International Journal of Electrical and Computer Engineering (IJECE)
Vol. 6, No. 4, August 2016, pp. 1551~1559
ISSN: 2088-8708, DOI: 10.11591/ijece.v6i4.9218

Design of 8-point DFT Based on Rademacher Functions

Zulfikar, Hubbul Walidainy
Department of Electrical Engineering, Syiah Kuala University, Indonesia

Article Info

Article history:

Received Oct 20, 2015
Revised Apr 26, 2016
Accepted May 10, 2016

Keyword:

4-point DFT
8-point DFT
FFT
Rademacher functions
Twiddle Factor
Walsh transforms

ABSTRACT

This paper presents a new circuit design for 8-point DFT algorithm based on product of Rademacher functions. The designed circuit is basically constructed base on 8-point DFT decimation in time that mainly construct of two 4-point and four 2-point DFTs. However, the operation of the design circuit is different. It utilized the advantages of Rademacher functions simplicity. Therefore, the proposed design is constructed from our previous design 4-point DFT which is based on product of Rademacher functions. Some analysis upon number types, internal connections and complex conjugate of the results to achieve the more efficient circuit have been made. Therefore, instead of four, the proposed design requires only three 2-point DFTs. Again, two output results of the design DFT have been removed since they are equal in terms of magnitude to the other results, but two negative circuit are required as a compensation. Moreover, the previous designed circuit of 4-point DFT has been replaced with the new circuit design. This circuit is special designed for non-standalone used, the circuit must be integrated inside the proposed 8-point DFT.

Corresponding Author:

Zulfikar,
Department of Electrical Engineering,
Syiah Kuala University,
Jl Syech Abdur Rauf, Darussalam, Banda Aceh, Indonesia.
Email: zulfikarsafrina@unsyiah.ac.id

1. INTRODUCTION

No doubt that Fourier transforms is used ubiquitous. The Fourier algorithms is available in terms of both continue and discrete models. Discrete model of Fourier which is often called Discrete Fourier Transforms (DFT) is more suitable for application since the development of computing machines that limits the ability of calculation. Unlike discrete, the continue model is very difficult to be implemented.

The development of Fourier transforms has been done since about a century ago. However, because of the huge number of applications, it is still attracted scientists to develop a more and more efficient and fast algorithms for implementing the transforms in the applications. Duhamel and Veterli summarized, analyzed and provided some suggestions of those development in 1990 [1]. The most significant improvement of Fourier is when Cooley and Tukey introduced a method for factorization of it [2]. After that, thousands number of paper appears for implementing Fourier in applications.

Alternatively, scientists have developed the algorithms of Fourier transforms that combines Walsh and Fourier transforms [3]-[5]. The developments is based on the simplicity calculation of Walsh transforms that often ignored in the previous researches. Those algorithms such as Walsh transforms is adopted through factorization of intermediate transforms T for calculation of DFT coefficients [3]. Monir T et al then proposed the efficient combination of DFT and Walsh calculations. This technique is used to perform Fast Walsh Hadamard Transforms (FWHT) by utilizing Radix-4 [4]. Later then, the efficient algorithm of calculating both Walsh transforms and DFT transforms using the famous Radix-2 was also published [5].

Those previous combination algorithms are designed for parallel both input and output. This leads to huge number of memory resources which is not suitable for small circuit applications. A method for reducing

circuit resources has been proposed in [6]. The circuit is designed by taking input serially and the output is gathered in parallel. The proposed method used to design 4-point DFT that adopts behavior of how Walsh transforms is performed.

The previous DFT has been designed only for 4-point, which is very simple and rarely used in the application. In the real application, it is required a DFT which able to perform higher than 4 point transformation. Therefore, in this paper, we propose a design of 8-point DFT that is constructed by using the previous 4-point DFT design. Two 4-point DFT and four 2-point DFT are required in the proposed design.

This paper is organized as follows: some basic theories of DFT are covered in the next section. Then in the section 3, step by step circuit design for area efficiency of 8-point DFT is described in detail. Section 4 views the analysis of results and a few discussions of the proposed design. Finally, the conclusions and some suggestions for future works are presented in section 5.

