ISTITUTO NAZIONALE DI FISICA NUCLEARE

Sezione di Bologna

INFN/TC-89/1
21 Febbraio 1989

I. D'Antone, P. Matteuzzi, S. Meneghini, G. Sanzani:

A QBUS-VMEBUS ADAPTER

Servizio Documentazione
dei Laboratori Nazionali di Frascati

INFN - Istituto Nazionale di Fisica Nucleare
Sezione di Bologna

INFN/TC-89/1
21 Febbraio 1989

A QBUS-VMEBUS ADAPTER.

(I.D'Antone, P.Matteuzzi, S.Meneghini, G.Sanzani)
Istituto Nazionale di Fisica Nucleare
Bologna, Italy

ABSTRACT

A connection between a microVAX and a VME crate has been realized using an interface Qbus I/O mapped.

This last is more convenient for small data block transfer because it does not require initialization time, as required for example in a DMA transfer.

The link has been tested and has been obtained an effective rate of about 0.3 Mbytes/sec.

1.-INTRODUCTION

Many recent experiments in the high energy particle physics use the VME standard for their data acquisition systems, because it is cheap and modular and it allows a high hardware/software flexibility in the system design.

On the other hand the microVAX machines, which are powerful and software compatible with the family of VAX computers, are also widely used.

The diffusion of these two standards and their contemporary use at different levels in the on-line applications requires a fast link between the two systems.

A link Qbus-VMEbus can be realized by using interfaces with DMA capability to transfer large data block or by using I/O mapped or memory mapped interfaces to transfer small data block or to send short messages between microvax machine and a VME CPU; this last is more convenient for small data block transfer because it does not require initialization time.

We have realized a connection using an I/O mapped interface developed in our laboratory.

In the next chapters we describe the busses features and the detail of the interface used: then we give a description of the software developed for the link and the results obtained.

2.-DATA BUSSES.

The VME bus structure is based on a 16 bit data bus and a 24-bit address space non multiplexed, both expandable to 32-bit.

The VME bus has a master/slave asynchronous data transfer format; this allows multiple bus masters (typically processors and DMA controllers), and besides memories and peripherals of different speeds can be used without slowing the bus.

Data transfer rate as high as 20 Mbytes/s in the expanded 32 bit configuration is possible.

Any bus allowing multiple mastership must be capable of allocating (arbitrating) which master has the use of the data transfer at a given time. The VMEbus arbitration consists of prioritized request levels and prioritized grant levels.

The bus interrupt structure, which is vectorized, has also options which are useful to multiple master systems. There are seven interrupt request lines by which modules may ask for servicing by an interrupt handler.

The Q22 bus implements 22-bit addressing and 16-bit data transfer. The bus operates asynchronousiy with a master/slave relationship. The bus master starts a bus transaction and the slave device responds acknowledging this operation in progress and exchanging data with the bus master.

In block mode multiple words can be transferred sequentially in both directions starting from a single bus address.

The bus interrupt structure is vectorized. There are four prioritized interrupt levels. When the processor acknowledges the interrupt request it receives from the device a vector address of the service routine.

3.- THE HARDWARE SYSTEM.

A way to realize a connection between two computers is obtained by adapting the bus cycles: in this way a CPU can see directly the address space of another CPU. An interface with this features has been developed in the INFN electronic laboratory of Bologna to link the Qbus and the VMEbus. A schematic view of this link is shown in fig.1.

To realize the connection we use a board plugged in the microvax backplane and carrying out some Qbus lines in two connectors as indicated in fig.2. Two flat cables carries this lines to the VME interface.

The interface adapts each Qbus cycle to the equivalent VME cycle so a computer Qbus based, as a microVAX or a vaxstation GPX, can directly read or write a memory or an I/O board in the VME.

The interface, located in the VME crate, is mapped on the microvax as a part (1 kbytes) of the Qbus I/O space. Therefore the microvax sees directly 1kbytes of VME locations and the base address of this block can be changed writing an offset register (Base Address register, BAR) in the interface.

The fig.3 shows as the VME is mapped on the I/O Qbus.

The interface is seen as a Qbus slave from the microvax, but when it gains the VME bus it works as a VME master; therefore in this system the microvax behaves as a CPU master of the VME crate.

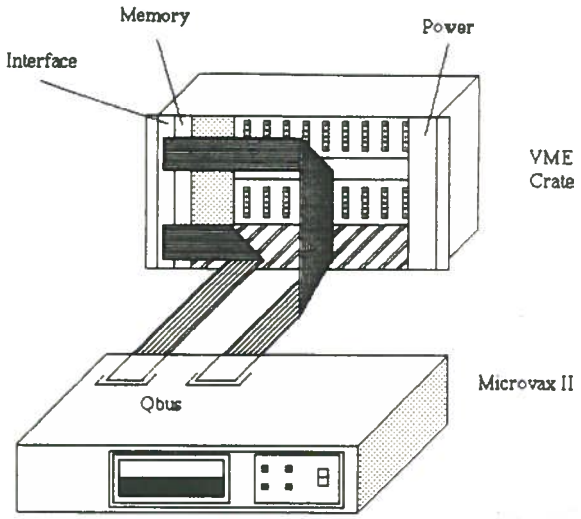

Fig. 1

Schematic view of the I/O mapped interface-microvax connection

pin	Q2 & Q3	Q1 & Q4
1:	GND	GND
3:		BSS7
5:		BWTBT
7:		BDAL16
9:		BDAL14
11:		BDAL13
13:		BDAL12
15:		BDAL11
17:		BDAL10
19:		BDAL9
21:		BDAL8
23:		BDAL7
25:		BDAL6
27:	BIACKO	BDAL5
29:	BIRQ4	BDAL4
31:		BDAL3
33:	BSYNC	BDAL2
35:		BDAL21
37:	BDIN	BDAL20
39:		BDAL19
41:	BRPLY	BDAL18
43:		BDAL1
45:	BDOUT	BDAL0
47:	BIRQ6	BDAL17
49:	BIRQ5	BDAL16

