Mixing Staged Data Flow and Stream Computing Techniques in Modern Telemetry Data Acquisition/Processing Architectures

James William Yates
L-3 Communications Telemetry & Instrumentation
15378 Avenue of Science
San Diego, California 92128

Phone: 858.674.5100, ext. 4140
e-mail: jyates@ti.L-3Com.com

Abstract

Today's flight test processing systems must handle many more complex data formats than just the PCM and analog FM data streams of yesterday. Many flight test programs, and their respective test facilities, are looking to leverage their computing assets across multiple customers and programs. Typically, these complex programs require the ability to handle video, packet, and avionics bus data in real time, in addition to handling the more traditional PCM format. Current and future telemetry processing systems must have an architecture that will support the acquisition and processing of these varied data streams. This paper describes various architectural designs of both staged data flow and stream computing architectures, including current and future implementations. Processor types, bus design, and the effects of varying data types, including PCM, video, and packet telemetry, will be discussed.

Key words

Flight Test Processing, Staged Data Flow, Symmetric Multi-Processing, Asymmetric Multi-Processing, Stream Computing Architecture, Mixed Architecture.

Processing Structures

In today's telemetry processing environment, many more complex data streams and tasks are required to satisfy current and emerging requirements. What was once the sole domain of analog FM streams or single Time Division Multiplexed (TDM) Pulse Code Modulation (PCM) streams is now filled with multiple data streams of varying content and format. Once, avionics systems in flight test and space vehicles were monitored in real time by embedding asynchronous bus data into synchronous PCM streams. The trend now is to

treat asynchronous data as such, placing it in an asynchronous transmission method such as CCSDS or the new IRIG-107 Standard and then telemetering or recording it. Additionally, now it is often the norm to require handling mixed data types and multiple data streams in real time.

So, the first question becomes what type of requirements are out there that require this mix of synchronous and asynchronous data in real time? Some current requirements that come to mind are listed in Table 1.

Table 1. Data Types by Platform

Avionics Flying Test Beds	Ground Stations for UAVs & RPVs	Mixed Processing Avionics Laboratories
PCM	PCM	PCM
MIL-STD-1553	CDMA	Pi Bus
ARINC-429 & -629	UHF	MIL-STD-1553
Ethernet	Video	CCSDS
ATM	CCSDS	MIL-STD-1773
CCSDS		Video
Video		HSDL
RS-232, -422, -485		STANAG

With the emerging changes in telemetry data structures came large increases in data quantity and bandwidth. All these changes make designing current telemetry processing systems a challenge and provide many unique opportunities to mix technology into adapted solutions.

In general, current computer processing structures (modern telemetry ground stations and advanced data acquisition systems included) can be divided into two main architectural types, assuming that all current and future systems are multi-processor in nature. *Multiprocessing systems can increase performance, hasten response time, and simplify software development. As its name implies, a multi-processing system uses more than one processor to solve an application task.*[1]

These generic architectural processing types are referred to as either symmetric multi-processing (SMP) or asymmetric multi-processing (AMP). Deciding between tightly coupled symmetric multi-processing and loosely coupled asymmetric multi-processing architectures is a matter of application rather than technology. When dealing with stringent real-time requirements, generally one architecture will perform better than the other. Additionally, subsets of the above architectures include non-uniform memory access (NUMA), network of workstations (NoW), and distributed shared memory (DSM). Since this paper is geared towards the architecture of telemetry and data acquisition systems, we will deal primarily with the most common architectural types.

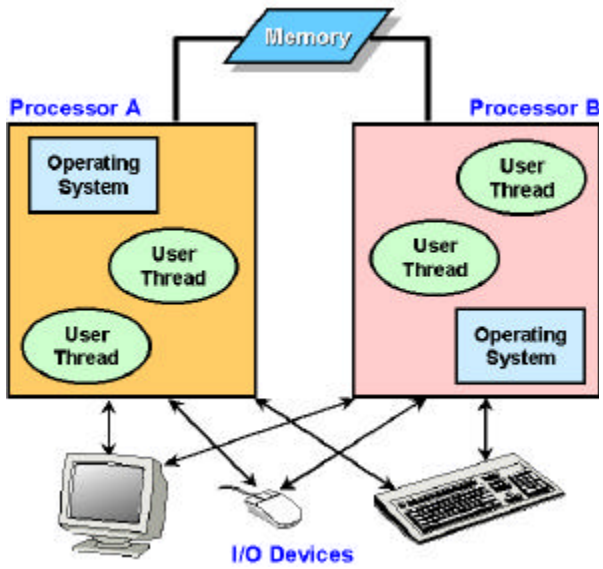

Figure 1. Symmetric Multi-Processing

In the tightly coupled approach, separate processors share the same memory space. This allows them to work on the same data simultaneously (see Figure 1). Such an approach necessitates using an SMP-capable operating system like Windows NT. This can work well for math-intensive applications, but requires the overall architecture to remain completely heterogeneous.

