LEVERAGING INTERNET PROTOCOL (IP) NETWORKS TO TRANSPORT MULTI-RATE SERIAL DATA STREAMS

Doug Heath
Marty Polluconi
Flora Samad
RT Logic Incorporated
Englewood, CO USA

ABSTRACT

As the rates and numbers of serial telemetry data streams increase, the cost of timely, efficient and robust distribution of those streams increases faster. Without alternatives to traditional point-to-point serial distribution, the complexity of the infrastructure will soon overwhelm potential benefits of the increased stream counts and rates. Utilization of multiplexing algorithms in Field-Programmable Gate Arrays (FPGA), coupled with universally available Internet Protocol (IP) switching technology, provides a low-latency, time-data correlated multi-stream distribution solution. This implementation has yielded zero error IP transport and regeneration of multiple serial streams, maintaining stream to stream skew of less than 10 nsec, with end-to-end latency contribution of less than 15 msec. Adoption of this technique as a drop-in solution can greatly reduce the costs and complexities of maintaining pace with the changing serial telemetry community.

KEY WORDS

Multiplexing, Telemetry, Field-Programmable Gate Array, Internet Protocol, low-latency, time-data correlation.

INTRODUCTION

Remote collection of telemetry on test ranges presents a variety of system design issues. One of the main concerns is how the collected data arrives at the central data processing facility. When post-event transfer of recorded data is not an acceptable option, a communication method that provides immediate transfer is required. There are a variety of options available that include direct connection, re-modulation for RF transmission on line-of-sight links, and WAN/LAN networks. The profusion of network infrastructure devices has greatly reduced the cost of implementing an IP solution for real-time data distribution. However, IP-based solutions have latency and time-data correlation issues that must be solved.

SYSTEM DESIGN

The primary goal of any distribution system for remotely collected data is to function effectively as a long wire. Signals originating at the distant source should appear at the central processing facility without errors and with their original timing relationships. One way of minimizing the number of connections between the remote and local sites, and at the same time preserving the relationship between independent signal sources, is to multiplex the data into a single aggregate stream at the remote location. The aggregate data is then transferred to the central facility and demultiplexed back into the original streams (see Figure 1).

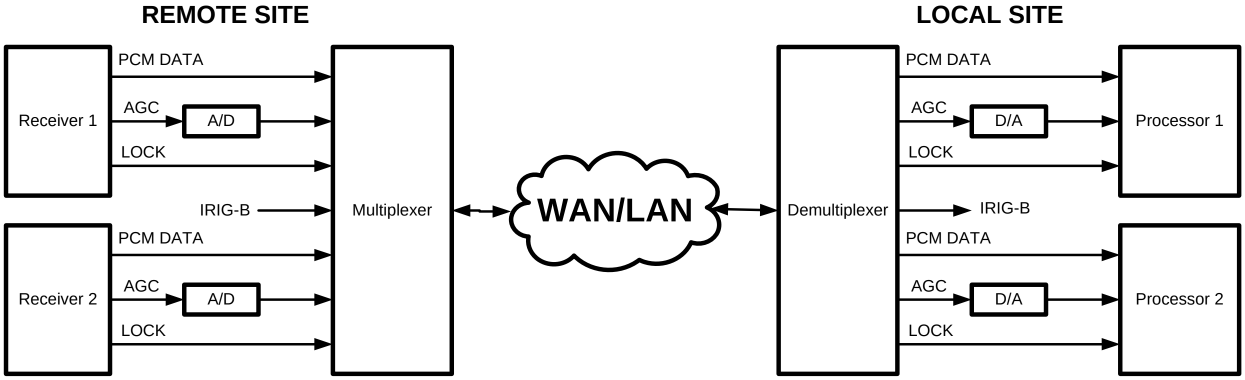

Figure 1. Remote Collection System Using Networks

The data to be transmitted could include timing signals (IRIG), demodulated and bit synchronized telemetry data (PCM streams), digitized analog signals (e.g., received signal power), or discrete status indicators (e.g., bit sync lock). The aggregated data is then sent to the destination facility using TCP/IP messaging over the existing wide area or local area network infrastructure. The demultiplexer system on the other end simultaneously regenerates the original streams so that the processing devices think they are tied directly to the source. Processing of the data can also be performed at the local site in parallel if multiple remote sites sources are available, addressing single point of failure scenarios and allowing multi-stream combination (best source selection) to guarantee data quality.

