Williamson-Hadamard Transforms: Design and Fast Algorithms

Sos Agaian[1], Hakob Sarukhanyan[2], Karen Egiazarian[3] and Jaakko Astola[3]

Abstract – - **In this paper, algorithms for fast computation of a special type of Hadamard transforms, namely 4tpoint (t is an order of so-called Williamson type matrices) Williamson-Hadamard transforms, are presented. These transforms are based on Williamson's construction of Hadamard matrices. Comparisons revealing the efficiency of the proposed algorithms with respect to the known ones are given. Also, of numerical examples are presented.**

Keywords – **Hadamard matrices, Hadamard transforms, fast algorithms**

I. Introduction

In the past decade fast orthogonal transforms have been widely used in many areas, such as data compression, pattern recognition and image reconstruction, interpolation, linear filtering, spectral analysis, watermarking, cryptography and communication. The computation of unitary transforms is a complicated and time consuming task. However, it would not be possible to use the orthogonal transforms in signal and image processing applications without effective algorithms is calculate them. An important question in many applications is how to achieve the highest computation efficiency of the Then discrete orthogonal transforms (DOTs). Among DOTs a spe cial role plays a class of Hadamard transforms based on the Hadamard matrices [1,-13,19-21] and they do not require any multiplication operation in their computation.

In this paper we have utilized Williamson's construction of Hadamard matrices in order to develop efficient computational algorithms of a special type of Hadamard transforms, called Williamson-Hadamard transforms.

The paper is organized as follows. Section 2 described the Hadamard matrix construction from Williamson matrices. Sections 3 and 4 present the block representation of Williamson-Hadamard matrices and fast Williamson-Hadamard block transform algorithm. In Section 5 Williamson-Hadamard transform algorithm on add/shift architecture is developed. Section 6 and 7 present a fast Williamson-Hadamard transform algorithms based on the multiplicative theorems. Section 8 gives the complexities of developed algorithms, and also a comparative estimate, revealing the efficiency of the proposed algorithms with respect to the known once.

[1]Sos Agaian is with the University of Texas at San Antonio, San Antonio, USA, sagaian@utsa.edu

[2]Hakob Sarukhanyan is with the Institute for Informatics and Automation Problems of national academy of sciences of Armenia, Yerevan, Armenia, hakop@ipia.sci.am

[3]Karen Egiazarian and Jaakko Astola are with the Institute of Signal Processing, Tampere University of Technology, Tampere, Finland, karen@cs.tut.fi, jta@cs.tut.fi

II. How to Build Hadamard Matrices from Williamson Matrices

In this section we first give a definition of Hadamard matrices and then describe an algorithm for generation of Williamson-Hadamard matrices.

Definition 2.1: A *Hadamard matrix* H_n of order n is an orthogonal matrix consisting of elements ± 1:

$$H_n H_n^T = H_n^T H_n = nI_n,$$

where T is a transposition sign, I_n is an identity matrix of order n.

Let as briefly describe the Williamson's approach to the Hadamard matrices construction.

Theorem 2.1: (Williamson [15]) Suppose there exist four (± 1)-matrices A, B, C, D of order n satisfying

$$\begin{aligned} &PQ^T = QP^T, \quad P, Q \in \{A, B, C, D\}, \\ &AA^T + BB^T + CC^T + DD^T = 4nI_n. \end{aligned} \qquad (1)$$

Then

$$W_{4n} = \begin{pmatrix} A & B & C & D \\ -B & A & -D & C \\ -C & D & A & -B \\ -D & -C & B & A \end{pmatrix} \qquad (2)$$

is Hadamard matrix of order $4n$.

The matrices A, B, C, D with properties (1) are called *Williamson matrices*. The matrix (2), is called the *Williamson-Hadamard matrix*.

If A, B, C, D are cyclic symmetric (± 1)-matrices of order n, then the first relation of (1) is satisfied automatically and the second condition becomes

$$A^2 + B^2 + C^2 + D^2 = 4nI_n.$$

The first rows of the Williamson type cyclic and symmetric matrices A, B, C, D of order n, $n = 3, 5, ..., 25$ can be found in [2,11,16]. For the more complete list of Williamson matrices see in J. Seberry's web page [17].

Note that any cyclic symmetric matrix A can be represented as $A = \sum_{i=0}^{n-1} a_i U^i$, where U is a cyclic matrix of order n with the first row $(0, 1, ..., 0)$, and $U^{n+i} = U^i$, $a_i = a_{n-i}$, +, for $i = 1, 2, ..., n-1$.

Below we give an algorithm of the WilliamsonHadamard matrices generation.

Algorithm 2.1: **Hadamard matrix generation via cyclic symmetric Williamson matrices.**

Input: the vectors $(a_0, a_1, ..., a_{n-1})$, $(b_0, b_1, ..., b_{n-1})$, $(c_0, c_1, ..., c_{n-1})$ and $(d_0, d_1, ..., d_{n-1})$.

Step 1. Construct matrices A, B, C, D by

$$A = \sum_{i=0}^{n-1} a_i U^i, \qquad B = \sum_{i=0}^{n-1} b_i U^i,$$
$$C = \sum_{i=0}^{n-1} c_i U^i, \qquad D = \sum_{i=0}^{n-1} d_i U^i.$$

Step 2. Substitute matrices A, B, C, D into the array (2).

Output: Williamson-Hadamard matrix W_{4n}.

Example 2.1: The following matrices are the Williamson type matrices of order 3.

$$A = \begin{pmatrix} + & + & + \\ + & + & + \\ + & + & + \end{pmatrix}, \quad B = C = D = \begin{pmatrix} + & - & - \\ - & + & - \\ - & - & + \end{pmatrix}.$$

Substituting these matrices in (2) we obtain a Williamson-Hadamard matrix W_{12} of order 12

III. Block Representation of Williamson-Hadamard Matrices

In this section we present an approach of block Hadamard matrices construction equivalent to the Williamson-Hadamard matrices. This approach is useful for designing fast transforms algorithms.

We begin with an example. Let (a_0, a_1, a_1), (b_0, b_1, b_1), (c_0, c_1, c_1) and (d_0, d_1, d_1) be the first rows of Williamson type cyclic symmetric matrices of order 3. Using Algorithm 2.1, we can construct the following matrix of order 12:

$$\begin{pmatrix}
\begin{pmatrix} a_0 & a_1 & a_1 \\ a_1 & a_0 & a_1 \\ a_1 & a_1 & a_0 \end{pmatrix} & \begin{pmatrix} b_0 & b_1 & b_1 \\ b_1 & b_0 & b_1 \\ b_1 & b_1 & b_0 \end{pmatrix} & \begin{pmatrix} c_0 & c_1 & c_1 \\ c_1 & c_0 & c_1 \\ c_1 & c_1 & c_0 \end{pmatrix} & \begin{pmatrix} d_0 & d_1 & d_1 \\ d_1 & d_0 & d_1 \\ d_1 & d_1 & d_0 \end{pmatrix} \\
-\begin{pmatrix} b_0 & b_1 & b_1 \\ b_1 & b_0 & b_1 \\ b_1 & b_1 & b_0 \end{pmatrix} & \begin{pmatrix} a_0 & a_1 & a_1 \\ a_1 & a_0 & a_1 \\ a_1 & a_1 & a_0 \end{pmatrix} & -\begin{pmatrix} d_0 & d_1 & d_1 \\ d_1 & d_0 & d_1 \\ d_1 & d_1 & d_0 \end{pmatrix} & \begin{pmatrix} c_0 & c_1 & c_1 \\ c_1 & c_0 & c_1 \\ c_1 & c_1 & c_0 \end{pmatrix} \\
-\begin{pmatrix} c_0 & c_1 & c_1 \\ c_1 & c_0 & c_1 \\ c_1 & c_1 & c_0 \end{pmatrix} & \begin{pmatrix} d_0 & d_1 & d_1 \\ d_1 & d_0 & d_1 \\ d_1 & d_1 & d_0 \end{pmatrix} & \begin{pmatrix} a_0 & a_1 & a_1 \\ a_1 & a_0 & a_1 \\ a_1 & a_1 & a_0 \end{pmatrix} & -\begin{pmatrix} b_0 & b_1 & b_1 \\ b_1 & b_0 & b_1 \\ b_1 & b_1 & b_0 \end{pmatrix} \\
-\begin{pmatrix} d_0 & d_1 & d_1 \\ d_1 & d_0 & d_1 \\ d_1 & d_1 & d_0 \end{pmatrix} & -\begin{pmatrix} c_0 & c_1 & c_1 \\ c_1 & c_0 & c_1 \\ c_1 & c_1 & c_0 \end{pmatrix} & \begin{pmatrix} b_0 & b_1 & b_1 \\ b_1 & b_0 & b_1 \\ b_1 & b_1 & b_0 \end{pmatrix} & \begin{pmatrix} a_0 & a_1 & a_1 \\ a_1 & a_0 & a_1 \\ a_1 & a_1 & a_0 \end{pmatrix}
\end{pmatrix} \quad (3)$$

