A FLEXIBLE MULTIFUNCTION TELEMETRY INPUT/OUTPUT MODULE

Richard Woicik
L-3 Communications Telemetry & Instrumentation
15378 Avenue of Science
San Diego, California 92128

ABSTRACT

Many high-performance, reconfigurable data functions can now be integrated into a single PCI circuit board, making possible low-cost and complex systems using PCs or UNIX workstations. FPGA and PCI technologies are an excellent match to telemetry applications where commercial off-the-shelf solutions are desired, but customization is common and performance critical. A Multifunction Telemetry I/O (MFT) module was designed to exploit these technologies for both flight test and space telemetry ground systems. The reconfigurability of the module has facilitated evolutionary hardware enhancements as well as custom applications. These enhancements have been used both as building blocks for system integrators and for commercial-off-the-shelf (COTS) graphic setup, processing, archiving, and display software. The MFT module includes a standard set of telemetry functions: up to two bit synchronizers, an IRIG time decoder, and two independent telemetry serial input and output channels. The MFT module is also available on a 6U VME board. This paper describes some of the proven capabilities and applications of this module.

KEY WORDS

PCI Multifunction Telemetry Input/Output, Reconfigurable FPGA, Personal Computer

INTRODUCTION

The demand within the telemetry industry for COTS solutions, particularly those using PC technologies, evolved during the mid-1990s and directly led to the development of the PCI MFT, which was first shipped in 1998. Today, the telemetry community is increasingly accepting PCs as core components in ground systems because of the recent advances in performance and reliability and because of widely adopted commercial standards. In many cases, PCs are usurping more expensive embedded systems as well as workstations. For a long time, both PC hardware and software were seen as poor and unreliable performers and were avoided in any system with the least hint of mission criticality. This started to change in the mid-1990s with a convergence of technological innovations led by Intel and Microsoft. The most

significant of these were the Pentium series of multi-processors, the Peripheral Interconnect Bus (PCI bus), and the Windows NT operating system.

For telemetry systems using the PCI bus, the major challenge is the limitation in card slots. Adopting technologies designed to save board "real estate," including field programmable processors (FPGAs) and packaging technologies such as surface mount, solved this problem. The result was a board that integrated most of the functions of up to six boards into one. The basic standard telemetry functions of the MFT are bit synchronization, decommutation, simulation, time decoding, tagging and generation, and commanding.

The FPGAs were chosen to be downloadable from disk files. This allows a great deal of flexibility for reuse of resources on the board by changing internal logic and interconnections in a manner similar to software (i.e., the board loads from a file and then executes). This method allows improved customer service because revisions, new standard features, and customizations can be delivered without returning modules back to the factory for physical rework. It also allows the deployment of multiple hardware configuration files that uniquely allocate limited physical resources for different applications.

At transfer rates of up to 132 Mbytes per second, the PCI bus far out-performs its predecessor, the ISA bus, which tops out at about 8 Mbytes per second, and can be faster then many implementations of the VMEbus. The PCI bus also implements a rational plug-and-play installation and an arbitrated bus mastering system that limits the effect of bandwidth-hogging by slow circuit boards. Finally, the PCI bus removes legacy interrupt and memory mapping limitations.

Today, it is widely believed that new top-end x86 processors are never far behind in performance of processors in more expensive computers (usually UNIX) and embedded systems. Dual Pentium systems are widely used now and quad systems are also available. Multi-processor PCs give the telemetry user confidence that other processes won't crowd well-implemented real-time threads.

In addition to hardware improvements, common operating systems have also contributed to the acceptance of PCs. The Windows NT operating system cut loose of old DOS legacy support while maintaining most Windows 9x support. This allowed Microsoft to add important features such as support for multi-processor systems, multiple users, and protection of the system from application mode program crashes. Subsequent versions of NT have shown increased stability. The open systems UNIX-derived Linux is also widely accepted as a stable, multi-processing operating system.

PHYSICAL RESOURCES

The MFT's simple generic physical architecture can be seen in Figure 1. Most processing functions are performed by the FPGAs. Connectors, digital transceivers, analog inputs, synthesizers, memory, and a PCI interface augment the FPGAs. Daughter (mezzanine) cards were avoided to reduce the costs of extra board design and interconnect components.