The main benefits of using an SMP architecture are:

- Processes are executed in priority order, with higher priority processes always executed over lower priority processes
- Each processor has its own queue of processes
- Synchronization is in the hardware
- Allows a single machine to do more work
- Allows better predictability for running tasks
- Allows systems to be scalable
- Inexpensive SMP computers are available off the shelf [2]

Without the requirement to merge time-coherent dissimilar data streams or provide small grain processing for time determinism, a heterogeneous SMP architecture provides a more cost-efficient and simpler solution. Many times, the solution can be provided in an inexpensive form factor such as ISA or PCI (see Figure 2).

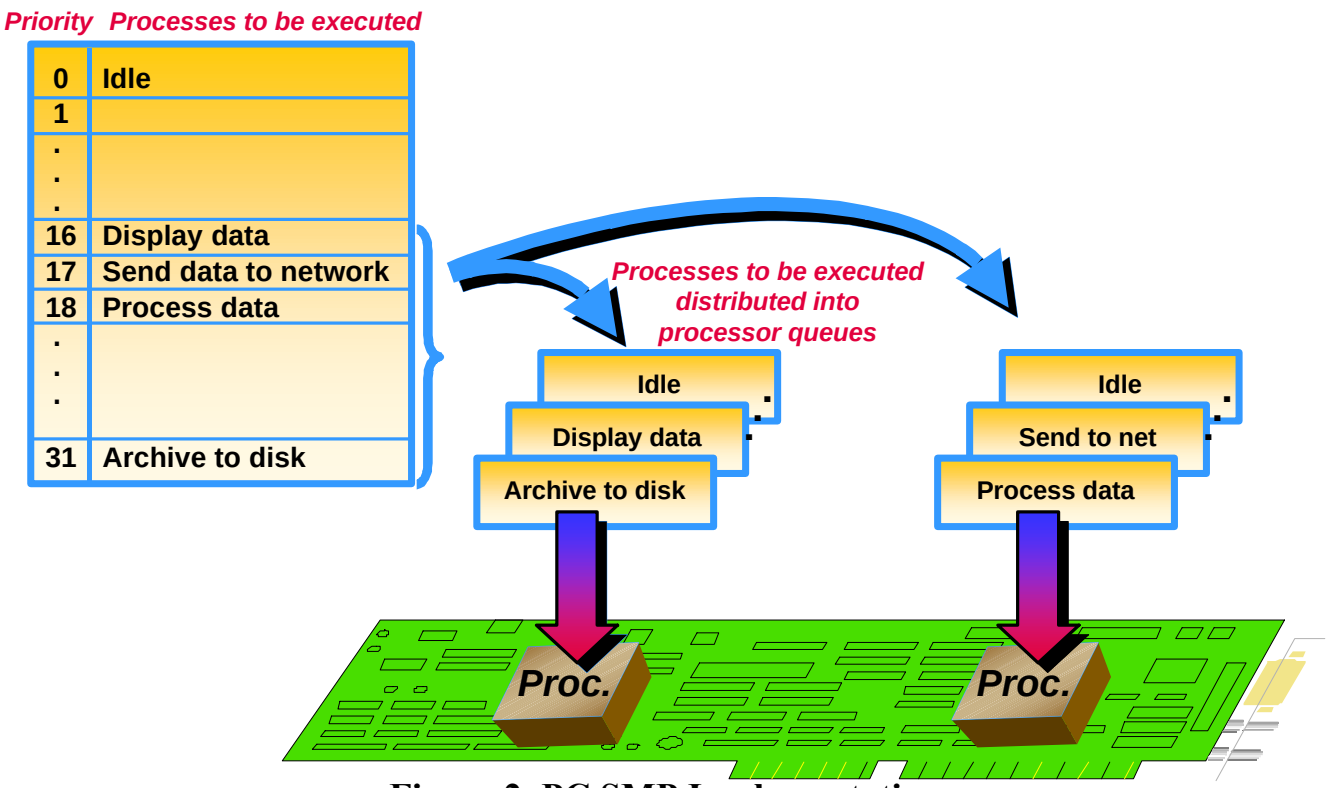

Figure 2. PC SMP Implementation

In the AMP configuration, each processor has its own memory and local devices, but shares information with other processors via a common interface such as Ethernet or ATM. A more efficient usage is found in AMP architectures that share the same high-speed computer backplane as the interface (see Figure 3).

