USER'S MANUAL
Revision J
May 1988

HK68/V20/V2F/V2Fa

VMEbus 68020-based Single Board Computer

HEURIKON CORP.®
OPEN SYSTEMS : : OPEN TOOLS

HK68/V20/V2F/V2Fa

VMEbus 68020-based Single Board Computer

USER'S MANUAL
Revision J
May 1988

HEURIKON® CORP.
OPEN SYSTEMS : : OPEN TOOLS

Heurikon HK68/V20 - User's Manual
Heurikon Corporation
Madison, WI

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

The purpose of this manual is to document the features of the Heurikon HK68/V20 (tm) (and HK68/V2F and HK68/V2FA) microcomputer boards. Unless stated otherwise, items discussed in this manual apply to all three boards even though only the HK68/V20 may be mentioned.

This manual covers the unique features of the HK68/V20 board. Although general information on the MPU, FPP and MFP is given, details should be obtained from the chip manufacturers data sheets.

Feel free to contact Heurikon Corporation (Customer Support Department) if questions arise. We are prepared to answer general questions as well as help with specific applications.

1.1 Disclaimer

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.

---

1 HK68, HK68/V20 and Hbug are trademarks of Heurikon Corporation.
2 This document was prepared using the UNIX nroff facility and the PWB/mm macros.
3 UNIX is a trademark of AT&T Bell Laboratories.

2. HK68/V20 FEATURE SUMMARY

MPU	Motorola 68020 microprocessor chip; 12.5 Mhz; 32-bit internal architecture, 32-bit address and data paths; 32 address lines; 4 gigabyte addressing range; 256-byte Instruction Cache. (Ref: section 5)
FPP	68881 Floating Point Co-processor. Implements the IEEE-P754 Binary Floating Point Standard. (Ref: section 6)
PMMU	(HK68/V20 only) Motorola 68851 chip (or equiv). Provides logical to physical address translation. Demand Paged Virtual Memory operation. (Ref: section 7)
RAM	Four megabyte capacity; One parity bit per byte; Hardware refresh. (Ref: section 9)
EPROM	Two ROM sockets; one socket on the HK68/V2FA. 128 Kbyte total capacity. Page Addressable ROM or EEpROM capability. (Ref: section 9)
VMEbus	32-bit addressing (4 gigabyte range); 32-bit data bus, compatible with 8-bit boards; Seven bus interrupts. (Ref: section 10)
VSB	High speed local memory expansion. Supports secondary bus masters. (Ref: section 10)
Serial I/O	One serial I/O port via the MFP chip; Internal baud rate generator; Asynchronous, synchronous modes; RS-232C interface, RS-422 (optional via external cable). (Ref: section 11.3)
MFP	Mostek MK68901 (or equiv) Multi-Function Peripheral chip; On-card interrupt controller; Four timers; Serial I/O controller; (Ref: section 11)
NV-RAM	Nonvolatile Static RAM; 256 x 4 configuration; Internal EEpROM; 100 year retention; 10,000 store cycle lifetime; For user definable functions. (Ref: section 9.7)
Mailbox	Allows remote control of the HK68/V20 via specified VMEbus addresses; MPU halt, reset, interrupt, and on-card bus lock functions. (Ref: section 10.8)
RTC	Optional Real-Time Clock module for time-of-day maintenance. With battery backup. (Ref: section 12)

3. BLOCK DIAGRAM

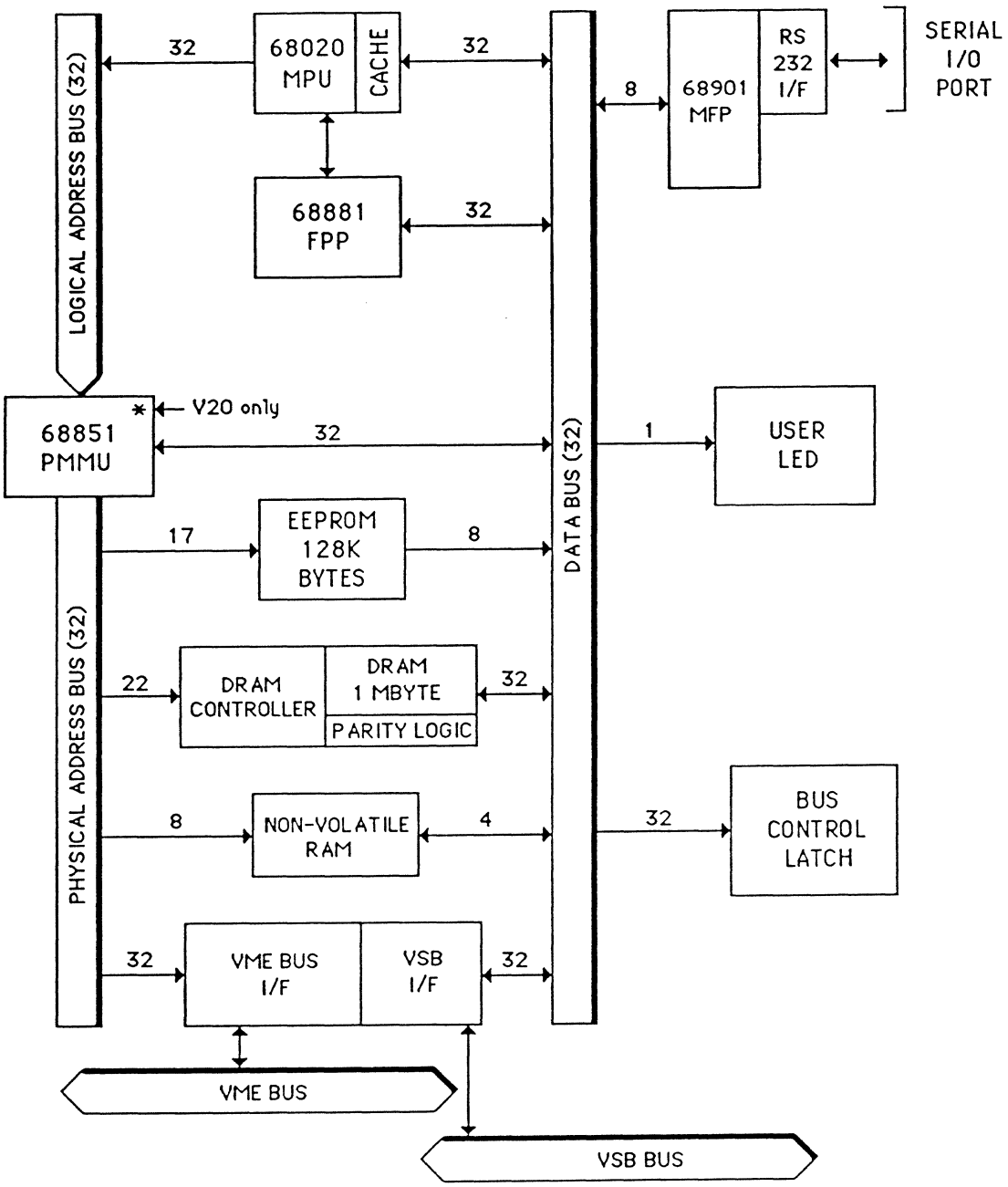


Figure 1. HK68/V20 Block Diagram

4. GETTING GOING

Here is what you need to get the Heurikon HK68/V20 "on-the-air":

- Heurikon HK68/V20 Microcomputer board
- Card cage and power supply
- Serial I/F cable (RS-232)
- CRT Terminal
- Heurikon Hbug monitor and bootstrap EPROM

4.1 Installation Steps

- CAUTION: All semiconductors should be handled with care. Static discharges can easily damage the components on the HK68/V20. Keep the board in an anti-static bag whenever it is out of the system chassis and do not handle the board unless absolutely necessary. Ground your body before touching the HK68/V20 board.
- CAUTION: High operating temperatures will cause unpredictable operation. Because of the high chip density, fan cooling is required for most configurations, even when cards are placed on extenders.

All products are fully tested before they are shipped from the factory. When you receive your HK68/V20, follow these steps to assure yourself that the system is operational:

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

[2] Connect a CRT terminal to the serial port, via connector P4. If you are making your own cables, refer to section 11.3. Set the terminal as follows:

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

[3] Connect AC power and turn the system on.

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

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

[6] Reconfigure the jumpers, etc, as necessary for your application. See Section 14 for a summary of I/O device addresses and configuration jumpers.

4.2 Troubleshooting and Service Information

In case of difficulty, use this checklist:

[1] Be sure the system is not overheating.

[2] Inspect the power cables and connectors. If the HK68/V20 board has power, the large chips should feel warm to your touch.

[3] If the Hbug monitor program is executing, run the memory diagnostics, via command 'um'.

[4] Check your power supply for proper DC voltages. If possible, look for excessive power supply ripple or noise using an oscilloscope. 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 EPROMs.

[6] Check your terminal switches and cables. Be sure the P4 connector is on properly. If you have made your own cables, pay particular attention to the cable drawings in section 11.3.

[7] Check the jumpers to be sure your board is configured properly. All jumpers should be in the "standard configuration" positions shown in section 14.3.

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

- The monitor program revision level (part of sign-on message).
- The HK68/V20 p.c.b. serial number (scribed along card edge).

- The complete HK68/V20 model number, including option codes.

- The serial number of the Operating System.

If you plan to return the board to Heurikon for service, contact our Customer Service Department to obtain a Return Merchandise Authorization (RMA) number. Be prepared to provide the items listed above, plus your Purchase Order number and billing information if your HK68/V20 is out of warranty. If you return the board, be sure to enclose it in the anti-static bag, such as the one in which it was originally shipped. Send it prepaid to:

Heurikon Corporation
Factory Service Department

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

4.3 Monitor Summary

The HK68/V20 monitor and bootstrap program, Hbug, is contained in one EPROM. It is intended to provide a fundamental ability to check the memory and I/O devices, to manually enter a program and to down-line load or bootstrap a larger program into memory. Advanced features and utilities may be loaded from media or via an operating system.

Refer to the Hbug manual for details on the commands and command formats.

5. MPU SUMMARY INFORMATION

This section details some of the important features of the 68020 MPU chip and, in particular, those items which are specific to the implementation on the Heurikon HK68/V20.

5.1 MPU Interrupts

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

Level	Interrupt	Interface
7	Parity error, ACFAIL, non-maskable (autovectored)	P1-B3
6	VMEbus interrupt level 6	P1-B25
5	VMEbus interrupt level 5	P1-B26
4	MFP Interrupt (See section 11)	MFP
3	VMEbus interrupt level 3	P1-B28
2	VSB Interrupt (IRQ) (autovectored)	P2-A18
1	VMEbus interrupt level 1	P1-B30
0	Idle, no interrupt	

Table 1. MPU Interrupt Levels

When an interrupt is recognized by the MPU, the current instruction is completed and an interrupt acknowledge sequence is initiated, whose purpose is to acquire an interrupt vector from the interrupting device. The vector number is used to select one of 256 exception vectors located in reserved memory locations (see section 5.2 for a listing.) The exception vector specifies the address of the interrupt service routine.

The MFP device on the HK68/V20 is capable of generating more than one vector, depending on the particular condition which caused the interrupt. This significantly reduces the time required to service the interrupt because the program does not have to rigorously test for the interrupt cause. Section 10.5 has more information on the HK68/V20 interrupt logic.

5.2 MPU Exception Vectors

Exception vectors are memory locations from which the MPU fetches the address of a routine to handle an exception (interrupt). All exception vectors are two words long (four bytes), except for the reset vector which is four words. The listing below shows the vector space as it appears to the Heurikon HK68/V20 MPU. It varies slightly from the 68020 MPU manual listing due to particular implementations on the HK68/V20 board. Refer to the MPU documentation for more details. The vector table normally occupies the first 1024 bytes of memory, but may be moved to other locations under software control. Unused vector positions may be used for other purposes (e.g., code or data) or point to an error routine.

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

Table 2. MPU Exception Vectors

Autovectoring is used for the parity error, ACFAIL and VSB interrupts. Interrupts from all other devices can be programmed to provide a vector number (which would likely point into the "User Interrupt Vector" area, above).

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

Level	Vector	Device	Condition
7	31	Memory	Parity error autovectored interrupt
		VMEbus	VMEbus ACFAIL* signal active
6	vv	VMEbus	Interrupt level 7
5	vv	VMEbus	Interrupt level 5
4	79	MFP	VMEbus Interrupt 7
	78		VMEbus Interrupt 6
	77		Timer A
	76		Receive Buffer Full
	75		Receive Error
	74		Transmit Buffer Empty
	73		Transmit Error
	72		Timer B
	71		Mailbox Interrupt
	70		VMEbus Interrupt 4
	69		Timer C
	68		Timer D
	67		VMEbus Interrupt Pending
	66		VMEbus Interrupt 2
	65		VMEbus ACFAIL
	64		VMEbus SYSFAIL
3	vv	VMEbus	Interrupt level 3
2	26	VSB	VSB autovectored interrupt
1	vv	VMEbus	Interrupt level 1

Table 3. Suggested Interrupt Vectors

The vector numbers "vv" are delivered by the interrupting device, which is another board on the VMEbus.

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

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

Device	Hex Value	Decimal Value
MFP	0x40	64

Table 4. Device Interrupt Vector Values (Suggested)

5.3 Status LEDs

There are three status LEDs which continuously show the state of the board as follows:

LED	Name	Meaning
L1	Fail	The SYSFAIL line is being driven active by this board.
L2	Master	The HK68/V20 is the master on the VMEbus.
L3	Slave	The HK68/V20 is a slave on the VMEbus.
L4	User	User LED (see section 10.10)

Table 5. Status LEDs

5.4 Control Panel Interface/MPU Status

There are four status outputs which allow remote monitoring of the HK68/V20 processor. Connections are made through a 10 pin connector, P3.

P3 pin	Name	Meaning
2	Supr	The MPU is in the supervisor state.
4	User	The MPU is in the user state.
6		n/c
8	Halt	The MPU has halted. (Double bus fault, odd stack address or the system reset line is active.)
10	Bus	Another VMEbus master has control of the local bus.
1,3,5,7,9	Vcc	Vcc (+5) volts

Table 6. Control Panel Interface (P3)

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

Recommended mating connectors for P3 are Ansley P/N 609-1001CE and Molex P/N 15-29-8108.

5.5 Co-Processors

The HK68/V20 supports the PMMU and FPP coprocessors. The HK68/V2F and HK68/V2FA support the FPP coprocessor only.

Co-Proc ID	Device	Function	Reference Section
0	68851	Paged Memory Management Unit (PMMU)	7
1	68881	Floating Point Coprocessor (FPP)	6

Table 7. 68020 Coprocessor ID Codes

6. FLOATING POINT CO-PROCESSOR (FPP)

The HK68/V20 allows the use of an optional MC68881 (or 68882) floating point processor chip. It runs as a coprocessor with the MPU.

6.1 FPP Feature Summary

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

FPP programming details are available in the 68881 technical manual.

6.2 FPP Bypass

The HK68/V20 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 Bus Control Latch), the software can determine if the FPP chip is installed. An attempt to access a non-existent FPP will result in a Watchdog timeout and a Bus Error, forcing a Line 1111 MPU Exception, vector number 11.

7. MEMORY MANAGEMENT CO-PROCESSOR (PMMU)

(Not supported on the HK68/V2F or HK68/V2FA.)

This section explains some of the relevant features of the 68851 PMMU chip. Refer to the PMMU technical manual for more details.

The PMMU operates as a coprocessor with the MPU.

The PMMU automatically enters a "transparent" mode following a system reset. Thus, all logical addresses and physical address will be the same. The PMMU must be programmed and enabled before any address translations will begin.

7.1 Function Code Definitions

The table below shows the MPU and FPP function codes which are generated for each memory reference. They indicate to the PMMU the particular type of reference being made, and are used to index into the PMMU Address Space Table (AST). Ultimately, the function codes determine the logical to physical mapping and the protection levels for the operation (e.g., write protect, user/supervisor space).

Hex	FC3	FC2	FC1	FC0	
0	0	0	0	0	(reserved)
1	0	0	0	1	User DATA
2	0	0	1	0	User PROGRAM
3	0	0	1	1	(reserved)
4	0	1	0	0	(reserved)
5	0	1	0	1	Supervisor, DATA
6	0	1	1	0	Supervisor, PROGRAM
7	0	1	1	1	CPU Space (FPP, PMMU)
8-F	1	x	x	x	Not used

Table 8. Function Code Assignments

7.2 PMMU Address Line Block Diagram

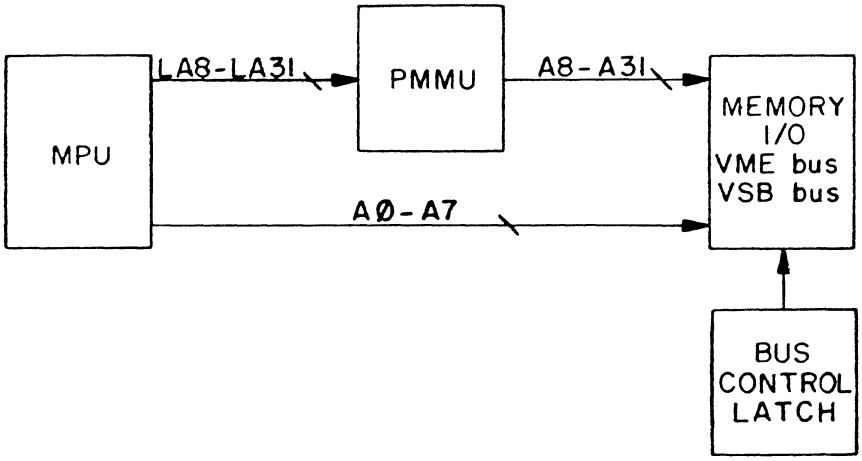


Figure 2. MPU Accesses to Memory or Bus

7.3 PMMU Bypass

The HK68/V20 will operate without an PMMU chip. If the PMMU is removed from the board, the following jumpers must be installed in its place. (A pre-wired bypass header is available from Heurikon.)

Signal	Connect Pins	Signal	Connect Pins
LA31	L3 and C5	LA30	M2 and A3
LA29	M3 and A2	LA28	L4 and B4
LA27	M4 and C4	LA26	N2 and B3
LA25	N3 and B2	LA24	L5 and D4
LA23	M5 and A1	LA22	N4 and C2
LA21	N5 and D3	LA20	L6 and D2
LA19	M6 and B1	LA18	N6 and C1
LA17	N7 and E3	LA16	N8 and E2
LA15	M8 and D1	LA14	L8 and E1
LA13	N9 and F3	LA12	N10 and F2
LA11	M9 and F1	LA10	L9 and G1
LA9	N11 and H1	LA8	N12 and H2
LBRI-0	A7 and A5	LAS	A4 and B5
LBGI-0	A6 and C6	PBR-PBG	B9 and A9

Table 9. PMMU Bypass Connections

If the Watchdog timer is enabled, the software can determine if the PMMU chip is installed. Any attempt to access a non-existent PMMU will result in a Watchdog timeout and thus a Bus Error, forcing a Line 1111 MPU exception, vector number 11.

7.4 Alternate Capabilities - MMB

In some applications, the MMB daughter board may be used instead of the 68851 PMMU. If you are using the MMB, it is addressed as follows:

Address	MMB Register (read/write)
00FE,6000	Root Pointer (RP)
00FE,6004	Translation Control Register (TC)

Table 10. MMB Registers

The MMB registers are in the MMB physical address space. Therefore, take care to assure that the MMB is always mapped to a logical block. Both registers are 32 bits. Refer to the MMB manual for programming details.

Since the MMB is a piggy-back board, one or two extra card slots are required in a standard rack.

