# Use of Network Processors for Data Multiplexing and Data Merging

J.-P. Dufey, B. Jost and N. Neufeld

*Abstract*—**Network Processors are an emerging technology targeted for high-end switch manufacturers to provide programmable and flexible means for implementing network protocols, quality of service algorithms etc.**

**In this paper we describe investigations performed in the framework of the LHCb Data Acquisition project on how these devices can be used to solve problems of data multiplexing and data merging. We present results of performance simulations and also a possible implementation of a general-purpose board.**

## I. Introduction

LHCb is one of the four experiments currently under construction at Cern's LHC accelerator. It is a special purpose experiment targeted at precision measurements of the CP violation mechanisms in the B-meson decay. The detector is described elsewhere [1]. The Trigger and Data Acquisition (DAQ) architecture has been described in previous publications [2]. One of the functional components of the LHCb DAQ system consists of merging and concatenating data fragments from several inputs into one fragment at the output. A first prototype of this functionality has been designed using conventional components (FPGAs, FIFOs memories, etc) [3].

We present here a novel approach to the same problem based on Network Processors (NP). NPs are an emerging technology targeted originally at the switch market to provide the input stage of high-end switches with a flexible (programmable) means to implement such functionalities as quality of service, traffic shaping, etc. It is expected that NPs will play an important role in the networking industry because of their flexibility to cope with changing requirements and changing network protocols [4],[5], which makes them superior to the currently used ASIC approach in the switch industry.

We investigated the possibilities of using NPs for solving the problem of merging the data arriving on several input links of the LHCb Readout Unit to one fragment at the output.

## II. Introduction to Network Processors

Network processors are collections of RISC processor cores, usually multithreaded in hardware, that are specialized for analyzing and altering network frames arriving at the input or leaving the NP at the output and/or to prepare the frame for the transfer to a switching fabric. For these purposes the RISC cores are supported by sets of coprocessors that implement special functions, such as tree-lookups to resolve output ports from Ethernet addresses or IP addresses, checksum calculations or implementing rate limitation policies. In our studies we focused the efforts on the IBM NP4GS3(B) Network processor [5]. Figure 1 shows the architecture of the chip.

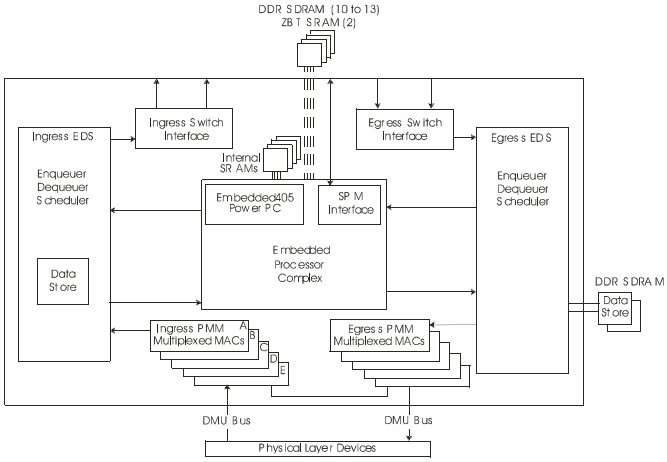


Figure 1 Overall architecture of the IBM NP4GS3 Network Processor. Shown inside the large solid rectangle are the components that are integrated on-chip, whereas the other components are external.

The main functional components are

- Integrated Media Access Controllers (MACs) for 10/100 Mbit and Gb Ethernet or POS (Packet over SONET)
- 128 kB of on-chip Ingress Buffer for receiving the data from the MACs
- Embedded Processor Complex (EPC) containing 8 Dyadic Protocol Processing Units (DPPUs). Each DPPU contains 2 processors sharing a set of coprocessors with 2 hardware threads
- Interfaces to a switch bus, DASL (Data Aligned Synchronous Link) that allows connecting several NPs to a switch chip. The maximum speed of the DASL is 4 Gb/s (Full Duplex, i.e. 8 Gb/s in total)
- Up to 64 MB of, externally implemented, Egress Buffer for storing the data before sending them to the output port. Actually this buffer space is organized as 2 times 64 MB with common memory

Manuscript received June 10, 2001.