Now we want to use this matrix in order to make an equivalent block cyclic matrix. The first block P_0 we form as follows: a) from the first row of above matrix the first, 4-th, 7-th, and 10-th elements (a_0, b_0, c_0, d_0) and make the first row of block P_0, b) from the 4-th row of above matrix the first, 4-th, 7-th, and 10-th elements $(-b_0, a_0, -d_0, c_0)$ we make the second row of block P_0, and so on. Hence, we obtain

$$P_0 = \begin{pmatrix} a_0 & b_0 & c_0 & d_0 \\ -b_0 & a_0 & -d_0 & c_0 \\ -c_0 & d_0 & a_0 & -b_0 \\ -d_0 & -c_0 & b_0 & a_0 \end{pmatrix}. \quad (4)$$

The second (and third) block P_1 we form as follows: a) from the first row of above matrix the second, 5-th, 8-th and 11-th elements we make the first row (a_1, b_1, c_1, d_1) of block P_1, b) from the 4-th row of above matrix the second, 5-th, 8-th, and 11-th elements $(-b_1, a_1, -d_1, c_1)$ we make the second row of block P_1, and so on. Hence, we obtain

$$P_0 = \begin{pmatrix} a_1 & b_1 & c_1 & d_1 \\ -b_1 & a_1 & -d_1 & c_1 \\ -c_1 & d_1 & a_1 & -b_1 \\ -d_1 & -c_1 & b_1 & a_1 \end{pmatrix}. \quad (5)$$

From (3), (4) and (5) we obtain

$$[BW]_{12} = \begin{pmatrix} P_0 & P_1 & P_1 \\ P_1 & P_0 & P_1 \\ P_1 & P_1 & P_0 \end{pmatrix} \quad (6)$$

which is a block cyclic block symmetric Hadamard matrix.

Using the properties of the Kronecker product, we may rewrite (6) as $[BW]_{12} = P_0 \otimes I_3 + P_1 \otimes U + P_1 \otimes U^2$.

In general, any Williamson-Hadamard matrix of order $4n$ can be presented as

$$W_{4n} = \sum_{i=0}^{n-1} Q_i \otimes U^i,$$
$$Q_i(a_i, b_i, c_i, d_i) = \begin{pmatrix} a_i & b_i & c_i & d_i \\ -b_i & a_i & -d_i & c_i \\ -c_i & d_i & a_i & -b_i \\ -d_i & -c_i & b_i & a_i \end{pmatrix}, \quad (7)$$

where $Q_i = Q_{n-i}$, $a_i, b_i, c_i, d_i = \pm 1$ and $\otimes$ is a sign of the Kronecker product [11].

The Hadamard matrices of the form (7) are called *block-cyclic block-symmetric Hadamard matrices* [2]. The Williamson-Hadamard matrix W_{12}(see previous section) can be represented as a block-cyclic block-symmetric matrix:

$$[BW]_{12} = \left(\begin{array}{cccc|cccc|cccc}
+ & + & + & + & + & - & - & - & + & - & - & - \\
- & + & - & + & + & + & + & - & + & + & + & - \\
- & + & + & - & + & - & + & + & + & - & + & + \\
- & - & + & + & + & + & - & + & + & + & - & + \\
+ & - & - & - & + & + & + & + & + & - & - & - \\
+ & + & + & - & - & + & - & + & + & + & + & - \\
+ & - & + & + & - & + & + & - & + & - & + & + \\
+ & + & - & + & - & - & + & + & + & + & - & + \\
+ & - & - & - & + & - & - & - & + & + & + & + \\
+ & + & + & - & + & + & + & - & - & + & - & + \\
+ & - & + & + & + & - & + & + & - & + & + & - \\
+ & + & - & + & + & + & - & + & - & - & + & +
\end{array}\right)$$

$$= \begin{pmatrix} Q_0(+,+,+,+) & Q_4(+,-,-,-) & Q_4(+,-,-,-) \\ Q_4(+,-,-,-) & Q_0(+,+,+,+) & Q_4(+,-,-,-) \\ Q_4(+,-,-,-) & Q_4(+,-,-,-) & Q_0(+,+,+,+) \end{pmatrix}$$

From (7) we may see that all the blocks are Hadamard matrix of Williamson type of order 4. In [18] it was proved that cyclic symmetric Williamson-Hadamard block matrices can be constructed using only 5 different blocks such as

$$Q_0 = \begin{pmatrix} + & + & + & + \\ - & + & - & + \\ - & + & + & - \\ - & - & + & + \end{pmatrix}, \qquad Q_1 = \begin{pmatrix} + & + & + & - \\ - & + & + & + \\ - & - & + & - \\ + & - & + & + \end{pmatrix},$$

$$Q_2 = \begin{pmatrix} + & + & - & + \\ - & + & - & - \\ + & + & + & - \\ - & + & + & + \end{pmatrix}, \qquad Q_3 = \begin{pmatrix} + & - & + & + \\ + & + & - & + \\ - & + & + & + \\ - & - & - & + \end{pmatrix}, \qquad (8)$$

$$Q_4 = \begin{pmatrix} + & - & - & - \\ + & + & + & - \\ + & - & + & + \\ + & + & - & + \end{pmatrix}$$

For example, Williamson-Hadamard block matrix $[BW]_{12}$ was constructed using matrices Q_0 and Q_4 only.

Note that when we fix first block then one needs maximum 4 blocks to design any Williamson-Hadamard block matrix and these 4 blocks is defined uniquely up to a sign. Thus, if the first row of the first block consists of an even +1, then the first rows of the others 4 blocks consist of an odd +1. And if the first row of the first block consists of an odd +1, then the first rows of the others 4 blocks consist of an even +1. The set of blocks with fixed first block with odd +1 is following $Q'_0 = Q_0(+,+,+,-)$, $Q'_1 = Q_1(+,-,-,+)$, $Q'_2 = Q_2(-,+,-,+)$, $Q'_3 = Q_3(-,-,+,+)$, $Q'_4 = Q_4(+,+,+,+)$.

IV. Fast Block Williamson-Hadamard Transforms

In this section we describe two algorithms for forward block Williamson-Hadamard transform calculation:

$$F = [BW]_{4n} f.$$

Let us split the vector-column f into n 4-dimensional vectors as $f = \sum_{i=0}^{n-1} X_i \otimes P_i$, where P_i are column-vectors of dimension n, whose i-th element is equal to 1, and the remaining elements are equal to 0, and $X_i = (f_{4i}, f_{4i+1}, f_{4i+2}, f_{4i+3})$, $i = 0, 1, ..., n-1$.

Now, using (7), we have

$$\begin{aligned} [BW]_{4n} f &= \Big(\sum_{i=0}^{n-1} Q_i \otimes U^i\Big)\Big(\sum_{j=0}^{n-1} X_j \otimes P_j\Big) \\ &= \sum_{i,j=0}^{n-1} Q_i X_j \otimes U^i P_j. \end{aligned} \qquad (9)$$

We may cheek $U^i P_j = P_{n-i+j}$, $j = 0, 1, ..., n-1$, $i = j+1, ..., n-1$.

Hence, the equation (9) can be presented as

$$[BW]_{4n} f = \sum_{i,j}^{n-1} Q_i X_i \otimes U^i P_j = \sum_{j=0}^{n-1} B_j, \qquad (10)$$

where $B_j = \sum_{i=0}^{n-1} Q_j X_j \otimes U^i P_j$.