8. SYSTEM ERROR HANDLING

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

8.1 Error Conditions

The following error conditions may arise during MPU cycles:

Condition	Meaning
RAM Parity	Incorrect parity was detected during a read cycle from on-card RAM memory. This may be due to a true parity error (RAM data changed,) or because the memory location was not initialized prior to the read and contained garbage. Parity errors generate a level 7 autovector interrupt. A pointer to the parity error handling routine should be loaded at location 00007C. Parity checking cannot be disabled.
Watchdog Timeout	During an access, usually to the bus, no acknowledge was received within a fixed time interval defined by a hardware timer. (1.64 milliseconds.) This is usually the result of no bus device being assigned to the specified bus address. A timeout could also occur if an access from the bus is not terminated by the bus master. For an access to the bus, the memory cycle is terminated, the BERR (Bus Error) exception is taken by the MPU and execution resumes at the location specified by the exception vector. If an access from the bus was in progress, no BERR exception occurs.
Double Bus Fault	Another bus error occurred during the processing of a previous bus error, address error or reset exception. This error is the result of a major software bug or a hardware malfunction. A typical software bug which could cause this error would be an improperly initialized stack pointer, which points to an invalid address. A double bus fault forces the MPU to enter the HALT state. Processing stops. The HALT status LED will come on. The only way out of this condition is to issue a hardware reset.

MMU Fault	(HK68/V20 only.) The MMU has detected a write violation or an undefined segment address. The memory cycle is terminated and the bus error exception is taken.
Divide by Zero	The value of the divisor for a divide instruction is zero. The instruction is aborted and vector 5 is used to transfer to an error routine.
Privileged Violation	A program executing in the user state attempted to execute a privileged instruction. The instruction is not executed. Exception vector 8 is used to transfer control.
Address Error	An odd address has been specified for an instruction. The bus cycle is aborted and vector 3 is used to transfer control.
Illegal Instruction	The bit pattern for the fetched instruction is not legal or is unimplemented. The instruction is not executed. Exception vector 4, 10 or 11 is used to transfer control.
Format Error	The format of the stack frame is not correct for an RTE instruction. The instruction is aborted and exception vector 14 is used to transfer control.
Line 1111 Emulator	The FPP or PMMU Coprocessor is not present and a coprocessor instruction was fetched. The instruction is not executed. Exception vector 11 will be taken.
FPP Exceptions	The FPP Coprocessor has detected a data processing error, such as an overflow or a divide by zero. The FPP causes the MPU to take one of eight exceptions in the range of 48 to 54.

(HK68/V20 only.) As the above list indicates, there are two causes for a bus error exception. In order to determine the cause of a bus error exception, test the fault status bits in the MMU. If the MMU indicates no fault then the bus error was caused by the watchdog timer.

9. ON-CARD MEMORY CONFIGURATION

The Heurikon HK68/V20 microcomputer will accommodate a variety of RAM and ROM configurations. There are two ROM sockets (one on the V2FA) for pROM, page addressable ROM or EEpROM, four SIP (or 36 ZIPs, on the V2FA) RAM positions, and a nonvolatile RAM. Off-card memory may be accessed via the VMEbus or the VSB.

9.1 ROM

Each ROM occupies a fixed 64K byte physical address space. At power-on, the MPU fetches the reset vector from the first eight locations of ROM 0. The reset vector specifies the initial program counter and status register values. ROM access time must be 435 nsec or less.

Base Address	ROM	Chip	Mode
0000,0000	0	U23	read only
0001,0000	1	U24	read only
0002,0000	1	U24	write only

Table 11. ROM Address Summary (V20 and V2F)

Base Address	Chip	Mode	Size
0000,0000	U15	read only	128 Kbytes
0002,0000	U15	write only	64 Kbytes

Table 12. ROM Address Summary (V2FA)

There are four jumpers which must be set according to the ROM type being used. On the V20 and V2F, if more than one ROM is used, they do not have to be of the same type.

EPROM Type	ROM Capacity	Total Board Capacity	Jumper Positions U24 (ROM 1)	U23 (ROM 0)
2764	8 Kbytes	16 Kbytes	J1-B J6-B	J2-B J7-B
27128	16 Kbytes	32 Kbytes	J1-B J6-B	J2-B J7-B
27256	32 Kbytes	64 Kbytes	J1-A J6-B	J2-A J7-B
27512	64 Kbytes	128 Kbytes	J1-A J6-A	J2-A J7-A
27513 Paged	64 Kbytes	128 Kbytes	J1-C J6-A	not supported
27516 Key ROM	8 Kbytes	128 Kbytes	J1-C J6-A	J2-C J7-A
2864 R/W EEpROM	8 Kbytes	16 Kbytes	J1-C J6-open	not supported

Table 13. ROM Capacity and Jumper Positions (HK68/V20 and V2F)

EPROM Type	ROM Capacity	------Jumper Positions------ J1	J2	J6	J7
2764	8 Kbytes	B	B	B	B
27128	16 Kbytes	B	B	B	B
27256	32 Kbytes	A	B	B	B
27512	64 Kbytes	A	B	A	B
27010	128 Kbytes	A	A	A	A
27513 Paged	64 Kbytes	C	B	A	B
27916 R/W Key ROM	8 Kbytes	C	B	A	B
2864 R/W EEpROM	8 Kbytes	C	B	open	B

Table 14. ROM Capacity and Jumper Positions (HK68/V2FA)

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

On the HK68/V2FA, the rom socket is 32 pins. When using a 28 (or 24) pin device, justify it so socket pins 1, 2, 31 and 32 (or those plus 3, 4, 29 and 30) are empty.

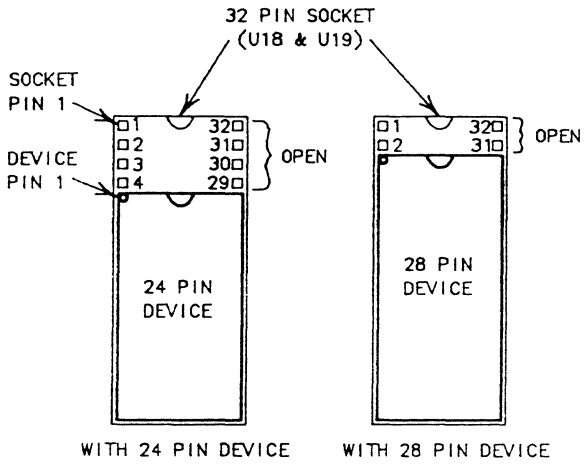


Figure 3. ROM Positioning Diagram (for HK68/V2FA only)

On the V20 and V2F, when ROM 0 is smaller than 64 KBytes, the two ROM positions are not contiguous (although a mirror of the lower ROM will be contiguous with the upper ROM). The best way to create a contiguous image is to copy the contents of both ROMs to contiguous RAM areas.

Electrically Erasable or paged pROMs may be An EEpROM allows specific addresses to be changed by writing to the ROM. used. (In the second ROM position only on the V20 and V2F). A different base address value is used for the EEpROM depending on whether you are reading or writing. 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 16K byte pages. A specific page is selected by writing the page value to the ROM. For example, to select page three of a 27513 in the ROM 1 position, write 0x03 to address 0002,0000.

9.2 On-Card RAM

9.2.1 HK68/V20 and HK68/V2F RAM Configuration

The HK68/V20 and HK68/V2F use four SIP RAM packages, each nine bits wide. There is one parity bit per byte. Standard memory configurations are 256K bytes, 1 megabyte or 4 megabytes. Two card slots are required for the

four megabyte version. On-card RAM occupies physical addresses starting at 0200,0000.

RAM Type	Quantity	Capacity
64K x 9 SIP	4	256 Kbytes
256K x 9 SIP	4	1024 Kbytes
1Meg x 9 SIP	4	4096 Kbytes

Table 15. On-card RAM Capacity (HK68/V20 and V2F)

9.2.2 HK68/V2FA RAM Configuration

The HK68/V2FA board uses 36 ZIP RAM packages, each one bit wide. This results in a total capacity of four megabytes, plus one parity bit per byte.

9.3 On-card Memory Sizing

(Not useful on the HK68/V2FA.) The following algorithm can be used to determine the amount of on-card RAM memory installed. This procedure takes advantage of "mirrors" which exist in higher addresses when the on-card physical memory size is less than the logical memory space.

[1] Clear 16 megabytes of memory starting at location 0200,0000.

[2] Write 5555 (hex) to location 0200,0000.

[3] Read a word from 0204,0000. If the value read is 5555 the board has 256 Kbytes of memory installed. If the value is zero, continue.

[4] Read a word from 0210,0000. If the value read is 5555 the board has one megabyte of memory installed. If the value is zero, continue.

[5] Read a word from 0240,0000. If the value read is 5555 the board has four megabytes of memory installed. If the value is zero, the board has 16 megabytes of memory.

9.4 Bus Memory

See section 10 for details concerning the bus interface.

9.5 Physical Memory Map

See section 14.2 for an I/O device address summary.

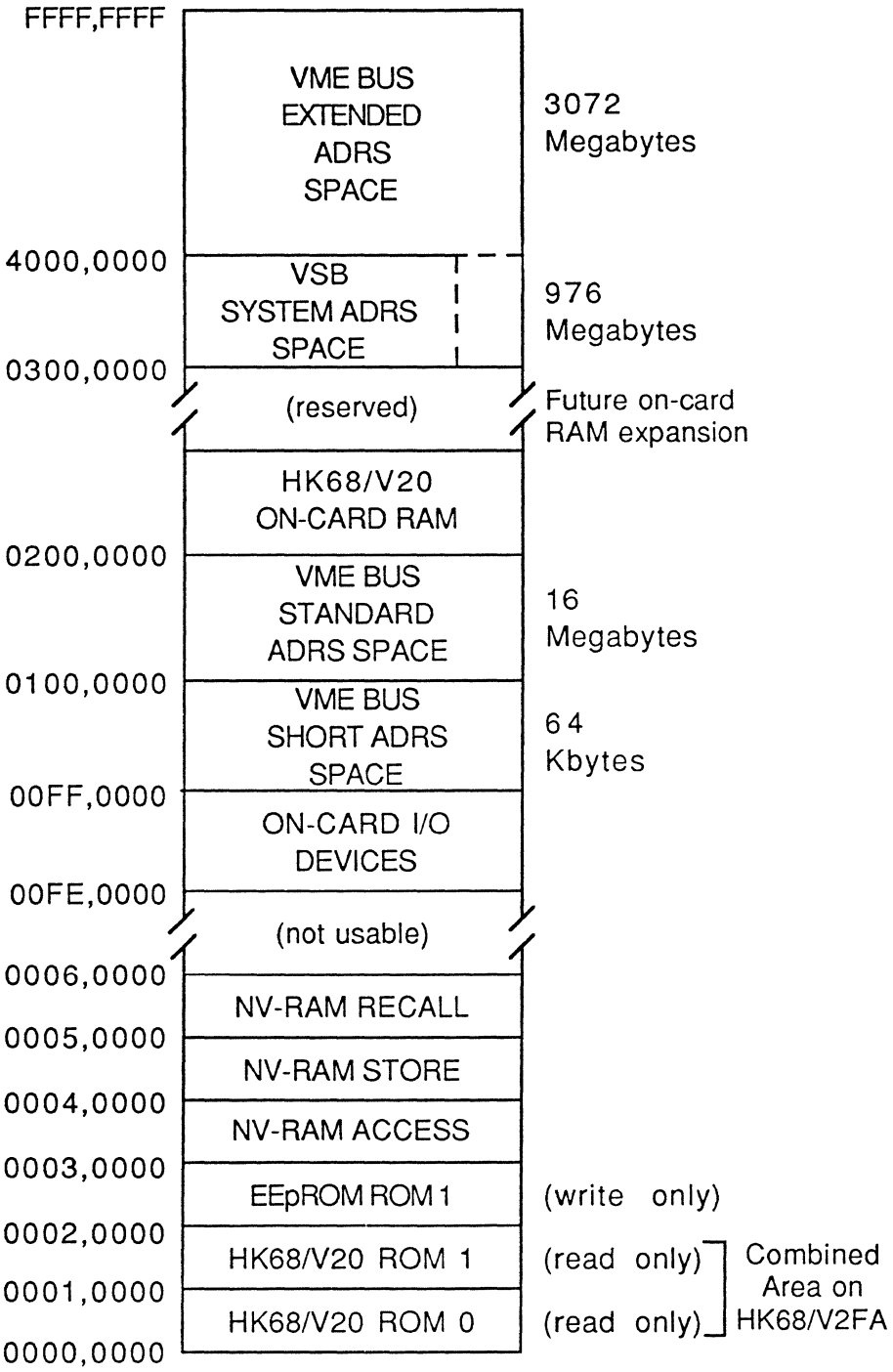


Figure 4. Physical Memory Map

9.6 Memory Timing

The HK68/V20 memory logic has been carefully tuned to give optimum memory cycle times under a variety of conditions. Considerations have been given to these factors:

[1] The MMU, if present, delays the generation of stable physical addresses on MPU accesses. (Translation Time.)

[2] Typical access times for ROMs are 100 to 200 nanoseconds longer than RAM. Since most programs will be in RAM (or could at least be copied to RAM for execution), ROM timing need not be optimized.

[3] Dynamic memory refreshing must be fast enough that a lengthy (or infinite) bus access cycle will not cause loss of the RAM contents. If a long access from the bus to on-card RAM occurs, which would be terminated by the Watchdog Timer, refreshing must resume and a complete refresh cycle must be done before the maximum refresh time allowed by the RAMs expires. Refreshing operates normally during accesses from the VMEbus which are redirected to the VSB, and all accesses to the bus. The HK68/V20 uses a hardware refresh.

Depending on the RAM speed, extra clock cycles are inserted in memory references to synchronize the MPU with the MMU and memory. The number of extra clock cycles required (in addition to the cycles built into the instruction timing charts) for a (RAM) memory read are shown below.

Condition (120 nsec. RAM speed)	Extra Cycles (at 16.67 MHz)	Total Cycles (at 16.67 MHz)
MPU on-card RAM access, no MMU	1	4
MPU on-card RAM access, with MMU	2	5

Table 16. On-card Memory Cycle Timing (RAM)

As faster DRAMs become available, the number of wait states will be reduced, according to the following chart (assumes no MMU).

	Required DRAM Speeds for		
MPU Speed	Zero Waits (r/w)	Zero Waits (read)	One Wait (read)
20.0 MHz		60 nsec. (est)	110 nsec. (est)
16.67 MHz		75 nsec. (est)	135 nsec.
12.5 MHz	80 nsec.	100 nsec.	180 nsec.

The following chart can be used to estimate relative MPU/RAM performance based on the MPU speed, RAM access time and percentage of cache hits. The first column of figures is the performance value if there are no memory accesses, i.e., the cache is hit 100% of the time. The remaining columns show the performance figures for various RAM speeds. The "100% Cache Hits" column shows the maximum performance; the other columns show the minimum performance. The actual value depends on the actual cache hit ratio; your mileage may vary. The chart takes into account the number of

wait states required to access the RAM.

Assumptions: No MMU installed. With an MMU, there will be an additional wait state for each RAM access. All RAM cycles are reads; in some cases, there is an additional wait state for a write.

Computations: Performance Value = MPU.Clock.Rate / Cycles.per.access
Cycles.per.access = 3 + wait.cycles.per.access

	100%	---------RAM SPEED---(no cache hits)----------			
MPU Speed	Cache Hits	60 nsec.	100 nsec.	120 nsec.	150 nsec.
25 MHz	8.33	6.25	(5.00)	5.00	4.17
20 MHz	6.67	6.67	(5.00)	5.00	4.00
16.67 MHz	5.33	5.33	(4.00)	4.00	3.20
12.5 MHz	4.17	(4.17)	4.17	(3.13)	3.13

Table 17. Relative MPU/RAM Performance Figures

Since the performance figures are anchored at zero, the performance of a system with a value of eight will be twice that of a system with a value of four. The numbers in parenthesis e.g, "(5.00)", represent a cell in the chart which is better implemented (less costly) by using the next slower RAM speed.

The user must weigh the tradeoffs between performance and the costs of faster MPU or memory.

The HK68/V20 uses hardware logic to control refreshing of the dynamic memory. The refresh clock runs at is 76,800 Hz. Thus, one row of the RAM array is refreshed every 13 microseconds. Worst case conditions result in a speed penalty of about 1.5% to accommodate the refresh cycles.

Memory timing is controlled by jumpers J3 and J4, which select the proper delays for DRAM address multiplexing, RAS/CAS timing and DTACK response. These jumpers are factory set; please don't fiddle with them.

9.7 Non-Volatile RAM

A unique feature of the HK68/V20 is its non-volatile RAM (NV-RAM), which allows precious data or system configuration information to be stored and recovered across power cycles. The RAM is configured as 256, four-bit words (low half of a byte). When the MPU reads a byte of data from the NV-RAM, the upper four bits of the value it receives are indeterminate. The NV-RAM is accessible as shown below.

Address	Mode	Function
0003,00xx	R/W	Read/Write RAM contents (4 bits).
0005,0000	Read	Recall RAM contents from Non-volatile memory.
0004,0000	Write	Store RAM contents in Non-volatile memory. The 68020 "tas" (test and set) instruction must be used for this operation.

Table 18. Non-Volatile RAM Addresses

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

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

The HK68/V20 monitor (Hbug) and certain system programs use the NV-RAM. The exact amount reserved for Heurikon usage depends on the system. A major portion of the RAM, however, is available for customer use. Heurikon usage is summarized below (details are available separately).

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

Table 19. NV-RAM Contents (partial)

10. VMEBUS CONTROL

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

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

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

10.1 Bus Control Signals

10.1.1 VMEbus, P1 Descriptions

The following signals on connector P1 and P2 are used for the VMEbus interface. For a complete listing of the pinouts, refer to section 13.

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

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

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

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

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

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

BBSY* — BUS BUSY. An open-collector driven signal low by the current MASTER to indicate that it is using the bus. When the MASTER releases this line, the resultant

rising edge causes the ARBITER to sample the bus grant lines and grant the bus to the highest priority requester. Early release mode is supported.

BCLR* BUS CLEAR. A totem-pole driven signal, generated by an ARBITER to indicate when there is a higher priority request for the bus. This signal requests the current MASTER to release the bus. This signal is an input to the HK68/V20. This signal is an output of the HK68/V20, associated with J28.

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

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

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

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

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

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

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

IACK* INTERRUPT ACKNOWLEDGE. An open-collector or three-state driven signal used by an INTERRUPT HANDLER

acknowledging an interrupt request. It is routed, via a backplane signal trace, to the IACKIN* pin of slot one, where it forms the beginning of the IACKIN*, IACKOUT* daisy-chain.

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

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

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

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

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

SERCLK — SERIAL CLOCK. A totem-pole driven signal which is used to synchronize the data transmission on the VMEbus. Not implemented on the HK68/V20.

SERDAT* — SERIAL DATA. An open-collector driven signal which is used for VMEbus data transmission. Not implemented on the HK68/V20.

SYSCLK — SYSTEM CLOCK. A totem-pole driven signal which provides a constant 16-MHz clock signal that is independent of any other bus timing. This signal is not used by the HK68/V20; it is an output (via J12) on the V2F and V2FA.

SYSFAIL* — SYSTEM FAIL. An open-collector driven signal that indicates that a failure has occurred in the system. Also used at power-on to indicate that at least one VMEbus board is still in its power-on initialization phase. This signal may be generated by any board on

the VMEbus. The HK68/V20 drives this line low at power-on. It is released by accessing address 00FE,2000.

SYSRESET* SYSTEM RESET. An open-collector driven signal which, when low, causes the system to be reset. This signal is associated with jumper J8.

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

+5V STDBY +5 Vdc STANDBY. This line supplies +5 Vdc to devices requiring battery backup. Not used on the HK68/V20.

10.1.2 VSB, P2 Descriptions - VME Subsystem Bus

The following signals on connector P2 are used for the VSB interface. For a complete listing of the P2 pinouts, refer to section 13. See section 10.9 for more discussion about the VSB.

