5-2009

NSMS probe recorder design and development

Daniel S. Crews
University of Tennessee

To the Graduate Council:

I am submitting herewith a thesis written by Daniel S. Crews entitled "NSMS probe recorder design and development." I have examined the final electronic copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Electrical Engineering.

Bruce W. Bomar, Major Professor

We have read this thesis and recommend its acceptance:

Accepted for the Council:
Carolyn R. Hodges

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

To the Graduate Council:

I am submitting herewith a thesis written by Daniel S. Crews entitled “NSMS Probe Recorder: Design and Development.” I have examined the final electronic copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Electrical Engineering.

Bruce W. Bomar, Major Professor

We have read this thesis
and recommend its acceptance:

L. Montgomery Smith_____

Bruce A. Whitehead_____

Accepted for the Council:

Carolyn R. Hodges_____
Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

NSMS PROBE RECORDER:
DESIGN AND DEVELOPMENT

A Thesis
Presented for the
Master of Science
Degree
The University of Tennessee, Knoxville

Daniel S. Crews
May 2009

DEDICATION

This thesis is dedicated to my wife, Tiffany, for her continuous love and support and to my son, Sterling, for his excitement and joy every time I return home.

ACKNOWLEDGEMENTS

I wish to thank all who helped with the development of this thesis and the work described within. I would like to thank my advisor, Dr. Bruce W. Bomar, for his guidance during the preparation of this thesis and also Dr. L. Montgomery Smith and Dr. Bruce A. Whitehead for serving on my thesis committee and their valuable inputs. I would also like to thank the other members of the NSMS team, Brent Cox, Bryan Hayes, Terry Hayes, and Ron Wood, for their efforts during the development of the system. I would also like to thank all those that participated in the technical peer review process for their insightful questions and comments.

ABSTRACT

The real-time Non-Contact Stress Measurement System (NSMS) currently used at AEDC calculates the vibration of rotating blades by capturing the time of arrival for each blade. The time of arrival is determined by a triggering circuit that is activated when the signal from the engine probe crosses a predetermined threshold. In its current configuration, the NSMS system only saves post-processed data. A system that records the raw signals from the probes was developed to allow reprocessing the data whenever necessary.

The probe recorder system consists of analog-to-digital conversion hardware to capture the signals, data storage for the files, and digital-to-analog hardware to replay the signals. The system accommodates a maximum of 32 channels, a maximum sampling rate of 20 MHz, and a total bandwidth of up to 160 megabytes per second. Sixteen-bit resolution is used in digitizing the analog waveforms to minimize quantization errors. The incoming data is transferred using FPDP, capable of 160 MB/sec, and PCI-X, capable of 528 MB/sec. Large amounts of high speed (3200 MB/sec) random access memory coupled with two dual-core processors were included for data transfer buffering and program execution. As the final destination, a RAID array connected to a PCI Express interface was implemented for 240 MB/sec data storage.

Laboratory tests were conducted on the system to verify performance. The RAID array exceeded expectations for disk writing but reduced bandwidth was observed for read operations. The relationship between the input analog signals and the reproduced waveforms was checked and, except for one case, performed identically to the simulated system transfer function. Long duration tests were performed to verify data transfers at the maximum settings and proved that the system could operate continuously without data loss.

Due to the large amounts of data, a brief study of offline compression techniques was conducted. Lossy compression was investigated but was not implemented at this time due to unwanted distortion and loss of critical data. Lossless compression using

WinZip was implemented as a compromise between ideal compression ratios and data retention expectations.

TABLE OF CONTENTS

Chapter **Page**

1. Introduction 1
2. Determination of System Requirements 4
3. Hardware System Design 7
4. Software System Design 17
5. Checkout and Results 24
6. Data Compression 44
7. Summary and Conclusions 49

LIST OF REFERENCES 52

VITA 55

LIST OF TABLES

Table **Page**

Table 1: Interpolation formulas 30
Table 2: Long duration throughput results 38
Table 3: Laboratory throughput specifications 43
Table 4: Lossy compression test set results 47
Table 5: Lossless compression test set results 48

LIST OF FIGURES

Figure **Page**

Figure 1: Example probe signal 2
Figure 2: 4th Generation NSMS System 2
Figure 3: Picture of ICS-645 analog input board 7
Figure 4: ICS-645 block diagram 9
Figure 5: Picture of ICS-500 FPDP to PCI-X adapter 10
Figure 6: ICS-500 Block Diagram 10
Figure 7: Schematic of 2 channel selectable gain amplifier 12
Figure 8: Probe Recorder hardware diagram 13
Figure 9: Data flow diagram for recorded signals 15
Figure 10: Data flow diagram for playback signals 15
Figure 11: Probe recorder acquisition front panel 17
Figure 12: Execution flowchart for data acquisition software 19
Figure 13: Execution flowchart for continuous loops 21
Figure 14: Probe Recorder top level file structure 22
Figure 15: Probe Recorder file viewer front panel 23
Figure 16: Low speed single disk write test results 26
Figure 17: Single disk drive write test results 27
Figure 18: RAID array write test results 28
Figure 19: RAID array read test results 29
Figure 20: Impulse Response - 2.5 MS/sec, 1x interpolation 32
Figure 21: Impulse Response - 1.25 MS/sec, 2x interpolation 33
Figure 22: Impulse Response - 625 kS/sec, 4x interpolation 34
Figure 23: Impulse Response - 312.5 kS/sec, 8x interpolation 35
Figure 24: Impulse Response - 156.25 kS/sec, 16x interpolation 36
Figure 25: Impulse Response - 1.25 MS/sec, 2x interpolation (modified simulation) 37
Figure 26: 100 kHz input, 20 MHz sample rate at maximum settings 39

Figure 27: 100 kHz input, 20 MHz sample rate FFT at maximum settings 39
Figure 28: 100 kHz input, 10 MHz sample rate at maximum settings..... 40
Figure 29: 100 kHz input, 10 MHz sample rate FFT at maximum settings 40
Figure 30: 100 kHz input, 5 MHz sample rate at maximum settings..... 41
Figure 31: 100 kHz input, 5 MHz sample rate FFT at maximum settings 41
Figure 32: 100 kHz input, 2.5 MHz sample rate at maximum settings..... 42
Figure 33: 100 kHz input, 2.5 MHz sample rate FFT at maximum settings 42
Figure 34: Histogram of typical probe signal 45
Figure 35: Block diagram of lossy compression algorithm 45
Figure 36: Original symmetric signal and lost information 47
Figure 37: Lossless compression ratios and time required 48

1. Introduction

Frequently used on turbine engine rotors, the Non-contact Stress Measurement System (NSMS) calculates the vibration of rotating blades by converting the blade time of arrival data to blade deflection data [1]. In order to capture the arrival time of each blade, laser probes are mounted around the turbine to detect the presence of a blade as it passes through the beam [13]. Each probe contains a fiber optic emitter for the laser beam and a fiber optic receiver that is interfaced to a photodiode to capture the reflected light. As the turbine engine rotor rotates, the reflected light produces a pulse train, as shown in Figure 1 [13], with each pulse corresponding to an individual turbine engine blade [14]. A triggering circuit, clocked in the tens to hundreds of megahertz, captures the time of arrival of each individual blade by generating a logic pulse and counter value at the same time that the probe signal exceeds a voltage threshold determined by the system operator. The triggered signals are also correlated to a once per revolution signal to identify the particular blade associated with each pulse, and an Inter-range Instrumentation Group (IRIG) time source for a precise timestamp for the calculated vibration data. The timing data is then communicated to the processing PC for the signal processing extraction of vibration frequency, phase, and amplitude [1].

Although deducing the time of arrival of each blade through the arrangement previously described has been proven as a reliable technique, this setup can introduce errors due to incorrect and inconsistent triggering of the pulse pattern in the analog triggering module. The time of arrival used to extract vibration information is heavily dependent on the adjustable threshold and data resulting from an incorrectly triggered waveform are propagated through the signal processing algorithms. As shown in Figure 2 [15], the current configuration of the NSMS system only saves data determined from the triggered time of arrival and incorrect triggering cannot be corrected. The NSMS probe recorder was envisioned to correct this problem by capturing the raw signals from the probes for future reprocessing to prevent incorrect conclusions from badly triggered timing data.

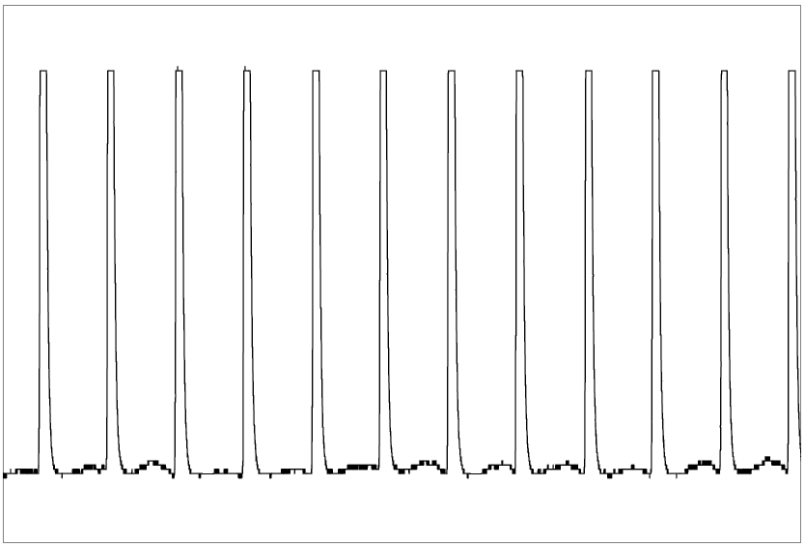

Figure 1: Example probe signal

(from [13])

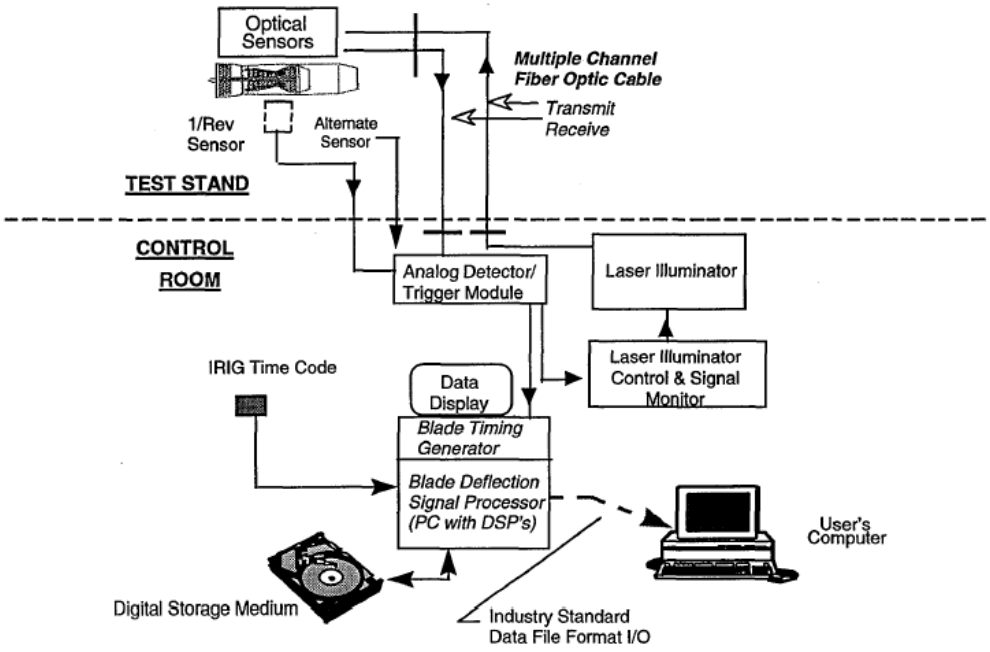

Figure 2: 4th Generation NSMS System

(from [15])

Saving the raw waveforms with a probe recorder also provides the opportunity to fine tune the measurement results. Even though the data may be triggered properly with respect to the correct number of detected pulses, different threshold values can produce different timing measurements due to variation on the edge of the signal. Having the data saved and accessible for review before setting trigger points and reprocessing will enable the analyst to determine the best possible threshold for data accuracy. Even though this recorder is specifically designed and developed as a complement to the NSMS system, the capability to record and extract high speed signals for long durations of time without interruption can obviously be applied to many different applications that have similar demands.

Before development of the probe recorder, the design criteria, presented in chapter 2, were analyzed and clearly defined. Based on the defined specifications, chapter 3 discusses the selection, design, and integration of the hardware components. Custom software was developed to control each hardware component and the interconnecting data links and is explained in chapter 4. The analysis and validation of the probe recorder is presented in chapter 5. As an additional feature for the system, data compression was implemented to reduce the demand on storage space. The investigation of data compression methods is discussed in chapter 6.

2. Determination of System Requirements

Several characteristics of the raw signals produced from the photodiodes had to be considered so that probe recorder system would not miss any important information. As the vibrations deduced from the timing data have a frequency component, all frequency content of the raw signals pertaining to vibrating blades must be present in the recorded waveforms. Since a voltage threshold is used to mark the time of arrival of the turbine engine blade, the digitized waveforms must have adequate resolution to prevent significant quantization errors from appearing in the signals. During data acquisition, the entire system should also be able to provide enough data throughput for continuous signal capture. The system also has to simultaneously record at least 32 channels to match the channel count of the NSMS system.

The sampling rate for the probe recorder analog-to-digital conversion process was derived using the specifications of the signal receive portions of the laser probes and the expected velocity ranges of the monitored blades. The probe recorder must be able to fully characterize the rapid rise time of the blade pulse for accurate triggering during playback to the NSMS system. The rise time, T_{rise}, of the reflected laser pulse width can be calculated using the formula, $T_{rise} = W_l / (\cos(\alpha_o) * 2\pi * R_b * \omega)$, where W_l is the width of the transmitted laser beam, α_o is the angular offset of the probe, R_b is the radius to the tip of the blade, and ω is the rotational velocity in revolutions per second.

Each of these variables has a wide range of possible values. The energy distribution width of the laser can also vary depending on the optimal focal distance of the lens at the end of the transmitting fiber optic. Probes are also placed at varying angles due to physical mounting constraints. Any angular offsets will increase the laser width, therefore increasing the rise time and reducing the sampling rate required. The range of one of these variables is also affected by the value of other variables; for example, turbine engines with high rotor speed ranges will typically have smaller blade tip radii.

Using the worst case values for each variable – 333.33 rev / sec for rotational velocity, 25 in for blade radius, 0 degree angular offset, and 0.023 in for the width of a perfectly focused laser probe – produces a rise time of 0.439 microseconds. In relating the rise time to the required sampling frequency, the bandwidth of a pulsed signal is $F_{knee} = 0.5 / T_r$, where F_{knee} is the highest frequency of interest and T_r is the rise time of the signal [10]. Using this equation, the highest frequency of interest is 1.139 MHz and the Nyquist sampling frequency is 2.277 MHz. In practical situations, the probes are often mounted at a 45 degree angle to the blade. This variation changes the fastest rise time to 0.621 microseconds, which results in a sampling frequency of 1.61 MHz. Since it has been shown, in [11], that the error is less than 2 percent when the rise time of the recorder is at least 3 times the rise time of the signal, the practical sampling frequency should be at least 3 times the calculated minimum. This extra factor results in a desired sampling frequency of 4.83 MHz, which is rounded up to 5 MHz for simplicity.

The required resolution of the digitizer can be determined from the voltage output limits of the photodiodes used to capture the reflected light. The pulse generated by the return signal is limited to ± 3.5 V, which is a 7 V dynamic range. Time of arrival triggering on the edge of the pulse waveform is quoted to have a precision of 5 mV. Using these values and the formula, $N_{bits} = 1 + \log_2 (\text{range} / \text{resolution})$ [2], the number of bits required to quantize the waveform is 11.45 which, for practical purposes, is rounded up to 12.

Using the calculated values above, the continuous throughput of the NSMS probe recorder will have to be 5 MHz multiplied by 12 bits, or 60 megabits per second per channel. As the system is expected to record at least 32 channels, the overall throughput will need to be 1920 megabits per second (240 megabytes per second). This bandwidth will need to be available in all components of the recording computer so that the complete system can provide the sustained data rate. The key components to be scrutinized are the peripheral bus that will communicate with the digitizing hardware, the core processor and RAM to handle the data exchange, and the disk writing speed. The known limitations present in most computers are the PCI standard bus that operates at

a theoretical maximum of 132 MB/sec, and the standard disk writing speed in the range of 30 to 40 MB/sec.