In some applications, it is necessary to send the same source, or a subset, to multiple destinations. Figure 2 shows a system concept where Sites B and C are receiving all the data from Site A but Site D is only getting PCM data. This allows range users, for example, to receive only the data they are interested in at their own facilities and not be burdened by data related to the infrastructure (e.g., range safety).

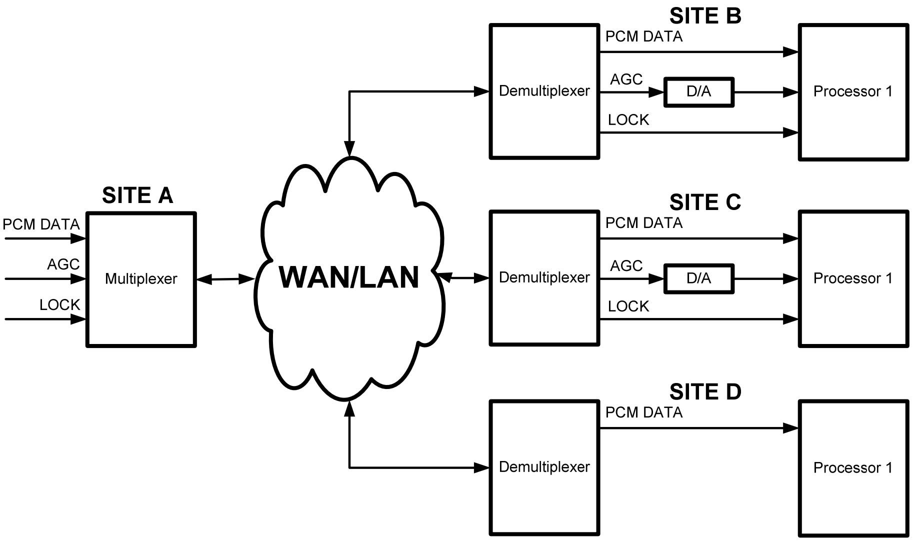

Figure 2. Data Distribution Using Networks

RT Logic has produced a solution called the T500MX Gateway to meet these operational requirements. The Gateway is a variable-rate, multiplexer-demultiplexer system that provides a low-latency, time-data correlated, IP network connection between a source and one or more destinations. Inputs to the system are serial clock and data pairs at any rate from 0 to 30 Mbps, plus an analog IRIG-B or IRIG-G timing signal. The system automatically detects the input bit rate so that it never requires operator input for the expected rate. Whatever shows up on a multiplexer input port will be recreated on the corresponding demultiplexer output port. This feature not only simplifies configuration of the system but also seamlessly delivers links with variable data rates.

Timing differences between the source and destination master clocks are automatically compensated so that the output clock rates are identical to the input rates. The multiplexer and demultiplexer systems do not have a common master clock signal since they are not typically in close proximity to each other. Even though they could be driven from highly stable and accurate 10 MHz reference signals that might have frequency differences of less than 1 ppm, adjustment at the demultiplexer is always necessary to avoid data overrun or underrun. The demultiplexer has logic that detects if data is starting to back up, or is drying up, and adjusts its clock generators accordingly. In addition to regenerating signals that match the input rates, this feature has the added benefit of removing input jitter.

In some applications, it is critical to deliver the remotely collected data to a central facility as quickly as possible. Other applications might require a settable, constant latency, for example as an aid to multiple-source, data correlation algorithms. The T500MX provides a fixed, settable latency as low as 15 msec, excluding the latency of the network itself. The minimum latency is driven by the maximum variation in latency from packet to packet on the network. If the latency on a packet ever exceeds the set point, data underrun occurs since the demultiplexer will not have any data to regenerate. When data finally arrives at the demultiplexer, it is immediately output; however, the recipients of the data will have experienced a gap in the stream. Every aggregate packet created by the multiplexer includes a time tag with extended resolution down to

100 nsecs. When IRIG is available at both the multiplexer and demultiplexer, the system reports the current, minimum, and maximum one-way latency which can then be used by the operator to set the latency to the lowest value that will not produce underrun.

