DATA ACQUISITION SYSTEM ISSUES FOR LARGE EXPERIMENTS

E. J. Siskind, NYCB Real-Time Computing, Inc., Lattingtown, NY 11560-1025, USA

Abstract

This talk consists of personal observations on two classes of data acquisition ("DAQ") systems for Silicon trackers in large experiments with which the author has been concerned over the last three or more years. The first half is a classic "lessons learned" recital based on experience with the high-level debug and configuration of the DAQ system for the GLAST LAT detector. The second half is concerned with a discussion of the promises and pitfalls of using modern (and future) generations of "system-on-a-chip" ("SOC") or "platform" field-programmable gate arrays ("FPGAs") in future large DAQ systems.

The data acquisition system pipeline for the 864k channels of Si tracker in the GLAST LAT consists of five tiers of hardware buffers which ultimately feed into the main memory of the (two-active-node) level-3 trigger processor farm. The data formats and buffer volumes of these tiers are briefly described, as well as the flow control employed between successive tiers. Lessons learned regarding data formats, buffer volumes, and flow control/data discard policy are discussed.

The continued development of platform FPGAs containing large amounts of configurable logic fabric, embedded PowerPC hard processor cores, digital signal processing components, large volumes of on-chip buffer memory, and multi-gigabit serial I/O capability permits DAQ system designers to vastly increase the amount of data preprocessing that can be performed in parallel within the DAQ pipeline for detector systems in large experiments. The capabilities of some currently available FPGA families are reviewed, along with the prospects for next-generation families of announced, but not yet available, platform FPGAs. Some experience with an actual implementation is presented, and reconciliation between advertised and achievable specifications is attempted. The prospects for applying these components to space-borne Si tracker detectors are briefly discussed.

1 GLAST LESSONS LEARNED

The Si tracking system ("TKR") in the GLAST LAT consists of 16 modules, each of which is a tower of 36 layers with 1,536 Si strips per layer. The DAQ path for the TKR consists of one multi-chip module ("MCM") per layer, one tower electronics module ("TEM") per tower, one global electronics box containing the event builder and other common logic ("GASU"), and three (2 active plus 1 cold spare) cPCI-based event processor PowerPC computer crates ("EPUs"). Event data flow from the MCMs through the TEMs to an event builder module ("EBM") within the GASU, and thence to the active EPU crates via a LAT communication board ("LCB") within each crate. The LCB performs direct memory access ("DMA") transfers via the crate's cPCI backplane to deposit incoming data directly into processor main memory. Events passing the onboard filter cuts in an EPU are returned via cPCI DMA transfer through the LCB to the EBM, which forwards them to the spacecraft solid state recorder for ultimate downlink.

The MCM contains 24 TKR front-end chips ("GTFEs") organized as a pair of serial daisy-chains, one running left-to-right and the other right-to-left. Each GTFE is an application-specific integrated circuit ("ASIC") that provides the complete interface to 64 Si strips. A separate TKR readout controller ASIC ("GTRC") is located at each end of the GTFE daisy-chains. Although 12 of the GTFEs in an MCM are typically read out by the GTRC at each end, the readout path can be reconfigured for an asymmetric split between the two GTRCs to mitigate a GTFE failure on the daisy-chains. Nine MCMs are mounted on each of the 4 sides of each tower, and the readout path for the 9 GTRCs on each end of the MCMs on one side are daisy-chained through separate TKR readout cables for each end of each side.

The bit stream from each of a tower's 8 TKR readout cables is routed to an individual TKR cable controller ASIC ("GTCC") in the TEM. Event data from the GTCC outputs pass via a 16-bit TEM-wide tri-state bus to a common readout controller FPGA, which builds and serializes a TEM's contribution to a LAT event and forwards it to the EBM. The TEM also contains 4 analogous tower calorimeter cable controller ASICs ("GCCCs"), plus another FPGA for trigger signal concentration and monitoring functions.

The TEM event data are routed to a pair of EBM input FPGAs, with 8 TEMs serviced by each FPGA. These FPGAs deserialize the data and store them as 32-bit words in a separate static memory ("SRAM") array for each FPGA. Upon command of the EBM output FPGA, an EBM input FPGA retrieves TEM event data from its SRAM and furnishes these data as a

byte stream to the output FPGA, which in turn forwards that byte stream to the LCB in a target EPU.

