FPGA Based Walsh and Inverse Walsh Transforms for Signal Processing

Zulfikar[1], Shuja A. Abbasi[2], A. R. M. Alamoud[2]
[1]*Department of Electrical Engineering, Syiah Kuala University, Darussalam, Banda Aceh 23111, Indonesia, phone: +62 651 7554336*
[2]*Department of Electrical Engineering, King Saud University, Riyadh, P.O. Box. 800, Saudi Arabia, phone: +966 1 4676752*
zulfikarsafrina@yahoo.co.id

***Abstract*—In this paper, we design and implement a set of Walsh transform and inverse Walsh transforms for signal processing. The Walsh and inverse Walsh transforms are designed to produce correct results for any input data combinations by providing sufficient word lengths at every steps of the design. Addition, subtraction and dyadic convolution processes have been chosen to demonstrate the performance of the designs. Detail word lengths designs in order to minimize the circuits are presented. It is found that the proposed Walsh transform structure is superior to many of the reported results when it is implemented on FPGAs in terms of area and speed.**

***Index Terms*—Digital integrated circuits, digital signal processing, discrete Fourier transforms, MATLAB.**

I. INTRODUCTION

Digital signal processing (DSP) is a well established area of research. The techniques for analysis, synthesis and processing of two or more digital signals have been established. However, certain tasks of DSP are difficult to be performed in time domain and hence the information has to be transformed to other domains [1].

Frequency domain is the most popularly used domain for the tasks which cannot be, or are difficult to be done in the time domain. Fourier transform is the most widely used technique for transforming the information from time domain to frequency domain. Discrete Fourier Transform (DFT) techniques for analyzing periodic digital signals exist, however the DFT technique is quite complex resulting in many problems during hardware realization and its use is justified only with very complex systems.

Orthogonal functions like Walsh Transform may also be used to analyze signals in frequency domain. The Walsh transform may be obtained using the Walsh functions [1], [2]. It is also well known that the Walsh functions may be evaluated using Rademacher functions [1]–[3]. Many scientists preferred this technique since the Rademacher functions may be conveniently realized using a counter [4], [5]. However, it should be noted that it is not always necessary to use the above technique and the Walsh transform (WT) may be performed just by using adders and subtractors [6]. This idea interested many scientists and engineers for hardware realization of the Walsh transform. This method is known as Fast Hadamard Transform (FHT) since it is derived from Hadamard matrices [7]–[10].

In order to further simplify the FHT, two types of structures - Distributed Arithmetic (DA) and Systolic Architechture (SA) have proposed by some wokers [7]–[10]. Amira et al proposed an improved structure and claimed better performance on the basis of an elaborate comparative study [11]. They also reported the results of power analysis. Later on, Meher and Patra [12] introduced a very simple technique based on combination of unified algorithm [6] and Rademacher functions. This technique is based upon the use of simple 4 points FHTs arranged in such a way that the higher points FHTs (8, 16, 32) may be obtained easily. The technique was also implemented on FPGAs. Superior results were claimed.

We found that the original Walsh functions, defined in terms of products of Rademacher functions can be used to transform the information into frequency domain faster than FHT and thus leads to speed up the DSP process. Therefore, the original Walsh transform technique based on Walsh functions and Rademacher functions is proposed and the results of hardware realization on FPGAs are presented. Further, we also designed and implemented the Inverse Walsh Transform (IWT) for conversion from frequency domain to time domain.

FPGA based hardware realization has been used to process two digital signals using the Walsh transform and the Inverse Walsh transform techniques. The results of hardware realization like the occupied area and delays have been compared with the results reported by other workers.

