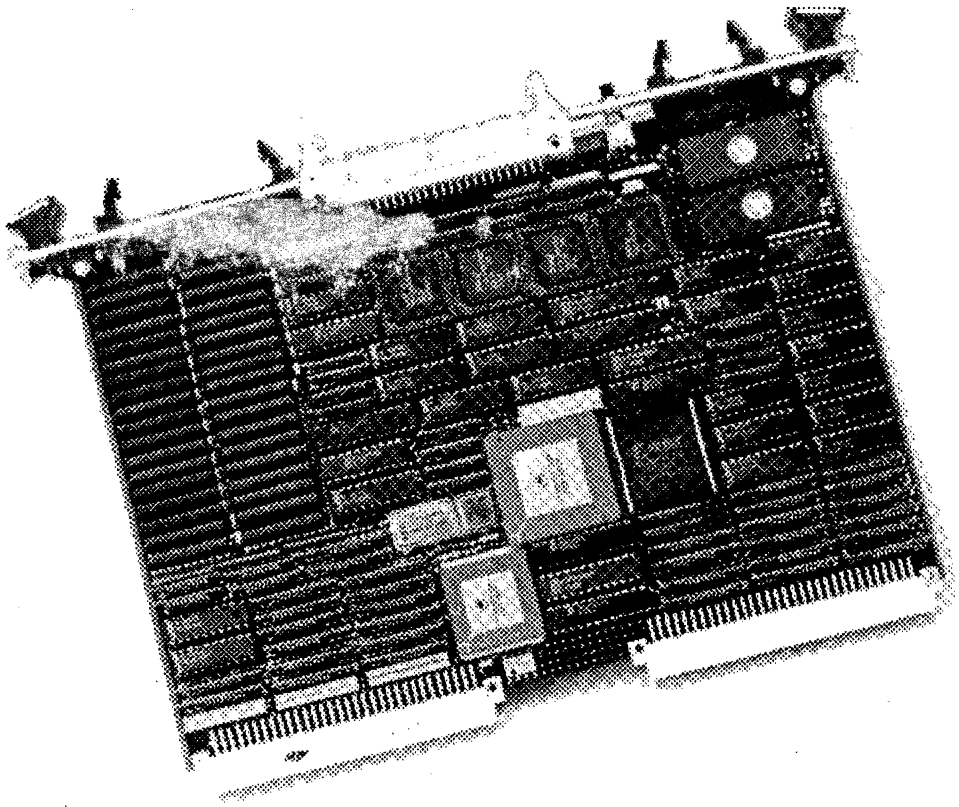
HEURIKON CORP.

HK68/V2E USER'S MANUAL



HK68/V2E USER'S MANUAL

001M528

REVISION A
OCTOBER 1989

 Heurikon Corporation *Madison, WI*

PREFACE

The purpose of this manual is to document the features of the Heurikon HK68/V2E™ microcomputer board.

This manual covers the unique features of the HK68/V2E board. Although general information, such as MPU, SCSI, CIO, and SCC programming is discussed, more detailed information is available directly from the chip manufacturers.

Feel free to contact Heurikon Corporation's Customer Support Department if you have questions. We are prepared to answer general questions as well as providing help with specific applications.

TABLE OF CONTENTS

Tables

Figures

Chapter 1

AN OVERVIEW OF THE HK68/V2E

1.1 FEATURE SUMMARY

CPU
Motorola 68020 microprocessor chip
16.67, 20 and 25 Mhz option
32-bit internal architecture
32-bit address and data paths
32 address lines
4-gigabyte addressing range
256-byte Instruction Cache. (Ref: section 3)

FPP
68881/68882 Floating Point Co-processor. Uses the IEEE-P754 Binary Floating Point Std. (Ref: section 4)

RAM
4 or 16 megabyte capacity
One parity bit per byte
Uses 1024K x 1 or 4096K x 1 DRAMs. Hardware refresh. (Ref: section 6)

EPROM
Two ROM sockets
2 Mbyte total capacity. Page Addressable ROM and EEpROM capability. (Ref: section 6)

VMEbus
32-bit addressing (4 gigabyte range)
32-bit data bus, compatible with 8-bit boards
Seven bus interrupts. (Ref: section 7)

VSB
High speed local memory expansion. Supports secondary bus masters. (Ref: section 7)

Serial I/O
Four serial I/O ports (two Zilog Z8530 Serial Communication Controllers)
Separate baud rate generators for each port
Asynchronous and synchronous modes
RS-232C interface, RS-422 option. (Ref: section 10)

SCSI ANSI X3T9.2 compatible SCSI controller
Supports up to 8 disk drive controllers or other devices
Synchronous protocol support. (Ref: section 11)

Centronics Control I/F and eight-bit output port for Centronics-type printer. (Ref: section 12)

LEDs Four user LEDs under software control
Three MPU/BUS status LEDs (Ref: section 8)

CIO Zilog Z8536 Counter/Timer and Parallel I/O Unit; three 16-bit counter/timers
Three parallel ports for on-card control functions. (Ref: section 9)

NV-RAM Nonvolatile Static RAM
256 x 4 configuration
Internal EEpROM
100 year retention
10,000 store cycle lifetime
For user definable functions. (Ref: section 6.8)

Mailbox Allows remote control of the HK68/V2E via specified VMEbus addresses
MPU halt, reset, interrupt, and on-card bus lock functions. (Ref: section 7.8)

RTC Optional Real-Time Clock module for time-of-day maintenance. With battery backup. (Ref: section 13)

1.2 BLOCK DIAGRAM

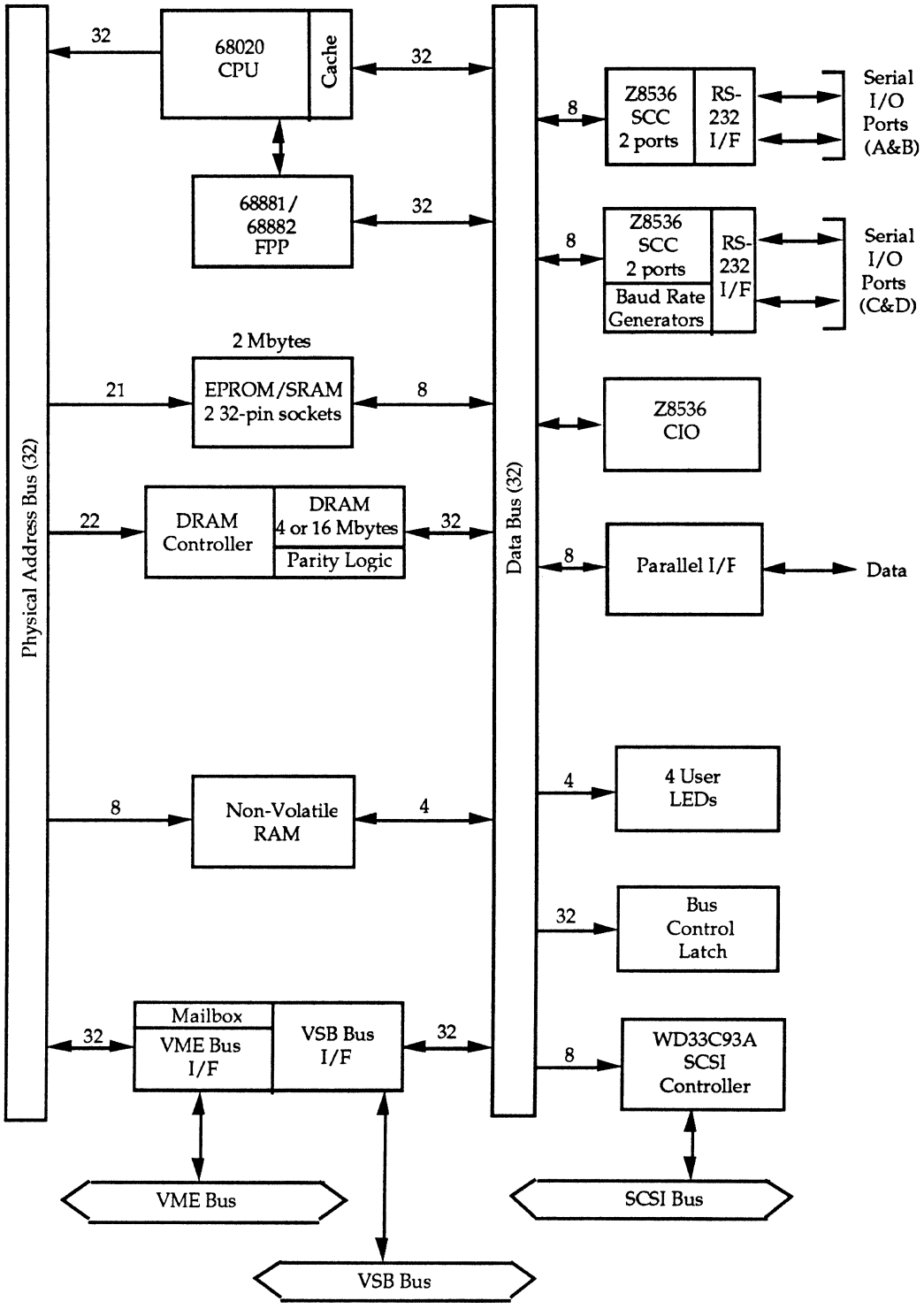


Fig. 1-1. HK68/V2E Block Diagram

Chapter 2

GETTING GOING

2.1 *INSTALLATION STEPS*

Here is what you need to get the Heurikon HK68/V2E "on-the-air":

1. Heurikon HK68/V2E Microcomputer board
2. Card cage and power supply
3. Serial I/F cable (RS-232)
4. CRT Terminal
5. Heurikon Hbug monitor and bootstrap EpROM

CAUTION: All semiconductors should be handled with care. Static discharges can easily damage the components on the HK68/V2E. Keep the board in an anti-static bag whenever it is out of the system chassis and *do not handle the board* unless absolutely necessary. Ground your body before touching the HK68/V2E board.

CAUTION: High operating temperatures will cause unpredictable operation. Because of the high chip density, fan cooling is required for most configurations, even when cards are placed on extenders.

All products are fully tested before they are shipped from the factory. When you receive your HK68/V2E, follow these steps to assure yourself that the system is operational:

1. Visually inspect the board(s) for loose components which could be the result of shipping vibrations. Visually inspect the chassis and all cables. Be sure all boards are seated properly in the card cage. Be sure all cables are securely in place.

2. Connect a CRT terminal to Serial Port B, via connector P5. If you are making your own cables, refer to section 12. Set the terminal as follows:

   - 9600 baud, full duplex.
   - Eight data bits (no parity).
   - Two stop bits for transmit data.
   - One stop bit for receive data.
   - If your terminal does not have separate controls for transmit and receive stop bits, select one stop bit for both transmit and receive.

3. Connect AC power and turn the system on.

4. Push the system RESET button. A sign-on message and prompt from the monitor should appear on the screen. If not, check your power supply voltages and CRT cabling.

5. Now is the time to read the monitor manual and the operating system literature. Short course: To boot the operating system, insert a diskette and enter 'bf' (for boot floppy) or 'bw' (to boot from Winchester.)

6. Reconfigure the jumpers, etc, as necessary for your application. See section 15 for a summary of I/O device addresses and configuration jumpers.

2.2 TROUBLESHOOTING AND SERVICE INFORMATION

In case of difficulty, use this checklist:

1. Be sure the system is not overheating.

2. Inspect the power cables and connectors. If the HK68/V2E board has power, at least one of the four user LEDs (near the Reset button) should be on.

3. If the Hbug monitor program is executing, run the diagnostics, via command 'uc' or 'um'.

4. Check your power supply for proper DC voltages. If possible, look for excessive power supply ripple or noise using an oscilloscope. Note that the use of P2 is required to meet the power specifications.

5. Check the chips to be sure they are firmly in place. Look for chips with bent or broken pins. In particular, check the EPROM.

7. Check the jumpers to be sure your board is configured properly. All jumpers should be in the "standard configuration" positions shown in section 17.3.

8. After you have checked all of the above items, call our Customer Service Department for help. Please have the following information handy:

   - The state of the user and status LEDs (near the Reset button).
   - The monitor program revision level (part of sign-on message).
   - The HK68/V2E p.c.b. serial number (scribed along card edge).
   - The complete HK68/V2E model number, including option codes.
   - The serial number of the Operating System.

If you plan to return the board to Heurikon for service, contact our Customer Service Department to obtain a *Return Merchandise Authorization* (RMA) number. Be prepared to provide the items listed above, plus your Purchase Order number and billing information if your HK68/V2E is out of warranty. If you return the board, be sure to enclose it in the anti-static bag, such as the one in which it was originally shipped. Send it prepaid to:

Heurikon Corporation
Factory Service Department
8310 Excelsior Drive
Madison, WI 53717

Please put the RMA number on the package so we can handle your problem most efficiently.

2.3 *Monitor Summary*

The HK68/V2E monitor and bootstrap program, Hbug, is contained in one EPROM. It is intended to provide a fundamental ability to check the memory and I/O devices, to manually enter a program and to down-line load or bootstrap a larger program into memory. Advanced features and utilities may be loaded from media or via an operating system.

Refer to the **Hbug User's Manual** for details on the commands and command formats.

Chapter 3

MPU SUMMARY INFORMATION

3.1 *INTRODUCTION*

This section details some of the important features of the 68020 MPU chip and, in particular, those items which are specific to the implementation on the Heurikon HK68/V2E.

3.2 *MPU INTERRUPTS*

The MPU can internally set an interrupt priority level in such a way that interrupts of a lower priority will not be honored. Interrupt level seven, however, cannot be masked off.

Level	Interrupt (bus)	Interrupt (on-card)
7	IRQ7	Parity error, highest priority,
		non-maskable, autovectored
6	IRQ6	CIO (vectored) (sub-priority:
		timer 3, port A, timer 2, port B, timer 1)
5	IRQ5	-
4	IRQ4	SCSI (autovectored)
3	IRQ3	-
2	IRQ2	SCC (vectored) (sub-priority: port C, D)
		(sub-sub-priority: rcv ready, tx ready,
		status change)
1	IRQ1	VSB Interrupt (autovectored)
0	-	Idle, no interrupt

Table 3-1. MPU Interrupt Levels

When an interrupt is recognized by the MPU, the current instruction is completed and an interrupt acknowledge sequence is initiated, whose purpose is to acquire an interrupt vector from the interrupting device. The vector number is used to select one of 256 exception vectors located in reserved memory locations (see section 3.3 for a listing.) The exception vector specifies the address of the interrupt service routine.

In case there are two interrupts pending at the same level, the on-card device is serviced before the bus interrupt. The VMEbus interrupts are masked on and off via the CIO. Refer to sections 11 and 9.4.

The SCC and CIO devices on the HK68/V2E are capable of generating more than one vector, depending on the particular condition which caused the interrupt. This significantly reduces the time required to service the interrupt because the program does not have to rigorously test for the interrupt cause. Section 7.5 has more information on the HK68/V2E interrupt logic. The VMEbus interrupts are vectored; the vector is automatically read from the interrupting device.

3.3 *MPU Exception Vectors*

Exception vectors are memory locations from which the MPU fetches the address of a routine to handle an exception (interrupt). All exception vectors are two words long (four bytes), except for the reset vector which is four words. The listing below shows the vector space as it appears to the Heurikon HK68/V2E MPU. It varies slightly from the 68020 MPU manual listing due to particular implementations on the HK68/V2E board. Refer to the MPU documentation for more details. The vector table normally occupies the first 1024 bytes of RAM, but may be moved to other locations under software control. Unused vector positions may be used for other purposes (e.g., code or data) or point to an error routine.

Vector	Address Offset	Assignment
0	000	Reset: Initial SSP (Supervisor Stack Pointer)
1	004	Reset: Initial PC (Supr Program Counter)
2	008	Bus Error (Watchdog Timer, MMU Fault)
3	00C	Address Error
4	010	Illegal Instruction
5	014	Divide by Zero
6	018	CHK Instruction (register bounds)
7	01C	TRAPV Instruction (overflow)
8	020	Privilege Violation (STOP, RESET, RTE, etc)
9	024	Trace (Program development tool)
10	028	Instruction Group 1010 Emulator
11	02C	Instruction Group 1111 Emulator
11	02C	FPP Coprocessor not present
12	030	(reserved)
13	034	(reserved)
13	034	FPP Coprocessor Protocol Violation
14	038	Format Error
15	03C	Uninitialized Interrupt
16-23	040-05F	(reserved-8)
24	060	Spurious Interrupt, not used
25	064	Level 1 autovector, VSB
26	068	Level 2 autovector, not used
27	06C	Level 3 autovector, not used
28	070	Level 4 autovector, SCSI Interrupt
29	074	Level 5 autovector, not used
30	078	Level 6 autovector, not used
31	07C	Level 7 autovector, parity error, ACFAIL
32-47	080-0BF	TRAP Instruction Vectors (16)
48-54	0C0-0DB	FPP Exceptions (8)
55-63	0DC-0FF	(reserved-8)
64-255	100-3FF	User Interrupt Vectors (192)

Table 3-2. MPU Exception Vectors

Autovectoring is used for the parity error, SCSI and VSB interrupts. Interrupts from all other devices can be programmed to provide a vector number (which would likely point into the "User Interrupt Vector" area, above). VMEbus interrupts (IRQ1 - IRQ7) are vectored; the vector is supplied by the interrupting device over the VMEbus.