The data path for incoming event data in the LCB begins with a deserializer in the LAT FPGA on this board. This converts the incoming byte stream into 32-bit words that are loaded into a discrete FIFO part. A separate PCI FPGA then removes blocks of words from this FIFO and writes these data into a circular buffer in main memory via a cPCI DMA transfer. The event data FIFO marks the interface between the domains of the 20 MHz LAT clock and the 33 MHz cPCI clock. Details of the paths via which data from events passing the onboard level-3 trigger filter cuts are returned from the EPU to the EBM and thence to the ground are beyond the scope of this talk.

Data Formats & Buffer Sizes

Each GTFE has 4 event buffers, where each buffer consists of the 64-bit mask of the presence/absence of a hit in each strip in an event. Each GTRC has 2 event buffers, with each buffer containing up to 64 11-bit addresses (5 bits of GTFE number plus 6 bits of strip number) of those strips with hits in an event. Each GTCC has a 128-entry FIFO where each 12-bit entry consists of the 11-bit strip address within a layer plus an end-of-layer bit. Within the SRAM of an EBM input FPGA, each TEM owns an 8-kilobyte ("kB") circular buffer region which serves as a data FIFO. The 12-bit data words from the GTCC FIFOs are packed into this FIFO at 3 nibbles per word. Finally, the event data FIFO between the two FPGAs in the LCB holds up to 4 kilobytes of data, which is in the same 3-nibble/hit format for TKR hit strip addresses.

Flow Control & Data Truncation

TKR data flow upstream of the TEM relies on the TEM issuing commands to pull data from one buffer stage to the next, based on the state of a buffer model maintained within the TEM. The TKR generates LAT dead time whenever the buffer model in any TEM indicates that all four of its GTFE buffers are full.

Data flow begins when the TEM issues a command to all its GTRCs to pull data from a specified GTFE buffer and store the resulting hit strip addresses in a specified GTRC buffer. Since there are two GTRC buffers, there can be up to two such commands outstanding from a TEM at any given time, with the second such command queued within the GTRC pending completion of the first outstanding command. A command of this type is issued whenever the source GTFE buffer is full and the destination GTRC buffer is empty. If more than two of the GTFE buffers are full, the commands for the remaining buffers are queued in the TEM pending emptying of the target GTRC buffers. Depending upon the configuration of the split in readout allocation between the GTRCs at the two ends of an MCM, a single GTRC may potentially find up to 1,536 hit strips per event. The GTRC terminates its scan of the GTFEs' buffers prior to processing all data from an event if the number of hit strip addresses already stored reaches a programmable limit set in each GTRC. The maximum value of this limit corresponds to the 64 words of storage available within each GTRC buffer. All remaining GTFE buffer data from the event are discarded by the GTRC once the limit is reached.

The TEM also issues a command to all its GTRC daisy-chains to transmit the contents of a specified GTRC buffer to the GTCC FIFOs whenever the command to fill that GTRC buffer is already outstanding and none of the GTCC FIFOs are over a programmable "almost full" threshold. If the process of filling the source GTRC buffer from a GTFE buffer has not completed when a GTRC receives this command, it defers the transmission pending completion of the buffer-filling operation. At most one such command can be outstanding at any time, since the state of the FIFOs must be reevaluated after each such data transfer. If there are outstanding requests to fill both GTRC buffers, the TEM queues the command to empty the second buffer to the GTCC FIFOs pending the completion of the first outstanding command. Note that the GTRC does not notify the TEM when it finishes transferring the contents of a GTFE buffer to a GTRC buffer except in that it defers execution of a command to empty that GTRC buffer. Therefore, the TEM buffer model cannot mark either the GTFE buffer or the GTRC buffer as empty until event data have been transferred from the GTRC buffer to the GTCC FIFO. If the event data volume in the buffers in the 9 daisy-chained GTRCs that feed into a single GTCC FIFO exceeds the available space within that FIFO, the excess data are discarded and an error is generated.

