

SIBO 'C' Software Development Kit

HARDWARE REFERENCE

Version 2.10

NOTE: *The data in this manual is provided for information only. You are strongly advised not to make direct access to any of the hardware since this may conflict with access made by EPOC. To guarantee future compatibility, hardware access should only be made via the appropriate operating system calls.*

(C) Copyright Psion PLC 1990-95

All rights reserved. This manual and the programs referred to herein are copyrighted works of Psion PLC, London, England. Reproduction in whole or in part, including utilization in machines capable of reproduction or retrieval, without express written permission of Psion PLC, is prohibited. Reverse engineering is also prohibited.

The information in this document is subject to change without notice.

Psion and the Psion logo are registered trademarks, and Psion, Psion MC, Psion HC, Psion Series 3, Psion Series 3a and Psion *Workabout* are trademarks of Psion PLC.

IBM, IBM XT and IBM AT are registered trademarks of International Business Machines Corp. Microsoft and MS-DOS are registered trademarks of Microsoft Corporation. Apple and Macintosh are registered trademarks of Apple Computer Inc. VAX and VMS are registered trademarks of Digital Equipment Corporation. 80C86, 8088, 8288, 8259 and SLD are registered trademarks of Intel Corporation. V30 and V30H are registered trademarks of NEC Corporation. Brief is a registered trademark of Underware Inc. Psion PLC acknowledges that some other names referred to are registered trademarks.

Contents

1 The Principal Chip Set	1
ASIC1	2
Bus controller	2
Bus converter	2
Timer	2
LCD controller	3
Memory decoding	3
I/O decoding	4
Interrupt controller	4
Watch dog timer	4
Address and I/O protection.....	4
ASIC2	4
System clock oscillator	5
On/off control	5
SIBO serial protocol controller.....	5
SIBO serial protocol slave.....	5
Keyboard interface	5
Buzzer interface	5
NMI generation	5
ASIC9	6
Increased memory addressing	6
Increased processor speed	6
Improved Pack performance with hardware wait	6
Addition of hardware 32 bit RTC.....	6
Improved sound facilities	7
Switch on via the membrane key pads	7
Facility to address two ROM sites	7
Additional FRC	7
Improved Grey scale.....	7
Triple Serial Control and Data Registers.....	7
Contiguous video memory map	7
Expansion support in SSD slots	7
Altered Interrupt allocation	8
PSU subsystem.....	8
The standby state	8
The backup state	8
The idle state	8
The operating state	8
Vcc1 supply	8
Vcc2 supply	9
Vcc3 supply	9
Vcc4 supply	9
Vcc5 supply	9
Vee1 supply	9
Vee2 supply	9
Vh supply	9
2 The SIBO Serial Protocol.....	11
Hardware Interface	11
Clock Line.....	11
Data line	12
The Physical layer	12
Frame structure	12
Null Frame	13
Control frame.....	13

Data Output Frame	13
Data Input Frame.....	13
Data line direction	14
The Transport layer	14
Slave select mode	15
Slave control mode.....	15
Timing	17
States.....	17

3 The SIBO Expansion Ports	19
Extended internal expansion	19
Physical connector	19
Signal definition	20
Direct I/O	20
The SIBO serial channel.....	22
Power supplies.....	22
Reduced external expansion	22
Physical connector	22
Signal definition	22
Example expansion devices	23
Extended expansion example one - 8 bit read/write port.....	23
Extended expansion example two - 16 channel ADC	23
Reduced expansion example three - TTL RS232 based interface.....	24

CHAPTER 1

THE PRINCIPAL CHIP SET

Current SIBO products in the MC, HC and Series 3 range are based on the same three principal chips. These are the V30H (a 8086 compatible processor) and two Psion custom chips known as *ASIC1* and *ASIC2*. Later SIBO products including the Series 3a have these three devices integrated into a single Psion custom chip known as *ASIC9*.

The V30H is an enhanced 16 bit CMOS version of the 8088 found in the original IBM PC. It is software compatible with the 8088. The V30H is a fully static design, which means that all the internal storage elements i.e. its registers are made from static rather than dynamic storage elements. This means there is no minimum clock speed required to refresh the storage elements and the system clock can be stopped at any time with no loss of internal state. This technique is used extensively in the SIBO architecture to save power while the processor is idle i.e. waiting for an event.

ASIC1 and *ASIC2* are Psion specific gate arrays based on a one micron full CMOS process. They are described in more detail in the next sections.

Figure 1 shows a simplified block diagram of the SIBO architecture showing the principal three chips.

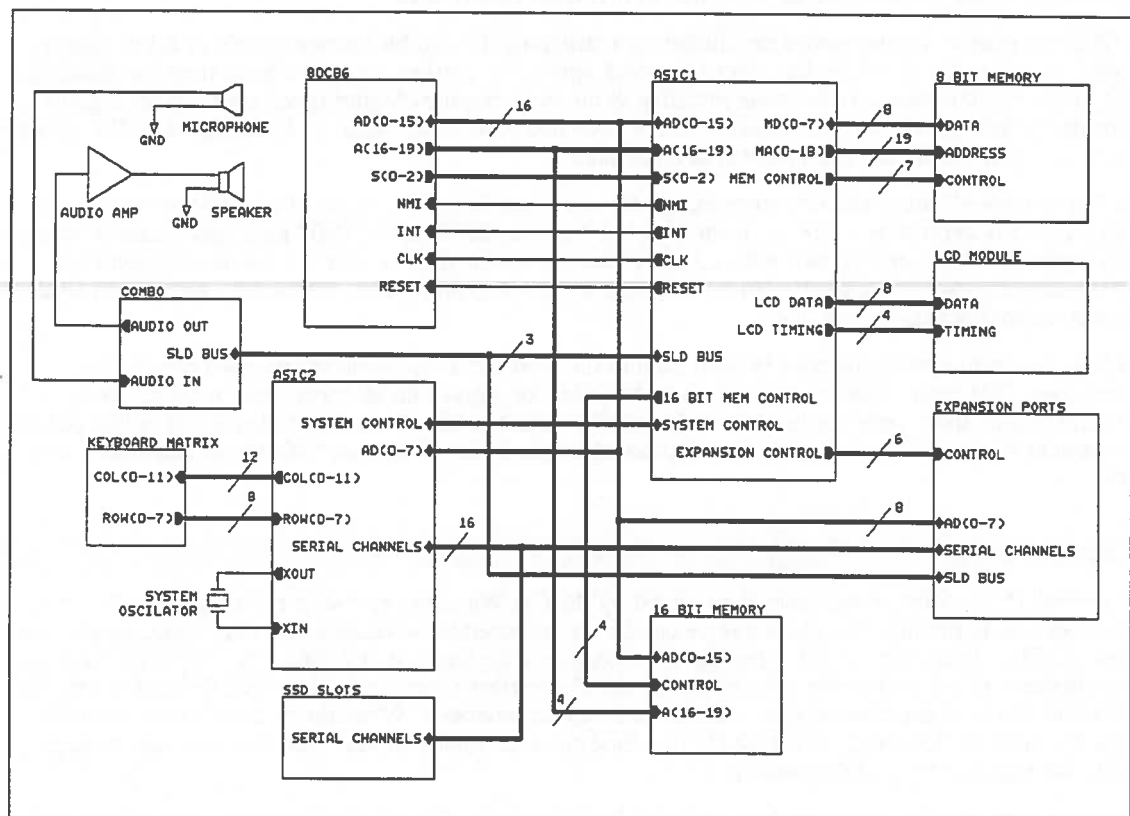


Figure 1.

ASIC1

ASIC1 contains approximately 4300 gates, implemented on a Hitachi one micron CMOS gate array in a 100 pin quad flat pack package.

ASIC1 is the main system controller for the SIBO architecture. It connects directly to the 80C86 in maximum mode controlling all bus cycles to and from the processor. This configuration effectively forms a micro-controller like device that executes 8086 instruction codes. ASIC1 is made up of a number of functional blocks described below.

Bus controller

The bus controller converts the encoded bus cycle requests (present on S0-S2) from the processor into the appropriate memory and I/O control signals. The bus controller is similar to the Intel 8288 device but with several enhancements, these are:-

- 1) Clock stopping on the HALT instruction. Whenever a HALT instruction is issued by the processor the clock line to the processor is stopped and held low until either an NMI or an interrupt is asserted.
- 2) Bus arbitration during video refresh cycles. Bus cycles to the video RAM segment are delayed while a video refresh is in progress.
- 3) Memory write and I/O blocking. Memory writes outside pre-programmed segments and all I/O instructions can be blocked by the bus controller. See following sections on address and I/O protection.

Bus converter

There are two distinct memory buses in the SIBO architecture, these are the 8-bit bus and the 16-bit bus. Memory access to the 8-bit bus are converted to 16 bit accesses by the bus converter. This means that the 80C86 can access memory on the 8-bit bus as if it were 16-bits wide.

ASIC1 can operate in two modes dependant on a strapping pin: 16 bit memory mode or 8 bit memory mode. 16 bit mode allows higher processor clock speeds for a given memory access time but requires a full 16 bit bus and hence 16 bit wide memory. 8 bit mode requires higher speed memory for a given processor clock speed but only requires an 8 bit bus and 8 bit wide memory. In all MC's *ASIC1* operates in 16 bit mode, while the HC operates in 8 bit mode.