3. Hardware System Design

After conducting a review of the available high speed analog-to-digital hardware, the decision was made to use the ICS-645 analog input board [3], shown in Figure 3, as the main component of the probe recorder. Comparing the specifications to the values calculated in the previous section, the maximum sample rate per channel of the ICS-645, 2.5 MHz, is half the ideal theoretical value. This deviation was deemed tolerable for the immediate future as this sample rate will be adequate for practical data acquisition. The 16-bit digitization of the input voltage exceeds the required 12-bit resolution, and the 32 channels available are exactly the amount needed for the system. The combination of a 2.5 MHz sampling rate, 2 bytes per channel, and 32 channels yields a maximum data rate of 160 MB / sec.

The ability to closely meet the required specifications with a single board was a significant advantage compared to other commercial solutions. At the time of the market survey, systems that require several boards with a backplane communication bus were available that would have met the theoretical specifications, but the slight increase in performance translates to a significant increase in cost and size.

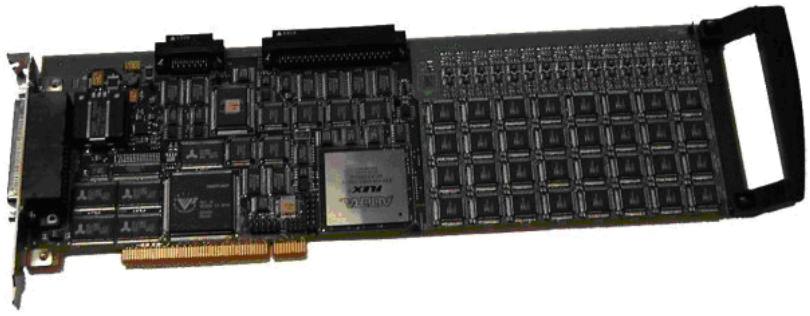

Figure 3: Picture of ICS-645 analog input board

As shown in Figure 4, the ICS-645 digitizes each channel signal with a dedicated analog-to-digital converter chip (Analog Devices AD9260) to achieve a guaranteed sample rate per channel. When operating with all thirty-two channels at 2.5 MHz, the A/D chip incorporates an 8x decimation filter to reduce the output rate from the clock rate of 20 MHz. This filter also prevents aliasing of frequencies above the Nyquist rate [4]. The digital data is then packaged based on data transfer settings and sent to the appropriate buffer before being transmitted onto the selected bus [3].

The expected data rate in the previous section, 240 megabytes per second, has been reduced by the new specifications to 160 megabytes per second. The ICS-645 has two inherent options for output data transfer, a standard PCI connection and a secondary Front Panel Data Port (FPDP) bus. The standard PCI bus has the limitation of 132 megabytes per second and can only be used when the number of channels and the sample rate per channel have simultaneously been reduced to lower the required bandwidth. The FPDP transfer protocol specifies a maximum transfer rate of 160 megabytes per second. In practical use, the FPDP bus can be slightly overclocked to provide timely transfer of the entire data stream at 160 megabytes per second and the necessary overhead of the communication protocol.

Since the board cannot handle the maximum data rate across the standard PCI connection provided, the secondary Front Panel Data Port (FPDP) bus of the ICS-645 was used to transport the digitized data. The ICS-500 (Figure 5) was added to the system to act as a bridge between the FPDP bus and the extended bandwidth of the 66 MHz, 64 bit PCI-X slot of the computer. The PCI-X transfer protocol provides an increased maximum bandwidth of 528 megabytes per second, which eliminates the bottleneck of transferring data from the analog-to-digital converters to the computer. As shown in Figure 6, the ICS-500 consists of the bus interfaces for FPDP and PCI-X with the buffering and translation circuitry in between the two communication protocols. Since the ICS-500 is packaged in the PCI Mezzanine Card (PMC) form factor, the module was physically mounted on a PMC to PCI-X carrier board inside the host computer [5].

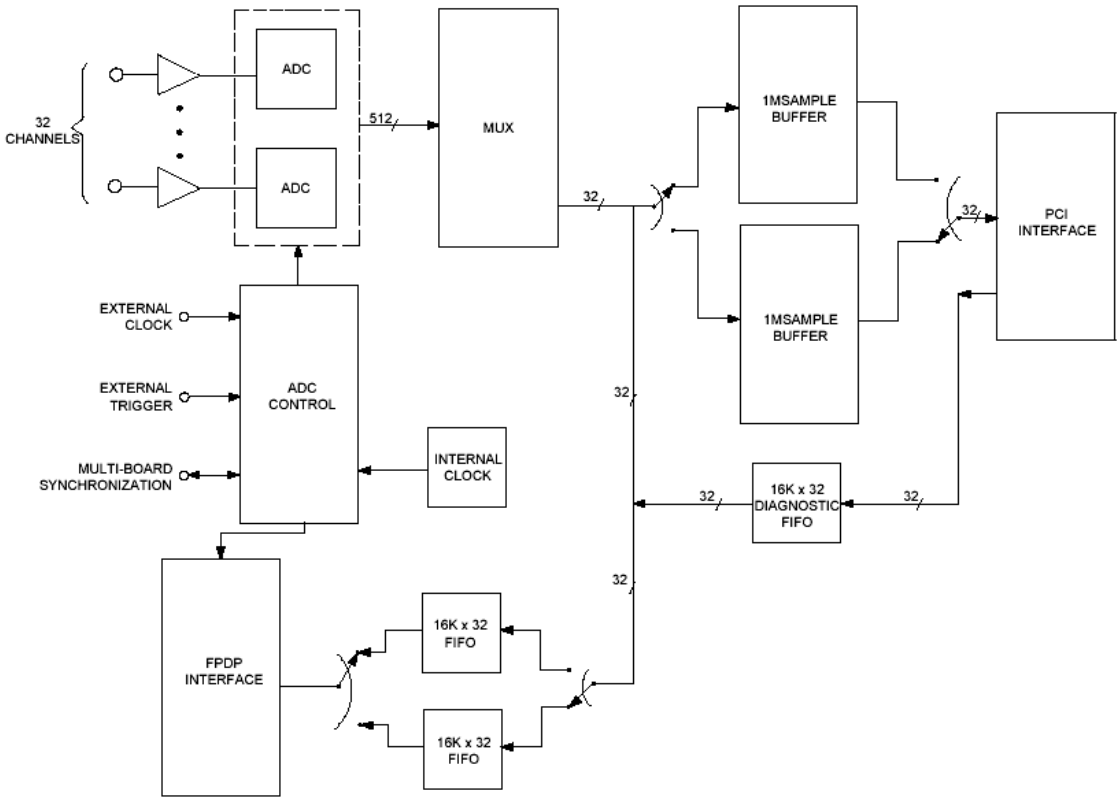

Figure 4: ICS-645 block diagram

(from [3])

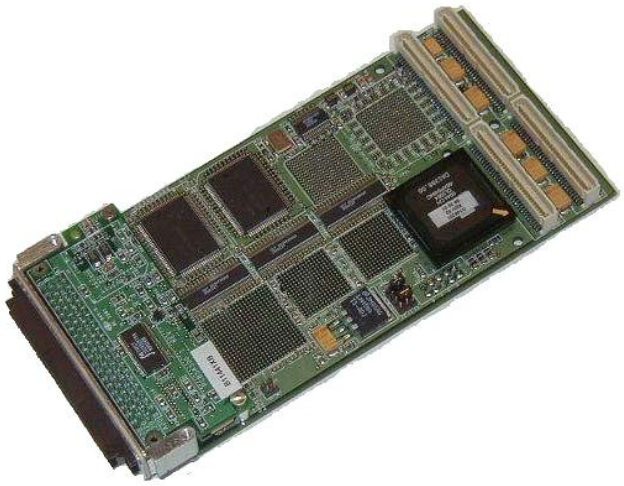

Figure 5: Picture of ICS-500 FPDP to PCI-X adapter

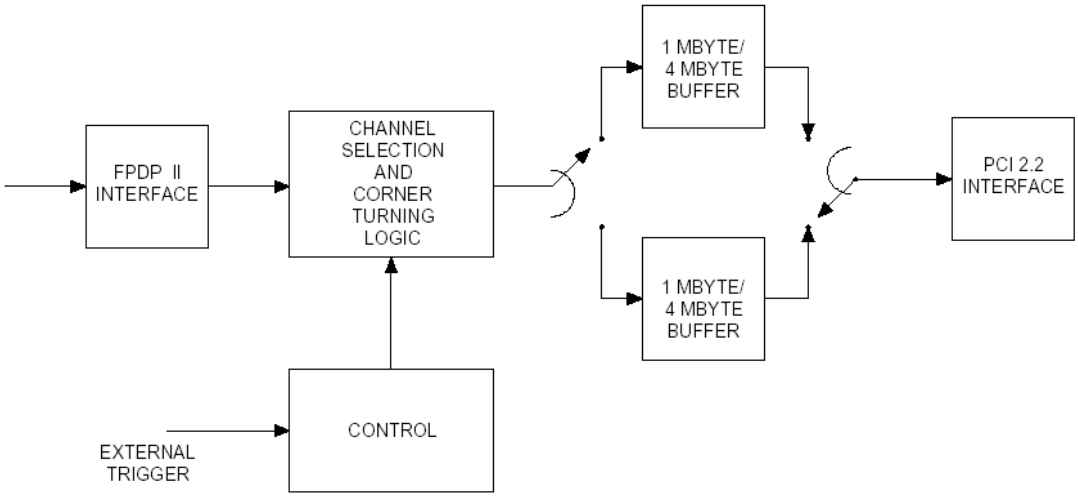

Figure 6: ICS-500 Block Diagram

(from [5])

One unintended consequence of choosing the off-the-shelf hardware for the probe recorder was the limitation of the signal input voltage range. The ICS-645 has a ±1V limit for each channel. An additional 32 channel adjustable gain box was designed so that the input signals will be appropriately attenuated or amplified before they are introduced to the digitizer board. A schematic of the two channel amplifier is shown in Figure 7, and this was duplicated fifteen additional times to accommodate thirty-two channels. The main component of the amplifier circuit is the THS7002 programmable gain amplifier. This component contains two pre-amplifiers for the input signals and two amplifiers that can be adjusted by the Dual In-line Package (DIP) switches in the schematic. The gain can be set to one of the following eight values: 10, 5.01, 2.52, 1.26, 0.63, 0.32, 0.16, and 0.08. Designed for high speed applications, the THS7002 has a bandwidth of 70 MHz, which is far above the bandwidth of the probe recorder [6].

As displayed in Figure 8, the probe recorder computer was built to specifications that would ensure reliable data throughput. To receive the data through the PCI-X bus, the host computer allocates a section of random access memory (RAM) for direct memory access (DMA) transfers [3]. After the data has been received into the allocated memory, the software transfers the data to the hard drive for storage.

To provide adequate space and bandwidth for data transfers and available memory for running applications, the computer was equipped with 4 gigabytes of DDR400–PC3200 RAM. According to the specifications of Double Data Rate (DDR) memory [7], the clock rate is 200 MHz and the maximum available throughput is 3200 megabytes per second. With the large amount of memory working in conjunction with two 2.2 GHz Opteron dual core processors, the probe recorder data acquisition software has an abundance of internal resources to transfer data from busses to disk and execute additional programs.

The final step in the data path is the permanent storage of the digital waveform. Standard single disk drives are limited to a practical throughput rate of 30 to 40 megabytes per second due to the maximum revolutions per minute of the spinning disk. The 160 megabytes per second throughput needed for the probe recorder is comparable to the storage needs of high speed storage in server architectures. To

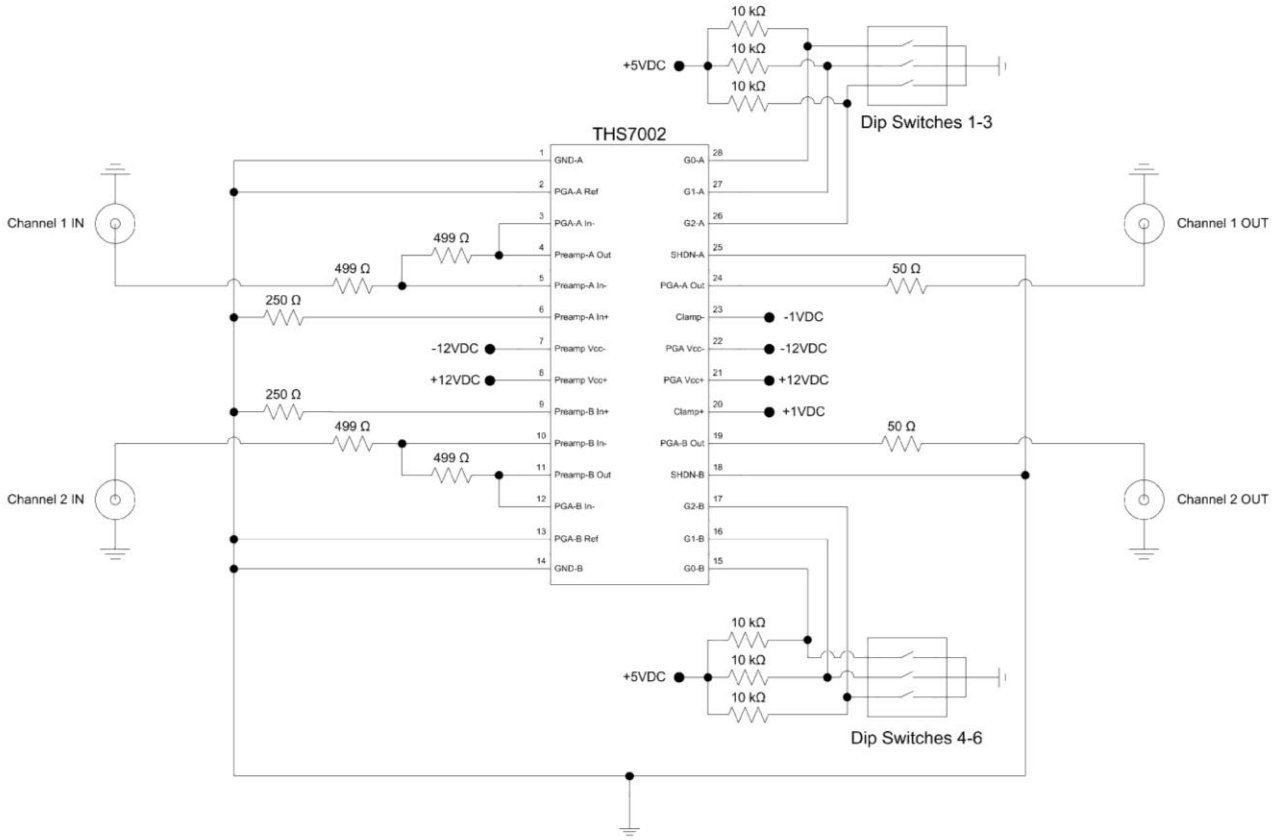

Figure 7: Schematic of 2 channel selectable gain amplifier

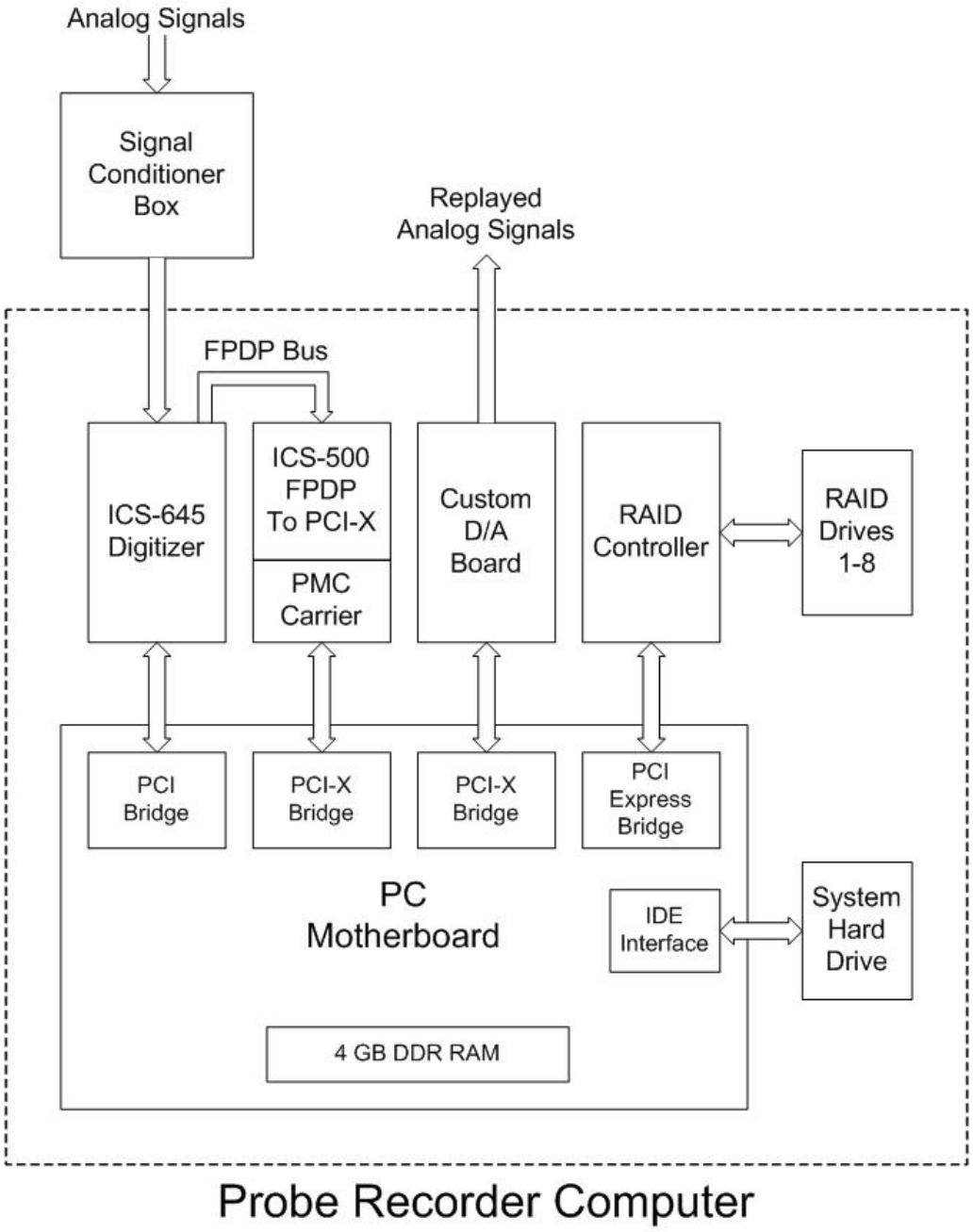

Figure 8: Probe Recorder hardware diagram

overcome this technical hurdle, servers are often equipped with Redundant Array of Inexpensive Disks (RAID) controllers that coordinate an array of individual drives to act as one storage area [16].

