EMBEDDED CAMAC CONTROLLER: HARDWARE/SOFTWARE CO-OPTIMIZATION FOR HIGH THROUGHPUT

P. M. Nair, K. Jha, P. Sridharan, A. Behere, M.P. Diwakar, C.K. Pithawa,
Electronics Division, BARC, Mumbai, India

Abstract

Advances in technology have resulted in availability of low-power, low form-factor embedded PC based modules. The Embedded CAMAC Controller (ECC) is designed with ETX (Embedded Technology eXtended) standard Single Board Computer (SBC) having PC architecture with Ethernet connectivity. The paper highlights the software and hardware design optimizations to meet high throughput requirements of multi-parameter experiments and scan mode accelerator control applications. The QNX based software is designed for high throughput by adopting design strategies like multi-threaded architecture, interrupt-driven data transfer, buffer pool for burst data, zero memory copy, lockless primitives and batched event data transfer to host. The data buffer and all control logic for CAMAC cycle sequencing for LIST mode is implemented entirely in hardware in Field Programmable Gate Array (FPGA). Through this design, sustained throughput of 1.5MBps has been achieved. Also, the host connectivity through Ethernet link enables support for multi-crate configuration, thus providing scalability. The ECC has been installed for accelerator control at FOTIA BARC, Pelletron and LINAC-Pelletron, TIFR and for multi-parameter experiments at NPD, BARC.

INTRODUCTION

PC compatible architectures have become a popular choice for embedded systems because of easy availability of low powered, low form-factor embedded PC based modules. Also, a large base of knowledge and resources exist for these architectures. An Embedded PC meets special needs of mechanical design, power consumption, product lifetime, customized software and professional support. All these readily available features of Embedded PC-based technology coupled with optimum hardware and software design allowed to develop an Embedded CAMAC Controller (ECC) for high throughput requirements of multi-parameter experiments. The paper discusses various design optimizations in the hardware and software of ECC.

EMBEDDED CAMAC CONTROLLER

Embedded CAMAC Controller [1] has been designed to cater to the needs of process control systems and high throughput of large nuclear physics experiments. Control applications need more number of physically distributed crates with regular scanning of all the parameters, the control being with a centralized computer, whereas nuclear physics experiments need a high throughput with a large number of parameters in one or more crates.

The Single Board Computer (SBC) of ECC interfaces with PCI-CAMAC Interface logic through PCI controller (see Fig. 1). The ECC allows interaction with remote host by means of standard Ethernet services, such as TCP socket based communication protocol. The processor runs a version of QNX optimized for low memory footprint. The FPGA holds List buffer, Data buffer and all control logic for CAMAC cycle sequencing. Three modes of operation have been supported:

- ***Single Cycle mode:*** In this mode, single ECC command is executed in every cycle.
- ***Scan mode:*** In scan mode of operation, a list of CAMAC commands is sent to the ECC and is executed at regular intervals, and is suitable for control applications.
- ***List mode:*** The list mode is optimized for multi-parameter experiments. In this mode a list of CAMAC commands is stored in Field Programmable Gate Array (FPGA) which is executed on every Look At Me (LAM) from signal acquisition module.

SALIENT FEATURES: HARDWARE

- List sequencing mode with 1.1 µs CAMAC cycle for multi-parameter experiments
- Scan mode with a period of 50msecs for control applications
- Trigger source: LAM / External event / Timer
- Ethernet connectivity for remote and multi-crate configurations.
- Built-in test features with CAMAC data way display

List Mode Optimization

ECC operates in List mode for multi-parameter experiments. There are three lists of NAF commands in multi-parameter mode of acquisition; EVENT_LIST, INI_LIST and SCALER_LIST. The INI_LIST is executed once at start of acquisition to initialize the ECC. EVENT_LIST of up to 256 CAMAC commands is executed upon generation of LAM and data is stored in FIFO buffer. In List mode processing, once the setup is complete and acquisition is started, the host does not intervene other than to stop the acquisition. In this mode, all acquired event data are stored in communication

ISBN 978-3-95450-124-3

buffers and are send to host PC in batches for achieving better throughput. The size of communication buffer is user programmable so that throughput can be optimised with reference to the number of parameters at experiment time. At the end of every batch of event data which are to be send to the host, SCALER_LIST is executed. The acquired batch of events and the scalar counts are sent to the host PC.

The list processing is completely implemented in FPGA hardware to achieve optimum CAMAC throughput of 1.1µs with only 100ns overhead per readout. Optimization in hardware is done by maintaining two event buffers in FPGA. When one buffer is full, event data is acquired in the other buffer and first data buffer is transferred to PC memory.

SALIENT FEATURES: SOFTWARE

The application software for ECC has been designed to achieve high throughput for a burst nature of data for nuclear physics experiments and the stringent periodic scan requirements of control applications. ECC application runs embedded QNX based software with a low memory footprint designed for high throughput by adopting design strategies like multi-threaded architecture, interrupt-driven data transfer, buffer pool for burst data, zero memory copy, lockless primitives and batched event data transfer to host.

