

# **DESIGN OF A GIGABIT DATA ACQUISITION AND MULTIPLEXER SYSTEM**

**Albert Berdugo**  
**Vice President of Advanced Product Development**  
**Teletronics Technology Corporation**  
**Bristol, PA**

## **ABSTRACT**

Gigabits and hundreds of megabit communication buses are starting to appear as the avionic buses of choice for new or upgraded airborne systems. This trend presents new challenges for instrumentation engineers in the areas of high speed data multiplexing, data recording, and data transmission of flight safety information.

This paper describes the approach currently under development to acquire data from several types of high-speed avionic buses using distributed multiplexer and acquisition units. Additional input data may include PCM, wideband analog data, discrete, real-time video and others. The system is capable of multiplexing and recording all incoming data channels, while at the same time providing data selection down to the parameter level from input channels for transmission of flight safety information. Additionally, an extensive set of data capture trigger/filter/truncation mechanisms are supported.

## **KEY WORDS**

Data Acquisition, Fibre Channel, FireWire, Multiplexer, Gigabit, Recorder – Solid State.

## **INTRODUCTION**

Flight test instrumentation finds itself in a constant struggle to accommodate the ever-increasing demands placed upon it by modern avionics systems. As avionics systems continue to expand and embrace modern digital communication technologies, test instrumentation finds itself with the difficult task of efficiently acquiring and preserving a wide range of input sources. From low-speed analog data, medium-speed PCM-encoded time-division multiplexed data, high-speed packetized digital messages and very-high speed (gigabit) protocols, it becomes increasingly difficult to build flexible, cost-effective, and performance-efficient systems that can handle this mix of data in a way that can satisfy the requirements imposed on the flight test engineer by the avionics community.

One approach used by the community today is premised upon building specialized and highly integrated units that merge the functions of data acquisition and data recording. Such a unit has the ability to handle a wide range of data acquisition rates and provides some degree of flexibility, but lacks as a cost-effective and performance-efficient solution when deployed as a total solution for

flight test instrumentation in a modern military fighter or large commercial jet aircraft. In addition, today's integrated data acquisition/multiplexer systems don't address the acquisition and selection of flight safety data, resulting in a need to implement two separate but equal data acquisition networks.

An alternate approach will be described here, which results in a better system level solution to the problem of flight test instrumentation when confronted with today's heterogeneous avionics systems. This approach follows from viewing the flight test instrumentation implementation as a distributed problem, which can be resolved by using a layered system approach (both physically and in bandwidth). Distributed data acquisition and multiplexing represents a natural and incremental step in the development path being followed by TTC products and whose need was driven by the requirement to accommodate gigabit data sources for a major fighter program in an evolutionary manner.

The remainder of this paper outlines the design goals used to shape the framework of our solution, details the process used to resolved three key system architecture decisions, and describes the main features and capabilities of our solution.

## **DESIGN GOALS**

The framework for our distributed gigabit data acquisition and multiplexer system was driven by the choices made for the design goals. The design goals themselves were derived from a variety of sources, including past experience, corporate strategy, customer input, and technology constraints. These goals were:

- Support a minimum system burst data rate of 1 gigabit per second to the flight recorder(s) with a future growth path to over 2 gigabits per second.
- Each system unit should be processor-based (not state machine based) to allow for maximum flexibility in programmability.
- Data transfers between multiple units (if needed) must be capable of reaching the same bandwidth as supported between a single unit and the flight recorder(s).
- Capability to configure a single unit to acquire data from a variety of sources with low to very high bandwidth requirements (few Mbits/sec to a few Gigabits/sec)
- The design should leverage, where possible, use of the latest proven commercial chip technology for high-speed interfaces and buses.
- The unit should be able to accept IRIG time codes and/or a CAIS bus interface without requiring the use of a dedicated peripheral slot.
- The hardware design for each unit must be flexible enough to allow for a variety of peripheral cards to be plugged into the unit to create multiple personalities:
  - A multiplexer and data recording unit with support for any media
  - A high-speed data acquisition unit
  - A high-bandwidth health monitoring unit, or

