IRQ6 depending on the status of the interrupt source bit.

Industrypack™ Interrupt Level Board Control Register: FF800F09
Size: 8 bit
Access: Write

7	6	5	4	3	2	1	0
INTREQD1	INTREQD0	INTREQC1	INTREQC0	INTREQB1	INTREQB0	INTREQA1	INTREQA0

INTREQXX - Industrypack™ Interrupt Level

When set the interrupt source will generate a level 6 interrupt to the processor, when clear a level 1 interrupt will be generated. After reset this bit is clear.

As only two interrupt lines are allocated to the IndustryPack™ controller it is likely that more than one Interrupt will be allocated to the same level. In this case, the Interrupt lines are prioritised inside the interrupt controller as follows:

Highest Priority:	INTREQA0
	INTREQA1
	INTREQB0
	INTREQB1
	INTREQC0
	INTREQC1
	INTREQD0
Lowest Priority	INTREQD1

7.11.2 VMEbus Interrupter

The BVME3000 can generate interrupts on the VMEbus. This function is completely independent of the processor Interrupter function. When acknowledged by the VMEbus interrupt handler, the BVME3000 returns a programmable 8-bit vector. The interrupter is implemented as Release On acknowledge (ROAK). Thus the interrupt is cleared by the interrupt acknowledge cycle.

The BVME3000 may generate an interrupt on any of the seven VMEbus IRQ levels. However, it can only generate on a single level at any one time. The level on which an interrupt is generated is set in the “VMEbus Slave Interrupt level board control register”.

VMEbus Interrupt level Board Control Register: FF800F1B
Size: 8 bit
Access: Write

7	6	5	4	3	2	1	0
					VIPL2	VIPL1	VIPL0

VIPL2 - VIPL0 - VMEbus interrupt out level

These bits select on which VMEbus Interrupt level the board will act as an interrupter. The binary code written selects the interrupt level, for example 101 corresponds to level 5. When set to 000, no interrupt can be generated on VMEbus. After reset these bits are cleared.

Assuming an interrupt level has been programmed into the “VMEbus Interrupt level Board Control Register”, an interrupt is asserted by the processor writing to the “VMEbus Interrupt Vector board control register” (VIVBCR) The value written to the VIVBCR Vector Register is then used as the value returned in the subsequent acknowledge cycle. The action of the vector fetch during the service routine will clear the interrupt.

VMEbus Interrupt Vector Board Control Register: FF800F19
Size: 8 bit
Access: Write

7	6	5	4	3	2	1	0
VEC7	VEC6	VEC5	VEC4	VEC3	VEC2	VEC1	VEC0

VEC7 -VEC0 - VMEbus interrupt Vector

Writing a value into this register will generate a VMEbus interrupt on the level programmed onto the "VMEbus Interrupt level Board Control Register". The value will act as the Vector during the service routine. The vector fetch will clear the Interrupt. After reset this register is clear.

7.12 VMEbus system Controller functions

The BVME3000 will act as a system controller when configured as such. Most functions are configured using Links, including RESET, BERR and SYSCLK generation however the internal architecture must also detect the system controller status. This is achieved by writing to the "VMEbus Arbitration board control register".

VMEbus Arbitration Board Control Register: FF800F0B
Size: 8 bit
Access: Write

7	6	5	4	3	2	1	0
							ARBIT

ARBIT - VMEbus Arbitration Enable

When set VMEbus arbitration is performed by the BVME3000. When the clear BVME3000 is not the system controller. After reset this bit is clear.

7.13 VMEbus Master Access

The BVME3000 can access VMEbus as a bus master. Depending on the Address Range used, different types of access are performed. VMEbus specifies three basic Address Mode schemes - A16 (Short I/O), A24 (Standard) and A32 (Extended). The BVME3000 supports A24 and A16 access modes. VMEbus also specifies three basic Data Transfer schemes - D08(EO), D16 and D32. The BVME3000 supports D08(EO) and D16 modes.

The BVME3000 does not support Block transfers A32 D32 A64 and D64.

7.13.1 A16:D16 (D08EO)

Base Address: FFFF0000.
Size: 64Kbyte.

Accesses to this area perform a Short I/O access to VMEbus with LWORD inactive. Long Word accesses are automatically broken down to Word (D16) cycles. Byte accesses produce a D08(EO) cycle. These accesses only involve signals available on the P1 VMEbus Connector.

The BVME3000 treats the short I/O space as a peripheral, using CS7 to select the space. Processor Acknowledges are derived externally.

OR7 Option Register 7: FF0010C4
Size: 32 Bit
Access: Read/Write

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TCYC 3	TCYC 2	TCYC 1	TCYC 0	AM27	AM26	AM25	AM24	AM23	AM22	AM21	AM20	AM19	AM18	AM17	AM16
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AM15	AM14	AM13	AM12	AM11	FCM3	FCM2	FCM1	FCM0	BCYC 1	BCYC 0	NU	PGME	SPS1	SPS0	DSEL
0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0

Cycle wait states: 0 Not supported
Mask Address: $FFFF0000.
Function Code support: 0 All (not CPU Space).
Burst Cycle Length: 0 Not supported.

Page Mode: 0 Disabled.
SRAM Port size: 2 External DSACKx.
SRAM/DRAM: 0 SRAM.

BR7 Base Register 7: FF0010C0
Size: 32 Bit
Access: Read/Write

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BA31	BA30	BA29	BA28	BA27	BA26	BA25	BA24	BA23	BA22	BA21	BA20	BA19	BA18	BA17	BA16
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BA15	BA14	BA13	BA12	BA11	FC3	FC2	FC1	FC0	TRLX Q	BACK 40	CSNT 40	CSNT Q	PARE N	WP	V
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Base Address: $FFFF0000.
Function Code Support: 0 Any.
Timing Relax: 0 No.
Burst Acknowledge 040: 0 Not supported.
CS Negate timing 040: 0 Not supported.

CS Negate timing: 0 at end.
Parity checking: 0 Disabled.
Write Protect: 0 Disabled.
Valid: 1 Yes (BVME3000).
0 no (RP3000)

For more information refer to “7.4.6 Memory Configuration”.

The following Address Modifier (AM) codes are generated:
CPU Supervisor Data Access = $2D
CPU User Data Access = $29

7.13.2 A24:D16 (D08EO)

Base Address: FE000000.
Size: 16Mbyte.

Accesses to this area perform a Standard Address access to VMEbus with LWORD inactive Long Word accesses are automatically broken down to Word (D16) cycles. Byte accesses produce a D08(EO) cycle. These accesses only involve signals available on the P1 VMEbus Connector.

Address decoding for this area is performed in logic.

The following Address Modifier (AM) codes are generated:

CPU Supervisor	Program Access	=	$3E
	Data Access	=	$3D
CPU User	Program Access	=	$3A
	Data Access	=	$39

7.14 VMEbus Slave Access

The BVME3000 allows other VMEbus masters to access the onboard SRAM. The memory can be accessed via the A24 address space. The BVME3000 also acts as a location monitor for accesses to a programmable memory area within the A24 address space.

7.14.1 Dual Ported SRAM

The BVME3000 allows other Bus Masters to access its 512Kbytes of SRAM in the A24 standard address space. It provides 16Mbytes of address space and the SRAM can be located anywhere in this space on 512Kbyte boundaries. The SRAM location is programmed in the “VMEbus Standard Slave Address board control register” (VSSABCR), by setting the address bits. The SRAM is enabled onto the bus by setting the “VMEbus Standard Slave Enable board control register” (VSSEBCR).