Processing devices on the demultiplexer side could require a timing signal along with the PCM stream. They might also assume a certain relationship between two PCM streams in order to extract or combine data from the two streams. In both these cases, it is critical to track and replicate the original timing relationship between all the signals input to the multiplexer. The T500MX holds stream-to-stream skew to less than 10 nsec, including IRIG. This value is far below the 1% variation allowed by the IRIG 200-04 specification, and contributes only a small amount to the frame-to-frame variation reported by devices that use IRIG to time-tag PCM data, like frame syncs and decommutators. Most of the reported differences will come from jitter in the IRIG carrier.

The IP scheme used by the multiplexer to send aggregate data to the demultiplexer consists of variable-length messages with a fixed amount of overhead per stream. This approach minimizes the amount of network bandwidth required for transmission. In certain applications where the IP message must be converted to a fixed-rate, serial stream for transfer via satellite link or over standard telephony lines, the T500MX can be configured to output a fixed-sized packet instead of its normal variable-length message. It can also output a serial aggregate stream directly in place of an IP message. When using a fixed-sized packet scheme, however, data loss can occur whenever the sum of the data rates (plus overhead) exceeds the fixed rate.

SYSTEM IMPLEMENTATION

Figure 3 shows the T500MX system architecture. The multiplexer and demultiplexer use identical hardware, which consists of a standard high-reliability server and an RT Logic RTL-DFP-PCI. The RTL-DFP-PCI is a general-purpose card with a Virtex-II Pro FPGA, a significant amount of SDRAM, and many I/O lines. The FPGA’s firmware, is downloaded from a file when the server is powered on. Different files are downloaded to configure the T500MX as either a multiplexer or a demultiplexer.

The T500MX application software is minimal. On the multiplexer side, the software is primarily used to read the SDRAM content on the RTL-DFP-PCI and put it on the network. On the demultiplexer, the software reads packets off the network and writes them to the SDRAM on its RTL-DFP-PCI. The actual multiplexing and demultiplexing functions are handled by the firmware without any other software intervention.

