FUNCTIONAL COMPONENT APPROACH
TO
TELEMETRY DATA CAPTURE SYSTEMS

Steve Sabia, Sarah Hand
Data Systems Technology Division, Code 520
Mission Operations and Data Systems Directorate
NASA, Goddard Space Flight Center
Greenbelt, Maryland 20771

ABSTRACT

To support the telemetry data rates and meet the needs of telemetry data users in the next decade, telemetry data capture systems will have to be radically different from today’s systems. At Goddard Space Flight Center, the Mission Operations and Data Systems Directorate is developing the capability to build user specific data capture systems from a library of high performance hardware and software elements that satisfy standard data capture processing requirements.

One or more telemetry functions are encapsulated in a single standard open bus system (e.g. VME, Multibus II, NuBus etc.) with supporting software to form a user data capture system. Each subsystem module (card or board) includes a local microprocessor supplying on board intelligence and programmability for changing requirements. Many of these subsystem designs include custom very large scale integration (VLSI) components to increase speed while minimizing cost and size. A standard hardware and software interface to each card subsystem is employed to simplify system integration in the open system environment.

INTRODUCTION

Imagine the following scenario: A scientist needs a telemetry system to capture data from his experiment aboard the Space Station, perform error correction, and extract and process “quick look” packets of science data. From his office workstation, he logs on to the Telemetry Component Library Processor at the Goddard Space Flight Center (GSFC). Information on the capabilities and availability of all telemetry processing components, both NASA specific and user specific, is housed in the system’s database. Through interaction with the Processor, the experimenter defines his data capture needs on a high level. The intelligent Processor resolves any design ambiguities through queries of the user and the detailed component information stored in the database. Finally, the user specific system is assembled and installed in a large data handling facility, or provided, fully tested, to the user for remote operation. While this scenario is far from reality today, early efforts to prototype a functional component architecture may help make it a reality in the future.

A functional component approach to telemetry data capture systems is being developed in the Mission Operations and Data Systems Directorate at GSFC which could support telemetry system needs in the next decade. The goal is to develop a library of functional modules which perform common telemetry functions. These components are then available off the shelf, to be assembled on a standard platform, in a user specific configuration. The one time effort to build a component, and its subsequent reuse in many systems is an important advantage of this approach. Smaller, faster and less expensive systems will make telemetry data capture capabilities available to a wider range of organizations and budgets. This opens up the possibility of decentralized telemetry data capture, where functionality and performance can be tailored to meet particular customer needs.

Some common telemetry data capture functions include NASA Communications (NASCOM) Block processing, frame synchronization, Reed/Solomon decoding, packet processing, and virtual channel sorting/multiplexing. One or more of these functions are modularized into a VME card(s), along with supporting software, to make a functional module. Generally, a data capture system is made up of a number of these modules combined in a standard open bus system with the necessary system support hardware and software.

CURRENT SYSTEMS VS. FUTURE NEEDS

Current telemetry systems are generally software intensive and application specific machines. They were built using medium scale integration (MSI) technology for special front end data capture with Mainframe CPUs used for actual data processing. These systems adequately support the data rates needed for many current missions. In contrast to the functional component approach, no standard platform exists for these data capture systems. Instead, completely separate custom systems were built for each facility despite the similarity of functions they perform. Many of these systems exist within NASA today, and many more will be needed with the advent of new missions and the Space Station era.

