CERN/EF 84-12
2 August 1984

CERN LIBRARIES, GENEVA

CM-P00061279

TRENDS IN NEW COLLIDER EXPERIMENT DATA ACQUISITION SYSTEMS

Ph. Gavillet
CERN, Geneva, Switzerland

ABSTRACT

New collider projects such as the SLC and the Tevatron in the States, or LEP in Europe are underway and future generation machines are under consideration for investigation of the TeV domain of high energy physics. Spectrometers at these facilities are of growing size and complexity due to both the increase in energy and the introduction of new detection techniques. The resulting trigger and data acquisition systems have to treat enormous data volumes. New approaches have to be adopted to deal with the collection and filtering of event data. The availability of high performance micro- and specialised processors and of high speed modular bus systems allows the design of data acquisition architectures which are versatile and powerful enough to ensure safe event collection in a form suitable for prompt analysis of new physics processes.

Invited Talk given at the "Topical Seminar on
Perspectives for Experimental Apparatus at Future High Energy Machines"
22-25 May 1984, S. Miniato, Firenze, Italy

1. INTRODUCTION

The current high energy physics environment is characterized by the successful exploitation of collider machines such as the CERN $p\bar{p}$ collider. Higher energy accelerator projects such as SLC and LEP are already launched. They aim to open the vast and exciting physics domain heralded by the recent discovery of the $W^\pm$ and Z^0 particles. Progress in beam technology leads to the expectation of very high luminosities (i.e. high collision rates) at these future facilities. The planned detectors have basic features similar to the present ones: solenoidal magnetic field surrounded by cylindrical detector components closed by end caps, but the increase in energy (with the increase in average number of tracks) necessitates new types and larger size of detectors. On the other hand the impressive advance in electronics technique makes it possible to build these devices with even finer granularity than previously and at a reasonable cost.

Large amounts of data for events produced at high rates will have to be collected by data acquisition systems including the necessary online filter processing of background data to keep the volume of stored information to a manageable level. The availability of new high performance bus systems, microprocessors and minicomputers should greatly ease this task, but major difficulties will be a challenge to be faced in the realisation of these acquisition systems. We try to identify them in this paper and discuss the present approaches undertaken to find suitable solutions.

In sect. 2 we briefly review the environment which conditions the data acquisition features of future collider experiments. A typical trigger and data acquisition system for such an experiment is described in sect. 3 and some introductory remarks on the computer methods used in the design of the system are given in sect. 4. Sects 5 to 8 present in more detail the three consecutive steps of the acquisition procedure: the freeing of the front-end electronics, the full event readout, and the storage on recording devices with their associated trigger and filter levels. We envisage in sect. 9 the integration of data acquisition system into outside environment and finally raise in sect. 10 how offline data analysis is directly connected to the online aspect of data handling.

2. CHARACTERISTICS OF COLLIDER EXPERIMENT DATA ACQUISITION SYSTEM

The data acquisition environment of collider experiments is primarily determined by the collider machine, the detector and the physics program at this detector. They impose the main features of the acquisition system with which the design has to cope, given various technological, financial and expertise constraints.

2.1 The collider

Several collider parameters fix basically the event rate and quality. They are the beam-crossing rate (10^3 to 10^6 Hz - 50 kHz at LEP), the luminosity, the beam energy and properties (Focus, spread, ...), the background conditions in the interaction region (which depend on the neighbouring magnets, collimators, vacuum state ...). Most of them can be subject to large fluctuations which the acquisition system should be able to accommodate.

2.2 The detector

Standard collider detectors such as "OPAL" shown in fig. 1, are an assembly of embedded cylindrical detector components of increasing radius closed at each extremity by end caps. The magnetic field is solenoidal with its axis along the detector axis. Typical detectors are vertex detectors, central detectors, electromagnetic and hadron calorimeters and muon chambers. Their number is often in excess of ten units which corresponds frequently to more than 100000 electronics channels, to be housed in more than 100 front-end crates. It is to be noted that the central detector itself can amount to more than half of the total electronics channels.

The averaged raw data per event can reach several megabytes largely due to the more common use of time-sliced digitising methods. It forces data compaction in situ, to keep the transported data to a level acceptable by the storage device, typically $\sim 1/4$ megabyte at a few Hz input rate.

2.3 The physics program

The physics aims naturally influence the design of the data acquisition system. From the study of the largest cross section interactions ("bread and butter" physics), through the analysis of rare processes up to the search for new particles, one normally wishes to completely cover the physics phenomenae in the energy range of the machine. Accompanying processes such as Bhabha scattering, $\mu^+ \mu^-$ production in $e^+ e^-$ machines should also be measured. Cosmic ray particles should be recorded as they are very useful for monitoring and surveying the detector.

This translates into the need for versatile acquisition systems prepared to pick up and treat all sorts of events produced at variable rates.

