RESEARCH ARTICLE OPEN ACCESS

The Haar-Recursive Transform and Its Consequence to the Walsh-Paley Spectrum and the Autocorrelation Function

H. M. Rafiq*, M. U. Siddiqi*
**Department of Electrical and Computer Engineering, International Islamic University Malaysia Kuala Lumpur, Malaysia*

ABSTRACT
The Walsh and Haar spectral transforms play a crucial part in the analysis, design, and testing of digital devices. They are most suitable for analysis and synthesis of switching or Boolean functions (BFs). It is well known that, the connection between the two spectral domains is given in terms of the Walsh-Paley transform. This paper derives an alternative expression of the Walsh-Paley transform in terms of the Haar transform. The work demonstrates the possibility of obtaining both the Haar spectrum and the Walsh-Paley spectrum using only the Haar transform domain. The paper introduces a new Haar-based transform algorithm (Haar-Paley-Recursive Transform, HPRT) in the form of a recursive function along with its fast transform version. The new algorithm is then explored in its interpretation of the Walsh-Paley transform and its connection to the Autocorrelation function (ACF) of a BF. The connection is given analogously in terms of the Haar-Paley power spectrum via the Wiener-Khintchine theorem. The paper then presents the simulation results on the execution times of both derived algorithms in comparison to the existing Walsh benchmark. The work shows the advantages of using the Haar transform domain in computing the Walsh-Paley spectrum and in effect the ACF.
Keywords **–** Autocorrelation, Haar/Walsh-Paley, Power Spectrum, Recursive Transform, Spectral Transform.

I. INTRODUCTION

The Walsh and Haar spectral transforms are significant in their use for various engineering applications [1-6]. They are considered suitable for representation of switching functions and have been applied not only in logic synthesis but as well in their related analysis including the design and testing of digital devices. Each one has its own advantages over the other when it comes to different area of applications [7]. The Walsh transform is global in nature while its Haar counterpart is characterized locally [1-7]. Their connection has been well studied and presented within existing research works including their respective hybrid transforms and underlying benefits [7-12]. The Walsh-Paley transform, in particular, is of more interest to this work as it is directly linked with the Haar transform. The link between the two is through their spectral zones and the transformation between the two domains can be induced via the Walsh-Paley transform [7-12]. In other words, the transformation from one spectral domain to another can simply be done through the Walsh-Paley transformation. On the other hand, the connection between the Autocorrelation function (ACF) of a BF and the Walsh transform domain is given by the well-known Wiener-Khintchine theorem [1,2,7,13]. The theorem states that the Walsh transform of the power spectrum gives the ACF of the respective BF. This connection coupled with the fast Walsh-Hadamard transform (FWHT) makes it easier and affordable to compute the ACF from the perspective of the computational complexity [7,13]. In this paper we focus on the transformation between the Walsh-Paley and Haar transform spectral domains. In this case, we introduce a new algorithm in form of a recursive function. This algorithm utilizes the Haar transform in its processing and answers the following posed question, "*what happens when the Haar transform is applied to a given vector and then repeatedly being applied to the zones within the transformed vector and their sub-zones recursively?*" We refer to this transformation process as the Haar-Recursive Transform (HRT).

In the process of answering the posed question, we deduce the consequence of the HRT and derive a new algorithm that we refer to as the Haar-Paley-Recursive Transform (HPRT). This algorithm is given in terms of the Haar recursive function and/or the HRT. We also present the HRT and HPRT from the matrix point of views and based on the related matrix structures, we then deduce the fast transform version of the HPRT algorithm (called here the Fast-Haar-Paley-Recursive Transform - FHPRT). It is also shown that the result of the HPRT is nothing other than the Walsh-Paley spectrum. The consequence of this interpretational point of view is the connection between the HPRT and the Autocorrelation function (ACF) of a BF. In this sense, we deduce the connection between the HPRT and the ACF by exploiting the well known Wiener-Khintchine theorem. In the process, we derive the expression of the ACF in terms of the Haar-Paley power spectrum. We then proceed to compare the

average execution times of the two derived algorithms (HPRT and FHPRT) with the existing Walsh benchmark (FWHT) and discuss the advantages of the new transform algorithm. The paper is organized as follows. Section 2 presents an overview of Boolean functions including the spectral transform methods and some of the known results to be employed in the later sections. In section 3, the HPRT is presented where the section is divided into three sub-sections each of which representing the HRT along with the related matrix representations, the HPRT, and the FHPRT respectively. The section also reviews the complexity of the FHPRT and compares it to that of the fast Walsh-Hadamard Transform (FWHT). Additionally, the HPRT connection to the ACF is presented in the same section as well. The derived algorithms have been simulated in terms of their average execution times and in comparison to the existing Walsh-Hadamard benchmark. The results of the simulation experiments are then presented in section 4 and including the discussion on the advantages and benefits of the HPRT and FHPRT. Finally, in section 5, we present the conclusion of the paper and discussion on future work.

II. PRELIMINARIES

1.1 Boolean Functions

An *n*-variable Boolean function f, is a mapping of *n* binary inputs to a single binary output. It can formally be defined as [1,2,3]: $f: \mathbb{F}_2^n \to \mathbb{F}_2$ Maps $(x_1, \dots, x_n) \in \mathbb{F}_2^n \mapsto f(x) \in \mathbb{F}_2$

The input is an *n*-dimensional binary vector $x = (x_1, x_2, \dots, x_n) \in \mathbb{F}_2^n$ $(x_i \in \mathbb{F}_2)$, while the output of the function is given by $f(x) \in \mathbb{F}_2$. The set of all Boolean functions is denoted by B_n. Any $f \in B_n$ has a unique representation in each of the following forms [2]:

The binary truth table of f – Ordered tuple given by $T_f = (f(x^{(0)}), f(x^{(1)}), \dots, f(x^{(2^n-1)}))$ which lists the output of the function for all 2^n input combinations, where $x^{(0)} = (0, \dots, 0)$ and $x^{(2^n-1)} = (1, \dots, 1)$ ($x^{(k)}$ is the binary vector representation of the integer k, for $0 \le k \le 2^n - 1$ with the relationship $k = \sum_{i=1}^{n} 2^{n-i} x_i$).

The polarity truth table of f – This is the real valued representation of the function referred to as the *sign function* $\hat{f}$ ($\hat{f} \in \{-1,1\}$), which is defined as $\hat{f}(x) = (-1)^{f(x)} \equiv 1 - 2f(x), \forall x \in \mathbb{F}_2^n$. Its truth table is called the *sequence* of f. This representation is considered more advantageous in some of engineering applications.

