One Gigasample Per Second Data Acquisition Using Available Gate Array Technology

K.W. Wagner
NASA Goddard Space Flight Center, Greenbelt, MD 20771

Abstract

A new National Aeronautics and Space Administration instrument forced demanding requirements upon its altimeter digitizer system. Eight-bit data would be generated at a rate of one billion samples per second. NASA had never before attempted to capture such high-speed data in the radiation, low-power, no-convective-cooling, limited-board-area environment of space. This presentation describes how the gate array technology available at the time of the design was used to implement this one gigasample per second data acquisition system.

I. Introduction

NASA Goddard Space Flight Center is building an instrument that will measure the height of the earth's polar ice caps. An instrument aboard a polar orbiting satellite will fire a laser at the earth. A portion of the laser's outgoing photons is immediately shunted back to the instrument's detector. A small portion of the laser's photons is reflected off the earth and return to the instrument's detector. The detector generates a voltage based on the rate at which photons are striking it. We measure the amount of time between the detector receiving the shunted outgoing photons and the reflected incoming photons. Using this measured time and the trajectory of the satellite's orbit, we can calculate the height of the earth's surface.

Since the speed of light in a vacuum is 2.9979×10^8 meters per second, light travels 1 meter in 3.3 nanoseconds. The project's scientists determined that digitizing the detector's voltage waveform to 8 bits at a rate of 1 gigahertz (1 sample per nanosecond) would produce the necessary measurement accuracy for measuring the time between detection of the shunted outgoing photons and the reflected incoming photons.

II. Requirements

The satellite will be flying in a low-earth orbit at an altitude of approximately 600 kilometers altitude. Radiation engineers predicted the worst-case environment of the electronics at between 10 KRad and 30 KRad. Ray-trace analysis is showing that this radiation number estimate is slightly pessimistic and will be reduced based on the instrument's physical configuration and further analysis of the orbit.

All electronics should fit on a single 8-inch by 9-inch board, a requirement that was eventually relaxed to 8 inches by 12 inches. A fully redundant cold spare board has been allocated

The sequence of events aboard the instrument is as follows (see Figure 1):

1) The laser is commanded to fire every 25 milliseconds (40 Hertz).

2) The laser fires 200±5 microseconds after the command to fire.

3) Some outgoing photons never leave the instrument. Rather, they are shunted back to the detector.

4) After reflecting off the earth's surface, a small number of photons re-enter the instrument and strike the instrument's detector.

The electronics need to be able to:

- Digitize the detector's voltage waveform during the period that includes the time shunted outgoing photons are striking the detector.
- Digitize the detector's voltage waveform during the period that corresponds to 11 kilometers and includes the expected time of the reflected photons' return.
- Filter the reflected photons' waveform. The filtering is a digital Finite Impulse Response filter on up to 11 kilometers surrounding the expected time of the photon return.
- Determine the exact time when the shunted outgoing photons were detected.
- Search the filtered waveform for the exact time when the ground return was detected. This time is determined by searching the filtered return waveform to find when the filtered value reached a programmable threshold value.
- Make an accurate measurement of the time between the outgoing laser shot and the incoming reflected photons' arrival.

The need to make an accurate measurement of the time between outgoing and incoming photons led to a design decision. In order to save memory, the digitized waveform could have been saved only during outgoing and incoming periods. If this were done, no memory would be needed for the period when the photons were traveling to and from earth. However, it was determined that it would be too difficult to turn collection off and on at precise times to accurately measure the time between outgoing and incoming photons.

Instead, a large amount of memory will be used so that the acquisition to memory will not have to be turned off and on during the laser shot. Digitized detector voltages will be written continuously to consecutive memory locations at 1 gigasample per second from the time when the command to fire is issued, through the time of outgoing photons, through

