A Comparison of Computing Architectures for Ultrasonic Signal Processing

Álvaro Hernández, Jesús Ureña, Manuel Mazo, Ana Jiménez, Juan J. García, Carlos Marziani, Alberto Ochoa, José M. Villadangos and José A. Jiménez
Electronics Department, University of Alcalá. E-28806 Alcal de Henares (Madrid), Spain. E-mail: alvaro@depeca.uah.es
Fernando J. Álvarez
Department of Electronics and Electromechanical Engineering, University of Extremadura. Cáceres, Spain.

***Abstract* – The design of advanced sonar modules implies the use of more and more complex processing algorithms, in order to obtain as much information as possible from the environment. The use of simultaneous emission and reception techniques in certain transducers, that form an ultrasonic sensorial module, allows to increase the benefits and the performance of this type of systems. These developments make possible to obtain more information for every ultrasonic emission, diminishing the scanning time and improving the interpretation and usefulness of the obtained results. These techniques are usually based on the encoding of the ultrasonic emission, so the received signals can be correlated in order to search for possible echoes. As a consequence, the computational complexity necessary to carry out the implementation of the new algorithms increases considerably. The analysis of possible computing platforms for the real-time processing of the new algorithms becomes a task of great importance. In this work, some computing architectures are analysed and compared, in order to determine which scheme is most suitable for the ultrasonic signal processing.**

***Keywords* – Computing architectures, DSP, FPGA, Ultrasonic signal processing.**

I. Introduction

Classical processing techniques used in ultrasonic transducers are based on the measurement of times-of-flight (TOF). There, only one isolated transducer emits a signal consisting of a train of pulses; after the emission, the arrival of the possible echoes is detected. This detection is usually carried out by integrating and thresholding the received signal. Nevertheless, there exist some problems related to these methods, which are described thoroughly in [1] [2] [3] [4]:

This work was supported by the Spanish Ministerio de Educación Ciencia (PARMEI project, ref. DPI2003-08715-C02-01), and by the University of Alcalá (ISUAP project, ref. PI2004/033).

- The measurements are carried out in an independent way for every transducer. These cannot be excited simultaneously to avoid crosstalk problems. This constraint implies long acquisition times until enough information from the environment is collected.
- After the emission of a pulse, the signal is only received by a transducer (usually the same one that carried out the emission). So, the probability of non-detecting certain obstacles is increased due to possible specular reflections.
- The measured distances have a low precision. Although this can be enough for certain applications, it is not suitable when the TOFs from different transducers in more complex systems should be processed. In these cases, by having available a suitable precision, it is possible to determine reception angles, in order to carry out reflector-type classifications (for example, among planes, corners, or edges), or to build maps of the environment, etc.

Regarding these problems from any sensorial system based on ultrasonic emissions, the use and development of assemblies, that allow the use of simultaneous emission and reception techniques (multimode), provides a logical solution to problems about scanning rate. These techniques, based on correlation processes for the search of certain sequences in the received signals, allow to collect more information for every ultrasonic emission, but they require more computing power, in order to achieve real-time signal processing.

Apart from correlation techniques, the ultrasonic signal can be processed in other ways. Recently, some developments based on frequency analysis are coming more and more important, since they can cope with more complex environments in a better way [5] [6] [7]. Nevertheless, these algorithms often present more computational load, so they are not considered for real-time embedded systems.

0-7803-9030-X/05/$20.00 ©2005 IEEE.

In this work a study is carried out about the feasibility, presented by certain architectures to implement the algorithms for the determination of ultrasonic times-of-flight. These algorithms are based on the emission of Golay complementary sequence pairs. Section II describes the proposed ultrasonic processing. The computing architectures used in the comparison are detailed in Section III. Some results are explained in Section IV, and, finally, some conclusions are discussed in Section V.