The Algebraic Normal Form (ANF) – The polynomial representation expressed uniquely as a sum (XOR) of products (AND): $f(x) = a_0 \oplus a_1 x_1 \oplus \cdots \oplus a_{12} x_1 x_2 \oplus \cdots \oplus a_{12\cdots n} x_1 x_2 \cdots x_n$ where $a_i, x_i \in \mathbb{F}_2$.

1.2 Spectral Transforms

In this section we look at the Haar and Walsh spectral transforms of Boolean functions. The focus of this work is on the Paley ordering of the Walsh transform. That is, all the considerations will be done with regards to the Walsh-Paley transform and its connection to the Haar transform. We also present some of the existing results that will be employed in the subsequent sections of the paper.

Throughout this paper the following notations and abbreviations will be assumed: *WH and WP* are the Walsh-Hadamard and Walsh-Paley orderings respectively; $\vec{y}_j$ is the *j*-th row (*Y* function) in the respective transform matrix; $\vec{r}_{0_s}$ is a row-vector whose elements are all ones ($\vec{1}$) with size 1×2^s; $\vec{r}_{1_s}$ is a balanced row-vector whose first half elements are all ones and the second half elements are all negative-ones with size 1×2^s; I_l as the $2^l \times 2^l$ identity matrix; and $Y_j \cdot f$ is the inner dot product between the elements of Y and f.

Walsh-Hadamard Transform (WHT) of a function $\hat{f}$ on $\mathbb{F}_2^n$ is denoted by $\hat{F}_{WH}$ and given by [1, 2]:

$$\hat{F}_{WH}(u) = \sum_{x,u \in \mathbb{F}_2^n} (-1)^{f(x) \oplus x \cdot u} \equiv \overrightarrow{WH}_u(x) \cdot \hat{f}(x) \quad (1)$$

Where $\overrightarrow{WH}_u(x) = (-1)^{u \cdot x}$ defines the WH function, and the transform can be given equivalently in matrix form as [12]:

$$\hat{F}_W = [W_n] \cdot \left[\hat{f}\right]^t \quad (2)$$

Where $[W_n] = \left[\overrightarrow{W_j}\right]$ is a $2^n \times 2^n$ Walsh transform matrix whose rows $(0 \le j < 2^n)$ constitutes the Walsh functions ($\overrightarrow{w_j}$), and $\left[\hat{f}\right]^t$ is a column vector of $\hat{f}$.

Generator Matrix: the Walsh transform matrices can be generated recursively depending on which ordering is being considered. In the context of this paper, we will consider the generator for the Walsh matrices in Paley ordering as it will be employed later in the derivations. The following is the recursive generation of the Walsh matrices in Paley ordering [9,10,11]:

$$[WP_n] = \left[\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \otimes \overrightarrow{rp}_{i_{n-1}}\right], \forall\, i \in [0, 2^{n-1}) \quad (3)$$

Where $[WP_0] = [1]$, $\otimes$ is the Kronecker product and $\overrightarrow{rp}_{i_{n-1}}$'s are the rows of the previous lower order matrix ($[WP_{n-1}]$). Note that each row in the $(n-1)^{th}$-order produces two rows in the $(n)^{th}$-order, i.e. the i-row in $(n-1)^{th}$-order produces rows $2i$ and $2i+1$ in the $(n)^{th}$-order. Consequently, the rows in the interval $[2^l, 2^{l+1})$ (in $(n-1)^{th}$-order) will produce the rows in the interval $[2^{l+1}, 2^{l+2})$ (in the $(n)^{th}$-order).

Autocorrelation Function (ACF): denoted as $(\hat{r}_f(a))$, the ACF of a BF f can be defined simply

based on equation (4) [1,2,13]. The expression $\hat{f}(x)\cdot\hat{f}(x\oplus a)$ is referred to as the directional derivative of the respective BF.

$$\hat{r}_f(a)=\sum_{x\in\mathbb{F}_2^n}\hat{f}(x)\cdot\hat{f}(x\oplus a) \quad (4)$$

Haar Functions: The set of Haar functions H_l^q (resp. H_j) forms a complete set of orthogonal rectangular basis functions [7,9]. They are defined on the interval $[0,2^n)$ as un-normalized taking the values of 0 and ±1 as follows:

$$H_0^{(0)}=\vec{H}_0(x)=1,\forall x\in[0,2^n)$$

$$\vec{H}_j(x)=\begin{cases}1, & u_0\cdot 2^{n-l-1}\le x<u_1\cdot 2^{n-l-1}\\ -1, & u_1\cdot 2^{n-l-1}\le x<u_2\cdot 2^{n-l-1}\\ 0, & else\ in\ [0,2^n)\end{cases} \quad (5)$$

Where $u_i=2q+i$; l and q are degree and order of the Haar functions respectively. With $j=2^l+q$ and for each value of $l=0,1,\dots,n-1$, we have $q=0,1,\dots,2^l-1$.

Haar Transform: The Haar transform ($\hat{F}_H$) of $\hat{f}$ is defined by equation (6) [7,9,14] and its equivalent matrix representation is given by equation (7) respectively [7,8,9,14].

$$\hat{F}_H(j)=\sum_{x=0}^{x=2^n-1}H_l^q(x)\cdot\hat{f}(x)\equiv\sum_x\vec{H}_j\cdot f(x) \quad (6)$$

$$\hat{F}_H=[H_n]\cdot\left[\hat{f}\right]^t \quad (7)$$

Where $[H_n]=[\vec{H}_j]$ is a $2^n\times 2^n$ Haar transform matrix whose rows consist of Haar functions (H_j *'s*), and with the following generator:

$$[H_n]\equiv\begin{bmatrix}\vec{r}_{0_n}\\ \left[I_l\otimes\vec{r}_{1_{n-l}}\right]\end{bmatrix}\quad(\text{For } l\in[0,n)) \quad (8)$$

The calculation of the Haar Spectrum in terms of fast transforms is based on a simple algorithm that involves *s*-recursive construction of the sequence $a_s(x)$ $(s=1,2,\dots,n;x=0,1,\dots,2^{n-s+1})$. This algorithm is a result of the following theorem which was first introduced by Karpovsky [7].

