3.5 Liste des commandes (Extraits)

Le tableau qui suit fournit de façon succinte la fonction de chaque commande et les différents équipements qui lui sont associés.

Commande	Fonctions	PMV	Camé -ras	UMT
A	Affichages : Transfert de mesures ou états en format détaillé 'A'			X
B	Bilans : Transfert de mesures ou états en tables semi-formatées "B"			X
BK	Break : Interruption des actions et de la réponse	X	X	
CFAL	Configurations des modes et paramètres de conditions d'Alerte	X		X
CFID	Configuration des identifiants et mots de passe des utilisateurs	X	X	X
CFMK	Configuration des macros de mouvement de la commande K		X	
CFMP	Configuration des macros de positionnement de la commande P	X		
CFPK	Configuration des paramètres permanents de la commande K		X	
CFPP	Configuration des paramètres permanents de la commande P	X		
DT	Lecture ou réglage de mise à l'heure et au jour	X	X	X
ID	Identification de sécurité par mot de passe	X	X	X
INIT	Réinitialisation générale des matériels et logiciels d'un site	X	X	X
K	Commandes de caméra		X	
M	Mesures : Transfert de mesures en format compact "M"			X
P	Positionnement des modules variables d'un équipement	X		
SET	Configuration logique d'un port de communication	X	X	X
SETU	Configuration physique des UARTS, du média utilisé, des temporisations	X	X	X
SETV	Configuration des signaux vidéo		X	
ST	Paramètres permanents et indicateurs de dysfonctionnement	X	X	X
ST AL	Status des circuits d'envois d'alertes	X		X
ST LCOM	Liste des Commandes et paramètres utilisables	X	X	
ST LCPI	Liste des caractéristiques d'un pilote informatique d'équipement	X	X	
TRACE	Trace des fichiers internes d'analyse et de fonctionnement	X	X	X
VIDE	Interruption d'une réponse en cours (ou commande "vide")	X	X	X

4 : Interprétation des commandes et paramètres

4.1 Interprétation des paramètres

L'interprétation des paramètres d'une commande s'effectue de gauche à droite. Dès qu'un paramètre est découvert comme invalide (étiquette, ou argument) la commande est refusée. Lorsqu'un paramètre avec un argument valide est répété au sein d'une même commande, la commande est acceptée. C'est l'argument du paramètre le plus à droite qui sera pris en compte.
Ainsi lorsqu'un paramètre est répété, si l'un d'entre eux a un argument invalide, la commande est refusée.
Exemples : Les exemples suivants sont relatifs au pilotage d'un PIC (Pilote Informatisé de Caméra) dont le détail des commandes est donné dans l'annexe 8.

Q : **STOP**
// *Pas de réponse car la commande est absente du dictionnaire général*
Q : **P AM=1.0 AF="xx"**
// *Pas de réponse car la commande est absente du dictionnaire d'un PIC*
Q : **KV RH=D/P RH=D/N**
R : **!**
// *La tourelle tournera en sens négatif*

4.2 Empilage des commandes

Certaines commandes peuvent aboutir à une réalisation complète après un temps assez long, une commande de rotation lente en mode absolu par exemple. Lorsque l'action relative à une commande est en cours, une nouvelle commande reçue peut aboutir :

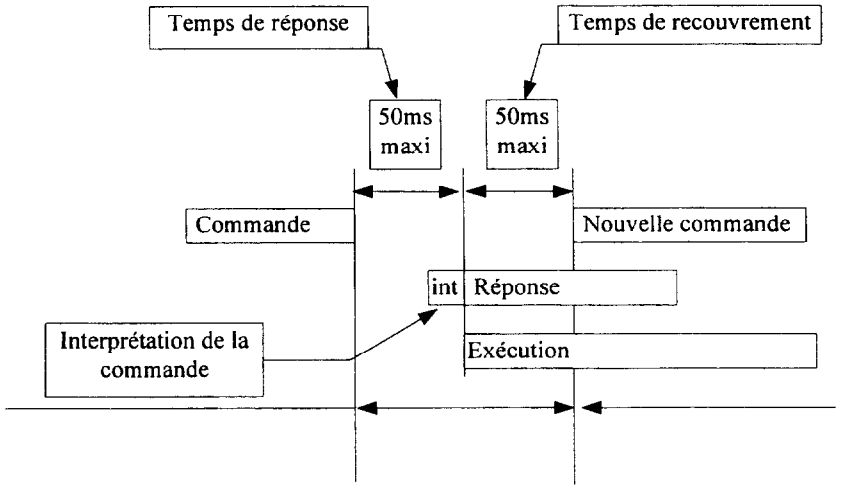
- à l'exécution parallèle d'une nouvelle action,
- à l'interruption de l'action en cours et l'exécution de la nouvelle commande,
- au refus de la nouvelle commande.

