NEW APPROACHES TO THE IMPLEMENTATION OF EMBEDDED TELEMETRY PROCESSING SYSTEMS

Bill Eckman, Dana Irvin, Kirill Lokshin, Amit Puri
Ingenicomm Inc.
14120 Parke Long Court #210, Chantilly, VA 20151
beckman@ingenicomm.com

ABSTRACT

The conventional architecture of modern telemetry and signal processing systems typically relies on some combination of specialized hardware and firmware components integrated with commercial rackmount computer platforms running off-the-shelf operating systems. The evolution of the modern system-on-a-chip (SoC) architecture, which tightly couples traditional field-programmable gate array (FPGA) elements with high-performance embedded microprocessor cores, offers an alternative to this conventional approach. When effectively utilized, the SoC approach can provide several advantages, including reduced system size and power, increased system security, and lower lifetime operating costs.

This paper discusses the benefits offered by embedded SoC architectures for the implementation of satellite telemetry processing systems, and presents an example design that demonstrates advantages with respect to security, reliability, maintainability, footprint, and cost.

INTRODUCTION

The conventional modern architecture for satellite telemetry and command processing systems began to emerge in the mid-1980s with the development of systems like NASA's TCDS (Telemetry and Communications Data System) and NCPS (Network Command Processing System), which used MULTIBUS computer cards and peripheral interface cards in much the same way that modern computers do. These early systems used Motorola processors running at clock frequencies around 50 MHz and using 4 MB of system memory.

The next generation of telemetry processing equipment began to be developed in the early 1990s, taking advantage of the explosive growth in the personal computer market and using increasingly powerful commercial x86-based computer systems to host modular telemetry processing software. Like the MULTIBUS and VME systems before them, these systems used plug-in architecture buses like ISA and PCI, which allowed equipment manufacturers to design custom interface cards to perform tasks like demodulation, bit and frame synchronization, serial data processing, and IRIG time signal processing. Industrial 19-inch rackmount computer server systems became the common form factor, with the average system footprint usually being around 3U to 4U.

Over time, the move towards TCP/IP network communications as a common mode of data transfer between systems has reduced the need for custom hardware interfaces. Telemetry processing systems that only use TCP/IP network interfaces are often marketed as software-only implementations, but the software is still running on the same rackmount server architecture as before (although the size of the system is typically smaller, around 1U to 2U).

DISADVANTAGES OF THE RACKMOUNT SERVER ARCHITECTURE

Most modern telemetry processors still use the rackmount server architecture more than 25 years after its initial development. However, this architecture has a number of disadvantages, which include security risks, operating system issues, size and power concerns, increased cost, sustainability issues, and channel density restrictions.

Security risks are becoming more prevalent and costlier to prevent. The most common security threat is malware targeted consumer and enterprise operating systems, such as Microsoft Windows or Linux, which are widely used by modern telemetry and command processing equipment. Malware is most commonly introduced through email, network file exchanges, or portable storage devices; most organizations apply system hardening and security controls in an effort to prevent this type of malicious activity, but the prevention comes at a great cost in terms of labor and the continuous monitoring, upgrades, and additional software required to maintain security scanning tools and anti-virus programs. Anti-virus software and other security tools also reduce the performance of the systems on which they are deployed and can interfere with normal operational activity.

Operating systems designed for consumer and enterprise applications are often optimized for their most common expected use; in most cases, this is not real-time data processing, but rather the use of interactive applications such as word processors, games, and Internet browsers. Data transfer functions are optimized for intermittent, user-generated traffic, which is not a typical use case for telemetry processing that more commonly transfer data at constant throughput rates. Further, the system processor time required to maintain throughput rates is difficult to control, and processor time available for applications is often superseded by other, higher-priority processes running at the kernel level. This can result in reduced throughput rate or data interruptions if not sufficiently compensated for by the telemetry processing software; but the need to over-design the software to compensate for deficiencies in the operating system results in inefficient software and increased system resource usage.

Commercial rackmount servers are typically not designed with size, weight, and power efficiency in mind, and typical deployments of multiple servers translate to wasted power, a need for additional cooling, and increased rack space. Typical power supplies for rackmount server systems range from 500W to 1000W, and a typical system will use ¼ of the rated power just sitting idle, without any processing software running. Some power savings can be achieved by placing these systems in sleep mode when they are idle, but this is almost never acceptable in operational environments that require high availability.

