HARDWARE-BASED FAST COMMUNICATIONS FOR FEEDBACK SYSTEMS

L. Doolittle, A. Ratti, C. Serrano, A. Vaccaro, LBNL, Berkeley, CA 94720, USA
L. R. Dalesio, Y. Tian, BNL, Upton, NY 11973, USA

Abstract

Feedback control entities in an accelerator need means for communication. Low latency, synchronism, determinism, and reliability are some of the factors determining feedback controls performance, and data communication is one of the processes involved in such systems.

The hardware-based fast communication system described here offers means for deterministic, fault-tolerant data transmission for feedback systems requiring low-latency communications. Some of the potential beneficiaries for a real-time beam dynamics network are, among others, beam position monitors (BPMs), corrector magnets, Low-Level Radio Frequency (LLRF) controllers, transverse and longitudinal feedback systems, and machine protection.

INTRODUCTION

When studying communications for feedback controls, we treat the beam as a single physical unit, suffering from external disturbances, measurements, and excitations, and responding to such as a whole. We therefore provide means for a global communication backbone responding to general feedback control requirements such as synchronism, determinism, low-latency, fault tolerance, flexibility, scalability, and connectivity. The communication system is based on a ring network of Field Programmable Gate Arrays (FPGAs), interconnected with optical fibers. A deterministic protocol has been designed to provide minimum latency data distribution around the ring, controlling latency by eliminating traditional computers from the communication design. A second link is used for redundancy and guarantees connectivity in case of a single-point fault.

DESIGN

Principles

The essential communication functions of a feedback system are to collect data from local feedback systems directly attached to each communicating node (called concentrators), distributing it around the ring, to collect data from other nodes, and to make data from the entire network available to feedback or functions. The data is time multiplexed into the ring network following a deterministic protocol, which can be configured to minimize latency, adapting to the number of nodes and amount of data distributed.

The final objective in a fast communication system is to minimize the time elapsed between data generation into a given node and its availability to the rest of the ring. In this respect, the performance of the data distribution depends on four parameters: The size of the data to be distributed, the size of the ring (the electrical distance the data needs to travel), the node local input rate (how fast the data is generated), and the system bandwidth (how fast the data can be handled).

The communication system described here adaptable over a range of these parameters, which are machine and application dependent. After determining the size and generation rate of the data to be distributed, the size of the ring, and the inter-node distances, the configuration parameters are loaded into the FPGAs. The newly configured communication protocol is tuned for minimal distribution latency.

Configuration Parameters

Next, we give an overview of the steps needed to determine the system configuration parameters, which will define how the data is multiplexed into the ring, and the logic followed by every FPGA to implement the communication protocol.

First, we calculate the minimum lap delay, defined as the minimal time that takes a signal to complete a lap around the ring. It is the sum of delays through fibers and the computation delay in all processing nodes (FPGAs). This determines the best possible delay according to the geography of the machine.

All nodes need to be perfectly synchronized in order to communicate in a deterministic way [1], which is achieved by deriving the FPGA clock references from a common distributed Local Oscillator (LO). This reference frequency combined with the amount of data transmitted each clock cycle defines the bit rate of the system.

Every time data is generated in one node, each FPGA introduces data into the network. It is then transmitted to one of its adjacent nodes, which forwards the data to the following node, etc. At the end of one lap, all nodes have introduced their data into the ring, and have retrieved that from the rest of the nodes.

Assuming as an example that one byte of data is transmitted every FPGA clock cycle, the minimal lap delay defines how long it takes one byte of data to complete a lap around the ring. Typically the data introduced by each node into the ring is larger than one byte, and the time it takes all the data introduced in each node to complete a lap around the ring is defined as one spin. By dividing one spin by the number of nodes, we define the time slot allocated to each one of them to introduce data into the ring, we call this time slot a block. These and other protocol definitions

are defined next.

PROTOCOL

The protocol used in the communication system is distributed, and the data is time-division multiplexed into the ring. One node is designated master, the rest being slaves. All nodes have the capacity to be a master or a slave, and the protocol provides mechanisms for both time synchronization, and auto-selection of this mode.

The flow of data generated in digital feedback control systems is quantized, not continuous. Low latency in transmitting that data is vital for the feedback system's performance. The communication protocol needs to keep synchronism, auto-configure, and respond quickly to eventual faults. The functionality of the protocol has therefore been divided into a timing and a data pulse.