The primary I/O, digital transceivers, and analog inputs are available to the user on the external panel on a DB50 connector. The parallel port is available on an internal header connector, where connection to a digital-to-analog converter (DAC) board is facilitated. Another header connector is also available for other auxiliary signaling.

The digital transceivers are primarily intended to support differential and single-ended serial applications. For telemetry, this normally means isochronous clock and data signals. The quantity of digital signals was selected to support two independent input and output channels. However, in some applications, these signals are reallocated to alternative uses. Eight TTL and eight pairs of differential signals are scaled for two pairs of clock and data out and two pairs of clock and data in.

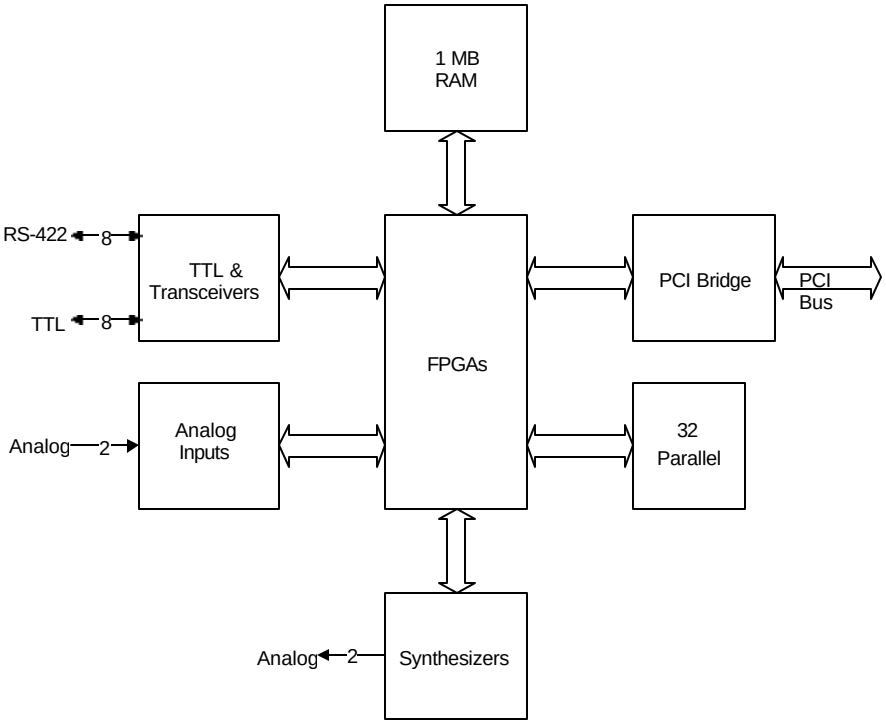


Figure 1. MFT Physical Resources

Two analog inputs were implemented to support PCM and IRIG time code demodulation. Each channel can accept a 40 dB signal range and programmable 75-ohm termination. Different FPGA configurations allow the two channels to be allocated between PCM and time or between two PCM inputs.

Signals up to 30 MHz can be generated by four frequency synthesizers. They are programmable to better than 1 Hz resolution and accurate to 4.6 PPM. These are used in various applications, including output clock generation, demodulation, and IRIG time signal generation.

One Mbyte of RAM is normally used for data buffering and sequencing. Access is arbitrated between the different functions depending on the configuration and user setup.

A proven COTS chip was selected for the PCI interface. The programmable chip implements all PCI signaling for both bus slave and mastering transactions. It implements PCI plug-and-play and includes two bus-mastering DMA engines.

STANDARD TELEMETRY CONFIGURATIONS

Four standard MFT configurations were designed for telemetry applications. They were designed by selecting commonly used sets of functions. (Each function usually includes an FPGA core plus some external non-FPGA physical resources.) To ensure deterministic behavior in real time, these functions all operate without host intervention, excepting the transfer of real-time data to and from the host.

