7 The data acquisition system

This section concentrates on the central part of the H1 data acquisition system, the front-end readout and triggering subsystems having been dealt with extensively in previous chapters.

From the real-time computational point of view, a total of over a quarter of a million analog channels are read-out and digitized, resulting in some 3 Mbyte of raw digitized information for a triggered event. As the time between successive electron-proton bunch crossings is just 96 ns, various levels of hardware triggering, software filtering and digital compression are employed before reducing final data sizes to acceptable storage-media recording rates. For the data acquisition several hundred processing elements are embedded largely within the IEEE VMEbus standard [54]. Descriptions of the system also exist elsewhere [55, 56].

Figure 20 summarises the overall data acquisition system and the key rates at full HERA design luminosity. Information is digitized and read-out in parallel from many subdetector partitions before being finally merged. Four main levels of hardware triggering and software filtering can be enabled on all or part of the various detector partitions. In parallel data compression and formatting reduce the 3 Mbyte of raw data to event sizes of between 50 Kbyte and 100 Kbyte so that final data recording rates are restricted, at present, to a maximum 1.2 Mbyte/s. The first level (L1) pipelined triggering system selects initial candidates for data processing from the background. After a L1 accept the front-end pipelines are held. Dead-time then begins. A more refined hardwired L2 decision, based on combined information, can then be activated within 20 μs. Front-end electronics readout is not initiated until an early-level triggering decision is made whereupon each component has a maximum digitization time of 800 μs before the pipelines are re-enabled; a large amount of the data compression also takes place during this phase. A third level of filtering can be enabled so that should readout commence the pipelines are immediately reactivated if a L3 reject occurs.

In logical structure the detector components are merged, in parallel, into individual subdetector VMEbus crates which each contain a readout controller and memory buffer plus a fibre optic link to a coordinating event management task. A parallel array of RISC processors provides the fourth level of software-coded filtering once all of the full-event information is combined over the fibre-optics [48]. At all stages workstation intervention provides for data monitoring and detector control. To avoid saturation on global VMEbusses, extensive use is made of the local VSB specification [57]. To minimise any further dead time contributions, memory buffering is provided at the key data processing stages. Finally a local area network (Ethernet) caters for the relatively slow exchange of general operating conditions, files and even event-records between different subsystems. This also serves the external international community with current status and event information over wide-area networks.

7.1 System components

To provide a coherently managed system, between over 200 electronics crates, a baseline set of standards was established to ease maintenance and software development overheads.

7.1.1 Basic hardware components

For general-purpose real-time control within VMEbus, a common 68020/30/40 series processor card was selected [26]. On-board memory is dual-ported so that it can be accessed externally from the VMEbus (for communication) as well as internally from the processor itself. VSB local bus access can be extended to memory boards in adjacent crates but is never used as an inter-crate connection. For even more calculation-intensive applications RISC processor-based modules are employed. The RAID 8235/8239 is a single-width VMEbus/VSB board based on the MIPS R3000 processor (25/40 MHz clock) and the R3010 Floating-Point Accelerator with up to 32 Mbyte of onboard DRAM [58]. To enhance memory bandwidths and reduce instruction access times, up to 128 Kbyte of independent data and instruction cache ensure that each board has an equivalent computing power of up to 80 % of an IBM 3090 mainframe. The DPM 8242 is the basic memory module and can be equipped with up to 16 Mbyte of 70 ns static RAM, having equal priority in arbitration from the VME and VSB ports [59]. A broadcast mode selection allows any VMEbus master to write to several memories simultaneously. Both the VMEbus and VSB memory base addresses are software programmable via VME accessible registers, permitting a managing Event Coordinator task (see later) to switch units between different events. Due to their graphics-user-interface and open NuBus architecture, Macintosh computers are used extensively for the purpose of software development and operator control. A maximum of 24 VMEbus crates can be mapped directly onto each Macintosh via MacVEE and Micron, MacVEE interface card resident on NuBus [60]. No sophisticated protocol needs to be initialised, unlike other DMA-based connections.

