An Approach in Designing 16-point DFT with Decimation in Time based on Rademacher Functions

Zulfikar Zulfikar and Hubbul Walidainy
Department of Electrical and Computer Engineering, Syiah Kuala University, Banda Aceh 23111, Indonesia
zulfikarsafrina@unsyiah.ac.id

***Abstract*—This paper presents a circuit design for 16-point DFT algorithm with Decimation in Time based on products of Rademacher functions. The designed circuit is constructed from two 8-point DFT and four 2-point DFT. However, the operation of the design circuit is different. It utilised the advantages of the similarity of Fourier transforms, and Rademacher functions. Therefore, the proposed design is constructed from previously designed 8-point DFT which is based on products of Rademacher functions. Some analysis of the type of numbers, internal connections and the complex conjugate of the results to achieve the more efficient circuit has been made. Therefore, instead of eight, the proposed design requires only five 2-point DFTs. Therefore, six output results of the design 16-point DFT have been removed since they are equal regarding magnitude to the other results, but six negative circuits are required as compensation. Therefore, the previously designed circuit of 8-point DFT has been replaced with the new circuit design. This circuit is specially designed for non-standalone used; the circuit must be integrated inside the proposed 16-point DFT.**

***Index Terms*—8-Point DFT; Decimation In Time; Fourier Transforms; Walsh Transform.**

I. Introduction

Nowadays, Fourier transforms used ubiquitously. The Fourier algorithms for converting the information to frequency domain are available concerning both continuous and discrete models. The discrete model of Fourier which is often called Discrete Fourier Transforms (DFT) is more suitable for hardware application since the capability of computing machines that limit the ability of calculation. Unlike discrete one, the continuous model was challenging to be implemented.

Fourier transforms have been developed since long time ago because of the huge number of applications that require this model. It is still an attracted work for scientists to develop a more efficient and fast algorithm for implementing it in the applications. Duhamel and Veterli described a brief history and development of the Fourier algorithm in 1990 [1]. They presented a detail explanation of advantages and drawbacks of each previously proposed algorithm. The most significant improvement of the Fourier transform is when Cooley and Tukey introduced a method for factorisation of it [2]. After that, thousands number of work published for implementing Fourier transform in real applications.

Meanwhile, the calculation process of Walsh transforms for converting information to the frequency domain is very simple. Even though, in the application, the calculation process may be performed using the integer and real number only. Therefore, scientists have been developing the algorithms of Fourier transforms that combines Walsh and Fourier transforms [3]-[5]. The developments are based on the simple calculation of Walsh transforms. Those algorithms of Walsh transform adopted through factorisation of intermediate transforms T for gathering of Fourier coefficients [3]. Monir T et al. then proposed the efficient combination of Fourier and Walsh calculations. This technique is used to perform the Fast Walsh Hadamard Transforms (FWHT) by utilising decimation in time (DIT) of Radix-4 [4]. Later then, the efficient algorithm for determining of both Walsh transforms and DFT transforms based on the Radix-2 model was also proposed [5].

Those previous combination algorithms were designed for entering information into the system in parallel and gathering the results also in parallel. This model leads to many memory resources which are not suitable for embedded realisation. Therefore, a method for reducing the usage of the resource has been proposed in [6]. The circuit is designed by taking information serially, and the results are extracted in parallel. The method utilised 4-point DFT that adopts the behaviour of how Walsh transforms is performed. Next, the design of 8-point DFT [7] has been proposed. It is constructed using the 4-point DFT designed in [6]. This design of 8-point itself is very simple; it constructed from two 4-point DFT and three 2-point DFT.

The previous DFT model has been designed only for 4-point and 8-point, which is very simple and rarely used in the real application. In the real application, it is required a DFT model which able to perform higher than 8-point transformation processes. Therefore, in this paper, we propose a design of 16-point DFT circuit that is constructed by using the previous 8-point DFT model. Two 8-point DFT and eight 2-point DFT are required in this design. This paper also provides an analysis of the type of number and complex conjugate for improvement purpose of the design.

This paper is organised as follows: a step by step circuit design for area efficiency of 16-point DFT is described in detail in Section 2. Section 3 views the analysis of results and discussions of the proposed design. Finally, the conclusions and some suggestions for future works are presented in section 4.

