CM-P00056677

ORGANIZATION FOR NUCLEAR RESEARCH

CERN-ECP 91-3

809111

CERN/ECP 91-3
12 January 1991

C1

PERFORMANCES OF THE CENTRAL L3 DATA ACQUISITION SYSTEM

T. Angelov³, B. Bertucci⁴, S. Falciano^{1,4}, M. Fukushima³, D. Linnhöfer^{1,2}, B. Martin¹,
G. Medici⁴

¹ CERN, Geneva, Switzerland

² Institut für Hochenergiephysik, ETH Zürich

³ Massachusetts Institute of Technology, Cambridge, Massachusetts, USA

⁴ INFN Sezione di Roma and University of Roma, La Sapienza, Italy

Abstract

In this paper we present the central L3 Data Acquisition system which is entirely built in FASTBUS. Performance and reliability during the first year of data taking at LEP is reported. Special emphasis is placed on the system functionality and the chosen data structure.

1. Introduction

The Data Acquisition system of the L3 experiment [1] is designed to combine the digitized data, delivered by the subdetectors, into one single event with a data structure suitable for online monitoring and physics analysis. The second and third level triggers, which reduce the data rate in the process of event building, are integrated into the system. Accepted events are then transferred through a fast interface to the online computer for storage and monitoring.

The modularity of the detector has allowed the building of a flexible architecture by splitting the system into independent partitions. Parallel data taking in the various branches is then possible and allows to maximize the system throughput. A further splitting into horizontal layers allows the data, which flow asynchronously from the subdetectors, to be resynchronized and formatted in several steps. The partitioning of the system is particularly useful for test and calibration runs. In case of a failure of a single detector component, it can be easily removed from the data stream in order to continue data taking with the rest of subdetectors.

Finally, to avoid dead time, the system provides enough bandwidth and intermediate storage capability.

We have chosen FASTBUS [2] as the standard to implement all the tasks listed above as it is specially suited for building high-speed modular data acquisition, data processing and control systems.

2. General description

A block diagram of the L3 Data Acquisition (DAQ) system can be found in Figure 1. This diagram shows the interaction between the different trigger levels and the DAQ path. The system consists of four parallel data streams from the four major subdetectors (vertex chamber, electromagnetic calorimeter, hadron calorimeter, μ -chambers) and a separate stream of trigger data with much coarser segmentation. In each stream, data flow upon a positive level-1 trigger [1,3] decision and are combined into a single output memory by a subdetector Event Builder. A second step of event building is performed only in case of a positive level-2 trigger [4] decision taken by the level-2 processors (XOP [5]) which analyse the trigger data. A central Event Builder combines all the subdetector data and the trigger data into a single event, which is dispatched to the memory of one of the level-3 trigger processors (IBM 3081/E emulators [6]). The emulators have access to the data with full detector granularity and perform a sophisticated analysis. On a positive result, the event is acquired by the main online computer and stored on tape.

3. Hardware description

At each horizontal layer of the subdetector partitions, repetitive tasks lead to a modular structure which is implemented with a few hardware devices (Figure 2): several identical source memories, a destination memory, a data moving device and an intelligent controller to steer the data flow. The same FASTBUS operations are carried out in parallel and twice in each stream: data are transferred from data sources on a backplane segment to a destination on a cable segment.

To buffer the data coming from the subdetector read-out electronics, we have chosen the memory LeCroy 1892 [7]. Data are fed into the memory via differential ECL-lines at a speed of 16 Mbytes/s and can be simultaneously read out via the FASTBUS backplane. Data assembled from several of these memories are moved to a destination buffer which, in turn, becomes the data source for the next step of event building. The Dual Slave Memory (DSM [8]) was used as the buffer. It can be addressed both on the FASTBUS cable (destination) and the backplane (source) segments.

Both memories allow the marking of record boundaries and have internal counters which keep track of the number of words written or read.

Data moving is performed at very high speed by a dual ported device called Block Mover (BM [9]). The BM reads the data on one segment and writes them on another segment adding only 50 ns to the access time of the slowest memory. It is controlled by a very fast programmable sequencer which can perform versatile FASTBUS operations and formatting of the data. We use as controlling device the General Purpose Master (GPM [10]) based on the Motorola 68000 microprocessor. It has full FASTBUS master capabilities, can be interrupted via FASTBUS and executes FORTRAN code compiled by the CERN cross compiler [11].

The BM and the GPM form an Event Builder at both subdetector and central level.