The following table gives suggested interrupt vectors for each of the possible (on-card) device interrupts which could occur. Note that the listing is in order of interrupt priority, highest priority first.

Level	Vector	Device	Condition
7	31		Parity err./ACFAIL autovectored interrupt
6	96	CIO	Timer 3
	79	CIO	External Interrupt (P6-11)
	77		EEpROM 1 Ready
	75		EEpROM 0 Ready
	73		Mailbox Interrupt
	71		SCSI DMA Request
	69		VME Interrupt in Progress
	67, 65		-
	98	CIO	Timer 2
	78	CIO	Centronics Interrupt
	76,74,72		-
	70		-
	68		-
	66		-
	64		-
	100	CIO	Timer 1
	102	CIO	Timer, error
4	28	SCSI	SCSI Interface (autovectored)
2	92	SCC	Port A, Receive character available
	94		Port A, Special receive condition
	88		Port A, Transmit buffer empty
	90		Port A, External/Status change
	84		Port B, Receive character available
	86		Port B, Special receive condition
	80		Port B, Transmit buffer empty
	82		Port B, External/Status change
	93		Port C, Receive character available
	95		Port C, Special receive condition
	89		Port C, Transmit buffer empty
	91		Port C, External/Status change
	85		Port D, Receive character available
	87		Port D, Special receive condition
	81		Port D, Transmit buffer empty
	83		Port D, External/Status change
1	25	VSB	VSB IRQ Interrupt (autovectored)

Table 3-3. Suggested Interrupt Vectors

The suggested interrupt vectors for the CIO and SCC devices take into account that the lower bit and upper four bits of the vectors are shared, e.g., all CIO Port A vectors have five bits which are the same for all interrupt causes.

Each vectored on-card device has interrupt enable and control bits which allow the actual interrupt priority levels to be modified under program control by temporarily disabling certain devices.

Of course, fewer vectors may be used if the devices are programmed not to use modified vectors or if interrupts from some devices are not enabled.

If you want to use the suggested vector numbers in the above table, the proper values to load into the device vector registers are:

Device	Hex Value	Decimal Value
SCC 1 (Ports A & B):	0x50	80
SCC 2 (Ports C & D):	0x51	81
CIO, Port A:	0x41	65
CIO, Port B:	0x40	64
CIO, C/T vector:	0x60	96

Table 3-4. Device Interrupt Vector Values (Suggested)

Making your way through the Zilog CIO and SCC manuals in search of details on the interrupt logic is quite an experience. We suggest you start with these recommended readings from the CIO and SCC technical manuals:

Device	Item
CIO Z8536	Technical Manual
	Vector register: section 2.10.1
	Bit priorities: section 3.3.2
SCC Z8530	Technical Manual
	Port priorities: section 3.2.2, table 3-5
	Vector register: section 4.1.3
	Vectors: section 4.1.10, table 4-3

3.4 STATUS LEDs

There are three status LEDs which continuously show the state of the board as shown in Table 3.5:

LED	Name	Meaning
L1	Fail	The SYSFAIL line is being driven active by this board.
L2	Master	The HK68/V2E is the master on the VMEbus.
L3	Slave	The HK68/V2E is a slave on the VMEbus.

Table 3-5. Status LEDs

3.5 CONTROL PANEL INTERFACE/MPU STATUS

There are four status outputs which allow remote monitoring of the HK68/V2E processor. Connections are made through a 14 pin connector, P6.

P6 pin	Name	Meaning
2	Supr	The MPU is in the supervisor state.
4	User	The MPU is in the user state.
6		n/c
8	Halt	The MPU has halted. (Double bus fault, odd stack address or the system reset line is active.)
10	Bus	The HK68/V2E is being accessed as a slave on the VMEbus.
1,3,5,7,9	Vcc	Vcc (+5) volts

Table 3-6. Control Panel Interface (P6)

The output signals are low when true. Each is suitable for connection to a LED cathode. An external resistor must be provided for each output to limit current to 15 milliamps.

Two input signals are also provided on P6 for interrupt and reset.

P6 pin	Name	Function
P6-11	INTR*	Connected to CIO bit A7, and pull-up (Refer to section 11.1)
P6-12	Gnd	
P6-13	RESET*	When low, causes a local reset
P6-14	Gnd	

Table 3-7. Control Panel Interface (P6)

A recommended mating connector for P6 is Molex P/N 15-29-8148.

3.6 MPU CACHE CONTROL

The 68020 cache may be controlled as follows:

Address	Function (write-only)
02B0,0002	MPU Cache Control D0 = 0, cache disabled (default) D0 = 1, cache enabled

Table 3-8. MPU Cache Control

The cache control register in the MPU itself must also be set properly to enable the MPU cache.

3.7 COPROCESSORS

The HK68/V2E supports the FPP coprocessor, which is described in more detail in section 4.

Chapter 4

FLOATING POINT COPROCESSOR (FPP)

4.1 FPP FEATURE SUMMARY

The HK68/V2E allows the use of an optional MC68881 (or 68882) floating point processor chip. It runs as a coprocessor with the MPU.

FPP Feature Summary

- Allows fully concurrent instruction execution with the main processor.
- Eight general-purpose floating-point data registers, each supporting a full 80-bit extended-precision real data format (a 64-bit mantissa plus a sign bit, and a 15-bit biased exponent).
- A 67-bit ALU to allow very fast calculations, with intermediate precision greater than the extended-precision format.
- A 67-bit barrel shifter for high-speed shifting operations (for normalizing, etc.)
- 46 instruction types, including 35 arithmetic operations.
- Fully conforms to the IEEE P754 standard, including all requirements and suggestions. Also supports functions not defined by the IEEE standard, including a full set of trigonometric and logarithmic functions.
- Supports seven data types: byte, word, and long integers; single, double, and extended-precision real numbers; and packed binary coded decimal string real numbers.
- Efficient mechanisms for procedure calls, context switches, and interrupt handling.

FPP programming details are available in the 68881/68882 technical manual.

4.2 FPP BYPASS

The HK68/V2E will operate without the FPP chip. Simply unplug the FPP if it is not required. No wires or jumpers are needed.

If the Watchdog Timer is enabled, the software can determine if the FPP chip is installed. An attempt to access a non-existent FPP will result in a Watchdog timeout and a Bus Error, forcing a Line 1111 MPU Exception, vector number 11.

SYSTEM ERROR HANDLING

5.1 ERROR CONDITIONS

There are numerous events which could cause an error to occur. The responses to these events are carefully controlled.

The following error conditions may arise during MPU cycles:

Condition	Meaning
RAM Parity	Incorrect parity was detected during a read cycle from on-card RAM memory. This may be due to a true parity error (RAM data changed,) or because the memory location was not initialized prior to the read and it contained garbage. Parity errors generate a *level 7 autovector interrupt.* A pointer to the parity error handling routine should be loaded at Vector Base Register offset 00007C. Parity checking cannot be disabled.
Watchdog Timeout	During an access, usually to the bus, no acknowledge was received within a fixed time interval defined by a hardware timer. (about 100 microseconds.) This is usually the result of no bus device being assigned to the specified address. A timeout could also occur if an access from the bus is not terminated by the bus master. For an access *to* the bus, the memory cycle is terminated, the BERR *(Bus Error)* exception is taken by the MPU and execution resumes at the location specified by the exception vector.

If an access *from* the bus was in progress, no BERR exception occurs.

Double Bus Fault Another bus error occurred during the processing of a previous bus error, address error or reset exception. This error is the result of a major software bug or a hardware malfunction. A typical software bug which could cause this error would be an improperly initialized stack pointer, which points to an invalid address.

A double bus fault forces the MPU to enter the *HALT* state. Processing stops. The HALT status LED will come on. The only way out of this condition is to issue a hardware reset.

Divide by Zero The value of the divisor for a divide instruction is zero. The instruction is aborted and *vector 5* is used to transfer to an error routine.

Privileged Violation A program executing in the user state attempted to execute a privileged instruction. The instruction is not executed. Exception *vector 8* is used to transfer control.

Address Error An odd address has been specified for an instruction. The bus cycle is aborted and *vector 3* is used to transfer control.

Illegal Instruction The bit pattern for the fetched instruction is not legal or is unimplemented. The instruction is not executed. Exception *vector 4, 10 or 11* is used to transfer control.

Format Error The format of the stack frame is not correct for an RTE instruction. The instruction is aborted and exception *vector 14* is used to transfer control.

Line 1111 Emulator The FPP or PMMU Coprocessor is not present and a coprocessor instruction was fetched. The instruction is not executed. Exception *vector 11* will be taken.

FPP Exceptions The FPP Coprocessor has detected a data processing error, such as an overflow or a divide by zero. The FPP causes the MPU to take one of eight exceptions in the range of *48* to *54*.

Chapter 6

ON-CARD MEMORY CONFIGURATION

6.1 INTRODUCTION

The Heurikon HK68/V2E microcomputer will accommodate a variety of RAM and ROM configurations. There are two ROM sockets for pROM, page addressable ROM or EEpROM, 36 ZIP RAM positions, and a nonvolatile RAM. Off-card memory may be accessed via the VMEbus or the VSB.

6.2 ROM

Each ROM occupies a fixed 4-megabyte physical address space. At power-on, the MPU fetches the reset vector from the first eight locations of ROM 0 (chip position U48). The reset vector specifies the initial program counter and status register values. ROM access time must be 435 nsec or less.

Base Address	ROM	Chip
0000,0000	0	U48
0040,0000	1	U56

Table 6-1. ROM Address Summary

There are four jumpers for each ROM position which must be set according to the ROM type being used. If more than one ROM is used, they do not have to be of the same type.

PROM Type	ROM Capacity	Total Board Capacity	Jumper Positions (U48/U56) J12/J13	J11/J14	J10/J15	J9/J16
2764	8 Kbytes	16 Kbytes	C	B	x	B
27128	16 Kbytes	32 Kbytes	C	B	x	B
27256	32 Kbytes	64 Kbytes	C	B	x	A
27512	64 Kbytes	128 Kbytes	B	B	x	A
27010	128 Kbytes	256 Kbytes	B	x	B	A
27020	256 Kbytes	512 Kbytes	B	A	B	A
27040	512 Kbytes	1 Meg	B	A	B	A
27080	1 Meg	2 Meg	B	A	A	A
27513 Paged	64 Kbytes	128 Kbytes	C	B	x	B
2864 R/W EEpROM	8 Kbytes	16 Kbytes	x	B	x	B
2817 R/W EEpROM	2 Kbytes	4 Kbytes	A	B	x	B

Table 6-2. ROM Capacity and Jumper Positions

Jumpers J9, J10, J11 and J12 are for ROM 0 (U48); J13, J14, J15 and J16 control ROM 1 (U56). See section 15.3 for help in locating the jumper locations.

Each ROM contains consecutive (both even and odd) addresses. When programming pROMs, do not split even and odd bytes between the two chips.

Both ROM sockets are 32 pins. When using a 28 pin device, justify it so socket pins 1, 2, 31 and 32 are empty. Twenty four pin devices are not supported. The ROM access time must be at most 250 nanoseconds.

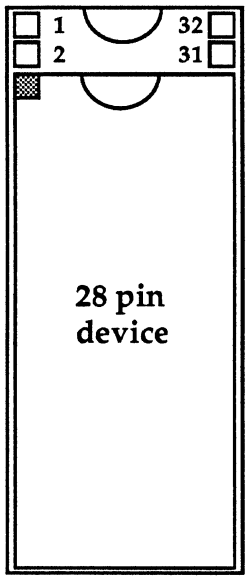


Fig. 6-1. ROM Positioning Diagram

The two ROM positions are not contiguous (although a mirror of the lower ROM will be contiguous with the upper ROM). The best way to create a contiguous image is to copy the contents of both ROMs to contiguous RAM areas.

Electrically Erasable or paged pROMs may be used. An EEpROM allows specific addresses to be changed by writing to the ROM. When writing to the EEpROM, a delay must be provided *by the software* between write operations. For the 2864, this delay is 10 milliseconds. The EEpROM Busy/Ready signals are available at the CIO to facilitate this timing; see section 9.1.

Paged ROMs allow future growth of ROM capacity, without adding address pins. A single device can contain multiple 16K byte pages. A specific page is selected by *writing* the page value to the ROM. For example, to select page three of a 27513, write 0x03 to address 0000,0000.

6.3 ON-CARD RAM

The HK68/V2E uses 36 ZIP RAM packages, each one bit wide. There is one parity bit per byte. Standard memory configurations

are four or 16 megabytes. On-card RAM occupies physical addresses starting at 0300,0000.

RAM Type	Quantity	Capacity	J17
1Meg x 1 ZIP	36	4 Megabytes	J17-A
4Meg x 1 ZIP	36	16 Megabytes	J17-B

Table 6-3. On-card RAM Capacity

6.4 ON-CARD MEMORY SIZING

The following algorithm can be used to determine the amount of on-card RAM memory installed. This procedure takes advantage of "mirrors" which exist in higher addresses when the on-card physical memory size is less than the logical memory space.

1. Clear 16 megabytes of memory starting at location 0300,0000.
2. Write 5555 (hex) to location 0300,0000.
3. Read a word from 0340,0000. If the value read is 5555 the board has four megabytes of memory installed. If the value is zero, the board has 16 megabytes of memory.

6.5 BUS MEMORY

See section 7 for details concerning the bus interface.

6.6 PHYSICAL MEMORY MAP

See section 15.2 for an I/O device address summary.

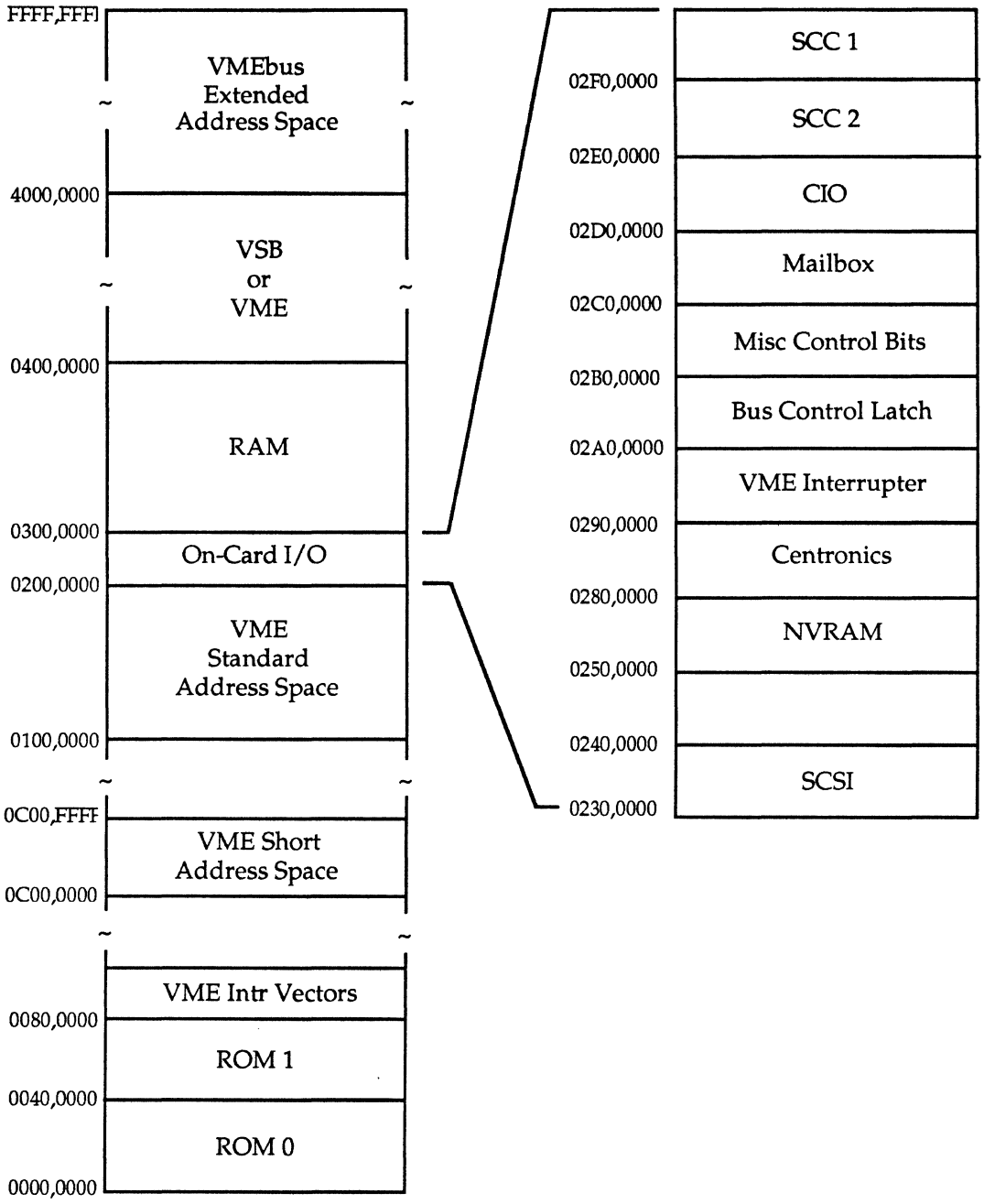


Fig. 6-2. Physical Memory Map

6.7 MEMORY TIMING

The information for this section has not been compiled for the HK68/V2E as of this writing. Please consult the factory if you need details.