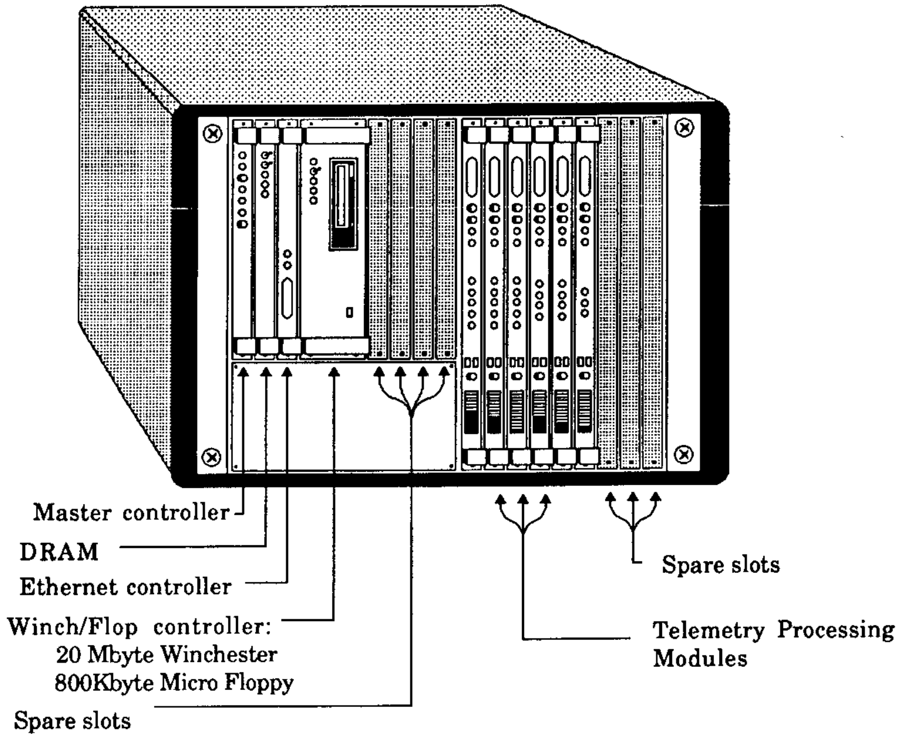
The functional component approach seeks to meet the needs for telemetry data capture of both current and future missions. The standardization of telemetry functions will facilitate this approach. Recommendations for advanced telemetry systems in the Space Station Era are being developed by the Consultative Committee on Space Data Systems (CCSDS). These recommendations include standards such as data formats, and error detection and correction codes. The standard data format will be the CCSDS frame with variable length packets contained inside. The error detection and correction scheme will be a Reed/ Solomon code, and data rates will range up to 300 Mbps.

GENERAL DESCRIPTION OF A COMPONENT BASED SYSTEM

Each component built telemetry system is a hierarchical combination of commercial hardware, NASA specific custom hardware and multitasking software. Discrete modules on hardware and software, with consistent interfaces, allow complex data capture systems to be configured to a user’s specific needs. The following sections describe the hardware and software structure of a component based telemetry data system prototype which has evolved over the last three years at the Goddard Space Flight Center.

SYSTEM LEVEL HARDWARE

The standard hardware platform for this prototype is a multiprocessor architecture with two bus systems housed in a triple height (9U) VME card cage (see figure #1). A master processor is dedicated to system control, while the other processors provide an interface to the master and local intelligence on the telemetry data processing modules. Commercially available VME cards are used wherever possible to minimize cost and engineering time with custom designs limited to NASA specific functions.



STANDARD HARDWARE PLATFORM

FIGURE 1

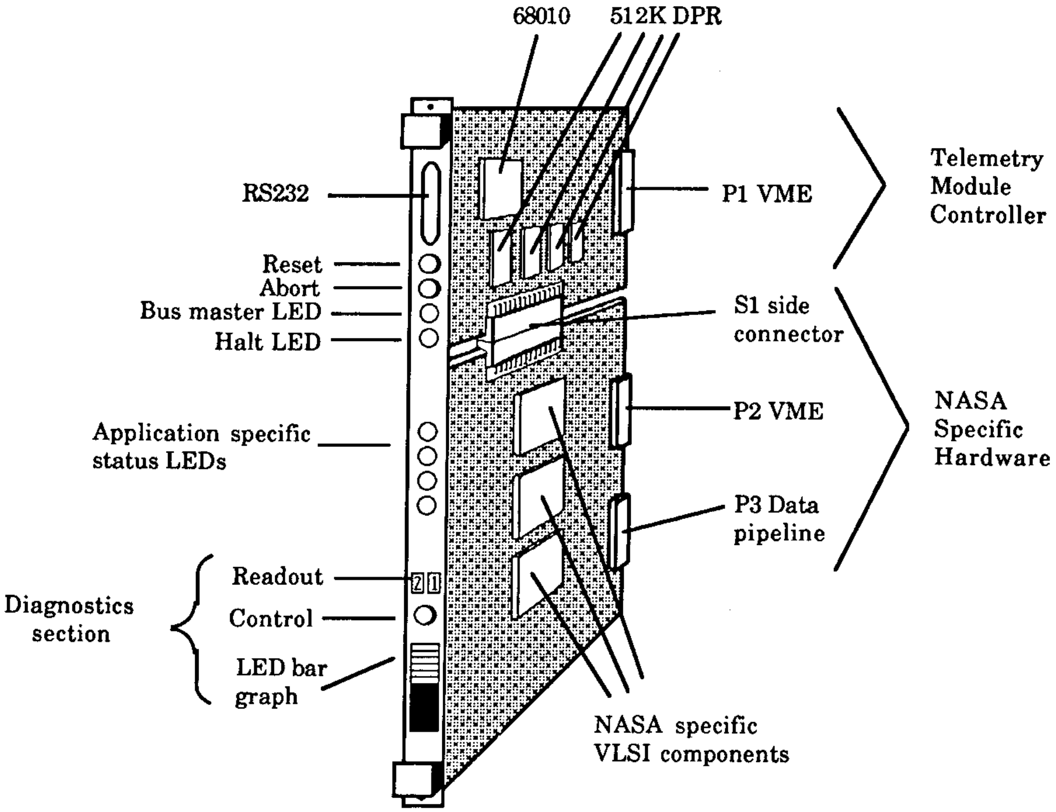
A telemetry processing module is a combination of NASA specific hardware and a commercially available single board computer which together form a triple high VME card. This single board computer contains a Motorola 68010 processor and 512 Kbytes of dual ported RAM in a single height VME card format. Other commercial hardware consists of a processor card containing a Motorola 68020 microprocessor and one Mbyte of RAM, a two Mbyte DRAM card, a Winchester/Floppy Controller card to drive the 20 Mbyte hard disk and 800 Kbyte microfloppy, and an Ethernet card. Dual bus systems are used to separate the telemetry data stream from control and status data. The dedicated "telemetry data pipeline bus" is used for the high speed transfer of telemetry data between processing modules, while the VMEbus is used for passing control and status information between the system master and the modules.