Channel density has always been an issue with rackmount computer systems. A large factor in this is physical space for connectors on add-in slots; the space available per backplane slot has been the same since the inception of the rackmount server. Higher density connectors can increase the number of interfaces possible, but higher density connectors are difficult to design into a satellite ground system that typically uses coaxial cabling.

Maintainability becomes an issue when using commercial off-the-shelf (COTS) computer systems. Commercial computing systems typically become obsolete in three years or less. This obsolescence is designed into the commercial marketplace, and operating systems and equipment older than 3−5 years are often no longer produced or supported. This is incompatible with a typical satellite ground system lifecycle, where a typical satellite takes 3−10 years to develop and is expected to have a lifetime of 10 years or more. Some satellite operators attempt to remediate this problem by making purchasing stockpiles of end-of-life components and larger number of spares, but this is often ineffective in the face of software and operating system obsolescence. An operating system that is no longer supported by the manufacturer is a security threat. Further, updates to telemetry and command processing software are typically only supported on current operating systems, so any software updates also require upgrades to operating systems, and ultimately to the underlying computing hardware. As a result, a typical ground system is forced to undergo several hardware and software refreshes throughout the operational life of the satellite.

AN EMBEDDED ARCHITECTURE APPROACH

Using modern SoC devices, it is possible to design embedded telemetry processing equipment with a greatly reduced form factor. SoC devices tightly integrate microprocessors and FPGAs into one physical device, allowing the system designer to select the most appropriate technology to accomplish a task. For example, one might use FPGA resources to do perform forward error correction (FEC) decoding, such as BCH or LDPC, while using the microprocessor to handle packet processing and network data transfer.

Figure 1, below, demonstrates the key differences between the conventional rackmount architecture and the embedded SoC architecture.