- A general-purpose airborne computer.
- The system should support the use of a general-purpose real-time operating system (preemptive kernel with real-time scheduling).
- The bus architecture of the unit should lend itself to supporting 4-12 slots.
- The system must not require the use of forced-air cooling.
- It is desirable to restrict the number of processors used in the system in order to reduce the heat, power, and space required and minimize operating system software complexity.
- The aggregate bandwidth of the system backplane, at a minimum, should be at least twice as large as the bandwidth of the fastest peripheral card plugged into the bus.
- The number and type of data acquisition ports is configurable through peripheral card selection by the user in his/her lab.

Finding a solution that optimizes all design goals is a very difficult problem, usually the number of variables involved results in an under-constrained system of equations. A common algorithm used in such a situation is known as the Greedy algorithm. Rank the variables according to some benchmark of importance to the final solution and choose the highest ranked variable and solve it independently. Reorder the variables again and repeat until a solvable system of equations is reached. During the design process used for the distributed system described here, the three top design variables found during this process were bus architecture, processor architecture and operating system.

### **BUS SELECTION PROCESS**

Early on in the design process, it became clear that building the customizable unit needed for our gigabit distributed system would require the availability of a multi-slot chassis. The chassis would incorporate a backplane to bus signals between the various peripheral cards within the unit. Based on this architecture, a suitable high-speed bus was required to fit the needs of the product family.

When selecting the backplane bus for the unit, several criteria were considered. The bus had to be capable of sustaining the bandwidths required to move data to/from the boards present on the backplane. It was also important to choose a bus that was supported by the industry. Using an industry standard bus usually implies that there is a wealth of compatible devices, software and tools available. The availability of such items lends itself to lower cost implementation as well as reduced development time. There are sometimes downsides to adopting standard bus structures. For example, being locked into a certain protocol that may not fulfill a given design need. These factors must be weighed when making the decision as to which bus to adopt. Another factor considered was the scalability of the bus. It was important for the design effort to choose a bus that would allow migration to higher bandwidths if and when needed.

Several buses were evaluated for use in the system. Initially the decision was made not to pursue a totally custom bus. Although a custom proprietary bus could be tailored to this application, the amount of effort required to develop the necessary backplane devices (i.e., ASICs and FPGAs) was deemed prohibitive. Also, as it would be desirable to support a wide variety of high-speed

commercial interfaces (such as FireWire and Fibre Channel), the effort to build bridges between a custom backplane and standard components would become excessive. As such, the evaluation focused on industry standard buses.

The decision was made to use a CompactPCI<sup>®</sup> Bus-like backplane. CompactPCI is based on the PCI Bus protocol. CompactPCI and PCI Bus are established industry standards and there is a wealth of support for these standards in the marketplace. The system would use a 64-bit wide backplane operating at 66 MHz. This translates to a peak bandwidth of 528 megabytes per second (MB/s). Accounting for real-world data traffic and protocol overhead, bandwidths of over 200 MB/s are realizable. For this application, however, additional signaling (such as IRIG time codes) would be required between boards. Also, the physical form factor defined by the CompactPCI standard was not suitable for this application. To work around these limitations, modifications were implemented. The unit would use boards that are somewhat wider than those defined by the 3U CompactPCI standard. Additionally, the bus itself was extended to add the additional timing signals needed to fulfill the design goals. These signals included, among other signals, a time bus and a master clock to support IRIG Chapter 10 data time tag format. These signals were added in a way that does not impact the CompactPCI defined bus signals. This allowed the use of third party boards and tools during the development phase of the product and the ability to plug our boards into standard PCI backplanes.