The memory may only be accessed provided one of the following Address Modifier codes is provided:-

CPU Supervisor	Program Access	= $3E
	Data Access	= $3D
CPU User	Program Access	= $3A
	Data Access	= $39

7.14.2 Location Monitor

The BVME3000 provides a location Monitor (Location Monitor) Interrupt which is set by an external Bus Master accessing standard A24 space at either the 512Kbytes above the SRAM, if the SRAM is on a 1Mbyte boundary (A19 not set in the VSSABCR), or by accessing the SRAM if it sits on a 512Kbyte boundary (A19 set in the VSSABCR). The Location monitor address space is enabled onto the bus by setting VSSEBCR.

The Location Monitor may only be accessed provided one of the following Address Modifier codes is provided:-

CPU Supervisor	Program Access	= $3E
	Data Access	= $3D
CPU User	Program Access	= $3A
	Data Access	= $39

7.14.3 VMEbus standard Slave board Control Register

VMEbus Standard Slave Address board control register: FF800F1D
Size: 8 bit
Access: Write

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
			VMEA23	VMEA22	VMEA21	VMEA20	VMEA19

VMEA23-19 VMEbus Address compare bits

These bits are used to set the comparitor for the VME Dual Port SRAM base address for example the bits were set 10011 the SRAM would be located at $980000. At reset all bits are set to 0.

VMEbus Standard Slave Address board control register: FF800F1F
Size: 8 bit
Access: Write

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
						LOCEN	SRAMEN

SRAMEN - SRAM Slave Address enable

This bit is used to enable the SRAM onto the standard A24 space. If clear SRAM is not enabled onto the standard A24 space. If enabled the SRAM is available in the standard A24 space at the address specified by the VMEAXX bits. After Reset this bit is clear and SRAM is disabled.

LOCEN- Location Monitor Address enable

This bit enables the 512Kbyte Location Monitor location onto the standard A24 space either directly above the SRAM space if VMEA19 is clear or on top of the SRAM if VMEA19 is set. When LOCEN is clear then the Location Monitor location is disabled. When LOCEN is set the Location Monitor location is enabled. At Reset this bit is clear and the Location Monitor location is disabled.

7.15 IndustryPack™ Controller

The BVME3000 provides 4 IndustryPack™ sites each supporting I/O, ID, MEMORY and Interrupt Cycles at both 8MHz or 32MHz programmable in the “IndustryPack™ Clock Speed board control register”. The IndustryPack™ controller supports 32 bit access to sites A and B as a pair and to sites C and D as a pair. D16 and D32 Memory IP’s are supported.

7.15.1 IndustryPack™ Interrupts

Each IP can generate interrupts on two separate IRQ lines, INT0 and INT1. The IP Interface contains registers for setting, in software, each IRQ source to either level 1 or level 6. IP interrupts are internally prioritised for the case where two interrupts are requesting on the same level. Refer to "7.11.1 Processor Interrupter" for details.

7.15.2 Memory Space Address Map

The Address Map logic decodes an 128Mbyte region for memory space, this region is then further decoded in the IP Interface.

Each IP can be accessed in two ways:

1. As a single, 16-bit wide device through an 8Mbyte window.
2. As a double-size, 32-bit wide device through a 16Mbyte window. In this mode, a pair of IP sites are used to take the double-size IP.

Thus A26 to A0 are used by the IPs or IP Interface for memory space decoding as shown in the following table.

Address Range	A[26:23]	Size	IP Selected	IP A[22-A1]
F0000000 - F07FFFFF	0000	8Mb (D16)	IP A	A22 - A1
F0800000 - F0FFFFFF	0001	8Mb (D16)	IP B	A22 - A1
F1000000 - F17FFFFF	0010	8Mb (D16)	IP C	A22 - A1
F1800000 - F1FFFFFF	0011	8Mb (D16)	IP D	A22 - A1
F4000000 - F4FFFFFF	100x	16Mb (D32)	IP A/B	A23 - A2
F5000000 - F5FFFFFF	101x	16Mb (D32)	IP C/D	A23 - A2

7.15.3 I/O & ID Space Address Map

The address map logic decodes a 1Mbyte space of the I/O region to IP I/O. This space is shared amongst the 4 IPs.

Each IP has an I/O space consisting of 128 bytes mapped as 64 words. Also each IP has 128 bytes of ID space, this again is word mapped.

In addition each pair of IPs can be combined to give 32-bit wide accesses, as for memory accesses.

Address Range	A[11:7]	Size	IP Selected	IP A[6-1]
FF800000 - FF80007F	0000 0	128 byte (D16)	IP A I/O	A6 - A1
FF800080 - FF8000FF	0000 1	128 byte (D16)	IP A ID	A6 - A1
FF800100 - FF80017F	0001 0	128 byte (D16)	IP B I/O	A6 - A1
FF800180 - FF8001FF	0001 1	128 byte (D16)	IP B ID	A6 - A1
FF800200 - FF80027F	0010 0	128 byte (D16)	IP C I/O	A6 - A1
FF800280 - FF8002FF	0010 1	128 byte (D16)	IP C ID	A6 - A1
FF800300 - FF80037F	0011 0	128 byte (D16)	IP D I/O	A6 - A1
FF800380 - FF8003FF	0011 1	128 byte (D16)	IP D ID	A6 - A1
FF800800 - FF8008FF	1000 X	256 byte (D32)	IP A/B I/O	A7 - A2
FF800900 - FF8009FF	1001 X	256 byte (D32)	IP C/D I/O	A7 - A2

7.15.4 IndustryPack™ DMA

The BVME3000 supports IP DMA access to the I/O spaces on IPA DMA channel 0 and to either IPA DMA channel 1 or IPC DMA channel 0, set in the “IndustryPack™ A Clock Speed board control register”, and the settings of LK10 and LK11. Memory DMA accesses are not supported.

The 68360 has two Independent DMA channels which act as the DMA controller to the IndustryPacks™. These channels are highly programmable and the user should refer to the MC68360 user manual for more details.

The DMA control signals are made available from the 68360’s port B, the pin direction, type and level must all be configured before operation. These functions can be selected in the Port B registers. Register description is available in “7.9.1 SPI configuration”. The following shows the register settings for DMA operation.

BPPAR Port B Pin Assignment register: FF0016BC
Size: 18 Bit
Access: Read/Write

17	16	15	14	13	12	11	10	9	8	5	6	5	4	3	2	1	0
DD 17	DD 16	DD15	DD14	DD13	DD12	DD11	DD10	DD9	DD8	DD7	DD6	DD5	DD4	DD3	DD2	DD1	DD0
X	X	X	X	X	X	X	X	1	1	1	1	1	1	X	X	X	X

BPDIR Port B Data Direction Register: FF0016B8
Size: 18 Bit
Access: Read/Write

17	16	15	14	13	12	11	10	9	8	5	6	5	4	3	2	1	0
DR17	DR16	DR15	DR14	DR13	DR12	DR11	DR10	DR9	DR8	DR7	DR6	DR5	DR4	DR3	DR2	DR1	DR0
X	X	X	X	X	X	X	X	1	1	1	1	1	1	X	X	X	X

BPODR Port B Open Drain Register: FF0016C2
Size: 16 Bit
Access: Read/Write

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OD15	OD14	OD13	OD12	OD11	OD10	OD9	OD8	OD7	OD6	OD5	OD4	OD3	OD2	OD1	OD0
X	X	X	X	X	X	0	0	0	0	0	0	X	X	X	X

For more information on Port B register configuration refer to “7.9.1 SPI configuration”

It is only necessary to program Bits 9 - 4 in the port B registers as these bits contain the dedicated DMA pins:-

Bit 4 DREQ1 This input receives the request from IPA DMA channel 0.

Bit 5 DACK1 This output to IPA DMA channel 0 acknowledges its DMA request for the current cycle.

Bit 6 DONE1 This Bi-directional signal indicates to/from the IPA DMA channel 0 that DMA transfers have ended.

Bit 4 DREQ2 This input receives the request from IPA DMA channel 1 or IPA DMA channel 0.

