Kayani, J. K. and Steve F. Russell, "A Computationally Efficient Correlator for Pseudo-Random Correlation Systems." IEEE 6th International Symposium on Spread-Spectrum Technology and Applications, New Jersey Institute of Technology, New Jersey, USA, Sept 6-8, 2000.



# A Computationally Efficient Correlator for Pseudo-Random Correlation Systems

Jahangir K. Kayani
Advanced Engineering Research Organization
P.O. Box 145, Haripur, Pakistan

Steve F. Russell
ECPE Department, Iowa State University
Ames, Iowa 50011

***Abstract*: Pseudo-random correlation systems find their application in many engineering fields like, communications, nondestructive testing, medical imaging, and geophysics. The heart of such systems is a correlator, which performs the crosscorrelation between the received signal and a reference waveform. A new method of DSP-based correlator implementation is discussed. It exploits the structural characteristics of pseudo-random waveforms and performs the crosscorrelation of any digitized waveform with a reference pseudo-random waveform, in a manner, much more efficient in terms of processing speed and hardware requirements. This new method can be applied for baseband or bandpass waveforms, and it can handle a wide range of modulation schemes and signaling structures. In order to achieve greater resolution of the calculated correlation function, it is possible to compute the correlation function for the lag values in fraction of the basic chip interval.**

## I. INTRODUCTION

Pseudo-random correlation systems find their application in many engineering fields like, communications, nondestructive testing, medical imaging, and geophysics [1, 2, 3]. They provide a convenient means to improve the dynamic range and the signal-to-noise ratio of the measurement, without reducing the resolution and without having to increase the peak power of the transmit waveform. There is, however, a price to pay in terms of greater system complexity and complicated signal processing. Until recently, the signal processing equipment has been prohibitively expensive for the widespread application of pseudo-random correlation systems. However, due to the advances in the ASIC technology, and the digital signal processing techniques, practical system implementations are becoming realizable.

A new method of DSP-based correlator implementation is discussed. It exploits the structural characteristics of pseudo-random waveforms and performs the crosscorrelation of any digitized waveform with a reference pseudo-random waveform, in a manner, much more efficient in terms of processing speed and hardware requirement. This new method can be applied for baseband or bandpass waveforms, and it can handle a wide range of modulation schemes and signaling structures. In order to achieve greater resolution of the calculated correlation function, it is possible to compute the correlation function for the lag values in fraction of the basic chip interval.

## II. PSEUDO-RANDOM CORRELATION SYSTEMS

The model of a typical pseudo-random correlation system is given in Figure 1. The heart of the system is a correlator that calculates the correlation between the pseudo-random excitation waveform and the received waveform. The correlator implemented digitally is generally much superior in performance. The correlator can either be designed to perform a periodic (circular) correlation or a linear (aperiodic) correlation. Whenever possible, a periodic correlation is preferred as it performs better than its linear counterpart.

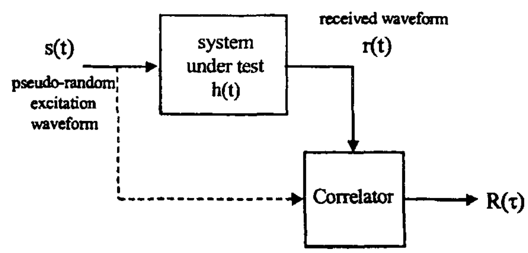


Figure 1: Pseudo-random correlation system model.

## III. EXISTING CORRELATORS

The earlier correlators were of analog type and their performance was relatively poor [1]. The present state-of-the-art correlators are DSP-based, with much improved performance. However, the process is still computation intensive and requires costly computational resources. Two main techniques exist for the implementation of a digital correlator. These are, (a) Time-domain delay-multiply-add correlator (Figure 2) and (b) Frequency-domain FFT based correlator (Figure 3).

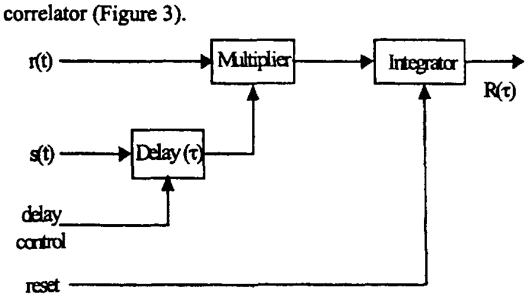


Figure 2: Basic time-domain correlator block diagram.