Author's Address:
J.-P. Dufey, B. Jost and N. Neufeld, Cern / EP Division, CH-1211 Geneva 23, Switzerland

management in order to double the access bandwidth

- An embedded Power PC core allows for configuration, controls and monitoring of the operation of the Network Processor core, but can also receive frames from the EPC in exceptional cases.

The main processing units (EPC and PowerPC) run at a clock frequency of 133 MHz. The total CPU power represented by the EPC amounts to 2100 MIPS. With this CPU power available the chip is designed to implement wire-speed switching at any frame size.

With the chip comes a very elaborate software development environment consisting of

- An assembler
- A source-level simulator/debugger
- A profiler, simulating the chip and the software running on it with clock-cycle precision taking into account shared resources, such as memory buses etc.

There also exists a hardware reference kit that allows connecting up to two NPs to each other or to a switching chip for applications testing and performance measurements. This reference kit will also allow verifying the profiling results using real hardware.

## III. Application to Data Multiplexing and Data Merging

In the rest of this paper we will describe an alternative application of the NP in the field of data multiplexing and data aggregation, i.e. the assembly of data fragments arriving at two or more input ports to one single data fragment to be sent to an output port.

The requirements for data multiplexing in the LHCb DAQ system are

- The input rate of fragments per link <100 kHz
- Number of input links 4-16 depending on location within the readout system
- Input fragment size compatible with Gb speed at output for the aggregated output fragment, i.e. the aggregation of the input fragments to an output fragment should not exceed ~1 Gb/s at the nominal output rate. [1]

We have developed, debugged and profiled two versions of software, namely small fragments at input (1-100 Bytes) and larger fragments (100-500 Bytes). The two algorithms differ in the way they make use of the buffer storage provided by the network processor. While for small fragment sizes it is advantageous to do the merging of the data in the ingress buffer (see Figure 1) and by copying the data, it is more performing for large fragment sizes to apply the merging algorithm on the data at the output (Egress) buffer. In the latter approach, to a large extent, only pointers have to be manipulated and not all the data have to be touched. The simulated performance of both algorithms is shown in Figure 2 and Figure 3.

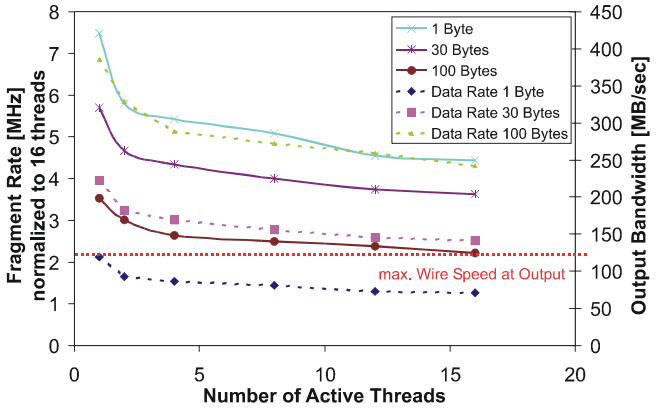


Figure 2 Simulated performance of the algorithm optimized for small incoming fragments (1-100 Bytes of average payload). The solid lines show the performance in terms of rate of incoming fragments that can be handled (left scale), whereas the dashed lines show the performance in data rate at the output (right scale). A 4:1 data-merging ratio is assumed. It should be noted that the maximum output rate for a Gb Ethernet port is 125 MB/s which is represented by the dashed horizontal line.

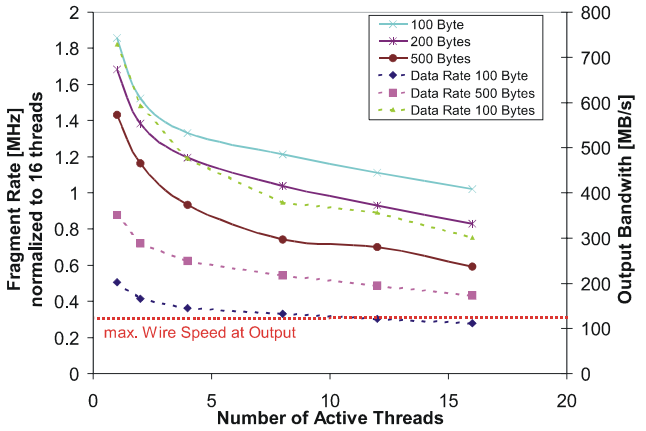


Figure 3 Simulated performance of the algorithm optimized for large incoming fragments (100-500 Bytes of average payload). The solid lines show the performance in terms of rate of incoming fragments that can be handled (left scale), whereas the dashed lines show the performance in data rate at the output (right scale). A 4:1 data-merging ratio is assumed for the determination of the output rate. It should be noted that the maximum output rate for a Gb Ethernet port is 125 MB/s which is represented by the dashed horizontal line.