With the advent of the PCI-X protocol, the PCI Bus has a migration path to higher bandwidths, albeit at the expense of the number of slots supported. PCI-X not only has the ability to operate at higher frequencies, the protocol itself is more efficient. This results in peak bandwidths exceeding 1 gigabyte per second. The backplane within the unit was designed to support the PCI-X protocol. Consequently, the system will be ready to take advantage of the benefits of this latest enhancement to the PCI Bus when needed.

The CompactPCI Bus is a good choice for this airborne instrumentation application. The standard is widely supported which results in a cost-effective unit with a reduced development cycle. The physical aspects of the packaging are adaptable to the demanding airborne instrumentation environment. Lastly, the bus performance enables the delivery of a high performance unit today with a migration path that will future-proof the system for tomorrow.

## **PROCESSOR SELECTION PROCESS**

When selecting the processor for the distributed system, several key features were required. First, the processor would need to provide an adequate amount of performance to fulfill the roles of system manager and data manipulator. It would need to accomplish this while consuming a relatively small amount of power and under specific environmental conditions. In addition, the processor would need the appropriate level of peripheral support in order to interface to other buses and circuits while minimizing the number of additional support devices. Lastly, the availability of an appropriate operating system with associated development tools and support would be required for the processor candidate.

Several processors with varying architectures and from various suppliers were considered. Ultimately, a PowerPC® (PPC) based processor was chosen. In addition to the PowerPC core, the processor provides integrated peripheral controllers on-chip. These include a PCI-X Bus controller, a DDR-SDRAM controller, a peripheral bus controller, two Fast Ethernet MACs, two UARTs, two I<sup>2</sup>C Bus controllers as well as other functions (e.g., DMA and interrupt controllers).

The processor core has a peak performance rating of 800 Dhrystone 2.1 MIPS. Peak performance is attained with the help of integrated Level 1 instruction and data caches. Another key element to the performance is the memory controller. A high performance Double Data Rate Synchronous Dynamic RAM (DDR-SDRAM) controller, which is capable of peak data rates of 2.1 gigabytes per second, is supported. Adding to the primary functional blocks of the processor is a PCI-X Bus controller. The controller supports both 64-bit conventional PCI and PCI-X Bus protocols. At a maximum speed of 133 MHz and in PCI-X mode, the bus has a peak data rate of just over 1 gigabyte per second. In conventional PCI Bus mode, the peak data rate is 533 megabytes per second. Another important interface to the processor is the peripheral bus. The available 32-bit demultiplexed, general-purpose bus is ideal for interfacing to non-volatile memory, SRAM and slower peripheral devices. The peripheral bus can operate up to 66 MHz for a peak rate of 266 megabytes per second.

Power consumption and thermal emissions was another important factor when choosing the microprocessor. Because the unit will need to make extensive use of conduction cooling, the processor has to provide an elevated level of performance with a minimum heat signature. Again the PPC architecture was able to deliver the performance needed while consuming less than 4 watts while running the core processing unit at 400 MHz. Packaging also has an impact on thermal performance. The chosen processor is packaged in a small ceramic ball grid array that lends itself to good thermal dissipation.

Industry support for the PPC architecture is extensive. There are many companies actively marketing software, development tools, etc. for these devices. Also, there are roadmaps from the chip manufacturers showing the migration path of the PPC family. This should ensure the advancement and longevity of these devices.

## **OPERATING SYSTEM SELECTION PROCESS**

An evaluation of embedded operating systems was made to choose a solution that would serve as the basis for application development for the distributed gigabit system. The following types of operating systems were considered for this unit:

- A customized 2.4.18 Linux distribution that provides a preemptive kernel with real-time scheduling for use in embedded systems.
- A customized 2.4 Linux distribution that uses a dual kernel approach to provide hard real-time performance.
- A POSIX-compliant real-time proprietary operating system that can emulate Linux at the binary level.