Table 1 shows which FPGA function cores are present in each configuration. The configurations are contained in binary files identified by name (TIO74x.BFC). The Binary FPGA Configuration (BFC) file format concatenates all FPGA data together and adds an ASCII header. Concatenation was preferable to individual FPGA files because host application software would otherwise be required to unnecessarily sort out compatibility between various combinations.

Integration of standard telemetry functions can be seen in the generalized logical data flow diagram in Figure 2. Advantage is taken of the powerful signal routing capabilities of the board, unlimited by cables and proprietary buses. For example, precision time tagging can be performed by the time function through direct access to the input buffers and in response to asynchronous triggers from the two input channels. This capability is difficult when the functions reside on different boards.

Table 1. Multifunction Board Configurations

Function	***TIO745***	***TIO746***	***TIO747***	***TIO748***
Bit Sync 1		**X**	**X**	
Bit Sync 2			**X**	
Telemetry I/O 1	**X**	**X**	**X**	
Telemetry I/O 2	**X**	**X**	**X**	
Internal Time Clock	**X**	**X**	**X**	**X**
IRIG Time Input	**X**	**X**		**X**
IRIG Time Output	**X**			**X**
Ternary I/O 1				**X**
Ternary I/O 2				**X**
Parallel Output	**X**	**X**	**X**	
PCI Interface	**X**	**X**	**X**	**X**