For the probe recorder computer, the RAID controller uses a PCI Express x8 connection, capable of 4 gigabytes per second, for the initial data transfer from memory [17]. The controller then is connected to eight separate 500 gigabyte hard drives and the configuration is set to operate as a single drive (RAID 0 configuration). Higher transfer speeds are achieved due to the RAID controller, in a process called striping, segmenting the data and distributing the information among the individual drives. This setup theoretically allows eight times the transfer speed of a single drive system.

The RAID array is also necessary for the system due to the volume of data that must be stored during operation. When the incoming data rate of 160 megabytes per second is converted to 9.375 gigabytes per minute or 562.5 gigabytes per hour, the need for a large storage capacity is evident. Utilizing the RAID array to provide four terabytes of storage space allows the user to record and save 7.28 hours worth of data.

Figure 9 shows the overall data flow for the incoming signal as the signal is digitized and stored in a file by the probe recorder. All components of the analog front end of the system have been selected to ensure bandwidths are greater than the 1.25 MHz Nyquist frequency so that no frequency content is removed before digitizing the signal. The data transfer paths have been chosen so that each transfer will operate at a greater speed than the maximum signal rate of 160 megabytes per second.

For the digital-to-analog (D/A) playback portion of the system, a custom board was developed by an electrical engineering colleague, Terry Hayes, because of the unavailability of commercial products, in 2006, that would meet the required channel count and bandwidth. Due to the shortage of chipsets for high speed busses such as PCI Express during design, the playback board utilizes a standard PCI interface chip for the computer interface. The bandwidth limitation of the PCI bus, 132 megabytes per second, changed the design of the board to a channel selectable design rather than generating all channels within a file. Figure 10 shows the overall data flow for the playback data, which is similar to a reversal of the recorded data path.

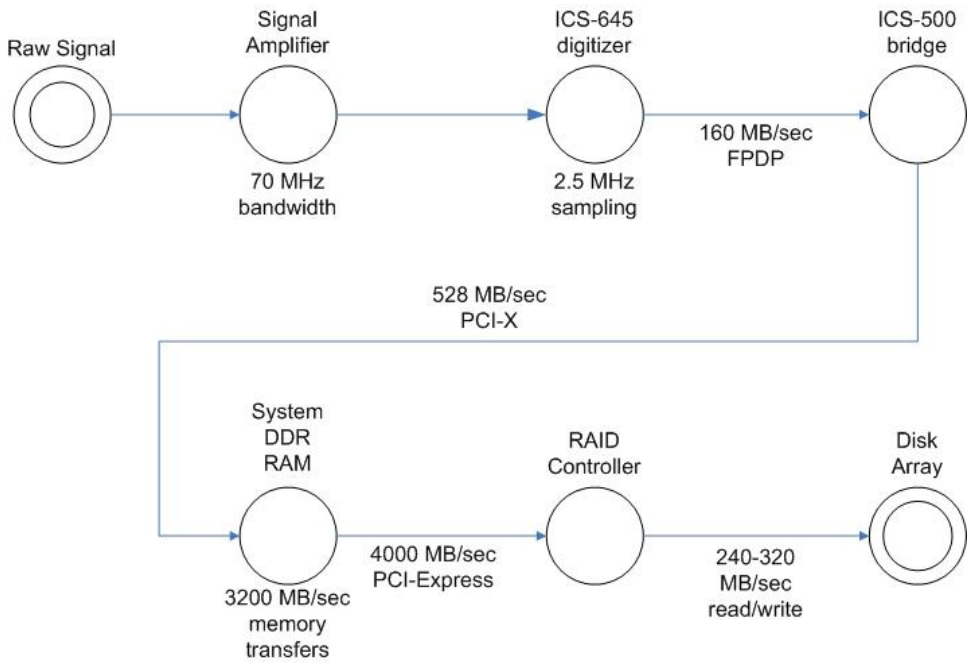

Figure 9: Data flow diagram for recorded signals

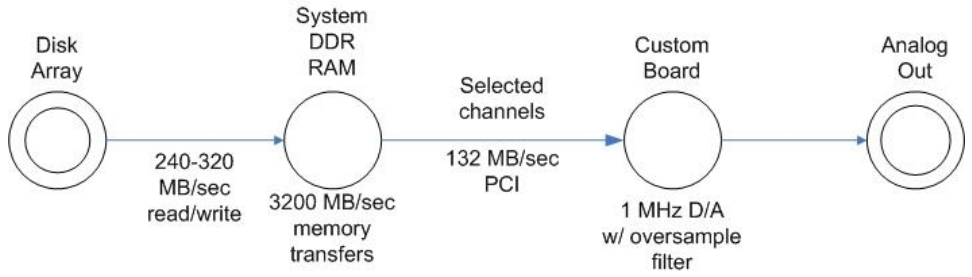

Figure 10: Data flow diagram for playback signals

The main component of the custom playback solution is the Texas Instruments DAC8580 D/A chip, which has 16-bit resolution, an oversampling filter, and frequency limitations of 1 MHz with interpolation and 2 MHz without [8]. The oversampling filter reduces the “stair step” jumps in the reproduced data that is inherent to sample-and-hold D/A circuits. This filtering was deemed as a higher priority as the NSMS system trigger circuitry is clocked at a faster frequency than the recorded 2.5 MHz. For data sampled at higher frequencies than the above limitations, a process was defined to playback the data at the highest rate and to apply a time based correction in the NSMS triggering system when processing the signals.

4. Software System Design

Data acquisition, file viewing, and playback are the essential software components of the probe recorder system. The data acquisition software controls the communication with the digitizing hardware and saves the data to the disk array in a continuous operation. The file viewing program allows a user to replay a recorded file for data verification and also provides selected tools for file modification and analysis. The playback interface controls the selection of channels and replays the chosen data through the D/A board.

Due to familiarity, availability, and compatibility, all software for the probe recorder system was developed using National Instruments LabWindows CVI software. This platform provided the versatility of the C programming language with easy integration for graphical interfaces. Interfacing the hardware was also simplified because the manufacturer supplied C-based device drivers for each component.

As with the selection and design of the hardware components of the probe recorder, the real time operation must be ensured in the data acquisition software. For visual reference, the front panel of the data acquisition program is shown in Figure 11.

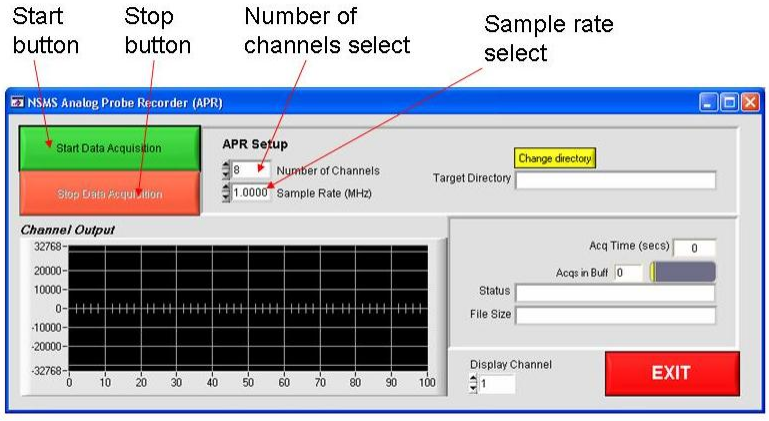

Figure 11: Probe recorder acquisition front panel

The execution flowchart for the data acquisition software in Figure 12 shows the steps necessary for controlling the hardware described in the previous section. Multithreaded operation is necessary to avoid stalls in the program execution due to sequential commands. Multiple threads also exhibit better utilization of the four processor cores available on each probe recorder. When the user initiates the data acquisition, the program checks for the critical parameters of number of channels, sample rate, and a location to store the data files. If all values are valid, the multithreaded operation begins by allocating a large, thread-safe buffer to store multiple acquisition blocks and starting the two real-time threads. The threads are divided into a communication thread for retrieving the data from the hardware and a saving / display thread for writing the data to disk and updating the graphical displays for signal monitoring, acquisition time, and file size.

To acquire the incoming signals from the A/D hardware, the communication thread first initializes each board with the proper settings for channel count, sample rate, and the custom parameters for communication and board operation. After these values are loaded, the onboard memory buffers for FPDP and PCI-X transfer and receiving PCI-X buffer in host memory are allocated for one acquisition data block based on 16 bit values for each sample, 32768 samples per channel, and the user-specified number of channels. Once the target memory locations are set, the boards are enabled and triggered for data acquisition.

As the data acquisition thread is preparing the pipeline for incoming signals, the file saving and graphing thread prepares for its operation. Utilizing the channel count entered by the user, a thread safe buffer for 300 acquisition blocks is initialized as an intermediate location to optimize the interaction between acquisition and disk writing operations. After checking the target file saving location for validity, a file with the current date and time in the filename is created and the custom header with the acquisition parameters is written to this file. Memory space for one channel of one block of data is allocated for the real time waveform display. At this point, each thread enters into a loop for continuous operation until the user issues a stop command.

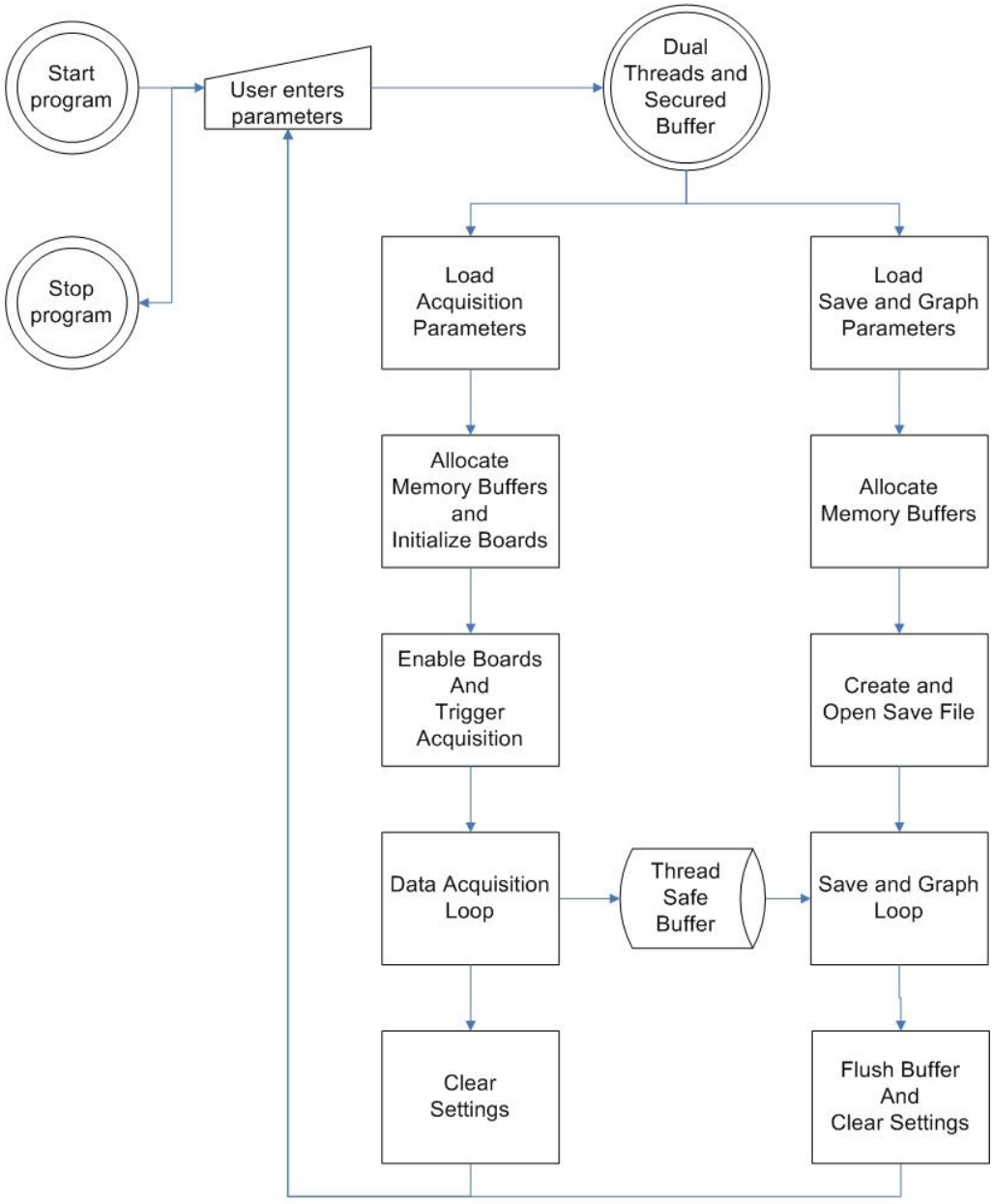

Figure 12: Execution flowchart for data acquisition software

To ensure optimum performance during data acquisition, the loops only execute the minimum amount of operations. The execution and interaction flowchart for each loop is shown in Figure 13. For the loop controlling the digitizing hardware, the software monitors the PCI-X bus for the incoming acquisition block, writes the data to a fixed location in system memory to free the onboard buffers, and then transfers the data out of the DMA buffer into the thread safe buffer. After the data has been placed for the saving and graphing thread, a thread safe variable for the number of blocks in the thread safe buffer is incremented. Before checking for the next interrupt from the acquisition hardware, the loop checks for the exit flag to stop acquisition.

During the execution of the save and graph loop, the number of blocks for the thread safe buffer is monitored for a value greater than 25. This value was chosen through several laboratory iterations as a compromise between large data arrays for optimum disk writing and frequent polling of the thread safe buffer to avoid buffer overrun in case of momentary system interruptions. When the number of blocks in the thread safe buffer meets or exceeds 25, the data is pulled from the buffer into a temporary array in memory and then written to the RAID disk. Since the thread safe buffer operates in circular fashion, once the data is pulled, the memory space previously occupied is freed for use for the incoming data. The first acquisition block for the user selected channel is also displayed at this time on the user interface for monitoring signals. Once the data has been written to disk, the number of blocks variable is decremented by 25. This variable is also displayed on the user interface as an indication of system performance. After writing the data, the current file size is checked for a size greater than 10 gigabytes. If this condition is true, a new file is created for easier post-processing of data. If the user has sent the command to stop acquisition, the thread pulls all remaining data in the buffer, saves the data to disk, updates the number of acquisitions in the file header, and stops execution.