- A POSIX-compliant real-time proprietary operating system.
- A customized 2.4 Linux distribution that offers near-real-time performance using customized loadable kernel modules.
- A real-time proprietary operating system.

Eight attributes were chosen which represented what were deemed the most important considerations for choosing the appropriate development operating system and development environment. Each attribute was assigned a percentage out of 100% relative to all the others based on its relative importance to the distributed system. Partial percentages are assigned for the range of encountered values appropriate for each attribute. The higher the value, the better the result. Maximum score would be 100.

Each decision attribute is described in more detail:

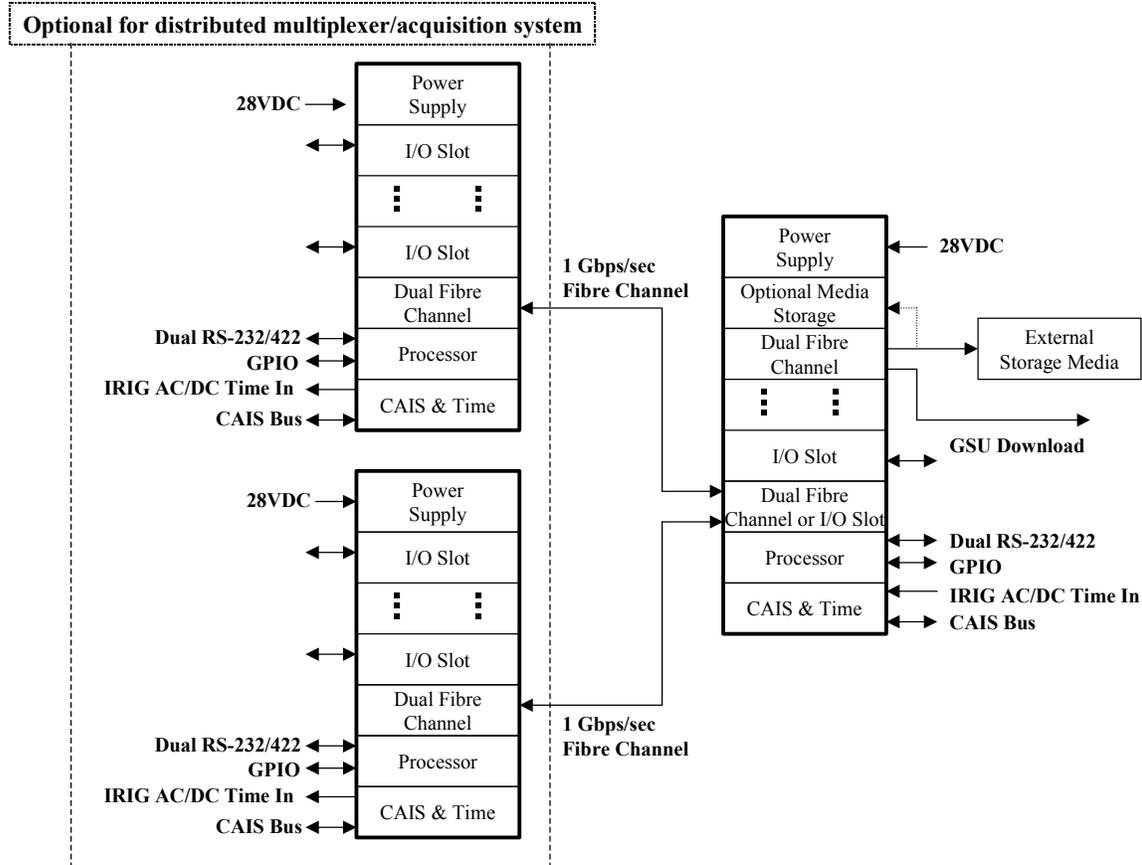
- *Development Cost.* This is the cost associated with a single developer seat plus 1 year of priority support from the vendor. Any one-time costs required by the vendor (such as choosing an architecture) are included. Assigned weight of 5%.
- *Runtime Cost.* This is the cost required to purchase 100 runtimes from the vendor when buying the first development seat. Some vendors charge more for runtimes bought incrementally. Assigned weight of 5%.
- *Source.* The ability to have source code to the operating system has proved itself to be invaluable in the past. The amount of source code provided by the vendor and the cost associated with obtaining this has been weighted here. The total assigned weight is 20%.
- *IDE.* A well integrated development environment with a rich set of tools increases the productivity of the software development team. Assigned weight of 10%.
- *Chosen Processor Support.* Preexisting and well-tested support in the operating system for the PPC processor itself and the associated development system would be invaluable. Assigned a scaled weight of 20%.
- *FireWire and Fibre Channel Support.* The availability of supported drivers in the operating system for both Fibre Channel and FireWire will be very important to the project. This includes both freeware and 3<sup>rd</sup> party (for sale) versions of this software. Assigned a scaled weight of 20%.
- *Qt/GTK+ Support.* Application support for the integration of cockpit display devices was deemed worthwhile. Assigned weight of 10%
- *Real-Time Capable.* The ability of the operating system to support various levels of real-time processes is an important attribute for a software system that performs data acquisition. It is difficult to assign a true value to this attribute because most real-time requirements turn out to be soft real-time, not hard real-time. Assigned weight of 10%.