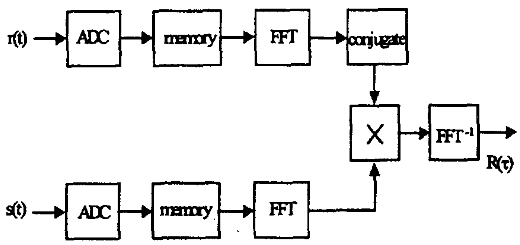


Figure 3: FFT-based correlator block diagram.

While the time-domain approach is simple and straight forward, it performs poorer in terms of computational efficiency. The frequency-domain approach is relatively efficient, though still slow and expensive. New devices and techniques are needed to reduce these costs and permit the wider use of correlators and the correlation systems.

## IV. NEW CORRELATOR

A new method of DSP-based correlator implementation is discussed. It exploits the structural characteristics of a pseudo-random waveform based on the maximal-length sequence. The method can be applied for baseband or bandpass waveforms, and it can handle a wide range of modulation schemes and signaling structures. In order to achieve greater resolution of the calculated correlation function, it is possible to compute the correlation function for the lag values in fraction of the basic chip interval. The new method eliminates various kinds of redundancies in the basic correlation process. The correlation operation is broken into pieces and transformed into a form where the benefits of fast Hadamard transform are utilized (Figure 4). The resulting data is regrouped and transformed back to the standard form. This method performs most of the mathematical operations in the fixed point arithmetic format, thereby saving lot of storage space and processing time.

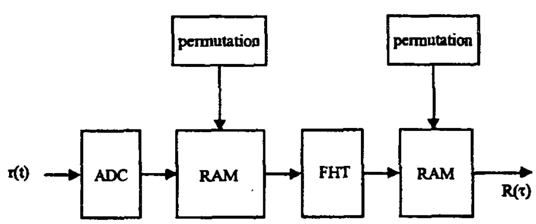


Figure 4: FHT-based correlator block diagram.

## V. MATHEMATICAL DESCRIPTION

The correlator calculates the cross-correlation function between the received waveform and a reference waveform. Let $r(n)$ and $s(n)$ represent the discrete-time versions of the received waveform and the reference waveform, respectively, each of period $M$. The waveforms in the vector notation can be represented as,

$$\underline{r} = [r_0 \quad r_1 \quad r_2 \quad \cdots \quad r_{M-1}] \tag{1}$$

$$\underline{s} = [s_0 \quad s_1 \quad s_2 \quad \cdots \quad s_{M-1}] \tag{2}$$

The pseudo-random reference waveform, $s(n)$, is derived from a two-valued maximal-length sequence, $c(n)$, and the signalling waveform, $p(n)$. This can be represented as,

$$\underline{c} = [c_0 \quad c_1 \quad c_2 \quad \cdots \quad c_{L-1}] \tag{3}$$

$$\underline{p} = [p_0 \quad p_1 \quad p_2 \quad \cdots \quad p_{N-1}] \tag{4}$$

$$\underline{s} = [c_0\underline{p} \quad c_1\underline{p} \quad c_2\underline{p} \quad \cdots \quad c_{L-1}\underline{p}] \tag{5}$$

where, $L$ is sequence length, $N$ is chip length, $M$ is waveform length, and $M = NL$. The non-normalized periodic (circular) cross-correlation of $r(n)$ and $s(n)$, is defined as,

$$\phi(k) = \sum_{n=0}^{M-1} r(n)s(n-k) \tag{6}$$

where $k$ represents the correlation lag value in samples such that, $k = 0, 1, 2, \ldots, (M-1)$. The cross-correlation operation represented by equation (6) is highly computation intensive. Each correlation value requires $M$ multiplications and $(M-1)$ additions. In order to reduce the computational load, equation (6) will be transformed into a form most appropriate for DSP-based implementation of the cross-correlation function.

