DAΦNE TECHNICAL NOTE

INFN - LNF, Accelerator Division

Frascati, June 13, 1994

Note: **C-13**

The DEVIL as a Real Time processor

L. Trasatti

The DAΦNE control system, DANTE, uses as peripheral VME CPUs the DEVILs, which are built using a standard Macintosh LC III logic board with a custom developed interface to VME, VSB, Ethernet and 4 additional MBytes of RAM.

The DEVILs take advantage of the ease of interaction provided by Apple System 7 and of the LabVIEW programming environment. The result is a VME CPU which makes it particularly simple to program and maintain the control system software.

However, in some time critical tasks, like machine timing, it is important to get rid of the System 7 facilities, which could make the real time response of the DEVIL unpredictable.

To accomplish this goal we have built a set of LabVIEW CINs (Code Interface Node) in 68030 Assembler Language which inhibits all system interrupts, leaving the entire system under program control.

The system is composed of two VIs:

- DisableINT, which disables all system interrupt levels up to the 7th (general reset), which is only accessible through a front panel pushbutton.

- EnableINT, which restores the normal operating conditions.

To test the system, we have built the following setup (see Fig. 1).

A pulser sends a train of pulses (20 Hz) to a digital I/O board (Burr Brown MPV930) installed in a VME crate together with a DEVIL that communicates with the Apple Network through Ethernet, a memory board, a Lextel LL2000 communication board and a VME Bus Analyzer.

