DAΦNE TECHNICAL NOTE
INFN - LNF, Accelerator Division

Frascati, June 10, 1992
Note: **C-6**

The DANTE Control System

G. Di Pirro, C. Milardi, A. Stecchi, L. Trasatti

and

Barbara Caccia, Vittorio Dante, Stefano Valentini
Istituto Superiore di Sanità and INFN, Sezione Sanità, Roma

DANTE (DAΦNE New Tools Environment) is a control system designed for the DAΦNE Φ-Factory.

The design goals of the system have been software simplicity and reliability through the use of standard programs and general structure streamlining through the use of point to point fast links and of a polling configuration.

Standard commercial components have been used both for hardware and software wherever possible, to take advantage of the huge amount of new development tools which are available in the field of computing, data acquisition and controls.

System Structure

An extensive multiprocessor environment has been adopted [1,2]. A large number of peripheral processors has been foreseen for the system (70 at the last estimate). This allows to achieve a very high bandwidth, which is necessary to gain fast feedback to the machine. Moreover, it is possible to customize different processors for different tasks, placing computing power where it is really needed. If every task could be performed by a different processor, both debugging and operation would be ideally simple. While this is an extreme situation, the general idea of minimizing complexity by increasing the number of processors is worth pursuing for reliability and performance.

We shall use interrupt structures as little as possible (we hope not at all). Polling mechanisms are used instead to make the system simple and easy to maintain.

The processors are not connected by a network, but through point to point high speed links. Communications are coordinated by a central controller, generating a system with distributed intelligence, but centralized control. Such a system is more reliable in an environment where the accelerator's control is handled by a central control room.

DAΦNE Implementation

Fig. 1 shows a list of the I/O control channels for DAΦNE. A channel here is defined generally as an input/output wire, carrying an ADC level, a DAC level, a digital bit, an RS/232 link, etc. Assigning 100 channels to the average CPU gives an estimate of 70 CPUs and 70 instrumentation crates.

# of channels	TEST B	LTA	ACC	ATR	RING S	TOT
MAGNETS	77	456	367	407	2769	**4076**
RF	**0**	0	32	0	320	**352**
VACUUM	10	29	45	51	421	**556**
DIAGNOSTIC	119	303	242	371	866	**1901**
FEEDBACK	0	0	31	0	50	**81**
TOT	**206**	**788**	**717**	**829**	**4426**	**6966**

Fig. 1: DANTE channel list

Fig. 2 shows the general scheme of the system. Three levels are defined:

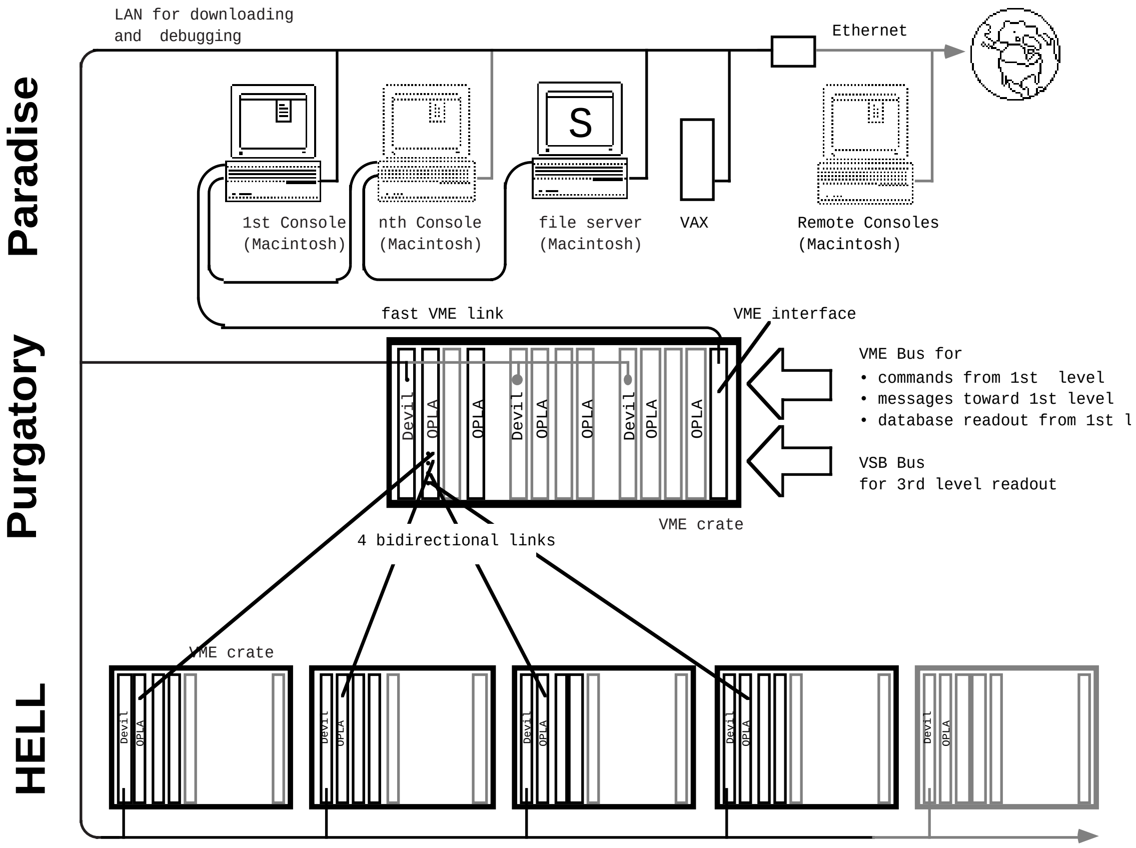

Fig. 2: DANTE schematic diagram

1) PARADISE (PARAllel DISplay Environment) This level implements the operator interface with several equivalent consoles.

2) PURGATORY (Prototype Unit for Recording and Gate TOward the Real time world Yonder). This level implements the communication controller and the database prototype. It is implemented by one central VME crate.

3) HELL (Hardware Environment Low Level) It is the hardware interface (many VME crates).

The first level is the user interface, implemented using Macintosh personal computers. The consoles are all equivalent, so that every one of them can control the whole system and none of them has any privileged functions on specific machine subsystems.

The second level, which consists of a single VME crate, is the communication supervisor. A single CPU receives commands or series of commands from the consoles, interprets and transfers them to the appropriate third level units for execution (e.g.: emittance measurement, beam steering etc.). A log of commands sent is kept in the central memory.

Several other CPUs receive through optical links and the VSB bus the messages sent by the HELL subsystems, which relate all changes in machine parameters. All of the data describing the machine status are immediately transferred to a RAM memory located in the supervisory crate, but accessible to the consoles through Direct Memory Access to the VME bus. This memory constitutes the database prototype, containing at every moment the complete machine status. This double bus method (VSB-VME) allows a very high and flexible bandwidth for data from HELL, since the system is strongly asymmetric in the data flow size: only commands go from Paradise to Hell, while big buffers can be sent by instruments in Hell (waveform digitizers, oscilloscopes, etc.).

The third and last level is made up of at least one CPU per crate, performing on-line control and/or monitoring functions on the accelerator hardware equipment. Hell CPUs must carry out three separate tasks:

i) reading and executing all commands coming from the consoles through Purgatory;

ii) relaying to the upper levels only the relevant changes in the hardware status (information filtering) and the warning and fault messages;

iii) continuously controlling the hardware.

SOFTWARE

Software is by far the biggest problem that a control system has to face. The cost of software development for a control installation is nowadays much higher than the cost of the hardware. At the same time, the reliability of software is a problem very little understood and very difficult to handle. The only available solution is to develop as little as possible in house, taking advantage of the high intrinsic reliability of commercial products.

Under this respect, it is worth mentioning that the reliability of a software component is directly proportional to its diffusion, since the only known way to debug a program is to have a large number of test benches. Incidentally this also means that the reliability is inversely proportional to the price.