Theorem 1: let f *be a step function representing a system of Boolean function of* n *arguments and* $F_H(x)$, *its Haar spectrum.Where:* $l\in[0,n)$; $q\in[0,2^l)$; $s\in[1,n]$. Set

$a_0(x)=f(x),(x=0,1,\dots,2^n-1)$

$a_s(x)=a_{s-1}(2x)+a_{s-1}(2x+1),\ x\in[0,2^{n-s})$

$a_s(2^{n-s}+x)=a_{s-1}(2x)-a_{s-1}(2x+1)$

Then, $F_{H_{n-s}}^{(q)}=2^{-s}a_s(2^{n-s}+q)$ (9)

Lemma 1: Relationship between the Haar functions $H_l^q(x)$ and the Walsh-Paley (WP) functions[7] is based on equation (10), $\forall l\in[0,n);q=0,1,\dots,2^l-1$ as:

$$H_l^q(x)=\begin{cases}WP_{2^l}(x), & x\in S_q^l\\ 0, & otherwise\end{cases} \quad (10)$$

Let $\left[SWP_{2^n}^l\right]$ be the Walsh-Paley sub-matrix and $\left[SH_{2^n}^l\right]$ be the Haar sub-matrix, then the two respective sub-matrices are of dimension $2^l\times 2^n$, and can be defined based on the spectral zones (degrees $l\in[0,n)$) as follows [10, 12];

$$\left[SWP_{2^n}^l\right]=\begin{bmatrix}\overrightarrow{WP}_{2^l}\\ \overrightarrow{WP}_{2^l+1}\\ \vdots\\ \vdots\\ \overrightarrow{WP}_{2^{l+1}-1}\end{bmatrix};\left[SH_{2^n}^l\right]=\begin{bmatrix}\vec{H}_{2^l}\\ \vec{H}_{2^l+1}\\ \vdots\\ \vdots\\ \vec{H}_{2^{l+1}-1}\end{bmatrix} \quad (11)$$

The sub-matrices are simply obtained by dividing the respective transform matrices based on the sub-intervals $[2^l,2^{l+1})$ defined by the degrees l and including the global functions. Which are equivalent to the structure of $\left[I_2\otimes\vec{r}_{1_1}\right]$ as given in equation (8).

Lemma 2: The Relationship between the Haar and the Walsh-Paley (WP) sub-matrices is defined by [10, 12]:

$$\left[SWP_{2^n}^l\right]=[WP_l]\cdot\left[SH_{2^n}^l\right],l=0,\dots,n-1 \quad (12)$$

Where $\left[SWP_{2^n}^l\right]$ is a $2^l\times 2^n$ Walsh-Paley sub-matrix, $\left[SH_{2^n}^l\right]$ is the Haar sub-matrix ($2^l\times 2^n$) and $[WP_l]$ is the Walsh-Paley transform matrix of order l.

Lemma 3: Relationship between the Haar ($\hat{\mathrm{F}}_{\mathrm{H}}(\mathrm{x})$) and the Walsh-Paley ($\hat{\mathrm{F}}_{\mathrm{WP}}(\mathrm{x})$) spectral coefficients $\forall l\in[1,n)$ [7, 10]:

$$\hat{F}_{WP}(x)=\begin{cases}\hat{F}_H(x), & x=0,1\\ [WP_l]\cdot\left[\hat{F}_H(x)\right]^t, & x\in[2^l,2^{l+1})\end{cases} \quad (13)$$

Theorem 2: A relation between the Walsh transform and the Autocorrelation function is given by (based on Wiener-Khintchine theorem) [1, 2]:

$$\hat{R}_W(x)=\hat{F}_W{}^2(x)\quad\forall x\in\mathbb{F}_2^n \quad (14)$$

III. THE HAAR-PALEY RECURSIVE TRANSFORM (HPRT)

In this section, we present the Haar-Paley Recursive Transform (HPRT). The section is divided into three parts. The first part examines the Haar-Recursive transform (HRT) as a recursion algorithm. This idea is then extended in the second part to introduce the related spectral transform (called here HPRT). The third part on the other hand, considers the fast transform version of the HPRT from the algorithmic point of view and fast signal flow. The resulting fast transform in this sense, is referred to as the FHPRT.

1.3 Haar-Recursive Transform (HRT)

In this part, we introduce and define the notion of the Haar-Recursive Transform (HRT) which will be deployed further in the next sub-section. We define this notion in terms of a recursive function and demonstrate its interpretation with relevant examples.

Definition 3.1: Let V be a vector with dyadic length (2^n elements) then define *dyadic partitioning* of V by its partition into sub-vectors of dyadic lengths $\forall l\in[0,n)$ as

$[V]=[V(0),V(1),\dots,V(2^n-1)]$

$$= [V(0), [V_l]] \quad (15)$$

Where $[V_l] = [V(2^l), V(2^l + 1), \dots, V(2^{l+1} - 1)]$

It should be noted that if V from Definition 3.1 represents the Haar spectrum, then it is obvious that $[V_l]$ represents nothing other than the Haar spectral zones defined by the respective degrees l.

Definition 3.2: Let V be a vector with dyadic partitioning ($lengt\hbar = 2^n, n \geq 1$), then define the Haar-Recursive Transform (HRT) of V ($HRT_V(n) = W$) by the following recursion steps:

1. Base-Case: $lengt\hbar(V) = 2$
   $W = HRT_V(n) = FHT(V)$
2. Recursion-step:

i. $W = HRT_V(n) = FHT(V)$

ii. $\forall l \in [1, n) \ \wedge \ \forall x \in [2^l, 2^{l+1})$:

$$W(x) = HRT_W(l) = FHT\big(W(x)\big) \quad (16)$$

Note: the recursion algorithm can also take a sub-matrix as its input. An alternative definition of the HRT's recursion-step (**Definition 2**) can be given in matrix representation as follows.

Definition 3.3: Let a $2^n \times 2^n$ matrix be denoted by $[HR_n]$ and represents the HRT matrix, then it can simply be defined as a product of a recursive matrix and the respective Haar transform matrix ($[H_n]$). The recursive matrix is a diagonal matrix with the lower HRT matrices ($[HR_i], i \in [0, n)$) as the sub-blocks down the diagonal. Its definition is given as follows:

$$[HR_n] = \begin{bmatrix} [1] & & & 0 \\ & [HR_0] & & \\ & & \ddots & \\ 0 & & & [HR_{n-1}] \end{bmatrix} \cdot [H_n] \quad (17)$$

Where $[HR_0] = [H_0] = [1]$

Definition 3.4: Let V be as defined in the Definition 3.2, then an equivalent definition of HRT of V in terms of the transform matrix $[HR_n]$ can be given by:

$$W = HRT_V(n) = [HR_n] \cdot [V]^t \quad (18)$$

Remark: What the HRT does is as follows: it takes a vector and then apply Haar transform to it recursively first by dealing with the entire vector. Then it divides the resulting transformed vector into sub-vectors (with dyadic length) and then for each of them, it repeats the same procedure until it gets to a sub-vector of length 2 only when it quits (see Fig. 1 below). The arrows in Figure 1 represents recursive process of applying the Haar transform to the original vector then followed by the same procedure to the zones and their sub-zones.