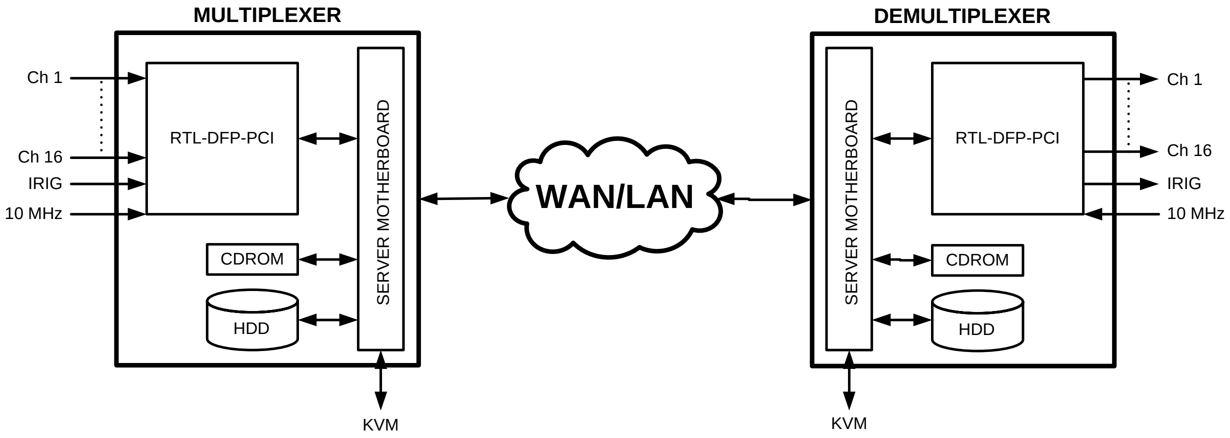

Figure 3. Hardware Elements

The central element in the multiplexer-demultiplexer implementation is the RTL-DFP-PCI (see Figure 4). It is a half-length, short format PCI card that supports 64-bit transfers at a max bus speed of 66 MHz. The PCI bus interfaces to a Xilinx Virtex2 Pro FPGA, which is connected to 32 pair of differential I/O lines. The board can be populated with RS-422, or LVDS drivers and receivers to provide a total of 16 input and 16 output clock and data sets. All the digital I/O is divided between two high-density SCSI connectors on the rear of the card. Four banks of SDRAM provide a total of 512 Mb of 100 MHz memory. Each bank can be independently addressed and is configured as 8 Mwords of memory (16-bit words). An analog-to-digital converter is fed from a BNC connector on the rear connector panel, which is used in this application to digitize an IRIG-B or IRIG-G signal. There is also an analog comparator tied to the analog input, which is used to convert the IRIG carrier directly to a zero-crossing clock which provides highly accurate timing information used by the time-data correlation logic. A digital-to-analog converter is used to regenerate the IRIG-B or IRIG-G signal from the demultiplexer. Its output is also routed to a BNC connector on the rear. A third BNC connector on the rear is for a 10 MHz reference signal which feeds an onboard PLL multiplier chip. A second PLL multiplier chip is fed by an onboard 20 MHz crystal and is used when an external reference is not available or not desired. All multiplexer-demultiplexer functions, except the interface with the network, reside in the FPGA.

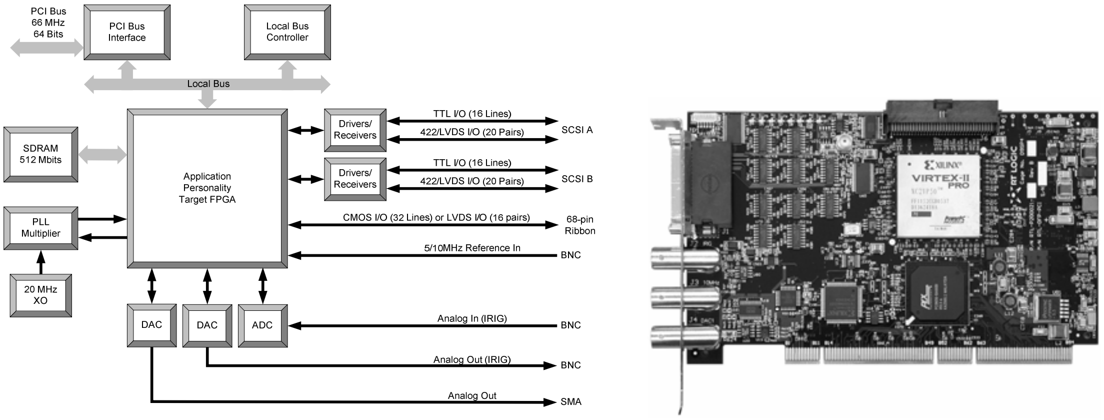

Figure 4. RTL-DFP-PCI Card

The firmware for both the multiplexer and demultiplexer is based on the concept of a *Time Slice*, which is simply a collection of bits for a fixed time period (see Figure 5). The duration of a Time Slice is programmable but must be the same on both sides, and determines the minimum latency. The nominal value is 2 msec since most bit rates in today's telemetry systems are at least 1 kbps. When bit rates are so slow that no clock transitions are detected within the Time Slice, additional data is required by the demultiplexer for proper data regeneration. Using a Time Slice duration that guarantees at least one bit per stream actually helps minimize network bandwidth. However, this may be of little concern depending on the available bandwidth of the underlying network. The latency will always be at least two Time Slices in duration, one for collection by the multiplexer and the second for regeneration by the demultiplexer. In practice, the uncertainty for software to service the firmware, the normal operation of the various IP layers and sharing the available network bandwidth with other devices adds to the minimum latency needed to avoid data underrun.

