

Efficient Real-Time Correlator for LS Sequences

M. C. Pérez¹, J. Ureña¹, A. Hernández¹, A. Jiménez¹

¹Department of Electronics
University of Alcalá
Alcalá de Henares, Spain
Email: carmen@depeca.uah.es

W. P. Marnane², F. Álvarez³

²Dept. of Electrical & Electronic Engineering
University College Cork, Cork, Ireland
Email: marnane@ucc.ie

³ Dept. of Electronics & Electromechanical Engineering
University of Extremadura, Cáceres, Spain
Email: fafranco@unex.es

Abstract—The cross-correlation function and the side-lobes of the auto-correlation function of LS codes are zero in a certain vicinity of the zero shift. Therefore, the effects of the Inter-Symbol-Interference and Multiple-Access-Interference that appears in CDMA and multi-sensor systems are mitigated. Conventionally, the detection of these sequences has been achieved by means of straight-forward matched-filter correlators. In this paper, a new correlator which significantly reduces the total number of operations to be performed is proposed, allowing real-time operation.

I. INTRODUCTION

Pulse compression sensor systems and CDMA (Code Division Multiple Access) systems are interference limited. There exists Inter-Symbol-Interference (ISI) due to the non zero auto-correlation (AC) sidelobes of the used sequences. They also have Multiple-Access-Interference (MAI) due to the non zero cross-correlations (CC) values. To avoid the effects of both ISI and MAI, the sidelobes of the AC and the CC values should be as small as possible.

Hence, a lot of research efforts have been devoted to finding sequences which fulfill both conditions as far as possible. Barker codes [2] have been widely used due to their good aperiodic AC. However, their maximal length is limited, so it is not possible to detect them in case of high noise. Furthermore, there are no Barker codes with low CC among them. Pseudo-random sequences such as m-sequences, Gold codes or Kasami codes [3], exhibit non zero off-peak AC and CC values in the case of asynchronous transmissions. Another possibility is Golay pairs [4], which have good aperiodic correlation properties, and their length is not limited. The detection of Golay sequences can be performed by means of efficient correlation algorithms which notably decrease the computational load and hardware complexity [5]. Nevertheless, they provide only two mutually orthogonal pairs, which is not useful for multi-user environments. Complementary sets of M sequences (M -CSS) [6] are a generalization of Golay codes containing more than two sequences. The elimination of the constraint in the number of sequences of a set yields on a high process gain and also M mutually orthogonal sets. To achieve these goals, it is necessary to add the AC functions of the sequences of the set, or the CC functions of the corresponding sequences in the M sets, respectively. The problem is that, in many systems, transducers have limited bandwidths, so it is not possible to

transmit or receive the M sequences of the set simultaneously. Unfortunately, it implies an undesired increase of both ISI and MAI.

Loosely Synchronized (LS) codes [7] exhibit an Interference Free Window (IFW), where the aperiodic AC sidelobes and CC values become zero. Consequently, ISI and MAI are completely reduced if the maximum transmission delay is less than the length of the IFW. Usually, the correlation of these codes is carried out by means of straightforward matched filter implementations [8]. These implementations provide large processing cost; and, in case of a large amount of data to be processed, real-time operation can not be possible without the aid of high complexity hardware.

The contribution of this paper is to propose an *Efficient LS correlator (ELSC)*, which significantly reduces the number of operations performed, in comparison with an straightforward matched filter implementation. The presented ELSC is designed for the correlation of LS codes, generated from Golay complementary pairs, as it is explained in [7]. Also, the practical implementation of the ELSC in a Field Programmable Gate Arrays (FPGA) has been carried out.

The paper is organized as follows. Section II introduces the family of LS codes. In Section III the Efficient Correlator of LS sequences is presented. Section IV shows the hardware implementation of the correlator and some results. In Section V tests with real signals are considered. Finally, conclusions are outlined in Section VI.

II. LS CODES

In [7], a systematic method to construct LS codes by using Golay complementary sequences is presented. It basically consists of using two orthogonal Golay pairs of length N , and of linking the sequences of these pairs directly, or negated, depending on the coefficients of a $(P \times P)$ Hadamard matrix. Also, a set of W_0 zeros has to be inserted in the centre of the LS code G_k . The total length of the LS codes is given by $L = KN + W_0$, where $K = 2P$ is the number of available codes with orthogonal properties in the IFW (with $K = 2^n$, $n \in \mathbb{N}$). The correlation properties of LS sequences ($G_k = [g_{k,1}, g_{k,2}, \dots, g_{k,L}]; 1 \leq k \leq K$) are given by (1). It can be noted that the AC function is obtained when $k_1 = k_2$, whereas the CC function when $k_1 \neq k_2$.

$$R_{G_{k_1}, G_{k_2}}[m] = \sum_{i=0}^{L-m-1} G_{k_1}[i]G_{k_2}[i+m] = \begin{cases} K \cdot N, & \text{for } m = 0, & k_1 = k_2 \\ 0, & \text{for } 1 \leq |m| \leq W_0, & k_1 = k_2 \\ 0, & \text{for } 0 \leq |m| \leq W_0, & k_1 \neq k_2 \end{cases} \quad (1)$$

Prior to describe the proposed ELSC, it is necessary to give more details about the generation method of LS codes presented in [7]. Also, the notation $LS(N, P, W_0)$ mentioned in [9] has been adopted.

