

# Cross-correlation Based Ultrasonic Multi Channel Quality Control Using a Virtex 5 SX50T Board

Sammy Verslype, Erik Blomme, Tijl Cool, Renaat De Craemer, Franky Loret, Joan Peuteman and Jean-Jacques Vandebussche

**Abstract** – Using ultrasound, it is possible to test in-line the quality of materials. By mounting several air-coupled piezo-based transducers in parallel, entire surfaces can be scanned. The real-time processing of the sensor signals requires a digital processing of the data. Here, a fast FPGA based platform with parallel processing of the data is crucial in order to implement the required cross-correlations.

**Keywords** – Ultrasonic testing, quality control, FPGA, Virtex 5, cross-correlation

## I. INTRODUCTION

Using ultrasound (with frequencies up to 3 MHz), it is possible to test materials in a non-destructive way. By launching a wave into the material, useful information about the medium can be retrieved by analyzing the transmitted or reflected waves. The use of high-quality piezo-based transducers allows to test materials without any physical contact between the transducers and the material to be tested. This highly increases the number of potential air-coupled ultrasonic applications [1-3].

A broad range of materials like paper, wood, textiles, metals, plastics, rubbers and even nutrition can be tested. The ultrasonic testing principle does not harm the medium to be tested, the environment nor the operator.

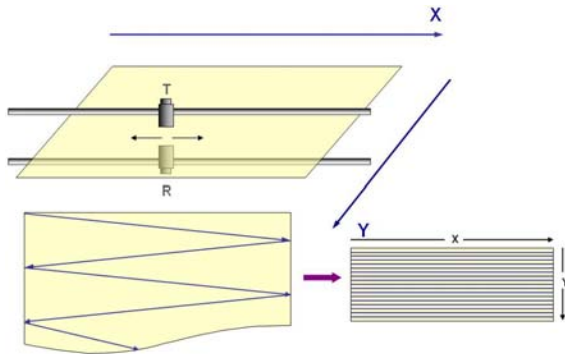


FIGURE 1. PRINCIPLE OF A ZIGZAG LINE SCAN

Test systems based on ultrasound already exist by using one transmitter and one receiver. Here, it is only possible to perform a line scan as visualized in Figure 1. When

S. Verslype, R. De Craemer, F. Loret, J. Peuteman and J-J. Vandebussche are with the KHBO – Katholieke Hogeschool Brugge Oostende, Departement IW&T, lab ECOREA, Zeedijk 101, Oostende B-8400, Belgium, e-mail: [sammy.verslype@khbo.be](mailto:sammy.verslype@khbo.be)

E. Blomme and T. Cool are with the KATHO – Katholieke Hogeschool Zuid-West-Vlaanderen, Dept.VHTI, lab NCU, Doorniksesteenweg 145, Kortrijk B-8500, Belgium, e-mail: [erik.blomme@katho.be](mailto:erik.blomme@katho.be)

considering surfaces, only sampling tests can be obtained during the production process. Indeed, during the movements of the pair of transducers (in the X direction), the surface moves (in the Y direction) implying the existence of a “dead zone” which will not be scanned.

In case the in-line quality control requires a full inspection of the whole surface, a large number of non-moving transducer pairs will be used in parallel. This approach is visualized in Figure 2.

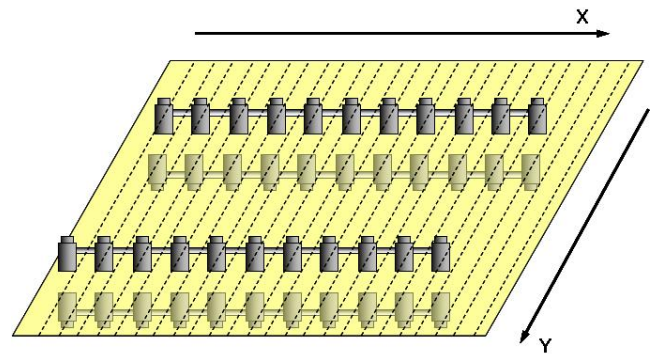


FIGURE 2. PRINCIPLE OF A FULL SCAN

The ultrasonic transmitters/receivers are grouped in modules of four transmitters/receivers. By placing these modules in parallel, it is possible to mount a larger number of transducer pairs.

The transmitting module contains an internal power supply, digital-to-analog converters (14 bit with a sampling rate of 10 Msps), reconstruction filters, power amplifiers and four piezo-based transmitters (Figure 3).