II. WALSH TRANSFORMS AND SIGNAL PROCESSING

We preferred definition of Walsh transform based upon derivation of Walsh functions from Rademacher functions which is found to be more appropriate for hardware implementation. The Rademacher functions are defined as follows [1]–[3]

$$\phi(n+1, x) = Sgn(\sin 2\pi 2^n x), \quad n = 0, 1, 2,...0 \le x < 1, \quad (1)$$

Manuscript received October 6, 2011; accepted May 17, 2012.
The authors gratefully acknowledge the financial support from National Plan for Science, Technology and Innovation (NPST), Saudi Arabia under project no. 09-ELE854-02.

where $\phi(0,x)=1$ and the signum function Sgn(y) is defined by

$$\mathrm{Sgn(y)}=\begin{cases}+1, & y\geq 0,\\ -1, & y<0.\end{cases} \tag{2}$$

The Walsh functions are defined in terms of product of Rademacher functions as [1]–[3]:

$$\psi(n,t)=\prod_{i=0}^{N}\left[\phi(i+1,t)\right]^{n_i},\quad n_i\in\{0,1\}, \tag{3}$$

$$n=\sum_{i=0}^{N}2^i n_i. \tag{4}$$

A signal x(t) of length N may be represented as a Walsh series given by [1]–[3]

$$x(t)=\sum_{n=0}^{N-1}A_n\psi(n,t). \tag{6}$$

The Walsh coefficients An are evaluated as [1]–[3]

$$A_n=\frac{1}{N}\sum_{k=0}^{N-1}x_k\psi(n,t). \tag{7}$$

If the signals h(t), p(t) and q(t) are defined as the summation, subtraction and multiplication of two signals given by:

$$h(t)=x(t)+g(t), \tag{8}$$

$$p(t)=x(t)-g(t), \tag{9}$$

$$q(t)=x(t)g(t), \tag{10}$$

where the function x(t) has the Walsh series expansion as in (5) and g(t) has the following Walsh series expansion

$$g(t)=\sum_{n=0}^{N-1}B_n\psi(n,t), \tag{10}$$

then the Walsh expansion of h(t), p(t) and q(t) are given by:

$$h(t)=\sum_{n=0}^{N-1}C_n\psi(n,t), \tag{11}$$

$$p(t)=\sum_{n=0}^{N-1}D_n\psi(n,t), \tag{12}$$

$$q(t)=\sum_{n=0}^{N-1}E_n\psi\psi(n,t), \tag{13}$$

where the expansion coefficients of C_n , D_n and E_n are computed as [3], [13], [14]:

$$C_n=A_n+B_n, \tag{14}$$

$$D_n=A_n-B_n, \tag{15}$$

$$E_n=\sum_{m=0}^{\infty}A_{n\oplus m}B_m, \tag{16}$$

where $\oplus$ refers to dyadic addition (XOR) and the last expression is the well known dyadic convolution.

III. Design of Walsh and Inverse Walsh Transforms

A set of circuits which perform signal processing of two digital signals is shown in Fig. 1. The design consists of two Walsh transform blocks, a DSP block, and one Inverse Walsh transform block. Input digital signals x(t) and g(t) are passed into the system serially. Similarly, the output signal is also produced in series. Two identical Walsh transform blocks are used in order to speed up the transform process.

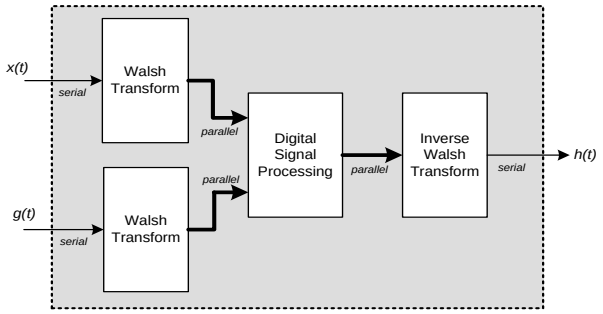

Fig. 1. Walsh and inverse Walsh transforms for signal processing.

IV. Walsh Transform

The previous Walsh transform methods arrange outputs and inputs in parallel or series. We prefer combinations of this arrangement. Table I shows comparison of the proposed method to the previous methods in term of inputs and outputs arrangement.

TABLE I. COMPARISON OF INPUT AND OUTPUT ARRANGEMENTS OF WALSH TRANSFORMS CIRCUITS.

	Input	**Output**
Proposed	Serial	Parallel
[9], [12]	Parallel	Parallel
[9], [11], [15], [16], [17]	Serial	Serial

By passing input data serially will obviously require less numbers of pins when the circuit is implemented. Meanwhile, because of signal processing purposes, the outputs of WT are arranged in parallel.