2. THEORIES

2.1. 2-Point and 4-Point Discrete Fourier Transforms

DFT plays a very important role in almost all digital applications. For example, in the fields of digital signal processing (DSP), spectrum analysis and filtering process. Many scientists proposed the theory and implementation of how DFT efficiently used. The basic calculation of DFT is using equation (1) as follow [7]-[9],

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{kn}, \quad 0 \le k \le N-1 \tag{1}$$

where, twiddle factor (W_N) may be evaluated according to equation (2) as follow:

$$W_N = e^{-j2\pi/N} \tag{2}$$

Small point number (N) of DFT is very simple and may be often ignored in discussion. However higher N-point DFTs are constructed from smaller ones. For instance, the 8-point DFT based on FFT algorithm is constructed using 2-point and 4-point DFTs. The 2-point DFT may be calculated easily according to equations (1) and (2). For instance, given x(n) = {2, 5}, let us first compute twiddle factors for the DFT.

$W_2^0 = \cos(0) - j \sin(0) = 1$
$W_2^1 = \cos(\pi) - j \sin(\pi) = -1$

Therefore, the output results may be now obtained as follow:

$X(0) = x(0)W_2^0 + x(1)W_2^0 = 2(1) + 5(1) = 2 + 5 = 7$
$X(1) = x(0)W_2^0 + x(1)W_2^1 = 2(1) + (-1)(5) = 2 - 5 = -3$

The circuit for implementing the 2-point DFT is very easy to be designed since it is required only the addition and subtraction process. Let us now move to the calculation of higher point (N) DFT (4-point). For example, given x(n) = {2,4,3,5}, the twiddle factors can be evaluated as follows:

$W_4^0 = \cos(0) - j \sin(0) = 1$
$W_4^1 = \cos(\pi/2) - j \sin(\pi/2) = -j$
$W_4^2 = \cos(\pi) - j \sin(\pi) = -1$
$W_4^3 = \cos(3\pi/2) - j \sin(3\pi/2) = j$

Therefore, the transformation results may be calculated as follow:

$X(0) = x(0)W_4^0 + x(1)W_4^0 + x(2)W_4^0 + x(3)W_4^0 = 2 + 4 + 3 + 5 = 14$
$X(1) = x(0)W_4^0 + x(1)W_4^1 + x(2)W_4^2 + x(3)W_4^3 = 2 - 4j - 3 + 5j = -1 + j$
$X(2) = x(0)W_4^0 + x(1)W_4^2 + x(2)W_4^0 + x(3)W_4^2 = 2 - 4 + 3 - 5 = -4$
$X(3) = x(0)W_4^0 + x(1)W_4^3 + x(2)W_4^2 + x(3)W_4^1 = 2 + 4j - 3 - 5j = -1 - j$

It can be seen that the computation process of 4-point DFT that is demonstrated here would not require multiplication. This calculation process is similar to the 2-point DFT's. However, because it does contains imaginary part, we should separate the results and store it into different buffers. This process has been demonstrated before in 2015 [6].

2.2. Rademacher Functions

Some researchers preferred performing Walsh transforms based upon product of Rademacher functions. This is found to be more appropriate for circuit realization [9]-[13]. The Rademacher functions itself are defined as stated in equation (3).

$$\phi(n+1,x) = |Sgn(\sin 2\pi 2^n x)|, \quad n = 0,1,2,\ldots 0 \le x < 1 \tag{3}$$

where $\phi(0,x)=1$ and the signum function Sgn(y) is determined as follows,

$$Sgn(y) = \begin{cases} +1, & y \ge 0, \\ -1, & y < 0, \end{cases} \tag{4}$$

In practice, the Rademacher functions can be easily generated by simply using a counter circuit.

3. DESIGN OF 8-POINT DISCRETE FOURIER TRANSFORMS

3.1. General Circuit Design

This paper proposes the circuit design of 8-point DFT based on product of Rademacher functions. The designed circuit is constructed from the previous work of 4-point DFT [6] combined with the 8-point DFT decimation in time design. Figure 1 shows the 8-point DFT based on decimation in time. The structure consists of several smaller point of DFTs. The structure also requires some arithmetic process such as real multiplication and imaginary multiplication.