Firstly, two orthogonal Golay pairs (c_0, s_0) and (c_1, s_1) of length N are defined. The AC and CC properties of these pairs should satisfy (2):

$$R_{c_i, c_j}[m] + R_{s_i, s_j}[m] = \begin{cases} 2 \cdot N, & \text{for } m = 0, & i = j \\ 0, & \text{for } m \neq 0, & i = j \\ 0, & \text{for } \forall m, & i \neq j \end{cases} \quad (2)$$

Golay pairs (c_0, s_0) and (c_1, s_1) are linked depending on the coefficients of a Hadamard matrix H , and on a vector π . $H = [h_{i,j}]$ is a $P \times P$ Hadamard matrix, whose $h_{i,j} = \pm 1$ elements determine the final polarity of the Golay pairs. The vector $\pi = [\pi_1, \dots, \pi_P]$, $\pi_k \in \{0, 1\}$ denotes a binary expansion of an arbitrary integer n , $0 \leq n < 2^P$ so that $n = \sum_i \pi_i \cdot 2^i$. This vector π indicates which Golay pair, (c_0, s_0) or (c_1, s_1) , is taken into account at every moment. Furthermore, a second vector $\pi^* = [\pi_1^*, \dots, \pi_P^*]$, $\pi_k^* = \pi_k + 1 \pmod{2}$ for each $1 \leq k \leq P$, has to be considered. The first P sequences of the set G_1, \dots, G_P are constructed as it is shown in (3). Whereas the next G_{P+1}, \dots, G_K are obtained when the vector π in (3) is substituted by its complement π^* . For brevity, polynomial notation has been used. In this notation, any bipolar sequence $a = [a_0, a_1, \dots, a_{N-1}]$, is simply replaced by the polynomial $A(z) = \sum_{n=0}^{N-1} a_n z^n$, in the Laurent Polynomial ring $\mathbb{Z}[z, z^{-1}]$.

$$G_k(z) = \sum_{i=1}^P h_{k,i} [C_{\pi_i}(z) + z^{PN+W_0} S_{\pi_i}(z)] z^{(i-1)N} \quad (3)$$

Thus, a set of K LS codes with length $L = KN + W_0$ is obtained. For periodic emissions it is necessary to insert a guard interval of, at least, W_0 zeros between sequences, preventing them for overlapping. In these cases $L = KN + 2W_0$. The LS codes generated exhibit an IFW of length $W = \min \{2N - 1, 2W_0 + 1\}$, considering $W_0 = N - 1$ in most of cases [7][9].

Fig. 1 shows the aperiodic AC of a $LS(16, 2, 15)$ sequence; whereas in Fig. 2 can be seen the aperiodic CC among different $LS(16, 2, 15)$ codes from the same set.

III. EFFICIENT LS CORRELATOR

When longer LS sequences have to be processed in real-time, the increasing in the computational load demands the use of efficient correlators, able to perform the detection

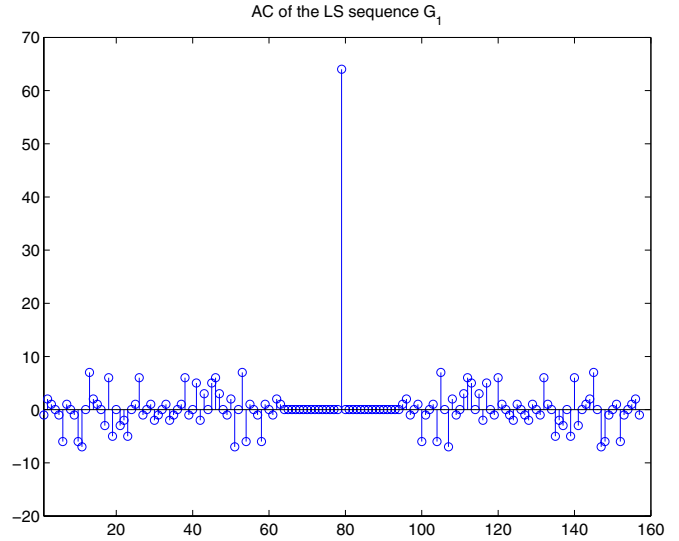


Fig. 1. AC function for a sequence $LS(16, 2, 15)$.