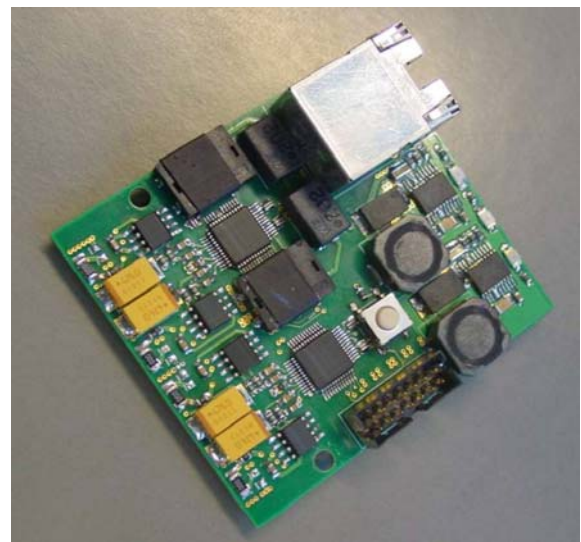


FIGURE 3. TRANSMITTING MODULE

In addition to four piezo-based sensors, the receiving module also contains low noise amplifiers, anti-aliasing filters, analog-to-digital converters (12 bit with a sampling rate of 10 Msp/s) and an internal power supply. Figure 4 shows such a four channel sensor module.

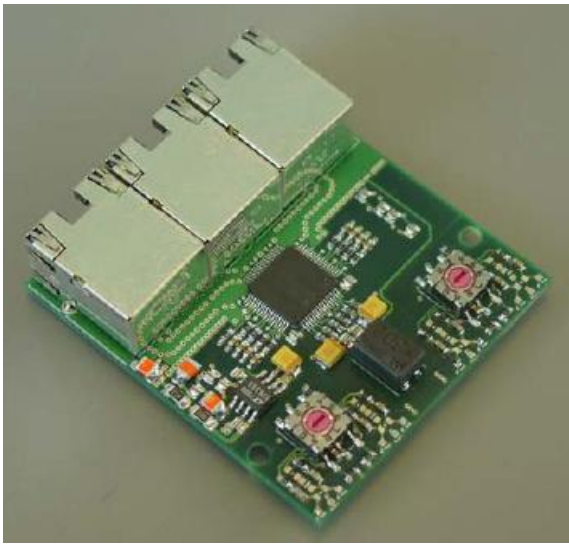


FIGURE 4. RECEIVING MODULE

By means of an LVDS data connection, the FPGA board transmits the desired ultrasonic waveform to the transmitter module. This provides the possibility to adapt the ultrasonic wave to the physical properties of the tested material. By means of four parallel LVDS data connections, the receiver module sends the pattern of the received ultrasonic waves to the FPGA board. This communication approach is visualized in Figure 5.

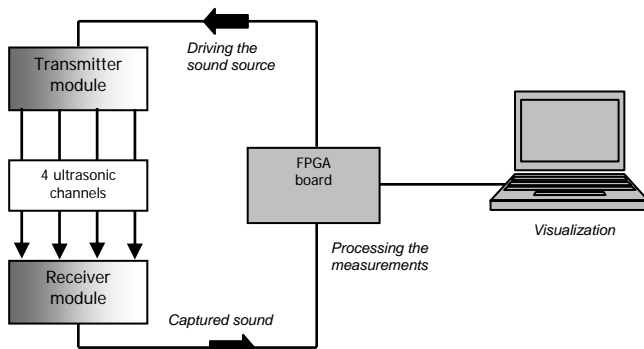


FIGURE 5. COMMUNICATION FPGA BOARD AND MODULES

The FPGA processes the data of several transmitter-receiver pairs and even the data of several modules in real-time. Processing such a large amount of data requires a lot of parallel processing which can be implemented in an FPGA such as the Xilinx® Virtex 5.

## II. TECHNICAL IMPLEMENTATION

### A. Hardware layout

The hardware of a single channel analyzer consists of dedicated functions which can be categorized into four major parts: signal acquisition, data recovery, data

processing and communication to a host computer. Figure 6 presents an overview of these four parts.