A timing pulse is a periodic event that allows the system to keep synchronism, perform a preliminary configuration of the system, and detect eventual broken links or dropped nodes. A data pulse takes place between two timing pulses for actual data transmission. By the time a node receives a data pulse trigger, it is configured and synchronized with the rest of the network, and is able introduce data into the ring during its allocated timing slot.

Terminology

Fig. 1 shows the terminology used in the communication protocol. The three levels refer to the general structure of a data pulse. One message is composed of N+1 spins. In one spin, all nodes (M) are allocated one block. One spin corresponds to a local node input, so in one spin each node will introduce data from one of its inputs in the ring in its allocated block, and will retrieve the corresponding data from the rest of the nodes. Depending upon the implementation, these inputs can be BPM readings, LLRF data, configuration data to be distributed to any feedback system, etc.

Any data that needs to be distributed using the fast communication link only needs to be multiplexed into the node input, which has a well defined interface. The number of spins, blocks, or the payload size is machine and application dependent. The communication protocol is completely configurable to adapt to these parameters.

One of the nodes is assigned as the master by providing it an external trigger, and will generate timing and data pulses according to its configuration. During the timing pulse, state machines in all nodes are initialized in cascade. This is done using comma detection in a 8B/10B encoding scheme. According to the state of its state machine, one node can determine the meaning of any bit received from its adjacent nodes, and since they derive their reference clock from the distributed LO, they will keep synchronism within one clock cycle.

During the timing pulse, a counter is initialized to 0 by the master node and sent to the other nodes. They will each increment this counter by one and determine their position

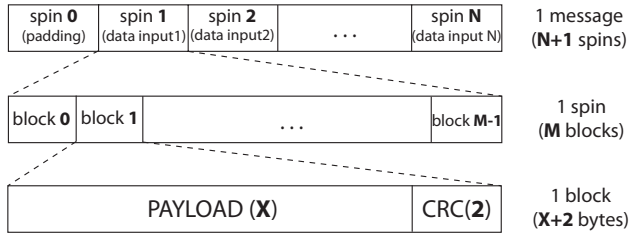

Figure 1: Terminology used in the hardware-based fast communication protocol.

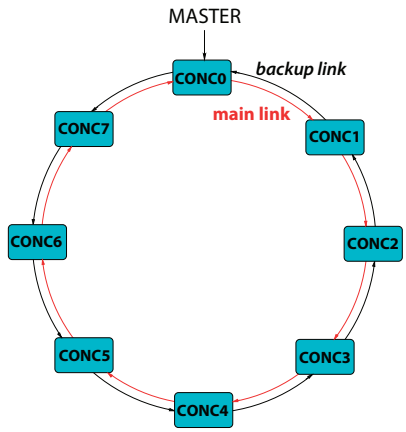

Figure 2: Ring network in normal operation mode. For clarity only 8 nodes are shown. Each node is assigned an ID number according to its location relative to the master, and the sense of data circulation.

in the ring by examining its value. According to their position relative to the master, each node is assigned an ID number from 0 to M-1. This number determines which block (or time slot) is assigned to them to introduce data into the ring. Conversely, all nodes can determine the origin of the retrieved data by its location into the spin (see fig. 1). Each block contains X bytes of payload and two CRC (Cyclic Redundancy Check) bytes to verify its integrity. The size of a block is configurable and identical in all spins.

Data Flow

Fig. 2 shows the configuration of the ring network in normal operation mode. For clarity only 8 nodes are shown. The link circulating data clockwise is used as the main link, and the one in the opposite direction as the backup link. In normal operation mode, data circulates in both directions. The data circulating in the backup link is only used to attempt data recovery when for some reason a CRC in a block of data fails in the main link.