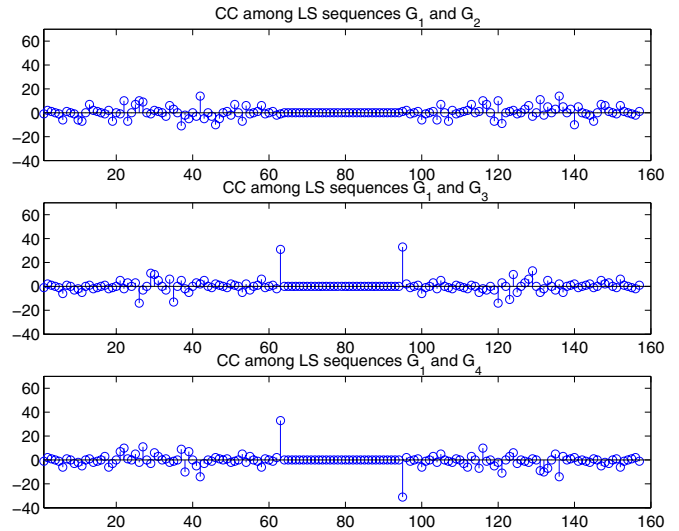


Fig. 2. CC function among the sequences of a set generated as $LS(16, 2, 15)$.

of these sequences with affordable computational cost. In this paper, an efficient correlator for LS sequences (ELSC) has been developed; which decreases the total number of operations performed, in comparison with a straight-forward matched filter implementation. This correlator can be easily implemented on reconfigurable hardware to achieve real-time operation.

The proposed ELSC correlator exploits the properties of the Golay codes to simplify the correlation process. Golay pairs, (c_0, s_0) and (c_1, s_1) , that form the LS sequences can be correlated by means of the Efficient Golay Correlator (EGC) developed in [5]. Later, the corresponding outputs of the EGCs are delayed and added or subtracted, depending on the values of the vector π and the Hadamard matrix H .

The block diagram of Fig. 3 shows the EGC algorithm structure. It is composed by a set of M similar stages, where $N = 2^M$, with $M \in \mathbb{N} - \{0\}$, is the length of the Golay

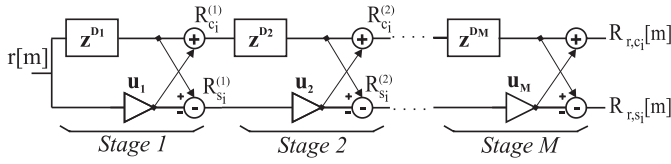


Fig. 3. Block Diagram of the Efficient Golay Correlator.

pairs. Every stage is composed of a delay module z^{D_i} , where $D_i = 2^{P_s}$, and P_s is any permutation of the numbers $0, 1, \dots, M-1$; a multiplier u_i , where $U = [u_0, u_1, \dots, u_{M-1}]$ is the generation seed of the Golay pair; an adder; and a subtracter. It is important to note that, since the coefficients u_i are binary, multiplications are reduced to additions and subtractions. Starting from any input signal $r[m]$, the EGC provides two outputs, $R_{r,c_i}[m]$ and $R_{r,s_i}[m]$; which are the correlation between the input signal $r[m]$, and the sequences of the Golay pair (c_i, s_i) . The number of multiplications in the EGC is equal to $\log_2 N$, whereas in the straightforward matched filter implementation it would be N . Also, the number of additions is reduced from the $N-1$ operations in the straightforward to $2 \cdot \log_2 N$ in the EGC.

Considering the EGC, the scheme has to be adapted to the features of LS codes. Since two orthogonal pairs, (c_0, s_0) and (c_1, s_1) , form the LS codes, two EGC are needed. One EGC carries out the correlation among the input sequence $r[m]$ and the Golay pair (c_0, s_0) , generated with seed U_0 . It provides the outputs $R_{r,c_0}[m]$ and $R_{r,s_0}[m]$. The other EGC performs the correlation with the Golay pair (c_1, s_1) , generated with seed U_1 , and gives the outputs $R_{r,c_1}[m]$ and $R_{r,s_1}[m]$.

Vectors π or π^* used to generate the LS sequence determine which correlation outputs, $R_{r,c_0}[m]$, $R_{r,s_0}[m]$, $R_{r,c_1}[m]$ or $R_{r,s_1}[m]$, have to be delayed. Note that the order of the delays in (3) has to be interchanged for the correlation. Therefore, the output corresponding with the correlation between the input signal and the sequence of the Golay pairs that firstly appear in the LS code, has to be delayed by $z^{PN+W_0+(P-1)N}$. Later, the delayed outputs are added or subtracted, according to the column of the Hadamard matrix used in the LS code generation. Eq. (4) summarizes the operations performed by the ELSC. Furthermore, Fig. 4 depicts the block diagram of the ELSC.