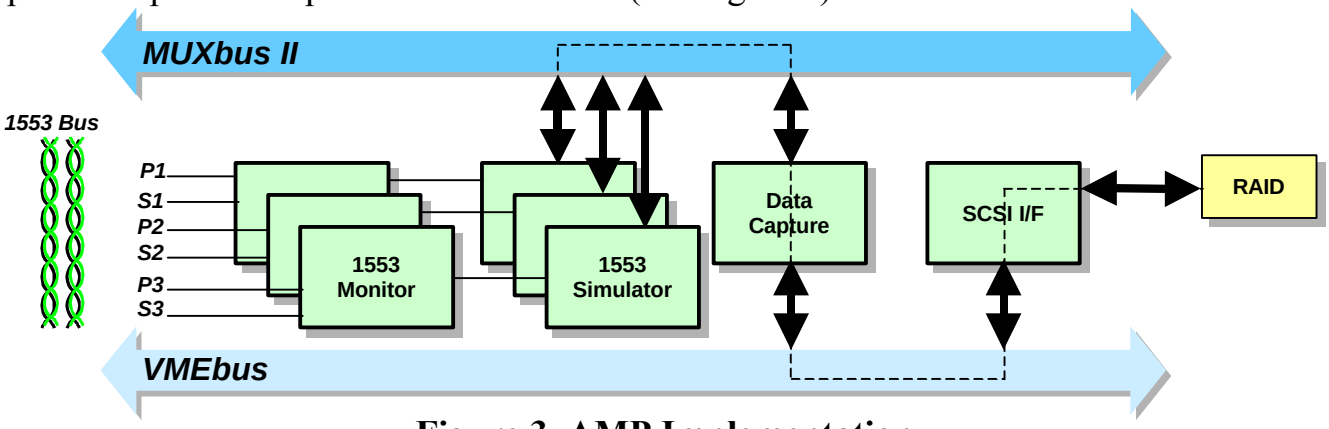

Figure 3. AMP Implementation

By nature, AMP configurations allow all processors to perform on their own data concurrently, thereby eliminating the latency generated by processor contentions. Additionally, AMP systems support the running of differing operating systems concurrently, thereby imposing fewer restrictions on component and software selections. Streaming tasks can be handled by dedicated chip sets and software suited specifically to given tasks, while staged data flow applications are most often handled by dissimilar hardware and software. A software example would be using a UNIX-based operating system to control graphical user interface functions such as setup, control, and data

display, while using an embedded operating system such as VxWorks to handle data flow, bus arbitration, and overall performance management tasks. In this way, tasks and processors can be divided, where one processor handles real-time interrupt processing and control, one handles GUI control, and another handles archiving.

DATA-DRIVEN ARCHITECTURE

In typical telemetry ground station processing architectures, data-driven designs are used as a baseline methodology that is "wrapped" around a system bus as the primary data flow path (see Figure 4). The necessity for this is validated by the timing characteristics of the reconstructed data, especially when the data is derived from multiple sources with different time bases.