Bit 5 DACK2 This output to IPA DMA channel 1 or IPA DMA channel 0 acknowledges its DMA request for the current cycle.

Bit 6 DONE2 This Bi-directional signal indicates to/from the IPA DMA channel 1 or IPA DMA channel 0 that DMA transfers have ended.

The IP DMA channel attached to the 68360 channel 2 depends on the setting in the “IndustryPack™ A Clock Speed board control register”.

If the DMA interface pins are required for a different task, LK10 and LK11 jumpers must be removed.

7.15.5 IndustryPack™ Clock Speed Board Control Registers

Each IndustryPack™ site has its own board control register for setting its clock speed. IPA has an additional bit to select the second processor DMA source.

IPA Clock Speed /DMA2 select board control register: FF800F01
Size: 8 bit
Access: Write

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
						IPDMA	IPACLK

IPACLK - IPA clock speed

When set IPA will operate at 32MHz, when clear IPA will run at 8MHz. After reset this bit is clear with IPA clock running at 8MHz to conform to the IndustryPack™ specification.

IPDMA -Processor DMA channel 2 Source

When set, DMA accesses to IPC channel 0 can be performed, when clear DMA accesses to IPA channel 1 can be performed. This bit is cleared on reset.

IPB Clock Speed board control register: FF800F03
Size: 8 bit
Access: Write

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
							IPBCLK

IPBCLK - IPB clock speed

When set IPB will operate at 32MHz, when clear IPB will run at 8MHz. After reset this bit is clear with IPB clock running at 8MHz to conform to the IndustryPack™ specification.

IPC Clock Speed board control register: FF800F05
Size: 8 bit
Access: Write

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
							IPCCLK

IPCCLK - IPC clock speed

When set IPC will operate at 32MHz, when clear IPC will run at 8MHz. After reset this bit is clear with IPC clock running at 8MHz to conform to the IndustryPack™ specification.

IPD Clock Speed board control register: FF800F07
Size: 8 bit
Access: Write

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
							IPDCLK

IPDCLK - IPD clock speed

When set IPD will operate at 32MHz, when clear IPD will run at 8MHz. After reset this bit is clear with IPD clock running at 8MHz to conform to the IndustryPack™ specification.

8. Specification

8.1 On-Board Functions

MC68EN360	Quad Integrated Communications Controller with Ethernet @ 32MHz. QUICC I/O available via P2 connector and 60 way IDC header. Background Debug Mode support.
SERIAL PORT	RS232 controlled from SCC4 on the MC68EN360. Connection via RJ45 socket through the Front Panel.
FLASH	1 or 2Mbyte (factory fitted), 16-bit wide, Bottom 128Kbytes Boot protected .
SRAM	512Kbytes CMOS SRAM, 32-bit wide, non-volatile for up to 7 days.
DRAM	4/8/12/16 Mbytes (factory fitted) 32-bit wide.
EXPANSION	4off 32pin JEDEC compatible sockets. Support 70nS -100nS devices. (Exact pinout PLD selectable).
EEPROM	X25020 256 x 8 bit non volatile storage. Accessed via the SPI interface on the MC68EN360.
DS1215S	Timer Clock Peripheral Battery Backed for 10 years.

LOCAL BUS TIMEOUT period 1024 CPU clocks (32µS @ 32MHz bus clock).

RED LED indicates CPU access.
GREEN LED indicates IndustryPack™ status.

RESET switch (if enabled).
ABORT header (level 7 auto-vectored interrupt).

8.2 VMEbus Master

Single Level 3 Requester
A24, A16
D16, D08(EO)
RMW
AM6

8.3 VMEbus Slave

A24
D16, D08(EO)
RMW
AM6
LOCATION MONITOR

8.4 VMEbus System Controller Functions

ARBITER: SGL, level3, FAIR ROR.
SYSCLK Driver.
SYSRESET Driver/Monitor power-up and switch.
VMEbus RESET minimum period = 200mS.
BUS TIMEOUT period 128µS.
ACFAIL monitor (level 7 auto-vectored interrupt).

8.5 VMEbus Interrupts

Interrupter D08(O) ROAK:
I(1-7) single level, software programmable.
Interrupt vector ID, software programmable.
Interrupt handler D08(O): I(1-7) all levels, software maskable.

8.6 IndustryPack™ Functions

Four IndustryPack™ compatible sites:

- 2 x Single IPs (16-bit) or 1 x Double IP (32-bit);
- 8MHz or 32MHz, software selectable;
- IP DMA to local memory support for IPA ch0; (**Not Rev B)**
- IP DMA to local memory support for IPA ch1 or IPC ch0, software selectable; **(Not Rev B)**
- Software programmable IP interrupts (Level 1 or Level 6);
- Front panel IP I/O connections.

8.7 Board Configuration

LINKS:
- EEPROM write enable;
- RESET switch enable;
- MC68EN360 I/O tri-state enable;
- ABORT Header;
- BOOT BLOCK write enable;
- SOFTWARE RESET enable;
- RESET switch enable;
- VMEbus SYSCLOCK enable;
- VMEbus RESET IN/OUT enable;
- VMEBERR enable;
- DMA Channel 1 enable; **(Not Rev B)**
- DMA Channel 2 enable **(Not Rev B)**

PROGRAM:
- IP Clock/DMA sources;
- IP Interrupt level 1/6;
- VMEbus Arbiter;
- EXPANSION memory wait state;
- EEPROM Chip select;
- AUTOVECTORED level 7 Source;
- LOCATION monitor Interrupt Control;
- VMEbus interrupt handler levels;
- VMEbus interrupt level;
- VMEbus interrupt vector ID;
- VMEbus SLAVE addressing;
- VMEbus SLAVE enables;

8.8 Operating Environment

Dimensions: 160mm x 233.35mm (6U) single slot.
Power: +5v 1.6A Typ (full memory) +12V 0mA Max, -12V 0mA
(excluding IP and Transition Module requirements),
Environmental: 0 to 70 °C, 95% humidity non-condensing (extended range to order).

Appendix A - Data Sheets & Manual References

MC68EN360

MOTOROLA MC68360 QUAD INTEGRATED COMMUNICATIONS CONTROLLER USER'S MANUAL (1995 MOTOROLA Order number: MC68360UM/AD REV.1)

AM29F040

AMD Am29F040 A Megabit (524,288 x 8-Bit) CMOS 5.0 Volt-only, Sector Erase FLASH Memory Data Sheet (Nov 1996 AMD Publication No. 17113 Rev E Amendment 0)

X25020

XICOR X25020 256 x 8 Bit SPI Serial E^2PROM with Block Lock™ protection Data Sheet (1996 Sheet No. 3834-1.8 6/10/96 T3/C1/D0 NS)

DS1215

DALLAS Semiconductors DS1215 Phantom Time Chip Data Sheet (1993 Sheet No. 021894)

VMEbus

The VMEbus SPECIFICATION (Sept 1987, VITA)

INDUSTRYPACK™ SPECIFICATION

GREENSPRING COMPUTERS IndustryPack™ Logic Interface Specification (1993 Revision 0.7.1).