As shown in Figure 14, the acquired data is saved in a simple binary file with a header space containing the acquisition parameters followed by the blocks of data. The probe recorder header contains a structure that includes the number of blocks in the file,

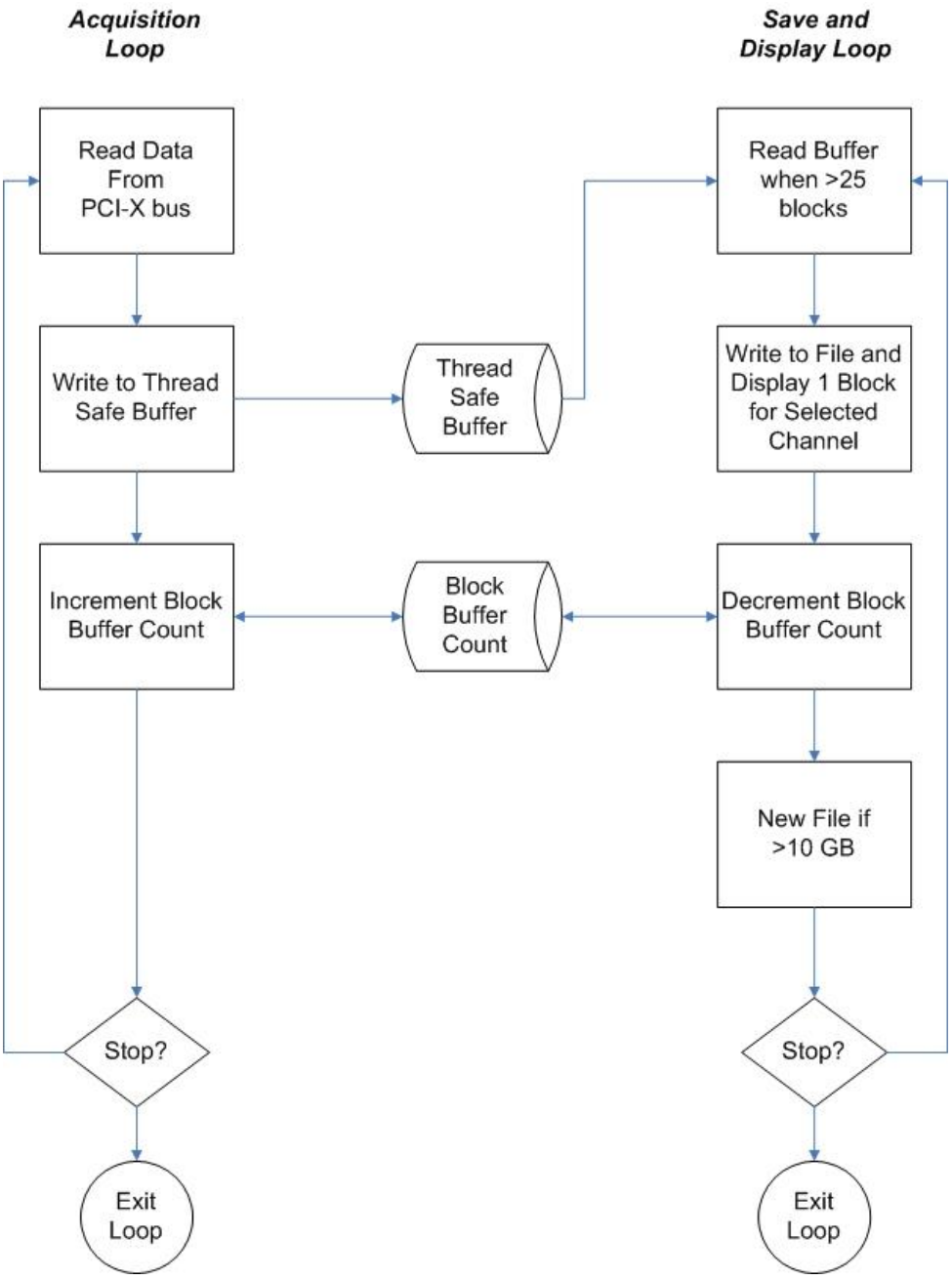

Figure 13: Execution flowchart for continuous loops

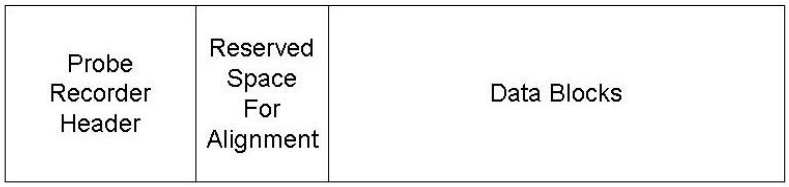

Figure 14: Probe Recorder top level file structure

the number of active channels, the sample rate in megahertz, and the number of samples in each data block. Since the system file cache is bypassed during data acquisition, extra bytes are allocated after the header to align the data in an integer multiple of the disk sector size due to the requirements of non-buffered disk writing in a Microsoft Windows environment [12]. Data blocks are saved sequentially after the headers. The data for individual channels inside the data blocks is interleaved due to uncorrected transfer from the onboard buffers to disk. Since the ICS-645 acquires blocks of data in binary multiples, the data blocks are inherently sector aligned for non-buffered disk writing.

To extract the information in these custom files, a file viewer was developed. The front panel of this application is shown in Figure 15. Since each file can contain a large amount of data, a sliding window approach to file loading was adopted. Once the user selects the file of interest, the playback controls are used to scan through the data blocks while the underlying code uses the current file pointer to control the current file section. For further analysis of the captured data, this program also allows the user to export sections to text files, create smaller data files from larger files, and splice together files to form one continuous data file. FFT analysis of the recorded data is also available as an option.

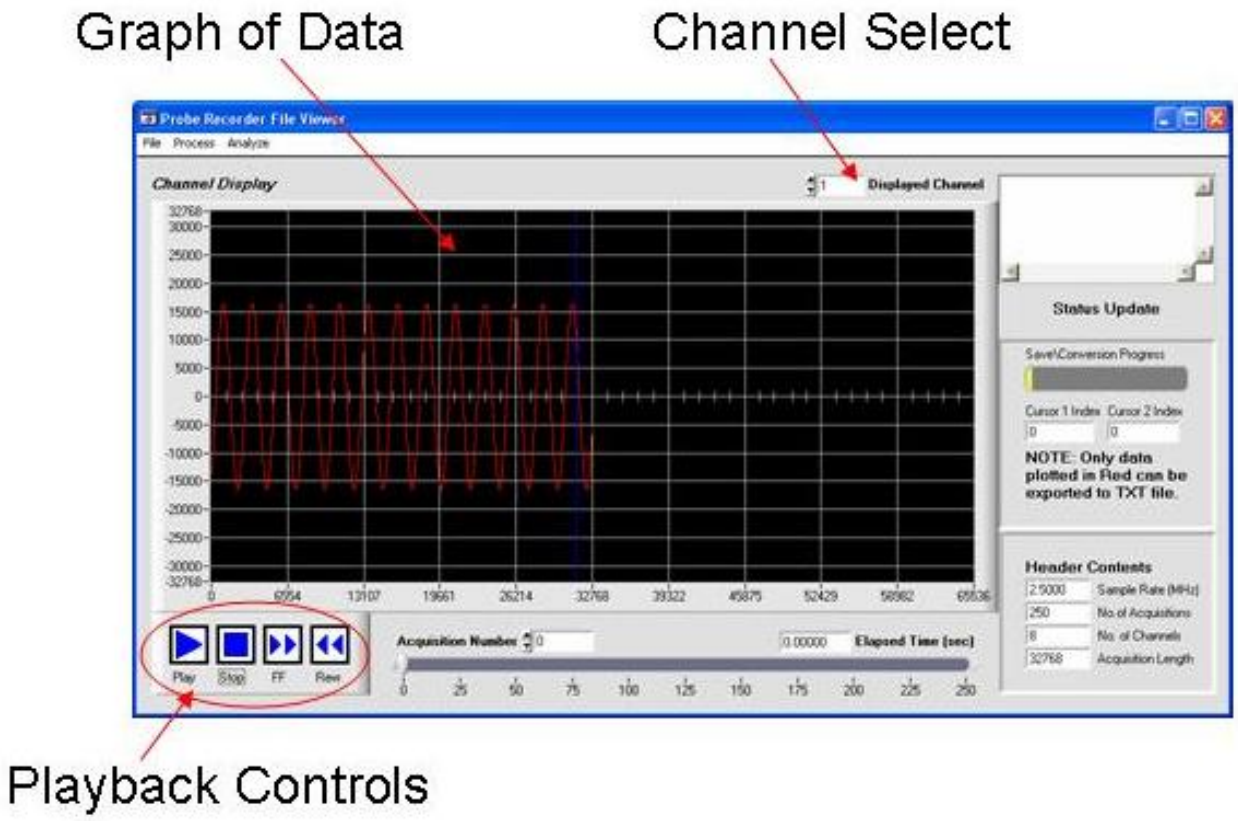

Figure 15: Probe Recorder file viewer front panel

5. Checkout and Results

To verify the capabilities of the probe recorder, three types of tests were conducted to measure important characteristics. The write speed of the RAID array was tested with a custom program duplicating the same methods of the acquisition program and timing the disk operations. To quantify differences between analog signal inputs and the recreated analog signals, the overall transfer function of the system was characterized by introducing an impulse data set to the playback portion, recording the analog waveform with the digitizing portion and with an oscilloscope, and comparing this result with a simulation of the known effects of the components. The data throughput was tested with a function generator input while the system was operated at maximum conditions.

For the RAID array test, a special program was written to test the disk writing and reading speeds for different sizes of data blocks. Since the largest acquisition block size available was 32768 samples per channel, or 65536 bytes per channel, and the ICS-645 is only configurable for even numbers of channels, the read and write tests were conducted in multiples of 131072 bytes up to the maximum size of 2097152 bytes. These different sizes were applied to four different writing methods – normal writing using the system file cache, the write-through method where the data is written to the cache and immediately flushed to disk, non- buffered writing where the system cache is neglected for data writes but the file information is cached, and the combination of write-through and non-buffered, where all file writes bypass the system cache and are immediately flushed to disk. The same four file accessing methods were used on a temporary file in the read test. Opening the file for non-buffered access followed by flushing the file system cache was necessary between methods to clear any existing references to the file in system memory and obtain true performance statistics.

To test the program, the tests were first run on a single disk drive through a standard IDE connection. A sample test was conducted with smaller data block transfers to confirm the need for the large data blocks. The results are shown in Figure 16 and clearly show that disk performance increases as the write block size is

increased. The single disk drive was then used to verify the speed testing program and the results are shown in Figure 17. An interesting result of this verification was the better consistency achieved by using non-buffered and write- though methods compared to the normal cached writing. Using the same program, the RAID array was tested with the eight drives arranged as one striped volume. The write test results are shown in Figure 18 along with a threshold line at 160 MB/sec for comparison. As seen from the results, the regular and write-through methods do not provide sufficient data rates over the entire range while non-buffered writing provides more than adequate performance for every scenario. The read test results are shown in Figure 19 and show that the performance for data sizes corresponding to low channel counts does not meet the required throughput. This problem can be circumvented by reading in multiple blocks of data from the disk before transferring to the playback board.

Since the probe recorder is designed to capture analog waveforms and reproduce the same analog signals at a later time, the system transfer function is essential knowledge for knowing the differences between the original analog inputs and the reproduced analog outputs. To test the system characteristics, the analog outputs of the playback portion of the system were routed into the digitizing portion so that an input set of impulse data could be compared with the digitally recorded waveforms. This input data set was constructed of impulses sampled at appropriate frequencies to yield an equivalent 2.5 MHz output signal when interpolated. The resulting analog waveform was then recorded on the probe recorder at a 2.5 MHz sampling rate and on an oscilloscope at 25 MHz, except for the 16x case which was sampled at 10 MHz due to the length of the response.

According to the DAC8580 datasheet [8], the interpolation filter is characterized by the first formula in Table 1. The expanded form of this generic equation is $H(z) = 1/N^3 * \left(1 - 3z^{-N} + 3z^{-2N} - z^{-3N} / 1 - 3z^{-1} + 3z^{-2} - z^{-3}\right)$ and the specific formulas for each case are shown in Table 1. To simulate the expected output, a program was written using National Instruments Labview software to process the input impulse using