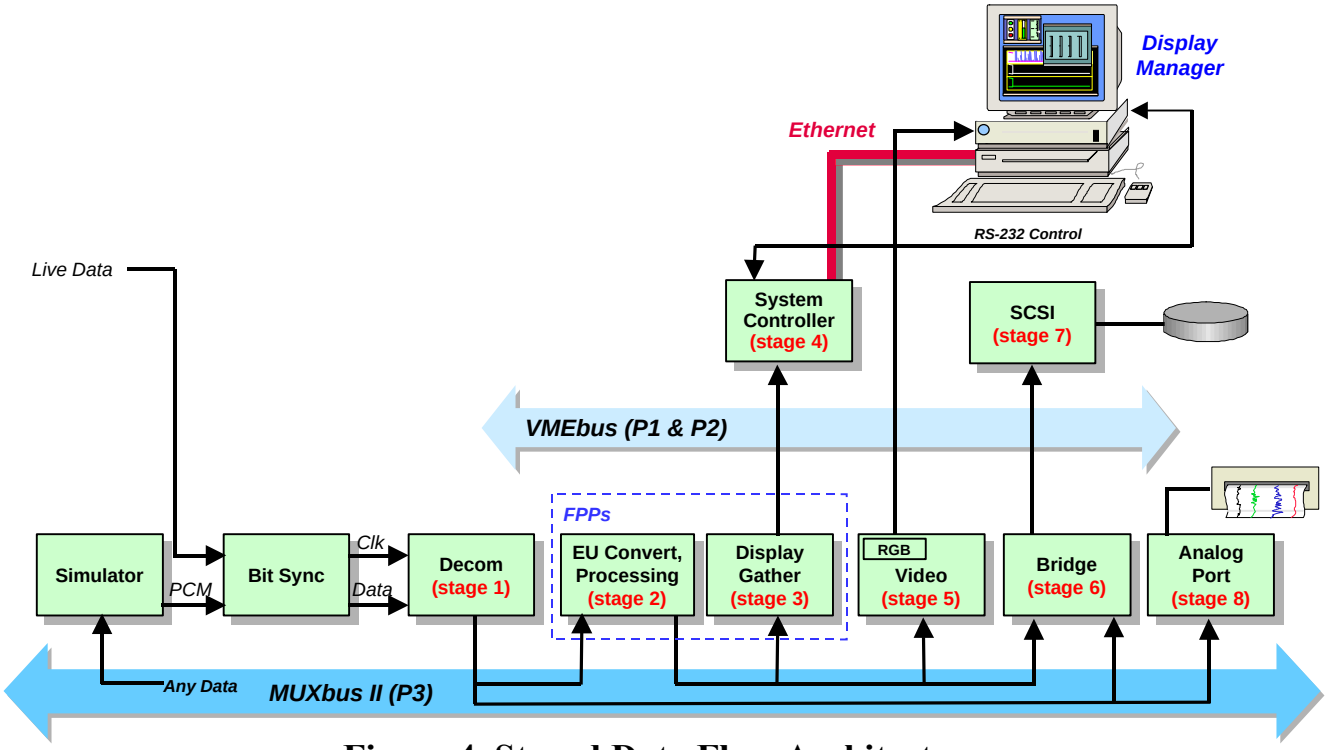

Figure 4. Staged Data Flow Architecture

Using a scalable multi-processor architecture permits additional computing and input/output resources to be added as required. In addition, a distributed data-driven architecture guarantees deterministic data merging and processing.

What's more, if the multi-processing architecture uses a broadcast mode arbitration scheme, independent processors and I/O controllers can accomplish a host of tasks without affecting the throughput of the bus. If you use the example of the system pictured in Figure 4, you will see that all the data is first processed at stage 1 (decom). The data is then passed on in stages 2, 5, and 8 to be processed independently and concurrently. Finally, the data is processed in stages 3, 4, and 7.

Staged data flow architectures are currently best served by RISC processors. These are defined by applications that take in a flow of data, process that data, and then move the refined data on. RISCs make good data flow processing engines. They have fast cycles, they can process data on each cycle, and they have regular wide bus interfaces. They can also be pipelined and easily process streams of data. The current fourth-generation RISCs are superscalar machines, i.e., they can effectively issue multiple instructions at each pipeline clock cycle. Leading-edge RISCs can actually sustain three or four instruction executions per cycle.

STREAM COMPUTING ARCHITECTURE

Stream computing refers to handling data from sources that are typically unrelenting and require processing in real time. Typically, these applications have been focused on high-rate sensors like radar, sonar, and medical data such as MRI and CT. Current telemetry sources that have 'stream' or unrelenting data sources include imagery from low earth orbiting (LEO) satellites and telemetered sensor data as well as video and other imagery. An example of a stream computing architecture is shown in Figure 5.

By definition, stream computing requires the transfer of high-volume data. Breaking the data up into packets increases latency and presents ordering problems (additional latency) during reconstruction. Solving the latency problem adds complexity, which in turn increases the size of the solution. Packet-type telemetry data in a high-rate single-stream application is one area in which SMP-type architectures excel (see Figure 6). Managing the breakdown of the data to its smallest processed component and taking care of the network administration is done with high efficiency in a multi-processor environment running under a single operating system.