From (10) we have that for performing the fast Williamson-Hadamard transform we need to calculate the spectral coefficients of the block transforms, such as $Y_i = Q_i X$. Here Q_i, $i = 0, 1, 2, 3, 4$ have the form (8), and $X = (x_0, x_1, x_2, x_3)^T$, and $Y_i = (y_i^0, y_i^1, y_i^2, y_i^3)^T$ are the input and output column-vectors, respectively.

The flow graph of the algorithm for joint computation of five 4-point Williamson-Hadamard transforms $Q_i X$, $i = 0, 1, ..., 4$ is given in Figure 1.

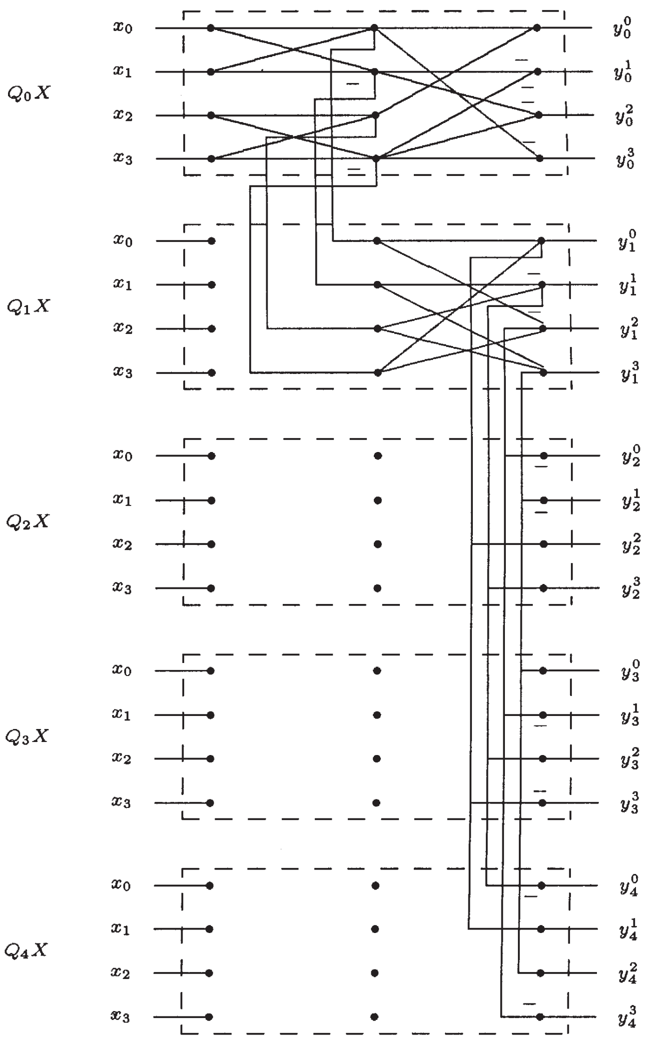


Fig. 1. Flow graph for the joint $Q_i X$ transforms, $i = 0, 1, ..., 4$

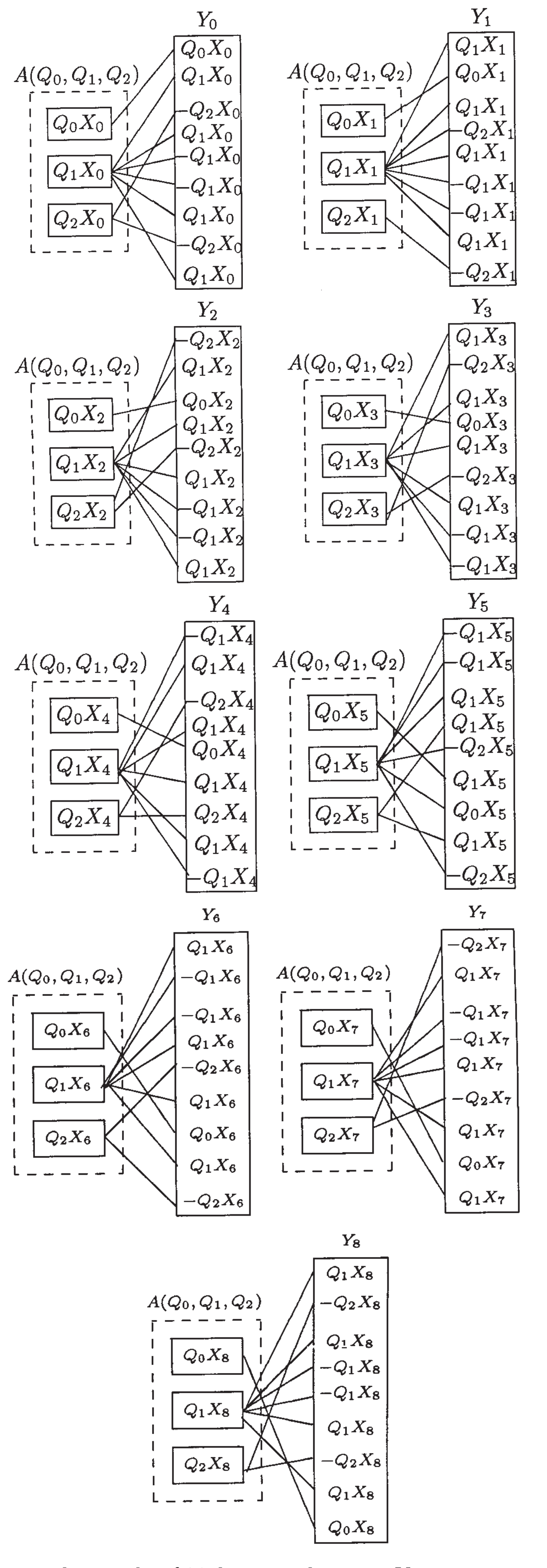
The joint computation of 4-point transforms $Q_i X$, $i = 0, 1, ..., 4$ requires only 12 addition/subtraction operations. Note that the separate calculation of Q_j, $j = 0, 1, ..., 4$ requires 40 addition/subtraction operations.

Really, from Figure 1 we can check that the transform $Q_0 X$ requires 8 addition/subtraction operations, and the transform $Q_1 X$ requires 4 addition/subtraction operations. We can see also that the joint computation of all 4-point transforms $Q_i X$, $i = 0, 1, ..., 4$ requires only 12 addition/subtraction operations.

Now we give a detailed description of 36-point block Williamson-Hadamard fast transform algorithm.

Example 4.1: **36-point fast Williamson-Hadamard transform using 396 operations.**

Input : vector-column $F_{36} = (f_i)_{i=0}^{35}$ and blocks Q_0, Q_1 and Q_2



Y_0
$A(Q_0,Q_1,Q_2)$
Q_0X_0
Q_1X_0
Q_2X_0
Q_0X_0 Q_1X_0 $-Q_2X_0$ Q_1X_0 $-Q_1X_0$ $-Q_1X_0$ Q_1X_0 $-Q_2X_0$ Q_1X_0
Y_1
Q_1X_1 Q_0X_1 Q_1X_1 $-Q_2X_1$ Q_1X_1 $-Q_1X_1$ $-Q_1X_1$ Q_1X_1 $-Q_2X_1$
Y_2
$-Q_2X_2$ Q_1X_2 Q_0X_2 Q_1X_2 $-Q_2X_2$ Q_1X_2 $-Q_1X_2$ $-Q_1X_2$ Q_1X_2
Y_3
Q_1X_3 $-Q_2X_3$ Q_1X_3 Q_0X_3 Q_1X_3 $-Q_2X_3$ Q_1X_3 $-Q_1X_3$ $-Q_1X_3$
Y_4
$-Q_1X_4$ Q_1X_4 $-Q_2X_4$ Q_1X_4 Q_0X_4 Q_1X_4 Q_2X_4 Q_1X_4 $-Q_1X_4$
Y_5
$-Q_1X_5$ $-Q_1X_5$ Q_1X_5 Q_1X_5 $-Q_2X_5$ Q_1X_5 Q_0X_5 Q_1X_5 $-Q_2X_5$
Y_6
Q_1X_6 $-Q_1X_6$ $-Q_1X_6$ Q_1X_6 $-Q_2X_6$ Q_1X_6 Q_0X_6 Q_1X_6 $-Q_2X_6$
Y_7
$-Q_2X_7$ Q_1X_7 $-Q_1X_7$ $-Q_1X_7$ Q_1X_7 $-Q_2X_7$ Q_1X_7 Q_0X_7 Q_1X_7
Y_8
Q_1X_8 $-Q_2X_8$ Q_1X_8 $-Q_1X_8$ $-Q_1X_8$ Q_1X_8 $-Q_2X_8$ Q_1X_8 Q_0X_8

Fig. 2. Flow graphs of 36-dimensional v ectors Y_i, $i = 0, 1, ..., 8$ computation

Step 1. Split vector F_{36} into 9 parts as $F_{36}^T = (X_0^T, X_1^T, ..., X_8^T)$, where $X_i^T = (f_{4i}, f_{4i+1}, f_{4i+2}, f_{4i+3})$, $i = 0, 1, ..., 8$.