$$R_{G_k} = \sum_{i=1}^P h_{k,i} [z^{PN+W_0} R_{C_{\pi_i}}(z) + R_{S_{\pi_i}}(z)] z^{(P-i)N} \quad (4)$$

This optimization presents some advantages compared to the implementation of a classical straight-forward correlation. Table I shows the operations carried out by the two correlation schemes. As has been stated before, multiplications are reduced to additions and subtractions, due to the binary nature of their coefficients. Regarding the notation, DW is the number of bits of the input signal; and $O_s = \frac{\text{acquisition frequency}}{\text{emission frequency}}$ is the over-sampling factor used when the sequences are acquired. It can be observed that the number of operations is

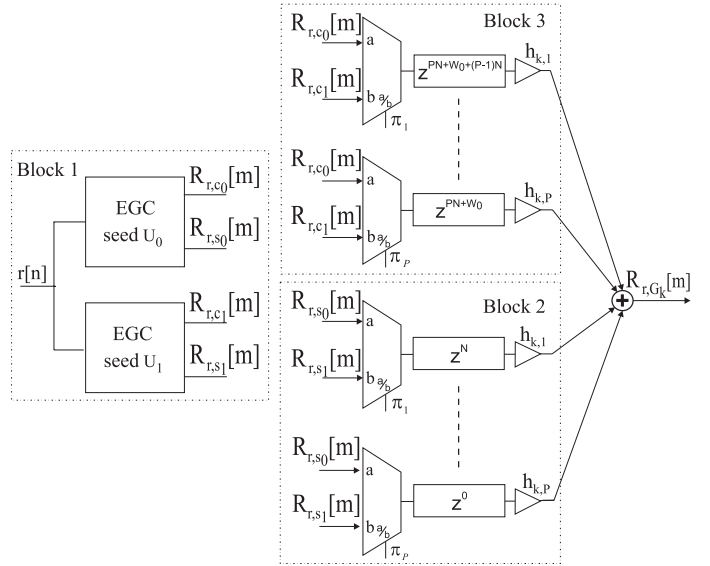


Fig. 4. Block Diagram of the Efficient Loosely Synchronized Correlator.

significantly reduced in the case of the ELSC implementation. Nevertheless, the total number of memory bits to store the data in the ELSC, is larger than the corresponding in the straight-forward implementation, as can be seen in Table II. To expound further, Fig. 5 and Fig. 6 depict the information given by Tables I and II respectively. To generate both figures, a set of LS codes with $K = 16$ sequences has been used, with $DW = 8$, and $O_s = 10$. The length N of the initial Golay sequences can take values $N = [8, 16, 32, 64, 128, 256, 512, 1024]$; and the number of zeros in the centre of the LS code is $W_0 = N - 1$.

IV. ELSC IMPLEMENTATION

A. Design Strategy

A generic hardware implementation in a FPGA of the ELSC has been developed. The correlator parameters can be changed through synthesizing the design. Therefore, it is possible to configure the number of LS codes available in the set through

TABLE I
OPERATIONS TO PERFORM FOR THE CORRELATION OF LS CODES, USING AN STRAIGHTFORWARD IMPLEMENTATION AND AN ELSC.

Implementation	Multiplications	Additions
Straight-forward	$L = KN + W_0$	$L - 1 = KN + W_0 - 1$
ELSC	$2\log_2 N + K$	$4\log_2 N + 1$

TABLE II
MEMORY REQUIRED TO STORE THE DATA IN THE CORRELATION OF LS CODES, USING AN STRAIGHTFORWARD IMPLEMENTATION AND AN ELSC.

Implementation	Memory requirements
Straight-forward	$L \cdot O_s \cdot DW + L + (DW + \log_2 L)$
ELSC	$2 \cdot O_s \cdot [DW \cdot \frac{N}{2} + (DW + 1) \cdot \frac{N}{4} + (DW + 2) \cdot \frac{N}{8} + \dots + (DW + \log_2 N - 1)] + O_s \cdot (DW + \log_2 N - 1) \cdot (\frac{K}{2} \cdot N + W_0) \cdot \frac{K}{2} + \sum_{i=1}^{\frac{K}{2}-1} [2 \cdot N \cdot i]$

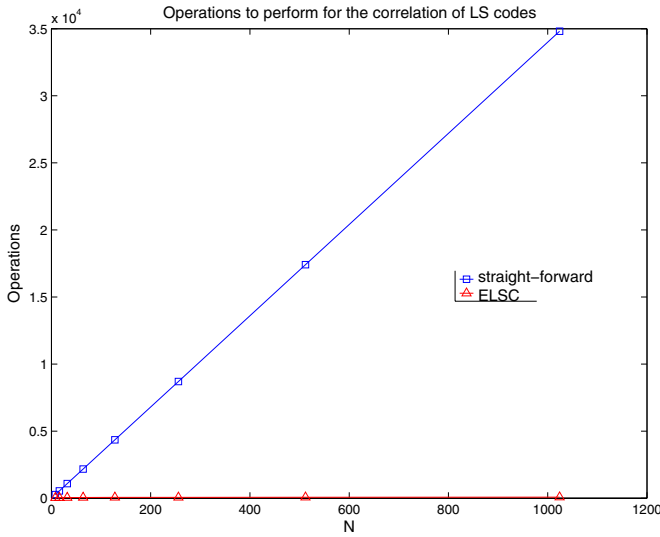


Fig. 5. Operations versus N performance comparison of an ELSC and a straight-forward matched filter implementation of a correlator for LS codes. $K = 16$ users are supported, with $DW = 8$ and $O_s = 10$.