2.4 Purpose, objectives and constraints of a data acquisition system

Once the experimental framework has been defined and well understood, the purpose of the acquisition system becomes that of gathering all events of physics interest with minimum amount of background data under conditions of minimum dead time to save precious machine time. In addition the system should be flexible enough to adopt to new, even unexpected physics as well as to fluctuating running conditions. Moreover a possible growth should be foreseen to incorporate new detectors or for upgrading of the system itself [1-5].

The main objective is well defined: read, filter and record basic event data after successive trigger and filter levels of increasing selectivity power. For this goal a modular, hopefully simple (easy to operate), and reliable system should be constructed. Its architecture is tree oriented, imposed by the reality that individual detectors are built separately in scattered institutes and have to operate themselves as elementary data acquisition systems.

The integration of detectors into a coherent system is performed in hardware via a connecting network such as FASTBUS. Control and monitoring are inserted at both the detector level via a distributed microprocessor system under the control of host minicomputer and at the system level, where central management of the various detector cells is achieved by a central operation computer.

The constraints in the realization of the system are first of technological order. The selection of the networking bus, the choice between commercial microprocessor boards or home made special processors, the specification of the computers, their interfaces to the bus, the recording media ... are never simple but should not obscure the remaining considerable hardware and software effort to assemble the whole system.

Money, availability of manpower and expertise put obvious limits to the project. One should have in mind that the data acquisition system (not including the front-end electronics) amounts to about 5% of the detector cost while the detector electronics itself represents about one third of this cost.

3. DESCRIPTION OF A TYPICAL TRIGGER AND DATA ACQUISITION SYSTEM

The projected data acquisition systems for SLC, the Tevatron [5] and LEP, in particular those of the OPAL, ALEPH, L3 and DELPHI Collaborations [1-4] have large similarities (figs 2-5). Their functionality follows a three-level structure as shown in fig. 6 and described below.

3.1 Fast triggers and freeing of the front-end

From the time of beam cross-over, a pretrigger is formed based on pick up signals or patterns. It aims to detect element(s) of track(s) and (or) large neutral energy as the indication of an interaction. It should deliver an answer before the next beam crossing ($\sim 20 \mu\text{s}$ in the LEP case). Its selectivity is expected to reduce the rate from beam-crossing frequency ($\sim 50 \text{ kHz}$) to less than 1 kHz .

The second level trigger coincides with the end of drift which can arise a few beam crossings later. This time is profitably used to elaborate the trigger answer. A dead time of a few % is introduced due to the lost crossings. One expects the rate to be reduced to less than 20 Hz .

In case of a valid trigger, one needs to free the front-end memories fast, in order to reactivate the detector as soon as possible. At a second level rate of 20 Hz (50 ms average spacing time between valid triggers), if one wants to keep the detector idle time to less than 1%, no more than $500 \mu\text{s}$ can be given to the detectors to transfer their data from the front-end memories to a front-end buffer. We shall see in sect. 5.2 that this buffer should be preferentially designed as a multievent buffer attached to each detector.

3.2 Trigger validation and full readout

De-randomisation is achieved by the front end multievent buffer which allows the second stage of the acquisition procedure to run asynchronously from the first one. Detector data are read simultaneously on parallel branches into a buffer where the full event data are collected (fast trigger information is transferred separately).

This process is bounded by the readout time of the more bulky detector branch – often the central detector – which can last up to 10–20 ms. Time is available to run parallel trigger validation computations based on selected data, in the shadow of data transfers (sect. 6.3). The selectivity of this third level trigger should reduce the rate from 20 Hz to about 5 Hz at the expense of 1–2% dead time. For similar reason it is convenient to give to the event buffer a multievent capacity as is done for the front-end one.

3.3 Filter/tagging and output of data streams

The last stage also runs asynchronously from the preceding one. Full event data are passed into one of parallel processors where filter and signature algorithms are run to purify further the sample and possibly assign events to dedicated output streams for immediate physics investigation. Typical time scales for this stage is of a few tenths of second. Two to three % time is lost to bring down the rate to the level of 1–2 Hz manageable by the storage devices (sect. 7).

Let us remark that the system is made of successive processes characterised by decreasing rates allowing for increasing processing time at each level. It is also worth mentioning that the system can be operated in test and debugging phases, in a simple synchronous mode where the front-end and multievent buffers are used as single event buffers, and the third and fourth trigger levels are switched off. In this mode an event can be transferred straight, after being accepted by the fast trigger.

4. COMPUTER AIDED CONCEPTION OF DATA ACQUISITION SYSTEM

The above system can well be abstracted and seen as a bus system with multicorrelated stages, submitted to a variable input depending on the luminosity, energy, background conditions ... As such it can be simulated at the design phase, with the aim of understanding better the characteristics of the input events and consequently of optimising the trigger and data acquisition architecture.

4.1 Use of simulated events for trigger design

The method consists of generating physics and background events and of tracking them through detectors. The resulting digitizings are used to evaluate the possible trigger schemes at each level. The choice of a suitable segmentation, the definition of correlations between detectors, and the efficiency and timing of filter algorithms can be optimized in this way.