Fig. 2

Q22 bus lines used

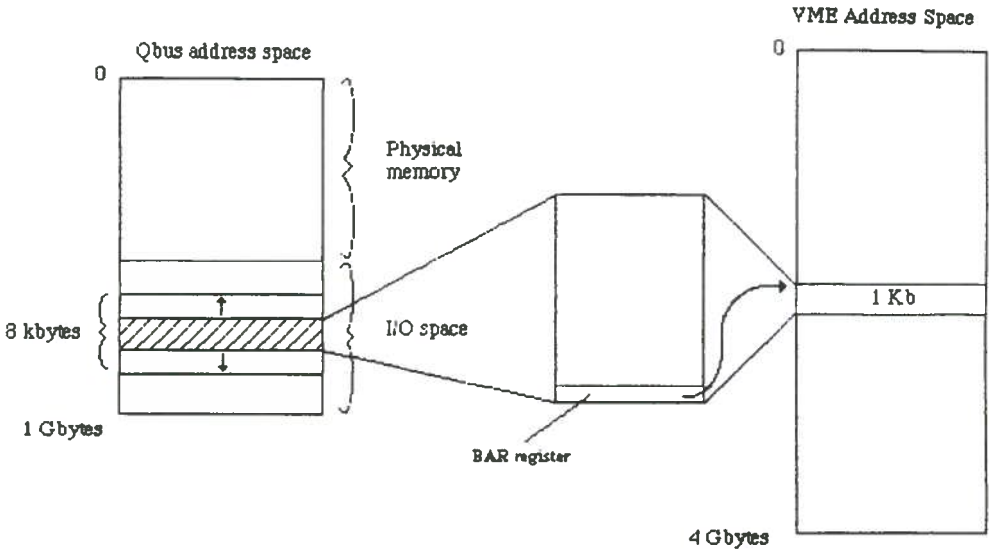

Fig. 3

VME and QBUS Address spaces relation

The board contains a number of data paths, address decoders, multiplexer, tranceivers and storage elements controlled by four PAL components. In the interface .besides, is implemented an ARBITER and a SYSCLOCK generator and therefore it may be configured as a system controller and it can work autonomously in the VME crate.

A functional block diagram is shown in fig.4.

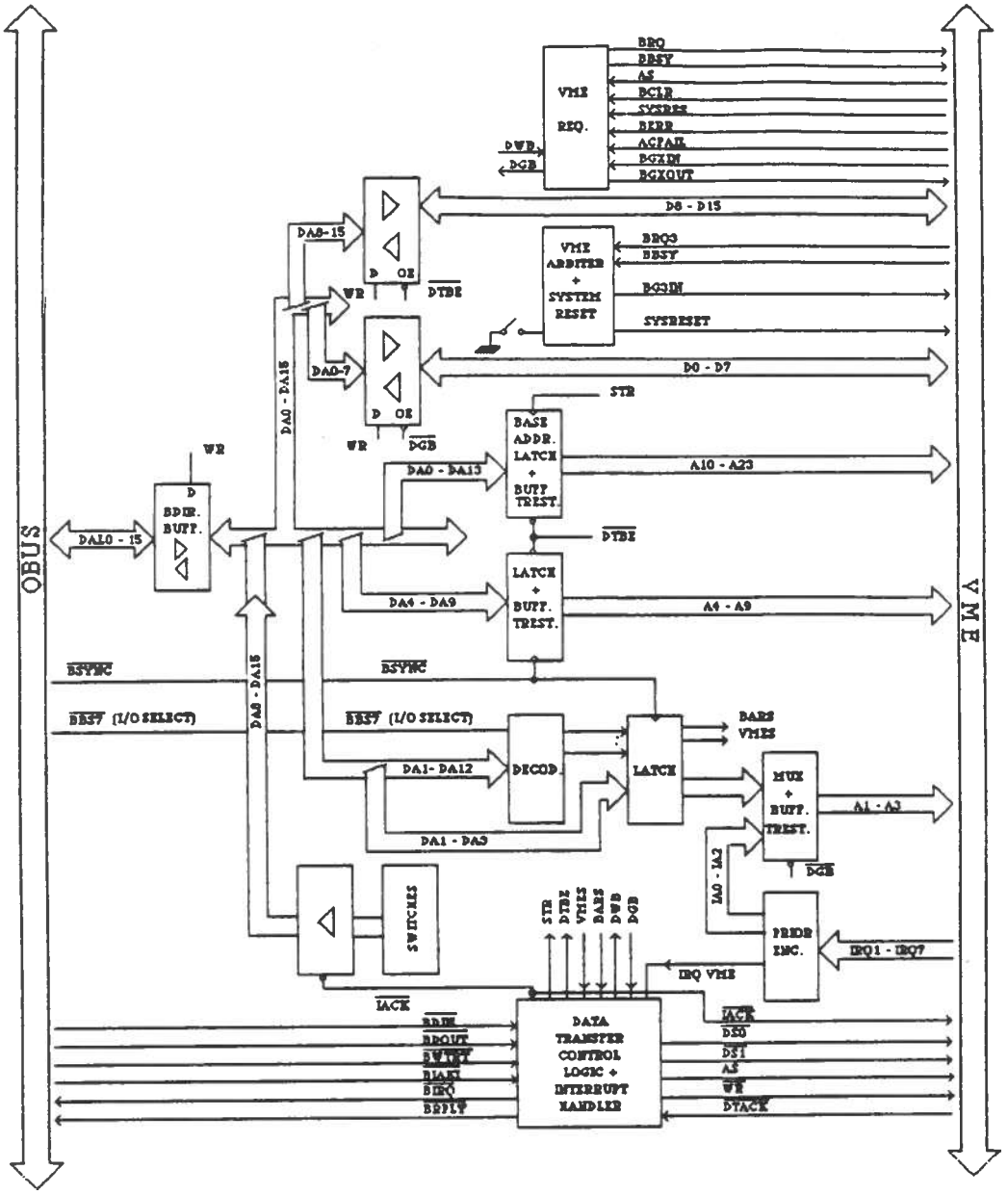

Fig.4 Functional interface block diagram

4.-FUNCTIONAL DESCRIPTION OF THE INTERFACE.