Ces mécanismes sont détaillés dans le paragraphe "Temps de réponse et temps de recouvrement" ci-après. Il n'y a jamais empilage de commande. (ce qui n'exclut pas les éventuels empilages relatifs aux couches basses)

4.3 Temps de réponse et temps de recouvrement

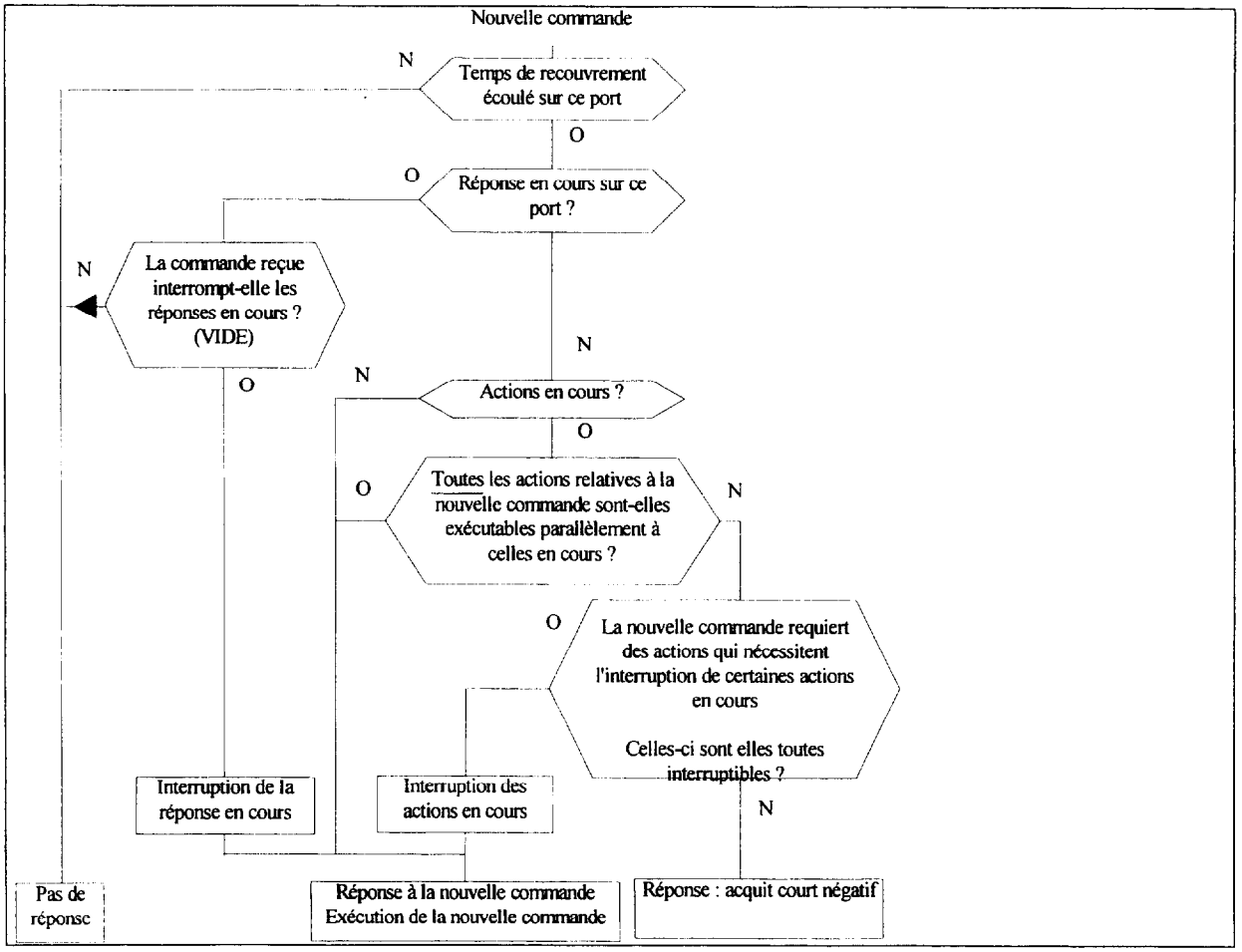
Les contraintes temporelles énoncées ci-dessous s'entendent relatives aux couches 6 et 7 du modèle OSI. Elles n'intègrent pas les délais qui peuvent être apportés par les protocoles des couches basses.

Comportement face à une nouvelle commande n'interrompant pas l'exécution de la première



On distingue ici deux types de commandes, celles générant une exécution et une réponse, et celles ne générant qu'une réponse (l'exécution consiste uniquement à répondre).
Une commande reçue accompagnée du paramètre R=N génère un comportement ordinaire hormis la durée de réponse nulle.

Dès sa réception complète, la commande est interprétée puis elle génère une réponse. Le temps de réponse (Tr), défini comme le temps séparant la fin de la commande et le début de la réponse, est au **maximum de 50ms**, pour les commandes d'exploitation, quelle que soit la classe du Pilote Informatisé. Au-delà, on considère que le Pilote ne répond pas.
Si la commande est acceptée, son exécution, débute dès le début de l'envoi de la réponse.
La durée effective de la réponse sera d'au **maximum 110%** de sa durée théorique, définie par les différents paramètres de la transmission.
A partir du début de la réponse s'écoule un temps, dit de recouvrement, fixé à un **maximum de 50ms**. Durant ce temps, une nouvelle commande partiellement ou complètement acheminée n'est pas interprétée et ne fait donc pas l'objet de réponse ni d'exécution. Au-delà, la commande est interprétée ; plusieurs situations peuvent survenir. L'ensemble des comportements est décrit dans le diagramme et le tableau qui suivent.



