Processing of Multiple Digital Signals Based on Real-time Walsh Transform

(IJCSNT) International Journal of Communication Systems and Network Technologies
ISSN Number: 2053-6283
Volume 1, Issue No. 2, 101–115

https://journals.mirlabs.in/index.php/ijcsnt
DOI: 10.18486/ijcsnt/1.2.009
IJCSNT

Zulfikar[1], Shuja A. Abbasi[2], A. R. M. Alamoud[3]

Abstract

In this paper, the design and implementation of processing of multiple digital signals using real-time Walsh transforms is presented. The design of real-time Walsh is able to produce output instantly even before all input data have been given to the system. The design system consists of Walsh Transforms, some Digital Signal Processing (DSP) circuits, and inverse Walsh transform. The real-time Walsh and inverse Walsh transforms are also designed to produce right results for all possible combinations of input data. DSP block is designed to perform addition, subtraction, and dyadic convolution process of Walsh coefficients of two or more digital signals. Comparisons to the previous methods are presented in details. It was found that the design of real-time Walsh transform structure performs better than many of the reported results in the literature.

Keywords

Fast Hadamard Transform, Real-time Walsh Transform, Walsh Transform, Real-time Inverse Walsh Transform, Inverse Walsh Transform, Walsh Functions, Walsh Series, Walsh Coefficients, Rademacher Functions, Dyadic Convolution.

Received: 28 May 2012; **Revised:** 30 July 2012; **Accepted:** 25 August 2012;
Published: 30 August 2012;

[1]Department of Electrical Engineering, Syiah Kuala University, Banda Aceh, Indonesia.
[2, 3]Department of Electrical Engineering, King Saud University, Riyadh, Saudi Arabia.

Corresponding author:
Email: zulfikarsafrina@yahoo.co.id

Introduction

At the present, 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, some of them required certain tasks of DSP which are difficult to be performed in time domain and hence the information has to be transformed to other domains.

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 information from time domain to frequency domain. No doubt that this is a very useful tool in DSP. In the group of 2^m transform length, the Fourier transform is better described as the Walsh transform.

The Walsh transform (WT) may be obtained using the Walsh functions. 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]. Distributed Arithmetic (DA) and Systolic Architecture (SA) are two types of structures that have been proposed in order to further simplify the FHT[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.

It is 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. Moreover, a good design of Walsh transform is able to produce output immediately. 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.

The paper is organized as follows. In section II, the basic theory of Walsh transform and signal processing is presented. Section III deals with the proposed structure of real time Walsh transform and inverse Walsh transform. Section IV deals with control signals of the design system. The hardware implementation on FPGAs is presented in section V. Section VI deals with analysis and discussions of the proposed Walsh and inverse Walsh transforms. Conclusions are presented in Section VII.

Walsh Transform and Signal Processing

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

$$\phi(n+1,x) = Sgn(\sin 2^n 2\pi x), \quad n = 0,1,2,\ldots\ ;\ 0 \le x \le 1 \tag{1}$$

Where, and the signum function $Sgn(y)$ is defined by:

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

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

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

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

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

$$x(t) = \sum_{n=0}^{N-1} A_n \psi(n,t) \tag{5}$$

The Walsh coefficients A_n are evaluated as[1–3]:

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

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

$$h(t) = x(t) + g(t) \tag{7}$$

$$p(t) = x(t) - g(t) \tag{8}$$

$$q(t) = x(t)g(t) \tag{9}$$

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[1]:

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

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

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

Where the expansion coefficients of C_n, D_n and E_n are computed as[1,13,14]:

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

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

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

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

Design of Complete Set of Real-Time Walsh and Inverse Walsh Transforms

Fig. 1 shows complete set of circuits which perform signal processing of multiple digital signals. The design consists of Walsh transform blocks, a DSP block, one Inverse Walsh Transform block and a dividing block. Input digital signals $x_1(t)$, $x_2(t)\ldots$etc are passed into the system serially. Similarly, the output signal $h(t)$ is also produced in series.

