This document provides a description of the features of the Sorter which is a Data Processing module of the Eurogam data acquisition system.

To simplify the discussions it is assumed that the following refers to the implementation for Daresbury Laboratory. It is possible that a further development may be necessary to support the complete array at Strasbourg.

Input will consist of blocks of event data from the Event Builder module.

The processing element will be based on the current offline Sort Engine and language as agreed by the Sorting Subcommittee.

Output will consist of blocks of event data for writing to the storage medium (tape).

Requirements

The Project Scientist Committee has asked for the system to be able to cope with a minimum of 200 KWords/second, with feedback required on the possibility of upgrading to 2 MWords/second.

Other requirements that have been assumed include ease of programming and production phase maintainability. All components suggested are readily available from stable and internationally known companies.

Background

The Sorter lies in the chain of event data flow between the Event Builder and Data Storage. Some of the decisions involve consideration of the Sorter and Event Builder together.

The Sorter for Daresbury Laboratory will have to be placed close to the GEC4190 acting as the tape server due to considerations concerning the connecting parallel interface.

The sorting language currently in use in the UK will be used to provide a user friendly interface for the physicist. As processing systems get more complicated with the use of multi-processing techniques, it was thought that user-written Fortran sorting programs would become more and more inappropriate. The advantage of a high level sorting language lies in its ease of use and computer-independence.

Components

It is proposed that the Sorter consist of boards contained in one VME crate. The current offline Sort Engine provides the fundamental design basis of using a VME crate and Motorola 68K series processors.

The crate will support an Ethernet link to the system Control and Setup network. This will be provided by the Motorola 147 cpu board which has an onboard Ethernet interface. See elsewhere for details of the network interface.

The interface between the Event Builder and Sorter will be a fibre optic point-to-point link. There are two versions readily available in the catalogues of Struck and CES. Both have maximum quoted rates of 6 Mbytes/second, and both have VSB interfaces. On first sight either product would be satisfactory and some further research will be necessary to make a choice.

The interface for tape storage at Daresbury will be a parallel link to the GEC. The board in current use will not be good enough for Eurogam purposes and it proposed to build a more suitable version. This will have a VSB interface to remove the transfer load from VMEbus. The exact proposals will be described elsewhere.

The data processing cpus have to be Motorola 68k series cpus to be compatible with current Sort Engine software. The Motorola 165 (25 or 33MHz 68040 with 4Mbytes memory) seems a good choice as a general processing engine with sufficient performance. It has been designed specifically for embedded controller functions. The onboard memory is ported to both VMEbus and VSBbus. This board is orderable now with production availability in Q3 1990. The 68040 has a quasi-risc design which provides high processing power whilst keeping the powerful instruction set of the 68k series.

The final component of the sort system is the spectrum memory. This memory will be VME accessible by

all processing cpus. The memory boards used will be standard ones available from several manufacturers. It is important that they be configurable for extended addressing mode only (A24 must be disabled). The system is proposed to have 64 Mbytes enabling 2D matrices to be updated online.

Data Rates

The input data rate is limited by the bandwidth of the optical fibre link from the Event Builder to a maximum of 6 Mbytes/second.

It is difficult to estimate the processing power required for sorting. One MVME165 may be sufficiently powerful to cope with the minimum requested 200 KWords/second. Since the system is modular and more cpus may be added with no change of software, there is an easy upgrade path.

However, other criteria affect the performance. The data output path is currently via the Struck 302 parallel interface into a GEC4190. This parallel interface is only accessible over VMEbus with a limitation of 16bit wide data. This will have to be replaced, or a severe bandwidth limitation will occur on VMEbus. It is proposed to build a more efficient interface accepting 32bit wide block transfers over VSBbus.

Spectrum updates will occur over VMEbus to globally accessible memory. Current experience places a limit approaching 1 Mupdates/second assuming there is no other use of VMEbus. This cannot be the case as spectrum reads for display purposes and local slicing will expect some reasonable response.

It should be noted that high fold events contain many combinations of doubles and triples. Hence the volume of spectrum updates will increase considerably over current levels. This is likely to become the major bottleneck to future expansions.

Modus Operandi

Data will arrive in the fibre optic controller and be transferred over VSBbus to one of several sorting processors. Data blocks to be output to Storage will be transferred over VSBbus again to the GEC interface controller. As in the Event Builder design, VSB is used to modularise the system and to split total bus bandwidth. As output to storage will necessarily be limited, then two passes over VSB should be allowable.

Thus with a 6 slot VSB backplane, two slots will be occupied by the Event Builder and Storage interfaces leaving upto four slots available for sorting processors.

This design leaves VMEbus free for spectrum access. Spectrum updates from the sorting processors will occupy most of the available VMEbus bandwidth. The Control and Setup network interface cpu will require spectrum access for read clear and slicing operations.

Software

A task running in the network interface cpu will accept control commands from the user and act on them as necessary. These will involve program halt/go/reset/flush and download commands. For a definition of the network interface software and the available command set, see other documents.

The software running in the fibre optic controller cpu will control the link and transfer data blocks only. This

will be an independent subsystem, accepting data blocks transferred from the Event Builder and initiating block transfers to one of several processors. It will receive acknowledgements from the processors, and send acknowledgements to the Event Builder. The code will be a fixed single task system, written in C or assembler as necessary.

The processing cpus for the Sorter will all execute the same code. This consists of a foundation layer allowing command and data transfer from the controller. This code will be essentially the same as running currently on the offline Sort Engine. Some changes will be necessary to accommodate the extra requirements of an online system.

On top of the foundation lies a block of code to process an event. This code will be generated as at present, from the output of the Sort compiler. It then passes through a cross-assembler and the result downloaded into each processor. The basic system will be the same as is currently in use in the offline version. The command set that the Sort compiler understands will be enhanced to accommodate the different trigger types and event formats of Eurogam, together with the agreed requirement for multiple storage streams output from the Sorter. There will be other changes to the sort language, but these will be the subject of another document.

It is proposed to provide online matrix slicing operations locally within the crate executed by remote user command. The code to execute the slicing operations will be part of the sort control task executing in the network interface cpu. The slicing operations will include horizontal, vertical, diagonal and polygonal window projections on 2D matrices.