6.8 NON-VOLATILE RAM

A unique feature of the HK68/V2E is its non-volatile RAM (NV-RAM), which allows precious data or system configuration information to be stored and recovered across power cycles. The RAM is configured as 256, four-bit words (low half of a byte). When the MPU reads a byte of data from the NV-RAM, the upper four bits of the value it receives are indeterminate. The NV-RAM is accessible as shown below.

Address	Mode	Function
0250,00xx	R/W	Read/Write RAM contents (4 bits).
0270,0000	Read	Recall RAM contents from Non-volatile memory.
0260,0000	Write	Store RAM contents in Non-volatile memory. The 68020 "tas" (test and set) instruction must be used for this operation.

Table 6-4. Non-Volatile RAM Addresses

Physically, the NV-RAM (a Xicor X2212 or equivalent) consists of a static RAM overlaid bit-for-bit with a non-volatile EEp-ROM. The store operation takes 10 milliseconds to complete. Recall time is approximately one microsecond. Allowances for those delays should be made in *software*, since the memory hardware does not stop the MPU during the store or recall cycles. The chip is rated for 10,000 store cycles, minimum. During a store operation, only those bits which have been changed are "cycled". The use of a "tas" instruction helps prevent an unintentional store operation by an errant program or a power failure glitch.

At power-up, the shadow RAM contents are indeterminate. Do a recall operation before accessing the NV-RAM for the first time. Recall cycles do not affect the device lifetime.

The HK68/V2E monitor (Hbug) and certain system programs use the NV-RAM. The exact amount reserved for Heurikon usage depends on the system. A major portion of the RAM, however, is available for customer use. Heurikon usage is summarized below (details are available separately):

Function
Magic Number
Checksum
Accumulated number of writes
Board type, serial number and revision level
Hardware configuration information
Software configuration information
System configuration information

Table 6-5. NV-RAM Contents (partial)

VMEBUS CONTROL

7.1 INTRODUCTION

The control logic for the VMEbus allows numerous bus masters to share the resources on the bus. Up to 21 boards may be used on the VMEbus.

The VMEbus interface uses 32 Address lines for a total of 4 gigabytes of VMEbus address space, as well as 32 data lines to support 8, 16, 24 or 32-bit data transfers. The "short address" mode, using only 16 address lines, is also supported. In addition, the VSB Expansion Interface is supported, which allows high speed, 32-bit data transfers.

There is an INTERRUPTER MODULE as well as an INTERRUPT HANDLER, both of which are capable of utilizing any or all of the seven VMEbus interrupt lines.

7.2 BUS CONTROL SIGNALS

7.2.1 VMEBUS, P1 DESCRIPTIONS

The following signals on connector P1 and P2 are used for the VMEbus interface. For a complete listing of the pinouts, refer to section 14.

A01-A15 ADDRESS bus (bits 1-15). Three-state driven address lines that are used to broadcast a short address.

A16-A23 ADDRESS bus (bits 16-23). Three-state driven address lines that are used in conjunction with A01-A15 to broadcast a standard address.

A24-A31 ADDRESS bus (bits 24-31). Three-state driven address lines that are used in conjunction with A01-A23 to broadcast an extended address.

ACFAIL* AC FAILURE. An open-collector driven signal which indicates that the AC input to the power supply is no longer being provided or that the required AC input voltage levels are not being met. This signal is connected to MPU interrupt level 7.

AM0-AM5 ADDRESS MODIFIER (bits 0-5). Three-state driven lines that are used to broadcast information such as address size and cycle type. These lines are very similar in usage to the function lines on the MPU.

AS* ADDRESS STROBE. A three-state driven signal that indicates when a valid address has been placed on the address bus.

BBSY* BUS BUSY. An open-collector driven signal low by the current MASTER to indicate that it is using the bus. When the MASTER releases this line, the resultant rising edge causes the ARBITER to sample the bus request lines and grant the bus to the highest priority requester. Early release mode is supported.

BCLR* BUS CLEAR. A totem-pole driven signal, generated by an ARBITER to indicate when there is a higher priority request for the bus. This signal requests the current MASTER to release the bus. This signal is an input and an output of the HK68/V2E, associated with J28.

BERR* BUS ERROR. An open-collector driven signal generated by a SLAVE or BUS TIMER. This signal indicates to the MASTER that the data transfer was not completed.

BG0IN*-BG3IN* BUS GRANT (0-3) IN. Totem-pole driven signals generated by the ARBITER and REQUESTERS. "Bus grant in" and "bus grant out" signals form bus grant daisy chains. The "bus grant in" signal indicates, to the board receiving it, that it may use the bus if it wishes to.

BG0OUT*-BG3OUT* BUS GRANT (0-3) OUT. Totem-pole driven signals generated by REQUESTERS. The bus grant out signal indicates to the next board in the daisy-chain that it may use the bus.

BR0*-BR3* BUS REQUEST (0-3). Open-collector driven signals generated by REQUESTERS. A low level on one of these lines indicates that some MASTER needs to use the bus.

D00-D31 DATA BUS. Three-state driven bidirectional data lines used to transfer data between MASTERS and SLAVES.

DS0*, DS1* DATA STROBE ZERO, ONE. A three-state driven signal used in conjunction with LWORD* and A01 to indicate how many data bytes are being transferred (1, 2, 3, or 4). During a write cycle, the falling edge of the first data strobe indicates that valid data is available on the data bus.

DTACK* DATA TRANSFER ACKNOWLEDGE. An open-collector driven signal generated by a SLAVE. The falling edge of this signal indicates that valid data is available on the data bus during a read cycle, or that data has been accepted from the data bus during a write cycle. The rising edge indicates when the SLAVE has released the data bus at the end of a READ CYCLE.

IACK* INTERRUPT ACKNOWLEDGE. An open-collector or three-state driven signal used by an INTERRUPT HANDLER acknowledging an interrupt request. It is routed, via a backplane signal trace, to the IACKIN* pin of slot one, where it forms the beginning of the IACKIN*, IACKOUT* daisy-chain.

IACKIN* INTERRUPT ACKNOWLEDGE IN. A totem-pole driven signal. The IACKIN* signal indicates to the VMEbus board receiving it that it is allowed to respond to the INTERRUPT ACKNOWLEDGE CYCLE that is in progress if it so wishes.

IACKOUT* INTERRUPT ACKNOWLEDGE OUT. A totem-pole driven signal. The IACKIN* and IACKOUT* signals form a daisy-chain. The IACKOUT* signal is sent by a board to indicate to the next board in the daisy-chain that it is allowed to respond to the INTERRUPT ACKNOWLEDGE CYCLE that is in progress.

IRQ1*-IRQ7* INTERRUPT REQUEST (1-7). Open-collector driven signals, generated by an INTERRUPTER, which carry interrupt requests. When several lines are monitored by a single INTERRUPT HANDLER the highest numbered line is given the highest priority.

LWORD* LONGWORD. A three-state driven signal used in conjunction with DS0*, DS1*, and A01 to select which byte location(s) within the 4 byte group are accessed during the data transfer.

RESERVED RESERVED. A signal line reserved for future VMEbus enhancements. This line must not be used.

SERCLK SERIAL CLOCK. A totem-pole driven signal which is used to synchronize the data transmission on the VMEbus. Not implemented on the HK68/V2E.

SERDAT* SERIAL DATA. An open-collector driven signal which is used for VMEbus data transmission. Not implemented on the HK68/V2E.

SYSCLK SYSTEM CLOCK. A totem-pole driven signal which provides a constant 16-MHz clock signal that is independent of any other bus timing. This signal is associated with J19.

SYSFAIL* SYSTEM FAIL. An open-collector driven signal that indicates that a failure has occurred in the system. Also used at power-on to indicate that at least one VMEbus board is still in its power-on initialization phase. This signal may be generated by any board on the VMEbus. The HK68/V2E drives this line low at power-on. It is released by writing a one to address 02B0,000E.

SYSRESET* SYSTEM RESET. An open-collector driven signal which, when low, causes the system to be reset. This signal is associated with jumper J29.

WRITE* WRITE. A three-state driven signal generated by the MASTER to indicate whether the data transfer cycle is a read or a write. A high level indicates a read operation; a low level indicates a write operation.

+5V STDBY +5 Vdc STANDBY. This line supplies +5 Vdc to devices requiring battery backup. Not used on the HK68/V2E.

7.2.2 VSB, P2 DESCRIPTIONS - VME SUBSYSTEM BUS

The following signals on connector P2 are used for the VSB interface. For a complete listing of the P2 pinouts, refer to section 14. See section 7 for more discussion about the VSB.

AD00-31 MULTIPLEXED ADDRESSED/DATA LINES. This multiplexed address/data path (32 lines) is controlled by the three-state drivers on the master and slave devices. All lines are active high signals.

PAS* VSB ADDRESS STROBE. The falling edge of PAS* indicates that a valid address is present on AD31-AD00.

SPACE0-SPACE1 VSB ADDRESS SPACE SELECT. These signals select one of four address spaces or signify an interrupt acknowledge or parallel arbitration cycle. On the HK68/V2E, these signals are not used; they are driven high when the HK68/V2E is the VSB master, which selects the System Address Space.

DS* VSB DATA STROBE. The falling edge of DS* indicates that transfer will occur over AD31-AD00. During write cycles, write data is valid at the falling edge of DS*.

WR* VSB WRITE. WR*, when low, indicates that a write operation is to be performed and, when high, indicates that a read operation will occur.

SIZE0,SIZE1 VSB BUS SIZE. These lines, in conjunction with addresses AD00 and AD01, determine the data transfer size and position on the data bus. Lines SIZE0 and SIZE1 are active high signals.

LOCK* VSB BUS LOCK. LOCK*, when low, indicates that the bus is locked and that no other master can obtain possession of the bus. This allows for non-interruptable cycles, such as Read-Modify-Write cycles, to occur from the VSB to a dual ported resource. LOCK* can also indicate that a block transfer cycle is in progress.

ASACK0*
ASACK1*

VSB ADDRESS/SIZE ACKNOWLEDGE. The slave device that is selected by address decoding drives at least one ASACK* signal to control switching the multiplexed address/data bus from address to data. ASACK0* and ASACK1* are encoded to indicate to the master the size of the data bus for the slave module to allow dynamic bus sizing.

WAIT*

WAIT* is gated with AC (Decode Complete) on the master device. The condition AC active and WAIT* inactive, while PAS* is asserted, means that no VSB slave module has decoded the address being driven at that time or that there are no VSB slave modules installed. This gives the VSB master the option to switch to the VMEbus when VSB slaves are not responding which allows VSB and VMEbus to share a common address space.

AC

VSB DECODE COMPLETE. AC is asserted by slave modules to indicate to the master that address decoding has been completed. A slave device allows AC to go high after completing decoding or other conditions (see WAIT*), regardless of whether the device is selected by the current address on the bus. AC is an active high signal.

CACHE*

VSB CACHEABLE. CACHE*, when low, indicates to the master that the selected address location is cacheable. CACHE* is asserted only by the selected VSB slave module. This signal is not used on the HK68/V2E.

ACK*

VSB DATA TRANSFER ACKNOWLEDGE. ACK* is asserted low by the selected slave module to complete the handshake for a transfer operation.

ERR*

VSB DATA ERROR. ERR* is asserted low by the selected slave device to indicate a fault condition while attempting the data transfer operation. This would typically be the result of a parity error detected on a slave device.

IRQ*

VSB INTERRUPT REQUEST. IRQ* when low indicates that a master or slave device is attempting to interrupt another master. On the HK68/V2E, this signal will generate a level 1 autovectored MPU interrupt.

BREQ* VSB BUS REQUEST. BREQ* is asserted low by a requester whenever bus mastership is required.

BGIN* VSB BUS GRANT IN. BGIN* is an input to a requester that, when low, indicates to the requester that it has been granted the bus.

BGOUT* VSB BUS GRANT OUT. BGOUT* is asserted low by either an arbiter or a requester to grant the bus to a requester via the BGIN* signal.

BUSY* VSB BUS BUSY. BUSY* is asserted low by a requester that has been granted the bus to indicate ownership of the bus.

GA0-GA2 VSB GEOGRAPHICAL ADDRESSES. These lines are connected to ground on the HK68/V2E; the geographical addressing feature is not implemented.

7.3 *Bus Arbitration and Release*

When the MPU makes a request for VMEbus facilities, the arbitration logic takes over. If necessary, the requesting board enters a wait state until the bus is available (but only for the maximum time allowed by the Watchdog timer).

Normally, the VMEbus System Controller card provides the system bus clock and access timer, and participates in the arbitration logic. However, a separate System Controller card is *not* needed. The HK68/V2E includes a bus timer and four level (prioritized) VME bus arbiter logic, enabled via jumpers. The following table details the System Controller functions provided by the HK68/V2E.

Function	Setting
System Clock (SYSCLK*)	J19 (install)
System Reset (SYSRESET*)	J29-B (output)
Bus Clear (BCLR*)	J28 (install)

Table 7-1. System Controller Functions

When the HK68/V2E is acting as a System Controller, it should be in VME slot 1.

There are four separate bus request lines on the VMEbus. Each bus request line has an associated bus grant daisy chain.

The following steps *must* be used to configure the HK68/V2E, whether or not the HK68/V2E is the system controller. Failure to follow these instructions could result in incorrect board operation.

1. Decide which level the board will use to request the VMEbus.

2. Set the Bus Request jumper, J18, to the chosen level according to the jumper diagram at the end of this manual.

3. Decide if the HK68/V2E will be the system controller on the VMEbus.

4. Install J24 through J27 corresponding to the configuration chosen above. Select the appropriate setting from the eight legal settings shown for those jumpers in the jumper diagram at the end of this manual.

Refer to section 15.3 for help in locating the jumpers.

If the HK68/V2E is the bus master, when the requested bus operation is completed, the bus will be released according to the state of two control bus control signals, BC1 and BC0. These signals are under software control.

BC1	BC0	Bus Release Status
0	1	(Release-When-Done) Release bus after every operation.
0	0	(Release-On-Request) Release the bus if any other board has a request for the bus (or if BCLR is true).
1	0	(Release-On-Priority) Release the bus only if BCLR is true. This means release only if a higher priority request is pending.
1	1	(No-Release) Never release the bus, once acquired. This state can be used to capture the bus.

Table 7-2. Bus Control Bits

The bus control bits are set (or reset) by writing to the appropriate bits of the Bus Control Latch, described below.

7.4 ACCESSES FROM THE VMEBUS (SLAVE MODE)

The slave address logic is enabled or disabled by writing the appropriate value to the Slave Mode control bit, as follows:

Address	Function (write-only)
02B0,000C	Slave Mode Enable D0 = 0, Slave Disable D0 = 1, Slave Enable

Table 7-3. Slave Mode Control

When the most significant VMEbus address lines match the "Slave Compare Address" and the address modifier matches the "Slave Address Modifier" code, as set in the Bus Control Latch, a slave access is recognized. The most significant address lines (A24-A31) are tested only if the selected address modifier is "Extended". The base address of the window into on-card RAM is also set by bits in the Bus Control Latch. The size of the window is specified by J20 through J23 as follows: (Refer to section 15.3 for jumper locations)

Slave Window Size	J20	J21	J22	J23	Address Compare	Replacement Address
1 megabyte	B	B	B	B	A20-A31	A20-A23
2 megabytes	A	B	B	B	A21-A31	A21-A23
4 megabyte	A	A	B	B	A22-A31	A22-A23
8 megabyte	A	A	A	B	A23-A31	A23 only
16 megabyte	A	A	A	A	A24-A31	none

Table 7-4. Slave Mode Control

A 24-bit latch is used to specify various parameters concerning the operation of the VMEbus. This is a write-only register. The default state at power-up is all zeros.

The latch is composed of three, 8-bit shift registers which are set as follows:

1. Disable the VME slave logic by writing a zero to address 02B0,000C.

2. Write a 32-bit long word to the Bus Control Latch at address 02A0,0000. This is done by performing eight consecutive writes to the Bus Control Latch. The data are automatically

shifted into the shift registers. (See the code fragment, below.)