The usual tools are FORTRAN programs run on main frames such as the LEP simulation program of physics events based on the Lund method [6]. The background events, mainly off momentum $e^\pm$ in the LEP case, are generated by Monte-Carlo simulation from the energy and angular distributions of electrons and photons in the vicinity of the interaction region, as provided by the machine experts [7].

The performances of the proposed triggers are analysed in a trigger evaluation program which will be tuned ultimately on real events. This program should later be extended to become the basic software processor through which new triggers for future experiments will be designed.

4.2 Simulation of data flows through buffering and local processing stages

The study of the data flow through the data acquisition tree can be tackled using queuing theory [8]. In this framework each step of the acquisition procedure is described in terms of an input queue with given time distribution (e.g. Poisson like with average λ), a buffering stage of event depth k and a processing stage of n processors (servers), each of them with the proper processing time distribution (constant, exponential with average processing time μ , ..) and a rejection rate R .

The expected outcome is the optimization of the number of servers and buffer sizes as function of the rates and tolerated deadtimes. A simple example is the full readout process described in sect. 3.2. The input rate is 20 Hz (50 ms) and the processing time i.e. readout time is in average 20 ms. The deadtime for this process is simply given by the formula

$$\text{Dead time (\%)} = \frac{(1 - \eta)\eta^k}{1 - \eta^{k+1}}, \quad \eta = \mu/\lambda$$

Table 1 gives the dead times induced by different values of η and k as deduced from the formula. One can readily see the benefit of having a multievent buffer 4 to 5 events deep to reduce the deadtime in our case from about 30% to less than 1%. Interesting estimates can hence be obtained by computation of simple queuing theory formulae (fig. 7). Some users write their own application programs using high level languages such as FORTRAN, PASCAL, ALGOL or ADA if not specific simulation packages like SIMULA.

The complete simulation of complex architectures such as the FASTBUS data acquisition network we are considering, is more difficult. The queuing theory equations become coupled between adjacent steps and have to be solved numerically. It is preferable to use Monte-Carlo methods to model the exact configuration. An interesting approach by the ALEPH team [9] has been to adopt the IBM General Purpose Simulation System (GPSS) normally used for computer architecture studies. The package is standard, efficient and easy to use. More important, its predictions can be cross checked which is often a major difficulty with other methods. Hence it could be the right sort of tool to fully simulate data flow in a data acquisition network.

5. FAST TRIGGERS AND FRONT-END FREEING

This is the first process in the data acquisition and it works at the beam crossing frequency.

5.1 Fast triggers

Such triggers aim to detect hints of interaction in the form of one or more charged track segments, release of electromagnetic or hadronic energy or muon signal pointing towards the crossing region. They work on pick up signals or already digitised data depending on the available time which does not exceed that of a few beam crossings ($< 100 \mu\text{s}$).

This is the domain of matrix coincidences, hardwired modules and track finding processors reviewed in the contribution of S. Conetti to this seminar.

5.2 Front-end freeing

After an event candidate has been accepted by the fast triggers, transfer of the corresponding data has to be initiated. The objectives are twofold:

- Free the front-end digitiser memories rapidly ($\leq 500 \mu\text{s}$) to minimise the detector idle time to a few %.
- Buffer locally these data in a front-end multievent buffer to allow the operation of the following stage - the full readout - with minimum dead time.

The realisation of this transfer and buffering depends on the detector characteristics, in particular on the digitiser type and the data volume. For example detectors performing time samplings frequently have to compact their data before transfer. More generally the time constraints impose a high parallelism of data paths and a high bandwidth bus.

Let us discuss in more detail four implementation schemes as foreseen by DELPHI, which cover the more usual cases. They are sketched in figs 8(a-d), using the notation T for the current trigger event (in the process of transfer after validation by the fast triggers) and R for the presently readout event (asynchronously extracted from the front-end buffer).

(a) Front-end buffer on front-end boards - No preprocessing

In this simple case (fig. 8(a)) T data are buffered on the board into time shared memories including the necessary level of control to suspend R data readout when a T trigger happens. This interruption occurs at the fast triggers rate and makes it difficult to pipeline the R data readout.

(b) Front-end buffer on front-end boards - Local preprocessing

This scheme is convenient for detectors with large amounts of data per channel for which zero suppression is mandatory but cannot be achieved easily within the time scale of $\sim 500 \mu\text{s}$. Raw T data (fig. 8(b)) are buffered locally. Data compaction is performed asynchronously as a preliminary phase of the full readout, i.e. R data are reduced on the front-end board to a small volume R'. This multiplexed preprocessing on the board has to be suspended on write request of a T event but now, R' data can be readout in pipelined block-transfer mode.

(c) Additional processing in front-end segments

More ambitious front-end processing can be achieved by local processors (scanners) residing in front-end crates (fig. 8(c)). First, all R' data are sequentially read and formatted into R'' data blocks for further readout. In addition part of the R'' data is treated (e.g. pedestals subtraction, clustering, ...) for consecutive third level trigger processes (sect. 6.3) which will operate on these selective R''' preprocessed data.