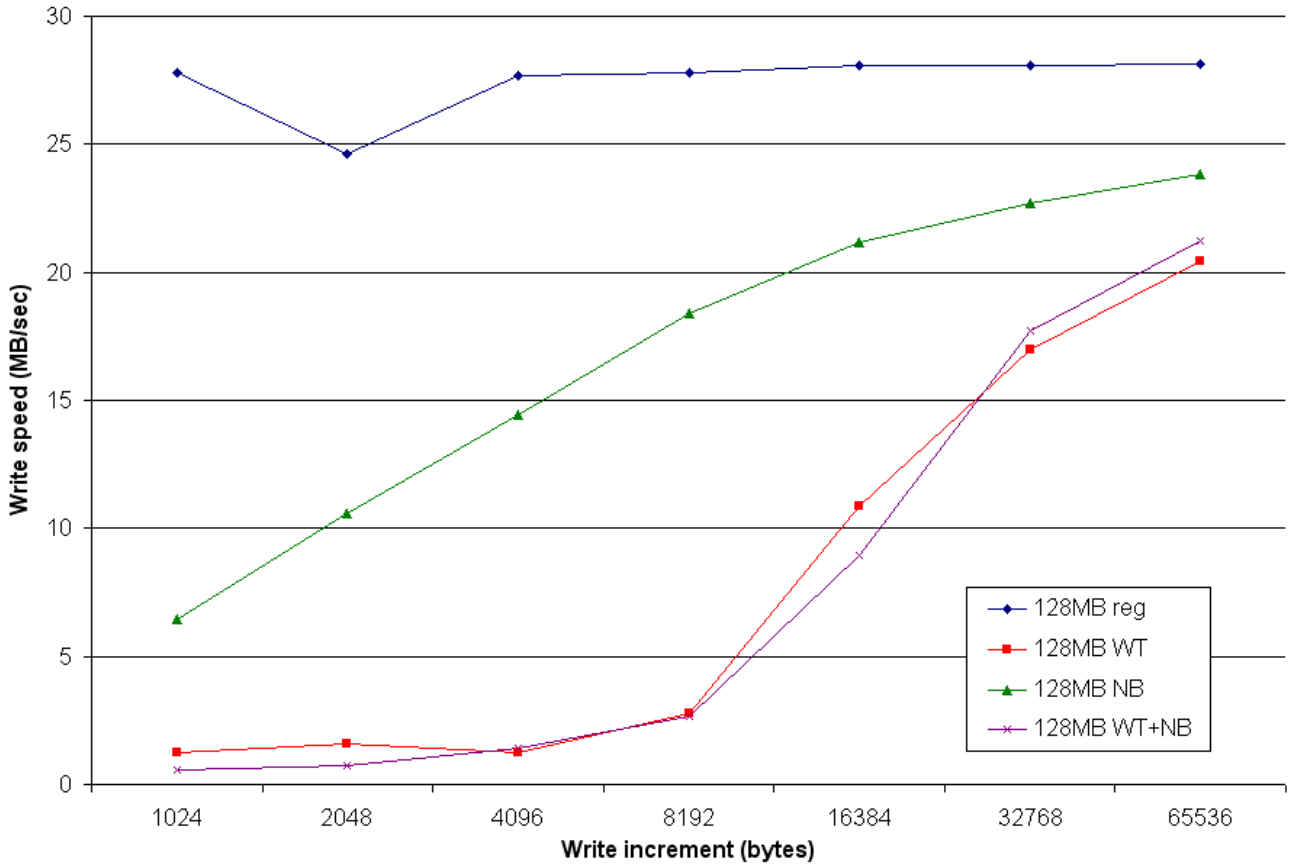

Figure 16: Low speed single disk write test results

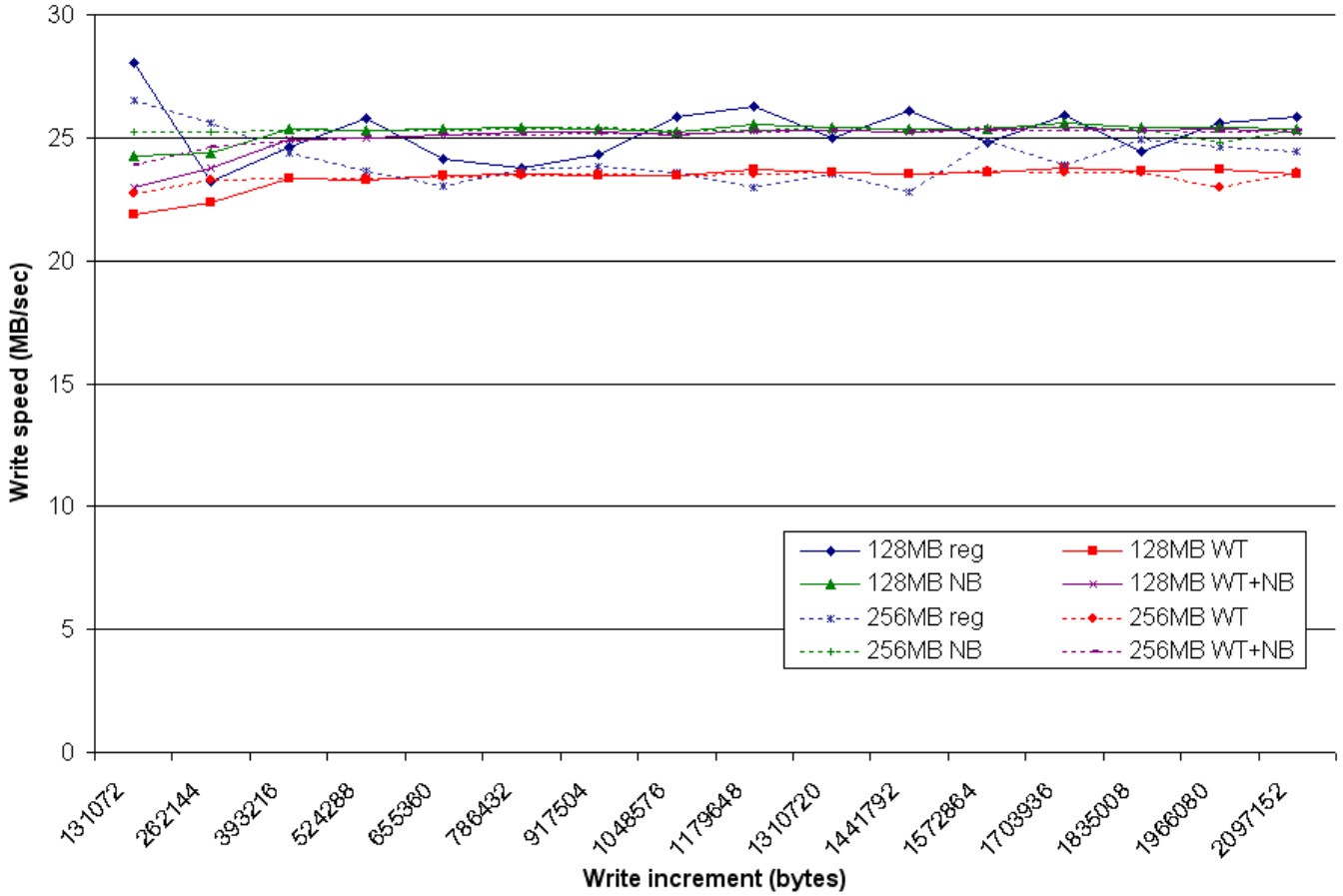

Figure 17: Single disk drive write test results

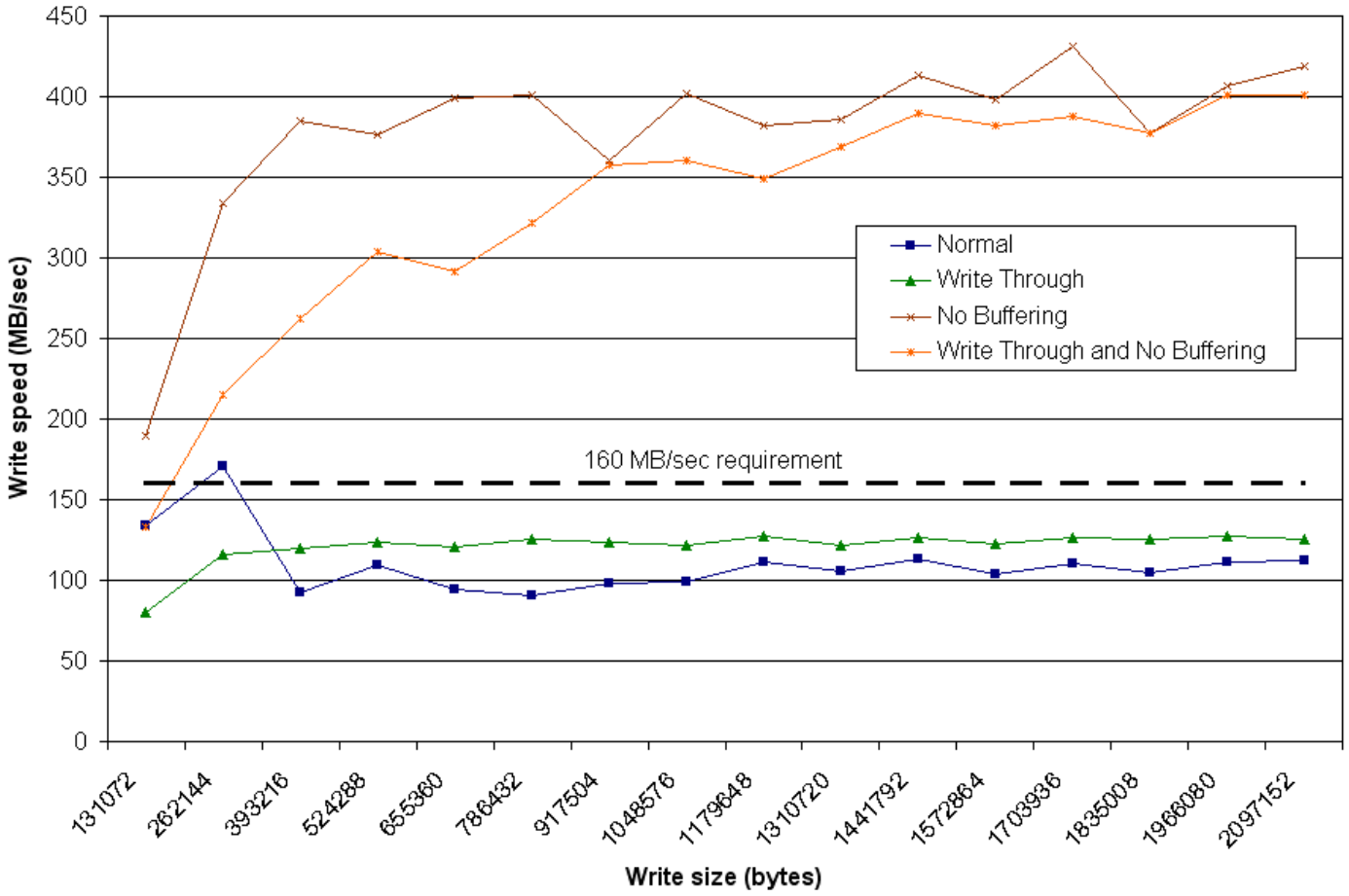

Figure 18: RAID array write test results

Figure 19: RAID array read test results

Description	Formula
Generic interpolation formula	$H(z) = \frac{1}{N^3}\left(\frac{1-z^{-N}}{1-z^{-1}}\right)^3$
2x interpolation	$H(z) = \frac{1}{8}\left(\frac{1-3z^{-2}+3z^{-4}-z^{-6}}{1-3z^{-1}+3z^{-2}-z^{-3}}\right)$
4x interpolation	$H(z) = \frac{1}{64}\left(\frac{1-3z^{-4}+3z^{-8}-z^{-12}}{1-3z^{-1}+3z^{-2}-z^{-3}}\right)$
8x interpolation	$H(z) = \frac{1}{512}\left(\frac{1-3z^{-8}+3z^{-16}-z^{-24}}{1-3z^{-1}+3z^{-2}-z^{-3}}\right)$
16x interpolation	$H(z) = \frac{1}{4096}\left(\frac{1-3z^{-16}+3z^{-32}-z^{-48}}{1-3z^{-1}+3z^{-2}-z^{-3}}\right)$
Modified 2x interpolation (simulation)	$H(z) = \frac{1}{8}\left(\frac{1-3z^{-2}+3z^{-4}-z^{-6}}{1-2z^{-1}+z^{-2}}\right)$

Table 1: Interpolation formulas

a time domain implementation of the appropriate interpolation filter. The results of the simulations and laboratory tests are displayed in Figures 20 through 24.

All of the cases matched well with the expected outputs except for the 2x oversampling case. Since the overshoot and ringing was evident on both the probe recorder and the oscilloscope and the other cases performed as expected, the problem was isolated to and assumed to be the implementation of the filter on the DAC8580. Tests of the simulation with slight modification of the filter equation were performed and a similar output was observed when the denominator was only squared instead of being cubed. The comparison between the captured data and the modified simulation is shown in Figure 25 and the modified equation is the last entry in Table 1. Due to the lack of an exact match, other unknown errors may be present.

For verification of the total system bandwidth, a 100 kHz input signal was recorded at the four different configurations that would produce the maximum data transfer of 160 megabytes per second. Since the data is 16-bit values packaged inside 32-bit and 64-bit words, the input signal was placed on the first two channels and then every third channel afterwards to create an asymmetric data array that would easily show incorrect data transfer. The four maximum configurations are 4 channels sampled at 20 megahertz, 8 channels sampled at 10 megahertz, 16 channels sampled at 5 megahertz, and 32 channels sampled at 2.5 megahertz. The 100 kHz frequency of the input was verified using a calibrated oscilloscope which measured the frequency as fluctuating between 99.7 and 100.5 kHz.

Since the probe recorder is intended to be used in continuous operation for long periods of time, duration tests for each maximum configuration were conducted. The throughput rate for each of these cases is shown in Table 2, and the results required revisiting the bandwidth expectation of 160 megabytes per second. The discrepancy was found to be related to the fact that one megahertz is one million hertz, while one megabyte is actually 1,048,576 bytes. Recalculating the expected bandwidth with this correction yields a value of 152.59 megabytes per second, which closely matches the calculated values and the error is due to the one second resolution of the time.