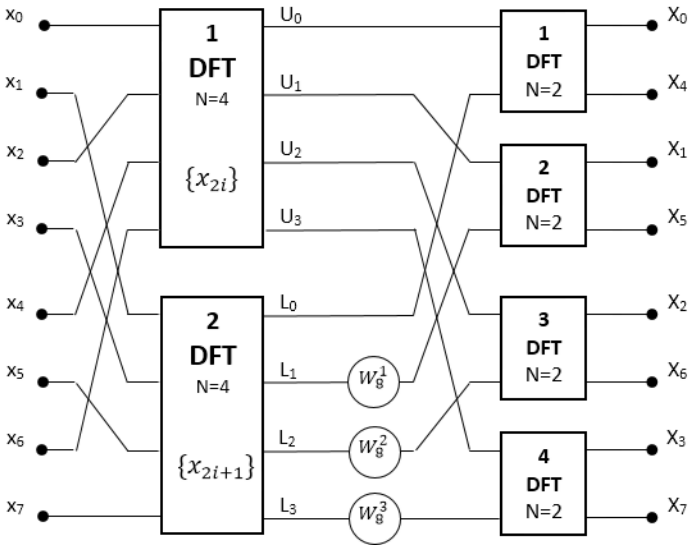

Figure 1. Scheme of 8-point DFT [1]

Input data x[x_0, x_1, x_2, x_3, x_4, x_5, x_6, x_7] will be transformed into frequency domain and become X[X_0, X_1, X_2, X_3, X_4, X_5, X_6, X_7]. Even inputs [x_0, x_2, x_4, x_6] are passed through the first (#1) 4-point DFT. Meanwhile, odd input [x_1, x_3, x_5, x_7] are passed through the second (#2) 4-point DFT. The calculation process of both 4-point DFTs is performed based on product of Rademacher functions [6]. Let's assume that U_0, U_1, U_2, U_3 are results of the first 4-point DFT and L_0, L_1, L_2, L_3 are results of the second 4-point DFT.

Four blocks of 2-point DFT are used to transform temporary results (U and L) to be the final 8-point DFT result X(k). Only inputs of the first block of 2-point DFT are connected directly from temporary results,

others have to be multiplied with twiddle factors. These multiplications process will be evaluated later in the next sections. The process have to be considered as additional resource that is used beside the main blocks of 4-point DFTs and 2-point DFTs. Internal circuit of 4-point DFT is shown in Figure 2.

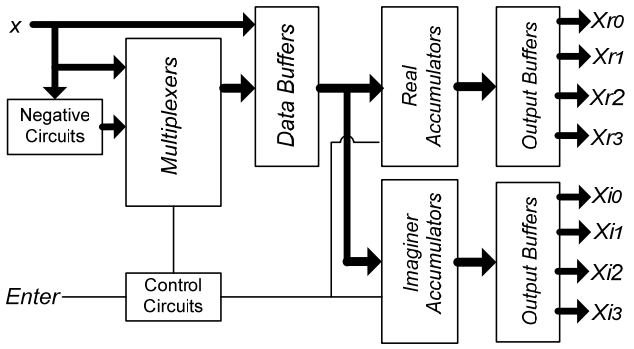

Figure 2. Internal cicuit of 4-point DFT [6]

Input data x is connected in parallel to negative circuit, multiplexers and data buffers. The negative circuit is used to provide negative value of input data x based on the selection of multiplexers. Since the first row of DFT matrix contains only positive value, the connection of data input x to the buffers is made directly in order to avoid multiplexers. The selected values will be passed through buffers to either real accumulators or imaginary accumulators controlled by the signal generated from control circuit. Finally, all stored values in output buffers are passed out and considered as the DFT results. Data buffers are controlled by the signal which is generated by control circuit, but the output buffers is not. These buffers are used to store values temporary before they passed out.

3.2. Type of Number

The circuit scheme in Figure 1 shows blocks of 4-point DFTs, 2-point DFTs and twiddle factors in general view. In order to integrate blocks and components, it requires specific handling that may involve real and imaginary numbers. The connections between blocks or components that involve both real and imaginary numbers requires more circuit. Figure 3 shows all possible of real (noted "R") and imaginary (noted "I") numbers for processing the 8-point DFT.