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

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

SPACE0-SPACE1 VSB ADDRESS SPACE SELECT. These signals select one of four address spaces or signify an interrupt acknowledge or parallel arbitration cycle. On the HK68/V20, these signals are not used; they are driven high when the HK68/V20 is the VSB master, which selects the System Address Space.

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

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

SIZE0,SIZE1 VSB BUS SIZE. These lines, in conjunction with addresses AD00 and AD01, determine the data transfer

size and position on the data bus. Lines SIZE0 and SIZE1 are active high signals.

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

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

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

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

CACHE* VSB CACHEABLE. CACHE*, when low, indicates to the master that the selected address location is cacheable. CACHE* is asserted only by the selected VSB slave module. This signal is not used on the HK68/V20.

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

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

IRQ*	VSB INTERRUPT REQUEST. IRQ* when low indicates that a master or slave device is attempting to interrupt another master. On the HK68/V20, this signal will generate a level 2 autovectored MPU interrupt.
BREQ*	VSB BUS REQUEST. BREQ* is asserted low by a requester whenever bus mastership is required.
BGIN*	VSB BUS GRANT IN. BGIN* is an input to a requester that, when low, indicates to the requester that it has been granted the bus.
BGOUT*	VSB BUS GRANT OUT. BGOUT* is asserted low by either an arbiter or a requester to grant the bus to a requester via the BGIN* signal.
BUSY*	VSB BUS BUSY. BUSY* is asserted low by a requester that has been granted the bus to indicate ownership of the bus.
GA0-GA2	VSB GEOGRAPHICAL ADDRESSES. These lines are connected to ground on the HK68/V20; the geographical addressing feature is not implemented.

10.2 Bus Arbitration and Release

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

Normally, the VMEbus System Controller card provides the system bus clock and participates in the arbitration logic. However, if only one level of arbitration is required, a separate System Controller card is not needed. via the Bus Control Latch. The following table details the System Controller functions provided by the HK68/V20.

Function	Setting
System Clock (SYSCLK*)	V20: Not provided V2F and V2FA: J12-B (output)
Single Level (3) Arbitration	Bus Control Latch Set bit D12
System Reset (SYSRESET*)	J8-B (output)

Table 20. System Controller Functions

When the HK68/V20 is acting as a System Controller, it should be in VME slot 1.

When the HK68/V20 is acting as a System Controller, bus arbitration level three is the only level supported; be sure to set J9 and J10 according to the description below. Also, be sure the HK68/V20 is in bus slot zero.

There are four separate bus request lines on the VMEbus. Each bus request line has an associated bus grant daisy chain. These lines may be prioritized, or they may be treated with equal importance, depending on the arbitration mode selected by the System Controller board.

On the HK68/V20, one bus request line is selected by installing a jumper in one of the four positions of J9. Corresponding jumpers are then installed in the same row of J10 in positions A and C to route the bus grant daisy-chain lines to the HK68/V20 bus grant circuitry.

The remaining columns of J10 must all have jumpers in the "B" position, which allows the remaining three bus grant daisy-chain signals to bypass the HK68/V20 unmolested.

The four possible configurations are shown below. These are the only valid arrangements of J9 and J10.

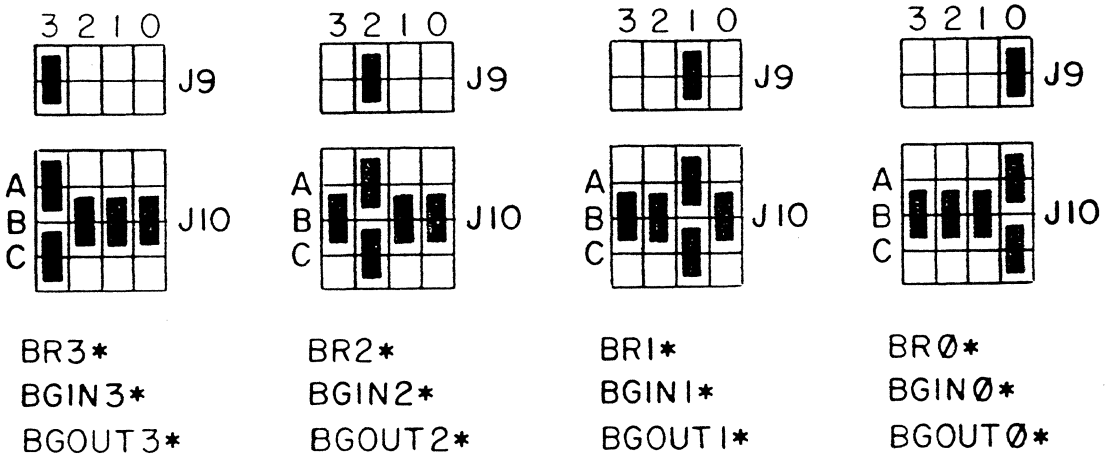


Figure 5. Bus Priority Arbitration Jumpers (HK68/V20 and V2F)

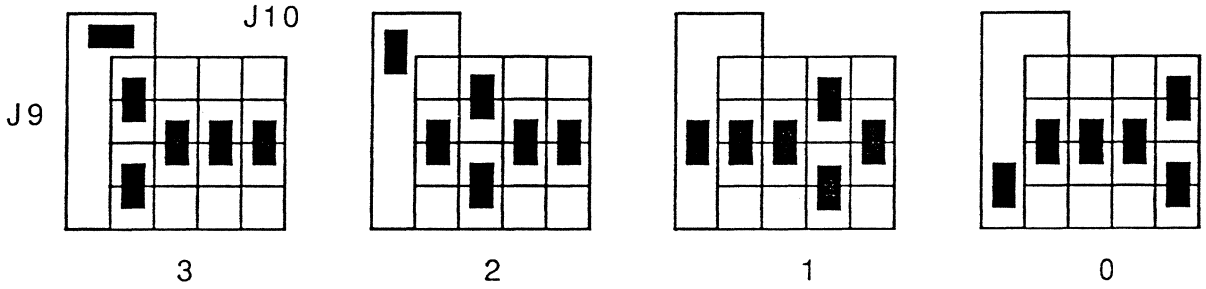


Figure 6. Bus Priority Arbitration Jumpers (HK68/V2FA)

Refer to section 14.3 for help in locating the jumpers.

When the requested operation is completed, the bus will be released according to the state of two control bus control signals, BC1 and BC0. These signals are under software control.

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

Table 21. Bus Control Bits

The bus control bits are set (or reset) by writing to the appropriate bits of the Bus Control Latch, described below. (See section 10.10.)

10.3 Accesses FROM the VMEbus (Slave mode)

The conventional method of board assignment in the VMEbus address space is to utilize a group of DIP switches or jumpers to specify a base address for each board. The Heurikon HK68/V20 uses bus mapping PLEs which monitor the VMEbus for particular combinations of the upper twelve address lines.

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

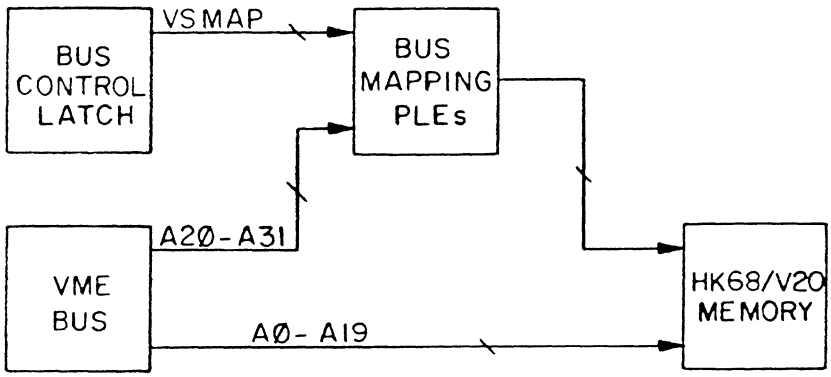


Figure 7. Memory Accesses from the VMEbus

After an access, control of the on-card bus is generally returned to the MPU. However, if the Release With Hold bit is set in the Bus Control Latch (see section 10.10), the bus will not be returned to the MPU for 500 nanoseconds. This mode can be used to speed HK68/V20 slave response time if you expect rapid bus requests, since the time required to arbitrate for the on-card bus will be reduced.

Lengthy memory cycles (hundreds of microseconds), originating from the bus, should be avoided. RAM refreshing is suspended during an access by the bus of on-card RAM. When DTACK* is received from the HK68/V20 board, the master processor must terminate the bus request. If the Watchdog Timer is enabled (via the Bus Control Latch), any access longer than 1.64

milliseconds from the bus will automatically be aborted. The watchdog timer will also terminate a long access to the bus. Status LED L3 will be on when a slave access is in progress.

10.4 Bus Map (Slave mode)

The MPU selects the address space by setting ten Bus Map Control lines, in the Bus Control Latch. The Bus Control Latch is defined in section 10.10.

There are 12 bits that control the map logic. They are arranged in three groups, as defined in the following tables. A valid slave access is recognized only when the condition specified by each group is matched; that is, the conditions are ANDed together. To compute the bus base address, add the two address values from the MP2 and MP3 tables.

VSMAP-			Standard Configuration	
10	9	8	Space	AM5* to AM0*
0	0	0	MP1-0	Off - no access allowed
0	0	1	MP1-1	0x3D
0	1	0	MP1-2	0x39 or 0x3D
0	1	1	MP1-3	0x39, 0x3A, 0x3D or 0x3E
1	0	0	MP1-4	0x0D
1	0	1	MP1-5	0x09 or 0x0D
1	1	0	MP1-6	0x0D or 0x0E
1	1	1	MP1-7	0x09, 0x0A, 0x0D or 0x0E

Table 22. Bus Map - Group MP1 (VSMAP10-8) - Off-card Coming On.

AM5* to AM0*	Address Space
0x09	Extended Non-Privileged Data
0x0A	Extended Non-Privileged Program
0x0D	Extended Supervisory Data
0x0E	Extended Supervisory Program
0x39	Standard Non-Privileged Data
0x3A	Standard Non-Privileged Program
0x3D	Standard Supervisory Data
0x3E	Standard Supervisory Program

Table 23. Address Modifier Codes (AM5*-AM0*)

The base address of the slave window corresponds to the bottom of the on-card RAM window. The VSB memory, if allocated (see the following tables), is contiguous with the end of the on-card RAM area specified in the table. Thus, the combination of spaces MP2-14 and MP3-6 specify a 17 megabyte window starting at bus address E000,0000 (E000,0000 + x000,0000); the first megabyte is on-card RAM and the remaining 16 megabytes are on the VSB. Likewise, MP2-14 and MP3-2 produce a one megabyte region at bus address E020,0000. The configuration may be changed by custom programming the bus mapping PLEs. The smallest block size which may be programmed is one megabyte. Note that although the HK68/V20 can be a master in the short address space, the standard Bus Map PLEs do not allow slave accesses from the short address space.

VSMAP-7	6	5	4	Space	**Standard Configuration** Adrs Lines	VME Base Adrs
0	0	0	0	MP2-0	32	0000,0000
0	0	0	1	MP2-1	32	0400,0000
0	0	1	0	MP2-2	32	0800,0000
0	0	1	1	MP2-3	32	1000,0000
0	1	0	0	MP2-4	32	2000,0000
0	1	0	1	MP2-5	32	4000,0000
0	1	1	0	MP2-6	32	8000,0000
0	1	1	1	MP2-7	32	9000,0000
1	0	0	0	MP2-8	32	A000,0000
1	0	0	1	MP2-9	32	C000,0000
1	0	1	0	MP2-10	32	C400,0000
1	0	1	1	MP2-11	32	C800,0000
1	1	0	0	MP2-12	32	CC00,0000
1	1	0	1	MP2-13	32	D000,0000
1	1	1	0	MP2-14	32	E000,0000
1	1	1	1	MP2-15	24	xx00,0000

Table 24. Bus Map - Group MP2 (VSMAP7-4) - Off-card Coming On.

VSMAP-3	2	1	0	Space	********Standard Configuration******* RAM	VSB	Lines	VME Base Adrs
0	0	0	0	MP3-0	1 Meg	0	32	x000,0000
0	0	0	1	MP3-1	1 Meg	0	32	x010,0000
0	0	1	0	MP3-2	1 Meg	0	32	x020,0000
0	0	1	1	MP3-3	1 Meg	2 Meg	32	x000,0000
0	1	0	0	MP3-4	1 Meg	4 Meg	32	x000,0000
0	1	0	1	MP3-5	1 Meg	8 Meg	32	x000,0000
0	1	1	0	MP3-6	1 Meg	16 Meg	32	x000,0000
0	1	1	1	MP3-7	4 Meg	0	32	x040,0000
1	0	0	0	MP3-8	4 Meg	0	32	x080,0000
1	0	0	1	MP3-9	4 Meg	0	32	x0C0,0000
1	0	1	0	MP3-10	4 Meg	8 Meg	32	x000,0000
1	0	1	1	MP3-11	4 Meg	16 Meg	32	x000,0000
1	1	0	0	MP3-12	16 Meg	0	32	x100,0000
1	1	0	1	MP3-13	1 Meg	0	24	xx00,0000
1	1	1	0	MP3-14	4 Meg	0	24	xx00,0000 *
1	1	1	1	MP3-15	16 Meg	0	24	xx00,0000 *

Table 25. Bus Map - Group MP3 (VSMAP3-0) - PLE # 68/V2-MP3-02

The last two entries in Bus Map Group MP3 (see previous table) reflect modifications made in early 1987. Prior to that time, and when using V20-Bug 1.1, the last two entries were as follows:

VSMAP-					********Standard Configuration*******			
3	2	1	0	Space	RAM	VSB	Lines	VME Base Adrs
1	1	1	0	MP3-14	1 Meg	0	24	xx90,0000
1	1	1	1	MP3-15	1 Meg	0	24	xxE0,0000

Bus Map - Group MP3 (VSMAP3-0) - old PLE # 68/V2-MP3-01

Contact our Customer Service Department if you have questions.

10.5 VME Bus Interrupts

The seven VMEbus interrupts are monitored and controlled by the MFP, MPU and special bus interrupt control logic. By proper programming, certain combinations of the bus interrupt lines may be monitored, and a vectored interrupt to the MPU can be generated when the desired state is realized.

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

10.5.1 Interrupter Module Operation

To generate a VMEbus interrupt, follow these steps:

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

[2] Mask off that level in the MPU (via the Bus Control Latch) or MFP so that the INTERRUPT HANDLER does not respond to the interrupt line you are about to use. If you fail to do this, you could interrupt yourself.

[3] Write an eight or 16-bit value to the appropriate VME Status/ID latch, as described below. This value is usually treated as a simple interrupt vector, but it could represent other information as well. This value is provided to the board that acknowledges the interrupt, which is done by executing an INTERRUPT ACKNOWLEDGE cycle on the VMEbus with your priority level encoded on address lines 1 to 3 (see the Interrupt Handler description, below.)

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

Address	Vector Size	Function (write-only)
00FE,A000	8	Clear Pending VMEbus interrupt from this board.
00FE,A004	8	Interrupt level 1
00FE,A008	8	Interrupt level 2
00FE,A00C	8	Interrupt level 3
00FE,A010	8	Interrupt level 4
00FE,A014	8	Interrupt level 5
00FE,A018	8	Interrupt level 6
00FE,A01C	8	Interrupt level 7
00FE,A024	16	Interrupt level 1
00FE,A028	16	Interrupt level 2
00FE,A02C	16	Interrupt level 3
00FE,A030	16	Interrupt level 4
00FE,A034	16	Interrupt level 5
00FE,A038	16	Interrupt level 6
00FE,A03C	16	Interrupt level 7

Table 26. VMEbus Interrupter Addresses

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

When an interrupt is sent to another HK68/V20 board, a 16-bit vector can be read by the target board using the target's Interrupt Handler logic, described below. However, if the interrupt is directed at the MPU (e.g., IRQ5), only the lower eight bits of the vector will be read during the MPU interrupt acknowledge cycle.

The state of the on-card interrupt logic can be tested by the MFP. The Interrupt Active bit will be true if an interrupt in still pending from this board. Also, the ICU can be programmed to generate a MPU interrupt when the VME Interrupt Active bit goes false, thus allowing the software to queue bus interrupt requests. Note: the interrupt pending bit will not be cleared by accessing address 00FE,A000.

10.5.2 Interrupt Handler Operation

VME Bus Interrupts are received by the MFP (IRQ7, 6, 4 and 2) and the MPU (IRQ6, 5, 3 and 1). Note that IRQ6 is connected to both devices. In order for the MPU to recognize any of the interrupt requests connected to it, the corresponding enable bits must be set in the Bus Control Latch, described in section 10.10. The SYSFAIL* and the ACFAIL* lines are also routed to the MFP to allow generation of an interrupt upon detection of a system failure. Incoming interrupts may be masked off using the mask register of the MFP and the Bus Control Latch. Refer to the MFP technical manual and section 11 for more programming information.

The MFP generates its interrupt to the MPU at Priority level 4, as detailed in section 5.2.

When an interrupt is generated on the VMEbus, the interrupt vector of the interrupting board may be determined by reading from the appropriate address, as shown below. The value returned is that value written by the interrupting board to its VMEbus Status/ID latch.

Priority Level	16-bit Vector Address (read-only)
IRQ2	00FE,E004
IRQ4	00FE,E008
IRQ6	00FE,E00C
IRQ7	00FE,E00E

Table 27. Interrupt Acknowledge Port Summary

Accessing one of the above addresses also sends an interrupt acknowledge signal to the interrupting board. Acknowledging a non-existent interrupt will result in a bus error. The bus interrupt acknowledge signals for levels 1, 3 and 5 are automatically generated when the MPU performs an interrupt acknowledge cycle, so those entries are not included in the above table. Interrupt acknowledge for level 6 should only be manually generated (via address 00FE,E00C) if IRQ6 is handled via the MFP.

Those bus interrupts which are routed to the MFP allow the value returned by the interrupting device during the acknowledge cycle to be treated as a general purpose value rather than as only a vector. Also, the MFP inputs are falling edge sensitive, while the direct interrupts are level sensitive. This variance should be considered in your system design.

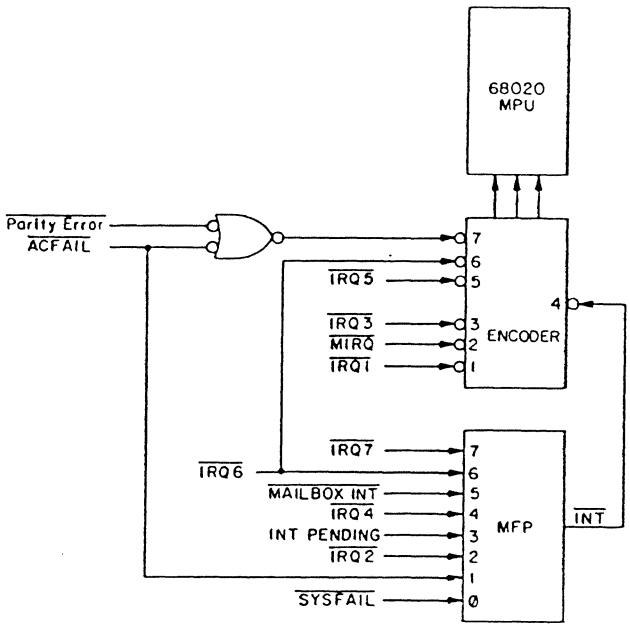


Figure 8. Interrupt Signal Routing

The four IRQx* signals connected to the encoder must be enabled via the Bus Control Latch in order to generate an interrupt. See section 10.10

10.6 SYSFAIL Control

The SYSFAIL line is driven low by the HK68/V20 after power-on. The SYSFAIL line will remain low on the VMEbus until all boards release this line after completing their initialization and self test sequences. The SYSFAIL line also signifies a system failure, and can generate an interrupt via the MFP if it should go on during normal system operation. The current state of this signal may be read via the MFP (see 11.1).