In 16 bit mode all the system memory except for the video RAM is connected to the 16 bit bus. The video RAM is defined as a 32K segment of RAM between B8000 and C0000 hex. Any memory access to this segment is converted to two 8 bit memory accesses on the 8 bit bus by the bus conversion logic. These accesses may delay the 80C86 by stopping the clock to allow time for the two accesses or to allow a video refresh access to take place.

In 8 bit mode all system memory is connected to the 8 bit bus so all memory accesses need to be converted. This means that memory needs to be faster for a given clock speed than in 16 bit mode, because two memory accesses need to be "squeezed" into one bus cycle. The video RAM is just part of the system RAM and is of variable size dependent on the LCD screen size with a base address of 400 hex.

Timer

A general 16 bit count down timer is provided by *ASIC1*. When the system is not in the standby state i.e. the processor is running, the clock source can be programmed to be either a fast (512 KHZ) clock or a slow (32 Hz) clock. Any 16 bit value can be written into the timer at any time. This value will then be decremented on the next rising edge of the clock. The current value in the timer can be read at any time. When the value in the timer reaches zero an interrupt is generated. When the system is in the standby state the timer is clocked from the 32 Hz real time clock oscillator in *ASIC3* so that time can be kept while the processor is powered down.

The timer can operate in two modes controlled by a programmable bit, free running mode or prescale mode. In free running mode, when the timer decrements to zero an interrupt is generated and the counter just wraps around to -1 (FFFF). In prescale mode when the interrupt is generated the original value written to the timer is re-loaded automatically, this allows the interrupt to become a rate programmable pulse stream. This pulse stream is also an output from *ASIC1*.

LCD controller

The LCD controller provides all the necessary timing and data signals to drive any graphic LCD module of up to 640 by 400 dots in size.

The data is output on up to eight signals. Each data signal outputs serial data that is synchronous with the dot clock. The data is a continuous bit stream formed from serialised data taken from the video refresh RAM. The data is serialised so that the first pixel in a line is bit 0 in the corresponding byte of the video RAM, this means the video RAM can be accessed in words i.e 16 bits wide.

The timing and data signals are controlled by a number of programmable registers which allow the following parameters to be set:

- 1) Line length - the number of dots in the line.
- 2) Frame length - the total number of dots in the screen.
- 3) Refresh rate - sets the refresh rate. This should be calculated from the system clock speed and the screen size to give a refresh rate of approximately 70 frames per second.
- 4) AC drive rate - sets the frequency of the AC drive signal to the LCD module as a function of the line rate. This should be calculated to be approximately 300 Hz.
- 5) Display mode - four different display modes are supported these are: single screen refreshed from video page one; single screen refreshed from video page two; single screen refreshed from video page one and two giving a two bit grey scale; and double screen refreshed from video page one and two.

Memory decoding

Memory decoding is dependent on what mode *ASIC1* is in. In 16 bit mode only a single 8 bit memory device can be addressed, in 8 bit mode up to six memory devices can be addressed. 16 bit memory devices must be decoded externally from *ASIC1*.

8 bit memory decoding allows for different sized physical memory devices to be connected to the logical memory slots. Three memory device size options are supported allowing for 32K x 8, 128K x 8 and 512K x 8 devices to be addressed. These options are selected by two programmable bits, *MSEL0* and *MSEL1*.

The following table summarises the memory device size options:-

MSEL1	MSEL0	Device size
0	0	32K x 8
0	1	128K x 8
1	0	Invalid
1	1	512K x 8

The following table summarises the memory decoding performed by *ASIC1*:-

Device size		32K x 8	128K x 8	512K x 8
Mode	Memory slot			
8 Bit	8 Bit slot 0	00000 - 07FFF	00000 - 1FFFF	00000 - 7FFFF
8 Bit	8 Bit slot 1	08000 - 0FFFF	20000 - 3FFFF	80000 - FFFFF
8 Bit	8 Bit slot 2	10000 - 17FFF	40000 - 5FFFF	-
8 Bit	8 Bit slot 3	18000 - 1FFFF	60000 - 7FFFF	-
8 Bit	8 Bit slot 4	20000 - BFFFF	80000 - BFFFF	80000 - BFFFF
8 Bit	8 Bit slot 5	C0000 - FFFFF	C0000 - FFFFF	C0000 - FFFFF
16 Bit	8 Bit slot 1	B8000 - BFFFF		
16 Bit	16 Bit slot	00000 - B7FFF + C0000 - FFFFF		

I/O decoding

I/O space is decoded by *ASIC1* according to the following table:

Address Range	Function
0300 - FFFF	Not used
0200 - 02FF	Expansion port 2
0100 - 01FF	Expansion port 1
0090 - 00FF	Not used
0080 - 008F	I/O space in <i>ASIC2</i>
0020 - 007F	Not used
0000 - 001F	Internal I/O in <i>ASIC1</i>

Interrupt controller

The interrupt controller in *ASIC1* prioritises eight interrupt request lines to generate a single interrupt request to the processor. When and if the processor executes an interrupt acknowledge cycle, the interrupt controller places the correct vector on the bus.

The interrupt controller is similar to the Intel 8259 device, however, the following restrictions apply; there is no provision for daisy-chaining additional interrupt controllers, level triggered mode only is supported and the interrupt vector base is fixed at 78 hex.

The 8 interrupts are defined internally in *ASIC1* as follows:

Interrupt	Vector	Name	Description
IRQ0	78	TINT	Tick interrupt at 2 or 32 Hz
IRQ1	79	EINT0	External interrupt usually connected to mains detect bit
IRQ2	7A	EINT1	External interrupt from expansion port one
IRQ3	7B	EINT2	External interrupt from expansion port two
IRQ4	7C	EINT3	External interrupt from <i>ASIC2</i>
IRQ5	7D	OVINT	Timer overflow interrupt
IRQ6	7E	SRXI	SLD sound receive interrupt
IRQ7	7F	STXI	SLD sound transmit interrupt

Watch dog timer

The watch dog timer is a two bit counter which counts up at a rate of 4 Hz. If the count reaches three a NMI is generated. The counter can be reset by a write to an I/O location inside *ASIC1*. Normally the counter is reset on every tick by the interrupt service routine. The NMI will be produced after 24 ticks have passed with no reset, this equates to 0.75 of a second with interrupts disabled.

Address and I/O protection

ASIC1 can be switched into protected mode, in this mode all I/O instructions and memory writes outside a predefined range are blocked. An attempted I/O instruction or write out of range will generate an NMI.

The range is set by writing the upper and a lower segment bounds for the given range to two internal 16 bit registers in *ASIC1*.

Protected mode is switched on and off by writing to or reading from a special I/O location in *ASIC1*.

ASIC2

ASIC2 contains approximately 2200 gates, implemented on a Hitachi one micron CMOS gate array in an 80-pin quad flat pack package.

ASIC2 contains the system clock oscillator, controls switching between the standby and operating states, and provides an interface to the power supply, keyboard, buzzer, SSDs and expansion ports. *ASIC2* consists of the following functional blocks:

System clock oscillator

An on-chip oscillator cell connected to an external crystal generates the main system clock in the operating state. The system clock frequency is divided down to provide clocks for the serial channels, the SLD bus interface and for RS232 communication. The oscillator is disabled in the standby and battery backup states to conserve power.

On/off control

ASIC2 controls switching between the standby and operating states. *ASIC2* puts the system into the operating state when: the keyboard "on" signal is asserted; the system "reset" signal is generated; an expansion device requests service; an external controller initiates a "high speed link" or the *ASIC1* timer reaches zero.

Switching from the operating state to the standby state is software controlled. When an "enter standby" command is written to *ASIC2* it shuts down the power supply via the serial interface and puts the system into the standby state.

SIBO serial protocol controller

ASIC2 contains an 8-channel SIBO serial protocol controller which provides an interface to the power supply, the SSDs, the expansion ports and the high speed link. Channel assignments are:

Channel	Function
0	Power supply (<i>ASIC3</i>)
1-4	SSDs
5,6	Expansion ports
7	High speed link

Serial clocks 5 & 6 may be set free running at 1.536Mhz, this frequency may be divided down by devices connected to the expansion ports to provide baud rates for RS232 communication.

To maximise the data transfer rate, a "controller busy" signal is generated by *ASIC2* during a data transfer. This signal is connected to the processor TEST pin.

For further details refer to "The SIBO serial protocol" chapter of this manual.

SIBO serial protocol slave

The SIBO serial protocol slave is used for the "high speed link" to receive data transmitted by an external controller. An interrupt is generated when a frame is received.

Keyboard interface

ASIC2 provides an interface to a matrix keyboard of up to 192 keys, made up from 16 rows by 12 columns. During a keyboard scan the columns are each driven high in turn by *ASIC2* and the row information is read from the keyboard row register. Normally the row buffer is 8 bits wide which gives a matrix of 8 x 12 i.e. 96 keys.

Buzzer interface

A piezo electric buzzer element can be connected directly to *ASIC2*. The buzzer can be driven at high or low volume, and either directly by toggling an *ASIC2* register bit or by the overflow bit from the *ASIC1* timer.

NMI generation

ASIC2 controls the generation of an NMI which is passed onto the processor via *ASIC1*. This NMI can be generated from a number of sources, these are, power fail conditions and SSD or expansion port door micro-switches. When the main power supply voltage (V_{sup}) falls to a level insufficient to power the system *ASIC3* signals *ASIC2* and a power fail NMI is generated. When an expansion device is removed or inserted, or if an SSD is removed or inserted micro-switches connected to *ASIC2* generate the appropriate NMI. The source of the NMI is identified by reading a register in *ASIC2*.

