ORION GATEWAY DESIGN FOR FEEDBACK CONTROLS CONNECTIVITY*

L. Doolittle, A. Ratti, C. Serrano, A. Vaccaro, LBNL, Berkeley, CA 94720, USA

Abstract

The Optical Redundant I/O Network (ORION) is a hardware-based fast communication system for feedback controls to be implemented at NSLS-II. It controls latency by eliminating traditional computers from the communication design. Redundant communication paths give basic single-point fault tolerance. This paper describes the peripheral infrastructure for data exchange between feedback control systems and diagnostics and the communication backbone.

INTRODUCTION

The Optical Redundant I/O Network (ORION) is under development to meet the communication and computation needs of storage ring fast communications, using distributed hardware and fiber optic communication channels [1]. By avoiding the use of computers in the real-time data path, and using a common clock frequency for all digital hardware, latency can be kept short and fully predictable.

Its baseline configuration is aligned with the needs of fast orbit feedback for the NSLS-II project [2]: 30 stations each receiving data from 8 two-axis BPMs and controlling 3 two-axis corrector magnets [3]. It is plausible to tie bunch-by-bunch feedback systems into ORION, providing integrated "woofer" functionality. A further tie to LLRF controls would give a network with full 3-dimensional real-time characterization and control of the stored beam.

COMMUNICATION DESIGN

Commodity fiber-optic communication technology is assumed, with a ring topology that parallels the physical construction of the storage ring, as shown in Fig. 1. By using a common clock for all nodes, jitter is removed at its foundation, and traditional restrictions on message lengths are eliminated. Each node is programmed within a single FPGA; within each node, all data processing and real-time communications happens in a single clock domain.

The essential idea behind ORION's message design is to configure one node's payload (plus CRC) length equal to the propagation time through that node. At any time, N messages from each of N nodes are in transit. Once a payload has circulated through the network once, it can be replaced with the next payload from the same source node. The latency through a node, and therefore the payload size, has a fixed minimum determined by serdes latency, fiber delay, and the pipeline delay introduced by the implementation in the FPGA fabric. That minimum is then padded with a FIFO to a suitable payload+CRC size. Table 1 shows those amounts for the NSLS-II baseline, when clocked at 125 MHz.

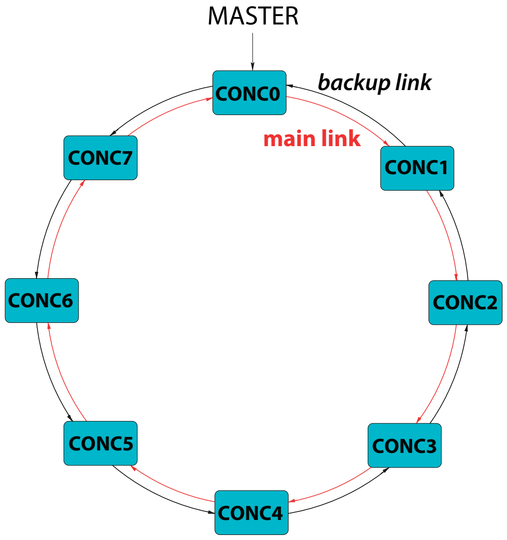

Figure 1: ORION ring network in normal operation mode. For simplicity only 8 nodes are shown. Each node is assigned an ID number according to its location relative to the master, and the sense of data circulation.

Table 1: Latencies for the NSLS-II Baseline when Clocked at 125 MHz

	time (ns)	cycles
serdes	176	22
fiber	137	17
fabric pipeline	56	7
FIFO	288	36
payload	640	80
CRC	16	2

We use the term "spin" to refer to one such circulation of messages around the ring. Thus, each node inserts one payload onto the network each spin. At least two spins are intended for communicating BPM data, and currently one additional spin is dedicated for internal network housekeeping.

ORION's key design feature is the fault-tolerance. Any single fiber (pair) or node can fail, and the remaining com-

* Work performed under auspices of the U.S. Department Of Energy under contract No. DE-AC02-98CH10886 with Brookhaven Science Associates, LLC.