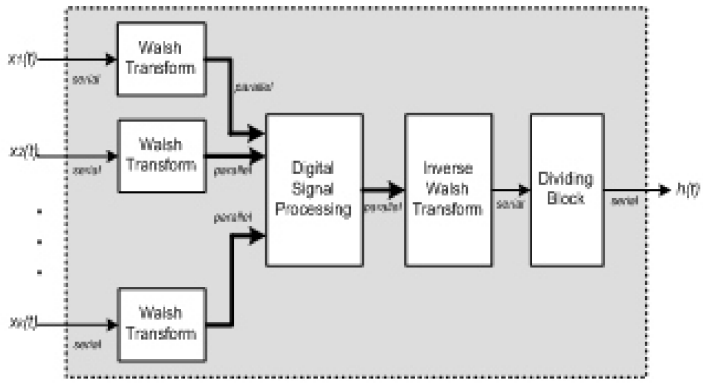

Figure 1. Walsh and Inverse Walsh for signal processing

Numbers of Walsh transform blocks used here depend on number of input signals used. Fig 2 Views basic design of the Walsh transform. The design consists of a negative circuit, a circuit to produce (N-1) orders of Walsh functions, a block of N-1 number of 2 to 1 multiplexers, N data buffers, N accumulators, and N output buffers.

Input data (samples) of X are passed serially. Walsh circuit is used to select the suitable data X or X and pass it through the multiplexers. The outputs of the multiplexers will accumulate at the accumulators and

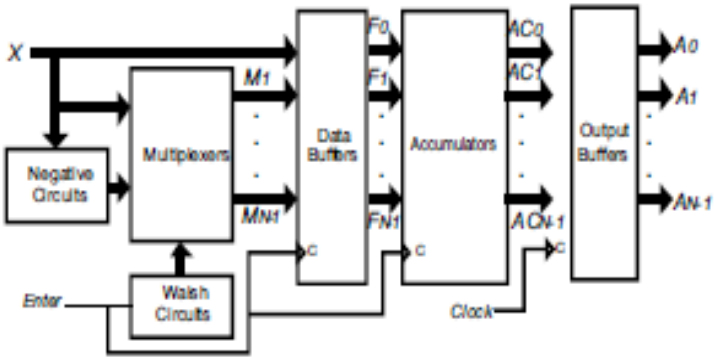

Figure 2. Design of Walsh transform for transform length of N

they will form the output transformed coefficients (A's). 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) have to be represented in 6 bits (output word lengths WO=6)[13].

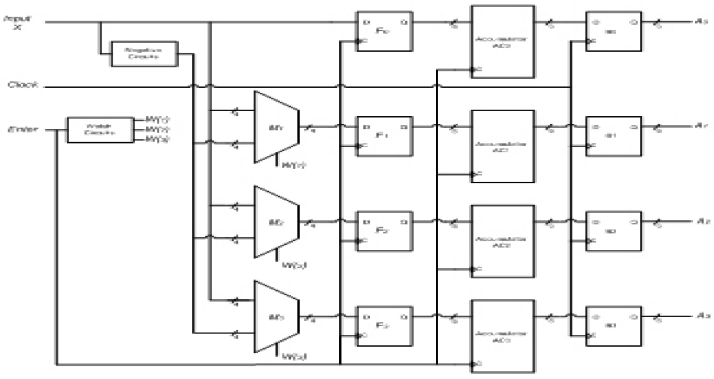

Figure 3. Circuit realization of WT for N=4 and WI=4

The proposed WT is able to produce output results instantly even before all input data have been given to the system. For example, input data of N=4 are X{-6, -2, 3, 7}, WT (Walsh coefficients) of these data can be calculated as follows,

```
A0 = (-6) + (-2) + 3 + 7 = 2
A1 = (-6) - (-2) + 3 - 7 = -8
A2 = (-6) + (-2) - 3 - 7 = -18
A3 = (-6) - (-2) - 3 + 7 = 0
```

When input data X have not completely entered into the system, say only -6 and -2 have just passed, the output Walsh coefficients are:

```
A0 = (-6) + (-2) + 0 + 0 = -8
A1 = (-6) - (-2) + 0 - 0 = -4
A2 = (-6) + (-2) - 0 - 0 = -8
A3 = (-6) - (-2) - 0 + 0 = -4
```

Those coefficients are true by assuming the rest of input data are all zero. For transform lengths N, the coefficients will be produced N times. Therefore, we call the design of Walsh transform as real-time Walsh transform.