II. Ultrasonic signal processing

The use of Golay complementary pairs for the detection of times-of-flight in ultrasonic systems has been already described in different works [8] [9]. These pairs are composed by two sequences, $A[n]$ and $B[n]$, whose independent auto-correlation function, and their later addition, provide an output signal according to (1):

$$C_{AA}[n] + C_{BB}[n] = \begin{cases} 2N & n = 0 \\ 0 & n \neq 0 \end{cases} \tag{1}$$

Where $A[n]$ and $B[n]$ are the pair of Golay complementary sequences; and N is the number of bits, or their length. Figure 1 shows the resulting signal after the addition of the two partial auto-correlation functions for a 32-bit Golay complementary pair. In the same way, two pairs of Golay complementary sequences, see $[A_1, B_1]$ and $[A_2, B_2]$, are orthogonal if the expression (2) is met.

$$C_{A_1A_2}[n] + C_{B_1B_2}[n] = 0 \qquad \forall n \tag{2}$$

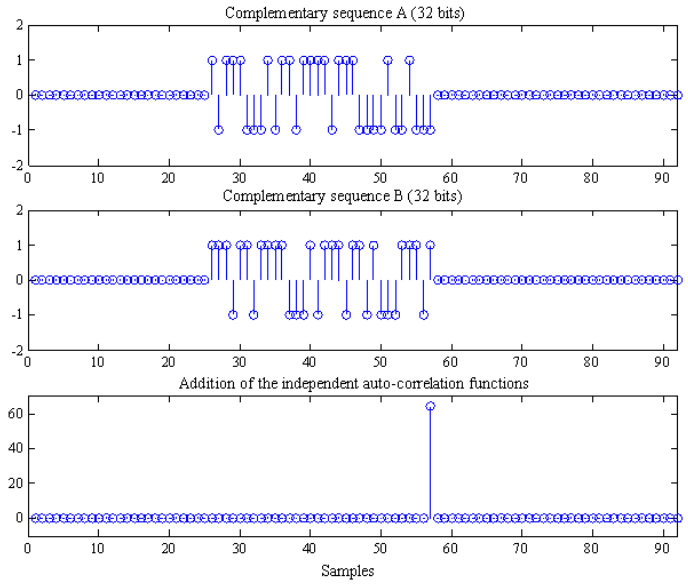

Fig. 1. Addition of the two auto-correlation functions for a 32-bit Golay complementary pair.

By using a Polaroid ultrasonic transducers whose maximum frequency response is centered at $50kHz$ [10], it is necessary to carry out a modulation of the Golay pair with a carrier of the same frequency. A digital variant of a QPSK modulation has been used, where the sequences $A_i[n]$ and $B_i[n]$, both from the same pair of a transducer i, have been associated to the component I and Q of the modulation, respectively. The mathematical expression (3) depicts this modulation process, in order to obtain the signal $e_i[n]$ to be emitted by transducer i.

$$\begin{aligned} e_i[n] &= A_i[n] * S_I[n] + B_i[n] * S_Q[n] = \\ &= \sum_{k=0}^{N \cdot M \cdot m - 1} A_i \left[\frac{k}{M \cdot m} \right] S[n-k] + \\ &+ \sum_{k=0}^{N \cdot M \cdot m - 1} B_i \left[\frac{k}{M \cdot m} \right] S \left[\left(n - \frac{M}{4} \right) - k \right] \end{aligned} \tag{3}$$

Where M is the number of samples per period of the symbol $S[n]$ (related to the sampling frequency f_S of the received signal); m is the number of periods per symbol; and N is the number of bits or the sequence length. The signals $A_i[n]$ and $B_i[n]$ constitute the Golay pair assigned to the transducer i: the first one assigned to the in-phase component I of the modulation, whereas the second is related to the in-quadrature component Q. On the other hand, the signals $S_I[n]$ and $S_Q[n]$ are the carriers of the components I and Q, obtained from the symbol $S[n]$ with the corresponding phase. As has been already commented, the symbol allows to center the emission at a frequency of $50kHz$; in an experimental way, a sampling frequency f_S of $400kHz$ has been chosen, what implies an oversampling $M = 8$. The parameter m has been fixed at $m = 2$. Figure 2 shows the emitted signal $e_i[n]$ obtained after the QPSK modulation, with the mentioned parameters -$N = 32$, $m = 2$ and $M = 8$-.