The basic blocks of the proposed WT circuit, shown in Fig. 2, consists of:

- Negative circuit (WI bits) – One no;
- Walsh circuit (N-1 order) – One no;
- 2 to 1 multiplexers (WI bits) – (N-1) nos;
- Accumulators (WO bits) – N nos;
- Data buffers (WI bits) – N nos;
- Output buffers (WO bits) N nos.

N input data (samples) X are passed into the circuit serially and they are controlled by Enter signal. Walsh circuit is used to select the suitable data X or –X and pass it through the multiplexers. The outputs of the multiplexers will be accumulated at the accumulators and they will form the output transformed coefficients (A's).

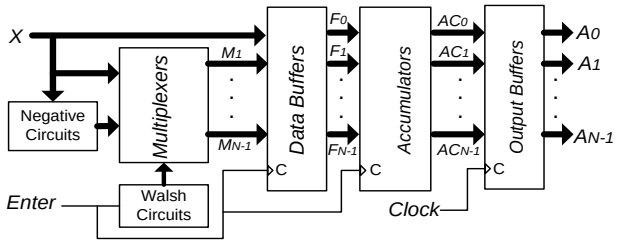

Fig. 2. Design of Walsh transform for transform length of N.

Fig. 3 views circuit realization of the proposed Walsh transform for N=4 and input word lengths WI=4 (used to represent input data X). The output transformed coefficients (A's) are represented in 6 bits (output word lengths WO=6) [13].

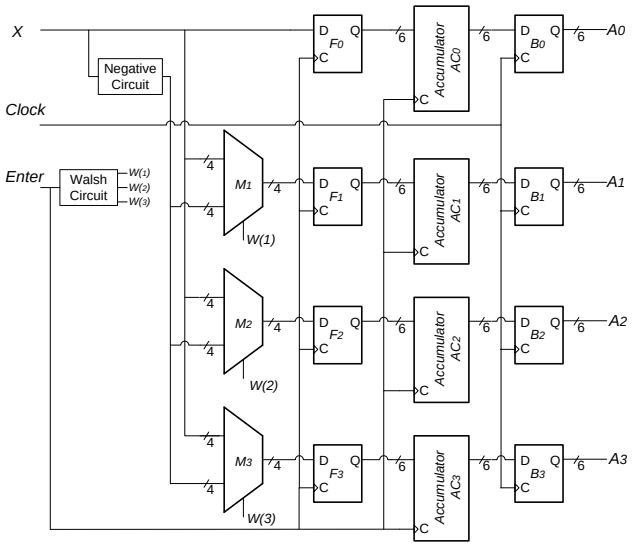

Fig. 3. Circuit realization of WT for N=4 and WI=4.

In determining coefficients A's, and to avoid floating numbers, factor 1/N in (6) is ignored for the time being. This factor will be added towards the end of the process of Inverse Walsh transform circuit.

Inverse Walsh Transforms

The input data of inverse Walsh transform are new coefficients as results processing of Walsh transform coefficients. Thus, the IWT is designed to receive inputs in parallel and produce outputs serially.

The basic blocks of the proposed inverse Walsh transform circuit are shown in Fig. 4. The design consists of:

- Negative circuit (WIC bits) – (N-1) nos;
- Walsh circuit (N-1 order) – One no;
- 2 to 1 multiplexers (WIC bits) – (N-1) nos;
- Adders (WOC bits) – (N-1) nos;
- Data buffer (WIC bits) – (N-1) nos;
- Output buffer (WOO bits) – One no.

Fig. 5 views circuit realization of the proposed Inverse Walsh transform for N=4 and input data (coefficients) word lengths WIC=6 (used to represent input coefficient C's). It is assumed here that no DSP process has been done before IWT circuit or the system is assumed under signal generation mode. The output data (H) are represented in 4 bits (output word lengths WOO=4).

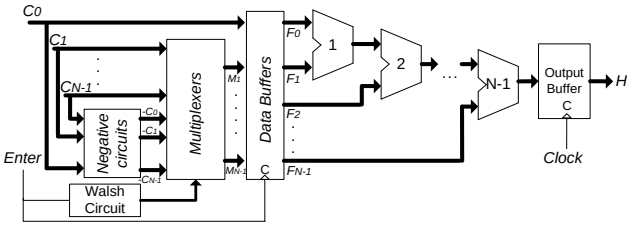

Fig. 4. Design of inverse Walsh transform for transform length of N.