The initialization of the whole system, which takes about 2 minutes, is done by a software package called FSM (Fastbus System Management [12]). It initializes all the routes which logically connect the different crates, writes initial values to memory locations and registers, downloads the programs into the FASTBUS master devices and verifies the system hardware.

4. Event building procedures

In the first step of event building, where partial events are built in parallel within each sub-detector partition, the GPM first ensures the event synchronization in the source memories by checking a common event number attached to the data. Then it sets up necessary parameters for the BM transfer, such as addresses and word counts, and initiates the transfer. In case of de-synchronization, the GPM tries an automatic recovery and reports to the operator console through its terminal port. The GPM also provides some of the ZEBRA control words described in the next chapter.

In the second step of event building where a complete event is built by assembling partial events from all the sub-detectors, the GPM checks again for event synchronization and then takes into account the decision of the level-2 trigger. If positive, it chooses a free emulator, initiates the BM for the data transfer and starts the execution of the emulator program upon termination of the transfer. On the negative decision, the GPM clears the event in the source memories by simply advancing the event pointer.

The end of the level-3 processing is signaled to the CHI by sending the Service Rquest (SR). The CHI then transfers the entire event to the VAX only in case of a positive decision. Regardless of the level-3 trigger result, a fraction of the events are passed to the VAX to monitor the functioning of the trigger system.

Besides the transfer of the main data, the CHI transmits messages generated asynchronously by various FASTBUS masters to the VAX. Messages and statistics on the event building serve for the detailed monitoring and diagnostics of the FASTBUS system.

5. Data structure

The L3 event format has been chosen to fulfill the needs of the online and offline software which require data to be assembled in logical units and to evolve from the simple hardware generated format to a final and complex structure with minimal overhead. The offline software, based on the ZEBRA memory management package [13], dictated the choice of the format.

ZEBRA allows the building of data structures with quick access to logical data units and to transport data easily among different computers. The basic unit of a ZEBRA structure is a "bank", a collection of informations with logical affinity. Several banks can be organized and linked to each other with pointers. Ten control words placed at the beginning of the bank specify the contents. In the online environment, a single event

is represented as a linear and linkless structure consisting of six banks, one for each subdetector, one for the trigger data and one with general informations (Figure 3a).

Depending on the subdetector segmentation and read-out electronics, data are spread among several LRS1892 memories before being merged by the Event Builders. Each memory contains a raw data blocklet which is the smallest data entity to be dealt with, and the final event structure is built around it in three consecutive steps. In addition to the detector data, a small amount of trigger data have to be added along the data flow. The internal structure of the data blocklets is fully transparent to the Event Builder and interpreted only by decoding routines within the level-3 trigger algorithms and the offline analysis programs.

In each subdetector partition, including the trigger partition, a standard bank frame is partially filled at initialization with data and control words unchanged for each event. The informations such as the event length or the status word, indicating the validity of the data, are filled event by event by the Event Builder. During the data transfer the raw data blocklets are enclosed by wordcounts from different devices (BM and DSM) and geographical markers identifying the data source. This format provides the minimal structure to easily search for, read or skip the data in multi-event buffer memories. It allows the tracing of data transmission errors, it is useful in locating the origin of the data and aids in identifying hardware faults. A total wordcount and a marker indicating the address of the source FASTBUS segment are placed before the data. The final bank format is illustrated in Figure 3b.

Once the data from all subdetector and trigger memories are verified, a data transfer takes place into the emulator memory with a special bank in front of the detector banks containing a description of the event such as the total event length, the data type, the run/event number and the beam energy. The emulator, receiving this full event structure, appends the results of the level-3 trigger at the end of the trigger bank. If the event is accepted, this will be the final format to be written on tape.

6. The FASTBUS/Host Interface

Interfacing between the FASTBUS system and the online computer (VAX 8800) is done by means of the CHI (Cern Host Interface [14]). This module is a powerful FASTBUS master/slave equipped with a fast interface card for a direct connection to a host computer. Data can be transferred over the interface at a maximum speed of up to 3 Mbytes/s. The local CHI processor (68030) holds a large variety of user callable procedures such as the Standard FASTBUS Routines in EPROM. Any of these routines is executed locally either called by a CHI program or from the host computer via Remote Procedure Calls (RPC [15]). In this way the event read-out can be executed locally at the CHI and only parameters and results have to be exchanged between the CHI and the host. Consequently, the software taking care of the data transfer from the hardware to the online computer, consists of two distinct parts. One part, running on the CHI, performs all FASTBUS transactions to read the data and makes data checks such as the validation of the event structure; the second part, running on the host, is resumed via

interrupt if data are available and its only task is to pass the data to the main online program.