ASIC9

The Series 3a differs from the above description in that the V30H and all the functions in *ASIC1* and *ASIC2* have been integrated into a single custom device known as *ASIC9*. This has allowed faster processing speeds and lower power consumption to be achieved. In addition a number of enhancements and alterations have been made to the SIBO architecture listed below.

Increased memory addressing

The Series 3a extends the V30H address range beyond the 1 MByte limit. This allows larger mask ROM's and RAM's to be addressed. This is done by implementing bank switching in *ASIC9* which will be similar to EMS on PC's.

Each bank is 64 KBytes in size and is selected by writing into one of four page registers, two to select a RAM address and two to select ROM addresses. These page registers are called PSEL0 to PSEL3 respectfully.

This method extends the V30H address capability by 4 bits to produce two 24 bit address bus ranges, one for RAM and one for ROM. This gives an affected linear address range of 32 MBytes. The 16 MByte RAM space is decoded into 4 Chip selects allowing up to 8 x 16 Mbit RAM devices (2 per chip select as 16 bit wide). The 16 MByte ROM space is decoded into two chip selects each selecting 8 MBytes each, one for mask ROM and one for FLASH memory.

The physical address map for the V30H now looks like this:-

Address range	Description
00000 - 5FFFF	Maps to physical bottom 384K of the 16 Mbyte RAM space.
60000 - 6FFFF	Maps to 64K RAM segment selected by page register PSEL0.
70000 - 7FFFF	Maps to 64K RAM segment selected by page register PSEL1.
80000 - 8FFFF	Maps to 64K ROM segment selected by page register PSEL2.
90000 - 9FFFF	Maps to 64K ROM segment selected by page register PSEL3.
A0000 - FFFFF	Maps to physical top 384K of the 16 Mbyte ROM space.

Increased processor speed

Although the V30H microprocessor is rated at clock speeds of up to 16 MHz, the Series 3 machine is currently limited to a clock speed of 3.84 MHz. This is due to the fact that *ASIC1* accesses the ROM as two 8 bit bytes to assemble a 16 bit word for the V30H. This equates to a required ROM access speed of approximately 250 nSec for a 3.84 MHz clock. Readily available mask ROM's have an access speed of 200 nSec which currently limits the clock speed of Series 3.

ASIC9 supports 16 bit wide accesses to memory space which allows a doubling of the clock speed for the same ROM access time. In addition *ASIC9* will support *programmable wait states* so that faster clock speeds can be used with slow ROM's or RAM's.

Improved Pack performance with hardware wait

Data throughput to and from SSD's has been significantly increased by integrating the serial controller more closely to the bus controller. This is now possible as they are both in the same chip. The serial controller can now automatically hold the processor (by stopping the CPU clock) if it attempts to issue a command to the controller whilst it is busy. This reduces to zero the overhead involved in determining this condition. This technique should also enhance the speed that Flash memory devices can be written to.

Addition of hardware 32 bit RTC

The *ASIC1/ASIC2* implementation of the SIBO architecture uses the FRC (Free running counter) to maintain time while the CPU is powered down (in standby). This means the CPU must wake up at a periodic rate (approx 8 hours) to keep time, if the CPU cannot start (no main batteries) time is lost. To overcome this limitation *ASIC9* contains a separate 32 bit real time clock that will continually maintain the system time independently from the CPU. This counter will be clocked at 1 Hz so its maximum range will be 2^{32} seconds or approx 136 years.

Improved sound facilities

ASIC9 contains a 16 FIFO buffer to reduce the CPU overhead involved in playing or recording sounds via the CODEC device.

Switch on via the membrane key pads

ASIC9 contains the logic to bring the system out of the standby state if any of a set number of row inputs goes high. This allows Switch on via the membrane key pads.

Facility to address two ROM sites

The 16 MByte paged ROM address will be split into two sections each 8 MBytes in size. It is intended that one section is for mask ROM one for FLASH memory.

Additional FRC

ASIC9 contains an additional FRC (Free Running Counter). This FRC is functionally identical to the current FRC but is indented for general user use and is not used by EPOC.

This second FRC can be clocked from either 512 KHz or 1024 Hz. This extends the use of the FRC by providing a counter with a resolution of 1 mSec and a range of just over one minute. This could help software time events in the mSec range i.e. a stopwatch.

Improved Grey scale

The mapping of the two video maps has been improved to reduce the overhead that grey scales impose on software. This improved scheme supports only one grey level together with black and white. The second video map is an attribute map and would have the effect of dimming only those pixels that are set i.e. black.

Triple Serial Control and Data Registers

ASIC9 contains three serial control registers and three read data registers. This allows independent expansion devices to be added to either SSD slot.

The main advantage of triple registers is that a multiple-mode pack read can be interrupted at anytime by a read or write to any of the other serial channels without loss of setup and data, for example servicing an RS232 interrupt from either SSD slot.

Contiguous video memory map

The *ASIC1/ASIC2* implementation of the SIBO architecture positions the two video maps so that they are separated by a fixed address space (16K 4000H). This restriction is overcome in *ASIC9* by starting the second video map directly after the first video map regardless of the size of the video maps.

Expansion support in SSD slots

Two active low external interrupt inputs are provided by *ASIC9* which can be connected to the V_{BACK} outputs on two SSD sockets. This will allow SSD sized expansion devices to be fitted internally to the Series 3a.

Altered Interrupt allocation

The Interrupt allocation has been changed in *ASIC9* so that sound processing is serviced at the highest priority. Interrupts are allocated as follows:-

Interrupt	Vector	Name	Comment
IRQ0	\$78	CSINT	CODEC sound interrupt.
IRQ1	\$79	TINT	32 Hz tick interrupt.
IRQ2	\$7A	SSINT	Serial slave interrupt.
IRQ3	\$7B	EINT0	External interrupt input 0.
IRQ4	\$7C	EINT1	External interrupt input 1 (inverted).
IRQ5	\$7D	EINT2	External interrupt input 2 (inverted).
IRQ6	\$7E	FRC1OI	FRC1 overflow interrupt .
IRQ7	\$7F	FRC2OI	FRC2 overflow interrupt.

PSU subsystem

SIBO architecture power supplies all differ depending on the system requirements. MC and HC power supplies have higher current ratings to support a wide variety of expansion devices and are based around 7.2 volt NiCd batteries. The Series 3 and its derivatives have a lower current rating and are based around two 1.5 volt alkaline batteries.

MC and HC power supplies are based on a full custom liner ASIC known as *ASIC3*. This custom chip is manufactured by Maxim and has the Maxim part number MAX616. The power supply provides a number of switched power rails from one unregulated supply rail known as *Vsup*. This supply is typically connected to the main battery in the system and can be in the range 5.5 to 12 volts. These supply rails are derived from *Vsup* from either PNP pass transistor linear regulators or from switch mode regulators. Series 3 power supplies generate a sub-set of the rails listed below from a number of switch mode regulators.

SIBO architecture power supplies all operate in four principal states, these are; standby, backup idle and operating.

The standby state

In the standby state the main oscillator and the processor are shut down. *ASIC1*, *ASIC2* and the memory are powered and the real time oscillator is running to keep time. The timer in *ASIC1* is used in this mode to keep time while the processor is powered down. All other supply rails are switched off in this state.

The backup state

The backup state only exists when the main battery supply has failed or is not present. This state is the same as the standby state except that *Vcc1* is now powered from the backup battery input and switching to the operating state is inhibited.

The idle state

In the idle state the processor is powered up and the main system oscillator is running: the switchable supply rails may be on depending on software instructions. The processor clock is stopped and the processor is idle and waiting for an event i.e. an interrupt. This means that the bus is not active apart from LCD screen refresh accesses.

The operating state

The operating state is the same as the idle state except that the processor clock is running and the bus is active.

Vcc1 supply

Vcc1 is a nominal +5 volt supply that is not programmable and is present in all states. On MC and HC power supplies it is derived from *Vsup* via a PNP pass transistor, the base drive of which is supplied by

ASIC3, this configuration is the same for all the linear regulators. On Series 3 power supplies it is derived from a DC to DC converter and on *ASIC9* based systems the nominal voltage is 3.3 volts.

Vcc1 supplies *ASIC1* and *ASIC2* or *ASIC9* plus all the memory. The Vcc1 supply is current limited to a few hundred micro-amps when the system is in the standby state.

Vcc2 supply

Vcc2 is a +5 volt supply that is either derived from Vsup in the same way as Vcc1 or from a DC to DC converter. Vcc2 is disabled when the system is in the standby state and is automatically switched on when the operating or idle state is entered. Vcc2 powers the processor and any other devices that can be shut down in the standby state. Vcc2 is also connected to the expansion ports for powering peripheral devices. Vcc2 is fixed at 5 volts even in a 3.3 volt system.

Vcc3 supply

Vcc3 is a software switchable supply. It is either 5 volts or 3.3 volts depending on the voltage of Vcc1. It is connected to the LCD display module and powers the display drivers.

Vcc4 supply

Vcc4 is a software switchable +5 volt supply. It is connected to the positive supply to the sound sub-system. It is only present in MC and HC power supplies.

Vcc5 supply

Vcc5 is a software switchable +5 volt supply that is either derived from Vsup in the same way as Vcc1 or switched from Vcc2. It is connected to the 5 volt supply pin on the external SSD slots.