In highly demanding stream computing applications like radar or medical imaging processing, a flow of data constantly enters the system from external sensors. These sensors are typically high rate and similar in type and sample rate. The data must be processed in real time and then sent on to be processed further and displayed or used to stimulate another system. Systems in such scenarios combine the best of scalar and vector processing in a heterogeneous mix of multiple processor types, including digital signal processors (DSPs), RISCs, and custom ASICs. More often than not, these solutions are based on switched fabric or ring topologies.

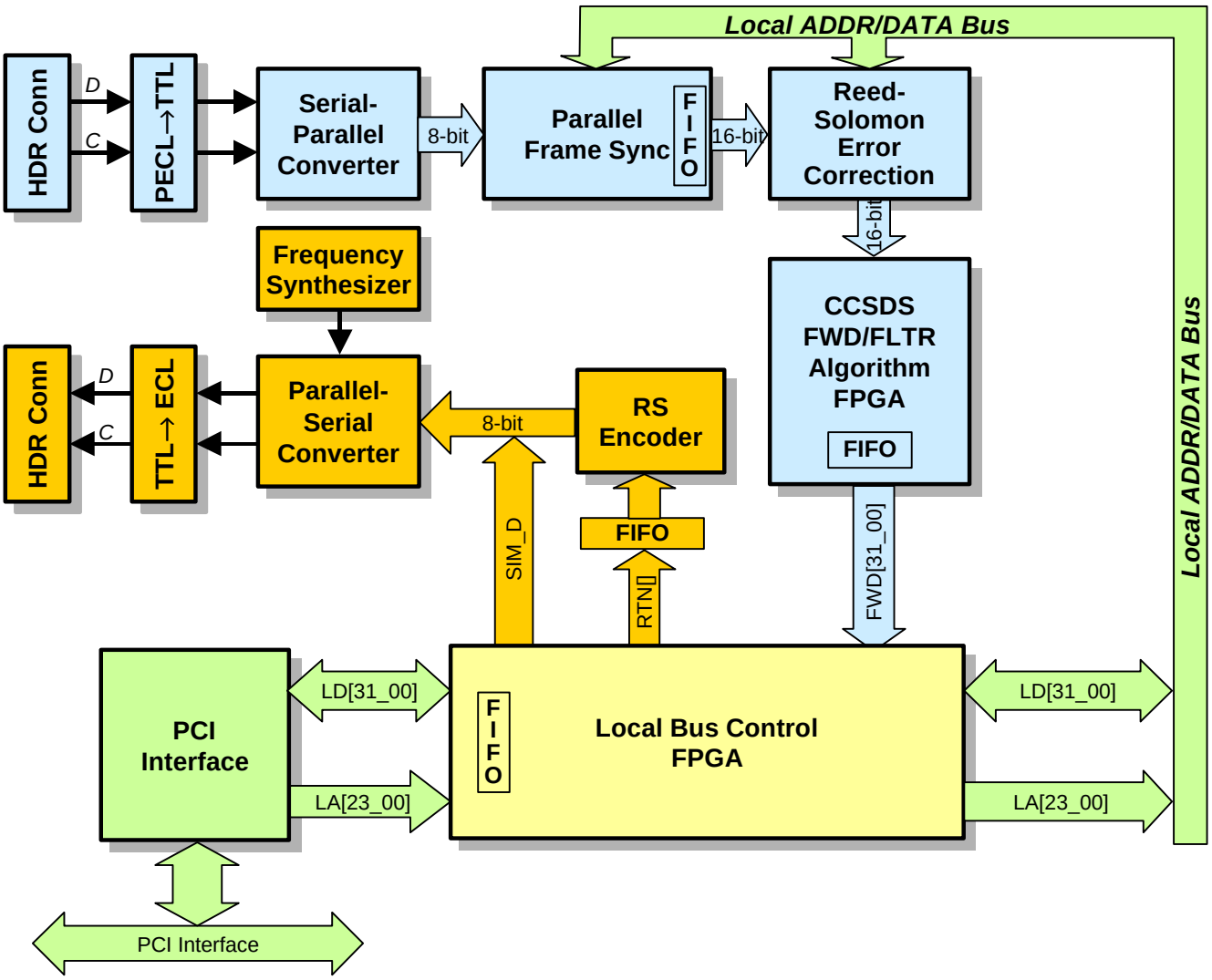

Figure 5. Stream Computing Architecture

In the telemetry world, stream computing applications are more likely to 'mix' sensor types and rates across a wide variety of hardware and platforms. Processing is slanted toward scalar operations like table look-ups, jumps, and conditionals. If there is a need for vector tasks like signal filtering, it is done in specialized single-board computers (SBCs) that are typically DSP-based. The need to retain signal conditioning flexibility and a modular architecture outweigh the requirements for homogenous architectures. The net result is that when multiple data sources are input and have to remain time coherent, system buses are a more deterministic solution than switched fabric or ring topologies. The main reason is that complexity becomes a limiting factor in real-time scalability. Latency and software overhead become issues to overcome as opposed to the more straightforward solution of a single system bus.