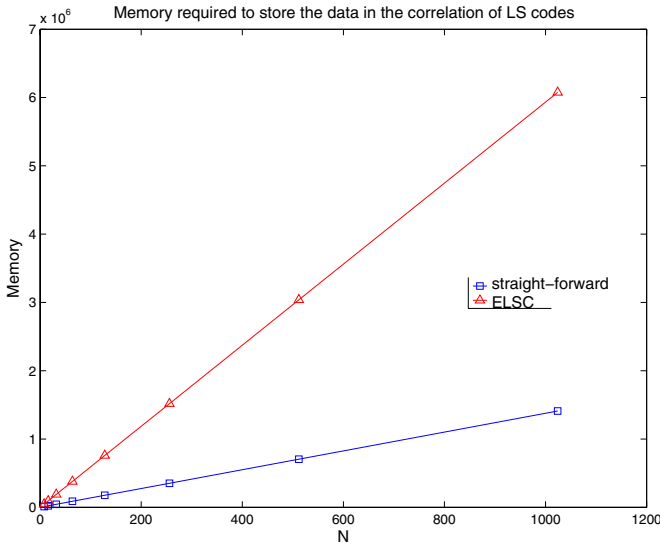


Fig. 6. Memory versus N performance comparison of an ELSC and a straight-forward matched filter implementation of a correlator for LS codes. $K = 16$ users are supported, with $DW = 8$ and $O_s = 10$.

the parameter n ($K = 2^n$). The number of zeros W_0 inserted in the centre of the LS code can be selected as well. It is possible to configure the number M of stages in the EGC, and therefore the length $N = 2^M$ of the initial Golay codes. The data-width DW can be selected, in order to have more accuracy in the obtained results. Finally, the the over-sampling rate O_s can be also selected by the user.

Fig. 7 shows the structure of the hierarchical root block for the ELSC, where $r[m]$ is the input signal; U_0 and U_1 are the generation seeds of the Golay pairs; P_i and Had are the vector π and the column of the Hadamard matrix H used in the generation of the LS codes; and finally R_{r,G_k} is the correlation

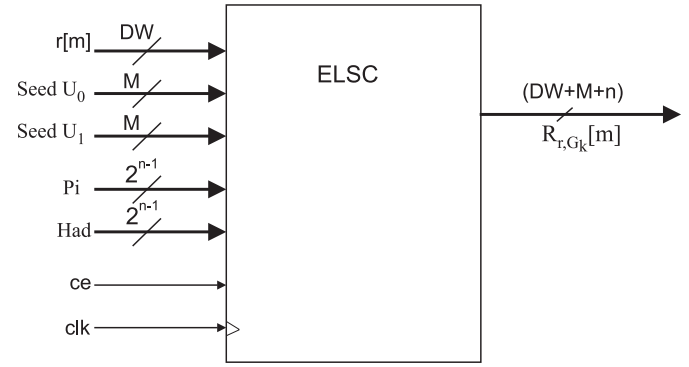


Fig. 7. Input and output ports of the proposed ELSC structure.

function between the input signal $r[m]$ and the considered LS sequence G_k . Note that the output R_{r,G_k} requires $DW + M + n$ bits if overflow is avoided in internal operations.

The internal architecture of the ELSC is illustrated in Fig. 8. It is based on six different modules. The first one implements the EGC (see Fig. 3), and it is based on M similar stages. As has been mentioned before, the number of stages is configurable in order to increase the adaptability of the system. Every stage is divided into two modules: a sequential and a combinational one. The sequential block contains the specific delay of the stage. Note that, at every stage, the number of bits required to store the partial results is increased. Hence, to reduce the total memory required by the correlator, the larger delays should be placed at the first stages, and the smaller ones at the last stages. On the other hand, all these delay elements have been increased by O_s , to consider the sampling factor used in the acquisition of the input signal. As a result, $D = [D_1, D_2, \dots, D_M] = [2^{M-1} \cdot O_s, 2^{M-2} \cdot O_s, \dots, 2^0 \cdot O_s]$. These delays have been implemented by making use of specific shift registers in Xilinx FPGA's architecture (SRL16 modules [10]). The combinational module carries out the operations of the stage. The binary values of the seeds U_0 and U_1 determine the final configuration of the adders and subtractors in the i -th stage.