With the above configuration, a great deal of processing power remains available in the VME environment. Additional user specific functions, such as "quick look" of specific data sets, could be custom developed by users or provided by commercially available VME components, to meet their particular application needs. These new modules could then be added to the library of available telemetry data system modules.

SYSTEM SOFTWARE

The software to manage this system is also designed as a generic shell on which a user develops his particular application. Software functions common to all systems include the setup of all hardware and software for processing a specific data stream, the reporting of status about the processing, and module level testing and diagnostics. In the component built system, the master's software must be flexible enough to handle any card placed in the system. This is accomplished by a highly modular design where generic software is linked with card specific software to support the hardware components included in a particular configuration.

Each system is a set of microprocessors, with one acting as the system master controller to command and monitor each telemetry module's local processor, which in turn controls and monitors the module's accompanying NASA specific hardware (see figure #2). Each processor's software runs under the control of a commercial realtime operating system. Using its multitasking services, the software is partitioned into a number of tasks which together implement the functions of the particular module. The master's software interfaces with the system operator and the individual telemetry processing modules, to set up the system and monitor its activity. These actions are supported by peer software running on each module.



TYPICAL TELEMETRY PROCESSING MODULE

FIGURE 2

MASTER CONTROLLER

A commercial realtime operating system provides the basic software development and operational environment. The operating system supports many processor cards which incorporate the MC680XX series processors and will accommodate upgrades to advanced processors as they become available. As the VME system controller, the master controller drives the system clock and reset lines, and serves as the bus arbiter. To implement its functions of controlling and monitoring the system, the master controller software consists of an operator interface task, a driver task for each module in the system, a system status task, and tasks used for system initialization, debugging and maintenance.

The tasks work together in the following manner. The operator interface has one or more setup and display pages for each module on which setup data is entered and status data is displayed. The information on these pages is packed into messages that are transferred to/from the corresponding module driver tasks via intertask communications. The module drivers send the data to their assigned module using the interprocessor communications layer software. The status task gathers individual module status blocks, on a timed cycle, and builds a system status block for the operator interface task to display. At the core of the control software, interprocessor commands and data are passed between the master controller and the module controllers. Each command reflects an action that the master controller wants the telemetry module to perform, for example, the initialization of a chip set based on operator or scheduled local reference table inputs.

TELEMETRY PROCESSING MODULE HARDWARE

Each telemetry processing module implements NASA specific data processing functions in a unique triple height (9U) configuration. A module consists of a single height (3U) processor card connected to a double height (6U) custom hardware card via a unique side connector (see figure #2). All three connectors are used on the card cage backplane: the standard VMEbus on PI and P2 and a custom “telemetry data pipeline” bus is on the P3 connector. This configuration is adequate for many telemetry functions with rates up to 20 Mbps. For still higher rates or more complex functions, custom or new CPU designs could replace the commercial single height CPU as long as the generic system level protocol is maintained. In this way, high level integration is indifferent to various performance levels required by individual users.

Application specific VLSI chips in combination with various configurations of buffers, FIFOs, and other control logic make up the telemetry module’s custom hardware. To achieve a reduction in size and an increase in throughput, many data processing functions previously performed in software are implemented in hardware. NASA specific VLSI components handle many of the standard telemetry processing functions. For example, correlation, NASCOM block processing, frame synchronization, and statistics accumulation are incorporated in present VLSI chips. Complex erasable programmable logic devices are also extensively used to implement local hardware functions such as I/O control.