3. Enable the VME slave logic by writing a one to address 02B0,000C.

ATTENTION HBUG USERS: Rather than using the 'sl' or 'fl' Hbug commands, V2E users should use the command

efl value

where value represents the number which you wish to write to the Bus Control Latch.

```
#define BUS_LATCH     (unsigned long *)0x02A00000
#define SLAVE_ENABLE (unsigned long *)0x02B0000C

WrBusLatch(value)
unsigned long value;
{
    int i;
    *SLAVE_ENABLE = 0;  /* disable slave interface
*/
    for (i=0; i<8; i++ ) {
        *BUS_LATCH = (value >> i); /* shift in D16,
D8 and D0 */
    }
    *SLAVE_ENABLE = 1;  /* enable slave interface */
}
```

Fig. 7-1. Bus Control Latch Loading Routine

Bit	Function
D31-D24	(not used)
D23	(reserved)
D22	(reserved)
D21	Slave Address Modifier 2
D20	Slave Address Modifier 1
D19	Slave Address Modifier 0
D18	VME Slave Release Without Hold
D17	Bus Control BC 1
D16	Bus Control BC 0
D15	Replacement Address 23
D14	Replacement Address 22
D13	Replacement Address 21
D12	Replacement Address 20
D11	Slave Compare Address 23
D10	Slave Compare Address 22
D9	Slave Compare Address 21
D8	Slave Compare Address 20
D7	Slave Compare Address 31
D6	Slave Compare Address 30
D5	Slave Compare Address 29
D4	Slave Compare Address 28
D3	Slave Compare Address 27
D2	Slave Compare Address 26
D1	Slave Compare Address 25
D0	Slave Compare Address 24

Table 7-5. Bus Control Latch (VME slave logic)

The Slave Address Modifier is selected by three SAM bits in the Bus Control Latch according to the following chart:

SAM2	SAM1	SAM0	Slave Address Space
0	0	0	No slave access allowed (disable)
0	0	1	Standard Supervisor Data
0	1	0	Standard Data
0	1	1	Standard (all)
1	0	0	No slave access allowed
1	0	1	Extended Supervisor and Data
1	1	0	Extended Data
1	1	1	Extended (all)

Table 7-6. Slave Address Modifiers

Once a valid bus request has been detected, an on-card bus request is generated to the MPU. When the current MPU cycle is completed, the MPU will release the on-card bus. The VMEbus address and data are then gated on. The bus address lines are utilized as follows:

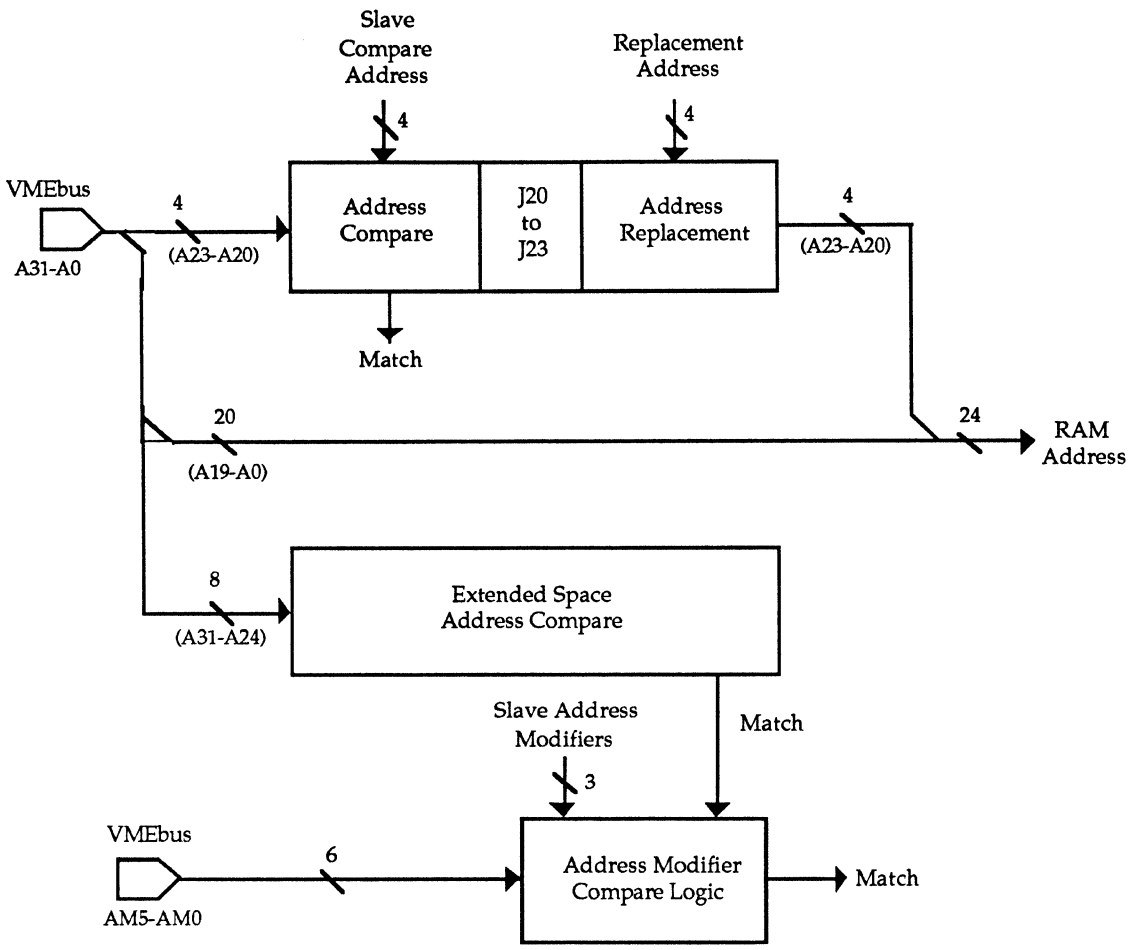


Fig. 7-2. Memory Accesses from the VMEbus

For example, if the Bus Control Latch is set to 0x383050 and J20-J23 are set to select a one megabyte window, then all extended space accesses from 5000,000 through 500F,FFFF are mapped to the fourth megabyte of on-card RAM at location 0330,0000.

After a slave access, control of the on-card bus will not be returned to the MPU for approximately 500 nanoseconds. However, if the Release Without Hold bit in the Bus Control Latch (see above) is set, the bus will be returned immediately following the slave access. This mode can be used to maximize bus response time to the MPU and DMAC at the expense of having more overhead on slave accesses. If you expect rapid requests from the VMEbus, you may not want to use this mode.

The Bus Timer will automatically terminate any slave access which lasts longer than 100 microseconds.

7.5 VME Bus Interrupts

The seven VMEbus interrupts are monitored and controlled by the MPU and CIO. A vectored interrupt to the MPU can be generated when a desired bus interrupt signal is on.

There are two functions described below. The *Interrupter* generates bus interrupts; the *Interrupt Handler* receives interrupts from the bus.

7.5.1 Interrupter Module Operation

To *generate* a VMEbus interrupt, follow these steps:

1. Decide which of the seven VMEbus interrupt lines you wish to activate. IRQ7* has the highest priority.

2. Disable that level via the CIO so that the INTERRUPT HANDLER does not respond to the interrupt line you are about to use. If you fail to do this, you could interrupt yourself.

3. Write an eight bit value to the appropriate VME Status/ID latch, as described below. This value is usually treated as a simple interrupt vector, but it could represent other information as well. This value is provided to the board that acknowledges the interrupt, which is done by executing an INTERRUPT ACKNOWLEDGE cycle on the VMEbus with *your* priority level encoded on address lines 1 to 3 (see the Interrupt Handler description, below.)

   The very act of writing to the Status/ID latch activates the INTERRUPTER circuitry, and the interrupt is generated.

Address	Vector Size	Function (write-only)
0290,0004	8	Interrupt level 1
0290,0008	8	Interrupt level 2
0290,000C	8	Interrupt level 3
0290,0010	8	Interrupt level 4
0290,0014	8	Interrupt level 5
0290,0018	8	Interrupt level 6
0290,001C	8	Interrupt level 7

Table 7-7. VMEbus Interrupter Addresses

Only one (outgoing) interrupt may be pending at a time.

The state of the on-card interrupt logic can be tested by the CIO. The Interrupt Active bit will be true if an interrupt in still pending from this board.

7.5.2 *Interrupt Handler Operation*

Each bus interrupt generates an interrupt to the MPU at a specific MPU interrupt priority level, as detailed in section 3.2. When an interrupt is recognized, the MPU will execute an interrupt acknowledge cycle on the VMEbus to read the vector from the interrupting board. This vector is used as an index into the MPU vector table.

When an interrupt is generated on the VMEbus, the interrupt vector of the interrupting board may be (manually) determined by reading from the appropriate address, as shown below. The value returned is that value written by the interrupting board to its VMEbus Status/ID latch. Since the MPU automatically does interrupt acknowledge cycles on the bus, the main use for these ports is to clear a pending interrupt on the HK68/V2E (or another VMEbus interrupt source).

The HK68/V2E can generate and read only 8-bit interrupt vectors.

Priority Level	8-bit Vector Address (read-only)
IRQ1	0080,0003
IRQ2	0080,0005
IRQ3	0080,0007
IRQ4	0080,0009
IRQ5	0080,000B
IRQ6	0080,000D
IRQ7	0080,000F

Table 7-8. Interrupt Acknowledge Port Summary

Accessing one of the above addresses also sends an interrupt acknowledge signal to the interrupting board. Acknowledging a non-existent interrupt will result in a bus error.

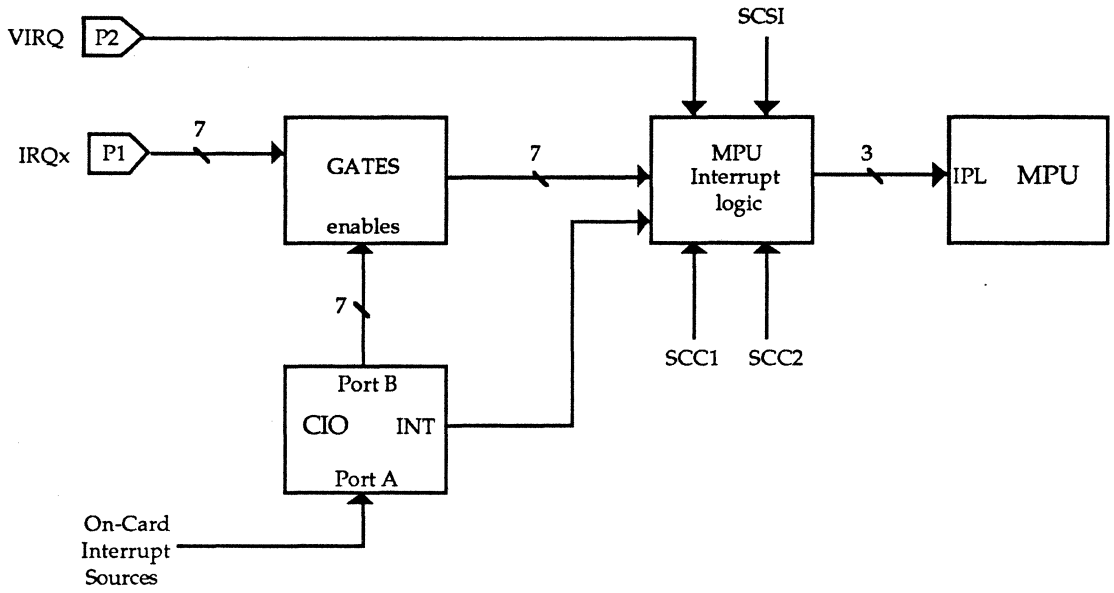


Fig. 7-3. Interrupt Signal Routing

7.6 SYSFAIL CONTROL

The SYSFAIL line is driven low by the HK68/V2E after power-on. The SYSFAIL line will remain low on the VMEbus until all boards release this line after completing their initialization and self test sequences. The SYSFAIL line also signifies a system failure. The current state of this signal may be read via the CIO (see section 9.4).

On the HK68/V2E, SYSFAIL must be released under software control by writing a one to address 02B0,000E.

7.7 BUS ADDRESSING (MASTER MODE)

The HK68/V2E supports three address modes, "short", "standard" and "extended". Short addresses use the lower 16 logical address lines to specify the target address. Standard addresses use 24 address lines, and extended addresses use all 32 address lines. The following table details the relationship between the on-card physical address and the corresponding VMEbus and VSB regions.

On-Card addresses	VMEbus Region
00C0,0000 thru 00FF,FFFF	VMEbus Short Address (0000 thru FFFF)
0100,0000 thru 01FF,FFFF	VMEbus Standard (00,0000 thru FF,FFFF)
0400,0000 thru 3FFF,FFFF	VSB Space, or VME Extended Space (0400,0000 thru 3FFF,FFFF)
4000,0000 thru FFFF,FFFF	VMEbus Extended (4000,0000 and up)

Table 7-9. VMEbus Regions

Extended VME addresses and VSB addresses from 0000,0000 thru 03FF,FFFF are not accessible. The region between 0400,0000 and 3FFF,FFFF is either VSB or VMEbus, depending on whether or not VSB memory exists. That is, if the VSB does not respond to an address in that region, the VMEbus will be used, instead. For this mechanism to work, the VSB ARBE bit (at address 02B0,000A) on the primary master HK68/V2E must be set true, *even if the VSB is not used.* The bus access hardware operates as follows between 0400,0000 and 3FFF,FFFF:

1. If this board is the primary VSB master and if the VSB ARBE bit is false, generate a bus error.
2. Else, arbitrate for the VSB and run a bus cycle.
3. If a VSB device responds (AC and WAIT), do a VSB cycle.
4. Else, if no VSB device responds (AC and no WAIT), restart the access on the VMEbus.
5. Arbitrate for the VMEbus.
6. If a VMEbus device responds (DTACK), complete the bus cycle.
7. Else, if no response on the VMEbus (DTACK), generate a bus error. This could be due to there being no device at the address or there being no bus arbitration (incorrect setting of the bus arbitration jumpers or a system controller malfunction).

7.8 MAILBOX INTERFACE

Certain on-card functions can be controlled via special addresses in the VMEbus *Supervisor Short Address Space,* that is, when the address modifier lines (AM5* to AM0*) are 0x2D. The HK68/V2E will respond (as a slave) to a short address which matches the Mailbox select lines, as described below. The mail-

box logic must be enabled by setting the control bit at address 02B0,0004.

Address	Function (write-only)
02B0,0004	Mailbox Control D0 = 0, Disable (default) D0 = 1, Enable

Table 7-10. Mailbox Control

Address	Function (Slave Mode)
Mbase + 0	CIO Input D4 (see section 9.2) (Mailbox Interrupt)
Mbase + 2	HK68/V2E Reset
Mbase + 4	VMEbus Lock On
Mbase + 5	VMEbus Lock Off
Mbase + 6	MPU Halt On
Mbase + 7	MPU Halt Off

Table 7-11. Mailbox Functions

The "Mbase" value is specified by 13 Mailbox Base bits in the Mailbox Address Latch at address 02C0,0000 (16-bits, write-only). Address lines A15 through A3 must match the corresponding data bits in the Mailbox Address Latch. The lower three bits of the latch are not used.

The Lock function, when ON, will deny the use of the on-card bus by the MPU after the *next* access from the bus. The Lock function must be cleared before the MPU will be allowed to resume operation. This feature can be used to reduce arbitration time during a block data transfer from the VMEbus. With the on-card bus locked, slave accesses will be acknowledged in 330 to 500 nanoseconds, depending on the memory address (whether or not there is a page fault).

The SYSFAIL signal must be off for the mailbox Halt function to operate. (See section 7.6.)

7.9 VSB INTERFACE

The VSB is a local bus extension designed for high speed access to memory or other facilities without the need to use the VMEbus. The HK68/V2E operates on the VSB in master or secondary modes

only; it cannot operate as a slave. It has the required arbitration logic to handle multiple VSB masters. The VSB is a super-set of the VMX32bus; VMX32bus slaves may be used.

Physically, the bus interface uses 32 multiplexed address and data lines. Data transfers may be 8, 16, 24 or 32 bits in length. It is an asynchronous bus.

There is one interrupt line, IRQ, associated with the VSB. This line is connected to MPU interrupt level 1, and generates an autovectored interrupt when recognized by the MPU.

The VSB signals must be properly terminated to assure correct bus operation. Use this chart to determine which resistor packs should be installed for your system configuration.

	End Board	Other Boards
Install	all (R18-R24)	none
Remove	none	all (R18-R24)

Table 7-12. VSB Terminations

Summary: Remove resistor packs R18 through R24 on all but *one* end board.

The VSB specification calls for the terminators to be within two inches of one end of the signal lines. If your VSB backplane includes the signal terminations, then the R-Packs should be removed on all of the VSB modules. The R-Packs are located near connector P2. Six or fewer boards may be used on the VSB.

There are two control bits which affect the operation of the VSB interface.

Address	Function (write-only D0)
02B0,0008	Release With Hold (see below)
02B0,000A	VSB Arbiter enable (see below)

Table 7-13. VSB Control

The HK68/V2E supports two VSB release modes. The bus can be released between every access or only if another master requests the bus. A one-bit latch at address 02B0,0008 controls the VSB release mode as follows:

VSB ROR	Function	Wait States (read)	Wait States (write)
0	Release only if another request	5	4
1	Release after every operation	7	6

Table 7-14. VSB Release Modes

The "VSB ARBE" bit *must be set true* on the "first" VSB master board - the the primary master. The secondary master should have this bit set false. The VSB ARBE bit indicates the beginning of the VSB arbitration daisy chain.

7.10 Watchdog and Bus Timer

The HK68/V2E has two timers which monitor board activity. One is used to monitor on-card activity; the other is for the VMEbus.

If the on-card watchdog timer is enabled and if the on-card physical address strobe stays on longer than 1.67 milliseconds, the timer will expire. This will cause the current memory cycle to be terminated. The watchdog timer is *disabled* by writing a one to address 02B0,0030. The timer is *enabled* by writing a zero to address 02B0,0030; this is the power-on default state.

See section 5.1 for more details on the watchdog timer.

The second timer is associated only with activity on the VMEbus. The timer will expire during a long bus access (greater than 100 microseconds) by *any* bus master and generate a VMEbus error (BERR). This is normally a VMEbus system controller function.

7.11 Relevant Jumpers - Bus Control

Jumper	Function	Position
J18	Bus Request Level	See section 7.2
J19	SYSCLK*	See section 7.2
J20	VME Slave Window Size	See section 7.3
J21	VME Slave Window Size	See section 7.3
J22	VME Slave Window Size	See section 7.3
J23	VME Slave Window Size	See section 7.3
J24	Bus Arbitration Level	See section 7.2
J25	Bus Arbitration Level	See section 7.2
J26	Bus Arbitration Level	See section 7.2
J27	Bus Arbitration Level	See section 7.2
J28	BCLR*	See section 7.2
J29	SYSRESET*	See section 7.2

Table 7-15. Bus Control Jumpers

MISCELLANEOUS DEVICES

8.1 USER LEDS

There are four LEDs (located near the reset button) whose meanings may be defined by the program.