If only the correlation lag values in multiples of the chip interval are computed, equation (6) gets modified to,

$$\phi(m) = \sum_{n=0}^{M-1} r(n)s(n-Nm) \tag{7}$$

where, $m = 0, 1, 2, \ldots, (L-1)$. By exploiting the structure in the reference vector $\underline{s}$, as shown in equation (5), the summation in equation (7) can be broken down into two parts, such that,

$$\phi(m) = \sum_{x=0}^{L-1} \sum_{n=0}^{N-1} p(n)c(x-m)r(xN+n) \tag{8}$$

Equation (8) in terms of matrix manipulation can be written as,

$$\begin{bmatrix} \phi_0 \\ \phi_1 \\ \cdot \\ \cdot \\ \phi_{L-1} \end{bmatrix} = \begin{bmatrix} c_0 & c_1 & c_2 & \cdot & \cdot & c_{L-1} \\ c_{L-1} & c_0 & c_1 & \cdot & \cdot & c_{L-2} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ c_1 & c_2 & c_3 & \cdot & \cdot & c_0 \end{bmatrix} \begin{bmatrix} r_0 & r_1 & \cdot & r_{N-1} \\ r_N & r_{N+1} & \cdot & r_{2N-1} \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & r_{LN-1} \end{bmatrix} \begin{bmatrix} p_0 \\ p_1 \\ \cdot \\ p_{N-1} \end{bmatrix}$$

$$\underline{\phi} = C \cdot R\,\underline{p} \tag{10}$$

where, $\boldsymbol{C}$ is a right circulant matrix whose first row is the vector $\underline{c}$, and each successive row is obtained from the previous one by a single circular shift operation. The matrix $\boldsymbol{R}$ consists of the elements of the received vector $\underline{r}$, arranged in $L$ rows and $N$ columns. Since the elements of $C$ are two valued, i.e., $c_i = \{A,B\}$, where $A$ and $B$ are real numbers, it is possible to transform $C$ into $X$ through the transformation $C = \{aX + bY\}$, such that $x_i = \{+1, -1\}$. Hence, equation (10) can be written as,

$$\underline{\phi} = a(X \cdot R\,\underline{p}) + b(Y \cdot R\,\underline{p}) \tag{11}$$

and in matrix form as,

$$\begin{bmatrix} \phi_0 \\ \phi_1 \\ \cdot \\ \cdot \\ \phi_{L-1} \end{bmatrix} = a \begin{bmatrix} x_0 & x_1 & x_2 & \cdot & \cdot & x_{L-1} \\ x_{L-1} & x_0 & x_1 & \cdot & \cdot & x_{L-2} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ x_1 & x_2 & x_3 & \cdot & \cdot & x_0 \end{bmatrix} \begin{bmatrix} r_0 & r_1 & \cdot & r_{N-1} \\ r_N & r_{N+1} & \cdot & r_{2N-1} \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & r_{LN-1} \end{bmatrix} \begin{bmatrix} p_0 \\ p_1 \\ \cdot \\ p_{N-1} \end{bmatrix}$$

$$+\, b \begin{bmatrix} 1 & 1 & 1 & \cdot & \cdot & 1 \\ 1 & 1 & 1 & \cdot & \cdot & 1 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 1 & 1 & 1 & \cdot & \cdot & 1 \end{bmatrix} \begin{bmatrix} r_0 & r_1 & \cdot & r_{N-1} \\ r_N & r_{N+1} & \cdot & r_{2N-1} \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & r_{LN-1} \end{bmatrix} \begin{bmatrix} p_0 \\ p_1 \\ \cdot \\ p_{N-1} \end{bmatrix}$$

where, $a=\{A-B\}/2$ and $b=\{A+B\}/2$ and $Y$ is an $LxL$ matrix with all elements equal to one. The constants $a$ and $b$ can merge into vector $\underline{p}$, such that $\underline{g}= a\underline{p}$ and $\underline{m}= b\underline{p}$, therefore,

$$\underline{\phi} = X \cdot R\,\underline{g} + Y \cdot R\,\underline{m} \tag{13}$$