The second block, called MUX, generates a set of P multiplexers governed by the input P_i . Every multiplexer has the same two inputs, a and b ; which can be connected either with (R_{r,c_0}, R_{r,c_1}) , or with (R_{r,s_0}, R_{r,s_1}) . If $P_i(i) = '0'$ then the multiplexer output is a , whereas if $P_i(i) = '1'$ the multiplexer output is b .

When the outputs of the EGCs are selected, they have to be delayed according to (4). There are two different delays blocks: one implements P delays, all of them with value $z^{(PN+W_0)}$; the other contains the $[z^0, z^1, \dots, z^{(P-1)N}]$ delays. Both blocks use the basic module SRL16 of Xilinx FPGA's architectures.

The Hadamard block determines the polarity of the delayed EGC's outputs. $Had(i) = '0'$ denotes negating the input sequence i ; while $Had(i) = '1'$ means that the output sequence i and the input sequence i are both equal.

Finally, the P outputs of both Hadamard blocks are added

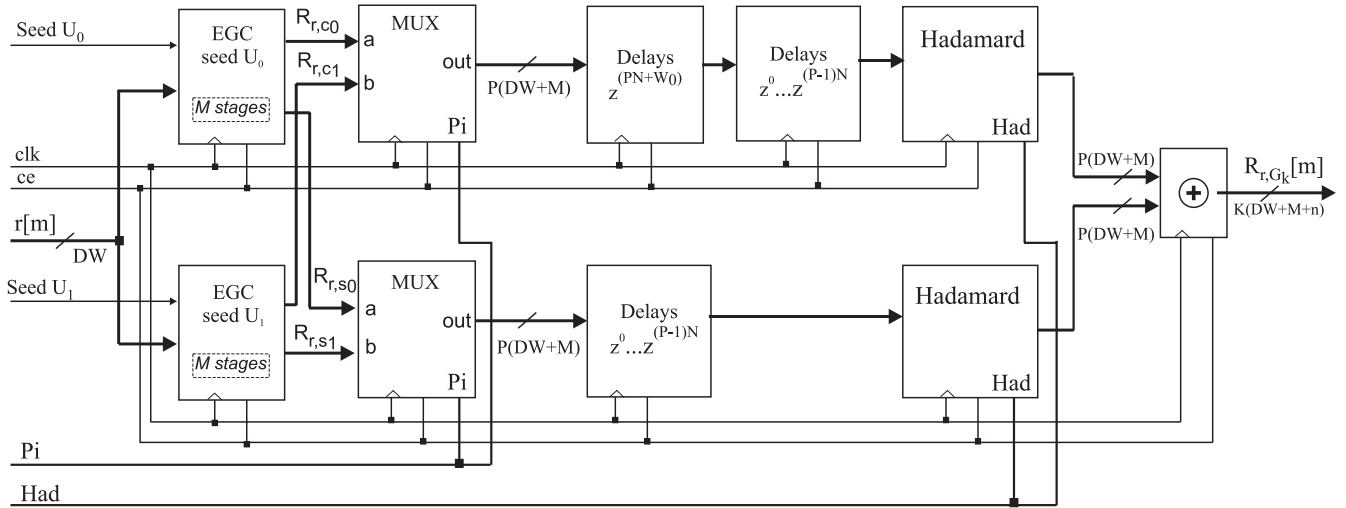


Fig. 8. Internal architecture of the ELSC.

to obtain the correlation output R_{r,G_k} .

B. Performance of the Hardware Implementation

The result requirements and maximum frequencies in the ELSC implementation depend on the (N, P, W_0) parameters of the LS code. In Table III the resource requirements of $LS(8, 2, 7)$, $LS(16, 2, 15)$ and $LS(8, 4, 7)$ are shown. An over-sampling factor $O_s = 10$ has been used, with a data-width $DW = 8$. The correlation system has been implemented in a Spartan3 xc3s1500 FPGA by Xilinx [10].

A performance comparison between the ELSC and a straight-forward matched filter implementation has been achieved. Fig. 9 depicts the hardware implementation scheme of a straightforward correlator based on internal BRAM memory [11]. The input signal is digitalized with a sampling frequency f_s , and stored in a buffer of size $L \cdot O_s$, that has been implemented in BRAM blocks [10]. This sampling buffer is read every $f_{FPGA} = L \cdot f_s$. At each reading, a sample is added or subtracted with the previous accumulated result, depending on the LS code evaluated. Whenever a new sample is received, the accumulated value is reset in order to compute a new correlation. The access to the sampling buffer is carried out in gaps of O_s positions, by taking into account the sampling frequency. In Table IV the requirements for this implementation are shown, in case of using a $LS(16, 2, 15)$. Correlation results are obtained every $t_s = t_{FPGA} \cdot L = 8.01 \text{ ns} \cdot 79 = 632.95 \text{ ns}$, being $t_s = \frac{1}{f_s}$ and $t_{FPGA} = \frac{1}{f_{FPGA}}$. Nevertheless, the ELSC implementation provides a new correlation result every FPGA clock cycle $t_{FPGA} = t_s = 10.23 \text{ ns}$, assuring real-time operation, although it requires more resources.