L'action d'une commande de configuration se termine lorsque tous les paramètres relatifs a cette commande sont devenus effectifs (ex: CFx) ou que l'état requis devient efficace (ex: INIT).
Une commande de configuration inhibe et arrête toute commande de mouvement en cours et refuse toute nouvelle commande de mouvement tant que l'action de configuration n'est pas terminée.
L'action d'une commande de positionnement se termine lorsque le module a atteint l'état requis.
L'action d'une commande de lecture se termine lorsque le PIC a préparé tous les éléments de la réponse ou les a transmis aux couches basses. Les réponses en cours ne sont interruptibles que par la commande VIDE.
Exemples : (voir ANNEXE 8 : LCR CAMERA)

Q :	**KV RH=A/C/2400/L**	// Demande de positionnement à l'azimut 240 en vitesse lente.
R :	**!**	// Le cycle est interrompu au profit de la rotation
Q :	**KV RH=R/N/300/R**	// Demande de positionnement à 30° à gauche en vitesse rapide.
R :	**!**	// La rotation en mode absolu s'interrompt dès la réception de la commande, au profit de la rotation relative.
Q :	**INIT**	// Demande d'initialisation.
R :	**!**	// Le mouvement s'arrête.

ANNEXE 3 : CARTE MVME 162

Requirements

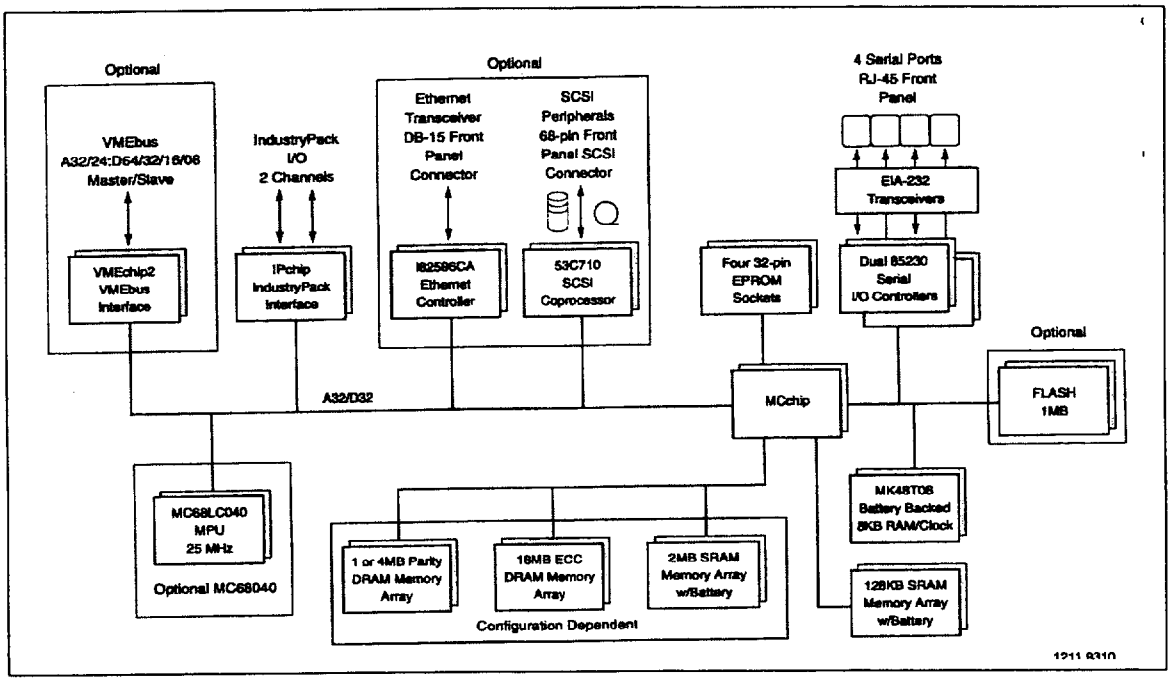
These boards are designed to conform to the requirements of the following documents:

- VMEbus Specification (IEEE 1014-87)
- EIA-232-D Serial Interface Specification, EIA
- SCSI Specification, ANSI
- IndustryPack Specification, GreenSpring

Features

- 25 MHz MC68040 or MC68LC040 Microprocessor
- 1 or 4 MB of DRAM with parity protection on a mezzanine module, or 16 MB ECC DRAM on a mezzanine board
- 128 KB of SRAM with battery backup, or 2 MB SRAM on a mezzanine board with battery backup
- Four JEDEC standard 32-pin DIP PROM sockets
- One Intel 28F008SA IM x 8 Flash memory device with write protection. Optional
- Status LEDs for FAIL, RUN, SCON, and FUSES
- 8K by 8 Non-Volatile RAM (NVRAM) and time of day (TOD) dock with battery backup
- RESET and ABORT switches
- Four 32-bit Tick Timers and Watchdog Timer (in the MCchip ASIC) for periodic interrupts
- Two 32-bit Tick Timers and Watchdog Timer (in the VMEchip2 ASIC) for periodic interrupts
- Eight software interrupts (for MVME162LX versions that have the VMEchip2)
- I/O
  - Optional SCSI Bus interface with DMA
  - Four serial ports with EIA-232-D interface (serial port controllers are the Z85230 chips)
  - Optional Ethernet transceiver interface with DMA
  - Two IndustryPack interfaces
- VMEbus interface
  - VMEbus system controller functions
  - VMEbus interface to local bus (A24/A32, D8/D16/D32 (D8/D16/D32/D64 BLT) (BLT = Block Transfer)
  - Local bus to VMEbus interface (A16/A24/A32, D8/D16/D32)
  - VMEbus interrupter
  - VMEbus interrupt handler
  - Global CSR for interprocessor communications
  - DMA for fast local memory
  - VMEbus transfers (A16/A24/A32, D16/D32/D64 BLT)

