TASS

Trigger and Acquisition System Simulator

An interactive graphical tool for Daq and trigger design

Visit www.top1.it

D. De Pedis INFN-Roma1

TASS is a simulation program with graphical and interactive interface.

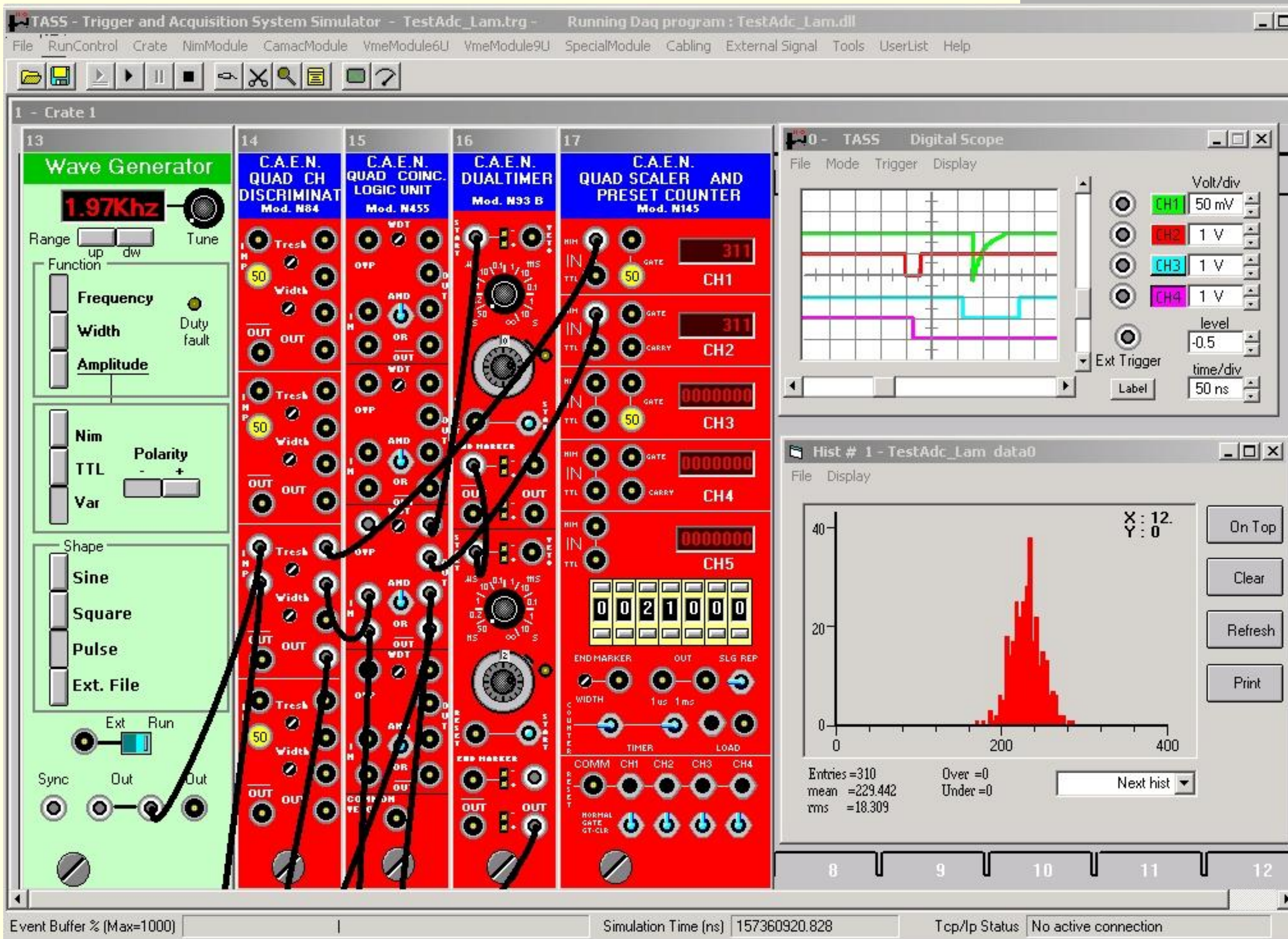
It reproduces in a realistic way the commercial Nim, Camac and Vme modules.