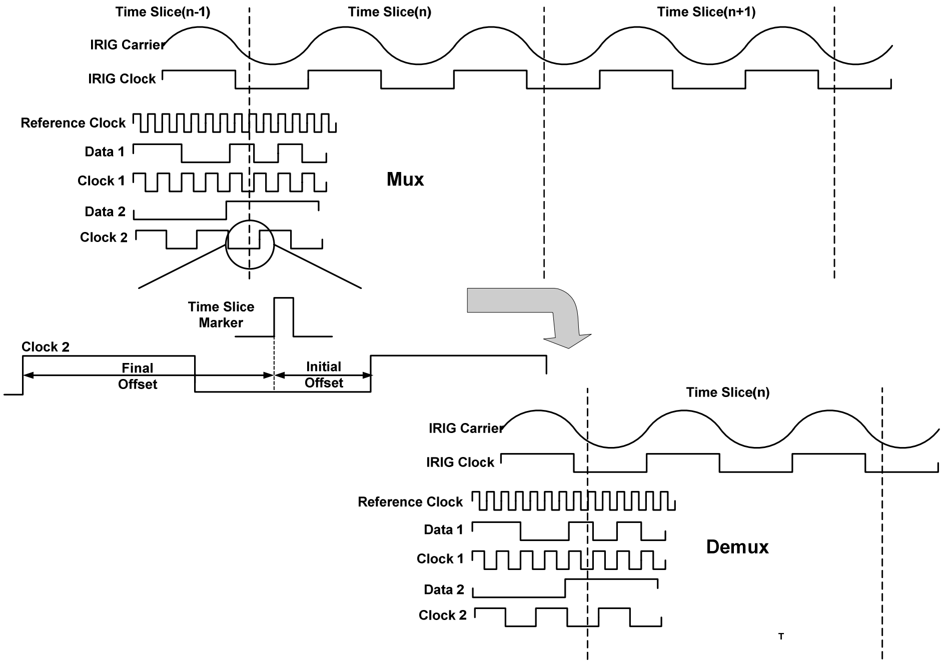

Figure 5. Time Slice Diagram

The demultiplexer regenerates the data by calculating a frequency for a clock generator based on the total bits collected during the Time Slice and the duration of a Time Slice. Clocks accompanying live or playback input data streams have jitter, which is removed during collection and regeneration by calculating an average bit rate across the entire Time Slice. The clock generators in the demultiplexer use Numerically Controlled Oscillators (NCO), which have intrinsic jitter based on the ratio of the generated clock to the NCO's reference clock. This type of jitter is typically more acceptable to bit syncs and clock recovery devices since it is very periodic, opposed to the stochastic nature of jitter from live or recorded data. Devices that rely on the demultiplexer clock output, such as frame syncs, decryptors and decommutators, are immune to jitter.

Knowing the total number of bits in a Time Slice is not sufficient for ensuring tight time-data correlation. If the demultiplexer simply started the data regeneration process for all streams at the beginning of a Time Slice, the original stream-to-stream relationship observed by the multiplexer would be lost. Additional information is recorded by the multiplexer to pinpoint the position of the first and last rising-edge of the clock on each input stream (see Figure 5). The initial and final offsets act to anchor the regenerated data stream for a Time Slice while all other clock edges will be determined by the NCO value computed for the Time Slice. Since all streams are anchored in the same way relative to the Time Slice boundary, their relative timing is preserved. The error in recreating the actual initial and final clock edges is plus or minus one period of the 120 MHz reference clock (8.5 nsec). In addition to the sampling error, measurements of channel-to-channel skew will also be affected by inherent jitter in each stream which could vary from bit to bit, in the worst case. Feeding the same stream into all channels and measuring both their channel-to-channel skew and each channels delay from the same point on the IRIG signal shows the errors introduced solely by the multiplexer-demultiplexer process.

The demultiplexer is responsible for ensuring that the regenerated output bit rates exactly match the input rates at the multiplexer so that underrun or overrun will not occur. The control parameter that increases or decreases all bit rates simultaneously is the number of reference clocks in a Time Slice. The Time Slice duration is set during multiplexer initialization by converting it to a count of the number of reference clocks. The demultiplexer is initialized to the same value but it changes the count based on the current latency. A proportional-integral-derivative (PID) loop is employed to smoothly adjust the demultiplexer Time Slice duration. The error term fed into the loop is the difference between the target and current latencies expressed as the number of backlogged Time Slice buffers. For example, if the target latency is 10 msec and the Time Slice duration is 2 msec, the demultiplexer should have four Time Slices as backlog.