Due to a limitation in the current version of the simulator only a maximum of 16 threads can be simulated. We have extrapolated the performance for numbers of active threads smaller than 16 to 16 treads, instead of the 32 threads available on the real hardware, to be able to assess the scaling behavior. As can be seen from the figures the performance does not scale perfectly when moving to larger numbers of threads. This is to be expected, since the event-building process entails accesses to non-shareable resources of the

[1] There is another application with similar functionality in LHCb, namely the Level-1 trigger. The basic difference to the DAQ application is that the input fragment rate is ~1.1 MHz with very small fragment sizes, of the order of 30-50 Bytes. This application is however not directly part of the studies presented here.

chip, like e.g. memory buses or data structures describing the event-building process. As can be seen from Figure 4, which shows the processing time per fragment normalized to that for 16 threads as a function of the number of threads, the losses due to non-scaling effects are of the order of a factor of two when going from 1 to 16 threads. In addition the non-scaling effects seem to flatten off for larger numbers of threads. This result is encouraging taking into account that the real hardware will have 32 threads available to the algorithm.

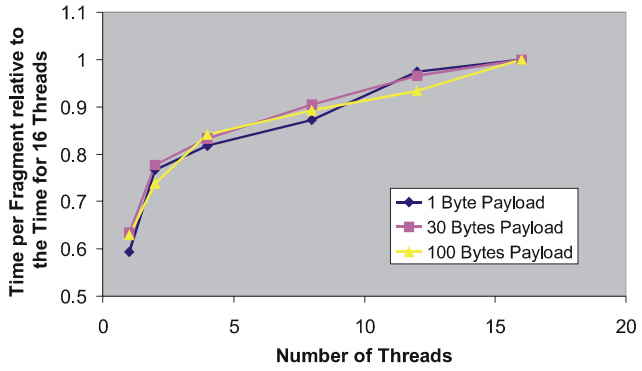


Figure 4 Measurement of the scaling behavior of the performance for small fragment sizes. Plotted is the processing time per incoming fragment normalized to the time taken by 16 threads as a function of the number of threads activated for various average payloads.

## IV. Board-Level Implementation

In this section we outline first ideas on an implementation of a board using the NPs suitable for the LHCb DAQ System. The idea is to implement the NP and its accompanying memory infrastructure on a mezzanine card forming a functional unit. The interfaces of this mezzanine card to the external world consist of the DASL connection, the PCI bus for configuring and controls and the DMU (Data Mover Unit) buses to connect to the physical network layer components. A throttle signal, as required by the LHCb data-flow specifications to signal high buffer occupancy, is also fed out of the mezzanine card.

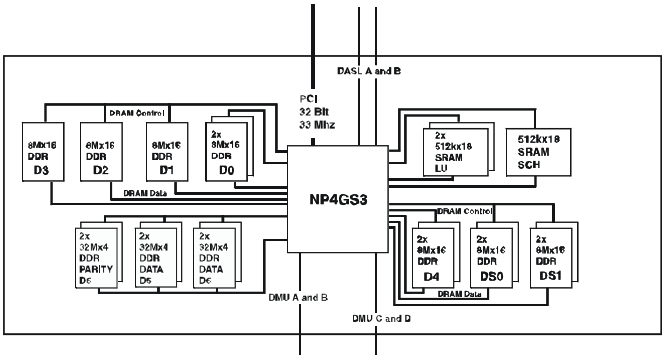


Figure 5 Block-Diagram of a mezzanine card containing all necessary memory components for the Network Processor. The only external interfaces needed are the PCI bus for configuration and monitoring, the DASL bus for connecting to another mezzanine card or a switch chip and the connections to the physical layer of the networking equipment plus power and clock.

A block diagram of the mezzanine card is shown in Figure 5 and its embedding on a mother board in Figure 6. Besides two mezzanine cards containing NPs the mother board also houses an interface to the Experiment Controls System (ECS) of LHCb in form of a Credit-Card PC (see [7]) and power converters and clock generators for the NP mezzanines. Of course any other implementation of an interface to a controls system, like e.g. a VME slave interface could easily be accommodated. This modularity makes the board very flexible in that the hardware implements basically just 8 Gbit Ethernet ports that are completely connected and it is the software running on the NPs that determines the functionality of the board.

