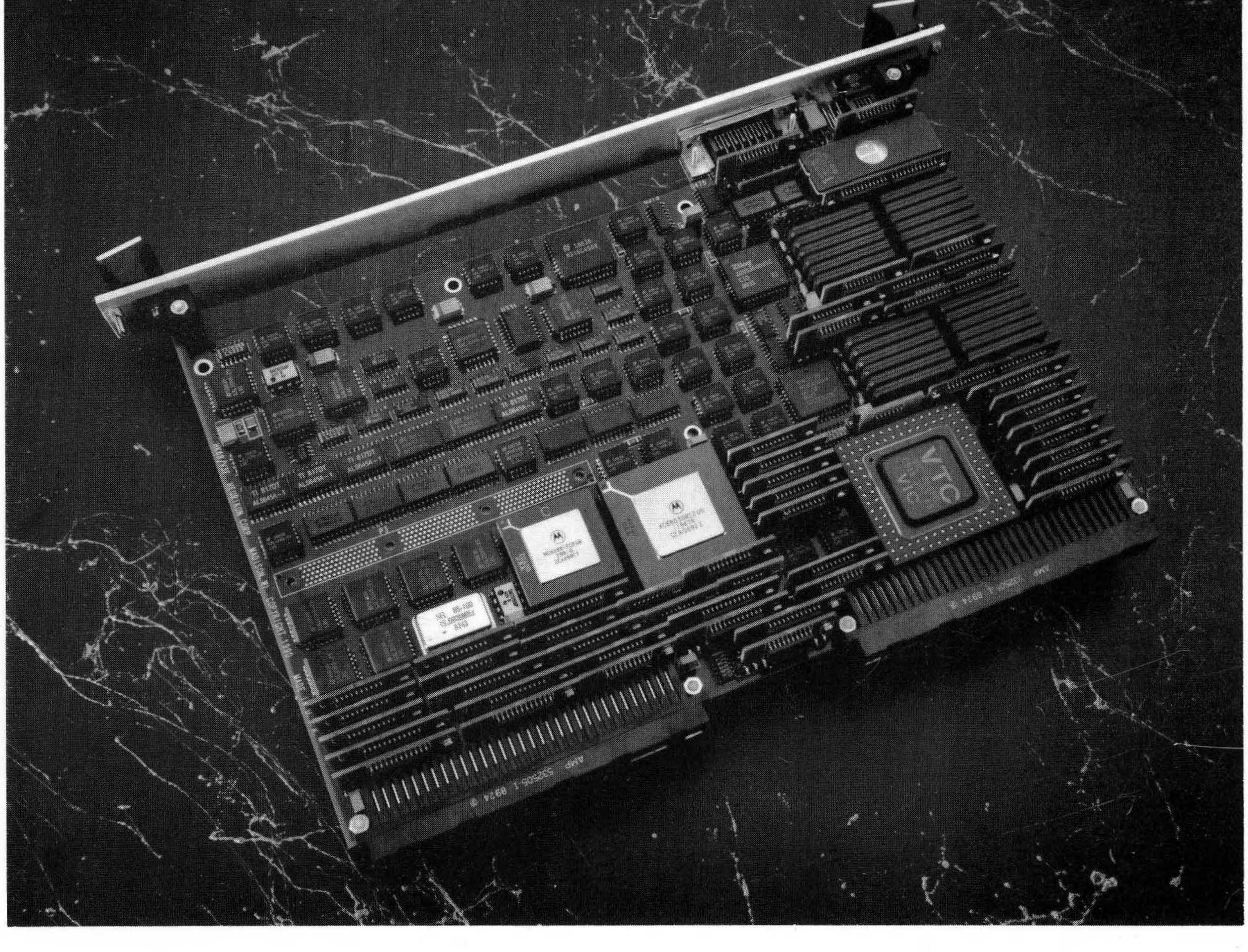
USER'S MANUAL
Revision D
September 1990

HK68/V3F

Heurikon Corebus™ 68030-based Computer



HEURIKON CORP.®
OPEN SYSTEMS : : OPEN TOOLS

HK68/V3F

Heurikon Corebus™ 68030-based Computer

USER'S MANUAL
Revision D
September 1990

HEURIKON® CORP.
OPEN SYSTEMS : : OPEN TOOLS

The information in this manual has been checked and is believed to be accurate and reliable. HOWEVER, NO RESPONSIBILITY IS ASSUMED BY HEURIKON FOR ITS USE OR FOR ANY INACCURACIES. Specifications are subject to change without notice. HEURIKON DOES NOT ASSUME ANY LIABILITY ARISING OUT OF USE OR OTHER APPLICATION OF ANY PRODUCT, CIRCUIT OR PROGRAM DESCRIBED HEREIN. This document does not convey any license under Heurikon's patents or the rights of others.

Heurikon, HK68/V, Hbug, and Corebus are trademarks of Heurikon Corporation.

REVISION HISTORY

Revision Level	Principal Changes	Date of Publication
A (Preliminary)		April 1990
B (Preliminary)		May 1990
C (Preliminary)		August 1990
D	Release version, board revision P	September 1990

Contents

1 — Overview

2 — Set-up and Installation

3 — Main Processor Unit

4 — Floating Point Coprocessor (FPP)

5 — System Error Handling

6 — On-card Memory Configuration

7 — VMEbus Control

8 — VME Subsystem Bus (VSB) Control

9 — Front Panel Interface

10 — Serial I/O

11 — Timer Clock Peripheral (TCP)

12 — Corebus Interface

13 — Summary Information

Index

Figures

Tables

1

Overview

1.1 INTRODUCTION

The HK68/V3F is a 32-bit single-board computer designed as a high-performance base for modular additions that interface to the board via Corebus™, a Heurikon-designed interface. The HK68/V3F uses the Motorola 68030 microprocessor and has two RS-232 serial ports, mailbox interrupt support, a built-in real-time clock, and full VMEbus compatibility.

1.2 HK68/V3F FEATURE SUMMARY

MPU The MPU is a Motorola 68030 32-bit microprocessor chip running at 25 or 33 MHz, depending on the option ordered. It has 32-bit internal architecture; 32-bit address and data paths; 4-Gbyte addressing range; a 256-byte instruction cache; and a 256-byte data cache. Section 3 contains details on the MPU.

FPP option The optional floating point coprocessor is a 32-bit Motorola 68882 for executing transcendental, trigonometric, and basic arithmetic functions. The FPP implements the IEEE standard for binary floating point arithmetic (ANSI/IEEE Std. 754-1985). FPP features are described in section 4.

RAM The RAM has 2- or 8-Mbyte capacity and optional parity (one bit per byte). It uses 256K x 4 or 1M x 4 static-column DRAMs. Hardware logic controls refresh. Section 6.3 contains details on RAM.

EPROM There is one 32-pin ROM socket with 1 Mbyte total capacity and an 8-bit data path. Section 6.2 contains details on the EPROM.

EEPROM An internal EEPROM provides 8 Kbytes of nonvolatile storage for user-definable and system parameters. The EEPROM has 100-year retention and 10,000 write-cycle lifetime. (Refer to section 6.7 for details.)

VMEbus The VMEbus interface is provided by a VTC VIC068 intelligent VMEbus controller/arbiter with a 32-bit address bus; 24- or 32-bit address modes (4-Gbyte range); a 32-bit data bus with 8-, 16-, or 32-bit board compatibility; and seven bus interrupts. Section 7 has details on the VMEbus interface.

VSB This high-speed local memory expansion subsystem supports secondary bus masters. See section 8 for details.

Mailbox The mailbox allows remote control of the HK68/V3F via specified VMEbus addresses. (Refer to section 7.5 for details.)

Serial I/O The HK68/V3F has two serial I/O ports controlled by a single Zilog Z85C30 serial communication controller (SCC). There are separate baud rate generators for each port, and asynchronous and synchronous modes. RS-232C interface is standard; RS-422 is optional. Section 10 contains details on the serial I/O.

TCP The timer clock peripheral provides a built-in real-time clock module. The TCP has a full-function real-time clock/calendar, power fail features, and 44 bytes of CMOS RAM. The TCP has two 16-bit timers, each with its own prescaler and selectable clock input with eight choices. Details on the TCP are in section 11.

LEDs There are three MPU/BUS status LEDs. Section 9 contains details.

Front Panel Interface The front panel interface connector has RESET input and HALT output. There are four user-programmable outputs. Details are in section 9.

Corebus Plugover Modules Corebus is a 32-bit interface that allows the addition of plugover modules such as Ethernet or SCSI for custom configurations. Corebus is described in section 12.

1.3 BLOCK DIAGRAM

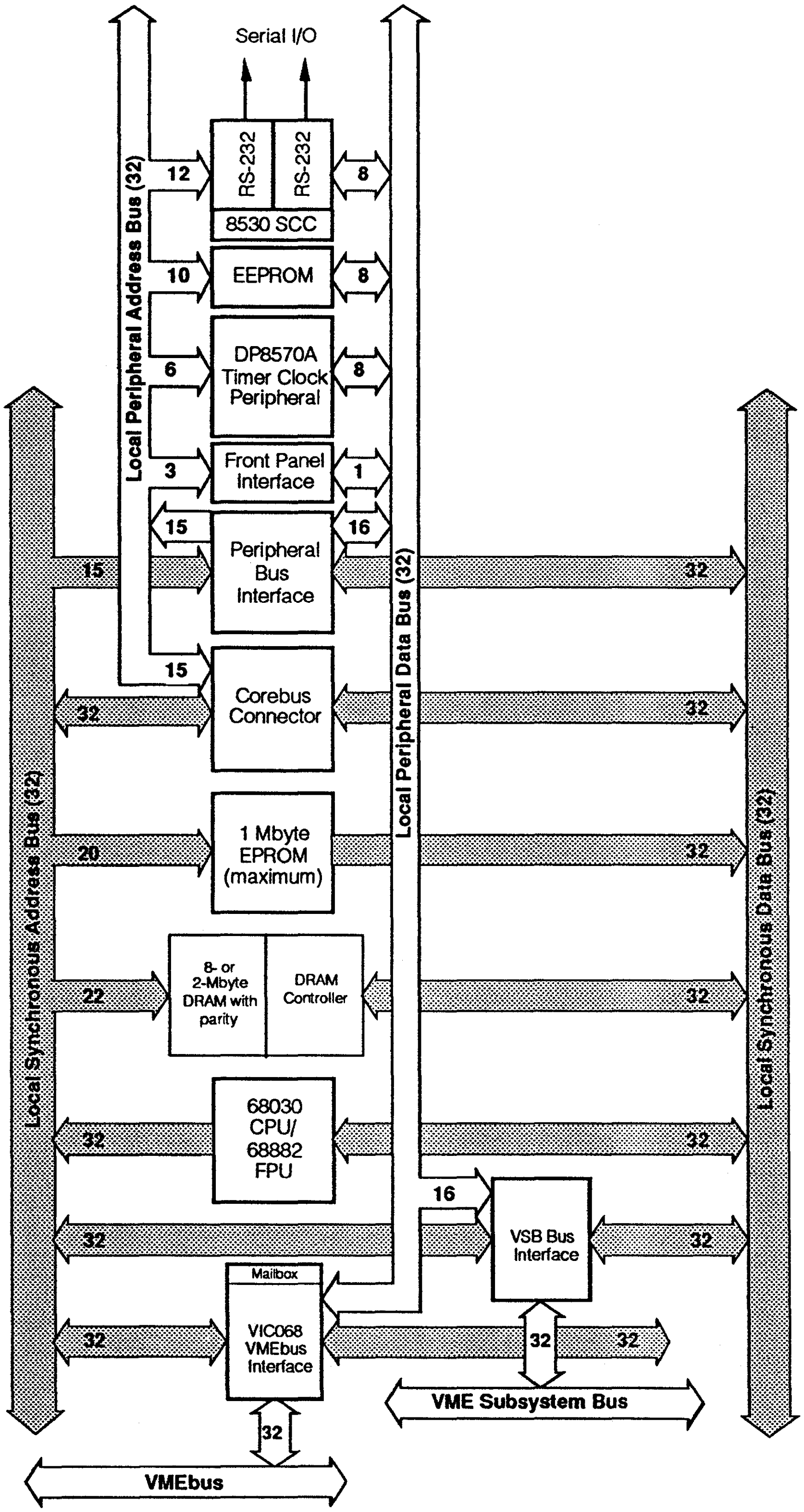


FIGURE 1-1. HK68/V3F block diagram

1.4 COMPONENT MAP

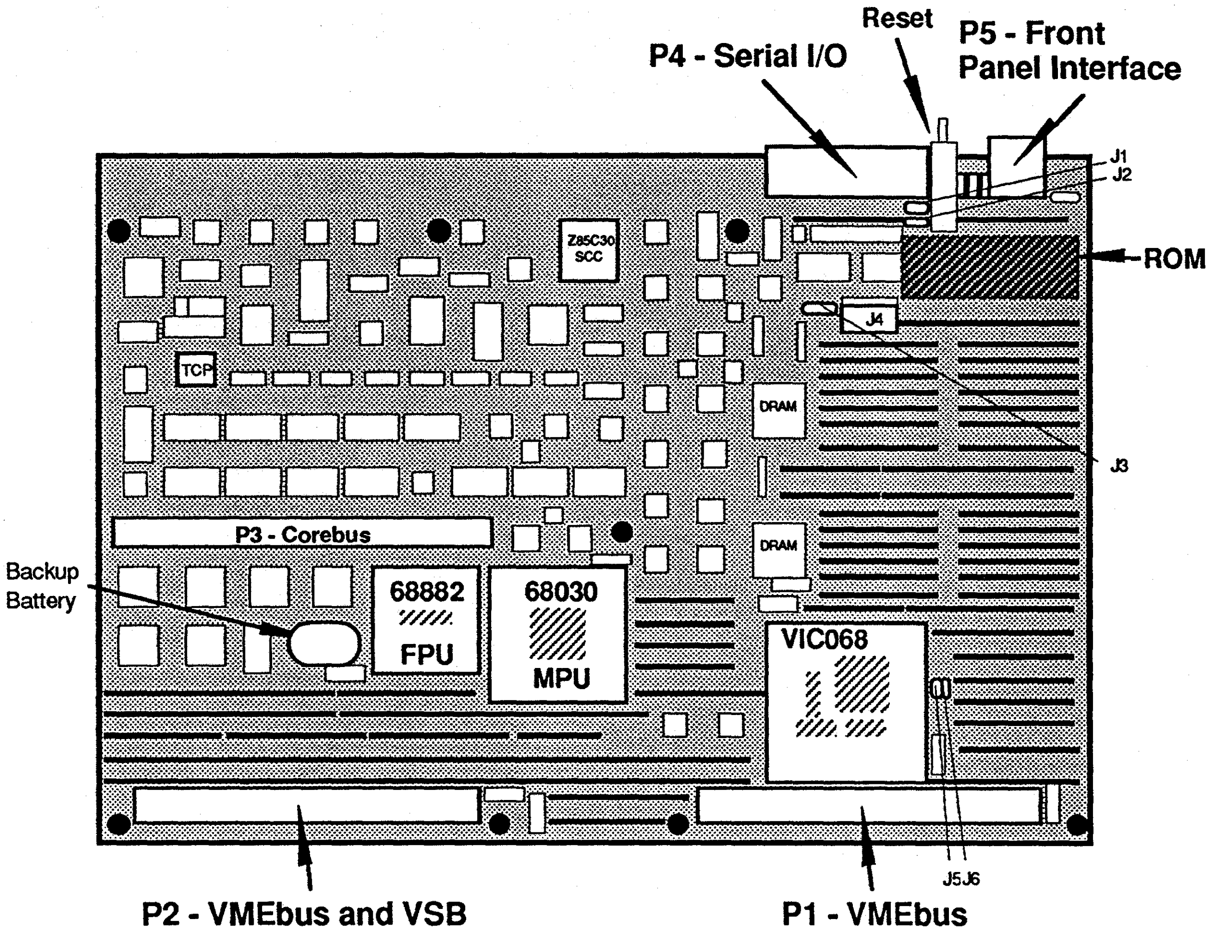


FIGURE 1-2. HK68/V3F component map

1.5 BUS SUMMARY

The HK68/V3F uses the Corebus Module Synchronous Bus and Module Peripheral Bus interfaces to provide a high level of performance along with the flexibility to expand on-card resources with plugover modules. The Corebus interface is described in section 12.

The VMEbus provides high throughput for data transfers between board or subsystems on the VMEbus, and is the main conduit for transferring system level information between processor subsystems. The VMEbus interface is described in section 7 and the VSB interface is described in section 8.

1.6 CONNECTORS, JUMPERS, AND SWITCHES

1.6.1 Connectors

The HK68/V3F has five connectors:

P1 and P2 P1 and P2 provide a standard interface for the VMEbus.

P3 Corebus connector for the two Corebus interfaces — Module Synchronous Bus and Module Peripheral Bus. The Corebus interface provides external connections for modules that contain Corebus-compatible extensions. The Corebus master interface allows modules to arbitrate for and gain control of the local bus. The arbitration supports the bus clear feature so that the module may retain ownership of the local bus until the bus is needed by a master. The Corebus master interface supports burst transfers, locked cycles, and other features. The Corebus master interface has highest priority as a requestor for the local bus. The VMEbus slave and VSB slave interfaces arbitrate at a lower priority.

Details and pin assignments are in section 12.

P4 The RS-232-C connector is a high-density 28-pin connector that holds two serial cables. Details and pin assignments are in section 10.

P5 The front panel interface uses a 10-pin connector. Pin assignments and details are in section 9.

1.6.2 Jumpers

There are five jumpers for custom configurations. Default settings and factory settings are summarized in Tables 2-1 and 13-2.

J1 — +5V power control for port A of P4.
J2 — +5V power control for port B or P4.
J3 — RS-232 handshaking
J4 — ROM size
J5 — System controller

1.6.3 Reset Switch

This switch resets the HK68/V3F and also resets the VMEbus if the HK68/V3F is the VME system controller.

1.7 OVERVIEW OF THE MANUAL

The manual has four main parts:

- Chapters 1 and 2 contain set-up information.
- Chapters 3 through 11 contain information for programming board components.
- Chapter 12 describes the Corebus interfaces.
- Chapter 13 contains summary information, including on-card I/O addresses and a jumper diagram.

1.7.1 Terminology and Notation

We have used *byte* for 8-bit quantities; *word* for 16-bit quantities; *long word* for 32-bit quantities; and *quad word* for 4-word quantities (that is, 64-bit quantities) throughout the manual.
Hexadecimal numbers are shown with a subscript *16* and binary numbers with a subscript *2*.

1.7.2 Additional Technical Information

This manual describes Heurikon's implementation of the intelligent components of this board. Further information on basic operation and programming can be found in the following documents:

TABLE 1-1
Technical references

MPU	*MC68030 User's Manual,* 2nd ed. (Englewood Cliffs, NJ: Prentice-Hall, 1989).
FPU	*MC68881/MC68882 Floating Point Coprocessor User's Manual,* 1st ed. (Englewood Cliffs, NJ: Prentice-Hall, 1985).
Corebus	*Corebus Technical Specification* (Madison, WI: Heurikon Corp., 1989).
VMEbus	*VIC068 VMEbus Interface Controller Specification* (Bloomington, MN: VTC Incorporated, 1989) and *The VMEbus Specification C.1* (Motorola, 1985).
VME Subsystem Bus	*The Parallel Sub-System Bus of the IEC 821 Bus, Revision C* (International Electromechanical Commission, 1986).
Serial Interface	*EIA Standard RS-232-C* (Washington, DC: Electronic Industries Association, 1969) and *Z8030 Z-BUS SCC/Z8530 SCC Serial Communications Controller Technical Manual* (Campbell, CA: Zilog, Inc., 1989).
Real-Time Clock	*Advanced Peripherals: Real Time Clock Handbook* (Santa Clara, CA: National Semiconductor Corporation, 1989).

Feel free to contact our Customer Support Department at 1-800-327-1251 if you have questions. We are prepared to answer general questions and provide help with specific applications.

2

Set-up and Installation

2.1 EQUIPMENT

Here is what you need to get the Heurikon HK68/V3F "on-the-air":

- Heurikon HK68/V3F microcomputer board
- VME card cage and power supply
- Serial interface cable (RS-232)
- CRT terminal
- Heurikon EPROM, which includes both monitor and bootstrap

CAUTION: **All semiconductors should be handled with care. Static discharges can easily damage the components on the HK68/V3F. Keep the board in an antistatic 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/V3F board.**

CAUTION: **Do not install the board in a rack or remove the board from a rack while power is applied, at risk of damage to the board.**

2.2 PRELIMINARY CONSIDERATIONS

2.2.1 Electrical

If you are adding the HK68/V3F to an enclosure, the power supply must be sufficient for the additional board.

TABLE 2-1
Power requirements

Voltage	Current	Usage
+5	9 A	All logic
+12	.5 A	RS-232 interface
-12	.5 A	RS-232 interface

Note: All of the "+5" and "Gnd" pins on P1 *and* P2 *must* be connected to ensure proper operation.

2.2.2 Physical

The board is a single-height VMEbus board (9.187" W x 6.299" H x 0.6" D) that occupies one slot in a VMEbus card cage.

2.2.3 Environmental

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

As with any printed circuit board, be sure that air flow to the board is adequate. Recommended air flow rate is about 2-3 cubic feet per minute, depending on card cage constraints and other factors. Operating temperature is specified at 0° to 55° C ambient, as measured at the board.

2.3 INSTALLATION AND POWER-UP

All products are fully tested before they are shipped from the factory (please contact us if you would like to have current information on mean time between failures). When you receive your HK68/V3F, follow these steps to assure yourself that the system is operational:

1. Visually inspect the board(s) for components that could have loosened during shipment. Visually inspect the chassis and all cables. Be sure the ROM is in place and the board is jumpered correctly (see the jumper diagram in section 13.3). Be sure all boards are seated properly in the VME card cage. Be sure all cables are securely in place.

2. Connect a CRT terminal to serial port B (port A for the VxWorks operating system), via connector P4. If you are making your own cable, refer to section 10.10. 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. 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. Read the monitor manual and the operating system literature. Short course: type **bf** (for "boot floppy") to boot the operating system or **bw** to boot from Winchester.