Based on the assigned weightings, the customized 2.4.18 Linux distribution that provided a preemptive kernel with real-time scheduling scored highest, with a final score of 92.

## DISTRIBUTED MULTIPLEXER/ACQUISITION ARCHITECTURE

The basic architecture consists of a multi-slot custom cPCI chassis populated with a common set of system and peripheral cards, configured for use as either a high-speed acquisition, general airborne computing, or multiplexing device through a software load, and interconnected into a group of 1-5 units for integration into a total onboard flight instrumentation package. An example of a valid system configuration is shown in Figure 1:

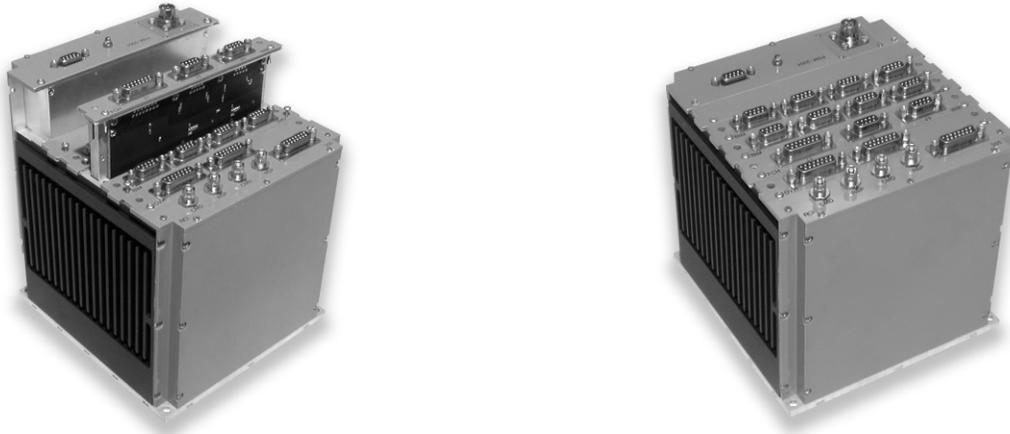
**Figure 1. Example of Distributed System Configuration.**



In Figure 1, two of the chassis have been software configured to perform the data acquisition function and loaded with high-speed data acquisition cards and one of the chassis has been software configured to perform the multiplexer function and loaded with a high-speed inter-connecting card and other peripheral cards.