LED Number	Address (write-only)
(MSB) 1	02B0,0040
2	02B0,0050
3	02B0,0060
(LSB) 4	02B0,0070

Table 8-1. User LEDs - Addresses

Writing a zero turns the chosen LED on; writing a one will turn it off. At power-on or after a system reset, the LEDs will be ON.

CIO USAGE

9.1 INTRODUCTION

The on-card CIO device performs a variety of functions. In addition to the three 16-bit timers which may be used to generate interrupts or count events, the CIO has numerous parallel I/O bits.

9.2 PORT A BIT DEFINITION

Port A handles various control signals. All bits should be programmed as inputs.

Bit	Function	Polarity	Interface	Reference
D7	External Interrupt	Negative True	P6-11	3.5
D6	EEpROM 1 Ready (U56)	Positive True	U56-1	6.2
D5	EEpROM 0 Ready (U48)	Positive True	U48-1	6.2
D4	Mailbox Interrupt	Negative True	-	7.8
D3	SCSI DMA Request	Negative True	-	11
D2	VME Interrupt in Progress	Negative True	-	7.5
D1	SCSI Reset	Negative True	P4-40	11
D0	(reserved)			

Table 9-1. CIO Port A Bit Definitions

Bit D2 may be used to test if there is a pending interrupt still active from *this* board. The mailbox interrupt is a pulse, so the ones catcher should be used for that input bit.

9.3 PORT B BIT DEFINITION

The B port of the CIO is used to handle the Centronics I/F interrupt (input) and generate the VME interrupt mask bits (outputs).

Bit	Function	Polarity	Interface	Reference
D7	Centronics Interrupt	Positive True	P3	12
D6	IRQ7 enable	Negative True	P1	3.2
D5	IRQ6 enable	Negative True	P1	3.2
D4	IRQ5 enable	Negative True	P1	3.2
D3	IRQ4 enable	Negative True	P1	3.2
D2	IRQ3 enable	Negative True	P1	3.2
D1	IRQ2 enable	Negative True	P1	3.2
D0	IRQ1 enable	Negative True	P1	3.2

Table 9-2. CIO Port B Bit Definitions

Internal priorities of the CIO place D7 as highest (D0 as lowest) for simultaneous interrupts from either port.

9.4 PORT C BIT DEFINITION

Port C on the CIO chip is used to read four on-card status signals.

Bit	Function
D3	VMEbus ACFAIL* (negative true)
D2	VMEbus SYSFAIL* (negative true)
D1	Port C RI (negative true)
D0	Port A RI (negative true)

Table 9-3. CIO Port C Bit Definitions

9.5 COUNTER/TIMERS

There are three independent, 16-bit counter/timers in the CIO. For long delays, timers 1 and 2 may be internally linked together to form a 32-bit counter chain. When programmed as timers, the following equation may be used to determine the time constant value for a particular interrupt rate.

TC = 2,457,600 / interrupt rate (in HZ)

When the timer is clocked internally, the count rate is 2.4576 MHz. The HK68/V2E board uses a 19.6608 MHz clock oscillator as the system time base. The frequency tolerance specification is ±0.01%. If you are using the 19.6608 MHz clock as the CIO time base, the maximum accumulative timing error will be about 9 seconds per

onds per day, although the typical error is less than one second per day. Better long-term accuracy may be achieved via a power line (60 Hz) interrupt, using a bus interrupt or the Real-Time Clock (RTC) option (see section 13).

9.6 Register Address Summary (CIO)

Register	Address	Function
Port C, Data	02D0,0001	Miscellaneous Control Bits
Port B, Data	02D0,0003	Miscellaneous Control Bits
Port A, Data	02D0,0005	Miscellaneous Control Bits
Control Regs	02D0,0007	CIO Configuration & Control

Table 9-4. CIO Register Addresses

All registers are eight bits wide.

9.7 CIO Initialization

The following figure shows a typical initialization sequence for the CIO. The first byte of each data pair in "ciotable" specifies an internal CIO register; the second byte is the control data. The specific directions of some of the PIO lines and interrupts need to be changed in the table, based on your application. An active low signal can be inverted (so that a "1" is read from the data port when the signal is true) by initializing the port to invert that particular bit. Refer to section 3 for information concerning CIO interrupt vectors.

```
char ciotable[] = {
0x00, 0x01, 0x00,/* reset, set chip ptr to reg zero */

/* port A initialization */
0x20, 0x06,  /* bit port, priority encoded vector */
0x22, 0x9c,  /* invert negative true bits */
0x23, 0xff,  /* all bits are inputs */
0x24, 0x10,  /* one's catcher */
0x25, 0x00,  /* pattern polarity register */
0x26, 0x00,  /* all levels (can't use transitions */
/* in "or priority mode") */
0x27, 0x10,  /* pattern mask, enable mailbox interrupt */
0x02, 0x41,  /* set interrupt vector */
0x08, 0xc0,  /* set int enable, no int on err */

/* port B initialization */
0x28, 0x06,  /* bit port, priority encoded vector */
0x2a, 0x00,  /* don't invert input bit */
0x2b, 0x80,  /* one bit is an input */
0x2c, 0x00,  /* normal input (no ones catchers) */
0x2d, 0xff,  /* bit interrupt on a one */
0x2e, 0x00,  /* no transition, levels only */
0x2f, 0x00,  /* no interrupts enabled */
0x03, 0x40,  /* set interrupt vector */
0x09, 0xc0,  /* set int enable, no int on err */

   /* port c initialization */
0x05, 0x0f,  /* invert negative true bits */
0x06, 0x0f,  /* all bits are inputs */
0x07, 0x00,  /* normal inputs */

/* timer 3 and other CIO initialization */
0x1e, 0x80,  /* set mode to auto reload */
0x1a, 0xa0,  /* high byte delay constant */
0x1b, 0x00,  /* low byte delay constant */
0x04, 0x60,  /* interrupt vector */
0x08, 0x20,  /* clear any port A ints */
0x08, 0x20,  /* clear any port A ints */
0x01, 0x94,  /* enable timer 3, port a and port b */
0x0c, 0xc6,  /* set interrupt enable and */
/* gate command bit and trigger cmd bit */
0x00, 0x9e   /* master int enable and vector includes */
/* status for timer 3, port A and port B */
};

struct cdevice {   /* CIO register structure */
    char dummy0;  char cdata;      /* port C */
    char dummy1;  char bdata;      /* port B */
    char dummy2;  char adata;      /* port A */
    char dummy3;  char ctrl; /* control port */
};
\#define CIO ((struct cdevice *)0x02d00000)

cioinit()
{
    int i, t3intr();
 /*Don't forget to set CIO interrupt vectors. Example: */
    *(int(*))(0x60*4) = (int)t3intr;/* Timer 3 interrupt */
    i = CIO->ctrl;           /* assure register sync */
    CIO->ctrl = ciotable[0]; /* avoid clr instruction*/
    i = CIO->ctrl;           /* assure register sync */
    for (i = 0; i < sizeof(ciotable); i++)
   CIO->ctrl = ciotable[i];/* send ciotable to CIO chip */
}
```

```
Aintr()     /* clear Port A interrupt */
    /* one of 8 routines */
{   /* process port A interrupts here */
    CIO->ctrl = 0x08; CIO->ctrl = 0x20;
}

Bintr()/* clear Port B interrupt */ /* one of 8 routines */
{   /* process port B interrupts here */
    CIO->ctrl = 0x09; CIO->ctrl = 0x20;
}

timer3() /* clear Timer 3 interrupt, get here via t3intr */
{   /* process timer interrupt here */
    CIO->ctrl = 0x0c; CIO->ctrl = 0x24;
}
```

Fig. 9-1. CIO Program Example (C Portion)

```
            .globl   t3intr%, timer3

      \# the vector at 0x60*4 points to this routine

      t3intr%: movm.l  &0xFFFF,-(%sp)   # save registers
               jsr     timer3           # to C portion
               movm.l  (%sp)+,&0xFFFF   # restore registers
               rte
```

Fig. 9-2. CIO Program Example (Assembly Code Portion)

9.8 CIO PROGRAMMING HINTS

1. To maintain compatibility with 68010 programs, do not use the 68020 "clr.b" instruction to set a CIO register to zero. On the 68000 and 68010, that instruction does a "phantom" read of the port before it does the zero write. The read operation will upset the CIO internal register selection sequencer. Similarly, when using a high level language, do not set a CIO register value to the constant "0" because the compiler may use a "clr.b". Use a variable which is set to zero, or output the values from a lookup table. For example:

```
zero = 0;
*CIOcntrl = 0x20;
*CIOcntrl = zero;
```

2. The ones catchers in a CIO port will be cleared whenever any bits are changed in the pattern mask register. Avoid changing the mask register if you are using a ones catcher. If this is not possible, a program that writes to the pattern mask register should first OR the CIO data register into a memory

variable. Later, that memory value can be ORed with the CIO data register to find out what the data register would have been if the CIO had not cleared it. Routines which respond to a ones catcher interrupt must clear the corresponding bits in the memory value and the CIO data register. There will still be a critical period where a fast input pulse could be missed, even when using this scheme.

3. If you get an unexpected interrupt from bit D0 of a CIO port, it may be because another enabled CIO input signal went false before the MPU initiated the interrupt acknowledge cycle. The use of a ones catcher may be appropriate to latch the input line.

4. If you turn on a bit in the pattern mask register, that bit will generate an interrupt (if the port is enabled) even if the input signal is false. To prevent this, disable the port while adjusting the pattern mask register.

5. The CIO may glitch the parallel port lines when a hardware reset is done, even if all lines are programmed as inputs. This may cause a problem in multi-processor systems because the glitches may produce spurious ACFAIL and SYSFAIL signals on other (operating) boards. To prevent this effect, disable the port (via software) prior to doing a board reset.

Refer to the Z8536 technical manual for more details on programming the CIO. Some people find the CIO technical manual difficult to understand. We encourage you to read all of it twice, before you pass judgement. Section 3.2 has a list of suggested readings from the CIO manual. Contact us (or Zilog) to obtain application notes.

SERIAL I/O

10.1 INTRODUCTION

There are four RS-232C serial I/O ports on the HK68/V2E board. Each port may optionally be configured for RS-422 operation with a special interface cable, as detailed in section 10.8. Each port has a separate baud rate generator and can operate in asynchronous or synchronous modes.

10.2 RS-232 PINOUTS

Data transmission conventions are with respect to the external serial device. The HK68/V2E board is wired as a "Data Set." The connector pinouts are as follows:

Pin	"D" Pin	RS-232 Function	Direction	SCC Pin Function
P5-1	2	Port A Tx Data	In	Rcv Data
P5-2	15	Tx Clock	In	
P5-3	3	Rcv Data	Out	Tx Data
P5-4	16	(not used)		
P5-5	4	Request to Send *	In	DCD
P5-6	17	Rcv Clock	In	
P5-7	5	Clear to Send	Out	DTR
P5-8	18	Ring Detect	In	Ring Ind
P5-9	6	Data Set Ready	Out	RTS
P5-10	19	(not used)		
P5-11	7	Gnd		Sig Gnd
P5-12	20	Data Terminal Ready*	In	CTS

Table 10-1a. Serial Port Pinouts (P5) - Port A

Pin	"D" Pin	RS-232 Function	Direction	SCC Pin Function
P5-13	2	Port B Tx Data	In	Rcv Data
P5-14	15	Tx Clock	In	
P5-15	3	Rcv Data	Out	Tx Data
P5-16	16	+12v (via J3)		
P5-17	4	Request to Send *	In	DCD
P5-18	17	Rcv Clock	Out	
P5-19	5	Clear to Send	Out	DTR
P5-20	18	+5v (via J4)		
P5-21	6	Data Set Ready	Out	RTS
P5-22	19	-12v (via J5)		
P5-23	7	Gnd		Sig Gnd
P5-24	20	Data Terminal Ready*	In	CTS

Table 10-1b. Serial Port Pinouts (P5) - Port B

Pin	"D" Pin	RS-232 Function	Direction	SCC Pin Function
P5-25	2	Port C Tx Data	In	Rcv Data
P5-26	15	Tx Clock	In	
P5-27	3	Rcv Data	Out	Tx Data
P5-28	16	(not used)		
P5-29	4	Request to Send *	In	DCD
P5-30	17	Rcv Clock	In	
P5-31	5	Clear to Send	Out	DTR
P5-32	18	Ring Detect	In	Ring Ind
P5-33	6	Data Set Ready	Out	RTS
P5-34	19	(not used)		
P5-35	7	Gnd		Sig Gnd
P5-36	20	Data Terminal Ready*	In	CTS

Table 10-1c. Serial Port Pinouts (P5) - Port C

Pin	"D" Pin	RS-232 Function	Direction	SCC Pin Function
P5-37	2	Port D Tx Data	In	Rcv Data
P5-38	15	Tx Clock	In	
P5-39	3	Rcv Data	Out	Tx Data
P5-40	16	+12v (via J6)		
P5-41	4	Request to Send *	In	DCD
P5-42	17	Rcv Clock	Out	
P5-43	5	Clear to Send	Out	DTR
P5-44	18	+5v (via J7)		
P5-45	6	Data Set Ready	Out	RTS
P5-46	19	-12v (via J8)		
P5-47	7	Gnd		Sig Gnd
P5-48	20	Data Terminal Ready*	In	CTS
P5-49		(not used)		
P5-50		(not used)		

Table 10-1d. Serial Port Pinouts (P5) - Port D

Note that the interconnect cable from P5 is arranged in such a manner that the "D" connector pinouts are correct for RS-232C conventions. Not all pins on the "D" connectors are used. Recommended mating connectors are Ansley P/N 609-5001CE and Molex P/N 15-29-8508.

Signals indicated with "*" have default pullup resistors, controlled by J1 (ports A and B) and J2 (ports C and D). NOTE: The serial ports may .I appear to be inoperative if J1 (or J2) is set to default "FALSE" and if the device connected to the port does not drive the DTR and RTS pins TRUE. The Hbug monitor software, for example, initializes the SCC channels to respect the state of DTR and RTS. The RI signals for ports A and C are routed to the CIO. See section 10.9.

10.3 SIGNAL NAMING CONVENTIONS (RS-232)

Since the RS-232 ports are configured as "data sets," the naming convention for the interface signals may be confusing. The interface signal names are with respect to the terminal device attached to the port while the SCC pins are with respect to the SCC as if it, too, is a terminal device. Thus all signal pairs, e.g., "RTS" & "CTS," get switched between the I/F connector and the

example, "Transmit Data," P5-1, is the data transmitted from the device to the HK68/V2E board; the data appears at the SCC receiver as "Received Data." For the same reason, the "DTR" and "RTS" interface signals appear as the "CTS" and "DSR" bits in the SCC, respectively. If you weren't confused before, any normal person should be by now. Study the chart below and see if that helps.

SCC Signal	I/F Signal	Direction
Tx Data	Rcv Data	to device
Rcv Data	Tx Data	from device
Tx Clock	Rcv Clock	from device (ports A & C)
Tx Clock	Rcv Clock	to device (ports B & D)
Rcv Clock	Tx Clock	from device
RTS	DSR	to device
CTS	DTR	from device
DTR	CTS	to device
DCD	RTS	from device
-	Ring Ind.	from device

Table 10-2. Signal Naming Conventions

The SCC was designed to look like a "data terminal" device. Using it as a "data set" creates this nomenclature problem. Of course, if you connect the HK68/V2E board to a modem ("data set"), then the SCC signal names are correct, however, a cable adapter is needed to properly connect to the modem. (Three pairs of signals must be reversed.)

SCC Signal	P5 Pin #s	"D" Pin # at HK68/V2E	"D" Pin # at modem	RS-232 Signal
x	x	1	1	Prot Gnd
Rcv Data	1	2	3	Rcv Data
Tx Data	3	3	2	Tx Data
DCD	5	4	6	DSR
RTS	9	6	4	RTS
DTR	7	5	20	DTR
CTS	12	20	5	CTS
(Ring Ind)	8	18	22	Ring Ind
(Sig Gnd)	11	7	7	Sig Gnd

Table 10-3. RS-232 Cable Reversal

Summary: The HK68/V2E may be directly connected to a data "terminal" device. A cable reversal is required for a connection to a modem.

10.4 CONNECTOR CONVENTIONS

Paragraph 3.1 of the EIA RS-232-C standard says the following concerning the mechanical interface between data communications equipment:

> *"The female connector shall be associated with...the data communications equipment... An extension cable with a male connector shall be provided with the data terminal equipment... When additional functions are provided in a separate unit inserted between the data terminal equipment and the data communications equipment, the female connector...shall be associated with the side of this unit which interfaces with the data terminal equipment while the extension cable with the male connector shall be provided on the side which interfaces with the data communications equipment."*

Substituting "modem" for "data communications equipment" and "terminal" for "data terminal equipment" leaves us with the impression that the modem should have a *female* connector and the terminal should have a *male.*

The Heurikon HK68/V2E microcomputer interface cables are designed with female "D" connectors, because the serial I/O ports are configured as data sets (modems). Terminal manufacturers typically have a female connector also, despite the fact that they are terminals, not modems. Thus, the extension cable used to run between a terminal and the HK68/V2E (or a modem) will have male connectors at both ends.

If you do any work with RS-232 communications, you will end up with many types of cable adapters. Double males, double females, double males and females with reversal, cables with males and females at both ends, you name it! We will be happy to help make special cables to fit your needs.