Vee1 supply

Vee1 is a software switchable negative supply rail which is connected to the LCD module to provide the LCD bias voltage. The voltage of Vee1 can be programmed in software to provide software adjustment of the LCD contrast. Vee1 is generated using a fly-back switch mode regulator. The voltage generated is determined by comparing the feedback voltage to the output of a DAC.

Vee2 supply

Vee2 is a software switchable negative supply rail which is connected to the negative supply of the sound sub-system. Vee2 is specified as -5 volts plus or minus 5% and is generated from Vsup using a fly-back switch mode regulator which is controlled by *ASIC3*. Vee2 is only present on MC and HC power supplies.

Vh supply

Vh is a software switchable positive supply rail which is connected to the SSD slots. It is used in the SSD's to provide the programming voltage needed to write and format Flash memory devices. Vh is generated using a fly-back switch mode regulator is specified to be +16 volts.

The second part of the document describes the various methods used to collect and analyze data. It highlights the need for a systematic approach to data collection and the importance of using reliable sources of information.

The third part of the document discusses the role of the auditor in the financial system. It explains that the auditor's primary responsibility is to provide an independent assessment of the financial statements and to ensure that they are free from material misstatements.

The fourth part of the document describes the various types of audits that can be performed. It includes a discussion of the different types of audit procedures and the factors that influence the choice of audit method.

The sixth part of the document describes the various types of audit reports that can be issued. It explains that the auditor's report is a key document in the financial system and that it provides a summary of the audit findings and conclusions.

The eighth part of the document describes the various types of audit procedures that can be used. It includes a discussion of the different types of audit tests and the factors that influence the choice of audit procedure.

The tenth part of the document describes the various types of audit procedures that can be used. It includes a discussion of the different types of audit tests and the factors that influence the choice of audit procedure.

The twelfth part of the document describes the various types of audit procedures that can be used. It includes a discussion of the different types of audit tests and the factors that influence the choice of audit procedure.

CHAPTER 2

THE SIBO SERIAL PROTOCOL

The SIBO serial protocol is a general purpose method of bidirectional serial data transfer. It has been designed to communicate between a controlling device and a number of slave devices

The interface consists of 2 wires as follows:

CLK Clock output from the controller to the slaves. Nominally 3.84 MHz.

DATA A bidirectional synchronous data line.

The data is transferred using a series of 12 bit frames including 8 data bits each. This equates to a theoretical maximum data transfer rate of approximately 312 Kbytes/second. Other bits of the frame contain control information.

The "system" is generically defined by 2 protocol layers:

The Physical layer defining the hardware interface and frame structure.

The Transport layer defines system control and register transfers between the controller and the slaves.

These layers are described in detail in the next two sections.

Using this system a large number of higher level implementations can be defined. Initially these will include:

- 1) An eight channel controller. This will be embodied within all SIBO computers to allow up to 8 isolated channels to be accessed.
- 2) Power control IC. This is also internal to the computers and operates as a slave.
- 3) Memory pack. This is an external plug-in device operating as a slave.

In normal use the controller will communicate to slaves in a point to point configuration. Multidrop configurations with a number of slaves attached to one channel of the controller are also supported.

Hardware Interface

This section specifies the timing and hardware required by the SIBO serial protocol.

This consists of two lines that switch at 5V CMOS voltage levels:-

CLK A clock output from the controller to the slaves. Nominally 3.84 MHz.

DATA A bidirectional synchronous data line.

Clock Line

This line is used to synchronously clock data between the controller and slaves. It is always output from the controller.

The clock should only be active during the transfer of data. At all other times it is tristate pulled low.

Clock Timing Parameters

Symbol	Parameter	Min	Typ	Max	Units
Tckh	Width of Clock High	65	130	-	nSec
Tckl	Width of Clock Low	130	130	-	nSec
Tcyc	Cycle time of clock	195	260	-	nSec
Fck	Clock Frequency		3.84	5.12	MHz

Data line

This is a bidirectional line used to transfer data synchronously between the controller and slaves. The direction of the data line is not determined by the physical layer but by the control information in the transport layer. This is described in the next section.

When no data transfers are in progress the data line is always set to input on both the controller and slaves. This line is pulled low.

Data Timing Parameters**Data Received by Controller**

Symbol	Parameter	Min	Typ	Max	Units
Trdset	Data set-up time	50			nSec
Trdhold	Data hold time	10			nSec

Data Transmitted by Controller

Symbol	Parameter	Min	Typ	Max	Units
Trdset	Data set-up time	Tclk-90			nSec
Trdhold	Data hold time	Tclk			nSec

Data is changed on the falling edge of clock by the transmit device and latched into the receiving device on the rising edge of clock.

The Physical layer

This section specifies the low level protocol of the SIBO serial protocol.

The physical layer protocol consists of a series of 12 bit frames. There are four types of frames:

Null frames	Transmitted by controller to synchronize slaves.
Control frames	Control information transmitted by controller to slaves.
Data output frames	Data frame transmitted by controller to slaves.
Data input frames	Data frame received by controller from a slave.

Frame structure

All 12 bit frames have the following structure:-

Bit	0	1	2	3	4	5	6	7	8	9	10	11
Name	ST	CTL	I1	D0	D1	D2	D3	D4	D5	D6	D7	I2

ST Start bit.	This bit goes high to indicate the start of a valid frame.
CTL Control bit.	When low indicates this is a control frame. High indicates a data frame.
I1 Idle bit.	Used to turn around direction of data line. Normally Low.
D0-D7 Data bits.	Used to turn around direction of data line. Normally low.
I2 Idle bit.	

Null Frame

This is a special frame transmitted by the controller to ensure all slaves are synchronized. It is generated by transmitting 12 clock pulses with the data line set to input. Since the data line is pulled low this results in 12 zeros being transmitted.

Control frame.

This frame is transmitted from the controller to one or more slaves. The data line is an output from the controller throughout the whole frame.

The bits in the frame have the following value in a control frame.

ST Start bit.	This bit goes high to indicate the start of a valid frame.
CTL Control bit.	Low to indicate this is a control frame.
I1 Idle bit.	Set low.
D0-D7 Data bits.	8 bits of control information.
I2 Idle bit.	Set low.

Data Output Frame.

This frame is transmitted from the controller to one or more slaves. The data line is an output from the controller throughout the whole frame.

The bits in the frame have the following value in a data output frame.

ST Start bit.	This bit goes high to indicate the start of a valid frame.
CTL Control bit.	High to indicate this is a data frame.
I1 Idle bit.	Set low.
D0-D7 Data bits.	8 bits of transmitted data.
I2 Idle bit.	Set low.

Data Input Frame.

This frame is received by the controller from a slave. The data line is an output from the controller for cycles 1 and 2 and input to the controller for cycles 4 to 11.

The bits in the frame have the following value in a data input frame.

ST Start bit.	Output from controller. This bit goes high to indicate the start of a valid frame.
CTL Control bit.	Output from controller. High to indicate this is a data frame.
I1 Idle bit.	Used to turn around direction of data line. Both controller and slave should tristate the data line during this bit. This bit should be low due to pull down resistor on data line. The controller changes the data line from output to input at the end of cycle 2. The slave changes the data line from input to output at the start of cycle 4.
Data bits.	Output from slave 8 bits of data transmitted by slave. Controller sets data line to input during these bits.

I2 Idle bit.

Used to turn around direction of data line. Both controller and slave should tristate the data line during this bit. Should be low due to pull down resistor on data line. The slave changes data line from output to input at the end of cycle 11.

Data line direction

The following table summarises the direction of the data line.

Condition	CONTROLLER		SLAVE	
	CK	DATA	CK	DATA
Outside Frame	T	I	I	I
Null frame	O	I	I	I
Control Frame	O	O	I	I
Data output from controller	O	O	I	I
Data input to controller:-				
Cycles 1-2	O	O	I	I
Cycle 3	O	I	I	I
Cycles 4-1	O	I	I	O
Cycle 12	O	I	I	I

Key T Tristate
 I Input
 O Output

The Transport layer

This section specifies the transport level protocol that operates above the SIBO serial communication physical layer.

The transport layer protocol controls the serial communication between the SIBO Protocol Controller (SPC) and a number of SIBO Protocol Slave (SPS) devices. The following rules apply:

- 1) The interface is controlled by the writing of control bytes from the controller to the slaves. Control bytes cannot be written by the slaves. Unsolicited data cannot be sent from the slave to the controller.
- 2) The controlling device contains two registers to communicate to the slaves. These are the control register (byte, write only) and the Data register (byte or word, read/write).

Control bytes are transmitted to the slaves by writing to the control register.

The format of the control byte is as follows:-

Bit	7	6	5	4	3	2	1	0
Name	S	x	x	x	x	x	x	x

The control word can have 2 distinct formats depending on the setting of bit 7 the Select (S) bit:

Select = 0

This is the slave select mode. This mode is for selecting, deselecting and resetting slaves.

Select = 1

This is the slave control mode. This mode is for communicating with a slave which has been previously selected using the select slave command.

Slave select mode

The format of the slave select byte is as follows:-

Bit	7	6	5	4	3	2	1	0
Name	0	R	I	I	I	I	I	I

Key: R = single reset bit, IIIII = 6 bit ID field.

The 6 bit ID field is a property only of the slave. No slave may have an ID of zero, hence there can be 63 different slaves connected to one controller.