TKR data flow downstream of the TEM utilizes a push model with backpressure from the EPUs' main memory circular buffers to the LCBs, from the LCBs to the EBM, and from the EBM to the TEMs. The EBM asserts backpressure to a TEM whenever the corresponding 8-kB circular buffer is at least half full. The TEM defers initiation of the transfer of an event contribution until this backpressure is absent, but does not respond to backpressure once a transfer is already in progress. The EBM discards the excess in an event contribution over 4,080 bytes and generates an error. Since the maximum length of a "normal" TEM event

contribution is slightly more than 3 kB, data are only discarded in the rare combination of an event with an exceedingly large volume of data concatenated with an error contribution which approaches its maximum length. Even in these cases it is the tail end of the error description that is truncated.

The remaining stages of data flow utilize backpressure to suspend and resume a data transfer, and never discard data from the DAQ pipeline. However, the data transfer from the EBM to the LCB always occurs in units of 128-byte cells, and the backpressure request to suspend transmission is not honored by the EBM until the end of the current cell. In addition, there is hysteresis in the flow control in that the backpressure is not removed until the LCB event data FIFO falls below ¾ full. The timing of the assertion of backpressure is also inexact, and is based on a worst-case estimate of the data volume within the FIFO, assuming that data were only written into the FIFO and not removed from the FIFO since it last became at least ¾ full.

Lesson 1: Chose Proper Data Formats

The Si strips in the LAT have a thickness that is approximately twice their width. The TKR is required to maintain acceptance down to cos θ of 0.2, where θ is the usual polar angle from the zenith in spherical coordinates. This implies that an individual track may intersect up to 10 strips in a single TKR layer. Since the available phase space tends to zero near the zenith, the mean number of hits per track in a layer is significantly larger than unity. Therefore, a data format that describes a cluster of adjacent hit strips rather than a single hit would have made more efficient use of available buffer space and data transfer bandwidth. A rational format might have appended 4 bits of cluster width (biased by 1) to the existing 11-bit format in the GTRC and 12-bit format downstream.

Lesson 2: Provide Adequate Buffer Volume

The volume of buffer space provided to hold zero-suppressed TKR hit strip addresses at each stage in the DAQ pipeline has a potential impact on track-finding efficiency, and thus on detector acceptance. Although the size of the LAT's LCB FIFO was dictated by available components, the volume of SRAM buffer space dedicated to each TEM was chosen to provide space for two maximum-sized TEM event contributions, and was only 1/64 of the space available within the SRAM part. However, the size of the GTRC buffers and GTCC FIFOs were determined with the guidance of a Monte Carlo simulation of DAQ system performance with insufficient physics input. Although the GTRC buffer size proved adequate (and would have been more efficiently utilized given the proper choice of data format), the 128-hit capacity of the GTCC FIFO appears to be insufficient to handle the worst-case requirements of 9 GTRC buffers each containing up to 64 hits, especially if data from previous events remain in the FIFO. Although this FIFO occupies a significant fraction of the available die area in the space-worthy GTCC design, the die size for the GCCC was increased to contain a data FIFO twice the size of that in the GTCC. A similar increase in GTCC FIFO size would have eliminated a major DAQ pipeline chokepoint.

Lesson 3: Understand Truncation/Acceptance

Si tracking systems have large numbers of channels with very low average occupancies. Occasionally, either the underlying physics or electronic noise results in an event whose occupancy stresses the capacity of buffers for zero-suppressed data either locally or globally. In extreme cases, it becomes necessary to truncate the event dataset and discard some zero-suppressed data. It is vitally important to have a clearly expressed, coherent policy for doing so, and to fully understand the effects of this policy on detector acceptance. It is also necessary to mark each event if its data are truncated anywhere in the pipeline, and advisable to note the data volume discarded.

It is highly tempting to take advantage of the low average occupancy to increase the pipeline's duty cycle by removing backpressure when it becomes statistically unlikely, but not impossible, for the size of the next upstream event to exceed the volume of downstream buffer space currently available. This temptation should be avoided unless the downstream buffer is not FIFO-like, i.e. it provides a fixed capacity for the next event regardless of the frequency of recent events or their average event size. Failure to do so can engender a track-finding efficiency, and thus an acceptance, that is dependent not only upon detector and event geometries but also on the recent time history of the detector when an event occurs. Specifically, it is useful to limit truncation to the initial zero-suppression stage of the pipeline, and to perform that suppression into a fixed-length output buffer rather than a FIFO.