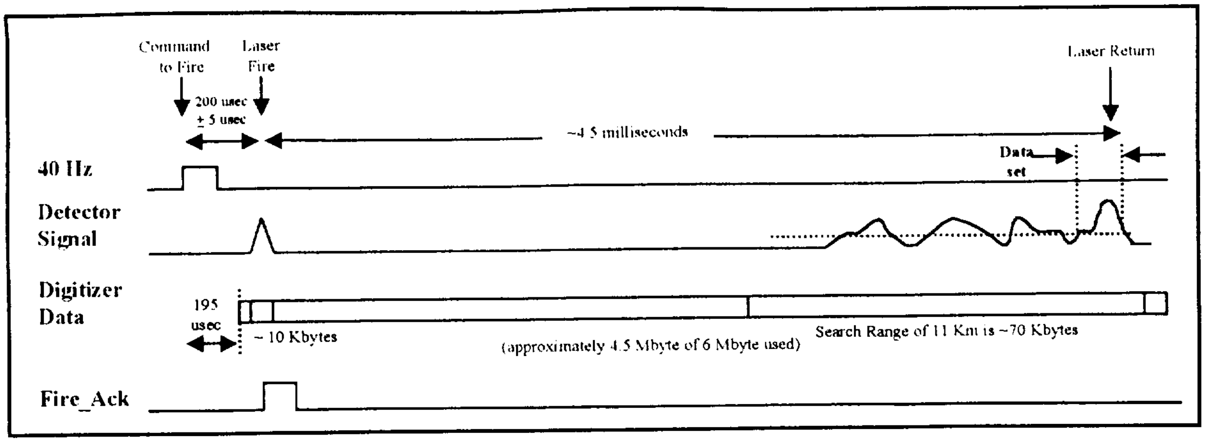

Figure 1: Instrument Event Timing

the travel time to earth and back, and past the time of arrival of incoming photons. To calculate the time between outgoing and incoming photons, the software needs only to find the difference between the addresses of the two events and convert the difference to time using a factor of 1 nanosecond per sample. In an orbit of 600 kilometers, it takes approximately 4.5 milliseconds for the laser's photons to travel to the earth's surface and back. At 1 nanosecond per 8-bit sample, 4.5 million bytes of memory are needed to store the entire waveform.

III. Initial Simple Asic Approach

The original simple concept was to use Rambus technology and high-speed Application Specific Integrated Circuits (ASICs) (Figure 2).

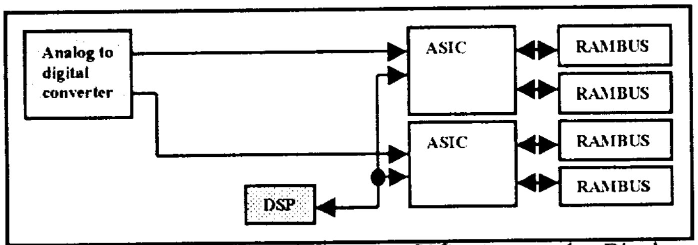

Figure 2: Initial Simple ASIC Approach Block Diagram

Rambus is a CMOS memory standard being used by the computer industry. It has a high-speed serial interface allowing for write and read operations at greater than 500 megabytes per second using formatted packets. Rambus uses a serial addressing and command scheme which leads to a low pin count and small package size. The Rambus protocol uses small-swing voltages to achieve high data rates, with the typical V_{LO} to V_{HI} swing being 1.15V. Many major computer-industry memory manufacturers presently produce Rambus-based products.

Each ASIC in the system would provide interfacing between three blocks (Figure 3): The analog-to-digital (a/d) converter block with its two 500-MHz emitter-coupled-logic (ECL) output channels; the CMOS Rambus block with its 250-MHz clock, 500-MHz data lines, and odd logic-level I/O; and the executor block global bus with a CMOS digital signal processor (DSP) at its center.

This simple design could not be implemented for several reasons. No programmable logic devices could be found that would meet the interface level and speed requirements of this design. Nor could a programmable logic device be found that could collect and store the a/d converter data and write it to the Rambuses in bursts necessary to achieve the greater than 500 megabyte per second write rate. ECL and Gallium Arsenide (GaAs) ASIC manufacturers estimated the chance of successfully manufacturing such a chip the first time around at only 70%, and the cost of the part would be huge. Additional versions of the ASIC to correct the errors would also be expensive.

IV. Completely Discrete Approach

With the ruling out of the simple approach of an ASIC serving as an interface directly between the three blocks, the design effort moved toward a completely discrete approach. In the completely discrete approach (Figure 4), the a/d converter outputs are saved into a series of ECL latches. These discrete latches slow down the data from its dual 500-MHz channels to a speed that can be written directly to regular Static Random Access Memories (SRAMs). While half the latches are receiving data, the other half remain stable while their data are written to the SRAMs through a field programmable gate array (FPGA). The gate array generates the necessary control and address signals to the SRAMs and ECL logic. Lockheed Martin and White Microelectronics manufacture SRAMs that are 128K 32-bit words per chip and 20-nanosecond cycle write time.