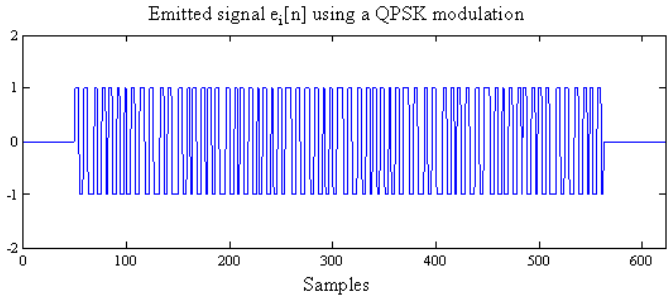

Fig. 2. Signal $e_i[n]$ emitted by transducer i after QPSK modulation.

In the reception process the demodulation of the received signal is carried out in order to extract the components $I_i[n]$ and $Q_i[n]$ from the received signal $r_i[n]$ in a receiver i, as shown in (4). The signal $S[n]$ is the modulation symbol again. This symbol has been chosen in such a way that, there exist always M/m null values in every half period. This fact implies that the component $I_i[n]$ can be obtained directly from the component $Q_i[n]$, by only delaying M/m samples. So, the demodulation process is reduced to only one correlation process.

$$
\begin{aligned}
I_i[n] &= C_{rS_I}[n] = r_i[n] * S_I[n] = \\
&= \sum_{k=0}^{N\cdot M\cdot m-1} r_i[k+n]S[k] \\
Q_i[n] &= C_{rS_Q}[n] = r_i[n] * S_Q[n] = \\
&= \sum_{k=0}^{N\cdot M\cdot m-1} r_i[k+n]S\left[k-\frac{M}{4}\right]
\end{aligned}
\tag{4}
$$

Once obtained the components, the following step is to carry out the search of the Golay complementary pair, each one of them in its corresponding component. This operation could be carried out by means of a classical correlation, what would allow to detect the emitted sequences.

Nevertheless, in [11] an optimized method is shown to obtain an efficient Golay correlator (EGC). This model allows to simplify the detection process, whenever Golay complementary pairs, whose length N is power of 2, are used ($N = 2s$, where s is the number of bits of the sequence seed **W**, such that $\mathbf{W} = [w_0, w_1, ...w_{s-1}]$). The block diagram in Figure 3 shows the algorithm.

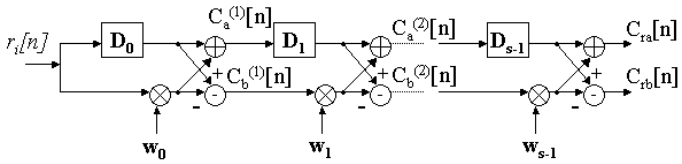

Fig. 3. BLOCK DIAGRAM OF THE EFFICIENT GOLAY CORRELATOR (EGC).

Where D_s is a delay module $D_s = 2 \cdot P_s$; P_s is any permutation of numbers $\{0, 1, ...s-1\}$, represented as $\{w_0, w_1, ...w_{s-1}\}$; $C_{ra}[n]$ and $C_{rb}[n]$ are the results from the correlation between the input signal $r_i[n]$ and the searched sequences $A_i[n]$ and $B_i[n]$.

This optimization presents some advantages compared to the implementation of a classical correlation. Firstly, in the classical version, N multiplications are necessary (being N the length of the used sequences), whereas, using the EGC scheme only $2 \cdot log2N$ multiplications are carried out. Also, the number of memory accesses to store intermediate data decreases from the N accesses to the $2 \cdot log2N$ in the EGC. It is important to remark, regarding the multiplications, that, due to the use of binary sequences with values in $\{-1, +1\}$, all these operations can be reduced to additions and subtractions.