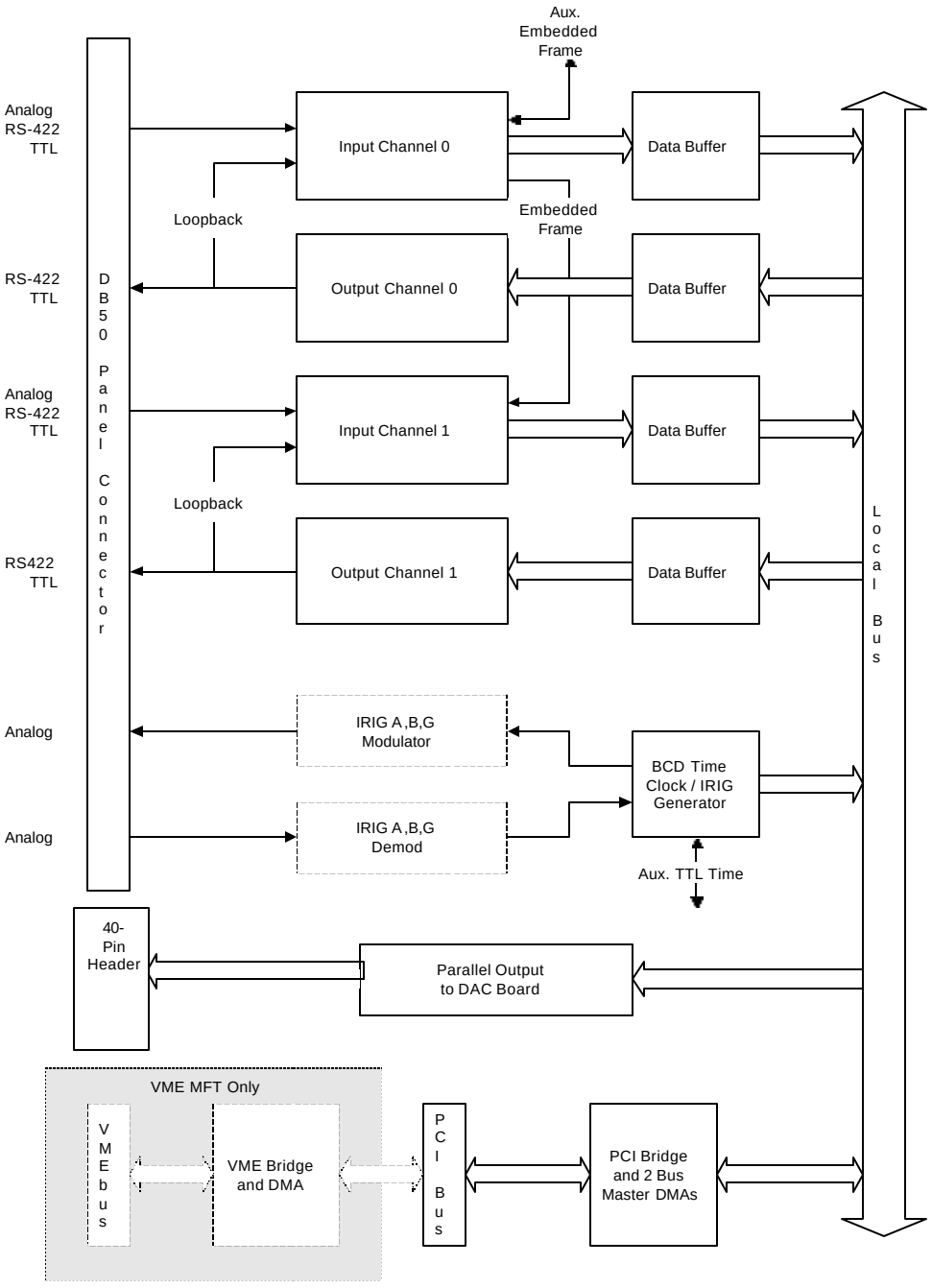


Figure 2. MFT Standard Configuration Data Flow

STANDARD TELEMETRY FUNCTIONS

Outstanding performance characteristics were obtained from the MFT's telemetry functions.

Digital PCM bit sync functions operate from 100 bps to 10 Mbps with characteristics that exceed most other board products. Acquisition and tracking of PCM signals is based on a proprietary 2nd-order phase lock loop. Bit error rate (BER) test results are shown in Figure 3. BER performance is mostly below 1.5 dB of theoretical at 10 Mbps per second and improves to a fraction of a dB at lower bit rates. NRZ bit acquisition was verified averaging within 10-bit periods at 10 Mbps per second and improving to within 1-bit period at lower rates, with or without 12 dB of noise. Programmable loop bandwidth, auto-offset, and auto-gain supporting a 40 dB signal range is included. The same bit sync core has been successfully ported to a remotely piloted vehicle for receiving commands from the ground.

Telemetry input and output functions support IRIG-106 Class I and common Class II decommutation and simulation. Burst and pass-all modes are also supported. The inputs accept isochronous data and clock signals from the bit sync function, from external TTL or differential signals, from the adjacent output channel, or from an embedded stream extracted by the other input channel. The outputs generate TTL and differential clock and data signals that are optionally PCM encoded or randomized. The I/O channels support interrupt service routines, polling, and slave and DMA data transfer methods using double buffers synchronized to data frames or built-in timers. The function cores originated in an ISA board first shipped in 1994. Since then, the functions have been improved and repeatedly demonstrated to transport data without loss or corruption.

Time functions support IRIG-200, -A, -B, and -G AM modulated time code standards. A 2nd-order phase lock loop in the digital demodulator locks onto the AM carrier and generates a 1 MHz signal used to drive the time clock. The digital implementation rejects the noise and distortion commonly found on instrumentation tapes. A narrow/wide loop bandwidth setting is included, which optimizes the system for particular circumstances. The narrow setting being best for real-time conditions and the wide being better for some instrumentation tapes. The AGC loop allows amplitudes from 0.2V to 10V peak to peak to be accepted. The AM modulator outputs IRIG-A, -B or -G synchronously to the time clock for time generation and translation applications.

The internal time-of-year clock either runs synchronously to the 1 MHz signal from the AM demodulator (decode/generate) or synchronously to a 1 MHz independently generated signal with a stratum 3 accuracy (4.6 PPM) (generate only). Time of year is kept in a binary-coded decimal (BCD) format. The BCD time clock is the time-tag source for the telemetry input channels, the AM modulator, and a register that can be read asynchronously by the host. The time clock performs pulse width modulated (PWM) symbol decoding from the AM demodulator or from an auxiliary TTL PWM source. It also performs error rejection and PWM symbol encoding, which can be output to the AM modulator and the auxiliary connector. The auxiliary feature is used in systems requiring more than one MFT board so that only one time demodulator needs to be instantiated and to ensure exact synchronization of their otherwise independent clocks.

In a recent space program, the time function design was carefully calibrated to be consistently within ±1 microsecond of the IRIG-B signal generated by a Datum CesiumPlus GPS time decoder/generator.