Step 2. Compute the vectors Y_i, $i = 0, 1, ..., 8$, as shown in Figure 2.

Note that the subblocks $A(Q_0, Q_1, Q_2)$in Figure 2 can be computed using the scheme in Figure 1.

Step 3. Evaluate the vector $Y = Y_0 + Y_1 + \cdots + Y_8$.

Output: transform coefficients, i.e. vector Y.

As we have seen from the fast algorithm of 4-point Williamson-Hadamard transforms (see Figure 1), the joint computation of the transforms Q_0X_i, Q_1X_i and Q_2X_i requires only 12 addition/subtraction operations. From (8) we can see that only these transforms are there in each vector Y_i. Hence, for all these vectors it is necessary to perform 108 operations. Finally, the 36-point Hadamard transform requires only 396 addition/subtraction operations, but in the case of direct computation it would require 1260 addition/subtraction operations.

Note that we have developed a fast Walsh-Hadamard transform algorithm without knowing the existence of Williamson-Hadamard matrices. We can speed up this algorithm if we would know a construction of these matrices. The first block-rows of the block-cyclic block-symmetric (BCBS) Hadamard matrices of Williamson type of order $4n$, $n = 3, 5, ..., 25$ can be found in [2,16].

V. Williamson-Hadamard Transform on Add/Shift Architectures

In this section we describe add/shift for Williamson-Hadamard transform.

Denoting by $z_1 = x_1 + x_2 + x_3$, $z_2 = z_1 - x_0$, and using (8), we can calculate $Y_i = (y_i^0, y_i^1, y_i^2, y_i^3) = Q_iX$ as:

$$y_0^0 = z_1 + x_0, \qquad y_0^1 = z_2 - 2x_2,$$
$$y_0^2 = z_2 - 2x_3, \qquad y_0^3 = z_2 - 2x_1;$$

$$y_1^0 = y_0^0 - 2x_3, \qquad y_1^1 = z_2,$$
$$y_1^2 = y_0^2 - 2x_1, \qquad y_1^3 = y_0^0 - 2x_1;$$

$$y_2^0 = -y_1^2, \qquad y_2^1 = -y_1^3,$$
$$y_2^2 = y_1^0, \qquad y_2^3 = y_1^1;$$

$$y_3^0 = y_1^3, \qquad y_3^1 = -y_1^2,$$
$$y_3^2 = z_2 = -y_1^1, \qquad y_3^3 = -y_1^0;$$

$$y_4^0 = -z_2 = -y_1^1, \qquad y_4^1 = y_1^0,$$
$$y_4^2 = y_1^3, \qquad y_4^3 = -y_1^2.$$

It is easy to check that the joint 4-point transforms computation requires less operations than their separate computations. The separate computations of transforms Q_0X and Q_1X require 14 addition/subtraction operations and 6 one-bit shifts, but for their joint computation it is necessary only 10 addition/subtraction operations and 3 one-bit shifts.

So, using this fact, the complexity of the fast Williamson-Hadamard transform will be discussed next.

VI. Multiplicative Theorem Based Williamson-Hadamard Matrices

In this section we describe of Williamson-Hadamard matrices constructions based on the following multiplicative theorem.

Theorem 6.1: (Multiplicative Theorem [14]) Let there exist Williamson-Hadamard matrices of orders $4m$ and $4n$. Then there exist Williamson-Hadamard matrices of order $4(2m)^i n, i = 1, 2, ...$

Theorem 6.2: Let there exist Williamson matrices of order n and Hadamard matrix of order $4m$. Then there exists a Hadamard matrix of order $8mn$.

The prove of Theorems 6.1 and 6.2 is presented in Appendix.

Algorithm 6.1: **Generation of Williamson-Hadamard matrix of order $8mn$ from Williamson-Hadamard matrices of orders $4m$ and $4n$.**

Input : Williamson matrices A, B, C, D and A_0, B_0, C_0, D_0 of orders m and n, respectively.

Step 1 . Construct matrices X and Y as

$$X = \frac{1}{2}\begin{pmatrix} A+B & C+D \\ C+D & -A-B \end{pmatrix}, \quad Y = \frac{1}{2}\begin{pmatrix} A-B & C-D \\ -C+D & A-B \end{pmatrix}. \tag{11}$$

Step 2 . For $i = 1, 2, ..., k$ construct recursively the following matrices

$$\begin{aligned} A_i &= A_{i-1} \otimes X + B_{i-1} \otimes Y, \\ B_i &= B_{i-1} \otimes X - A_{i-1} \otimes Y, \\ C_i &= C_{i-1} \otimes X + D_{i-1} \otimes Y, \\ D_i &= D_{i-1} \otimes X - C_{i-1} \otimes Y. \end{aligned} \tag{12}$$

Step 3 . For $i = 1, 2, ..., k$ construct the Williamson-Hadamard matrix as

$$[WH]_i = \begin{pmatrix} A_i & B_i & C_i & D_i \\ -B_i & A_i & -D_i & C_i \\ -C_i & D_i & A_i & -B_i \\ -D_i & -C_i & B_i & A_i \end{pmatrix}$$

Output : Williamson-Hadamard matrices $[WH]_i$, $i = 1, 2, ..., k$.

Example 6.1: **Construction of Williamson matrices**.

Using Williamson matrices of order 3 and 5 from the Example 2.1 and (11), we obtain

a) for $n = 3$:

$$X = \begin{pmatrix} + & 0 & 0 & + & - & - \\ 0 & + & 0 & - & + & - \\ 0 & 0 & + & - & - & + \\ + & - & - & - & 0 & 0 \\ - & + & - & 0 & - & 0 \\ - & - & + & 0 & 0 & - \end{pmatrix}, \quad Y = \begin{pmatrix} 0 & + & + & 0 & 0 & 0 \\ + & 0 & + & 0 & 0 & 0 \\ + & + & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & + & + \\ 0 & 0 & 0 & + & 0 & + \\ 0 & 0 & 0 & + & + & 0 \end{pmatrix}.$$

b) for n=5:

$$X = \begin{pmatrix} + & - & - & - & - & + & 0 & 0 & 0 & 0 \\ - & + & - & - & - & 0 & + & 0 & 0 & 0 \\ - & - & + & - & - & 0 & 0 & + & 0 & 0 \\ - & - & - & + & - & 0 & 0 & 0 & + & 0 \\ - & - & - & - & + & 0 & 0 & 0 & 0 & + \\ + & 0 & 0 & 0 & 0 & - & + & + & + & + \\ 0 & + & 0 & 0 & 0 & + & - & + & + & + \\ 0 & 0 & + & 0 & 0 & + & + & - & + & + \\ 0 & 0 & 0 & + & 0 & + & + & + & - & + \\ 0 & 0 & 0 & 0 & + & + & + & + & + & - \end{pmatrix}$$

$$Y = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & + & - & - & + \\ 0 & 0 & 0 & 0 & 0 & + & 0 & + & - & - \\ 0 & 0 & 0 & 0 & 0 & - & + & 0 & + & - \\ 0 & 0 & 0 & 0 & 0 & - & - & + & 0 & + \\ 0 & 0 & 0 & 0 & 0 & + & - & - & + & 0 \\ 0 & - & + & + & - & 0 & 0 & 0 & 0 & 0 \\ - & 0 & - & + & + & 0 & 0 & 0 & 0 & 0 \\ + & - & 0 & - & + & 0 & 0 & 0 & 0 & 0 \\ + & + & - & 0 & - & 0 & 0 & 0 & 0 & 0 \\ - & + & + & - & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