The second term in equation (13) results into a column vector whose elements have identical values. Hence, its contribution to the correlation vector $\underline{\phi}$ is a constant dc shift. In most practical applications, the absolute value of the correlation function is of little interest and only the normalized correlation function is important. This implies that the second term of equation (13) can be dropped, resulting in,

$$\begin{bmatrix} \phi_0 \\ \phi_1 \\ \cdot \\ \cdot \\ \phi_{L-1} \end{bmatrix} = \begin{bmatrix} x_0 & x_1 & x_2 & \cdot & \cdot & x_{L-1} \\ x_{L-1} & x_0 & x_1 & \cdot & \cdot & x_{L-2} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ x_1 & x_2 & x_3 & \cdot & \cdot & x_0 \end{bmatrix} \begin{bmatrix} r_0 & r_1 & \cdot & r_{N-1} \\ r_N & r_{N+1} & \cdot & r_{2N-1} \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & r_{LN-1} \end{bmatrix} \begin{bmatrix} g_0 \\ g_1 \\ \cdot \\ g_{N-1} \end{bmatrix}$$

$$\underline{\phi} = X \cdot R\,\underline{g} \tag{15}$$

The cross-correlation operation reduced to equation (14) is the first step in the reduction of computational load. Since the elements of $X$ are all +/-1, only additions and subtractions are required to perform the matrix multiplication. Finding each element of the correlation vector $\underline{\phi}$ now requires $(M-1)$ additions and only $N$ multiplications instead of $M$. Considering that, in general, $M$ is much larger than $N$, this is a significant reduction of the computational load. It is, however, possible to reduce the computational requirement even further by exploiting the fast Hadamard transform [2].

Fast Hadamard Transform A Hadamard matrix is defined recursively as,

$$H_1 = 1 \tag{16}$$

$$H_{2i} = \begin{bmatrix} H_i & H_i \\ H_i & -H_i \end{bmatrix} \tag{17}$$

Only orders $2^k$, where $k$ is a positive integer exists. A matrix of order $8x8$ would be,

$$H_8 = \begin{bmatrix} +1 & +1 & +1 & +1 & +1 & +1 & +1 & +1 \\ +1 & -1 & +1 & -1 & +1 & -1 & +1 & -1 \\ +1 & +1 & -1 & -1 & +1 & +1 & -1 & -1 \\ +1 & -1 & -1 & +1 & +1 & -1 & -1 & +1 \\ +1 & +1 & +1 & +1 & -1 & -1 & -1 & -1 \\ +1 & -1 & +1 & -1 & -1 & +1 & -1 & +1 \\ +1 & +1 & -1 & -1 & -1 & -1 & +1 & +1 \\ +1 & -1 & -1 & +1 & -1 & +1 & +1 & -1 \end{bmatrix} \tag{18}$$

The Hadamard transform is defined as,

$$\underline{y} = H\,\underline{x} \tag{19}$$

The input vector $\underline{x}$ is transformed into vector $\underline{y}$ through the Hadamard matrix multiplication. Based on the structure of $\boldsymbol{H}$, it is possible to digitally implement this transform very efficiently. The flow graph for an 8-point fast Hadamard transform (FHT) is shown in Figure 5.

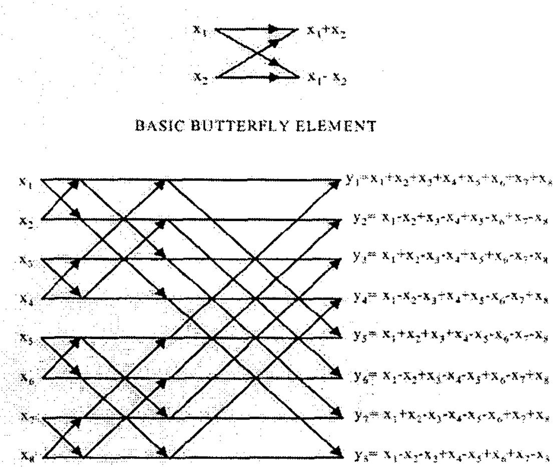


Figure 5: Flow graph for 8-point fast Hadamard transform.