The Walsh circuit realization for N=4 is shown in Fig. 4. It is designed to produce complement of 2nd, 3rd and 4th orders of Walsh functions based on Hadamard ordering. For higher transform lengths, obviously it will require more XOR gates in order to perform multiplications of Rademacher functions.

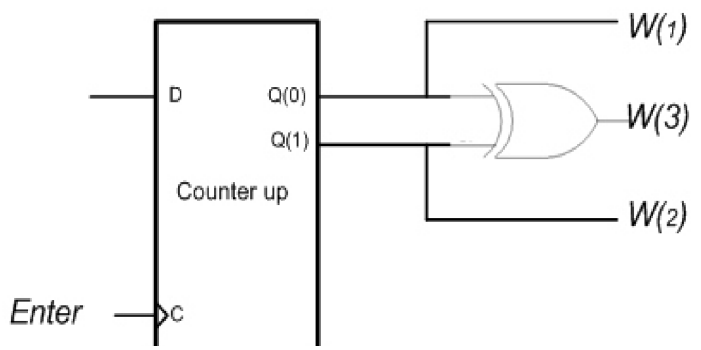

Figure 4. Walsh Circuit realization for N=4

The inverse Walsh transform may be obtained using most of the blocks designed Walsh transform. The complete block diagram is shown in Fig. 5. The circuit requires a set of N negative circuits to receive N parallel inputs, (N-1) number of 2 to 1 multiplexers, N data buffers, (N-1) adders, and an output buffer.

Fig. 6 views circuit realization of the proposed Inverse Walsh transform for N=4 and input word lengths WI=6 (used to represent input coefficient C's). The output data (H) are represented in 4 bits (output word lengths WO=4). This (WO¡WI) is due to dividing by factor of N=4.

The process flow of inverse Walsh transform is similar to Walsh transform, all input data either C or C are selected to pass the multiplexers based on outputs of Walsh circuits. These data will then be added one by one using the adders sequentially.

In determining coefficients A's and B's, and to avoid floating numbers, factor 1/N in Eq. 6 is ignored for a while. This factor will be implemented towards the end of the process in dividing block. The circuits to perform this job are installed together with the Inverse Walsh transform circuit. For addition and subtraction, the outputs of inverse Walsh transform have to be divided by N. Meanwhile, for multiplication process, the outputs should be divided by factor of N^2 [13].

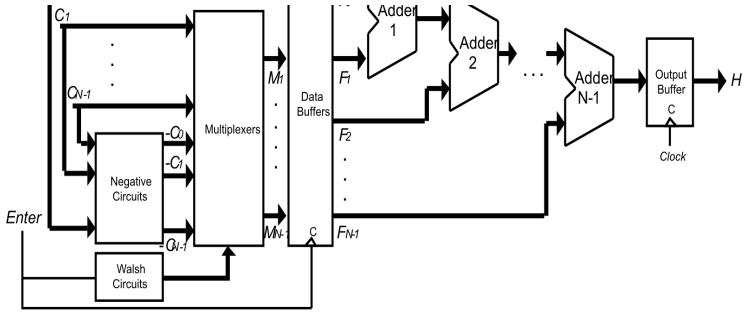

Figure 5. Design of inverse Walsh transform for transform length of N

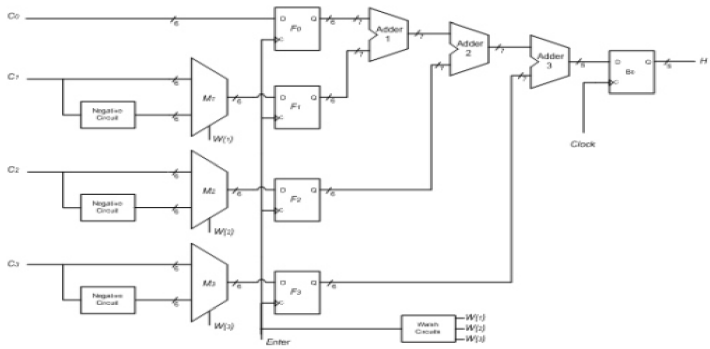

Figure 6. Circuit realization of IWT for N=4 and WI=6

Circuit Control Signals

The operation of the proposed WT and IWT has to be controlled carefully in order to gather optimum results. There are three basic control signals as follows:

- Clock; it is used to synchronize buffers and other storage elements.
- Reset; it is used to reset (clear) all buffers and storage elements. Before input data are fed to the circuit, signal Reset normally goes high.
- Enter; it is used to control input data. How fast the data enters the system, is based upon this signal. This signal has to be high for at least twice of clock period,

$$T_{Enter} \geq 2\,T_{Clock} \tag{17}$$

Therefore, this signal is non periodic. Fig. 7 shows relation between signal Enter and Clock.

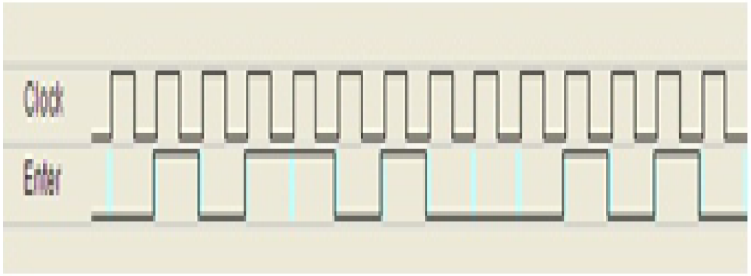

Figure 7. Relation between Clock and signal Enter

The availability of this signal is based on the availability of input data. The signal arrives just after the input data (X) available at input port of the circuits. Fig. 8 shows relation between signal Enter and input data.

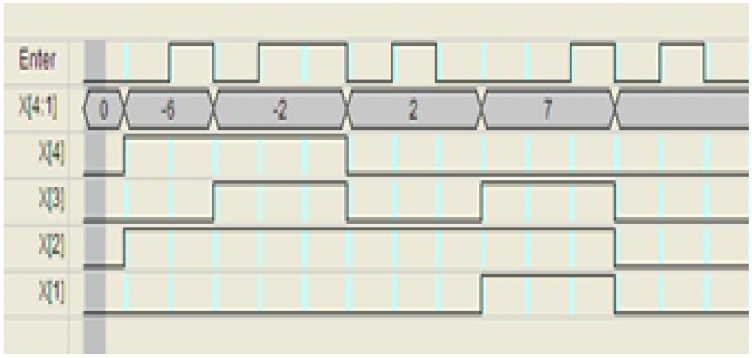

Figure 8. Relation between signal Enter and input data (X)

Implementation Results

In order to show step by step process of the design on chip, the implementation will be performed and displayed for every step (Walsh transform, DSP, Inverse Walsh transform) and limited to two input signals only. Three types of signal processing steps viz. addition, subtraction and multiplication will be demonstrated for transform length N=4 and word lengths of input signals WI=4 based on natural or Hadamard ordering. The implementations are targeted to Virtex chips from Xilinx.

Figs. 9 and 10 show Walsh transforms of input signals $x(t)$ and $g(t)$. The transformed coefficients A and B are represented in word lengths WO=6 bits.

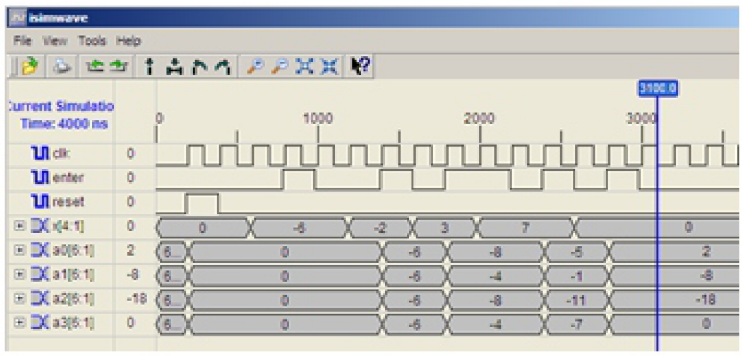

Figure 9. Walsh transform of signal x(t)