High throughput rates are achieved by using a dedicated telemetry data path (a custom design on the P3 connector) for transfer of data blocks and frames, separate from the VMEbus. Telemetry data entering a module on the data pipeline is processed by VLSI components on board with very little software intervention. These devices interrupt the local processor at key points in the data flow triggering software service routines. Setup and status data are passed over the VMEbus under software control, where speed is not as critical.

Each module performs extensive power up self diagnostics which include a check out of the local data pipeline and the components surrounding it. Each module includes a VLSI Test Pattern Generator Chip, also designed at Goddard, which simulates a data stream through the data pipeline. Command driven software triggers and reports on each module’s self-testing.

TELEMETRY PROCESSING MODULE SOFTWARE

Each module's software expands on the capabilities of the accompanying hardware; together they implement the module's specific functions. The software consists of the following: interprocessor communications task, card level status gathering task, data processing tasks and interrupt handlers, and a "debug" task.

The interprocessor communications task executes the module's specific command set. The commands allow the module's custom hardware and software to be set up and monitored by the master. The status task collects the card status from various hardware devices and software tables, formats a status block and writes it to dual ported RAM, where the master expects it. A status block may include information such as the number of frames input, frames in error, frames filtered, as well as information on the card's health and operational state. Realtime processing of the pipeline data is accomplished by a varying number of tasks which are typically interrupt driven. The number and design of these tasks is dependent on the hardware and the functions required of the module. Functions such as data filtering, routing and accounting are handled by these processes. The debug task localizes code that would normally be spread throughout the tasks of the module and also allows interactive setup and monitoring of the module's hardware and software without the need for a master processor.

CONCLUSION

To fully develop the science and application potential of future space activities, NASA's telemetry data systems must do more than simply meet specific technical requirements. They must provide for reliable, low cost, and modular systems which NASA and its user community can easily tailor in size and performance to particular needs. These systems must allow for growth and expansion in the years to come.

The new Telemetry and Command system (TAC), from which the component approach at GSFC was born, utilizes functional modules combined in the VME open system. This new system represents a major ground breaking effort. Many new techniques and technologies were developed during the design of this first component data system.

This functional component approach to telemetry data capture systems has been under development in the Mission Operations and Data Systems Directorate at the Goddard Space Flight Center for over three years. We now have a variety of NASA specific CMOs VLSI, components which perform standard telemetry functions (e.g. correlation, frame synchronization, NASCOM deblocking, etc.) and special support VLSI components such as the Test Pattern Generator chip. These components have been used alone or in combination to develop generic telemetry processing modules (cards) such as the

Synchronizer Module which has the capability to perform NASCOM Block processing and frame synchronization on a single VME board at up to 20 Mbps. An Emitter Coupled Logic (ECL) version of this card is now under development for support of data rates up to 300 Mbps. Other telemetry processing modules currently under development include virtual channel sorting/multiplexing, packet processing, and Reed/Solomon decoding. Telemetry processing modules to perform support functions, such as data simulation and time decoding, are also being developed in support of end-to-end system level functionality, testing and verification. An Ethernet hardware and software interface, a standard feature of this system, can be used to support control and status functions of a network based, data handling facility. Future enhancements include the adaption of NASA’s Transportable Applications Environment (TAE™) as the basic user interface for system development and operation.

As described in this paper, the design and development of VLSI components to perform standard telemetry functions, the use of one or more of these components to form functional modules, and the integration of these functional modules to form a functional component telemetry data system, demonstrates a prototype system which could provide the architecture to meet many of NASA’s future systems requirements.

REFERENCES

1. “DIF cost and Technology Evaluation Study”, NASCOM Task S224. Ford Aerospace and Communications Corporation, College Park, MD, December 1987.

2. “Talking to Spacecraft: NASA’s New Multi-Processing Telemetry and Command System”, Bus/Board Users Show and Conference, Marlborough, MA. Sabia and Homer, NASA, Goddard Space Flight Center, Greenbelt, MD, October 1987.

3. CCSDS 701.0-R-1: Recommendation For Space Data System Standards, Issue 1, Red Book. Consultative Committee for Space Data Systems. April 1988.

NOMENCLATURE

CCSDS	Consultative Committee for Space Data Systems
CMOS	Complimentary Metal Oxide Semiconductor
CPU	Central Processing Unit
DRAM	Dynamic Random Access Memory
FIFO	First In First Out
GSFC	Goddard Space Flight Center
I/O	Input/Output
Kbyte	Kilobyte

Mbps	Megabits per second
Mbyte	Megabyte
MSI	Medium Scale Integration
NASA	National Aeronautics and Space Administration
NASCOM	NASA Communications
RAM	Random Access Memory
TAC	Telemetry and Command
VLSI	Very Large Scale Integration
VME	Versabus Module Eurocard Format