Comparing the one pulse per second signals coming from the Datum decoder/generator and that coming from the MFT's time clock function with a counter timer validates this.

The parallel output port is used to send decommutated words to a digital-to-analog (DAC) card. Raw data from the telemetry input functions is sent out the port when it is written into the data buffer.

Playback of archived data to DACs while preserving the original timing is difficult in some systems. This is not a problem for the MFT, where all data can be recorded in real time using pass-all mode to a file. Data can be played back through an MFT output channel in continuous streaming mode and the output can be looped back to an input channel configured for decommutation and output to the DACs. In applications requiring real-time decommutation and display, one channel can be set up in pass-all mode for archiving and the other can be set up in sorted or tag-data mode. The input signal can then be daisy chained to both channels (see Figure 3).

Another problem for some systems is DAC output of real-time processed data due to variable latency in processing. This is particularly problematic with non-embedded systems like PCs. A unique capability of the MFT is support for time-coherent output of host-processed data to the DACs. When this feature is enabled, part of the telemetry input buffer is configured to output extra data to the parallel port. A processed data word is sent to the parallel port when the same words from a later frame are written. A constant latency can be maintained for processed DAC output because the input channel demands it. The user must set a fixed delay in buffer times for the channel between the raw input and processed DAC output to ensure that the worst-case processing time cannot exceed it.

Ternary functions support SGLS standard format (1, 0, S) for commanding applications. Command packets are sent in a burst mode. The timing implicit in the packet is maintained at the output. Between packets, the board automatically transmits NULLs. An integral feature is dibit to ternary and ternary to dibit conversion.

EXAMPLE OF AN MFT SATELLITE APPLICATION

A ground system was constructed using PCs and MFTs for a commercial satellite venture. The system consists of a mission control center (MCC) and remote ground terminals (RGTs) located around the globe. Operators at the MCC issue commands to satellites via one of the RGTs where they are transmitted on the S-band. The X-band telemetry downlink includes a 256 kbps data stream on the main carrier and a variable 4 kbps/8 kbps data stream on a subcarrier. The two telemetry streams are routed back to the MCC via the RGTs and are also stored locally. Dedicated links are in place between the RGTs and MCC. Using the TCP/IP protocol, the components of the ground system are networked together for data transport and maintenance.