MVME 162LX Block Diagram



Memory Map

Address Range	Devices Accessed	Port Width	Size	Software Cache Inhibit
Programmable	DRAM on Parity Mezzanine	D32	1MB-4MB	N
Programmable	DRAM on ECC Mezzanine	D32	16 MB	N
Programmable	On-Board SRAM	D32	128 KB	N
Programmable	SRAM on Mezzanine	D32	2 MB	N
Programmable	VMEbus A32/A24	D32/D16	--	?
Programmable	IP_a Memory	D32-D8	64KB-8MB	?
Programmable	IP_b Memory	D32-D8	64KB-8MB	?
$FF800000-$FF9FFFFF	Flash/EPROM	D32	2 MB	N
$FFA00000-$FFBFFFFF	EPROM/Flash	D32	2 MB	N
$FFC00000-$FFDFFFFF	Not Decoded	D32	2 MB	N
$FFE00000-$FFE1FFFF	On-Board SRAM Default	D32	128 KB	N
$FFE80000-$FFEFFFFF	Not Decoded	--	512 KB	N
$FFF00000-$FFFEFFFF	Local I/O Devices (Refer to next table)	D32-D8	878 KB	Y
$FFFF0000-$FFFFFFFF	VMEbus A16	D32/D16	64 KB	?

ANNEXE 4 : CARTE SUPPORT I4000

Introduction
The I4000 is a modular passive peripheral-controller for the VME-bus. Up to four M-modules can be connected to the board to form a peripheral-controller.

1 Technical overview

The I4000 has the following features:
- 4 M-module slots
- One VMEbus slot needed
- short or standard VMEbus addressing (A16/A24)
- byte or word data transfer (D08/D16)
- D08(O) VMEbus interrupter
- peripheral I/O connections either up front or via the VME-P2 connector.
- Additional mounting holes to secure the modules

2 Block diagram

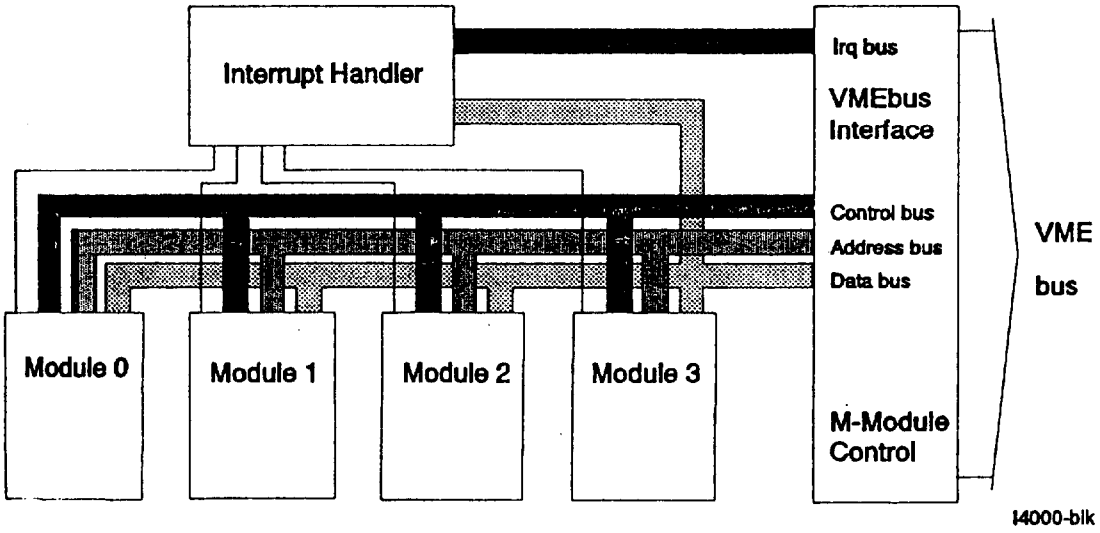


Figure 1 I4000 Block Diagram

3 Address interface

The I4000 can be selected to accept either short addressing, using 16 bit addresses (A16) or standard addressing, using 24 bit addresses (A24).

When using short addressing up to 32 I4000 boards can be put in a single VMEbus System since the lower 11 address lines (A0-A10) are used internally on the I4000. If standard addressing is used, a larger address space is available to select the board.
The memory space occupied by the I4000 is $800 bytes long. This memory space is equally spread across the four modules, so each module occupies $200 bytes. From these $200 bytes address space half is used by the interrupt controller. The address space for each module is therefore $100 bytes long (8 bit addressing).

The address space of the I4000 is selectable using dip switches (SWI and SW2) which are accessible even when the four module slots are occupied. When a switch is "on" the corresponding address line is a logical "zéro" and when a switch is "off" the corresponding address line is "one".