6. Reconfigure the jumpers, etc., as necessary for your application. See section 13 for a summary of I/O device addresses and configuration jumpers.

2.4 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.

3. If the monitor program is executing, run the diagnostics by using the monitor command **uc** or **um**.

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

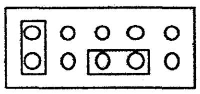
5. Check the chips to be sure they are firmly in place. Look for chips with bent or broken pins. In particular, check the EPROM.

6. Check your terminal switches and cables. Be sure the P4 connector is secure. If you have made your own cables, pay particular attention to the cable drawing in section 10.

7. Check the jumpers to be sure your board is configured properly. All jumpers should be in the "standard configu-

ration" positions shown in Table 2-2. Check the EPROM jumpers, especially.

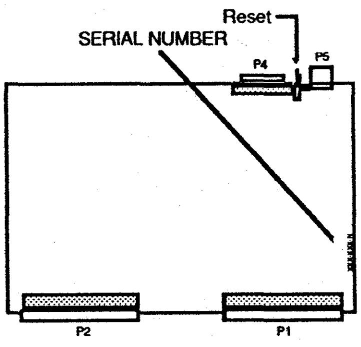
TABLE 2-2
Standard jumper settings

Jumper	Function	Reference Section	Standard Configuration
1	+5V power for Port A	10	Not installed.
2	+5V power for Port B	10	Not installed.
3	RS-232 handshaking A: +12V B: -12V	10	J3-A True J3-B False
4	ROM size	6	**Front panel ↑**  **512 Kbit**
5	System controller	3	Not installed.

8. Since the HK68/V3F monitor uses its on-card EEPROM to configure and set the baud rates for its console port, the lack of a prompt might be caused by incorrect terminal settings, an incorrect configuration of the EEPROM, or a malfunctioning EEPROM. Try pressing the **H** character a few times after a reset. If the prompt comes up, the EEPROM was most likely configured incorrectly. For more information about the way that the EEPROM configures the console port baud rates, refer to the *Heurikon Hbug User's Manual.*

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

    - The monitor program revision level (part of sign-on message)
    - The serial number of the operating system.
    - The HK68/V3F p.c.b. serial number (inscribed along the card edge).
    - Whether your board has been customized for options such as processor speed or configuration for networking and peripherals.



If you plan to return the board to Heurikon for service, contact our Customer Service Department at 1-800-327-1251 to obtain a **Return Merchandise Authorization** (RMA) number. We will ask you to list which items you are returning and the board serial number, plus your purchase order number and billing information if your HK68/V3F is out of warranty. If you return the board, be sure to enclose it in an antistatic 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 outside of the package so we can handle your problem efficiently. Our service department cannot accept material received without an RMA number.

2.5 MONITOR SUMMARY

An optional EPROM-based debug-monitor/bootstrap for the HK68/V3F is available. General features and functions include the ability to:

- Manually download data or MC68030 program code.
- Check the processor, memory, VME, VSB, and I/O devices.
- Execute a bootstrap (for example, boot an operating system).

The monitor uses the area between $0000,0000_{16}$ and $0000,1000_{16}$ for stack and uninitialized-data space. Any *writes* to that area can cause unpredictable operation of the monitor. The monitor initializes this area (that is, writes to it) to prevent parity errors, but it is the programmer's responsibility to initialize any other memory areas that are accessed.

A full description of monitor commands is in the *Heurikon Hbug Monitor User's Manual.*

3

Main Processor Unit

3.1 INTRODUCTION

This section details some of the important features of the 68030 MPU and the specifics of its implementation on the Heurikon HK68/V3F.

Refer to the MC68030 User's Manual for more information on the features described in this section.

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.

The MPU has seven interrupt levels. The VIC chip acts as interrupt controller for 19 possible interrupt sources. All interrupt priorities are handled by the VIC. These interrupts have a fixed priority within the VIC, but the encoding on the MPU IPL lines is programmable. The source of the interrupt vector — VIC or device — is programmable. A summary of the VIC interrupt levels is shown in section 7.4.2.

When the MPU recognizes an interrupt, it completes the current instruction and initiates an interrupt acknowledge sequence 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.

Section 7.4.2 has more information on the HK68/V3F 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 (four bytes) long, except for the reset vector, which is four words long. The listing below shows the vector space as it appears to the HK68/V3F MPU. It is more specific than the 68030 MPU manual listing because of particular implementations on the HK68/V3F 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 (such as code or data) or point to an error routine.

TABLE 3-1
MPU exception vectors

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)
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 (for program development)
10	028	Instruction Group 1010 Emulator
11	02C	FPP coprocessor not present
12	030	(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.
25	064	Level 1 autovector
26	068	Level 2 autovector
27	06C	Level 3 autovector
28	070	Level 4 autovector
29	074	Level 5 autovector
30	078	Level 6 autovector
31	07C	Level 7 autovector
32-47	080-0BF	TRAP Instruction Vectors — 16
48-54	0C0-0DB	FPP Exceptions — 8
55	0DC	(reserved)
56	0E0	MMU Configuration Error
57-63	0E4-0FF	(reserved — 6)
64-255	100-3FF	User Interrupt Vectors — 192

3.4 STATUS LEDs

Three status LEDs continuously show the state of the board as follows:

TABLE 3-2
Status LEDs

LED Label	Name	Meaning
HLT	HALT	The MPU has halted. (Double bus fault, odd stack address or the system reset line or mailbox halt line is active.)
ALT	ALT	A device other than the MPU owns the local bus. This device could be the VME bus, the VSB, or the Corebus plugover module.
SYF	SYSFAIL	The SYSFAIL line is being driven active by this board.

3.5 MPU CACHE CONTROL

The MPU cache is always enabled externally; however, it can be disabled by means of the cache control register (CACR). When D0 = 0, cache is enabled (default). When D0 = 1, cache is disabled.

The cache control register in the 68030 must be set properly to enable the MPU cache and cache burst accesses. See section 6.1 of the Motorola *MC68030 User's Manual.*

If caching is enabled, data and instructions from on-card ROM and DRAM will be cached. Data and instructions from the VMEbus will also be cached. VMEbus is cached for the standard or extended space when the CIOUT signal of the MPU is not asserted. The CIOUT pin follows a bit in the MMU page table entries. The bit allows caching on a page-by-page basis. See section 7.4.3 for further information on VMEbus caching.

3.5.1 Instruction Cache

The MC68030 contains a 256-byte direct-mapped instruction cache organized as 16 lines. Each line is composed of four long-word blocks. Each long word constitutes a separate cache entry. The instruction cache is controlled by five bits, 4-0, in the cache control register.

A tag for each line contains the 24 most significant logical address bits, the function code bit used to distinguish between user and supervisor accesses, and four valid bits corresponding to each long word. A comparator compares the address and function code bits in the selected tag with bits from the internal prefetch request to determine if the requested word is in the cache. The result of the comparison determines whether a word is supplied to the instruction pipe or a normal instruction prefetch occurs.

The instruction cache enhances the MC68030's performance by storing instruction prefetches (instruction words and extension words).

The instruction cache may be enabled or disabled via the **movec** instruction. See the MC68030 user's manual for details.

3.5.2 Data Cache

The MC68030 also contains a 256-byte direct-mapped data cache organized as 16 lines. Like the instruction cache, each line is composed of four long-word blocks. Each long word constitutes a separate cache entry. The data cache is controlled by six bits, bits 8-13 of the cache, in the cache control register.

Operation of the data cache is similar to the instruction cache. A tag for each line contains the 24 most significant logical address bits, the function code bit used to distinguish between user and supervisor accesses, and four valid bits corresponding to each long word. The tag is used to determine the cacheability of the data.

The data cache stores data references to any address space except CPU space. The data cache may be enabled or disabled via the **movec** instruction. See the MC68030 user's manual for details.

3.6 COPROCESSORS

The HK68/V3F supports a floating point coprocessor, which is described in section 4.

3.7 WATCHDOG TIMER

The HK68/V3F watchdog timer is controlled by the VIC chip. The timeout period is programmable from 4 to 512 microseconds and can be disabled. See section 7.11 for more details on the watchdog timer.

4

Floating Point Coprocessor

4.1 INTRODUCTION

The HK68/V3F allows the use of an optional MC68882 floating point processor chip. It runs as a coprocessor with the MPU.

4.2 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 *MC68881/MC68882 User's Manual.*

4.3 FPP BYPASS

The HK68/V3F 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 (via the VIC068), the software can determine if the FPP chip is installed. An attempt to access a nonexistent FPP causes a watchdog timeout and a bus error, forcing a Line 1111 MPU exception, vector number 11.

5

System Error Handling

Error conditions can result from many sources. The following error conditions might occur 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$_{16}$.
Watchdog Timeout	During an on-card access or VMEbus slave access, no acknowledge was received within a fixed time interval defined by a register in the VIC068. The usual cause is that no bus device was 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 on-card bus cycle, 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 that could cause this error is an improperly initialized stack pointer that 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 a hardware reset from the reset switch, SYSRESET* on the VMEbus, or another VMEbus master writing to ICR7 bit 6.

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.

Privilege 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 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 *48* to *54*.

6

On-card Memory Configuration

6.1 INTRODUCTION

The Heurikon HK68/V3F microcomputer will accommodate several RAM and ROM configurations. There is one 32-pin ROM socket for PROM or page-addressable ROM or EEPROM; 24 ZIP RAM positions; and an electrically erasable PROM (EEPROM). Off-card memory may be accessed via the VMEbus, the VSB, or Corebus.

6.2 ROM

The ROM occupies a fixed 1-Mbyte physical address space. At power-up or after a system reset, ROM is mirrored throughout the entire MPU address space. The MPU fetches the reset vector from location 0, which specifies the initial program counter and stack pointer values. Then RAM is turned on and the standard memory map is activated on the first access to the range $FC00,0000_{16}$ – $FC40,0000_{16}$ where EPROM usually resides. Thus, the reset vector may point directly to ROM (at base address $FC00,0000_{16}$).

The ROM contains consecutive (both even and odd) addresses.

TABLE 6-1
ROM address summary

Base Address	ROM
$FC00,0000_{16}$	0

Associated with the ROM socket is a set of jumpers that must be set according to the type of ROM being used. The HK68/V3F supports EPROM from 256 Kbits to 8 Mbits (27256 – 27080), as shown in Table 6-2. The ROM size and associated configuration are shown in Figure 6-1:

TABLE 6-2
ROM capacity and J4 jumper positions

PROM Type	ROM Capacity	Jumper Positions
27256	256 Kbits	J4: 2–4 and 5–7
27512	512 Kbits	J4: 1–2 and 5–7
27010	1 Mbits	J4: 1–2, 4–6, and 7–9
27020	2 Mbits	J4: 1–2, 3–5, 4–6, and 7–9
27040	4 Mbits	J4: 1–2, 3–5, 6–8, and 7–9
27080	8 Mbits	J4: 1–2, 3–5, 6–8, and 9–10

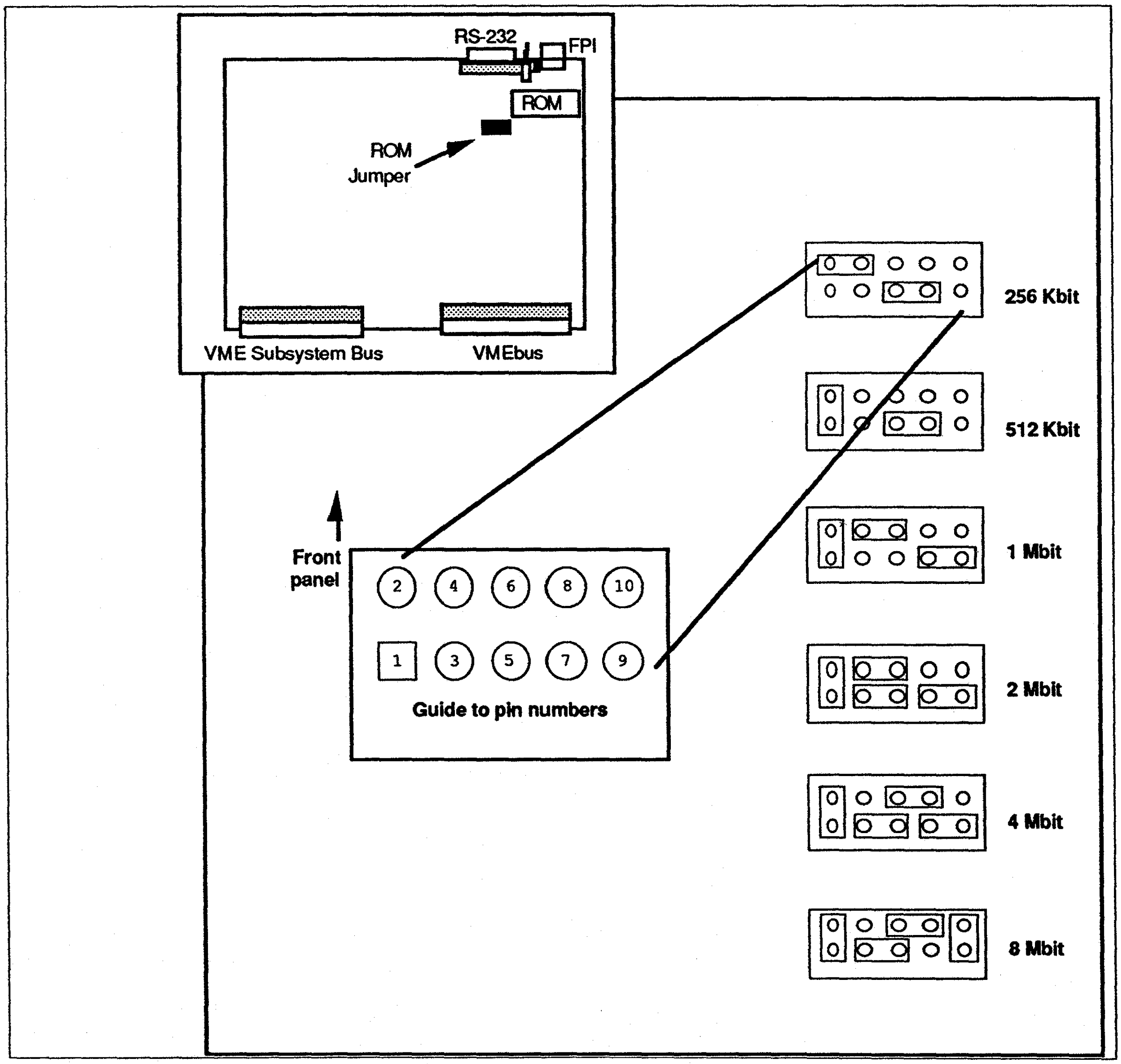


FIGURE 6-1. Jumper settings for ROM options

The ROM socket has 32 pins. If you use a 28-pin device, justify it so that socket pins 1, 2, 31, and 32 are empty. Twenty-four-pin devices are not supported. The ROM access time must be ≤250 nanoseconds.

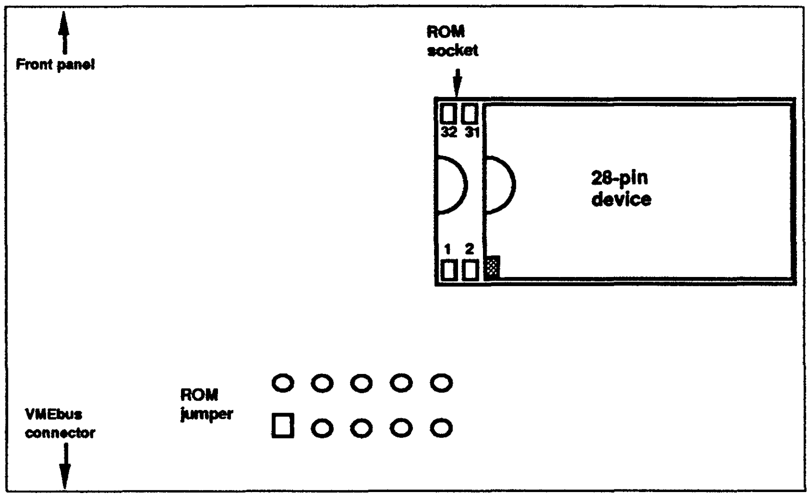


FIGURE 6-2. Placement for a 28-pin ROM

You may also use electrically erasable or paged PROMs. 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.

Paged ROMs allow future growth of ROM capacity without adding address pins. A single device can contain multiple 16-Kbyte pages. A specific page is selected by ***writing*** the page value to the ROM. For example, to select page three of a 27513, write 03_{16} to address $FC00,0000_{16}$.

6.3 ON-CARD RAM

The HK68/V3F uses 16 ZIP RAM packages, each four bits wide, for data storage. There is optional parity of one bit per byte; 8 ZIPs are used for parity. Standard memory configurations are 2 or 8 Mbytes. On-card RAM occupies physical addresses starting at $0000,0000_{16}$. The base address of the HK68/V3F RAM on the VMEbus is controlled by a set of base address bits in the VMEbus slave control register. The HK68/V3F is able to match 24-bit or 32-bit addresses and therefore resides in either the VMEbus standard or extended space. The address of the space is set by bits in the VMEbus control latch. The VSB base address is controlled by the VSB slave control register.

6.3.1 On-card Memory Sizing

The size of on-card memory is fixed at 2 Mbytes or 8 Mbytes with VSB following, so memory sizing is not possible.

6.4 BUS MEMORY

The bus interface is described in section 7.

6.5 PHYSICAL MEMORY MAP

See section 13.2 for an I/O device address summary.

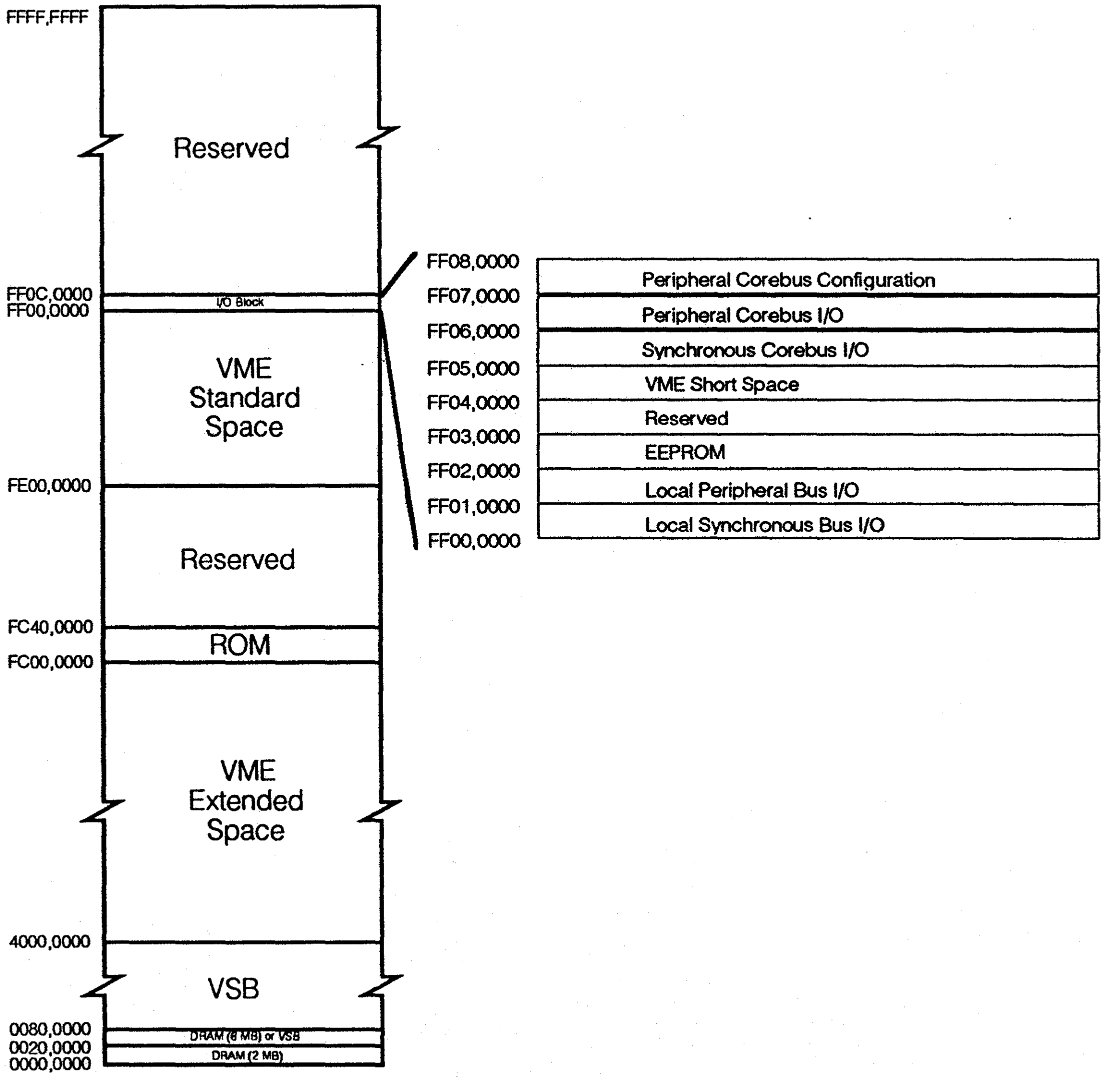


FIGURE 6-3. Physical memory map

6.6 MEMORY TIMING

The HK68/V3F memory logic has been carefully tuned for optimum memory cycle times under a variety of conditions.

The base cycle time for a MC68030 is two clock cycles for a RAM read or write and one clock cycle for subsequent burst cycles. The following chart shows total access times required to get these base cycle times out of a RAM interface. It should be noted that this is the time from address valid to data input setup of the MC68030, including clock skew and various other factors.

TABLE 6-3
Access time required for no wait states

CPU Speed	(read cycle)	(write cycle)	(burst cycle)
25 MHz	18 ns	38 ns	38 ns
33 MHz	15 ns	30 ns	30 ns

The HK68/V3F utilizes several features to provide the memory bandwidth the processor requires.

Reads

The HK68/V3F provides a bank interleave memory structure that allows the concurrent access of adjacent long words in memory. The bank interleaving of read cycles allows the processor to achieve no wait state on the second, third, and fourth accesses of a burst read cycle at 33 MHz.