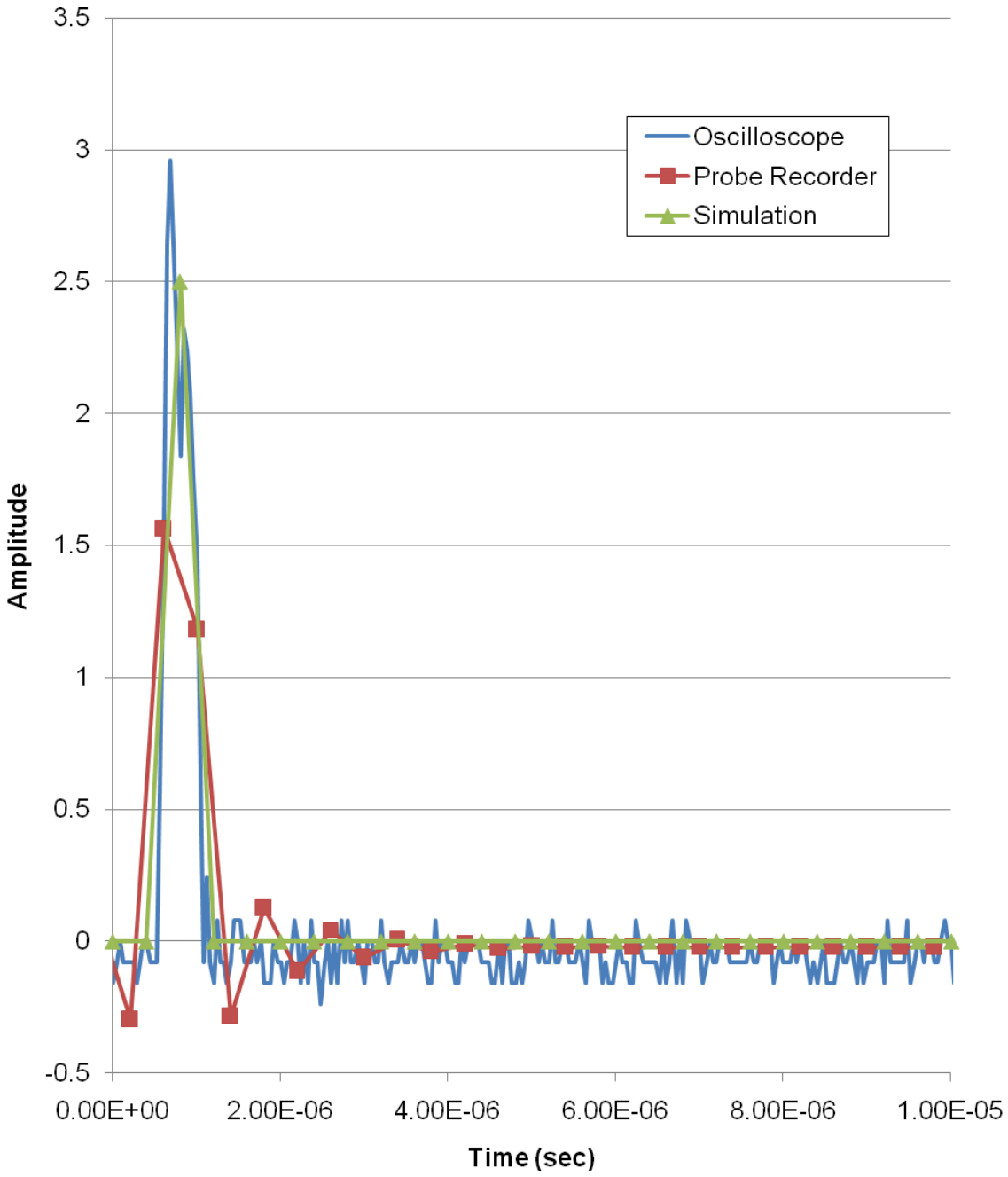

Figure 20: Impulse Response - 2.5 MS/sec, 1x interpolation

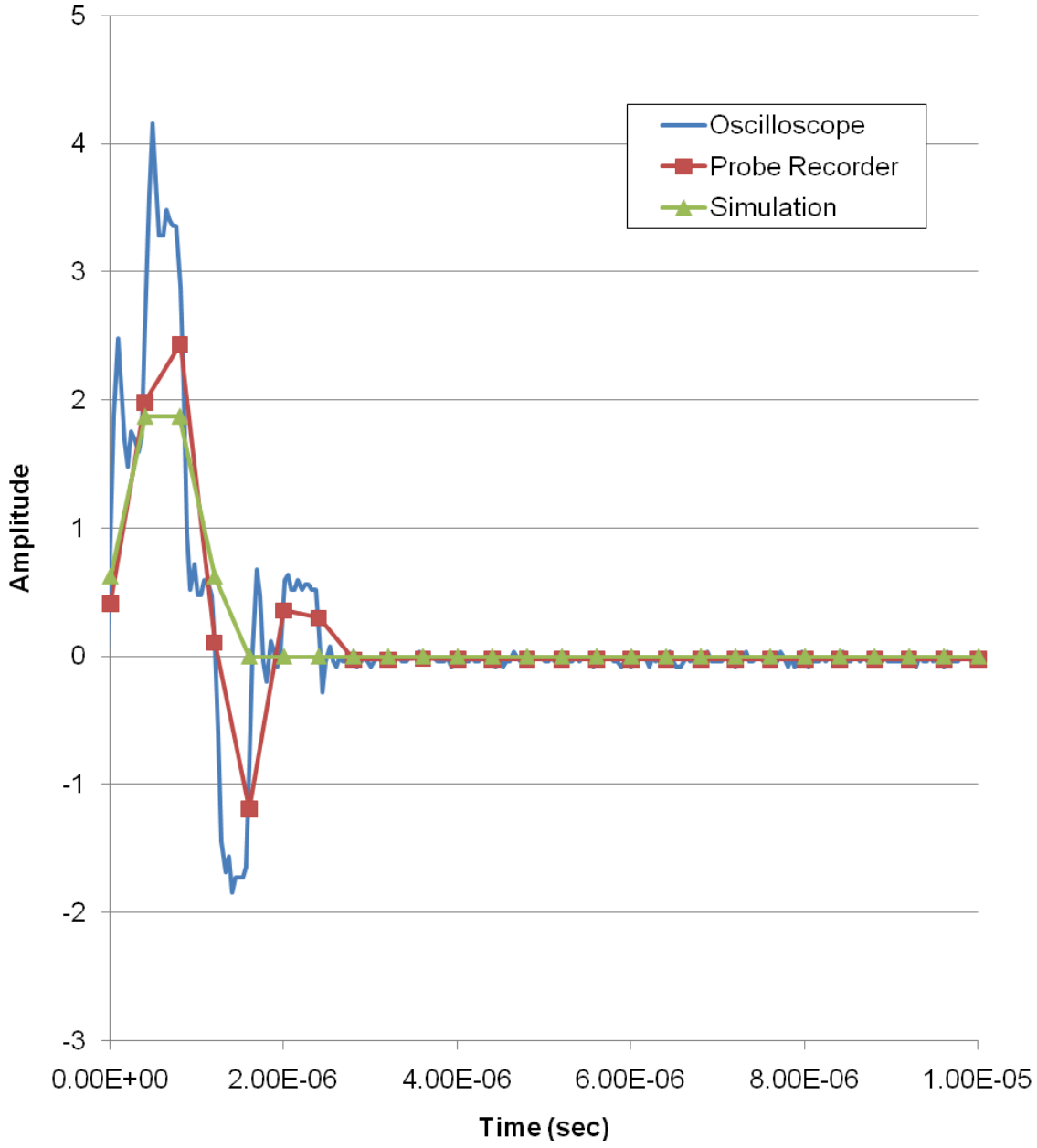

Figure 21: Impulse Response - 1.25 MS/sec, 2x interpolation

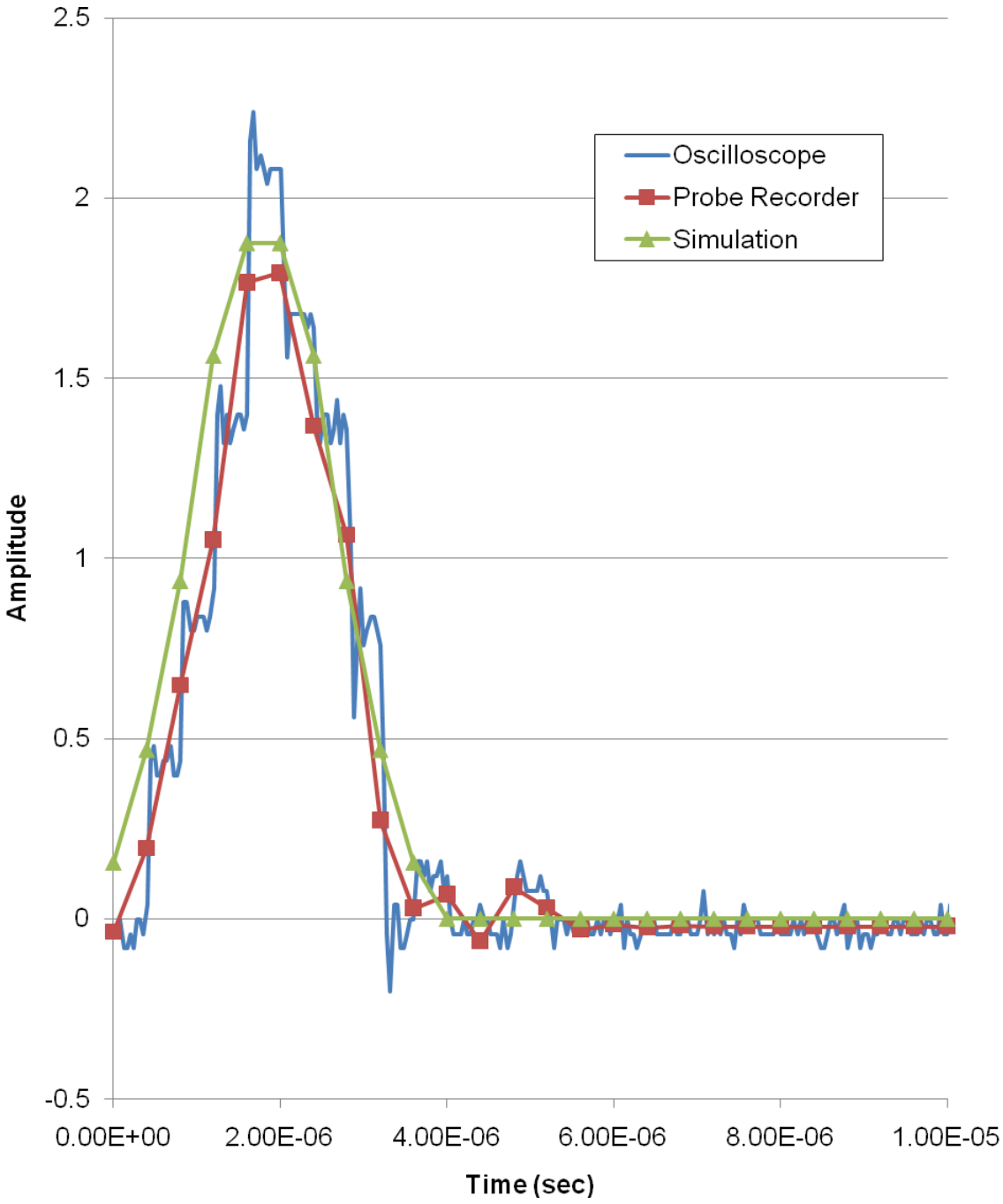

Figure 22: Impulse Response - 625 kS/sec, 4x interpolation

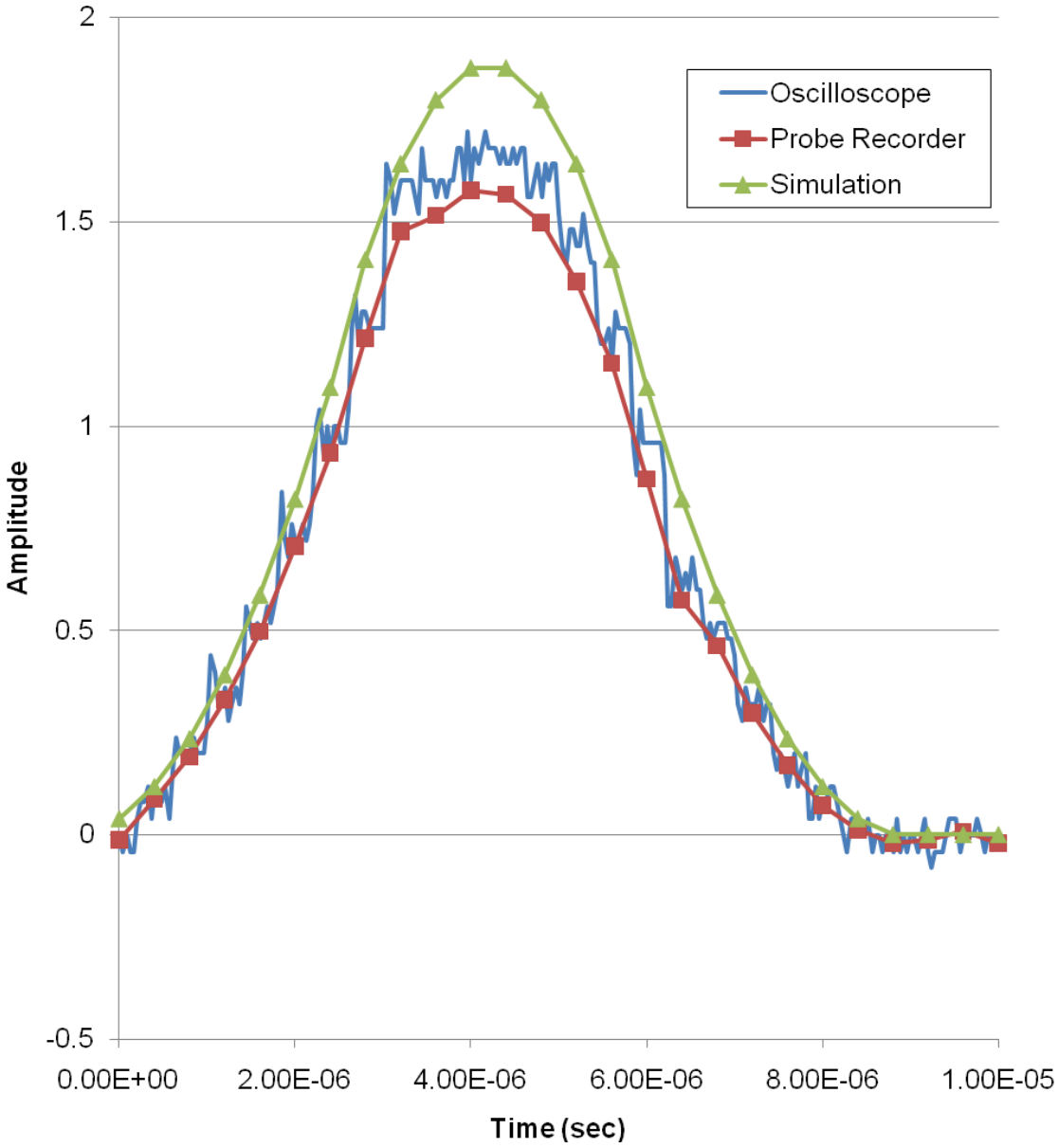

Figure 23: Impulse Response - 312.5 kS/sec, 8x interpolation

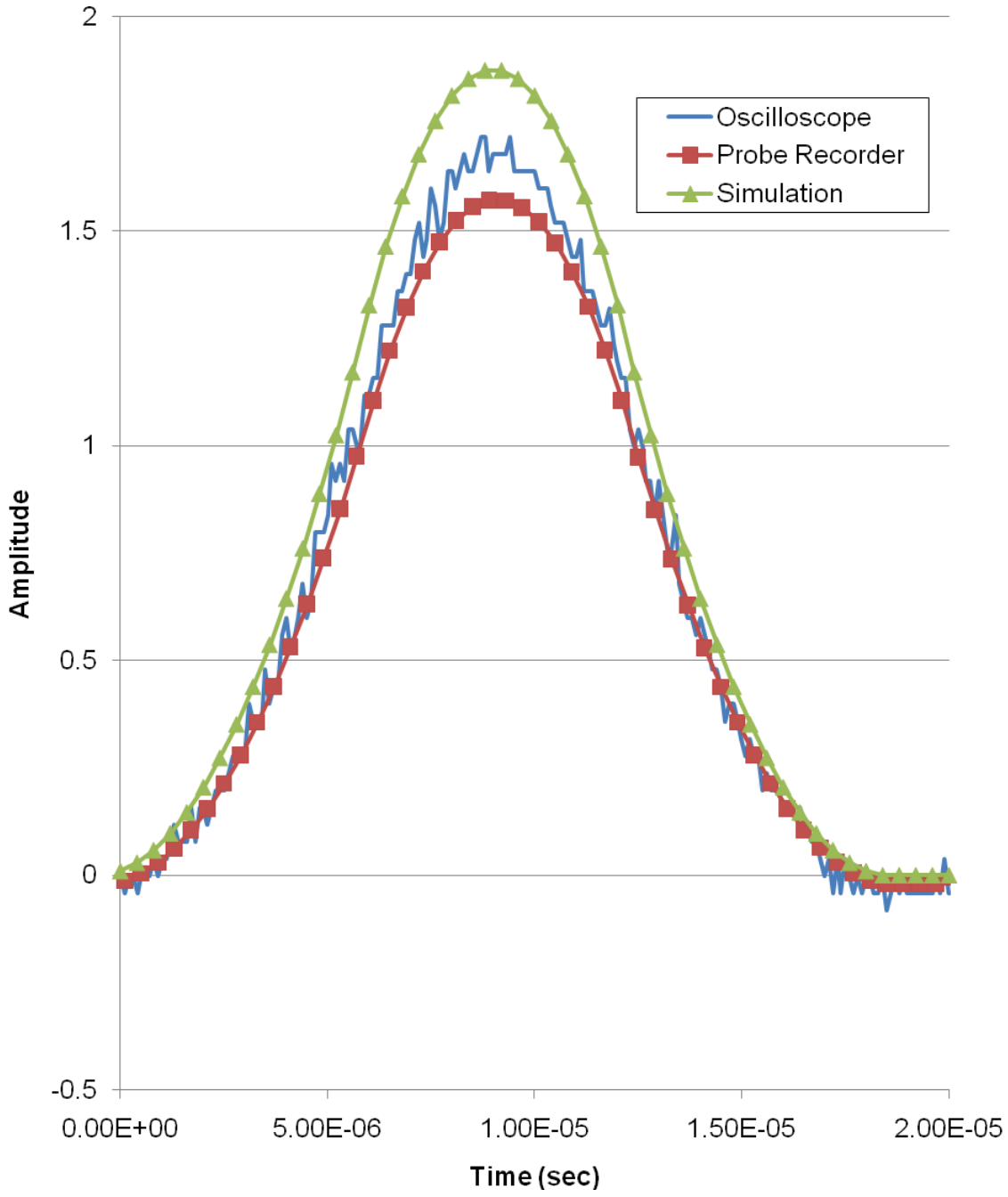

Figure 24: Impulse Response - 156.25 kS/sec, 16x interpolation

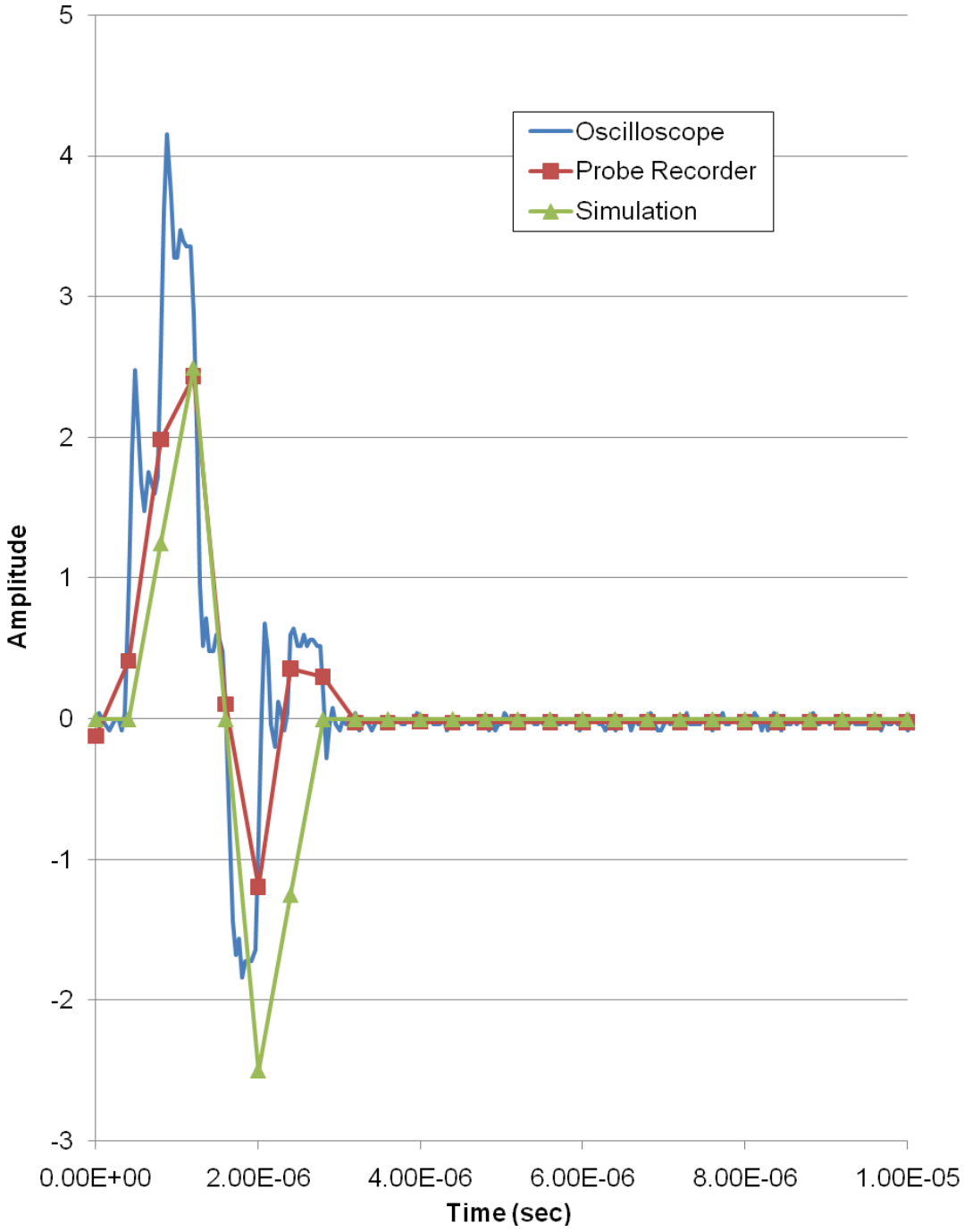

Figure 25: Impulse Response - 1.25 MS/sec, 2x interpolation (modified simulation)

Settings	Bytes recorded	Time (sec)	Calculated MB/sec
32 channels @ 2.5 MHz	284,656,820,224	1779	152.597
16 channels @ 5 MHz	339,074,088,960	2119	152.603
8 channels @ 10 MHz	327,669,702,656	2048	152.583
4 channels @ 20 MHz	293,246,828,544	1833	152.571

Table 2: Long duration throughput results

The recorded time domain and frequency data for each corresponding sample rate are shown in Figures 26 through 32. Table 3 contains the expected number of samples per sinusoidal cycle for each sample rate and the sample number plotted on the x-axis of each of the time domain charts confirms the correct number of samples per sinusoidal cycle. An FFT was performed on 50 acquisition blocks of these recorded signals to look for unwanted frequencies that would indicate dropped portions of the waveform. This grouping created a 1,638,400 point FFT, and the resolution of this FFT for each sample rate is also shown in Table 3. The frequency plots confirm that the calculated frequency falls within the range measured by the oscilloscope.