Address modifier AM4 can be used to select either standard or short memory addressing. If AM4 is "one" (the switch is in the off position) standard addressing is selected. If AM4 is "zero" (switch on) short addressing is selected. When using short addressing, the address lines A16-A23 are 'don't care'.

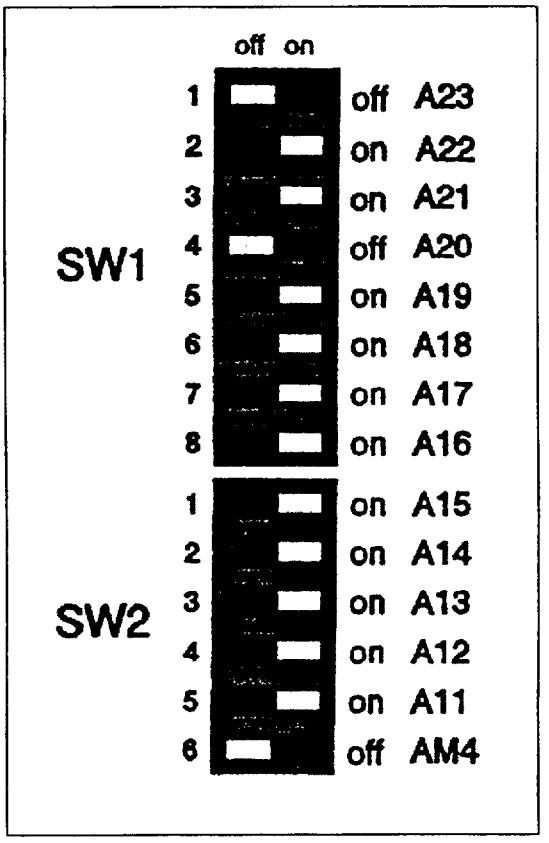


Figure 2 Address Selection

EXAMPLE :
Base Address $900000
(Standard address space)

4 The interrupter

The I4000 has a special interrupter which is largely compatible with the M68C153. The interrupter is a so called D08(O) type interrupter which means that the interrupter during an interrupt acknowledge cycle will put a status byte on the data line D0-D7. The interrupter has one interrupt level (IRQ1-IRQ7) selectable by dip switches (SW3) for all the modules.
See Figure 3 Switch value L0 = 1
Switch value L1 = 2
Switch value L2 = 4

Each module can generate its own vector.
There are two classes of interrupters which are both supported by the I4000: release on acknowledge (ROAK) or release on register access (RORA). The ROAK interrupter negates its interrupt request line in response to an interrupt acknowledge cycle. This mechanism will work with all handlers. The RORA interrupter releases its request when the handler accesses an on-board register during the interrupt service routine. The handler performs the acknowledge cycle but the interrupter does not immediately negate its request. Sometime during the service routine the handler will have to write to a register on the interrupter which causes it to negate the request.

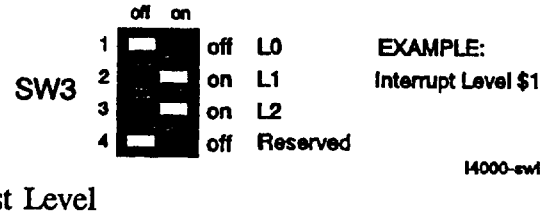


Figure 3 Interrupt request Level

5 Software issues

5.1. Address mapping

As mentioned before the address space occupied by the I4000 board is $800 bytes (A0-A10). These $800 bytes are divided into 4 identical spaces. Every $200 bytes block is assigned to a module slot. The first $100 byte address space is assigned to the module itself and the second $100 byte is used for the access part of the interrupt controller.

Using this method of address decoding provides an identical address map for each module on the I4000 board. This makes writing the software easier since just the base address of the module, not the base address of the I4000 has to be known. Every module has its own Interrupt-Vector and Interrupt-Control register.
The $100 bytes from each module used to access the interrupt controller are not completely decoded. Both registers of the Interrupt controller are mirrored several times within the $100 bytes address space. If the module itself the $100 bytes completely decodes, depends on the used module.

Address Map of the I4000:

$000..$0ff **$101** **$103**	**Module** **Control register (CR0)** **Vector register (VR0)**	**Module 0**
$200..$2ff **$301** **$303**	**Module** **Control register (CR1)** **Vector register (VR1)**	**Module 1**
$400..$4ff **$501** **$503**	**Module** **Control register (CR2)** **Vector register (VR2)**	**Module 2**
$600..$6ff **$701** **$703**	**Module** **Control register (CR3)** **Vector register (VR3)**	**Module 3**

The base address of a module can be calculated using the following formula :
ModuleBaseAddress = I4000BaseAddress + ModuleNumber * $200

EXEMPLE

The installed base address of the I4000 is $800000. A module is fitted into slot 2. The module base address is then $800000 + 2 * $200 == $800400. When using a 32 bit master which accesses the standard address space at address $ffxxxxxx, then the module will be accessed at address $ff800400. The corresponding interrupt control register address is then $ff800501.