Now let $A_0 = (1)$, $B_0 = (1)$, $C_0 = (1)$, $D_0 = (1)$ and $A = (+ + +)$, $B = C = D = (+ - -)$ be cyclic symmetric matrices of order 1 and 3, respectively. Then, from the equation (12), matrix we obtain Williamson matrices of order 6, i.e.

$$A_1 = A_3 = \begin{pmatrix} A & C \\ D & -B \end{pmatrix}, \quad A_2 = A_4 = \begin{pmatrix} B & D \\ C & -A \end{pmatrix},$$

$$A_1 = A_3 = \begin{pmatrix} + & + & + & + & - & - \\ + & + & + & - & + & - \\ + & + & + & + & - & + \\ + & - & - & - & + & + \\ - & + & - & + & - & + \\ - & - & + & + & + & - \end{pmatrix},$$

$$A_2 = A_4 = \begin{pmatrix} + & - & - & + & - & - \\ - & + & - & - & + & - \\ - & - & + & - & - & + \\ + & - & - & - & - & - \\ - & + & - & - & - & - \\ - & - & + & - & - & - \end{pmatrix}.$$

Let $A_0 = (1)$, $B_0 = (1)$, $C_0 = (1)$, $D_0 = (1)$ and $A = B = (+ - - - -)$, $C = (+ + - - +)$, $D = (+ - + + -)$ be cyclic symmetric matrices of order 1 and 5, respectively. Then, from the equation (12), we obtain Williamson matrices of order 10, i.e.

$$A_1 = A_3 = \begin{pmatrix} + & - & - & - & - & + & + & - & - & + \\ - & + & - & - & - & + & + & + & - & - \\ - & - & + & - & - & - & + & + & + & - \\ - & - & - & + & - & - & - & + & + & + \\ - & - & - & - & + & + & - & - & + & + \\ + & - & + & + & - & - & + & + & + & + \\ - & + & - & + & + & + & - & + & + & + \\ + & - & + & - & + & + & + & - & + & + \\ + & + & - & + & - & + & + & + & - & + \\ - & + & + & - & + & + & + & + & + & - \end{pmatrix},$$

$$A_2 = A_4 = \begin{pmatrix} + & - & - & - & - & + & - & + & + & - \\ - & + & - & - & - & - & + & - & + & + \\ - & - & + & - & - & + & - & + & - & + \\ + & - & - & + & - & + & + & - & + & - \\ - & + & - & - & + & - & + & + & - & + \\ + & + & - & - & + & - & + & + & + & + \\ + & + & + & - & - & + & - & + & + & + \\ - & + & + & + & - & + & + & - & + & + \\ - & - & + & + & + & + & + & + & - & + \\ + & - & - & + & + & + & + & + & + & - \end{pmatrix}.$$

VII. Multiplicative Theorem Based Fast Williamson-Hadamard Transforms

In this section we present a fast transform algorithm based in Theorems 6.1 and 6.2. First we give an algorithm of generation of Hadamard matrix based on Theorem 6.2.

Algorithm 7.1: **Generation of Hadamard matrix via Theorem 6.2.**

Input : Williamson matrices A, B, C and D of order n and Hadamard matrix H_1 of order $4m$.

Step 1 . Construct the matrices X and Y according to (11).

Step 2 . Construct Hadamard matrix as

$$P = X \otimes H_1 + Y \otimes S_{4m} H_1, \qquad (13)$$

where S_{4m} is a monomial matrix with conditions: $S_{4m}^T = -S_{4m}$, $S_{4m} S_{4m}^T = I_{4m}$.

Output Hadamard matrix P.

Example of a monomial matrix of order 8 is given below

$$\begin{pmatrix} 0 & + & 0 & 0 & 0 & 0 & 0 & 0 \\ - & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & + & 0 & 0 & 0 & 0 \\ 0 & 0 & - & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & + & 0 & 0 \\ 0 & 0 & 0 & 0 & - & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & + \\ 0 & 0 & 0 & 0 & 0 & 0 & - & 0 \end{pmatrix}.$$

Algorithm 7.2: **Fast transform with matrix (13).**

Input vector-column $F^T = (f_1, f_2, ..., f_{8mn})$, and Hadamard matrix P from (13).

Step 1 . Perform P as

$$P = (X \otimes I_{4m} + Y \otimes S_{4m})(I_{2m} \otimes H_1). \qquad (14)$$

Step 2 . Split vector F as $F = (F_1, F_2, ..., F_{2n})$, where $F_j = (f_{4m(j-1)+1}, f_{4m(j-1)+2}, ..., f_{4m(j-1)+4m})$.

Step 3 . Compute the transform $Q_i = H_1 F_i$, $i = 1, 2, ..., 2n$.

Step 4 . Split vector $Q = (Q_1, Q_2, ..., Q_{2n})$ into $4m$ $2n$-dimensional vectors as $Q = (P_1, P_2, ..., P_{4m})$, where

$$P_j = (f_{2n(j-1)+1}, f_{2n(j-1)+2}, ..., f_{2n(j-1)+2n}).$$

Step 5 . Compute the transforms XP_j and YP_j.

Output : transform coefficients.

Let us give an example of computation of transforms XF and YF ($F = (f_1, f_2, ..., f_6)$), where A, B, C, D are Williamson matrices of order 3, and X and Y from the Example 2.1. First we compute

$$XF = \begin{pmatrix} f_1 + f_4 - (f_5 + f_6) \\ f_2 + f_5 - (f_4 + f_6) \\ f_3 + f_6 - (f_4 + f_5) \\ f_1 - f_4 - (f_2 + f_3) \\ f_2 - f_5 - (f_1 + f_3) \\ f_3 - f_6 - (f_1 + f_2) \end{pmatrix}, \quad YF = \begin{pmatrix} f_2 + f_3 \\ f_1 + f_3 \\ f_1 + f_2 \\ f_5 + f_6 \\ f_4 + f_6 \\ f_4 + f_5 \end{pmatrix} \qquad (15)$$

From (15) it follows that the joint computation of XF and YF requires only 18 addition/subtruction (see Figure 3).

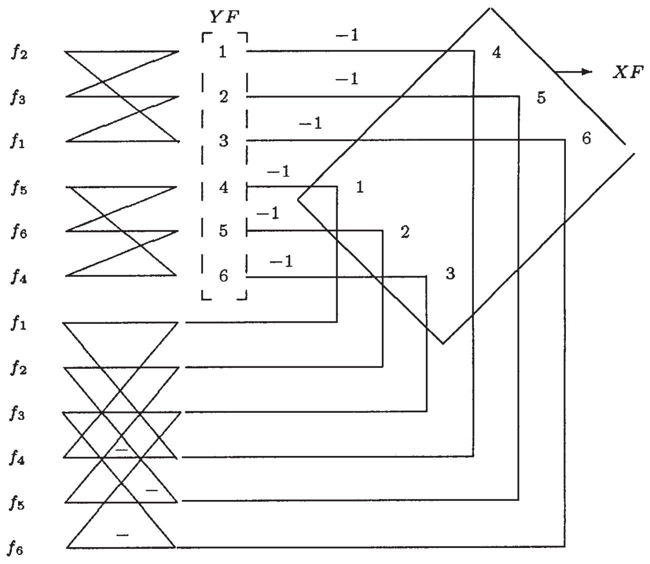


Fig. 3. Flow graph of joint computation of XF and YF

Then, from (14), we can conclude that the complexity of PF transform algorithm can be obtained by

$$C(24m) = 48m(2m + 1).$$

Note, that if X, Y are matrices of order k defined by (11), H_m is an Hadamard matrix of order m, and S_m is a monomial matrix of order m, then for any integer n

$$H_{mk^n} = X \otimes H_{mk^{n-1}} + Y \otimes S_{mk^{n-1}} H_{mk^{n-1}} \qquad (16)$$

is an Hadamard matrix of order mk^n.