10.5 SCC INITIALIZATION SEQUENCE

The following table shows a typical initialization sequence for the SCC. This example is for port A. Other ports are pro-

grammed in the same manner, substituting the correct control port address.

Data	Register Adrs.	Function
00	02F0,0003 (write)	Reset SCC register counter
09,C0	" "	Force reset (do for ports A & C only)
04,4C	" "	Async mode, x16 clock, 2 stop bits tx
05,EA	" "	Tx: RTS, Enable, 8 data bits
03,E1	" "	Rcv: Enable, 8 data bits
01,00	" "	No Interrupt, Update status
0B,56	" "	No Xtal, Tx & Rcv clk internal,BR out
0C,baudL	" "	Set Low half of baud rate constant
0D,baudH	" "	Set high half of baud rate constant
0E,03	" "	Null, BR enable

Table 10-4. SCC Initialization Sequence

Note: the notation "09,C0" (etc) means the values 09 (hex) and C0 should be sent to the specified SCC port. The first byte selects the internal SCC register; the second byte is the control data. The above sequence only initializes the ports for standard asynchronous I/O without interrupts. The 'baudL' and 'baudH' values refer to the low and high halves of the baud rate constant which may be determined from the Baud Rate Constants section below.

For information concerning SCC interrupt vectors, refer to section 3. Refer to the Z8530 technical manual for more details on SCC programming. Contact us (or Zilog) to obtain application notes.

To maintain compatibility with 68010 programs, do not use the 68020 "clr.b" instruction to set a SCC register to zero. On the 68000 and 68010, that instruction does a "phantom" read of the port before it does the zero write. The read operation will upset the SCC internal register selection sequencer. Similarly, when using a high level language, do not set a SCC register value to the constant "0" because the compiler may use a "clr.b". Use a variable which is set to zero, or output the values from a lookup table. For example, this is correct:

```
zero = 0;
*SCCcntrl = 0x20;
*SCCcntrl = zero;
```

10.6 PORT ADDRESS SUMMARY

Register	Port A	Port B	Port C	Port D
Control	02F0,0003	02F0,0001	02E0,0003	02E0,0001
Data	02F0,0007	02F0,0005	02E0,0007	02E0,0005

Table 10-5. SCC Register Addresses

All ports are eight bits.

10.7 BAUD RATE CONSTANTS

If the internal SCC baud rate generator logic has been selected, the actual baud rate must be specified during the SCC initialization sequence by loading a 16-bit time constant value into each generator. The following table gives the values to use for some common baud rates. Other rates may be generated by applying the formula given below.

Baud Rate	x1 clock rate	x16 clock rate
110	22,340	1,394
300	8,190	510
1200	2,046	126
2400	1,022	62
4800	510	30
9600	254	14
19,200	126	6
38,400	62	2

Table 10-6. Baud Rate Constants

The time constant values listed above are computed as follows:

TC = 4,915,200/(2 * baud * (factor 1 or 16)) - 2

The x16 mode will obtain better results with asynchronous protocols because the receiver can search for the middle of the start bit. (In fact, the x1 mode will probably produce frequent receiver errors.)

The maximum SCC data speed is one megabit per second, using the x1 clock and synchronous mode. For asynchronous transmission, the maximum practical rate using the x16 clock is 51,200 baud.

10.8 RS-422 Operation

As an option, one or more of the serial ports on the HK68/V2E may be configured for RS-422 operation. To accomplish this, the on-card RS-232 interface chips are removed, and a special cable is used which includes the RS-422 interface logic.

To install the RS-422 option kit, follow these steps:

1. Remove the RS-232 interface chips for the affected ports. These are U41 and U50 (for port A), U42 and U51 (for port B), U43 and U52 (for port C) and U44 and U53 (for port D). U45 should also be removed.

2. Install the special headers in the RS-232 interface positions. The headers are wired as follows:

75188 Outputs (U50 to U53)	75189 Inputs (U41 to U45)
2 to 3	1 to 3
4 and 5 to 6	4 to 6
8 to 9 and 10	8 to 10
11 to 12 and 13	11 to 13

Table 10-7. RS-422 Header Wiring

3. Install jumper shunts J3 through J8. This supplies power to the RS-422 interface chips which are located on the cable.

10.9 RELEVANT JUMPERS (SERIAL I/O)

Jumper	Function	Position
J1	RS-232 A & B Status Default	J1-A (True) J1-B (False)
J2	RS-232 C & D Status Default	J2-A (True) J2-B (False)
J3	+12 power to P7	
J4	+5 power to P7 (ports A & B)	
J5	-12 power to P7	
J6	+12 power to P6	
J7	+5 power to P6 (ports C & D)	
J8	-12 power to P6	

Table 10-8. Relevant Jumpers - Serial I/O

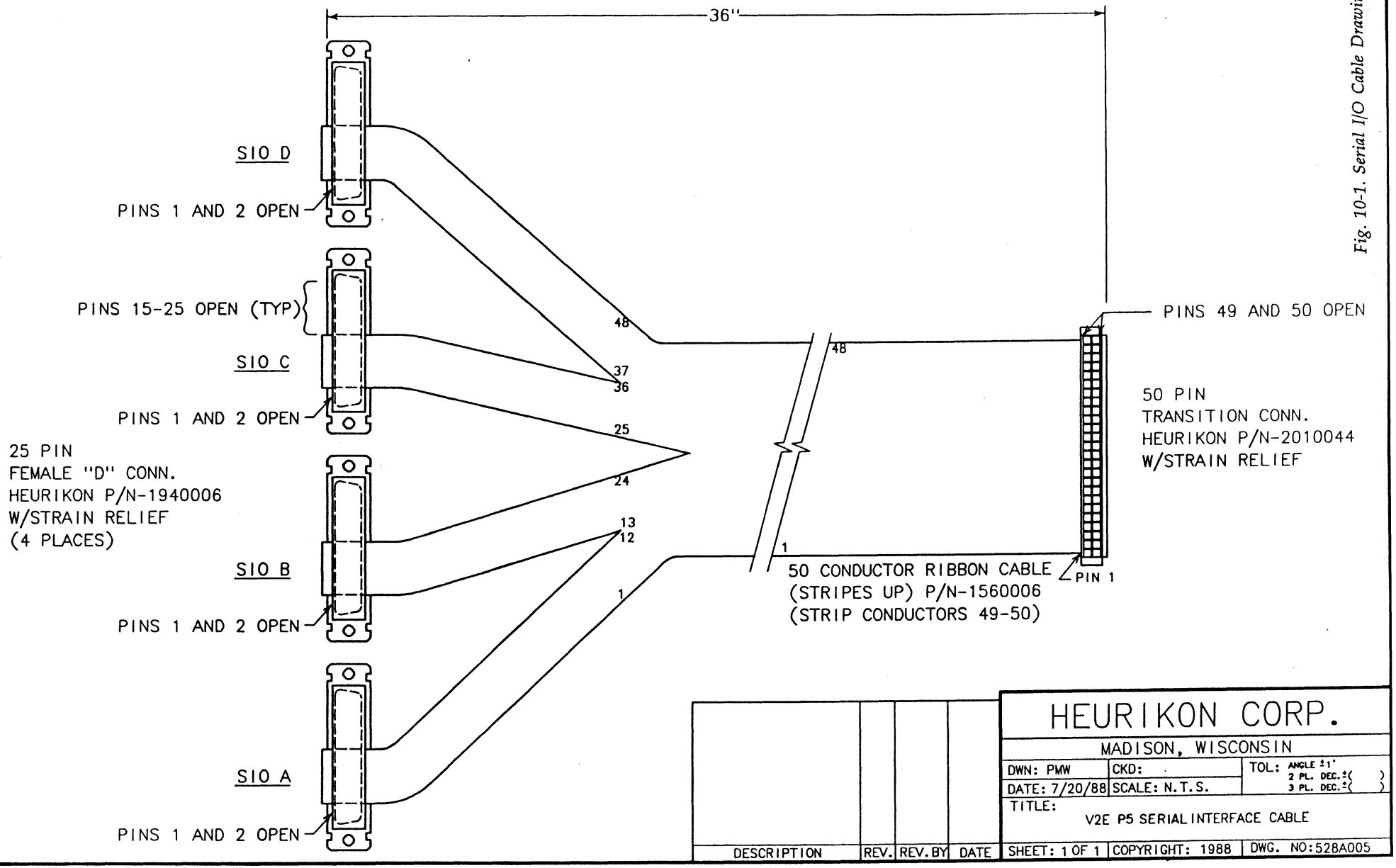


Fig. 10-1. Serial I/O Cable Drawing

Chapter 11

SCSI PORT

11.1 INTRODUCTION

The HK68/V2E uses the Wester Digital WD33C93 chip to implement a Small Computer System Interface (SCSI) port. (Commonly called "Scuzzy".)

The SCSI port may be used to connect to a variety of peripheral devices. Most common are Winchester disks, floppy diskettes, and streamer tape drives.

This port is pseudo-DMA driven, transferring up to 1.5 megabytes per second. Actual rates depend on the characteristics of the attached controller and device being used.

Supported features and modes include:

- Initiator role
- Target role
- Arbitration
- Disconnect
- Reconnect
- Pseudo-DMA interface

11.2 SCSI IMPLEMENTATION NOTES

The SCSI Data Ready signal is routed to the CIO which can cause a MPU interrupt. The interrupt from the SCSI chip generates a level 4 autovector. See MPU Exception Vectors, section 3.3 for details. Data transfer functions can be handled in a polled I/O mode or by using the pseudo-DMA functions provided by on-card logic and the MPU.

The pseudo-DMA allows the SCSI data request signal to either interrupt the MPU (via the CIO) or to synchronize the MPU wait signal (DTACK) to the availability of data. The synchronization logic is enabled by accessing the SCSI data register in a special memory mapped area (at 0240,0000). By so doing, the MPU will enter a wait state until data becomes available. This fea-

ture allows a block transfer of data to or from the SCSI interface to be initiated by an interrupt (when the first byte is ready) and continued through MPU control with minimal overhead.

The MPU will be released from the wait state when either the watchdog expires (indicating that there has been a pause in the data transfer as could happen between sectors) or when the WD33C93 interrupts the MPU (indicating the end of transfer or an error).

11.3 Register Address Summary (SCSI)

Address	R/W	Bits	Function
0230,0001	W	8	Set Controller Address Register
0230,0001	R	8	Read Auxiliary Register
0230,0003	R/W	8	SCSI Controller Registers
0240,0000	R/W	8	SCSI Data Register (pseudo-DMA)
02B0,0006	W	1	SCSI Bus Reset (1=reset, 0=release)
02B0,0020	W	1	SCSI Interrupt Enable (1=enable)

Table 11-1. SCSI Register Address Summary

11.4 SCSI PORT PINOUTS

Pin number	Name	Function
Odd pins		Ground
P4-2	DB0/	Data bit 0
P4-4	DB1/	Data bit 1
P4-6	DB2/	Data bit 2
P4-8	DB3/	Data bit 3
P4-10	DB4/	Data bit 4
P4-12	DB5/	Data bit 5
P4-14	DB6/	Data bit 6
P4-16	DB7/	Data bit 7
P4-18	DBP/	Data parity bit
P4-32	ATN/	Attention
P4-34	Spare	
P4-36	BSY/	SCSI Bus busy
P4-38	ACK/	Transfer acknowledge
P4-40	RST/	Reset
P4-42	MSG/	Message
P4-44	SEL/	Select
P4-46	C/D/	Control/Data
P4-48	REQ/	Transfer request
P4-50	I/O/	Data movement direction

Table 11-2. SCSI Pinouts

Recommended mating connectors are Ansley P/N 609-5001CE and Molex P/N 15-29-8508.

The terminating resistors, RN4, RN5 and RN6 should be installed only if the HK68/V2E is located at an *end* of the SCSI interface cable.

Chapter 12

CENTRONICS PORT

12.1 INTRODUCTION

This 8-bit parallel port is designed for direct connection to a Centronics compatible (printer) device. Since the handshake lines (STROBE and INIT) are under software control, this interface can be used as a general purpose output port.

12.2 CENTRONICS PORT CONFIGURATION

P3 Pin	Centronics Pin	Direction	Signal
P3-17	9	Output	DATA8 (D7)
P3-15	8	Output	DATA7
P3-13	7	Output	DATA6
P3-11	6	Output	DATA5
P3- 9	5	Output	DATA4
P3- 7	4	Output	DATA3
P3- 5	3	Output	DATA2
P3- 3	2	Output	DATA1 (D0)
P3- 1	1	Output	STROBE/
P3-19	10	Input	ACK/
P3-21	11	Input	BUSY
P3-23	12	Input	PE
P3-25	13	Input	SELECT
P3-26	31	Output	INIT/
P3-28	32	Input	ERROR/
P3-30	33	Input	spare 1
P3-32	34	Input	spare 2
P3-34	35	Input	spare 3
P3-27	14		Gnd
P3-31	16		Gnd
P3-(2-24)	(19-30)		Gnd
(even)			
P3-29	15		n/c
P3-33	17		n/c
-	18		n/c
-	36		n/c

Table 12-1. Centronics Pinout (Connector P3)

Recommended mating connectors are Ansley P/N 609-3401CE and Molex P/N 15-29-8348.

The falling edge of ACK/ is used to turn on the Centronics interrupt signal going to CIO bit A4. To clear the interrupt signal, *read* from the interrupt reset location, 0280,0007.

12.3 CONTROL PORT ADDRESSES - CENTRONICS

The Centronics interface logic uses the following physical memory addresses for data and control functions:

Address	Dir	Function
0280,0001	W	Data Latch (see below)
0280,0001	R	Status Port (see below)
0280,0003	W	Turn STROBE on
0280,0003	R	Turn STROBE off
0280,0005	W	Turn INIT on
0280,0005	R	Turn INIT off
0280,0007	R	Reset ACK Interrupt

Table 12-2. Centronics Control Addresses

Bit	0280,0001 (Write) Data Latch	0280,0001 (read) Status Port
D7	DATA8	(spare 1)
D6	DATA7	(spare 2)
D5	DATA6	(spare 3)
D4	DATA5	ERROR/
D3	DATA4	SELECT
D2	DATA3	PE
D1	DATA2	BUSY
D0	DATA1	ACK/ (Negative true pulse)

Table 12-3. Centronics Data/Status Addresses

After power-on, the state of the Data Latch is indeterminate; STROBE and INIT will be false. The Data Latch is not changed by a board reset; however, STROBE and INIT will go false.

Follow this procedure when using this port for a Centronics printer:

1. Wait for the printer BUSY signal to go false.

2. Write the character to port 0280,0001.

3. Assert STROBE (write to 0280,0003).

4. Delay at least one microsecond.

5. De-assert STROBE (read from 0280,0003).

6. Wait for ACK (poll CIO bit A4, or wait for an interrupt via the CIO). The ACK signal at the Centronics status port (bit D0 of 0280,0001) will be just a fleeting pulse.