(d) Front-end buffer in front-end segments

It is a simple implementation of a front-end buffer (fig. 8(d)). The electronics channel boards are read sequentially into the local scanner memory. On receipt of a

fast trigger, the scanner must be interrupted from the current task. The major limitation of this solution is the difficulty to cope with the severe time constraints. Consider front-end crates with about twenty digitiser modules to be read sequentially in less than 500 μ s i.e. in less than 25 μ s each. A very high bandwidth bus is necessary as soon as the data volume per board is in excess of a few tens of words.

Previous considerations show that the front-end data collection for new collider acquisition systems has now to include local multievent buffering, data preprocessing and a fast connecting bus, i.e. the demand will grow for buffering and optional preprocessing on the electronics channel board itself and for availability of multipurpose front-end scanners based on microprocessors embedded in the standard bus.

6. THIRD LEVEL TRIGGER AND FULL READOUT

This stage involves complex concurrent or parallel procedures such as:

- The full readout from front-end buffers to the multievent buffer which consists of parallel transfers of large data blocks on several detector branches controlled by a distributed microprocessor system.
- The intermittent spying on data from individual detectors.
- The third level trigger which is a succession of selective readout, local computations and central decision processes.

It illustrates well why a versatile multipurpose bus system should be adopted and we proceed to describe this acquisition phase in the FASTBUS framework [10] which seems to be the more suitable choice.

6.1 Full readout procedure

Various possible schemes are usually considered (until a full simulation of the system selects the best one!). We shall review the implementation sketched in fig. 9.

Let us start with the stand-alone detector case. On fig. 9 one identifies the front-end units (with their front-end buffers) connected via a Segment Interconnect (SI) to an Equipment Computer Crate (ECC) attached to the host control computer

(called Equipment Computer: EC) via a Computer to FASTBUS Interface (CFI). In the local ECC a General Purpose Master (GPM) intelligent module analogous to the scanner previously mentioned acts, at this moment, as a local readout controller or Event Supervisor (ES).

The front-end buffer status is known to the ES when it decides to extract an event. The ES processor triggers a block mover module (BM) encharged of fast data blocks I/O, to perform a so-called "sparse data broadcast" to find out which front-end modules have valid data for the event under consideration. In its turn, the BM converts the received source module list into a sequence of block transfers to a dual port memory (Dual Slave Memory: DSM) in ECC crate. The detector minicomputer (EC) can then access these data through the CFI interface.

The full spectrometer readout is an extrapolation from the stand-alone case. Another GPM residing in a central crate operates as Central Event Supervisor (ES), i.e. it takes full control mastership. To read an event, it first dispatches to the local ES's via broadcast addressing, the necessary information such as the event identifier and the route map. It then releases control to the local ES's which initiate the full readout of their detector part as in the stand-alone case, except that now the destination is the Multi Event buffer (MEB). On completion of readout local ES's notify the central ES which stores the control information (number of blocks, partitions, error messages, ...). Finally, the central ES appends the third level trigger results to the event data in MEB (sect. 6.3).

6.2 Spy facility

Spying from the EC is a variation of the full readout. In this mode the BM in the front-end units is told by the local ES - in agreement with the central ES and on request of the detector EC - to either broadcast the detector data to the DSM in the ECC in addition to the MEB, or to operate a second move of the same data from the front-end buffer to the ECC.

6.3 Third level trigger

At the full readout stage the event rate has been drastically reduced by the fast triggers to less than 20 Hz in the case of LEP (sect. 3.1). A powerful third level trigger can be introduced in the same stage without additional dead time, provided it does not last more than about 20 ms (including the readout time (sect. 3.2)). Obviously the heavily loaded detector branch(es) which can take up to 20 ms to be

readout cannot participate in this trigger. Fortunately the majority of detectors can be read at FASTBUS speed in a few ms only. Therefore most of the detectors can find time after (or before) being read for trigger calculations. Moreover these computations can be directed by the earlier fast trigger predictions i.e. dedicated tasks can be defined beforehand which will operate on a restrictive fraction of detector data.

A possible implementation scheme could work as follows. When the central ES decides to treat an event residing in the front-end buffers, it first gets the fast trigger predictions from the central trigger supervisor (fig. 3) which holds this information buffered as normal detector data in its own front-end buffer. Then the central ES can preclassify the type of trigger to determine which task to execute on which detector data and define the selective readout pointers. Next it triggers the local ES's to perform their own selective readout and local computations. It is a distributed real time processing of parallel (possibly coupled) tasks whose execution is under control of the central ES. On completion of their task, local EC's will notify the central ES, via semaphore type of communication protocol, and this in its turn will collect all individual results. The central ES can from this moment evaluate the final decision from the partial answers. In case of trigger rejection it will reset pointers in the multievent buffer, otherwise it will append the third level information to the event data.