Writes

The HK68/V3F also performs write posting of memory write cycles. This allows the processor to terminate the write cycle early, permitting the memory to complete the write cycle. With the combination of bank interleaving and write posting, the HK68/V3F can achieve no-wait-state cycles for writes and burst writes.

The use of bank interleaving and write posting provides the HK68/V3F with nearly no-wait-state performance. The only wait state occurs on read cycles and the first cycle of a read burst cycle.

Table 6-4 describes the expected wait states for the HK68/V3F:

TABLE 6-4
RAM access time required for the HK68/V3F

Cycle	Total Clocks	Wait States
Reads	4	2
Writes	2	0
Burst Read (4 accesses)	7	2-0-0-0
Burst Write (4 accesses)	5	0-0-0-0

The HK68/V3F will provide either 2 or 8 Mbytes of memory using 70-nanosecond DRAMs.

There are two other sources of wait states that DRAM architectures can exhibit.

The first is caused when a refresh must be performed and the DRAM controller is unable to perform the refresh during non-RAM cycles. This happens so infrequently that any performance degradation is usually unnoticeable.

The second occurs when the processor is required to perform back-to-back memory cycles with no delays. This also rarely occurs because of the instruction cache and data cache. In the event of a back-to-back memory cycle, an additional two-clock-cycle wait is inserted between accesses.

While the above information is important in comparing the relative performance of DRAM designs, the performance of individual DRAM designs has much less impact on overall system performance than one might expect. The reason for this is that the internal instruction and data caches built into the 68030 chip help to decouple the processor from slower speed memories such as DRAMs.

To summarize, the higher the cache hit rates, the less impact external memory has on system performance.

6.7 EEPROM

The HK68/V3F has an EEPROM for storing and recovering data or system configurations across power cycles.

The HK68/V3F has 8 Kbytes of EEPROM, which is configured for byte-wide (8-bit) access. There are 8 bytes between each byte, so EEPROM occupies 64 Kbytes of address space. The EEPROM is accessible as shown below.

TABLE 6-5
EEPROM addresses

Address (hex)	Mode	Function
FF02,0000$_{16}$	R/W	Read/Write EEPROM contents. Writes must be enabled through access to a separate location.
FF01,0C00$_{16}$	RMW	Temporarily enables writes to EEPROM. EEPROM write must be next peripheral operation. The 68030 **tas** instruction must be used for this operation.

Write operations take a maximum of 10 milliseconds to complete. Upon writing, reads from the location written will return complimented data until the write completes, at which point the data written will be read.

There is no limit on read operations, but operation is only guaranteed up to 10,000 write cycles to any single location. The use of a **tas** instruction to enable writes was implemented to prevent unintentional writes by an errant program or power failure.

The EEPROM is mapped from FF02,0000$_{16}$ to FF03,0000$_{16}$; however, the upper quarter from FF02,C000$_{16}$ to FF03,0000$_{16}$ is hardware protected and can only be modified when the EEPROM protect jumper is installed.

The HK68/V3F monitor (Hbug) and certain system programs use the EEPROM. The exact amount reserved for Heurikon usage depends on the system. A major portion of the device, however, is available for customer use. Heurikon usage is summarized in Table 6-6 below (details are available separately):

TABLE 6-6
EEPROM contents (partial)

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

7

VMEbus Control

7.1 INTRODUCTION

The HK68/V3F has a VMEbus interface that conforms to the specifications set forth in the following section. The VMEbus interface consists of the VIC068 VMEbus Interface Controller (VIC) and required support circuitry to perform all VMEbus functions. 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.

Please refer to the *VIC068 VMEbus Interface Controller Specification* from VTC for detailed descriptions of the registers.

The HK68/V3F VME interface has the following features:

Address The VMEbus interface uses 32 address lines for a total of 4 Gbytes of VMEbus address space. Supported are the "short," "standard," and "extended" address modes, which use 16, 24, and 32 address lines, respectively.

Data The VMEbus interface uses 32 data lines to support 8-, 16-, 24-, or 32-bit data transfers.

Interrupts The VIC handles the seven VMEbus interrupts and multiple local interrupts.

Mailbox The mailbox consists of a collection of 8-bit registers that can be used for interprocessor communications over the VMEbus.

System Controller The HK68/V3F may be configured as the VMEbus system controller to perform the necessary system controller functions of SYSCLK, BCLR, SYSRESET, bus watchdog, and bus arbiter.

7.2 BUS CONTROL SIGNALS, VMEbus P1 DESCRIPTIONS

VME pins are defined on P1 and part of P2. VSB is defined on the rest of P2. Refer to the Motorola VMEbus specification, revision C.1, for detailed usage of these signals. All signals are bidirectional unless otherwise stated. Refer to sections 7.12 and 7.13 for a complete listing of the pins.

The bus request level is set in the arbiter and requestor configuration register (B3).

The following signals on connectors P1 and P2 are used for the VMEbus interface.

A01-A15 ADDRESS bus (bits 1-15). Three-state address lines that are used for short, standard, and extended addresses.

A16-A23 ADDRESS bus (bits 16-23). Three-state address lines that are used for standard and extended addresses.

A24-A31 ADDRESS bus (bits 24-31). Three-state address lines that are used for extended addresses only.

ACFAIL* AC FAILURE. This signal is an input to the HK68/V3F and may be used to generate an interrupt to the MC68030 by programming the VIC accordingly.

AM0-AM5 ADDRESS MODIFIER (bits 0-5). Three-state lines that are used to broadcast information such as address size and cycle type.

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

BBSY* BUS BUSY. An open-collector signal driven 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 signal generated by the ARBITER to indicate when there is a higher priority request for the bus. This signal requests the current MASTER to release the bus.

BERR*

BUS ERROR. An open-collector 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 signals generated by the ARBITER and REQUESTERS. Bus-grant-in and bus-grant-out signals form bus grant daisy chains. An input to the HK68/V3F, the bus-grant-in signal indicates that it may use the bus if it wants.

BG0OUT*-BG3OUT*

BUS GRANT (0-3) OUT. Totem-pole signals generated by REQUESTERS. An output from the HK68/V3F, 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 signals generated by REQUESTERS. Assertion of one of these lines indicates that some MASTER needs to use the bus.

D00-D31

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

DS0*, DS1*

DATA STROBE ZERO, ONE. A three-state 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 are available on the data bus.

DTACK*

DATA TRANSFER ACKNOWLEDGE. An open-collector signal generated by a SLAVE. The falling edge of this signal indicates that valid data are available on the data bus during a read cycle, or that data have 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 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 signal and an input to the HK68/V3F. The IACKIN* indicates that the board may respond to the INTERRUPT ACKNOWLEDGE CYCLE that is in progress.

IACKOUT*

INTERRUPT ACKNOWLEDGE OUT. A totem-pole signal and an output from the HK68/V3F. The IACKIN* and IACKOUT* signals form a daisy-chain. The IACKOUT* signal indicates to the next board in the daisy-chain that it may respond to the INTERRUPT ACKNOWLEDGE CYCLE in progress.

IRQ1*-IRQ7*

INTERRUPT REQUEST (1-7). Open-collector 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*

LONG WORD. A three-state 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 signal that is used to synchronize the data transmission on the VMEbus. This signal is not implemented on the HK68/V3F.

SERDAT*

SERIAL DATA. An open-collector signal that is used for VMEbus data transmission. This signal is not implemented on the HK68/V3F.

SYSCLK

SYSTEM CLOCK. A totem-pole signal that provides a constant 16-MHz clock signal that is independent of any other bus timing. This signal is driven if the HK68/V3F is a system controller.

SYSFAIL*

SYSTEM FAIL. An open-collector signal that indicates 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 VIC drives this signal low at power-up and may be programmed to generate an interrupt if asserted by another board in the system. Details are given in section 7.9.

SYSRESET*

SYSTEM RESET. An open-collector signal that, when asserted, causes the system to be reset.

WRITE*

WRITE. A three-state 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. This signal is not used on the HK68/V3F.

7.3 VIC REGISTER MAP

The base address of the VIC chip is $FF00,0000_{16}$. The following table shows the VIC register offsets from the base. Please refer to the *VIC068 VMEbus Interface Controller Specification* from VTC for detailed descriptions of the registers.

The registers are byte wide.

TABLE 7-1
VIC register map

Offset Address	Acronym	Register Name
3	VIICR	VMEbus Interrupter Interrupt Control Register
7	VICR1	VMEbus Interrupter Control Register 1
B	VICR2	VMEbus Interrupter Control Register 2
F	VICR3	VMEbus Interrupter Control Register 3
13	VICR4	VMEbus Interrupter Control Register 4
17	VICR5	VMEbus Interrupter Control Register 5
1B	VICR6	VMEbus Interrupter Control Register 6
1F	VICR7	VMEbus Interrupter Control Register 7
23	DSICR	DMA Status Interrupt Control Register
27	LICR1	Local Interrupt Control Register 1
2B	LICR2	Local Interrupt Control Register 2
2F	LICR3	Local Interrupt Control Register 3
33	LICR4	Local Interrupt Control Register 4
37	LICR5	Local Interrupt Control Register 5
3B	LICR6	Local Interrupt Control Register 6
3F	LICR7	Local Interrupt Control Register 7
43	ICGSICR	ICGS Interrupt Control Register
47	ICMSICR	ICMS Interrupt Control Register
4B	EGICR	Error Group Interrupt Control Register
4F	ICGSIVBR	ICGS Interrupt Vector Base Register
53	ICMSIVBR	ICMS Interrupt Vector Base Register
57	LIVBR	Local Interrupt Vector Base Register
5B	EGIVBR	Error Group Interrupt Vector Base Register
5F	ICSR	Interprocessor Communications Switch Register
63	ICR0	Interprocessor Communications Register 0
67	ICR1	Interprocessor Communications Register 1
6B	ICR2	Interprocessor Communications Register 2
6F	ICR3	Interprocessor Communications Register 3
Continues.		

73	ICR4	Interprocessor Communications Register 4
77	ICR5	Interprocessor Communications Register 5
7B	ICR6	Interprocessor Communications Register 6
7F	ICR7	Interprocessor Communications Register 7
83	VIRSR	VMEbus Interrupt Request and Status Register
87	VIVR1	VMEbus Interrupt Vector Register 1
8B	VIVR2	VMEbus Interrupt Vector Register 2
8F	VIVR3	VMEbus Interrupt Vector Register 3
93	VIVR4	VMEbus Interrupt Vector Register 4
97	VIVR5	VMEbus Interrupt Vector Register 5
9B	VIVR6	VMEbus Interrupt Vector Register 6
9F	VIVR7	VMEbus Interrupt Vector Register 7
A3	TTR	Transfer Timeout Register
A7	LBTR	Local Bus Timing Register
AB	BTDR	Block Transfer Definition Register
AF	VICR1	VMEbus Interface Configuration Register 1
B3	ARCR	Arbiter and Requester Configuration Register
B7	AMSR	Address Modifier Source Register
BB	BESR	Bus Error Status Register
BF	DMASR	DMA Status Register
C3	SS0CR0	Slave Select 0 Control Register 0
C7	SS0CR1	Slave Select 0 Control Register 1
CB	SS1CR0	Slave Select 1 Control Register 0
CF	SS1CR1	Slave Select 1 Control Register 1
D3	RCR	Release Control Register
D7	BTCR	Block Transfer Control Register
DB	BTLR0	Block Transfer Length Register 0
DF	BTLR1	Block Transfer Length Register 1
E3	SYSRR	System Reset Register
E7		Undefined
EB		Undefined
EF		Undefined
F3		Undefined
F7		Undefined
FB		Undefined
FF		Undefined

7.4 VMEbus INTERRUPTS

VMEbus interrupt generation and handling capability is provided by the VIC chip. The following features are included:

- Conformance to the Motorola VMEbus specification revision C.1
- The capability to interrupt other boards on the VMEbus using any of the seven VMEbus interrupt levels
- The capability to generate interrupts on multiple levels at the same time
- The capability to intercept VMEbus, VIC, and on-board interrupts and provide an interrupt to the MPU
- Capability to provide vectors for VIC and local interrupts
- A timer interrupt

The seven VMEbus interrupts are monitored and controlled by the VIC chip. An interrupt to the MC68030 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.

For details on the VIC processor, read the *VIC068 VMEbus Interface Controller Specification* by VTC Incorporated. A summary of the interrupt architecture on the HK68/V3F is shown in Figure 7-1.

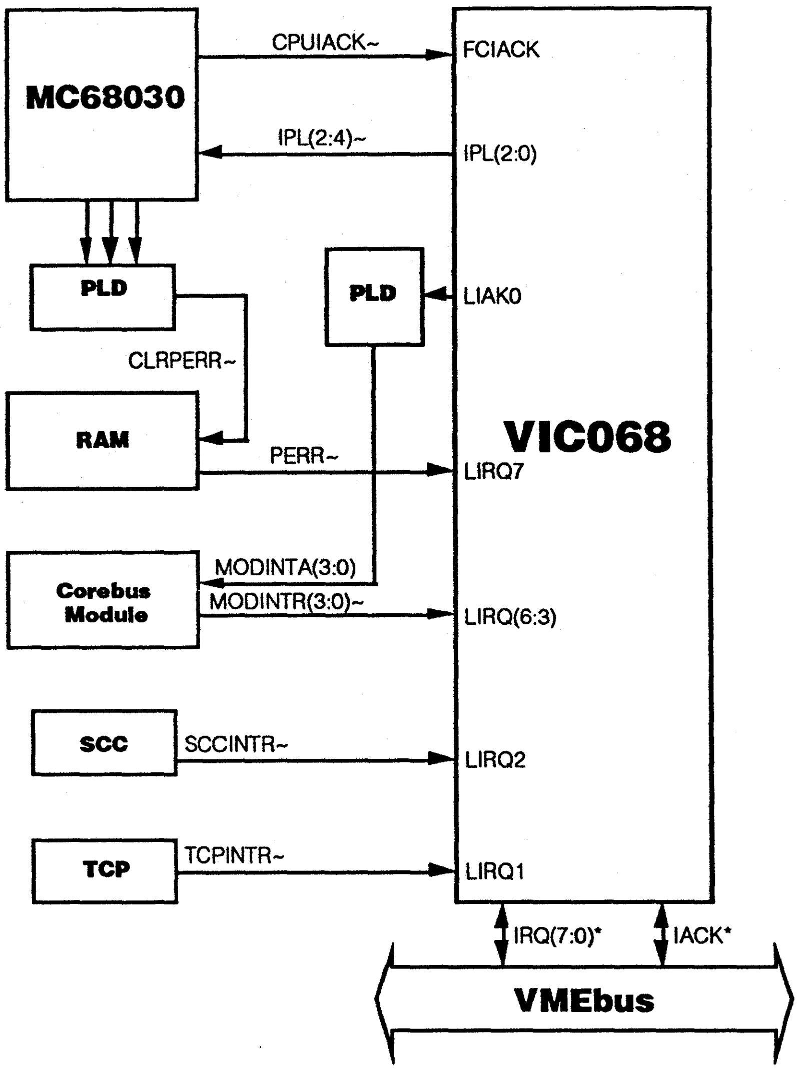


FIGURE 7-1. HK68/V3F interrupt architecture

7.4.1 Interrupter Operation

The VIC may assert interrupt requests on the VMEbus at all of the seven interrupt levels. It may generate interrupt requests on multiple levels simultaneously.

Interrupt generation is programmed through the VMEbus interrupt request/status register (VIRSR) of the VIC processor. This register allows each interrupt to be set and reset by writing a 1 or a 0 to the corresponding bit in the register.

The VIC068 also includes seven VMEbus interrupt vector registers (VIVR1-VIVR7) that must be initialized before the interrupt is turned on. When a VMEbus interrupt is acknowledged, an internal interrupt can be generated to complete the handshake without polling the VMEbus interrupt request and status register (VIRSR) for the acknowledge of an interrupt. The local interrupt for acknowledges is programmed using both the VMEbus interrupter interrupt control register (VIICR) and the error group interrupt vector base register (EGIVBR).

7.4.2 Interrupt Handler Operation

The VIC controller handles all VMEbus interrupts (IRQ1*-IRQ7*) and all local interrupts.

7.4.2.1 VIC Interrupt Requests

VIC interrupts are presented to the MC68030 on three lines IPL0-IPL2. The VIC chip can be programmed to present one of seven priority levels on the IPL lines. The MC68030 interrupt controller treats these lines as dedicated interrupt requests.

7.4.2.2 VIC Interrupt Acknowledges

The VIC provides a vector for all interrupt levels except LIRQ(6:3) (module interrupts). The VIC indicates an interrupt condition to the processor on IPL2, IPL1 and IPL0. Performing an interrupt acknowledge cycle causes an 8-bit vector to be read and the interrupt to be removed. An attempt to read a vector when no interrupt is present results in a bus error.

The VIC can be programmed to generate interrupts to the MC68030 for the following sources:

- **Error Group Interrupts:** Refer to the VIC error group control register (EGICR) and the error group interrupt vector base register (EGIVBR).

  ACFAIL: If a power failure module is installed on the VMEbus backplane, the VIC may be programmed to generate an interrupt if a power failure occurs (that is, VMEbus ACFAIL* asserted).

  SYSFAIL: The VIC may be programmed to generate an interrupt when a system failure is indicated (that is, VMEbus SYSFAIL* asserted).

  Arbitration timeout: When the VIC times out on arbitration, the VIC can be programmed to generate an interrupt.

Write posted cycle failure: If a write cycle that was posted by the processor fails, the processor is notified by this interrupt.

- **Local Interrupts (LIRQ7-LIRQ0)**: The VIC can be programmed to generate interrupts for Ring Detection on SCC Port A, Corebus modules, and the TCP. Refer to the VIC local interrupt control registers (LICR1-LICR7) and local interrupt vector base register (LIVBR).

- **ICGS Group Interrupts**: The interprocessor communications global switches (ICGS) allow other VMEbus boards to interrupt the HK68/V3F for global events. Refer to the VIC ICGS interrupt control register (ICGSICR) and ICGS interrupt vector base register (ICGSIVBR).

- **ICMS Group Interrupts**: The interprocessor communications module switches (ICMS) allow other VMEbus boards to interrupt the HK68/V3F for HK68/V3F-specific events. Refer to the VIC ICMS interrupt control register (ICMSICR) and ICMS interrupt vector base register (ICMSIVBR).

- **VMEbus Interrupts (IRQ7-IRQ0)**: The VIC can be programmed to receive and generate the seven VMEbus interrupts with status/id (vector) information. Refer to the VIC VMEbus interrupt control registers (VICR1-VICR7) and VMEbus interrupt vector registers (VIVR1-VIVR7).

- **DMA Status/Complete Interrupt**: If this interrupt is enabled, the VIC generates an interrupt if either the DMA completes, or a BERR occurs (local *or* VME) during the DMA transfer. Refer to the DMA status interrupt control register (DSICR) and error group interrupt vector base register (EGIVBR).

- **VME interrupter handshake**: When a VMEbus interrupt generated by the HK68/V3F is acknowledged, this interrupt can be used to indicate the acknowledge has taken place. Refer to the VMEbus interrupter interrupt control register (VIICR) and the error group interrupt vector base register (EGIVBR).

Interrupts are internally prioritized, as shown in the following table.

TABLE 7-2
Interrupt priorities

Rank	Interrupt
19	LIRQ7 Parity Error (highest priority)
18	Error Group Interrupt
17	LIRQ6 (Corebus MODINTR3)
16	LIRQ5 (Corebus MODINTR2)
15	LIRQ4 (Corebus MODINTR1)
14	LIRQ3 (Corebus MODINTR0)
13	LIRQ2 (SCC Interrupt)
12	LIRQ1 (TCP Interrupt))
11	ICGS Group Interrupt
10	ICMS Group Interrupt
9	IRQ7 (VME Interrupt Request Level 7)
8	IRQ6 (VME Interrupt Request Level 6)
7	IRQ5 (VME Interrupt Request Level 5)
6	IRQ4 (VME Interrupt Request Level 4)
5	IRQ3 (VME Interrupt Request Level 3)
4	IRQ2 (VME Interrupt Request Level 2)
3	IRQ1 (VME Interrupt Request Level 1)
2	DMA Status/Complete Interrupt
1	VME Interrupt Acknowledged

Vector base registers are provided for each of the following groups of interrupts:

- ICGS
- ICMS
- Local interrupts
- Error interrupts

If the interrupt source is a VME interrupt, then the VIC latches the status/id (vector) onto the local bus during the local IACK cycle.

7.5 MAILBOX INTERFACE

Interprocessor communication (also known as mailbox) is provided by the VMEbus Interface Controller (VIC) processor. This section provides a brief description of the interprocessor communications facilities of the HK68/V3F. For a detailed description, read the VIC068 specification.

The mailbox interface consists of a collection of 8-bit registers and memory locations that can be used for communications with the HK68/V3F through the VMEbus. A description of the VIC registers follows:

VIC REGISTER	DESCRIPTION
ICR4-ICR0	Five general-purpose, dual-port "Interprocessor Communications" registers, which can be accessed from the VMEbus and from the HK68/V3F local bus. These registers are 8 bits wide and each has an associated semaphore bit in ICR7.
ICR5-ICR7	Four global switch registers.
ICR5	ICR5 is a VIC-specific register that specifies the revision level of the VIC.
ICR6	ICR6 is read only from the VMEbus and provides the status of the HK68/V3F.
ICR7	ICR7 is a dual-port register accessible from the VMEbus and the HK68/V3F local bus. This register provides semaphore bits for ICR0-ICR4, status of the HK68/V3F, and a means for remote resetting of the HK68/V3F.
ICGS0-ICGS3	Four interboard communications "global switch" registers, which are used to generate interrupts for global events.
ICMS0-ICMS3	Four interboard communications "module switch" registers, which are used to generate interrupts for HK68/V3F-specific events.

The local bus register addresses are described in Table 7-1. The VMEbus mailbox structure is described in Figure 7-2.