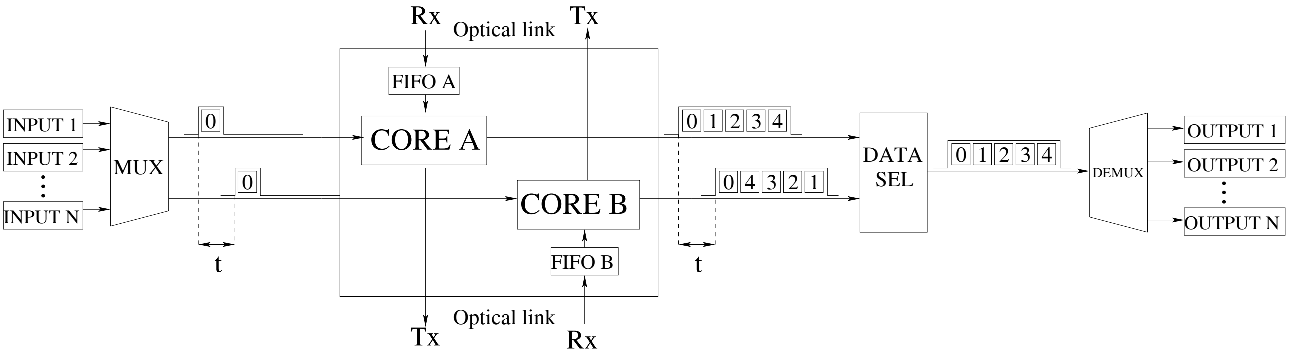

Figure 2: Structure of the local input and output interfaces of the communication node.

ponents continue to operate with no degradation of performance. As will be seen below, redundancy and fault-tolerance have deep implications for the application interfaces.

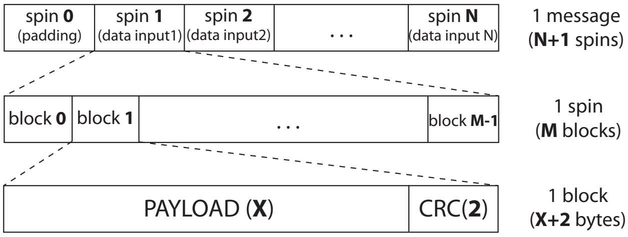

Figure 3: Terminology for the data structure in ORION.

COMMUNICATION ARCHITECTURE

A communication node exchanges data with other nodes using the redundant optical fiber links. It inserts data into the ring using its local data input interface, and retrieves data from the ring using the local output interface. Figure 2 shows the communication structure of one communication node (more details on these well-defined interfaces are given later in this section).

Two identical cores exchange data with the other nodes in the two directions (a and b). The same input data is multiplexed and inserted in both directions. Both cores provide all the data from all the nodes in case of no failure, and the only function of the data selection module shown in Fig. 2 is to forward the data from one of the links to the output demultiplexer. However, when a fault is detected in the system, data flowing in both links will be needed to recover all the data. The validate signal shown in Fig. 5 is used by the data selector to find valid blocks of data for the output multiplexer.

An address bus is provided to the output demultiplexer to signal the origin and type of the data being transmitted on the data bus. Figure 3 shows the data structure in ORION. One message is divided into as many spins as types of data are transmitted, for example providing each information for a feedback system in the ring. Each spin is divided into blocks, and there are as many blocks as nodes in the ring. Thus the address bus provides spin and block number for the output demultiplexer to identify type and origin of the data.

Data Input Interface

Spins are numbered, and an input multiplexer selects the data source for each spin in turn. The input interface for the data for one spin is therefore very simple, as shown in the timing diagram in Fig. 4. In Figures 4 and 5, X represents don't care, and D represents the series of data octets making up one payload. All nodes and all spins provide the reset pulse at the same time (plus or minus one clock cycle). The time between reset and gate depends only on the spin number of the data source.

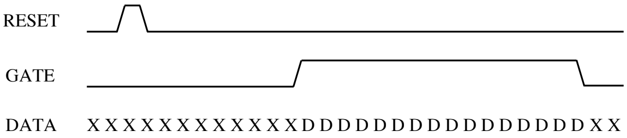

Figure 4: Input timing diagram.

The only complication to the above interface is that there are two such ports, one for each redundant message circulation direction, which we label a and b. The reset pulse is common to the two directions. While in principle the two gates should happen in synchrony, misadjusted FIFO lengths will lead to a time skew. A robust implementation will function correctly even in the presence of such skew.

Our test and example input modules tend to follow the strategy of filling two memories when triggered by the sync pulse. Then each memory can be read out independently as commanded by the a and b gate signals. Other strategies are possible.

One possible source of BPM data is a chain of Libera BPM modules, using their dedicated Ethernet daisy-chain link. We have demonstrated reception of these packets, and can convert their data to an ORION message payload.