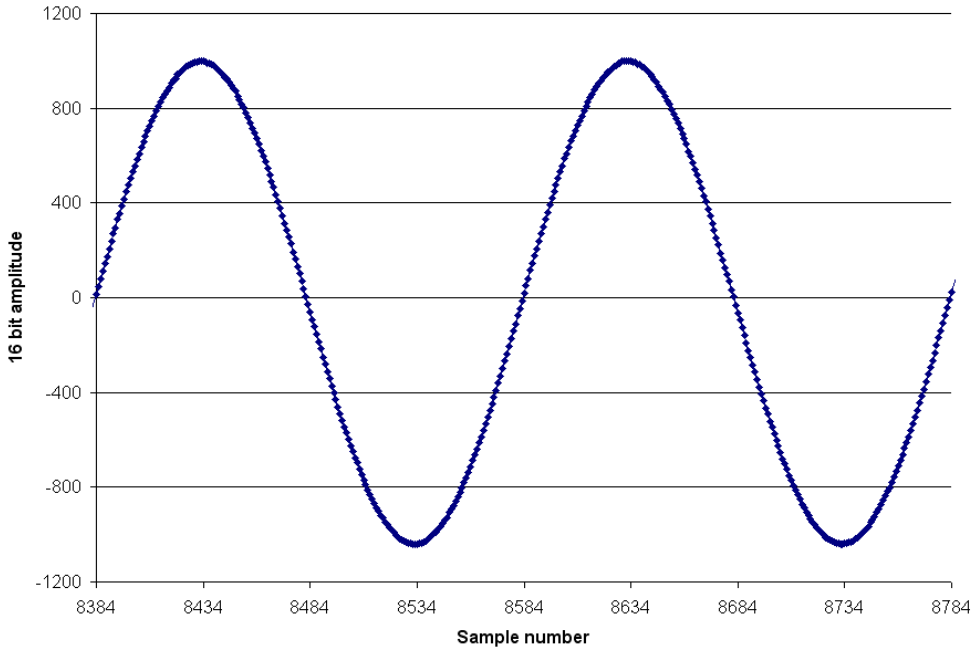

Figure 26: 100 kHz input, 20 MHz sample rate at maximum settings

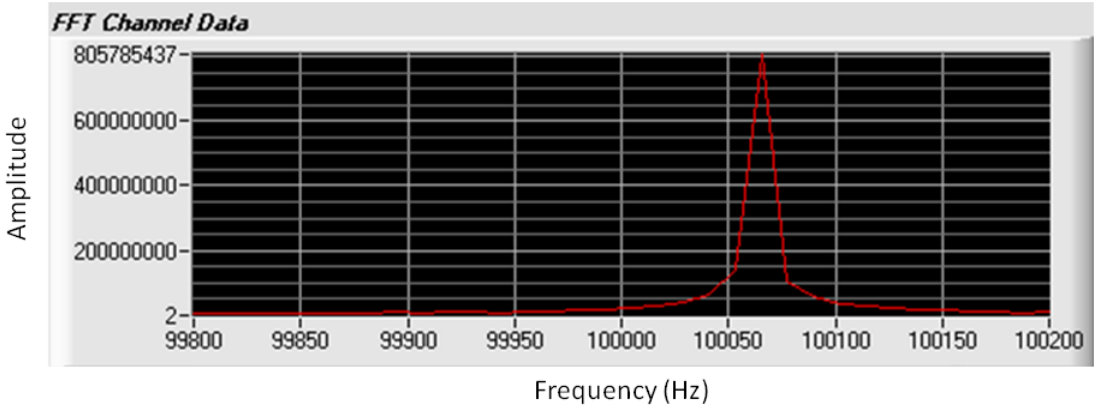

Figure 27: 100 kHz input, 20 MHz sample rate FFT at maximum settings

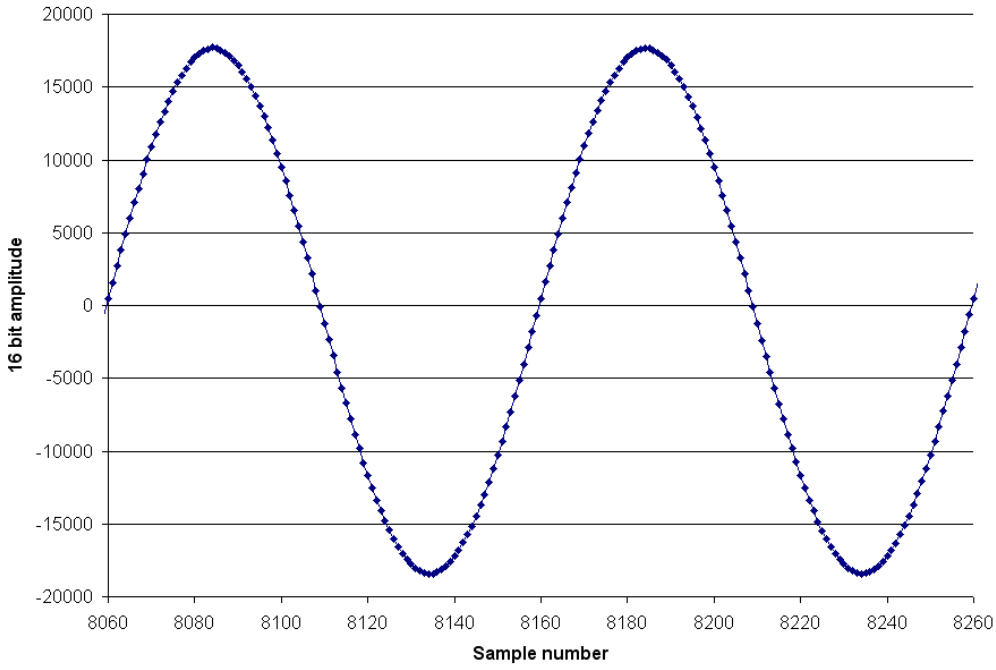

Figure 28: 100 kHz input, 10 MHz sample rate at maximum settings

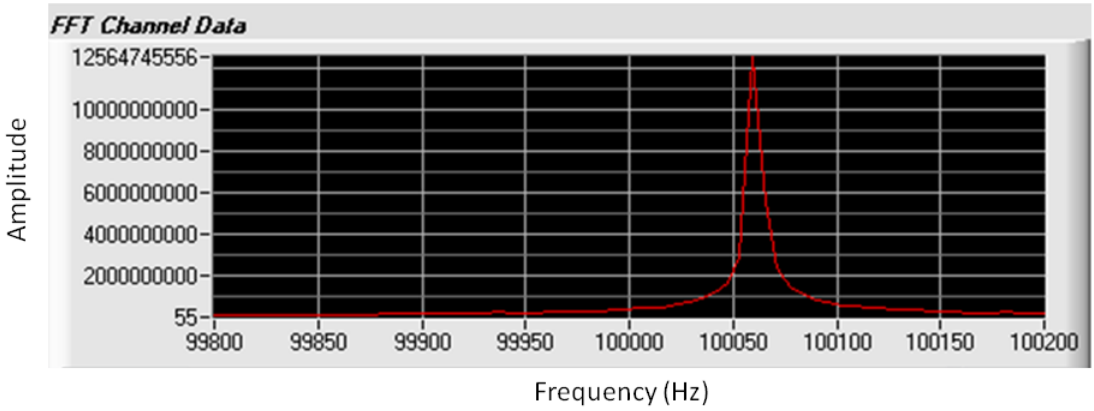

Figure 29: 100 kHz input, 10 MHz sample rate FFT at maximum settings

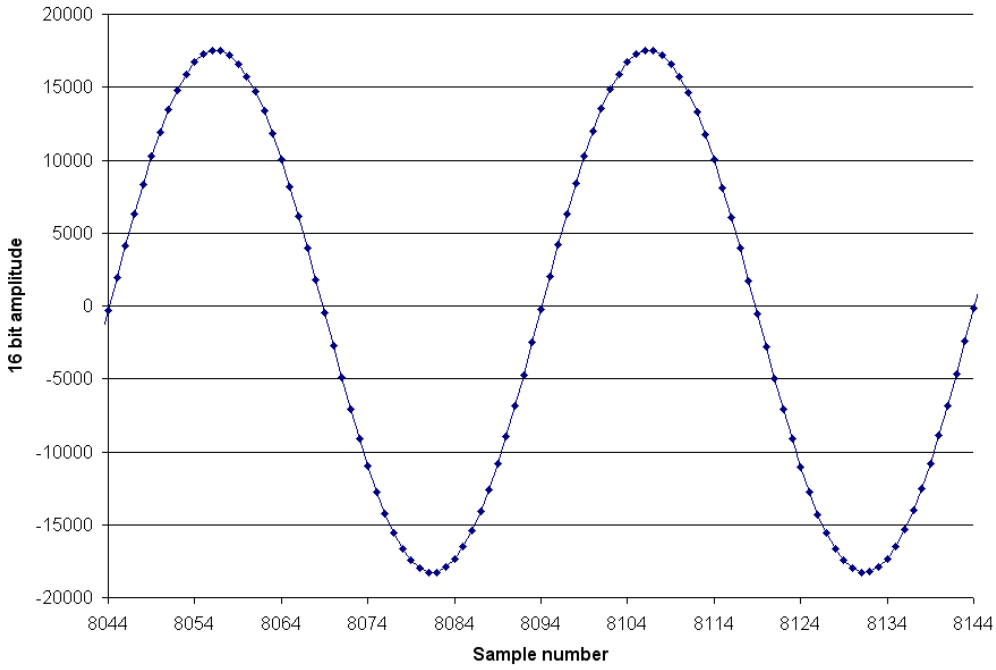

Figure 30: 100 kHz input, 5 MHz sample rate at maximum settings

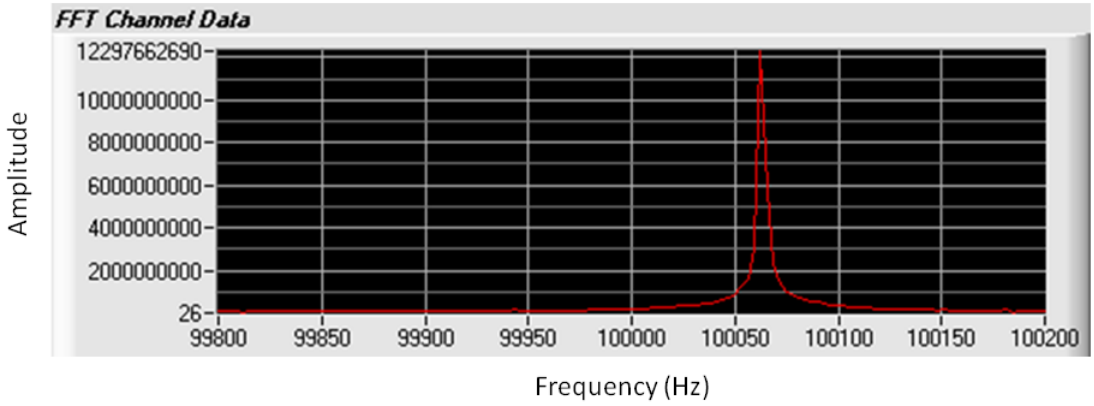

Figure 31: 100 kHz input, 5 MHz sample rate FFT at maximum settings

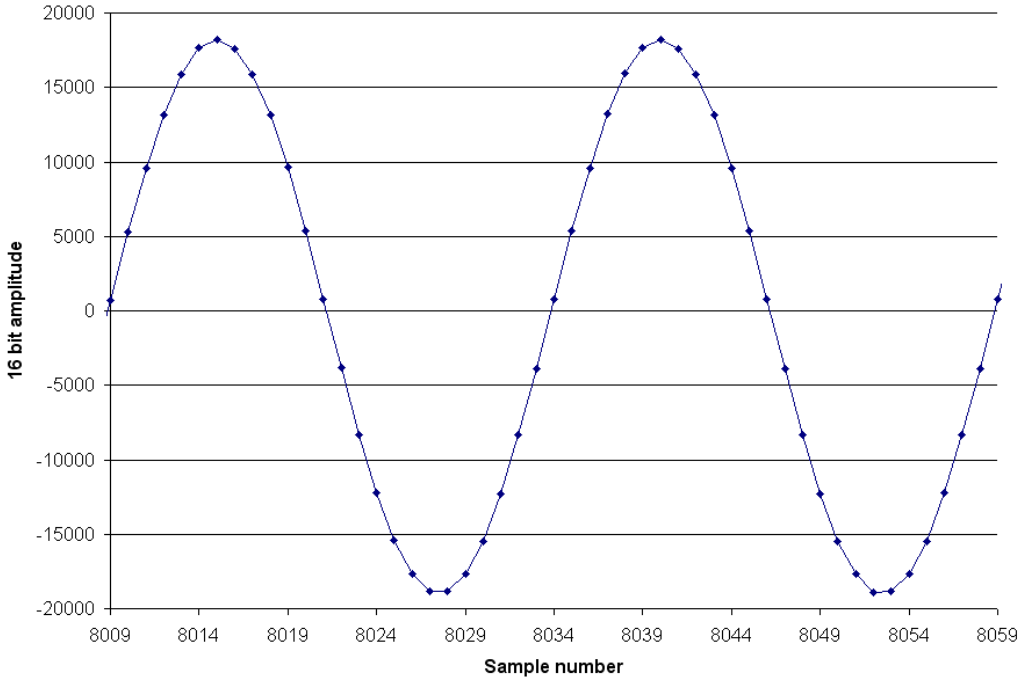

Figure 32: 100 kHz input, 2.5 MHz sample rate at maximum settings

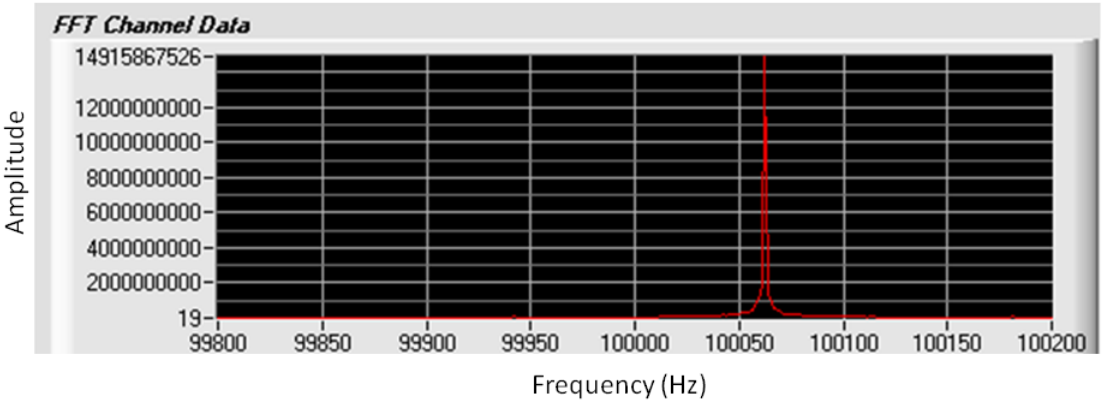

Figure 33: 100 kHz input, 2.5 MHz sample rate FFT at maximum settings

Sample rate (MHz)	Number of samples per 100 kHz sinusoidal period	50 acquisition block FFT resolution (Hz)
2.5	25	1.526
5	50	3.052
10	100	6.104
20	200	12.207

Table 3: Laboratory throughput specifications

6. Data Compression

For more efficient use of the system's disk space, data compression is utilized to reduce the amount of disk space each file will require. During development, the idea of a real time data compression technique was considered but not investigated due to the large capacity of the disk array and the desire to have all computer resources available for real time data transfers.

Because lossy compression can achieve much higher results than lossless compression techniques, an ideal data compression algorithm was evaluated that would theoretically retain all of the important information of the pulse and discard the trivial region between pulses. Since the typical signals recorded by the system are similar to the one shown in Figure 1, the lossy technique was customized for detection and compression efficiency based on a signal with significant space in between short, periodic pulses.

