VLSI Architecture for Repetitive Waveform Measurement with Zero Overhead Averaging

Alexander N. Kalashnikov, Vladimir Ivchenko, Richard E. Challis
School of Electrical and Electronic Engineering, University of Nottingham, Nottingham, UK

Abstract **– The architecture and performance of a digital waveform acquisition instrument with built in averaging is discussed. The substantial time required for averaging in software was eliminated in the developed architecture by using hardware averaging at the speed of waveform occurrence, due to a fully pipelined operation of constituent units. This architecture was implemented using an FPGA development board, and was tested at a number of averages above 10,000 that should theoretically provide a reduction of the additive noise by more than 100 times. The effect of the noise reduction was clear from experiment. However, lower than predicted improvements were achieved when the level of the input noise before averaging was low. This occurred due to the quantization of the input signal, and should not be attributed to the averaging. Therefore, averaging of digitized data can impose particular limits on the achievable noise reduction but this only occurs when the noise level is very low and does not require much reduction *per se*.**

Keywords **– averaging, digital signal processing, VLSI for intelligent signal processing, waveform measurement**

I. INTRODUCTION

Averaging represents probably the most ubiquitous technique used in measurement processing. It is frequently applied to reduce the variance of measured values by the number of averages taken, provided that the measured variable remains unchanged. The additive noise is to be random with zero mean and symmetrical probability density function. Averaging can be applied to repetitive waveform measurement (ensemble averaging [1]) if two issues are addressed. First, the acquired waveforms should remain coherent [2]. Frame jitter reduces this coherence, and the best way of its reduction is to synchronize tightly the excitation and acquisition instruments architecturally [3]. Second, transmission of subsequent waveforms to a remote processing device might require substantial time due to communication network latencies [4]. This overhead could be reduced substantially if averaging is performed at the source of the measured waveform. As averaging of analogue signals is rarely used, the operation is performed at the analogue-to-digital conversion point. This adds intelligence to the ADC unit of a measurement system because its output is computed out of many (up to several hundred and more) repetitive waveforms.

The motivation for our research came from the necessity for reduction of uncertainties in ultrasonic measurements that can be achieved using averaging [5]. A typical waveform contains from several hundreds to several thousand samples affected by an additive noise. The excitation and acquisition parts of an ultrasonic instrument are located in a close proximity that allows the above-mentioned architectural reduction of the frame jitter to be achieved easily. The excitation frequency exceeds 1 kHz in most cases thus, 1,000 waveforms for averaging can be collected in less than a second. However, instruments like digital storage oscilloscopes and custom made ultrasonic devices available to us failed to achieve the high speeds of averaging required for observation of rapid processes. This is likely to be related to architectural limitations of the software implementation of computing averages. We aimed to develop a hardware based architecture capable of intelligent averaging of waveforms as they are acquired. A reconfigurable VLSI hardware was targeted for implementation that ensures portability and transferability of the design.

In this paper we report the development of a VLSI architecture for intelligent averaging of ultrasonic waveforms that was achieved on the fly without overheads for processing.

II. AVERAGING ARCHITECTURE ALTERNATIVES

Averaging in hardware can be accomplished by using a memory channel architecture shown in fig.1 [6]. It requires a separate hardware accumulator for every sample that restricts its usability severely. A more viable option is to use standard RAM and a single accumulating device.

Intelligent acquisition with averaging could be implemented using a general purpose or specialized processor as shown in fig.2. If the CPU can keep pace with the required sampling rate, it inputs data from the ADC directly and stores them in a local CPU memory (fig.2a). However in many cases