7. Reset the ACK interrupt signal by reading from 0280,0007.

8. Repeat for the next character.

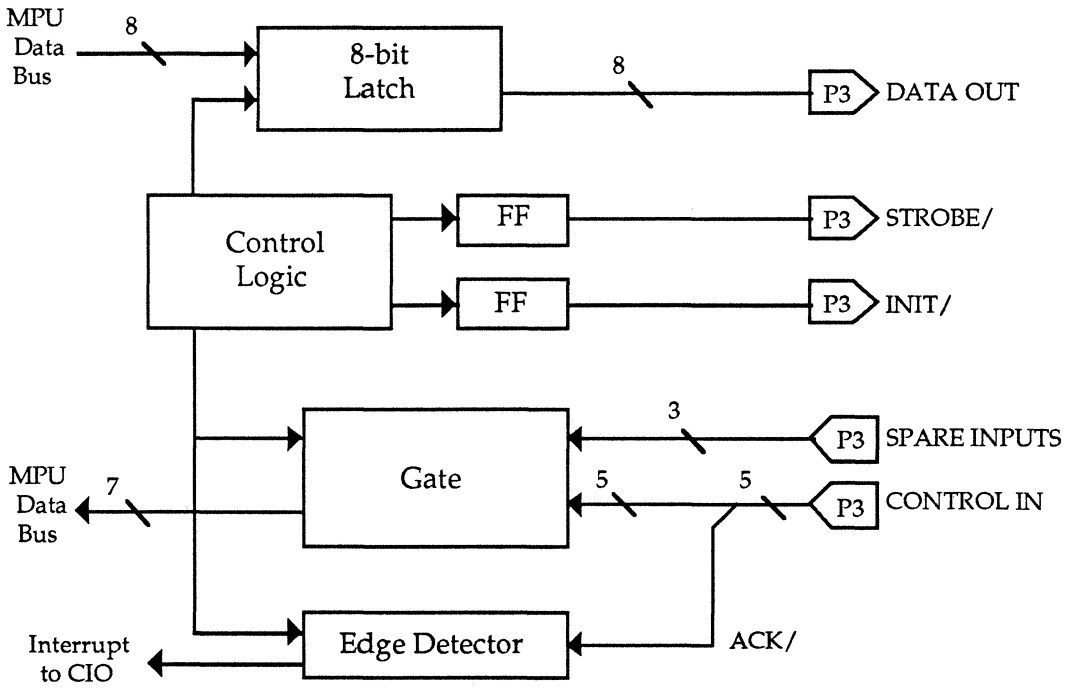
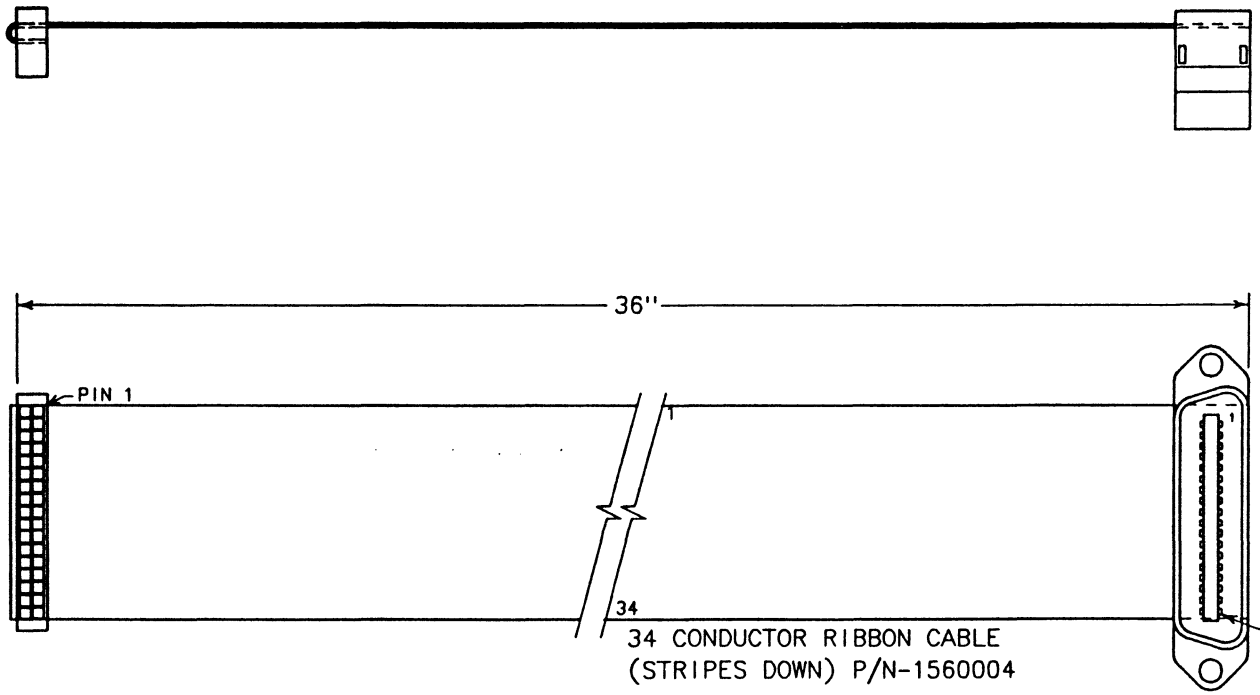


Fig. 12-1. Centronics Interface - Block Diagram



34 PIN
TRANSITION CONN.
ANSLEY P/N-609-3401M
HEURIKON P/N-2010036
W/STRAIN RELIEF

36 PIN
RIBBON CONN.
ANSLEY P/N-609-36M
HEURIKON P/N-1990082

Fig. 12-2. Centronics Printer Interface Cable

DESCRIPTION	REV.	REV. BY	DATE

HEURIKON CORP.

MADISON, WISCONSIN

DWN: RJC	CKD:	TOL: ANGLE ±1° 2 PL. DEC. ±( ) 3 PL. DEC. ±( )
DATE: 7/18/88	SCALE: N.T.S.	

TITLE: V2E CENTRONICS I/F CABLE

SHEET: 1 OF 1	COPYRIGHT: 1988	DWG. NO: 528A004

Chapter 13

REAL-TIME CLOCK (RTC) - OPTIONAL FEATURE

13.1 INTRODUCTION

As an option, one pROM can be fitted with a special socket which has a built-in CMOS watch circuit and a lithium battery (Dallas Semiconductor, part number DS1216F).

The module socket is installed in the second HK68/V2E pROM position (U56). The RTC module must be plugged into the existing socket because there are components under the standard ROM socket. This makes the board profile wider (assuming standard pROM thickness), as shown below.

Configuration	Component Height Above Board	Minimum Board Spacing
RTC module plugged into existing pROM socket:	.75 in.	.85 in. (2 slots)

Table 13-1. RTC module, physical effects

Only one card slot is required if the board is in the end slot. The RTC logic does not generate interrupts; a CIO timer channel is still used for that purpose. The RTC contents, however, may be used to check for long-term drift of the HK68/V2E system clock, and as an absolute time and date reference after a power failure. Leap year accounting is included. Heurikon can provide complete operating system software support for the RTC module.

The RTC module time resolution is 10 milliseconds. The RTC internal oscillator is accurate to one minute per month, at 25 degrees C.

The clock contents are set or read using a special sequence of ROM read commands, as detailed in the program example, below. The RTC module "monitors" ROM accesses and, if a certain sequence of 64 ROM addresses occur, takes temporary control of the ROM

space, allowing data to be read from or written to the module. Writing is done by twiddling an address line, which the module uses as a data input bit. There are never any MPU write cycles directed to the pROM space.

Do not execute the module access instructions out of ROM. The instruction fetch cycles will interfere with the module access sequence. Also, be certain the reset disable bit (rtc_data.day bit D4) is always written as a "1".

```
#define WATCHBASE (unsigned char *)0x00040000 /* ROM socket */
#define WR0_WATCH (unsigned char *)(WATCHBASE+2) /* write 0 */
#define WR1_WATCH (unsigned char *)(WATCHBASE+3) /* write 1 */
#define RD_WATCH  (unsigned char *)(WATCHBASE+4) /* read */
struct rtc_data {        /* D7 D6 D5 D4   D3 D2 D1 D0   range */
    unsigned char dotsec;/*-0.1 sec-:-0.01 sec-;00-99 */
    unsigned char sec;   /* --10 sec-:-seconds-; 00-59 */
    unsigned char min;   /* --10 min-:-minutes-; 00-59 */
    unsigned char hour;  /*A  0  B Hr:-hours-; 00-23 */
    unsigned char day;   /*0  0  0  1:-day--; 01-07 */
    unsigned char date;  /*-10 date-:-date-; 01-31 */
    unsigned char month; /*-10 month-:-month-; 01-12 */
    unsigned char year;  /*-10 year--: --year----- ; 00-99 */
}; /* "A" =  "0" for 00-23 hour mode, "1" for 01-12 hour mode */
   /* "B" =  MSB of the 10 hours value (if 00-23 hour mode) else
          =  "0" for PM or "1" for AM  (if 01-12 hour mode) */

rtc_wr(data)             /* set the real-time clock */
register unsigned char *data;  /* rtc_data pointer */
{
    register int i, bit;
    unsigned char temp;
    static unsigned char key[] = { /* the unlock pattern */
        0xC5, 0x3A, 0xA3, 0x5C, 0xC5, 0x3A, 0xA3, 0x5C };

    if ( data ) {
        rtc_wr(0);       /* send key pattern */
    } else { /* this is the unlock function */
        i = *RD_WATCH; /* reset */
        data = key;
    }
    for( i=0; i<8; data++, i++ )
        for( bit = 1; bit & 0xff; bit <<= 1 )
            temp = ( *data & bit ) ? *WR1_WATCH : *WR0_WATCH;
}

rtc_rd(data)           /* read the real-time clock */
register unsigned char *data;  /* rtc_data pointer */
{
    register int i, bit;

    rtc_wr(0); /* send key pattern */
    for( i=0; i<8; data++, i++ ) {
        *data = 0;
        for( bit = 1; bit & 0xff; bit <<= 1 )
            *data |= (*RD_WATCH & 1) ? bit : 0 ;
    }
}
```

Fig. 13-1. Real-Time Clock, Example Software

Chapter 14

VMEBUS INTERFACE

14.1 INTRODUCTION

The VMEbus consists of P1 address, data, and control signals. P2 is used for the extended VMEbus address and data lines as well as the VSB.

14.2 P1 (VMEbus) Pin Assignments

P1 Pin Number	Row A Signal Mnemonic	Row B Signal Mnemonic	Row C Signal Mnemonic
1	D00	BBSY*	D08
2	D01	BCLR*	D09
3	D02	ACFAIL*	D10
4	D03	BG0IN*	D11
5	D04	BG0OUT*	D12
6	D05	BG1IN*	D13
7	D06	BG1OUT*	D14
8	D07	BG2IN*	D15
9	Gnd	BG2OUT*	Gnd
10	SYSCLK	BG3IN*	SYSFAIL*
11	Gnd	BG3OUT	BERR*
12	DS1*	BR0*	SYSRESET*
13	DS0*	BR1*	LWORD*
14	WRITE*	BR2*	AM5
15	Gnd	BR3*	A23
16	DTACK*	AM0	A22
17	Gnd	AM1	A21
18	AS*	AM2	A20
19	Gnd	AM3	A19
20	IACK*	Gnd	A18
21	IACKIN*	SERCLK	A17
22	IACKOUT*	SERDAT*	A16
23	AM4	Gnd	A15
24	A07	IRQ7*	A14
25	A06	IRQ6*	A13
26	A05	IRQ5*	A12
27	A04	IRQ4*	A11
28	A03	IRQ3*	A10
29	A02	IRQ2*	A09
30	A01	IRQ1*	A08
31	-12V	+5V STDBY	+12V
32	+5V	+5V	+5V

Table 14-1. P1 (VMEbus) Connector Pinout

Not all of the signals are used on the HK68/V2E. See section 7 for details and signal descriptions.

14.3 P2 (VSB) PIN ASSIGNMENTS

P2 is used for both the VMEbus and the VSB. The center row (B) of pins are the upper address and data lines of the VMEbus. The outer two rows (A and C) make up the VSB.

P2 Pin Number	Row A VSB Signal Mnemonic	Row B VMEbus Signal Mnemonic	Row C VSB Signal Mnemonic
1	AD00	+5	AD01
2	AD02	Gnd	AD03
3	AD04	(reserved)	AD05
4	AD06	A24	AD07
5	AD08	A25	AD09
6	AD10	A26	AD11
7	AD12	A27	AD13
8	AD14	A28	AD15
9	AD16	A29	AD17
10	AD18	A30	AD19
11	AD20	A31	AD21
12	AD22	Gnd	AD23
13	AD24	+5	AD25
14	AD26	D16	AD27
15	AD28	D17	AD29
16	AD30	D18	AD31
17	Gnd	D19	Gnd
18	IRQ*	D20	Gnd
19	DS*	D21	Gnd
20	WR*	D22	Gnd
21	SPACE0	D23	SIZE0
22	SPACE1	Gnd	PAS*
23	LOCK*	D24	SIZE1
24	ERR*	D25	Gnd
25	Gnd	D26	ACK*
26	Gnd	D27	AC
27	Gnd	D28	ASACK1*
28	Gnd	D29	ASACK0*
29	Gnd	D30	CACHE*
30	Gnd	D31	WAIT*
31	BGIN*	Gnd	BUSY*
32	BREQ*	+5	BGOUT*

Table 14-2. P2 (VMEbus, VSB) Connector Pinout

The use of P2 is *required* in order to meet VME power specifications.

14.4 POWER REQUIREMENTS

Voltage	Current	Usage
+5	9.0A, max	All logic
+12	20ma, max	RS-232 I/F
-12	20ma, max	RS-232 I/F

Table 14-3. Power Requirements

The "+5" and "Gnd" pins on P2 *must* be connected to assure proper operation.

14.5 ENVIRONMENTAL

Operating temperature: 0 to +55 degrees Centigrade, ambient, at board.

Humidity: 0% to 85%.

Storage temperature: -40 to +70 degrees C.

NOTICE: Power dissipation is about 45 watts.

Fan cooling is required if the HK68/V2E board is placed in an enclosure or card rack.

Fan cooling is also recommended when using an extender board for more than a few minutes.

14.6 MECHANICAL SPECIFICATIONS

Width	Depth	Height	(above board)
9.187 in.	6.299 in.	0.6 in.	0.8 in.
233.35 mm	160 mm	15.25 mm	20.35 mm

Table 14-4. Mechanical Specifications

If the Real Time Clock (RTC) module is installed, see section 13.

Standard board spacing is 0.8 inches. The HK68/V2E is a 10 layer board.

Chapter 15

SUMMARY INFORMATION

15.1 SOFTWARE INITIALIZATION SUMMARY

This section outlines the steps for initializing the facilities on the HK68/V2E board. Certain steps must be performed in sequence, while others may be rearranged or omitted entirely, depending on your application.

1. The MPU automatically fetches the reset vector following a system reset and loads the supervisor stack pointer and program counter. The reset vector is in the first 8 bytes of ROM.
2. Recall the NV-RAM contents. (Reference: section 6.8)
3. Determine RAM configuration. (Reference: section 6.4)
4. Set the Bus Control Latch. (Reference: section 7.8)
5. Clear on-card RAM to prevent parity errors due to uninitialized memory reads. (Reference: section 5.1)
6. Load the 68020 Vector Base Register with the location of your exception vector table (usually at the start of RAM).
7. Initialize the exception vector table in RAM (at the selected base address.) This step links the various exception and interrupt sources with the appropriate service routines. (Reference: section 3.3)
8. Initialize the CIO. (Reference: section 9.6)
9. Initialize the serial ports. (Reference: section 10.6)
10. Initialize the SCSI port. (Reference: section 11)
11. Initialize the Centronics port. (Reference: section 12)
12. Initialize the User LED port. (Reference: section 8.1)

13. Release the VMEbus SYSFAIL line. (Reference: section 7.7)

14. Initialize off-card memory and I/O devices, as necessary.

15. Enable system interrupts, as desired. (Reference: section 3.2)

15.2 ON-CARD I/O ADDRESSES

This section is a summary of the on-card port addresses. It is intended as a general reference for finding additional information about a particular device. Refer to section 6.6 for a pictorial description of the system memory map.

Hex Address	Type	Device	Reference Section
4xxx,xxxx	R/W	VMEbus (Extended Adrs mode)	7.7
04xx,xxxx	R/W	VSB or VMEbus	7.7, 7.9
03xx,xxxx	R/W	HK68/V2E on-card RAM	8.2
02F0,000x	R/W	SCC1 (Ports A & B)	12
02E0,000x	R/W	SCC2 (Ports C & D)	12
02D0,000x	R/W	CIO	11
0280,000x	R/W	Centronics	14
02C0,0000	W	Mailbox Base Address	7.8
02B0,0070	W	User LED 4	10.1
02B0,0060	W	User LED 3	10.1
02B0,0050	W	User LED 2	10.1
02B0,0040	W	User LED 1	10.1
02B0,0030	W	VMEbus Watchdog Enable	7.10
02B0,000E	W	VME SYSFAIL Release	7.6
02B0,000C	W	VME Slave Enable	7.4
02B0,000A	W	VSB Arbiter Enable	7.9
02B0,0008	W	VSB Release Mode	7.9
02B0,0006	W	SCSI Reset	11
02B0,0004	W	Mailbox Enable	7.8
02B0,0002	W	MPU Cache Disable	3.6
02A0,0000	W	Bus Control Latch	7.4
0290,000x	W	VMEbus Interrupt Request	7.5
0270,0000	R	NV-RAM Recall	6.8
0260,0000	W	NV-RAM Store (tas)	6.8
0250,00xx	R/W	NV-RAM Data	6.8
0230,000x	R/W	SCSI	11
0240,0000	R/W	SCSI Psuedo DMA	11
01xx,xxxx	R/W	VMEbus (Standard Space)	7.7
0C00,xxxx	R/W	VMEbus (Short Space)	7.7
0080,000x	R	VMEbus Interrupt Vectors	7.5
0040,0000	R	pROM 1	6.1
0000,0000	R	pROM 0	6.1

Table 15-1. Address Summary

15.3 HARDWARE CONFIGURATION JUMPERS

Jumper settings are detailed in the manual section pertaining to the associated device. This section can be used as a cross reference for finding additional information about the jumpers.