The reset bit (R) controls whether the slave(s) are selected or reset. If R = 0 slave(s) are reset, R = 1 slave(s) are selected.

Slave select control bytes can be summarised by the following table:

S	R	ID	Description
0	0	0	Reset all slaves
0	0	xx < > 0	Reset specific slave with ID = xx
0	1	0	Deselect slave (does not reset slave)
0	1	xx < > 0	Select slave with ID=xx and read slave info (see below).

The Reset function is dependant on the slave. It would normally put the slave into a known passive reset state.

Select Slave with ID=xx (S=0,R=1)

This is a special command that causes a slave with ID=xx to transmit to the controller an 8 bit information field. This field depends entirely on the slave but must be non zero. A reply of 0 indicates that there is no slave of the requested ID present.

Slave control mode

This mode is for communicating with a slave which has been previously selected using the select slave command described above.

The format of the control word in slave select mode is as follows:

Bit	7	6	5	4	3	2	1	0
Name	1	R/W	B/W	S/M	X	X	X	X

Key

R/W Read/write select. 0 = write, 1 = read

B/W Data transfer size. 0 = 1 byte transfer, 1 = word (2 byte transfer).

S/M Single/Multi transfer mode. 0 = single, 1 = multibyte.

XXXX 4 bits of data to slave.

Note the meaning of the 4 bits of data (XXXX) is entirely dependent on the slave.

The settings of R/W,B/W,S/M bits in the control word determine the size, type and direction of subsequent data transfers in the following manner:

R/W	B/W	S/M	
0	0	0	write a single byte to slave
0	0	1	write a number of single bytes to slave
0	1	0	write a byte pair to slave
0	1	1	write a number of byte pairs to slave
1	0	0	read a single byte from slave
1	0	1	read a number of single bytes from slave
1	1	0	read a byte pair from slave
1	1	1	read a number of byte pairs from slave

Write a single byte

This command readies the currently selected slave to receive a byte of data and sets up the controller so that the next byte (or the LSB of a word) written to its data register will be transmitted to that slave. Anything further written to the controller's data register will have no effect.

Write a number of single bytes

This command readies the currently selected slave to receive a number of sequential bytes of data. The slave will expect to receive data bytes until another control byte is received. The controller is set up so that the next byte (or the LSB of a word) written to its data register will be transmitted to that slave. All subsequent bytes written to the controller's data register will be transmitted to the slave. This will continue until another byte is written to the controller's control register.

Write a byte pair

This command readies the currently selected slave to receive two bytes of data and sets up the controller so that the next word written to its data register will be transmitted to that slave (LSB first). Anything further written to the controller's data register will have no effect.

Write a number of byte pairs

This command readies the currently selected slave to receive a number of sequential byte pairs of data. The slave will expect to receive byte pairs until another control byte is received. The controller is set up so that the next word written to its data register will be transmitted to that slave (LSB first). All subsequent words written to the controller's data register will be transmitted to the slave. This will continue until another byte is written to the controller's control register.

Read a single byte

This command triggers a byte to be transmitted from the selected slave to the controller. This byte can then be read from the LSB of the controller's data register. Further reads of the controller's data register will return the same data but have no effect on the protocol.

Read a number of single bytes

This command triggers a byte to be transmitted from the selected slave to the controller. This byte can then be read from the LSB of the data register. This read will trigger the next byte to be transmitted to the data register of the controller. All subsequent reads of the controller's data register will trigger further bytes to be transmitted to the controller. This will continue until another byte is written to the controller's control register.

Read a byte pair

This command triggers a byte pair to be transmitted from the selected slave to the controller. This word can then be read from the controller's data register. Further reads of the controller's data register will return the same data but have no effect on the protocol.

Read a number of byte pairs

This command triggers a byte pair to be transmitted from the selected slave to the controller. This word can then be read from the controller's data register. This read will trigger the next byte pair to be transmitted to the data register of the controller. All subsequent reads of the controller's data register will trigger further byte pairs to be transmitted to the controller. This will continue until another byte is written to the controller's control register.

Timing

The time taken for commands to be processed and data sent is shown below. The time is given in SIBO pack protocol clock cycles. The length of a clock cycle is nominally 260 nanoseconds for a clock frequency of 3.84 MHz.

Receive and process the control byte	12 cycles
Byte transfer to or from slave	12 cycles
Byte pair transfer to or from slave	24 cycles

When writing to the controller's data and control registers the following rules apply:-

- 1) After writing to the control register there must be a delay of at least 12 cycles before the data register is accessed or another control word is written.
- 2) To read a word from the data register after the command to read byte pair is issued there must a delay of at least 12 (for control byte) + 24 (for the byte pair transfer) = 36 cycles.
- 3) To perform a multiple byte pair write there must be a delay of at least 12 cycles after the command is written to the control register before the first word can be written to the data register and a delay of at least 24 cycles between subsequent writes to the data register.

States

A slave can be in one of 5 states. Note a control byte can be received and interpreted at any time.

- 1) Waiting to receive a data byte or control byte
- 2) Waiting to receive a data byte pair or control byte
- 3) Waiting to transmit a data byte or control byte
- 4) Waiting to transmit a data byte pair or control byte
- 5) Waiting to receive control byte only

The following table shows the state after each control word is received

Control Word	Comment	State
S R IIIII		Slave select
0 0 0	Reset all	RC
0 0 < > 0	Reset slave	RC
0 1 0	Deselect slave	RC
0 1 < > 0	Select slave	TB
S RW B2B SM XXXX		Slave control
1 0 0 0 x	write byte	RB
1 0 0 1 x	write bytes	RB
1 0 1 0 x	write byte pair	RBP
1 0 1 1 x	write byte pairs	RBP
1 1 0 0 x	read byte	TB
1 1 0 1 x	read bytes	TB
1 1 1 0 x	read byte pair	TBP
1 1 1 1 x	read byte pairs	TBP

The first part of the document is a letter from the President of the United States to the Congress, dated January 1, 1863. It is a very important document, as it is the first time that the President has addressed the Congress since the beginning of his administration.

The second part of the document is a report from the Secretary of the Treasury, dated January 1, 1863. It is a very important document, as it is the first time that the Secretary has reported to the Congress since the beginning of his administration.

The third part of the document is a report from the Secretary of the Interior, dated January 1, 1863. It is a very important document, as it is the first time that the Secretary has reported to the Congress since the beginning of his administration.

The fourth part of the document is a report from the Secretary of the War, dated January 1, 1863. It is a very important document, as it is the first time that the Secretary has reported to the Congress since the beginning of his administration.

The fifth part of the document is a report from the Secretary of the Navy, dated January 1, 1863. It is a very important document, as it is the first time that the Secretary has reported to the Congress since the beginning of his administration.

The sixth part of the document is a report from the Secretary of the State, dated January 1, 1863. It is a very important document, as it is the first time that the Secretary has reported to the Congress since the beginning of his administration.

The seventh part of the document is a report from the Secretary of the War, dated January 1, 1863. It is a very important document, as it is the first time that the Secretary has reported to the Congress since the beginning of his administration.

The eighth part of the document is a report from the Secretary of the Navy, dated January 1, 1863. It is a very important document, as it is the first time that the Secretary has reported to the Congress since the beginning of his administration.

CHAPTER 3

THE SIBO EXPANSION PORTS

The SIBO architecture allows for two types of expansion device, extended internal expansion and reduced external expansion. Extended internal expansion slots provide direct I/O with the processor, a SIBO serial channel, and connection to the power supply. It allows for higher powered expansion devices to be added by including a direct connection to the main 7.2 volt battery. The MC and HC products both support two independent extended internal expansion slots. Reduced external expansion is based on a single SIBO serial channel only and has very limited power available (< 25 mA). The Series 3 and 3a supports a single reduced external expansion slot only.

Extended internal expansion

Physical connector

The extended expansion port is made up of a 25 way single row connector spaced on a 0.1 inch pitch as illustrated in Figure 2.

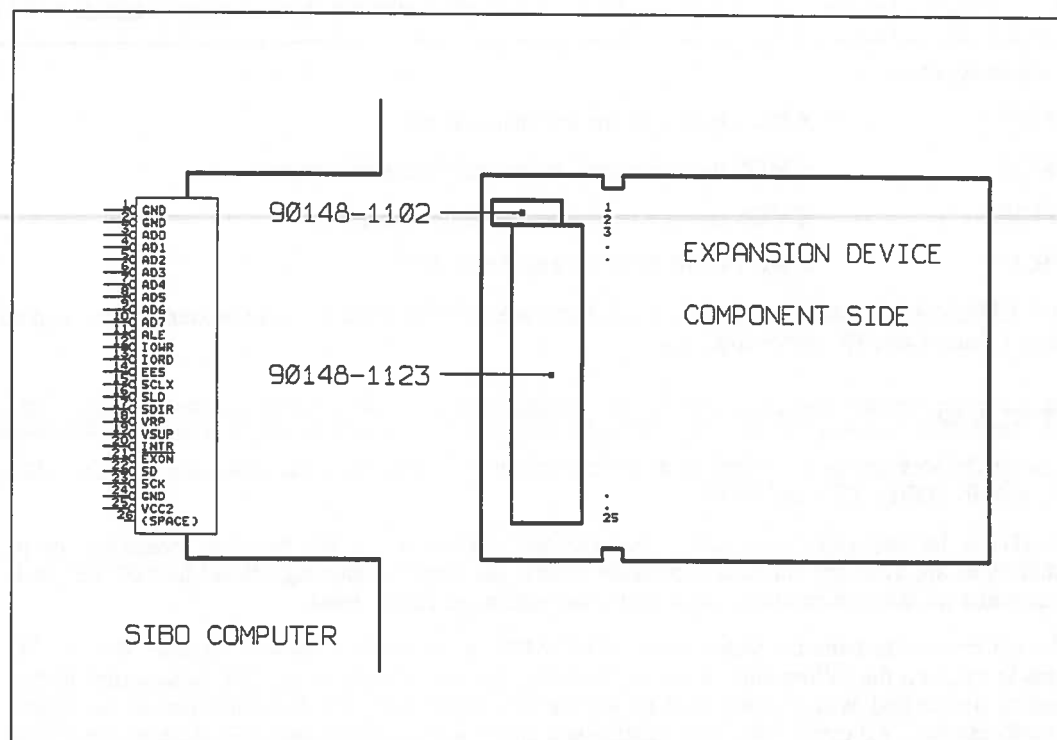


Figure 2

The 26th position is a polarising key and should be left blank. The correct mating connector on the expansion device is made up from a number of Molex C-Grid series 90148 connectors. Pins 1 and 2 are

GND and should have there own connector placed nearer the board edge to ensure the GND connection is made first when the expansion device is inserted. The required connectors are Molex 90148-1102 for the GND contacts and Molex 90148-1123 for the signal contacts.