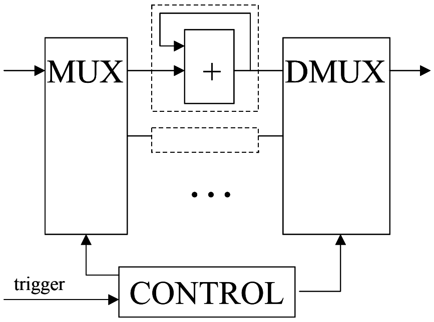
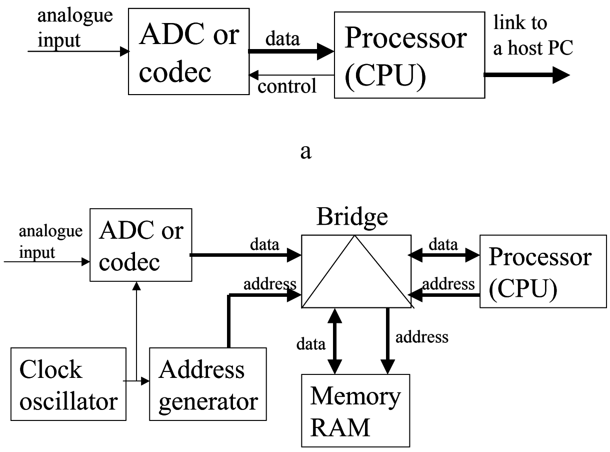


Fig.1. A hardware architecture for averaging [6]



b

Fig.2. Software architectures for intelligent acquisition/averaging

when the sampling frequency is of the order of 100 MHz and above the data need to be acquired into the RAM using dedicated hardware (address generator and bridge) that relieves the CPU from data acquisition but allows it subsequent access to the acquired data (fig.2b). In this case data can be acquired at the full speed of the ADC whilst the processor can still can be used for processing. This architecture employs three separate hardwired hardware elements (ADC, bridge and memory). Unfortunately, close ties between these elements cause significant difficulties for upgrades of existing systems. This architecture is implemented predominantly as a plug in data acquisition board (DAQ board) for a PC (with some customization as an option).

Let us analyze the suitability of this architecture for fast averaging. If the memory required is to be minimized, the ADC data are placed into a **BUFFER** area, and the running averaged record is placed into a **AVER** area. The following C pseudo code illustrates the computations involved:

```
// aver=0;
for (j=0;j<Naver;j++)
    { // acquire new frame into BUFFER;
       for (i=0;i<Nsamples;I++)
             AVER[i]+=BUFFER[i];
    }
// aver/=Naver;
```

where **Naver** is the number of averages required and **Nsamples** is the number of samples in a single waveform.

Processing of every sample requires three memory accesses, and because of this the procedure lasts several tens of seconds for some equipment available to us. This observed duration exceeds the time required for measurement by many times. Therefore, software architectures can be regarded as unsuitable for full speed averaging.

The design objectives (on the fly averaging, tight synchronization between the excitation and acquisition parts, upgradeable implementation) were met by using the architecture presented in fig.3. Using an adder in the data path

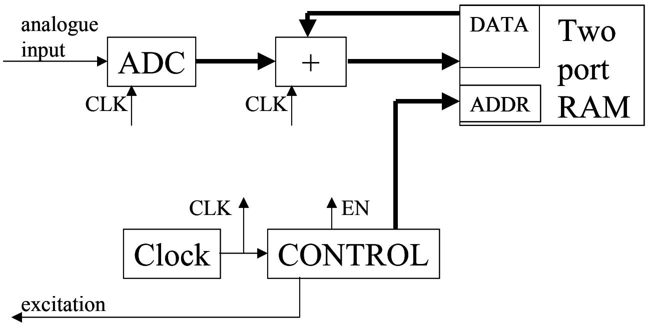


Fig.3. Zero overhead averaging hardware architecture

from the ADC to the memory enables full speed averaging. The adder is supplied with the running average at the same time from the RAM as a dual port RAM is used. An updated value is written back on its place, and there is no need for a separate buffer for a current record, this minimizes the demand for RAM. A host computer provides necessary parameters at the initialization stage, and this computer triggers the operation. When the averaging is completed, it is notified and extracts averaged data back using a dedicated interface. The control block not only generates address and enable signals, but also provides a trigger signal for generation of the excitation pulse to eliminate the frame jitter [3]. The averaged data are truncated to the number of bits of the ADC. Therefore, the averaged waveform can be seen by a host computer as a waveform acquired by the ADC itself.