The Q22bus signals from the microvax are buffered and carried to the VME interface by means of two flat cables.

Since the address and data bus are multiplexed, each bus cycle can be divided in two phases: address transfer and data transfer. For this reason the interface must latch the address bits and holding this condition during all the bus cycle.

During the addressing phase, the bus master outputs the address for the desired slave device (BDAL<21:00> lines), asserts BBS7 signal to indicate that the address in the I/O page and after that it asserts BSYNC signal to allow the slave to latch into its internal logic BDAL and BBS7, which are active only during the addressing cycle.

For each cycle the BDAL1-BDAL12 Qbus address lines are decoded and, if the VME is selected, during the Qbus data transfer the interface asks control of the VMEbus, via its on-board REQUESTER, asserting the DWB (Device Wants Bus) signal. When DGB (Device Granted Bus) signal is received the bus is available and the interface executes a VME cycle asserting WR, DS0, DS1, AS control signals; the VME WR signal, that indicates the transfer direction, is related to the Qbus BDIN and BDOUT signals.

The addressed VME slave module responds with the acknowledge DTACK signal that is sent by the interface to the Qbus as BRPLY signal that is the slave's acknowledgement. This signal completes the cycle.

The timing diagram of a write cycle is shown in fig.5.

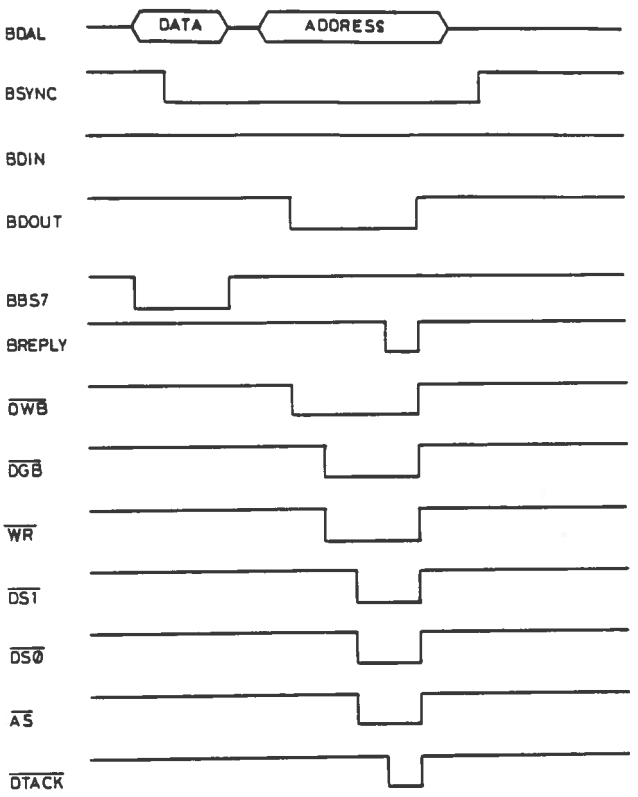

Fig. 5 Timing diagram of a write cycle

An interrupt handler is present on the interface to manage interrupts coming from the VME. When a VME interrupt IRQ1 - IRQ7 is detected by the interface, an interrupt is sent to the Qbus. When the microvax is ready to service this request generates the acknowledge to the interface which sends a signal to its on-board VME REQUESTER, indicating that it needs the VMEbus. When the bus is available starts the VME interrupt cycle.

The 16bit interrupt vector address sent to the Qbus is composed with the 8bit vector coming from the VME and others 8bit set with switches, as follows:

```
D15--------------D8 D7--------------D0
      switches          VME vector
```

The microvax executes the service routine using this vector address.

On the interface there are seven switches to select:

-BDAL12. DBAL11 address bits to choose the Qbus I/O space occupied by the interface

-interrupt 4 or 5 priority level to the Qbus

-ARBITER enabled or disabled: when configured as the system controller, the interface accepts bus requests on level 3 and issues the bus grant

-SYSCLOCK and SYSRESET system clock and reset enabled or disabled; as system controller the interface provides a 16 MHz clock signal to other VMEbus modules and upon reset can drives the SYSRESET line.

In fig.6 is shown the layout of the VME printed circuit board.

4.-COMMUNICATION SOFTWARE.

As we have seen the address lines are decoded to produce two selection signals: one of these selects the Base Address Register and the other selects a location in the VME window.

To perform a data transfer, before all the base VME address (A10-A23) must be written into the BAR register. In the next cycle the interface must acquire control of the VMEbus via its on-board REQUESTER and when the bus is available 1kbytes of VME contiguous address can be directly accessed by the microVAX without changing the BAR address.

VME address are obtained as follows:

```
A23----------------A10 A9----------------A1
      from BAR            A9--A1 Qbus
```

To move the 1kbyte window, in order to address all the VME address space, one must rewrite the BAR register.