On the HK68/V20, SYSFAIL must be released under software control by accessing the following address (in byte mode):

Address	Function (write-only)
00FE,0000	Release SYSFAIL

10.7 Bus Addressing (Master Mode)

The HK68/V20 supports three address modes, "short", "standard" and "extended". Short addresses use the lower 16 logical address lines to specify the target address. Standard addresses use 24 address lines, and extended addresses use all 32 address lines. The following table details the relationship between the on-card physical address and the corresponding VMEbus and VSB regions.

On-Card addresses	VMEbus Region
00FF,0000 thru 00FF,FFFF	VMEbus Short Address (0000 thru FFFF)
0100,0000 thru 01FF,FFFF	VMEbus Standard (00,0000 thru FF,FFFF)
0300,0000 thru 3FFF,FFFF	VSB Space, or VME Extended Space (0300,0000 thru 3FFF,FFFF)
4000,0000 thru FFFF,FFFF	VMEbus Extended (4000,0000 and up)

Table 28. VMEbus Regions

Extended VME addresses and VSB addresses from 0000,0000 thru 02FF,FFFF are not accessible. The region between 0300,0000 and 3FFF,FFFF is either VSB or VMEbus, depending on whether or not VSB memory exists. That is, if the VSB does not respond to an address in that region, the VMEbus will be used, instead. For this mechanism to work, the VSB ARBE bit in the Bus Control Latch (D23) on the primary master HK68/V20 must be set true, even if the VSB is not used. The bus access hardware operates as follows between 0300,0000 and 3FFF,FFFF:

[1] If this board is the primary VSB master and if the VSB ARBE bit is false, generate a bus error.

[2] Else, arbitrate for the VSB and run a bus cycle.

[3] If a VSB device responds (AC and WAIT), do a VSB cycle.

[4] Else, if no VSB device responds (AC and no WAIT), restart the access on the VMEbus.

[5] Arbitrate for the VMEbus.

[6] If a VMEbus device responds (DTACK), complete the bus cycle.

[7] Else, if no response on the VMEbus (DTACK), generate a bus error. This could be due there being no device at the address or there being no bus arbitration (incorrect setting of J9/J10 or a system controller malfunction).

10.8 Mailbox Interface

Certain on-card functions can be controlled via special addresses in the VMEbus Supervisor Short Address Space, that is, when the address modifier lines (AM5* to AM0*) are 0x2D. The HK68/V20 will respond (as a slave) to a short address which matches the Mailbox select lines, as described below. The mailbox logic must be enabled by setting bit D21 of the Bus Control Latch.

Address	Function (Slave Mode)
Mbase + 0	MFP Input 5 (see section 11.1) (Mailbox Interrupt)
Mbase + 2	HK68/V20 Reset
Mbase + 4	VMEbus Lock On
Mbase + 5	VMEbus Lock Off
Mbase + 6	MPU Halt On
Mbase + 7	MPU Halt Off

Table 29. Mailbox Functions

The "Mbase" value is specified by eight Mailbox Base bits in the Bus Control Latch. Address lines A15, A14, A13, A7, A6, A5, A4 and A3 must match the corresponding Mailbox Base bit. Bus address lines A12, A11, A10, A9 and A8 must be false.

A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
a	a	a	0	0	0	0	0	a	a	a	a	a	x	x	x

Table 30. Mailbox Addressing

The Lock function, when ON, will HALT the MPU after the next access from the bus. The Lock function must be cleared before the MPU will be allowed to resume operation. This feature can be used to reduce arbitration time during a DMA block data transfer from the VMEbus. With the on-card bus locked, slave accesses will be acknowledged in 150 nanoseconds plus two MPU clock cycles.

10.9 VSB Interface

The VSB is a local bus extension designed for high speed access to memory or other facilities without the need to use the VMEbus. The HK68/V20 operates on the VSB in master or secondary modes only; it cannot operate as a slave. It has the required arbitration logic to handle multiple VSB masters. The VSB is a super-set of the VMX32bus; VMX32bus slaves may be used.

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

There is one interrupt line, IRQ, associated with the VSB. This line is connected to MPU interrupt level 2, and generates an autovectored interrupt when recognized by the MPU.

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

	End Board	Other Boards
Install	all (R8-R14)	none
Remove	none	all (R8-R14)

Table 31. VSB Terminations

Summary: Remove resistor packs R8 through R14 on all but one end board.

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

The HK68/V20 supports two VSB release modes. The bus can be released between every access or only if another master requests the bus. Bit D22 in the Bus Control Latch (see below) controls this function as follows:

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

Table 32. VSB Release Modes

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

10.10 Bus Control Latch

This 32-bit latch is used to specify various parameters concerning the operation of the VMEbus and VSB interface. This is a write-only port, and all 32 bit must be written with each access. The default state at power-up is all zeros. The latch is at address 00FE,8000. Attention Hbug users: do not use the 'sl' command; use 'fl'.

Bit	Function	Reference Section
D31	Mailbox Base 15	10.8
D30	Mailbox Base 14	10.8
D29	Mailbox Base 13	10.8
D28	Mailbox Base 7	10.8
D27	Mailbox Base 6	10.8
D26	Mailbox Base 5	10.8
D25	Mailbox Base 4	10.8
D24	Mailbox Base 3	10.8
D23	VSB ARBE	10.7, 10.9
D22	VSB ROR	10.9
D21	Mailbox Enable	10.8
D20	Watchdog Disable	10.3
D19	VMEbus IRQ6 Enable	10.5
D18	VMEbus IRQ5 Enable	10.5
D17	VMEbus IRQ3 Enable	10.5
D16	VMEbus IRQ1 Enable	10.5
D15	BC 1	10.2
D14	BC 0	10.2
D13	User LED (L4), 0 = On	5.3
D12	VME Single Level Arbiter	10.2
D11	Release With Hold	10.3
D10	MP-1 VSMAP 10	10.4
D9	MP-1 VSMAP 9	10.4
D8	MP-1 VSMAP 8	10.4
D7	MP-2 VSMAP 7	10.4
D6	MP-2 VSMAP 6	10.4
D5	MP-2 VSMAP 5	10.4
D4	MP-2 VSMAP 4	10.4
D3	MP-3 VSMAP 3	10.4
D2	MP-3 VSMAP 2	10.4
D1	MP-3 VSMAP 1	10.4
D0	MP-3 VSMAP 0	10.4

Table 33. Bus Control Latch

10.11 Watchdog Timer

The HK68/V20 has a timer which monitors board activity. If the timer is enabled and if the on-card physical address strobe stays on longer than 1.64 milliseconds, the timer will expire. This will cause the current memory cycle to be terminated. The timer is disabled by setting bit D20 in the Bus Control Latch.

See section 8.1 for more details on the watchdog timer.

10.12 Relevant Jumpers - Bus Control

Jumper	Function	Position
J8	Bus SYSRESET* Select	J8-A SYSRESET* is an input from bus J8-B SYSRESET* is an output to bus
J9	VMEbus Request	See section 10.2
J10	VMEbus Grant	See section 10.2
J12	SYSCLK* Enable	(V2F/V2FA only) J12-A Disabled (V2F/V2FA only) J12-B Enabled (output)

Table 34. Bus Control Jumpers

11. MULTI-FUNCTION PERIPHERAL (MFP)

The MFP chip handles certain interrupts, contains four timers and provides the HK68/V20 serial I/O port. This section only provides implementation details which are specific to the HK68/V20. Please refer to the MFP Technical Manual for programming information.

11.1 Interrupt Input Port

The MFP's general purpose I/O port is used to monitor certain VMEbus signals. These state of these signals can be read at any time by the MPU by polling the GPIP register, or the MFP can be programmed to interrupt the MPU if certain lines change state.

Bit	Function	Polarity	Interface
D7	VMEbus Interrupt 7	Negative True	P1-B24
D6	VMEbus Interrupt 6	Negative True	P1-B25
D5	Mailbox Interrupt	Neg True Pulse	(See section 10.8)
D4	VMEbus Interrupt 4	Negative True	P1-B27
D3	VMEbus Interrupt Pending	Positive True	(See section 10.5.1)
D2	VMEbus Interrupt 2	Negative True	P1-B29
D1	VMEbus ACFAIL Interrupt	Negative True	P1-B3
D0	VMEbus SYSFAIL Interrupt	Negative True	P1-C10

Table 35. MFP Interrupt Input Port

The Interrupt Pending bit indicates that this HK68/V20 board has generated a VMEbus interrupt which has not yet been acknowledged.

See section 10.5 for a diagram of the HK68/V20 interrupt logic.

11.2 Counter/Timers

There are four timers in the MFP chip. All timers can be used to generate clock interrupts for the MPU. Timers A and B can be programmed as event counters. Timers C and D may be used as baud rate generators for the serial I/O section. The time base is 2.4576 MHz.

Here are some sample time constant values for MFP timers A and B.

TACR TBCR	Prescale	TADR TBDR	Interrupt Rate
0x05	64	192	200 Hz
0x06	100	128	192 Hz
0x07	200	192	64 Hz

Table 36. Sample MFP Time Constants

11.3 Serial I/O

The serial port is intended to provide a basic communication capability for board and system testing. It is not intended to be a general purpose port. For example, it would be appropriate for an operating system console to be connected to this port, but there are no controls to accommodate a modem.

The serial I/O port may be operated in interrupt or polled mode. In interrupt mode, a separate vector (see section 5.2) can be used receive and transmit functions. To operate in polled mode, the transmitter and receiver state can be tested via separate status registers.

11.3.1 Serial I/F Pinouts

The "RS-232 Function" and "Direction" columns in the following table are with respect to the data terminal device connected to the HK68/V20. The pinout and cable are designed to allow a direct connection to a data terminal device.

P4 Pin	"D" Pin	RS-232 Function	Direction
	1	x	
1	2	Tx Data	(from device)
2	15	Data Clock	(from device)
3	3	Rcv Data	(to device)
	16	x	
	4	x	
	17	x	
4	5	Clear to Send	(to device)
5	18	x	
6	6	Data Set Ready	(to device)
7	19	x	
8	7	Signal Ground	
9	20	x	
10	8	Carrier Detect	(to device)
	9-13	x	
	21-25	x	

Table 37. Serial Port Pinouts (P4)

Note that the ribbon cable is split at the "D" connector to achieve the above pinouts.

Clear to send (CTS), Data Set Ready (DSR) and Data Carrier Detect (DCD) are always held true by the HK68/V20.

11.3.2 Relevant Jumpers (Serial I/O)

The baud clock source may be either on-card or external. MFP timers C and D may be used as on-card sources, or the baud rate clock may be brought in via an RS-232 signal, on P4 pin 2.

Jumper	Function	Position
J11-A,B,C	Data Out Clock	J11-A,B MPF Timer C J11-B,C P4-2
J11-C,D,E	Data In Clock	J12-C,D P4-2 J12-D,E MFP Timer D

Table 38. Relevant Jumpers (Serial I/O)

Due to space limitations, the J11 connections are factory set using wires rather than posts and shunts. On the HK68/V20 and V2F, these jumpers are hiding between crystal Y2 and the DRAMs. On the HK68/V2FA, these jumpers are under U10 (74LS393).

11.3.3 Serial Baud Rates

MFP timers C and D may be used as baud rate generators for the serial port. The table below shows the time constant values to use for particular baud rates. The MFP reference clock is 2,457,600 Hz. The following formula is used to compute the timer data values (for the TCDR and TDDR registers):

$$\text{Time Constant} = \frac{2{,}457{,}600 \;/\; 4}{\text{Baud Rate} * 16 * 2}$$

The extra divide by two is because the timer output toggles when the count hits zero. After setting the baud rate, delay in software long enough for the old count value to expire; the new baud rate does not take effect immediately.

The Timers should be programmed in "Delay Mode, Prescale = 4" (e.g., the TCDCR register should be set to 0x11). The upper bit of the serial port Control Register, UCR, should be set to one to use the divide by 16 clock mode.

Baud Rate	Time Constant
19200	1
9600	2
4800	4
2400	8
1200	16
600	32
300	64
150	128

Table 39. Baud Rate Timer Constants

19,200 baud is the maximum rate available when using the on-card baud rate generator.

11.3.4 Serial I/O Cable

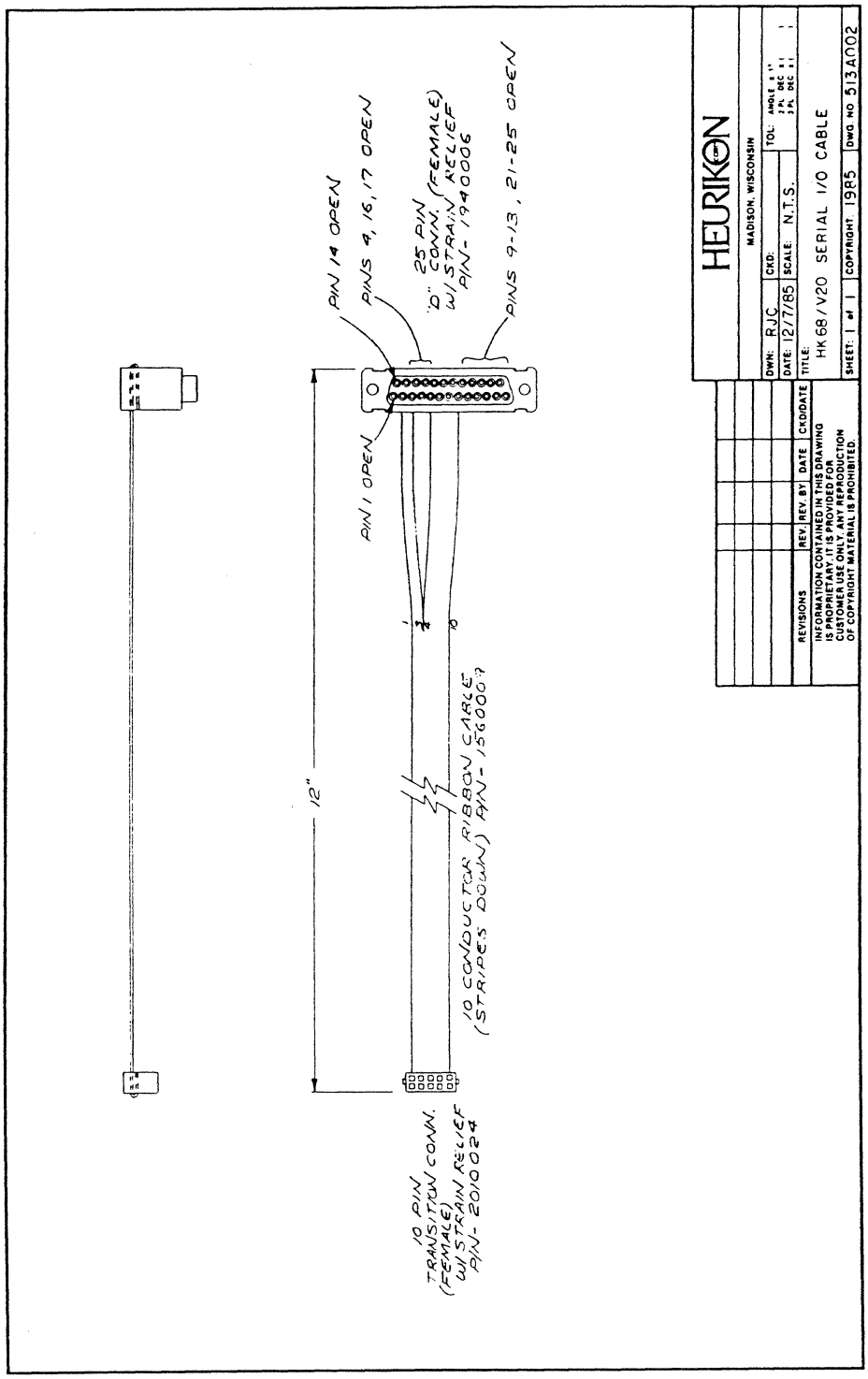


Figure 9. Serial I/O Cable

11.4 Register Summary

The suggested initial values will set the MFP for a 64 Hz interrupt via timer A, mailbox interrupts, and 9600 baud, interrupt driven serial communications.

Address	Name	Suggested Initial Value	Function
00FE,C000	GPIP		Data I/O Port
00FE,C002	AER	0x08	Active Edge Register
00FE,C004	DDR	0x00	Data Direction Register
00FE,C006	IERA	0x34	Interrupt Enable Register A
00FE,C008	IERB	0x80	Interrupt Enable Register B
00FE,C00A	IPRA	0	Interrupt Pending Register A
00FE,C00C	IPRB	0	Interrupt Pending Register B
00FE,C00E	ISRA	0	Interrupt In-Service Register A
00FE,C010	ISRB	0	Interrupt In-Service Register B
00FE,C012	IMRA	0x34	Interrupt Mask Register A
00FE,C014	IMRB	0x80	Interrupt Mask Register B
00FE,C016	VR	0x48	Vector Register
00FE,C018	TACR	0x07	Timer A Control Register
00FE,C01A	TBCR		Timer B Control Register
00FE,C01C	TCDCR	0x11	Timers C and D Control Register
00FE,C01E	TADR	0xC0	Timer A Data Register
00FE,C020	TBDR		Timer B Data Register
00FE,C022	TCDR	0x02	Timer C Data Register
00FE,C024	TDDR	0x02	Timer D Data Register
00FE,C026	SCR		Sync Character Register
00FE,C028	UCR	0x98	USART Control Register
00FE,C02A	RSR	0x01	Receiver Status Register
00FE,C02C	TSR	0x01	Transmitter Status Register
00FE,C02E	UDR		USART Data Register

Table 40. Register Summary (MFP)

All registers are eight bits in length. Interrupt priorities are listed in section 5.2.