Signal definition

The following table defines the 25 way expansion connector:

Pin No.	Name	Sig. type	Comments
1	GND	Power	Should mate first when device inserted
2	GND	Power	Should mate first when device inserted
3	AD0	B CMOS	8 bit multiplexed address and data bus pulled low with 100 K resistors
4	AD1	B CMOS	
5	AD2	B CMOS	
6	AD3	B CMOS	
7	AD4	B CMOS	
8	AD5	B CMOS	
9	AD6	B CMOS	
10	AD7	B CMOS	
11	ALE	O CMOS	Address latch enable - high when valid address on AD0-AD7
12	IOWR	O CMOS	I/O write strobe - active high, data valid on falling edge of IOWR
13	IORD	O CMOS	I/O read strobe - high when device can place valid data on AD0-AD7
14	EES	O CMOS	External Expansion Select - high during I/O cycles to expansion device
15	-	-	Reserved
16	-	-	Reserved
17	-	-	Reserved
18	Vrp	Power	Used to enable reprogramming of internal system Flash memory
19	Vsup	Power	Unregulated battery voltage - present all the time
20	INTR	I CMOS	Active high interrupt input
21	EXON	I CMOS	Active low input pulled up to Vcc1 - pull low to switch machine on
22	SD	B CMOS	SIBO serial protocol data line - pulled low
23	SCLK	T CMOS	SIBO serial protocol clock line - Hi-Z in standby needs a pull down
24	GND	Power	
25	Vcc2	Power	+5 volt supply, switched off in standby. Max current available = 50 mA

The signal types are:

O CMOS	CMOS output to the expansion device.
B CMOS	CMOS bi-directional line to the expansion device.
T CMOS	CMOS tri-state output to the expansion device.
I CMOS	CMOS input from the expansion device.

All the CMOS signals including the AD0-AD7 bus are buffered from the main system busses and so present a load of one HC series logic gate.

Direct I/O

Expansion devices can be connected to direct processor I/O space using the following signals; AD0-AD7, ALE, IOWR, IORD, EES and INTR.

AD0-AD7 is the least significant half of the multiplexed address and data bus, this means that up to 128 I/O addresses are available for each expansion device. As only the least significant half of the bus is available and no bus conversion is done only even addresses can be used.

ALE must be used to latch the address from AD0-AD7 for devices that require a stable address. The address is valid on the falling edge of ALE. Note that A0 will always be low for valid writes to the expansion device and as such should not be used as an address line, A1 should be used as the lowest order address line. A0 can be used as an additional enable signal to stop odd I/O accesses disturbing the expansion device (see example 1).

EES is the External Expansion Select and is high during all I/O accesses to the expansion device, i.e. for I/O reads and writes to address range 100 to 1FF hex. for expansion port 1, and 200 to 2FF for expansion port 2.

IOWR is an active high signal which is high during all I/O write bus cycles. The data on AD0-AD7 is guaranteed to be stable before the rising edge of IOWR and after the falling edge of IOWR.

IORD is an active high signal which is high during all I/O read bus cycles. The AD0-AD7 bus is guaranteed to be tri-state before the rising edge of IORD and after the falling edge of IORD. The expansion device must present valid data on the bus when IORD is high, see the timing details below.

INTR is an active high interrupt input to *ASIC1*. This can be used as a directly readable bit or as a dedicated interrupt input. It must not be driven high when the system is in the standby state as this input is pulled down and will cause excessive standby current consumption.

Figure 3 shows the timing of the I/O write and read cycles.

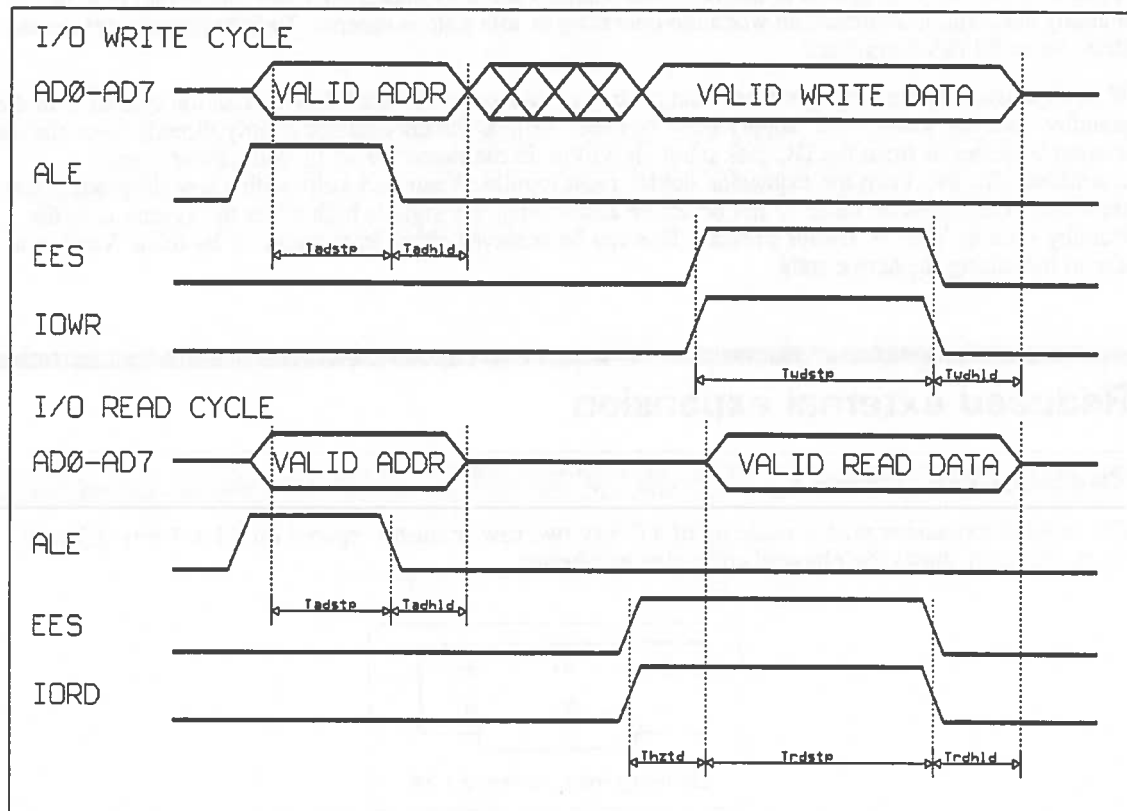


Figure 3

Symbol	Parameter	Min	Typ	Max	Units
Tdstp	Address set up time	30	50	185	nSec
Tdhld	Address hold time	25	40	-	nSec
Twdstp	Write data set up time	130	195	-	nSec
Twrcyc	Write cycle pulse width	130	195	390	nSec
Twdhld	Write data hold time	40	75	-	nSec
Thztd	Time from Hi-Z to data active	0	-	120	nSec
Trdstp	Read data set up time	65	-	-	nSec
Trdhld	Read data hold time	0	-	-	nSec

Note the typical values given are for an MC with a system oscillator of 15.36 MHz, the HC times will be approximately twice as long. The minimum times given are for projected higher speed MC devices with oscillator frequencies of up to 23.04 MHz.

The SIBO serial channel

A single SIBO serial protocol channel is provided on each expansion port. Expansion port 1 is connected to serial channel 5, expansion port 2 is connected to serial channel 6. The serial channel clock can be continuously enabled to provide a free running clock for expansion devices. The frequency is fixed at 1.536 MHz regardless of the system clock frequency. This frequency is a multiple of the SLD clock rate and of all normal RS232 baud rates.

Power supplies

Two power supplies are available for expansion devices these are Vcc2 and Vsup.

Vcc2 is a +5 volt supply that is derived from Vsup. Vcc2 is switched off when the system is in the standby state and is switched on when the operating or idle state is entered. Each expansion device can draw up to 50 mA from Vcc2.