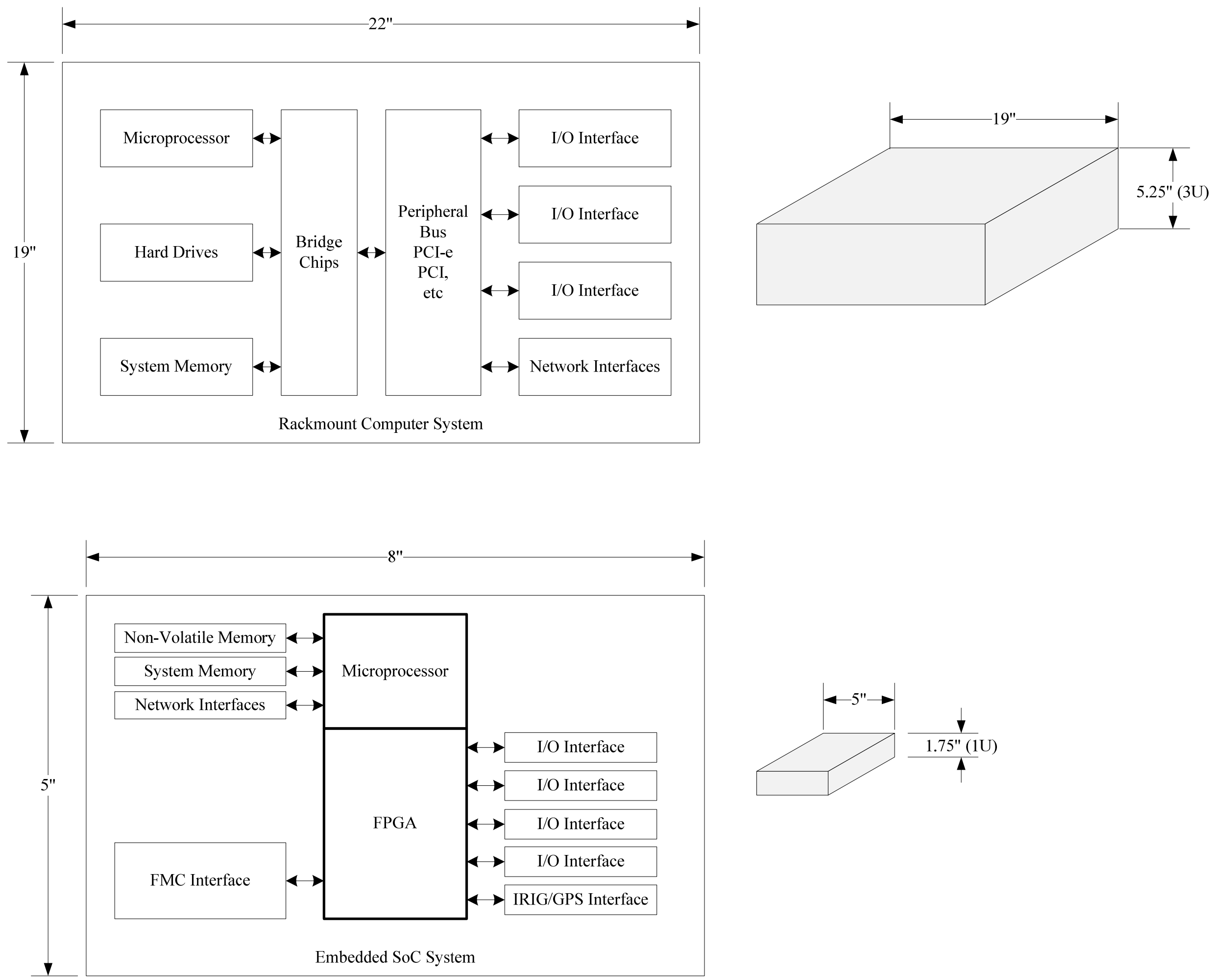

Figure 1: Architecture Comparison

The top portion of Figure 1 shows a typical rackmount server telemetry processor with a 3U chassis. The system is configured with three serial interface channels using PCI or PCIe bus interface cards. The bottom portion of Figure 1 shows an embedded version of this architecture. The system footprint has been reduced from a 3U sized system with three channels to a 1U sized system with 4 serial channels. The width of the embedded system is so small that three embedded systems could be mounted in a single 1U rack space.

Figure 2, below, shows a representative implementation of a SoC-based embedded telemetry processing system, the “Rattlesnake”, designed by Ingenicomm in 2015.

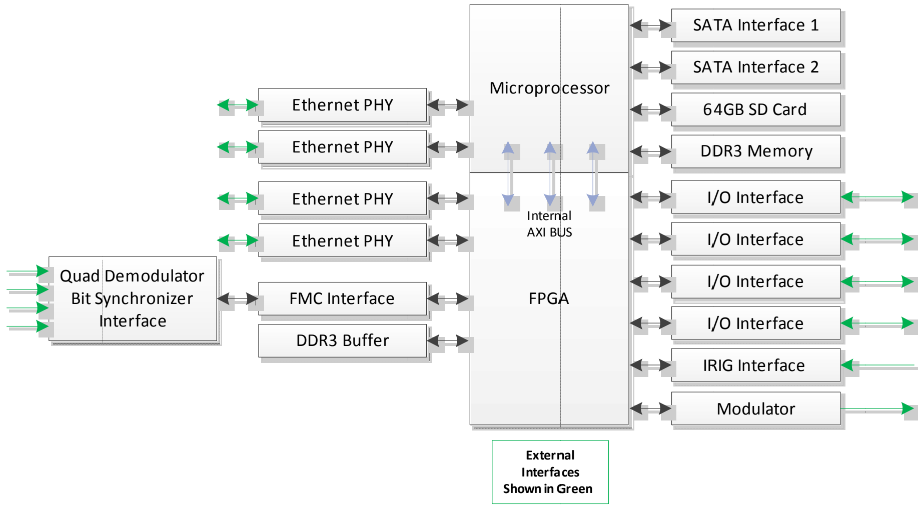

Figure 2: Rattlesnake Telemetry Processor Schematic

The embedded design allows multiple interfaces (including four RJ-45 Ethernet ports, four coaxial demodulator/bit synchronizer interfaces, a modulator output, four full-duplex serial interface ports, and an IRIG timing input) to be packaged into a form factor that allows three of systems to be mounted in a single 1U rack space.

In order to allow for scalability to support a greater number of independent signal interfaces as well as configuration flexibility, the Rattlesnake main board (shown in Figure 3) includes a mezzanine interface that is compatible with the FMC VITA-57 standard, which supports a wide variety of commercially available cards, such as the four-channel analog interface shown in Figure 4.

