"DATA DIGITIZING UNIT" ELIMINATES THE NEED FOR ANALOG RECORDERS

Timothy B. Bougan
Science Applications International Corporation

ABSTRACT

Today's telemetry environment is becoming increasingly digital. Highly reliable and relatively inexpensive digital recorders readily available, and most telemetry facilities are migrating away from the older analog recorders which are difficult to calibrate and expensive to maintain. Unfortunately, most site managers find they still have one or more "legacy" signals (such as FM-FM, PAM, and pre-detect PCM) that still require analog recording. To exclusively use digital recorders the TM site must integrate some device to convert the analog signals to digital format before recording.

Until recently, the TM site managers had very few options short of building custom equipment to convert and capture the legacy signals. One solution available from Racal (for their Storeplex digital recorder) is to purchase their Analog Record/Play Signal Module. Unfortunately, their module uses a 16-bit Sigma-Delta converter and has a maximum bandwidth of 45.5 KHz, which is woefully inadequate for many analog signals. Other manufacturers offer similar solutions with similar bandwidth restrictions. Another solution is to purchase a multiplexor "front-end" which is capable of mixing multiple signal types (both digital and analog) on to the recorder's serial-digital data stream. This option can provide higher analog bandwidths, but represents a significant investment (greater than $100K and often more than the recorder itself).

This paper discusses the conceptualization, design, and performance of a unit to fill the gap between the low-bandwidth analog channel module and the high-end signal multiplexors. We will discuss how high-speed field-programmable gate arrays (FPGAs) can be configured to provide a low-cost interface between the digital recorder and the analog-to-digital converters (ADCs) and digital-to-analog converters (DACs) to capture and playback the analog signals. Our design focuses on achieving the maximum possible bandwidth for each analog signal while ensuring that IRIG-A or IRIG-B timecode are recorded simultaneously (so the analog signals can be later synchronized with their digital counterparts). We have found that such a solution permits multiple analog signals from 400 KHz up to 3 MHz to be easily and inexpensively recorded on the current generation of

digital recorders. Our conclusions show that such a device can permit most telemetry sites to transition completely to more reliable, cheaper, and easier-to-maintain digital recorders.

KEYWORDS

Recorders, Digital Recorders, Multiplexors, A/D Conversion, D/A Conversion, PCM, PAM, FM-FM

INTRODUCTION

The DDU-1004, or "Data Digitizing Unit", was designed at the request of the USAF to allow newly purchased Racal Storeplex digital recorders to replace aging analog recorders in the USAF's fleet of E-9A aircraft at Tyndall AFB, FL. The E-9A is a telemetry relay aircraft whose mission is to intercept telemetry signals from drones, missiles, and aircraft and relay the signals to ground stations for recording and/or processing. The E-9A also carries telemetry operators to monitor and locally record selected signals. Most of the signals are pulse code modulation (PCM) and are suited for direct recording on the digital recorders. The operators do encounter several "legacy" analog signals, however, such as pulse amplitude modulation (PAM), FM-FM signals, and pre-detect PCM signals that need to be relayed and recorded beside the PCM streams. These signals range in frequency from 20 KHz to 2 MHz.

The Racal Storeplex recorder is a 51.2 megabit per second digital recorder. Racal provides digital interface modules that allow bit-serial streams to be recorded at full speed (51.2 Mbs) or power of two fractions thereof (25.6 Mbs, 12.8 Mbs, 6.4 Mbs, etc). Racal also sells analog modules for the Storeplex, but these boards
have a maximum input bandwidth of 45 KHz (far below the requirement for the E-9A).

Our challenge was to design a two-stage unit. The first stage needed to capture and digitize multiple analog signals and feed them to the Storeplex at the maximum possible bit rate (to maximize bandwidth). The second stage must accept a signal from the Storeplex (on playback), detect the number of signals originally recorded, and accurately reproduce the original signals.

The analog-to-digital (A/D) and digital-to-analog (D/A) conversions are straightforward and are easily accomplished by any number of off-the-shelf components. The multiplexor (MUX) and demultiplexor (DEMUX) is the heart of the DDU and the primary subject of this paper.