Both the front panel appearance and the electrical and logical behaviour are simulated.

As on real setup, the user can :

- push buttons and move switches
- turn knobs
- set Camac/Vme registers
- ... and so on

Example of a small setup



TASS is devoted :

- To help experimental physicists developing their trigger systems.

- To help the students learning the fundamental aspects of this job.

Indeed ….

… up to now, the design and construction of a trigger system has been based on the experience of its designers and the simulations always based on software programs.

On the other hand, the transfer of know how among the members responsible of trigger systems can become very difficult if based on pencil and paper and thousands of lines program code.

Therefore an interactive graphical tool becomes an essential part of the trigger design process.

The main idea is based on built-in “library of devices”.

TASS provides a picture of virtual electronic counting room. There, user can allocate racks, crates, fan units ... and so on, as in real setup.

TASS provides tools usually found in an experimental area like digital scope, wave generator, voltmeter.

The user builds her/his virtual trigger system choosing from the library the modules she/he needs, places them in the crates, makes the cable connections and runs the simulation.

Any parameter can be set interactively and input signals, provided by waveforms generators, can be used to stimulate the system. The resulting output can be shown on a virtual digital scope and saved on data files.

For each module, the program assumes the mean values of parameters (gain, delay, pulse width etc.) as reported in data sheets of their constructors.

The program doesn’t attempt to analyze the internal circuit of the modules but simply takes in account their transfer functions.

The simulation is “event driven”, an event being a change in the level of a signal.

Each time an event occurs, the program computes the transfer function of all modules whose inputs are affected by that event and reflects the changes on their outputs.

How TASS Works - cont

The advantage of this philosophy is that the user can change and try, very quickly, different configurations for the system under construction, by simply replacing modules and cable connections.

Since the complete hardware behavior is simulated, these actions will reproduce effects exactly corresponding to the ones of the real system.

The simulation can run continually or step by step.

Time is internally stored by 64 bits (19 digit) allowing a total symulation time longer than 10 days with resolution better than ps.

User can set breakpoints to halt the simulation if a predefined event occurs.
There are three type of breakpoint:

- BrkP break and pause simulation
- BrkW break and write to file
- BrkI break and send interrupt to the Daq

Using programmable devices (Camac and Vme) the user has to provide an external Daq program steering the flow of command to the hardware.

A library of Esone routines and general Vme support functions is provided.

TASS supports hardware interaction by

- Look At Me (LAM) interrupt
- Polling techniques

A histogramming library, similar to HBook, is available.

TASS adopts a powerful cabling technique:

- **Nim, Ecl and 3M bundle connection allowed.**
- **Cable’s length (in ns) is automatically computed.**
- **Automatic check of plug and cable consistency.**
- **Capability to label any cables.**
- **Complete list of properties of any connections.**
- **Statistic for cable lenght.**

TASS supplies automatically:

- **The picture of layout of the trigger system (counting room)**
- **The picture of any crate**
- **List of used crates**
- **List of used modules**
- **List of used cables**
- **List of connections**

The package includes a ‘‘Device Editor’’ that may be used to develop, in standard and fast way, code for modules not present in the library. This tool provides an extended set of Electronic Component Control (ECC) like connectors, switches, knobs, etc. simulating the corresponding components usually found on real modules.

User builds up the module placing the ECC on its front panel. The Device Editor, on the basis of insert ECC, directly prepares the corresponding code routines. About 70% of code is written in this way.

Only the specific behavior of modules is responsibility of the user.