II. Design of a 16-point DFT

A. Main Design

This paper proposes the design of the circuit for implementing 16-point DFT based on products of Rademacher functions. This function has been appeared in many designs for realising Walsh transforms [8]-[12]. That implementation theoretically based upon algorithms of Walsh transforms and its application in different ordering [13]. The circuit is constructed from the previous work of 4-point DFT [6], 8-point DFT [7] combined with the design of 16-point DFT decimation in time. Figure 1 shows the 16-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.

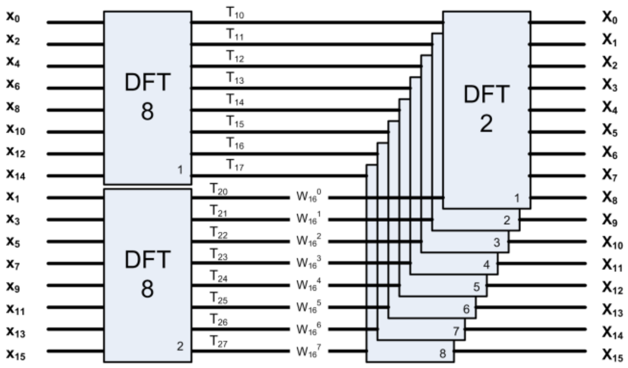

Figure 1: Structure of 16-point DFT decimation in time

Input data x[x_0, x_1, x_2,..., x_{15}] will be transformed into frequency domain and become X[X_0, X_1, X_2,..., X_{15}]. Even inputs [x_0, x_2, x_4, x_6, x_8, x_{10}, x_{12}, x_{14}] are passed through the first (#1) 8-point DFT. Meanwhile, odd input [x_1, x_3, x_5, x_7, x_9, x_{11}, x_{13}, x_{15}] are passed through the second (#2) 8-point DFT. The calculation process of both 8-point DFTs is performed based on products of Rademacher functions [7]. Let's assume that T_{10}, T_{11}, T_{12}, T_{13}, T_{14}, T_{15}, T_{16}, T_{17} are results of the first 8-point DFT and T_{20}, T_{21}, T_{22}, T_{23}, T_{24}, T_{25}, T_{26}, T_{27} are results of the second 8-point DFT.

Eight blocks of 2-point DFT are used to transform temporary results (Ts) to be the final 16-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 by twiddle factors. These multiplications process will be evaluated next. The multiplication processes have to be considered as an additional resource that is used beside the main blocks of 8-point DFTs and 2-point DFTs. The internal circuit of 8-point DFT will be evaluated next.

B. *Type of Number*

The circuit scheme in Figure 1 shows blocks of 8-point DFTs, 2-point DFTs and twiddle factors in general view. To integrate blocks and components, it requires specific handling that may involve real and imaginary numbers. The connections between blocks or components that require both real and imaginary numbers require more circuit. Figure 2 views all possible of imaginary (noted "I") and real (noted "R") numbers for processing the 16-point DFT.

It is assumed that inputs of 16-point DFT are all real numbers. Then based on the calculation inside 8-point DFT, the temporary results (Ts) will be in real, imaginary or might contain both real and imaginary numbers. Those type of numbers has been derived from the twiddle factor of both 8-point DFT blocks. For processing the multiplications of some output of 8-point DFTs, we should examine all possible twiddle factor's type of number. Table 1 lists and derivation of the kind of the number of several twiddle factors that involve in the calculation.