In the available 5 to 10 ms, advanced trigger validation and the application of pretagging algorithms can be carried out such as linearised or spline fits on one or more tracks for momentum determination, intersection of tracks within the interaction region, hadronic or electromagnetic energy estimate, jet axis computation ... Worth noticing that such a realisation of a third level trigger based on the ability of the GPM's to perform trigger computations as well as readout functions, does not require additional hardware.

The third level trigger is not foreseen by all LEP collaborations but it seems really attractive, as can be judged on fig. 10 where the expected efficiency improvement for the upgraded UA1 acquisition system is shown.

We have stressed how the advanced functionality of the third level trigger and readout stage spreads among several processes which use profitably the built in facilities or characteristics of FASTBUS such as:

- high bandwidth,
- sparse data scan,
- broadcast block transfer,
- traffic route definition,
- arbitration scheme for competing tasks,
- multicrate assembly which eases the implementation of distributed microprocessor systems.

It is, in fact, the system approach of FASTBUS which is conveniently exploited to initialise and run a data acquisition and processing network as an homogenous system.

Several collider experiments such as SLD and CDF in the States or ALEPH, L3, OPAL (for the front-end electronics) and DELPHI at LEP have adopted FASTBUS. Developments and new issues are growing, and broad support from industry is expected. The next generation of data acquisition systems as seen by the ECFA recent workshop on future colliders [11] will request a "VERYFASTBUS" of about one gigabyte bandwidth! Such a perspective for the nineties makes it even more important to evaluate and use FASTBUS on a large scale.

We have shown that the full readout and third level phases will be the domain of versatile microprocessor applications. The desired qualities of these microprocessors include 32 bit word size, large addressing space and memory, high speed, support of high level languages such as PASCAL or FORTRAN, an efficient operating system, in particular with a high performance real-time kernel ... and one will certainly prefer them directly implemented on a standard board of the bus system to simplify the interfacing.

Some candidates seem to be already studied by the LEP experiments such as FAMP [12], GPM [13], XOP [14]. The choice is clearly guided by the adoption of standards, as is the case for the 68 K family at CERN and (or) the guarantee of wide use and strong local support. Surprisingly few experiments envisage to buy commercial boards but the situation is evolving fast. In particular, in the VME bus environment 68 K based modules now available are being selected by experiments, such as UA1, which prefer this commercial standard. On the other hand FASTBUS supporters are expecting industrial firms to provide 68 K based FASTBUS processors such as the GPM.

7. FILTER/TAGGING STAGE

At the output of the third level trigger the event rate is expected to be less than 5 Hz if triggers were selective which sometimes is not desirable. Even in this favourable case the background data dominate. In the case of LEP physics the good over background event ratio will be at best 10 to 20% in the Z^0 region and as low as a few parts per thousand outside the Z^0 peak. Clearly physicists cannot satisfy themselves by just collecting events at magnetic tape speed. The accumulated data (about 100 tapes per day and 10000 tapes per year) necessitate enormous CPU power to be processed (sect. 10) and cannot be treated in step with the data taking. Online filtering of events is therefore mandatory for both maximising the good over background event ratio on tape and picking up candidates for rare processes for immediate offline analysis.

The method consists of running online filter algorithms normally run offline on full event data, i.e. of moving upstream part of the event processing. Real time execution obviously limits the available processing time. Indeed simple queuing theory (sect. 4.2) shows that a filter configuration with up to five parallel servers preceded by a multievent buffer 5 to 10 events deep does not allow for much more than half a second in input/output rate conditions of 5 and 2 Hz respectively.

The filter algorithms are complex computations subject to frequent modifications, therefore written in a high level language such as FORTRAN. If executed on-line, regular cross checks have to be made by comparing their results to those obtained offline on the same data. It implies that the algorithms can be transported in a transparent way from offline to online, i.e. the online processor should run FORTRAN as does the main frame.

The event filter stage is implemented via a battery of emulators inserted inbetween the multievent buffer and the data logger computer (figs 2 and 3) and interfaced to them through high bandwidth interface modules. Events selected by the emulators, in particular the candidates for rare processes, can be directed to dedicated output streams.

In the same configuration emulators can sample events for exhaustive validation controls up to full event reconstruction. In this way interactive analysis can be performed during data taking for regular check up.

Some comments have to be added to the online use of emulators. One should first remember that these special processors were designed [15] for supplementing offline event processing and are widely used as such. When pilot experiments showed how helpful they are for event tagging – the 168/E's contributed a lot to the on-line recognition of W and Z^0 events at UA1 [16] – they were accepted as powerful online processors. Nowadays it is not clear if emulators should be devoted to filtering out background events or to the hunting of rare events, for example, candidates for new particles. This dilemma could still exist at LEP time. It results from the fact that full event treatment cannot be exhausted online because of the lack of time. To overcome this obstacle, systems with large event buffers as front-end reserve for numerous emulator branches have to be designed. The recent ECFA workshop on future LHC experiments [11] estimate the need for emulator-type processors to be up to 1000 units per data acquisition system! Unfortunately the present state of the art does not permit to ground this extrapolation. Of course modern emulators such as the 370/E, an upgraded version of the 168/E, already exist [17], or will be soon available in the form of the 3081/E [18] but their commercialisation is still an open problem. No real alternative appears currently. Medium term candidates such as 68 K (or microVAX?) based multiprocessor systems such as the VIRTUS project [19] can compete if they improve substantially in speed.