In Figure 3, it is observed that, starting from any input signal $r_i[n]$, the system provides two outputs, $C_{ra}[n]$ and $C_{rb}[n]$, which are the results from the correlation between the input signal $r_i[n]$, and the sequences of the Golay pair $A_i[n]$ and $B_i[n]$, respectively. Considering this idea, the scheme has to be adapted to the specifications of the proposal, since the aim is now to detect or to correlate the Golay pair with the corresponding modulation components I and Q. If the in-quadrature component $Q_i[n]$ becomes the input of the EGC module, then, at the output, the result of the correlation of the component $Q_i[n]$ with the sequence $A_i[n]$ is obtained in the top branch, $C_{Q_iA_i}[n]$, and with the sequence $B_i[n]$ in the bottom one, $C_{Q_iB_i}[n]$. This last term is necessary to obtain the final result $s_i[n]$; however, the sequence $A_i[n]$ should actually be correlated with the component $I_i[n]$. In order to avoid the introduction of a new EGC to obtain the second term, $C_{Q_iA_i}[n]$, this aspect is solved by inserting a delay M/m at the output of the top branch of the EGC, since both components $I_i[n]$ and $Q_i[n]$ are equal, but delayed M/m samples. In this way, the signal $C_{I_iA_i}[n]$ is obtained, and added to $C_{Q_iB_i}[n]$, provides the final signal of the processing $s_i[n]$. Figure 4 shows this adaptation using the general EGC scheme, where it can be observed also that delays D appear multiplied by a factor $M \cdot m$ ($M = 8$, $m = 2$), corresponding to the oversampling existing in the reception.

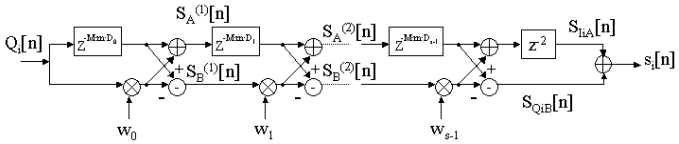

Fig. 4. EGC SCHEME MODIFIED FOR THE PROPOSED ALGORITHM.

Figure 5 shows the signal $s_i[n]$ obtained after all this processing, where the maximum value can be easily identified, corresponding with the instant of the echo arrival.

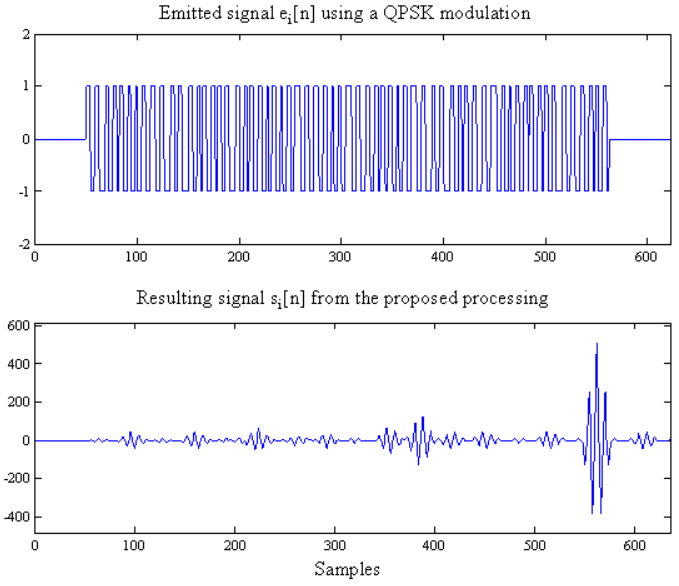

Fig. 5. RESULTING SIGNAL $s_i[n]$ AFTER THE PROPOSED PROCESSING.

The resulting signal $s_i[n]$ should be analyzed by a maximum detector module, whose function is to detect the local maximum values and to validate them as received echoes in the signal $p_i[n]$. A peak is validated as an echo if its value is greater than the threshold U_e, and if it is the maximum value inside an analysis window V_0 of $m \cdot M = 16$ samples. This analysis window V_0 eliminates the validation of the modulation sidelobes as echoes. Figure 6 shows an example for the validation of an echo.