III. Implementation of zero overhead averaging architecture using FPGA technology and high level design software

Architectural developments require adequate tools capable of fast prototyping. For high-performance data acquisition it means availability of hardware that includes a medium specification ADC(s)/DAC(s) and reconfigurable at will FPGA chip, because of the limitations of processors discussed above. The design software should support a high level of abstraction, design verification capabilities, and PC software interface. These requirements were met by the Xilinx Xtreme DSP development kit [7] that includes all the hardware necessary for our development. The kit includes System Generator software that is a plug-in facility for the Simulink toolbox of MATLAB. It provides a library of optimized components, for example, dual port RAMs; ensures simulation of the design both purely in software and co-simulation with hardware; and allows easy data exchange between the development board and MATLAB environment.

The block diagram of the averager captured from the System Generator is presented in fig.4. On board ADC1 supplies data to the arithmetic module that provides accumulation of samples, and additionally, clearing of DualPortRAM and streaming processed data to on board EzternalRAM. During averaging it reads a running total from the DualPortRAM located in the FPGA chip, adds it to the currently acquired sample, and writes the result back.

The on board ExternalRAM can hold results of many averaged acquisitions that ensures a high frame acquisition rate.

The MemoryControl unit increments memory addresses when triggered by the MainControl unit. It is used to control