In order to create a multi channel analyzer, the single channel hardware will be implemented several times on the

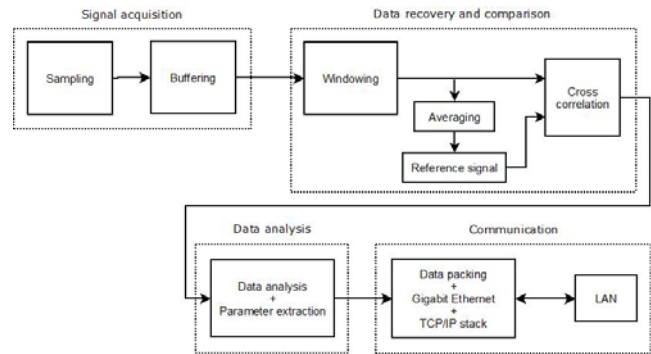


FIGURE 6. LAYOUT OF A SINGLE CHANNEL ANALYSER

same chip, with exception of the communication hardware. The hardware of the different channels runs completely parallel, at a high speed and real-time. Considering the above requirements, a state-of-the-art FPGA has been used as a single chip solution.

### B. FPGA selection

A modern FPGA can perform complex tasks at a very high speed. It can be equipped with dedicated hardware such as DSP-blocks, Ethernet MAC sublayers, RAM memory and microprocessors. The choice of an appropriate FPGA is application dependent. In the present application, a lot of digital signal and data processing is required and it is also important to realise a communication link over a LAN network. A high performance FPGA for signal processing applications is chosen, namely a Xilinx Virtex 5 SX50T.

A Virtex 5 SX50T contains DSP blocks, four Ethernet MAC layers and sufficient RAM memory. The FPGA does not provide an on-board microprocessor, but it is possible to implement a softcore processor such as the 32 bit RISC MicroBlaze.

### C. Calculation of the cross-correlation

The ultrasonic analyzer samples the incoming amplified sensor signal and compares it with a reference signal by means of a cross-correlation. Both signals contain 2048 samples with a 12 bit resolution. The reference signal is the average of several ultrasonic measurements obtained from an error-free reference sample. The cross-correlation allows to compare the quality of the material under test with a high-quality product.

Since the measurements on the material under test proceed at production speed, there is a continuous stream of measurements to the analyzer. The cross-correlation needs to be calculated at high speed in real-time in order to avoid any delay. Therefore, a correct and fast hardware implementation of the correlation is crucial.

The cross-correlation can be performed in the time-domain based on the expression of Eq. 1 ([4], pp. 186).

$$r(j) = \frac{1}{N} \sum_{n=0}^{N-1} x_r(n) x_m(n+j) \quad (1)$$

Here,  $x_r$  denotes the reference signal and  $x_m$  is the measurement performed on the material under test ( $r(j)$  is calculated for  $j = 0, \dots, N-1$  with  $N = 2048$ ). Based on the expression of Eq. 2, the cross-correlation can also be performed in the frequency domain ([4], pp. 207).

$$r = \frac{1}{N} F^{-1} \left[ F(x_r) F(x_m)^* \right] \quad (2)$$

Here,  $F$  denotes the Discrete Fourier Transform and  $F^{-1}$  denotes the Inverse Discrete Fourier Transform ( $N = 2048$ , the FFT can be used).  $*$  denotes the complex conjugate;  $F(x_r)$  and  $F(x_m)^*$  are pointwise multiplied providing the vector  $r$  with elements  $r(j)$  for  $j = 0, \dots, N-1$ .

Although using Eq. 1 is easy and straightforward, it requires a lot of computational effort. Indeed, the cross-correlation  $r(j)$  is calculated for each  $j$  requiring a total of  $N^2$  multiplications.

Calculating the cross-correlation using Eq. 2, which relies on a comparison of the frequency spectra  $F(x_r)$  and  $F(x_m)$ , requires significantly less computational effort. As Eq. 2 indicates, twice an FFT, one IFFT and only  $N$  multiplications are required. Performing an FFT or an IFFT requires  $(N/2)\log_2(N)$  multiplications ([4], pp. 76). Notice however, that Eq. 2 is more complex to implement and requires more hardware than the implementation of Eq. 1.