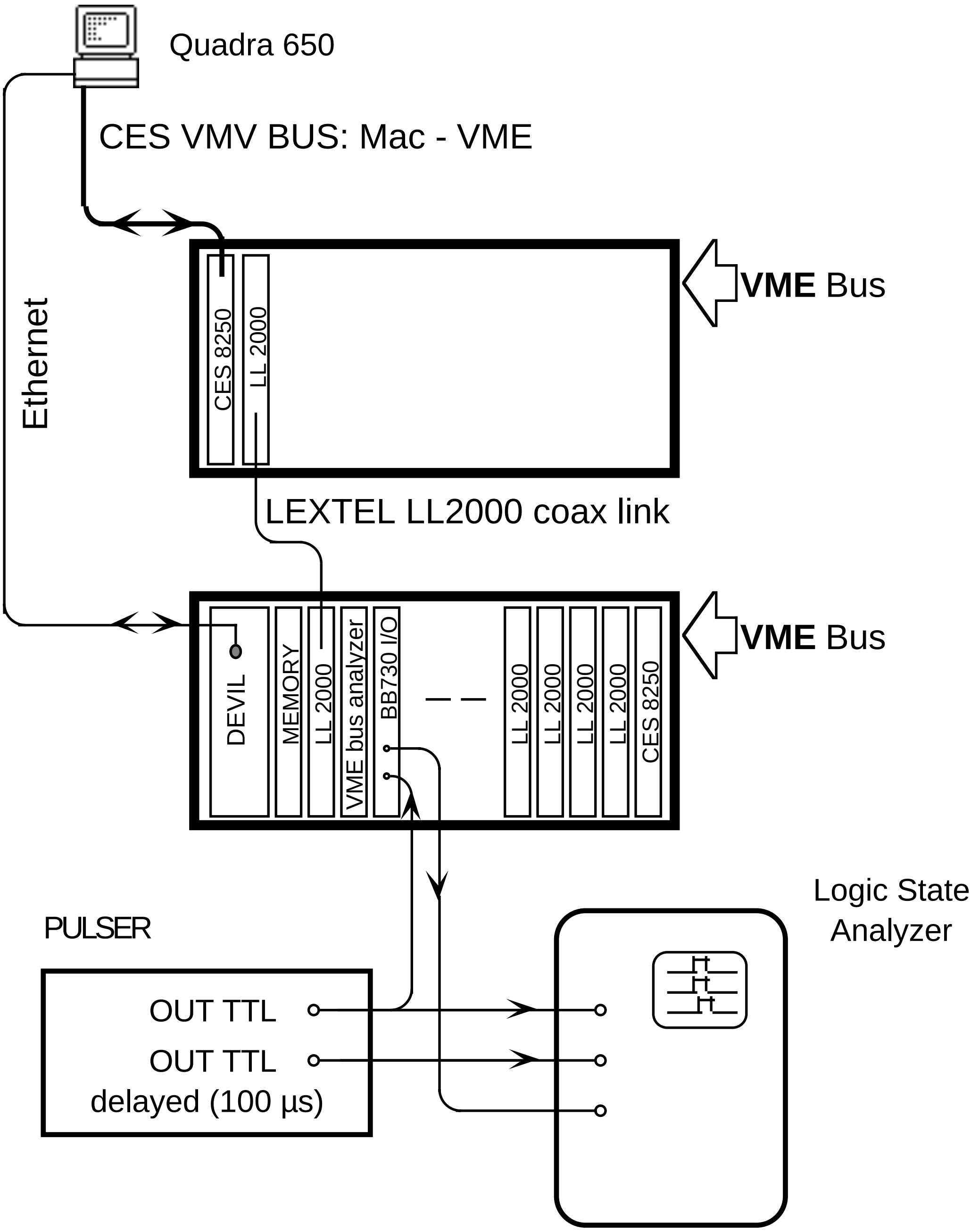

Fig. 1: Experimental setup

The DEVIL, using a test CIN written for this purpose, reads the status of the pulser continuously and copies its value to one of the output bits of the Burr Brown MPV930 board. This process takes about 10 µs.

The two signals are sent to a logic state analyzer together with a second signal from the pulser which copies the first with a delay of 100 µs.

The resulting situation is shown in Fig. 2, where the Logic State Analyzer is triggered on the coincidence of the two signals coming from the pulser (normal and delayed).

At this point the trigger of the analyzer is changed to include the negation of the signal generated by the DEVIL. In this condition the analyzer is triggered only when the DEVIL is stopped for more than 100 µs in the execution of its assembler program by an external request of the operating system, typically an interrupt.

It is very easy to generate such events artificially: Fig 3 shows a typical event generated by the movement of the mouse, while Fig. 4 shows on a different scale (25 ms/div) a typical event generated by a request of the network (password).

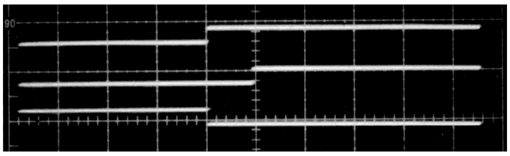

*Fig 2: Normal situation: 100 µs/div, trigger on a*b*

a): Upper trace: Output from the pulser
b): Middle trace: Delayed output from the pulser
c): Lower trace: Output from the DEVIL, copying and negating the upper trace

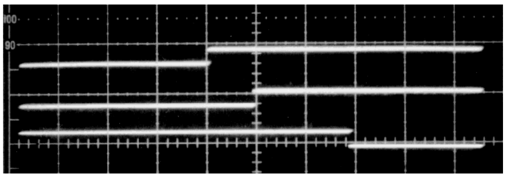

*Fig 3: Mouse event: 100 µs/div, trigger on a*b*c*

a): Upper trace: Output from the pulser
b): Middle trace: Delayed output from the pulser
c): Lower trace: Output from the DEVIL, copying and negating the upper trace with a dead time of 300 µs

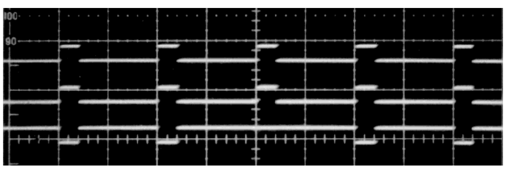

*Fig 4: Network event: 25 ms/div, trigger on a*b*c*

a): Upper trace: Output from the pulser
b): Middle trace: Delayed output from the pulser
c): Lower trace: Output from the DEVIL, copying and negating the upper trace with a dead time > 10 ms

The program we are using for this test does not change the Status Register of the CPU.

If we run a different program, that is identical to the former but masks all of the interrupts at the beginning and restores them at the end, the following situation is achieved:

- the mouse is no longer active.
- the DEVIL disappears from the network
- the Macintosh clock is stopped
- no trigger is generated on the analyzer after a 2 h wait.

An additional feature of the test programs is a continuous test of an external VME location contained in a memory module. The programs keep running until the contents of this location are equal to zero, and stop as soon as the value is changed. It is possible to change the contents of the memory with an access to VME from a Quadra through the LEXTEL LL2000.

This feature proves the possibility of controlling a program which disables the interrupts through a shared memory semaphore.

The program cycle as seen by a VME Bus Analyzer is shown in Table 1.

Table 1: The program cycle as seen from the VME bus Analyzer

	Time	BR3:0*	BgL	AM	Address	Data	+Size	Cycl	Stat	IRQ7:1*
1	0.87us	....	----	0D	00800000	00000000	LONG	RD	OK	....
2	1.03us	....	----	2D	00FFF008	....FFFF	WORD	RD	OK	....
3	0.97us	....	----	2D	00FFF006	....FF00	WORD	WRI	OK	....

As a conclusion it appears possible to use a DEVIL getting rid completely of the influence of the operating system of the Macintosh.