FIRMWARE OPERATION

The firmware downloaded to the FPGA at boot time determines if the T500MX’s internal RTL-DFP-PCI acts like a multiplexer or a demultiplexer. No physical changes are required on the card, only configuration of the direction of the I/O lines will be affected. The firmware design for both the multiplexer and the demultiplexer is channelized to prevent slower streams from impacting higher rate streams.

Data entering the multiplexer passes through the Input Selector block which allows substitution of any stream with a simulated signal (see Figure 6).

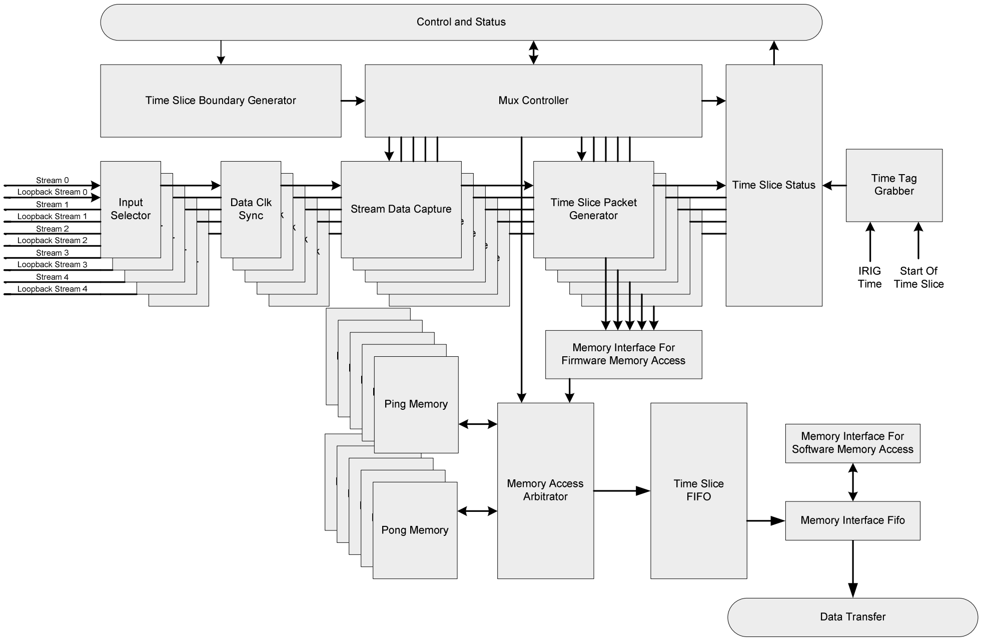

Figure 6. Multiplexer Firmware Block Diagram

From there, the data and clock are synchronized by the Data Clk Sync logic to the reference clock domain. The Stream Data Capture function stores the state of the data line every time it detects a rising edge on the associated clock line. Bits are stored in the ping-pong memory when 32 of them have been collected. This type of memory structure allows data capture in one of the memories while the other is being written to the main FIFO for transfer to the software. The Time Slice Packet Generator creates stream-level header words for the stream, which includes a data word count, stream ID, bit count, initial offset, and a final offset. The Mux Controller uses timing signals from the Time Slice Boundary Generator to close off collection and storage of the current Time Slice. It then starts the transfer of the Time Slice into the Time Slice FIFO by first writing header information from the Time Slice Status block. This header includes a sync pattern, byte count (viz., the sum of all stream packet bytes), a sequence number for detecting missed Time Slices, and a time-tag from the Time Tag Grabber, which has a resolution of 100 nsec. It then proceeds to write the contents of each stream’s ping or pong memory into the Time Slice FIFO starting with the IRIG stream and continuing on with PCM channels in order from lowest to highest. The order is only important to the demultiplexer since it needs to distribute the data to its data regenerators in the same order or to a software application using the IP packets without reserialization. The software uses the Data Transfer interface to retrieve packets from the Time Slice FIFO whenever it is not empty. Retrieval uses the DMA engine built into the PCI interface chip on the RTL-DFP-PCI card to minimize software involvement and maximize transfer rates.