8. FROM EMULATORS TO MAGNETIC TAPE

The outputs from emulators have to be transported over a long distance (150–200 m) to the storage medium. This transmission from the location of the detector electronics in the experiment pit up to the ground level, should maintain the FASTBUS bandwidth. This can be ensured by using fiber optics technology. The last stage of dumping data on recording device depends onto the data logger computer (bus and memory bandwidth), the input interface and output device performances.

To illustrate where are the real constraints and limits we consider the VAX model already chosen by various experiments. Fig. 11 shows the VAX11/780 architecture and table 2 gives relevant figures for the discussion. First point to be made clear: although the main bus has a respectable bandwidth the computer throughput heavily depends on the operating system, the load induced by tasks run concurrently with the data acquisition, and the way this latter task is performed: pure DMA transfer, event formatting, additional event processing, ... Measured throughputs are often well below the main bus specification.

Interface to a public I/O bus is limited to few megabytes/s and only a FASTBUS interface to UNIBUS (via CAMAC) is currently available. Special fast interfaces to a dedicated bus, such as DR780, exist but not widely used and only for the VAX 11/780 host computer.

Traditional magnetic tapes could be soon superseded by optical disks already on the market. As the bandwidth of these devices is still limited to about 1 Megabyte/s, some experiments keep an interest in dedicated hyperchannel data links connected to high capacity mass storage devices.

9. INTEGRATION OF DATA ACQUISITION SYSTEM IN THE LOCAL ENVIRONMENT

Communication is known to be the most vital element in a distributed computer system. The 10 to 15 EC's, the DAQ and OP computers of a collider acquisition system have to be interconnected to allow easy access from the host laboratory as well as from outside institutes.

The proposed solution (fig. 12) is to link all computers in the experiment environment by a Local Area Network (LAN) . The majority of terminals are attached to the LAN via packet assembler/disassemblers (PADs) making it possible for any terminal to communicate with any computer in the network. The bandwidth (~ 10 Mbits/s) should support the anticipated traffic consisting of normal terminal I/O activity, commands and messages exchanges, file transfer and limited event data circulation.

Experiments do not foresee to develop and build their own network facilities. They would rely on standard industrial products, now on the market, with the appropriate level of performance and support, such as ETHERNET.

Communication from the experiment location to the outside world goes through a gateway to the host laboratory network (e.g. CERNET) which itself is linked to the national telecommunication networks via an X25 connection. Three levels of networking are therefore required to allow access from (to) any place in the collaborating institutes to (from) the experiment site and to ensure a good level of communication.

10. OFFLINE DATA HANDLING

Collecting data at full magnetic tape speed makes onliners satisfied but is only part of the complete data processing. Next comes the huge task of analysing the accumulated data with the goal of extracting physics results quickly.

Table 3 presents the breakdown of computer time as estimated for the first year of a LEP experiment. It crudely reveals those aspects of high energy physics computing which are large CPU consumers: number crunching production, program development, interactive event analysis. These activities have to be approached in a new fashion in view of both their increasing importance and the advances in computing technology and methods.

An exhaustive discussion of the multiple aspects of this field is far beyond the aim of this article. We will restrict ourselves to brief mention of the steps already undertaken in these domains and present a tentative solution to integrate offline data handling in a multipurpose computing system.

Number crunching computations include Monte-Carlo studies, event filtering and reconstruction. The increasing complexity of physics events with higher energy, coupled to the ability of new colliders to produce them at high rates, results in a tremendous need for CPU power to treat them. Up to 4 or 5 IBM-168 equivalent will be necessary each year to process the events of one LEP experiment. The corresponding financial investment exceeds the available resources. To overcome this difficulty, it has been envisaged to build dedicated emulator farms assigned to run repetitive tasks on numerous event samples. The CPU speed being not as crucial as in the online case, commercial cheap processors now (or soon) available could take over in this job. This should ease the assembly and control of these streams both from the hardware and software points of view.

Program development is going to be more and more widely spread through the collaborating institutes. This produces new needs to enable this distributed work to be undertaken efficiently, such as powerful networks, universal availability of facilities such as common data bases, software tools (source code manager, libraries, memory managers, ...), mail service etc... These requests are common to both online and offline programmers. They force the adoption of standards which limit the choice in computer equipment to compatible devices and restrict software packages to be written in a well-defined framework.