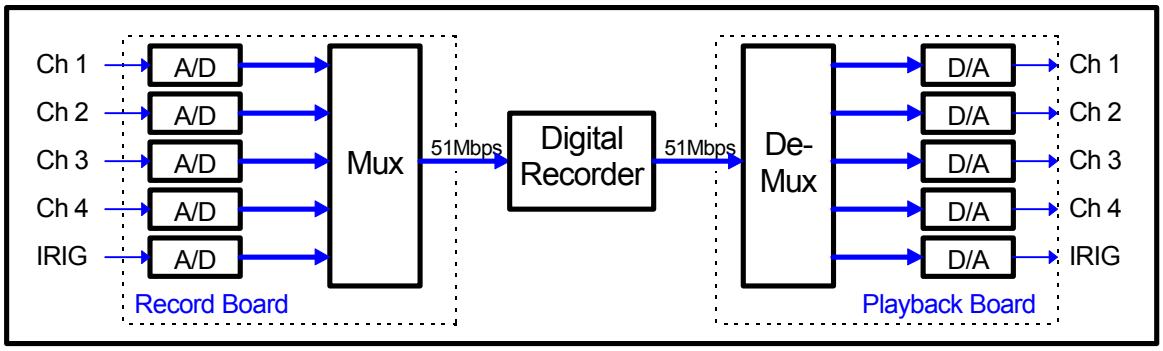

Figure 1 -- DDU Block Diagram

THEORY OF OPERATION

Our first task was to determine the maximum signal bandwidth that could be recorded by the Storeplex. We chose an ADC sample size of 8 bits (which met our resolution requirements). Running at full speed (51.2 megabits per second), and assuming we only record a single channel, the maximum theoretical recording rate is 51.2M divided by 8 or 6.4 million samples per second. Applying Nyquist's criteria we arrive at a maximum theoretical rate of 3.2 MHz.

Our requirements, however, specified that we must record IRIG timecode (in analog form) along side the signals of interest. This signal requires a minimum of 19 KHz and can be considered "low-speed" in comparison to our signal of interest. Furthermore, we needed to design to capability to select the number of high-speed channels to be recorded (between one and four). We also realized that we would need to insert some synchronization bits into the stream (so the playback board could determine where each frame of data begins). Since these sync bits represent "overhead" and directly reduce available bandwidth, we decided to limit them to one bit per 17 bits of data (for a calculated overhead rate of 5.89%).

Given the above requirements and assumptions, we decided on a sixteen sample "frame". Each frame would contain eight sync bits, eight low-speed channel bits (one sample), and sixteen high-speed samples from each of the selected channels (from one to four). To prevent jitter we must ensure the samples are evenly spaced. We did this be inserting only one bit of the sync byte before each high-speed sample, as such:

| SYNC Bit 7 | High-Speed Sample | SYNC Bit 6 | High-Speed Sample | SYNC Bit | ....

Each "high-speed sample" can be from 8 bits (if only one channel is being recorded) to 32 bits (if all four are being recorded). Obviously, we exhaust the sync pattern by interleaving them with eight high-speed samples. The next eight high-speed channels are interleaved with the eight bits from a single low-speed channel, as such:

| Low-Speed Bit 7 | High-Speed Sample | Low-Speed Bit 6 | High-Speed Sample | ....

A frame then, can consist of 144 bits, 272 bits, 400 bits, 528 bits (depending on whether you want to record one, two, three, or four high-speed channels). Each frame is contiguous... there is no "gap" between frames. Therefore, we can compute recording bandwidth as follows:

One channel = 144 bits/16 samples = 9 bits/sample
51.2 Mbs/9 bits/sample = 5.68M samples/sec = 2.84 MHz (max)

Two channels = 272 bits/16 samples = 17 bits/sample
51.2 Mbs/17 bits/sample = 3.01M samples/sec = 1.51 MHz (max) per channel

Three channels = 400 bits/16 samples = 25 bits/sample
51.2 Mbs/25 bits/sample = 2.05M samples/sec = 1.02 MHz (max) per channel

Four channels = 528 bits/16 samples = 33 bits/sample
51.2 Mbs/33 bits/sample = 1.55M samples/sec = 778 KHz (max) per channel

The low-speed channel includes only one sample per frame. Correspondingly, we can compute its recording bandwidth as follows:

One HS channel = 144 bits/frame = 1 sample/144 bits
51.2 Mbs * 1 sample/144 bits = 356 K samples/sec = 177.8 KHz (max)

Two HS channels = 272 bits/frame = 1 sample/272 bits
51.2 Mbs * 1 sample/272 bits = 188 K samples/sec = 94 KHz (max)
Three HS channels = 400 bits/frame = 1 sample/400 bits
51.2 Mbs * 1 sample/400 bits = 128 K samples/sec = 64 KHz (max)

Three HS channels = 528 bits/frame = 1 sample/528 bits
51.2 Mbs * 1 sample/528 bits = 97 K samples/sec = 48.5 KHz (max)

Determining the frame structure and recording bandwidths was the easy part. Realizing a device that could digitize from two to five signals, construct such frames as described above, and reconstruct the original signals on playback is a singularly non-trivial task when you consider the bit stream is 51.2 megabits per second. Using a general purpose processor was out of the question (most can't possibly handle the rate and those that can are very highly priced). Our options were threefold:

1) Design the system using fast discrete logic
2) Design the system using custom semiconductors (ASICs)
3) Design the system using programmable logic (PALs or FPGAs)

We disregarded the second option as being cost prohibitive for our quantity desired. Either of the other two options would work, but to minimize required space and optimize post-design flexibility, we opted to use field programmable gate arrays (FPGAs) for the bulk of the digital design. In particular, we choose to use Xilinx 4005 FPGAs.

Our system also needs some user interface (to select the number of channels to record) and higher-level computational power (to "detect" the number of channels recorded on playback). We had a requirement to read a rotary switch and respond to RS-232 commands from a remote controller. To add these additional capabilities without a lot of overhead, we decided to place a low-cost Intel 8751 microcontroller on the each board. These microcontrollers have built in serial ports (for the RS-232 control) and have sufficient I/O lines to read a rotary switch. Additional I/O lines go to the FPGA to command number of channels to record or detect.

Figure 2 shows the basic layout of the record board; figure 3 shows the playback board.