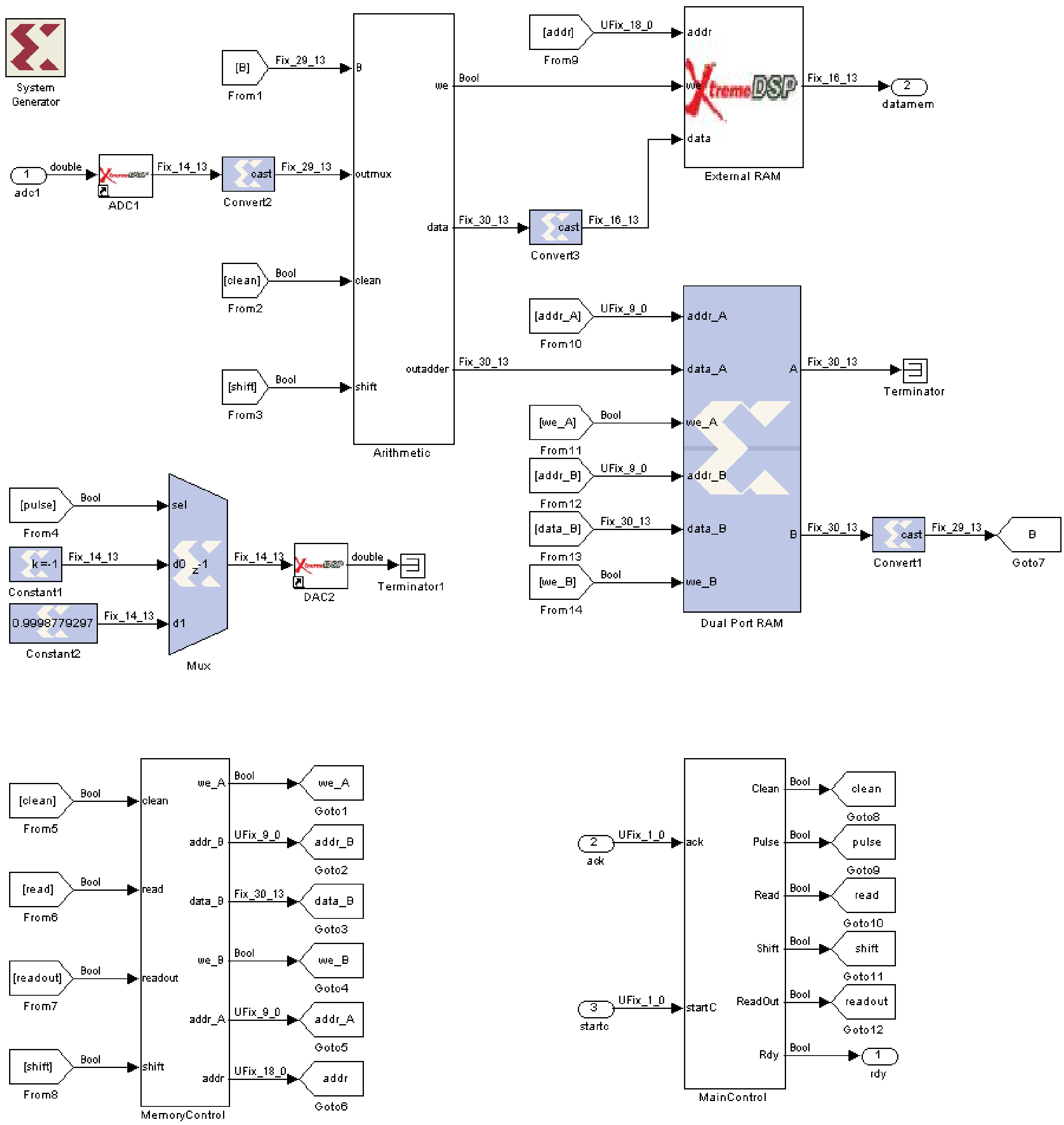


Fig.4. System Generator implementation of zero overhead averaging architecture

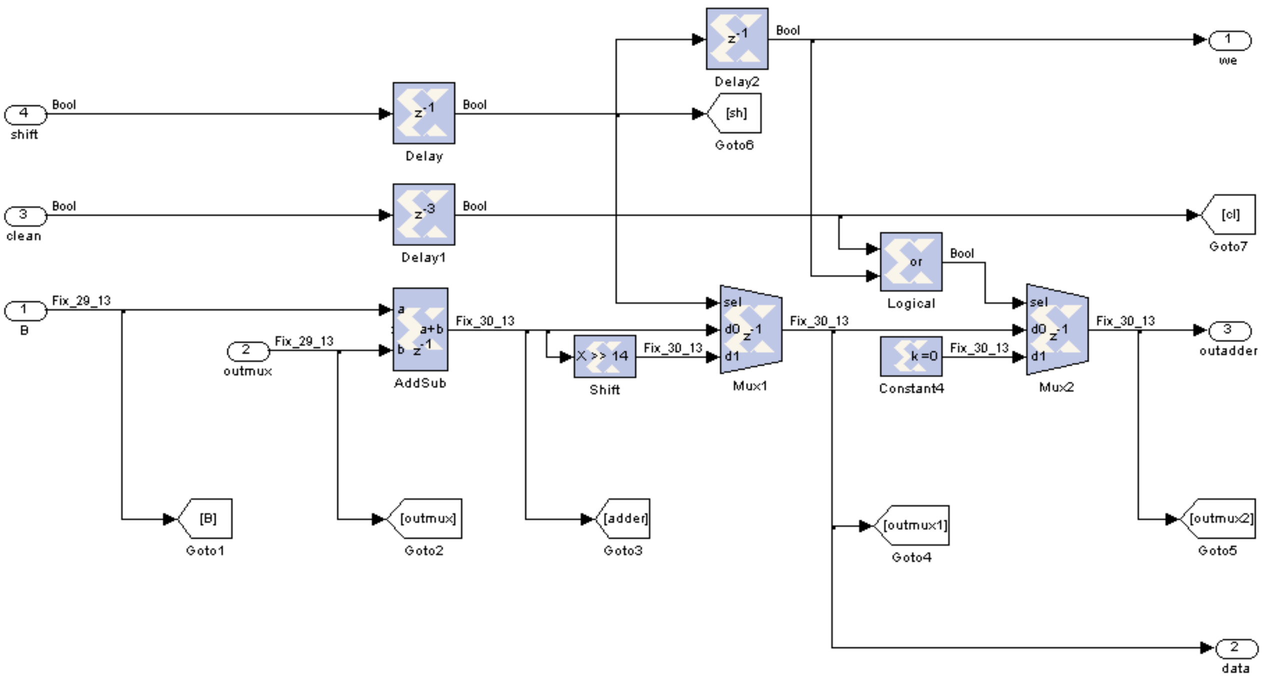


Fig.5. The blok diagram of the Arithmetic module

both the DualPortRAM and ExternalRAM for acquisition/averaging and (the later) for data uploading modes.

The MainControl unit triggers both excitation to minimize the frame jitter (through the DAC2) and acquisition/averaging according to softcoded parameters, and interacts with MATLAB.

