PROCESSOR TECHNOLOGY OFFERS UNREALIZED FLEXIBILITY IN TOMORROW'S TELEMETRY GROUND STATIONS

GARY A. THOM
PROJECT ENGINEER
AYDIN MONITOR SYSTEMS
FORT WASHINGTON, PA.

ABSTRACT

Today's state of the art in semiconductor technology coupled with innovative computer architecture techniques can provide tomorrow's telemetry industry with advanced ground station capabilities. Computer systems have traditionally been used to process all of the telemetry data. As data transmission speeds increase, the computer system can no longer handle real time processing so preprocessors are being used to handle the additional computational requirements.

An alternative approach is to embed special purpose processors into applicable elements of the front-end equipment. These processors can be optimized for the function they are to perform, which prevents under utilization of processing power and enhances the flexibility and performance of the front-end element.

These special purpose processors take up little real estate when implemented with todays LSI and VLSI semiconductors. The modules which are ideally suited for this type of technology are serial data correlators, decommutators, real time data correction, engineering units conversion, quick look display, data simulation and many special application modules. These processing elements provide the building blocks for a very powerful, cost effective family of modular telemetry and communications products for the 80's and beyond.

INTRODUCTION

As data rates increase, and the need for real time processing grows, a new system architecture is necessary to fulfill ground station requirements. The current trend is to develop computer peripheral equipment to handle the telemetry and high speed processing functions. This approach only furthers the use of the general prupose computers which are quickly reaching their limits of performance.

Today's LSI and VLSI semiconductor technologies make it increasingly cost effective to develop special purpose processors. This paper will discuss an architecture which is based on distributing the processing power of the system throughout its elements. These elements take the form of special purpose processors which are optimized to perform a specific function. The result is a family of telemetry and communications products which will provide the performance and flexibility required for tomorrow's ground station environment.

A TYPICAL SYSTEM

A typical application will be presented in order to illustrate this distributed processing approach. The application is a 150 megabit/second frame synchronizer system. The elements required are a frame synchronizer and a decommutator for data acquisition, and a data simulator and bit error rate tester for system test and link quality monitoring.

The system is designed with two modes of operation. The first is local operation. In this mode, an external CRT terminal is used as the man-machine interface, eliminating front panel switches and indicators while providing off line maintenance and operation. The second mode is remote control operation. This mode allows the system to be controlled by an external device or be used in a network with other processing equipment.

To incorporate these two modes of operation, a user interface processor was designed. This system element uses a microprocessor as the link between the CRT or external processor and the elements within the system. In the local mode, the CRT provides the operator with an easy to use, "English" language interface to the system. The microprocessor prompts the operator for necessary information, verifies that it is valid, and then performs the necessary hardware setup functions. It can also display operational status and error alarms. In the remote mode, the microprocessor accepts commands from the remote device. Like the local mode, it verifies that the commands are valid and sets up the other system elements. The microprocessor offers the user a number of remote interface types: Serial (RS-232, RS-422), Parallel, IEEE-488, and others depending on the remote device requirements.

The benefit of the user interface processor is that it makes the system hardware transparent to the user. The system commands do not depend on the particular hardware implementation. This allows a generic telemetry command language to be developed, which is independent of the system hardware. The commands are converted into the bit level setup words and directed to the appropriate system elements. This allows extremely complicated setup procedures to be performed using simple commands. Similarly, status from a number of elements can be collected to compose meaningful status reports. The reports can be generated on request, at predetermined intervals, or when the status

changes. The user does not have to decode the various hardware bits to determine the state of the system.

The user interface processor has access to the setup logic of the other system elements over a high speed internal bus. The bus is also structured to allow real time data transfers for applications which require decommutated or processed data to be available to other elements. This particular application only requires the bus to be used for setup and status.

The next element of this system is the frame analyzer. This element is implemented as a programmable serial data processor which correlates the incoming data stream with known format characteristics in order to extract meaningful information about the data stream. The processor can keep track of number of valid frames, expected frames, flywheel frames, inverted frames, and frame sync dropouts. This status is available to the user interface processor. It also provides timing signals for the decommutator.

The decommutation function for this application was very simple. The sync information and a PN-Preamble were stripped off and the rest of the data stream was converted to parallel words and output to a high speed array processor. A stored program decommutator module was used. A more complex decom module will be discussed later.

The data simulation function is performed by combining a parallel to serial converter with a parallel data processor. This processor has an instruction set which is optimized to perform a single function: output a data field, word length information, and data orientation information. In this system the data field is used for simulated output data and the length and orientation information is used to control the parallel to serial converter for shifting out the data. The combination allows extremely complex formats to be generated at data rates exceeding 250 megabits/sec. The same processor can also be used with a serial to parallel converter to perform data decommutation. In this case the data field contains the ID tag and routing information while the length and orientation fields provide control for the serial to parallel converter. The deconmutated data can be put on the high speed bus and routed to subframe sync modules, floating point processors, reformatting processors, etc.

The last processing element in this system is the bit error rate processor. This processor compares an incoming PN-data stream with two internally predicted PN-patterns to produce linear and non-linear error rate data. The two rates are correlated to ensure the reference generator, which is used to measure the link bit error rate, is kept in synchronization with the received data stream. This approach greatly reduces the possibility of false reference generator resync in the presence of error bursts or long duration high error rates. The BER remains in sync with BERs approaching 40%.

ADDITIONAL ELEMENTS

The system described above is composed of a number of data processing elements, each implemented in a way to optimize the flexibility and performance of the specific functions. The elements can be used in many different applications. With the addition of the elements described below, this family of telemetry products can handle requirements from low end receive and record station to extremely complex environments doing real time data processing and display.

The first element required is a floating point processor for engineering units conversion. The unit is optimized to perform polynomial calculations by giving the processor access to two high speed multipliers, one generates powers of “x” and the other multiplies the power of “x” by their coefficients. “X” can either be the incoming data, or table look up values which require linear or polynomial interpolation. The result is a floating point or scalar solution to a fifth order polynomial in 3 microseconds. The processor gets its data from the high speed bus and can output processed data back onto the bus, or to an external device for further handling or display.

A second important processor is developed from a reformatting requirement where incoming data had to be re-organized and put into blocks before being re-transmitted. This processor is implemented as a bit slice processor with a writable control store. It takes only a single cycle to access incoming data, process it and write it out to its destination. With a cycle time of 100 nanoseconds a large amount of processing per data word is realized when a few microseconds per word time is available.

Mentioned in the previous section was a subframe sync processor. This processor takes data off the high speed bus which is designated as raw subframe data. The processor performs synchronization on subframe ID count or recycle subframe patterns. The subframe data can then be extracted from the data stream, assigned an ID tag and passed on for further processing. A number of these modules can be put into a system to meet format complexity.

CONCLUSION

The elements described above are implemented on a standard card size. They fit into a standard card rack and are designed with functional and electrical compatibility afforded by the high speed bus presented to all modules. The approach is very modular and allows functions to be selected and plugged in as required, or removed in much the same fashion as computer peripherals are added to a backplane. This provides maximum flexibility and adaptability to the user’s needs.

This distributed processing approach is similar to that of a mainframe computer. Interface modules are selected based on I/O requirements while processing modules are selected based on the complexity of the data manipulations to be performed. The processing elements act as co-processors operating simultaneously. The systems can be used as stand-alone stations or with a back-end computer system to handle any non-real-time functions. The result is a custom system for each application without the custom system cost.