This flow graph is identical to that of fast Fourier transform except that there is no multiplication involved in the present case [3].

Application of FHT to Correlator The FHT algorithm can not be directly applied to the cross-correlation operation described in equation (14) because $X$ is not a Hadamard matrix. However, it is possible to transform $X$ into the required form by the substitution,

$$X = P_2 \cdot S_2 \cdot H \cdot S_1 \cdot P_1 \tag{20}$$

where, $P_1$ and $P_2$ are permutation matrices whose purpose is to permute the rows and the columns of $H$. In general the size of $X$ is $LxL$, where $L = (2^k - 1)$, while $H$ can exist in sizes $(L+1)x(L+1)$ only. Therefore matrices $S_1$ and $S_2$ are required to transform the size of $H$ to that of $X$, by suppressing the first column and the row of matrix $H$. Equations (15) and (20) result into,

$$\underline{\phi} = (P_2 \cdot (S_2 \cdot (H \cdot (S_1 \cdot (P_1 \cdot R))))) \tag{21}$$

where the parentheses indicate the sequence of operation leading to the following interpretation. The measurement matrix $R$ is permuted according to $P_1$ and a row of zero elements is affixed to the beginning of the matrix. The resulting matrix is transformed by the fast Hadamard algorithm. Then the first row of the resulting matrix is dropped and the rest of it is permuted according to $P_2$. Finally the correlation vector is obtained by the multiplication of the previous results with the gain vector $g$. The correlation operation defined by equation (21) is now in the most appropriate form for an efficient digital implementation either in software or hardware. The calculation of each element of the correlation vector now requires about $N.log(L)$ additions and $N$ multiplications.

Fractional-Lag Correlation The set of correlation values obtained through equation (21) correspond to the correlation lags in integer multiples of a chip duration only. In certain situations it is desirable to calculate the correlation function for lag intervals smaller than the chip interval. In order to calculate the fractional-lag correlation values, a slight modification to the received vector $\underline{r}$ is required prior to its decimation into matrix $R$ according to equations (9) and (10). In particular, if the received vector $\underline{r}$ is given a left circular shift by one position, such that,

$$\underline{r}' = [r_1 \ \ r_2 \ \ r_3 \ \cdot \ \cdot \ r_M \ \ r_0] \tag{22}$$

and,

$$R' = \begin{bmatrix} r_1 & r_2 & \cdot & r_N \\ r_{N+1} & r_{N+2} & \cdot & r_{2N} \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & r_0 \end{bmatrix} \tag{23}$$

The resulting set of correlation values will correspond to the fractional-lag of *{k+1/N, where k is an integer}* of the basic chip interval. Similarly, by performing multiple circular shift operations on vector $\underline{r}$, correlation values for lags in all integral multiples of $1/N$ can be calculated.

## VI. HARDWARE IMPLEMENTATION

The proposed correlator can be implemented in hardware, either on a general purpose DSP board, or on a dedicated plug-in board based on various ASIC chips. In the later case, the digitizer can also be included on a single specialized board (Figure 6).

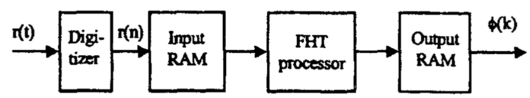


Figure 6: Single Card Correlator

The digitizer determines the size of the input RAM, and also the quantization noise level. The maximum sampling rate and the quantizer size is dictated by the specific application. However, 12 bit quantizer with 50 MHz sampling rate is a typical figure. The input RAM stores the received waveform $\underline{r}$. The decomposition of $\underline{r}$ into matrix $R$ and the permutation of vectors $R_i$ do not require any separate processing. These operations can be performed during the read or write cycle for the input RAM, without demanding any extra time.

In a similar manner, on the output side, the permutation of output vector is also performed during the data storage cycle. The size of the output RAM depends on the number of stages in the fast Hadamard transformer which in turn corresponds to the order of maximal-length sequence.

FHT Transformer The basic element of a FHT transformer is a two-input / two-output fixed point adder / subtractor (Figure 7).