The detailed structure of the arithmetic unit is presented in fig.5. It is implemented from the System Generator blocks solely. MUX1 and MUX2 were configured to ensure latency required for pipelined operation whilst the AddSub unit performs addition. The Shift unit converts the averaged waveform to the original resolution of the ADC. This imposes a restriction on the number of averages to be $N=2^M$, $M\in Z$ only. We found this restriction quite acceptable in practice, but it can be avoid by employing a dividing unit instead of the Shift unit. The width of the DualPortRAM equals to the width of the ADC plus log_2N to accommodate the running total in the worst case.

Compilation into the FPGA bitstream code is performed by clicking on the System Generator symbol (fig.4, left upper corner). The device can be used to provide up to 2^{16} averages at a sampling frequency of up to 90 MHz. The averager for 512 samples utilizes 694 Xilinx FPGA slices and 4 FPGA RAM blocks.

Further refinements of this design included generation of an arbitrary softcoded excitation signal and fast data exchange with MATLAB using the DMA mode of the PCI bus.

IV. EXPERIMENTS WITH VARIABLE NUMBER OF AVERAGES

The developed architecture was used successfully for acquisition of numerous ultrasonic waveforms (see examples of the experimental setup and typical waveforms in [2,3,5]) . The number of averages required was set depending on the total measurement time allowed (1024 typically). Sometimes (under good signal to noise input ratio) this number seemed excessive, and systematic measurements were performed in order to determine the number of averages both convenient and sufficient for particular signal and noise conditions.

Experimental waveforms were acquired with different numbers of averages that was set to 2^k (where $k\in N$, $2<k<10$). The number of samples in each waveform was 128, and 2048 hardware averaged waveforms were collected for every k. Initially the acquisition was performed at an interval where a strong signal that is normally of interest was present. However, the subsequent processing showed that the frame jitter (although significantly reduced in an architectural way) led to notable bias in the results. Therefore, more reliable results were obtained when the additive noise dominated the acquired waveforms. Some of the acquired samples are presented in fig.6 for two different numbers of averages, where q is the magnitude of the quantization step. The output signal of an ultrasonic transducer was amplified by 27 dB before digitization. The substantial reduction of the waveform variability achieved by using 1024 averages compared to only 2 is evident from the graphs.

The average of 2048 experimental waveforms was calculated and subtracted from the raw waveforms; the standard deviations of the residuals were then calculated :

$$\begin{aligned} aver &= \frac{1}{2048}\sum_{i=1}^{2048} waveform_i \\ residuals_i &= waveform_i - aver \qquad . \\ s\tan dard\_deviations_i &= std(residuals) \end{aligned} \qquad (1)$$

Two of the obtained sets of standard deviations are shown in fig.7. For these data the output signal of an ultrasonic transducer was amplified by 56 dB. Again, the graphs clearly highlight the advantages of averaging. The quantification of this advantage was made by calculating the ratio of the output to input STDs (standard deviations) of the intelligent averager. The theoretical value for this ratio would be

$$\frac{\sigma_{out}}{\sigma_{in}} = \frac{1}{\sqrt{N}} , \qquad (2)$$

where N is the number of averages, σ_{in} and σ_{out} are the standard deviation of the input and output signals of the averager respectively. The experimental data (fig.8) show that this relation holds for the case of up to 1024 averages for high input noise STDs. However it was observed that if the amplitude of the input noise is low, the amount of noise at the output tends to became higher than expected (fig.5, circles). The cause for this is discussed in the following section.