One of the nodes is assigned as the master. Some of the tasks of the master are: starting the data transfer when a burst of data is generated, delivering the proper synchronization messages, and initializing the counter for node ID assignment. The master node is assigned ID number 0,

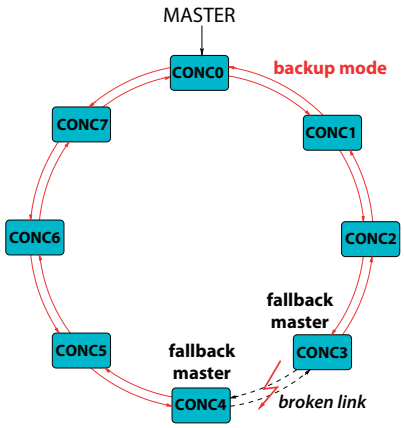

Figure 3: Ring network in fault recovery operation mode. When a link between two nodes is broken all the other nodes detect missing flow of data. Even if that link is broken, all nodes keep connectivity through the backup link.

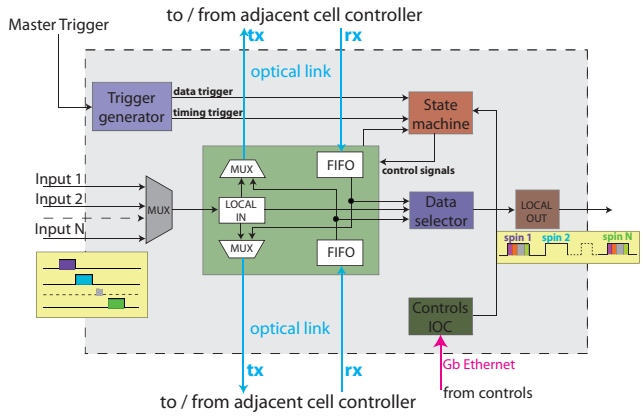

Figure 4: Block diagram of a hardware-based fast communication node implemented into an FPGA.

next number 1, etc. After a synchronization pre-pulse, the ID number of the node emitting data into the link–as well as the total number of nodes–is contained in the header of the message. It is by incrementing the ID number by one that each node discovers its location in the ring. Each message is preceded by a comma, which is used by every node to synchronize and share the same timing scheme. Fig. 2 shows the node ID assignment corresponding to the normal operation mode, where the main link circulates data around the ring clockwise.

Fault Recovery Mode

In normal operation mode, data circulates clockwise (for example), and the node ID assignment determines the order in which they introduce data into the link. When a link (or a pair of links) between two nodes is broken, the backup link is used to keep connectivity between any pair of nodes in the ring. If this situation arises, the system falls into the fault recovery operation mode. In the fault recovery mode, the system has localized the broken link and re-configures automatically to maintain connectivity using the backup link. Concentrators detecting a broken link with their adjacent node fall into a fallback master mode.

IMPLEMENTATION

Fig. 4 shows a block diagram of the fast communication node. A state machine is initialized either by the trigger generator (master node), or by detected commas from the fiber link. The state machine then generates control signals that determine the behavior of the node. Configuration parameters are loaded into the state machine to adapt those control signals to the application and machine it is being used.

Local inputs are multiplexed into the node according to the spin number. The node either introduces it into the link or forwards data from adjacent nodes according to the block number. Ideally, the system is tuned so every node introduces data into the network at the same time (clock cycle). This condition is met when the delay between two adjacent nodes is exactly the size of one block. This is achieved by setting the length of the FIFO at the receive ports in the node to the length of one block, minus the delay through the link electronics and the actual fiber that connects two nodes.

The master node has an additional, dynamically allocated length for its FIFOs to perform spin alignment. This ensures that a new spin will not be started as long as the last bit of data from all the nodes corresponding to the previous spin has reached all the others. The individual FIFOs in each node used for block alignment need to be manually set, and the additional FIFO length in the master is set automatically by the master by detecting the comma it transmitted (signaling the start of a pulse) coming back to it.

CONCLUSIONS

A new approach for fast data communication for feedback systems has been adopted eliminating computers from the communication design. This approach is implemented in FPGAs interconnected through optical fibers, guaranteeing low-latency and deterministic data transmission. Its adaptability, configurability and reliability allow this communication system to be used by a wide variety of applications, including communication for accelerator feedback controls such as fast orbit feedback [2], LLRF, transverse and longitudinal feedback, and machine protection.

REFERENCES

[1] L. Doolittle, “Synchronism, Jitter, and Communications in Accelerator Controls”, LLRF Workshop, October 2007, Knoxville

[2] Y. Tian, L. R. Dalesio, L. Doolittle, C. Serrano “Synchronous Device Interface and Power Supply Control System at NSLS-II”, PAC, May 2009, Vancouver