7.1.2 VMEtaxi

In many of the front-end subsystems the standard VICbus is used to interconnect crates [61]. However for coordinating the fast data acquisition transmission protocol between the different readout subsystems, VMEtaxi modules are employed [62]. These can connect VMEbus crates up to several kilometres with fibre-optics. Figure 21 illustrates the basic philosophy. All boards are interconnected with multimode optic fibres to form a ring, using AMD 7968/7969 TAXI-chip transmitter/receiver pairs. Because of this, there is theoretically no limit on the number of such devices which can be interconnected within a single ring; in practice one is limited by the software overhead in setting up transfers. Each single-width board has a CPU controller and single level VMEbus arbiter. The software protocol, discussed further in the following sections, is purpose-written and optimised for speed and efficiency in a data acquisition environment; no FDDI or similar networking protocol is adopted. In setting up transfers between any two crates, an initial description packet is sent around the ring; those modules not engaged in the subsequent activity can enable by-pass registers so as to establish a direct connection between the two VMEtaxis that are involved, analogous with SCI [63]. During the early stages of data-taking, a 25 MHz 68020-based board was used with 125 MHz taxi chips. Upgraded Mark-2 modules are now installed which exploit 50 MHz 68030 processors and either 175 MHz or 250 MHz taxi chips. By driving double-links in parallel there is the potential for 55 Mbyte/s point-to-point data transfers between crates up to distances of several km. The link reliability has been tested to a bit error rate of less than 1 in 10^{13}. Program memory is provided for by 128 Kbyte of on-board fast static RAM with an additional 2 Mbyte of on-board extension static RAM accessible over the VMEbus by external processors. On-board EPROM and EEPROM provide for firmware storage and configuration parameters. VME64 and VSB block transfer modes are realised on the Mark-2 modules with XILINX gate arrays.

7.1.3 Software

The architectural structure of the hardware, discussed in the following section, defines how the software is written and developed. Dedicated tasks run on dedicated processors throughout the VMEbus system with operator intervention provided for via graphics-orientated computers such as Macintosh. Different languages and approaches are used depending on the application being developed. Much of the online data acquisition systems code which executes on 68000 series microprocessors, within the VMEbus, is written in either C or Assembler. Final-level filtering algorithms, which execute on the R3000 boards, have large code resources originating offline and written in Fortran [48]. Control programs which run on the Macintosh computers exploit the latest object-orientated and graphics-based packages currently on the market [64]. It can be seen that the Macintosh provides a convenient integral component in both software development and operator-control. In order to encompass the framework of VMEbus, extra tools have been developed to ease the integration into the VMEbus [65]. As a result, no formal operating system is required to run on the VMEbus boards; all basic functions can be controlled from a Macintosh via external directives through dual-ported mailbox memories. In order to enhance information exchange between control computers a dedicated package has been written based on international networking protocols [66]; this has the added advantage of enabling the complete system to be monitored from external laboratories and institutes.

7.2 System Integration

By convention the data acquisition system breaks down into three main categories for the purpose of central coordination and management. First there are the front-end "producers" of data; the end result of extensive electronics digitization from the subdetectors or "branches". Second, the data are merged and distributed to several full-event "consumers"; subsystems which monitor and record the data onto permanent storage media. Third, the system is initiated and controlled via external processes for overall system supervision and operator intervention. Figure 22 shows the physical layout of the complete system, managed centrally through several layers of software protocol, purpose-written around the VMEtaxi fibre-optic ring [67]. Common, shared, memory blocks provide for the communication between all system processors and external computer stations. The master of the ring executes the "Event Coordinator" task, controlling the whole sequence of management processes. A separate object module provides a software library to interface the individual subsystem elements, with features ranging from full module testing and control through to basic buffer management and system coordination.

7.2.1 The architecture of the front-end producers

The readout system of each branch is autonomous up to and including a central subdetector VMEbus crate. Each central subdetector crate contains a dedicated supervisory readout controller, a multi-event buffer (MEB), a VMEtaxi and any input drivers necessary to access the data (Figure 23). Consequently the design allows a particular subdetector to be decoupled from the rest of the system, during installation and test phases, and does not exclude the use of other busses for front-end digitization. Data are placed into the multi-event buffer over VMEbus and then extracted by the VMEtaxi either over VSB or locally, depending on whether the buffer uses a DPM or exploits the extension RAM of the VMEtaxi. The Event Coordinator management task runs on the master VMEtaxi of the ring and interacts with each readout controller via shared memory blocks. Before commencing data acquisition, the Event Coordinator is responsible for hardware recognition and initialization of the various branches. During acquisition, when it is free for event building, it searches the subdetector multi-event buffers for the next event. When all branches are ready with the same event number, the separate banks are transferred via the optical ring into a full-event buffering system. On completion, each corresponding buffer is released. A subdetector crate may also contain additional processors for data reduction, an interface to a subdetector monitoring computer and a subsystem trigger controller (STC). All STCs are interfaced to a dedicated central trigger controller which, in turn, coordinates the sequence of all hardware triggering levels together with obtaining information from the HERA machine [68]. Each STC provides signal ports for the component electronics and communicates with the readout through VMEbus read-write cycles and interrupts. A consequence of this is that the Event Coordinator needs no hardwired connection with the central trigger controller itself; all management sequences are handled by software, so providing a portable solution to any large multi-crate VMEbus system [67].