Mailbox Base Address →		A1 - A0 = 00	A1 - A0 = 01	A1 - A0 = 10	A1 - A0 = 11
Supervisory or user accesses AM5 - AM0 = 29_{16} or $2D_{16}$	0_{16}	Undefined	ICR0	Undefined	ICR1
	4_{16}		ICR2		ICR3
	8_{16}		ICR4		ICR5
	C_{16}		ICR6		ICR7
Supervisory access only AM5 - AM0 = $2D_{16}$	10_{16}	ICGS0 Set	ICGS0 Clear	ICGS1 Set	ICGS1 Clear
	14_{16}	ICGS2 Set	ICGS2 Clear	ICGS3 Set	ICGS3 Clear
Supervisory or user accesses AM5 - AM0 = 29_{16} or $2D_{16}$	18_{16}	Undefined			
	$1C_{16}$				
	20_{16}	ICMS0 Set	ICMS0 Clear	ICMS1 Set	ICMS1 Clear
	24_{16}	ICMS2 Set	ICMS2 Clear	ICMS3 Set	ICMS3 Clear
	28_{16}	Undefined			
	$2C_{16}$				
	30_{16}				

FIGURE 7-2. VME mailbox structure

All accesses are defined as 8-bit, and accesses to undefined areas may result in a bus error. An access from the VMEbus to the appropriate address in the VMEbus short space results in the VIC's responding (as a slave) to the access.

The VMEbus mailbox can be mapped to any of 256 256-byte boundaries within the VMEbus *short* addressing space. All registers are accessible from the supervisory short space (AM5 - AM0 = $2D_{16}$) and all but the global switches are accessible from the user short space (AM5 - AM0 = 29_{16}).

The mailbox base address is an 8-bit value stored in a latch that is compared to address lines A15-A8 on a VMEbus short space access. The mailbox base is modified by writing to $FF01,2010_{16}$.

7.6 VMEbus SYSTEM CONTROLLER

Nearly all VMEbus operations of the HK68/V3F are handled by the VMEbus Interface Controller processor (VIC068). The VIC processor can be jumpered to provide the VMEbus system controller functions via jumper J5.

As the system controller, the VIC drives Sysclk (SYSCLK), Bus Clear (BCLR), and System Reset (SYSRESET). The system controller also provides the system bus arbitration in one of three modes: "prioritized," "round robin," and "single level" arbitration.

See the VIC arbiter and requester configuration register (ARCR). If configured as the system controller, the VIC also monitors the VMEbus interface as a watchdog timer (with a programmable time-out; see section 7.10).

The VIC processor is configured as the system controller by installing jumper J5. When the HK68/V3F is configured as the system controller, it *must* be installed in slot 1 with a programmable time-out.

TABLE 7-3
Bus control jumpers

Jumper	Function
J5	System Controller Enable — When this jumper is installed, the HK68/V3F acts as a VMEbus system controller as described in the VIC User's Manual.

NOTE: Only one board in a VME system should be system controller.

7.7 VMEbus MASTER INTERFACE

The HK68/V3F can access the VMEbus with any of the three address modes "short," "standard," and "extended" on any of the four bus request levels. Refer to the VIC registers — arbiter and requester configuration register (ARCR) and the address modifier source register (AMSR). Short addresses use 16 address lines to specify a target address. Standard addresses use 24 address lines, and extended addresses use all 32 address lines. Table 7-4 shows the relationship between the on-card physical address and the corresponding VMEbus region.

TABLE 7-4
Relationship of physical address to VMEbus and VSB memory region

On-card Addresses (hex)	Memory Region
FF04,0000 – FF04,FFFF	VMEbus short address (0000 – FFFF)
FE00,0000 – FEFF,FFFF	VMEbus standard address (00,0000 – FF,FFFF)
4000,0000 – FBFF,FFFF	VMEbus extended address (4000,0000 - FBFF,FFFF)

Extended VME addresses from $0000{,}0000_{16}$ to $4000{,}0000_{16}$ and from $\mathrm{FC00{,}0000}_{16}$ to $\mathrm{FFFF{,}FFFF}_{16}$ are not accessible.

The VMEbus master release modes are programmed by writing to the RCR (release control register) of the VIC chip. If the HK68/V3F is the bus master when the requested bus operation is completed, the bus will be released according to the state contained in the RCR register. The release mode is either:

ROR — Release-on-request will release the VMEbus (BBSY*) when a request is detected and there are no HK68/V3F bus requests.

RWD — Release-when-done will release the bus when there are no further HK68/V3F bus requests.

ROC — Release-on-clear will retain the bus until BCLR* has been asserted by the system controller.

BCAP — Bus-capture-and-hold (never release) can be used to capture the bus. BCAP will retain the bus until a different release mode is programmed.

7.8 VMEbus SLAVE INTERFACE

Three registers external to the VIC, as well as VIC registers, are used for slave accesses (Table 7-5).

TABLE 7-5
Registers used for slave accesses

Address (hex)	Register	Description
FF01,2000	Local address register	The address driven on local address lines 31-24 during slave accesses.
FF01,2008	VMEbus base address register	Base address of the HK68/V3F on the VMEbus. The address driven on local address lines 23-21 during standard space slave access.
FF01,2010	Mailbox base	Base address of mailbox on VMEbus.

The HK68/V3F can be accessed from the VMEbus in both "extended" and "standard" space. "Short" space is used for the mailbox only (Fig. 7-3). The local address is driven on local A(31:24) during VME slave cycles. All registers shown in Figure 7-3 are long-word registers (32 bits wide).

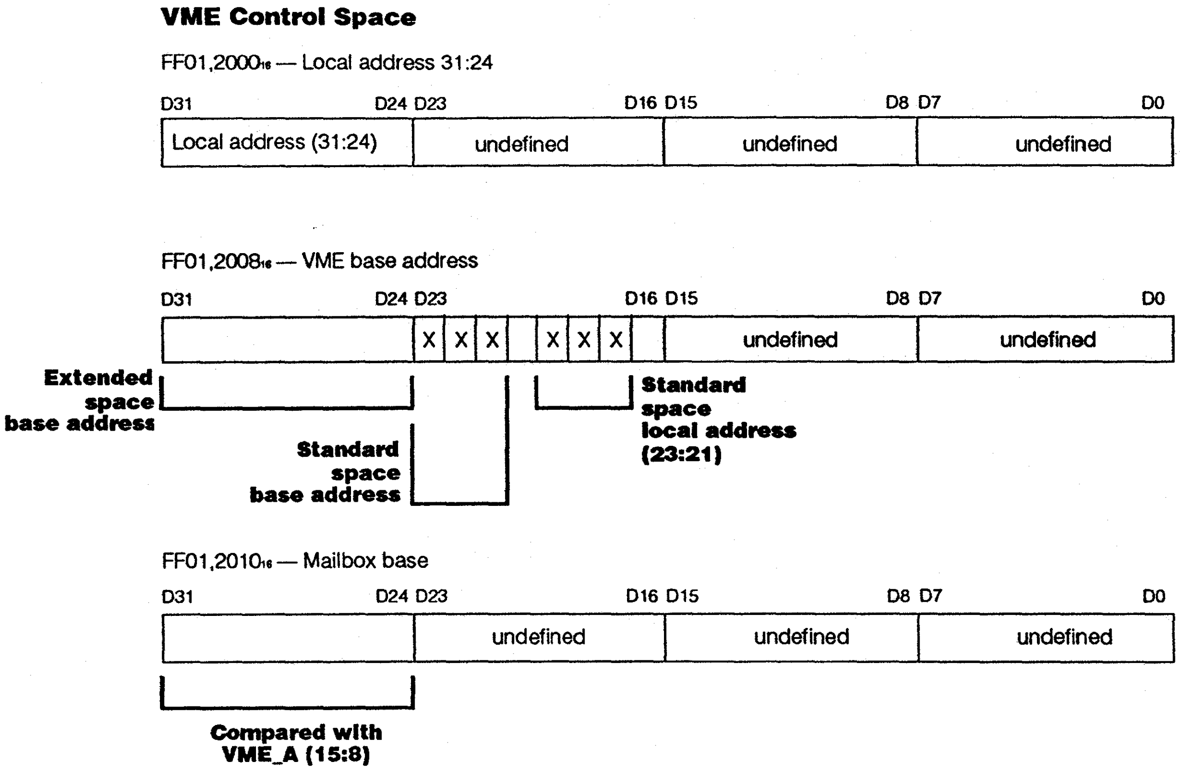

VME Control Space
FF01,2000$_{16}$ — Local address 31:24
D31
D24 D23
D16 D15
D8 D7
D0
Local address (31:24)
undefined
undefined
undefined
FF01,2008$_{16}$ — VME base address
D31
D24 D23
D16 D15
D8 D7
D0
X X X
X X X
undefined
undefined
Extended space base address
Standard space base address
Standard space local address (23:21)
FF01,2010$_{16}$ — Mailbox base
D31
D24 D23
D16 D15
D8 D7
D0
undefined
undefined
undefined
Compared with VME_A (15:8)

FIGURE 7-3. VME control space during slave cycles

7.8.1 Extended Space

For the HK68/V3F to respond to a VMEbus extended address, the following steps *must* be taken:

1. The VIC register SS1CR0 (slave select one, control register 0) must be configured to respond to A32/D32 types of cycles (bits 2, 3, and 4 must be set to 100$_2$).

2. The extended space compare address must be written to the register at FF01,2008$_{16}$.

3. The on-card base address register (FF01,2000$_{16}$) must be set up with the base address of RAM (0000,0000$_{16}$).

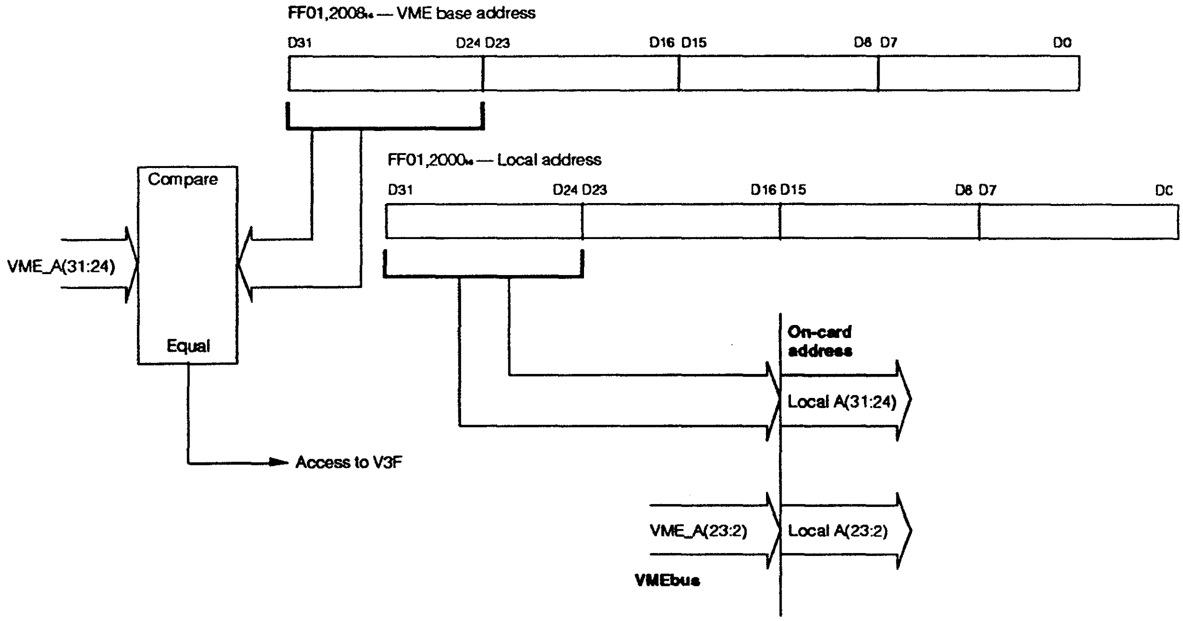


FIGURE 7-4. Extended space slave access

The slave extended space compare address can map the HK68/V3F internal RAM to one of 256 16-Mbyte boundaries. The compare address is stored in the register and is compared with the VMEbus address lines A31-A24 (Fig. 7-4). The slave compare address is modified by writing to FF01,2008$_{16}$. When the HK68/V3F is selected as a slave in the extended space, all on-card RAM is mapped to the bus, starting at the base of the 16-Mbyte region that corresponds to the slave compare address.

7.8.2 Standard Space

For the HK68/V3F to respond to a VMEbus standard address, the following steps *must be taken:*

1. VIC register SSOCRO (slave select 0, control register 0) must be configured to respond to A24/D32 types of cycles (bits 2, 3, and 4 must be set to 101$_2$).

2. The standard space compare address must be written to the register at FF01,2008$_{16}$.

3. The on-card base address register (FF01,2000$_{16}$) must be set up with the base address of RAM (0000,0000$_{16}$).

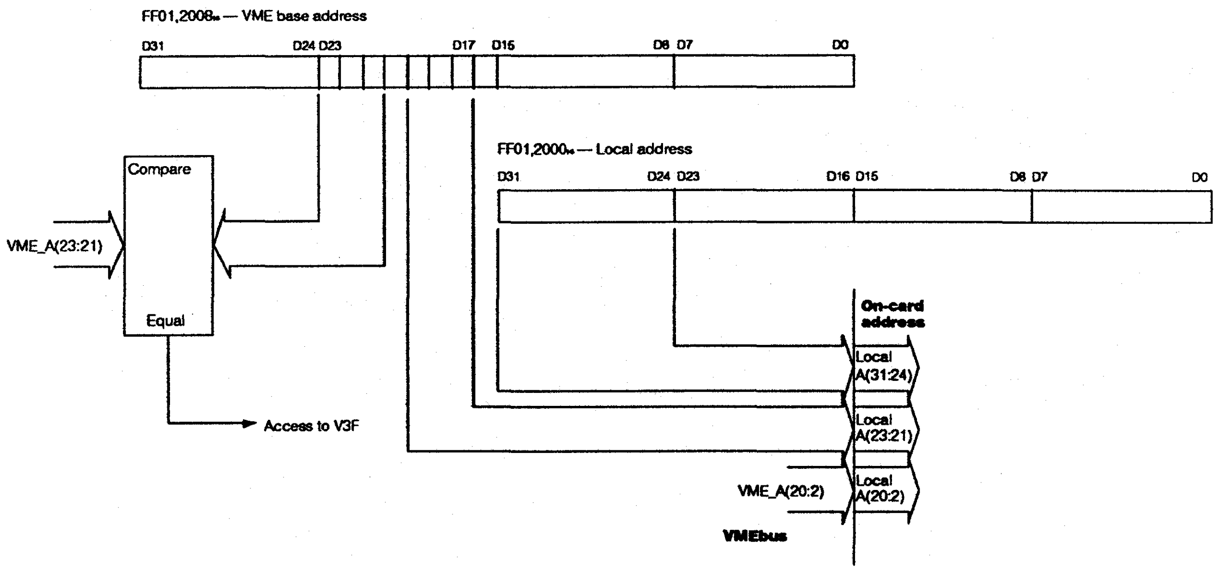


FIGURE 7-5. Standard space slave access

The slave standard space compare address can map 2 Mbytes of the MC68030's RAM to one of eight 2-Mbyte boundaries. The compare address for the standard space is stored in the register at $FF01,2008_{16}$ and is compared with VMEbus address lines A23-A21 (Fig. 7-5). When the HK68/V3F is selected as a slave in the standard space, 2 Mbytes of internal RAM are mapped to the bus, as described in Tables 7-6 and 7-7:

TABLE 7-6
HK68/V3F "standard" space slave mapping on VMEbus, 2-Mbyte HK68/V3F

VME Base Address Register Bits 23:21	VMEbus Base Address	Local Address Register Bits 31:21	Memory Mapped to VMEbus
000_2	$XX00,0000_{16}$	$0000,0000,000_2$	$0000,0000 - 001F,FFFF_{16}$
001_2	$XX20,0000_{16}$	$0000,0000,000_2$	$0000,0000 - 001F,FFFF_{16}$
010_2	$XX40,0000_{16}$	$0000,0000,000_2$	$0000,0000 - 001F,FFFF_{16}$
011_2	$XX60,0000_{16}$	$0000,0000,000_2$	$0000,0000 - 001F,FFFF_{16}$
100_2	$XX80,0000_{16}$	$0000,0000,000_2$	$0000,0000 - 001F,FFFF_{16}$
101_2	$XXA0,0000_{16}$	$0000,0000,000_2$	$0000,0000 - 001F,FFFF_{16}$
110_2	$XXC0,0000_{16}$	$0000,0000,000_2$	$0000,0000 - 001F,FFFF_{16}$
111_2	$XXE0,0000_{16}$	$0000,0000,000_2$	$0000,0000 - 001F,FFFF_{16}$

TABLE 7-7
HK68/V3F "standard" space slave mapping on VMEbus, 8-Mbyte HK68/V3F

VME Base Address Register Bits 23:21	VMEbus Base Address	Local Address Register Bits 31:21	Memory Mapped to VMEbus
000_2	XX00,0000$_{16}$	0000,0000,000$_2$	1st two MB (0000,0000 – 001F,FFFF$_{16}$)
		0000,0000,001$_2$	2nd two MB (0020,0000 – 003F,FFFF$_{16}$)
		0000,0000,010$_2$	3rd two MB (0040,0000 – 005F,FFFF$_{16}$)
		0000,0000,011$_2$	4th two MB (0060,0000 – 007F,FFFF$_{16}$)
001_2	XX20,0000$_{16}$	0000,0000,000$_2$	1st two MB (0000,0000 – 001F,FFFF$_{16}$)
		0000,0000,001$_2$	2nd two MB (0020,0000 – 003F,FFFF$_{16}$)
		0000,0000,010$_2$	3rd two MB (0040,0000 – 005F,FFFF$_{16}$)
		0000,0000,011$_2$	4th two MB (0060,0000 – 007F,FFFF$_{16}$)
010_2	XX40,0000$_{16}$	0000,0000,000$_2$	1st two MB (0000,0000 – 001F,FFFF$_{16}$)
		0000,0000,001$_2$	2nd two MB (0020,0000 – 003F,FFFF$_{16}$)
		0000,0000,010$_2$	3rd two MB (0040,0000 – 005F,FFFF$_{16}$)
		0000,0000,011$_2$	4th two MB (0060,0000 – 007F,FFFF$_{16}$)
011_2	XX60,0000$_{16}$	0000,0000,000$_2$	1st two MB (0000,0000 – 001F,FFFF$_{16}$)
		0000,0000,001$_2$	2nd two MB (0020,0000 – 003F,FFFF$_{16}$)
		0000,0000,010$_2$	3rd two MB (0040,0000 – 005F,FFFF$_{16}$)
		0000,0000,011$_2$	4th two MB (0060,0000 – 007F,FFFF$_{16}$)
100_2	XX80,0000$_{16}$	0000,0000,000$_2$	1st two MB (0000,0000 – 001F,FFFF$_{16}$)
		0000,0000,001$_2$	2nd two MB (0020,0000 – 003F,FFFF$_{16}$)
		0000,0000,010$_2$	3rd two MB (0040,0000 – 005F,FFFF$_{16}$)
		0000,0000,011$_2$	4th two MB (0060,0000 – 007F,FFFF$_{16}$)
101_2	XXA0,0000$_{16}$	0000,0000,000$_2$	1st two MB (0000,0000 – 001F,FFFF$_{16}$)
		0000,0000,001$_2$	2nd two MB (0020,0000 – 003F,FFFF$_{16}$)
		0000,0000,010$_2$	3rd two MB (0040,0000 – 005F,FFFF$_{16}$)
		0000,0000,011$_2$	4th two MB (0060,0000 – 007F,FFFF$_{16}$)
110_2	XXC0,0000$_{16}$	0000,0000,000$_2$	1st two MB (0000,0000 – 001F,FFFF$_{16}$)
		0000,0000,001$_2$	2nd two MB (0020,0000 – 003F,FFFF$_{16}$)
		0000,0000,010$_2$	3rd two MB (0040,0000 – 005F,FFFF$_{16}$)
		0000,0000,011$_2$	4th two MB (0060,0000 – 007F,FFFF$_{16}$)
111_2	XXE0,0000$_{16}$	0000,0000,000$_2$	1st two MB (0000,0000 – 001F,FFFF$_{16}$)
		0000,0000,001$_2$	2nd two MB (0020,0000 – 003F,FFFF$_{16}$)
		0000,0000,010$_2$	3rd two MB (0040,0000 – 005F,FFFF$_{16}$)
		0000,0000,011$_2$	4th two MB (0060,0000 – 007F,FFFF$_{16}$)

7.8.3 Short Space

Refer to section 7.5 ("Mailbox Interface") for information on short space.

7.9 SYSFAIL CONTROL

The SYSFAIL line is controlled by the VIC processor. It is both an input and an open-collector output.

As an *output,* the VIC asserts it after power-up and remains asserted until self-tests and diagnostics are complete. It can then be removed (or set) by setting control bits in Interprocessor Communication Registers 6 and 7 (that is, ICR6 and ICR7). See the VIC manual for further details. At power-up, all other boards in the system will also do the same, that is, they will assert SYSFAIL until their diagnostics are complete. Once all boards are initialized, SYSFAIL should not be asserted by any board, except to indicate a failure of some kind.

As an *input,* the SYSFAIL line is used to indicate a system failure. If the VIC detects SYSFAIL asserted and if the VIC "Error Group Interrupt" is enabled, and the Error Group Interrupt Control Register (EGICR) has the SYSFAIL interrupt enabled, then the processor will be interrupted, and the Error Group Interrupt Vector Base Register (EGIVBR) will indicate a SYSFAIL interrupt.

7.10 WATCHDOG AND BUS TIMER

All local accesses and accesses to the VMEbus are monitored by the VIC chip. The VIC chip has two timers; one for the local bus (default is 32 microseconds) and one for the VMEbus (default is 64 microseconds). These values may be changed via the VIC Transfer Timeout Register (TTR). The local timer defaults to being on, while the VMEbus timer is only on if the VIC is configured as the system controller.

If the *VMEbus* timer expires, BERR is asserted (on-card) to the MC68030 and BERR* is asserted on VMEbus.

If the *local* timer expires, BERR is asserted (on-card only) to the MC68030.

CAUTION: If the timer values are changed to "infinite," the corresponding bus will hang indefinitely if a nonexistent or nonresponding location is accessed.