The following example (Example 1) demonstrates the idea behind Definitions 3.3 and 3.4 with the matrix interpretation of the HRT.

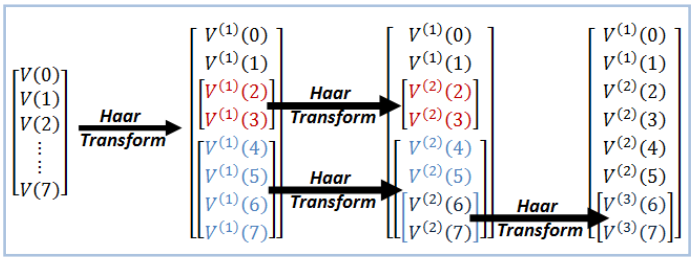


Fig. 1: Visual Aspect of HRT for n = 3 ($lengt\hbar(V) = 2^3$)

Example 1: Consider the case of $n = 1,2,3$ then based on Definition 3.3 we have the HRT matrices as follows ($[HR_0] = [1]$):

$$n = 1 \ \Rightarrow \ [HR_1] = \begin{bmatrix} [1] & 0 \\ 0 & [HR_0] \end{bmatrix} \cdot [H_1] = [H_1]$$

$$n = 2 \ \Rightarrow \ [HR_2] = \begin{bmatrix} [1] & & 0 \\ & [HR_0] & \\ 0 & & [HR_1] \end{bmatrix} \cdot [H_2]$$

$n = 3$:

$$\Rightarrow \ [HR_3] = \begin{bmatrix} [1] & & & 0 \\ & [HR_0] & & \\ & & [HR_1] & \\ & 0 & & [HR_2] \end{bmatrix} \cdot [H_3]$$

The following proposition (Proposition 1) entails the consequence of the HRT matrix in its connection to the Walsh-Paley transform matrix by summarizing the relationship between the two transform matrices.

Proposition 1: Given an n-variable domain, then the HRT matrix is equal to the Walsh-Paley transform matrix. That is, for $n \geq 1$

$$[HR_n] = [WP_n] \quad (19)$$

Proof: we prove this proposition using induction on the HRT matrix as follows

Base case: Let $n = 1$, then the HRT matrix is $[HR_1] = [H_1] = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = [WP_1]$. Which is *true* for the proposition.

Induction step*: Let $n \geq 1$ be given and suppose that the proposition is true for $n = k$. This means that the proposition holds for $n \leq k$, then by induction hypothesis and deploying Lemma 1, 2 with equation (17) we have*

$$[HR_{k+1}] = \begin{bmatrix} [1] & & & 0 \\ & [HR_0] & & \\ & & \ddots & \\ 0 & & & [HR_k] \end{bmatrix} \cdot [H_{k+1}]$$

Now, with the induction hypothesis followed by the use of Lemma 1 and Lemma 2 we then get

$$= \begin{bmatrix} [1] & & & 0 \\ & [HR_0] & & \\ & & \ddots & \\ 0 & & & [HR_k] \end{bmatrix} \cdot \begin{bmatrix} \vec{H}_0 \\ [SH^0_{2^{k+1}}] \\ [SH^1_{2^{k+1}}] \\ \vdots \\ [SH^{k-1}_{2^{k+1}}] \\ [SH^1_{2^{k+1}}] \end{bmatrix}$$

$$= \begin{bmatrix} \vec{H}_0 \\ [1] \cdot [SH^0_{2^{k+1}}] \\ [WP_1] \cdot [SH^1_{2^{k+1}}] \\ \vdots \\ [WP_{k-1}] \cdot [SH^{k-1}_{2^{k+1}}] \\ [WP_k] \cdot [SH^k_{2^{k+1}}] \end{bmatrix} = \begin{bmatrix} \overrightarrow{WP}_0 \\ [SWP^0_{2^{k+1}}] \\ [SWP^1_{2^{k+1}}] \\ \vdots \\ [SWP^{k-1}_{2^{k+1}}] \\ [SWP^k_{2^{k+1}}] \end{bmatrix} = [WP_{k+1}]$$

Thus, the proposition holds for $n = k + 1$ and this completes the proof of the induction step. Hence by using the principle of induction, the proposition holds for all $n \geq 1$. □

Now, the proposition can naturally be extended to the relationship between the Haar and Walsh-Paley sub-matrices. This is done simply by modifying the relationship given in Lemma 2 and use the HRT in place of the Walsh-Paley transform matrix.

Proposition 2: Consider the Haar sub-matrix $[SH^l_{2^n}]$, $l \in [1, n)$, $q \in [0, 2^l)$, and dyadic intervals $[2^l, 2^{l+1})$, then the Haar recursive transform of this sub-matrix ($denoted\ by\ HRT_{SH}(l)$) results into the Walsh-Paley sub-matrix $[SWP^l_{2^n}]$:

$$HRT_{SH}(l) = [SWP^l_{2^n}] \quad (20)$$

Proof: The proof follows directly from definition of the HRT, the use of Lemma 2 and the proposition 1 as follows:

$$\Rightarrow HRT_{SH}(l) = [HR_l] \cdot [SH^l_{2^n}] \quad (Definition\ 3.4)$$
$$= [WP_l] \cdot [SH^l_{2^n}] \quad (Proposition\ 1)$$
$$= [SWP^l_{2^n}] \quad (Lemma\ 2)$$ □

The next example demonstrates the results given by proposition 2.

Example 2: Consider a given Haar matrix of order 3 ($[H_3]$) and its Haar sub-matrices $([SH^1_{2^3}], [SH^2_{2^3}])$, then the steps (with respect to the degrees l) involved in the execution of the HRT are given by:

$$l = 1 \Rightarrow [HR_1] \cdot [SH^1_{2^3}] = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 1 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \end{bmatrix}$$

$$l = 2 \Rightarrow [HR_2] \cdot [SH^2_{2^3}]$$
$$= \begin{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \end{bmatrix} \\ \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \cdot [I_1 \otimes \vec{r}_{1_1}] \end{bmatrix} \cdot [I_2 \otimes \vec{r}_{1_1}]$$
$$= [WP_2] \cdot [SH^2_{2^3}] = [SWP^2_{2^3}]$$
$$= \begin{bmatrix} 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix}$$

(*End of Example*)

1.4 Haar-Paley Recursive Transform (HPRT)

The HRT presented in the previous section gives the connection between the Haar and the Walsh-Paley sub-matrices. The consequence is that, the Walsh-Paley transform matrix can be given alternatively based on the proposition 1. This is true since, the first two global coefficients for both the Haar and the Walsh-Paley spectra are the same and the rest are connected via the spectral zones. The zones from the two spectra can be connected in this sense through the HRT relationship.Additionally, performing the HRT transform on a given vector only once, gives the Haar spectrum of the respective vector. On the other hand, performing it repeatedly on a given vector gives the Walsh-Paley spectrum of the same vector. *For the case of the transformed vector being a Boolean function, then we refer to the **HRT** process in this work as the "**HPRT**". In turn, the effect of applying the **HPRT** to a given vector is nothing other than the **Haar-Paley-Recursive-Spectrum** (**HPRS**). In essence, the HPRS is simply the Walsh-Paley spectrum obtained through the use of the Haar transform.* A summary of the relationship between the Haar and Walsh-Paley transforms is given in the following figure (see Fig. 2 below).

