

**Cambridge Microprocessor
Systems Limited
Micro-Module
Hardware**

Publication Date : April 1996

c Cambridge Microprocessor Systems Ltd 1994-6

The information contained in this document has been thoroughly checked and is believed to be factually correct. However Cambridge Microprocessor Systems Ltd. assumes no responsibility for any mistakes, omissions, or inaccuracies.

Cambridge Microprocessor Systems do not accept any liability out of the use of this or any other Cambridge Microprocessor Systems product.

Although Cambridge Microprocessor Systems Ltd. is committed to supplying its customers a standard, well tested product, we reserve the right to make modifications to the product described in this manual.

This page intentionally left blank

Table Of Contents

Micro-Module Hardware

1.1	Introduction	1-1
-----	------------------------	-----

The MC68307 Micro-Controller

2.1	The 68000 CPU	2-2
2.1.1	What is the Advantage of the 68000?	2-2
2.1.2	The 68000 programming model	2-3
2.1.3	Exception Processing	2-4
2.1.4	Interrupt Controller	2-6
2.1.5	Interrupt Vector Generation	2-6
2.1.6	68000 Function Codes	2-8
2.2	The Bus Interface	2-9
2.2.1	Main Memory Map	2-9
2.2.2	Local Memory Map	2-10
2.2.3	The 68000 Bus	2-10
2.2.4	The 8051 bus	2-13
2.2.5	The I ² C Bus (M-Bus)	2-14

On Board Peripherals

3.1	The UART	3-1
3.2	The Timer Module	3-2
3.2.1	General Purpose Timers	3-2
3.2.2	Watch Dog Timer	3-5
3.3	Digital I/O	3-7

Power Saving

4.1	Low Power Modes	4-1
4.2	Example	4-2

Registers

5.1	Introduction	5-1
5.2	System Configuration Registers	5-5
5.3	I/O Port Control Registers	5-7
5.4	Interrupt Control Registers	5-9
5.5	Chip Select Registers	5-11
5.6	UART Registers	5-13
5.7	Timer Module Registers	5-18
5.8	I ² C bus (M-Bus) Module Registers	5-21

Memory

6.1	Read Only Memory	6-1
6.2	Static RAM	6-2

Prototyping Area

7.1	The Micro-Module	7-1
7.2	The Micro-Midget	7-1

Module Bus expansion

8.1	Introduction	8-1
8.2	Bus Width Considerations	8-1
8.3	Asynchronous Operation	8-2
8.4	Memory Expansion	8-2
8.5	Peripheral Expansion	8-4
8.6	Interrupts	8-4

Links

9.1	External Reset LK1	9-2
9.2	RS-232 or RS-485 LK2	9-2

9.3	EEPROM Option LK3	9-2
9.4	Static RAM Size LK4	9-3
9.5	EPROM Size LK5, LK6, LK7	9-3
9.6	RS-485 Terminator LK8	9-4

Connectors

10.1	Power PL1	10-1
10.2	Module Bus PL2	10-1
10.3	Serial Port PL3	10-3
10.4	Prototype area interface	10-4
10.4.1	Micro-Module (K-030)	10-4
10.4.2	Micro-Midget (K-038)	10-4

Specification

11.1	Processor	11-1
11.2	Memory	11-1
11.3	Serial Port	11-1
11.4	Digital I/O	11-1
11.5	I ² C bus (M-Bus)	11-2
11.6	Counter Timers	11-2
11.7	Watch dog Timer	11-2
11.8	Prototyping Area (Micro-Module Only)	11-2
11.9	8051 Bus Access	11-2
11.10	Micro-Module	11-3
11.11	Micro-Midget	11-3
11.12	Power Supply	11-3
11.13	Operating Temperature	11-3

Application Notes

12.1 Converting to RS-485 12-1
12.2 Converting to RS-232 12-2
12.3 Micro-Midget Mounting Dimensions 12-3

Appendix

13.1 Useful data sheets 13-1
13.2 User Manuals and Books 13-1
13.3 Returns Policy 13-1
13.4 Designer Kits 13-2
13.5 Acknowledgements 13-2
13.6 Circuit Diagram 13-2

1 Micro-Module Hardware

Important

The sections of code contained in this manual are not necessarily written to operate with the C cross compiler support package supplied. Many of them are written so that they can be blown directly into EPROM once they have been assembled. Other code segments may not be programs at all but just examples to demonstrate a particular point.

Micro-Module and Micro-Midget

Throughout this manual the words Micro-Module can be read as Micro-Midget. The two boards are virtually identical, except that the Micro-Module has a prototyping area and the Micro-Midget does not. Where there are differences it will be made clear which board the documentation is referring to.

1.1 Introduction

The Micro-Module is a small 32/16/8 bit micro-controller board which combines a high performance 68000 compatible CPU with a wide range of on board peripherals. The I/O facilities provided by the Micro-Module include digital I/O ports, a serial port which can be used for communication using either RS-232 or RS-485 serial standards, and two timer/counters. The CPU has facilities to operate peripheral I/O devices on a wide range of buses including the 68000 16-bit expansion bus, the 8051 8-bit expansion or the popular I²C bus or M-Bus serial interface bus. The Micro-Module makes full use of the dynamic bus sizing facility provided by the processor allowing a single EPROM and RAM to be fitted and having a 16-bit bus interface. Because the RAM and EPROM only take up two sockets, half of the board space on the Micro-Module can be given over to a prototyping area which will enable the user to build their own custom interfaces or I/O requirements. All the necessary signals, such as address lines, data lines, strobe lines and chip selects, are easily available to this area on a patch type connector. The Micro-Module measures just 118mm x 100mm, the Micro-Midget 64mm x 100mm. Both boards are only 13mm high.

Low power modes are available in which the main system clock can either be stopped or slowed down. Power can also be reduced by turning off the

Micro-Module Hardware

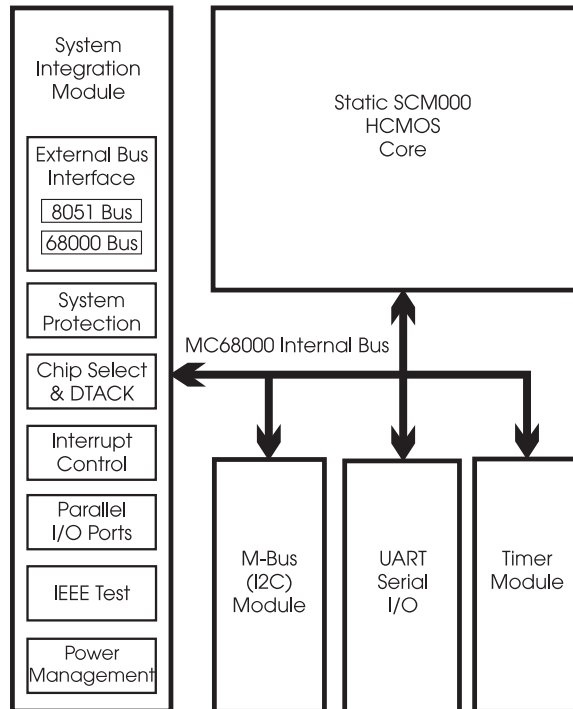
clocks to the timers and the I²C bus (M-Bus) section if they are not required. The UART clock can also be turned off. It can be turned back on automatically if data is received in the receiver buffer.

This section of the manual describes the Micro-Module hardware and gives users all the information required to configure the board and the use the expansion buses to add other peripherals. Many of the features of the CPU are described although more detailed information can be found in the manufacturers user manuals, details of which can be found in the appendix.

2 The MC68307 Micro-Controller

The MC68307 is an advanced 32/16/8 bit micro-controller based on the industry standard 68000 microprocessor. The MC68307 integrates the 68000 CPU core with a 68681 type serial port, a timer module, an I²C bus (M-Bus) module, some digital I/O lines and bus interface logic including up to seven user configurable select lines, dynamic bus sizing and up to 9 interrupt inputs, one of which is non-maskable.

Figure 1 Block Diagram of MC68307



2.1 The 68000 CPU

The CPU contained within the MC68307 is a version of the standard 68000 with full software compatibility to other members of the 68000 family. The MC68307 uses the 68EC000 core which is optimised for use in embedded controllers. This version of the 68000 does not support E cycles for 6800 type peripherals. For a full list of features for the version of the 68000 CPU contained on this board see the manufacturers data sheet on the 68EC000, details in the appendix.

For those that have used the 68000 processor family before skip to section 2.1.3 (Exception Processing).

2.1.1 What is the Advantage of the 68000?

The 68000 is a very powerful 16-bit processor. It will operate at speeds of up to 16 MHz where most 8-bit machines will only operate at speeds of up to 6 MHz. As the 68000 is a 16-bit machine it is able to access 16-bits at a time where the 8-bit machine would have to do two access cycles to get the same amount of data. This will result in faster instruction cycles.

The internal architecture is very flexible with 15 32-bit general purpose registers as opposed to the usual two or three that you get with 8-bit machines. The 68000 directly address 16 M-bytes of memory - you will not be able to use this up very quickly!. All this memory space is available for use directly - you do not need fixes to get round limits imposed by the hardware or operating system such as DOS and the 8086 family of processors.

The interrupt structure is equally clean. The peripheral generates an interrupt which can given a priority level between 1 and 7. The processor will execute the service routine associated with the highest level interrupt pending at any time.

The 68000 is ideal for using with high level languages such as C, C++, Modula-2 and PASCAL. A wide range of operating systems are available for the 68000 including UNIX and Microwares OS9. Cambridge Microprocessor

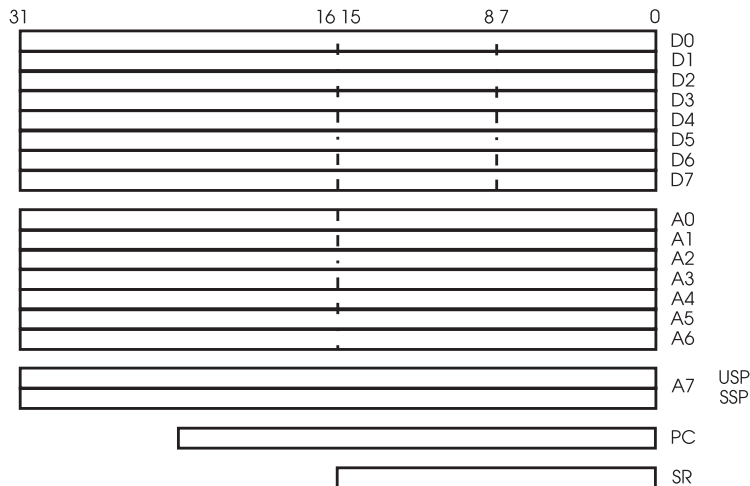
Systems have our own operating system called Minos which can be purchased for all our 68000 based products.

The 68000 is the base model in a whole range of processors that are upwardly object code compatible. Other versions include the 68008 which has an 8-bit data bus instead of the 16-bit data bus of the 68000. The 68010 supports virtual memory and has enhanced instruction execution timing. The 68010 is pin compatible with the 68000 offering a very simple upgrade for people using the standard 68000 microprocessor. The 68020 is a full 32-bit machine offering up to 4 G-bytes of address space. Other members of the family are the 68030 and the 68040, each having more advanced additions to their instruction set but still object code compatible with the older 68000 code.

2.1.2 The 68000 programming model

The 68000 contains sixteen 32-bit registers and a 32-bit program counter. The first eight registers are used as data registers for either 8-bit, 16-bit or 32-bit operations. The next eight registers are the address registers and the user stack pointer. These can be used as software stack pointers and base address registers when accessing I/O devices. There is also a condition code register which contains system information such as the overflow and carry bits. The 68000 instruction set contains 56 instruction types and 14 addressing modes. For details on the addressing modes and instruction set consult one of the many books on the market about the 68000 Assembly language programming. Some of the books we would recommend are listed in the appendix.

Figure 2 Programming Model of the 68000



2.1.3 Exception Processing

The exception processing state is used with interrupts, trap instructions and tracing. The exception can be generated internally by executing an instruction or externally by generating an interrupt, a reset or a bus error. A table of the exception vectors for the MC68307 is shown below. Unassigned or reserved vectors should never be used. All exception vectors are accessed in Supervisor Data mode except the reset vector which is accessed in Supervisor Program mode.

Vector	Address	Space	Description
0	000	SP	Reset : Initial SSP
1	004	SP	Reset : Initial PC
2	008	SD	Bus Error
3	00C	SD	Address Error
4	010	SD	Illegal Instruction
5	014	SD	Zero Divide

Vector	Address	Space	Description
6	018	SD	CHK Instruction
7	01C	SD	TRAPV Instruction
8	020	SD	Privilege Violation
9	024	SD	Trace
10	028	SD	Line 1010 Emulator
11	02C	SD	Line 1111 Emulator
12	030	SD	(Unassigned, Reserved)
13	034	SD	(Unassigned, Reserved)
14	038	SD	(Unassigned, Reserved)
15	03C	SD	Unitialised Interrupt Vector
16 - 23	040-05C	SD	(Unassigned, Reserved)
24	060	SD	Spurious Interrupt by BERR
25	064	SD	(Unassigned, Reserved)
26	068	SD	(Unassigned, Reserved)
27	06C	SD	(Unassigned, Reserved)
28	070	SD	(Unassigned, Reserved)
29	074	SD	(Unassigned, Reserved)
30	078	SD	(Unassigned, Reserved)
31	07C	SD	(Unassigned, Reserved)
32-47	080 - 0BC	SD	TRAP Instruction Vectors
48-59	0C0 - 0EE	SD	(Unassigned, Reserved)
60 - 63	0F0 - 0FC	SD	Base address & System Configuration
64 - 255	100 - 3FC	SD	User Interrupt Vectors

SD Supervisor Data Space

SP Supervisor Program Space

2.1.4 Interrupt Controller

The interrupt controller on the MC68307 is used to handle all the interrupt sources to the processor. Interrupts can be generated by the on chip peripherals or the general purpose interrupt inputs (Port B 8 to 15). The interrupt controller will receive the interrupt request from the peripheral, it will then prioritise it before passing the highest priority request valid at any time on to the processor. When the processor responds to the interrupt request with an interrupt acknowledge cycle the interrupt controller will provide the correct vector number for the peripheral generating the interrupt.