Appendix B - BVME3000 System Initialisation Code

```
name System Initialize
ttl Hardware dependent routines for BVME3000

**------------------------------------------------------------------------------**
* SysInit and related hardware dependent routines
**------------------------------------------------------------------------------**
*
* Copyright (c) 1996 BVM Ltd.
*
* This source code is the proprietary and confidential
* property of BVM Ltd., and has been provided solely
* for the purposes of documentation and education.
* Reproduction, publication, or distribution in any
* form to any other party is strictly prohibited.
*
**------------------------------------------------------------------------------**
* Edition History
*
* Ed# Date     By  Comments
* --- -------- --- -----------------------------------------------------
*   0 25.09.96 NJH Creation.
*
Edition equ 0 current edition number

 use defsfile
 use <sysglob.m>
 use <reach32.m>
 use <ldbra.m>

**------------------------------------------------------------------------------**
* Debugger usage
*
ifdef DEBUGGER
USING_DEBUGGER set 1 define primitive debugger in use
endc

ifdef ROMBUG
USING_DEBUGGER set 1 define smart debugger in use
endc

**------------------------------------------------------------------------------**

 psect sysinit,0,0,Edition,0,0

 use <rompak.m>

**------------------------------------------------------------------------------**
* use: RAMCHK  BRn,A-Reg-Temp,D-Reg-Temp[,D-Reg-Counter]
*
RAMCHK  macro
        move.l  \1(a5),\3                   ; get base register for region
        andi.l  #0xfffff800,\3              ; mask out non address bits
        move.l  \3,\2                       ; need it in an address register
        move.l  #0x7e57ed17,\3              ; get pattern
        move.l  \3,(\2)                     ; store pattern
        not.l   \3                          ; get different pattern
        move.l  \3,4(\2)                    ; store pattern
        not.l   \3                          ; get original pattern
        cmp.l   (\2),\3                     ; was pattern stored?
        beq.s   ramchk.\@                   ; branch if so
        ifeq \#-4                           ; if counter given
        subq.l  #1,\4                       ; decrement counter if not fitted
        endc
        moveq.l #0,\3                       ; get a zero
        move.l  \3,\1(a5)                   ; disable chip select
        ramchk.\@
        endm

**------------------------------------------------------------------------------**

EJTesize        equ 10                              ; exception jump table entry size
Auto7vector     equ 31*4                            ; auto-level 7 vector offset
Debounce        equ 200000                          ; debounce count on abort switch

**------------------------------------------------------------------------------**
* bootmenu detection locations - these locations are the first 8 bytes
* of the 68360 DPR
*
MenuSync        equ     DPRBase
MenuCount       equ     DPRBase+4
```

```
**-----------------------------------------------------------------------------**
* BVME3000 board specific values for the GMR
* See section 9.1 and section 6.13.1
*
* o 15.625 microsec DRAM refresh period
* o Refresh cycle length 6 clocks
* o Page size 1024 addresses
* o 32 bit port size
* o Parity disabled
* o CS/RAS of slave will not assert when writing CPU space
* o Internal address MUXing enabled
*
GMR_Table:     dc.l    0x002C51A0            ; no DRAM
GMR_1Bank      dc.l    0x1EAC51A0            ; 1 DRAM bank
GMR_2Banks     dc.l    0x16AC51A0            ; 2 DRAM banks
GMR_3Banks     dc.l    0x0FAC51A0            ; 3 DRAM banks
GMR_4Banks     dc.l    0x07AC51A0            ; 4 DRAM banks

align

**-----------------------------------------------------------------------------**
* SysInit:  perform system specific initialization (part 1)
*
SysInit:

* Set the base address for the mc68360 on-board modules
*
       move.l  #FC_SCPU,d1                   ; function code for CPU space
       move.l  #FC_SDS,d2                    ; function code for supervisor data space
       move.l  #DPRBase,d0                   ;
       move.l  d0,a5                         ; a5 points to 68360 DPR
       ori.l   #0x00000101,d0                ; address valid and AS7 mask CPU space
       movec   d1,DFC
*      movec   d1,SFC
       moves.l d0,MBAR                       ; set in the module base address
       movec   d2,DFC
*      movec   d2,SFC

       move.l  #80000,d0                     ; delay a 20 ms (to drain SCC)
drain.delay.loop
       nop
       ldbra   d0,drain.delay.loop

* Clear Reset Status Register after saving reason for last reset
*
       btst.b  #7,RSR(a5)                    ; test external reset flag
       bne.s   skip.reset                    ; branch if external reset
       reset
       skip.reset
       move.b  #$80,RSR(a5)                  ; clear external reset flag

       move.l  GMR_4Banks(pc),GMR(a5)        ; load default GMR value for 4 DRAM banks

* Ensure that 100 usec. has elapsed before CS 1,2,3 & 4 are set up
       move.w  #$3ff,d0                      ; delay
sysinit.1
       tst.l   (a5)
       dbra    d0,sysinit.1

* CS0: 2Mb FLASH $00000000-$001FFFFF, 1 wait, 16 bit
*
* TCYC  = 2, AM = FFE00000, FCM = 0, BCYC = 0, PGME = 0, SPS = 01, DSEL = 0
*
* BA    = 00000000, FC    = 0, TRLXQ = 0, BACK40 = 0, CSNT40 = 0,
* CSNTQ = 1,        PAREN = 0, WP    = 0, V      = 1
*
       move.l  #$2fe00002,OR0(a5)            ;
       move.l  #$00000009,BR0(a5)            ;

* CS1: 4Mb DRAM $01000000-$013FFFFF, 4 wait, 32 bit
*
* TCYC  = 1, AM = FFC00000, FCM = 0, BCYC = 0, PGME = 0, SPS = 00, DSEL = 1
*
* BA    = 01000000, FC    = 0, TRLXQ = 0, BACK40 = 0, CSNT40 = 0,
* CSNTQ = 0,        PAREN = 0, WP    = 0, V      = 1
*
       move.l  #$1fc00001,OR1(a5)            ;
       move.l  #$01000001,BR1(a5)            ;

* CS2: 4Mb DRAM $01400000-$017FFFFF, 4 wait, 32 bit
*
* TCYC  = 1, AM = FFC00000, FCM = 0, BCYC = 0, PGME = 0, SPS = 00, DSEL = 1
*
* BA    = 01400000, FC    = 0, TRLXQ = 0, BACK40 = 0, CSNT40 = 0,
* CSNTQ = 0,        PAREN = 0, WP    = 0, V      = 1
*
```

```
        move.l  #$1fc00001,OR2(a5)              ;
        move.l  #$01400001,BR2(a5)              ;

* CS3: 4Mb DRAM $01800000-$01BFFFFF, 4 wait, 32 bit
*
* TCYC  = 1, AM = FFC00000, FCM = 0, BCYC = 0, PGME = 0, SPS = 00, DSEL = 1
*
* BA    = 01800000, FC    = 0, TRLXQ = 0, BACK40 = 0, CSNT40 = 0,
* CSNTQ = 0,        PAREN = 0, WP    = 0, V      = 1
*
        move.l  #$1fc00001,OR3(a5)              ;
        move.l  #$01800001,BR3(a5)              ;

* CS4: 4Mb DRAM $01C00000-$01FFFFFF, 4 wait, 32 bit
*
* TCYC  = 1, AM = FFC00000, FCM = 0, BCYC = 0, PGME = 0, SPS = 00, DSEL = 1
*
* BA    = 01C00000, FC    = 0, TRLXQ = 0, BACK40 = 0, CSNT40 = 0,
* CSNTQ = 0,        PAREN = 0, WP    = 0, V      = 1
*
        move.l  #$1fc00001,OR4(a5)              ;
        move.l  #$01c00001,BR4(a5)              ;

* CS5: 512Kb SRAM $00200000-$0027FFFF, 1 wait, 32 bit
*
* TCYC  = 2, AM = FFF80000, FCM = 0, BCYC = 0, PGME = 0, SPS = 00, DSEL = 0
*
* BA    = 00200000, FC    = 0, TRLXQ = 0, BACK40 = 0, CSNT40 = 0,
* CSNTQ = 1,        PAREN = 0, WP    = 0, V      = 1
*
        move.l  #$2ff80000,OR5(a5)              ;
        move.l  #$00200009,BR5(a5)              ;

* CS6: 4Mb Expansion $00400000-$007FFFFF, 2 wait, 32 bit
*
* TCYC  = 3, AM = FFC00000, FCM = 0, BCYC = 0, PGME = 0, SPS = 00, DSEL = 0
*
* BA    = 00400000, FC    = 0, TRLXQ = 0, BACK40 = 0, CSNT40 = 0,
* CSNTQ = 0,        PAREN = 0, WP    = 0, V      = 1
*
        move.l  #$3fc00000,OR6(a5)              ;
        move.l  #$00400001,BR6(a5)              ;

* CS7: VME Short I/O $FFFF0000-$FFFFFFFF, external
*
* TCYC  = 0, AM = FFFF0000, FCM = 0, BCYC = 0, PGME = 0, SPS = 11, DSEL = 0
*
* BA    = FFFF0000, FC    = 0, TRLXQ = 0, BACK40 = 0, CSNT40 = 0,
* CSNTQ = 0,        PAREN = 0, WP    = 0, V      = 1
*
        move.l  #$0fff0006,OR7(a5)              ;
        move.l  #$ffff0001,BR7(a5)              ;

 ifne 0
        ori.w   #$4000,PLLCR(a5)                ; Write protect on
        move.b  #$7C,SYPCR(a5)                  ; Disable SWT, enable bus monitor functions
 else
 ifne 0
        ori.w   #$607f,PLLCR(a5)                ; Write protect on
*                                               ;  divide-by-128, 128 multiplication factor
 else
        ori.w   #$4000,PLLCR(a5)                ; Write protect on
        endc
        endc
        move.b  #$8F,CLKOCR(a5)                 ; Write protect on, PLL Loss of Lock
*                                               ; clock output disable
        move.w  #$8000,CDVCR(a5)                ; Not using the low power clock
*                                               ; divider. Write Protect the register
        move.w  #$0200,PITR(a5)                 ; Prescale the SWT clock by 512
        move.b  #$55,SWSR(a5)                   ; Service watchdog to effect change
        move.b  #$AA,SWSR(a5)

        move.l  #$00006cff,MCR(a5)              ; Async Bus Timing, Async Arbitration

        move.w  #$0740,SDCR(a5)                 ; setup SDMA configuration register

*** Step 8 - Clear Entire Dual-Port RAM and delay for DRAM initialise
*
        move.w  #$3fd,d0                        ; Count
        lea.l   8(a5),a4                        ; skip menu count and sync bytes in DPR
sysinit.2
        clr.l   (a4)+
        dbra    d0,sysinit.2

        move.w  #$8001,CR(a5)                   ; CPM Reset Command reinitializes Dual
*                                               ; Port RAM
```

```
*** Step 10 - Write the PEPAR
*
       move.w  #$0420,PEPAR(a5)              ; Enable Bus Clear /BCLR0

*** Step 12 - Initialize System RAM
*
* The only requirement here is to ensure that 120 usec. has elapsed between
* CS 1,2,3 & 4 being set up and the first DRAM access.

       move.l  BR1(a5),d1                    ; save DRAM base address from CS1

*** Check for DRAM on CS 1, 2, 3 & 4
       moveq.l #4,d2                         ; initialise DRAM bank counter
       RAMCHK  BR1,a0,d0,d2
       RAMCHK  BR2,a0,d0,d2
       RAMCHK  BR3,a0,d0,d2
       RAMCHK  BR4,a0,d0,d2

*** Update DRAM refresh according to number of DRAM banks fitted
       move.l  GMR_Table(pc,d2.w*4),GMR(a5)
```