Interactive computing is playing a growing role in offline computing particularly in program development and tuning and in event analysis (event scanning, simulation, calibration studies, ...). It is better provided by dedicated systems with graphics capabilities ranging from high resolution graphics terminals attached to a host computer up to more autonomous personal workstations.

Although computing activities will be scattered, it is desirable to integrate them either locally in each institution and (or) more centrally close to the experiment. The CDF Collaboration at Fermilab [20] has proposed a concentrated system in the form of a computer cluster built around the VAX family of computers (fig. 13).

All CPU's are linked via a fast (70 MHz) computer interconnect (CI) which enables them to share efficiently common resources: disks, magnetic tape units, plotters and printers. A second interconnection through ETHERNET allows remote access to the centre. The CPU power is provided equally by farms of micro VAX's (reserved for number crunching) and VAX 11/790's, and is equivalent to 15 IBM-168's in total. The VAX 11/790 part of the complex can be configured easily either to supplement in batch oriented production or for more local interactive computing with addition of graphics terminals or workstations. Upgrading of the centre is relatively easy within the selected manufacturers products and can be tailored to the requirements of the experiment.

The CDF approach certainly favours the creation of a computer complex attached to a detector facility instead of the growth of large computer centre common to all activities of a laboratory. Thus it can be seen as a new trend in the disposition of computer resources.

11. CONCLUSIONS

We have seen that future collider experiments differ from the current ones in many aspects, in particular in the amount of electronics channels to be readout, resulting in an enormous data volume per event. To face this new situation and at least maintain the throughput distributed intelligence has to be introduced as early as possible to filter and regulate data flows. Preprocessing in front-ends by means of resident microprocessors is becoming the rule. Successive buffering and processing stages are naturally inserted to cut the flow and gradually identify good events from background data. The occurrence of performance micro- and dedicated processors

decisively help in these tasks. The construction of versatile and coherent systems is made possible when new bus standards such as FASTBUS are especially designed to monitor acquisition networks of high data fluxes. Online event processing can be extended to refined data treatment for both enriching event samples and selecting candidates of rare event types. Easy communication between the acquisition system and the outside environment should be of benefit to the system operation, in particular when the offline data handling has to interact with the data taking itself. It should finally be observed that although the new generation of collider experiments are becoming complex and sophisticated, prompt issue of physics outcomes is expected by the many people who contributed to build them.

Acknowledgements

I wish to thank my colleagues of the DELPHI data acquisition group for their contribution to this paper which was largely inspired by common discussions at the design stage of the DELPHI data acquisition system.

REFERENCES

- [1] ALEPH Collaboration, Data acquisition and data analysis, CERN/LEPC 84-8.
- [2] DELPHI Collaboration, Data acquisition and analysis in the DELPHI experiment, CERN/LEPC 84-3.
- [3] L3 Collaboration, Trigger and data acquisition system of L3, CERN/LEPC/84-5.
- [4] OPAL Collaboration, Report on data acquisition and data analysis, CERN/LEPC 84-4 and 84-2.
- [5] A.E. Brenner et al., Design of the collider detector facility data acquisition system, CDF-108.
- [6] T. Sjostrand, Lund University, TP82-3 and 82-7.
- [7] S. Stergiou, Background from off-momentum particles in the DELPHI detector, DELPHI 84-32 GEN 9.
- [8] A.O. Allen, Probability, statistics and queuing theory, Acad. Press 1978.
- [9] J.F. Renardy, Fastbus simulation, private communication from ALEPH Coll.
- [10] FASTBUS, A modular high speed data acquisition system for high energy physics and other applications ESONE/FB/01.
- [11] Summary report of the workshop on the feasibility of hadron colliders in the LEP tunnel, Lausanne, 21-27 March 1984.
- [12] L.O. Hertzberger, Comp. Phys. Commun. 26 (1982) 79.
- [13] H. Müller, The GPM, a general purpose programmable Fastbus master (F6808), CERN, EP Division, 20 May 1984.
- [14] T. Lingaerde, XOP, DELPHI 82/45 ELEC; A second generation fast processor for online use in high energy physics experiments, Proc. of Topical Conf. on the application of microprocessors to high energy physics experiments, CERN 81-07.
- [15] P.F. Kunz, Use of emulating processors in high energy physics, Proc. of the Int. Conf. on experimentation at LEP, Phys. Scr. 23 (1981) 492.
- [16] J.T. Carroll et al., Data acquisition using the 168/E, Proc. of the three days in depth review on the impact of specialized processors in elementary particle physics, Padua, 23-25 March 1983, 47 (1983).
- [17] H. Brafman et al., A fast general purpose IBM hardware emulator, Proc. of the three days in depth review on the impact of specialized processors in elementary particle physics, Padua, 23-25 March 1983, 71 (1983).
- [18] P.F. Kunz et al., the 3081/E Processor, Proc. of the three days in depth review on the impact of specialized processors in elementary particle physics, Padua, 23-25 March 1983 (1983) 84.
- [19] VIRTUS Project, UCLA-VIRT-8405.
- [20] C.T. Day, An all VAX model for the next computer acquisition, CDF Note.