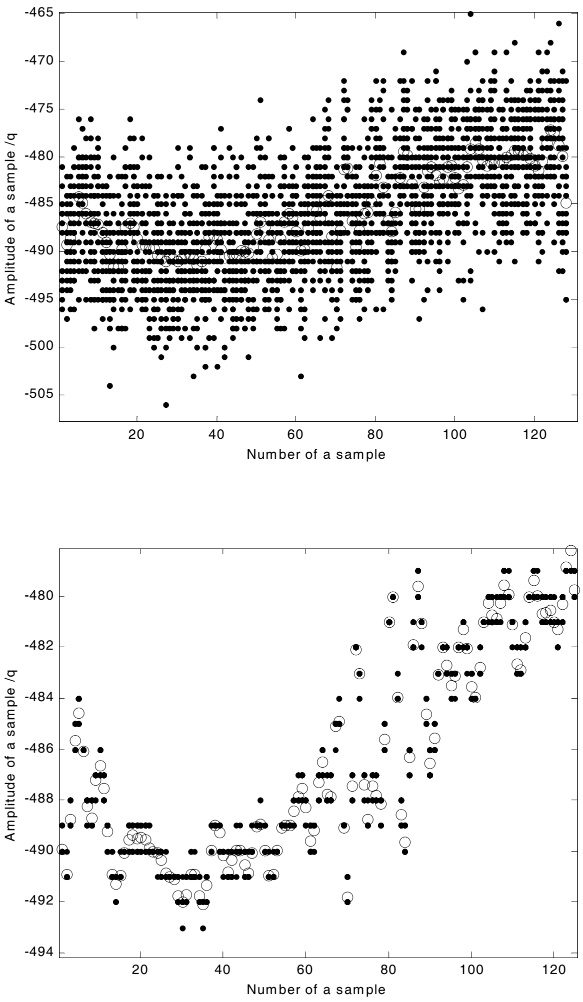


Fig.6. Examples of acquired samples after 2 (top) and 1024 (bottom) averages performed in hardware when the output signal of the ultrasonic transducer was amplified by 27 dB; circles represent average values

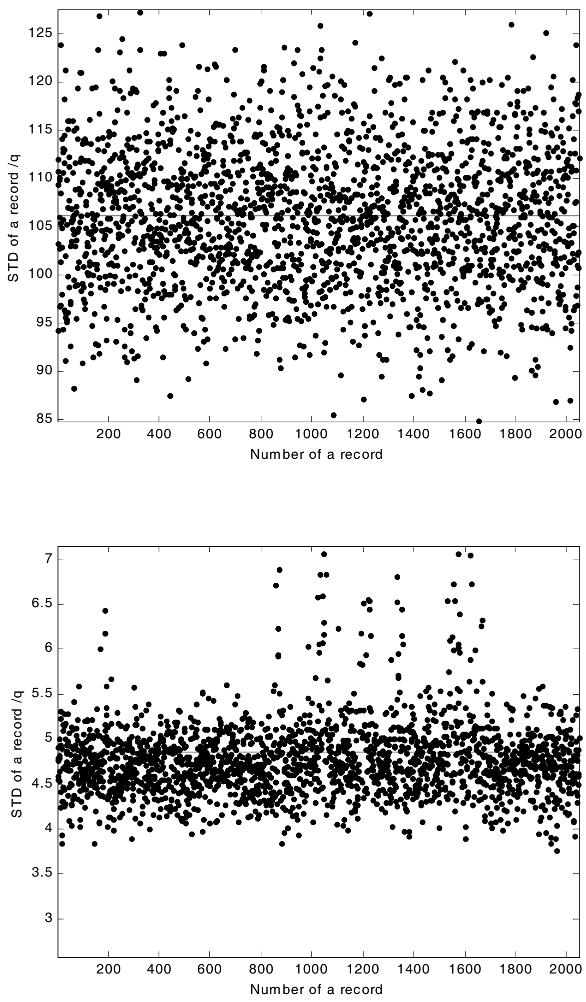


Fig.7. Examples of the calculated standard deviations for datasets obtained after 2 (top) and 1024 (bottom) averages performed in hardware when the output signal of the ultrasonic transducer was amplified by 56 dB

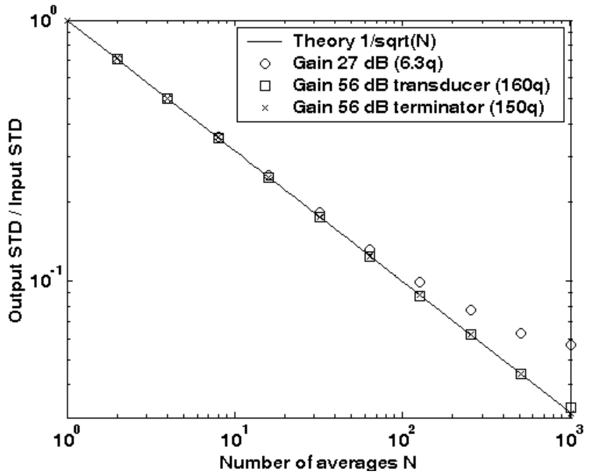


Fig.8. Experimental results averaged over the set of 2048 ensembles. The legend indicates the STD of the input noise related to the quantization step