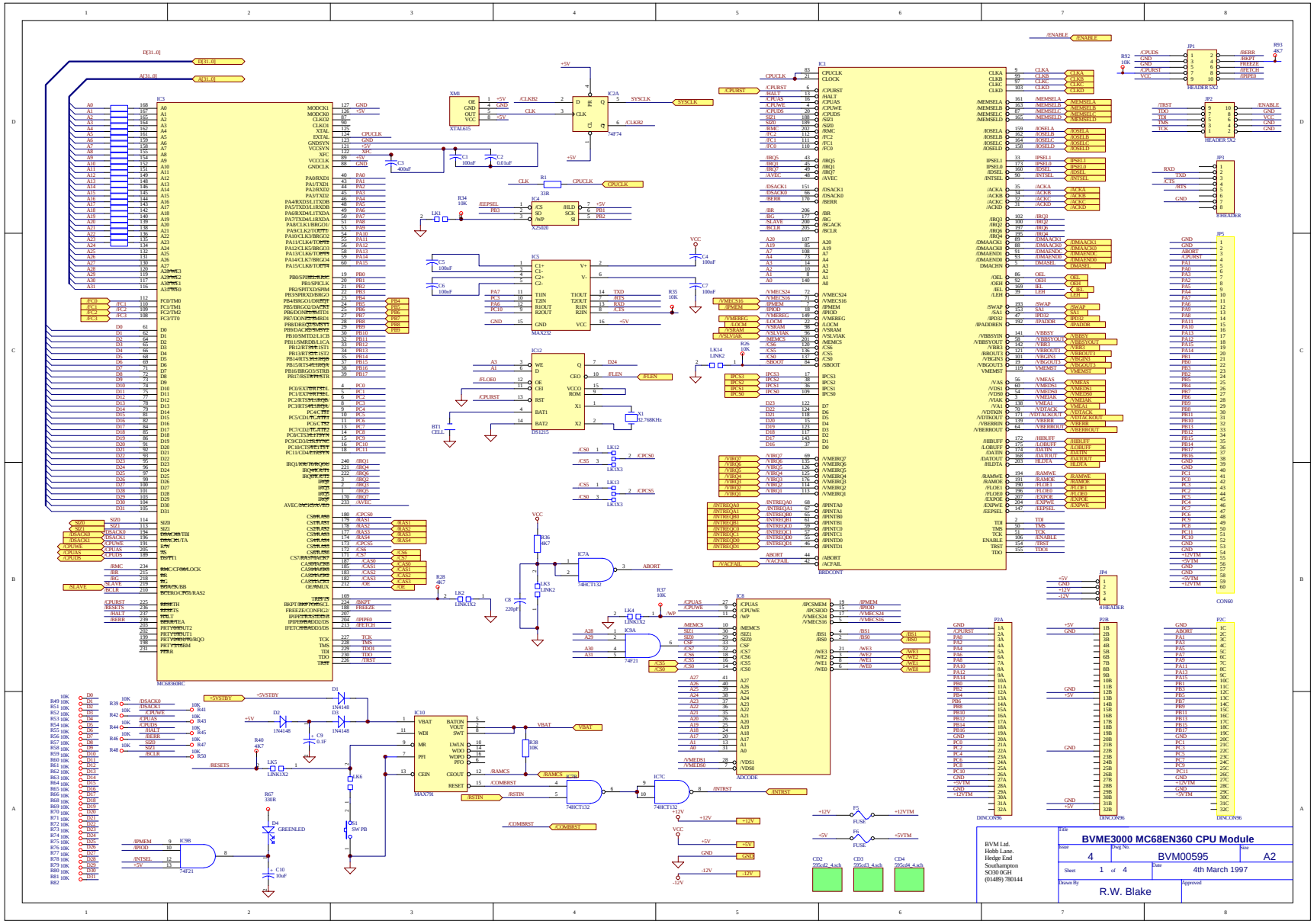
```
*** Check if DRAM is fitted on CS1, if not remap SRAM
*** or expansion RAM in its place

       tst.l   BR1(a5)                      ; was there DRAM on CS1
       bne.s   skip.remap                   ; branch if so
       lea.l   BR5(a5),a1                   ; default remap to SRAM

       RAMCHK  BR5,a0,d0                    ; check if SRAM is fitted

       tst.l   BR5(a5)                      ; was there SRAM on CS5
       bne.s   apply.remap                  ; branch if so

       lea.l   BR6(a5),a1                   ; remap to expansion RAM

apply.remap
       move.l  (a1),d0                      ; get base address register value
       andi.l  #0x3ff,d0                    ; mask out address bits
       andi.l  #0xfffff800,d1               ; mask out non address bits
       or.l    d1,d0                        ; change base to original DRAM address
       move.l  d0,(a1)                      ; apply change

skip.remap

*** Step 13 - Copy the EVT to System RAM
       movea.l VBRPatch(pc),a0
       movec   a0,vbr
* This ram must be sufficiant in size for OS-9 to place a vector
* table, stack and system globals.
*
       move.l  VectTbl(pc),0(a0)            ; initialize reset SSP...
       move.l  VectTbl+4(pc),4(a0)          ; ...and PC

***    Check & increment the RAM counter for bootmenu enable
       move.l  #$DAF750D5,d0
       cmp.l   MenuSync,d0                  ; got sync pattern?
       beq.s   IncCounter                   ; branch if so ..

       move.l  d0,MenuSync                  ; set sync pattern
       clr.l   MenuCount                    ; reset counter

IncCounter
       addq.l  #1,MenuCount                 ; increment count for boot menu test

* locate and execute extention code
* Note: MUST have RAM enabled for the stack before this is done!!!
*
       ROMPAK1

       bra             SysRetrn             ; return to boot.a

**------------------------------------------------------------------------**
* SInitTwo:  perform system specific initialization (part 2)
*

SInitTwo:
*** Setup board control registers
       bsr             r32_nvr_bcr_setup    ; must be done after global storage is setup

  ifdef USING_DEBUGGER
*** Setup handler for abort switch
       movem.l a0-a1,-(sp)                  ; save regs    movec   vbr,a0
                                            ; get vbr
       lea.l   AbortHdlr(pc),a1             ; get address of handler
       move.l  a1,Auto7vector(a0)           ; abort switch calls our handler
       movem.l (sp)+,a0-a1                  ; restore regs
       endc USING_DEBUGGER

*** Initialise console I/O
       bsr             r32_ConsInit         ; initialize console hardware
```

```
*** Setup serial port baud rate
       bsr            r32_nvr_consbaud      ; get baud rate code from EEPROM
       bsr            r32_ConsBaudSet       ; setup console baud rate
       bsr            r32_nvr_portbaud      ; get baud rate code from EEPROM
       bsr            r32_PortBaudSet       ; setup comms port baud rate

*** Print BVM part of the startup message
       lea            TypeMsg(pc),a0
       bsr            r32_PrintStr          ; print BVM type message

*** Display reset counter
       cmp.l   #1,MenuCount                 ;
       beq.s   SI2.SkipShowCount            ; don't display count if first reset

       lea            ResetCntMsg(pc),a0
       bsr            r32_PrintStr          ; display reset count heading
       move.l  MenuCount,d0                 ; get reset counter
       bsr            OutDec                ; print decimal string
       bsr            r32_PrintCR           ; start new line
SI2.SkipShowCount

       move.l  #4000000,d0                  ; delay a second (gives user a chance
*                                           ;  to reset again before boot file loaded)
delay.loop
       nop
       ldbra   d0,delay.loop

* Run additional setup that may or may not be in the extension module
       ROMPAK2
       rts

**---------------------------------------------------------------------------**
* Print decimal string of value in d0.l
*
OutDec:
       movem.l d0-d2/a0,-(a7)               ; save registers
       move.l  a7,a0                        ; get pointer past end of buffer