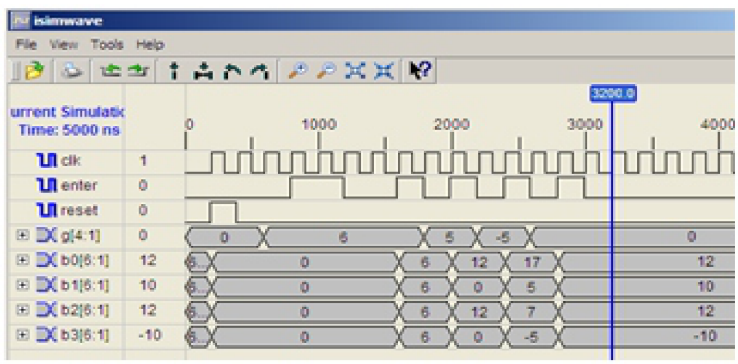

Figure 10. Walsh transform of signal g(t)

Output coefficients A and B are produced in real-time by the system. Here, the coefficients are produced four times. The first coefficients are available just after two cycles of signal Enter. The second and the third series of coefficients are available after three and four cycles of signal Enter respectively. The last coefficients will be available after signal Enter goes high five times.

Since the Walsh coefficients for A's and B's are available, we may now perform addition, subtraction and multiplication of signal $x(t)$ and $g(t)$. As mentioned earlier in section II, the addition and subtraction can be performed by simply adding and subtracting both coefficients A's and B's as follow[13]:

Addition process:

```
C0 = A0 + B0 = 2 + 12 = 14
C1 = A1 + B1 = -8 + 10 = 2
C2 = A2 + B2 = -18 + 12 = -6
C3 = A3 + B3 = 0 + (-10) = -10
```

Subtraction process:

```
D0 = A0 - B0 = 2 - 12 = -10
D1 = A1 - B1 = -8 - 10 = -18
D2 = A2 - B2 = -18 - 12 = -30
D3 = A3 - B3 = 0 - (-10) = 10
```

The coefficients C's and D's are then transformed back into the time domain by using IWT proposed in Fig. 6. The results are shown in Figs. 11 and 12 for addition and subtraction of signal $x(t)$ and $g(t)$ respectively.