Because the slowdown of the data is now being done in many discrete ECL parts rather than in a gate array, the board space required is now larger than the allocation. The power to drive the ECL parts is also too large because, unlike CMOS parts, ECL parts' power consumption is too high and independent of the operating frequency. The pin count of the gate array would be too large for known devices because it

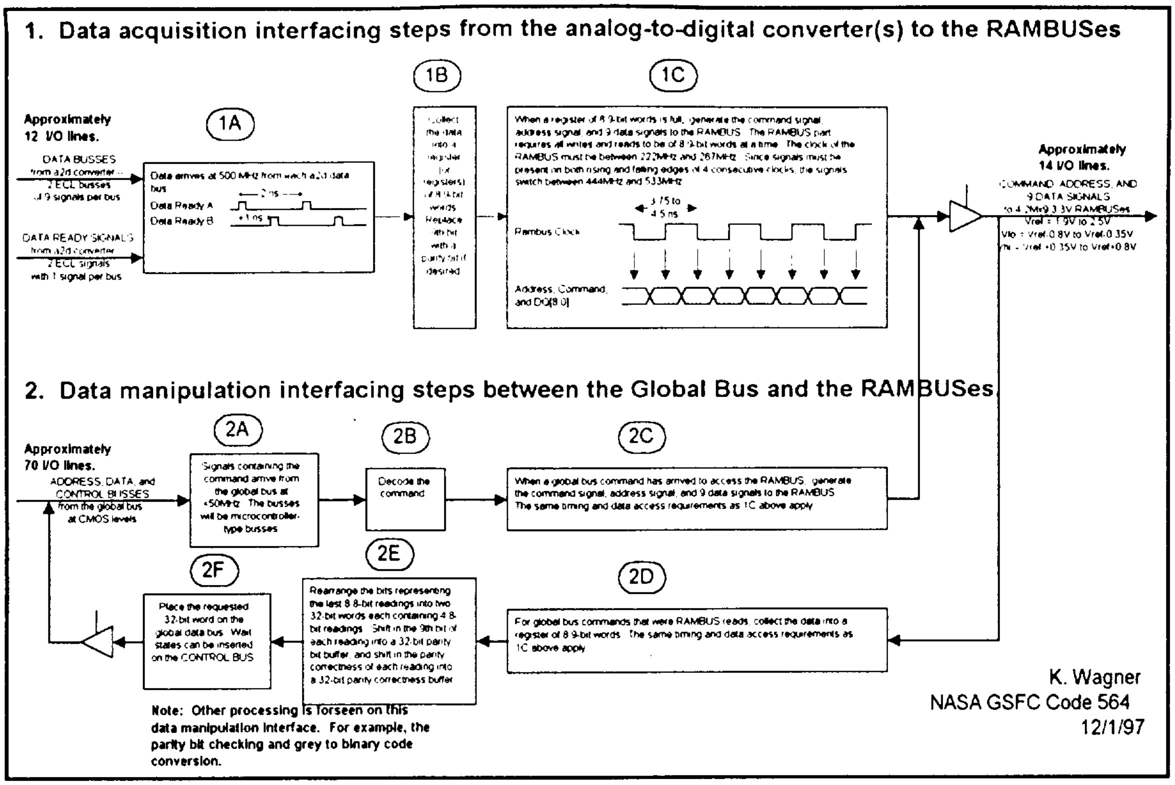
1. Data acquisition interfacing steps from the analog-to-digital converter(s) to the RAMBUSes
1A
1B
1C
Approximately 12 I/O lines.
DATA BUSSES from a2d converter -- 2 ECL busses of 9 signals per bus
DATA READY SIGNALS from a2d converter 2 ECL signals with 1 signal per bus
Data arrives at 500 MHz from each a2d data bus
2 ns
Data Ready A
Data Ready B
1 ns
Collect the data into a register (or registers) of 8 9-bit words. Replace 9th bit with a parity bit if desired
When a register of 8 9-bit words is full, generate the command signal, address signal, and 9 data signals to the RAMBUS. The RAMBUS part requires all writes and reads to be of 8 9-bit words at a time. The clock of the RAMBUS must be between 222MHz and 267MHz. Since signals must be present on both rising and falling edges of 4 consecutive clocks, the signals switch between 444MHz and 533MHz
3.75 to 4.5 ns
Rambus Clock
Address, Command, and DQ[8:0]
Approximately 14 I/O lines.
COMMAND, ADDRESS, AND 9 DATA SIGNALS to 4 2Mx9 3.3V RAMBUSes
Vref = 1.9V to 2.5V
Vlo = Vref-0.8V to Vref-0.35V
Vhi = Vref+0.35V to Vref+0.8V
2. Data manipulation interfacing steps between the Global Bus and the RAMBUSes
2A
2B
2C
2D
2E
2F
Approximately 70 I/O lines.
ADDRESS, DATA, and CONTROL BUSSES from the global bus at CMOS levels
Signals containing the command arrive from the global bus at <50MHz. The busses will be microcontroller-type busses
Decode the command
When a global bus command has arrived to access the RAMBUS, generate the command signal, address signal, and 9 data signals to the RAMBUS. The same timing and data access requirements as 1C above apply
For global bus commands that were RAMBUS reads, collect the data into a register of 8 9-bit words. The same timing and data access requirements as 1C above apply
Rearrange the bits representing the last 8 8-bit readings into two 32-bit words each containing 4 8-bit readings. Shift in the 9th bit of each reading into a 32-bit parity bit buffer, and shift in the parity correctness of each reading into a 32-bit parity correctness buffer
Place the requested 32-bit word on the global data bus. Wait states can be inserted on the CONTROL BUS
Note: Other processing is forseen on this data manipulation interface. For example, the parity bit checking and grey to binary code conversion.
K. Wagner
NASA GSFC Code 564
12/1/97