5.2. The interrupt controller

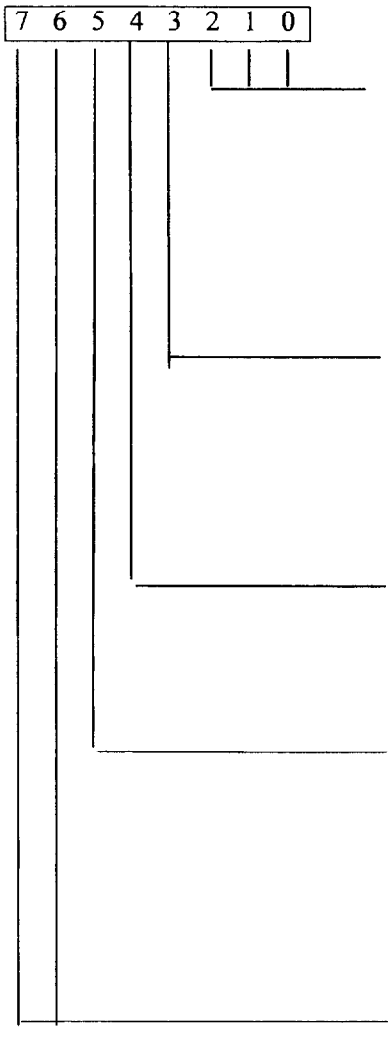
5.2.1. Functional description

The Interrupt controller used with the I4000 is largely compatible the MC68153 Interrupt controller from Motorola.
The interrupt controller provides a means for the modules to ask for an interrupt of the processor activity and receive service from the processor. The Interrupt controller on the I4000 acts as an interface device requesting and responding to interrupt acknowledge cycles for up to 4 independent modules.

5.2.2. Register description

The Interrupt controller of the I4000 contains 8 programmable registers. There are four control registers (CRO-CR3) that govern operation of the Interrupt controller and four vector registers (VRO-VR3) that contain the vector data used during an interrupt acknowledge cycle. Every module is assigned one register pair.

5.2.3. Control register



L2-LO (Interrupt level, dummy bits)

The least significant 3-bit field of the register determines the level at which an interupt will be generated.
These three bits are NOT used in the I4000 interrupt controller since the interrupt level for all the four modules is selected via dip switches. However to maintain software compatibility with the A201 it is advisable to program these 3 bits with the same interrupt level as the dip switches.

IRAC (Interrupt Auto-Clear)

If the IRAC is set (Bit 3), IRE (Bit 4) is cleared during an interrupt acknowledge cycle responding to this request. This action of clearing IRE disables the module interrupt request. To re-enable the module interrupt request associated with this register, IRE must be set again by writing to the control register.

IRE (Interrupt enable)

This field (Bit 4) must be set (high level) to enable the module interrupt request associated with the control register. Thus, if the module asserts IRQ and IRE is cleared, no interrupt request to the VMEbus will be generated.

X/IN* (External/Internal)

Bit 5 of the control register determines the response of the I4000 interrupt controller during an interrupt acknowledge cycle. If the X/IN* bit is clear (low level) the Interrupt Controller will respond with vector data and a DTACK* signal, i.e., an internal response. If X/IN* is set, the vector is not supplied and no DTACK* is given by the Interrupt Controller, i.e., an external module should respond.

RESERVED

These two bits are not used in the current implementation of the Interrupt Controller and reserved for future use.

5.2.4. Vector registers

Each module interrupt input has its own associated vector register. Each register is 8 bits wide and supplies a data byte during its interrupt acknowledge cycle if the associated External/Internal (X/IN*) control register bit is clear (zero). This data can be status, identification, or address information depending on system usage. The information is programmed by the system user.

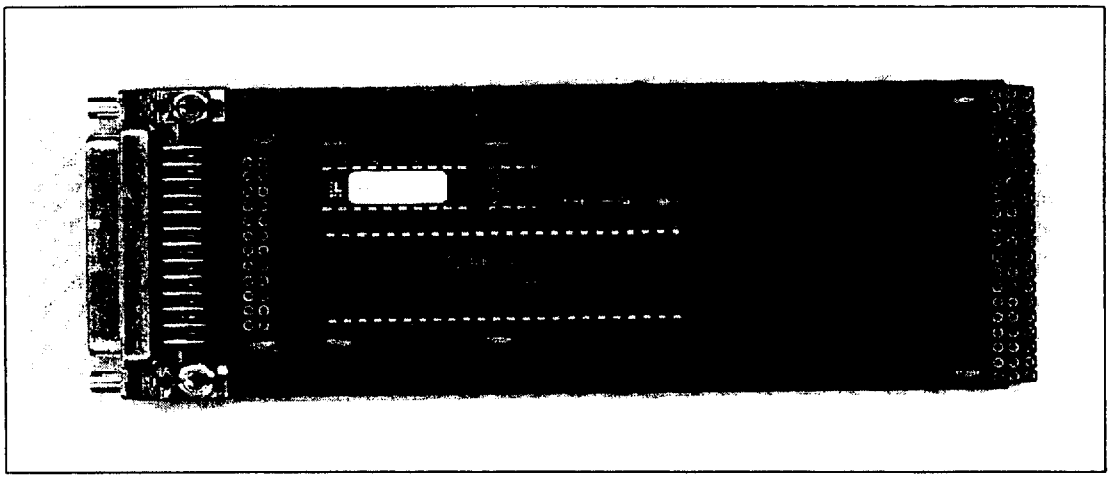
5.2.5. Interrupt controller reset

When a VMEbus reset is applied, the control registers of the I4000 interrupt controller are set to all zeros (low). The vector registers however are uninitialized and should be programmed before use.