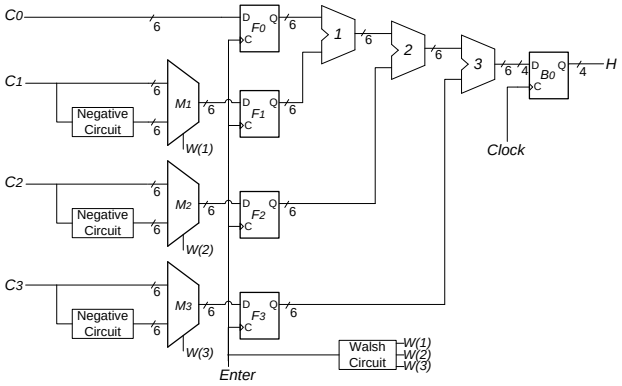

Fig. 5. Circuit realization of IWT for N=4 and WI=6.

V. DESIGN OF WORD LENGTHS

The proposed Walsh and inverse Walsh transforms are designed as part of complete system for signal processing. Therefore, in order to achieve efficient hardware utilization, it is necessery to carefully design the word lengths for all steps of processes.

Word lengths to represent the output information of Walsh transform may be chosen directly from equation 6. Consequently, the sufficient word lengths to represent coefficients A (WO) are as follow

$$WO = WI + \log_2(N). \tag{17}$$

This number of bits will guarantee that any input data combinations can be accommodated by coefficients A's.

The processed coefficients C, D and E described before have to be represented in different number of bits (word lengths) according to DSP types. It needs special care to determine the right number of bits to represent these coefficients.

For addition and subtraction processes, the word lengths WIC of the processed coefficients C and D are equal. These word lengths are evaluated as follows

$$WIC = WO + 1. \tag{18}$$

For multiplication, the word lengths WIC of the processed coefficients E can be estimated directly from (16) which gives teh following expression

$$WIC = \log_2(N) + 2\{(WI - 1 + \log_2(N)\} + 1. \tag{19}$$

However, in order to obtain a more efficient area required, the WIC according to (19) is more carefully analyzed using MATLAB. It was found that, in practice, the processed

coefficients E do not require the amount of bits that are suggested by that equation. Instead, the amount of bits given by the following equation is enough

$$WIC = 2\{(WI - 1 + \log_2(N)\} + 1. \quad (20)$$

For inverse Walsh transform, it is assumed that input coefficients C are represented by WIC bits for transform length N. The maximum number of bits required, WOC may be obtained by

$$WOC = WIC - \log_2(N). \quad (21)$$

However, again in order to obtain a more efficient resource uilization, the maximum bits required according to (21) and corresponding to Walsh transform is more carefully analyzed by providing all possible input values in MATLAB. It was found that, in practice, the maximum bits for performing IWT do not require the number suggested by (21). Instead, the number given by the following equation is enough

$$WOC = WIC. \quad (22)$$

It should be noted, the input coefficients of IWT have certain patterns and donot take all possible values like the input signal x(t) and g(t).

The output signal h(t) of inverse Walsh transform has to be represented in WOO bits as given in (23)

$$WOO = WOC - \log_2(K). \quad (23)$$

The subtraction factor $\log_2(K)$ is due to ignoring factor of 1/N in performing Walsh transform. Where K depends on the digital signal processing types performed before IWT. For generation, addition and subtraction processes, K=N and for multiplication process, $K=N^2$.

Table II summarizes all word lengths required for signal generation, addition, subtraction and multiplication processes of transform lengths N and input word lengths WI.

TABLE II. WORD LENGTHS DESIGN FOR TRANSFORM LENGTHS N AND INPUT WORD LENGTHS WI.

DSP types	**WO**	**WIC = WOC**	**WOO**
Signal Generation	WI+$\log_2$(N)	WIC=WO	WI
Addition/ Subtraction	WI+$\log_2$(N)	WO+1	WI+1
Multiplication	WI+$\log_2$(N)	2{(WI–1+$\log_2$(N)}+1	2WI-1

VI. WALSH CIRCUIT

One of the most useful properties of the Rademacher functions and Walsh functions is that they take only two values +1 and –1. Hence they are ideal for implementation on digital systems. If the ±1 amplitudes of the Walsh functions are converted to a binary logic {0,1} representation with the conversions +1 → "0" and −1 → "1", then multiplication of Rademacher functions is reduced to Exclusive-OR (XOR) or modulo-2 addition operation.

