Custom solution for a data readout architecture: a system level simulation.

A. Aloisio [1,2], F. Cevenini[1,2], D. della Volpe[1,2], L. Merola[1,2], P. Parascandolo[2]

[1] Università degli Studi di Napoli "Federico II", Dipartimento di Scienze Fisiche.
[2] INFN Sezione di Napoli.

Abstract

Behavioral simulations of data readout architecture based on VME and custom high speed buses show that it is suitable as data acquisition and event building system for high energy physics experiments.
This paper describes a reliable but simple auxiliary bus designed to afford asynchronous transactions up to 10 Mtransfers/s, sparse data scan operations and crate-level event building. An intercrate connection is also presented to accomplish system level event building and data concentration by means of synchronous transactions of rates up to 10 Mtranfers/s.
This architecture has been simulated using Verilog HDL. Preliminary performance estimates are presented and briefly discussed in view of system application in KLOE experiment at the DAΦNE Φ-factory in Frascati (Italy).

Introduction

In H.E.P. experiments, Data Acquisition systems use high performance parallel buses to gather data from the front-end electronics to event builder engine and on-line processors.

Software protocols and CPU driven operations may slow down the whole system even if state-of-the-art devices and communication channels are used.

We describe in this paper an architecture which relies on custom buses and hardware controllers in order to achieve high bandwidth without software penalties and CPU overheads.

System architecture

The proposed architecture is shown in Fig.1

Each chain contains a number of VME crates, which hosts a CPU board, acquisition modules and a Readout Controller (ROC).

Through VME backplane, the CPU can access both the Front-End modules and the ROC. This controller is the only master on a custom auxiliary bus (AUXbus) which connects all the Front-End modules.

The structure and the protocol of the AUXbus are optimized to meet the requirements of high speed data transfer. In our scheme the VME CPU initializes the whole system, runs diagnostic tests and, if necessary, samples events from time to time for performance monitoring of the system itself.

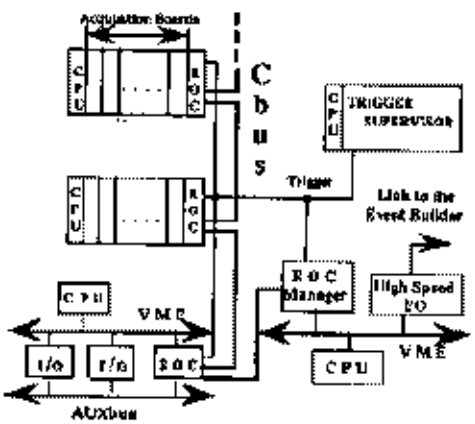

Fig. 1 - Read out Architecture

All these tasks may be afforded by a low-medium performance CPU in a standard software environment.

All the ROCs of a chain are connected through a single master cable-bus, called Chain-bus (Cbus), which is driven by a ROC Manager.

This last module is housed in a VME crate together with a high performance CPU and a high speed I/O board used to transfer data to the event builder.

Protocols overview

Each acquisition board is designed to respond to a trigger number request sending all the data associated to it onto the AUXbus. Acquisition results are automatically associated to the channel number inside the module.

The ROC drives the readout cycle on the AUXbus in two steps: broadcast and transfer.

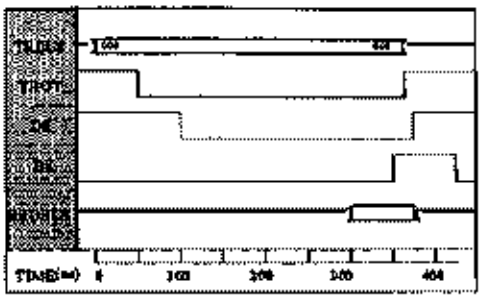

Fig. 2 - Broadcast cycle

During the broadcast cycle (Fig. 2), the ROC puts on the TRbus lines a trigger number forwarded asynchronously to the whole system by the trigger supervisor of the experiment. The boards with data associated to that trigger number assert a line on the BRDbus. The slowest board then validate this bus with the BK line.

As a result, the ROC records the BRDbus pattern by means of a single transaction synchronous to the BK assertion.

The broadcast cycle then proceeds with the identification of the modules to be read and the preparation of the table containing the addresses (sparse data scan).

In the following transfer cycle (Fig.3) the ROC addresses successively the modules to be read. Data transfer comes in blocks for each module using an asynchronous VME-like protocol.

An EOB line is used to flag in hardware the end of a block data transfer. It allows handling indefinite long block operations, overcoming the VME limitations. Data cycle pipelining is also provided to improve data transfer rates.