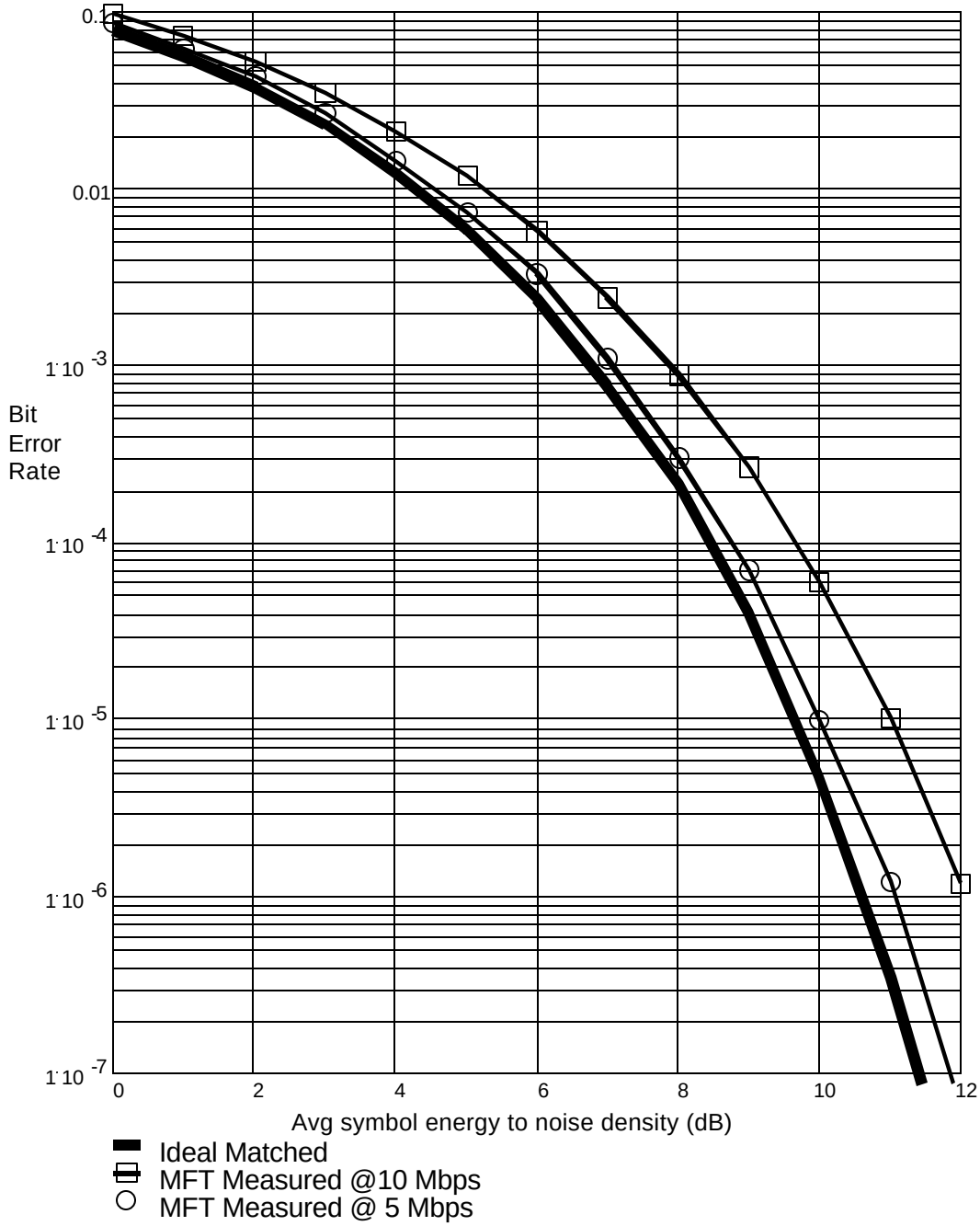
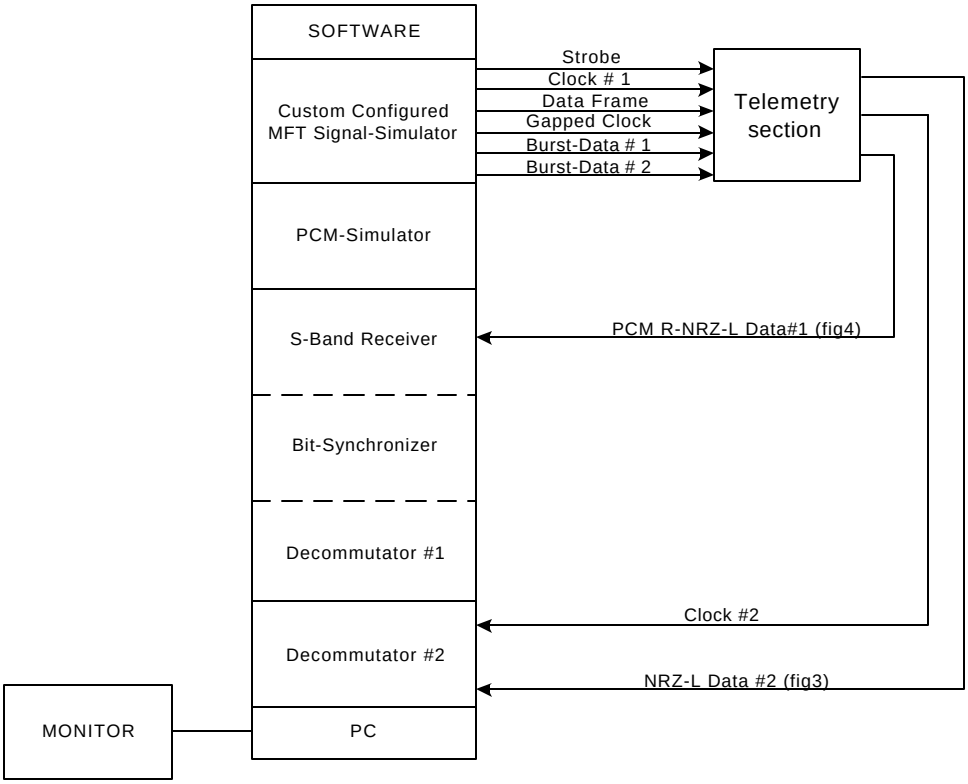