A device driver module has been written for the interface. It consists of the standard components of a device driver following the VMS specifications. It is composed by:

--an initialization routine that reads the device for operation when the driver is loaded

--a driver-prologue table that describes the driver and the device type

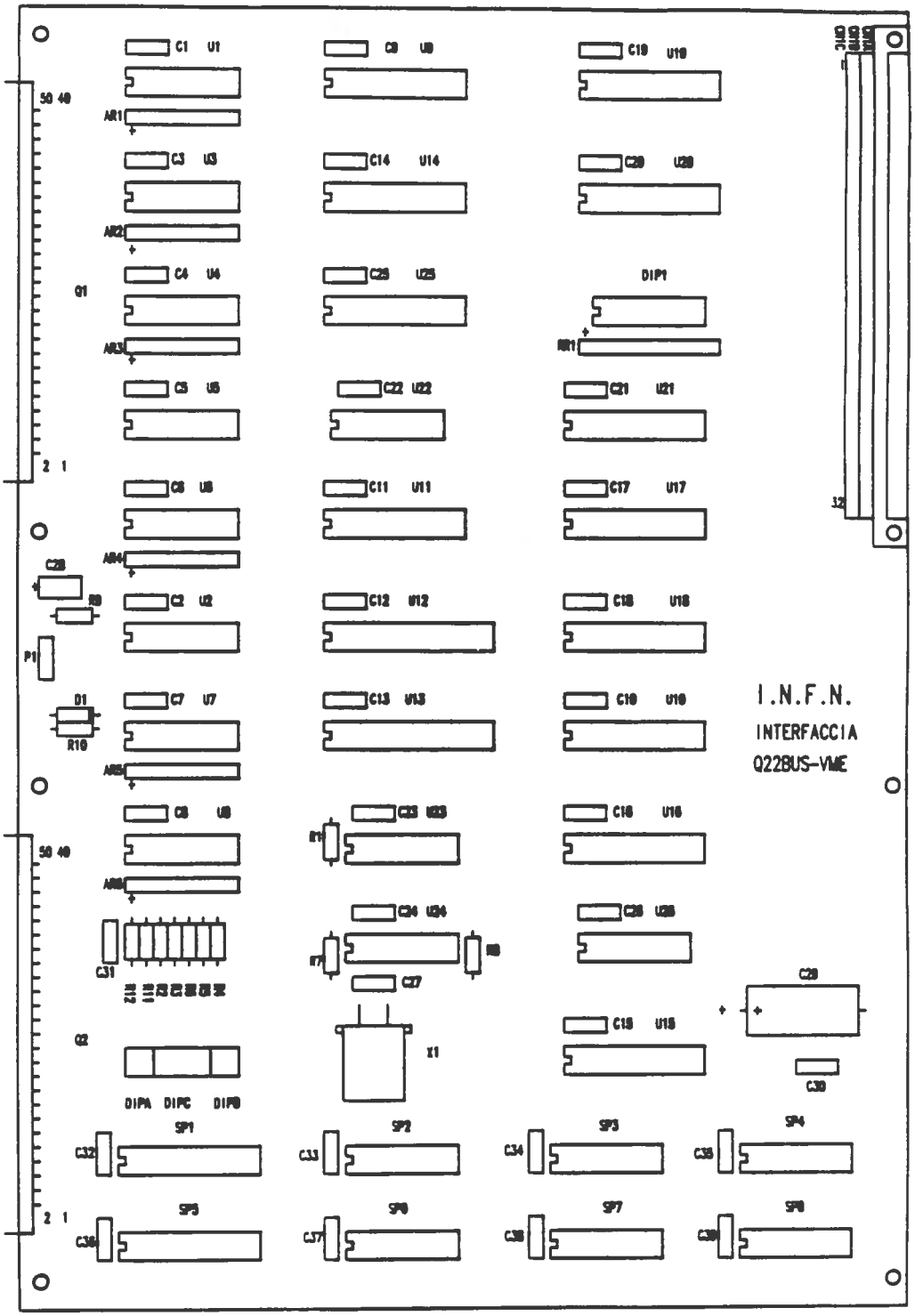

Fig. 6 VME printed circuit layout

--a driver-dispatch table that lists the entry point addresses of standard driver routines

--a function-decision table that lists all valid function-codes for the device (in particular has been written routines to read word, write word, read block of words, write block of words, write BAR register).

The driver is loaded during computer bootstrap with the standard LOAD and CONNECT commands of SYSGEN VMS utility and it is called by a user program that can be written in FORTRAN.

The user program sends, before all, a SYS$QIO with a special function to set the interface and the VME window. After that, read and write operations are performed with standard SYS$QIO calls where are specified the function code, the starting address inside the VME window, the number of words to transfer and the user buffer address.

5.-PERFORMANCES.

A test program has been written to check the system performances.

To transfer data blocks a buffer memory was created into the microvax in order to transfer 1Kbytes of VME contiguous data sequentially without changing the BAR address.

The BAR register write operation requires about 750 microsec; after that the VME access is performed with a period of 5 microsec. Then the time spent in the transfer (tI/O) related to the quantity of words (qw) is represented by the equation

$$tI/O = 750 * k + 5 * qw \qquad \text{(in microsec)}$$

$$\text{with } k = INT (qw / 512) + 1.$$

This function is plotted in fig.7.

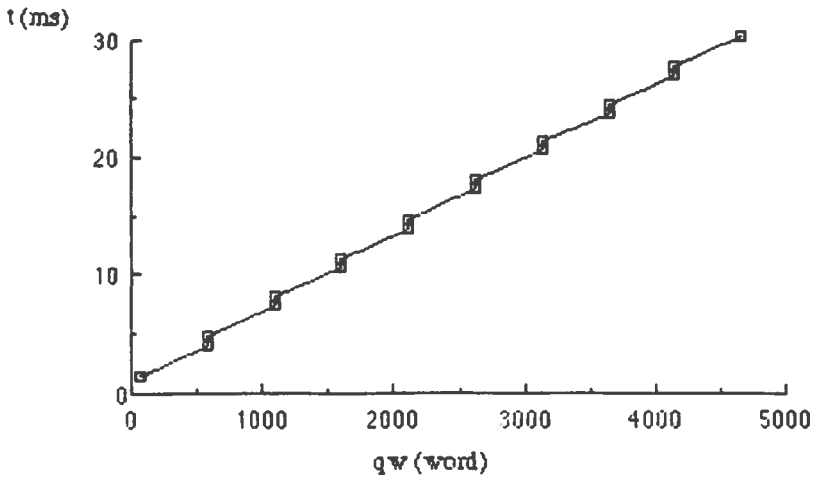

Fig.7
I/O mapped transfer time diagram

6.-CONCLUSION.