Remark 7.1: For $A = B = C = D = (1)$ from (11) we have $X = \begin{pmatrix} + & + \\ + & - \end{pmatrix}$, $Y = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$. And if $H_2 = \begin{pmatrix} + & + \\ + & - \end{pmatrix}$, then the matrix (16) is the Walsh-Hadamard matrix of order $2n + 1$ [1].

Algorithm 7.3: **Construction of Hadamard matrices of order** $m(2n)^k$.

Input : Williamson matrices A, B, C, D of order n and Hadamard matrix of order m.

Step 1 . Construct matrices X and Y according to (11).

Step 2 . Construct the matrix $H_{2nm} = X \otimes H_m + Y \otimes S_m H_m$.

Step 3 . If $i < k$, then $i \leftarrow i+1$, $H_{m(2n)^i} \leftarrow H_{m(2n)^{i+1}}$, $S_{m(2n)^i} \leftarrow S_{m(2n)^{i+1}}$, and go to the **Step 2**.

Output : Hadamard matrix $H_{m(2n)^k}$.

Let us represent a matrix H_{mk^n} as a product of sparse matrices.

$$\begin{aligned} H_{mk^n} &= (X \otimes I_{mk^{n-1}} + Y \otimes S_{mk^{n-1}})(I_k \otimes H_{mk^{n-1}}) \\ &= A_1(I_k \otimes H_{mk^{n-1}}). \end{aligned}$$

Continue this factorization process for all matrices $H_{mk^{n-i}}$, $i = 1, 2, ..., n$, we obtain

$$H_{mk^n} = A_1 A_2 \cdots A_n (I_{k^n} \otimes H_m), \tag{17}$$

where

$$A_i = I_{k^{i-1}} \otimes (X \otimes I_{mk^{n-i}} + Y \otimes S_{mk^{n-i}}), \quad i = 1, 2, ..., n.$$

Example 7.1: Let H_m be a Hadamard matrix of order m, and X and Y have the form as in Example 2.1, and let $F = (f_i)_{i=1}^{6m}$ be an input vector. Then we have a Hadamard matrix of order $6m$ of the form

$$H_{6m} = X \otimes H_m + Y \otimes S_m H_m.$$

Like in (17), we have $H_{6m} = A_1(I_6 \otimes H_m)$, where $A_1 = X \otimes I_m + Y \otimes S_m$, and

$$X \times I_m = \begin{pmatrix} I_m & 0 & 0 & I_m & -I_m & -I_m \\ 0 & I_m & 0 & -I_m & I_m & -I_m \\ 0 & 0 & I_m & -I_m & -I_m & I_m \\ I_m & -I_m & -I_m & -I_m & 0 & 0 \\ -I_m & I_m & -I_m & 0 & -I_m & 0 \\ -I_m & -I_m & I_m & 0 & 0 & -I_m \end{pmatrix}$$

$$Y \otimes S_m = \begin{pmatrix} 0 & S_m & S_m & 0 & 0 & 0 \\ S_m & 0 & S_m & 0 & 0 & 0 \\ S_m & S_m & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & S_m & S_m \\ 0 & 0 & 0 & S_m & 0 & S_m \\ 0 & 0 & 0 & S_m & S_m & 0 \end{pmatrix} \tag{18}$$

The input column-vector is represented as $F = (F_1, F_2, ..., F_6)$, where F_i is an m-dimensional vector.

Now we estimate the complexity of transforms

$$\begin{aligned} H_{6m}F &= A_1(I_6 \otimes H_m)F \\ &= A_1 \cdot \text{diag}\{H_m F_1, H_m F_2, ..., H_m F_6\}. \end{aligned} \tag{19}$$

Denote $T = (I_6 \otimes H_m)F$. Computing $A_1 T$, where $T = (T_1, ..., T_6)$, from (18) we obtain

$$(X \otimes I_m)T = \begin{pmatrix} T_1 + T_4 - (T_5 + T_6) \\ T_2 + T_5 - (T_4 + T_6) \\ T_3 + T_6 - (T_4 + T_5) \\ T_1 - T_4 - (T_2 + T_3) \\ T_2 - T_5 - (T_1 + T_3) \\ T_3 - T_6 - (T_1 + T_2) \end{pmatrix},$$

$$(Y \otimes S_m)T = \begin{pmatrix} S_m(T_2 + T_3) \\ S_m(T_1 + T_3) \\ S_m(T_1 + T_2) \\ S_m(T_5 + T_6) \\ S_m(T_4 + T_6) \\ S_m(T_4 + T_5) \end{pmatrix} \tag{20}$$

From (19) and (20) it follows that the computational complexity of transform H_{6mF} is

$$C(H_{6m}) = 24m + 6C(H_m),$$

where $C(H_m)$is a complexity of an m-point Hadamard transform.

VIII. Complexity and Comparison

A. *Complexity of block-cyclic block-symmetric Williamson-Hadamard Transform*

Since every block-row of block-cyclic block-symmetric Hadamard matrix contains block Q_0 and other blocks are from a set $\{Q_1, Q_2, Q_3, Q_4\}$ (see Appendix), it is not difficult to find that the complexity of the block Williamson-Hadamard transform of order $4n$ can be obtained from the following formula

$$C(H_{4n}) = 4n(n+2).$$

From representation of a block Williamson-Hadamard matrix we can see that some of block pairs are repeated.

Two block sequences of length k $(k < n)$ in the first block row of block Williamson-Hadamard matrix of order $4n$ $(-1)^{p_1}Q_i, (-1)^{p_2}Q_i, ..., (-1)^{p_k}Q_i$ and $(-1)^{q_1}Q_j, (-1)^{q_2}Q_j, ..., (-1)^{q_k}Q_j$ $(i, j = 0, 1, 2, 3, 4)$, where $p_t, q_t \in \{0, 1\}$ and for all $t = 1, 2, ..., k$ $q_t = p_t$, or $q_t = \bar{p}_t$ $(\bar{1} = 0, \bar{0} = 1)$, we call *cyclic congruent circuits* if $dist[(-1)^{p_t}Q_i, (-1)^{p_{t+1}}Q_i] = dist[(-1)^{q_t}Q_j, (-1)^{q_{t+1}}Q_j]$ for all $t = 1, 2, ..., k-1$, where $dist[A_i, A_j] = j - i$, for $A = (A_i)_{i=1}^{m}$.

For example, in first block row of the block-cyclic block-symmetric Hadamard matrix of order 36 there are 3 cyclic congruent circuits of length 2. These circuits are underlined as

$$Q_0, \underline{Q_1}, \underline{\underline{-Q_2}}, \underline{\underline{\underline{Q_1}}}, -Q_1; -Q_1, \underline{Q_1}, \underline{\underline{-Q_2}}, \underline{\underline{\underline{Q_1}}}.$$

With this observation, one can reduce some operations in summing up the vectors Y_i (see Step 3 above example and corresponding ?ow graphs).

Let m be a length of the cyclic congruent circuits into block-row of the block-cyclic block-symmetric Hadamard matrix of order $4n$, t_m be a number of various cyclic congruent circuits of length m, $N_{m,j}$ be a number of cyclic congruent circuits of type j and length m. Then the complexity of the Hadamard transform of order $4n$ takes the form

$$C_r(H_{4n}) = 4n\Big(n + 2 - 2\sum_{i=2}^{m}\sum_{j=1}^{t_m}(N_{m,j} - 1)(i-1)\Big). \tag{21}$$

The values of parameters n, m, t_m, $N_{m,j}$ and the complexity of the Williamson type Hadamard transform of order $4n$ are given in the following table 1.

Thus, the complexity of the block Williamson-Hadamard transform can be calculated from the formula

$$C^{\pm} = 2n(2n+3), \qquad C^{sh} = 3n,$$

where $C^{\pm}$ is a number of addition/subtractions, and C^{sh} is a number of shifts.

Table 1.