Data Output Interface

The output interface is also dual, and very similar to the input interface, with the addition of an address word and post-validate. The output from both a and b directions is shown in Fig. 5.

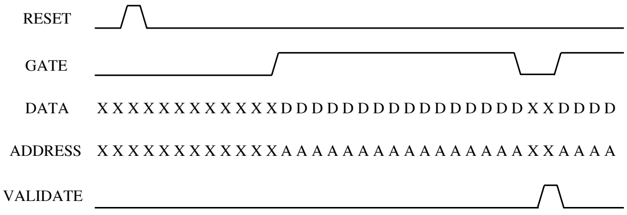
RESET
GATE
DATA
X X X X X X X X X X X X D D D D D D D D D D D D D D D D X X D D D D
ADDRESS
X X X X X X X X X X X X A A A A A A A A A A A A A A A A X X A A A A
VALIDATE

Figure 5: Output timing diagram.

The post-validation pulse is derived from the CRC check done on ORION messages. While an output module may store the payload and even stage computations, it cannot commit the result until and unless the validate pulse occurs.

The address word gives both the spin number and the origin node number for this payload. In case of a failed link, ORION's design guarantees that each node still provides data from all nodes to its two output ports, although in that case the data is split between the a and b ports.

Ethernet Access

ORION can not stand in isolation; it needs some connection to commodity hardware and legacy software. To meet this need, we have implemented a hardware (FPGA fabric) Ethernet/IP/UDP stack, that connects a GMII interface (connection to GigE PHY) to an on-chip register read/write local bus. That bus, in turn, has access to the on-chip controls and status readout of ORION itself, as well as application controls and status such as orbit feedback parameters and the current BPM readings.

A small amount of extra hardware (not present in our development system) would be required to make this interface nominally compatible with Synchronous Ethernet as used in White Rabbit [4].

An engineered (with FIFO) transition takes place between the signal processing clock domain (124.92 MHz ±20ppm beam for NSLS-II) and the two 125.0 MHz ±50ppm Ethernet domains required for compatibility with commodity hardware (GMII Physical layer). Application of this design to a DSP frequency markedly different from 125 MHz would require a more elaborate transition between clock domains, involving longer FIFOs and careful packet scheduling.

RESULTS

Our prototyping and development platform is made up of between two and six Avnet AES-XLX-V5LXT-PCIE50-G boards with Xilinx XC5VLX50T FPGAs. Each board includes the two SFP modules needed for the ORION network, and two Gigabit Ethernet links. We can use one such link for the controls interface and described above, and the other for BPM inputs. Additional hardware is under construction to give the opportunity for a full system test, including corrector magnet output.

Individual bit errors on the line are logged but have no other effect. With the short (3m) fibers used in our tests, no random errors have been observed.

One measure of timing skew can be performed when all links are connected. If link delays are symmetric, this information is in principle adequate to establish ±1 clock cycle synchronism of trigger pulses around the ring. Under link failure conditions, this measurement can no longer be performed.

Unlike White Rabbit, effort has not been invested in subcycle synchronization. The common-clock paradigm allows jitter-free, cycle-counting operation, and the communication setup will provide triggers with approximately one cycle unpredictability between power-on cycles. Fine timing is expected to be beam-derived in the front-end hardware. If front-end hardware is not fully beam-synchronous, even the jitter-free character of communication and control will be lost.

CONCLUSIONS

With a common clock infrastructure, it is easy to stream data at high rates and with zero jitter through commodity fiber optic communications hardware. The ORION project aims to implement fast communications for a storage ring on such a foundation. A ring communications topology naturally gives the opportunity to provide fault-tolerance, always a desired trait when running an accelerator. The design of fault-tolerant fixed-latency communications requires redundant data channels to be propagated to the application layer.

ACKNOWLEDGEMENTS

The authors wish to thank Bob Dalesio and the team at BNL for their ideas and encouragement. The help of Mike Chin and the ALS electronics group is also much appreciated.

REFERENCES

[1] L. Doolittle *et al.*, "Hardware-based Fast Communications for Feedback Systems," TH6REP076, PAC'09, May 2009, Vancouver, Canada.

[2] L. R. Dalesio *et al.*, "NSLS-II Control System," TUP104, ICALEPCS'09, October 2009, Kobe, Japan.

[3] Y. Tian *et al.*, "Power Supply Control System of NSLS-II," WEP040, ICALEPCS'09, October 2009, Kobe, Japan.

[4] J. Serrano *et al.*, "The White Rabbit project," TUC004, ICALEPCS'09, October 2009, Kobe, Japan.