We have realized an efficent connection between a microVAX II and a VME crate using a Qbus I/O mapped adapter.

We have obtained a reliable block transfer at an effective rate of about 0.3 Mbytes/sec: to transfer small data blocks we have found more convenient the I/O mapping method because there is no initialization that instead weighs heavily in the DMA transfers.

Moreover the I/O mapped interface may be used as a master in the VME crate and then it is useful in small stand-alone system to control VME strumentation using only a microvax and the interface.

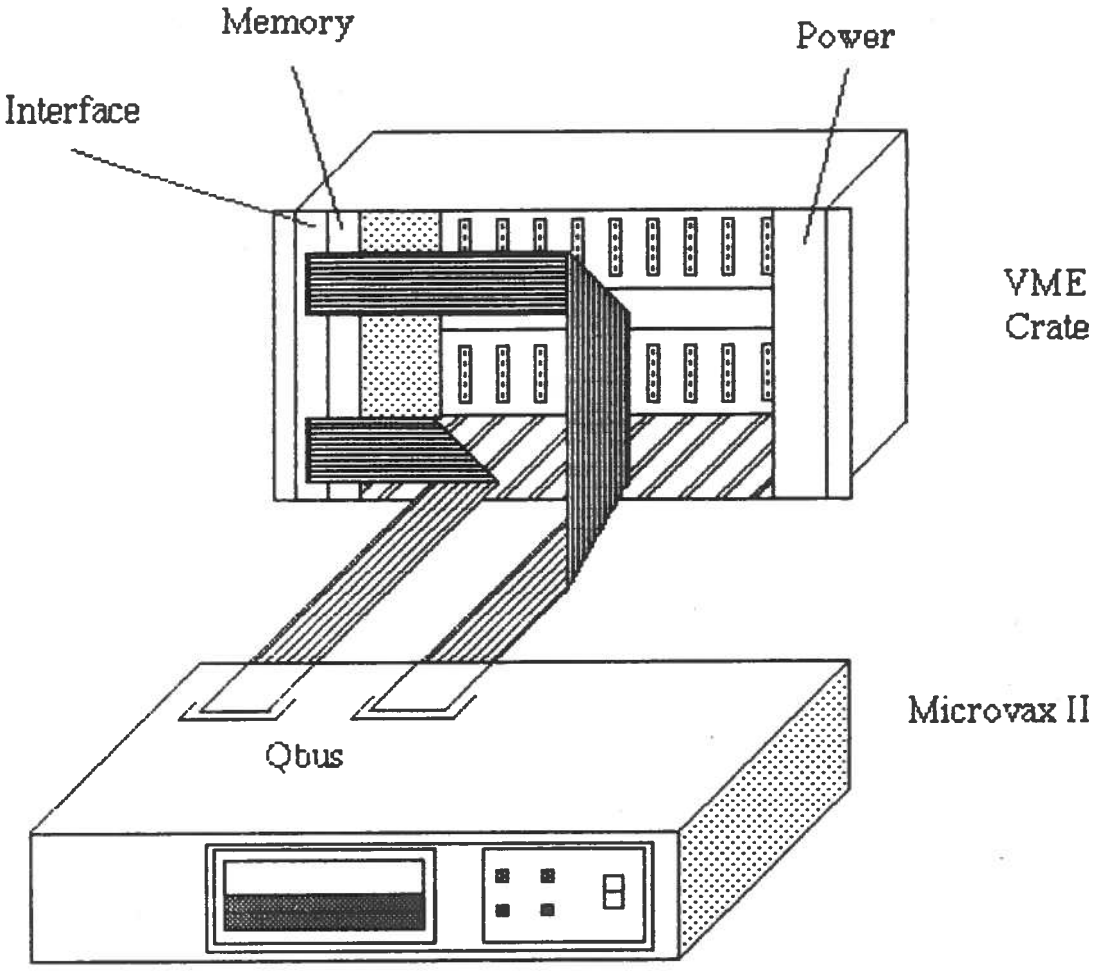

Fig. 1

Schematic view of the I/O mapped interface-microvax connection

		Q2 & Q3	Q1 & Q4
pin	1:	GND	GND
	3:		BSS7
	5:		BWTBT
	7:		BDAL16
	9:		BDAL14
	11:		BDAL13
	13:		BDAL12
	15:		BDAL11
	17:		BDAL10
	19:		BDAL9
	21:		BDAL8
	23:		BDAL7
	25:		BDAL6
	27:	BIACKO	BDAL5
	29:	BIRQ4	BDAL4
	31:		BDAL3
	33:	BSYNC	BDAL2
	35:		BDAL21
	37:	BDIN	BDAL20
	39:		BDAL19
	41:	BRPLY	BDAL18
	43:		BDAL1
	45:	BDOUT	BDAL0
	47:	BIRQ6	BDAL17
	49:	BIRQ5	BDAL16