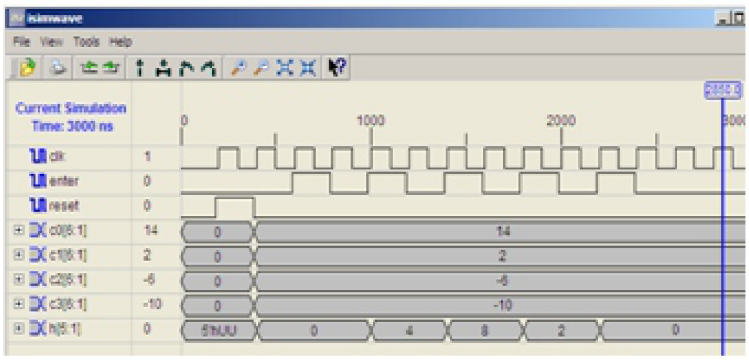

Figure 11. Inverse Walsh transform of addition coefficients A's and B's

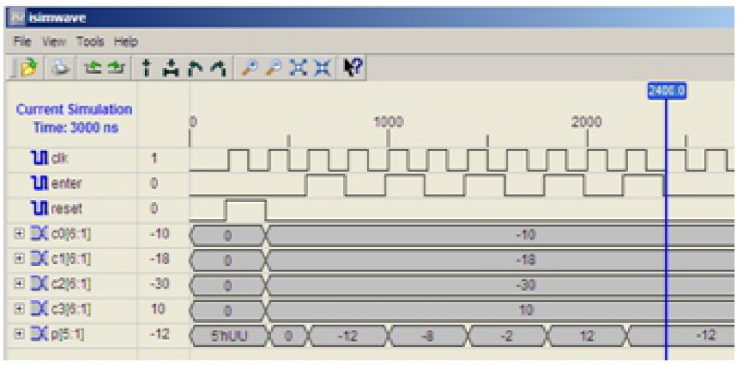

Figure 12. Inverse Walsh transform of subtraction coefficients A's and B's

For multiplication process, we need to perform dyadic convolution between coefficients A's and B's in order to find Coefficients E's[14]. Manual calculations for this process based on Eq. 16 are listed below:

```
E0 = A0*B0 + A1*B1 + A2*B2 + A3*B3
   = 2*12 + (-8)*10 + (-18)*12 + 0*(-10) = -272
```

```
E1 = A1*B0 + A0*B1 + A3*B2 + A2*B3
   = -8*12 + 2*10 + 0*12 + (-18)*(-10) = 104
E2 = A2*B0 + A3*B1 + A0*B2 + A1*B3
   = -18*12 + 0*10 + 2*12 + (-8)*(-10) = -112
E3 = A3*B0 + A2*B1 + A1*B2 + A0*B3
   = 0*12 + (-18)*10 + (-8)*12 + 2*(-10) = -296
```

The FPGA implementation of this convolution is shown in Fig. 13. These coefficients are then passed to the inverse Walsh circuit and the result signal $h(t)$ is shown in Fig. 14.

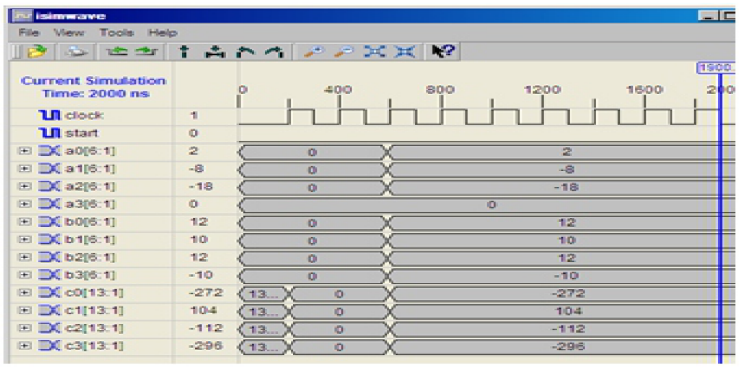

Figure 13. Dyadic convolution between coefficients A's and B's

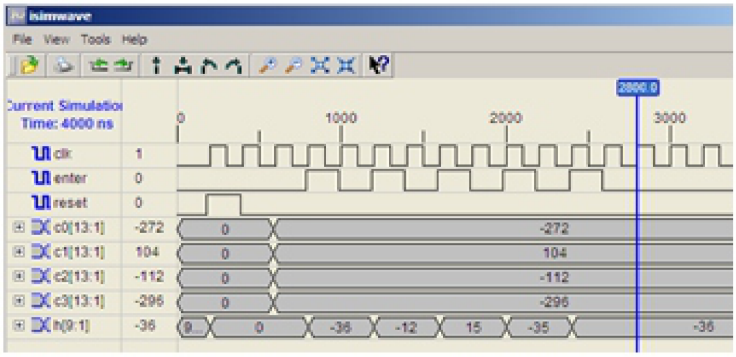

Figure 14. Inverse Walsh transform of dyadic convolution between coefficients A's and B's

To summarize the implementation of the proposed design, all the signals and coefficients are listed in Table I.

Table 1. 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

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

Table 2. Comparison of Area, Speed and Word Lengths for Walsh Transform, Inverse Walsh And Dyadic Convolution

Process	**Slice**	**Flip Flops**	**LUTs**	**Speed (MHz)**	**WI (bit)**	**WO (bit)**
Walsh Transform	20	39	36	636	4	6
Inverse Walsh Transform#	27	20	46	828	6	5
Inverse Walsh Transform^	74	43	116	816	13	9
Dyadic Convolution	428	–	732	–	6	13

In the design of the proposed real-time Walsh transform and its inverse, special care is needed for the selection of word lengths representing input signals, coefficients and the output signal. This requires detailed analysis of each step of the processes. The implementation results shown in the previous section clearly indicate that the word lengths for different coefficients and output signals vary.

The design is implemented on Xilinx Virtex-4 chip (xc4vlx15-12-sf363). Various results are shown in Table II.