7.11 VMEBUS INTERFACE

The VMEbus interface uses P1 and P2. P1 is used for most of the VME address, data, and control lines. P2 is used for the *extended* VME address and data lines, and all of the VSB lines. Not all of the P1 signals are used on the HK68/V3F.

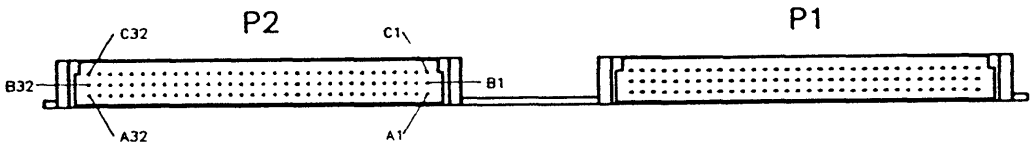


FIGURE 7-6. VMEbus and VME subsystem bus connectors, P1 and P2

7.12 VMEBUS PIN ASSIGNMENTS (P1)

TABLE 7-8
VMEbus connector pin assignments (P1)

Signal	Row A P1	Row B	Row C
Mnemonic	Pin Number	Signal	Signal
	Mnemonic	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
Continues.			

TABLE 7-8 — *Continued.*

VMEbus connector pin assignments (P1)

Signal Mnemonic	Row A P1 Pin Number Mnemonic	Row B Signal Mnemonic	Row C Signal
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

7.13 VMEBUS/VSB PIN ASSIGNMENTS (P2)

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

TABLE 7-9
VMEbus/VSB pin assignments (P2)

VMEbus Signal Number	Row A VSB Signal Mnemonic	Row B VSB Signal Mnemonic	Row C P2 Pin
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*

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

8

VME Subsystem Bus (VSB) Control

8.1 INTRODUCTION

The HK68/V3F supports the VME subsystem bus (VSB) expansion interface, which allows high-speed 8-, 16-, 24-, 32-bit data transfers without the need for the VME bus.

8.2 BUS CONTROL SIGNALS, VSB P2 DESCRIPTIONS

The VSB P2 pin descriptions are given in the following section. Refer to the Motorola VMEbus specification, revision C.1, for detailed use of these signals. All signals are bidirectional unless otherwise stated. Refer to section 7 for a complete listing of the pin assignments.

AD00-31 MULTIPLEXED ADDRESS/DATA LINES. The three-state multiplexed address/data path (32 lines) that is controlled by the three-state drivers on the master and slave devices.

PAS* VSB ADDRESS STROBE. A three-state line in which the falling edge indicates that a valid address is present on AD31-AD00.

SPACE0-SPACE1 VSB ADDRESS SPACE SELECT. Three-state signals that select one of four address spaces or signify an interrupt acknowledge or parallel arbitration cycle. On the HK68/V3F, these signals are not used; they are driven high when the HK68/V3F is the VSB master, which selects the System Address Space.

DS* VSB DATA STROBE. Three-state signal whose falling edge indicates a transfer will occur over AD31-AD00. During write cycles, write data are valid at the falling edge of DS*.

WR* VSB WRITE. WR*. A three-state signal used to indicate a *read* or *write* operation. When the signal is low, the operation is a write. When the signal is high, the operation is a read.

SIZE0,SIZE1 VSB BUS SIZE. Three-state lines that, in conjunction with addresses AD00 and AD01, determine the data transfer size and position on the data bus.

LOCK* VSB BUS LOCK. When asserted, this line indicates that the bus is locked and that no other master can obtain possession of the bus. This allows for indivisible cycles, such as Read-Modify-Write cycles, to occur from the VSB to a dual-ported resource.

ASACK0* and ASACK1* VSB ADDRESS/SIZE ACKNOWLEDGE. Open-collector lines that serve two functions:

(1) The responding SLAVE drives its size code on these lines, and (2) The responding slave drives at least one of these lines to inform the HK68/V3F to switch the multiplexed address/data bus from address to data.

WAIT* WAIT. An open-collector line that 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.

AC VSB DECODE COMPLETE. An open-collector line that 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 whether the device is selected by the current address on the bus.

CACHE* VSB CACHEABLE. An open-collector signal that, when asserted, 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/V3F.

ACK* VSB DATA TRANSFER ACKNOWLEDGE. An open-collector line that is asserted by the selected slave module to complete the handshake for a transfer operation.

ERR* VSB DATA ERROR. An open-collector line that is asserted 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. An open-collector line that, when asserted, indicates that a master or slave device is attempting to interrupt another master. On the HK68/V3F, this signal is not used.

BREQ* VSB BUS REQUEST. An open-collector line that is asserted by a requester whenever bus mastership is required.

BGIN* VSB BUS GRANT IN. A totem-pole line that, as an input to the HK68/V3F, indicates that it has been granted the bus. BGIN and BGOUT form a bus grant daisy-chain.

BGOUT* VSB BUS GRANT OUT. A totem-pole line that, as an output from the HK68/V3F, indicates to the next board in the daisy chain that it may use the bus.

BUSY* VSB BUS BUSY. An open-collector line that is asserted 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/V3F; the geographical addressing feature is not implemented.

8.3 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/V3F operates on the VSB in master, secondary, or slave mode. 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.

8.3.1 VSB OPERATION

VSB is accessible from the top of RAM (0020,0000$_{16}$ or 0080,0000$_{16}$) to 4000,0000$_{16}$.

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 are several control bits that affect the operation of the VSB interface.

8.3.1.1 Release Modes

The HK68/V3F supports four VSB release modes: (1) release with hold (RWH); (2) release on request (ROR); (3) release between every access (RWD), and (4) never release (RNV). The HK68/V3F powers up in RWH state. Two bits, one at FF01,3080$_{16}$ and one at FF01,3088$_{16}$, control VSB release mode.

TABLE 8-1
VSB release modes

Size: Byte. Type: Write.		
Port address FF01,3088$_{16}$	Port address FF01,3080$_{16}$	
D0	**D0**	**Function**
0	0	Release with hold
0	1	Release on request
1	0	Release between every access
1	1	Never release

8.3.1.2 Arbiter Enable

The "first" VSB master board — the primary master, should be the arbiter. Other VSB masters should not be the arbiter. The arbiter indicates the beginning of the VSB arbitration daisy chain. The VSB arbiter-enable bit *must be set true* if the HK68/V3F is the "first" board, that is, the arbiter. A one-bit latch at address FF01,3098$_{16}$ controls the VSB arbiter enable. The HK68/V3F powers up with the VSB arbiter not enabled.

TABLE 8-2
VSB arbiter enable

Port address: FF01,3098$_{16}$. Size: Byte. Type: Write.	
D0	**Function**
0	Not enabled: HK68/V3F is not arbiter.
1	Enabled: HK68/V3F is arbiter.

8.3.1.3 Slave Enable

The HK68/V3F supports VSB slave operation. Initially, the slave is disabled and the HK68/V3F will not allow VSB cycles on card. Prior to enabling slave operation, two registers must be initialized: the VSB compare address register and VSB local address register (see Figure 8-1).

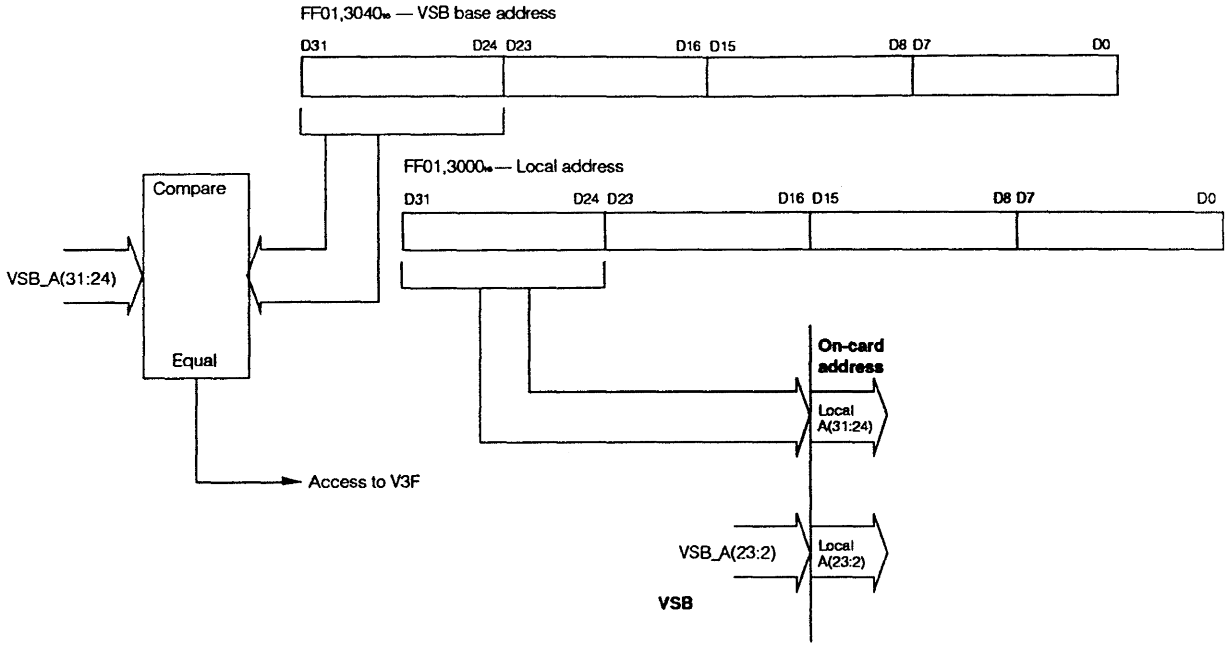


FIGURE 8-1. Extended space slave access

Figure 8-1 shows how VSB addresses are mapped to local addresses. Any 16-Mbyte-aligned 16-Mbyte block in the HK68/V3F memory map can be mapped onto any 16-Mbyte-aligned 16-Mbyte block in the VSB system address space. The 16-Mbyte block of system address space the Hk68/V3F maps into is determined by the long-word VSB compare address register at address FF01,3040$_{16}$. To map the HK68/V3F to any 16-Mbyte block in VSB system address space, simply write any address in that block to the VSB compare address register at FF01,3040$_{16}$.

The address presented to the local bus on A31 through A24 during slave cycles is held in the VSB local address register at address FF01,3000$_{16}$. Using this register, RAM, ROM, module, or any 16-Mbyte on-card space can be made accessible to the bus on slave cycles (Fig. 8-2).

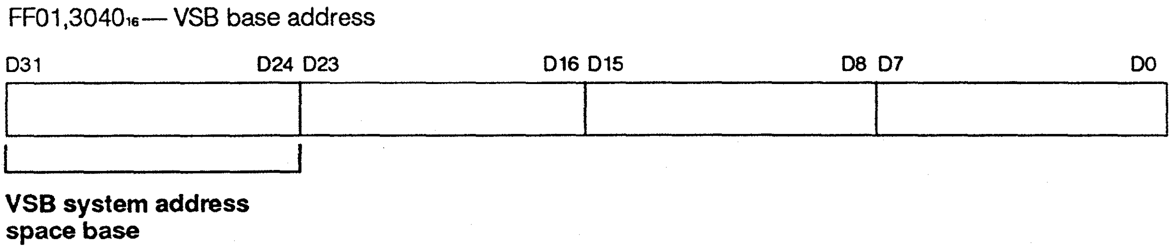

FF01,3040$_{16}$ — VSB base address
D31
D24 D23
D16 D15
D8 D7
D0
VSB system address space base

FIGURE 8-2. VSB system address space base

To enable the slave interface requires setting a bit in the VSB slave enable register at FF01,3090$_{16}$ (Fig. 8-3).

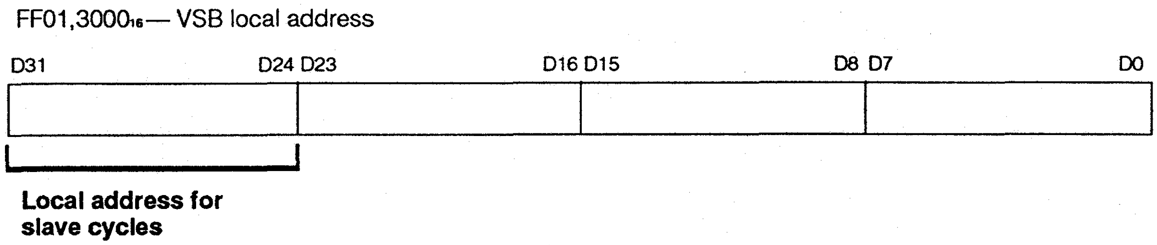

FF01,3000$_{16}$— VSB local address
D31
D24 D23
D16 D15
D8 D7
D0
Local address for slave cycles

FIGURE 8-3. Local address for slave cycles

TABLE 8-3
VSB slave enable

Port address FF01,3090$_{16}$. Size: Byte. Type: Write.	
D0	**Function**
0	VSB slave disabled
1	VSB slave enabled

It is important to note that even when the HK68/V3F slave is not enabled or the VSB address does not match the HK68/V3F, the HK68/V3F still responds to VSB cycles as a nonparticipating slave. This means that, unlike VME, if the slave is enabled or the map is changed after the VSB cycle has started, the HK68/V3F will not acknowledge and complete the cycle, because it has already responded to the cycle and indicated that it will not acknowledge.

8.3.1.4 Block Transfers

To support VSB slaves that cannot handle block transfers, the HK68/V3F has a bit that disables generation of block transfers by the HK68/V3F. A one-bit latch at address FF01,30A0$_{16}$ facilitates this. Default state on power-up is block mode enabled.

TABLE 8-4
VSB block transfer

Port address FF01,30A0$_{16}$. Size: Byte. Type: Write.	
D0	**Function**
0	VSB block mode enabled
1	VSB block mode disabled

Front Panel Interface

The front panel interface uses a 10-pin connector, P5 (Fig. 9-1), next to the reset switch. The connector pinout is outlined in Table 9-1. The interface includes four user-programmable outputs, LED0 through LED3.

Addresses for the user-programmable outputs are given in Table 9-2. The output signals are low when true. Each user-programmable output is suitable for connection to an LED cathode. An external resistor must be provided for each output to limit current to 15 milliamperes.

A recommended mating connector for P5 is Molex P/N 15-29-8148.

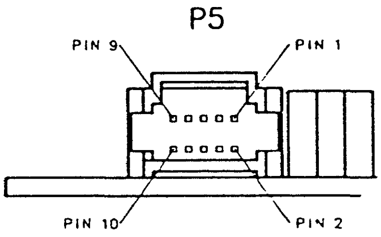


FIGURE 9-1. Front panel interface connector, P5

TABLE 9-1
Front panel interface connector pin assignments, P5

Pin	Signal Name	Pin	Signal Name
1	LED0	2	RESET (input)
3	LED1	4	GND
5	LED2	6	HALT (MPU status LED)
7	LED3	8	Not connected
9	VCC	10	Not connected

TABLE 9-2
Addresses of user-programmable outputs

User-programmable Signal Name	Hexadecimal Address (write-only)
LED0	$FF01,0900_{16}$
LED1	$FF01,0908_{16}$
LED2	$FF01,0910_{16}$
LED3	$FF01,0918_{16}$

Writing a 0 turns the selected output signal on; writing a 1 turns it off. At power-on or after a system reset, the output signals will be ON.

10

Serial I/O

10.1 INTRODUCTION

There are two RS-232C serial I/O ports on the HK68/V3F board. Each port may be configured for RS-422 operation. Each port has a separate baud rate generator and can operate in asynchronous or synchronous modes.

Refer to the Z85C30 SCC manual for programming details.

The SCC interrupts are tied to the LIRQ2 input of the VIC068 (Ports A and B).

10.2 RS-232 PIN ASSIGNMENTS (P4)

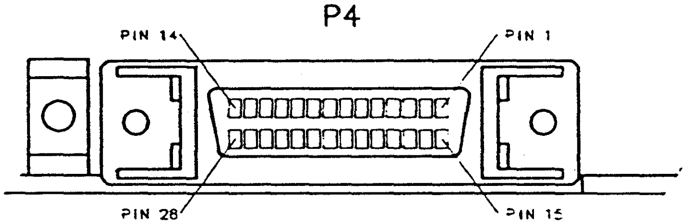


FIGURE 10-1. RS-232-C connector, P4

The RS-232-C connector is a high-density 28-pin connector that holds two serial cables. The connector pin assignments are given in Table 10-1.

Data transmission conventions are with respect to the external serial device. The HK68/V3F board is wired as a "Data Set" (or data communications equipment [DCE]).

TABLE 10-1a
Serial port pin assignments (P4) — Port A

Pin	"D" Pin	RS-232 Function	Direction	SCC Pin Function
2	2	Port A Tx Data	In	Rcv Data
3	15	Tx Clock	In	Rcv Clock
4	3	Rcv Data	Out	Tx Data
5	16	(not used)		
6	4	Request to Send *	In	DCD
7	17	Rcv Clock	In	Tx Clock
8	5	Clear to Send	Out	DTR
9	18	+5V (via J1)		
10	6	Data Set Ready	Out	RTS
11	19	(not used)		
12	7	Gnd		Sig Gnd
13	20	Data Terminal Ready*	In	CTS

TABLE 10-1b
Serial port pin assignments (P4) — Port B

Pin	"D" Pin	RS-232 Function	Direction	SCC Pin Function
16	2	Port B Tx Data	In	Rcv Data
17	15	Tx Clock	In	Rcv Clock
18	3	Rcv Data	Out	Tx Data
19	16	not used		
20	4	Request to Send *	In	DCD
21	17	Rcv Clock	In	Tx Clock
22	5	Clear to Send	Out	DTR
23	18	+5v (via J2)		
24	6	Data Set Ready	Out	RTS
25	19	not used		
26	7	Gnd		Sig Gnd
27	20	Data Terminal Ready*	In	CTS

Note that the interconnect cable from P4 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 pull-up resistors, controlled by J3.

NOTE: The serial ports may *appear* to be inoperative if J3 is set to default "FALSE" and if the device connected to the port does not drive the DTR and RTS pins TRUE. The monitor software, for example, initializes the SCC channels to respect the state of DTR and RTS.

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, were a terminal device. Thus all signal pairs, for example, "RTS" and "CTS," are switched between the interface connector and the SCC chip. For example, "Transmit Data," Px-1, is the data transmitted from the device to the HK68/V3F 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.

TABLE 10-2
Signal naming conventions

SCC Signal	I/F Signal	Direction
Tx Data	Rcv Data	to device
Rcv Data	Tx Data	from device
Tx Clock	Rcv Clock	from device (Port A)
Tx Clock	Rcv Clock	to device (Port B)
Rcv Clock	Tx Clock	See Table 10-1.
RTS	DSR	to device
CTS	DTR	from device
DTR	CTS	to device
DCD	RTS	from device
-	Ring Ind.	from device

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/V3F 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.

TABLE 10-3
RS-232 reversal cable

SCC Signal	P4 Pin #s	"D" Pin # at HK68/V3F	"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

Summary: The HK68/V3F may be directly connected to a "data terminal" device. However, a *cable reversal* is required for a connection to a "data set" device (for example, 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/V3F 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/V3F (or a modem) will have male connectors at both ends.

When you work with RS-232 communications, you will probably use many types of cable adapters — double males, double females, double males and females with reversal, cables with males and females at both ends, etc. We can 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. Port B is programmed in the same manner, substituting the correct control port address.

TABLE 10-4
SCC initialization sequence

Data (Hexadecimal)	Register Address (Hexadecimal)	Function
00	FF01,0010$_{16}$(write)	Reset SCC register counter
09,C0	FF01,0010$_{16}$(write)	Force reset (do for Port A only)
04,4C	FF01,0010$_{16}$(write)	Async mode, x16 clock, 2 stop bits tx
05,EA	FF01,0010$_{16}$(write)	Tx: RTS, Enable, 8 data bits
03,E1	FF01,0010$_{16}$(write)	Rcv: Enable, 8 data bits
01,00	FF01,0010$_{16}$(write)	No Interrupt, Update status
0B,56	FF01,0010$_{16}$(write)	No Xtal, Tx & Rcv clk internal,BR out
0C,baudL	FF01,0010$_{16}$(write)	Set low half of baud rate constant
0D,baudH	FF01,0010$_{16}$(write)	Set high half of baud rate constant
0E,03	FF01,0010$_{16}$(write)	Null, BR enable

Note: The notation "09,C0" (etc.) means the values 09 (hex) and C0 (hex) 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 Table 10-6 (Baud Rate Constants) below.

For information concerning SCC interrupt vectors, refer to section 3. Read the Z8530 technical manual for more details on SCC programming.

10.6 PORT ADDRESS SUMMARY

TABLE 10-5
SCC register addresses (hexadecimal)

Register	Port A	Port B
Control	FF01,0010$_{16}$	FF01,0000$_{16}$
Data	FF01,0018$_{16}$	FF01,0008$_{16}$

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 a formula.

TABLE 10-6
Baud rate constants

Baud Rate	x1 clock rate	x16 clock rate
110	72,725	4541
300	26,665	1665
1200	6665	415
2400	3331	206
4800	1665	102
9600	831	50
19,200	415	24
38,400	206	11

The time constant values listed above are computed as follows:

$$TC(x16) = \frac{500,000}{baud} - 2$$

$$TC(x1) = \frac{8,000,000}{baud} - 2$$

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 1 Mbit/sec, using the x1 clock and synchronous mode. For asynchronous transmission, the maximum practical rate using the x16 clock is 62,500 baud.