The Walsh circuit realization for transform lengths N=4 and N=8 are shown in Fig. 6. The circuit is designed to produce 2nd, 3rd and 4th orders of Walsh functions based on Hadamard ordering.

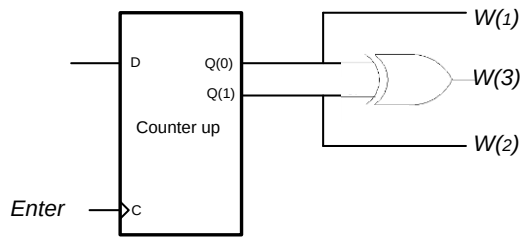

Fig. 6. Walsh Circuit realization for N=4.

The 1st order of Walsh Hadamard functions or any Walsh ordering is always +1 → "0", therefore, it is not necessary to generate it.

VII. ARRANGEMENT OF NET CONNECTIONS

In the Walsh transform circuit of Fig. 3, the accumulator AC_1 is used to accumulate input data which are made to enter into data buffer F_1. Accumulators are designed to accommodate data up to WO=6 bits. Since the output of data buffers are only 4 bits, net of $F_1(3)$ is to be connected to three inputs of the accumulator AC_1 as shown in the circuit in Fig. 7. For performing Walsh transform with higher transform lengths such as N=8, the nets have to be connected to 4 accumulator inputs. These connections are based on additional bits require as derived from (17).

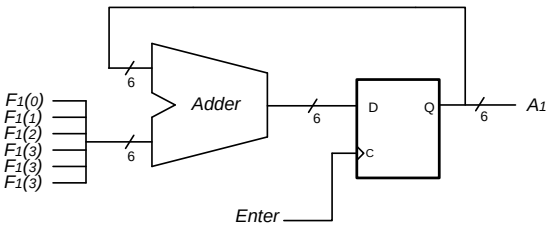

Fig. 7. Internal circuit of accumulator AC_1 and net connections.

In case of inverse Walsh transform circuit in Fig. 5, only 4 of 6 nets of adder 3 are connected to the buffer B_0, nets of $A_3(0)$ and $A_3(1)$ are ignored (not used). This net manipulation is due to subtraction factor K=N=4. Fig. 8 shows net connections between adder 3 and buffer B_0.

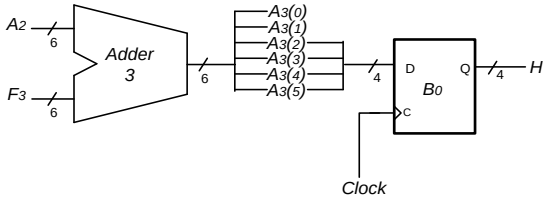

Fig. 8. Net connections between adder 3 and ouput buffer B_0.

VIII. IMPLEMENTATION RESULTS

The implementations are targeted to Virtex chips from Xilinx. Fig. 9 shows Walsh transform of input signal x(t).

Fig. 10 views result of inverse Walsh transforms of coefficeients C's. Where C's are addition results of coefficients A's and B's (14). These simulations have been done using ISE Simulator under Windows 7 on intel Core Duo processor (T2050) computer.

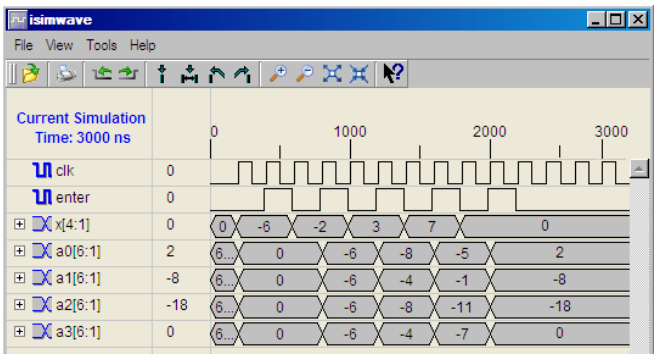

Fig. 9. Walsh transform of signal x(t).