11.5 MFP Programming

The following program example shows how to use the MFP timer interrupt and serial ports. The example shows the serial receive and transmit interrupt routines; additional data buffer logic is required.

```
#include "mfp.h"  /* defines mfpdevice structure */
#define MFP       ((struct mfpdevice *)0x00FEC000)

#define VECTbase 0x02000000   /* at base of on-card RAM */
#define VECTOR   0x40    /* MFP interrupt vector number */

mfpinit()
{
    int trans(), rcv(), timer(), mail();
    register int cnt, (**vectptr)();
    static int (*vecttable[])() = {
        0, 0, 0, 0, 0, 0, 0, mail,   /* mailbox intr vector */
        0, 0, trans, 0, rcv, /* transmit and receiv vectors */
        timer, 0, 0          /* timer interrupt vector */
    };
    static struct mfpentry {
        unsigned char *reg;
        unsigned char val;
    } mfptable[] = {
        &MFP->tacr,  0x07,  /* timer A, delay mode, 200 prescale */
        &MFP->tadr,  192,   /* timer A, count value */
        &MFP->vr, VECTOR | 0x08, /* vector, in-service reg enable */
        &MFP->iera,  0x34,  /* enable timer A, Rcv, Tx interrupts */
        &MFP->ierb,  0x80,  /* enable mailbox interrupt */
        &MFP->ipra,  0x00,  /* cleat interrupt pending regs */
        &MFP->iprb,  0x00,
        &MFP->isra,  0x00,  /* clear interrupt in service regs */
        &MFP->isrb,  0x00,
        &MFP->imra,  0x34,  /* interrupt mask reg */
        &MFP->tcdcr, 0x11,  /* timer C & D, delay mode, 4 prescale */
        &MFP->tcdr,  0x02,  /* timer C, count value (for 9600 baud) */
        &MFP->tddr,  0x02,  /* timer D, count value (for 9600 baud) */
        &MFP->ucr,   0x98,  /* 16 prescale, 8 bits, no parity, 2 stop */
        &MFP->rsr,   0x01,
        &MFP->tsr,   0x01,  /* enable Tx and Rcv */
    };

    vectptr = (int (**)())( VECTbase + (VECTOR * 4) );
    for (cnt = 0; cnt < sizeof(vecttable)/sizeof(int(**)()); cnt++)
        *vectptr++ = vecttable[cnt];
    /* be sure 68020 vector base register is set to VECTbase */

    for (cnt = 0; cnt < sizeof(mfptable)/sizeof(struct mfpentry); cnt++)
        *mfptable[cnt].reg = mfptable[cnt].val;

    for (cnt = 0; cnt < 10000 ; cnt++);   /* baud rate settle time */
}
/* don't forget to enable 68020 interrupts via "orw #0x0700,sr" */
```

Table 41. MFP Programming Example (C Portion Part 1)

```
long ticks = 0;
tintr()    /* timer A interrupt, 64 Hz, from _timer */
{   ticks++;
    MFP->isra = 0xDF;
}

txintr()    /* transmit interrupt service routine, from _trans */
{   MFP->udr = nextchar;  /* send character, skip this if done */
    MFP->isra = 0xFB;
}

txstart()                    /* start the transmitter */
{   MFP->udr = firstchar;    /* get a character and send */
}

rcvintr()    /* receiver interrupt service routine, from _rcv */
{   if ( RCVxBUFFERxFULL )
        MFP->imra &= 0xEF;          /* stop the receiver */
    (unsigned char)rcvchar = MFP->udr;  /* get rcv char */
    MFP->isra = 0xEF;
}

rcvstart()    /* start the rcv interrupts */
{   MFP->imra |= 0x10;
}

mailint()    /* mailbox interrupt service routine, from _mail */
{   MFP->isrb = 0x7F;
}
```

Table 42. MFP Programming Example (C Portion Part 2)

```
        .text
        .globl  _trans, _rcv, _timer, _mail
_trans: moveml #0xFFFF,sp@-   | save all regs
        jsr     _txintr       | go to C code
        bra     restr
_rcv:   moveml #0xFFFF,sp@-   | save all regs
        jsr     _rcvintr      | go to C code
        bra     restr
_mail:  moveml #0xFFFF,sp@-   | save all regs
        jsr     _mailint      | go to C code
        bra     restr
_timer: moveml  #0xFFFF,sp@-  | save all regs
        jsr     _tintr        | go to C code
restr:  moveml  sp@+,#0xFFFF  | restore regs
        rte
```

Table 43. MFP Programming Example (Assembly Portion)

12. REAL-TIME CLOCK (RTC) - Optional Feature

As an option, one pROM can be fitted with a special socket which has a built-in CMOS watch circuit and a lithium battery (Dallas Semiconductor, part number DS1216E).

On the V20, the module socket is installed in the first HK68/V20 pROM position (U23). On the V2F board, due to mechanical considerations, U24 should be used; In that case, the value for WATCHBASE will be 0x010000. Use U15 on the V2FA. It can be plugged into the existing socket (in which case the board profile is wider) or it may replace the standard pROM socket (except on the HK68/V2FA). The following table lists resulting board thickness values, depending on the installation method. The values include a standard pROM thickness.

Configuration	Component Height Above Board	Minimum Board Spacing
RTC module plugged into existing pROM socket:	.75 in.	.85 in. (2 slots)
RTC module soldered into HK68/V20 board:	.55 in.	.65 in. (1 slot)

Table 44. RTC module, physical effects

Only one card slot is required if the board is in the end slot. The RTC logic does not generate interrupts; a MFP timer channel is still used for that purpose. The RTC contents, however, may be used to check for long-term drift of the HK68/V20 system clock, and as an absolute time and date reference after a power failure. Leap year accounting is included. Heurikon can provide complete operating system software support for the RTC module.

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

The clock contents are set or read using a special sequence of ROM read commands, as detailed in the program example, below. The RTC module "monitors" ROM accesses and, if a certain sequence of 64 ROM addresses occur, takes temporary control of the ROM space, allowing data to be read from or written to the module. Writing is done by twiddling an address line, which the module uses as a data input bit. There are never any MPU write cycles directed to the pROM space.