Taking a histogram of a typical signal data block, as shown in Figure 34, reveals the noise floor as the majority of the data values while the values for the pulses are much less frequent. To maximize compression, the custom lossy algorithm was constructed to replace the noise floor values with the mean value of the noise to achieve long repetitive sequences. The mean value of a data block was considered to be the average offset of the waveform since the contribution of the pulses is insignificant. The standard deviation of the waveform is considered to be the noise distribution for the same reason, and one standard deviation from the mean is used as the derived boundary for the noise floor. This modified array, with all detected noise regions replaced with the mean value, is then run length compressed [9] using the mean value as the indication of a compression and the length as the next immediate value. Since the compressed data can be of varying size, the termination flag of the maximum possible value followed by the minimum possible value is added at the end of the compressed data. A stepwise diagram of the custom lossy algorithm is shown in Figure 35.

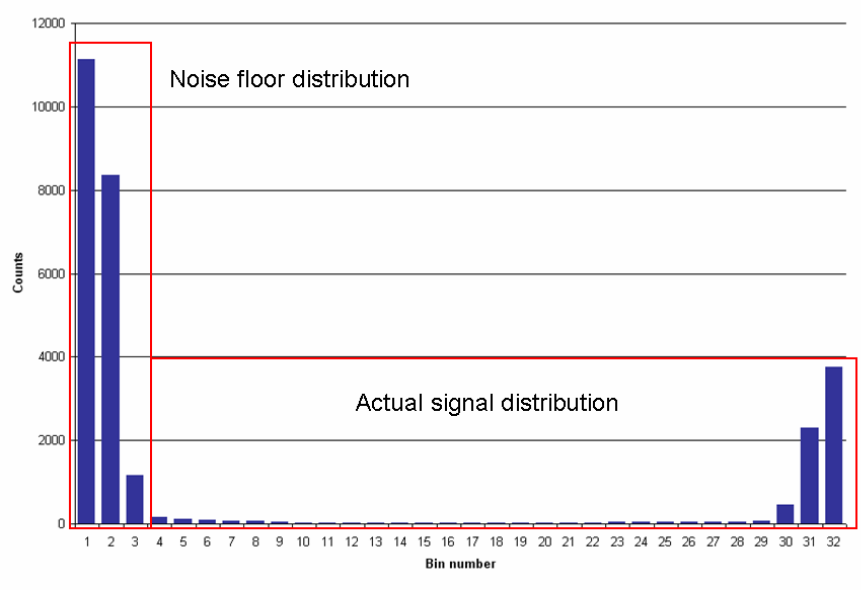

Figure 34: Histogram of typical probe signal

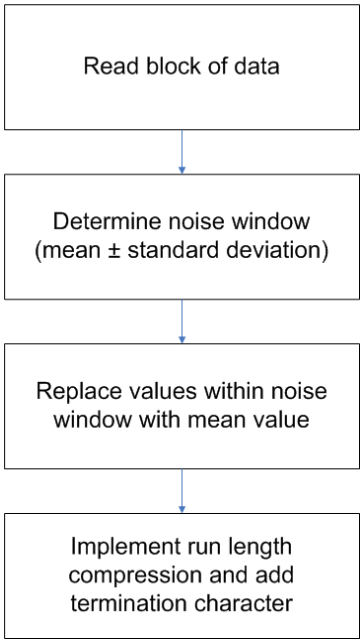

Figure 35: Block diagram of lossy compression algorithm

A test set of five recorded files with typical signals was used to validate the custom compression technique. For this test set, the lossy algorithm exhibited exceptional compression. The results of this test are shown in Table 4. Although this test set demonstrated the value of the method for these typical signals, other signal types produced different results. When using the algorithm to compress channels with no signal, the compressed data would not achieve the long run lengths needed for good compression ratios due to the noise frequently jumping outside the calculated window. This would result in files of similar size if not larger than the originals, which was still acceptable. For other signal types, such as any waveform with a symmetric pulse about the mean value or pure sine waves, the compression algorithm would treat an essential part of the signal as noise since it fell within the calculated range. These signals, when decompressed, would yield distorted results as shown in Figure 36.

Due to the unacceptable distortion of certain signals, lossy compression was tabled for future consideration and lossless compression was implemented to ensure signal integrity. The lossless compression was implemented using the WinZip program due to its availability. In WinZip, three different lossless techniques are provided. These methods are PPMd, a variation of prediction with partial match (ppm) methods, Bzip2, a conglomeration of several lossless algorithms applied in sequence, and the regular Zip algorithm based on standard dictionary techniques [9]. The methods were tested on a sample set of files to determine the best algorithm for archiving. The results of the lossless test set are shown in Table 5. While conducting the test to determine the best compression ratios, it was observed that the time required for each algorithm would also be a factor in choosing the best method. The results for each method applied to a standard file while monitoring execution time are shown in Figure 37. Considering both compression ratio and execution time, the Bzip2 algorithm was implemented.

File number	Original size (KB)	Compressed size (KB)	Compression ratio
1	41211	1612	25.57
2	36329	1546	23.50
3	980469	36967	26.52
4	508399	18674	27.22
5	440430	15777	27.92
		Average	**26.15**

Table 4: Lossy compression test set results

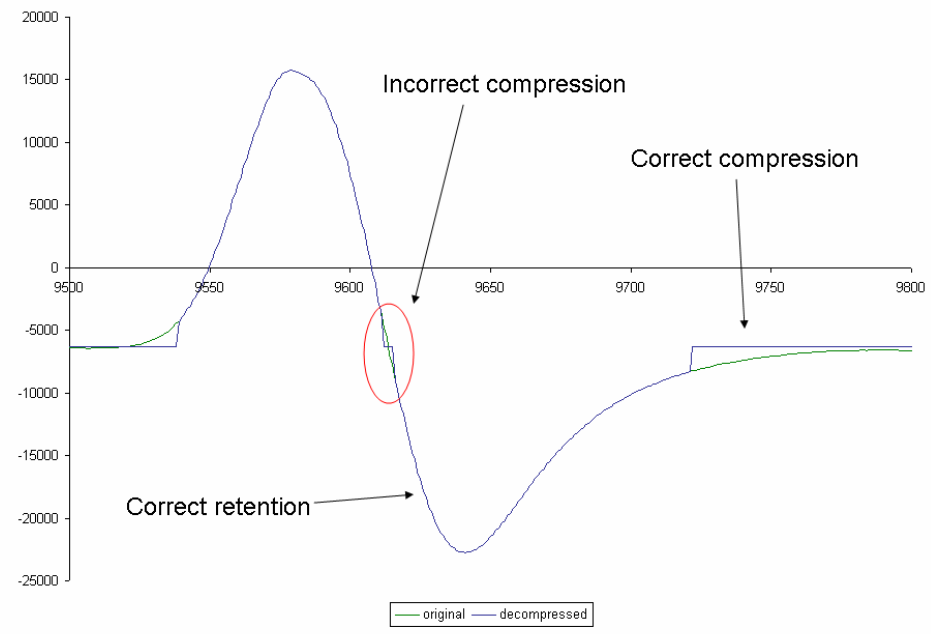

Figure 36: Original symmetric signal and lost information

File number	PPMd ratio	Bzip2 ratio	ZIP ratio
1	2.1015	1.9805	1.4564
2	2.4083	2.2594	1.5805
3	1.7360	1.6447	1.3266
4	1.7602	1.6720	1.3467
5	1.6942	1.5999	1.3016
6	1.7321	1.6396	1.3269
Average	**1.9054**	**1.7994**	**1.3898**

Table 5: Lossless compression test set results

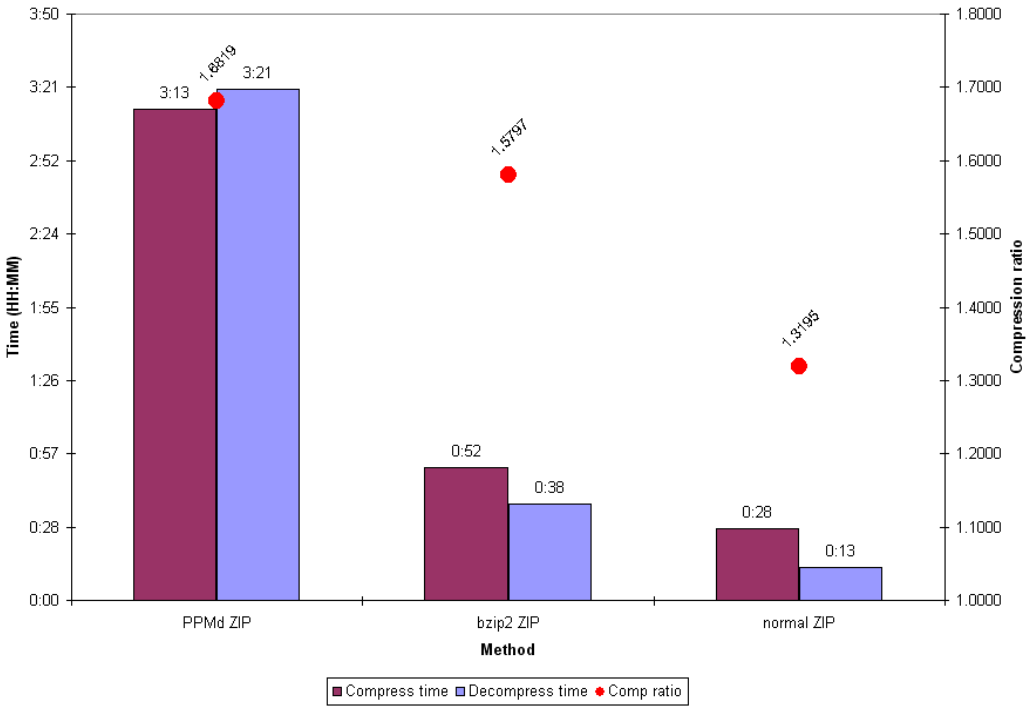

Figure 37: Lossless compression ratios and time required

7. Summary and Conclusions

After determining the desired sampling rates, precision, and channel counts for the probe recorder system, commercially available hardware was selected to digitize the data. This hardware was interfaced to a computer with internal memory, communication busses, and a RAID storage array specifically chosen for adequate data transfer speeds. Custom software was developed that ensured reliable and timely data acquisition, and another program was developed to view and analyze the acquired data. After the system was assembled and programmed, several tests were conducted to verify individual component and overall system performance.

Speed tests on the RAID array confirmed that the write speed was more than the required 160 MB / sec, usually around 350 to 400 MB / sec. According to the laboratory data, the read speed only met requirements when the larger block sizes, corresponding to eight channels or greater, were implemented. The read speed limitation may be addressed in the future, but since the playback board is limited to a 1 or 2 MHz output rate for 12 channels, the current speed limit is not a hindrance.

To characterize the system's overall transfer function, digital impulse data was introduced to the playback portion of the system, and the result was recorded on the digitizing portion of the system along with an oscilloscope and compared to a simulated response. Except for the overshoot and ringing observed in the case of an impulse interpolated at twice the sampling rate, no unexpected effects were observed in the recordings and the simulated response, the probe recorder data, and the oscilloscope data were well correlated.

System performance was also characterized with long duration tests at each of the maximum settings - 4 channels sampled at 20 MHz, 8 channels sampled at 10 MHz, 16 channels sampled at 5 MHz, and 32 channels sampled at 2.5 MHz. Time domain analysis of the waveforms recorded during the long duration tests confirmed the appropriate number of samples per sinusoidal cycle. Frequency domain analysis of the waveforms exhibited the correct frequency, 100 kHz, within the resolution of each FFT. After determining a more precise expected maximum throughput, 152.59 MB/sec, the

derived total throughput for each of the maximum settings was within 0.012 percent of the expected value. Since locating data glitches in the large files is a tedious task and can consume large amounts of time, an automated intelligent analysis tool to identify distortion or data interruption would greatly enable practical data verification.

Offline lossless data compression was implemented to reduce demands on archival storage space after attempting a lossy algorithm that could not retain all important signal characteristics. Since the method was exclusively based on the typical probe signal, it was not able to properly handle the other types of signals that can be recorded on the system. The current lossless method only provides a compression ratio varying between 1.5 and 2, and the implementation of a lossy data compression algorithm, either offline or real-time, should be revisited in the future due to the significant reduction in file size that can be achieved.

As with most development tasks, lessons were learned and ideas were formed during the design and verification process for the probe recorder. Decisions were made for reducing some requirements early in the design process from their ideal conditions to fit with currently available hardware. As shown in the data flow diagram, Figure 7, the bandwidth of the entire system is limited at the analog-to-digital conversion and the interface between the acquisition hardware and the computer memory. Upgrades in these areas will increase the throughput up to and beyond the ideal performance. Greater capacity for on-board memory buffers and larger DMA transfers would also help improve the capabilities of the system by reducing the processor time dedicated to monitoring the interrupt driven data transfers from hardware to computer memory. Although the RAID configuration exhibited exceptional writing speed, larger drives would increase the available space and enable longer time periods of continuous data acquisition.

As far as implementing the original vision, four systems have been built and each probe recorder has been field tested and proven effective at accurately recording raw signals and enabling the recovery of mistriggered or questionable data. In addition to the system characteristics already addressed, future development should also focus on expanding the analysis capabilities of the system. As explained in [1], a customizable

firmware design expanded the capabilities of a section of the NSMS system due to the added flexibility and size reduction from hardware to software. Having developed the tool to record the raw signals, software implementations of existing and unproven processing techniques can be developed without the need of expensive hardware configuration changes and tested with real data before operational use.

LIST OF REFERENCES

LIST OF REFERENCES

[1] C. R. Vining, B. W. Hayes, S. A. Arnold, L. L. Holt, B. Jones, and B. W. Bomar. “PC-Based Generation 4 Non-Contact Stress Measurement System”. High Cycle Fatigue Conference, New Orleans, LA. March 2005.

[2] John G. Proakis. *Digital Signal Processing: principles, algorithms, and applications.* Third Edition. New Jersey: Prentice Hall, 1996.

[3] *ICS-645 Operating Manual.* Nov. 2002. E10592 revision J. ICS Ltd.

[4] *AD9260 Datasheet.* Aug. 2004. Revision C. Analog Devices Inc.

[5] *ICS-500R Operating Manual.* Jan. 2005. E10651 revision C. ICS Ltd.

[6] *THS7001/THS7002Datasheet.* Aug.1999. SLOS214B. Texas Instruments Inc.

[7] *Double Data Rate (DDR) SDRAM specification.* March 2003. JESD79C. Jedec Solid State Technology Association.

[8] *DAC8580 Datasheet.* August 2005. SLAS458B. Texas Instruments Inc.

[9] Khalid Sayood. *Introduction to Data Compression.* Second Edition. Academic Press, 2000.

[10] H. W Johnson and M. Graham. *High Speed Digital Design: A Handbook of Black Magic.* Englewood Cliffs, New Jersey: Prentice Hall, 1993.

[11] C. Mittermayer and A. Steininger. “On the Determination of Dynamic Errors for Rise Time Measurement with an Oscilloscope”. *IEEE Transactions on Instrumentation and Measurement*, Vol. 48, pp. 1103-1107, Dec. 1999.

[12] Johnson M. Hart. *Windows System Programming.* Third Edition. Boston, Massachusetts: Pearson Education, 2005.

[13] C. R. Vining, S. A. Arnold, B. W. Hayes, and R. P. Howard. “Comparison of Eddy Current Probes to Generation 4 Line Probes.” High Cycle Fatigue Conference, New Orleans, LA. March 2005.

[14] Ray M. Chi and Henry T. Jones. “Demonstration Testing of a Noninterference Technique for Measuring Turbine Engine Rotor Blade Stresses.” AIAA 24th Joint Propulsion Conference, Boston, MA. July 1988.

[15] R. C. Anderson, W. Atkinson, T. Bonsett, and J. Osani. “The Propulsion Instrumentation Working Group – Introduction and Discussion of Subteam Activities.” AIAA 20th Advanced Measurement and Ground Testing Technology Conference, Albuquerque, NM. June 1998.

[16] Marc Farley. *Storage Networking Fundamentals: An Introduction to Storage Devices, Subsystems, Applications, Management, and Filing Systems.* Indianapolis, IN: Cisco Press, 2004.

[17] *RocketRAID 2320 SATAII Host Adapter User’s Guide.* Aug. 2005. Revision 1.0. Highpoint Technologies, Inc.

VITA

Daniel S. Crews was born in Lawrenceburg, TN on September 30, 1975. He was raised in Nashville, TN and attended David Lipscomb Elementary, Middle, and High School. Following graduation from DLHS in 1993, he attended Nashville State Community College then transferred to Tennessee Technological University and received a B.S. in electrical engineering in 2002. While working at Arnold Engineering Development Center, he attended graduate school at the University of Tennessee Space Institute and received a M.S. in electrical engineering in 2009.

Daniel is currently employed at Arnold Engineering Development Center as an electrical engineer concentrating on signal processing applications and techniques.