Fig. 2
Q22 bus lines used

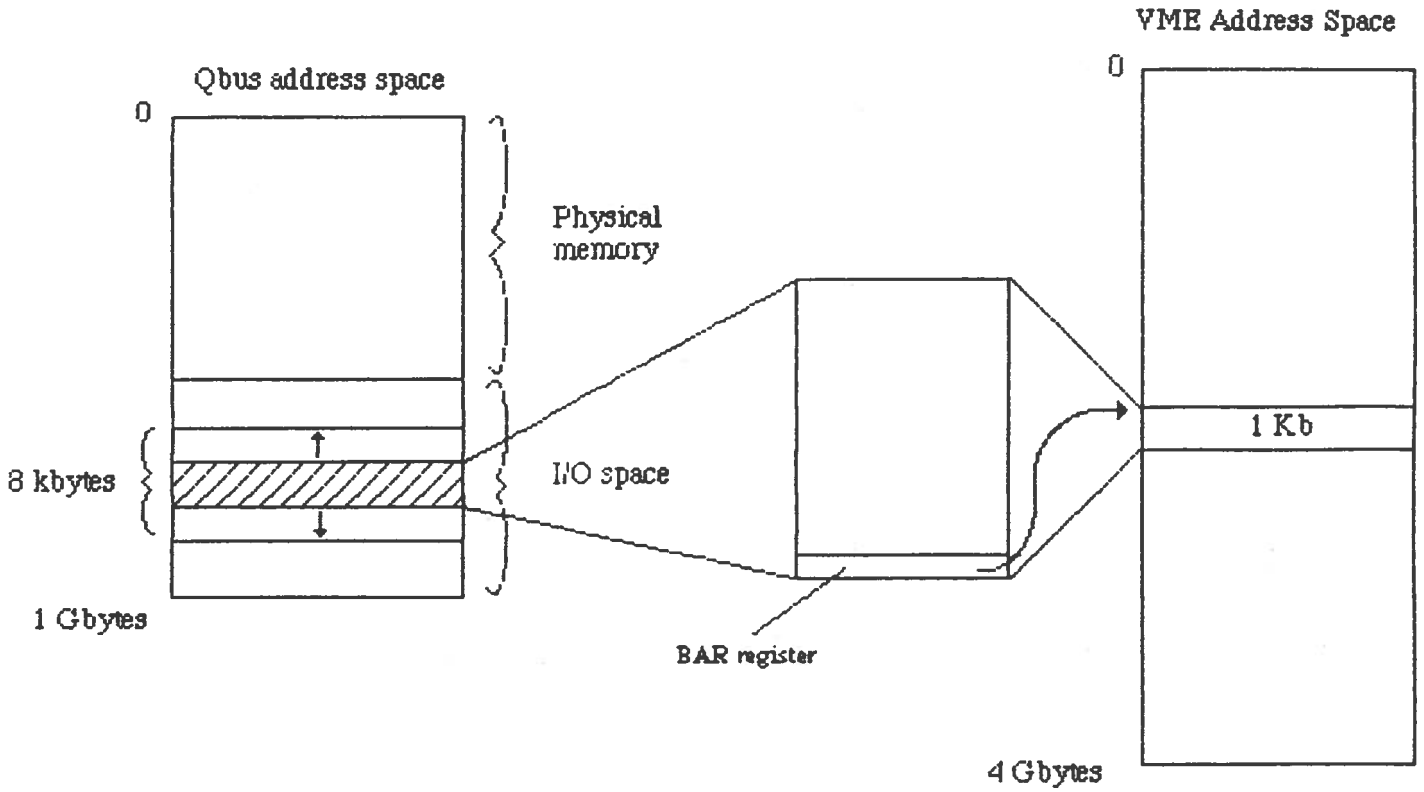

Fig. 3

VME and QBUS Address spaces relation

		Q2 & Q3	Q1 & Q4
pin	1:	GND	GND
	3:		BSS7
	5:		BWTBT
	7:		BDAL16
	9:		BDAL14
	11:		BDAL13
	13:		BDAL12
	15:		BDAL11
	17:		BDAL10
	19:		BDAL9
	21:		BDAL8
	23:		BDAL7
	25:		BDAL6
	27:	BIACKO	BDAL5
	29:	BIRQ4	BDAL4
	31:		BDAL3
	33:	BSYNC	BDAL2
	35:		BDAL21
	37:	BDIN	BDAL20
	39:		BDAL19
	41:	BRPLY	BDAL18
	43:		BDAL1
	45:	BDOUT	BDAL0
	47:	BIRQ6	BDAL17
	49:	BIRQ5	BDAL16