Figure 3. Typical MFT BER Performance

Two MFT boards are installed in each RGT PC. One receives the two encrypted or non-encrypted telemetry downlink streams. Application software takes the pass-all data from the two channels, archives it locally, and sends it over the network to the MCC. The two output channels are used to transmit simulated downlink data. The transmitted data is looped back to the inputs for testing. The other MFT at the RGT is connected to a SGLS modulator and demodulator. This MFT is configured with ternary functions.

Application software receives encrypted or non-encrypted commands from the MCC and sends them out the ternary output channel to command the satellites. The input channel receives the non-encrypted data in an RF loopback for echo checking. The received ternary commands are sent back to the MCC for comparison verification.

Two MFT boards are also installed in the MCC PC. The first board is connected to two decryptors. Application software at the MCC receives the two telemetry downlink streams from the RGT. If encrypted, the data is sent out the two output channels to the decryptors. The input channels receive and frame synchronize the clear data returned from the decryptor. The second MFT board is connected to a command encryptor. The MFT is configured with ternary functions. The application software sends unencrypted commands to the encryptor and receives them back, encrypted, on the MFT's input channel. The software



then sends the encrypted commands to the RGT via the network for transmission to the satellite (see Figure 4).

Figure 4. Custom Production Test System

EXAMPLE OF A CUSTOM MFT MISSILE TEST SIMULATOR APPLICATION

In one scenario, PC systems are being used to production test the telemetry section of a missile. The test systems generate simulated missile signals and output them to the missile telemetry section and receive and analyze the returning PCM telemetry. Each system contains two MFT boards. One board uses a standard configuration for IRIG-106 frame synchronization. The other board uses a custom signal simulator

configuration for generating eight required signals. The output signals vary in data and timing content from test to test. Three data streams are generated. Data rates for most tests are at 6 Mbps for one stream and 12.288 Mbps each for the others. Another normally 80 Hz strobe output signal is generated and used to coordinate the data output of the streams. Two streams burst data identically, but the third is different, making predictable and reliable timing of the bursts the critical design challenge.

In a full test cycle, three steps are required. First, test data files are generated using a custom C++ program according to the needs of the test from text instruction files created by the customer. The test files contain data values and timing information that will be sent to the missile's telemetry section. In the second step, the MFT signal simulator and the standard decommutators are enabled. The signal simulator host interface was designed to be compatible with a standard telemetry output mode so that COTS software could be used in this real-time step. A host process writes data to the signal simulator MFT as it reads it from the test file. Another process writes data to an archive file as it receives it from the decommutators. In the final step, another custom C++ program compares the archive file to the test file using rules set by the operator. If the comparison is successful, the device under test is considered to have passed the test. Otherwise, the reports from the custom program are used to help debug the problem.

The signal simulator FPGA implementation does not use standard functions because of the uniqueness of the application. Data flow is split in two on the MFT from a single pair of data buffers and processed separately. This reflects the application's requirement for the two timing streams. The processes run at different clock rates but are synchronized by the 80 Hz strobe. Each process separately parses the buffer for data and timing information relevant to its output stream. The result is a system where robust simulation timing and data can be changed using simple ASCII files and without the need to reprogram the FPGAs.

CONCLUSION

Extensive experience with the PCI MFT board has proven it reliable and robust in a variety of demanding applications. The board has been integrated with L-3's COTS software and customer software in both test and operational environments. The MFT's success in meeting both high-performance and small size design goals was facilitated by the use of FPGAs, PCI, and new packaging technologies. The MFT's host interface has proven to be easy to program, even for multi-threaded, multi-processor applications. Operating system drivers were created both at L-3 and by customers for Windows NT, Linux, SGI, and others. These operating systems all work well, but vary in difficulty of implementation. Bit sync BER performance, time-tagging precision, and data transfer are on par with many more expensive systems.