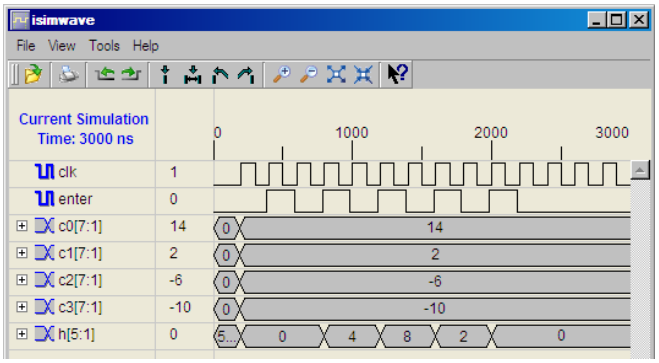

Fig. 10. Inverse Walsh transform of addition coefficients C's.

Table III shows the numerical results of all signals and coefficients. These values are represented in signed number format.

TABLE III. LIST ALL SIGNALS AND COEFFICIENTS.

Signal/ Coefficients	1	2	3	4
x(t)	-6	-2	3	7
g(t)	6	6	5	-5
A	2	-8	-28	0
B	12	10	12	-10
C	14	2	-6	-10
D	-10	-18	-30	10
E	-272	104	-112	-296
h(t)	0	4	8	2
p(t)	-12	-8	-2	12
q(t)	-36	-12	15	-35

The coefficients C's and D's are represented in WIC=7 bits based on (18). Meanwhile, the coefficients E's are represented in WIC=11 bits based on (20). Implementation of signal generation is not shown here because it may be obtained just by combining Walsh transform and inverse Walsh transform (see Table II for signal generation).

In order to make a comparison with the results reported by other works, we implemented Walsh transform for N=4, 8, 16 and WI=8 on Xilinx Virtex-E, Virtex-IIP and Virtex-4. The results of comparison are presented in Table IV.

TABLE IV. COMPARISON THE PROPOSED WALSH TRANSFORM TO PREVIOUS METHODS FOR VARIOUS XILINX VIRTEX SERIES (WI=8).

FPGA Platform	N	Proposed		Structure [12]		Structure [11]	
		Slices	Speed (MHz)	Slices	Speed (MHz)	Slices	Speed (MHz)
Virtex-E	4	37	185	32	179	82	127
	8	84	167	96	203	162	115
	16	167	191	256	205	335	67
Virtex-IIP	4	37	370	32	294	83	204
	8	83	357	96	283	163	183
	16	163	335	256	288	377	100
Virtex-4	4	37	586	32	294	82	227
	8	83	573	96	471	163	212
	16	165	530	256	418	358	121

It may be seen that the proposed structure for the Walsh Transform is superior to the others except for N=4, in which case, the area occupied is a bit higher than the one proposed by Meher et al [12].

We also implemented the Inverse Walsh transform for N=4, 8, 16 and WO=8 on Xilinx Virtex-IIP and Virtex-4 based on Hadamard ordering. Table VI shows the results of these implementations.

TABLE V. IMPLEMENTATION OF INVERSE WALSH TRANSFORM TO VARIOUS XILINX VIRTEX SERIES (WO=8).

FPGA Platform	N	Inverse Walsh Transform	
		Slices	Speed (MHz)
Virtex-IIP	4	40	630
	8	108	468
	16	275	437
Virtex-4	4	40	835
	8	108	636
	16	276	628

IX. CONCLUSIONS

FPGA based hardware realization of Walsh and Inverse Walsh transforms for signal processing has been proposed and demonstrated on various Xilinx Virtex chips. Three types of DSPs have been chosen in order to demonstrate the usefulnes of the proposed Walsh and inverse Walsh transforms. In general, it may be concluded that the proposed Walsh transform structure is superior to many of the structures reported in literature.

It is possible to design a system that performs signal generation, addition, subtraction, multiplication or other types of digital signal processing on a single chip. Also it is possible for future work to control the input signals by clock, instead of signal Enter.