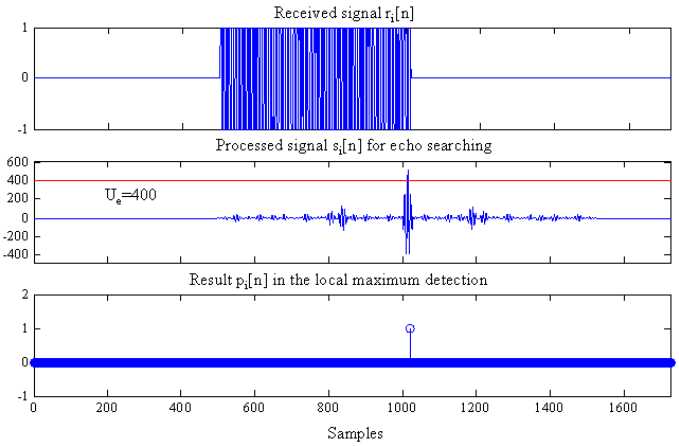

Fig. 6. RESULTING SIGNAL $p_i[n]$ AFTER THE ECHO VALIDATION.

As can be observed in Figure 7, the basic ultrasonic transducer has an emission stage and another reception stage. The emission stage consists of a modulation QPSK, with a 50kHz carrier and a 32-bit Golay complementary pair, characteristic of the transducer. In (2) the emission process of the desired ultrasonic signal is described mathematically, where the sequence $A_i[n]$ is associated to the component I (in-phase), and the sequence $B_i[n]$ to the component Q (in-quadrature). On the other hand, it is possible to carry out the search of until k echoes coming from k different emissions, always orthogonal, in the signal captured by the transducer i. Every different emission has its own EGC detector for the search of the Golay complementary pair; and then a peak detector that allows to detect the local maximum values and to validate them as detected echoes.

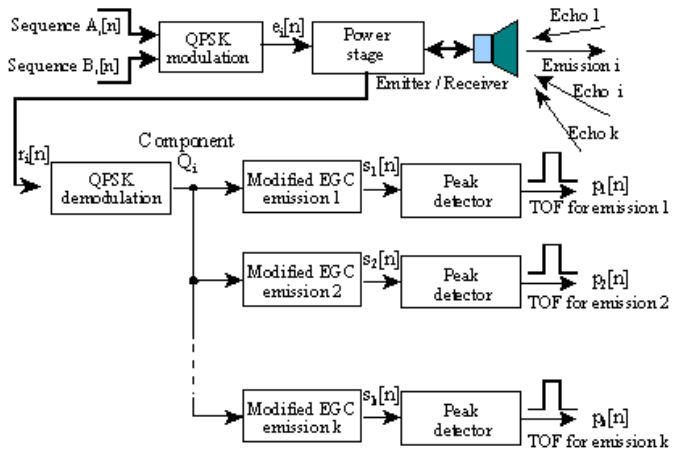

Fig. 7. GENERAL BLOCK DIAGRAM FOR THE GLOBAL ULTRASONIC PROCESSING IN A TRANSDUCER.

Finally, since it is desired to implement in real-time the processing described in this section, a temporal constraint should be considered. This comes determined by the sampling frequency f_S of the received signal whose value is $400kHz$. This implies a processing time interval of $2.5\mu s$ for every new sample. The selected architecture for this implementation should support the execution of the whole algorithm in a shorter time, meeting the considered temporal constraint.

III. HARDWARE ARCHITECTURES ANALYSIS

Once considered all the conditions and constraints presented by the algorithm to be developed, it should be selected which device or architecture is more suitable for their implementation.