In Vers.4.0 a new feature has been introduced allowing TASS to be connected to the network via Tcp/Ip protocol. This capability opens a new class of possible uses of TASS system, that is:

New features

Integration with the real Daq program

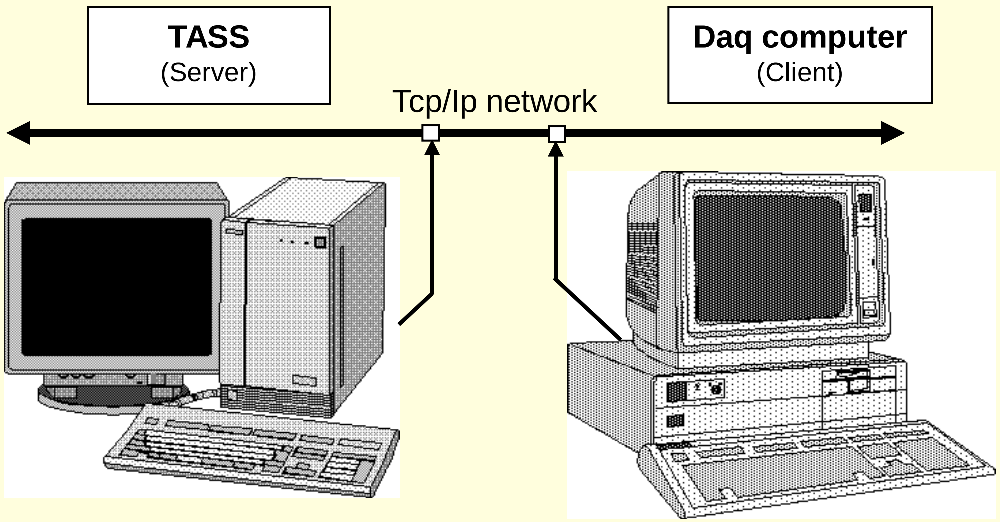
New features - Integration with the real Daq program

Full integration with the real Daq program can be easily achieved by the exchange of Camac / Vme commands via Tcp/Ip communication.

TASS provides suitable RemoteDaq class to interface Daq programs written in C++, Java, Visual Basic, and LabView languages. The supplied RemoteDaq class allows writing the Daq Camac/Vme calls in the same way as in the real world making the translation process, between the external Daq environment and the TASS system, fully transparent from user's point of view.

This feature allows user to develop and test her/his Daq program also in absence of real hardware.