REFERENCES

[1] M. G. Karpovsky, R. S. Stankovic, J. T. Astola, *Spectral Logic and Its Applications for The Design of Digital Devices*. New Jersey, John Wiley & Sons Inc., 2008.

[2] K. G. Beauchamp, *Applications of Walsh and Related Functions – With an Introduction to Sequency Theory.* Academic Press, 1984.

[3] M. Maqusi, *Applied Walsh Analysis*, 1st ed., London, Heyden and Son Ltd., 1981.

[4] A. M. A. Bin Ateeq, S. A. Abbasi, A. R. M. Alamoud, "Hardware Realization of Walsh Functions and Their Applications Using VHDL

and Reconfigurable Logic", *ICM – Beirut, Lebanon,* 2002, pp. 58–61.

[5] S. M. Qasim, S. A. Abbasi, "Single Chip FPGA Based Realization of Arbitrary Waveform Generator using Rademacher and Walsh Functions", in *Proc. of the Second International Conference of Emerging Technologies (ICET)*, Peshawar, Pakistan, 2006, pp. 205–210.

[6] B. J. Fino, V. R. Algazi, "Unified Matrix Treatment of the Fast Walsh-Hadamard Transform", *IEEE Transactions on Computers*, vol. 42, pp. 1142–1146, 1976. [Online]. Available: http://dx.doi.org/10.1109/TC.1976.1674569

[7] A. Amira, A. Bouridane, P. Milligan, M. Roula, "An FPGA Implementation of Walsh-Hadamard Transforms for Signal Processing", *in Proc. of IEEE International Conference on Acoustic, Speech and Signal Processing*, 2001, pp. 1105–1108.

[8] A. Amira, A. Bouridane, P. Milligan, "A Novel Architecture of Walsh Hadamard Transform using distributed Arithmetic Principles", *in Proc. of The 7th IEEE International Conference on Electronics, Circuit & Systems (ICECS'2K)*, Beirut, Lebanon, 2000.

[9] A. Amira, A. Bouridane, P. Milligan, M. Roula, "Novel FPGA Implementations of Walsh-Hadamard Transforms for Signal Processing", *in Proc. of IEE Vision, Image, and Signal Processing*, 2001.

[10] B. J. Falkowski, T. Sasao, "Unified Algorithm to Generate Walsh Functions in Four Different Orderings and Its Programmable Hardware Implementations", *in Proc. of IEE on Vision, Image and Signal Processing*, 2005, pp. 819–826.

[11] A. Amira, S. Chandrasekaran, "Power Modeling and Efficient FPGA Implementation of FHT for Signal Processing", *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 15, no. 3, pp. 286–295, 2007.

[12] P. K. Meher, J. C. Patra, "Fully-Pipelined Efficient Architechtures for FPGA Realization of Discrete Hadamard Transform" *in Proc. of International Conference on Application-Specific Systems, Architechture and Processors (ASAP 2008)*, 2008, pp. 43–48.

[13] M. Y. Zulfikar, S. A. Abbasi, A. R. M. Alamoud, "FPGA Based Analysis and Multiplication of Digital Signals", *in IEEE Proc. of Second International Conference on Advances in Computing, Control and Telecommunication Technologies (ACT 2010), Jakarta*, Dec., 2010, pp. 32-36.

[14] M. Y. Zulfikar, S. A. Abbasi, A. R. M. Alamoud, "FPGA Based Processing of Digital Signals using Walsh Analysis", *in IEEE Proc. of International Conference on Electrical, Control and Computer Engineering (INECCE 2011)*, Pahang, Malaysia, Jun. 2011, pp. 440–444.

[15] L. W. Chang, M. C. Wu, "A bit level systolic array for Walsh–Hadamard transforms", *Transaction on Signal Processing*, vol. 31, no. 3, pp. 341–347, 1993. [Online]. Available: http://dx.doi.org/10.1016/0165-1684(93)90091-N

[16] S. Y. Kung, *VLSI Array Processors*. Englewood Cliffs, NJ, Prentice-Hall, 1988.

[17] S. S. Nayak, P. K. Meher, "High throughput VLSI implementation of discrete orthogonal transforms using bit-level vector-matrix multiplier", *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.* vol. 46, no. 5. pp. 655–658, 1999.