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

```
#define WATCHBASE (unsigned char *)0x000000 /* first socket */
#define WR0_WATCH (unsigned char *)(WATCHBASE+2) /* write 0 */
#define WR1_WATCH (unsigned char *)(WATCHBASE+3) /* write 1 */
#define RD_WATCH  (unsigned char *)(WATCHBASE+4) /* read */
struct rtc_data {          /* D7 D6 D5 D4   D3 D2 D1 D0   range */
    unsigned char dotsec;/* --0.1 sec-- : --0.01 sec- ; 00-99 */
    unsigned char sec;    /* --10 sec--- : --seconds-- ; 00-59 */
    unsigned char min;    /* --10 min--- : --minutes-- ; 00-59 */
    unsigned char hour;   /*  A  0  B Hr : --hours---- ; 00-23 */
    unsigned char day;    /*  0  0  0  1 : --day------ ; 01-07 */
    unsigned char date;   /* --10 date-- : --date----- ; 01-31 */
    unsigned char month; /* --10 month- : --month---- ; 01-12 */
    unsigned char year;   /* --10 year-- : --year----- ; 00-99 */
}; /* "A" =  "0" for 00-23 hour mode, "1" for 01-12 hour mode */
   /* "B" =  MSB of the 10 hours value (if 00-23 hour mode) else
         =  "0" for PM or "1" for AM  (if 01-12 hour mode) */

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

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

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

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

Figure 10. Real-Time Clock, Example Software

13. VMEBUS INTERFACE

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

13.1 P1 (VMEbus) Pin Assignments

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

Table 45. P1 (VMEbus) Connector Pinout

Not all of the signals are used on the HK68/V20. See section 10 for details and signal descriptions.

13.2 P2 (VSB) Pin Assignments

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

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

Table 46. P2 (VMEbus, VSB) Connector Pinout

See section 10 for details and signal descriptions.

13.3 Power Requirements

Voltage	Current	Usage
+5	8.0A, max	All logic
+12	0.5A, max	RS-232 I/F
-12	0.5A, max	RS-232 I/F

Table 47. Power Requirements

The "+5" and "Gnd" pins on P2 must be connected to assure proper operation, even if the VSB interface is not used.

13.4 Environmental

Operating temperature: 0 to +55 degrees Centigrade, ambient, at board. Humidity: 0% to 85%. Storage temperature: -40 to +70 degrees C. NOTICE: Power dissipation is about 40 watts. Fan cooling is required if the HK68/V20 board is placed in an enclosure or card rack. Fan cooling is also recommended when using an extender board for more than a few minutes.

13.5 Mechanical Specifications

		Height (above board)	
Width	Depth	under 4 Meg Ram	4 MBytes Ram
9.187 in.	6.299 in.	0.6 in.	0.8 in.
233.35 mm	160 mm	15.25 mm	20.35 mm

Table 48. Mechanical Specifications (HK68/V20 and HK68/V2F)

Width	Depth	Height (above board)
9.187 in.	6.299 in.	0.4 in.
233.35 mm	160 mm	10.17 mm

Table 49. Mechanical Specifications (HK68/V2FA)

Standard board spacing is 0.8 inches. The HK68/V20, V2F and V2FA are 10 layer boards.

14. SUMMARY INFORMATION

14.1 Software Initialization Summary

This section outlines the steps for initializing the facilities on the HK68/V20 board. Certain steps must be performed in sequence, while others may be rearranged or omitted entirely, depending on your application.

[1] The MPU automatically fetches the reset vector following a system reset and loads the supervisor stack pointer and program counter. The reset vector is in the first 8 bytes of ROM.

[2] Recall the NV-RAM contents. (Reference: section 9.7)

[3] Determine RAM configuration. (Reference: section 9.3)

[4] Clear the Bus Control Latch. (Reference: section 10.10)

[5] Clear on-card RAM to prevent parity errors due to uninitialized memory reads. (Reference: section 8.1)

[6] Load the 68020 Vector Base Register with the location of your exception vector table (usually 0).

[7] Initialize the exception vector table in RAM (at the selected base address.) This step links the various exception and interrupt sources with the appropriate service routines. (Reference: section 5.2)

[8] Initialize the MMU. (Reference: section 7)

[9] Initialize the MFP devices. (Reference: section 11)

[10] Release the VMEbus SYSFAIL line. (Reference: section 10.6)

[11] Initialize off-card memory and I/O devices, as necessary.

[12] Set the Bus Control Latch bits as desired. (Reference: section 10.10)

[13] Enable system interrupts, as desired. (Reference: section 5.1)

14.2 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 9.5 for a pictorial description of the system memory map. All ports are on the physical address bus.

Hex Address	Type	Device	Reference Section
4xxx,xxxx	R/W	VMEbus (Extended Adrs mode)	10.7
03xx,xxxx	R/W	VSB or VMEbus	10.7, 10.9
02xx,xxxx	R/W	HK68/V20 on-card RAM	9.2
01xx,xxxx	R/W	VMEbus (Standard Space)	10.7
00FF,xxxx	R/W	VMEbus (Short Adrs mode)	10.7
00FE,E00x	R	VMEbus Interrupt Ack	10.5
00FE,A0xx	W	VMEbus Status/ID Latch	10.5
00FE,C0xx	R/W	MFP (Ints, Serial, Timers)	11.4