7. System performance

The system has been used for the data taking for over a year. The failure rate of the modules was low and the availability of the system was about 95%. The average event size is about 40 kbytes and the maximum level-1 trigger rate was 10 Hz.

The transfer rate between the LRS1892 memory as a source and the DSM as a destination is 11.4 Mbytes/s in the block transfer mode and is determined by the memory access speed of the DSM. In the first step of event building, most of the time is spent by the GPM and the time actually used by the BM for the data transfer is less than 10 percent of the total time. The GPM requires about 1.25 ms to extract information from each source memory such as the event length, trigger type and event number.

The second step of event building involves more complicated calculations than the first step. The CHI processor, which is used in place of the GPM for its higher processing power, needs about 3.3 ms to check event synchronization in all DSM's, choose an emulator and prepare the BM transfer. The transfer rate between the DSM and the emulator memory is 8 Mbyte/s and an average time spent in the transfer is 5 ms.

Data are moved from the emulator to the VAX using the CHI interface as intermediate buffer. The data transfer from the emulator memory to the CHI memory runs at about 6.6 Mbyte/s and from the CHI memory to the VAX memory the limit is about 1 Mbyte/s due to the software overhead. A typical data transfer rate of 150 kBytes/s to the VAX is well supported by this throughput.

Although we found the FASTBUS hardware reliable in general, several specific problems were encountered. The FASTBUS cable segments occasionally showed intermittent problems in the FASTBUS transaction and a painstaking debugging was necessary to locate and identify the problem.

The DSM showed a data transfer error when two ports were simultaneously accessed. The error rate was very low but it was enough to stop the system a few times a day due to the high rate of data, 500 Mbytes per hour, carried through the system. We have circumvented the problem by avoiding the simultaneous access of the DSM by a software flag. This has caused only a minor degradation of the throughput.

The standard FASTBUS software, although very useful for debugging, is not well suited for fast and efficient transactions. Many routines had to be tailored and rewritten to match the speed and performance requirements of the L3 experiment.

In order to enhance the reliability and throughput of the system, we are planning to exchange GPM's by the CHI processor modules with approximately 3 times faster processing speed. The designing of a new dual port memory is under way and the current DSM modules will be replaced soon by the new memory. These changes will make the system capable to handle a total level-1 trigger rate of 100 Hz which is expected for the high luminosity operation of the LEP collider.

8. Conclusions

The Fastbus Data Acquisition system of L3 has proved its reliability during one year of data taking. It handles the current trigger rates and data sizes without any problem.

9. Acknowledgements

We wish to thank A.Fucci/CERN-DD for helpful discussions and we would like to acknowledge the contribution from D.Fong/University of Maryland and from the CERN DD/OC group. We particularly would like to thank R.Divia for many useful discussions and for the continuous help he provided.

References

- [1] L3 collaboration, B.Adeva et al., "The construction of the L3 experiment", Nucl. Instr. and Methods A289, (1990) 35
- [2] IEEE Standard Fastbus Modular High-Speed Data Acquisition and Control System Hardware: IEEE Std. 960-1989, Software: IEEE Std. 1177-1989
- [3] R.Bizzarri et al., "The L3 Energy Trigger", Nucl. Instr. and Methods A283, (1989) 799
- [4] S.P.Beingessner et al., "The Second Level Trigger of the L3 Experiment", AIP Conference Proceedings 209, (1990) 598
- [5] P.Bähler et al., "XOP a fast versatile processor as a building block for parallel processing in HEP experiments", CERN/DD/85-15
- [6] P.F.Kunz et al., "The 3081/E Processor", CERN/DD/84-04
- [7] LeCroy LRS1892 user manual
- [8] Struck STR351 DSM user manual
- [9] Kinetics F930 Block Mover instruction manual
- [10] Struck STR500 GPM user manual
- [11] J.Blake, "Microprocessor Cross Software under VAX VMS", CERN/DD/PRIAM (1989)
- [12] E.M.Rimmer, "Software for FASTBUS", Nucl. Instr. and Methods A283, (1989) 399
- [13] R.Brun et al., "ZEBRA user guide", CERN/DD (1987)
- [14] T.Berners-Lee et al., "Cern Host Interface", CERN/DD
- [15] T.Berners-Lee et al., "RPC User Manual", CERN/DD

Figure 1: Structure of the L3 DAQ and Trigger system

Figure 2: The L3 FASTBUS system

Figure 3: The L3 Event format: a) linear bank structure; b) bank format