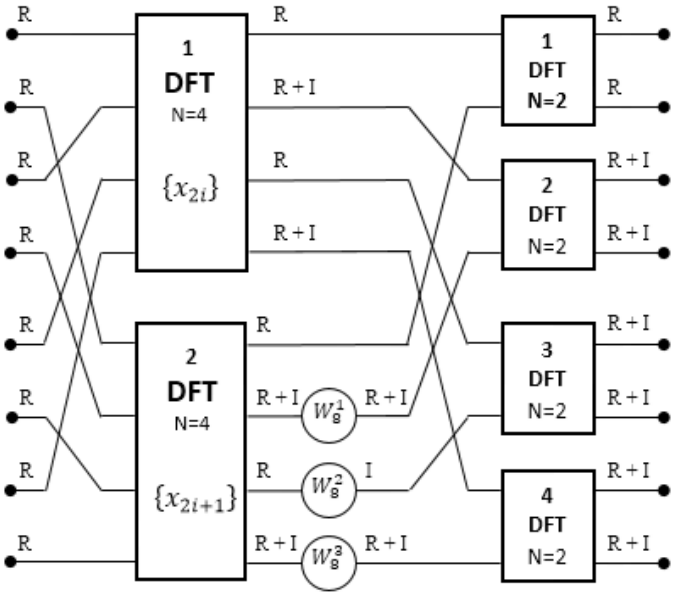

Figure 3. Type of numbers for connections

It is assumed that inputs of 8-point DFT are all real numbers. Then based on the calculation inside 4-point DFT, the temporary results (U and L) will be in real, imaginary or might contains both real and imaginary numbers. Those type of numbers have been derived from the twiddle factor of both 4-point DFT blocks. Let us consider both 4-point DFT blocks, Table 1 shows all possible number types of the result for all input real numbers.

Table 1. Possible types of numbers based on twiddle factor of 4-point DFT

k		Twiddle Factor		Type of Input	Type of Output
0	W_4^0	Cos (0) – j Sin (0)	1	x_0 : Real	U_0 : Real
1	W_4^1	Cos (2π/4) – j Sin (2π/4)	- j	x_2 : Real	U_1 : Real + Imaginary
2	W_4^2	Cos (4π/4) – j Sin (4π/4)	- 1	x_4 : Real	U_2 : Real
3	W_4^3	Cos (6π/4) – j Sin (6π/4)	j	x_6 : Real	U_3 : Real + Imaginary

Some results of the second 4-point DFT (L_1, L_2, L_3) are multiplied with twiddle factors (W_8^1, W_8^2 and W_8^3). These multiplications can be examined as follow,

$$(R + I) \times W_8^1 = (R + I) \times (\text{Cos } (2\pi/8) - j \text{ Sin } (2\pi/8)) = (R + I) \times (R + I) = R + I \quad (5)$$

$$(R) \times W_8^2 = (R) \times (\text{Cos } (4\pi/8) - j \text{ Sin } (4\pi/8)) = (R) \times (I) = I \quad (6)$$

$$(R + I) \times W_8^3 = (R + I) \times (\text{Cos } (6\pi/8) - j \text{ Sin } (6\pi/8)) = (R + I) \times (R + I) = R + I \quad (7)$$

As the result, after performing all of 2-point DFT processes, the output of 8-point DFT contains real and imaginary number except for X_0 and X_4 which contain only real numbers. This is because both inputs of the first (#1) 2-point DFT are contain real numbers only. These analysis play a very important thing in determining the amount of buffers required for implementing the circuit, since the real and imaginary numbers will be placed or stored in different buffers. This design will be further analyzed for determining the amount of required buffer. The connections that involve both real and imaginary requires twice number of buffer for storing data temporarily.

3.3. Interconnect Configuration

The designed 8-point DFT mainly requires two 4-point DFTs and four 2-point DFTs. These amount of DFTs will requires huge numbers of circuit. However, in terms of circuit perspective, there is a space to reduce the circuit. A depth analysis is required for determining which part of the whole circuit can be optimized. In the previous section, the type of numbers used for connecting blocks has been determined. Here, we provide deep analysis of those numbers.