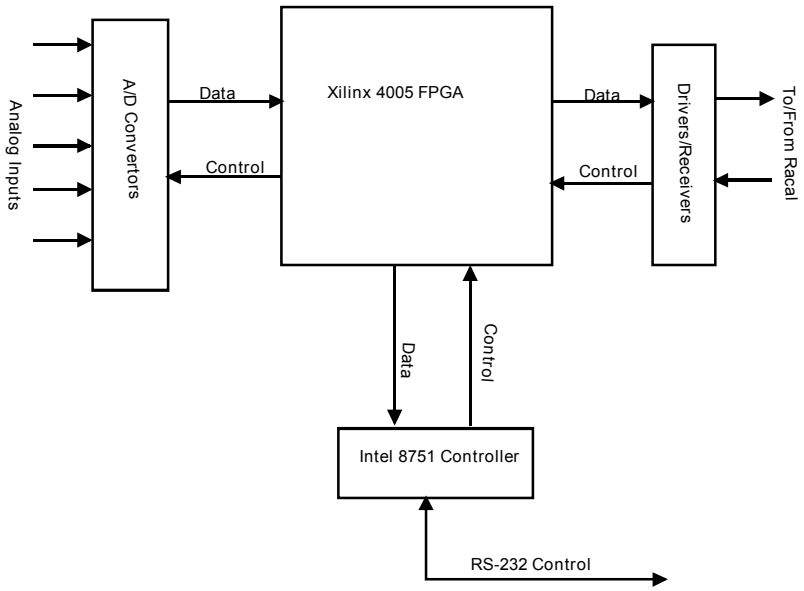

Figure 2-- Record Board Basic Layout

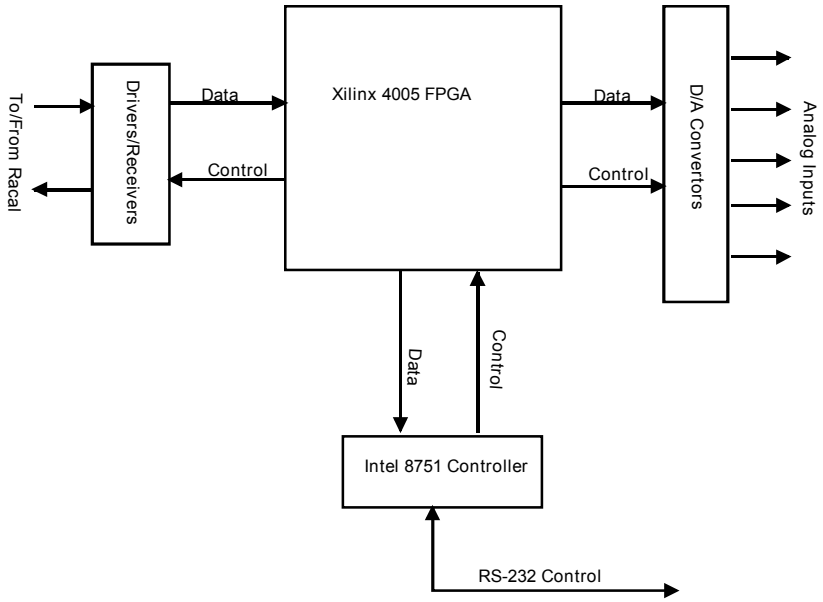

Figure 3 -- Playback Board Basic Layout

The most difficult task was designing the internal loads for the FPGAs. Using Xilinx supplied tools, we were able to specify the internal loads as schematic diagrams and compile them into a bit pattern that can be loaded into the FPGAs at run time. Although the designs themselves are straightforward, the reader should be aware that FPGAs, unlike discrete logic, can have propagation and routing delays that can vary from compile to compile (since complex designs can be routed in an infinite number of ways). Designing a set of state machines, registers, comparators, and shift registers that can run at 51.2 MHz is difficult at best and requires detailed working knowledge of the operation of both the routing tools and the target device. The fast clock allows only 19 nsec between transitions, and state machines must process all decision logic, propagate to the next stage, and meet setup times all within that narrow window.

The heart of the record FPGA is a series of shift registers, a 16-to-1 multiplexor, and an extremely fast state machine. The sampled data from each high-speed channel is loaded into a separate shift register (in line with the other channels). The state machine commands a simultaneous load for each channel. It then shifts the bits out one at a time.

At the end of the high-speed data, the state machine enables the 16-to-1 multiplexor and allows one of the 16 inputs to be appended to the serial stream. The inputs to the multiplexor are the 8 sync bits (hard-wired) and the 8 bits from a register holding the low-speed data (IRIG). See figure 4 for a block diagram of the serialization circuitry.