There are nine external interrupt sources. The first, and the one with the highest priority, is IRQ7. This signal always has a priority level of 7 and it will always be dealt with first by the processor. The other 8 interrupt sources are the INT1 to INT8 pins. Each of these sources can be programmed to generate any level interrupt request. The interrupt controller includes a daisy chain decoder for when there is more than one interrupt pending for a particular level. In this chain the IRQ7 input is the highest priority, then comes INT1 to INT8, followed by Timer 1, Timer 2, UART and the I²C bus (M-Bus) with the least priority. For a description of how to set up the interrupts see section 5

2.1.5 Interrupt Vector Generation

The interrupt controller passes the highest priority interrupt request to the processor. The processor responds with an IACK signal to receive a vector number. The vector number is used by the processor to access the interrupt service routine. The interrupt controller will provide the vector for the highest priority interrupt source currently pending. The vector is made up of two four bit parts. The most significant part is provided by the Peripheral Interrupt Vector Register (PIVR). This will select a block of memory to use for the vector table. The least significant four bits are encoded by the interrupt controller depending on the source. The table below gives the vector numbers for the different interrupt sources.

Vector	Interrupt Source
PIVR+\$0	Spurious Interrupt
PIVR+\$1	IRQ7 Input
PIVR+\$2	INT1 Input
PIVR+\$3	INT2 Input
PIVR+\$4	INT3 Input
PIVR+\$5	INT4 Input
PIVR+\$6	INT5 Input
PIVR+\$7	INT6 Input
PIVR+\$8	INT7 Input
PIVR+\$9	INT8 Input
PIVR+\$A	Timer 1
PIVR+\$B	Timer 2
PIVR+\$C	UART Module
PIVR+\$D	M-Bus Module

To calculate the memory location for the service routine the value contained in the Programmable Interrupt Vector Register PIVR is added to the interrupt source vector number. In Minos systems the PIVR register contains the value \$40. This value is then multiplied by four to get a 24-bit address. This address is the start address of the interrupt service routine. As an example if timer 1 generates an interrupt the vector number will be \$4A. When this is multiplied by four to get an address we end up with \$128. When the 32-bit value at this address is read as an address it will point to a location in the EPROM. As this can not contain the service routine this address contains another address which is in the RAM. Eventually the processor will get to this address to execute the interrupt service routine.

2.1.6 68000 Function Codes

The 68000 has three output pins called FC0, FC1, FC2. These signals are used in decoding logic to determine the mode and the cycle type. The 68000 has two modes of accessing data, the supervisor mode and the user mode. When the processor is running in supervisor state it has access to all resources on the system and all instructions can be executed. All operating system functions and any exception processing are done in the supervisor mode. In user mode some instructions are not allowed. Programs are run in user mode to prevent them from accessing areas of the system that they should not modify such as system configuration registers.

On the MC68307 the function codes are not available externally. They are however available internally for use by the address compare logic for the chip selects, allowing access to peripherals and memory in particular modes only. The table below describes the function codes and the meaning of each code.

FC2	FC1	FC0	Cycle Type
0	0	0	Undefined, Reserved
0	0	1	User Data
0	1	0	User Program
0	1	1	Undefined, Reserved
1	0	0	Undefined, Reserved
1	0	1	Supervisor Data
1	1	0	Supervisor Program
1	1	1	Interrupt Acknowledge

The Interrupt Acknowledge function code should never be programmed into the chip select registers on the MC68307.

2.2 The Bus Interface

The part name given to the MC68307 is the Integrated Multiple-Bus Processor or IMBP for short. This section handles all accesses to the bus, whether it is the 68000 expansion bus, 8051 bus or the I²C bus (M-Bus) serial bus. Also contained in this section is the chip select logic, the DTACK delay generator and the interrupt handler. The memory map used in this section is programmed into the operating system and so the addresses given are valid for all Micro-Module boards provided with an operating system. If the operating system is not used then the memory map is entirely up to the user. EPROM must always be present at address \$000000 to provide the reset vector. For full details of setting up the memory map see section 5 Registers.

2.2.1 Main Memory Map

The main memory map includes all the memory address space available on the Micro-Module, a total of 16 M-bytes. The peripheral address strobe (PAS) is selected to be in the same place as it is on our other processor cards to allow all Module Bus peripheral boards to be used with the Micro-Module.

	Size	Start Address	End Address
EPROM	1 M-byte	\$000000	\$0FFFFFF
Local I/O	4 k-byte	\$100000	\$100FFF
PAS (CS2)	32 k-byte	\$128000	\$12FFFF
RAM	512 k-byte	\$180000	\$1FFFFFF
Spare	14 M-byte	\$200000	\$FFFFFF

2.2.2 Local Memory Map

The local memory map contains all the on-board peripherals. The address of these devices is determined by the setting of the Module Base Address Register. This area takes up 4 K-bytes from the main memory map. For Minos systems the base address of these peripherals is \$100000 (as below).

	Start Address	End Address
Port A	\$100010	\$100015
Port B	\$100016	\$10001B
Interrupts	\$100020	\$100027
UART	\$100100	\$10011F
Timers	\$100120	\$100139
M-Bus (I ² C)	\$100140	\$100149

2.2.3 The 68000 Bus

2.2.3.1 User Chip Selects

The MC68307 will generate four chip selects to decode the memory map. For each of the chip selects the user can define an internal DTACK generation after a defined number of wait states. This eliminates the need for DTACK generation using external counters and gates. Each of the chip selects can individually programmed for an 8-bit bus or a 16-bit bus width (see Section 2.2.3.2 Dynamic Bus Sizing). There are also facilities to allow the each chip select line to only be valid for read or write accesses or for particular function codes.

The first chip select is CS0 which the Micro-Module uses to select the EPROM on the board. This chip select is programmed to operate in the 1 M-byte space between address \$000000 and \$0FFFFFF on Minos systems. It is an 8-bit bus access as there is only a single EPROM on the board. The DTACK that is required to end the EPROM access cycle is generated

internally by the interface logic with no wait states. This allows 350ns EPROMs to be used in the EPROM socket.

The second chip select is CS1 which the Micro-Module uses to chip select the static RAM on the board. This chip select is programmed to operate in the 512 K-byte space between \$180000 and \$1FFFFFF on Minos systems. This is also an 8-bit bus cycle and the DTACK is again generated in the controller with no wait states.

The third chip select, CS2, is different from the other chip selects in that it can be divided up into four chip selects. The Micro-Module is programmed to use just the one chip select leaving the other chip selects as general purpose I/O lines. This chip select decodes the 32 k-bytes between \$128000 and \$12FFFF in the memory map using the 16-bit bus mode. DTACK has to be asserted externally when addressing a device in this range. This area is used on the Module Bus system as the peripheral address space or PAS. This select line is positioned here to enable the Micro-Module to operate with our range of Module Bus compatible products.

If the target does not require the use of the Module Bus PAS signal CS2 can be divided up into four 16 k-byte sections each with their own chip select by setting the Enable Peripheral Chip Select (EPCS) bit in the System Control Register (SCR). The Port A registers will also need to be set up as CS2B to CS2D are an alternative function of some of the Port A lines. If used in this mode the size of CS2 must be set to 64 k-bytes. CS2A, which is the same pin as CS2, will decode the first 16 k-bytes, CS2B (Port A0) the next 16 k-bytes, CS2C (Port A1) the next block and finally CS2D (Port A2) decodes the last 16 k-bytes. These chip select lines can then be used to select peripherals placed on the prototyping area. If these four signals are used in this way then CS2B, CS2C and CS2D will each require a pull up resistor. Following a cold reset or power on as all the port A channels power up as general purpose inputs. If these signals are not pulled up the devices decoded by them would all be selected temporarily on power up. CS2A does not require the pull up as a resistor is already fitted to the Micro-Module.

The fourth chip select line, CS3, can either be used as an 8051 device chip select or as an extra 68000 bus chip select. To use this chip select as an 8051 cycle the Enable 8051 (E8051) bit and the Bus Width 3 (BUSW3) bit must be set in the System Control Register (SCR). When used as a 68000 chip select it can be used with either the 8-bit bus or the 16-bit bus mode.

This chip select is not set up on the Micro-Module. This chip select line can be used for adding peripherals or memory to the board using the prototyping area.

For each of the four chip selects there are two registers in the micro-controller. The first register is called the Base Register (e.g. BR0, BR1 etc.). This register contains the base address of the chip select range, the address space function code, read or write only bit and an enable bit to turn the chip select line on. The second register is the Option Register (e.g. OR0, OR1 etc.). This register contains information about the DTACK generator, a read/write mask bit and an address mask which is used to determine the size of the chip select. A bit determining the width of the data bus for each chip select is contained in the System Control Register (SCR). More details can be found on the registers in Section 5 Registers.

2.2.3.2 Dynamic Bus Sizing

The IMBP adds the facility of dynamic bus sizing to the standard 68000 expansion bus. Each of the chip selects that the MC68307 generates can operate with an 8-bit data bus or the full 16-bit data bus. The width of the data bus is set using the BUSW bit for each of the chip selects in the System Control Register (SCR). The BUW0 and BUSW1 should never be changed on the Micro-Module as they are used to chip select the EPROM and RAM on the board. If the board is used with Module Bus extensions then BUSW2 should not be changed either as all 8-bit peripherals on this bus use the lower data bus and odd addresses as used on the standard 68000.

If the processor is in 8-bit mode then only the upper 8 data lines are valid and DS1 is asserted. Valid address lines are A0 to A23. If the processor is using byte operations there will not be any speed difference when using the 8-bit mode. If word and long word operations are used then the performance will be reduced to half that if it were using the full 16-bit bus.

In 16-bit bus mode address lines A1 to A23 are valid and the whole of the data bus is used along with both data strobes, DS0 and DS1.

2.2.4 The 8051 bus

The MC68307 can interface directly with 8051 peripherals using its own 8051 expansion bus. The 8051 bus is enabled using CS3 and setting the E8051 bit and clearing the BUSW3 bit in the System Control Register (SCR).

The 8051 bus uses a multiplexed address and data bus. When an address within the range defined for the 8051 peripherals is selected the processor places the address on the address bus and a signal called Address Latch Enable (ALE) goes low. This signal should be used to latch the lower eight address lines. Once the address lines are latched the 8-bit data can be placed on the lower eight address lines either by the processor or the device being accessed depending on whether it is a read cycle (RD goes low) or a write cycle (WR goes low). For customers wishing to use 8051 peripherals the three control lines, ALE, RD and WR, are provided on the prototyping area only. They are not available on the Module Bus expansion connector. Data will appear on address lines A0 to A7 when ALE is low.

Figures 3 and 4 show timing diagrams for 8051 read and write cycles.