ANNEXE 5 : M-MODULE M11

The M11 is an M-Module for handshaked input and output of binary signals and for generation of time-controlled interrupts.
The 16 I/O lines of the M11 can be used as two separate 8-bit ports or as one 16-bit port. In addition, there is a 2-bit port.

Control of the I/O lines is via the four handshake lines. Depending on the operation mode, the four handshake lines generate an interlocked handshake, a pulsed handshake, interrupt inputs or simple inputs.



Features

TTL I/O :
- 18 TTL inputs/outputs

68230 Parallel Interface Timer :
- 68000 bus compatible
- port modes include:
  - bit I/O
  - unidirectional 8-bit and 16-bit
  - bidirectional 8-bit and 16-bit
- programmable handshaking options
- 24-bit timer
- five separate interrupt vectors
- separate port and timer interrupt service requests

Input Voltages and Currents :

- 2..5V; 0..10 µA (high level)
- -0.3..+0.8V; 0.1..1mA (low level)

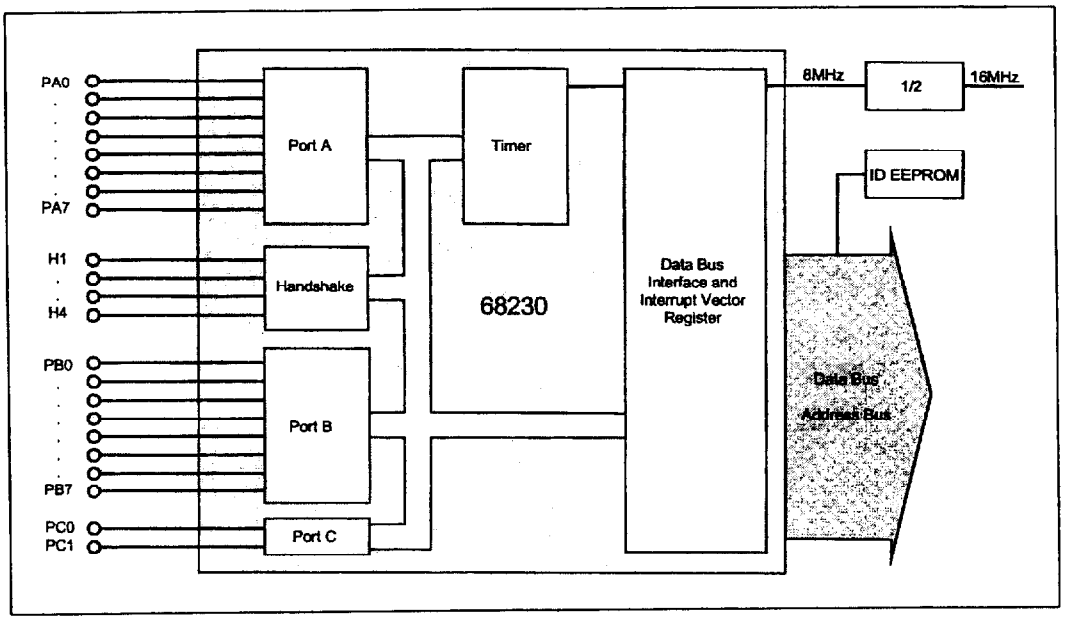
Output Voltages and Currents :
- 2.4..5V; 0.15mA max. (high level)
- 0..0.5V; 2.4mA max. (low level)

Peripheral Connections :
- via front panel on a shielded 25-pin D-Sub connector (female)
- via base board connector

M-Module Characteristics :
- A08, D08, INTC, IDENT

Software Support :
- MDIS driver concept

Block Diagram



Technical Data

Electrical Specifications

Input voltages and currents	input voltage "high" min. 2V, max. 4.75V input voltage "low" min. -0.3V, max. 1.8V input leakage current max. 10µA
Output voltages and currents	output current in "off-state" min. -0.1mA, max. -1mA output voltage "high" min. 2.4V (load < -0.15mA) output voltage "low" max. 0.5V (load < 2.4mA)
Supply voltage range	+5V: 4.85V..5.25V / 200mA typ.
Mean Time Between Failures	120,000h/50°C

Mechanical Specifications

Connector types	25-pin D-Sub receptacle: • according to DIN41652/MIL-C-24308, with thread bolt UNC 4-40 • mating connector: 25-pin D-Sub plug according to DIN41652/MIL-C-24308, available for ribbon cable (insulation piercing connection), hand-soldering connection or crimp connection
Dimensions	conform to M-Module Standard
Weight	tbd.

1 Installation and Setup

1.1 Getting Started

To test proper functioning of the module you can proceed as follows:

- Install the system with the M-Module base board but without the M11.
- Test the base board.
- If O.K., plug the module into slot 0 of the M-Module base board.
- Load a suitable debugger.
- Access the base address as set, reading word by word.
- If a bus error is occurring now, the module is not plugged properly.
- Reading a word from the base address plus 0x0A should yield the value 0x000F.
- Observe the installation manuals for operating system dependent software.

1.2 Connection of the Module

1.2.1 Power Supply

Power supply to the logic part is done via the base board. The necessary voltage is +5V.