       lea.l   -32(a7),a7                   ; allocate buffer space
       move.b  #0,-(a0)                     ; store null terminator

*** build string of digits in reverse order
       moveq.l #10,d2                       ; get divisor
PD.Loop
       divul.l d2,d1:d0                     ; d0 divided by 10 = d0 remainder d1
       add.b   #'0',d1                      ; make ASCII digit
       move.b  d1,-(a0)                     ; store it
       tst.l   d0                           ; finished?
       bne.s   PD.Loop                      ; branch if not

       bsr     r32_PrintStr                 ; print string of digits

       lea.l   32(a7),a7                    ; free buffer and restore stack pointer
       movem.l (a7)+,d0-d2/a0               ; restore registers
       rts                                  ; return

**---------------------------------------------------------------------------**
* Routines called externally for reading/writing the boot menu locations
*
read_menu_count:
       move.l  MenuCount,d0
       rts

write_menu_count:
       move.l  d0,MenuCount
       rts
```

```
**------------------------------------------------------------------------**
* Abort Switch Handler
*
  ifdef USING_DEBUGGER
*
* This is an interrupt handler for the ABORT switch on the CPU
* front panel.  Notice that it must debounce the switch.
*
AbortHdlr:
       lea.l   -6(sp),sp                    ; space for SR and continue point
       move.w  sr,0(sp)                     ; save SR
       movem.l d0-d1/a0,-(sp)               ; save those registers needed
*
* First check for ACFAIL!!!
*
       btst.b  #B_ACFAIL,BCR$LEVEL7         ; ACFAILed ?
       bne.s   UserIRQ7                     ; yes..go to user's routine (if any)
*
* Check for ABORT
*
       btst.b  #B_ABORT,BCR$LEVEL7          ; has abort switch been pressed?
       beq.s   UserIRQ7                     ; no..go to user's routine (if any)
*
* Debounce the push button
*
       move.l  #Debounce,d0                 ; debounce counter
WaitLoop
       nop                                  ;
       subq.l  #1,d0                        ; decrement debounce counter
       bne.s   WaitLoop                     ; make sure fully debounced
*
* If the debugger is not enabled don't call it!
*
       bsr     nvr_debugger                 ; get debugger enabled flag
       tst.b   d0                                                    ;
       beq.s   UserIRQ7                     ; call users handler if disabled

*
* Continue at DEBUGGER'S Auto7 handler
*
DebugIRQ7
       movem.l (sp)+,d0-d1                  ; restore some registers
       movea.l #Aut7Trap-*-8,a0             ; offset to debugger's abort routine
       lea.l   (pc,a0.l),a0                 ; address of debugger's abort routine
       move.l  a0,6(sp)                     ; continue at auto7 handler
       movea.l (sp)+,a0                     ; restore final reg
       rtr                                  ; enter debugger

*
* Continue at USER'S Auto7 handler
*
UserIRQ7
       movem.l (sp)+,d0-d1                  ; restore some registers
       movea.l #ExcptJmp+EJTesize*29,a0     ; user's level 7 autovector routine
       move.l  a0,6(sp)                     ; continue at user's auto7 routine
       movea.l (sp)+,a0                     ; restore final reg
       rtr
  endc USING_DEBUGGER
```

```
**----------------------------------------------------------------------------**
* External Calls Vectoring
*
* these "internal vectors" provide the means for this code to
* call external routines that are further than 16-bit offsets.
*
        reach32 nvr_bcr_setup
        reach32 nvr_consbaud
        reach32 nvr_portbaud
        reach32 inttoascii
        reach32 delay
        reach32 PrintStr
        reach32 PrintCR
        reach32 ConsInit
        reach32 ConsBaudSet
        reach32 PortBaudSet

**----------------------------------------------------------------------------**
* BVM message string
*
TypeMsg         TypeNam                         ; specific string from systype.d
ResetCntMsg     dc.b C$CR,"Reset count: ",0

*  He who is responsible for this excursion.....
*
PortMan: dc.b "Portman: Nick Holgate",0,0

 ends

**-----------------------------< end of file >------------------------------**
```