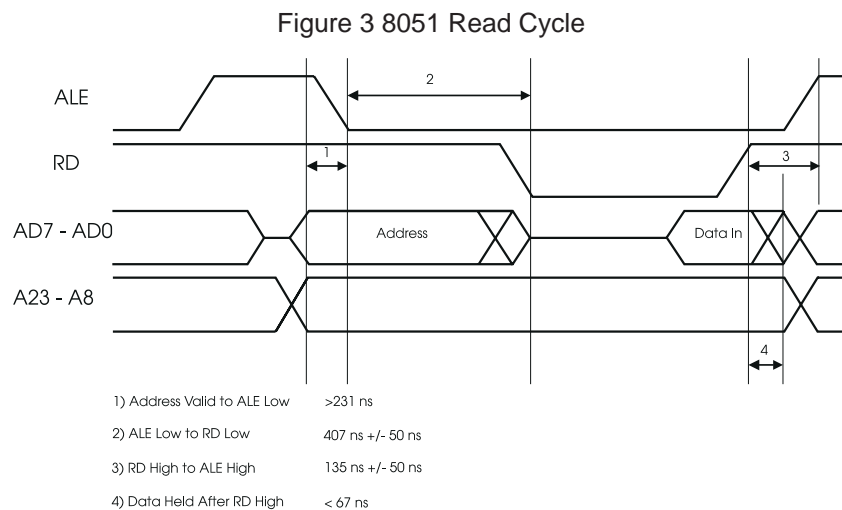
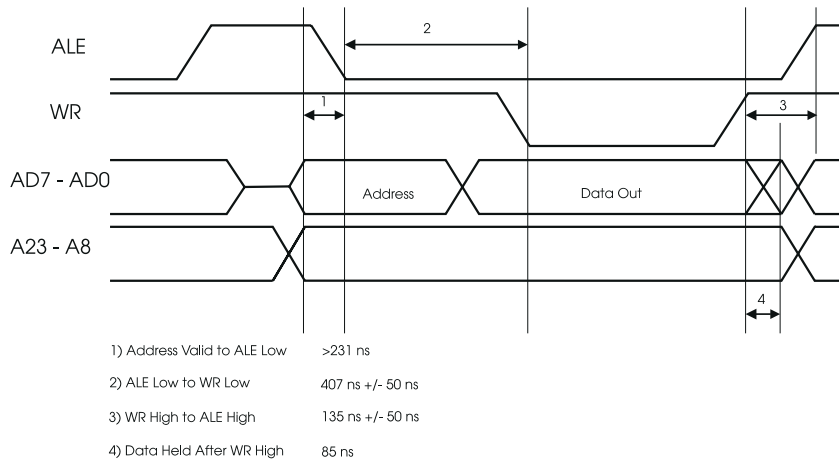


Figure 4 8051 Write Cycle



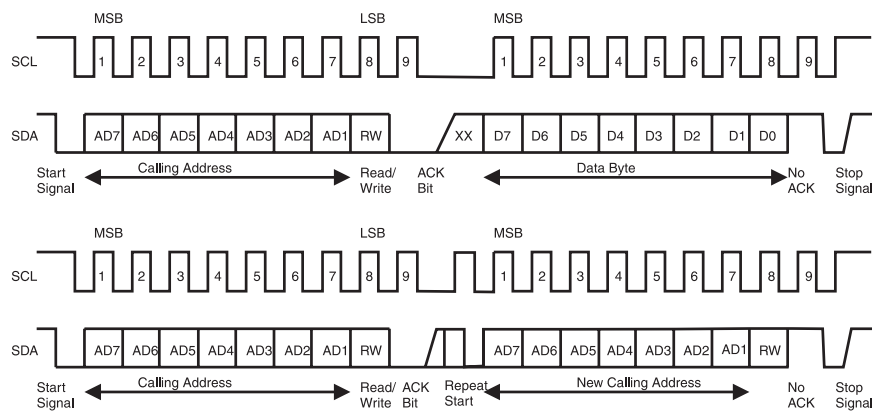
2.2.5 The I²C Bus (M-Bus)

The I²C bus (M-Bus) is a two wire, bi-directional serial bus which provides a simple method of exchanging data between devices. As the bus uses just two wires there is very little interconnection between the different devices on the bus. The bus is designed to operate at rates of 100 k-bits per second with maximum loading in order to meet the specification of the I²C bus. The MC68307 is capable of operating at greater baud rates than this with reduced bus loading. The maximum number of devices that can be connected to the I²C bus (M-Bus) is limited by the maximum bus capacitance of 400pF. The I²C bus (M-Bus) is a multi-master bus with built in arbitration and collision detection in case more than one master tries to control the bus at a time.

The I²C bus (M-Bus) uses just two wires, Serial Data Line (SDA) and Serial Clock Line (SCL), to perform all functions on the bus. All devices on the bus are connected to these two wires with open collector I/O pins. Communication on the bus is made up of four parts, the start signal, the slave address transmission, data transfer and the stop condition.

The start condition, sent out by a master, will initiate a transfer on the bus if it is not active at that time. The bus is inactive if both the SCL and SDA lines are high. The start signal is defined as a high to low transition on the SDA line while the SCL line is high. This will wake up all the slaves. The next piece of information sent is the address of the slave to be accessed. This is a seven bit address with a RW bit at the end. The RW bit will inform the slave what type of transfer is required. The slave that is addressed will send back an acknowledge bit to the master. The acknowledge signal is sent by pulling the SDA line low on the 9th clock. Once the slave has responded the data transfer can take place one byte at a time. The data word is 8 bits long and can only be changed while the SCL line is low. There is one clock pulse on the SCL line for every bit of data. The most significant bit is transferred first. At the end of each byte the receiver will acknowledge receipt of the byte by pulling SDA low on the 9th clock. If the receiver does not acknowledge the receipt of the data it signifies the end of the transfer. The slave must then release the SDA line so that the master can issue a stop instruction by making a low to high transition on the SDA line while the SCL line is high.

Figure 5 M-Bus Transmission Signals



A wide range of peripherals are available to be connected to the I²C bus (M-Bus). Philips Components manufacture devices for digital I/O, analogue I/O, real time clocks, LCD drivers, video decoders, voice synthesisers and

tone generators and many more. Memory devices are also available, this type of interface being particularly useful for EEPROMs. It is also possible to network processors using the I²C bus (M-Bus). An example of this is given in the application section of the 68307 user manual from Motorola.

The MC68307 can operate either as a master or a slave on the I²C bus (M-Bus). The slave address for the MC68307 is programmed into the I²C bus (M-Bus) Address Register so that it can be accessed by other masters on the I²C bus (M-Bus). The routines supplied with the board for use with Minos systems only support the 68307 as a bus master.

3 On Board Peripherals

3.1 The UART

The MC68307 basically contains a 68681 UART. The difference between the 68681 and the serial port in the micro-controller is that only one of the serial ports are used, there are no digital I/O ports and the timer can not be used for general use. The timer is used for generating the baud rate clock only.

The UART can be used in four different modes, normal mode, automatic echo, local loop back and remote loop back. In automatic echo mode all characters that are received by the UART are automatically retransmitted back to the transmitter. In local loop back mode the transmitter output is internally connected to the receiver input causing the UART to receive every thing that it sends. This mode is useful for checking that the UART is working correctly.

The UART can be configured to operate with 5, 6, 7, or 8 bits per character, even, odd or no parity, between 1 and 2 stop bits at standard or non standard baud rates up to 5 Mbps. It is a full duplex asynchronous transmitter and receiver. The UART can generate interrupts to improve the flow of data between systems.

The Micro-Module is available with two different types of serial communication. There is a RS-232 serial port and a RS-485 serial port. Both these types of serial require different functions from the driver code. The driver code for both serial types are available and details of how to modify boards from RS-232 to RS-485 is contained in section 12.1.

The RS-485 is a differential multi drop network port with common transmit and receive signals. It is ideal for long distance communication in noisy environments. When the serial port is using the RS-485 driver, the transmitter is automatically enabled when data appears in the transmit buffer. The transmitter is disabled when the last character is sent. The receiver buffer is turned off when the transmitter is enabled to prevent the serial port receiving all that is sent out. The receiver is enabled when nothing is being transmitted to allow the Micro-Module to listen to the line. When operating in this mode the serial port uses three of the Port B I/O lines, TXD, RXD and RTS.

The RS-232 serial port is configured to work with a P.C. computer. The micro-controller uses TTL level signals which are converted up to RS-232 levels by a buffer chip with built in charge pumps. A 9-pin D type connector can be connected to the serial port on the Micro-Module using a 10 way IDC cable and connector. The Minos driver for the serial port uses RTS and CTS handshaking for flow control. Data is made up of eight bits per character with one stop bit and no parity. There are internal buffers for both the receiver and the transmitter. The serial port uses four of the Port B I/O pins for the TXD, RXD, CTS and RTS signals. Other connections on the 9 pin P.C. connector are not supported on the Micro-Module.

Full details of the UART can be found in section 8 of the 68307 user manual.

3.2 The Timer Module

The Micro-Module has two 16-bit general purpose counter timers and a software watch dog timer. On Minos systems timer 1 is used to generate the 100 Hz ticker module for the task switcher. If Minos is not used then both the timers can be used. Full details of the timers can be found in section 6 of the 68307 user manual.

3.2.1 General Purpose Timers

Each of the general purpose timers consist of a timer mode register, a timer capture register, a timer counter register, a reference register and an event register. The timer mode register also contains a pre-scale value to divide down the clock source. The timers can be used for counting a fixed number of pulses and then interrupting the processor when they reach the reference count. Alternatively they can be used to the measure the time before an event occurs on the timer input pin.

3.2.1.1 Timer Clock Source

The two timers can be clocked from one of three sources.

The first is the main system clock. On the Micro-Module this system clock is running at 7.3728 MHz giving a minimum resolution of 135ns in the timers. When using the main system clock for the counter clock the maximum time

period is 2.28 seconds, when the reference value is set to \$FFFF and the pre-scaler in the mode register is set to 256.

The second source is the system clock divided by 16. This will give a clock frequency of 460.8 kHz, a minimum resolution of 2.17 μ s and a maximum time out period of 36.4 seconds when the reference value is set to \$FFFF.

The third source for the clock is the timer input pin. Using this pin any time period can be set by adjusting the frequency of the source.

3.2.1.2 Timer Modes

The two general purpose timers can operate in three different modes. In the first mode, the counter is incremented on each clock cycle until the value in the current count register matches that in the timer reference register. When the reference is reached the counter value can either be reset to zero and restart counting or the counter can continue to increment. When the reference value is reached an interrupt can be generated to carry out a special operation each time period. The code segment given below sets each timer up in output reference compare mode, outputting a pulse on each channels TOUT pin after the preset period. The interrupts are disabled in this example.

* Masks and bit patterns for timers

rst	equ	\$1	enable timer
iclk01	equ	\$2	use system clock
iclk10	equ	\$4	use system clock divide by 16
iclk11	equ	\$6	use Tin as clock
frr	equ	\$8	restart on reference compare
ori	equ	\$10	enable interrupt on reference compare
om	equ	\$20	toggle Tout
ce01	equ	\$40	capture on +ve edge of clock & interrupt
ce10	equ	\$60	capture on -ve edge of clock & interrupt
ce11	equ	\$80	capture on both edges of clock & interrupt
ref	equ	\$2	isolate TER REF bit
cap	equ	\$1	isolate TER CAP bit

```
or.b    #$18,pacnt(a0)  enable timer output pins
```

* set timers to run independently, active low pulses on Tout, interrupts disabled.

* Tout1 pulse every (2 x 9) CPU clocks, prescaler : 2; reference 9

```
move.w  #8,trr1(a0)      reference value 8
```

```
move.w  #rst + iclk01 + frr + $0100, tmr1(a0)
```

* Tout2 pulse every (3 x 7) CPU clocks, prescaler : 3; reference 7

```
move.w  #6,trr1(a0)      reference value 6
```

```
move.w  #rst + iclk01 + frr + $0200, tmr1(a0)
```

The second mode allows the timer/counters to be used to count events on an input signal. The code segment below shows how to set the timer up to operate as a counter, generating an interrupt when four pulses have been received.

* Set up Timer 1 to count events on T1in

```
or.w    #$0040,pbcnt(a0) Tin1 pin dedicated
```

* Set up Tin1 as clock source, interrupts generated every 4 clocks

* counter reset after reference reached and an interrupt generated

* Tout1 pulse active low, capture interrupts disabled.

```
move.w  #3,trr1(a0)      interrupt every 4 clocks
```

```
move.w  #rst + iclk11 + frr + ori, tmr1(a0)
```

```
stop    $2000            wait for interrupt
```

```
int_t1  andi.b  #ref,ter1(a0)  clear the interrupt
```

(service the interrupt)

```
rte
```

The third mode allows the timers to measure the time between events. The timer input pin can be set up to generate an interrupt on a change of state. When the timer input pin changes state, the current value of the timer is latched into the Timer Capture Register. The timers can be set up to capture the timer value on a falling edge, a rising edge or any transition on the timer input pin. Following a capture event an interrupt can be generated to service the timer. Before the interrupt request can be cleared the CAP bit must be set to 1 in the Timer Event Register. The following piece of code will set timer 2 up in this mode.

* Enable timer 2 input pin in port B control register

```
or.w    #$0080,pbcnt(a0) Tin2 dedicated pin
```

* Capture on +ve edge of Tin2, use system clock as clock source

* counter is free running, interrupts disabled for reference

* interrupts enabled for capture.

```
int_t2  move.w  #rst + iclk01 + ce01,tmr2(a0)
        andi.b  #cap,ter2(a0)    clear CAP event
        move.w  tcr2(a0),d5      read value of counter into d5
        rte
```

3.2.2 Watch Dog Timer

Inside the micro-controller is a third timer. This timer is special in that when it reaches its reference count value the system is reset. This is very useful to ensure that the software is running and has not just bombed. In order to stop the system being reset a program must write to a location in the micro-controller on a regular basis. This feature is called the watch dog because it keeps an eye on a system to ensure that it is running.

When the Micro-Module is first powered up the watch dog timer is enabled and set to go off approximately every 35 seconds. This time period can be reduced by changing the value in the Watch dog Reference Register. Also in the reference register is a bit which allows the watch dog feature to be disabled. For the Micro-Module to run correctly either the watch dog has to be turned off, by clearing bit 0 in the Watch dog Reference Register or typing nodog in the debug monitor, or kicked at least once within the time out period.

The watch dog is kicked by any write cycle to the Watch dog Counter Register. The code below shows how to turn off the watch dog so that it will not go off.

*Turn off the watch dog by clearing the EN bit in the WRR.

```
wrr    equ    $12b          watch dog Reference Register
                                (low byte of 16-bit register)
                                bclr    #0,wrr(a0)    clear EN bit
```

The following routine shows how to kick the watch dog to keep the Micro-Module running all the time.