V. Discussion

The underlying principle of ensemble averaging is based on the first principles of statistics. If the conditions listed in section I are met then equation (2) should undisputedly hold. However, a different case occurs when the input signal is quantified before averaging, because quantization introduces an error that can be treated as extra additive noise [8]. Computer simulations clearly show that the output STD of an averager is worse than predicted by equation (2) if the input noise STD is compatible with the quantization step (*e.g.*, [5,9]). This issue can be addressed theoretically from both the statistical theory of quantization [10] or the dither theory [11] points of view. Both approaches agree that if the input noise is "sufficient", the STD of the output noise would be in line with equation (2). If the amount of input noise decreased to levels compatible with the quantization step, part of the quantization noise will remain at the output regardless of the number of averages employed [1].

VI. Conclusions

Although averaging can consume a substantial time in conventional instruments due to software implementation, it can be implemented with zero overheads if appropriate hardware architecture is employed. This pipelined architecture includes ADC, adder and dual port RAM. It can be implemented using VLSI, *e.g.* an FPGA development board.

The implementation of this architecture using a commercially available FPGA development package proved the high efficiency of averaging achievable.

The experimental data showed the performance of the averager was in line with theory in the range of values from 2 to 1024. However, this only holds for pure noise. The presence of a strong signal introduces additional variability to the waveform samples due to the frame jitter. Therefore, the reduction of the additive noise by the averager should be assessed for pure noise input. The obtained improvement of the signal-to-noise ratio was lower than expected when the level of noise at the input was low.

The analysis of both analytical considerations and simulations found in literature showed this degradation in averager performance occurred due to quantization of the input signal, and not due to the operation principle and/or implementation of the device.

Overall, the described architecture ensured zero overhead averaging that substantially improved acquired waveforms. The performance of the averager was found in line with theory except in cases of relatively low levels of input noise. However, in these situations averaging might not be required at all or can be performed with a small number of averages that is beneficial for the total measurement time.

Acknowledgement

This research was fully funded by the EPSRC (UK). The donations of design software and training from Xilinx are highly appreciated.

References

[1] R.Skartlien and L.Øyehaug, "Quantization error and resolution in ensamble averaged data with noise", *IEEE Trans. Instrum. Measur.*, vol.54, No 3, Jun 2005, pp.1303-1312.

[2] A.N.Kalashnikov, R.E. Challis, M.E.Unwin and A.K.Holmes, "Quantification of the frame jitter in data acquisition systems", in *Proc. of IEEE Int. Symp. Intelligent Signal Process. (WISP'2003)*, Budapest, Hungary, 4-6 Sept 2003, pp.15-20.

[3] A.N.Kalashnikov, R.E.Challis, M.E.Unwin and A.K.Holmes, "Effects of frame jitter in data acquisition systems", accepted for a *special issue on intelligent signal processing of IEEE Trans. Instrum. Measur.*

[4] E.Truebenbach, "Communication buses: Bandwidth ≠ speed", *IEEE Instrum. Measur. Magazine*, vol.8, No 1, March 2005, p.32.

[5] A.N.Kalashnikov and R.E.Challis, "Errors in ultrasonic measurements", *accepted for IEEE Trans Ultrason, Ferroel, Freq Control in January 2005*.

[6] J.Jan, Digital signal filtering, analysis and restoration, IEE Press, London, 2000, Ch.6.

[7] Xilinx, Xtreme DSP Development Kit. http://www.xilinx.com/ipcenter/dsp/development_kit.htm

[8] W.R. Bennet,"Spectra of quantized signals", *Bell Syst. Tech. J.*, vol.27, July 1948, pp. 446-472.

[9] M.K. Brown," On quantization of noisy signals", *IEEE Trans. Signal Process.*, vol.SP-39, No 4, April 1991, pp.836-841.

[10] B. Widrow, I. Kollár, M.-C.Liu, "Statistical theory of quantization", *IEEE Trans. Instrum. Measur.*, vol. IM-45, No 2, Apr 1996, pp.353-361.

[11] R.Belchamber and G.Horlick, 'Use of added random noise to improve bit-resolution in digital signal averaging', *Talanta*, vol.28, 1981, pp.547-549.