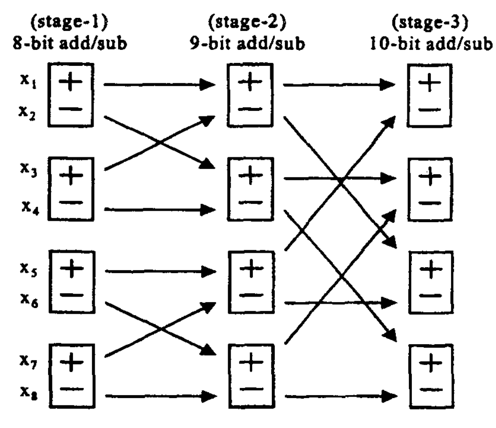


Figure 7: FHT Transformer

0-7803-6560-7/00/$10.00 (C) 2000 IEEE

A group of adders / subtractors is arranged into various stages. There are $(L+1)/2$ adders/subtractors in each stage, where $L$ is the length of the corresponding maximal-length sequence. The total number of stages is also a function of the sequence length. The size of the two inputs to the adder/subtractor depends on, 1) the quantizer size and 2) the number of stage to which the adder/subtractor belongs. In general the input data size progressively increases with increasing stage number.

## VII. SOFTWARE IMPLEMENTATION

The software implementation of correlator consists of, (a) main program, (b) FHT subroutine, and (c) permutation subroutine. The permutation subroutine (Figure 8) generates the permutation matrices $P_1$ and $P_2$. For a given maximal-length sequence, these matrices can be generated once and stored. Hence, repeated correlations corresponding to the same maximal-length sequence do not require that the permutation subroutine be called each time.

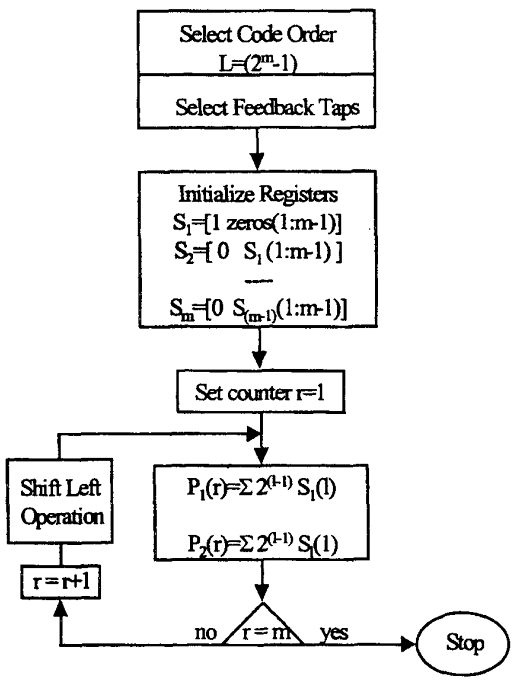


Figure 8: Permutation subroutine.

The FHT subroutine (Figure 9) is the core processing function of the correlation process. The main program (Figure 10) calls this subroutine several times for calculating a single set of correlation values. The correlator program was implemented in Matlab and its operation tested and verified [6].

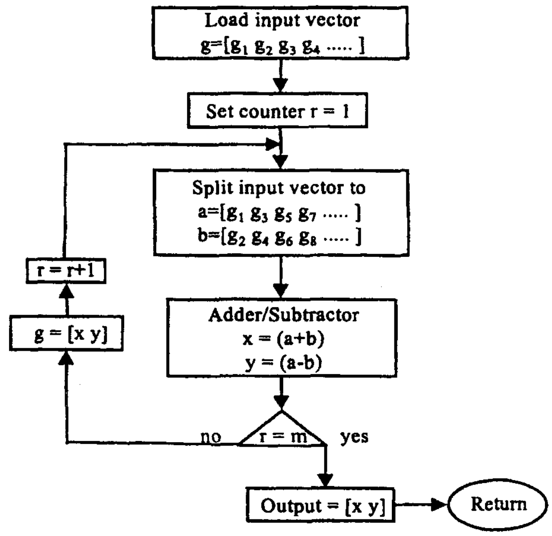


Figure 9: FHT subroutine.