* Watch dog kicking example

```
wrr    equ    $12A          Watch dog Reference Register
wcr    equ    $12C          Watch dog Control Register

                                move.w  #$0FFF,wrr(a0)  set Watch dog time out to about 0.2s

main   move.w  #$FFFF,d0     set up a delay
loop   nop
                                dbra   d0,loop
                                move.b  #$FF,wrc(a0)    kick the watch dog
                                bra     main
```

3.3 Digital I/O

The Micro-Module has one 8-bit digital I/O port in which each of the 8 lines can generate interrupts. Depending on what other functions are required there can be up to 22 digital I/O lines (or 24 if you remove the serial port buffer from the board!).

The dedicated 8-bit port uses Port B channels 8 to 15. The other 8 channels that make up the 16-bit Port B have alternative functions. The special functions of all the digital I/O lines are listed in the table below. Any channel can be used for its special function by setting the relevant bit in the port control register. If the bit in the control register is 0 then that channel can be used as a general purpose I/O line whose direction depends on the setting in the data direction register. The eight Port A lines can also be used as general purpose I/O lines by setting the relevant bits in the Port A Control register. This will give a maximum 24 digital I/O lines. On Minos systems PB15 is used by interrupt I3, which is used by most of the Module Bus expansion peripheral cards. PB14 is I4 and PB13 is I5. PB8 is connected to a red LED on the board and is a general purpose I/O line.

Port & Channel	Dedicated I/O Function
Port B, 15	INT8 Input
Port B, 14	INT7 Input
Port B, 13	INT6 Input
Port B, 12	INT5 Input
Port B, 11	INT4 Input
Port B, 10	INT3 Input
Port B, 9	INT2 Input
Port B, 8	INT1 Input
Port B, 7	Timer 2 Input
Port B, 6	Timer 1 Input
Port B, 5	CTS
Port B, 4	RTS

Port & Channel	Dedicated I/O Function
Port B, 3	RxD
Port B, 2	TxD
Port B, 1	SDA M-Bus Data Line
Port B, 0	SCL M-Bus Clock Line
Port A, 7	BGACK
Port A, 6	BG
Port A, 5	BR
Port A, 4	Timer 2 Output
Port A, 3	Timer 1 Output
Port A, 2	Chip Select 2D Output
Port A, 1	Chip Select 2C Output
Port A, 0	Chip Select 2B Output

As some of the I/O lines are used for the serial port, an RS-232 buffer is connected to four of the Port B lines. Two of these lines are inputs into the buffer and so, if the serial port is not used, these lines may be used as general purpose I/O lines. The other two lines are, however, outputs from the buffer and so the buffer will be driving these lines. These two lines, RXD (Port B3) and CTS (Port B5), can not be used for general purpose lines unless the serial port buffer (MAX232A) is completely removed from the board.

The following code segment will flash the LED on the Micro-Module. It is written in such a way that the code can be blown into an eprom and run from reset.

```

        dc.l    0
        dc.l    start

mbar   equ     $f2           Module Base register
or1    equ     $46           Ram option register
br1    equ     $44           Ram base register
wrr    equ     $12a         Watchdog reference register
pbdatt equ     $1a           port B data register
pbcnt  equ     $16           port B control register
pbdirt equ     $18           port B direction register

start  move.w   #$0100,mbar   set register base to $100000
        movea.l #$100000,a0   a0 points to register base
        move.w   #$1f00,or1(a0) set RAM size
        move.w   #$0301,br1(a0) set RAM base to $180000

rloop  move.w   #$ff,d0       clear the RAM CS latch
        move.b   $180000,d1
        dbra    d0,rloop

        move.w   #$0,wrr(a0)   turn the watchdog off
        bclr    #$0,pbcnt(a0)  PB8 general purpose I/O
        bset    #$0,pbdirt(a0) PB8 is Output

loop   bchg    #$0,pbdatt(a0)  change state of PB8
        move.w   #$ffff,d0     flash delay loop
wait   nop
        dbra    d0,wait
        bra     loop

```

This page is intentionally left blank

4 Power Saving

4.1 Low Power Modes

The Micro-Module has several ways of reducing its current consumption. Each of the main modules within the micro-controller has got their own clock source. If the module is not used then this clock can be turned off. The table below shows the relevant power saving to be made by turning off each of the peripherals.

Module	Percentage I _{DD}
All Operational	100%
CLKOUT	5%
M-Bus	6%
Timer	7%
UART	15%
CPU	61%
Low-Power Sleep	6%

The modules that this affects are the UART, the Timers and the I²C bus (M-Bus) serial I/O bus. Another way to reduce the power consumption is to turn off the clock output signal. This will stop the back plane oscillator so this signal should never be stopped when using expansion boards.

As the MC68307 is a static type CPU the clock to the processor can be slowed down or even stopped altogether to further reduce the current required to operate the board. By turning off all the peripheral clocks and running the CPU at 28.8 kHz the current consumption can be reduced by about half that when running at full speed. To stop the main system clock under software control the MOS bit in the System Control Register should be set and then a STOP instruction executed to finally turn off the clock. Once the system clock has been turned off the Micro-Module can be started up again by a peripheral generating an interrupt. The interrupt can be generated by an external peripheral, one of the timers after a programmed length of time or by the UART when it receives a character. If the timer or the I²C bus (M-Bus) module is required to generate the interrupt to wake up the processor their clocks must not be turned off. The UART clock can be turned

off as long as the UART Clock Wake up (UACW) bit is set in the System Control Register.

To run the Micro-Module at a reduced rate the Low Power Enable (LPEN) bit should be set in the System Control Register. When this bit is set the clock speed to the processor is determined by the setting of the Low Power Clock Divider (LPCD) bits in the same register. The table below shows the CPU clock speed for each setting of the LPCD bits

LPCD2	LPCD1	LPCD0	Ratio	Approx. Clock Speed
0	0	0	2	3.6864 MHz
0	0	1	4	1.8432 MHz
0	1	0	8	921.6 kHz
0	1	1	16	460.8 kHz
1	0	0	32	230.4 kHz
1	0	1	64	115.2 kHz
1	1	0	128	57.6 kHz
1	1	1	256	28.8 kHz

4.2 Example

The code shown below can be used to blow an EPROM which operates the Micro-Module at 28.8 kHz with all the auxiliary clocks turned off. This code will reduce the Micro-Modules current consumption from about 60 mA to less than 30 mA.

```
*****
* Low Power Mode
* This code should be blown into an EPROM and run from switch on.
*
* The code will flash an LED on the Micro-Module to show that the
* board is still working. The watch dog is kicked on a regular basis to
* prevent it from timing out. The LED is connected to Port B 8.
*
* Power consumption is reduced by:
* Turning off the Timer Module Clock
* Turning off the UART Module Clock
```

- * Turning off the I²C bus (M-Bus) Module Clock
- * Turning off the CLKOUT signal from the controller
- * Clocking the CPU at 28.8 kHz
- *

```

        dc.l    0
        dc.l    start

        org    $400          start of the code in EPROM

mbar    equ    $f2          module base address register
src     equ    $f4          system control register
br0     equ    $40          base address for ROM offset
or0     equ    $42          ROM option register offset
br1     equ    $44          base address for RAM offset
or1     equ    $46          RAM option register offset
pbddr   equ    $18          Port B direction register offset
pbdat   equ    $1a          Port B data register offset
wrc     equ    $12c         watch dog register offset

start   move.w  #$0100,mbar  set base register to $100000
        move.l  #$2e777,src   low power modes 28.8 kHz operation
                                etc
        movea.l #$100000,a0   set a0 to hold base address
        move.w  #$c001,br0(a0) set ROM base to $000000
        move.w  #$3e00,or0(a0) 1 Wait State, 1 M-byte
        move.w  #$0301,br1(a0) set RAM base to $180000
        move.w  #$3f00,or1(a0) 1 Wait State, 512 K-byte
        bset   #8,pbddr(a0)    set Port B 8 to be output

loop    bchg   #8,pbdat(a0)    change value of bit 8
        move.b  #$ff,wrc(a0)   kick the watch dog
        move.l  #$ff,d0        set up delay time
dly     nop                    waste some time

```

Micro-Module Hardware

dbra d0,dly
bra.s loop

5 Registers

5.1 Introduction

This section contains a list of all the registers in the 68307. It gives a brief description of the function of the register. More information can be found in the 68307 user manual (page numbers for the 68307 user manual are given in the tables below).

System Configuration Registers

Address	Register Name		Page
\$0000F0	Reserved		
\$0000F2	Module Base Address Register	MBAR	5-22
\$0000F4	System Control Register high word	SCRh	5-23
\$0000F6	System Control Register low word	SCRI	5-23
\$0000F8	Reserved		
\$0000FA	Reserved		
\$0000FC	Reserved		
\$0000FE	Reserved		

Note-The addresses given below depend on the base address in MBAR

External Bus Interface Registers

Address	Register Name		Page
\$100011	Port A Control Register	PACNT	5-34
\$100013	Port A Data Direction Register	PADDR	5-35
\$100015	Port A Data Register	PADAT	5-35
\$100016	Port B Control Register	PBCNT	5-36
\$100018	Port B Data Direction Register	PBDDR	5-36
\$10001A	Port B Data Register	PBDAT	5-37

Interrupt Control Registers

Address	Register Name		Page
\$100020	Latched Interrupt Control Register 1	LICR1	5-38
\$100022	Latched Interrupt Control Register 2	LICR2	5-38
\$100024	Peripheral Interrupt Control Register	PICR	5-39
\$100027	Programmable Interrupt Vector Register	PIVR	5-40

Chip Select Registers

Address	Register Name		Page
\$100040	Base Register 0	BR0	5-30
\$100042	Option Register 0	OR0	5-32
\$100044	Base Register 1	BR1	5-30
\$100046	Option Register 1	OR1	5-32
\$100048	Base Register 2	BR2	5-30
\$10004A	Option Register 2	OR2	5-32
\$10004C	Base Register 3	BR3	5-30
\$10004E	Option Register 3	OR3	5-32

UART Registers

Address	Register Name		Page
\$100101	UART Mode Register 1	UMR1	8-15
\$100101	UART Mode Register 2	UMR2	8-17
\$100103	UART Status Register (read)	USR	8-19
\$100103	UART Clock Select Register (write)	UCSR	8-21
\$100105	UART Command Register (write)	UCR	8-23
\$100107	UART Receiver Buffer (read)	URB	8-25
\$100107	UART Transmitter Buffer (write)	UTB	8-25
\$100109	UART Input Port Change Register (read)	UIPCR	8-26

Address	Register Name		Page
\$100109	UART Auxiliary Control Register (write)	UACR	5-26
\$10010B	UART Interrupt Status Register (read)	UISR	5-27
\$10010B	UART Interrupt Mask Register (write)	UIMR	5-28
\$10010D	UART Baud Rate Pre-scaler high byte	UBG1	5-29
\$10010F	UART Baud Rate Pre-scaler low byte	UBG2	5-29
\$100119	UART Interrupt Vector Register	UIVR	5-29
\$10011B	UART Input Port Register (read)	UIP	5-29
\$10011D	UART Output Port Data Register 1 (write)	UOP1	5-30
\$10011F	UART Output Port Data Register 0 (write)	UOP0	5-30

Timer Module Registers

Address	Register Name		Page
\$100120	Timer 1 Mode Register	TMR1	6-4
\$100122	Timer 1 Reference Register	TRR1	6-5
\$100124	Timer 1 Capture Register	TCR1	6-5
\$100126	Timer 1 Counter	TCN1	6-5
\$100129	Timer 1 Event Register	TER1	6-6
\$10012A	Watch dog Reference Register	WRR	6-7
\$10012C	Watch dog Counter Register	WCR	6-7
\$100130	Timer 2 Mode Register	TMR2	6-4
\$100132	Timer 2 Reference Register	TRR2	6-5
\$100134	Timer 2 Capture Register	TCR2	6-5
\$100136	Timer 2 Counter	TCN2	6-5
\$100139	Timer 2 Event Register	TER2	6-6

I²C bus (M-Bus) Module Registers

Address	Register Name		Page
\$100141	M-Bus Address Register	MADR	7-6
\$100143	M-Bus Frequency Divider	MFDR	7-6
\$100145	M-Bus Control Register	MBCR	7-7
\$100147	M-Bus Status Register	MBSR	7-9
\$100149	M-Bus Data I/O Register	MBDR	7-10

Please note that the addresses given refer to the most significant byte of the register. For 32 bit registers this will be bits 31 to 24, for 16 bit registers this will be bits 15 to 8.

5.2 System Configuration Registers

Module Base Address Register

\$0000F2

FC2	FC1	FC0	CFC	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

This register contains the address and function codes to enable the CPU to address the on chip peripherals. It can only be accessed in supervisor mode. The upper 12 address lines are used to decode the position of the registers in the memory map. By setting the compare function code bit the on chip peripherals can only be accessed in the defined function codes. The function code for the 68000 interrupt acknowledge cycle should never be programmed into this register. For full details see page 5-22 in the 68307 user manual.

- FC2 - 0 Function Codes
- CFC Compare Function Codes - If set then function codes are used
- A23 -12 Defines the base address for the registers

On a Minos system this register is programmed with the value \$0100 to give the base address \$100000, function codes are ignored.