Fig. 2
Q22 bus lines used

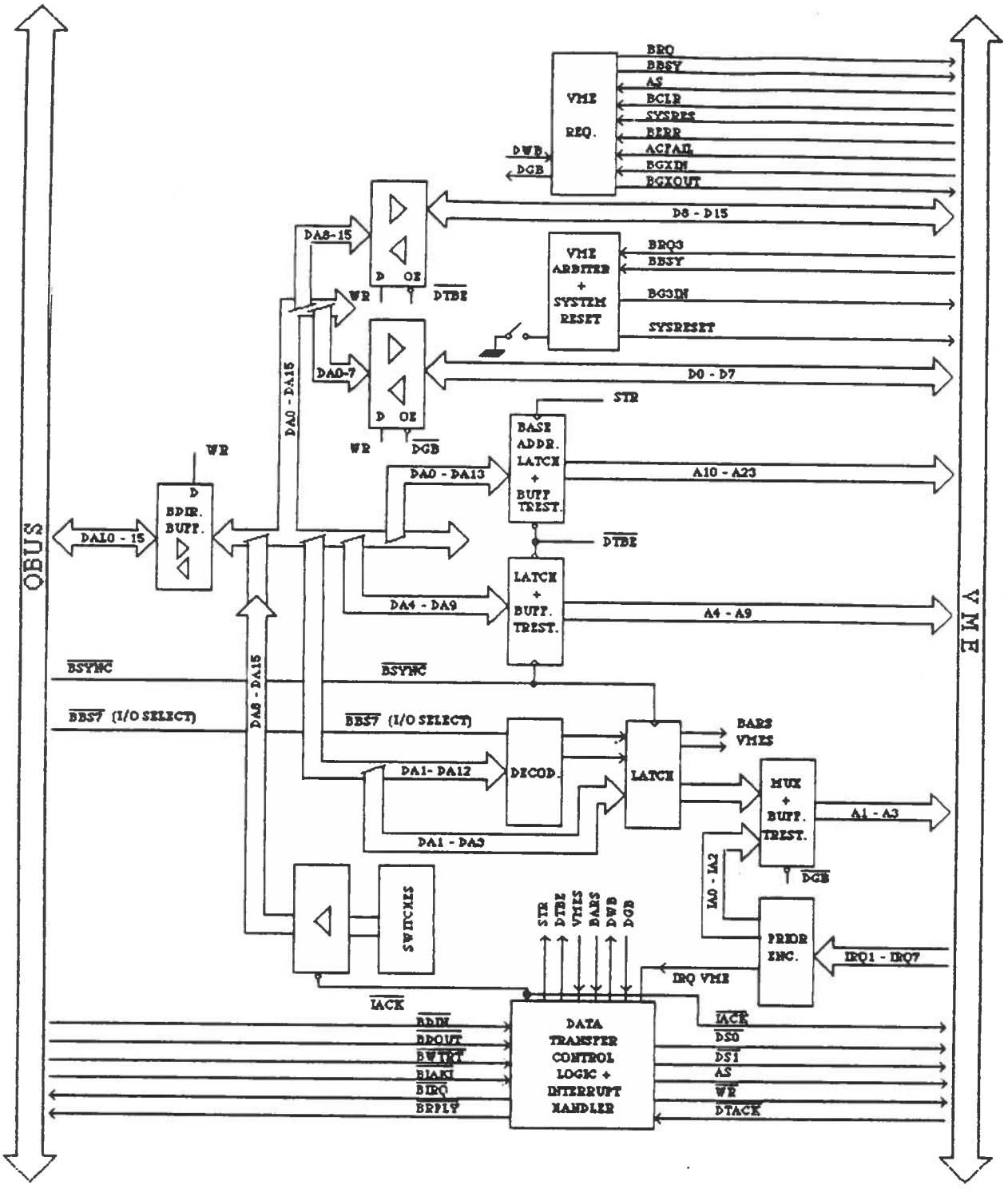

Fig.4 Functional interface block diagram

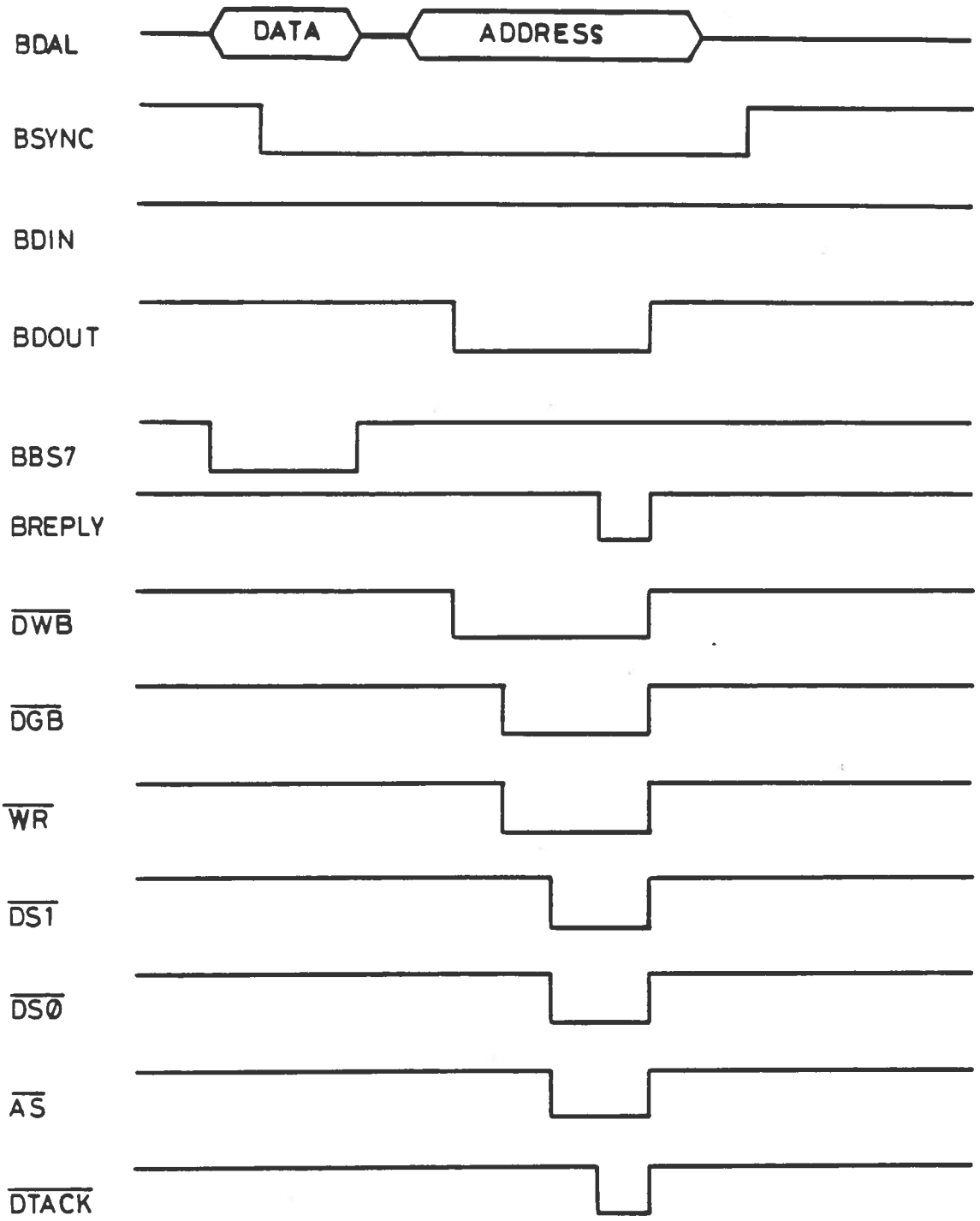

Fig. 5 Timing diagram of a write cycle