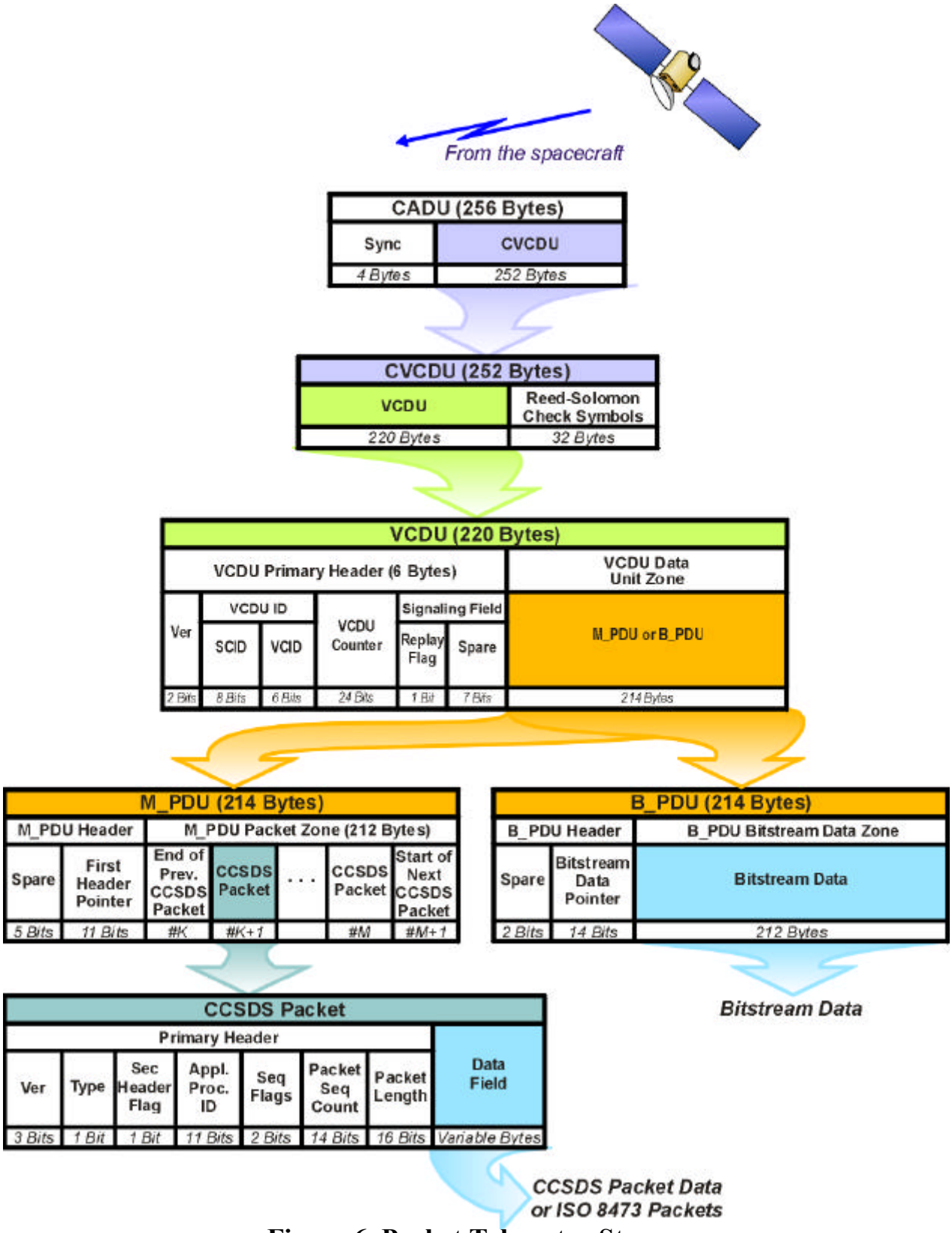

Figure 6. Packet Telemetry Stream

MIXED ARCHITECTURE DESIGN CONSIDERATIONS

Many issues need to be addressed when starting to review data processing requirements and selecting the best new system design architecture. The following are the most salient areas of concern for the system designer/integrator.

- How designing either symmetric or asymmetric multi-processing elements into future telemetry acquisition systems affect concurrent data operations and processing tasks.
- The effect of deciding what type of arbitration scheme will best handle the flow of data.
- The advantages and disadvantages of designing systems that follow heterogeneous processing architectures and structured arbitration schemes.
- The effects bus, processor, and network selection can have on hardware selection during the design process, as well as their impact on future growth and migration paths.
- The benefits and drawbacks to system performance parameters, such as time coherency and data latency, as they relate to stream and data flow architectures.
- The effects on cost and schedule of implementing a new system design based on early decisions about commercial products versus in-house design.
- How the change in current telemetry system design trends is partially defined by changing network paradigms and the interrelationship between processor types and their network counterparts.
- Design considerations given in choosing a PC architecture in lieu of a larger VME system and the trade-offs associated with such decisions.

MIXED ARCHITECTURE SYSTEM

The following section demonstrates a current state-of-the-art mixed architecture processing system. This system is used as an airborne acquisition platform for acquiring, processing, displaying, and archiving multiple types of data in real time (see Figure 7). The data types can be input from multiple sources and have varying physical properties. The system acquires and processes data at over 32.4 Mbytes/sec. Table 2 shows the variety of input data stream types, interfaces, and aggregate rates.

Table 2. Data Types for Mixed Architecture System

Type	Quantity	Rate	Aggregate Rate Mbps	Interface Type
Video	2	124.8 Mb	249.6 Mbps	DCM / VCC