Jumper	Function	Reference Section	Standard Configuration
J1	RS-232 A & B Defaults	10.7	J1-A (True)
J2	RS-232 C & D Defaults	10.7	J2-A (True)
J3	P5-16 +12	10.9	removed
J4	P5-20 +5	10.9	removed
J5	P5-22 -12	10.9	removed
J6	P5-40 +12	10.9	removed
J7	P5-44 +5	10.9	removed
J8	P5-46 -12	10.9	removed
J9	ROM 0 type	6.2	J9-A (27512)
J10	ROM 0 type	6.2	J10-open (27512)
J11	ROM 0 type	6.2	J11-B (27512)
J12	ROM 0 type	6.2	J12-B (27512)
J13	ROM 1 type	6.2	J13-B (27512)
J14	ROM 1 type	6.2	J14-B (27512)
J15	ROM 1 type	6.2	J15-open (27512)
J16	ROM 1 type	6.2	J16-A (27512)
J17	RAM Size	6.4	(factory set)
J18	Bus Request level	7.3	J18-D (level 3)
J19	SYSCLK*	7.3	(removed)
J20	VME Slave Window Size	7.4	J20-B (1 Meg)
J21	VME Slave Window Size	7.4	J21-B (1 Meg)
J22	VME Slave Window Size	7.4	J22-B (1 Meg)
J23	VME Slave Window Size	7.4	J23-B (1 Meg)
J24	Bus Arbitration level	7.3	(lvl 3, non-cntrl)
J25	Bus Arbitration level	7.3	(lvl 3, non-cntrl)
J26	Bus Arbitration level	7.3	(lvl 3, non-cntrl)
J27	Bus Arbitration level	7.3	(lvl 3, non-cntrl)
J28	BCLR*	7.3	(removed)
J29	SYSRESET*	7.3	J29-A (input)

Table 15-2. Jumper Summary

REV.	DATE	DESCRIPTION

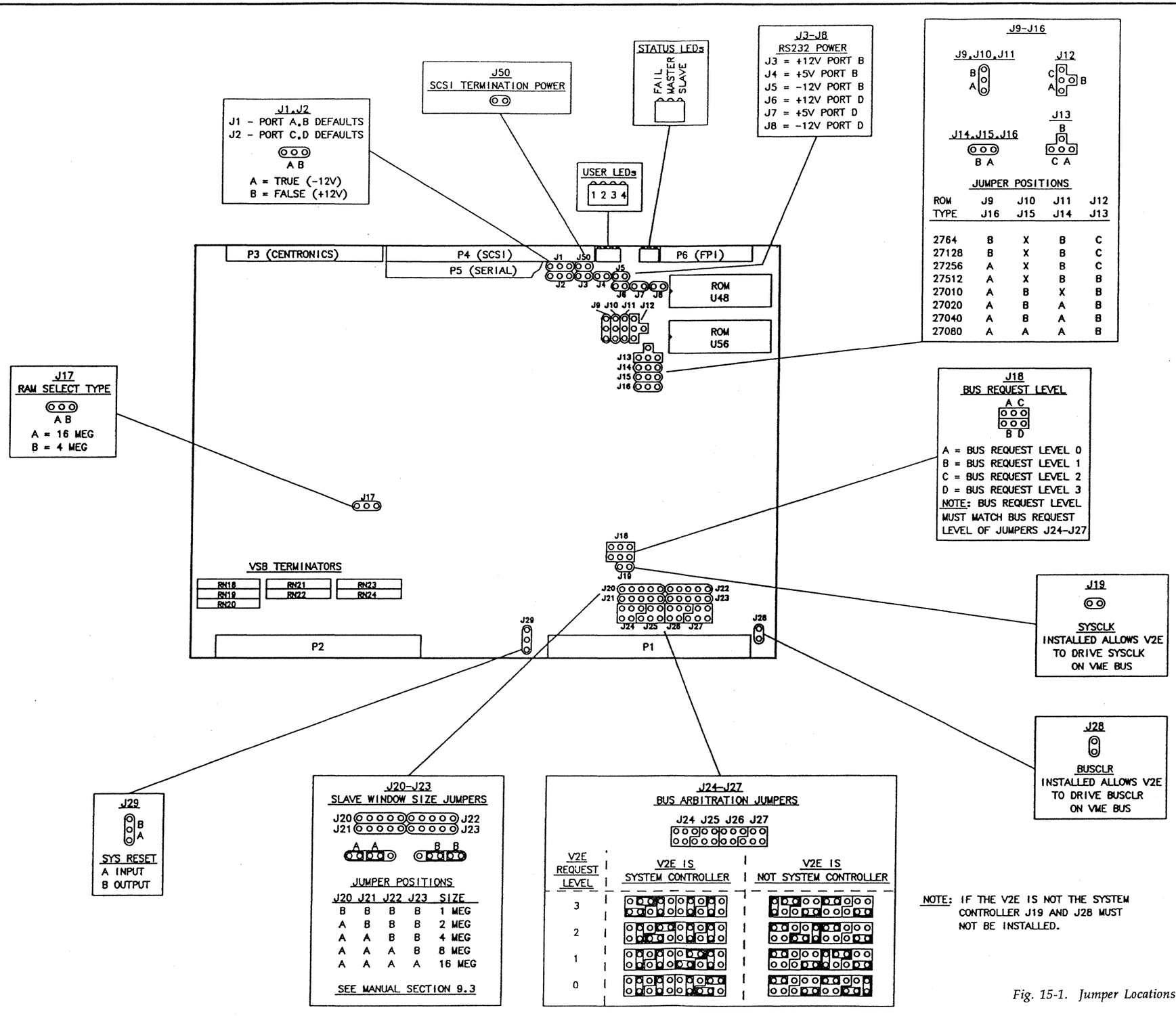


Fig. 15-1. Jumper Locations

HEURIKON CORP.

MADISON, WISCONSIN

DWN: RJC	CKD:	TOL: ± FRAC. ± DEC.) BREAK ALL SHARP CORNERS
DATE: 7/28/88	SCALE: NONE	ANGLE ± 1/2° ± DEC.)

TITLE: V2E JUMPER LOCATIONS

SHEET: 1 OF 1	COPYRIGHT: 1988	DWG. NO: 5280003

REV.	DATE	DESCRIPTION
A	8/25/88	FIX J1, J2

J1,J2
J1 - PORT A,B DEFAULTS
J2 - PORT C,D DEFAULTS
B A
A = TRUE (-12V)
B = FALSE (+12V)

J50
SCSI TERMINATION POWER

STATUS LEDs
FAIL
MASTER
SLAVE

USER LEDs
1 2 3 4

J3-J8
RS232 POWER
J3 = +12V PORT B
J4 = +5V PORT B
J5 = -12V PORT B
J6 = +12V PORT D
J7 = +5V PORT D
J8 = -12V PORT D

J9-J16

J9,J10,J11: B / A

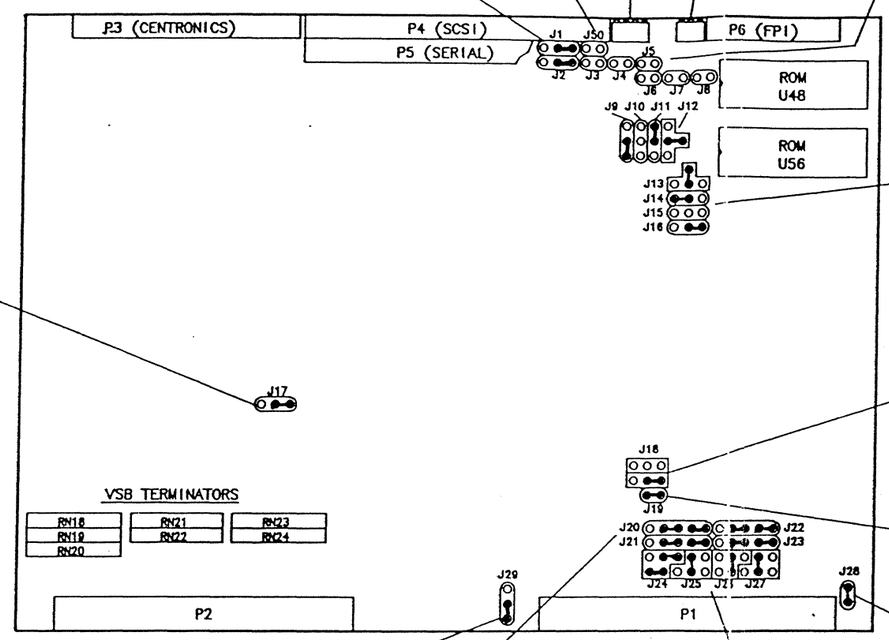
J12: C / A / B

J13: B / C A

J14,J15,J16: B A

JUMPER POSITIONS

ROM TYPE	J9 J16	J10 J15	J11 J14	J12 J13
2764	B	X	B	C
27128	B	X	B	C
27256	A	X	B	C
27512	A	X	B	B
27010	A	B	X	B
27020	A	B	A	B
27040	A	B	A	B
27080	A	A	A	B



J17
RAW SELECT TYPE
A B
A = 16 MEG
B = 4 MEG

J18
BUS REQUEST LEVEL
A C
B D
A = BUS REQUEST LEVEL 0
B = BUS REQUEST LEVEL 1
C = BUS REQUEST LEVEL 2
D = BUS REQUEST LEVEL 3
NOTE: BUS REQUEST LEVEL MUST MATCH BUS REQUEST LEVEL OF JUMPERS J24-J27

J19
SYSCLK
INSTALLED ALLOWS V2E TO DRIVE SYSCLK ON VME BUS

J28
BUSCLR
INSTALLED ALLOWS V2E TO DRIVE BUSCLR ON VME BUS

J29
B
A
SYS RESET
A INPUT
B OUTPUT

J20-J23
SLAVE WINDOW SIZE JUMPERS

J20 J22
J21 J23
A A B B

JUMPER POSITIONS

J20	J21	J22	J23	SIZE
B	B	B	B	1 MEG
A	B	B	B	2 MEG
A	A	B	B	4 MEG
A	A	A	B	8 MEG
A	A	A	A	16 MEG

SEE MANUAL SECTION 9.3

J24-J27
BUS ARBITRATION JUMPERS

J24 J25 J26 J27

V2E REQUEST LEVEL	V2E IS SYSTEM CONTROLLER	V2E IS NOT SYSTEM CONTROLLER
3		
2		
1		
0		

NOTE: IF THE V2E IS NOT THE SYSTEM CONTROLLER J19 AND J28 MUST NOT BE INSTALLED.

HEURIKON CORP.
MADISON, WISCONSIN

DWN: RJC	CKD:	TOL: 2 PL. ±.1 / 3 PL. ±.5 / BREAK ALL SHARP CORNERS
DATE: 7/28/88	SCALE: NONE	ANGLE ±1/2°
TITLE: V2E JUMPER LOCATIONS		
SHEET: 1 OF 1	COPYRIGHT; 1988	DWG. NO: 5280003 - A

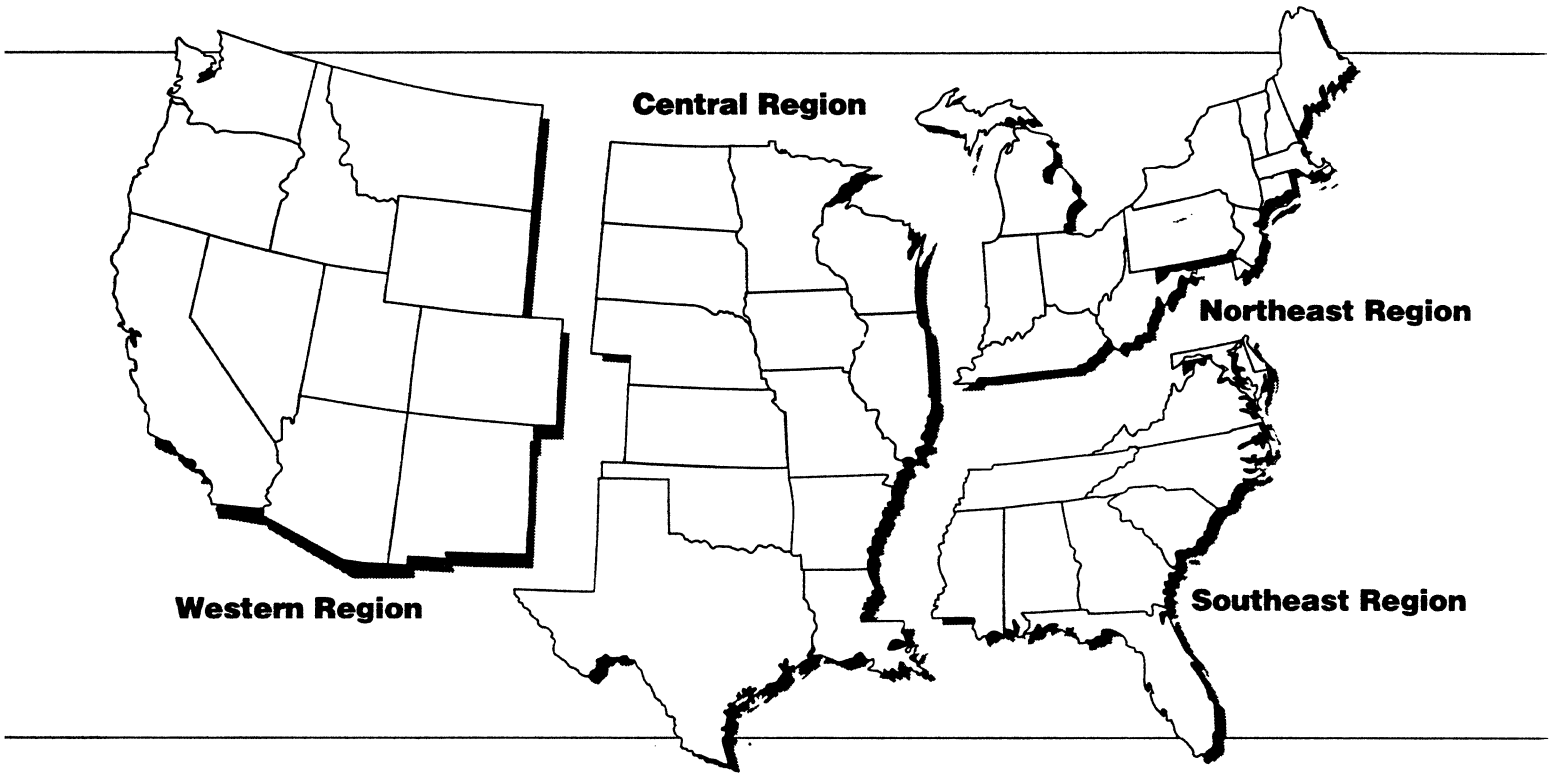
15.4 Additional Technical Literature

Additional information is available on the HK68/V2E peripheral chips, either from Heurikon sales or directly from the chip manufacturers.

Device	Number	Reference Document	Section
MPU	68020	Motorola 68020 Spec	5
FPP	68881	Motorola MC68881 Spec	6
CIO	Z8536	Zilog CIO Technical Manual	11
SCC	Z8530	Zilog SCC Technical Manual	12
SCSI	WD33C93	WD33C93 Technical Spec	13
RTC	DS1216F	Dallas Semiconductor Clock Module	15

Table 15-3. Additional Technical Literature

Sales and Customer Service Offices



Heurikon Corporate Office
8000 Excelsior Drive
Madison, WI 53717
Watts: 800-356-9602
Phone: 608-831-0900
Fax: 608-831-4249

Heurikon Customer Support and Factory Service Office
8310 Excelsior Drive
Madison, WI 53717
Watts: 800-327-1251
Phone: 608-831-5500

Heurikon Northeast Regional Office
67 South Bedford, Suite 400 W.
Burlington, MA 01803
Phone: 617-229-5831
Fax: 617-272-9115

Heurikon Southeast Regional Office
2010 Corporate Ridge, Suite 700
McLean, VA 22102
Phone: 703-749-1474
Fax: 703-556-0955

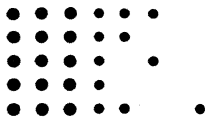
Heurikon Central Regional Office
13100 West 95th Street, Level 4D
Lenexa, KS 66215
Phone: 913-599-1860
Fax: 913-599-1918

Heurikon Western Regional Office
23121 Plaza Pointe Dr., Suite 109
Laguna Hills, CA 92653
Phone: 714-581-6400
Fax: 714-581-8509

HEURIKON CORP.

OPEN SYSTEMS :: OPEN TOOLS

Regional Sales Representatives



• NORTHEAST REGION

CT, DE, ME, MA, NH, NJ, NY, Eastern PA, RI and VT

Daner-Hayes, Inc.
62 West Plain Street
Wayland, MA 01778
Tel: (508) 655-0888
Fax: (508) 655-0939

IN, KY, MI, OH, Western PA and WV

The Systems Group
5340 Plymouth Road
Ann Arbor, MI 48105
Tel: (313) 930-1800
Fax: (313) 930-1803

• CENTRAL REGION

AR, LA, OK and TX

Acudata, Inc.
720 Avenue F
Suite 104
Plano, TX 75074
Tel: (214) 424-3567
Fax: (214) 422-7342

MN, ND, SD and Northwest WI

Micro Resources Corp.
4640 W. 77th Street
Suite 109
Edina, MN 55435
Tel: (612) 830-1454
Fax: (612) 830-1380

IL, IA, KS, MO, NE and Southeast WI

Panatek
2500 West Higgins Road,
Suite 305
Hoffman Estates, IL 60195
Tel: (708) 519-0867
Fax: (708) 519-0897

• SOUTHEAST REGION

MD, VA and Washington, DC

Spectro Associates
1107 Nelson Street
Suite 203
Rockville, MD 20850
Tel: (301) 294-9770
Fax: (301) 294-9772

AL, FL, GA, MS, NC, SC, TN

Heurikon Corporation
2010 Corporate Ridge
Suite 700
McLean, VA 22102
Tel: (703) 749-1474
Fax: (703) 556-0955

• WESTERN REGION

AZ, CO, NV, NM and UT

Compware Marketing
100 Arapahoe Avenue
Suite 7
Boulder, CO 80302
Tel: (303) 786-7045
Fax: (303) 786-7047

ID, MT, OR, WA, WY and Canada (Alberta and British Columbia)

Electronic Component Sales
9311 S. E. 36th Street
Mercer Island, WA
98040-3795
Tel: (206) 232-9301
Fax: (206) 232-1095

CA

Heurikon Corporation
23121 Plaza Pointe Drive
Suite 109
Laguna Hills, CA 92653
Tel: (714) 581-6400
Fax: (714) 581-8509

HEURIKON CORP.

OPEN SYSTEMS :: OPEN TOOLS

revised 7.91

INDEX

Heurikon Corporation
8310 Excelsior Drive
Madison, WI 53717

Customer Support: 1•800•327•1251

Sales: 1•800•356•9602

Heurikon Part Number 001M528