It is now apparent that software for personal computers is much better than software for big and expensive machines, simply because much more effort can be put into its development and debugging.

All this has led us to try several packages for the Macintosh family of computers. Experience with these high level tools (MacUA1, HyperCard[3], Timbuktu[4], LabView[5]) has proven very positive. Development times and code length decrease substantially when compared to traditional languages (FORTRAN, C, BASIC, etc.), and reliability increases accordingly.

We finally decided to use LabView® as the main software tool for DANTE. LabView® is a development environment especially designed for the laboratory. A large set of tools for data acquisition and control is available (from histograms to Fourier transforms), graphics instruments are abundant, a graphic programming language is part of the package and a very good debugging structure is immediately available. Multitasking is also included.

While LabView® is obviously a good candidate for the console programs, it can also be a very efficient tool for the peripheral processors. This has led us to a system where all of the processors, including the peripherals, belong to the Macintosh family, with all of the simplicity of a completely uniform software environment throughout the whole system.

An ETHERNET link connecting all the processors will allow both centralized downloading and very good debugging, using a communication program (Timbuktu®) which allows to take complete control of a networked Macintosh from another. The desktop of any of the 70 peripherals can be shown and controlled by any of the consoles.

HARDWARE

The choice for hardware has fallen on the VME standard, because of its high industrial diffusion. VXI crates can be incorporated in the system for diagnostic purposes. We have tried to use standard commercial components wherever possible, and only in two cases we have developed dedicated hardware: the CPUs and the optical links.

DEVILS

We want to use Macintosh computers for the VME CPUs, DEVILS (Developing and Executing Virtual Instruments at Low level). While it is possible to connect a standard Macintosh through a NuBus card to a VME crate, the arrangement is much more expensive and less compact than having a CPU resident in the crate. Therefore we have designed and prototyped a VME interface card which connects directly to the PDS (Processor Direct Slot) of a Macintosh LCII. Access time to VME has been measured to be 400 nsec for a 32 bit read operation. The choice of the LCII has been forced by mechanical considerations. The final CPUs will be built by industry.

Optical Link

We have implemented through a collaboration with ISS (Istituto Superiore di Sanità, Roma) a simple VME slave card OPLA (OPtical Link Adapter), containing four transmitters and four receivers for an electrical/optical link at 100 Mbit/s. This speed is sufficient to cover any VME bus traffic generated by a CPU. The input of every channel is a single 16 bit buffer, while the output is a 4096 16 bit words FIFO. The board serial input/output is through ECL electrical links. The optical transceivers are kept outside the board for two reasons: mechanical strength and the possibility to use different adapters depending on distance. Short distances can be covered directly by the ECL links.

STATUS

The system is entirely designed. Several discussions with other control system experts from other laboratories have convinced us that the system is vi-able and efficient. DEVIL and OPLA have been prototyped and we are in the course of handling them to industry for production. The online Database design is almost complete and LabView prototypes of PARADISE and HELL software have been implemented for a machine subsystem, the Booster.

Acknowledgments

We want to thank Prof Moreno Coli and Ing. Andrea Bartoli (Istituto di Ingegneria, Università di Roma) for their help in the development of the DEVIL prototype.

We are grateful to the Control System experts from all over the world, and in particular Dr. Steve Magyary from LBL who helped us in the design with criticism and encouragement.

We thank the other membrs of the DAΦNE project for continuing support.

References

[1] G. Di Pirro et al.: A centralized control system for the DAΦNE F-factory, presented at RT91, JÜlich, june 1991, to be published in Nuclear Instruments and Methods.

[2] G. Di Pirro et al.: Centralized multiprocessor control system for the Frascati Storage Rings DAΦNE , presented at ICALEPCS 91, Tsukuba, japan, Nov 1991.

[3] HyperCard v2.1, Apple Computer Inc, © 1987-90.

[4] LabVIEW, National Instruments Corp., © 1990.

[5] Timbuktu, Farallon Computing Inc, © 1987-91.