Full integration with the real Daq program

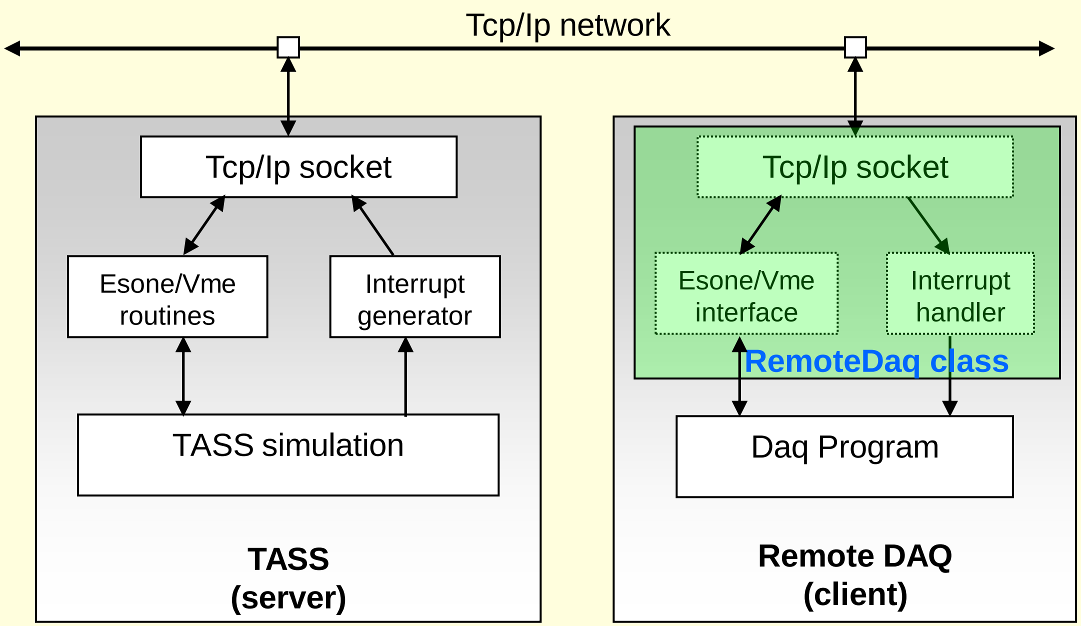


Java example

```
//define Camac register Adc0
Camac.CDREG (Adc0, br, cr, Adc0Stat, SubAdd0);

//define Camac register StatA
Camac.CDREG (StatA, br, cr, StatAStat, SubAdd);

//initialize Camac modules
Camac.CCCZ (Adc0);                          //init crate
Camac.CSSA (17, StatA, Mask, q);     //Write Mask
Camac.CSSA (26, Adc0, Data0, q);     //enable Lam
```



RemoteDaq class provided for:

- Visual Basic
- Java
- C++
- Lab View

New features

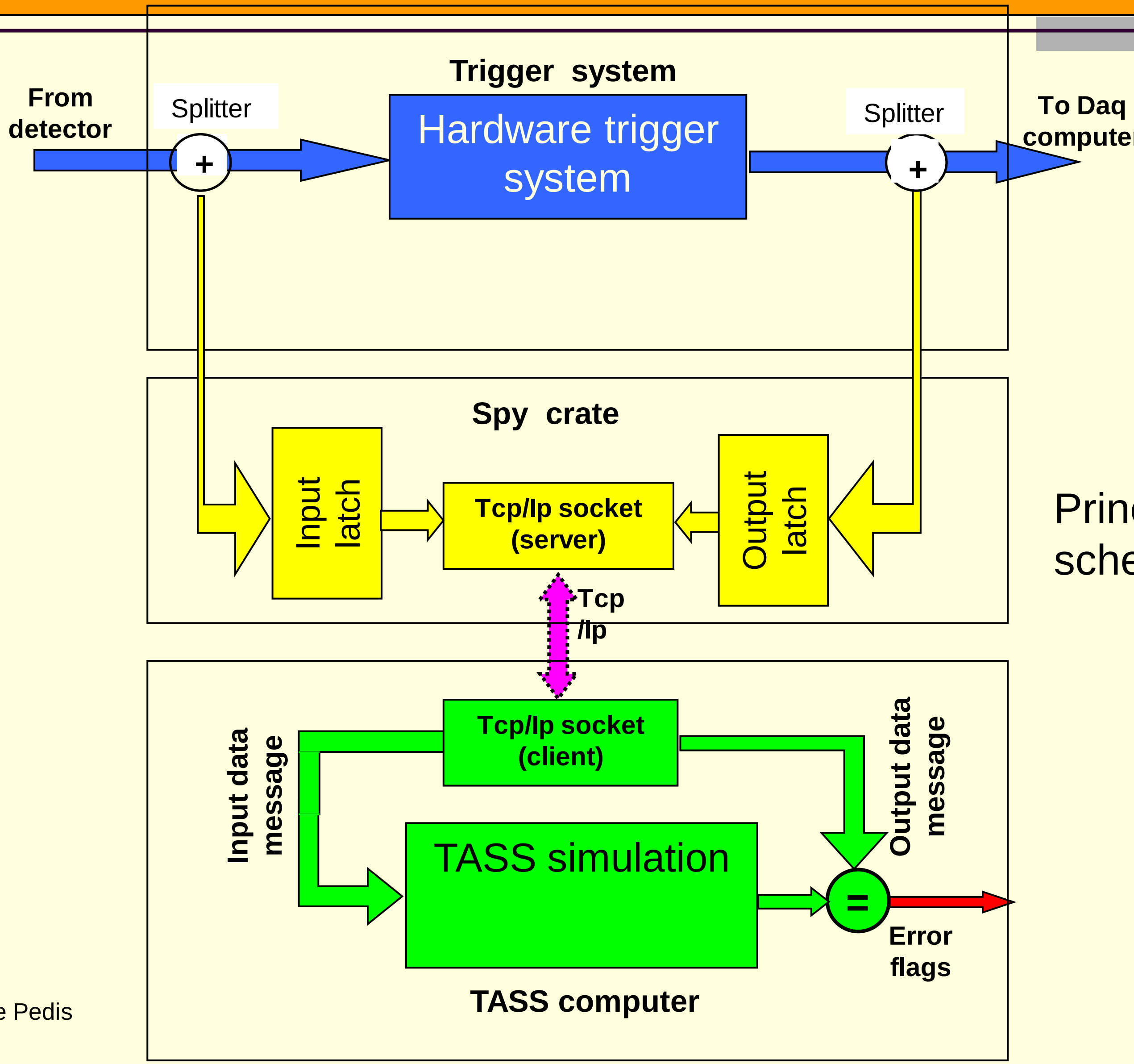
Hardware monitor

TASS can accept input data representing external electrical signals and send output data via a Tcp/Ip network connection. This new feature promotes TASS to become not only a simulator but also a very powerful monitor of hardware trigger systems:

Let TASS simulate the real trigger. Input data (on a sampling basis) can be latched and sent, via Tcp/Ip, to TASS program where they will be treated as in the real hardware. The TASS output can then be compared with the output from the real system and flags can be set according to the comparison, therefore, a different behavior between real systems and the simulated one can be prompt discovered.