No.	4n	m	t_m	$N_{m,j}$	$C_r(H_{4n})$	direct comp.
1	12				60	132
2	20				140	380
3	28	2	1	2	224	756
4	36	2	1	3	324	1260
5	44	2	1	2	528	1892
6	52	3	1	2	676	2652
7	60	2	3	3, 2, 2	780	3540
8	68	2	2	2, 3	1088	4558
9	76	2	3	2, 4, 3	1140	5700
10	84	2	3	2, 2, 5	1428	6972
11	92	2	3	4, 2, 2	1840	8372
12	100	2	3	2, 7, 2	1850	9900

Now, using repetitions of additions of vectors Y_i and the same notations as in the previous subsection (see equation (21)), the complexity of Williamson-Hadamard transform can be presented as

$$C_r^{\pm} = 2n\Big(2n + 3 - 2\sum_{i=2}^{m}\sum_{j=1}^{t_m}(N_{m,j} - 1)(i-1)\Big),$$
$$C^{sh} = 3n.$$

Formulas of complexities of the fast Williamson-Hadamard transforms without repetitions of blocks and with repetitions and shifts, and their numerical results are given in formula (22) and in Table 2, respectively,

$$\begin{aligned} C &= 4n(n+2),\\ C_r &= 4n\Big(n + 2 - 2\sum_{i=2}^{m}\sum_{j=1}^{t_m}(N_{m,j} - 1)(i-1)\Big),\\ C^{\pm} &= 2n(2n+3),\\ C^{sh} &= 3n,\\ C_r^{\pm} &= 2n\Big(2n + 3 - 2\sum_{i=2}^{m}\sum_{j=1}^{t_m}(N_{m,j} - 1)(i-1)\Big),\\ C^{sh} &= 3n. \end{aligned} \tag{22}$$

Table 2.

No.	4n	C	$C^{\pm}$	C^{sh}	C_r	$C_r^{\pm}$	direct comput.
1	12	60	54	9	60	54	132
2	20	140	130	15	140	130	380
3	28	252	238	21	224	210	756
4	36	396	378	27	324	306	1260
5	44	572	550	33	528	506	1892
6	52	780	754	39	676	650	2652
7	60	1020	990	45	780	750	3540
8	68	1292	1258	51	1088	1054	4558
9	68	1292	1258	51	1020	986	4558
10	76	1596	1558	57	1140	1102	5700
11	84	1932	1890	63	1428	1386	6972
12	92	2300	2254	69	1840	1794	8372
13	100	2700	2650	75	1900	1850	9900

B. *Complexity of Hadamard transform from multiplicative theorem*

Recall that if X, Y are matrices from (11) of order k and H_m is an Hadamard matrix of order m, then the Hadamard matrix constructed recursively

$$H_{mk^n} = X \otimes H_{mk^{n-1}} + Y \otimes S_{mk^{n-1}} H_{mk^{n-1}},$$

can be factorized as

$$H_{mk^n} = A_1 A_2 \cdots A_n (I_{k^n} \otimes H_m),$$

where

$$A_i = I_{k^{i-1}} \otimes (X \otimes I_{mk^{n-i}} + Y \otimes S_{mk^{n-i}}). \tag{23}$$

Let us now evaluate the complexity of a transform

$$H_{mk^n} F, \qquad F^T = (f_1, f_2, ..., f_{mk^n}).$$

First, we find the required operations for transform $A_i P$, $P^T = (p_1, p_2, ..., p_{mk^n})$.

Represent $P^T = (P_1, P_2, ..., P_{k^{i-1}})$, where

$$P_j = \Big((j-1)mk^{n-i+1} + t\Big)_{t=1}^{mk^{n-i+1}},$$

and $j = 1, 2, ..., k^{i-1}$.

Then, from (23), we have

$$\begin{aligned} A_i P = \text{diag}\{&(X \otimes I_{mk^{n-i}} + Y \otimes S_{mk^{n-i}})P_1, ...,\\ &..., (X \otimes I_{mk^{n-i}} + Y \otimes S_{mk^{n-i}})P_{k^{i-1}}\}. \end{aligned} \tag{24}$$

Denote the complexities of transforms XQ and YQ by C_X and C_Y, respectively. We have

$$C_X < k(k-1), \qquad C_Y < k(k-1).$$

From (24) we obtain the complexity of transform $\prod_{i=1}^{n} A_i P$, $(C_X + C_Y + k)mnk^{n-1}$.

Table 3.

H_m	Complexity
$H_m = X = H_2$ (see Remark 7.1)	$(n+1)2^{n+1}$
Walsh-Hadamard	$(C_X + C_Y + k)mnk^{n-1} + k^n m \log_2 m$.
BCBS Williamson-Hadamard: a) with block repetition	$(C_X + C_Y + k)mnk^{n-1} + k^n m (\frac{m}{4} + 2)$
b) with block repetition and congruent circuits	$(C_X + C_Y + k)mnk^{n-1} + k^n m (\frac{m}{4} + 2 - 2\sum_{i=2}^{r}\sum_{j=1}^{t_r}(N_{r,j} - 1)(i-1))$
c) with block repetition	$(C_X + C_Y + k)mnk^{n-1} + k^n \frac{m}{2}(\frac{m}{2} + 3)$
and shifts	$k^n \frac{3m}{4}$
d) with block repetition and congruent circuits,	$(C_X + C_Y + k)mnk^{n-1} + k^n \frac{m}{2}(\frac{m}{2} + 3 - 2\sum_{i=2}^{r}\sum_{j=1}^{t_r}(N_{r,j} - 1)(i-1))$
and shifts	$k^n \frac{3m}{4}$

Hence, the total complexity of transform $H_{mk^n}F$ is

$$C(H_{mk^n}) < (C_X + C_Y + k)mnk^{n-1} + k^n C(H_m),$$

where $C(H_m)$is a complexity of m-point Hadamard transform.

For a given matrices X and Y we can compute the exact value of C_X, C_Y, and therefore we can obtain the exact complexity of the transform. For example, for $k = 6$, from (15) we see that $C_X + C_Y = 18$, hence $m6^n$-point Hadamard transform requires only $24mn6^{n-1} + 6nC(H_m)$ operations.

IX. Conclusion

Three new efficient algorithms of $4t$-point (t is a 'arbitrary' integer number, for which there is a construction of a Hadamard matrix) Williamson-Hadamard transforms computation are developed. The design algorithms are based on block representation of Williamson-Hadamard matrices, on multiplicative theorem, and on iterative constructions.

Using the structures of the existing Williamson matrices the computational complexity of the developed algorithm is greatly reduced. Williamson-Hadamard transform algorithms on add/shift architectures are also described. The complexity of developed algorithms are demonstrated. Comparative estimates revealing the efficiency of the proposed algorithms with respect to the known ones are given. The results of numerical examples were presented.

Appendix

Proof of Theorem 6.1 . Let A, B, C, D be Williamson matrices of order m. Introduce $(0, \pm 1)$-matrices X, Y of order $2n$

$$\begin{aligned} X &= \tfrac{1}{2}\begin{pmatrix} A+B & C+D \\ C+D & -A-B \end{pmatrix}, \\ Y &= \tfrac{1}{2}\begin{pmatrix} A-B & C-D \\ -C+D & A-B \end{pmatrix}. \end{aligned} \tag{25}$$

One can check that matrices X, Y satisfy the conditions

$$\begin{aligned} &X \odot Y = 0, \quad \odot \text{ is a Hadamard product}, \\ &XY^T = YX^T, \\ &X \pm Y \quad \text{is a } (\pm 1)\text{-matrix}, \\ &XX^T + YY^T = 2mI_{2m}. \end{aligned} \tag{26}$$

Let A_0, B_0, C_0, D_0 be Williamson type matrices of order n. Introduce the matrices

$$\begin{aligned} A_i &= A_{i-1} \otimes X + B_{i-1} \otimes Y, \\ B_i &= B_{i-1} \otimes X - A_{i-1} \otimes Y, \\ C_i &= C_{i-1} \otimes X + D_{i-1} \otimes Y, \\ D_i &= D_{i-1} \otimes X - C_{i-1} \otimes Y. \end{aligned} \tag{27}$$

Prove that for any natural number i A_i, B_i, C_i, D_i are Williamson type matrices of order $(2m)^i n$. From the formulas (27) for $i = 1$ we obtain

$$\begin{aligned} A_1A_1^T =& A_0A_0^T \otimes XX^T + B_0B_0^T \otimes YY^T \\ &+ A_0B_0^T \otimes XY^T + B_0A_0^T \otimes YX^T, \\ B_1B_1^T =& B_0B_0^T \otimes XX^T + A_0A_0^T \otimes YY^T \\ &- B_0A_0^T \otimes XY^T - A_0B_0^T \otimes YX^T. \end{aligned}$$