Firstly, an architecture based on a TI C6701 DSP has been chosen [12] [13]. The architecture, whose block diagram can be observed in Figure 8, has been programmed by means of a C source code, which implements the algorithm described in the previous section. Assuming the clock frequency is $166MHz$, an approximated execution time of $38.8\mu s$ has been obtained for the processing of the possible echoes coming from two different emissions in a received signal. This value is longer than the maximum one allowed by the sampling frequency f_S, so this platform is not suitable for this real-time processing.

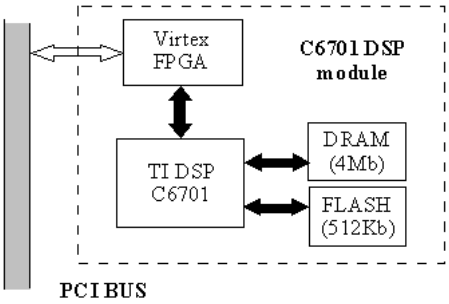

Fig. 8. BLOCK DIAGRAM OF THE TI C6701 DSP-BASED ARCHITECTURE.

Also, the implementation has been proposed and solved in an architecture based on a Xilinx XC4005E FPGA [14] [15], with an external 128Kb RAM memory (see Figure 9). The requirements of the algorithm has been studied carefully, in order to implement only those necessary resources. These resources are used intensively along time for the processing of every sample, so their use is optimized and adapted. This temporal implementation allows to obtain an execution time of $2.5\mu s$, for two emitting transducers in a captured signal. The used clock frequency is $12MHz$, very far from the superior bound of these devices (around $80MHz$). The differences in the resulting times, in comparison with the solution carried out by the C6701 DSP, come mainly from the fact that the system developed in the FPGA device is more adapted to the requirements of the algorithm. On the contrary, it becomes less flexible if some modifications in the processing algorithms were necessary.

Although the results obtained in the previous platform fulfill the temporal constraints of a real-time processing, the reduced number of available resources in the used XC4005 device makes not very appropriate the implementation. It is possible to structure the algorithm in a spatial implementation, so all the required resources are available along time to exploit parallelism. Then, the processing is divided into tasks that can be

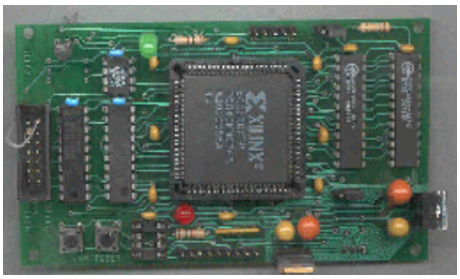

Fig. 9. GLOBAL VIEW OF THE PLATFORM BASED ON A XILINX XC4005E FPGA.

overlapped temporarily to obtain a better performance. Also, it would be suitable to have independent memory banks for every stage, so that the memory accesses do not suppose a constraint for the execution times.

In order to implement the last commented improvements, a Xilinx Virtex FPGA has been used [9] [13] [16]. Figure 10 shows the block diagram of the used platform; the central element is a Xilinx XCV1000E device whose resources allow to implement a pipeline, where every stage has its own internal memory bank. The processing time of a sample is a FPGA clock cycle, i.e. $20ns$ ($f_{CLK-FPGA} = 50MHz$); and the pipeline has been organized in four stages (new sample acquisition, demodulation, EGC, peak detector). Since all the four stages are completed in one cycle, only 20ns are required for every new sample assuming a latency of four samples at the output (see Figure 11). Obviously, this system could be used to improve the proposed algorithm, since the temporal constraint is far away from the values obtained. Possible improvements can be the increase of the sequence length N, or to compute more than one reception with the same platform.

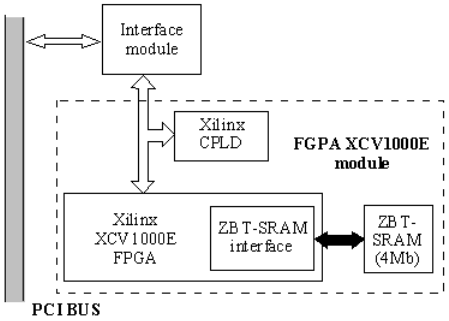

Fig. 10. BLOCK DIAGRAM OF THE XILINX XCV1000E FPGA-BASED ARCHITECTURE.