The results of 8-point DFT shows the unique phenomena, because some of them complex conjugate to the other result [7],[8]. For example given x={1,2,3,4,5,6,7,8}, the DFT results are X={36, -4+9.66i, -4+4i, -4+1.66i, -4, -4-1.66i, -4-4i, -4-9.66i}. Where, X_1 is complex conjugate with X_7, X_2 is complex conjugate with X_6 and X_3 is complex conjugate with X_5. In general, this is according to equation (8).

$$X\left(\frac{N}{2} - k\right) = X\left(\frac{N}{2} + k\right)^*, \quad for\ k = 1,2,\ldots,\frac{N}{2} - 1 \quad (8)$$

where N=4,8,16, ….. This behavior also similar to the 4-point DFT results, where $U_1=U_3^*$ and $L_1=L_3^*$. Figure 4 shows the mapping of all possible complex conjugate results of the designed 8-point DFT.

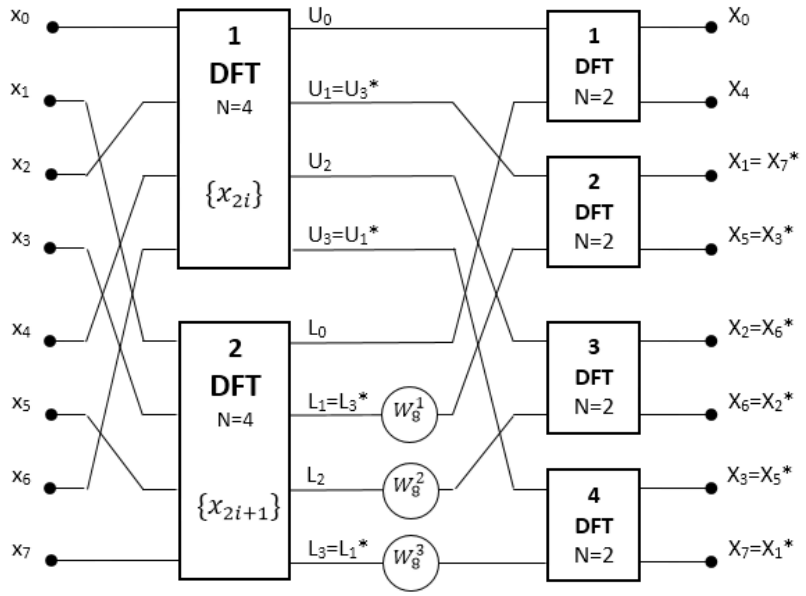

Figure 4. Complex conjugate results

By determining complex conjugate of some DFT results, the circuit can be optimized.

4. CIRCUIT COMPLEXITY

In the previous section, analysis of number's type and complex conjugate of the results has been made. Therefore, the designed circuit may now be optimized by reducing unneeded components or blocks. However, there is a cost for this improvement.

From the Figure 4, it can be seen that the results of 2nd and 4th 2-point DFTs are complex conjugate to each other. Therefore, one of these blocks can be removed. As a consequence of removing the block, it is required a negative circuit. Another advantage of removing the DFT block, either component of twiddle factor W_8^1 or W_8^3 is not required anymore. Let us remove the last block of 2-point DFT, as a result, twiddle factor W_8^2 can be also removed. This leave connections from U_3 and L_3 disconnected.

The multiplication process in the W_8^2 also can be removed because the magnitude of W_8^2 is -1. Based on previous analysis of twiddle factors multiplication indicated in equation (6). The result 4-point DFT L_2 may now be connected directly to the input of the third 2-point DFT block and assumed it as an imaginary number. Two negative circuits are required for compensation of removing one block of 2-point DFT. These circuits are shown in the part of Figure 5. The first circuit is used to make a negative value of X_{I1} and considered as X_{I7} and the second one is used to make a negative value of X_{I5} and considered as X_{I3}.

Another efficiency can be applied in the both blocks of 4-point DFT due to the unconnected of result U_3 and L_3. Figure 5 shows the efficient circuit design of 8-point DFT.