Architecture

Multi-threaded programming architecture was selected to achieve maximum CPU utilization and to make the application responsive to the user. Acquisition of experiment data and transmission of data to host are done concurrently. Through proper assignment of thread priority, data transmission to the host is interspersed while waiting for CAMAC readout. Acquisition is assigned highest priority and sufficient buffers with a provision to expand/contract the buffer pool have been provided to help absorb event burst.

Communication module for data transmission to host is designed based on Reactor pattern which allowed simple coarse-grain concurrency without adding complexity of multiple threads to the system. This pattern also allowed prompt data transmission of List mode data to host by use of common de-multiplexer (select) to notify communication events immediately while remaining responsive to user commands.

Object Pool

Event data generated during nuclear physics experiments in bursts with an average rate of about 1K events per second. Hence sufficient buffering is provided to handle this burst data. Application, on Start-up, pre-allocates all anticipated memory requirements on memory pool. Allocation of memory on pool reduced the need for dynamic memory allocation during application run thereby optimizing on memory allocation delays. The buffer pool is expanded as needed to maintain the specified number of free buffers in the pool. The buffer pool contracts as necessary, to prevent the pool from consuming system resources permanently as the result of usage peaks. Application software provides a feature to batch the acquired event data by providing adequate communication buffers. This strategy reduces communication overhead. Number of events that can be batched is programmable from the host PC.

Interrupt-Driven Data Transfer

To achieve maximum throughput in list mode, software has implemented Interrupt-Driven Data transfer. This offloaded the CPU and resulted in low overhead, low latency and better performance.

Lockless Primitives

The traditional approach to multi-threaded programming is to use locks to synchronize access to shared resources. Synchronization primitive such as mutex, semaphore and critical section are kernel objects and hence kernel switch times and also associated bookkeeping add lot of overhead to acquire and release lock. Application software was, therefore, optimized using lock-free data structures such as Lockless Queues. This avoided lock acquisition and associated overheads. In cases where it is impossible to eliminate the need for locks, application software used spin-locks based on atomic operations such as compare and swap (CAS).

Zero Memory Copy

A zero copy message transfer mechanism was incorporated which completely eliminated the overheads involved in copying or cloning. The same pre-allocated memory object was used throughout from data acquisition to data transmission. This relieved the CPU of the task of copying data from one memory area to another and improved performance by saving processing power and memory use while sending acquired data to the host.

Objected Oriented Design

Adoption of Object oriented design for the application software kept the design modular, reusable and composable. The use of well proven architectural and design patterns [2] and generics has resulted in the software being adaptable and extendable.

Selection of OS

QNX OS with its robust scalable microkernel architecture and bounded response times ensured deterministic behaviour to meet high throughput requirements. The embedded software also needed to be optimized for low memory footprint which could be achieved using QNX. QNX with its microkernel architecture allows embedded applications to be highly configurable. The configured footprint for ECC consists of ECC application software, QNX microkernel; TCP/IP stack, drivers and QNX file system.

ISBN 978-3-95450-124-3

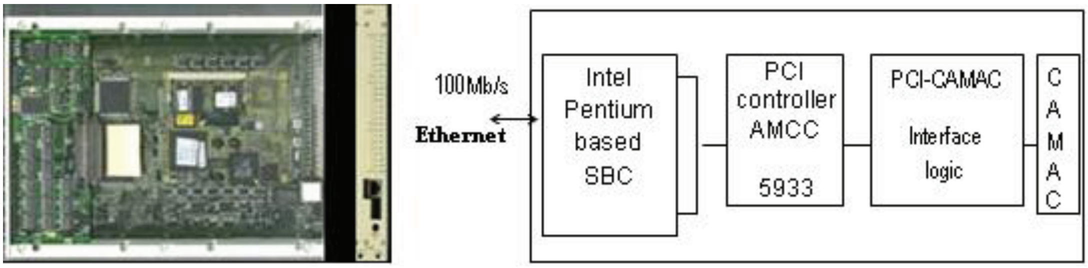

Figure 1: The Embedded CAMAC Controller with its block Schematic.

APPLICATIONS

ECC have been deployed for accelerator control at FOTIA BARC, Pelletron and LINAC-Pelletron, TIFR and for multi-parameter experiments at NPD, BARC, TIFR and SINP.

PERFORMANCE

The major improvement for accelerator based multi-parameter experiments is in the CAMAC readout, which has been reduced to 1.1µs. With optimum embedded software using QNX operating system, sustained throughput in excess of 1.5MBPS has been observed in the lab irrespective of number of parameters and event rate. This is a vast improvement over earlier CAMAC controllers developed in-house with capability of 250KB/s. ECC also meets requirement of accelerator beam line control application. A scan time of 50 ms has been achieved for 280 parameters in scan mode operation.

CONCLUSION

PC compatible architectures coupled with optimization done in software and hardware design has helped in achieving high throughput for CAMAC based data acquisition systems, which are widely used in accelerator based nuclear physics experiments.

REFERENCES

[1] K. Jha et al., “Ethernet CAMAC crate controller for data acquisition and control,” in DAE Symposium of Nuclear Physics,2008.

[2] Erich Gamma, Richard Helm, Ralph Johnson and JohnVlissides, “Design Patterns: Elements of Reusable Object-Oriented Software”, Addison-Wesley, 1995.

ISBN 978-3-95450-124-3