If an expansion device requires more than 50 mA or cannot be powered down when the system is in the standby state the Vsup power supply must be used. Vsup is the unregulated supply directly from the main system batteries or from the DC jack input. It will be in the range 5.5 to 12 volts under normal conditions. To use Vsup the expansion device must regulate Vsup to 5 volts with a low drop-out linear regulator. Care must be taken to not be active and driving any signals high when the system is in the standby state as Vsup is always present. This can be achieved either in software or by using Vcc2 as a signal indicating the active state.

Reduced external expansion

Physical connector

The reduced expansion port is made up of a 6 way two row connector spaced on a 2 x 3 way 0.1 inch pitch. Figure 4 shows the physical connector numbering.

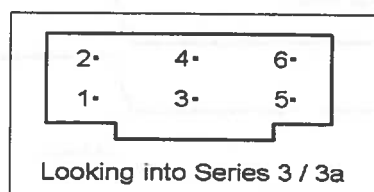


Figure 4

Signal definition

The following table defines the 6 way expansion connector:

No.	Name	Sig. type	Comments
1	MSD	I/O	Master SIBO serial protocol data line - pulled low
2	MCLK	O	Master SIBO serial protocol clock line - low in standby
3	Vcc	Power	+5 volt supply, switched off in standby. Limited to 25 mA
4	GND	Power	Signal ground, this signal should mate first when connector inserted
5	SSD/INT	I/O	Slave SIBO serial data line or active high interrupt input
6	SCK/EXON	I	Slave SIBO serial clock line or active high switch on input

MSD and MCLK form a single master SIBO serial protocol channel, this is normally channel 7. The serial channel clock can be continuously enabled to provide a free running clock for expansion devices. The frequency is fixed at 1.536 MHz regardless of the system clock frequency. SDS/INT and SCK/EXON are both dual function pins. SDS and SCK form a single slave SIBO serial protocol channel, this can be combined with MSD and MCLK to form a bidirectional high speed data link. SDS/INT can also be used as an active high interrupt input. The function of SDS/INT can be programmed in ASIC2 or ASIC9. A rising edge on the SCK/EXON input will bring the system out of the standby state into the operating state. VCC is a +5 volt supply that is switched off when the system is in the standby state and

is switched on when the system is in the operating or idle state. The maximum current that can be drawn is 25 mA.

Example expansion devices

Two examples are given to illustrate using the direct I/O expansion signals present in an extended SIBO expansion port. A further example is given using a SIBO serial channel for both reduced and extended expansion slots.

Extended expansion example one - 8 bit read/write port

Example one is a simple 8 bit read/write port. 8 LEDs can be set by an OUT instruction and the state of an 8 bit DIL switch can be read by an IN instruction. Figure 5 shows the schematic of such a port.

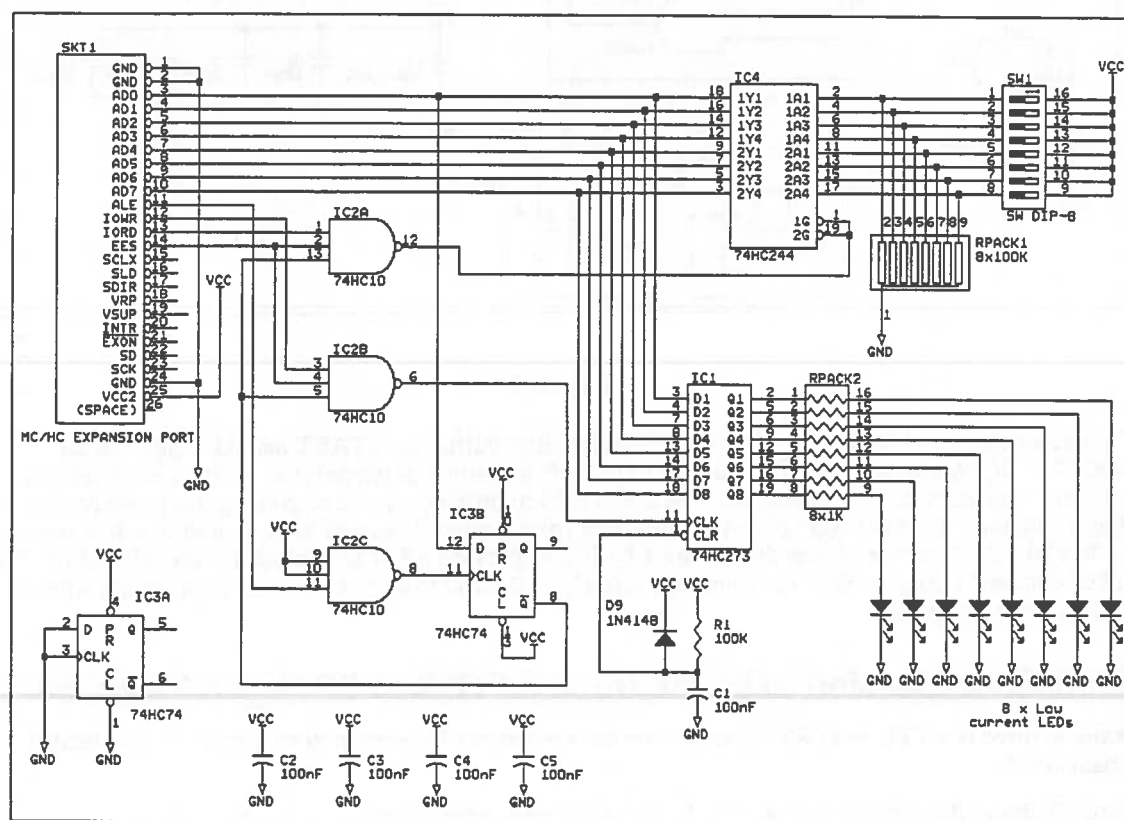


Figure 5

Note that the port is powered from Vcc2 which has a maximum current budget of 50 mA so low current LEDs must be used.

IC3B latches the state of AD0 to provide a stable A0 line, this is used as an additional enable signal, so that only even I/O cycles will effect the state of the port. IC2A combines A0, EES and IORD to provide a negative going pulse during any even read to the expansion port address range. IC2B combines A0, EES and IOWR to provide a negative going pulse during any even writes to the expansion port address range. IC2C just inverts the sense of ALE so that address can be latched on the rising edge by IC3B.

Extended expansion example two - 16 channel ADC

Example two is a 16 channel analog input interface. It is based on the National Semiconductor part ADC0816. The interface is arranged so that a write to the expansion port address range will start a conversion and select the required analog channel to read. The interrupt input (INTR) is connected to the EOC signal to generate an interrupt when the conversion is complete.

Figure 6 shows the schematic of the 16 channel analog interface.

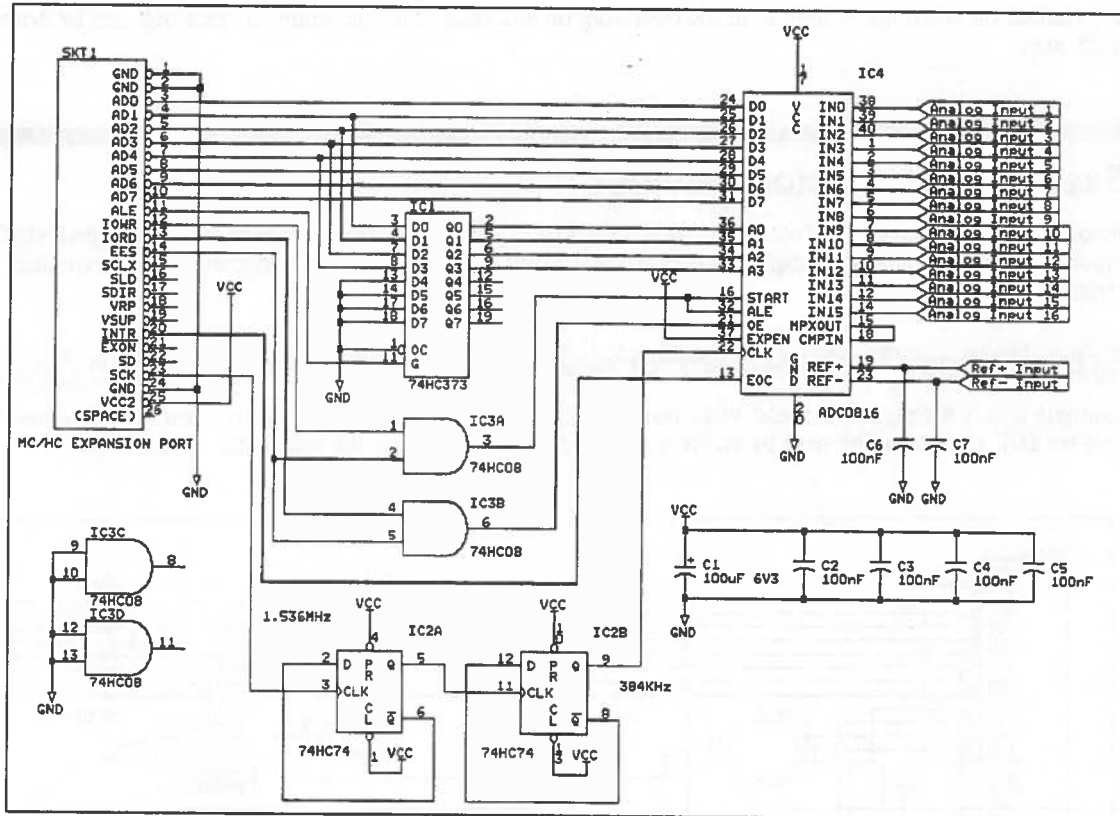


Figure 6

IC1 latches AD1 to AD4 to provide 4 stable address lines during the START and ALE pulse to the ADC0816. IC3A combines IOWR and EES to provide a positive going pulse to set the channel address and start a conversion. IC3B combines IORD and EES to provide a positive going pulse to enable the digital data onto the AD0-AD7 bus. The clock line from the serial channel SCK is used as a free running 1.536 MHz clock source, this is divided by 4 by IC2 to provide a 384 KHz clock for the ADC0816. The end of conversion signal (EOC) is connected directly to the INTR input to provide an interrupt when the conversion is complete.