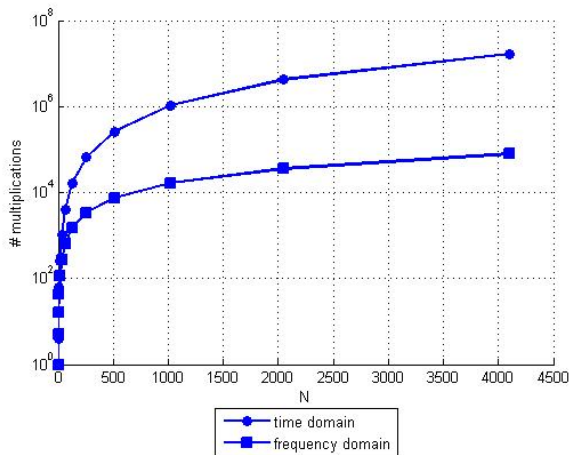


FIGURE 7. NUMBER OF MULTIPLICATIONS REQUIRED FOR CALCULATING THE CROSS-CORRELATION

Figure 7 shows the number of multiplications in case either Eq. 1 or 2 is used to calculate the cross-correlation. As already indicated, calculating the cross-correlation in the frequency domain requires less multiplications than in the time domain. For a data length of  $N = 2048$  samples, the number of multiplications is reduced by approximately a factor 100 which is very important in the present real-time application.

#### D. Design of the real-time cross-correlator

Performing the cross-correlation requires a number of operations executed in a sequential manner. Therefore, the

cross-correlation hardware is designed as a pipeline consisting of different hardware elements each performing an operation. The successive main operations are the buffering of incoming samples, the FFT of the buffered data, the complex multiplication of the frequency spectra of measured and reference data, the scaling of the product and finally the IFFT which gives the correlation signal  $r$ . The hardware elements are activated in the correct sequence by a controller. Figure 8 gives an overview of the real-time cross-correlator. Notice that the FFT of the reference signal is already available.

The hardware elements are designed and optimized for fast data processing, minimal use of FPGA resources and reusability. Indeed, a small cross-correlator block can be instantiated multiple times in the FPGA.

The receiving module of Figure 4 provides an uninterrupted stream of data to the FPGA. To buffer this stream of data, a circular buffer unit of 2048 samples is used. Once the buffer is full, the next sample will be stored at the lowest memory location and the cycle of buffering 2048 samples starts again.

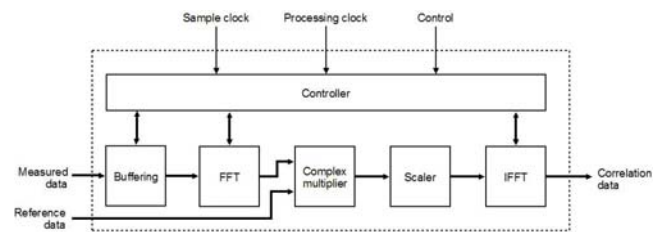


FIGURE 8. REAL-TIME CROSS-CORRELATION HARDWARE

The circular buffer is dual ported which allows to read and write samples simultaneously. A first data and address bus is used for the read operation, a second bus is used for the write operation.

The FFT-block and the IFFT-block are the most complex hardware elements of the cross-correlator. Xilinx provides a customizable intellectual property (IP) that can be used to implement these hardware blocks into the FPGA. Eq. 3 represents the FFT (actually a DFT) and Eq. 4 represents the IFFT (actually an IDFT) ([6], pp. 2).

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-jnk 2\pi / N} \quad (3)$$

with  $k = 0, \dots, N-1$

Similarities between these expressions imply that with some minor modifications, the same type of hardware can be used to calculate the FFT and the IFFT.

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) e^{jnk 2\pi / N} \quad (4)$$

with  $n = 0, \dots, N-1$

The FFT-block design can be based on many different hardware architectures. The fastest implementation is based on a pipelined architecture which allows to perform the transformation in real-time ([5], Table 6), e.g. a transform

of length 1024 requires 1024 clock cycles. Although this seems ideal for our real-time application, this architecture is also the most resource consuming ([5], Table 6) prohibiting the implementation of a large number of channels in one FPGA.

Alternatively, the Radix 2-Lite architecture is the most efficient regarding to FPGA resource consumption. Unfortunately, the Radix 2-Lite architecture is more time-consuming ([5], Table 12). For instance, a transform of length 1024 requires 11288 clock cycles. In order to maintain the desired real-time behaviour, it is necessary to use a clock frequency in the FFT-block which is much higher than the sampling rate.

A modern FPGA like the Virtex 5 SX can operate at clock frequencies up to 550 MHz [5] which is 55 times the sampling rate. In our application, a clock frequency of 300 MHz is used in combination with a read frequency of 300 Msps for the circular buffer. The write frequency of the circular buffer remains 10 Msps. This makes it possible to read the samples from the buffer and calculate the FFT in a time interval of 204.8  $\mu$ s.