Each unit in the distributed system contains a fixed number of slots and the number of slots is dependent upon the required backplane speed. The baseline unit supports a total of 4 peripheral slots and contains a backplane that provides a peak system throughput of 533 Mbytes/sec. The unit is enclosed in a ruggedized airborne enclosure with outer chassis dimensions of 6.3 inches by 6.0 inches by 6.6 inches. The total estimated power consumption is 43 watts. A photo of the finished four slot unit, stuffed with an overhead card and several peripheral cards is shown in Figure 2.

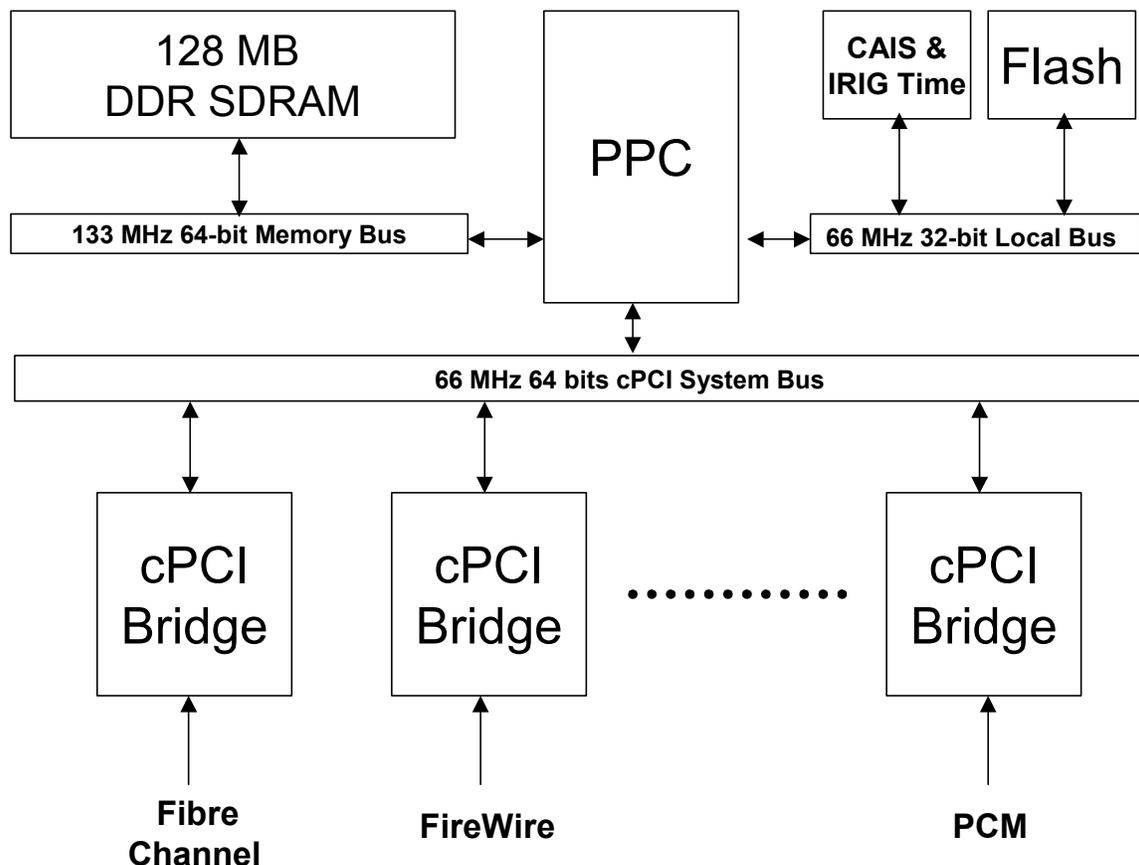
**Figure 2. Completed Chassis and Boards.**



An alternate chassis using a backplane that provides a peak bandwidth of 264 Mbytes/sec and allowing for 6 peripheral slots is also supported. All peripheral cards are interchangeable between different chassis sizes and allow the user to configure a unit to support a large variety of different interfaces and port count per interface. Every unit is required to support an overhead card called the OVH-300 that provides system-wide support for general-purpose programming. This card does not occupy a peripheral slot in the chassis, but instead is inserted into a doublewide slot opposite the power supply card. The PowerPC processor is contained on this card along with 128 megabytes of high-speed SDRAM and 32 megabytes of non-volatile storage. Two RS-232/422 configurable serial ports are supported along with eight general-purpose input/output signals.