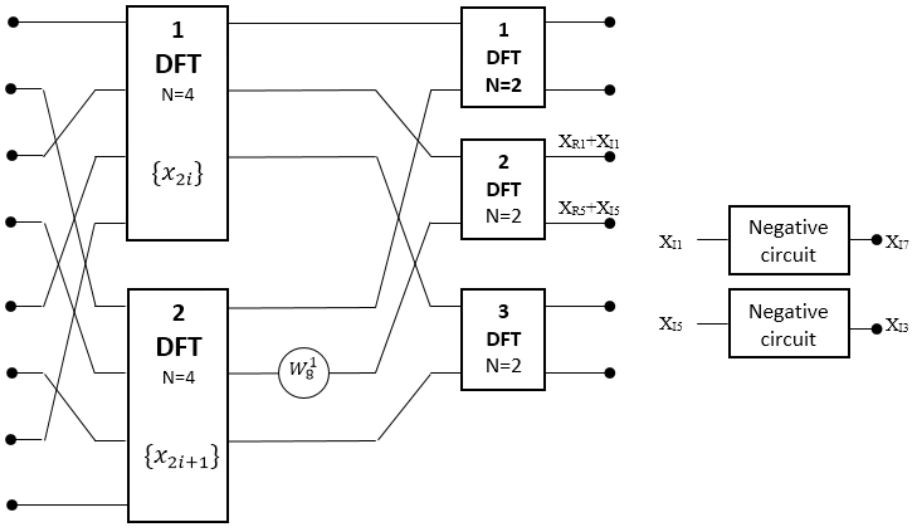

Figure 5. Propose efficient 8-point DFT

Figure 6 shows the circuit of 2-point DFT. The circuit has been developed based on the simplicity of twiddle factors $W_2^0 = 1$ and $W_2^1 = -1$ analyzed in section two. Therefore, for determining the results of 2-point DFT, just simply add and subtract both given inputs. The subtraction is performed through second complement method by just simply invert x_1 and then add to x_0. Figure 7 shows the modified 4-point DFT of the previous proposed one [6].

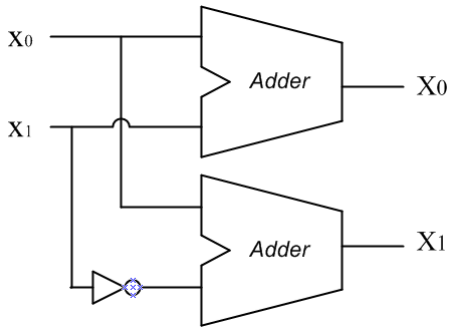

Figure 6. Circuit of 2-point DFT

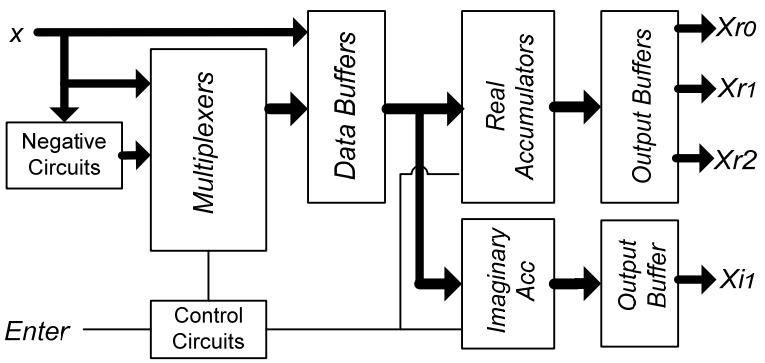

Figure 7. Propose modified 4-point DFT

The modified circuit of 4-point DFT is more efficient, it utilized less accumulator and buffer. It can be seen that there are four accumulators and four output buffers. The previous design required eight accumulators and eight output buffers [6]. However this circuit cannot used as a single standalone application, it must be integrated together to form the proposed 8-point DFT.

5. CONCLUSIONS

The designed circuit of 8-point DFT based on product of Rademacher functions has been done successfuly. Initially, the circuit consists of smaller DFT blocks which are two 4-point DFTs and four 2-point DFTs. The analysis of type number used, internal connections and complex conjugate has been accomplished. Based on these, the efficient 8-point DFT has been achieved. The efficient circuit involved two modified 4-point DFTs and three 2-point DFTs. Moreover, the design of the modified 4-point DFT and the simple 2-point DFT also has been constructed. Several output results of the designed DFT have been removed since they are equal in terms of magnitude, two negative circuit are required as a compensation.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge the financial support from Syiah Kuala University, Ministry of Education and Culture, Indonesia under project Hibah Bersaing, No. 035/SP2H/PL/Dit.Litabmas/II/2015, Feb 5, 2015.