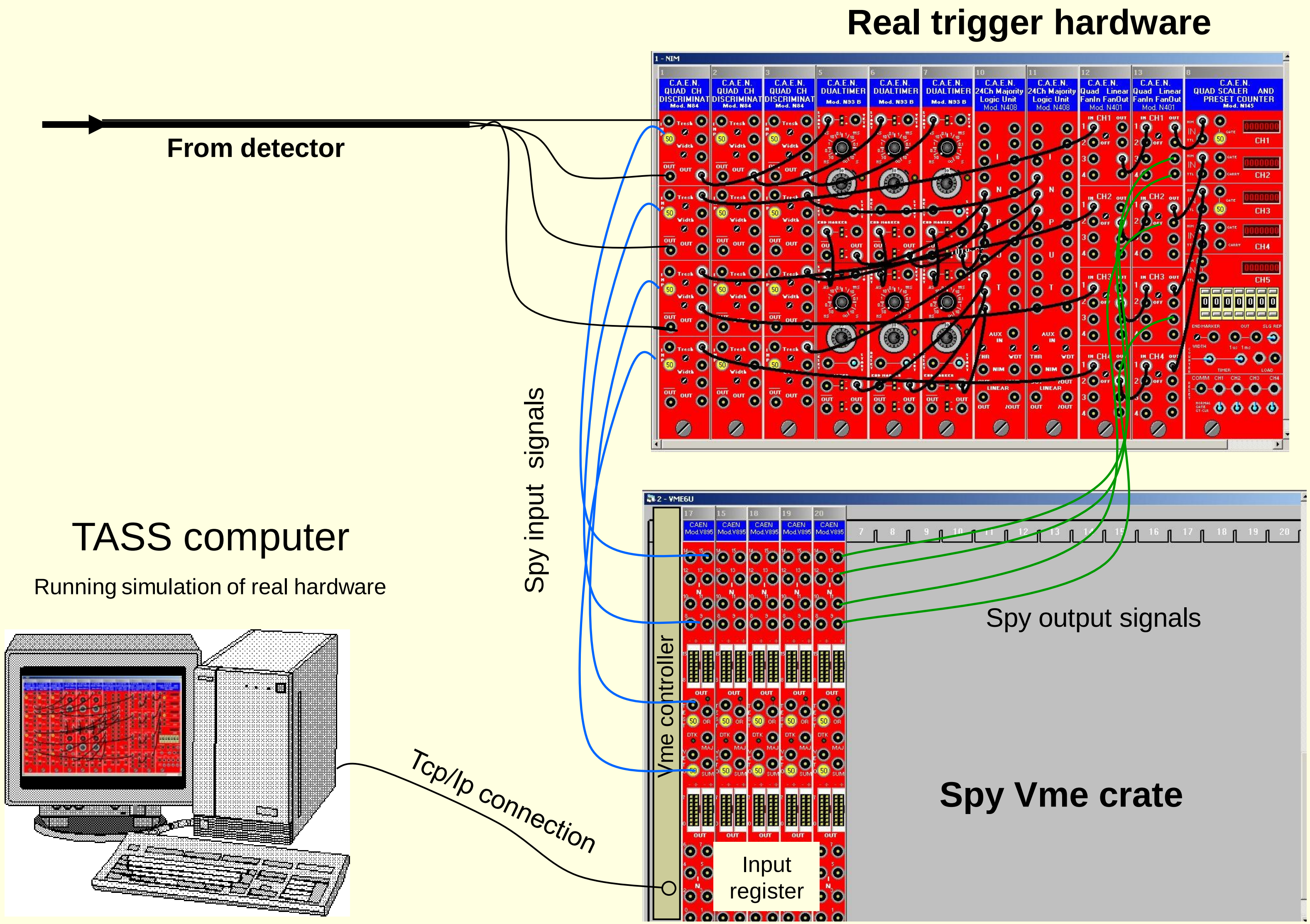
Even more, TASS can help to identify the bad working hardware doing a detailed crosscheck between the real setup and the simulated one.

Hardware monitor – Principle scheme



Principle scheme

Hardware monitor - Possible implementation



- **Design, optimization and documentation of the trigger system in physics experiments.**

  This is the main TASS goal. Several different configurations can be tested without the need to have the real modules available. Also the importance of being able to reproduce the system behavior corresponding to an 'old' configuration should not be underestimated.

- **Development and test of DAQ program.**

  During the development of DAQ program, the real trigger can be replaced by its TASS simulation.
  No real hardware needed to be involved.

- **Monitor and on line debugging of real systems.**

  Making the comparison between the real output and the simulated one, a hardware fault can be prompt discovered.

Possible application

- **Evaluation of hardware dead time and efficiency.**

  TASS keeps trace of delays and duration time of signals of any module, cable connections and so on. Evaluation of hardware dead time and efficiency are built-in in TASS output.

- **Speed up of the tuning phase in beam activity.**

  Most of beam time spent to tune the real system can be saved if any critical part has been simulated in advance. This can be extremely important in case of short beam allocation periods.

- **Student and on line team training.**

  TASS offers to students and instructors the ability to quickly design and test general-purpose systems without any real available modules. Training of online team for trigger system diagnostics should also be considered.

- **Framework environment changes.**

  Changes to the real trigger can be simulated in advance evoiding to modify or dismount the existing set up.

Conclusion

- So far more than 50 modules are simulated.
- Full package consists of more than 500K code lines.
- Presented in several conferences.

We think that TASS may represent a novel approach to the problem of simulation of complex trigger systems in HEP. Its modular structure allows a simple procedure to test different trigger setup. Its interactive behavior will be of great help in understanding problems that sometimes appear only after off line data analysis.

A demo package comprehensive of many examples is available on Web at:

www.top1.it