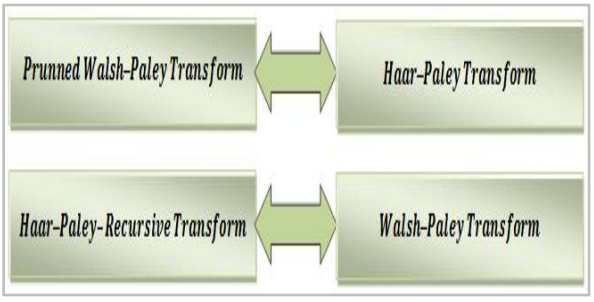


Fig. 2: Relationship between the Haar and Walsh-Paley Transforms

In the figure below (Fig. 3), we give the algorithm for the steps involved in the computation of the HPRS. The algorithm in this case, depicts the recursion steps defined in the HPRT. The input to

the algorithm is an arbitrary Boolean function f, which is transformed using the HPRT to get the respective HPRS as an output vector. The first two steps of the algorithm determine the length of f and its corresponding number of variables respectively. The third step represents the *base case* for the recursive algorithm, while steps 4 and 5 gives the respective recursion step defining the HRT process. The HPRS is given as the output vector $current_{vector}$ in the final step of the algorithm.

Remark: The HPRT is given here as a *recursion* algorithm in terms of the *Haar-recursive function*. In the next sub-section we will consider it from a *fast transform* perspective with the related computational complexity.

1.5 Fast Haar-Paley Recursive Transform (FHPRT)

The Haar transform as given in the preliminary section involves a sequence of transforms denoted as $a_i's\ (i \in [0, n))$. It is carried out as follows:

Algorithm: $Haar\_Paley\_Recursive\_Transform(f)$
Input: A Boolean function truth table, f
Output: Haar-Paley Recursive Spectrum
Steps:
Step 1: Determine the length of the input vector, $size$
$size = length(f)$
Step 2: Determine the number of variables, n_{local}
$n_{local} = \log_2(size)$
Step 3: $If\ n_{local} = 1\ (size = 2),\ then\ Do$
i. $current_{vector} = Haar\_Transform(f)$
ii. Go to Step 6
$Else\ Go\ to\ Step\ 4$
Step 4: Case $n_{local} > 1\ (size > 2)$
$current_{vector} = Haar\_Transform(f)$
Step 5: Recursion step, $\forall\ l \in [1, n_{local})\ Do$
i. $temp = current_{vector}(x),\ \forall x \in [2^l, 2^{l+1})$
ii. $current_{vector}(x) = Haar\_Recursive\_Transform(temp)$
Step 6: Output $current_{vector}$

Fig. 3: The HPRT Algorithm

The original signal to be transformed is taken first as a_0 and then recursively transformed n times in a sequential order (n is the number of variables). The resulting vector in this case is the Haar spectrum given by a_n. During the sequence of transforms, the length of the transformed signal for any two successive sequences a_i and a_{i+1} is reduced by a factor of 2. That is, the length of a_i is twice as much that of a_{i+1}. The signal-flow example of the Haar transform of a 3-variable Boolean function is depicted in Fig. 4(a) below.

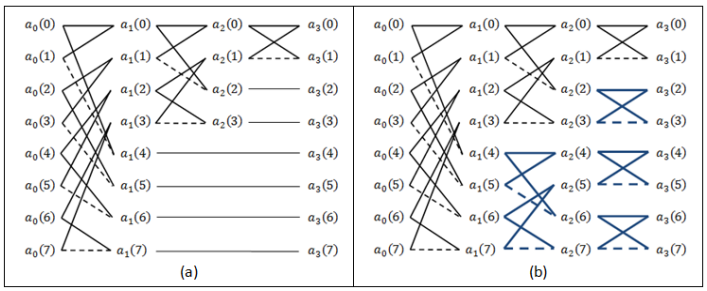


Fig. 4: Signal Flows for $n = 3$ - (a) The FHT (b) The FHPRT

On the other hand, the Haar transform process is applied repeatedly to the zones of the Haar spectrum and recursively to their subzones for the case of the HPRT algorithm. At any given sequence of transform a_i, the HPRT basically process the entire sequence by simply repeating the same procedure of the Haar transform for the entire sequence and not just a portion of it. This fills up the entire current processed sequence and therefore not having the processing reduction of a factor of 2. In turn, the number of processing for each sequence of the transform is the same. The signal flow of the *Fast Haar-Paley-Recursive Transform (FHPRT)* for the case of a 3-variable function is depicted in Fig. 4(b). The HRT over the Haar transform in this case, is portrayed by the blue colour highlight in the signal flow. That is, the added recursion or the repetitions of the Haar transform procedure for the entire processed signal. We define the fast Haar-Paley-Recursive transform as according to the following definition.

Definition 3.4 Let $f(x)$ be an n-variable Boolean function and $F_{HPR}(x)$ its Haar-Paley recursive spectrum (HPRS), set

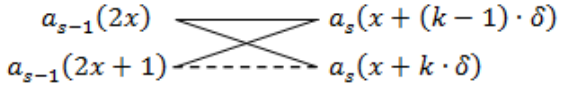
$a_0(x) = f(x), \quad x \in [0, 2^n)$

For $s \in [1, n]$, $k \in [1, 2^{s-1}], x \in [(k-1)\cdot\delta, k\cdot\delta]$, and $\delta = 2n - s$

$$a_s(x + (k-1)\cdot\delta) = a_{s-1}(2x) + a_{s-1}(2x+1),$$
$$a_s(x + k\cdot\delta) = a_{s-1}(2x) - a_{s-1}(2x+1),$$

Then, $F_{HPR}(x) = a_n(x)$ (21)

This definition is an extension of Theorem 1 given in preliminary section. It can simply be pictured through the butterfly pair as:



Note that, the parameter δ represents the step for the butterfly output pair (how far apart) of coefficients. The parameter k on the other hand, defines the Haar-recursive transform over the processed signal sequence. If k is only restricted to one ($k = 1$ only) then the algorithm becomes nothing other than the Haar transform. A summary of the steps describing this algorithm is given in the figure below (Fig. 5), as well as the flow chart depicting the algorithm's steps (Fig. 6).