Although initial hard-wired LAT design features that permitted the removal of backpressure when space for a maximum-length event fragment was not available in a downstream FIFO have been largely eliminated from the DAQ system, it remains possible to configure the combination of the maximum hits stored in each GTRC buffer and the "almost full"

threshold in the GTCC FIFOs so that the transport of event data from a GTRC daisy-chain to its GTCC may be initiated when there is potentially insufficient space in the FIFO to absorb those data. Discussion of the appropriate settings for these configuration parameters is ongoing, and seems likely to persist until the LAT reaches orbit.

2 PLATFORM FPGAS

With the announcement of the Xilinx Virtex-II Pro family of FPGAs in 2002, and its subsequent availability in small quantities in 2003, it became possible to design DAQ architectures for Si tracking detectors with PowerPC processing systems embedded in the FPGAs within the DAQ pipeline. The FPGAs in this family contain up to two PowerPC-405 processor cores as hard macros at specific locations on the die. Each processor runs at up to 300 MHz internal clock frequency and has a Harvard architecture with a 16 kB, two-way set associative level-1 cache for each of the data and instruction paths. Each part also has up to 20 multi-gigabit transceiver ("MGT") hard macros operating at up to 3.125 gigabits/second in most parts, but up to 6.25 gigabits/second in a few parts. The surrounding FPGA fabric provides up to 444 18-kilobit block memories, up to 444 18 x 18 multipliers, and up to 44,096 logic slices, where each slice contains two 4-input lookup tables ("LUTs"), two register bits, and additional carry/multiplier/multiplexer support logic.

The subsequent Virtex-4 FX sub-family increases the maximum number of available MGTs to 24, of block memories to 552, and of slices to 63,168, while increasing the maximum PowerPC clock frequency to 450 MHz and increasing the performance of the FPGA fabric. It replaces the multipliers by up to 192 flexible 500 MHz multiply-adder/multiply-accumulator digital signal processing ("DSP") slices with registered 48-bit adders. Larger numbers of these slices are available in another sub-family optimized for DSP applications without embedded PowerPCs. Larger members of the Virtex-4 FX sub-family are not yet widely available. The recently announced Virtex-5 family is based on the migration of the Si process from 90 to 65 nanometer feature size. Block memories grow from 18 to 36 kB, LUTs change from 4 to 6 inputs, and DSP performance increases from 500 to 550 MHz while one multiplier input width grows from 18 to 25 bits. However, details of the FX sub-family with embedded hard processors have not yet been announced. Nevertheless, it is reasonable to assume that performance will continue to improve as feature size decreases.

Initial Implementation Experience

Experience at implementing platform FPGA systems was gained with an XC2VP50 Virtex-II Pro part with a single PowerPC running at 250 MHz, a single MGT at 1.25 gigabits/second, and most internal busses clocked at 125 MHz. A custom real-time operating system ("RTOS") providing thread scheduling and inter-thread communication services was developed, and resides wholly within a small portion of the 128 kB of on-chip program memory and 64 kB of on-chip data memory implemented in 96 of the fabric block memories. In addition to these memories, the processor is equipped with two 8-megabyte banks of 64-bit external zero bus turnaround static memory, again clocked at 125 MHz. This system was developed to test advanced concepts for embedding processing in the DAQ component of control systems for pulsed accelerators with significant real-time requirements on pulse-to-pulse fast feedback, and is currently undergoing final application software development prior to future deployment at SLAC.

The techniques developed in this Virtex-II Pro board are now being extended to an XC4VFX60 Virtex-4, with an initial increase of processor clock frequency to 350 MHz and of bus clock frequency to 175 MHz. The external memory is replaced by reduced latency dynamic memory ("RLDRAM-II") with separate pins for write and read data that can operate concurrently. A 64-bit bank of these parts thus provides up to 5.6 gigabytes/second of combined write/read bandwidth at 175 MHz clock frequency because of the double-data-rate ("DDR") protocol employed on the data lines. The configuration for accelerator controls is based on an asymmetric combination of two processors sharing a single bank of RLDRAM-II, with a custom RTOS operating on one processor for hard real-time functions and a commercial RTOS (e.g. VxWorks or RTEMS) with a full TCP/IP stack on the second processor for more feature-rich but less time-critical applications. Design variants with separate RLDRAM-II banks on each processor, and employing a commercial RTOS, are also under development for potential applications in the control module for the SLAC "Petacache" project and the camera control system of the Large Synoptic Survey Telescope ("LSST"). Portions of these FPGA designs have been successfully placed and routed by the development tools, but prototype boards containing these parts are still under design.