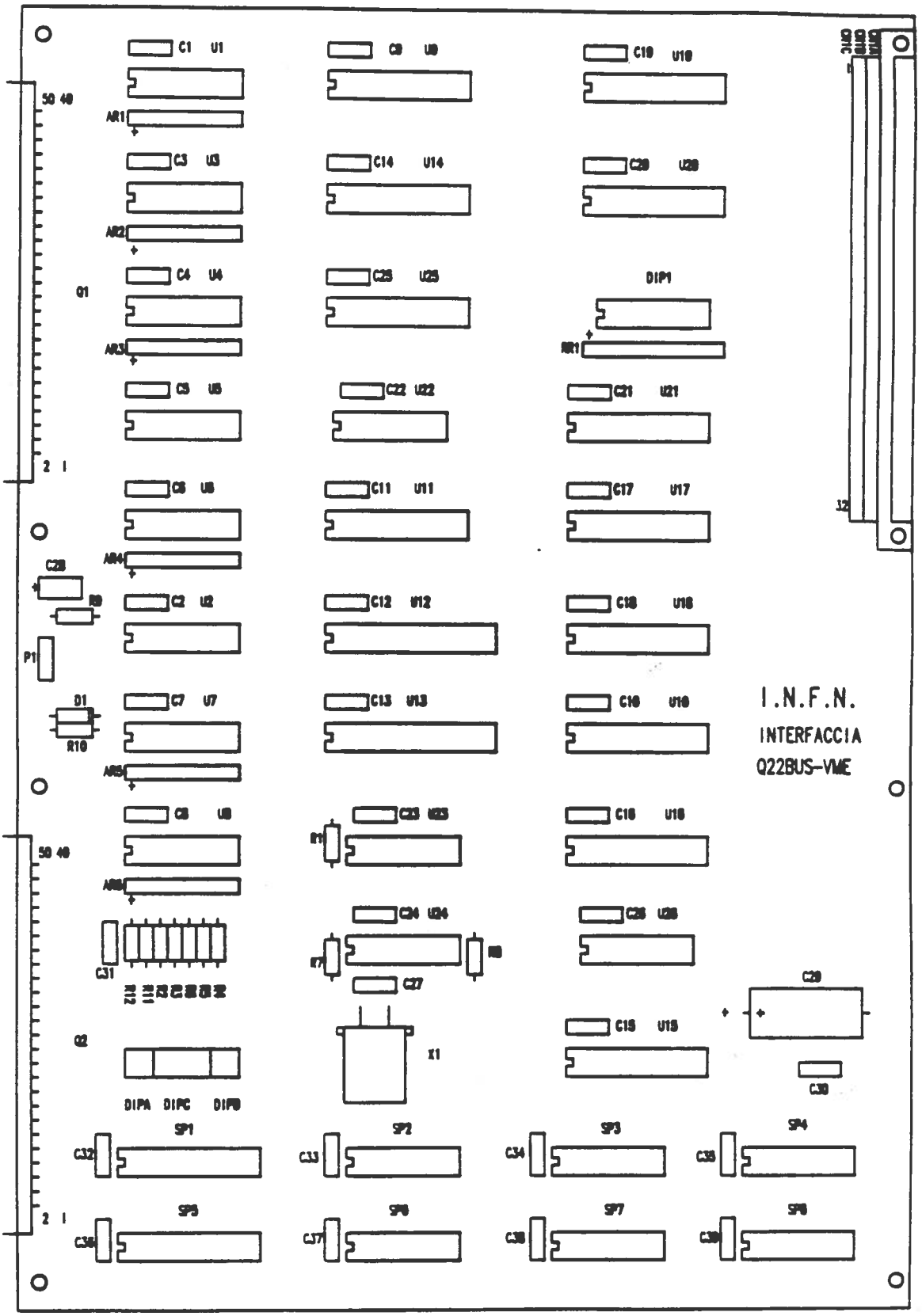

Fig. 6 VME printed circuit layout

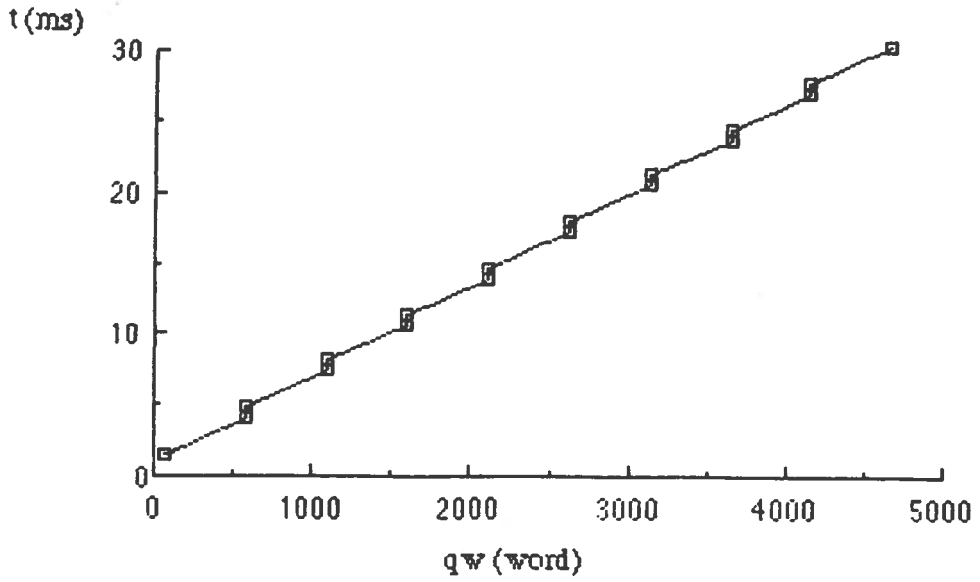

Fig. 7
I/O mapped transfer time diagram