1.2.2 Peripherals

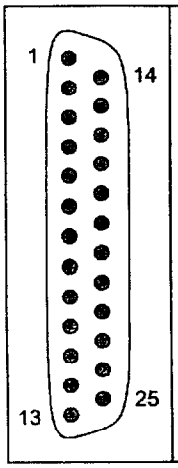
There are two possibilities for connecting peripherals:

- connection via 25-pin D-Sub connector or
- connection via the base board.

When a base board with a 96-pin DIN 41612 PCB connection is used for peripheral signals (for example a 6U VMEbus base board), these are fed to the module through the female 24-pin connector. You can connect up to four 21-pin connectors to the 96-pin connector (cf. base board manual). When these connectors are used, for each module three pins of the DIN 41612 PCB connector cannot be used. The pin numbers for the 96-pin connector shown below are valid for module slot number 3. If other module slots (2, 1 or 0) are used, the value 8, 16 or 24 must be added as appropriate.

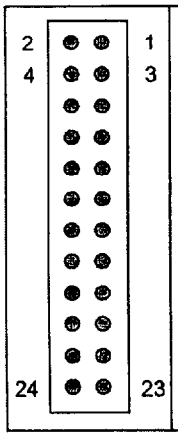
The M11 module has the following pin assignments:

Table 1a: Pin Assignment of the female 25-Pin D-Sub Connector P1



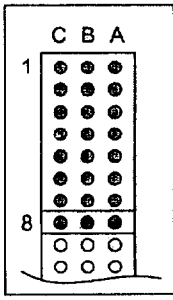
1	PA0	14	PA1
2	PA2	15	PA3
3	PA4	16	PA5
4	PA6	17	PA7
5	PB0	18	PB1
6	PB2	19	PB3
7	PB4	20	PB5
8	PB6	21	PB7
9	H1	22	H2
10	H3	23	H4
11	GND	24	VCC
12	PC0	25	PC1
13	GND		

Table 1b: Female 24-Pin Connector P2



2	PA1	1	PA0
4	PA3	3	PA2
6	PA5	5	PA4
8	PA7	7	PA6
10	PB1	9	PB0
12	PB3	11	PB2
14	PB5	13	PB4
16	PB7	15	PB6
18	H2	17	H1
20	H4	19	H3
22	PC0	21	GND
24	VCC	23	PC1

Table 1c: Signal Correspondence between 24-Pin Module and 96-pin Base Board Connector



	C	B	A
1	PA2	PA1	PA0
2	PA5	PA4	PA3
3	PB0	PA7	PA6
4	PB3	PB2	PB1
5	PB6	PB5	PB4
6	H2	H1	PB7
7	GND	H4	H3
8	VCC	PC1	PC0

1.2.3 Host Interface

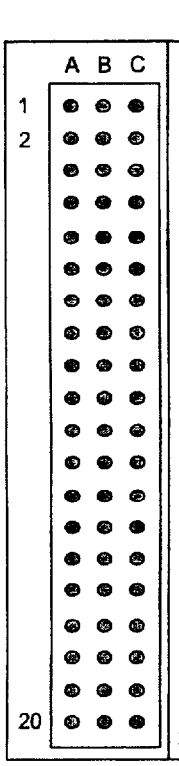
The M11 module supports the following pin assignment of the 60-pin base board interface connector:

Note: Only two rows - A and B – of the 60-pin connector are mounted on the M11.

Table 2: Signal Mnemonics

Signal	Direction	Function
PA0..7	in/out	port A lines 0..7
PB0..7	in/out	port B lines 0..7
H1..4	in/out	handshake lines
PC0..1	in/out	port C lines 0..1
GND	-	ground
VCC	-	+5V supply

Table 3: Pin Assignment of the 60-Pin Female Connector



	A	B	C
1	/CS	GND	-
2	A01	+5V	-
3	A02	-	-
4	A03	-	-
5	A04	GND	-
6	A05	-	-
7	-	-	-
8	A07	GND	-
9	-	D00	-
10	-	D01	-
11	-	D02	-
12	-	D03	-
13	-	D04	-
14	-	D05	-
15	-	D06	-
16	-	D07	-
17	-	-	-
18	/DTACK	/WRITE	-
19	/IACK	/IRQ	-
20	/RESET	SYSCLK	-

2 Address Organization

When using the driver software supplied, you do not need to be familiar with the hardware of the M11 module in detail. However familiarity with the address organization of the board is essential if you wish to write your own software for the module or do low-level development.

The 256-byte I/O area of the M11 module is hardware-mapped. The address at which individual functions can be addressed from the base board is computed from the base address of the module plus the address in the following table.

The address mapping mirrors the 68230 internal registers.

The address space is not completely filled by the module's registers, therefore they are mirrored four times in the 256-byte address space.

Table 4 : Address Map

Note :
The registers of the 68230 are shown in grey color in the table

Address	D15...D8	D7...D0
0x00	-	
0x02	-	
0x04	-	
0x06	-	
0x08	-	
0x0A	-	
0x0C	-	
0x0E	-	
0x10	-	
0x12	-	
0x14	-	
0x16	-	
0x18	-	
0x1A	-	
0x20	-	
0x22	-	
0x26	-	
0x28	-	
0x2A	-	
0x2E	-	
0x30	-	
0x32	-	
0x34	-	
0xFE	-	M-Module Identification (r/w)