Figure 3 shows the relationship between the overhead card, system backplane and peripheral cards within a single unit. The PowerPC processor provides a centralized location where system configuration and management are performed, and acts as a traffic cop in scheduling the movement of data between peripheral cards in the unit. Central to the performance of the unit is the use of a 16.8 gigabits per second memory bus that allows for peak data throughput within the system.

**Figure 3. Unit Bus Architecture.**



The system backplane makes use of a 66 MHz, 64-bit backplane to move data between cards. The backplane is transparently compatible with both the PCI 2.2 specification and the PCI-X 1.0 specification, allowing an incremental path for the user in the future to achieve backplane speeds approaching 8 gigabits per second. Each peripheral card contains either a PCI 2.2 or PCI-X 1.0 compatible bridge to interface with the system backplane, depending upon the bandwidth requirements need.

The supported peripheral cards as described in detail below.

- The RCI-305 card provides 1 looped port of a 5 megabits per second CAIS bus interface and an IRIG AC/DC input, DC output port. This card doesn't actually plug into a peripheral slot; instead it is attached to the overhead card via a mezzanine connector. This allows all peripheral slots to remain open while providing a modular way of supporting multiple system command and control interfaces. This card accepts IRIG-B time for purposes of time stamping data acquired directly on the unit. The CAIS bus interface is used for programming the system (although programming and setup can also be done through the RS-232/422 ports) and providing support for selected data retrieval, recorder status, and peripheral card status for transmission.
- The FCH-302E peripheral card supports 2 independent 1.0625 gigabits per second electrical Fibre Channel ports. This card serves a dual role in the distributed architecture; it can be used

to connect a single data acquisition unit to the multiplexer unit in order to expand the capacity of the flight instrumentation system or it is used to interface the multiplexer to the data storage devices using Fibre Channel arbitrated loop.

- The FCH-304L card supports 4 independent 2.125 gigabits per second optical Fibre Channel ports for data acquisition.
- The BIM-394Q card supports 4 independent 1394b FireWire ports running at 400 megabits per second for data acquisition.
- The PCI-304 card supports 4 independent 20 megabits per second PCM data ports for data acquisition.
- The VID-302 card supports dual MPEG-II video and audio compression ports capable of bit rates of 1 to 25 megabits per second.
- The BIM-553Q card supports 4 independent dual-redundant MIL-STD-1553 bus interfaces for data acquisition.
- The GIO-302 card supports a variety of general-purpose input/output signals.
- The MCI-305 card performs the function of a CAIS bus master, allowing a unit to aggregate data from a variety of downstream low to medium speed data acquisition devices.

The four slot chassis, when configured as a Multiplexer, makes use of an external data storage device (either hard-drive or solid-state). The default interface is a 1.0625 gigabit per second electrical Fibre Channel configured to support arbitrated loop (thus allowing for multiple devices to be attached to a single interface), but other interfaces can be supported (such as FireWire or SCSI). The six slot chassis allows for an internal data storage slot using the same media interfaces as supported by the four slot chassis.

## CONCLUSION

Choosing to build a distributed high-speed data acquisition and multiplexing architecture from a single customizable and programmable unit allows for maximal return on the design and development investment in a market where the number of systems sold is very small. The distributed system architecture discussed here allows for the creation of a flight test instrumentation solution that allows over 100 megabytes per second of data acquisition, multiplexing, health monitoring, recording and general-purpose airborne processing in a modular, yet compact package rugged enough for the avionics environment.