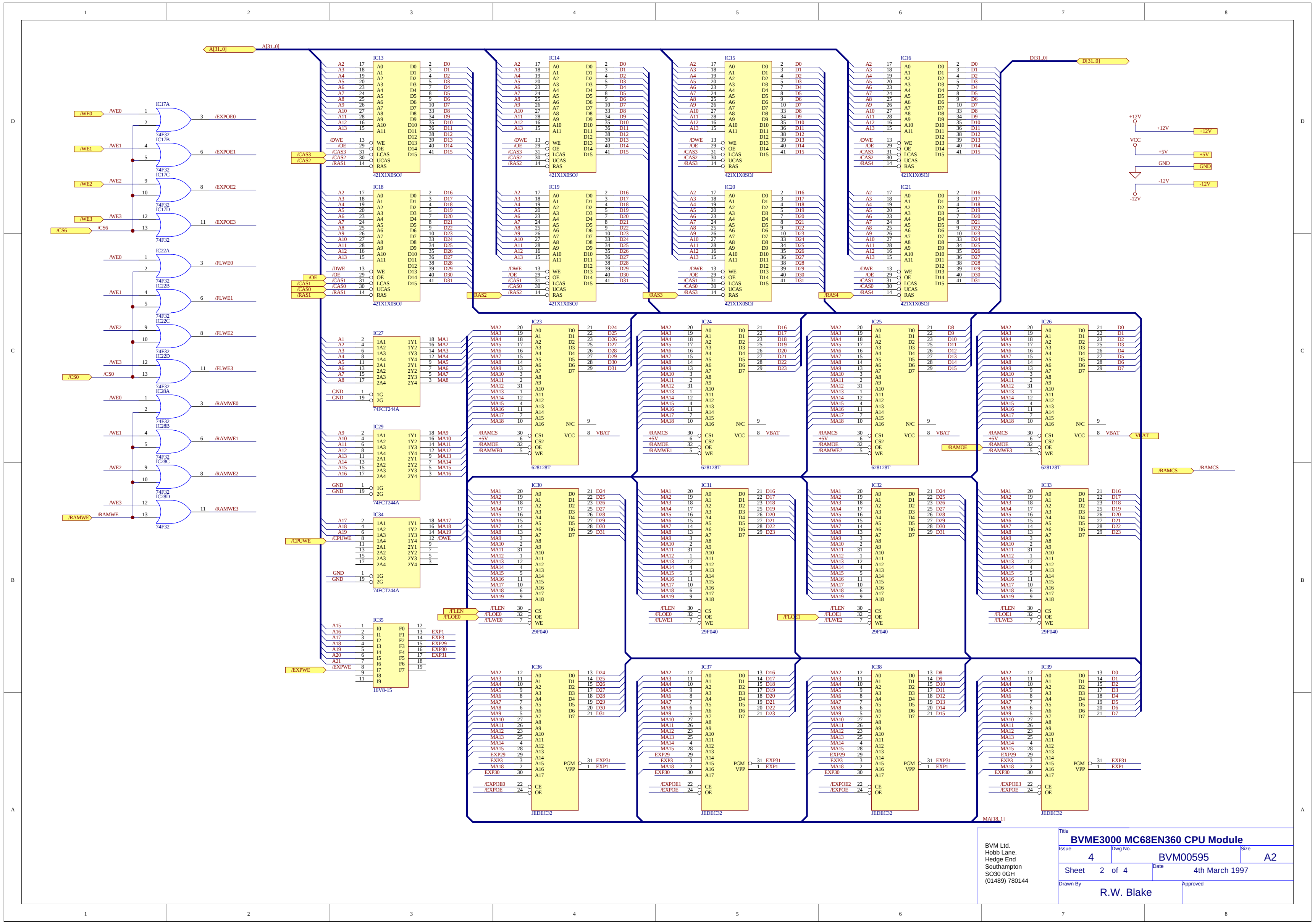
Appendix C - Circuit Diagrams



BVM Ltd.
Hobb Lane.
Hedge End
Southampton
SO30 0GH
(01489) 780144

BVME3000 MC68EN360 CPU Module

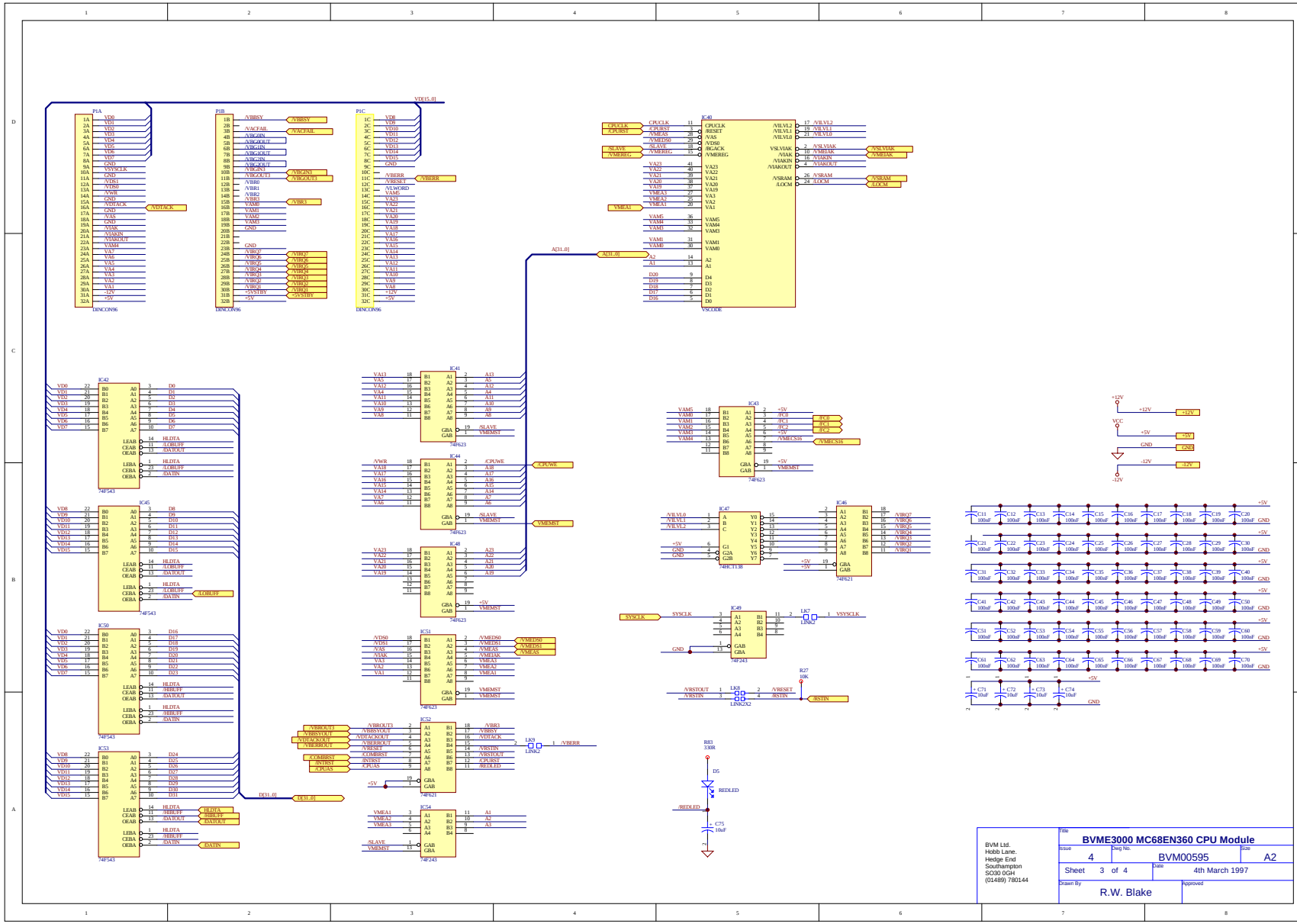
Issue	Dwg No.	Size
4	BVM00595	A2
Sheet 1 of 4	Date 4th March 1997	
Drawn By R.W. Blake	Approved	