Input: n**-variable Boolean function** $f(x)$
Output: HPRS (F_{HPR})
Step 1: set $a_0(x) = f(x)$
Step 2: $\forall s \in [1, n]$ Do the following
1. Compute δ
2. $\forall k \in [1, 2^{s-1}]$ *Do*
   a. Compute sub-interval: $x \in [(k-1)\cdot\delta,\ k\cdot\delta)$
   b. $\forall x \in [(k-1)\cdot\delta,\ k\cdot\delta)$ Compute the pairs: $a_s(x+(k-1)\cdot\delta)\ \wedge\ a_s(x+k\cdot\delta)$

Step 3: Output a_n

Fig. 5: The FHPRT Algorithm

Fig. 5 depicts the steps involved in the FHPRT algorithm. Step 1 involves assigning the Boolean function *f*, as the initial sequence to be processed during transformation. The transformation is carried out through *n* steps (*n* is the number of variables for the Boolean function) defined by the parameter *s* where each step involves computing the current sequence (a_s) using the previous one (a_{s-1}) (step 2 of the algorithm). For each of the current processed step *s*, two parameters are used (δ and k) in step 2 of the algorithm. These two parameters were defined along with the definition of FHPRT in the previous paragraphs.The Flow chart given in Fig. 6 provides a better view of the steps of the algorithm. There are three main loops in the algorithm, the first one determined by the parameter *s*, the second one by the parameter *k*, and the last one with the variable *x*. The first one decides the transform steps, the second decides how many Haar transform repetitions to be carried out for the current step, and the last one does the butterfly operation for each repeated Haar transform. The algorithm stops when $s > n$, giving out the HPRS as a_n.

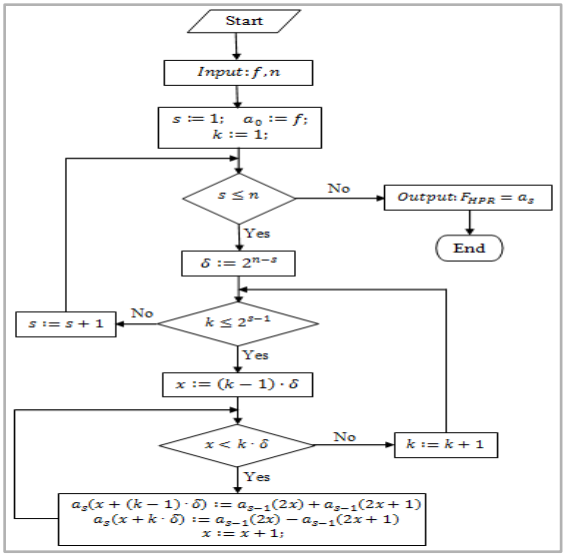


Fig. 6: Flow Chart for the FHPRT Algorithm

Remark: As the HPRT coincides with the Walsh-Paley transform then it is obvious that the HPRT matrix ($[HR_n]$) is also a self inverse matrix with appropriate normalizing weight factor, that is the inverse matrix is simply $[HR_n]^{-1} = 2^{-n} \cdot [HR_n]$. Consequently, the HPRT is a self inverse transform

meaning that the fast algorithm can be applied for both forward and inverse (with the respective weight factor) transforms.

FHPRT Time Complexity: Operations considered are *Additions* and *Subtractions* during the process of sequence of transforms from the initial sequence to the final sequence. For each butterfly pair, there are two operations involved which are: *1 addition* and *1 subtraction*. In turn, the number of processing for each sequence of the transform is the same. Each sequence of transforms now has 2^{n-1} butterfly pairs to be processed, each of which involves two operations of addition and subtraction. This gives a total number of operations within a step of transform as $2 \times 2^{n-1} = 2^n$, which consequently makes a total of $n \cdot 2^n$ number of operations for the completion of all the n steps of the transform. This coincides with the complexity of the fast Walsh-Hadamard transform [7].The presented algorithms in this section were implemented using MATLAB software and their performances on a PC machine were compared along with the Walsh-Hadamard algorithm as a benchmark. The last but one section (see Section 4 below) presents the results of this experiment. The following sub-section (Section 3.4) explores the connection between the HPRT and the ACF.

1.6 The FHPRT and Its Consequence to the Autocorrelation Function of a BF

As the FHPRT gives directly the Walsh-Paley spectrum, it is natural to connect this interpretation to the Wiener-Khintchine theorem for expressing the Autocorrelation function in terms of the HPRT. The following proposition summarizes this relationship

Proposition 3: Let $\hat{f}(x)$ be an *n*-variable Boolean function, $F_{HPR}(x)$ its Haar-Paley recursive spectrum (HPRS), and $\hat{r}_f(a)$ its Autocorrelation function. Then the Autocorrelation function can be expressed in terms of the Haar-Paley power spectrum as follows:

$$\hat{r}_f(x) = 2^{-n} \cdot \left(\hat{F}_{HPR}{}^2(x)\right)_{HPR} \quad \forall x \in \mathbb{F}_2^n \qquad (22)$$

Proof: we utilize theorem 2 and the fact that the HPRT coincides with the Walsh-Paley transform (WPT). From theorem 2 we have, the Walsh-Paley transform of the Autocorrelation function is the Power spectrum in Paley ordering which is given by $\hat{R}_{WP}(x) = \hat{F}_{WP}{}^2(x) \ \ \forall x \in \mathbb{F}_2^n$. The Walsh-Paley inverse transform of the Power spectrum in turn gives the Autocorrelation function as $\hat{r}_f(x) = [WP_n]^{-1} \cdot \left[\hat{F}_{WP}{}^2\right]^t = 2^{-n} \cdot [WP_n] \cdot \left[\hat{F}_{WP}{}^2\right]^t$. Since the HPRT is equivalent to the WPT and their transform matrices are self inverses with appropriate weight factors then it follows that $\hat{r}_f(x) = 2^{-n} \cdot [HR_n] \cdot \left[\hat{F}_{WP}{}^2\right]^t = 2^{-n} \cdot \left(\hat{F}_{HPR}{}^2(x)\right)_{HPR}$. □

The next example demonstrates the relationship portrayed by the Proposition 3. In this case the consideration is done for three 4-variable BFs including a Bent function, a nonlinear balanced BF, and a Linear BF. Bent functions are considered as the class of BFs farthest from being linear and comprise of all *zero* ACF coefficients except the global element [1,2,3,14].