00FE,8000	W	VMEbus Control Latch	10.10
00FE,600x	R/W	MMB Registers (V20 only)	7.4
00FE,0000	R	Release SYSFAIL	10.6
0005,0000	R	NV-RAM Recall	9.7
0004,0000	R	NV-RAM Store (tas)	9.7
0003,00xx	R/W	NV-RAM Access Space	9.7
0002,xxxx	W	EEpROM 1	9.1
0001,xxxx	R	ROM 1 (not on V2FA)	9.1
0000,xxxx	R	ROM 0	9.1

Table 50. Address Summary

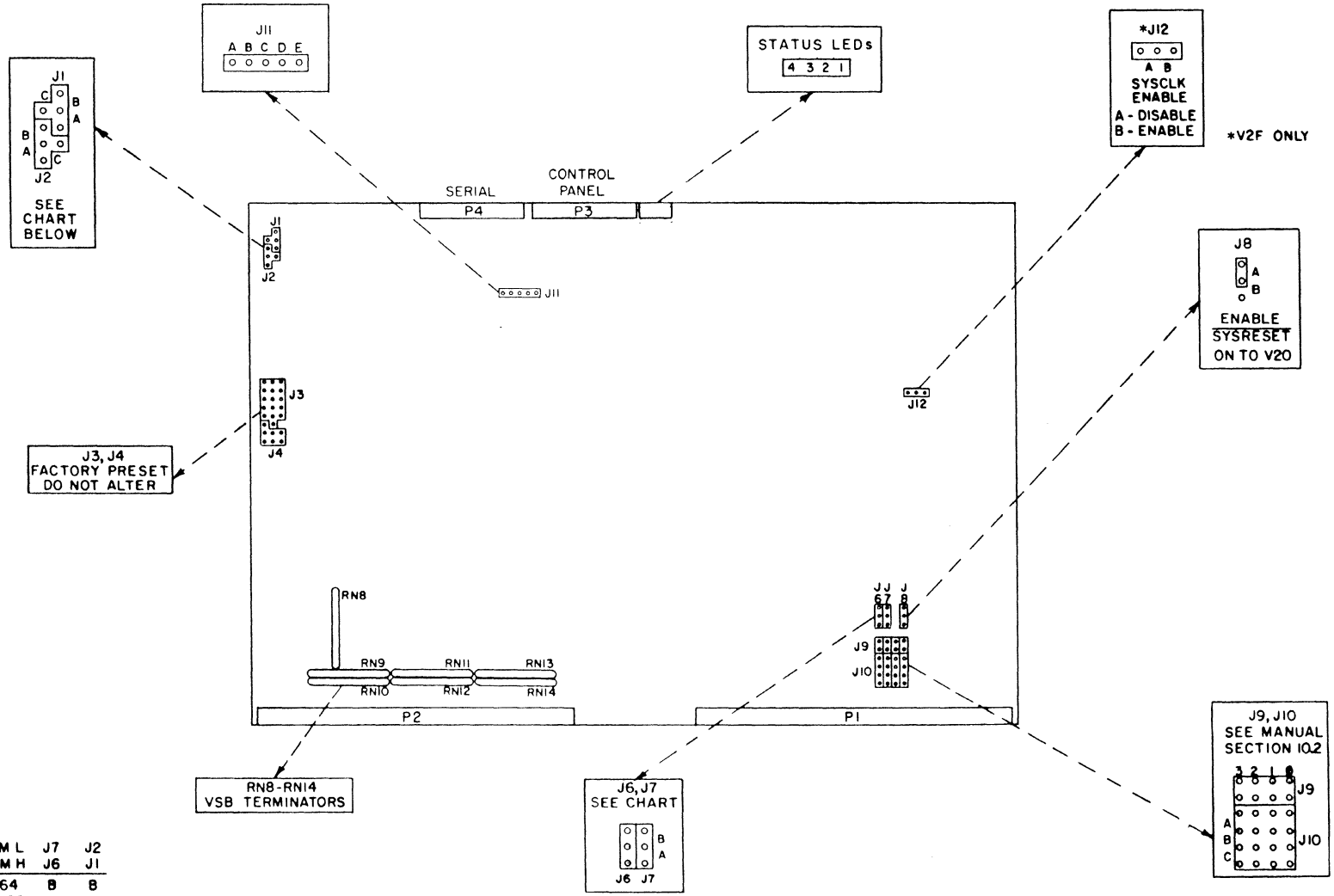
14.3 Hardware Configuration Jumpers

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

Jumper	Function	Reference Section	Standard Configuration
J1	ROM type	9.1	J1-B (2764)
J2	ROM type	9.1	J2-B (2764)
J6	ROM type	9.1	J6-B (2764)
J7	ROM type	9.1	J7-B (2764)
J3	Memory Timing	9.6	(Factory Set)
J4	Memory Timing	9.6	(Factory Set)
J5	(not used)		
J8	SYSRESET Control	10.11	J8-A (input)
J9	VMEbus Request/Grant	10.2	J9-3 (BR3)
J10	VMEbus Request/Grant	10.2	J10-3A,3C and J10-0B,1B,2B (BR3)
J11	Tx Serial Clock	11.3	J11-A,B (Timer C)
	Rcv Serial Clock	11.3	J11-D,E (Timer D)
J12	SYSCLK* Drive (not on V20)	10.2	J12-A (disabled)

Table 51. Jumper Summary

The HK68/V20 has been designed to have a minimum number of configuration jumpers. As many options as possible are under software control. For example, the Bus Control Latch, described in section 10.10, allows software control of bus mapping and mailbox logic. The NV-RAM can be used to store board and system configuration information.



ROM L	J7	J2
ROM H	J6	J1
2764	B	B
27128	B	B
27256	B	A
27512	A	A
27513	A	C
27916	A	C
2864	OPEN	C

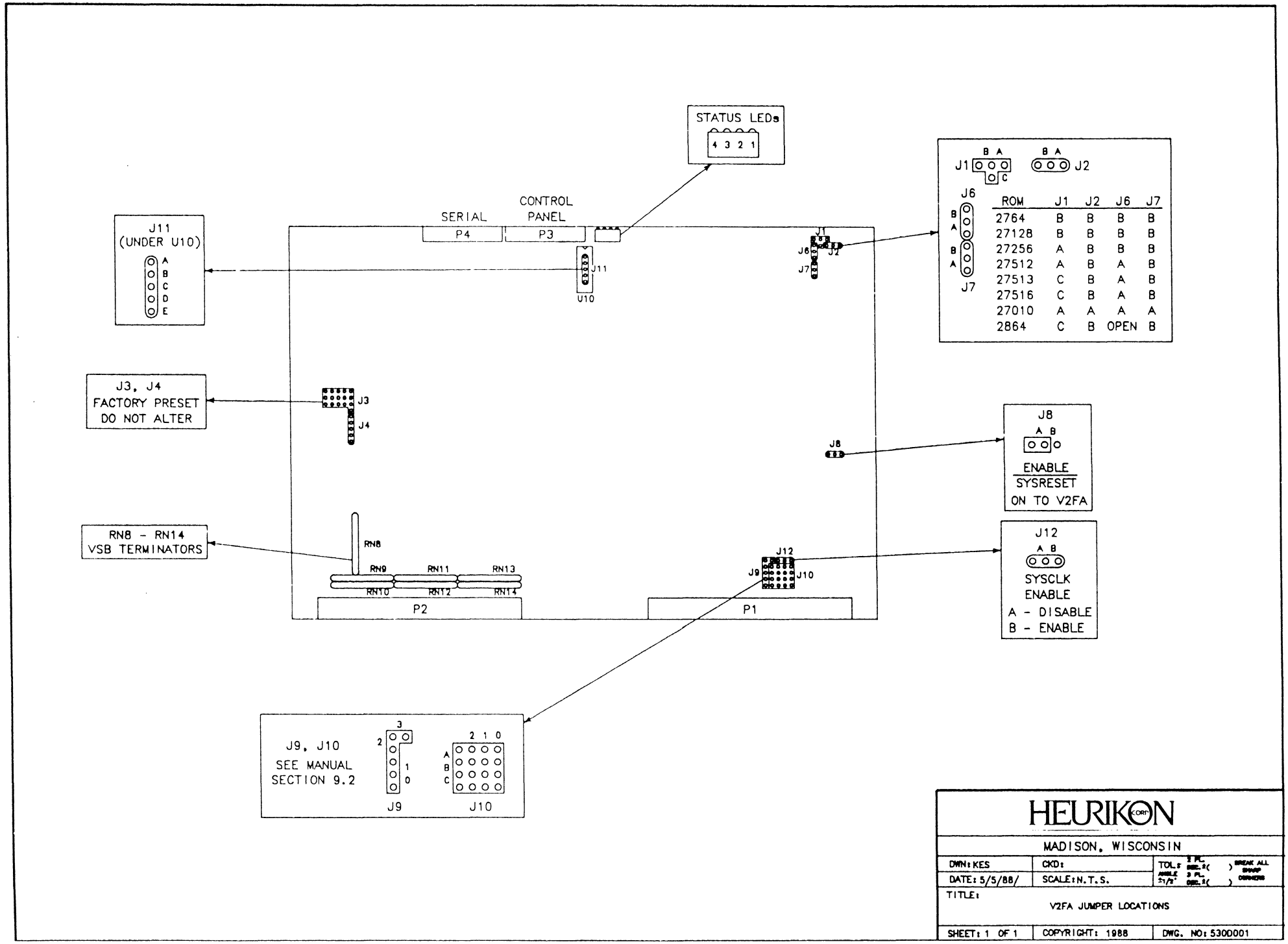
REVISIONS	REV.	REV. BY	DATE	CKD/DATE
ADDED J12	D	KES	2/87	
ADDED V33 TERMINATORS	C	KES	8/86	
ADD J11	B	KES	3/86	
ADD STATUS LED's	A	RJC	3/86	

HEURIKON CORP
MADISON, WISCONSIN

DWN: RJC	CKD:	TOL: 2 PL. DEC. ± ( ) ANGLE ± 1/2° 3 PL. DEC. ± ( )	BREAK ALL SHARP CORNERS
DATE: 1/14/86	SCALE: N.T.S.		

TITLE: HK68/V20 JUMPER LOCATIONS

SHEET: 1 of 1	COPYRIGHT: 1986	DWG. NO: 513D004-D



HEURIKON

MADISON, WISCONSIN

DWN: KES	CKD:	TOL: 3 PL. ±.4( ) BREAK ALL SHARP CORNERS; ANGLE ±1/2°; 3 PL. ±.4( )
DATE: 5/5/88/	SCALE: N.T.S.	

TITLE:

V2FA JUMPER LOCATIONS

SHEET: 1 OF 1	COPYRIGHT: 1988	DWG. NO: 5300001

15. READER COMMENT FORM

We would appreciate any comments you have concerning this guide. Please let us know if you have found any errors or feel that certain sections should be expanded. Thank you.

Name:_____ Title:_____

Company:_____ Date:_____

Address:_____

City:_____ State:_____ ZIP:_____

Telephone: ( )-_____-_____

Section	Comments

Mail to: Heurikon Corporation

HK68/V20 Rev J

16. APPENDICES

16.1 Additional Technical Literature

Additional information is available on the HK68/V20 peripheral chips, either from Heurikon sales or directly from the chip manufacturers.

Device	Number	Document	Reference Section
MPU	68020	Motorola 68020 Spec	5
FPP	68881	Motorola MC68881 Spec	6
PMMU	68851	Motorola MC68851 Spec	7
MFP	68901	Motorola Multi-Function Peripheral	11
RTC	DS1216E	Dallas Semiconductor Clock Module	12

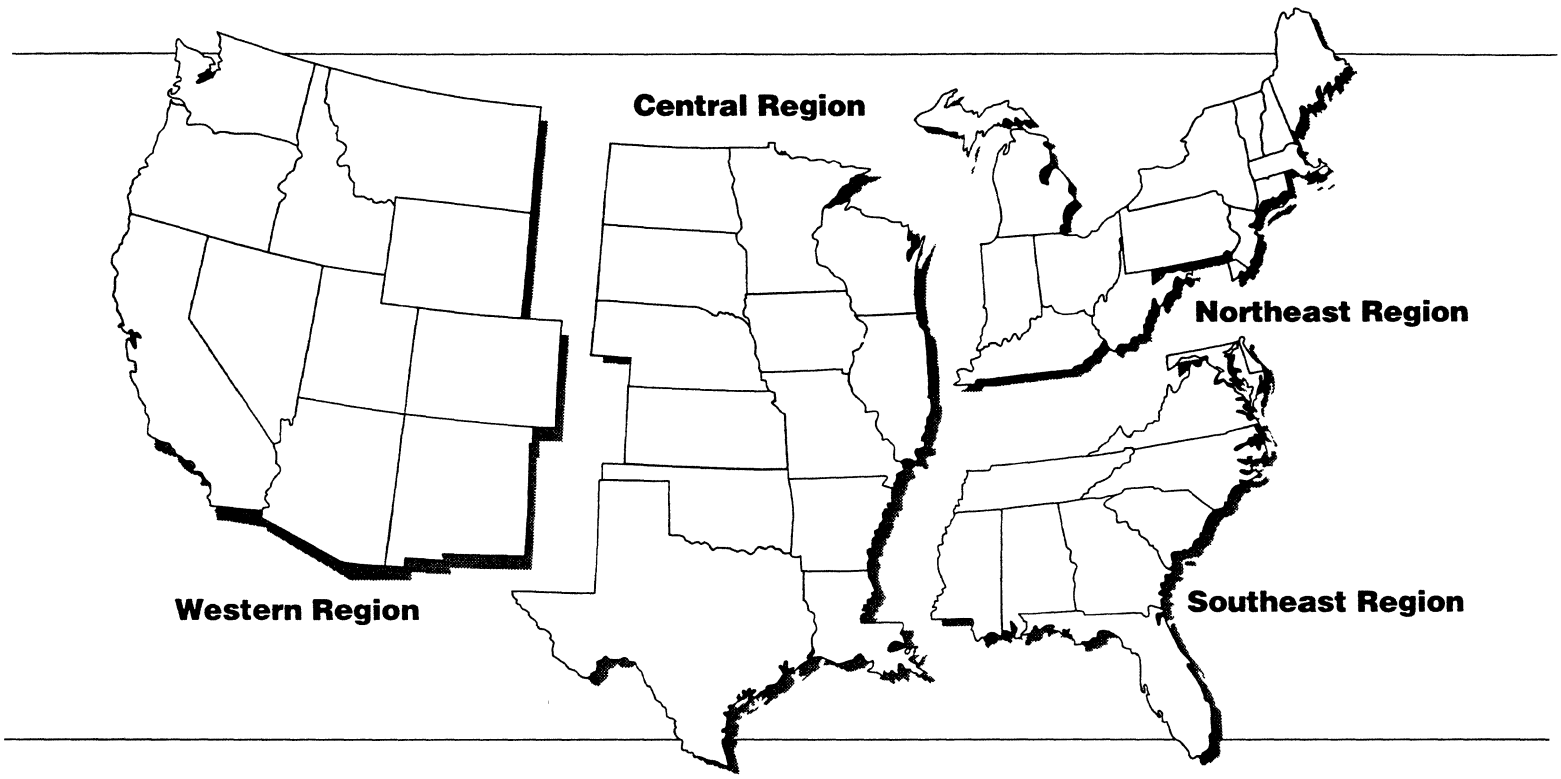
Table 52. Additional Technical Literature

16.2 HK68 VME Family - Feature Summary

Feature	HK68 V10/VF	HK68 VE	HK68 V20/V2F/V2FA
System Bus	VME	VME	VME
Expansion	-	-	VSB
MPU	68010	68000	68020
Speed, MHz	10/12.5	12.5	12.5
MMU/PMMU	68451/no/no	-	68851/no
FPP	68881	(SBX)	68881
DMAC	68450/440	68440	-
ROM	128-KB	256-KB	128-KB
skts	2 Skts	4 Skts	2/2/1 Skts
width	16 bits	16 bits	8 bits
EEpROM	-	yes	yes
On-card RAM	1 Meg	4 Meg	4 Meg
Parity	yes	yes	yes
NV-RAM	-	-	1-Kbit
SCSI	5380/opt	(SBX)	-
Control	CIO	CIO	MFP
Serial ports	2 (SCC)	2 (SCC)	1 (MFP)
Streamer I/O	(SCSI)	-	-
Parallel I/O	-	-	-
Mailbox	std	std	std
SBX	-	1	-
TOD Clock	option	option	option
User LEDs	4	4	1
Status LEDs	5 (ext)	5 (ext)	7 (4ext)
User Jumpers	8	8	-

Table 53. HK68 VME Family Feature Summary

Sales and Customer Service Offices



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

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

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

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

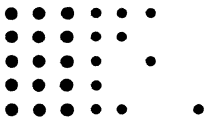
Heurikon Central Regional Office
13100 West 95th Street, Level 4D
Lenexa, KS 66215
Phone: 913-599-1860
Fax: 913-599-1918

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

HEURIKON CORP.

OPEN SYSTEMS :: OPEN TOOLS

Regional Sales Representatives



• NORTHEAST REGION

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

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

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

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

• CENTRAL REGION

AR, LA, OK and TX

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

MN, ND, SD and Northwest WI

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

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

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

• SOUTHEAST REGION

MD, VA and Washington, DC

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

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