To increase the operation frequency of the straight-forward correlator, tasks have to be overlapped temporarily, which implies introducing registers to store intermediate data. Therefore, the amount of required resources increases, approaching to the required by the ELSC implementation. Thus, it can be stated that the ELSC implementation is more suitable for real-time correlation of LS sequences.

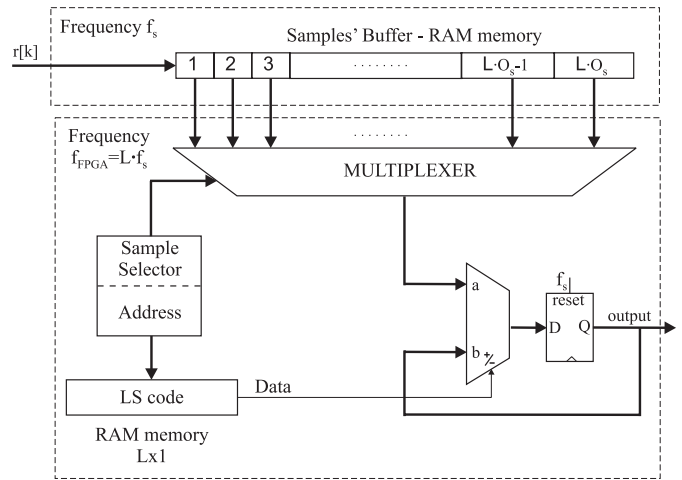


Fig. 9. Sequential implementation of a straight-forward correlator based on internal BRAM memory.

TABLE III
RESOURCES REQUIRED BY AN ELSC IMPLEMENTATION.

xc3s1500	LS(8,2,7)	LS(16,2,15)	LS(8,4,7)
Slices	720	1369	2112
Luts	929	1636	2432
IOBs	33	36	38
Max. Freq. $f_s = f_{FPGA}$	116.550 MHz	97.771 MHz	84.374 MHz
Correlation time $t_s = t_{FPGA}$	8.58 ns	10.23 ns	11.85 ns

TABLE IV
RESOURCES REQUIRED BY A STRAIGHT-FORWARD IMPLEMENTATION.

xc3s1500	LS(16,2,15)
Slices	51
Luts	83
IOBs	27
Max. Freq. f_{FPGA}	124.813 MHz
Max. Freq. f_s	1.58 MHz
Correlation time t_s	632.95 ns

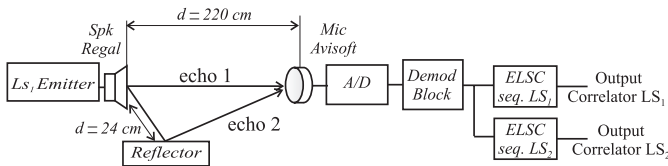


Fig. 10. Experimental setup for the tests carried out.

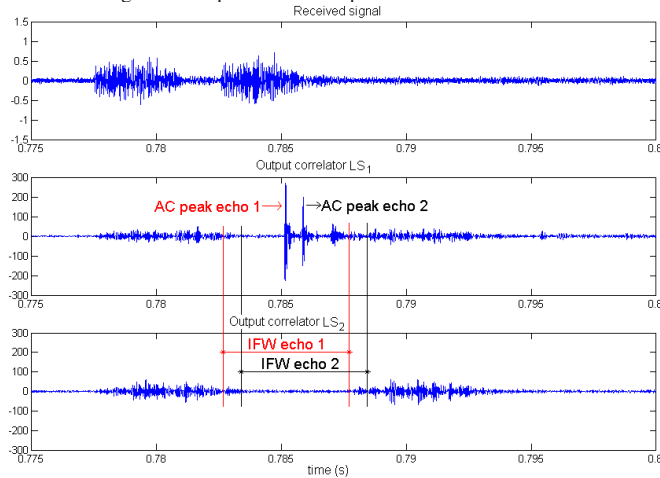


Fig. 11. Reception and correlation results of a real ultrasonic signal.