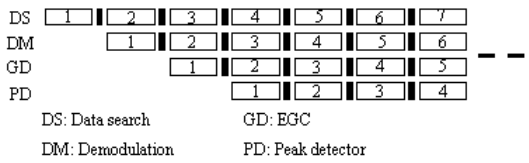

Fig. 11. PIPELINE SCHEME PROPOSED.

TABLE I.
COMPARATIVE ANALYSIS OF PROCESSING TIMES IN THE C6701 DSP PLATFORM FOR DIFFERENT NUMBER OF EMITTERS AND RECEIVERS.

TI C6701 DSP	1 emission	2 emissions	3 emissions
1 receptor	$19.84\mu s$	$38.8\mu s$	$57.7\mu s$
2 receptors	$39.7\mu s$	$77.6\mu s$	$115.5\mu s$
4 receptors	$79.4\mu s$	$155.2\mu s$	$231\mu s$

TABLE II.
COMPARATIVE ANALYSIS OF PROCESSING TIMES IN THE XC4005E FPGA PLATFORM FOR DIFFERENT NUMBER OF EMITTERS AND RECEIVERS.

Xilinx XC4005E FPGA	1 emission	2 emissions	3 emissions
1 receptor	$1.5\mu s$	$2.5\mu s$	$4.3\mu s$
2 receptors	$3\mu s$	$5\mu s$	$8.6\mu s$
4 receptors	$6\mu s$	$10\mu s$	$17.2\mu s$

IV. RESULTS

The data mentioned before correspond to the case of analyzing only one captured signal and searching two orthogonal emissions whose sequences have a length of 32 bits. If the number of the received signals to be analyzed increases, or there exists a larger number of orthogonal emissions, the solution based on the DSP processor will increase its processing time, due to its sequential operation. Nevertheless, as the FPGA-based solutions allow several parallel execution lines, these implementations can be adapted easier to new modifications; although, the amount of required resources increases.

Next the execution times commented previously are collected, so it can be observed how the increase of the number of emitters and receivers affects to each platform. There is a table for each one of the three mentioned platforms, analyzing the variation of the processing time in function of the number of orthogonal emissions existing in a captured signal (1, 2, or 4 emissions), as well as in function of the number of received signals to analyze, coming from different transducers (1, 2 or 4 receivers). In all the cases, 32-bit Golay complementary pairs are used.

Then it seems obvious that the two specific implementations, based on FPGA devices, are more efficient compared to those obtained from DSP-based systems, even when these are specialized in the digital signal processing. A last detail to consider, analyzing the difference between the spatial or temporal implementation of the algorithm in FPGAs, is the amount of resources used by the two possibilities. Table IV shows those values for a length $N = 32$ bits, where it can be confirmed that the spatial implementation is quite more expensive in terms of the amount of used resources.

V. CONCLUSIONS

In the development of more precise ultrasonic sensorial systems and with shorter acquisition time, algorithms are usually based on correlation techniques. These algorithms have been analyzed, establishing their real-time constraints, so, in that way,

TABLE III.

COMPARATIVE ANALYSIS OF PROCESSING TIMES IN THE XCV1000E FPGA PLATFORM FOR DIFFERENT NUMBER OF EMITTERS AND RECEIVERS.

Xilinx XCV1000E FPGA	1 emission	2 emissions	3 emissions
1 receptor	$20\mu s$	$20\mu s$	$20\mu s$
2 receptors	$40\mu s$	$40\mu s$	$40\mu s$
4 receptors	$80\mu s$	$80\mu s$	$80\mu s$

TABLE IV.

RESOURCES USED BY THE TWO PROPOSED FPGA IMPLEMENTATIONS FOR $N = 32$ BITS.