10.8 RS-422 OPERATION

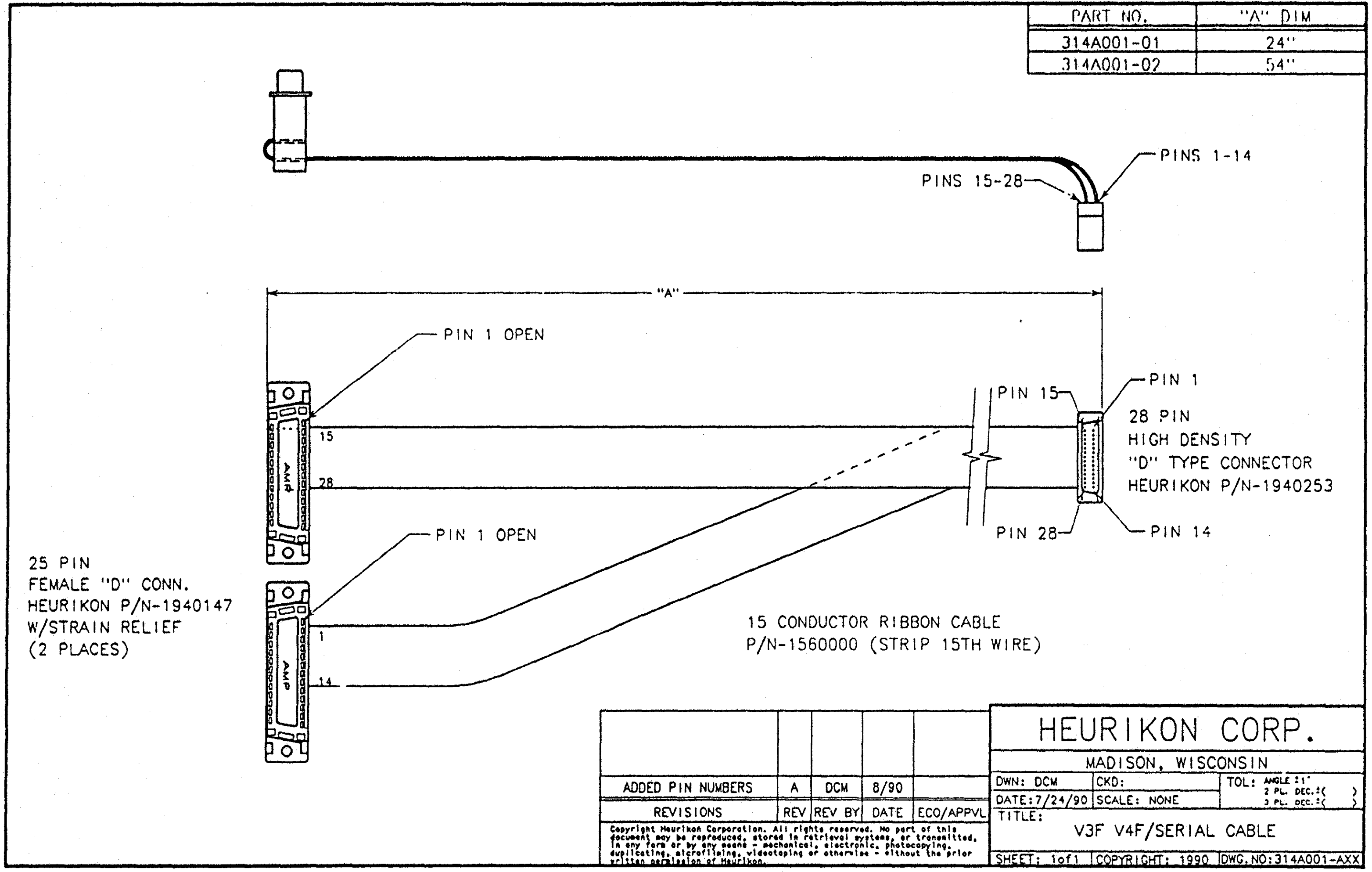
As an option, one or both of the serial ports on the HK68/V3F may be configured for RS-422 operation. The RS-422 option may either be installed when the board is ordered, or an existing HK68/V3F board may be factory-upgraded to add the option. Please contact Heurikon's Customer Service department for more information.

10.9 RELEVANT JUMPERS (SERIAL I/O)

TABLE 10-7
Relevant jumpers — serial I/O

Jumper	Function	Position
J1	+5 power Port A	Not installed.
J2	+5 power Port B	Not installed.
J3	RS-232 handshaking A: +12V B: -12V	J3-A (True) J3-B (False)

10.10 SERIAL I/O CABLE DRAWING



PART NO.	"A" DIM
314A001-01	24"
314A001-02	54"

ADDED PIN NUMBERS	A	DCM	8/90	
REVISIONS	REV	REV BY	DATE	ECO/APPVL

HEURIKON CORP.
MADISON, WISCONSIN

DWN: DCM	CKD:	TOL: ANGLE ±1° 2 PL. DEC.±( ) 3 PL. DEC.±( )
DATE:7/24/90	SCALE: NONE	
TITLE: V3F V4F/SERIAL CABLE		
SHEET: 1of1	COPYRIGHT: 1990	DWG.NO:314A001-AXX

FIGURE 10-2. Serial I/O cable

11

Timer Clock Peripheral (TCP)

11.1 INTRODUCTION

The built-in timer clock peripheral (DP8570A) is a real-time clock with counter timers that can be used to generate interrupts or count events. The clock runs from a rechargeable nickel-cadmium battery when power to the HK68/V3F is off.

The TCP has a full-function, real-time clock/calendar for 12- or 24-hour mode timekeeping, and day of week and day of year counters.

The TCP has two independent multifunction 10-MHz 16-bit timers, which operate in four modes. Each timer has a prescaler and can select any of eight possible clock inputs. A range of about 400 nanoseconds to 65,535 seconds is possible by programming the input clocks and timer counter values. The TCP has 31 bytes of battery-backed CMOS RAM available for storing programs.

Power fail logic and control are built into the TCP, with automatic log of time into RAM at power failure. Periodic, alarm/compare, timer, and power fail interrupts are also available.

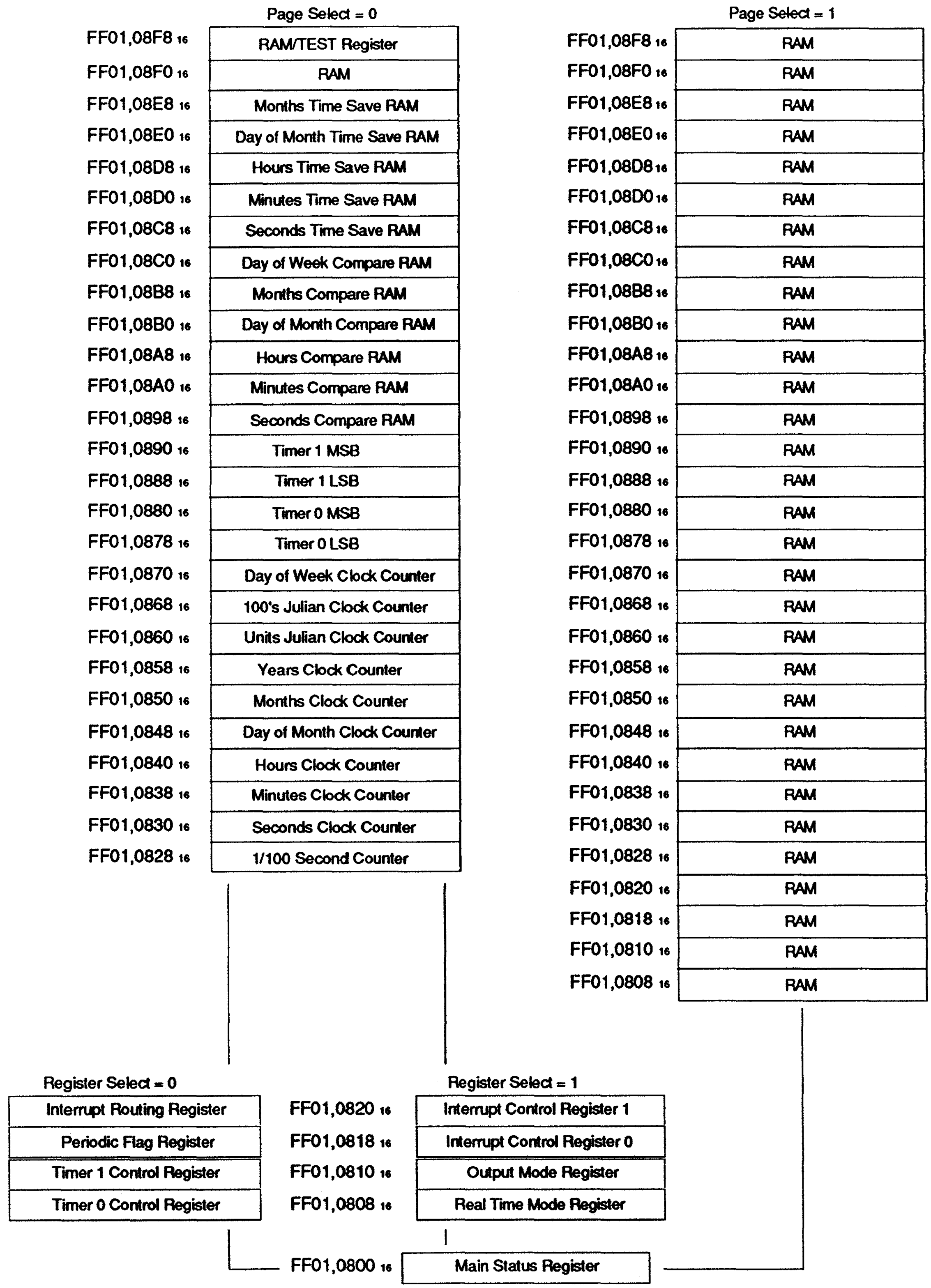
TABLE 11-1
TCP register addresses

Register	Address	Function
Control Registers	$FF01,0800_{16}$	TCP configuration and control

All functions of the TCP are controlled by a set of nine registers:

- Main Status Register
- Timer 0 Control Register
- Timer 1 Control Register
- Periodic Flag Register
- Interrupt Routing Register
- Real Time Mode Register
- Output Mode Register
- Interrupt Control Register 0
- Interrupt Control Register 1

The TCP memory is organized as two 31-register pages, with 8 bytes between registers, that share the Main Status Register (Figure 11-1). One page (Page 0) contains the remaining eight registers and all of the clock timer functions. The other page (Page 1) contains scratch pad RAM. A control bit in the Main Status Register is used to select one of the two pages. The eight registers are organized into two groups of four registers within Page 0.



Address	Page Select = 0
FF01,08F8$_{16}$	RAM/TEST Register
FF01,08F0$_{16}$	RAM
FF01,08E8$_{16}$	Months Time Save RAM
FF01,08E0$_{16}$	Day of Month Time Save RAM
FF01,08D8$_{16}$	Hours Time Save RAM
FF01,08D0$_{16}$	Minutes Time Save RAM
FF01,08C8$_{16}$	Seconds Time Save RAM
FF01,08C0$_{16}$	Day of Week Compare RAM
FF01,08B8$_{16}$	Months Compare RAM
FF01,08B0$_{16}$	Day of Month Compare RAM
FF01,08A8$_{16}$	Hours Compare RAM
FF01,08A0$_{16}$	Minutes Compare RAM
FF01,0898$_{16}$	Seconds Compare RAM
FF01,0890$_{16}$	Timer 1 MSB
FF01,0888$_{16}$	Timer 1 LSB
FF01,0880$_{16}$	Timer 0 MSB
FF01,0878$_{16}$	Timer 0 LSB
FF01,0870$_{16}$	Day of Week Clock Counter
FF01,0868$_{16}$	100's Julian Clock Counter
FF01,0860$_{16}$	Units Julian Clock Counter
FF01,0858$_{16}$	Years Clock Counter
FF01,0850$_{16}$	Months Clock Counter
FF01,0848$_{16}$	Day of Month Clock Counter
FF01,0840$_{16}$	Hours Clock Counter
FF01,0838$_{16}$	Minutes Clock Counter
FF01,0830$_{16}$	Seconds Clock Counter
FF01,0828$_{16}$	1/100 Second Counter

Address	Page Select = 1
FF01,08F8$_{16}$	RAM
FF01,08F0$_{16}$	RAM
FF01,08E8$_{16}$	RAM
FF01,08E0$_{16}$	RAM
FF01,08D8$_{16}$	RAM
FF01,08D0$_{16}$	RAM
FF01,08C8$_{16}$	RAM
FF01,08C0$_{16}$	RAM
FF01,08B8$_{16}$	RAM
FF01,08B0$_{16}$	RAM
FF01,08A8$_{16}$	RAM
FF01,08A0$_{16}$	RAM
FF01,0898$_{16}$	RAM
FF01,0890$_{16}$	RAM
FF01,0888$_{16}$	RAM
FF01,0880$_{16}$	RAM
FF01,0878$_{16}$	RAM
FF01,0870$_{16}$	RAM
FF01,0868$_{16}$	RAM
FF01,0860$_{16}$	RAM
FF01,0858$_{16}$	RAM
FF01,0850$_{16}$	RAM
FF01,0848$_{16}$	RAM
FF01,0840$_{16}$	RAM
FF01,0838$_{16}$	RAM
FF01,0830$_{16}$	RAM
FF01,0828$_{16}$	RAM
FF01,0820$_{16}$	RAM
FF01,0818$_{16}$	RAM
FF01,0810$_{16}$	RAM
FF01,0808$_{16}$	RAM

Register Select = 0	Address	Register Select = 1
Interrupt Routing Register	FF01,0820$_{16}$	Interrupt Control Register 1
Periodic Flag Register	FF01,0818$_{16}$	Interrupt Control Register 0
Timer 1 Control Register	FF01,0810$_{16}$	Output Mode Register
Timer 0 Control Register	FF01,0808$_{16}$	Real Time Mode Register
	FF01,0800$_{16}$	Main Status Register

FIGURE 11-1. TCP memory map

Using two blocks enables the nine control registers to be mapped into 5 locations. The only register that does not get switched is the Main Status Register, which contains the page select bit and the register select bit as well as status information. Register addressing is shown in Table 11-2.

TABLE 11-2
Register/counter/RAM addressing

A0-4	Page Select	Register Select	Description
Control Registers			
FF01,0800$_{16}$	X	X	Main Status Register
FF01,0808$_{16}$	0	0	Timer 0 Control Register
FF01,0810$_{16}$	0	0	Timer 1 Control Register
FF01,0818$_{16}$	0	0	Periodic Flag Register
FF01,0820$_{16}$	0	0	Interrupt Routing Register
FF01,0808$_{16}$	0	0	Real Time Mode Register
FF01,0810$_{16}$	0	1	Output Mode Register
FF01,0818$_{16}$	0	1	Interrupt Control Register 0
FF01,0820$_{16}$	0	1	Interrupt Control Register 1
Counters (Clock Calendar)			
FF01,0828$_{16}$	0	X	1/100, 1/10 sections (0-99)
FF01,0830$_{16}$	0	X	Seconds (0-59)
FF01,0838$_{16}$	0	X	Minutes (0-59)
FF01,0840$_{16}$	0	X	Hours (1-12, 0-23)
FF01,0848$_{16}$	0	X	Days of Month (1-28/29/30/31)
FF01,0850$_{16}$	0	X	Months (1-12)
FF01,0858$_{16}$	0	X	Years (0-99)
FF01,0860$_{16}$	0	X	Julian Date (LSB) (1-99)
FF01,0868$_{16}$	0	X	Julian Date (0-3)
FF01,0870$_{16}$	0	X	Day of Week (1-7)
Timer Data Registers			
FF01,0878$_{16}$	0	X	Timer 0 LSB
FF01,0880$_{16}$	0	X	Timer 0 MSB
FF01,0888$_{16}$	0	X	Timer 1 LSB
FF01,0890$_{16}$	0	X	Timer 1 MSB
Continues.			

TABLE 11-2 — *Continued.*
Register/counter/RAM addressing

Timer Compare RAM			
FF01,0898$_{16}$	0	X	Sec Compare RAM (0-59)
FF01,08A0$_{16}$	0	X	Min Compare RAM (0-59)
FF01,08A8$_{16}$	0	X	Hours Compare RAM (1-12, 0-23)
FF01,08B0$_{16}$	0	X	DOM Compare RAM (1-28/29/30/31)
FF01,08B8$_{16}$	0	X	Months Compare RAM (1-12)
FF01,08C0$_{16}$	0	X	DOW Compare RAM (1-7)
Time Save RAM			
FF01,08C8$_{16}$	0	X	Seconds Time Save RAM
FF01,08D0$_{16}$	0	X	Minutes Time Save RAM
FF01,08D8$_{16}$	0	X	Hours Time Save RAM
FF01,08E0$_{16}$	0	X	Day of Month Time Save RAM
FF01,08E8$_{16}$	0	X	Months Time Save RAM
FF01,08F0$_{16}$	0	1	RAM
FF01,08F8$_{16}$	0	X	RAM/Test Mode Register
FF01,0808$_{16}$- FF01,08F8$_{16}$	1	X	2nd Page General Purpose RAM

NOTE: Page select is bit D7 of the Main Status Register. Register select is bit D6 of the Main Status Register.

11.2 REAL-TIME CLOCK

The clock has 10 bytes of counters that count from 1/100 second to years. Each counter counts in binary coded decimal (BCD) and is synchronously clocked. The count sequence of the individual byte counters within the clock is shown in Table 11-2.

The day of week, day of month, day of year, and month counters all roll over to 1. The hours counter in 12-hour mode rolls over to 1 and the AM/PM bit toggles when the hours rolls over to 12 (AM=0, PM=1). The AM/PM bit is bit D7 in the hours counter. All other counters roll over to 0. The day of year counter is 12 bits long and occupies two addresses.

At initial power-up of both external and battery voltages, the counters contain random information. By "initial power-up," we mean the very first time the TCP is powered up, *or* the first time the TCP is powered up after battery replacement or discharge.

11.2.1 Reading the Clock

Clocking of the counter is asynchronous to reading the counter, so it is possible to read the counter while it is being incremented. This might result in an incorrect time reading. Checking a rollover bit can prevent an incorrect reading from this cause. The periodic interrupt status bits can be used, as described below:

1. Initialize program for reading clock.
2. Dummy read of periodic status bit to clear it.
3. Read counter bytes and store.
4. Read rollover bit and test it.
5. If rollover occurred, go back to step 3.
6. If no rollover, done.

Individual periodic status bits can be polled to detect the rollover. The periodic bit chosen should be equal to the highest frequency counter register to be read. That is, if only SECONDS through HOURS counters are read, then the SECONDS periodic bit should be used.

Enabling the periodic interrupt mask bits causes interrupts just as the clock rolls over. Enabling the desired update rate and providing an interrupt service routing that executes in less than 10 milliseconds enables clock reading without checking for a rollover.

Writing a 1 into the Time Save Enable bit (D7) of the Interrupt Routing Register and then writing a 0 reads the clock without checking the rollover bit. Writing a 1 into this bit enables the clock contents to be duplicated in the Time Save RAM. Changing the bit from a 1 to a 0 freezes and stores the contents of the clock in Time Save RAM. The time can then be read without concern for clock rollover, because internal logic takes care of synchronization of the clock. Because only the bits used by the clock counters will be latched, the Time Save RAM should be cleared prior to use to make sure that random data stored in the unused bits do not confuse the MPU. This bit can also provide time save at power failure (see the TCP *Handbook* for details). With the Time Save Enable bit at a logical zero, the Time Save RAM may be used as RAM if the latched read function is not necessary.

11.2.2 Initializing and Setting the Calendar-Clock

Time must be written to the clock at initial power-up and for time corrections. The usual procedure is to stop the clock by writing the start/stop bit in the Real Time Mode Register to a 0. This stops the clock from counting and disables the carry circuitry. When the clock's Real Time Mode Register is initialized, it is recommended that first the various mode bits be written while maintaining the start/stop bit reset, and then writing to the register a second time with the start/stop bit set.

This method is useful when the entire clock is being corrected. If only one location is being updated, the clock does not need to be stopped because doing so would reset the prescaler, and time would be lost. A good example of this is correcting the hours for daylight savings time. The best method for writing to the clock "on the fly" is to wait for the 1/100 second periodic interrupt. Then wait an addition 16 microseconds, and then write the data to the clock. See section 3 for information concerning TCP interrupt vectors.

11.3 COUNTER/TIMERS

The TCP has two independent multimode timers. Each is composed of a 16-bit negative-edge-triggered binary down counter and associated control. The timers can operate in four modes, and the input clock frequency can be selected from a prescaler over a wide range of frequencies. The timers are capable of generating interrupts as well as hardware output signals, and both the interrupt and timer outputs are fully programmable active high, or low, open drain, or push-pull.

Each timer consists of a 16-bit counter, two 8-bit input registers, two 8-bit output registers, clock prescaler, mode control logic, and output control logic. The timer and data registers are organized as two bytes for each timer. Under normal operations, a read/write to the timer locations will read or write to the data input register. The timer contents can be read by setting the counter Read bit (RD) in the timer control register.

11.3.1 Timer Initialization

The timer's operation is controlled by the set of registers shown in Table 11-3.

TABLE 11-3
Timer registers

Register Name	Register Select	Page Select	Address
Timer 0 Data MSB	X	0	FF01,0880$_{16}$
Timer 0 Data LSB	X	0	FF01,0878$_{16}$
Timer 0 Control Register	0	0	FF01,0808$_{16}$
Timer 1 Data MSB	X	0	FF01,0890$_{16}$
Timer 1 Data LSB	X	0	FF01,0888$_{16}$
Timer 1 Control Register	0	0	FF01,0810$_{16}$
Interrupt Routing Register	0	0	FF01,0820$_{16}$
Interrupt Control Register 0	1	0	FF01,0818$_{16}$
Output Mode Register	1	0	FF01,0810$_{16}$

All timer registers must be initialized before any timer is started. The Timer Control Register should first be set to select the timer mode with the timer start/stop bit reset. Then, when the timer is to be started, the control register should be rewritten in exactly the same way, but with the start/stop bit set.

Each timer is capable of operation in any of four modes that are programmed in each timer's Control Register (see Table 11-3). The four modes operate on the two 8-bit data words stored into the Data Input Register. At the beginning of a counting cycle, the two bytes are loaded into the timer and the timer starts counting down toward zero. The action taken when zero is reached depends on the selected mode, but in general the timer output changes state and an interrupt is generated if the timer interrupts are unmasked.

TABLE 11-4
TCP register summary

D7	D6	D5	D4	D3	D2	D1	D0
Main Status Register: PS=0, RS=0, Address=FF01,0800$_{16}$							
R/W	R/W	R[a]	R[a]	R[a]	R[a]	R[b]	R[c]
Page Select	Register Select	Timer 1 Interrupt	Timer 0 Interrupt	Alarm Interrupt	Periodic Interrupt	Power Fail Interrupt	Interrupt Status

[a] Reset by writing 1 to bit.