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

• WESTERN REGION

AZ, CO, NV, NM and UT

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

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

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

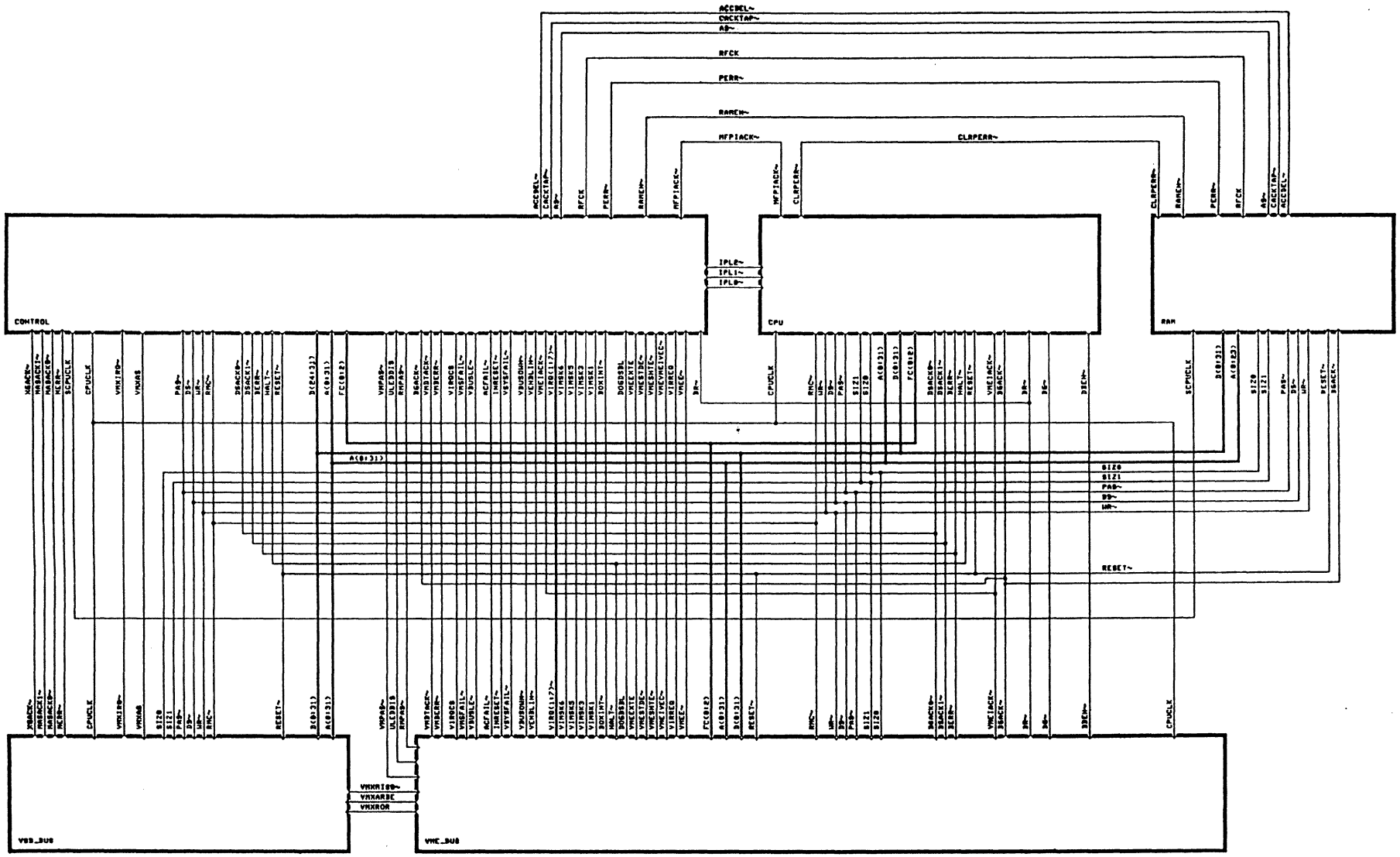
CA

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

HEURIKON CORP.

OPEN SYSTEMS :: OPEN TOOLS

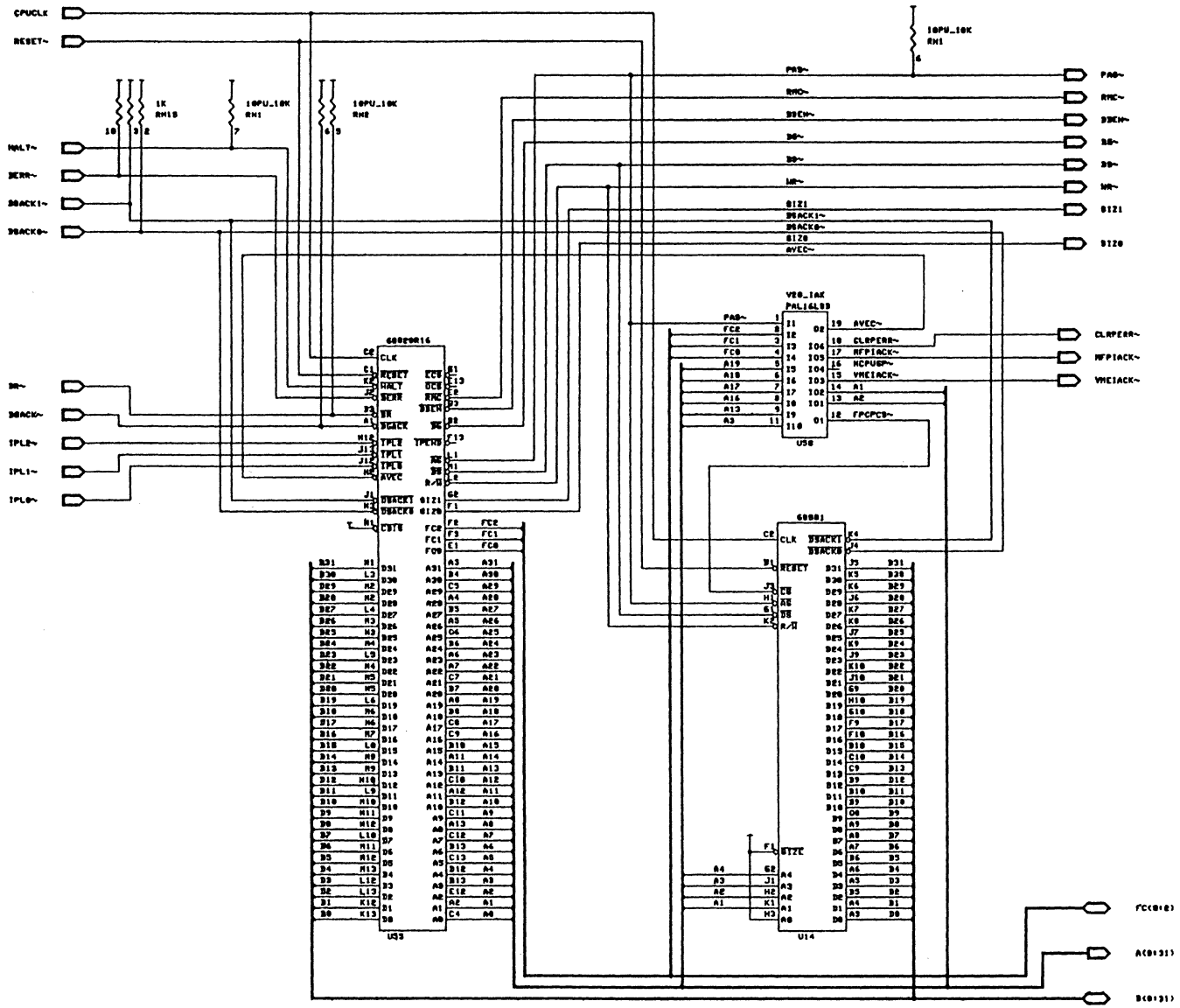
revised 7.91



HEURIKON

MADISON, WISCONSIN

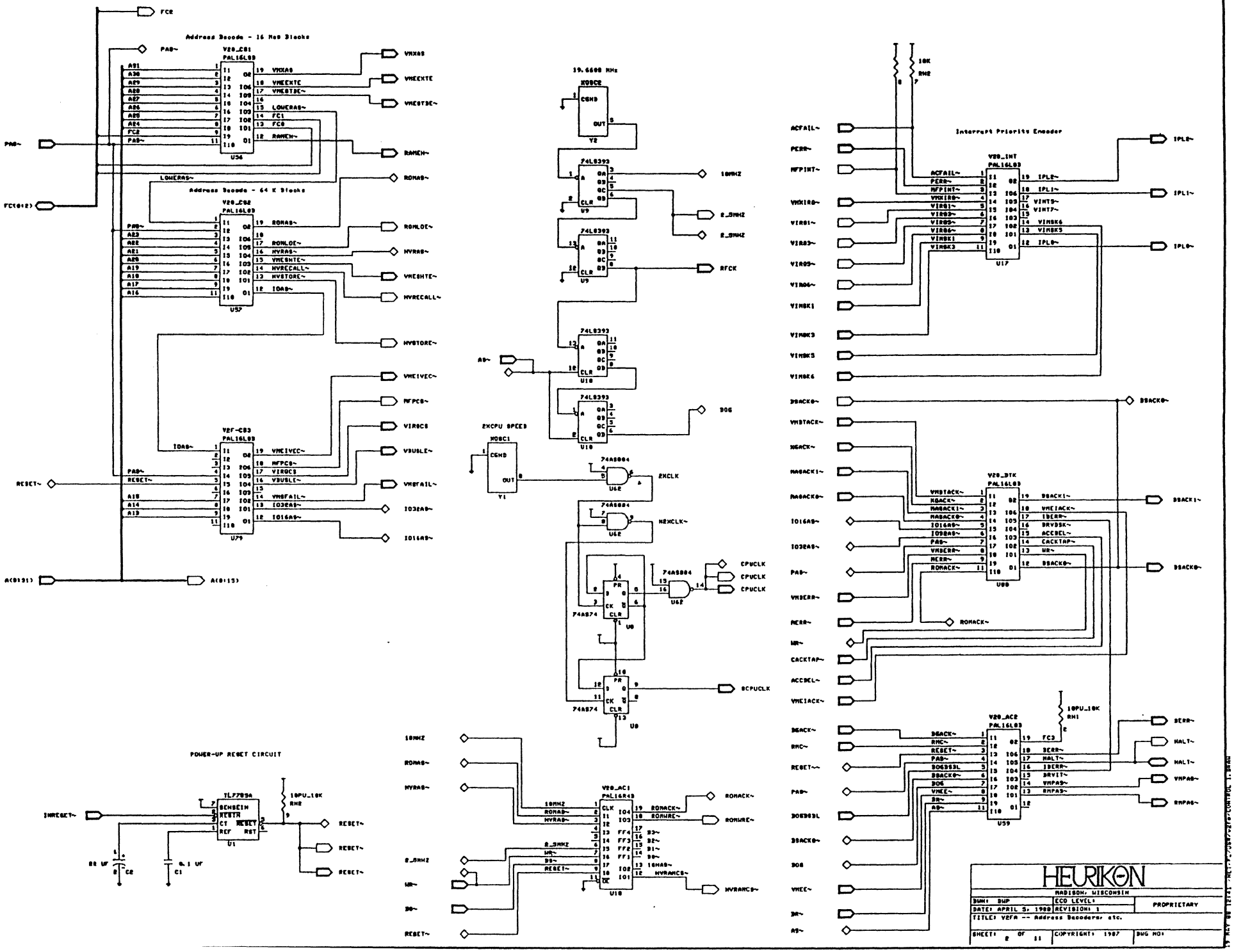
DWN: BWP	ECO LEVEL:	PROPRIETARY
DATE: APRIL 5, 1988	REVISION: 1	
TITLE: VBFA -- Block Diagram		
SHEET: 8 OF 11	COPYRIGHT: 1987	DWG NO:

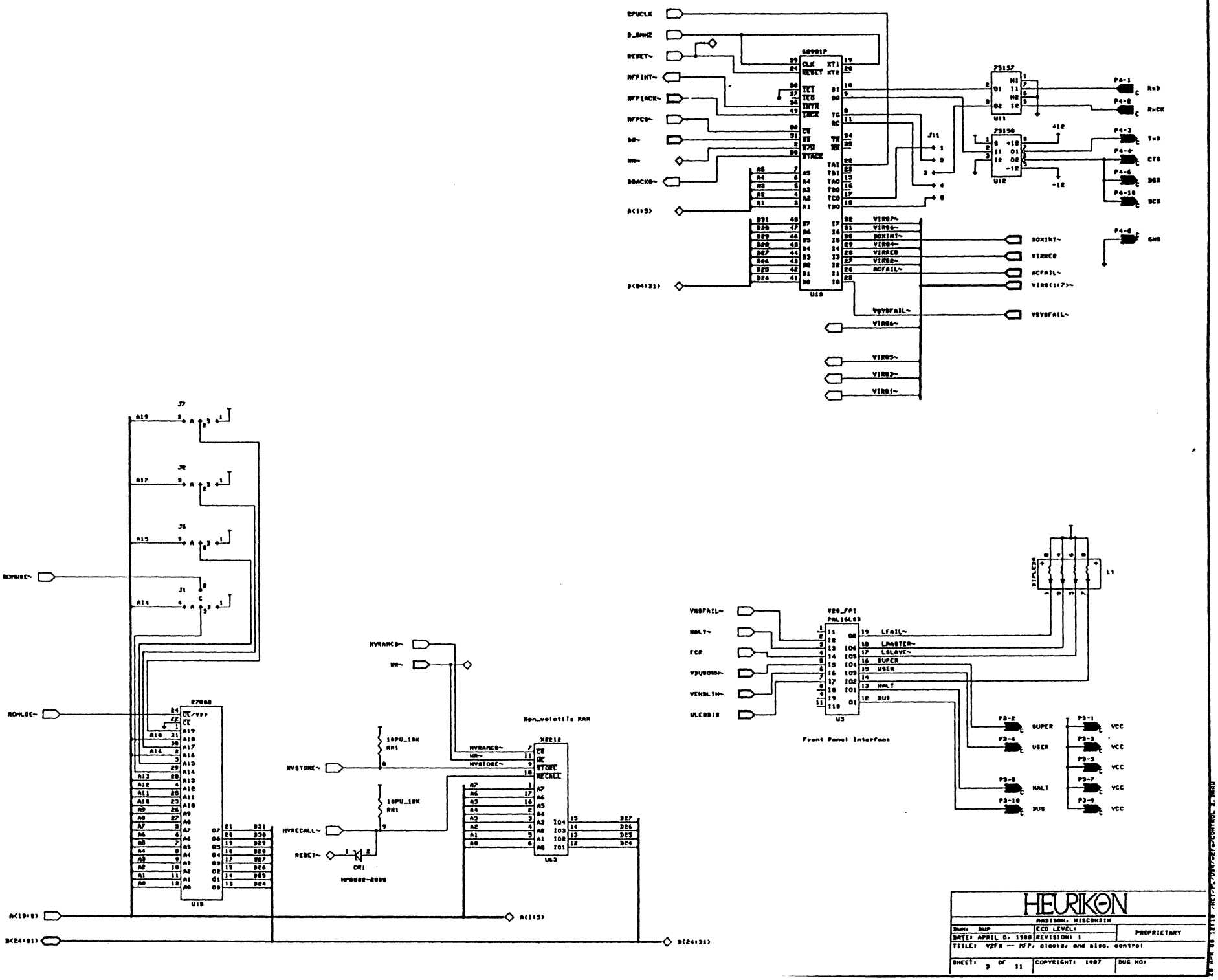


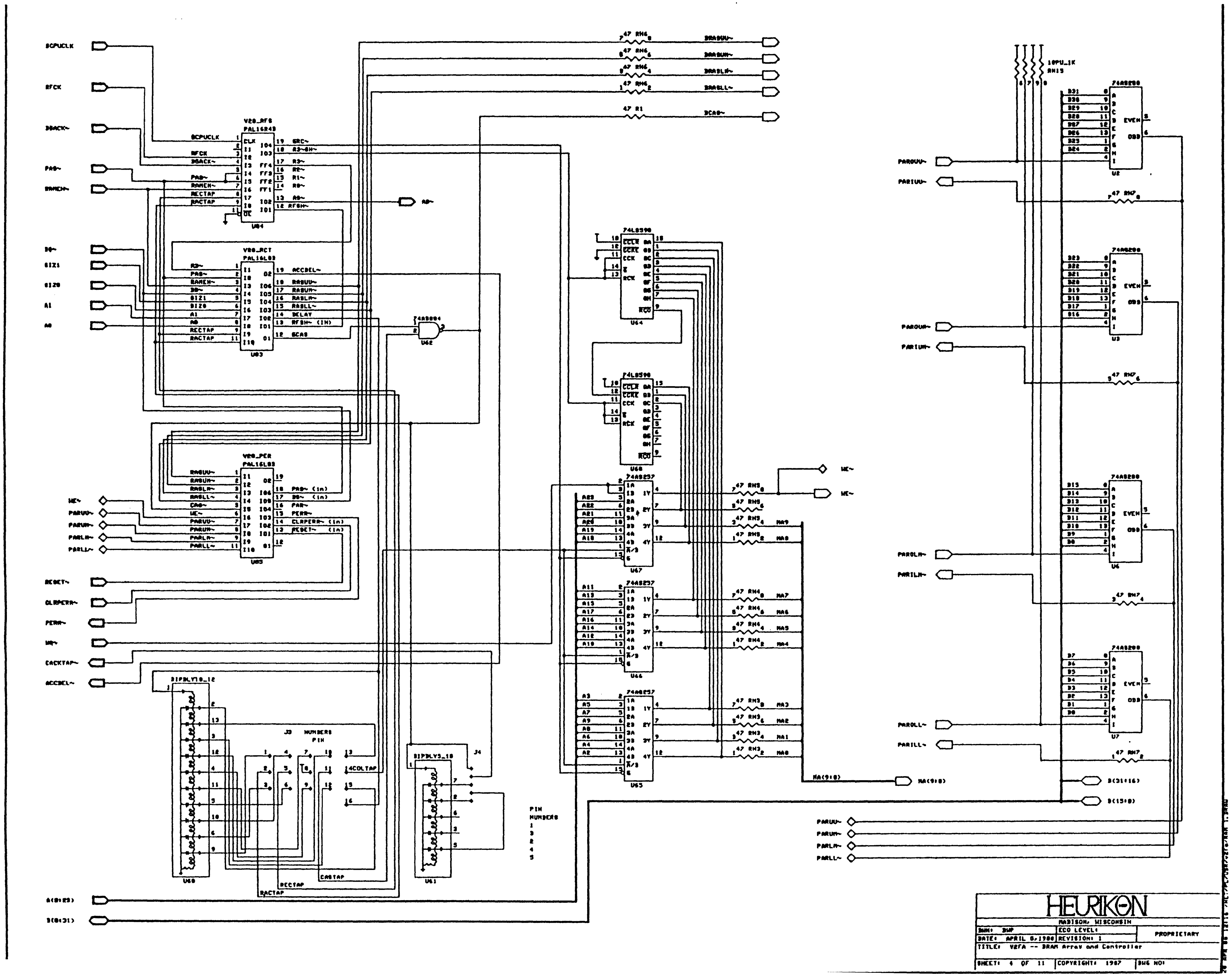
HEURIKON

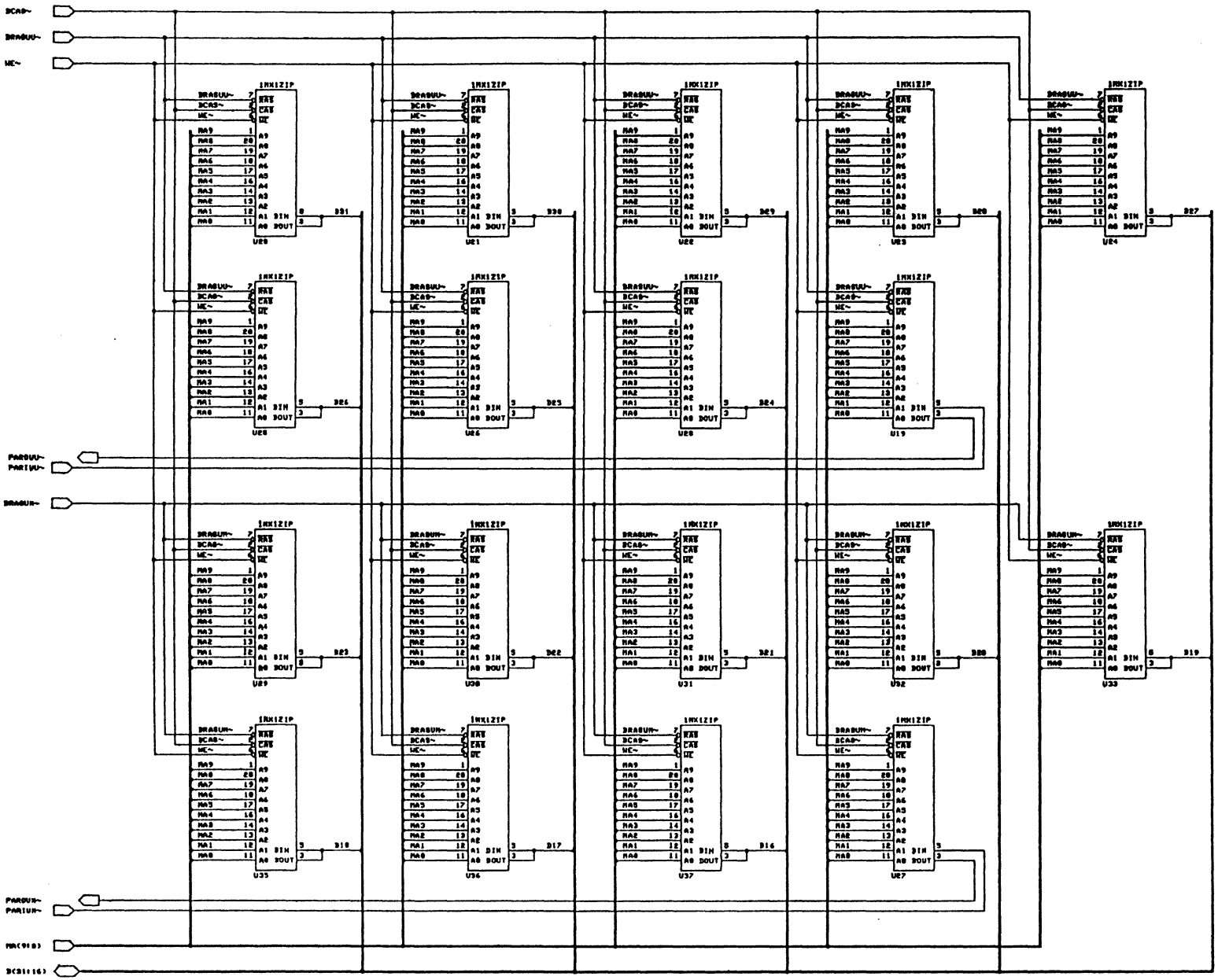
MADISON, WISCONSIN

DWN: DWP	ECO LEVEL:	PROPRIETARY
DATE: Oct. 9,1985	REVISION: 1	
TITLE: VEGA -- MPU AND FPCP		
SHEET: 1 OF 11	COPYRIGHT: 1987	DWG NO:

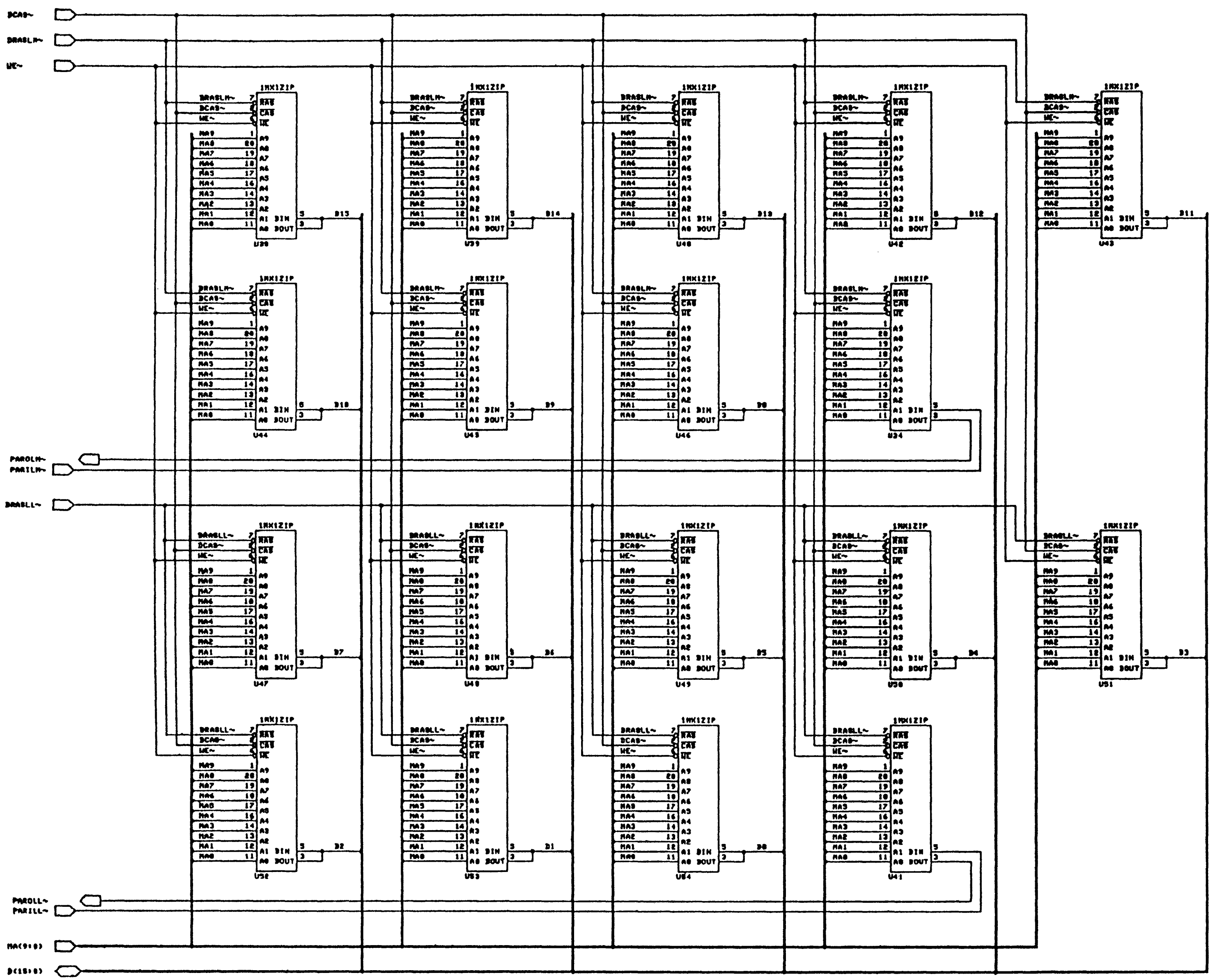

Address Decode - 16 Meg Blocks
Address Decode - 64 K Blocks
19.6608 MHz
Interrupt Priority Encoder
POWER-UP RESET CIRCUIT
HEURIKON
MADISON, WISCONSIN
PROPRIETARY
DWN: DWP
ECO LEVEL:
DATE: APRIL 5, 1988
REVISION: 1
TITLE: VEFA -- Address Decoders etc.
SHEET: 2 OF 11
COPYRIGHT: 1987
DWG NO:


Non_volatile RAM
Front Panel Interface
HEURIKON
MADISON, WISCONSIN
PROPRIETARY
TITLE: VZFA — MFP, clocks, and misc. control
COPYRIGHT: 1987


SCPUCLK
RFCK
SBACK~
PAS~
RAMEN~
DS~
SIZ1
SIZ0
A1
A0
V20_RFS
PAL16R4B
U04
V20_RCT
PAL16L8B
U03
V20_PER
PAL16L8B
U05
74AS004
U62
74LS590
U64
U68
74AS257
U67
U66
U63
74AS280
U2
U3
U6
U7
10PU_1K
RN15
RESET~
CLRPERR~
PERR~
WR~
CACKTAP~
ACCDEL~
DIPDLY10_12
U60
DIPDLY5_10
U61
J3
J4
PIN NUMBERS
RECTAP
RACTAP
CASTAP
WE~
MA(9:0)
PARUU~
PARUM~
PARLM~
PARLL~
D(31:16)
D(15:0)
A(0:23)
D(0:31)
HEURIKON
MADISON, WISCONSIN
DWN: DWP
DATE: APRIL 6, 1988
ECO LEVEL:
REVISION: 1
PROPRIETARY
TITLE: V2FA -- DRAM Array and Controller
SHEET: 4 OF 11
COPYRIGHT: 1987
DWG NO:



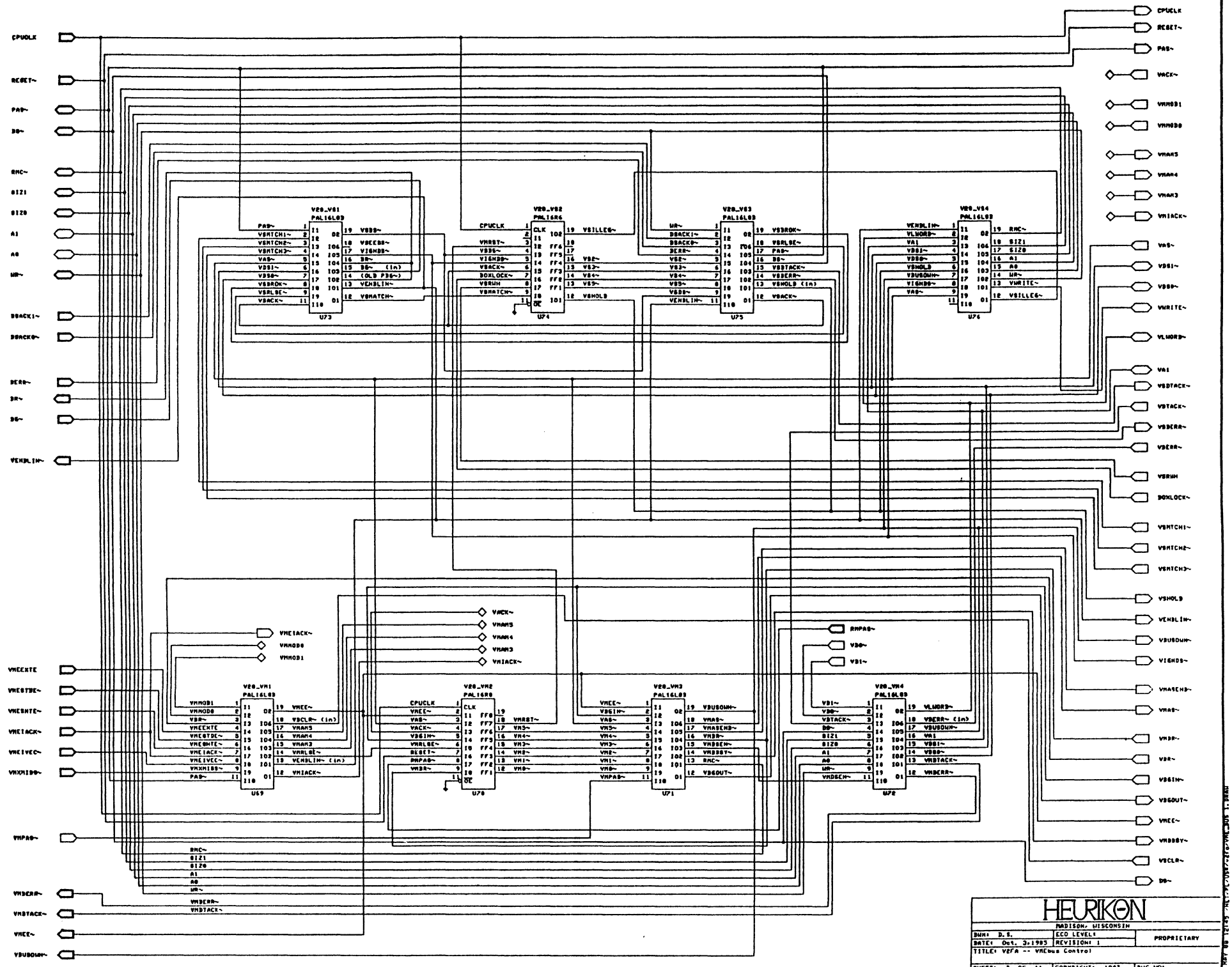
HEURIKON		
MADISON, WISCONSIN		
DWN: DHP	ECO LEVEL:	PROPRIETARY
DATE: April 5,1988	REVISION: 1	
TITLE: VZFA Ram Bank		
SHEET: 5 OF 11	COPYRIGHT: 1987	DWG NO:



HEURIKON

MADISON, WISCONSIN

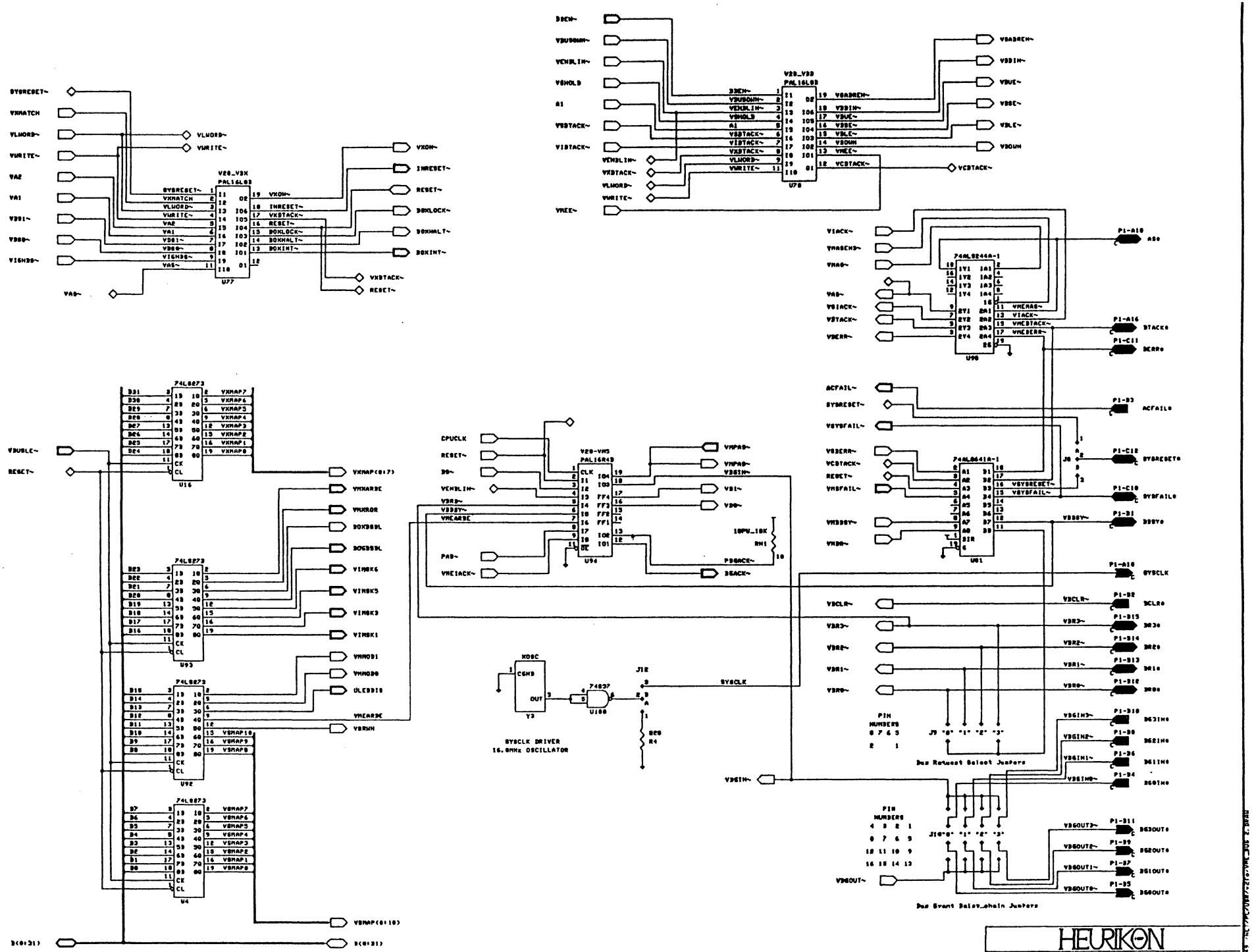
DWN: DWP	ECO LEVEL:	PROPRIETARY
DATE: April 8,1988	REVISION: 1	
TITLE: V2FA Ram Bank		
SHEET: 6 OF 11	COPYRIGHT: 1987	DWG NO:

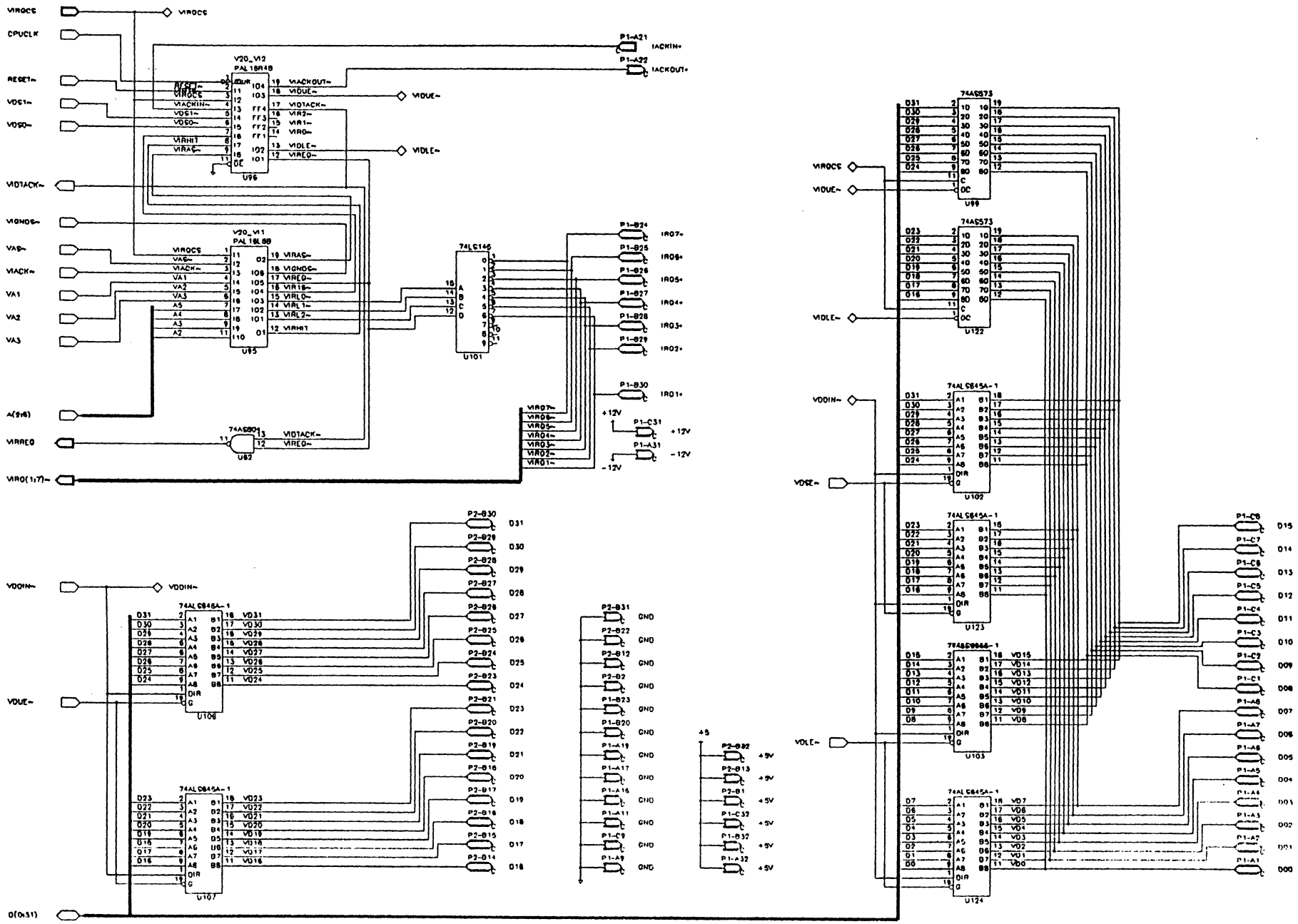


HEURIKON

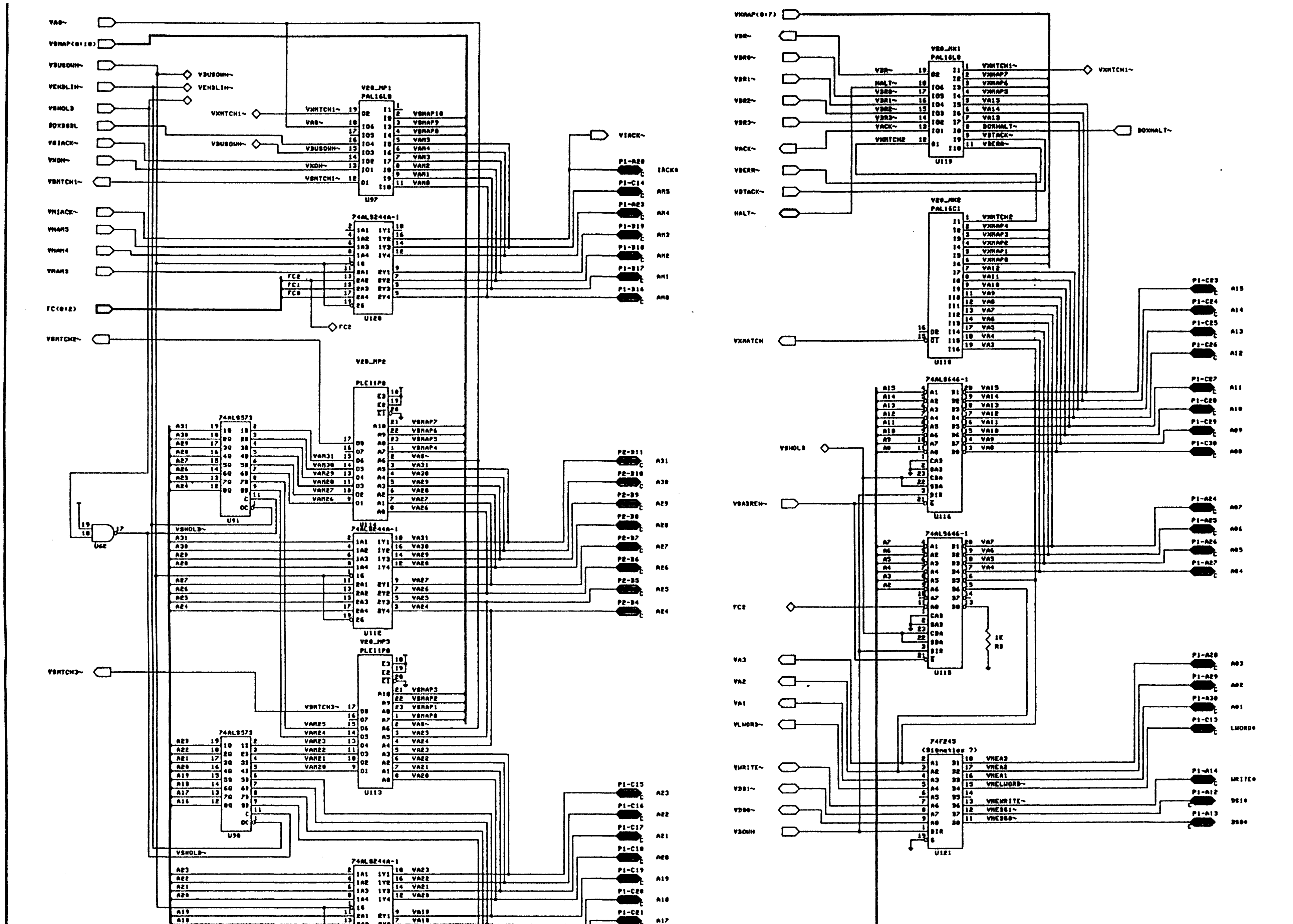
MADISON, WISCONSIN

DWN: D.S.	ECO LEVEL:	PROPRIETARY
DATE: Oct. 3,1985	REVISION: 1	
TITLE: V2FA -- VMEbus Control		
SHEET: 7 OF 11	COPYRIGHT: 1987	DWG NO:


SYSRESET~
VXMATCH
VLWORD~
VWRITE~
VA2
VA1
VB91~
VB80~
VIGNB8~
VA8~
V28_V8X
PAL16L8B
U77
VXON~
INRESET~
RESET~
BOXLOCK~
BOXHALT~
BOXINT~
VXBTACK~
RESET~
BSEN~
VBUBOWN~
VENBLIN~
VBHOLD
A1
VBBTACK~
VIBTACK~
VXBTACK~
VLWORD~
VWRITE~
VWEE~
V28_V88
PAL16L8B
U78
VBABREN~
VB3IN~
VBUE~
VB8E~
VBLE~
VBDWN
VWEE~
VCBTACK~
VBUSLE~
RESET~
74LS873
U16
VXMAP(0:7)
VMXARBE
VMXRLR
BOXB8BL
BOXB9BL
VIMBK6
VIMBK5
VIMBK9
VIMBK1
U93
VMMOD1
VMMOB0
ULEB9I8
VMEARBE
VBRWN
U92
VBMAP10
VBMAP9
VBMAP8
U4
VBMAP(8:10)
B(0:31)
CPUCLK
RESET~
B8~
VENBLIN~
VBRB~
VBBBY~
VMEARBE
PAB~
VMEIACK~
V28-VM8
PAL16R4B
U94
VMPAB~
VMPAB~
V3BIN~
VBI~
VB8~
10PU_10K
RN1
PBBACK~
BEACK~
VIACK~
VMABEN8~
VMAB~
VAB~
VBIACK~
VBTACK~
VBERR~
74LS244A-1
U98
VMERAB~
VIACK~
VMEBTACK~
VMEBERR~
P1-A10
A8e
P1-A16
BTACKe
P1-C11
BERRe
ACFAIL~
SYSRESET~
VBYBFAIL~
VBBERR~
VCBTACK~
RESET~
VMBFAIL~
VMBBBY~
VMBR~
74LS641A-1
U81
VBYBRESET~
VBYBFAIL~
P1-B3
ACFAILe
P1-C18
BYBRESETe
P1-C10
BYBFAILe
P1-B1
BBBYe
P1-A18
BYSCLK
VBCLR~
VBR3~
VBR2~
VBR1~
VBR0~
P1-B2
BCLRe
P1-B15
BR3e
P1-B14
BR2e
P1-B13
BR1e
P1-B12
BR0e
PIN NUMBERS
8 7 6 5
4 3 2 1
J9 "0" "1" "2" "3"
Bus Request Select Jumpers
VBGIN3~
VBGIN2~
VBGIN1~
VBGIN0~
P1-B10
BG3INe
P1-B8
BG2INe
P1-B6
BG1INe
P1-B4
BG0INe
VBGIN~
PIN NUMBERS
4 3 2 1
8 7 6 5
12 11 10 9
16 15 14 13
J10 "0" "1" "2" "3"
VBGOUT~
Bus Grant Daisy_chain Jumpers
VBGOUT3~
VBGOUT2~
VBGOUT1~
VBGOUT0~
P1-B11
BG3OUTe
P1-B9
BG2OUTe
P1-B7
BG1OUTe
P1-B5
BG0OUTe
K08C
CSMB
OUT
Y2
74S37
U100
J12
BYSCLK
820
R4
BYSCLK DRIVER
16.8MHz OSCILLATOR
HEURIKON
MADISON, WISCONSIN
DWN: DWP
ECO LEVEL:
PROPRIETARY
DATE: APRIL 5,1988
REVISION: 1
TITLE: V2FA -- VMEbus Control
SHEET: 8 OF 11
COPYRIGHT: 1987
DWG NO:

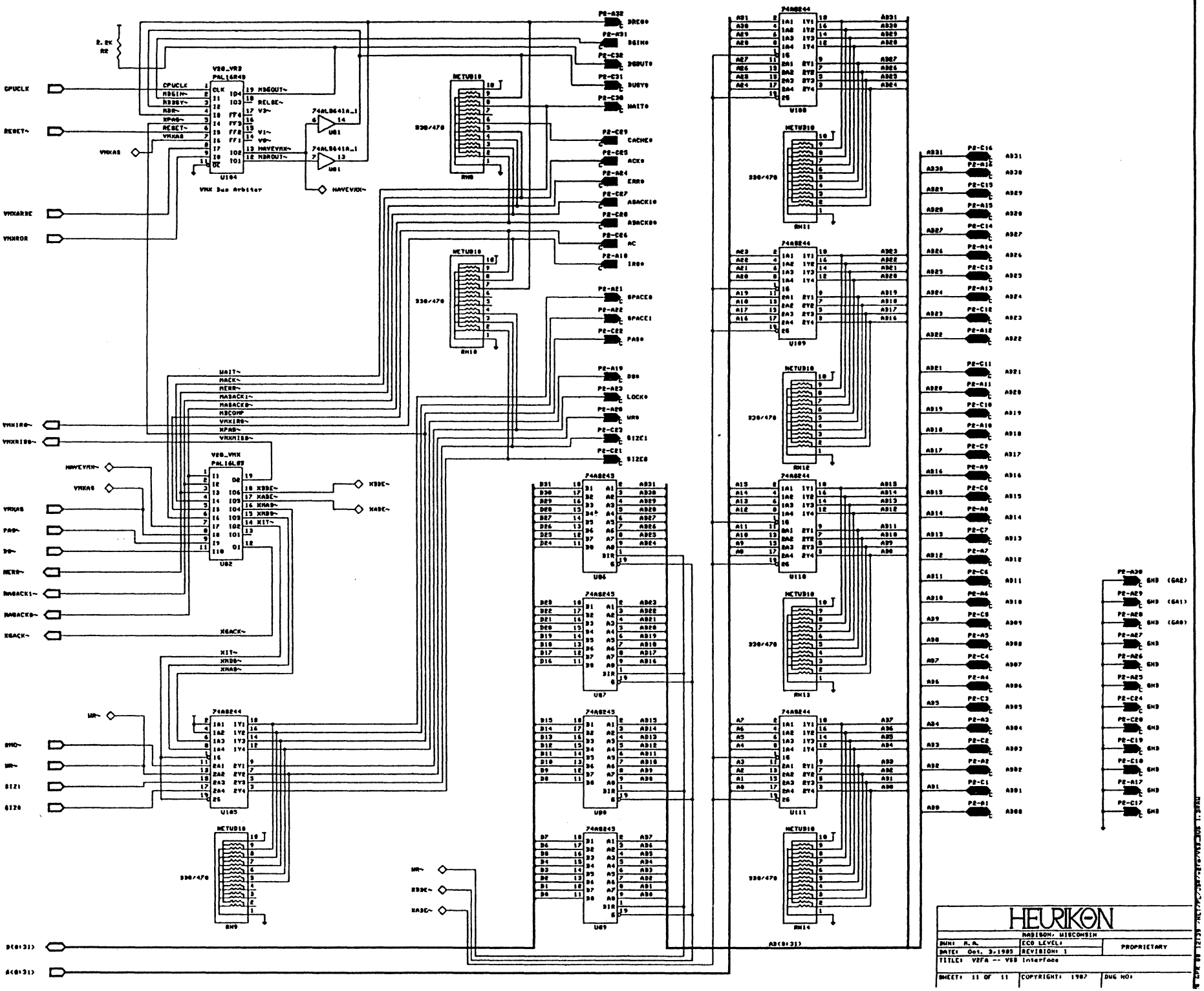


MADISON, WISCONSIN		
DWN: D.S.	ECO LEVEL:	PROPRIETARY
DATE: Oct. 3,1985	REVISION: 1	
TITLE: V2FA -- Data Buffers		



HEURIKON
MADISON, WISCONSIN

DWN: D.S.	ECO LEVEL:	PROPRIETARY
DATE: Oct. 3, 1985	REVISION: 1	
TITLE: V2FA -- VMEbus Address Buffers and Slave Maps		
SHEET: 10 OF 11	COPYRIGHT: 1987	DWG NO:



HEURIKON			
MADISON, WISCONSIN			
DWN: R.A.	ECO LEVEL:	PROPRIETARY	
DATE: Oct. 3,1985	REVISION: 1		
TITLE: V2FA -- VSB Interface			
SHEET: 11 OF 11	COPYRIGHT: 1987	DWG NO:	

Heurikon Corporation
8310 Excelsior Drive
Madison, WI 53717

Customer Support: 1•800•327•1251

Sales: 1•800•356•9602

Heurikon Part Number 001M513