Example 3: Consider the following 4-variable BFs $\hat{f}_1$, $\hat{f}_2$ and $\hat{f}_3$. The first function is a Bent BF, the second one is a nonlinear balanced BF, and the third is a linear BF. The functions' Haar-Paley-Recursive spectra (HPRS, $\hat{F}_{HPR}$), including the Haar-Paley power spectra ($\hat{F}_{HPR}{}^2$) and their HPRTs ($h\hat{r}_{\hat{f}} = 2^{-n} \cdot \left(\hat{F}_{HPR}{}^2\right)_{HPR}$) are given in the Table 1 below. Their polarity truth-tables are given respectively as follows:

$\hat{f}_1 = [1,1,1,-1,1,1,-1,1,-1,-1,-1,1,1,1,-1,1]$,

$\hat{f}_2 = [1,1,-1,-1,1,-1,-1,1,1,-1,1,-1,-1,-1,1,1]$,

$\hat{f}_3 = [1,1,1,1,-1,-1,-1,-1,-1,-1,-1,-1,1,1,1,1]$

Table 1: HPRSs, Haar-Paley Power Spectra and their HPRTs for BFs given in Example 3

x	$\hat{F}_{1\,HPR}$	$\hat{F}_{2\,HPR}$	$\hat{F}_{3\,HPR}$	$\hat{F}_{1\,HPR}{}^2$	$\hat{F}_{2\,HPR}{}^2$	$\hat{F}_{3\,HPR}{}^2$	$h\hat{r}_{f_1} = \hat{r}_{f_1}$	$h\hat{r}_{f_2} = \hat{r}_{f_2}$	$h\hat{r}_{f_3} = \hat{r}_{f_3}$
0	4	0	0	16	0	0	16	16	16
1	4	0	0	16	0	0	0	0	16
2	-4	0	0	16	0	0	0	-8	16
3	4	0	16	16	0	256	0	-8	16
4	4	0	0	16	0	0	0	0	-16
5	4	8	0	16	64	0	0	0	-16
6	-4	8	0	16	64	0	0	0	-16
7	4	0	0	16	0	0	0	0	-16
8	-4	4	0	16	16	0	0	0	-16
9	4	-4	0	16	16	0	0	0	-16
10	4	4	0	16	16	0	0	0	-16
11	4	-4	0	16	16	0	0	0	-16
12	4	4	0	16	16	0	0	-8	16
13	-4	4	0	16	16	0	0	-8	16
14	-4	-4	0	16	16	0	0	8	16
15	-4	-4	0	16	16	0	0	8	16

IV. SIMULATION RESULTS

The algorithms for the HPRT and FHPRT were implemented using the MATLAB software, and the experiments on performance comparisons between these two algorithms and the Walsh-Hadamard algorithm were conducted. The

experiments were carried out on laptop computer with the following specifications; Dell-Vostro-3450, Intel Core i5-2410M, CPU @ 2.30GHz, and 4.0GB RAM. The focal point of the experiments was on execution time (using the tic-toc MATLAB function). The fast Walsh-Hadamard transform (FWHT) provided as a built-in MATLAB function was used as the benchmark.The execution of each algorithm was conducted in the following manner: number of iterations on which the same algorithm executed was picked as 200 (ignoring the first time execution for the fetch delay), at the end of each run the average execution time was computed for the number of variables between 3 and 20 inclusive. The following tables (Table 2 and Table 3) show the results of the experiment, for the average execution times of the algorithms. The execution times during the different iterations for the given number of variables are presented in the figures below (see Fig. 7, Fig.8 and Fig. 9).

Table 2: Algorithms' Average Execution Times (sec) for $n \in [3,11]$

n	HPRT	FHPRT	FWHT
3	0.00008	0.00001	0.00021
4	0.00017	0.00002	0.00023
5	0.00039	0.00005	0.00028
6	0.00073	0.00009	0.00040
7	0.00154	0.00019	0.00065
8	0.00320	0.00043	0.00122
9	0.00671	0.00091	0.00246
10	0.01337	0.00212	0.00539
11	0.02636	0.00460	0.01222

Table 3: Algorithms' Average Execution Times (sec) for $n \in [12,20]$

n	**HPRT**	**FHPRT**	**FWHT**
12	0.05944	0.01042	0.03036
13	0.11640	0.02488	0.05678
14	0.23005	0.04871	0.12858
15	0.47885	0.10482	0.26154
16	0.98741	0.27912	0.68846
17	2.49517	0.57414	1.45983
18	4.96914	1.20209	3.09637
19	9.92213	2.46331	6.29882
20	18.73959	4.10494	12.05639

As it can be seen from both tables (Table 2 and Table 3) and the figures (Fig. 7, 8 and 9), on average execution times and with low number of variables the algorithms are performing at almost the same level ($n \in [3,5]$). At the beginning of the experiment (first iteration), the execution time is somewhat high, but as the number of iterations change then the execution time for the algorithms tend to stabilize to a certain level. On the other hand, as the number of variables increases, the performance of the fast algorithms is almost same (see Fig. 9 below) but for the recursion function (HPRT), the execution time increases relative to the other two. Even though the two fast algorithms perform way better than the one based on recursion, yet the FHPRT outperforms the MATLAB built-in FWHT by a factor of almost three. Note that, the average execution time given in Fig. 9 is computed in common logarithm.

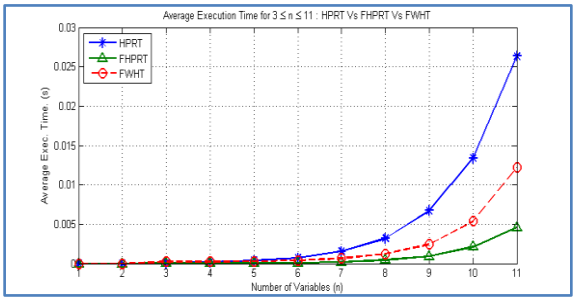


Fig. 7: Algorithms' Average Execution times (sec) for 200 Iterations and $n \in [3,11]$

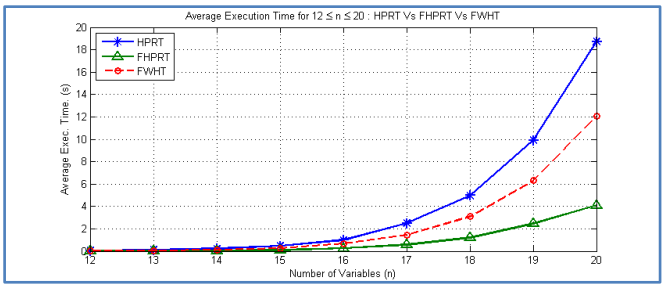


Fig. 8: Algorithms' Average Execution times (sec) for 200 Iterations and $n \in [12,20]$