[b] Set/reset by voltage at TCP $\overline{\text{PFAIL}}$ pin.

[c] Reset when all pending interrupts are removed.

Continues.

Timer 0 Control Register: PS=0, RS=0, Address=FF01,0808$_{16}$

Count Hold Gate	Timer Read	Input Clock Select C2	Input Clock Select C1	Input Clock Select C0	Mode Select M1	Mode Select M0	Timer Start/ $\overline{\text{Stop}}$

All bits R/W.

Timer 1 Control Register: PS=0, RS=0, Address=FF01,0810$_{16}$

Count Hold Gate	Timer Read	Input Clock Select C2	Input Clock Select C1	Input Clock Select C0	Mode Select M1	Mode Select M0	Timer Start/ $\overline{\text{Stop}}$

All bits R/W.

Periodic Flag Register: PS=0, RS=0, Address=FF01,0818$_{16}$

R/W	R/W[d]	R[e]	R[e]	R[e]	R[e]	R[e]	R[e]
Test Mode	Osc. Fail/Single Supply	1-ms Flag	10-ms Flag	100-ms Flag	Seconds Flag	10-sec Flag	Minute Flag

[d] Read Osc fail. Write 0 Batt-Backed Mode. Write 1 Single Supply Mode.

[e] Reset by positive edge of read.

Interrupt Routing Register: PS=0, RS=0, Address=FF01,0820$_{16}$

R/W	R[f]	R/W	R/W	R/W	R/W	R/W	R/W
Time Save Enable	Low Battery Flag	Power Fail Delay Enable	Timer 1 Int. Route MFO/ $\overline{\text{INT}}$	Timer 0 Int. Route MFO/ $\overline{\text{INT}}$	Alarm Int. Route MFO/ $\overline{\text{INT}}$	Periodic Int. Route MFO/ $\overline{\text{INT}}$	Power Fail Int Route MFO/ $\overline{\text{INT}}$

[f] Set and reset by V_{BB} voltage.

Real Time Mode Register: PS=0, RS=1, Address=FF01,0808$_{16}$

Crystal Freq. XT1	Crystal Freq. XT0	Timers EN on Back-up	Interrupt EN on Back-up	Clock Start/ $\overline{\text{Stop}}$	12/ $\overline{24}$ Hr Mode	Leap Year MSB	Leap Year LSB

All bits R/W.

Output Mode Register: PS=0, RS=1, Address=FF01,0810$_{16}$

MFO as Crystal	MFO as Timer 0	MFO PP/ $\overline{\text{OD}}$	MFO Active HI/ $\overline{\text{LO}}$	INTR PP/ $\overline{\text{OD}}$	INTR Active HI/ $\overline{\text{LO}}$	T1 PP/ $\overline{\text{OD}}$	T1 Active HI/ $\overline{\text{LO}}$

All bits R/W.

Continues.

Interrupt Control Register 0: PS=0, RS=1, Address=FF01,0818$_{16}$							
Timer 1 Interrupt Enable	Timer 0 Interrupt Enable	.1-ms Interrupt Enable	10-ms Interrupt Enable	100-ms Interrupt Enable	Seconds Interrupt Enable	10-Second Interrupt Enable	Minute Interrupt Enable
All bits R/W.							

Interrupt Control Register 1: PS=0, RS=1, Address=FF01,0820$_{16}$							
Power Fail Interrupt Enable	Alarm Interrupt Enable	DOW Interrupt Enable	Month Interrupt Enable	DOM Interrupt Enable	Hours Interrupt Enable	Minute Interrupt Enable	Second Interrupt Enable
All bits R/W.							

11.3.2 Reading the Timers

Reading the timer data register addresses, 0F$_{16}$ and 10$_{16}$ for Timer 0 and 11$_{16}$ and 12$_{16}$ for Timer 1, results in reading the input data register that contains the preset value for the timers. During timer operation it is often useful to read the contents of the 16-bit down counter. This reading may be an erroneous value of FFFF$_{16}$.

To read a timer, the microprocessor first sets the timer read bit in the appropriate Timer Control Register high. This causes the counter's contents to be latched to two 8-bit output registers, and enables those registers to be read if the microprocessor reads the timer's input data register addresses. On reading the LSB byte, the timer read bit is internally reset, and subsequent reads of the timer locations will return the input register values.

11.4 PROGRAMMABLE PRESCALER/OSCILLATOR

A programmable prescaler that divides the crystal oscillator frequency to 32 kHz and further to 100 Hz feeds the counter chain. Crystal frequency selections are 32 kHz, 32.768 kHz, 4.9152 kHz, and 4.194304 MHz. The HK68/V3F uses a 32.768 crystal, and the TCP must be programmed to reflect this selection.

Once 32.768 kHz is generated, it feeds both timers and the clock. The clock and timer prescaler can be independently enabled by controlling the timer or clock start/stop bits.

11.5 TCP IMPLEMENTATION

The TCP has one dedicated general-purpose interrupt output and another output on its MFO (multiple function output) pin. Each may be selected to generate an interrupt from any source. The MFO pin can be programmed either as oscillator output or Timer 0's output. The interrupt output is the only output that can generate a TCP interrupt on the HK68/V3F. The MFO output is routed to the test point TP1 to allow calibration of the TCP clock.

The TCP clock is calibrated at the factory and should not be recalibrated unless absolutely necessary. To recalibrate, a program must be run to set up the TCP so that the 32.768 kHz clock is driven on the MFO output. Heurikon's Hbug monitor performs this initialization automatically on reset. A frequency counter can then be attached to TP1 and CVxx can be adjusted to produce another 32.768 kHz on TP1.

Note: The clock is only as accurate as the equipment used to calibrate it. An inaccuracy as small as .01% causes a loss or gain of 4 minutes per month.

Counters can be linked externally to the TCP on the HK68/V3F. A pair of gate inputs on the TCP gates the checking of each counter. For counter 1, this input is tied true so that the counter runs continuously. For counter 0, it is programmable; writing 01_{16} to address $FF01,0700_{16}$ causes the output T1 of counter 1 to be tied to the gate input G0 of counter 0. Writing 00_{16} to $FF01,0700_{16}$ causes the G0 input to be always active so that counter 0 always counts. The HK68/V3F powers up with G0 always active. By programming the TCP so that counter 1 produces a short pulse on terminal count and writing counter 1's output to counter 0's input, counter 0 can be made to count only when counter 1 reaches terminal count, thus cascading counters 1 and 0.

TABLE 11-5
Cascading TCP counters

Port address: $FF01,0700_{16}$. Size: Byte. Type: Write.	
D0	**Function**
0	Sets G0 of counter 0 always active; counter 0 always counts.
1	Ties T1 of counter 1 to G0 at counter 0.

11.5.1 Interrupts

The TCP interrupt output INTR is routed to the VIC LIRQ1. To generate interrupts, the TCP must be set up to generate the interrupt and the VIC must be set to handle the interrupt. Plans for future revisions of the HK68/V3F call for an option to route the MFO to a second VIC input.

11.6 BATTERY LIFE

The rechargeable nickel-cadmium battery takes about 4 months (125 days) to discharge. The 4-month duration is a worst-case estimate. It takes about 2-3 days (70 hours) to recharge — again, a worst-case estimate.

12

Corebus Interface

12.1 INTRODUCTION

The Corebus consists of two interfaces: the Module Synchronous Bus and the Module Peripheral Bus. These two interfaces are optimized for different bandwidth devices. The Module Synchronous Bus is optimized for high speed transfers; the Module Synchronous Bus provides a high-speed synchronous interface for communications between a baseboard and module. This interface is designed to provide a bus bandwidth comparable to that of the MPU on the baseboard, while still providing a simple, straightforward interface. The interface provides a master interface, a slave interface, and a slave-override interface.

The Module Peripheral Bus is optimized for slower devices, such as I/O registers, which are accessed infrequently. Each bus has address and control lines, but they share the data bus and interrupt lines. Figure 12-1 shows a typical system utilizing Corebus.

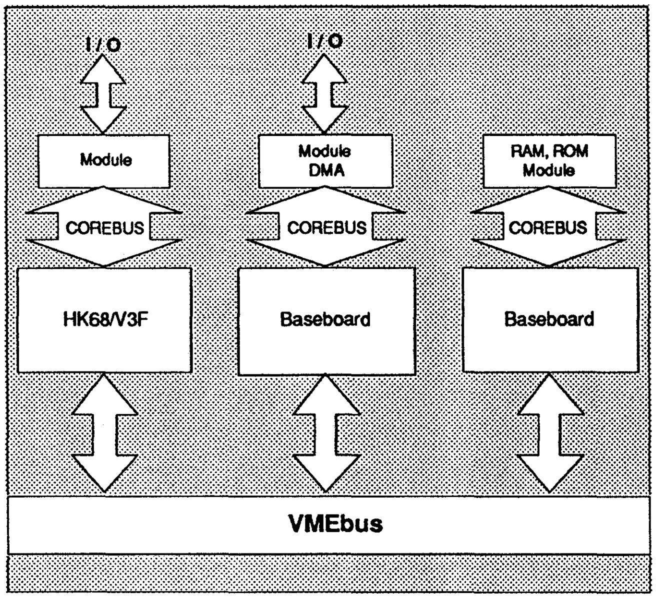


FIGURE 12-1. Typical Corebus application

12.2 COREBUS MODULE SYNCHRONOUS BUS

The Module Synchronous Bus is a synchronous interface that facilitates communication between the module and baseboard. It provides the following support:

- Master devices are supported through a set of arbitration signals and a protocol for arbitrating the ownership of the baseboard bus.
- Slave devices are supported through a 64-Kbyte slave space and select signal.
- Slave-override devices are supported through a set of override signals that allow the module to map itself into any space in the baseboard's memory map.
- Interrupts and interrupt acknowledges are supported through a set of interrupt control signals.
- Transfer cycles are supported through a synchronous protocol. The synchronous bus signals use a protocol that is synchronous with the bus clock, which may be up to 50 MHz. The clock rate may be increased in future revisions.
- Burst transfers that transfer multiple data words from one address are supported. Sequencing of burst addresses can occur in multiple modes.
- Locked cycles are supported through a lock signal that flags sequences of cycles as locked.

12.3 COREBUS MODULE PERIPHERAL BUS

The Module Peripheral bus provides the following support:

- Enough flexibility is provided in the timing and interrupts to support most peripheral devices.
- The bus handshake is set up to allow control of key timing parameters: address to read/write strobe set-up, read/write length, read/write to address hold time, data three-state time, and recovery time. These parameters are controlled by the module through handshake lines.

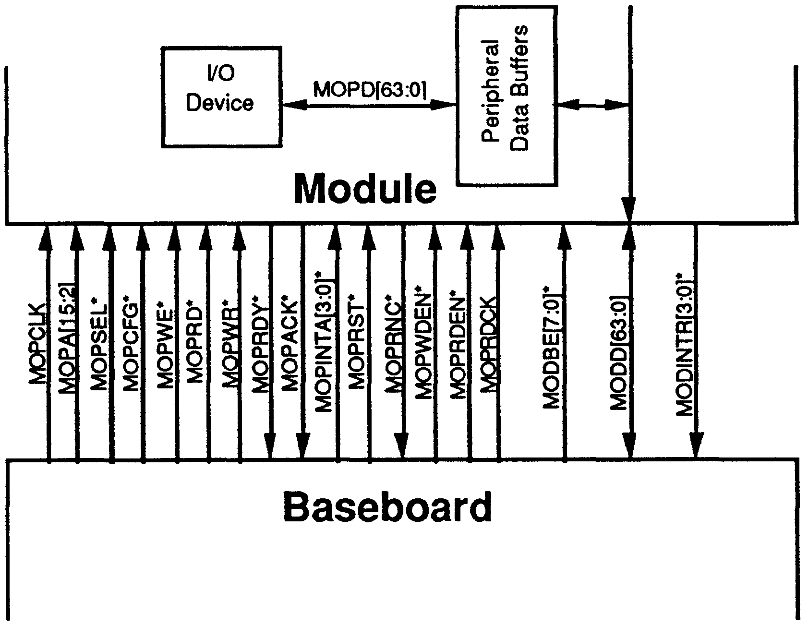


FIGURE 12-2. The Corebus Module Peripheral Bus

- There are four interrupt lines from the module to the baseboard. Each interrupt has a separate associated acknowledge. Module devices respond to interrupt acknowledges by providing a vector. The timing of interrupt acknowledge cycles is controlled by the same handshake lines that control normal accesses.

- The Module Peripheral Bus has a separate address space for configuration and initialization. This allows the baseboard to determine the type of module installed and to configure module attributes. There is support for intelligent modules that must perform diagnostics and initialization before they are ready for accesses from the baseboard.

- Transfer cycles on the Module Peripheral Bus are controlled by a select, a read or write strobe, and two handshake lines. The module controls the cycle timing with the two handshake lines. One of the handshake lines controls the read and write strobes and the other controls the select line.

- The use of a latching data buffer between the peripheral and the system data buses allows the timing of read data on the system bus to be independent of the end of the peripheral cycle. When the data from the peripheral is valid, it is latched in the data buffer, and the peripheral cycle can be terminated. The system bus can then control how long the data remains driven from the peripheral data buffer and when the data lines become three-stated.

12.4 COREBUS SIGNALS

12.4.1 Module Synchronous Bus

12.4.1.1 Arbitration Signals

MOSHLDR* HOLD REQUEST. The MOSHLDR* signal is asserted when a module wants the bus and when it is using the bus. Modules that do not have the capability of becoming an initiator need not drive MOSHLDR*.

MOSHLDA* HOLD ACKNOWLEDGE. The MOSHLDA* signal is normally inactive and is only driven active by the baseboard when it is acknowledging a request from the module. Modules requesting the bus must monitor MOSHLDA*. When they see it go active they know the module synchronous bus is cleared for use.

MOSBCLR* BUS CLEAR. Any module that currently is using the baseboard local bus must monitor this signal and release control of the Module Synchronous Bus as soon as possible after it sees this signal asserted.

12.4.1.2 Slave Control Signals

MOSENB* SLAVE ENABLE. Modules that contain synchronous bus slave devices must monitor this signal. When MOSENB* is asserted, the module may respond as target of the cycle.

MOSOVR* OVERRIDE. Modules that support override must monitor the address bus, but not MOSAVAL, looking for a match to their preprogrammed address. When a module sees a match, it must assert this signal with a very short delay. The address decode circuitry on the baseboard monitors MOSOVR* and disables on-card select logic for cycles where it is asserted.

MODINTR[3:0]* INTERRUPT REQUEST. Modules that need to interrupt the baseboard processor for handling an exception condition may inform the baseboard via these signals.

MOSINTA[3:0]* INTERRUPT ACKNOWLEDGE. Modules that contain devices capable of performing slave interrupt acknowledge cycles must monitor one of the MOSINTA[3:0]* signals. When one of them is asserted, the module may respond as target of the cycle, or may route the cycle to the Module Peripheral Bus.

MOSIAP* INTERRUPT ACKNOWLEDGE PERIPHERAL. When a module detects any asserted interrupt acknowledge signals MOSINTA

[3:0]*, it may indicate that the interrupt is to be acknowledged on the Module Peripheral Bus by asserting MOSIAP*. The baseboard contains logic to detect MOSIAP* asserted and runs the interrupt acknowledge cycle on the Module Peripheral Bus.

12.4.1.3 Address Bus Signals

MOSA [31:2] ADDRESS BUS. The Module Synchronous Bus was designed so that in most cases the address lines will be connected directly to the baseboard's internal address bus. Address line MOSA2 is invalid for 64-bit initiators and should be ignored by targets when accessed by a 64-bit initiator.

MODBE[7:0]* BYTE ENABLES. The MODBE[7:0]* signals determine which byte or bytes in a 64-bit word are valid during a transfer. Table 12-1 shows the relationships between the MODBE[7:0]* lines and the data bus. This relationship holds true for both 32-bit and 64-bit devices.

TABLE 12-1
Summary of MODBE7* through MODBE0* functions

Byte Enable	Data Lines Valid
MODBE7*	MODD[63:56]
MODBE6*	MODD[55:48]
MODBE5*	MODD[47:40]
MODBE4*	MODD[39:32]
MODBE3*	MODD[31:24]
MODBE2*	MODD[23:16]
MODBE1*	MODD[15:8]
MODBE0*	MODD[7:0]

MOSAVAL ADDRESS VALID. Address valid is a qualifier for the address. If asserted, it indicates that the cycle is a Module Synchronous Bus cycle and is intended for a target on the synchronous bus.

12.4.1.4 Data Bus Signals

MODD[63:0] DATA BUS. The Module Synchronous Bus was designed so that the local data bus of both the baseboard and module could be directly connected. Initiators and targets that have only a 32-bit data bus should neither drive nor receive MODD[63:32].

12.4.1.5 Cycle Control Signals

MOSTS* TRANSFER START. The owner of the Module Synchronous Bus initiates a bus cycle by asserting MOSTS*. The target must monitor MOSTS* to determine when a cycle is taking place and qualify it with the address to see if the module is being accessed.

MOSTA[1:0]* TRANSFER ACKNOWLEDGE. MOSTA[1:0] is driven by the target of a transfer to indicate the status of the bus cycle. MOSTA[1:0] may be driven to any of four states to signal the status of the cycle. Table 12-2 summarizes the possible states of transfer acknowledge and their meaning:

TABLE 12-2
Summary of MOSTA[1:0]* states

MOSTA1*	MOSTA0*	Acknowledge State
H	H	Transfer Not Complete
H	L	Data Transfer Acknowledged
L	H	Exception Acknowledge
L	L	Relinquish And Retry Acknowledge

MOSTX* TRANSFER EXTEND. For a normal read bus cycle, data would only be asserted for a short, fixed period of time. The initiator of the bus cycle may assert the transfer extend signal to inform the target that it wants the acknowledge portion of the cycle to be extended for an additional clock cycle.

MOSWR* WRITE. This signal must be driven by the initiator of a bus cycle to indicate the direction of data flow for the bus cycle. When asserted, it indicates that data are to be transferred from the initiator to the target. When deasserted, it indicates that the data are to be transferred from target to initiator.

MOSBSTR* BURST REQUEST. MOSBSTR* is used by the initiator to indicate that it wants to perform burst transfer cycles. As long as the initiator can perform burst data transfers, it may drive this signal low to request additional data cycles.

MOSBSTA* BURST ACKNOWLEDGE. MOSBSTA* is the target's indication that it is capable of providing burst data for the cycle requested. Initiators that drive MOSBSTR* must monitor the MOSBSTA* and use it to determine if additional data transfers are forthcoming. If the target can support burst transfers on the cycle requested, it must issue burst acknowledge to inform the initiator that it can do so.

MOSIS* IS SEQUENTIAL. When asserted by the initiator during a burst transfer, it signals that the burst is sequential and that successive data operands are associated with successive words in memory.

When deasserted by the initiator during a burst transfer, it indicates that the successive data portions of the burst reference successive operands modulo-4. When asserted by the initiator during an in-block sequence, it indicates that the addresses for successive cycles will be to successive locations. When deasserted by the initiator during an in-block sequence, it indicates that the in-block sequence may include cycles whose addresses have any arbitrary order.

MOSNC* NONCACHEABLE. When asserted by the initiator, MOSNC* indicates that the cycle is to a noncacheable area of memory. In some cases it may be useful for the initiator of a read cycle to indicate to the target that the cycle is noncacheable so that the target is not forced to read all bytes of the word addressed. Initiators that make no distinction between cacheable and noncacheable information need not drive MOSNC*.

MOSCI* CACHE INHIBIT. When asserted by a target, MOSCI* indicates that the area of memory is noncacheable. Initiators that contain caches may monitor MOSCI* and disable internal caching of locations that return MOSCI* asserted.

MOSLOCK* LOCK. When asserted by the initiator, MOSLOCK* indicates that the target being accessed may not allow any other devices to access it until the lock signal is released. Initiators that never perform locked cycles need not drive MOSLOCK*.

MOSIB* IN BLOCK. When asserted by the initiator through a sequence of accesses, MOSIB* indicates that all accesses in the sequence reference locations within the same 512-word block.

12.4.1.6 Central Control Signals

MOSCLK[2:0] PRIMARY BUS CLOCKS. These signals are provided by the baseboard and received by the module. They are used to synchronize all activity on the Module Synchronous Bus.

MOSRST* RESET. This signal is normally inactive and is only driven active by the baseboard when it detects some condition requiring a reset of module logic.

MODB64* BASEBOARD 64 BITS WIDE. Assertion of MODB64* by the baseboard indicates that the baseboard can be the initiator or target of accesses that use all 64 bits of the data bus.

MODM64* MODULE 64 BITS WIDE. Assertion of MODM64* by the module indicates that the module can be the initiator or target of accesses that use all 64 bits of the data bus.

12.4.2 Module Peripheral Bus

12.4.2.1 Cycle Control Signals

MOPA[15:2] ADDRESS. The fourteen active high address lines are driven by the baseboard when it is accessing devices or configuration space on the module.

MODBE[7:0]* BYTE ENABLE. These eight signals determine which byte or bytes in a 64-bit word are valid during a Module Peripheral Bus or a Module Synchronous Bus transfer. Table 12-3 shows the relationships between the MODBE[7:0]* lines and the data bus.

TABLE 12-3
Summary of MODBE7* through MODBE0* functions

Byte Enable	Data Lines Valid
MODBE7*	MODD[63:56]
MODBE6*	MODD[55:48]
MODBE5*	MODD[47:40]
MODBE4*	MODD[39:32]
MODBE3*	MODD[31:24]
MODBE2*	MODD[23:16]
MODBE1*	MODD[15:8]
MODBE0*	MODD[7:0]

MOPSEL* SELECT. The MOPSEL* signal is used by the baseboard to indicate that it is beginning a cycle to the peripheral bus.

MOPWE* WRITE ENABLE. The MOPWE* signal is used by the baseboard to indicate that the current cycle is a write cycle.

MOPRDY* READY. The MOPRDY* signal is driven by the module to control the timing of the read and write strobes (MOPRD* and MOPWR*).

MOPRD* READ. The MOPRD* signal is driven by the baseboard to signal a read cycle to the module.