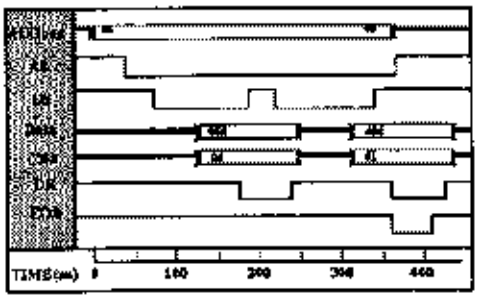

Fig 3 - Transfer cycle

Into the ROCs, data are organized in frames starting with a trigger number. Data words collected through the AUXbus from the acquisition modules follow. The frame is then terminated with a control word (End-Of-Frame). The frames are queued into a dual port FIFO buffer, accessible from the intercrate link.

The ROC Manager controls a chain of ROCs through the Cbus in a token-ring fashion.

Its readout cycle starts by presenting and validating a trigger number on the Cbus lines. Then a token is sent onto the bus.

The ROC nearest to the master catches the token and then start the synchronous transmission of the data associated to the given trigger number. The end of the data is flagged with the EOF control word. The token is then passed to the next ROC in the daisy-chain.

This procedure is repeated until all the ROCs of the chain have sent either the data related to the processed trigger number or an empty frame - if no data are available.

Traffic on AUXbuses and Cbus are fully decoupled by means of FIFO buffers to afford possible large change of trigger rate.

The data frame inside the ROC Manager is similar to that inside the ROC but also contains the crate address.

Simulations

Architecture and protocols presented above have been modeled and simulated at behavioral level using Verilog-HDL [1].

The timing of the transactions on both AUXbus and Cbus comes from the hypothesis of using off-the-shelf TTL components and from conservative criteria.

In our simulations, we considered the data acquisition layout proposed for the KLOE experiment [2] at the DAΦNE Φ-Factory in Frascati (Italy). The KLOE DAQ system [3] will rely on chains of 8 crates. One crate houses 16 acquisition boards with 32 channels each.

The simulation of the activities on AUXbus was done for an increasing number of words randomly spread into the crate.

Time the ROC needs to complete a readout cycle increases linearly with the amount of data to be transferred (Fig. 4).

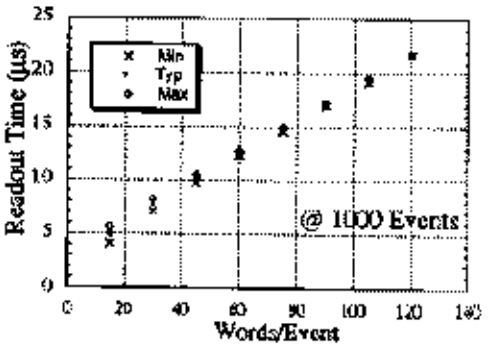

Fig. 4 - AUXbus simulated performances

The analysis of the whole chain (8 crates) has been carried out from 120 up to 480 words/event randomly spread into the system. For each occupancy 1000 events have been simulated.

The typical acquisition time at 250 words/event is about 70 µs as shown in Fig. 5.

Conclusions

The introduction of auxiliary buses in a DAQ scheme relaxes the requirements on the CPU and makes it possible to use VME as an additional channel for monitoring, diagnostics and on-line event sampling.

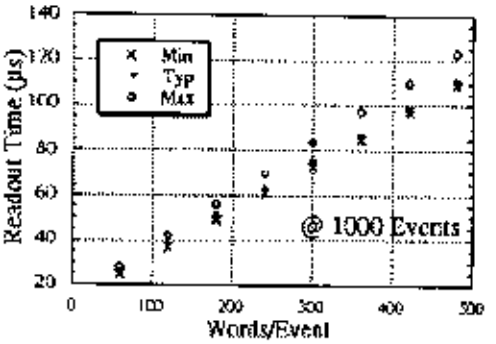

Fig. 5 - Cbus simulated performances

Data transmission in our proposal is driven by hardware processors through very reliable protocols, developed *ad hoc* to meet the requirements of a typical H.E.P. experiment.

A hardware implementation of the described architecture is under development in Naples in view of application in the KLOE DAQ system. The AUXbus controller is based on a XC3164 XILINX Gate Array [4] while the Cbus interface fits in a MACH210 AMD device [5].

The AUXbus (64 lines) shares the VME/J2 backplane, the Cbus uses a 50 twisted pairs cable. Prototypes of both VME/AUXbus custom backplanes and Cbus cable segments have been already designed and successfully tested.

In KLOE experiment, at 10 KHz expected trigger rate the DAQ system will be able to afford a 15 MByte/s data traffic per chain - assuming 32-bit wide words and a maximum occupancy of 400 words/event.

References

[1] Cadence Design System Inc., VERILOG-XL Reference Manual, Version 1.6, 1991.

[2] The KLOE Collaboration, " KLOE A general purpose detector for DAΦNE", LNF - 92/019 (IR).

[3] A. Aloisio *et al.* , "Fast readout system for KLOE", KLOE Note # 88, Jan. 94.

[4] XILINX, The Programmable Gate Array Data Book, 1993.

[5] Advanced Micro Devices, MACH 1 and 2 Family Data Book, 1993.