Figure 7 shows how the Time Slice data is distributed and decoded by the demultiplexer. The Demux Controller uses timing signals from the Time Slice Boundary Generator to pre-fetch the header data for the next packet in the Time Slice FIFO. This data is used by the Time Slice Packet Reader to determine how many bytes need to be transferred for each channel, the number of bits that will be generated for this Time Slice and when the first clock edge occurs. It then calculates a new NCO value for each stream based on the bit count, initial offset, and final offset. The current backlog buffer count is checked and used to update the PID loop in the Time Slice Boundary Generator. The demultiplexer also uses a ping-pong memory so that the data for the next Time Slice can be loading while the current Time Slice's bits are playing out. The Stream Data Generator fetches 32 bits at a time and serializes the data at the NCO's rate. Its clock stops during a data underrun. The Output Selector is used to substitute a simulated signal for any output channel and is useful for checking data integrity. The IRIG generator is different than the PCM generators since it creates an IRIG carrier signal with a digital direct synthesizer (DDS) and AM modulates it with the data transferred from the multiplexer.

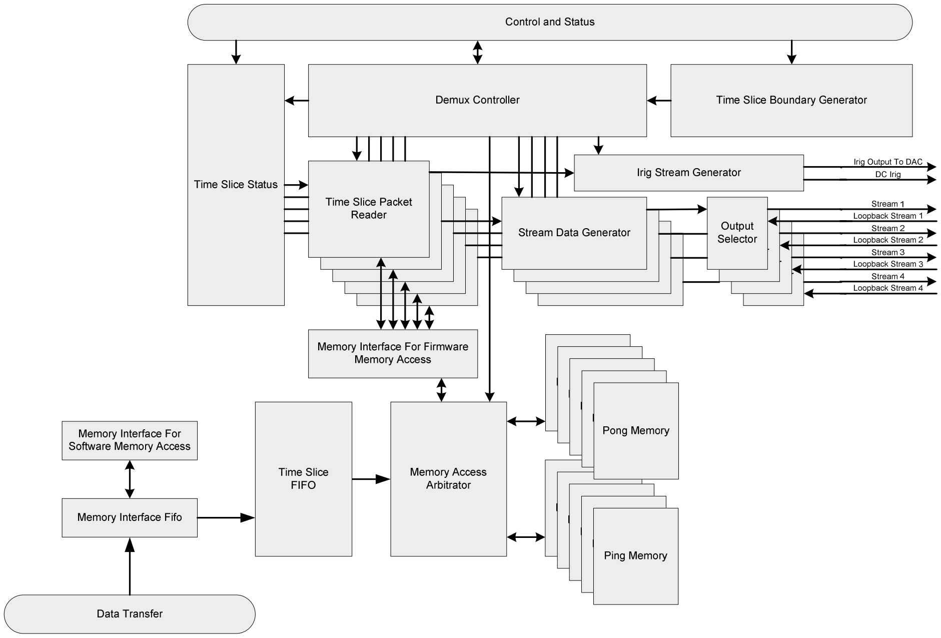

Figure 7. Demultiplexer Firmware Block Diagram

CONCLUSIONS

The T500MX Gateway multiplexer-demultiplexer system provides low-latency transfer of IRIG and multiple PCM data streams over any IP network while preserving the time-data relationships. It is implemented on the RTL-DFP-PCI which is an FPGA-based, short PCI card that is compatible with differential RS-422 or LVDS signals. The cards can be installed in existing servers running Microsoft® Windows® 2000 or XP or, if guaranteed low-latency is paramount, a T500MX server-based solution is available from RT Logic which uses LINUX with real-time extensions.

ACKNOWLEDGEMENTS

We would like to sincerely thank Chuck Brans, Randy Culver, Thad Genrich, Russ Johnson, Mark McMillen, Brian Olson, Jason Phillips, Dave Rolenc and Eddie Tuggle for reviews, suggestions and assisting with the development of the T500MX Gateway system.