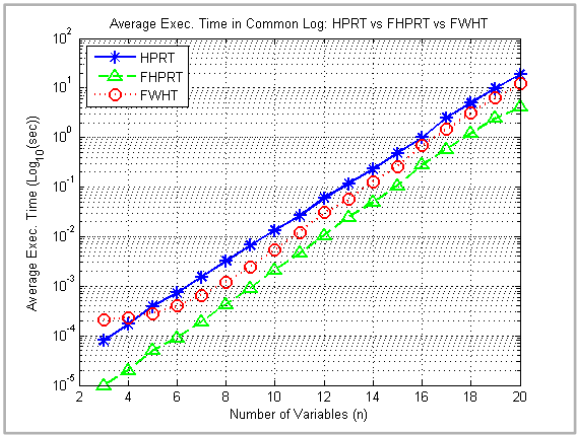


Fig. 9: Algorithms' Average Execution times (log(sec)) for 200 Iterations and $n \in [3,20]$

The advantage of the HRT is simply that, it gives one the freedom of determining how far they would want to go recursively and in turn flexibility in computations and complexity. The following considerations can be taken with regard to the HRT advantages:

1. **Local Property Flexibility and Linear Correlation:** Since the recursive transform can be applied locally then there is a computational flexibility. This transform is applied to each zones (defined by the degree l) of the Haar spectrum independently. Each zone is related to a correlation between the given Boolean function and a specific set of Linear Boolean functions (related to the degree l). This gives the flexibility of restricting computations locally depending on one's interest on finding the respective linear correlations.

2. **Local Property Flexibility and Testing:** Consider a case where the resulting Walsh-Paley spectrum is required to have a flat spectrum, where all the spectral coefficients have to be with the same absolute magnitude. The testing for such case-scenario becomes easier and computationally efficient since it can be conducted zone-wise and if the requirement is violated in one zone then there is a prevention of further unnecessary computations with the rest of the zones. The worst case scenario is when the violation is within the last zone of the spectrum. This advantage does not count only for flat-based spectrum, but rather the testing can be conducted by computing the spectrum partially and avoiding computing the entire spectrum first and then conduct the testing afterwards.

3. **Local Property Flexibility and the Possibility of Parallel Processing:** Since each zone of the Haar-spectrum is processed independent of the other zone, then this gives rise to the possibility of parallel processing being applied on each individual zone independently.

The following section presents the conclusion of the paper.

V. CONCLUSION

Spectral transforms play a crucial part in the analysis, design, and testing of digital devices. Such transforms based on the Walsh and Haar basis are the two most suitable for analysis and synthesis of switching or Boolean functions. The main contribution of this work is related to the Walsh-Paley transform from the Haar domain point of view.The paper has reviewed the connection between the Walsh-Paley and the Haar transforms. The work has introduced another alternative view on the connection between the two spectra. The paper derived the expression of the Walsh-Paley transform in terms of the Haar transform. In the process, the

work has demonstrated the possibility of obtaining both the Haar spectrum and the Walsh-Paley spectrum using only the Haar transform domain. In turn, the paper introduced a new Haar-based transform algorithm (Haar-Paley-Recursive Transform, HPRT) in the form of a recursive function along with its fast transform version called the fast-Haar-Paley-Recursive transform (FHPRT). The proposed algorithm coincides with the Walsh-Paley transform in terms of its interpretation. The consequence of this interpretation was then exploited in the work to derive the relationship between the Autocorrelation function (ACF) of a BF and the HPRT. This relationship has been given based on the Wiener-Khintchine theorem, and analogously the ACF has been expressed in terms of the Haar-Paley power spectrum. The paper then presented the simulation results on the average execution times of both derived algorithms in comparison to the existing Walsh benchmark. The work has shown the advantages of using the Haar transform domain in computing the Walsh-Paley spectrum.

VI. ACKNOWLEDGEMENT

The work presented here was partially funded by a grant from the IIUM Endowment Fund.

REFERENCES

[1] C.W. Thomas & S. Pantelimon, *Cryptographic Boolean functions and applications* (Academic Press, Elsevier Inc., 2009).

[2] C. Carlet, Boolean functions for cryptography and error correcting codes, in Y. Crama & P.L. Hammer (Eds.), *Boolean Models and Methods in Mathematics, Computer Science and Engineering* (Cambridge University Press, 2010) 257-397.

[3] C.K. Wu & D. Feng, *Boolean functions and their applications in cryptography* (Springer: Berlin Heidelberg, 2016).

[4] P.K. Sahu & A. Mishra, Haar wavelet and its application for problem solving in optimal control system, *Int. Journal of Engineering Research and Applications, 4(4),* 2014, 274-279.

[5] C.M. Patil & K.P. Paradeshi, Skin colour information and Haar feature based face detection, *Int. Journal of Engineering Research and Applications, 6(2),* February 2016, 72-75.

[6] S.J. Abbas & R. Manjhi, NSWT: Network security using Walsh table algorithms, *Int. Journal of Engineering Research and Applications, 3(2),* April 2013, 721-724.

[7] M.G. Karpovsky, R.S. Stanković & J.T. Astola, *Spectral logic and its applications for the design of digital devices* (John Wiley & Sons Inc., 2008).

[8] M.A. Thornton, D.M. Miller & R. Drechsler, Transformations amongst the Walsh, Haar, Arithmetic and Reed-Muller spectral domains, *Proc. 4th Intl. Workshop on Applications of Reed-Muller Expansion in Circuit Design,* 2001, 215-225.

[9] R.S. Stanković & B. J. Falkowski, The Haar wavelet transform: its status and achievement, *Computers and Electrical Engineering, 29(1),* 2003, 25-44.

[10] B.J. Falkowski & S. Rahardja, Walsh-like functions and their relations, *Proc. IEE Vision, Image and Signal Processing, 143(5),* 1996, 279-284.

[11] B.J. Falkowski & T. Sasao, Unified algorithm to generate Walsh functions in four different orderings and its programmable hardware implementations, *Proc. IEE Vision, Image and Signal Processing, 152(6),* 2005, 819-826.

[12] B.J. Fino, Relations between Haar and Walsh/Hadamard transforms, *Proc. IEEE, 60(5),* 1972, 647-648.

[13] M. Radmanović, R. Stanković & C. Moraga, Efficient calculation of the Autocorrelation of Boolean functions with large number of variables, *Facta Universitatis, Series: Electronics and Energetics, 28(4),* 2015, 597-609.

[14] H.M. Rafiq & M.U. Siddiqi, Haar transformation of linear Boolean functions, *Proc. IEEE International Conference on Signal Processing Systems,* Singapore, May 2009, 802-805.