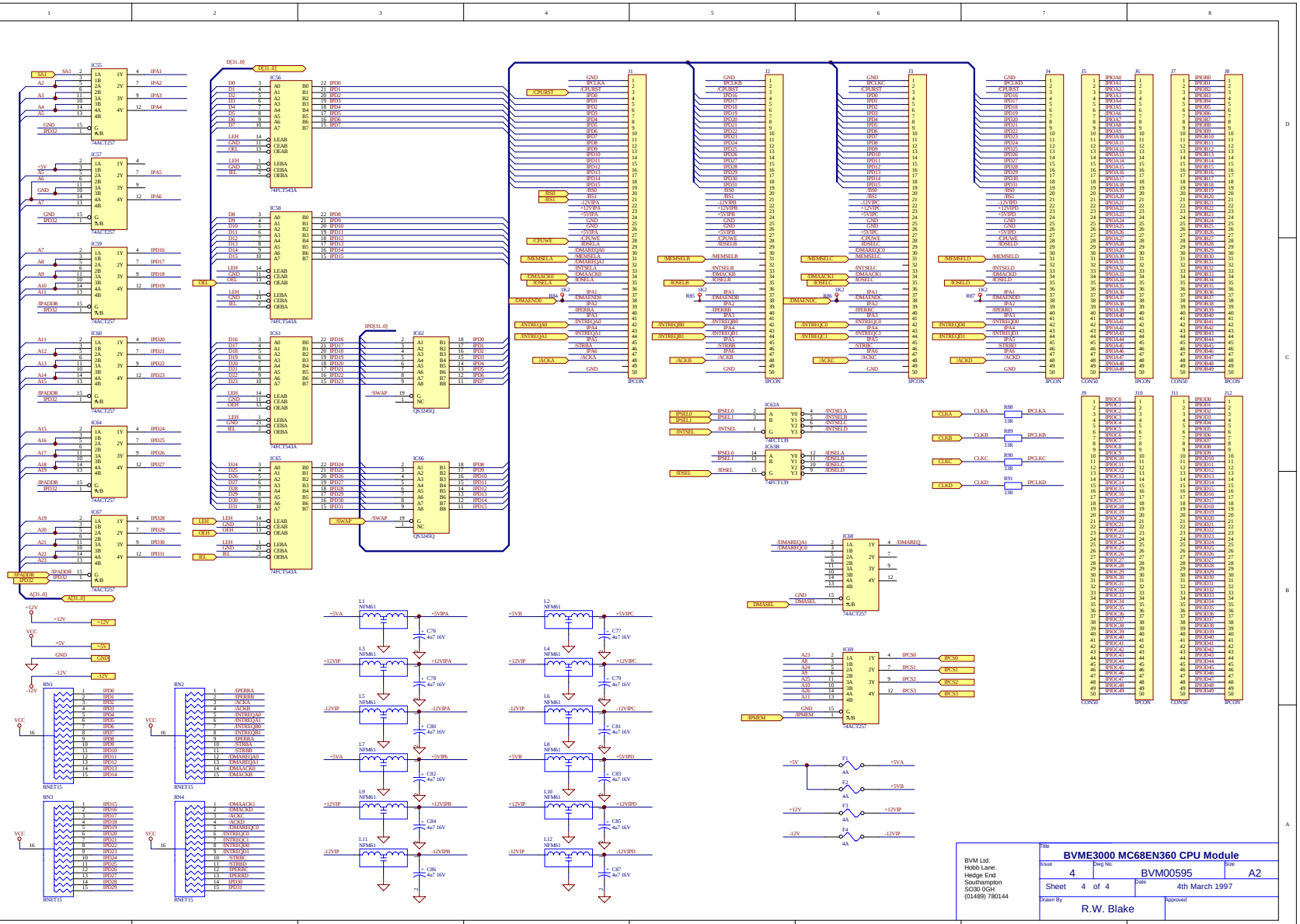


BVME3000 MC68EN360 CPU Module

BVM Ltd.
Hobb Lane.
Hedge End
Southampton
SO30 0GH
(01489) 780144

Title	BVME3000 MC68EN360 CPU Module		
Issue	4	Dwg No. BVM00595	Size A2
Sheet 2 of 4		Date 4th March 1997	
Drawn By	R.W. Blake	Approved	


P1A
P1B
P1C
DINCON96
VD[15..0]
/VDTACK
/VBBSY
/VACFAIL
/VBGRIN
/VBGOUT
/VBGIN
/VBGOUT
/VBGRIN
/VBGOUT
/VBGIN3
/VBGOUT3
/VBR0
/VBR1
/VBR2
/VBR3
/VBR7
/VBRQ6
/VBRQ5
/VBRQ4
/VBRQ3
/VBRQ2
/VBRQ1
/SYSFBY
/VBERR
/VRESET
/VLWORD
/VAM5
/VBERR
/VBR3
/VBRQ7
/VBRQ6
/VBRQ5
/VBRQ4
/VBRQ3
/VBRQ2
/VBRQ1
/SYSTBY
A[31..0]
IC40
CPUCLK
/CPURST
/VMEAS
/VMEDS0
/SLAVE
/VMEREG
/VILVL2
/VILVL1
/VILVL0
/VSLVIAK
/VIAIK
/VIAKIN
/VIAKOUT
/VSRAM
/LOCM
VSCODE
VMEA1
IC41
IC42
IC43
IC44
IC45
IC46
IC47
IC48
IC49
IC50
IC51
IC52
IC53
IC54
74F623
74F543
74F621
74HC138
74F243
/CPUWE
/VMEMST
/SLAVE
/LOBUFF
/HIBUFF
/DATOUT
/DATIN
/HLDTA
DI[31..0]
DO[31..0]
/VBROUT3
/VBRSYOUT
/VDTACKOUT
/VBERROUT
/VRESET
/COMRST
/INTRST
/CPUAS
/VBERR
/VBBSY
/VDTACK
/RSTIN
/RSTOUT
/CPURST
/REDLED
LK9
LINK2
LK7
LK8
LINK2X2
SYSCLK
/VSYSCLK
/VRSTOUT
/VRSTIN
/VRESET
R27
10K
RR3
330R
D5
REDLED
C75
10uF
+12V
VCC
+5V
GND
-12V
C11 100nF
C12 100nF
C13 100nF
C14 100nF
C15 100nF
C16 100nF
C17 100nF
C18 100nF
C19 100nF
C20 100nF
C21 100nF
C22 100nF
C23 100nF
C24 100nF
C25 100nF
C26 100nF
C27 100nF
C28 100nF
C29 100nF
C30 100nF
C31 100nF
C32 100nF
C33 100nF
C34 100nF
C35 100nF
C36 100nF
C37 100nF
C38 100nF
C39 100nF
C40 100nF
C41 100nF
C42 100nF
C43 100nF
C44 100nF
C45 100nF
C46 100nF
C47 100nF
C48 100nF
C49 100nF
C50 100nF
C51 100nF
C52 100nF
C53 100nF
C54 100nF
C55 100nF
C56 100nF
C57 100nF
C58 100nF
C59 100nF
C60 100nF
C61 100nF
C62 100nF
C63 100nF
C64 100nF
C65 100nF
C66 100nF
C67 100nF
C68 100nF
C69 100nF
C70 100nF
C71 10uF
C72 10uF
C73 10uF
C74 10uF
BVM Ltd.
Hobb Lane.
Hedge End
Southampton
SO30 0GH
(01489) 780144
Title
BVME3000 MC68EN360 CPU Module
Issue
4
Dwg No.
BVM00595
Size
A2
Sheet 3 of 4
Date
4th March 1997
Drawn By
R.W. Blake
Approved



BVM Ltd. Hobb Lane. Hedge End Southampton SO30 0GH (01489) 780144	Title: **BVME3000 MC68EN360 CPU Module** Issue: 4 — Drwg No.: BVM00595 — Size: A2 Sheet 4 of 4 — Date: 4th March 1997 Drawn By: R.W. Blake — Approved