MOPWR* WRITE. The MOPWR* signal is driven by the baseboard to signal a write cycle to the module.

MOPACK* ACKNOWLEDGE. The MOPACK* signal is driven by the module to terminate data transfer cycles on the peripheral bus and to delay the start of cycles while data are being driven on the bus from a previous cycle.

MODD[31:0] MODULE DATA. The 32 active high data lines carry all data between the baseboard and the module. They are used for Module Peripheral Bus transfers and Module Synchronous Bus transfers.

MOPRDEN* READ DATA ENABLE. The MOPRDEN* signal is used by the baseboard to enable read data from the peripherals on the module to the MODD[31:0] data lines.

MOPWDEN* WRITE DATA ENABLE. The MOPWDEN* signal is used by the baseboard to enable write data from the MODD[31:0] data lines to the peripherals on the module.

MOPRDCK READ DATA CLOCK. The MOPRDCK signal is used to clock data into registered transceivers on the module.

12.4.2.2 Interrupt Signals

MODINTR[3:0]* INTERRUPT. The MODINTR[3:0]* signals are used by the module to interrupt the baseboard. They are active low and asynchronous.

MOPINTA[3:0]* INTERRUPT ACKNOWLEDGE. The MOPINTA[3:0]* signals are used by the baseboard to acknowledge MODINTR[3:0]* from the module. (See section 12.4.1.) The Interrupt Acknowledge signals act as select signals in place of MOPSEL*.

12.4.2.3 Configuration Signals

MOPCFG* CONFIGURATION SPACE SELECT. The MOPCFG* signal is driven by the baseboard to indicate that it is accessing the configuration space of the module.

MOPRNC* RESET NOT COMPLETE. The MOPRNC* is driven by the module to indicate that it has not completed its initial configuration and diagnostic tests.

12.4.2.4 Central Control Signals

MOPCLK CLOCK. The MOPCLK provides synchronization for the transfer control signals on the Module Peripheral Bus. It is driven by the baseboard.

MOPRST* RESET. The MOPRST* signal provides a reset for the Module Peripheral Bus. It is driven by the baseboard.

12.5 COREBUS INTERFACE PIN ASSIGNMENTS (P3)

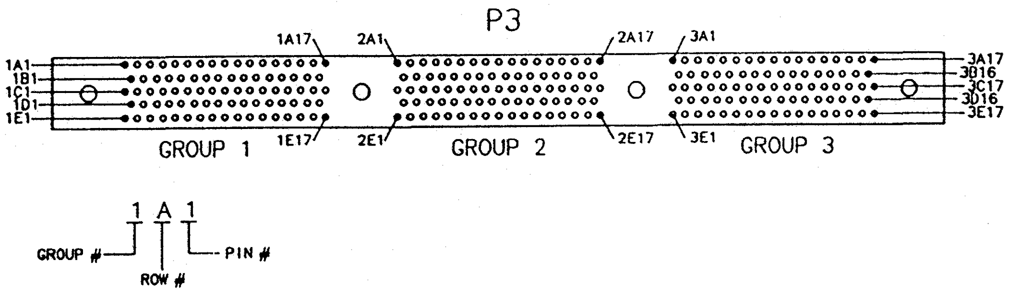


FIGURE 12-3. Corebus connector, P3

TABLE 12-4
Corebus pin assignments

Group 1									
Row A		**Row B**		**Row C**		**Row D**		**Row E**	
Pin	**Signal**	**Pin**	**Signal**	**Pin**	**Signal**	**Pin**	**Signal**	**Pin**	**Signal**
1	MOPRNC*	1	MOPA15	1	VCC	1	MODD63	1	Reserved
2	Reserved	2	MOPA14	2	VCC	2	MODD62	2	Reserved
3	MOPCFG*	3	MOPA13	3	VCC	3	MODD61	3	Reserved
4	MOPSEL*	4	MOPA12	4	VCC	4	MODD60	4	MODD47
5	MOPWE*	5	MOPA11	5	VCC	5	MODD59	5	MODD46
6	MOPRDY*	6	MOPA10	6	VCC	6	MODD58	6	MODD45
7	MOPRD*	7	MOPA9	7	VCC	7	MODD57	7	MODD44
8	MOPWR*	8	MOPA8	8	VCC	8	MODD56	8	MODD43
9	MOPACK*	9	MOPA7	9	GND	9	MODD55	9	MODD42
10	MOPRDEN*	10	MOPA6	10	GND	10	MODD54	10	MODD41
11	MOPRDCK	11	MOPA5	11	GND	11	MODD53	11	MODD40
12	MOPWDEN*	12	MOPA4	12	GND	12	MODD52	12	MODD39
13	MOPINTA3*	13	MOPA3	13	GND	13	MODD51	13	MODD38
14	MOPINTA2*	14	MOPA2	14	GND	14	MODD50	14	MODD37
15	MOPINTA1*	15	Reserved	15	GND	15	MODD49	15	MODD36
16	MOPINTA0*	16	MOPCLK	16	GND	16	MODD48	16	MODD35
17	MOPRST*			17	GND			17	MODD34
Continues.									

TABLE 12-4 — *Continued.*
Corebus pin assignments

Group 2									
Row A		**Row B**		**Row C**		**Row D**		**Row E**	
Pin	**Signal**	**Pin**	**Signal**	**Pin**	**Signal**	**Pin**	**Signal**	**Pin**	**Signal**
1	Reserved	1	Reserved	1	GND	1	MODD31	1	MODD33
2	Reserved	2	Reserved	2	GND	2	MODD30	2	MODD32
3	Reserved	3	Reserved	3	GND	3	MODD29	3	Reserved
4	MODBE7*	4	MODBE6*	4	GND	4	MODD28	4	MOSA31
5	MODBE5*	5	MODBE4*	5	GND	5	MODD27	5	MOSA30
6	MODBE3*	6	MODBE2*	6	GND	6	MODD26	6	MOSA29
7	MODBE1*	7	MODBE0*	7	GND	7	MODD25	7	MOSA28
8	MODM64*	8	MODB64*	8	GND	8	MODD24	8	MOSA27
9	MOSRST*	9	MODINTR0*	9	VCC	9	MODD23	9	MOSA26
10	MODINTR1*	10	MODINTR2*	10	VCC	10	MODD22	10	MOSA25
11	MODINTR3*	11	MOSINTA0*	11	VCC	11	MODD21	11	MOSA24
12	MOSINTA1*	12	MOSINTA2*	12	VCC	12	MODD20	12	MOSA23
13	MOSINTA3*	13	MOSIAP*	13	VCC	13	MODD19	13	MOSA22
14	MOSENB*	14	MOSHLDA*	14	VCC	14	MODD18	14	MOSA21
15	MOSHLDR*	15	MOSBCLR*	15	VCC	15	MODD17	15	MOSA20
16	MOSLOCK*	16	MOSIS*	16	VCC	16	MODD16	16	MOSA19
17	MOSIB*			17	VCC			17	MOSA18
Group 3									
Row A		**Row B**		**Row C**		**Row D**		**Row E**	
Pin	**Signal**	**Pin**	**Signal**	**Pin**	**Signal**	**Pin**	**Signal**	**Pin**	**Signal**
1	Reserved	1	Reserved	1	VCC	1	MODD15	1	MOSA17
2	Reserved	2	Reserved	2	VCC	2	MODD14	2	MOSA16
3	Reserved	3	Reserved	3	VCC	3	MODD13	3	MOSA15
4	Reserved	4	MOSAVAL	4	VCC	4	MODD12	4	MOSA14
5	Reserved	5	GND	5	VCC	5	MODD11	5	MOSA13
6	Reserved	6	MOSCLK3	6	VCC	6	MODD10	6	MOSA12
7	-12V	7	GND	7	VCC	7	MODD9	7	MOSA11
8	Reserved	8	MOSCLK2	8	VCC	8	MODD8	8	MOSA10
9	+12V	9	GND	9	GND	9	MODD7	9	MOSA9
10	Reserved	10	MOSCLK1	10	GND	10	MODD6	10	MOSA8
11	Reserved	11	GND	11	GND	11	MODD5	11	MOSA7
12	MOSOVR*	12	MOSCI*	12	GND	12	MODD4	12	MOSA6
13	MOSNC*	13	MOSBSTA*	13	GND	13	MODD3	13	MOSA5
14	MOSBSTR*	14	MOSTA0	14	GND	14	MODD2	14	MOSA4
15	MOSTA1	15	MOSWR*	15	GND	15	MODD1	15	MOSA3
16	MOSTX*	16	MOSTS*	16	GND	16	MODD0	16	MOSA2
17	Reserved			17	GND			17	Reserved

13

Summary Information

13.1 SOFTWARE INITIALIZATION SUMMARY

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 fetched from the first location in EPROM which, on power-up, is mapped at location $0000{,}0000_{16}$. The initial supervisory stack pointer is fetched from EPROM at location $0000{,}0004_{16}$.

2. Enable the standard memory map. At power-up, EPROM is mapped everywhere. Any access to a location in the range $FC00{,}0000_{16}$ to $FC3F{,}FFFF_{16}$ enables the standard memory map. Enabling the map can be done by branching to EPROM at location $FC00{,}0000_{16}$ to $FC3F{,}FFFF_{16}$ or loading the reset vector with an address in that range.

3. Any reset may be the result of a power failure or a system reset. A reset due to a power failure may require additional processing. If initialized to indicate power failure, the TCP may be read to determine if the current reset is due to a power-up or system reset.

4. Perform any required TCP initialization and/or testing.

5. Test and initialize the SCC.

6. Read the EEPROM to determine RAM size.

7. If desired, perform power-up memory testing.

8. If this is a power-up, initialize all of memory to prevent parity errors.

9. Determine where to place the exception vector table in RAM and initialize the 68030 Vector Base Register to reference the base of the table.

10. Initialize an exception vector table. This step links the various exception and interrupt sources with the appropriate service routines.

11. Initialize the VMEbus interface and VIC.
12. Initialize the VSB interface.
13. Release the VMEbus SYSFAIL line (via VIC).
14. Initialize and/or test off card devices as necessary.
15. Enable system interrupts, as desired.

13.2 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 for a pictorial description of the system memory map.

TABLE 13-1
Address summary

Address (Hexadecimal)	Type	Device	Reference Section
FF08,0000 – FFFF,FFFF	-	Reserved	
FF07,0000 – FF07,FFFF	R/W	Module Configuration	12
FF06,0000 – FF06,FFFF	R/W	Module Peripherals	12
FF05,0000 – FF05,FFFF	R/W	Module Synchronous Devices	12
FF04,0000 – FF04,FFFF	R/W	VMEbus Short Space	7.5, 7.7
FF03,0000 – FF03,FFFF	-	Reserved	
FF02,0000 – FF02,FFFF	R/W	EEPROM	6.7
FF01,30A8 – FF01,FFFF	-	Reserved	
FF01,30A0	W	VSB Block Mode Disable	8.3.1.4
FF01,3098	W	VSB Arbiter Enable	8.3.1.2
FF01,3090	W	VSB Slave Enable	8.3.1.3
FF01,3088	W	VSB Bus Mode Bit 1	8.3.1.1
FF01,3080	W	VSB Bus Mode Bit 0	8.3.1.1
FF01,3040	W	VSB Compare Address	8.3.1.3
FF01,3000	W	VSB Local Address	8.3.1.3
FF01,2018 – FF01,2FFF	-	Reserved	
FF01,2010	W	VMEbus Mailbox Base Address	7.5, 7.8
FF01,2008	W	VMEbus Base Address	7.8
FF01,2000	W	VMEbus Local Address	7.8
FF01,0D00 – FF01,1FFF	-	Reserved	
FF01,0C00	RMW	EEPROM Write Enable	6.7
FF01,0B00	RMW	Parity Inversion Enable	
FF01,0A00	R	Parity Error Clear	
FF01,0928 – FF01,09FF	-	Reserved	
FF01,0918	W	User LED4	9
FF01,0910	W	User LED3	9
FF01,0908	W	User LED2	9
FF01,0900	W	User LED1	9
FF01,0800	R/W	TCP Registers	11.1
FF01,0700	W	TCP Cascade	11.5
FF01,0100 – FF01,06FF	-	Reserved	
Continues.			

TABLE 13-1 — *Continued.*
Address summary

FF01,0000	R/W	SCC Registers	10.5, 10.6
FF00,0100 – FF00,FFFF	-	Reserved	
FF00,0000 – FF00,00FC	R/W	VIC Registers	7.3
FE00,0000 – FEFF,FFFF	R/W	VMEbus Standard Space	7.8.2
FC40,0000 – FDFF,FFFF	-	Reserved	
FC00,0000 – FC3F,FFFF	R	ROM	6.2
4000,0000 – FBFF,FFFF	R/W	VMEbus Extended Space	7.8.1
0020,0000 – 3FFF,FFFF	R/W	VSB System Address Space	8.3.1
0000,0000 – 001F,FFFF	R/W	RAM	6.3

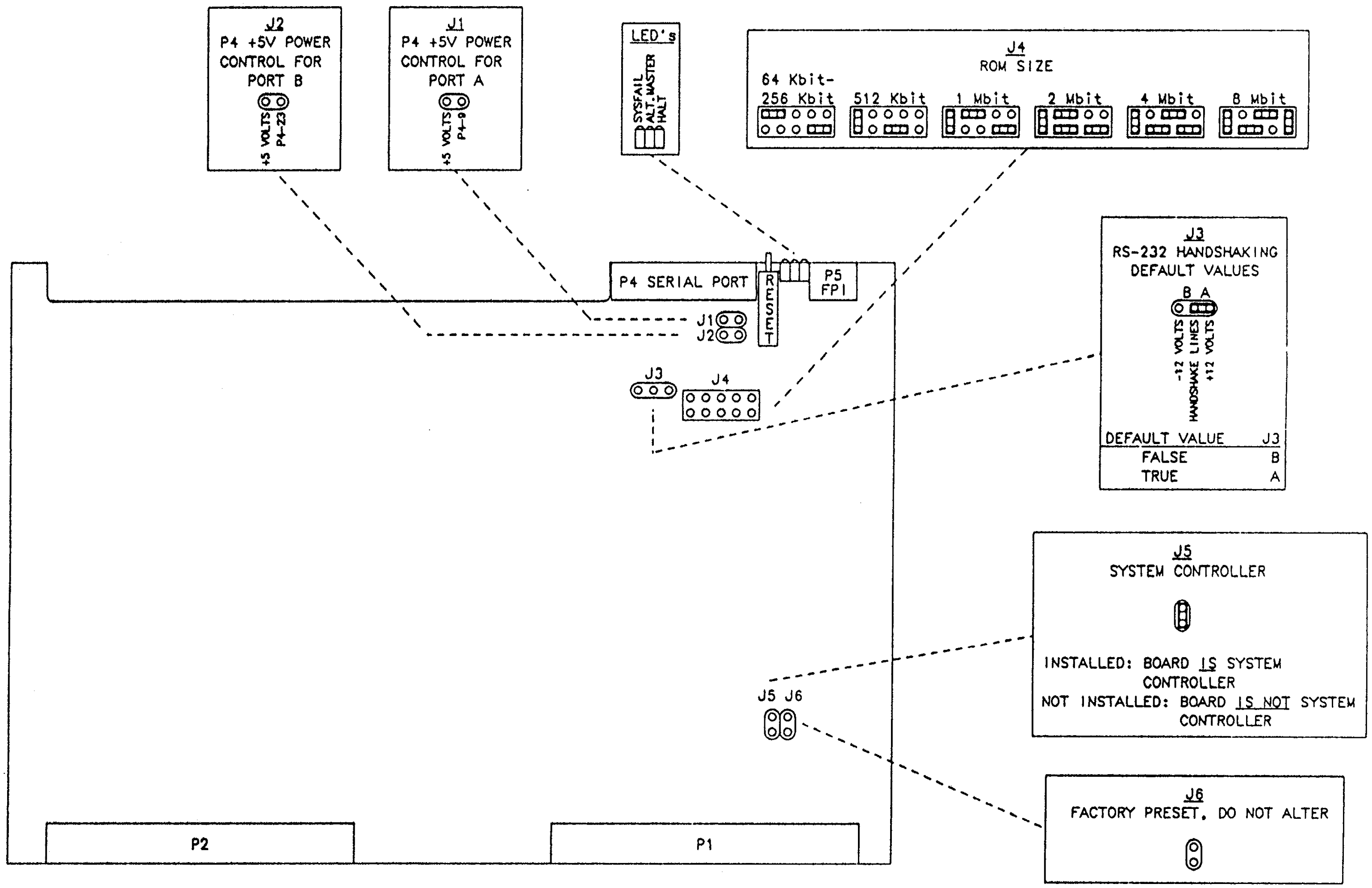
Note: RAM extends from 0000,0000$_{16}$ to 007F,FFFF$_{16}$ and VSB from 0080,0000$_{16}$ to 3FFF,FFFF$_{16}$ on an 8-Mbyte board.

13.3 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.

TABLE 13-2
Jumper summary

Jumper	Function	Reference Section	Standard Configuration
1	+5V power for Port A	10.9	Not installed.
2	+5V power for Port B	10.9	Not installed.
3	RS-232 handshaking A: +12V B: -12V	10.2	J3-A True J3-B False
4	ROM size	6.2	512 Kbit
5	System controller	7.6	Not installed.



536D002-C

FIGURE 13-1. HK68/V3F jumper configurations

13.4 POWER REQUIREMENTS

TABLE 13-3
Power requirements

Voltage	Current	Usage
+5	9 A	All logic
+12	.5 A	RS-232 interface
-12	.5 A	RS-232 interface

All "+5" and "Gnd" pins on P1 **and** P2 *must* be connected to assure proper operation.

13.5 ENVIRONMENTAL REQUIREMENTS

Operating temperature: 0 to +55 degrees Centigrade, ambient, at board.
Humidity: 0% to 85%
Storage temperature: -40 to +70 degrees C
Typical power dissipation: 57 W

FAN COOLING IS REQUIRED for the HK68/V3F board whenever power is applied, even when the board is on an extender card.

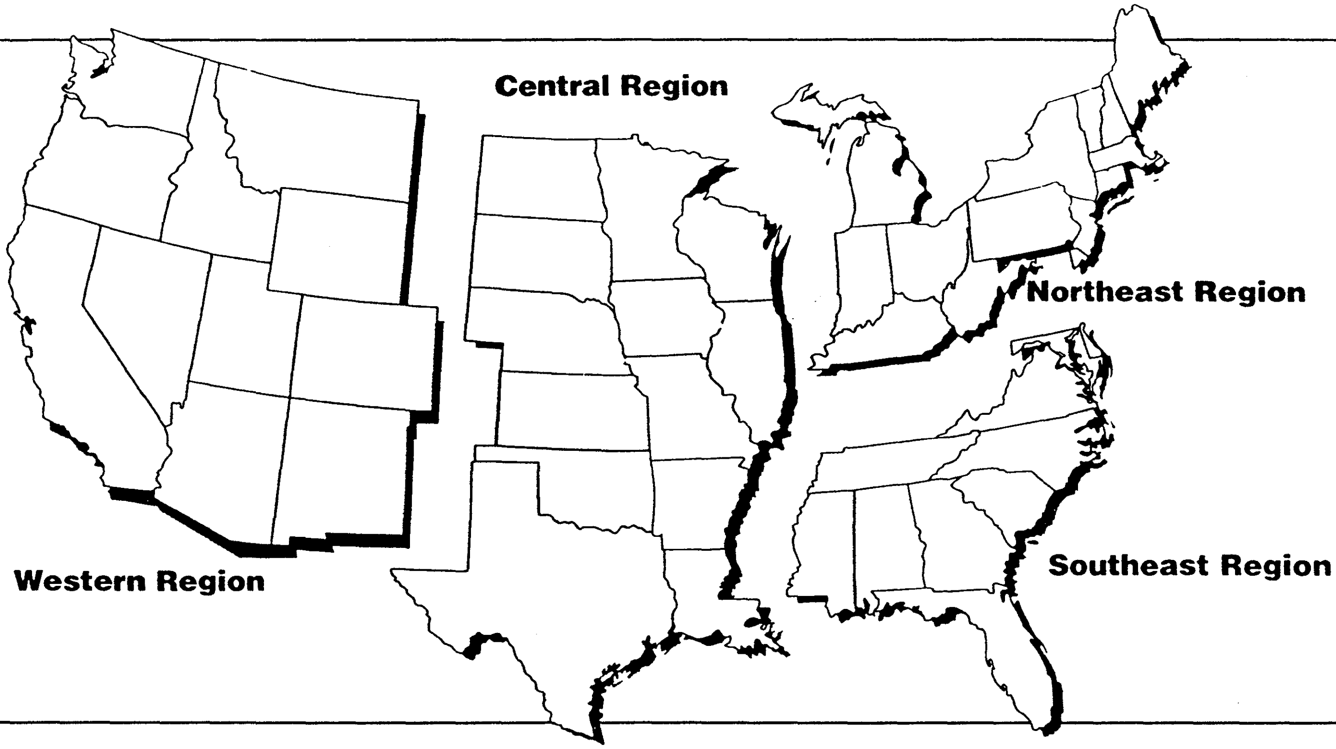
13.6 MECHANICAL SPECIFICATIONS

TABLE 13-4
Mechanical specifications

Width	Depth	Height
9.187 in.	6.299 in.	0.6 in.
233.35 mm	160 mm	15.25 mm

Standard board spacing is 0.8 inches. The HK68/V3F is a 10-layer board.

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
16496 Bernardo Center Drive, Suite 213
San Diego, CA 92128
Phone: 619-487-9771
Fax: 619-487-2562

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

Systems Components, Inc.
1327 Jones, Suite 104
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) 424-7342

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, #203
Rockville, MD 20850
Tel: (301) 294-9770
Fax: (301) 294-9772

• WESTERN REGION

AZ, CO, NV, NM and
UT

Compware Marketing
100 Arapahoe Ave.
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

Qualtech
333 West Maude Avenue,
Suite 108
Sunnyvale, CA 94086
Tel: (408) 732-4800
Fax: (408) 733-7084

Index

Heurikon Corporation
8310 Excelsior Drive
Madison, WI 53717
Customer Support: 1-800-327-1251