New trigger implementation for CERES/NA45 at CERN

D. Miśkowiec, GSI

The CERES/NA45 experiment at the SPS was upgraded in 1997-1998 by adding a cylindrical radial drift time projection chamber (TPC) in order to improve the invariant mass resolution. A new data acquisition system was needed to cope with the increased event size. Furthermore, the old trigger became obsolete and had to be replaced. In this note I describe the new trigger, prepared in early 1999 and successfully employed in the 40 GeV/nucleon lead run in fall 1999.

CERES/NA45 triggers on centrality which is accessible via charged particle track multiplicity. First, a minimum bias trigger is generated by requiring signals in two Cherenkov detectors, located in the beam upstream of the target, and requiring an anticoincidence with a similar detector located right after the target. Second, a centrality condition is imposed by requiring a certain number of charge particle tracks in the silicon drift detector. This information is available only 4 μs after the collision. Moreover, the silicon detector is sensitive to the electrons knocked out from the target by passing beam particles. Thus the trigger must make sure that, within several microseconds before and after the collision, there has been no other beam particle passing by (beam before- and after-protection). Similarly, the TPC drift time window is protected by requiring no other collision within $\pm 70\mu$s (reaction before- and after-protection).

The new trigger logic is implemented in one VME and one NIM crate, using standard electronics modules. Two VME logic units V495 are the heart of the system. In the second of them, working in the lookup-table mode, the coincidences between the beam detectors are realized. The first one, plugged directly before the second, is used in the transparence mode (output bits being copies of the input bits) during normal data taking. By switching it to the pattern generation mode one can simulate various sequences of detector signals and test the response of the trigger logic. This is particularly useful for testing the before- and after-protections.

In the old CERES trigger the VME modules were controlled using software written in Real-Time FORTRAN, running on a FIC processor under OS9. In order to make the new software less platform dependent, and to profit from the friendly UNIX environment both during development and usage, we decided to split the software in two parts:

- A simple server, running on a processor in the VME crate, talks to VME modules on one side and to clients on the other.
- A client program running on a linux PC provides a graphical user interface (GUI) for the experiment crew.

The communication between the client and the server goes via a socket. The client requests the server to read (write) data from (to) a given physical address corresponding to a VME module. The server, 250 lines of C code, is so simple that it can easily be adapted to a new operating system. In fact, it was first developed on a PC (linux), then tested at GSI on an E7 (Lynx), and finally installed at CERN on the FIC (OS9). The GUI client translates the user requests into read/write commands which it then sends to the server. The responses of the server are again translated into something which can be understood by the user. The code, 6500 lines in C++, is based on ROOT. There are specific classes for different VME modules and a generic VME-module parent class. Each specific class has a corresponding GUI class, and there is one parent GUI class from which the GUI classes inherit. Finally, there is a class which is specific for the CERES trigger logic, and a corresponding GUI class. A normal user is confronted with an object of this class (see Fig. 1). The upper part allows to set the trigger parameters. The lower part schematically represents the trigger logic. Connections are shown as lines, and the actual state of switches is visualized. The actual counts of signals in the current burst are shown. Twice a second the window is updated using the actual hardware information. The integrated counts of each burst are saved on a file. Another program allows for plotting the counts versus time and thus tracing the history of the beam and data taking over the entire duration of the experiment. The source code is available at http://wwwinfo.cern.ch/ misko/trigger/description.html.

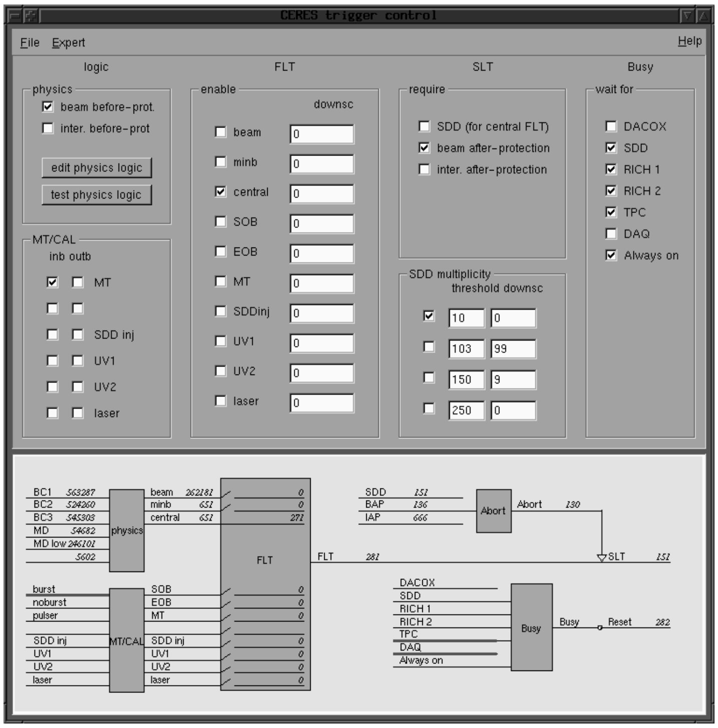

Figure 1: The user interface of the new CERES trigger control software.