Reduced expansion example three - TTL RS232 based interface

Example three is a TTL level RS232 expansion device that can be connected to a reduced or extended expansion slot.

Figure 7 shows the schematic of A TTL RS232 based expansion device.

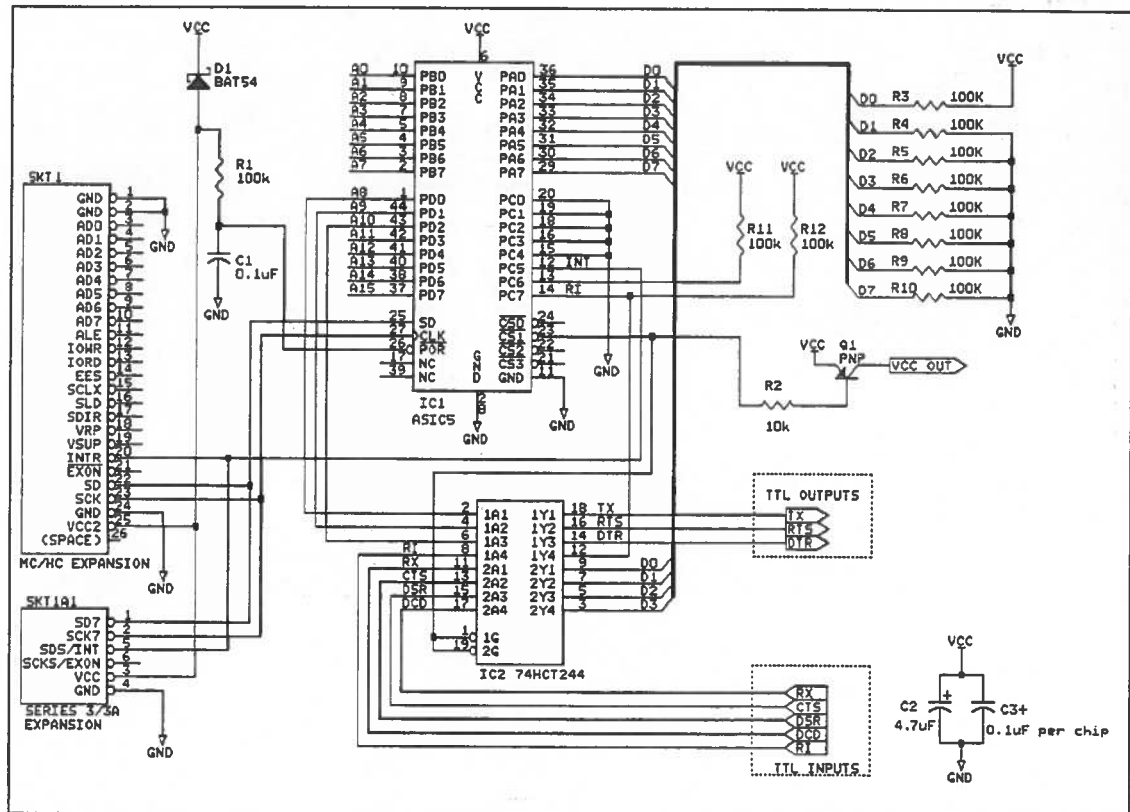


Figure 7

The interface is based around a Psion custom chip known as *ASIC5*, this device is a 44 pin plastic quad flat pack device and is available from Psion for custom expansion development. This device converts the SIBO serial channel into standard RS232 signals and allows the standard RS232 device driver contained in EPOC to be used.

Transistor Q1 provides a switched power rail for any other expansion device. This rail will automatically be switched off when the RS232 port is closed. D1 is required to isolate the supply so that the external device does not back power the SIBO computer when it enters the standby state. If the additional device does not require a supply or only uses a few micro-amps then Q1 and D1 can be omitted, Vcc and the supply for the expansion device can be connected directly to Vcc2.

If CMOS level RS232 signals are required IC1 can be replaced with a 74HC244 device. If inverted sense RS232 signals are required IC1 can be replaced with a 74HC241 device.

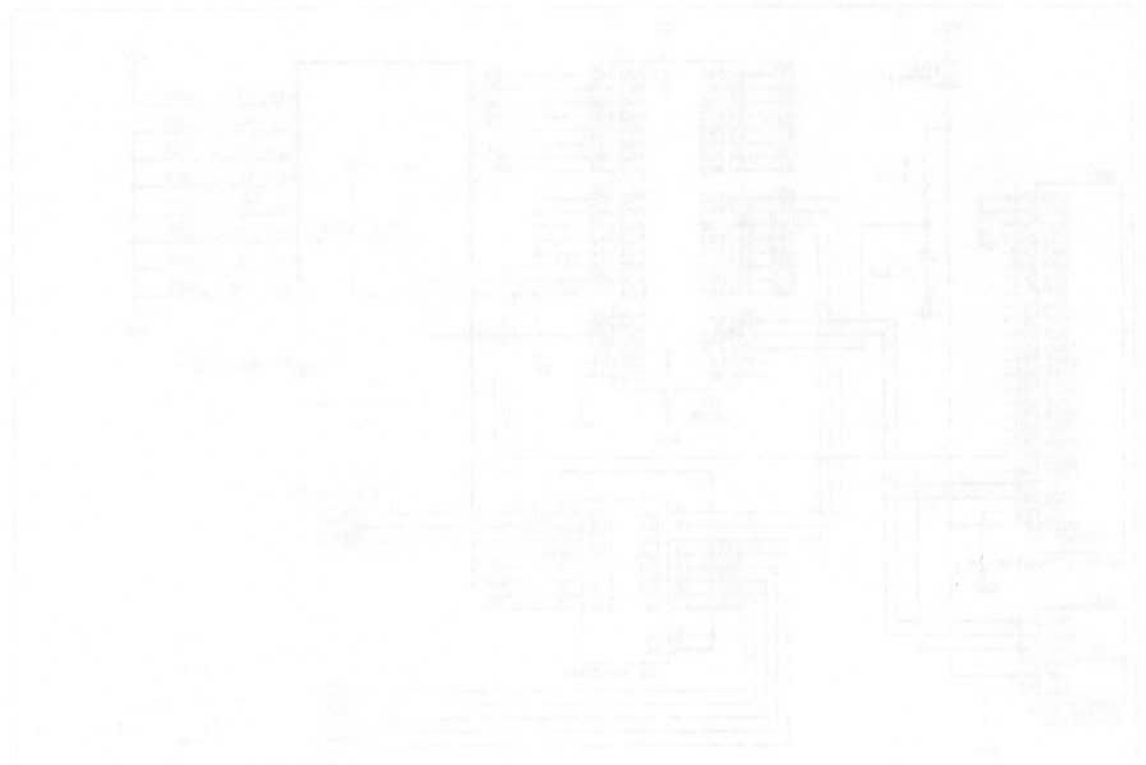


Figure 1

The diagram shows the layout of the building, including the main entrance, corridors, and rooms. The rooms are labeled with numbers and letters, and the corridors are labeled with letters. The diagram is a schematic representation of the building's layout, showing the relative positions of the rooms and corridors. The diagram is enclosed in a rectangular border, and there are various labels and lines indicating the electrical system's components and wiring paths.

INDEX

AD0-AD7 signal 20
ALE signal 20
analog interface 23
ASIC1 1
 8/16-bit bus 2
 I/O decoding 4
 interrupts 4
 memory decoding 3
 protected mode 4
ASIC2 1, 4
 serial protocol 5
ASIC3 2, 9
 power supply 8
ASIC5 25
ASIC9 1, 9
 addressing 6
buzzer 4
CODEC 7
 sound interrupt 8
Control frames 12
Data input frames 12
Data output frames 12
EES signal 20
eight channel controller 11
expansion connector 22
expansion port 5, 22
extended expansion port 19
Flash 6
 addressing 7
FRC 6, 7
 interrupt 8
free running clock 22
Free running counter 6, 7
HALT 2
HC 1
 ASIC1 2
high speed link 5
i/o timing 21
IN instruction 23
INTR signal 21
IORD signal 21
IOWR signal 21
keyboard 4
 interface 5
MC 1
 ASIC1 2
memory pack 11
Null frames 12
OUT instruction 23
power control IC 11
power supply 5
 ASIC2 4
 expansion 22
 HC 8, 9
 MC 8, 9

Series 3 8, 9
RAM
 addressing 6
 read/write port 23
 reduced expansion port 22
ROM
 addressing 6
 addressing two sites 7
RS232 interface 24
serial channel clock 22
serial transfer rate 11
Series 3 1
Series 3a 1
 ASIC9 6
 expansion support 7
SIBO Protocol Slave (SPS) devices 14
SSD 4, 5, 6, 7
 power supply 9
SSDs 5
V30H 1
 addressing 6
 ASIC9 6
 speed 6