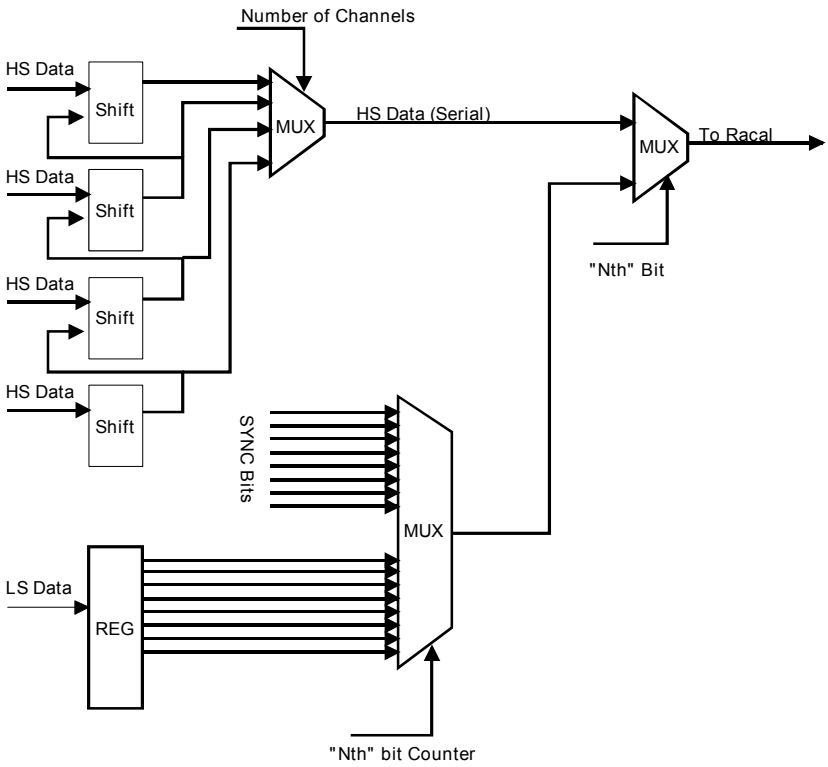

Figure 4 -- Record Serialization Circuit

The 51.2 MHz clock comes from the Racal Storeplex. The “Nth-bit” counter and “Nth-bit” signal are generated by the internal state machine. The “Nth-bit” refers to the last bit in a subframe (i.e. the SYNC or low-speed data bit appended to the high-speed data). For instance, if you are recording only one high-speed channel, the “Nth-bit” would be the “9th-bit”, since one channel requires 8 bits. For two channels N=17, for three, N=25, and for all four N=33. The “Nth-bit” counter is a 4-bit counter that cycles from 0 to 15 (and determines which SYNC or low-speed data bit is appended to the stream).

The “Number of Channels” control comes from the Intel Microcontroller and is set by user input or remote control.

Reversing the process for playback is more complex since the playback unit must “detect” how many channels were originally recorded. This involves sequentially shifting bits in and attempting to reconstruct the sync pattern (which occurs in eight sequential “Nth” bits). The playback board cycles through the possible number of channels and attempts to recover sync for each one. If sync is not detected after an entire frame is processed, the next channel increment is attempted. This process will repeat until successful. Once found, the sync pattern is predictable and looked for in every frame. If sync was erroneously detected, or subsequently lost, the playback board restarts the sync acquisition process (see figure 5).

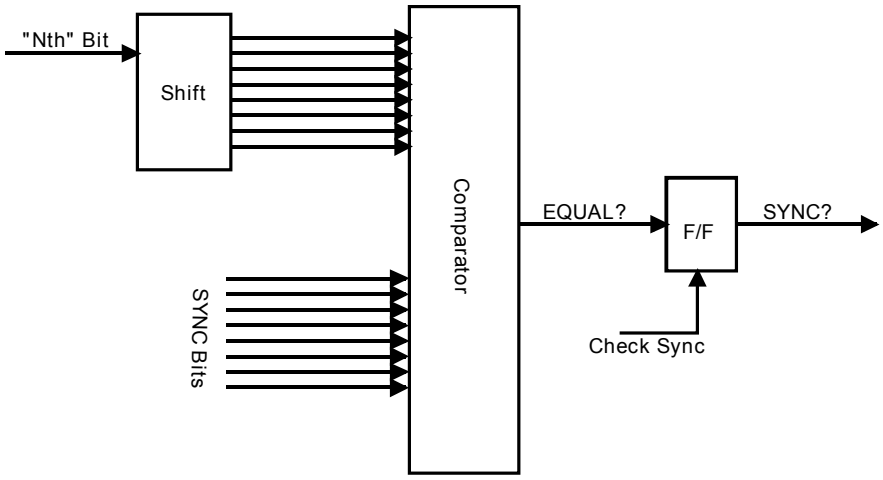

Figure 5 -- Sync Detection Circuit

The microcontroller knows how many channels have been recorded because it sets “N” (for the “Nth” bit) based on the number of channels it is testing for. When SYNC is detected (the output from the shift register matches the hardwired SYNC pattern) the microcontroller stops changing the number of channels to check (until sync is subsequently lost).

Recovering the original data, once sync is detected, is straightforward. Remember, however, that even though the logic is simple, we still must deal with exceptionally fast logic and very short intervals to meet routing, setup, and hold requirements.

CONCLUSIONS

In February 1996, the US Air Force deployed four DDUs in their E-9A aircraft at Tyndall AFB, FL. The units have permitted the USAF to permanently retire the old analog recorders on their E-9A aircraft and still maintain the ability to record legacy analog signals along with PCM streams. Testing reveals that the DDUs reached their predicted maximum capability. Because the clock is provided by the Storeplex, and because the system records data digitally, it is impossible to exceed the predicted capability.

The successful deployment of the DDUs demonstrates the feasibility of constructing, at a relatively low cost, devices which can multiplex several high-speed analog signals into a bit serial stream while simultaneously utilizing the full capability of inexpensive and highly reliable digital recorders. These devices will allow virtually any facility to migrate completely to digital recorders.