V. APPLICATION EXAMPLE

The suitability of the ELSC has been tested in the ultrasonic pulse compression system shown in Figure 10. A set of two LS codes $LS(64, 1, 63)$ is generated, although only one sequence (LS_1) is transmitted. A REGAL-RH16E [12] high power speaker has been used to transmit LS_1 with a BPSK modulation scheme, which employs a symbol with one period of a 25 kHz squared signal. Two echoes (echo 1 and echo 2) of LS_1 are received, due to the multipath caused by a reflector. These echoes are captured by a condenser microphone capsule Avisoft Bioacoustic CM16 [13], amplified and digitalized at a sampling frequency of 500 kHz, so $O_s = 20$. Afterwards, the resulting signal is demodulated and, by using two ELSCs, it is correlated with the two original sequences (LS_1 and LS_2) of the set. Figure 11 shows the results obtained in every ELSC. At the output of the correlator LS_1 only echo 1 will be validated. Echo 2 arrives within the IFW of echo 1, and, due to the AC property, there is no interference impact in this zone. Also, because of the CC property, there is no interference at the output of the correlator LS_2 .

VI. CONCLUSIONS

Traditional sequences used in pulse compression and CDMA systems, present ISI and MAI. LS sequences have ideal aperiodic correlation properties in a small window around the zero shift, removing the ISI and MAI completely in this window. If real-time operation is a requirement of the application, traditional straight-forward matched filter implementations are unsuitable. In this paper, an efficient correlator for LS sequences, generated with the method detailed in [7], is

presented. It notably decreases the total number of operations to carry out, achieving real-time operation.

A generic hardware implementation in FPGAs of the proposed LS correlator has been developed. The design is characterized by five generic parameters: the number of codes in the set; the number of zeros in the centre of the LS code; the length of the Golay pairs used to generate the LS codes; the width of the data-path; and the sampling factor. Hence, the result is a configurable module, able to be adapted to the requirements of the application. The performance of the correlator has been tested on a Xilinx Spartan3 FPGA, and compared with the results obtained with a straight-forward implementation. Tests have shown that, for real-time operation, the correlator presented in this paper is more adequate. Furthermore, it has been included on an ultrasonic sensory system, to verify the detection capability in case of working in a multipath environment.

ACKNOWLEDGMENT

This work has been possible thanks to the Madrid Community (FPI grant), from the Spanish Ministry of Science and Technology (RESELAI: TIN2006-14986-CO2-01) and Spanish Ministry of Promotion (VIATOR: ref 70025-T05).

REFERENCES

- [1] Pingzhi Fan, Michael Darnell, "Sequence Design for communications applications," *Research Studies Press LTD*, ISBN: 0 471 96557 X, 1996.
- [2] S. W. Golomb, R. A. Scholtz, "Generalized Barker sequences" *IEEE Trans. Inf. Theory*, vol. IT-11, no. 4, Oct. 1965, pp. 533-537.
- [3] D. Sarwate, M. Pursley, "Crosscorrelation Properties of Pseudorandom and Related Sequences," *Proc. IEEE*, vol.68, no. 5, May 1980, pp. 593-619.
- [4] M. J. E. Golay, "Complementary series," *IRE Trans. Inform. Theory*, vol. 7, no.2, Apr. 1961, pp. 82-87.
- [5] B. Popovic, "Efficient Golay Correlator," *Electron. Lett.*, vol. 35, no. 17, Aug. 1999, pp. 1427-1428.
- [6] C. C. Tseng, C. L. Liu, "Complementary sets of sequences," *IEEE Trans. Inf.*, vol. IT-18, no. 5, Sept. 1972, pp. 644-652.
- [7] S. Stańczak, H. Boche, "Are LAS-codes a miracle?," *Proc. IEEE Global Telecommunications Conf. GLOBECOM'01*, San Antonio, Texas, vol. 1, Nov. 2001, pp. 589-593.
- [8] M. Hahn, G. Friedman, E. Titlebaum, "A comparison of Analog and Digital Circuit Implementations of Low Power Matched Filters for Use in Portable Wireless Communication Terminals," *IEEE Trans. on circuits and systems II: analog and digital signal processing*, Vol. 44, no. 6, Jun. 1997, pp. 498-506.
- [9] S. Ni, H. Wei, J. S. Blough, L. Hanzo, "Network Performance of Asynchronous UTRA-like FDD/CDMA Systems using Loosely Synchronised Spreading Codes," *Proc. IEEE Vehicular Technology Conf. 2003*, Orlando, FL, USA, vol. 2, Oct. 2003, pp. 1359-1363.
- [10] Xilinx, Inc., "Spartan-3E FPGA Family: Complete Datasheet"; *Product Documentation*, Nov. 2005.
- [11] M. C. Pérez, A. Hernández, J. Ureña, C. De Marziani, A. Jiménez, "FPGA-based Implementation of a Correlator for Kasami sequences," *Proc. 11th IEEE International Conference on Emerging Technologies and Factory Automation ETFA'06*, Sept. 2006, 1141-1144.
- [12] REGAL ELECTRONIC: "Regal Electronic Speakers"; *Product Spec.* (2006).
- [13] AVISOFT BIOACOUSTICS: "Avisoft Bioacoustics UltraSoundGate microphone capsule CM16"; *Product Spec.* (2006).