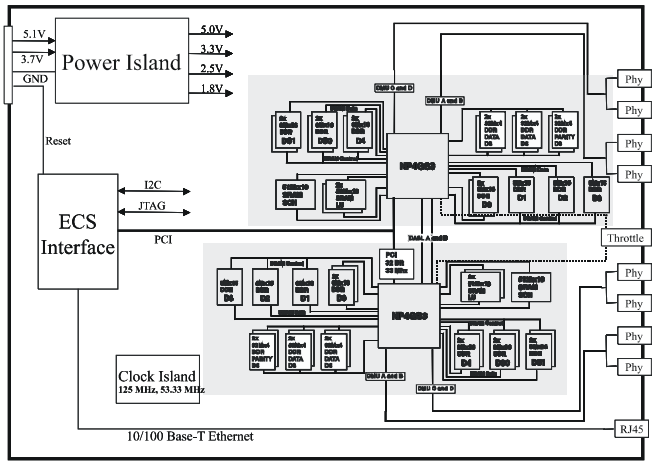


Figure 6 Block Diagram of a board containing two mezzanine cards as depicted in Figure 5. The auxiliary components like power and clocks are provided for on the carrier board. The ECS interface would be compatible with the LHCb standard controls interface (Credit-Card PC [7]) and is connected directly to the controls network.

Possible applications in the LHCb Data-Flow system could be

- 8-port switch as a building block for the LHCb Readout Network (see [8])
- n:1 multiplexer/event-builder as required in the LHCb DAQ at the Levels of the Front-End Multiplexers and Readout Units
- 4:4 Event-Builder module to be applied at the last stage of the Readout Network to build complete events and send them to the sub-farm controllers for further dispatch to the CPU farm. This application is an alternative to the use of "smart NICs" as described in [9], should their availability not be guaranteed or their cost be prohibitive.

The module outlined above could handle all functions in the LHCb dataflow system. Obviously this would be a tremendous advantage from the maintenance and operation point of view.

## V. Summary

Network Processors are an emerging technology in the network industry targeted at applications for manipulation and alteration of packets and frames in high-end switches. In this paper we present an application of network processors in

the domain of data merging and data multiplexing in the framework of the LHCb Data Acquisition system.

Code for this application has been written and debugged for the IBM NP4GS3(B) Network Processor optimized for small and large input fragment sizes. Simulations show, that for all practical purposes event-building speeds exceeding wire-speed at the output port can be achieved.

An implementation of a board implementing Network Processors for the LHCb DAQ system has been sketched, that would represent a unique module for the entire LHCb dataflow system.

## VI. Acknowledgment

We are indebted to P. Paerdaems of IBM Switzerland for his help in acquiring the IBM NP software Toolkit.

We also acknowledge many valuable discussions with our colleagues in the LHCb collaboration.

## VII. References

[1] LHCb Technical Proposal
[2] J.-P. Dufey et al.," The LHCb trigger and data acquisition system", IEEE Trans. Nucl. Sci. : 47 (2000) no.2, pp.86-90
[3] H. Muller et al., "The Readout Unit for High Rate Applications in the LHCb Experiment", These proceedings
[4] Linley Gwennap, "Net processor makers race toward 10-Gbit/s goal", EETimes, http://www.eetimes.com/story/OEG20000619S0011 19. June 2000.
[5] W. Bux, W. E. Denzel, T. Engbersen, A. Herkersdorf and R. P. Luijten, "Technologies and Building Blocks for Fast Packet Forwarding", IEEE Comm. Mag., Jan 2001, pp. 70-77
[6] Documentation for the NP4GS3 can be found on the IBM web Pages under http://www-3.ibm.com/chips/techlib/techlib.nsf/products/IBM_PowerNP_NP4GS3
[7] R. Beneyton et al., " Controlling Front-End Electronics Boards using Commercial Solutions", These Proceedings.
[8] J.-P. Dufey et al. The LHCb Event-Building Strategy", These Proceedings
[9] J.-P. Dufey et al., "Use of smart NICs for Event-Building", proceedings of the NSS 2000 conference, Lyon, October 2000.