Figure 3: Simple Design's ASIC Functions

interfaces with a large number of ECL latches and there are many data and address pins on the SRAMs.

V. COMBINED SOLUTION

The solution was to combine the two approaches (Figure 5). ECL discrete latches demultiplex the a/d converter outputs to slow them down. Eventually the data are stable on each channel long enough so that it can be latched into gate arrays. In the gate array the data are further latched to slow it down until the gate array can latch the data into the SRAMs. This solution lowered the number of ECL discrete latch chips to an acceptable value for power and area limitations, and it lowered the number of pins needed to interface to the SRAMs.

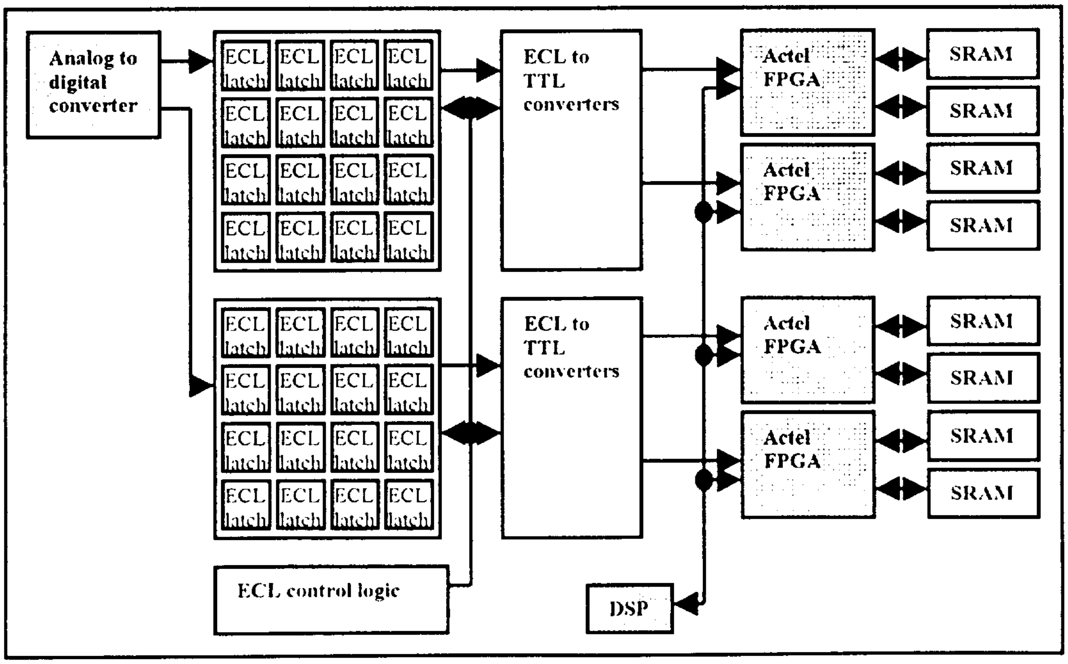

Figure 4: Completely Discrete Approach Block Diagram

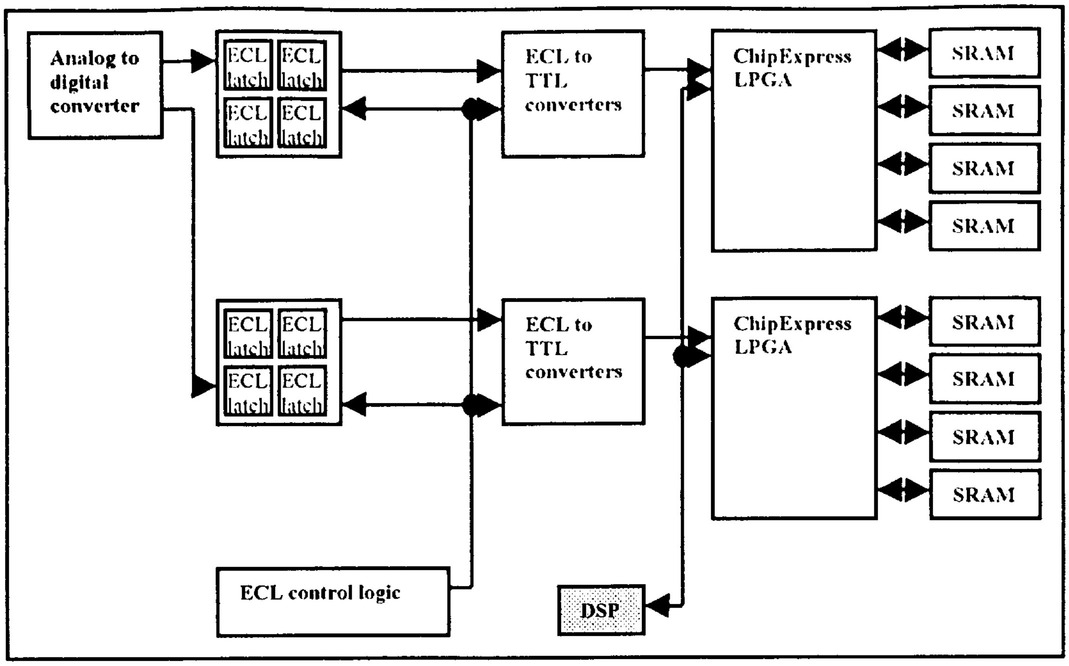

Figure 5: The Combined Solution Block Diagram

ECL latches lower the data rate from the a/d converter's pair of 500-MHz 8-bit channels down to twelve 83.333-MHz 8-bit channels. Each 83.333-MHz channel has its own "data ready" signal indicating stable data is on the channel. The twelve 83.333-MHz channels and data readies are converted from ECL to TTL levels. Each of four gate arrays receives three of the 83.333-MHz (12 nanosecond) channels and their data readies. The data bytes arrive 120 degrees out of phase from each other (that is, data arrives to a gate array every 4 nanoseconds from one of its three connected 83.333-MHz channels on a rotating-channel basis.) When a gate array has received four bytes through those channels, the gate array performs a write to a 32-bit SRAM, which is a write frequency of 20.833 MHz.

The gate array also serves other purposes. It decodes the a/d converter data from gray code to binary code. The gate array interleaves the data from the 12 SRAMs so that it appears to the DSP as a single 1.5 megaword by 32-bit device. The interleaved data is fetched from all the SRAMs in the order in which it was written to the SRAMs, not in the order in which it appears in a particular SRAM.