Taking into account conditions (25) and (26) and summarizing last expressions, we find

$$A_1A_1^T + B_1B_1^T = (A_0A_0^T + B_0B_0^T) \otimes (XX^T + YY^T).$$

Similarly, it turns out that

$$C_1C_1^T + D_1D_1^T = (C_0C_0^T + D_0D_0^T) \otimes (XX^T + YY^T).$$

Now, summarizing last two equations and taking into account that A_0, B_0, C_0, D_0 is a Williamson type matrices of order n, and X and Y satisfy to conditions (26), we have

$$A_1A_1^T + B_1B_1^T + C_1C_1^T + D_1D_1^T = 8mnI_{2mn}.$$

Let's prove now equality $A_1B_1^T = B_1A_1^T$. Really, from (27)

$$\begin{aligned} A_1B_1^T =& A_0B_0^T \otimes XX^T - A_0A_0^T \otimes XY^T \\ &+ B_0B_0^T \otimes YX^T - B_0A_0^T \otimes YY^T, \\ B_1A_1^T =& B_0A_0^T \otimes XX^T + B_0B_0^T \otimes XY^T \\ &- A_0A_0^T \otimes YX^T - A_0B_0^T \otimes YY^T. \end{aligned}$$

Comparing both expressions, we conclude, that $A_1B_1^T = B_1A_1^T$. Similarly, it can be shown, that $PQ^T = QP^T$, where $P, Q \in \{A, B, C, D\}$. Thus, the matrices A_1, B_1, C_1, D_1 are Williamson type matrices of order $2mn$.

Now we assume that the theorem is correct for $k = i \geq 1$, i.e. A_k, B_k, C_k, D_k are Williamson matrices of order $(2m)^k n$. Let's prove, that A_{i+1}, B_{i+1}, C_{i+1}, D_{i+1} are also Williamson matrices. Let's check up only the second condition of the equation (25). Compute

$$\begin{aligned} A_{i+1}A_{i+1}^T =& A_iA_i^T \otimes XX^T + A_iB_i^T \otimes XY^T \\ &+ B_iA_i^T \otimes YX^T + B_iB_i^T \otimes YY^T, \\ B_{i+1}B_{i+1}^T =& B_iB_i^T \otimes XX^T - B_iA_i^T \otimes XY^T \\ &- A_iB_i^T \otimes YX^T + A_iA_i^T \otimes YY^T, \\ C_{i+1}C_{i+1}^T =& C_iC_i^T \otimes XX^T + C_iD_i^T \otimes XY^T \\ &+ D_iC_i^T \otimes YX^T + D_iD_i^T \otimes YY^T, \\ D_{i+1}D_{i+1}^T =& D_iD_i^T \otimes XX^T - D_iC_i^T \otimes XY^T \\ &- C_iD_i^T \otimes YX^T + C_iC_i^T \otimes YY^T. \end{aligned}$$

Summarizing the obtained equations, we find

$$\begin{aligned} &A_{i+1}A_{i+1}^T + B_{i+1}B_{i+1}^T + \cdots + D_{i+1}D_{i+1}^T \\ &\quad = (A_iA_i^T + \cdots + D_iD_i^T) \otimes (XX^T + YY^T). \end{aligned} \tag{28}$$

Since A_i, B_i, C_i, D_i are Williamson matrices of order $(2m)^i n$, and the matrices X, Y satisfy to conditions (26), then the equation (28) has a form

$$\begin{aligned} &A_{i+1}A_{i+1}^T + B_{i+1}B_{i+1}^T + \cdots + D_{i+1}D_{i+1}^T = \\ &\qquad = 4(2m)^{i+1}nI_{(2m)^{i+1}n}. \end{aligned}$$

S_{4m} is a monomial matrix with condition $S_{4m}^T = -S_{4m}$ and H_1 is a Hadamard matrix of order $4m$.

Thus, $P = X \otimes H_1 + Y \otimes S_{4m}H_1$ is a Hadamard matrix of order $8mn$.

References

[1] Ahmed, Rao. Orthogonal Transforms for Digital Signal Processing. Springer-Verlag, New York, 1975.

[2] Agaian S.S. Hadamard Matrices and Their Applications. Lecture Notes in Mathematics, vol. 1168, 1985.

[3] G.R. Reddy, P. Satyanarayana. Interpolation Algorithm Using WalshHadamard and Discrete Fourier/Hartley Transforms. IEEE, 1991, p. 545-547.

[4] CheungFat Chan. Efficient Implementation of a Class of Isotropic Quadratic Filters by Using WalshHadamard Transform. IEEE Int. Symposium on Circuits and Systems, June 912, Hong Kong, 1997, p. 2601-2604.

[5] Brian K. Harms, Jin Bae Park, Stephan A. Dyer. Optimal Measurement Techniques Utilizing Hadamard Transforms. IEEE Trans. on Instrumentation and Measurement, vol. 43, No. 3, June 1994, p. 397-402.

[6] Chen Anshi, Li Di, Zhou Renzhong. A Research on Fast Hadamard Transform (FHT) Digital Systems. IEEE TENCON 93/Beijing, 1993, p.541-546.

[7] Sarukhanyan H.G. Hadamard matrices: Construction methods and applications. In Proc. of The Workshop on Transforms and Filter Banks, Feb. 21-27, Tampere, Finland, 1998, 35p.

[8] Sarukhanyan H.G. Decomposition of the Hadamard matrices and fast Hadamard transform. Computer Analysis of Images and Patterns, Lecture Notes in Computer Science, vol. 1296, 1997, p.

[9] Yarlagadda Rao R.K., Hershey E.J. Hadamard Matrix Analysis and Synthesis with Applications and Signal/Image Processing, 1997.

[10] Sylvester J.J. Thoughts on Inverse Orthogonal Matrices, Simultaneous Sign Successions and Tesselated Pavements in Two or More Colours, with Applications to Newton's Rule, Ornamental Til-Work, and the Theory of Numbers. Phil. Mag., vol. 34, 1867, p. 461-475.

[11] Seberry J., Yamada M. Hadamard Matrices, Sequences and Block Designs. Surveys in Contemporary Design Theory, Wiley-Interscience Series in Discrete Mathematics. Jhon Wiley, New York, 1992.

[12] Samadi S., Suzukake Y., Iwakura H. On Automatic Derivation of Fast Hadamard Transform Using Generic Programming. Proc. 1998 IEEE Asia-Pacific Conference on Circuit and Systems, Thailand, 1998, p. 327-330.

[13] Coppersmith D., Feig E., Linzer E. Hadamard Transforms on Multiply/Add Architectures. IEEE Trans. on Signal Processing, vol. 42, No. 4, 1994, p. 969-970.

[14] Agaian S.S., Sarukhanian H.G. Reccurent Formulae of the Construction Williamson Type Matrices. Math. Notes, vol. 30, No. 4, 1981, p. 603-617.

[15] Williamson J. Hadamard Determinant Theorem and Sum of Four Squares. Duke Math. J., No. 11, 1944, p. 65-81.

[16] H. Sarukhanyan, S. Agaian, K. Egiazarian, J. Astola. On Fast Hadamard Transforms of Williamson Type. EUSIPCO, 2000.

[17] http://www.cs.uow.edu.au/people/jennie/lifework.html

[18] Agaian S.S. Construction of Planar and Spatial Block Hadamard Matrices. Proc. of Computer Centre of Armenian Academy of Sciences,(in Russian), vol. 12, 1984, p. 550.

[19] Z. Li, H.V. Sorensen, C.S. Burrus. FFT and Convolution Algotithms an DSP Microprocessors. Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing. 1986, pp. 289-294.

[20] R.K. Montoye, E. Hokenek, S.L. Runyon. Design of the IBM RISC System/6000 floating point execution unit. IBM J. Res. Develop., 1990, vol. 34, pp. 71-77 p. 5-50.

[21] http://www.top500.org/reports/1993/section2-12-7.html