TABLE CAPTIONS

- Table 1 Example of queuing theory predictions of dead time as function of buffer capacity and reduced processing time.
- Table 2 Characteristics of data logger (VAX 11/780), input interfaces and output storage devices.
- Table 3 Breakdown of computer time required in first year of LEP.

TABLE 1

Dead time in %:

η	1	2	3	4	5
					(Buffer capacity)
.1	9.1	.9	-	-	-
.2	16.6	3.2	.6	-	-
.4	28.5	10.3	3.9	1.6	0.6
.6	37.5	18.5	10.1	5.7	3.2
.8	44.4	26.2	17.4	12.2	8.9
1.0	50.0	33.3	25.0	20.0	16.7
2.0	66.6	57.1	53.3	51.6	50.8

TABLE 2

<p>1. <u>VAX11/780 data logger</u></p> <ul style="list-style-type: none">- SBI bandwidth: 13.3 Megabytes/s- Major load from memory access (.8 to 1.9 μs per write cycle)- CPU is second load producer due to frequent memory to memory copy- Mass storage adapters (UBA-MBA) bandwidth ~ 1 Megabyte/s
<p>2. <u>Input interface</u></p> <p>(a) To public I/O bus</p> <ul style="list-style-type: none">- VAX11/780: To UNIBUS (1-2 Megabytes/s) using CFI interface- MicroVAX: To Q22 bus (2-3 Megabytes/s) - Interface under development <p>(b) To special bus</p> <ul style="list-style-type: none">- VAX11/780: To DR780 32 bit parallel bus (6-7 Megabytes/s)
<p>3. <u>Output storage devices</u></p> <p>(a) Magnetic tapes: 6250 bpi, 125 ips, 2400 feet have maximum capacity of 160 megabytes and 700 kilobytes/s bandwidth.</p> <p>(b) Available optical disks: ~ 2 Gigabytes capacity and ~ 1 Megabyte/s bandwidth.</p>

TABLE 3

	<u>CP h of IBM 168 equi.</u>
(a) Assumptions	
Effective running time	2500 h
LEP luminosity at the Z ⁰	$6 \times 10^{30} \text{ cm}^{-2} \text{ s}^{-1}$
No. of real events	1.5×10^6
Background ratio	4 : 1
Analysis time per real event	20 s of IBM 168
Filtering time per event	2 s of IBM 168
No. of Monte-Carlo events	1.5×10^6
Average generation time	5 s
(b) CP time requirements	
Total analysis time for real events	8000 h
Filtering time for 7.5×10^6 background events	4000 h
Base-load for general activities	2000 h
Monte-Carlo event generation and analysis	10000 h
Calibration and DST analysis	8000 h
TOTAL	32000 h

Breakdown of computer time required in first year of LEP

FIGURE CAPTIONS

- Fig. 1 The OPAL detector.
- Fig. 2 ALEPH logical sketch of main data acquisition system.
- Fig. 3 DELPHI data acquisition block diagram.
- Fig. 4 L3 data flow design.
- Fig. 5 CDF data acquisition block diagram.
- Fig. 6 Block diagram of a typical collider acquisition system.
- Fig. 7 Efficiency estimates of the UA1 upgraded acquisition system.
- Fig. 8 (a) Front-end buffering in front-end boards.
 (b) Front-end buffering in front-end boards with local preprocessing.
 (c) Front-end buffering in front-end boards with additional processing
 in front-end scanner.
 (d) Front-end buffering in front-end scanner.
- Fig. 9 FASTBUS implementation of the full readout and third level trigger
 procedure.
- Fig. 10 Efficiency estimates of the UA1 upgraded acquisition system as function
 of the 3rd level trigger inclusion.
- Fig. 11 VAX11/780 main architecture.
- Fig. 12 DELPHI online system and local area network interconnection.
- Fig. 13 CDF all VAX models for computer cluster.

OPAL 1983

OPAL
GENERAL VIEW

Fig. 1

LEP 216.001T 1000.1

ALEPH LOGICAL SKETCH OF MAIN DAO SYSTEM

DELPHI DATA ACQUISITION

L3 DATA FLOW DESIGN

CDF DATA ACQUISITION BLOCK DIAGRAM

Fig. 5

Fig. 6

DATA TAKING EFFICIENCY

SINGLE BUFFER SYSTEM

TWO BUFFER SYSTEM

1st buffer very fast

FULL READOUT SYSTEM

1st buffer : 4ms
2nd " : 30ms

168/E < 33Hz
MT < 3Hz

UA 1 EFFICIENCY ESTIMATES

Fig. 7

(a)

(b)

Fig. 8(a,b)

Fig. 8(c,d)

Fig. 9

Fig. 10

VAX 780 MODEL

Fig. 11

DELPHI on-line system

Fig. 12

CDF ALL VAX model for computer cluster

Fig. 13