At the subdetector branch level, the software library provides multi-event-unit routines catering for initialization, accessing status information, requesting buffers and signalling when event data are ready for merging. Readout error codes are normally indicated as standard parameters, but messages can be sent through the system in standard ASCII format to describe branch-specific anomalies. Figure 23 also shows how the protocol has been transparently mapped from a main branch through to a sub-branch structure by incorporating standard VIC links to merge smaller subsystem components.

7.2.2 The management of full-event consumers

As the Event Coordinator VMEtaxi builds full-event records it simultaneously broadcasts them along its VMEbus to dual-ported memories associated with parallel sets of " full-event units ". Event tasks are performed by connecting the unit processors with their respective full- event buffer (FEB) memories via VSB in order to minimise any bandwidth overheads (Figure 24). Since the memory has read/write access from both the event task processors and the Event Coordinator, full-event units need not only be "consumers" of data. An event unit can also become a "producer" of data, "feeding-back" data into the system. Consumers are able to determine which data they wish to receive, i.e. directly built events from the front-end branches or data fed-back into the system from, for example, a parallel filter farm. All this is handled modularly by the software protocol and its associated library of routines.

Typical full-event tasks are local data-logging, event display, data monitoring and histogramming. A more sophisticated form of unit is the fourth level filter "farm" consisting of many MIPS series processor "nodes" working in parallel. Each node executes the same algorithm independently on different events. This provides not only a final level of triggering but a further stage of data processing and online reconstruction, used for online event display [69].

Event records are passed to the different nodes under control of 3 filter input controllers, which communicate with the Event Coordinator as normal consumer full-event tasks. When a node is free it signals this to its filter input task which waits for an event ready in its full-event buffer. The data are then extracted over the VSB and sent to the free node, releasing that particular buffer and enabling a further node to be serviced. When a node has finished processing an event it signals completion to the filter output controller (a "feedback" event task). The latter then decides whether to pass the event back into the central system managed by the Event Coordinator and frees the node buffer for further processing. As a full-event unit, the parallel filter can be easily enabled or disabled. The filter output controller in addition handles output to permanent mass storage.

Final event records are sent to an SGI Challenge computer some 3 km distant [70].

The data is transferred via 4Mbyte VIC8251F crate interconnects to FORCE VMEbus based UNIX single board computers with SBUS FDDI cards to a Netstar Gigarouter in the 3 km distant central DESY computer centre, from where the data is finally transferred via HIPPI to a SGI Challenge mainframe and via SCSI to an Ampex DST800 tape robot. In case of link failure a backup event task, that drives a storage device directly from VMEbus, can be enabled.

7.2.3 System supervision and operator-control

The H1 system is capable of running entirely in VMEbus; Figure 25 illustrates the composition of the central part where dedicated processors run dedicated tasks so that the net result is one of a conventional multi-tasking system. This leads to a natural division of tasks and processes. The primary tasks of data acquisition are performed by the subsystem readout controllers, event builders, filter control tasks and event units, all managed by the Event Coordinator. All fast processing and readout is done within VMEbus. An exhaustive software library caters for external control by providing a full set of routines for system configuration, testing, system-status and monitoring [67]. NuBus-based Macintosh computers take over the responsibility of program development and run-time operator-control. The System Supervisor Mac [71] initialises the readout and checks the status of all elements by initiating VMEbus tasks such as the Event Coordinator. To provide a central focus of monitoring it also communicates with event tasks and subsystems either through VMEbus or over Ethernet. Additionally it serves the network with status information for external monitoring. The graphics-based philosophy of the Mac ensures that the operator is presented with, arguably, one of the most human-interactive interfaces, available today, to a complex real-time system.

7.3 Observations and performance

The H1 data acquisition system was initially commissioned during cosmic-ray tests in Spring 1991 and then used to read out the complete detector during first HERA operation in Summer 1992. Its advanced preparation, and relatively stable operation, was in no small measure due to the choice of an open, industrial bus-standard and the incorporation of modern commercially available hardware and software where suitable. Of particular note has been the extensive use of the networking facilities in addition to the main data flow. The ability to control and monitor the system externally, even from the latest generation of portable note-book computers, has been extensively capitalised upon.

From the master VMEtaxi the system can manage up to 32 branches and 16 full-event units, including the parallel filter farm, providing status information and accepting control command sequences from the System Supervisor Macintosh. Presently events of up to 100Kbytes can be coordinated from 12 branches at rates up to 200 Hz using the Mark-2 VMEtaxi modules, depending on the filter farm rejection rate. The primary sources of dead time remain at the front-end, where future injections of greater processing and refined triggers are made more comfortable by working within an international bus standard.