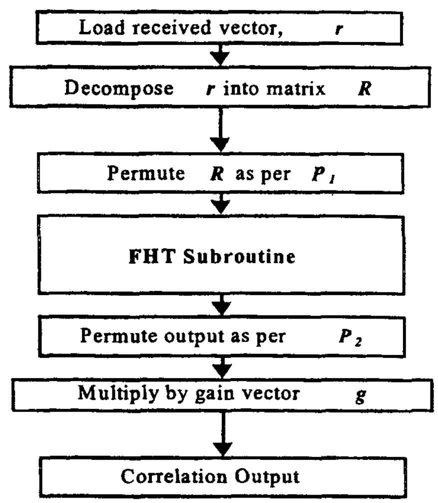


Figure 10: Main correlator program

## VIII. PERFORMANCE COMPARISON

A rough comparison of the performance of proposed correlator implementation with other standard implementations can be readily made. If $L$ is the length of the ml-sequence that constitutes the transmit waveform and there are $N$ samples per symbol. One period of the waveform will consist of $NL$ samples. Hence, the basic time-domain correlator will require $NL$ real multiplications and $NL$ real additions, in order to calculate one correlation value.

The FFT-based approach requires, (a) one FFT of length *NL*, (b) one IFFT of length *NL*, and (c) *NL* complex multiplications. These numbers, however, correspond to all the *NL* correlation values. Assuming that,

(a) one FFT requires $NL.log_2(NL)$ complex multiplications and about the same number of complex additions,
(b) one IFFT requires $NL.log_2(NL)$ complex multiplications and about the same number of complex additions,
(c) one complex multiplication requires four real multiplications and two real additions,
(d) one complex addition requires two real additions.

This gives us the average of $8xlog_2(NL)$ real multiplications and $8xlog_2(NL)$ real additions for each correlation value. The proposed approach, on the average requires only $N$ real multiplications and $N\ log_2(L)$ real additions for each correlation value. In general, $L$ is much larger compared to $N$ and therefore, the new approach is mostly dominated by additions. The following table compares the computational requirements of the proposed approach with the existing approaches, taking $L$=1,000 and $N$=10.

| Correlator Type | # of real multiplications | # of real additions |
|---|---|---|
| Basic time-domain | 10,000 | 10,000 |
| FFT-based | 110 | 108 |
| Proposed approach | 10 | 100 |

Table 1: Processing requirement per correlation value.

## IX. CONCLUSION

A new computationally efficient method of DSP-based correlator implementation is discussed. This method is applicable for pseudo-random correlation systems employing maximal-length sequence in a continuous mode. The new method exploits the structural characteristics of pseudo-random waveforms and performs the crosscorrelation of any digitized waveform with a reference pseudo-random waveform, in a manner, much more efficient in terms of processing speed and hardware requirement. This new method can be applied for baseband or bandpass waveforms, and it can handle a wide range of modulation schemes and signalling structures. In order to achieve greater resolution of the calculated correlation function, it is possible to compute the correlation function for the lag values in fraction of the basic chip interval.

## X. REFERENCES

[1] James Jordan, Peter Bishop, Bijan Kiani, *"Correlation-Based Measurement Systems"*, Ellis Horwood Limited, West Sussex, England, 1989.

[2] Mohamed A. Benkhelifa, Marcel Gindre, *"Echography Using Correlation Techniques, Choice of Coding Signal"*, IEEE Transactions on UFFC, Vol. 41, No. 5, September 1994, Page 579-586.

[3] Eric A. Lindgren, M. Rosen, *"Ultrasonic Characterization of Attenuative Materials by means of a Correlator System"*, Nondestructive Characterization of Materials IV, Plenum Press, New York, 1991.

[4] Borish, J. and Angell, J., *"An efficient algorithm for measuring the impulse response using pseudorandom noise"*, Journal Audio Engineering Society, Vol. 31, No. 7, Page 478-487, July/August 1983.

[5] Varney, M., *Hadamard versus Fourier transformation*, [1987].

[6] Jahangir K. Kayani, *"Development and Application of Spread-Spectrum Ultrasonic Evaluation Technique"*, Ph.D. Dissertation, Iowa State University, Ames, USA, 1996.