RS-232	7	287 kb	2.09 Mbps	High-Speed SIO
RS-232	2	2 Mb	4 Mbps	High-Speed SIO
MIL-STD-1553B	3	1 Mb	3 Mbps	1553 Bus Monitor
ARINC-429	2	100 kb	.2 Mbps	ARINC Monitor
RS-485	2	100 kb	.2 Mbps	High-Speed SIO w/ IP
Analog	5	35 kHz	.175 Mbps	Analog Ports Module

Since the data types vary substantially in both rate and type, an AMP-type system with a broadcast mode arbitration scheme, independent processors, and I/O controllers will work the best. The video data is streaming in nature and varies with image size. All other data sources are synchronous.

The streaming video data is fed into a 512 MB FIFO buffer on a Data Capture Module, allowing the system to handle up to 40 seconds of high-rate bursting data. This is possible because the hardware handling this process runs independently of the hardware processing the other types of data. The data stream is placed into local memory and is transferred via DMA to the SCSI controller under the supervision of a 'state machine' or logic that is resident on a local field programmable gate array (FPGA). Using an FPGA saves board real estate and power by not having to control a microprocessor or DSP locally. Eventually, all the data is fed through an Ultra SCSI interface to be archived on a RAID subsystem. Time coherency is maintained by applying time tags to all data as it 'hits' the bus. The latencies are easily defined for each I/O type of interface.

Data processing, such as engineering units conversion, bit manipulation, and derived measurement calculations, is done in the Field Programmable Processor (FPP). The FPP is designed with a state-of-the-art RISC processor that runs the VxWorks real-time operating system. This provides fast and predictable interrupt responses like:

- 10 microsecond interrupt latency
- 20 microsecond raw context switching
- 50 microsecond complete interrupt service time

This processor allows fine-grain processing on every data sample when required and runs concurrently with other I/O operations. The system's utilization of an embedded real-time operating system to control time-critical functions, while simultaneously running a UNIX-based operating system for display and control functions, is a hallmark of an AMP-type system architecture.