In reviewing the parts available at the time of this design, the Chip Express QYH580 Laser Programmable Gate Array (LPGA) met the functional requirements. Each 304-pin gate array allowed for the large number of input/output (I/O) pins that are needed: Data and data ready pins for the three input 83.333-MHz channels; address, data, and control pins for three 128Kx32 SRAMs; and address, data, and control pins for the DSP interface. Only 10,000 of the part's 80,000 NAND gates were used. The large array was needed to satisfy the I/O pin count requirement. Careful synthesis, layout, and analysis allowed the LPGAs to handle the data that each LPGA received every 4 nanoseconds from one of the three 83.333-MHz data channels connected to it. Since the hardest part of the routing was getting the channels from the board, through the input cell, and to the flip-flops, triple module redundancy (TMR) architecture was used in the LPGA design. TMR uses three flip-flops for each memory cell, with the majority value of the flip-flops used as the memory cell's value. This allows single event upset to an individual flip-flop without an effect of the function of the LPGA.

Although the Chip Express QYH580 did meet the requirements for the design, it was not a perfect part for the application. It is not a field programmable gate array, and as such the manufacturer had to program the parts. Although turnaround time is one to four weeks, this is not the same as burning a part in our own lab. Iterations in design are expensive. The place-and-route tools are not in-house, which didn't allow for extended experimenting in various floorplans.

For future designs, it would be useful if there were programmable logic devices that had advanced attributes. Faster input cells would have been helpful. For our application, having a superfast internal cell speed was not the most important criterion. Rather, what was needed was apart that could take in signals at 250 or 500 MHz. CMOS parts that can accommodate ECL logic-level inputs or user-specified inputs would have reduced chip count. A field programmable part would have been helpful. A part that can handle the Rambus interface standard would have allowed us to design in the Rambus, a memory device being used in the computer industry.