Pitfalls

The principal limitations on the performance of the embedded processors in the Virtex-II Pro and Virtex-4

FX families stem from the following architectural features:

1. The processor clock frequency must be an integral multiple, usually a factor of two, of the clock frequency of the processor local bus ("PLB"), which is the external connection to the level-1 caches.
2. The design of the PLB, and particularly of its arbitration logic, leads to large amounts of combinatorial logic and long delays between pipeline registers, thus engendering low bus clock frequency when systems contain large numbers of potential bus masters.
3. The lack of a level-2 cache and the small size of the level-1 cache lead to frequent data cache misses when executing DAQ applications in which the pipeline passes through the processor itself rather than around it.
4. Although the PLB protocol itself permits pipelining of an arbitrary depth, the level-1 cache interface to the PLB does not support having more than two read requests and one write request outstanding simultaneously. This largely negates the advantages of having high-performance code "touch" cache lines, thereby pre-fetching their contents into the cache, well in advance of the time that these contents are required by the underlying algorithm.
5. Errata in the interface between the cache and the on-chip block memory controller prevent the use of any pipelining between the processor's execution elements and caches when a mix of on-chip memory and PLB-accessible memory is employed.
6. The combination of the complexity of the PLB protocol and the necessity for deep pipelines in the read data paths for high-performance DDR memory technologies such as RLDRAM-II lead to large delays when a cache miss results in the need to fetch a cache line from external memory. These delays can easily reach 20 bus clock ticks or 40 processor clock ticks, even when bus/memory arbitration is won after the minimum possible delay.

The combination of these features leads to the observation that, at least for now, the most reasonable use of the embedded processor is to control a DAQ pipeline which actually passes through the on-chip DSP components, rather than through the processor itself. A contemplated application employing 36 DSP slices to multiply a 144 x 144 matrix and a 144-element vector every 4 microseconds to perform a channel-to-channel crosstalk correction on incoming pixels from the LSST camera in real-time might serve as an example of such an architecture.

Space-Borne Application Prospects

The use of digital logic in space-borne applications is always complicated by single-event upsets ("SEUs") resulting from ionization deposited by cosmic rays. GLAST dealt with this problem by choosing an Actel FPGA family that was radiation hardened, employed triplication and voting within the feedback paths of each register bit in the FPGA fabric, and utilized one-time programmable anti-fuses to establish both logic function and signal routing within the fabric. In contrast, Virtex and related FPGA architectures employ static memory to specify fabric logic function via lookup table, as well as to configure routing resources. This leads to a requirement that mitigation for SEUs must be developed for the configuration memory and on-chip processor data and program memories, as well as for data buffers and other functions implemented in block memory, in addition to the usual SEU protection via triplication and voting in register storage and ECC in external memories.

Despite these obstacles, the large feature sets, high performance and density, and presence of both embedded processors and DSP slices makes the use of platform FPGAs in space-borne Si trackers exceedingly attractive. As a result, SEU mitigation is being developed via a combination of manufacturing techniques (radiation hardening, block memories with hardware error correction), additional software steps in the development chain (automatic triplication and voting of fabric registers), and additional logic to periodically restore the contents of configuration memories to their initial state by rewriting their contents during the course of normal operation from an SEU-tolerant known good source (configuration memory "scrubbing").

3 ACKNOWLEDGMENTS

The author had the distinct privilege of participating in the implementation and debugging of the GLAST LAT DAQ system, joining a collaboration each of whose members was highly competent. Enlightening conversations, especially with G. Haller, M. Huffer, J.J. Russell, and R. Johnson, are gratefully acknowledged. Much of the non-GLAST work described herein has been supported by the U. S. Department of Energy under SBIR grant no. DE-FG02-05ER84366, as well as subcontracts under the SLAC operating contract.