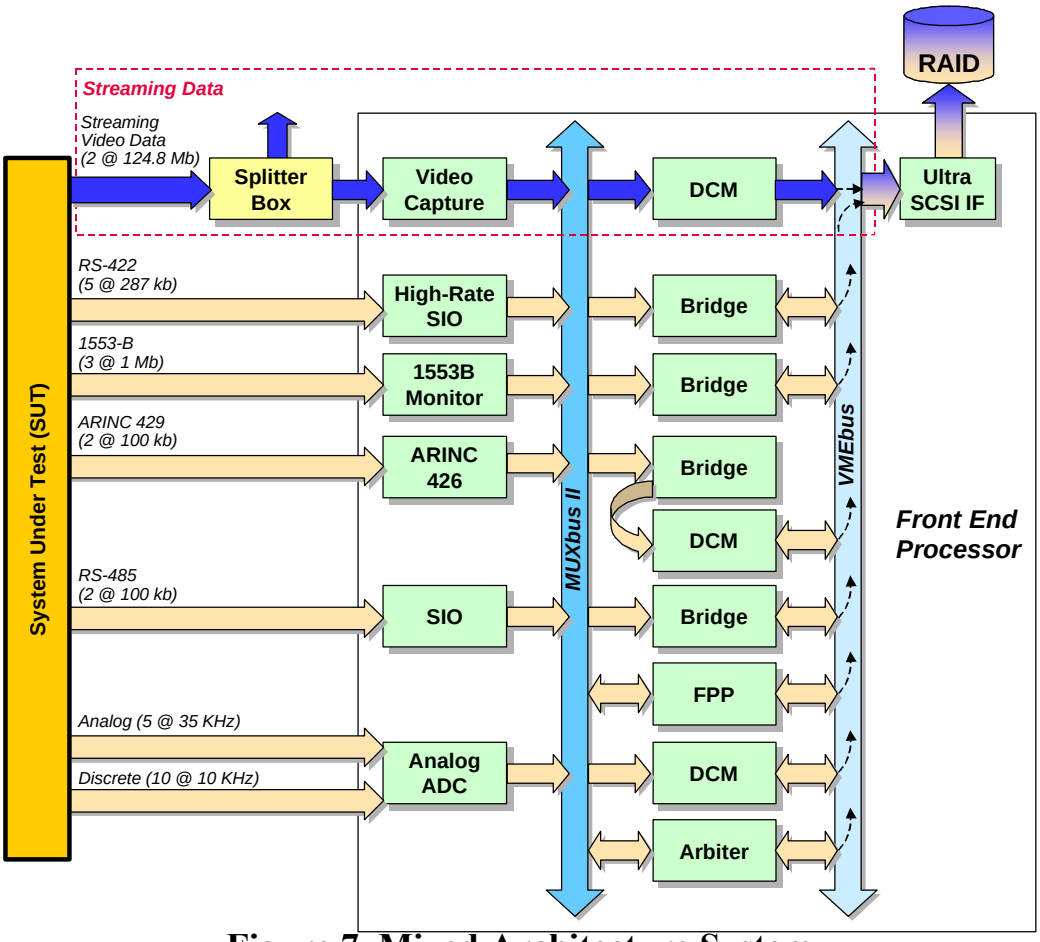

Figure 7. Mixed Architecture System

CONCLUSION

Data sources in the telemetry world are changing. Packet telemetry, high-rate imagery, and asynchronous data sources are now the norm instead of time division multiplexed data streams. When there is a requirement to mix these types of data streams, a scalable asymmetric multi-processing system that shares a high-speed computer backplane is the most efficient architecture. Using a high-speed bus as the primary data flow path allows a high degree of time-determinism and defined latencies. The integration of multiple data type-specific processors allows for concurrent operations on separate data sets, thereby eliminating the latency generated by processor contentions. Additionally, if a broadcast mode arbitration scheme is used, concurrent processing tasks will not affect bus throughput.

By contrast, if the requirement is to process, archive, and route a single telemetry stream from a high-rate streaming source (such as imagery from a satellite), then a symmetric multi-processing type of architecture running under a single operating system will provide a more cost-effective solution, while retaining the performance necessary to accomplish the task.

REFERENCES

[1] Kopchinski, Greg, “Asymmetric Multiprocessing for CompactPCI,” RTC VolumeVI, #10, October 1998.

[2] Wexler, Marty, “Using State-of-the-Art Technology to Provide Distributed Processing in an NT Environment,” European Telemetry Conference, June 1998.