System Control Register (SCR)

\$0000F4

ADC	WPV	HWT		RS1	RS0			ADCE	WPVE	EPCS	E8051	BUS W0	BUS W1	BUS W2	BUS W3
HWDE	HW2	HW1	HW0	UACW	UACD	TMCD	MBCD	LPEN	CDEN	CKD	EBU SW		CD2	CD1	CD0

This 32-bit register describes how the system is set up. It can be read or written at any time, in either byte, word or long word accesses. The top 8 bits, when read, will reveal the status of the system. The remaining 24-bits are used to configure the micro-controller. This register can only be accessed in supervisor mode. After a reset the status bits are set to 0. An occurrence of the relevant event will set the bit. A bit can be cleared by writing a 1 to the appropriate bit. In the 68307 user manual (page 5-23) most of the time periods given are for a clock speed of 16.67 MHz. The Micro-Module is clocked at 7.3728 MHz and so times should be recalculated accordingly. Note that the 68000 stores the most significant byte first i.e. the byte at address \$F4 will access the first byte, \$F5 accesses the second byte (ADCE - BUSW3), \$F6 byte HWDE to MBCD etc.

Status bits

ADC Address Decode Conflict
WPV Write Protection Violation
HWT Hardware Watch dog Time out (Bus Error)
RS1 - 0 Reset Source- gives the cause of the last reset

Control bits

ADCE Address Decode Conflict Enable- generates BERR
WPVE Write Protect Violation Enable - generates BERR
EPCS Enable Peripheral Chip Selects - divides CS2 into four
E8051 Enable 8051 bus
BUSW0 Bus Width 0 - size of data bus for CS0
BUSW1 Bus Width 1 - size of data bus for CS1
BUSW2 Bus Width 2 - size of data bus for CS2
BUSW3 Bus Width 3 - size of data bus for CS3
HWDE Hardware Watch dog Enable - enables the BERR timer
HW2 - 0 Hardware Watch dog Count - sets the time period for BERR
UACW UART Clock Wake up - allows UART clock to start when a character is received
UACD UART Clock Disable - turns the UART module clock off
TMCD Timer Clock Disable - turns the Timer module clock off
MBCD I²C bus (M-Bus) Clock Disable - turns the M-Bus clock off
LPEN Low Power Enable - enables low power modes
CKD Clock Output Disable - turns the CLKOUT signal off
EBUSW Enable Bus Width - determines bus width for non chip-select devices
LPCD2 -0 Low Power Clock Divider set the clock speed if LPEN is set

5.3 I/O Port Control Registers

Port A Control Register (PACNT)

\$100011

CA7	CA6	CA5	CA4	CA3	CA2	CA1	CA0
-----	-----	-----	-----	-----	-----	-----	-----

This register is used to specify the function of the I/O port A. Each of the I/O pins making up port A has two functions. They can either be a dedicated special function pin or a general purpose I/O pin. To configure any of the port A pins to function as their special I/O function set the required bit in this register. To enable this port to function as general purpose I/O port, clear the bits in this register. The table on page 3-7 shows the alternative functions for each of the Port A lines. At reset this register is set to 0, enabling all of the port A lines as general purpose I/O lines. Full details can be found on page 5-34 of the 68307 user manual

Port A Data Direction Register (PADDR)

\$100013

DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0
-----	-----	-----	-----	-----	-----	-----	-----

This 8-bit register is used to set the direction of each of the port A lines. By setting a bit in this register the associated port A line is configured as an output. On reset this register is cleared and so all the port A pins will function as inputs. Full details can be found on page 5-35 of the 68307 user manual.

Port A Data Register (PADAT)

\$100015

PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
-----	-----	-----	-----	-----	-----	-----	-----

This 8-bit register is used to store the logic states of the port A lines. When the port is configured as outputs a logic zero will be output on all the port A lines unless the data register is configured before the direction is changed. In output mode this register stores the value output on the port a lines, in input mode this register is read to input the value of the port A lines. At reset all the bits in this register are 0. Full details of this register can be found on page 5-35 of the 68307 user manual.

Port B Control Register (PBCNT)

\$100016

CB15	CB14	CB13	CB12	CB11	CB10	CB9	CB8	CB7	CB6	CB5	CB4	CB3	CB2	CB1	CB0
------	------	------	------	------	------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

This 16-bit register is used to specify the function of the I/O port B. Each of the I/O pins making up port B has two functions. They can either be a dedicated special function pin or a general purpose I/O pin. To configure any of the port B pins to function as their special I/O function set the required bit in this register. To enable this port to function as general purpose I/O port, clear the bits in this register. The table on page 3-7 shows the alternative functions for each of the port B lines. At reset this register is set to 0, enabling all of the port B lines as general purpose I/O lines. Full details of this register can be found on page 5-36 of the 68307 user manual.

Port B Data Direction Register (PBDDR)

\$100018

DB15	DB14	DB13	DB12	DB11	DB10	DB9	DB8	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
------	------	------	------	------	------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

This 16-bit register is used to set the direction of each of the port B lines. By setting a bit in this register the associated port B line is configured as an output. On reset this register is cleared and so all the port B pins will function as inputs. Further details on this register can be found on page 5-36 of the 68307 user manual.

Port B Data Register (PBDAT)

\$10001A

PB15	PB14	PB13	PB12	PB11	PB10	PB9	PB8	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
------	------	------	------	------	------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

This 16-bit register is used to store the logic states of the port B lines. When the port is configured as outputs a logic zero will be output on all the port B lines unless the data register is configured before the direction is changed. In output mode this register stores the value output on the port a lines, in input mode this register is read to input the value of the port B lines. At reset all the bits in this register are reset to zero. More details on this register can be found on page 5-37 of the 68307 user manual.

5.4 Interrupt Control Registers

Latched Interrupt Control Register (LICR)

\$100020

PIR1	INT1L2	INT1L1	INT1L0	PIR2	INT2L2	INT2L1	INT2L0	PIR3	INT3L2	INT3L1	INT3L0	PIR4	INT4L2	INT4L1	INT4L0
PIR5	INT5L2	INT5L1	INT5L0	PIR6	INT6L2	INT6L1	INT6L0	PIR7	INT7L2	INT7L1	INT7L0	PIR8	INT8L2	INT8L1	INT8L0

This 32-bit register is used to control the interrupt priorities of the 8 general purpose interrupt inputs (alternative function for Port B8-15). Each of the 8 interrupt lines has four bits assigned to it in these registers. One of the bits is a interrupt reset bit which allows the software to clear a pending interrupt. Writing a 1 to the reset bit will clear the interrupt latch. The other three bits for each input is the priority level between 0 and 7. If the level is programmed as 0 the interrupt is disabled. Level 7 is the highest priority. Full details can be found on page 5-38 of the 68307 user manual.

Note: the 68000 stores its registers most significant byte first i.e. \$100020 accesses the byte PIR1 to INT2L0, \$100021 PIR3 to INT4L0 etc.

PIR8 - 1 Pending Interrupt reset - clears a latched interrupt on the relevant interrupt input pin

INT1L2 - 0 Interrupt Level - allow programming the interrupt level for INT1

INT2L2 - 0 Interrupt Level - allow programming the interrupt level for INT2

INT3L2 - 0 Interrupt Level - allow programming the interrupt level for INT3

INT4L2 - 0 Interrupt Level - allow programming the interrupt level for INT4

INT5L2 - 0 Interrupt Level - allow programming the interrupt level for INT5

INT6L2 - 0 Interrupt Level - allow programming the interrupt level for INT6

INT7L2 - 0 Interrupt Level - allow programming the interrupt level for INT7

INT8L2 - 0 Interrupt Level - allow programming the interrupt level for INT8

Peripheral Interrupt Control Register (PICR)

\$100024

PI_T1	T1L2	T1L1	T1L0	PI_T2	T2L2	T2L1	T2L0	PI_UA	UAL2	UAL1	UAL0	PI_MB	MBL2	MBL1	MBL0
-------	------	------	------	-------	------	------	------	-------	------	------	------	-------	------	------	------

This 16-bit register is used to set up the interrupt levels generated by the on chip peripherals. For each of the four on chip interrupt sources there are four bits in this register. The value placed in three of these bits will determine the interrupt level for that peripheral. The fourth bit, when read will determine whether the peripheral has an interrupt pending. Writing will not affect this

5.5 Chip Select Registers

For each of the four chip select lines that are generated by the MC68307, there are two 16-bit registers. The base address register and the option register. These registers may be written to individually or at the same time using as a long write which may be better in some cases.

If two or more chip select lines are programmed for the same address then an Address Decode Conflict will occur, setting the ADC bit in the top part of the system configuration register. A bus error can be generated if it is enabled. The chip select logic of the 68307 will prevent two chip select lines going low at the same time by implementing a priority system. CS0 has the highest priority while CS3 has the lowest.

Base Register 0 (BR0)

\$100040

FC2	FC1	FC0	BA23	BA22	BA21	BA20	BA19	BA18	BA17	BA16	BA15	BA14	BA13	RW	EN
-----	-----	-----	------	------	------	------	------	------	------	------	------	------	------	----	----

This register contains the base address of the EPROM on the Micro-Module and any function codes that need to be compared before the EPROM is chip selected. Also contained in this register is a read or write only bit and an enable bit.

- FC2 - 0 Function code for the chip select if enabled
- BA23 - 13 Base address for chip select 0
- RW Defines the type of access if MRW bit is set in option register
- EN Enables the chip select line

On reset this register is set to \$C001. In Minos systems this register is left with the default value inside it, enabling the EPROM base address at 0. Full details on the base registers can be found on page 5-30 of the 68307 user manual.

Option Register 0 (OR0)

\$100042

DTK2	DTK1	DTK0	M23	M22	M21	M20	M19	M18	M17	M16	M15	M14	M13	MRW	CFC
------	------	------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

This register is used to set the number of wait states for the EPROM and the size of the block that the EPROM occupies.

- DTK2 - 0 DTACK delay - defines the delay time for DTACK, on the Micro-Module the standard No Wait State access time is 350ns. For each wait state 135ns is added.
- M23-M13 Base Address Mask - defines the size of the chip select
- MRW Mask RW bit - this allows Read or Write access only in conjunction with the RW bit in the Base Register
- CFC Compare Function Codes - enables the function code comparison

On a Minos system this register is set to \$1E00. Full details on the option registers can be found on page 5-32 of the 68307 user manual.

Base Register 1 (BR1) \$100044

This register is used to select the base address of the RAM on the Micro-Module. For a full description of this register see Base Register 0. On Minos systems this register is set to \$0301 to enable the RAM base at \$180000.

Option Register 1 (OR1) \$100046

This register determines the size of the block for the RAM chip select and the number of wait states to be inserted before the DTACK signal is generated. For a description of this register see Option Register 0.

On a Minos system this register is set to \$1F00, zero wait state, 512 K-byte block size.

Base Register 2 (BR2) \$100048

This register is used to select the base address of chip select 2 on the Micro-Module. For a full description of this register see Base Register 0.

On Minos systems this chip select is used for the peripheral address strobe (PAS) and so its base address is programmed as \$128000 by setting this register to \$0251.

Option Register 2 (OR2) \$10004A

This register determines the size of the block for chip select 2 and the number of wait states to be inserted before the DTACK signal is generated. For a description of this register see Option Register 0.

On a Minos system this register is set to \$FFF0, external DTACK 32K-byte block size.

Base Register 3 (BR3) \$10004C

This register is used to select the base address of chip select 3 on the Micro-Module. For a full description of this register see Base Register 0.

Option Register 3 (OR3) \$10004E

This register determines the size of the block for chip select 2 and the number of wait states to be inserted before the DTACK signal is generated. For a description of this register see Option Register 0.

5.6 UART Registers

Full details on the UART registers and programming can be found in the 68681 data sheet or section 8 of the 68307 user manual, details in the Appendix. The 68307 user manual refers to timings obtained using 16.67 MHz or 3.6864 MHz clock speed. The Micro-Module is clocked at 7.3728 MHz, this should be taken into account when calculating baud rates etc.

Mode Register 1 (UMR1) \$100101

RxRTS	RxIRQ	ERR	PM1	PM0	PT	BC1	BC0
-------	-------	-----	-----	-----	----	-----	-----

This register controls some of the serial port configuration. At reset this register is cleared to 0. Full details on this register can be found on page 8-15 of the 68307 user manual.

- RxRTS enables control of data flow using RTS
- RxIRQ determines the source of the receiver interrupt
- ERR controls the function of the RB, FE & PE bits in the status register (USR)
- PM1 - 0 defines the parity mode for the serial port
- PT used to define the parity type
- BC1 - 0 sets the number of bits per character between five and eight

Mode Register 2 (UMR2)

\$100101

CM1	CM0	TxRTS	TxCTS	SB3	SB2	SB1	SB0
-----	-----	-------	-------	-----	-----	-----	-----

This register contains some more set up bits for the serial port. It is accessed immediately after the UMR1 is accessed. At reset all the bits in this register are cleared. Full details on this register can be found on page 8-17 in the 68307 user manual.

- CM1 - 0 set the communication mode
- TxRTS controls the RTS signal
- TxCTS enables data flow control using CTS
- SB3 - 0 determines the number of stop bits

UART Status Register (USR)

\$100103

RB	FE	PE	OE	TxEMP	TxRDY	FFULL	Rx RDY
----	----	----	----	-------	-------	-------	-----------

This read only register is used to return the status of the serial port. At reset all the values are set to 0. Full details can be found on page 8-19 of the 68307 user manual.