Device Resources	Spatial implementation XCV1000E	Temporal Implementation XC4005E
Logical cells (slices)	1085	94
No. equivalent gates	105123	8640

it can be possible to determine the most suitable architecture for their implementation. The tests have shown as, due to the high degree of existing parallelism in the reception processing phases of the ultrasonic signal, the digital signal processors (DSP) do not provide the desired performances, overcoming the maximum execution time available for the analysis of every new acquired sample. On the other hand, the FPGA devices result very suitable for the implementation of this processing, even those with reduced resources, because they allow mainly a great parallelism degree, with several execution lines and pipeline-type structures.

REFERENCES

[1] K. Audenaert, H. Peremans, Y. Kawahara and J. Van Campenhout, "Accurate ranging of multiple objects using ultrasonic sensors", in *Proc. of IEEE Int. Conf. on Robotics and Automation*, 1992, pp. 1733-1738.

[2] H. Hamadene and H. Colle, "Optimal estimation of the range for mobile robots using ultrasonic sensors", in *Proc. 3rd IFAC Symposium on Intelligent Components and Instrument for Control Applications (SICICA'97)*, Annecy (France), 1997, pp. 141-146.

[3] J. Borenstein and Y. Koren, "Error Eliminating Rapid Ultrasonic Firing for Mobile Robots Obstacle Avoidance", in *IEEE Trans. on Robotics and Automation*, vol. 11(1), pp. 132-138, 1995.

[4] H. Peremans, A. Koenraad and J. Van Compenhout, "A High-Resolution Sensor Based on Tri-aural Perception", in *IEEE Trans. on Robotics and Automation*, vol. 9, no. 1, pp. 36-48. 1993.

[5] B. Barshan and B. Ayrulu, "Fractional Fourier transform pre-processing for neural network and its application to object recognition", in *Elsevier Science Ltd. Neuronal Networks*, no. 14, pp. 355-373, 2001.

[6] P. J. McKerrow and N. L. Harper, "Plant Acoustic Density Profile Model of CTFM Ultrasonic Sensing", in *IEEE Sensors Journal*, vol. 1, no. 4, pp. 245-255, 2001.

[7] H. Peremans and R. Muller, "A comprehensive robotic model for neural & acoustic signal processing in bats", in *Proc. First International IEEE Conference on EMBS Neural Engineering*, Capri (Italy), 2003, pp. 458-461.

[8] V. Díaz, J. Ureña, J. J. García, M. Mazo, E. Bueno and A. Hernández, "Using Golay complementary sequences for multi-mode ultrasonic operation", in *Proc. 7th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA99)*, vol. 1, 1999, pp. 599-604.

[9] A. Hernández, J. Ureña, J. J. García, M. Mazo, D. Hernanz, J. P. Dérutin and J. Sérot "Ultrasonic ranging sensor using simultaneous emissions from different transducers", in *IEEE Trans. on Ultrasonics, Ferroelectrics and Frequency Control*, vol. 51, no. 12, pp. 1660-1670, 2004.

[10] Polaroid Corporation Ltd., *Ultrasonic ranging systems*, Manual and handbook, 1991.

[11] B. M. Popovic, "Efficient Golay correlator", *IEE Electronics letters*, vol. 35, no. 17, 1999.

[12] Texas Instruments Inc., *TMS320C6701. Floating-point digital signal processing*, Datasheet specification, 2000.

[13] Sundance Multiprocessor Technology Ltd., hhtp://www.sundance.com, 2005.

[14] Xilinx Inc., *XC4000E and XC4000X series Field programmable gate arrays*, Product specification, 1999.

[15] A. Hernández, J. Ureña, D. Hernanz, J. J. García, M. Mazo, J. P. Dérutin, J. Sérot and S. Palazuelos, "Real-time implementation of an efficient Golay correlator (EGC) applied to ultrasonic sensorial systems", *Microprocessors and Microsystems*, no. 27, pp. 397-406, 2003.

[16] Xilinx Inc., *Virtex-E 1.8V Field programmable gate arrays*, Advance product specification, 2000.