In order to obtain a further reduction of the FPGA resources needed for implementing the FFT-block, the embedded DSP hardware blocks are used. These blocks provide several functions like multiplications and accumulations without using general FPGA resources.

As seen in Eq. 2, the complex spectrum  $F(x_r)$  is pointwise multiplied with the complex conjugate of the spectrum  $F(x_m)$  implying the need of a complex multiplier. The product of a complex number  $a + j.b$  and the complex conjugate of  $c + j.d$  (with  $j = \sqrt{-1}$ ) is given by

$$(a.c + b.d) + j(b.c - a.d) \quad (5)$$

Hence the practical implementation of the complex multiplication of the spectra can be realised using basic multiply-and-accumulate operations. As already mentioned, the DSP hardware blocks have specific functions to implement the procedure. The complex multiplier uses a fixed-point data representation, hence the product can have more bits than the operands. Therefore, the product is scaled providing a format which is suitable for the IFFT calculation.

As mentioned before, the FFT and the IFFT are implemented as instances of the same hardware IP. The output of the IFFT provides the cross-correlation information. The controller unit of Figure 8 controlling each hardware element in the pipeline is designed as a finite state machine.

#### E. Determination of the reference signal

The air-coupled NDT technique is based on the cross-correlation between the measured signal  $x_m$  and the reference signal  $x_r$ . For that reason it is very important to have a reliable reference signal  $x_r$ . In a calibration stage, the output of the sensors is measured for a high-quality sample with known properties. By considering  $M = 2048$  measurements each containing  $N = 2048$  samples of  $b = 12$  bit and taking the average, the reference signal is obtained.

Avoiding rounding errors, when adding  $M$  sample values of  $b$  bit, requires a buffer of  $B$  bit elements given by

$$B = \log_2 \left( (2^b - 1) \cdot M \right) \quad (6)$$

In order to restrict the memory required in the FPGA to realize the averaging buffer, the same averaging buffer is used for all the channels. An economic use of the hardware in the FPGA allows to increase the number of channels which can be implemented in the FPGA.

#### F. Further data analysis and communication

The IFFT in Eq. 2 results in the cross-correlation data. This data is analyzed in a next stage, as visualized in Figure 6, where parameters are extracted from the correlation signal. Parameters (like maximum, place of the maximum, integrated response) are gathered for every channel by a MicroBlaze processor. This processor also communicates the resulting parameters over a Gigabit Ethernet link to a host computer. There, the resulting parameters can be interpreted and/or visualized depending on the application.

### III. CONCLUSION

Using an FPGA, parallel data processing at high speed is possible on one single chip. This allows to realize a real-time multi channel quality control system using ultrasound. A modern FPGA (like the Virtex 5 SX50T) performs the major part of the required data processing which eliminates the use of very powerful and expensive computer systems.

#### ACKNOWLEDGEMENT

The authors wish to thank J. Deveugele and F. Declercq from KATHO (Kortrijk, Belgium) for the valuable discussions about the subject.

#### REFERENCES

- [1] E. Blomme, D. Bulcaen, F. Declercq. *Air-coupled ultrasonic NDE: experiments in the frequency range 750 kHz – 2 MHz*, NDT&E International, 2002, Vol 35, pp 417-426.
- [2] E. Blomme, D. Bulcaen, F. Declercq, P. Lust. *Air-coupled ultrasonic detection of errors in textile products*, in "Emerging Technologies in NDT", Eds. V. Hemelrijck, Anastasopoulos & Melanitis, Publ. Balkema, Rotterdam, 2003, pp 95-100.
- [3] E. Blomme, D. Bulcaen, T. Cool, F. Declercq and P. Lust. *Air-coupled ultrasonic assessment of wood veneer*, Electronic Proc. Int. Congress on Ultrasonics (ICU), Santiago, 12-16 Jan. 2009, <http://fisica.usach.cl/~ic2009/>.
- [4] E.C. Ifeachor, B.W. Jervis. *Digital Signal Processing: A Practical Approach*, Addison-Wesley, New York, 1998.
- [5] Xilinx, *Virtex-5 Family Overview LX, LXT, and SXT Platforms: Advance Product Specification*, DS100 (v3.2), September 4, 2007.
- [6] Xilinx, *Fast Fourier Transform v4.1*, DS260, April 2, 2007.