- RB set if a break character has been received
- FE set if no stop bit was received after a character (framing error)
- PE set if a parity error has occurred
- OE set if one or more characters have been lost due to the buffer being full
- TxEMP set if after transmission of the last stop bit , there are no characters in the transmitter buffer.
- TxRDY set if the transmitter buffer is empty and ready to be loaded
- FFULL set if a character has been received and is in the receiver buffer waiting to be read. Up to 2 characters could be in the buffer
- Bit 0 set if one or more characters are waiting to be read

UART Clock Select Register (UCSR) \$100103

RCS3	RCS2	RCS1	RCS0	TCS3	TCS2	TCS1	TCS0
------	------	------	------	------	------	------	------

This write only register selects the baud rate clocks for both the receiver and the transmitter. There are two sets of baud rates. Set 1 is selected if bit 7 in UACR is cleared and set 2 is selected if this bit is set. Full details can be found on page 8-21 of the 68307 user manual. Note that the Micro-Module serial port is clocked at 7.3728 MHz so baud rates will be double those shown in the tables in the user manual.

- RCS3 - 0 Receiver baud rate select
- TCS3 - 0 Transmitter baud rate select

UART Command Register (UCR) \$100105

	MISC2	MISC1	MISC0	TC1	TC0	RC1	RC0
--	-------	-------	-------	-----	-----	-----	-----

This write only register is used to pass commands to the UART. Multiple commands can be given if they are non conflicting. Full details can be found on page 8-23 of the 68307 user manual.

- MISC2 - 0 select an individual command
- TC1 - 0 reset the transmitter
- RC1 - 0 reset the receiver

Receiver Buffer (URB) \$100107

RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0
-----	-----	-----	-----	-----	-----	-----	-----

This eight bit read only register contains the character in the receiver buffer. At reset this register is cleared to zero. Details of this register can be found on page 8-25 of the 68307 user manual.

Transmitter Buffer (UTB) \$100107

TB7	TB6	TB5	TB4	TB3	TB2	TB1	TB0
-----	-----	-----	-----	-----	-----	-----	-----

This eight bit write only register contains the character about to be transmitted. At reset this register is cleared to zero. Full details are on page 8-25 of the 68307 user manual.

Input Port Change Register (UIPCR) \$100109

			COS				CTS
--	--	--	-----	--	--	--	-----

This read only register shows the current state of the CTS input pin. At reset it is cleared to zero. Only two bits have any meaning, all others are reserved. Full details can be found on page 8-26 of the 68307 user manual.

- COS set if a change of state lasting longer than 25 - 50µs on the CTS input has occurred
- CTS reflects the current state of the CTS input pin

Auxiliary Control Register (UACR) \$100109

BRG	CTM S2	CTM S1	CTM S0				IEC
-----	--------	--------	--------	--	--	--	-----

This write only register selects which baud rate set is used and controls the receiver and transmitter handshaking. At reset this register is cleared to zero. Full details can be found on page 8-26 of the 68307 user manual.

- BRG if this bit is set then baud rate set 2 is used otherwise set 1 is used.
- CTMS2 must be set
- CTMS1 must be clear
- CTMS0 must be clear
- IEC if set then a change of state on CTS will generate an interrupt

Interrupt Status Register (UISR) \$10010B

COS					DB	RxRDY	TxRDY
-----	--	--	--	--	----	-------	-------

This read only register gives the status of all potential interrupt sources on the UART. This register is masked by the UIMR before an interrupt will be generated. Details on page 8-27 of the 68307 user manual.

- COS if set then a change of state has occurred on CTS and caused an interrupt
- DB either the beginning or end of a break has been detected
- RxRDY either FFULL or RxRDY depending on setting in UMR1
- TxRDY this bit is a duplication of the TxRDY bit in the USR.

Interrupt Mask Register (UIMR)

\$10010B

COS					DB	Rx RDY	TxRDY
-----	--	--	--	--	----	-----------	-------

This write only register is used to enable or disable a particular interrupt source. If the relevant bit in this register is set and the same bit in the UISR is set then an interrupt will be generated. Full details can be found on page 8-28 of the 68307 user manual.

Timer Upper Preload Register (UBG1)

\$10010D

This eight bit write only register holds the 8 most significant bits of the preload value to be used by the timer in order to provide the required baud rate. Details can be found on page 8-29 of the 68307 user manual.

Timer Lower Preload Register (UBG2)

\$10010F

This eight bit write only register holds the 8 least significant bits of the preload value to be used by the timer in order to provide the required baud rate. Details can be found on page 8-29 of the 68307 user manual.

Interrupt Vector Register (UIVR)

\$100119

IVR7	IVR6	IVR5	IVR4	IVR3	IVR2	IVR1	IVR0
------	------	------	------	------	------	------	------

This eight bit register contains the vector register for the UART interrupt vector fetches. It contains an offset from the base of the vector table where the address for the exception handler can be found. Full details can be found on page 8-29 of the 68307 user manual.

Input Port Register (UIP)

\$10011B

							CTS
--	--	--	--	--	--	--	-----

This read only register gives the current state of the CTS input pin. At reset this register is set to \$FF. Only one bit is defined in this register, all others are reserved. Details can be found on page 8-29 of the 68307 user manual.

CTS Change the state of the CTS line

Timer 1 Reference Register (TRR1) \$100122

This register contains the reference value for the counter. The counter will count up from 0 to the value contained in this register. When the counter register value matches the value in this register the REF bit is set in the Timer Event Register and an interrupt may be generated. Following a reset all the bits in this register are set to 1 to enable the maximum count. Further details can be found on page 6-5 of the 68307 user manual.

On Minos systems this register is set to \$9000 giving a 100 Hz interrupt.

Timer 1 Capture Register (TCR1) \$100124

This register is used to latch the value of the timer counter register (TCN) when an edge occurs on the Timer 1 input pin as defined in the timer mode register. At reset this register is cleared. Details on this register are on page 6-5 of the 68307 user manual.

Timer 1 Count Register (TCN1) \$100126

This register is a 16-bit up counter. It contains the current value of the counter. After a reset this register is cleared to 0. Full details are on page 6-5 of the 68307 user manual.

Timer 1 Event Register (TER1) \$100129



This 8-bit register is used to report events generated by the timer. When an event occurs the relevant bit in this register will be set regardless to the setting of the interrupt enable bits. Only two bits are defined in this register, all other bits are reserved. Each bit is cleared by writing a one to it. Full details on this register are on page 6-6 of the user manual.

- REF set when counter value reaches its reference value.
- CAP set when counter value reaches its capture value.

Watch dog Reference Register (WRR) \$10012A

REF15	REF14	REF13	REF12	REF11	REF10	REF9	REF8	REF7	REF6	REF5	REF4	REF3	REF2	REF1	EN
-------	-------	-------	-------	-------	-------	------	------	------	------	------	------	------	------	------	----

This register contains the reference value for the watch dog timer. At reset this register is set to \$FFFF enabling the watch dog with its maximum time out period. If the watch dog counter reaches the value contained in this register the micro-controller will be reset. Full details can be found on page 6-7 of the user manual.

REF15 - 1 watch dog time out counter value.
 EN when set this enables the watch dog

Watch dog Counter Register (WCR) \$10012C

This register contains the current value of the watch dog counter. This register must be written to periodically to prevent the system being reset by the watch dog. Full details on this register can be found on page 6-7 in the 68307 user manual.

Timer 2 Mode Register (TMR2) \$100130

This is the timer 2 mode register. For a description of how to set this register up see Timer 1 Mode Register.

On Minos systems this register is left with its default value.

Timer 2 Reference Register (TRR2) \$100132

This is the timer 2 reference register. For a description of how to set this register up see Timer 1 Reference Register.

On Minos systems this register is left with its default value.

Timer 2 Capture Register (TCR2) \$100134

This is the timer 2 capture register. For a description of how to set this register up see Timer 1 Capture Register.

On Minos systems this register is left with its default value.

Timer 2 Count Register (TCN2) \$100136

This is the timer 2 count register. For a description of how to set this register up see Timer 1 Count Register.

On Minos systems this register is left with its default value.

Timer 2 Event Register (TER2) \$100139

This is the timer 2 event register. For a description of how to set this register up see Timer 1 Event Register.

On Minos systems this register is left with its default value.

5.8 I²C bus (M-Bus) Module Registers

Full descriptions of the I²C bus (M-Bus) module can be found in section 7 of the 68307 user manual.

I²C bus (M-Bus) Address Register (MADR) \$100141

ADR7	ADR6	ADR5	ADR4	ADR3	ADR2	ADR1	
------	------	------	------	------	------	------	--

This register contains the slave address to be used by the I²C bus (M-Bus) module. Following a reset this register is set to 0. Bit 0 is not defined. Full details can be found on page 7-6 of the 68307 user manual.

ADR7 - 1 address of 68307 when accessed as a slave

I²C bus (M-Bus) Frequency Divider (MFDR) \$100143

			MBC4	MBC3	MBC2	MBC1	MBC0
--	--	--	------	------	------	------	------

This register contains the pre-scale value used to scale the system clock to generate the I²C bus (M-Bus) clock. With a 7.3728 MHz system clock the I²C bus (M-Bus) clock frequency can be between 1695 Hz and 335 kHz. Full details can be found on page 7-6 of the 68307 user manual.

I²C bus (M-Bus) Control Register (MBCR) \$100145

MEN	MIEN	MSTA	MTX	TXAK	RTSA		
-----	------	------	-----	------	------	--	--

This register can be read or written at any time. For a full description of this register please see page 7-7 of the 68307 user manual.

- MEN enable the I²C bus (M-Bus) module
- MIEN if set then I²C bus (M-Bus) interrupts are enabled.
- MSTA selects the mode of operation of the I²C bus (M-Bus) module.
- MTX selects the direction of transfer for master and slave transfers.
- TXAK determines whether an acknowledge signal is placed in the 9th clock bit of the data for both master and slave receivers.
- RSTA if this bit is set then a repeated start condition is placed on the bus while the MC68307 is the master.

I²C bus (M-Bus) Status Register (MBSR) \$100147

MCF	MAAS	MBB	MAL		SRW	MIF	RXAK
-----	------	-----	-----	--	-----	-----	------

This is a read only register except the MIF and MAL bits which can be cleared by writing to them. At reset this register reads 0. Full details can be found on page 7-9 of the 68307 user manual.

- MCF cleared while data is being transferred. It is set on the 9th clock of a byte transfer.
- MAAS set if the 68307 is being addressed as a slave
- MBB set when the I²C bus (M-Bus) is busy.
- MAL set when arbitration is lost.
- SRW indicates the value of the R/W command bit of the calling address sent from the master.
- MIF set when there is an interrupt pending.
- RXAK cleared if an acknowledge has been received.

I²C bus (M-Bus) Data I/O Register (MBDR) \$100149

This register contains data that is to be either placed on the bus or has just been read from the bus. Details can be found on page 7-10 of the 68307 user manual.

6 Memory

6.1 Read Only Memory

The Micro-Module is fitted with a 32-pin JEDEC type socket for the read only memory. This socket can take all the standard 8-bit EPROM sizes available including 27C128, 27C256, 27C512, 27C010, 27C020, 27C040 and 27C080. The use of CMOS EPROMs are recommended. As well as taking EPROMs the read only memory on the Micro-Module could be fitted with a Flash EPROM up to 1 M-byte in size or with EEPROM up to 256 K-byte.

The Flash EPROMs are ideal for those still developing code because they are electrically erasable in the programmer and so do not have to be erased for 30 minutes with a UV light before they can be reprogrammed. This will speed up development of code no end. As Flash EPROMs require a 12 Volt supply to program them they can not be programmed in the Micro-Module itself. Flash memories that are supported include 28F010, 28F020 and 28F040.

Electrically Erasable Programmable Read Only Memories (EEPROM) are byte alterable and can be programmed at 5 Volts. If these devices are used in the EPROM socket then code can be updated later using the serial port with out changing the device. These devices are very expensive for the size of data that they can hold.

The Read only memory is mapped into the 1 M-byte slot between \$000000 and \$0FFFFFF for Minos systems. Smaller devices than the 1 M-byte maximum will be reflected.

The 68000 interface uses an asynchronous bus where the access time for devices is determined by the device. The MC68307 can be set up to insert a number of wait states into all ROM access cycles. When using boards with the Minos operating system the ROM is accessed with no wait states allowing devices with access times of up to 350ns to be used with out having to change the number of wait states.

The table in section 9.5 shows the link settings for different sized EPROMs that can be fitted to the Micro-Module. Also in section 9.3 is another link which

is used to enable EEPROMs to be used in the board by connecting up the R/W line from the processor.

6.2 Static RAM

The Micro-Module is fitted with a 32 pin JEDEC type socket for the static RAM. This socket can be populated with either 32 K-byte, 128 K-byte and 512 K-byte static RAM devices. Starter packs are supplied with 128 K-byte of RAM so that there is space for the program to be developed. Target boards are shipped with 32 K-bytes of static RAM as this memory will only be required for variables and stack space. All the programs will be stored in the ROM. The RAM on the Micro-Module is not battery backed and so the contents of the RAM will be lost when the power supply is removed. When the CPU is accessing the RAM there are no wait states and the access time for the RAMs should be better than 350ns, but most static RAMs now have access times of between 70 and 100ns.

On the Micro-Module the Static RAM is decoded in the 512 K-bytes between \$180000 and \$1FFFFFF. If devices smaller than 512 K-bytes are used they will be reflected throughout this memory area. The table in section 9.4 shows the link settings for each size of Static RAM that can be fitted to the Micro-Module.