Analysis and Discussion

In order to make a comparison with the results reported by other works, we implemented real-time Walsh transform for N=4, 8, 16 and WI=8 on Xilinx Virtex-IIP and Virtex-4.

Table 3. Comparison the proposed Walsh Transform to previous methods for various Xilinx Virtex series (WI=8)

FPGA Platform	N	Proposed		Structure[11]		Structure[12]	
		Slices	Speed	Slices	Speed	Slices	Speed
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 in[11].

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 IV shows the results of these implementations.

Table 4. Implementation of Inverse Walsh transform to various Xilinx Virtex series (WO=8)

FPGA Platform	N	Slices	Speed (MHz)
Virtex-IIP	4	37	370
	8	83	357
	16	163	335
Virtex-4	4	37	586
	8	83	573
	16	165	530

Conclusions

A complete set of real-time Walsh and Inverse Walsh transforms for signal processing has been proposed and demonstrated on various Xilinx Virtex chips. The design consists of real-time Walsh transforms, inverse Walsh transform, DSP block and dividing block.

Real-time Walsh transform is controlled by independent, random signal Enter. The availability of this signal is based on the availability of data in input ports of the design system. It requires N+1 cycles of signal Enter for completing the process of transforming input data to the frequency domain.

Generally, it may be concluded that the proposed Walsh transform structure is superior to many of the structures reported in literature. The Inverse Walsh transform, requires more area but is faster than the Walsh transform itself. The results clearly show that the DSP (Dyadic convolution) for multiplication process requires 7 times larger area than other processes.

When combining the proposed real-time Walsh transform, DSP block and inverse Walsh transform into a single system, the designed inverse Walsh transform is fully controlled by Clock, instead of signal Enter.

Acknowledgments

The authors gratefully acknowledge the financial support from National Plan for Science, Technology and Innovation (NPST), Saudi Arabia under project no. 09-ELE854-02.

References

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

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

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

[4] A. M. A. Bin Ateeq, S. A. Abbasi, and A. R. M. Alamoud, Hardware Realization of Walsh Functions and Their Applications Using VHDL and Reconfigurable Logic, ICM 2002, Beirut, Lebanon, pp 58–61, 2002.

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

[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.

[7] A. Amira, A. Bouridane, P. Milligan, and M. Roula, An FPGA Implementation of Walsh-Hadamard Transforms for Signal Processing, Proceeding of IEEE International Conference on Acoustic, Speech and Signal Processing, Vol. 2, pp: 1105–1108, 2001.

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

[9] A. Amira, A. Bouridane, P. Milligan, and M. Roula, Novel FPGA Implementations of Walsh-Hadamard Transforms for Signal Processing, Proceeding of IEE Vision, Image, and Signal Processing, Vol. 148, Issue: 6, 2001.

[10] B. J. Falkowski and T. Sasao, Unified Algorithm to Generate Walsh Functions in Four Different Orderings and Its Programmable Hardware Implementations, Proceeding of IEE on Vision, Image and Signal Processing, Vol. 152, Issue: 6, pp: 819–826, 2005.

[11] A. Amira and S. Chandrasekaran, Power Modeling and Efficient FPGA Implementation of FHT for Signal Processing, IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 15, Issue: 3, pp: 286–295, 2007.

[12] P. K. Meher and J. C. Patra, Fully-Pipelined Efficient Architectures for FPGA Realization of Discrete Hadamard Transform, Proceeding of International Conference on Application-Specific Systems, Architecture and Processors (ASAP 2008), pp: 43–48, 2008.

[13] Zulfikar, S. A. Abbasi and A. R. M. Alamoud, FPGA Based Processing of Digital Signals Using Walsh Analysis, Proceeding of IEEE International Conference on Electrical Control & Computer Engineering (INECCE 2011), Pahang, 21–22 June, 2011.

[14] M. Y. Zulfikar, S. A. Abbasi and A. R. M. Alamoud, FPGA Based Analysis and Multiplication of Digital Signals, IEEE Proceeding of Second International Conference on Advances in Computing, Control and Telecommunication Technologies (ACT 2010), pp: 32–36, Jakarta, 2–3 Dec, 2010.