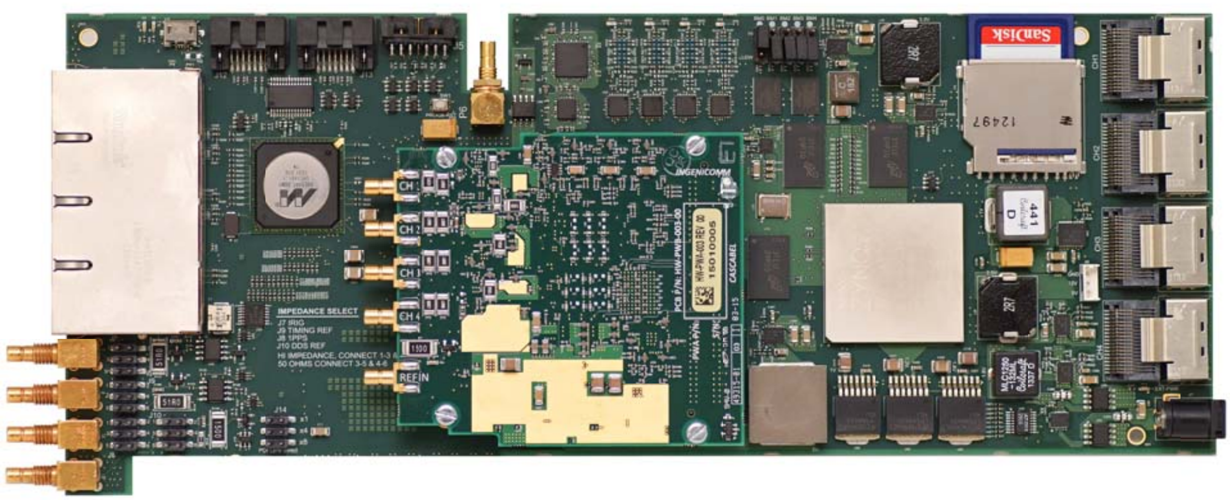

Figure 3: Rattlesnake Telemetry Processor Main Board with FMC Card Mounted

Figure 4: Four-Channel FMC Analog Card

This four-channel analog interface, which is capable of simultaneously receiving and processing four analog signals, is designed for subcarrier and IF demodulation and bit synchronization applications between 0 Hz and 150 MHz. Other commercially-available FMC cards provide a range of capabilities which includes analog-to-digital conversion, digital-to-analog conversion, clock distribution, signal level conversion, and many others.

ADVANTAGES OF THE EMBEDDED DESIGN

The embedded telemetry processor design provides a solution to many of the problems associated with the rackmount server architecture used by most common telemetry and command processing equipment, including security risks, operating system deficiencies, size and power concerns, lifetime maintainability issues, and channel density restrictions.

Security risks are reduced by eliminating the consumer or enterprise operating system commonly used in telemetry processing equipment, and replacing it with an embedded version of the Linux kernel that is virtually immune to common security risks. The use of an embedded OS allows the telemetry processor to be classified as a network appliance, similar to a network switch or router, which allows the configuration of the device to remain more static, compared to a consumer or enterprise OS that has updates released on a daily basis. The network appliance designation also reduces the need to install third-party security programs.

In addition, the use of an embedded operating system allows the system to be configured with only the specific processes and programs required for processing satellite telemetry and command data. Unlike telemetry processors based on consumer or enterprise operating systems, an embedded telemetry processor cannot be used for playing games or web browsing. There are no unnecessary programs or processors competing for CPU time; the only job of the device is to process telemetry and command data, and all data throughput can be optimized for that purpose.

Size, weight, and power are also reduced. The example Rattlesnake system shown can be powered by a 50W power supply, while total size has been reduced and channel density has been increased, as shown in Figure 1. This translates to large rack space savings in environments where space is a premium, and reduces the per-channel power requirements by a factor of 10.

The embedded architecture also provides increased maintainability, as the design is not based on consumer-grade components with short obsolescence times. The components selected for the example Rattlesnake design have a minimum design support life of 10 years, and lifetime buys can be used to extend the support life of the telemetry processor an additional 10 years or more. Further, in terms of sustaining software support, the embedded design is highly advantageous, as software is easier to maintain and support when the system configuration is more static, and the open-source OS kernel is not subject to proprietary restrictions and forced obsolescence.

CONCLUSIONS

The rackmount server architecture has become the standard architecture for implementing satellite telemetry and command processing equipment. When this architecture was first developed more than 25 years ago, it had numerous advantages in terms of size, processing power, and ease of development in comparison to predecessor architectures. However, over time, this architecture has shown weaknesses in terms of security, maintainability, and cost.

An embedded architecture for telemetry processing allows users to achieve reductions in system footprints, increase security, enhance reliability and maintainability, and reduce lifetime system cost. The smaller size of the embedded telemetry processor reduces the rack space required, decreasing the real estate required to support a mission. Significant reductions in power and cooling requirements lead to cost savings in power consumption. Improved security capabilities result in cost savings in terms of implementation and monitoring, while better maintainability allows users to avoid costly system rebuilds during the lifetime of a mission.