7 Prototyping Area

7.1 The Micro-Module

The Micro-Module contains its own built in prototyping area. This area can be used by the customer to modify the Micro-Module to perform their own particular task. All the CPU signals are easily available for access to the prototyping area on a 3 row x 27 hole contact area. This should not be used except as a patch area for the prototyping section. To give the customer some idea of how to expand the Micro-Module in this way there are a series of Designer Kits available covering a range of peripheral chips for performing different functions. For details of these kits see the appendix at the back of this manual.

The full 24 bit address bus, the 16 bit data bus, address and data strobes, chip select lines, timers, serial ports, digital I/O lines and 8051 strobe lines are all available for use on the prototyping area so whether you want to add 14 M-bytes of memory or more serial ports all the signals required are easily accessed. Once you have done your development on the prototyping area CMS can integrate your own circuit onto the Micro-Module and build the complete unit for you. In most cases the cost of doing this will be only a small amount dearer than buying the standard Micro-Module even in low volume.

7.2 The Micro-Midget

The Micro-Midget does not have the prototyping area since it is designed to be a plug in target card. Any code that is developed on a Micro-Module system can be used on a Micro-Midget in the target.

This page is intentionally left blank

8 Module Bus expansion

8.1 Introduction

The Micro-Module uses the Module Bus expansion to enable it to interface to a wide range of peripheral boards, including graphic displays, serial communications, digital I/O ports, printers etc. This section of the manual describes the working of the bus and suggests some methods of interfacing to it.

The Module Bus uses two basic modes of operation allowing access to 68000 memory and 68000 peripherals. Each mode uses the same address and data lines but a different set of control lines. The Module Bus also has five interrupt sources, four of which are maskable, which can be used to improve I/O efficiency.

8.2 Bus Width Considerations

The 68000 memory and peripheral modes both use the full 16-bit data bus on the 68000. This means that as the address increases bytes are read alternatively from D8-D15 and D0-D7. This is not a problem with memory as two devices can be used, one for each half of the data bus. However peripheral devices which have only 8-bit data buses must multiplex between D8-D15 and D0-D7. The simplest, and most common, solution is to connect the peripheral to D0-D7 data lines and connect A1 on the bus to A0 on the peripheral etc. This offset address lines mean that the registers on the peripheral will only be accessible at alternate addresses (the same half of the data bus). Using D0-D7 means that the registers are accessible on odd addresses only and gives a degree of compatibility with 68000 systems which also use D0-D7 for interrupt vector fetches. The 68000 instruction set has a group of instructions (movep) which are designed to read and write multiple bytes to peripheral devices connected in this way. On the Micro-Module it is possible to use the 8-bit bus width to access peripherals on the Module Bus connector. This is not recommended for compatibility because, when using the 8-bit bus width of the MC68307, the peripheral must be connected to the UPPER D8-D15 data lines not the lower as with standard 68000 bus peripheral cycles.

8.3 Asynchronous Operation

The 68000 bus uses an asynchronous mode of operation to access peripherals and memory. This means that the speed of each bus cycle is determined dynamically during each bus cycle. The processor starts a bus cycle by supplying the address and data followed by some control signals to say that they are valid. The participating peripheral must then wait until it has accepted the CPU's data (write cycle) or provided data for the CPU (read cycle) before asserting the DTACK (data transfer acknowledge) signal to tell the CPU to complete the cycle.

The Micro-Module can generate the DTACK internally after a programmed number of wait states after a chip select is asserted if required. This is not recommended when accessing the Module Bus peripherals as all expansion boards on this bus will generate their own DTACK signal when they are ready.

8.4 Memory Expansion

The memory expansion mode is used to add large blocks of external memory to Module Bus systems. In this mode all the external Module Bus address lines must be decoded and the AS (or MAS) signal should be used to enable the decoders. The memory may be read or written 8 or 16 bits at a time with the size controlled by DS0 and DS1 signals. If DS0 only is asserted then the data byte is to be transferred on the lower data bus D0-D7. If DS1 only is asserted then data is transferred on the upper data bus D8-D15. If both DS0 and DS1 are asserted then it is a 16-bit access and the whole data bus is used. A simple OR gate can be used with DS0, DS1 and Write to produce separate strobe lines for the two halves of the data bus.

Figures 6 and 7 show timing diagrams for 68000 read and write cycles.

Figure 6 68000 Read Cycle

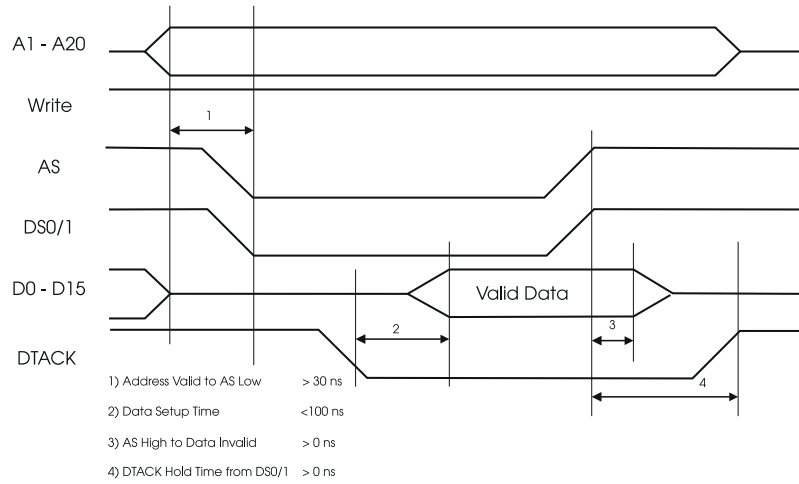
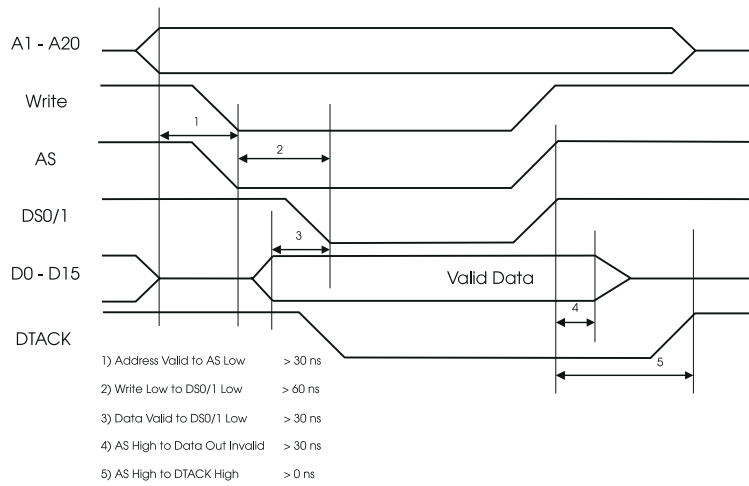


Figure 7 68000 Write Cycle



8.5 Peripheral Expansion

The peripheral expansion mode is used to add I/O devices which have non multiplexed address and data busses. Transfers can be made in 8 or 16 bits but if 8 bit devices are used they should be connected to D0-D7 and should have any address lines offset by 1 ie A1 on the bus goes to A0 on the peripheral etc. The address decoding only needs to include A14 downwards and decoders should be enabled with the PAS (Peripheral Address Strobe) signal.

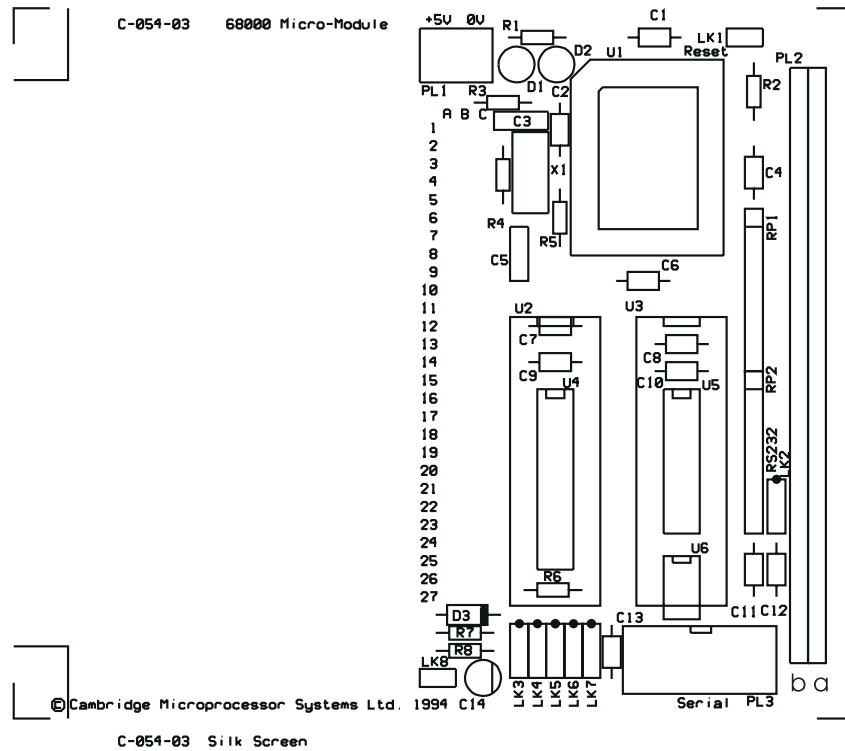
8.6 Interrupts

The Module Bus includes a number of interrupt inputs which can be used freely by expansion boards. Three of the lines correspond to levels 3, 4 and 5, but they can be programmed in the micro-controller to generate any level if required. The other line is the non-maskable interrupt and will always generate an interrupt level 7. The interrupt lines are level sensitive and should be driven with open collector drivers. They are pulled up on the controller board.

9 Links

This section describes the function of all the links and test points on the Micro-Module. The location of all the links on the board are shown in the drawing below. The drawing is based on the silk screen of the Micro-Module.

Figure 8 Micro-Module Silk Screen



9.1 External Reset LK1

The external RESET pins on the Micro-Module are 2x1 gold pins on a 0.1" pitch. In order to reset the Micro-Module these two pins may be shorted together. If these pins are left shorted out the Micro-Module will not run at all.

9.2 RS-232 or RS-485 LK2

This 3 way link is provided to allow the serial port to receive data from either the RS-232 serial port or the RS-485 multidrop network port. Data is transmitted on both standards unless the relevant buffer is removed from the board. The board is sold as standard linked to operate with RS-232. To modify a RS-232 board to work with RS-485 please see section 11.1.

Serial Format	LK2
RS-232	North
RS-485	South

9.3 EEPROM Option LK3

This 3 way link allows EEPROMs to be fitted into the EPROM socket on the Micro-Module. The link determines the signal that is applied to pin 31 on the EPROM socket. This link defaults to north which allows EPROMs up to 27C080 to be fitted. If the link is fitted south then EEPROMs can be fitted up to 28C020.

Pin 31	LK3
A18 (EPROM)	North
R/W (EEPROM)	South

9.4 Static RAM Size LK4

This 3 way link is provided to allow a 512 k-byte static RAM to be fitted in the RAM socket on the Micro-Module. This link is left NORTH for all other sizes of static RAM fitted on the board. The table below shows the link settings for all the different static RAMs that can be fitted to the Micro-Module.

Ram Size	LK4
32 k-byte	North
128 k-byte	North
512 k-byte	South

9.5 EPROM Size LK5, LK6, LK7

These three 3-way links are used to configure the type of EPROM fitted in the EPROM socket on the Micro-Module. The links settings are the same for Flash EPROMs of the same size. The table below shows the link settings for the different EPROM sizes that can be used on the Micro-Module.

Eprom Size	Eprom Type	LK5	LK6	LK7
16 k-byte	27C128	North	North	North
32 k-byte	27C256	North	North	North
64 k-byte	27C512	South	North	North
128 k-byte	27C010	South	South	North
256 k-byte	27C020	South	South	North
512 k-byte	27C040	South	South	North
1 M-byte	27C080	South	South	South

9.6 RS-485 Terminator LK8

This 2 way link allows a terminating resistor to be fitted across the RS-485 transmission line. The resistor is fitted to all RS-485 boards. If the Micro-Module is at either end of the serial link then the link can be terminated simply by fitting a jumper between these two pins. This link is only fitted on RS-485 modules (see section 12.1 on how to modify boards to work with RS-485 serial links).

10 Connectors

10.1 Power PL1

The power connector on the Micro-Module is a 2 pin Screw Terminal Block. +5V is applied to one of the terminals, 0V to the other.

10.2 Module Bus PL2

This connector will either be a 64 way ab DIN41612 connector or a pin strip depending on the version of the board purchased.