REFERENCES

[1] P. Duhamel and M. Vetterli, "Fast Fourier Transforms: a tutorial, review and a state of the art," *Trans. Signal Processing*, vol/issue: 19(4), pp. 259-299, 1990.
[2] J. W. Cooley and J. W. Tukey, "An Algorithm for the Machine Calculation of Complex Fourier Series," *Math. Comp.*, vol. 19, pp. 297-301, 1965.
[3] S. Boussakta and A. G. J. Holt, "Fast algorithm for calculation of both Walsh-Hadamard and Fourier transforms (FWFTs)," *Electron. Letter*, vol/issue: 25(20), pp. 1352-1354, 1989.
[4] M. T. Hamood and S. Boussakta, "Fast Walsh–Hadamard–Fourier transform algorithm," *Trans. Signal Processing*, vol/issue: 59(11), pp. 5627-5631, 2011.
[5] T. Su and F. Yu, "A Family of Fast Hadamard–Fourier Transform Algorithms," *Signal Processing Letters*, vol/issue: 19(9), pp. 583-586, 2012.
[6] Zulfikar and H. Walidainy, "A Novel 4-Point Discrete Fourier Transforms Circuit based on Product of Rademacher Functions," *IEEE Proceeding of International Conference of Electrical Engineering and Informatics (ICEEI)*, Bali, Indonesia, August 10-11, pp. 142-147, 2015.
[7] J. G. Proakis and D. G. Manolakis, "Digital signal processing: principles, algorithms, and applications," 4th ed., Pearson Prentice Hall, New Jersey, 2007.
[8] S. Salivahanan, *et al*., "Digital Signal Processing," McGraw Hill, New Delhi, 2000.
[9] M. G. Karpovsky, *et al.*, "Spectral Logic and Its Applications for The Design of Digital Devices," John Wiley & Sons Inc. Publication, New Jersey, 2008.
[10] M. Y. Zulfikar, *et al.*, "FPGA Based Analysis and Multiplication of Digital Signals," *Proceeding of IEEE Second International Conference on Advances in Computing, Control, and Telecommunication Technologies (ACT 2010)*, Jakarta, Indonesia, pp. 32-36, 2010.
[11] M. Y. Zulfikar, *et al.*, "FPGA Based Processing of Digital Signals using Walsh Analysis," *Proceeding of IEEE International Conference on Electrical, Control and Computer Engineering (INECCE 2011)*, 21-22 June, Pahang, Malaysia, pp. 440-444, 2011.
[12] Zulfikar, *et al.*, "A Novel Complete Set of Walsh and Inverse Walsh Transforms for Signal Processing," *Proceeding of IEEE International Conference on Communication Systems and Network Technologies (CSNT 2011)*, Katra, Jammu, 3-5 June, pp. 504-509, 2011.
[13] Zulfikar, *et al.*, "FPGA Based Complete Set of Walsh and Inverse Walsh Transforms for Signal Processing," *Transaction of Electronics and Electrical Engineering*, vol/issue: 18(8), pp. 3-8, 2012.

BIOGRAPHIES OF AUTHORS

Zulfikar. He was born in Beureunuen, Aceh, Indonesia, in 1975. He received his B.Sc. degree in Electrical Engineering from North Sumatera University, Medan, Indonesia, the M. Sc. Degree in Electrical Engineering from King Saud University, Riyadh, Saudi Arabia, in 1999 and 2011, respectively. He joined as teaching staff in the Department of Electronics at Politeknik Caltex Riau, Pekanbaru, Indonesia in 2003. He served as head of Industrial Control Laboratory, Politeknik Caltex Riau from 2003 to 2006. In 2006, he joined the Electrical Engineering Department, Syiah Kuala University. He has been appointed as head of Digital Laboratory for two successive years. His current research interests include VLSI design and System on Chips (SoC).

Hubbul Walidainy. He was born in Banda Aceh, Aceh, Indonesia, in 1973. He graduated from Electrical Engineering Department at Gadjah Mada University, Yogyakarta, Indonesia, in 1998, The Master Degree in Electrical Engineering from Gadjah Mada University, Yogyakarta, Indonesia, in 2003. He joined in the Department of Electrical Engineering, Syiah Kuala University, Aceh, Indonesia in 2000, as a teaching staff. His current position is the head of Telecommunication Laboratory. His current research interests include all issues in Digital Signal Processing (DSP).