Pin Number	Row B	Row A
1	D0	D8
2	D1	D9
3	D2	D10
4	D3	D11
5	D4	D12
6	D5	D13
7	D6	D14
8	D7	D15
9	Ground	Ground
10	CLK	SCL
11	Ground	A17
12	DS1	NRST
13	DS0	A18
14	Write	A19
15	A20	A0
16	DTACK	A21
17	Ground	A22
18	PAS	NMI
19	AS	N/C

Micro-Module Hardware

Pin Number	Row B	Row A
20	A23	I5
21	IACK (+5V)	I4
22	A16	I3
23	SDA	A15
24	A7	A14
25	A6	A13
26	A5	A12
27	A4	A11
28	A3	A10
29	A2	A9
30	A1	A8
31	-12 Volts	+12 Volts
32	+5 Volts	+5 Volts

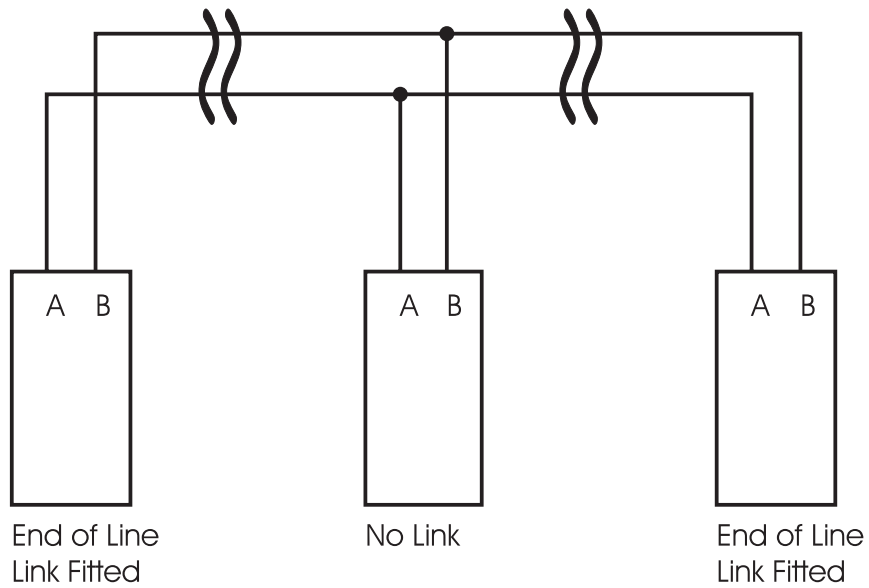
D0 - D15 68000 Data Bus
 A1- A23 68000 Address Bus
 A0 68000 Address Bus A0 8 bit bus transfers only
 CLK Main System Clock
 DS0/1 68000 Data Strobes
 Write 68000 Write Cycle
 DTACK Data Transfer Acknowledge
 AS 68000 Address Strobe
 PAS Peripheral Address Strobe
 IACK Interrupt Acknowledge (Not Used)
 SDA I2C bus (M-Bus) Data Line
 SCL I2C bus (M-Bus) Clock Line
 NRST System Reset
 NMI Non Maskable Open Collector Interrupt
 I3-I5 Open Collector Interrupt lines

10.3 Serial Port PL3

The serial port connector on the Micro-Module is a 10 way vertical boxed header for a bump polarised IDC type socket. A and B are the RS-485 signals. The RS-232 signals are pinned out to be compatible with a 9 pin P.C. type connector.

Signal	Pin Number	Pin Number	Signal
A (+ve)	10	9	Ground
B (-ve)	8	7	N/C
RTS	6	5	RxD
CTS	4	3	TxD
+5 Volts	2	1	N/C

Figure 9 RS-485 Serial Configuration



10.4 Prototype area interface

10.4.1 Micro-Module (K-030)

This interface area does not have a connector fitted on the Micro-Module. This is provided to allow the designer easy access to all the CPU signals when they are adding devices on to the prototyping area of the board. Devices should NEVER be plugged into this section of the board. This patch area is clearly labelled with a coordinate system to make it as easy to use as possible. The three columns are numbered A, B and C. The 27 rows are numbered from 1 to 27 on the silk screen.

10.4.2 Micro-Midget (K-038)

This connector is fitted with long pins on the Micro-Midget, which is designed to plug into a mother board. The signals with the same names as on the Module Bus connector are connected through on the Micro-Midget. There is no need to connect them on the mother board.

Pin Number	Column A	Column B	Column C
1	DS0	ALE	AS
2	D7	SCL	RD
3	D6	WR	NRST
4	D5	PAS	CS3
5	D4	CLK	T1in
6	D3	CTS	PB14
7	D2	PA6	PA5
8	D1	NMI	PA7
9	D0	PA4	T2in
10	PB15	PA3	PA2
11	A22	PA1	A21
12	PB9	PB8	A18
13	A23	RTS	A16
14	PB13	A15	A14
15	PA0	A9	A12
16	PB12	PB11	A7
17	PB10	A20	A6
18	SDA	DS1	A5
19	A13	A19	A4
20	Ground	+5 Volts	A3
21	TxD	Write	A2
22	RxD	A8	A1
23	A10	A11	A0
24	+12 Volts	D15	D8
25	-12 Volts	D14	D9
26	A17	D13	D10
27	D11	D12	DTACK

This page is intentionally left blank

11 Specification

11.1 Processor

MC68307 CMOS 68K Micro-Controller
68000 Instruction set
7.3728 MHz operation (14.7456 MHz option)

11.2 Memory

Up to 1 M-byte of EPROM in 32 pin JEDEC socket
Up to 1 M-byte of Flash EPROM (optional)
Up to 256 K-byte EEPROM (optional)
Up to 512 K-byte Static RAM
350ns No Wait State access.

11.3 Serial Port

1 RS-232 Buffered Serial Port
1 RS-485 Buffered multi drop network port option
75 to 38.4 Kbaud
5-8 data bits, 1- 2 stop bits, parity checking
Hardware handshaking (RS-232)
Auto transmit enable (RS-485)
9 pin P.C. Serial Port Compatible (RS-232)

11.4 Digital I/O

8 dedicated I/O lines
Up to 22 digital I/O lines
8 interrupt inputs with programmable interrupt levels

11.5 I²C bus (M-Bus)

128 Station Numbers
100 k bits per second transfer speed
Two wire multi master bus protocol

11.6 Counter Timers

Two 16-bit Counter/Timers
Cascadable for 32-bits
Two 16-bit Capture/Compare registers
Three clock sources

11.7 Watch dog Timer

34 second standard time out
Programmable time out
Software enable/disable

11.8 Prototyping Area (Micro-Module Only)

Large area for user circuits
22 x 37 0.1" pad matrix
Easy access to all CPU signals
Over 14 M-byte of expansion space

11.9 8051 Bus Access

Memory mapped 8051 peripheral expansion
ALE, RD and WR strobes on prototype expansion

11.10 Micro-Module

64 way DIN 41612 ab connector
Size 100 x 118mm
Compatible with Module range of products

11.11 Micro-Midget

124 Gold pins 11.5mm long
Size 100 x 64mm

11.12 Power Supply

5 Volt only operation at 60mA
Low power mode less than 30mA

11.13 Operating Temperature

0 to 60 degC

This page is intentionally left blank

12 Application Notes

12.1 Converting to RS-485

Micro-Modules can be ordered with a RS-485 multi drop network port fitted instead of a RS-232 serial port. If you want to convert a board from RS-232 to RS-485 then please follow the instructions below. The only tools required are a soldering iron, a pair of wire cutters and a pair of pliers. Since semiconductor devices are sensitive to static it is recommended that you observe static handling precautions when carrying out this modification.

Components List

Description	Quantity	RS Order Code
MAX485CPA	1	299-935
100 Ohm 5% 0.25W	1	131-132
1K Ohm 5% 0.25W	1	131-255
1x2 pin strip	1	453-145 (Not gold)
Jumper (Optional)	1	334-561

1. Use the pliers to bend the two resistors to 0.4" long.
2. Solder the 100 Ohm resistor in location marked R7 on the PCB.
3. Solder the 1K Ohm resistor in location marked R8 on the PCB.
4. Solder the 1x2 pin strip in location marked LK8.
5. Solder the MAX485 chip into location marked U6 (remove the EPROM first as this device is underneath this socket).
6. Move LK2 South i.e. away from the label RS232.
7. If this module is to be used at either end of the RS-485 link then the jumper should be fitted on LK8.

The Micro-Module will now function with an RS-485 network serial port. Please use the 485 driver which can be found in the MINOS directory on your installation disk to use the serial port correctly. The RS-485 serial data will transferred on pins 8 (-ve) and 10 (+ve) of PL3. To reduce the power consumption the MAX232ACPE RS-232 serial port buffer could be removed from the board.

An alternative part to the MAX485CPA device from MAXIM used above is Linear Technologys LTC485CN8. Industrial temperature versions of both of these devices are available if required, although you probably will not be able to purchase them from main line catalogue distributors.

12.2 Converting to RS-232

Micro-Modules can be ordered with a RS-232 serial port fitted instead of a RS-485 multi drop network port. If you want to convert a board from RS-485 to RS-232 then please follow the instructions below. The only tools required are a soldering iron, a pair of wire cutters and a pair of pliers. Since semiconductor devices are sensitive to static it is recommended that you observe static handling precautions when carrying out this modification.

Components List

Description	Quantity	Farnell Order Code
MAX232ACPE	1	MAX232ACPE
0.1 μ F Ceramic Axial	4	108-993

1. Use the pliers to bend the four capacitors to 0.3" long.
2. Remove the EPROM from its socket.
3. Solder the four capacitors in the locations marked C10, C11, C12 and C13.
4. Solder the MAX232ACPE into the location marked U5.
5. Move LK2 North i.e. towards the label RS232.
6. Refit the EPROM into its socket.

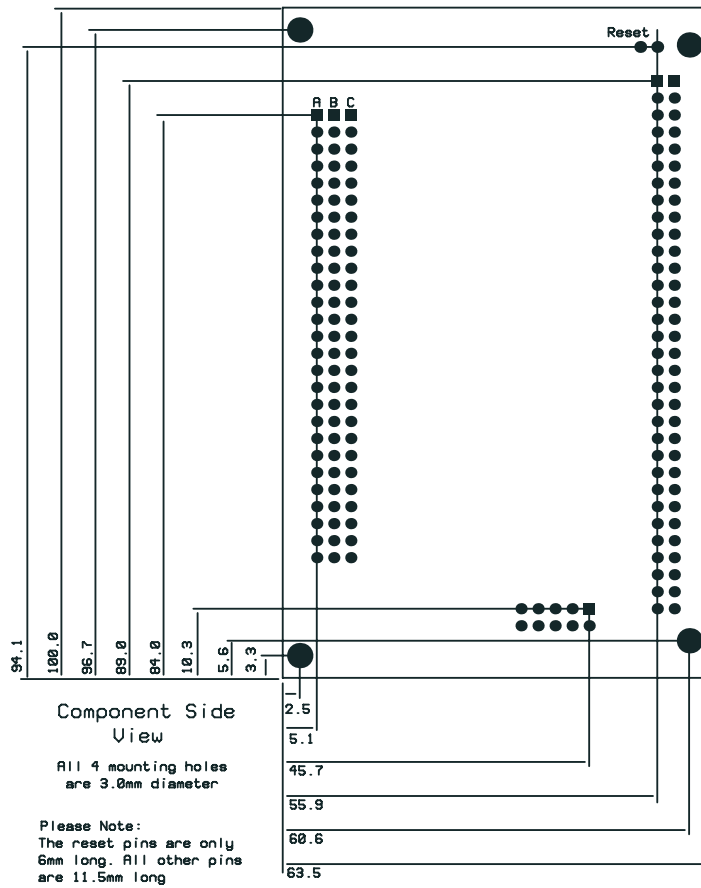
The board will now function as RS-232. Any jumper on LK8 can now be removed and the MAX485 (or LTC485) could be removed from the board. See section 10.3 for details of the RS-232 serial port connector.

Analogue Devices manufacture an equivalent part for the MAX232ACPE. The Analogue Devices part number is ADM232A. This part is available over industrial temperature ranges if required but you may not be able to purchase it from the main line catalogue companies.

12.3 Micro-Midget Mounting Dimensions

Figure 10 Micro-Midget Mounting Dimensions

All dimensions are +/- 0.05mm
 All connectors are on a 0.1" grid



This page is intentionally left blank

13 Appendix

13.1 Useful data sheets

MC68307/D	The micro-controller used in this product	Motorola
ADI988R1	The 68681 DUART data sheet	Motorola
MC68EC000/D	68EC000 data sheet	Motorola

13.2 User Manuals and Books

The manuals and books listed below could be of use to people wanting to program this product using the 68000 assembly language. There are many books on the market which describe the 68000 and how to program it. This list is by no means exhaustive. The 68307 user manual is provided by CMS as part of the documentaion for this product in a PDF format.

MC68307UM	User Manual for the Micro-Controller	Motorola
MC68000UM	16-bit Microprocessor User Manual	Motorola
M68000PM	Programmers reference manual	Motorola

13.3 Returns Policy

As the Micro-Module is sold in such a simple state it will normally cost more to repair a board that has been damaged than to replace it. If this is the case the damaged Micro-Module should be replaced with a new board. If a Micro-Module fails to operate once the customer has added circuits to the prototyping area we will be unable to repair the board and a replacement board should be purchased. If you have any problems using your

Micro-Module please check both the notes file and the readme file on the installation disk for the latest information.

13.4 Designer Kits

These are available if required. They show how to add extra devices to the Micro-Module. At the time of going to press the following designer kits are available.

Single Channel Serial Port using SCN2661

I2C bus (M-Bus) Examples.

Alphanumeric LCD

Graphics LCD

13.5 Acknowledgements

Information provided in this manual on the MC68307 micro-controller is a summary of the information contained in the 68307 user manual published by Motorola.

Part numbers are quoted from the following catalogues:

R S Components Ltd Sales Telephone 01536 201201

Farnell Electronic Comps. Sales Telephone 0113 263 6311

13.6 Circuit Diagram

The circuit diagram for the Micro-Module is shown over the page.