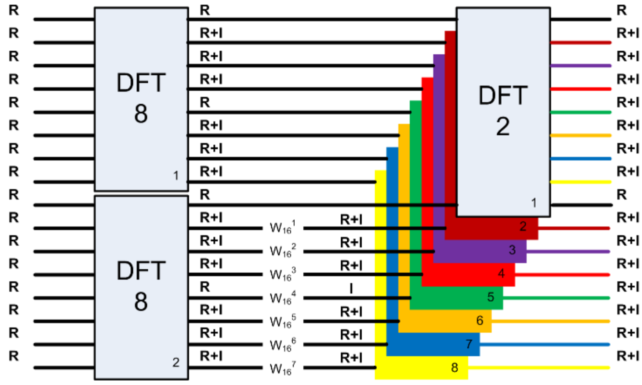

Figure 2: Type of numbers of internal connections of 16-point DFT

Table 1
Types of Numbers of Twiddle Factors

K	Twiddle-Factor			
0	W_{16}^0	Cos(0) – j Sin (0)	1	R
1	W_{16}^1	Cos(2π/16) – jSin (2π/16)	0,924–j0,382	R + I
2	W_{16}^2	Cos(4π/16) – jSin (4π/16)	0,707–j0,707	R + I
3	W_{16}^3	Cos(6π/16) – jSin (6π/16)	0,382–j0,924	R + I
4	W_{16}^4	Cos(8π/16) – jSin (8π/16)	- j	I
5	W_{16}^5	Cos(10π/16) – jSin(10π/16)	-0,382,–j0,924	R + I
6	W_{16}^6	Cos(12π/16) – jSin(12π/16)	-0,707–j0,707	R + I
7	W_{16}^7	Cos(14π/16) – jSin(14π/16)	-0,924–j0,384	R + I

It can be seen that most of twiddle factors requires calculation in real and imaginary. Twiddle factors W160 can be ignored since it equal to 1. Some results of the second 8-point DFT (T_{21}, T_{22}, T_{23}, T_{24}, T_{25}, T_{26}, T_{27}) are multiplied with twiddle factors (W_{16}^1, W_{16}^2, W_{16}^3, W_{16}^4, W_{16}^5, W_{16}^6, W_{16}^7). These multiplications can be examined as follow,

$$(T_{21})\,(W_{16}^1) = (R{+}I)\,(R{+}I) = R{+}I \quad (1)$$
$$(T_{22})\,(W_{16}^2) = (R{+}I)\,(R{+}I) = R{+}I \quad (2)$$
$$(T_{23})\,(W_{16}^3) = (R{+}I)\,(R{+}I) = R{+}I \quad (3)$$
$$(T_{24})\,(W_{16}^4) = (R)\,(I) = I \quad (4)$$
$$(T_{25})\,(W_{16}^5) = (R{+}I)\,(R{+}I) = R{+}I \quad (5)$$
$$(T_{26})\,(W_{16}^6) = (R{+}I)\,(R{+}I) = R{+}I \quad (6)$$
$$(T_{27})\,(W_{16}^7) = (R{+}I)\,(R{+}I) = R{+}I \quad (7)$$

As a result, after performing all of 2-point DFT processes, the output of 16-point DFT contains real and imaginary number except for X_0 and X_8 which include only real numbers. This is because both inputs of the first 2-point DFT include real numbers only. These analyses play an essential thing in choosing the number 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 analysed for determining the exact amount of required buffer. The connections that involve both real and imaginary requires a two-fold amount of buffer for storing data temporarily.

C. *Interconnect Configuration*

The designed 16-point DFT mainly requires two 8-point DFTs and eight 2-point DFTs. These number of DFTs will need huge numbers of the circuit. However, regarding circuit perspective, there is a space to reduce it. A depth analysis is required for determining which part of the whole circuit that might be removed. In the previous section, the type of numbers used for connecting blocks has been determined. Here, we provide the detailed analysis of those figures.

The results of 16-point DFT shows the unique phenomena, because some of them complex conjugate to the other result [14], [15]. For example, given input data x={1, 2, 3, 4, 5, 6, 7, 8, 11, 4, 1, 3, 5, 6, 2, 9}, the DFT results are X={77, -12.6-4.7i, 4.8+16.3i, -9.1-1.7i, 9+6i, -6.5+8.1i, -0.8+6.3i, -11.5+5.1i, -7, -11.5-5.1i, -0.8-6.3i, -6.5-8.1i, 9-6i, -9.1+1.7i, 4.8-16.3i, -12.6+4.7i}. Where, X_1 is complex conjugate with X_{15}, X_2 is complex conjugate with X_{14} and so on. In general, this is according to equation (8).

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

where N=4, 8, 16 ….. This behaviour was also similar to the 8-point DFT results, where $T_{11}=T_{17}^*$, $T_{12}=T_{16}^*$, $T_{13}=T_{15}^*$, $T_{21}=T_{27}^*$, $T_{22}=T_{26}^*$, and $T_{23}=T_{25}^*$. Figure 3 shows the mapping of all possible complex conjugate results of the designed 16-point DFT. By determining complex conjugate of some DFT results, the circuit can be optimised.

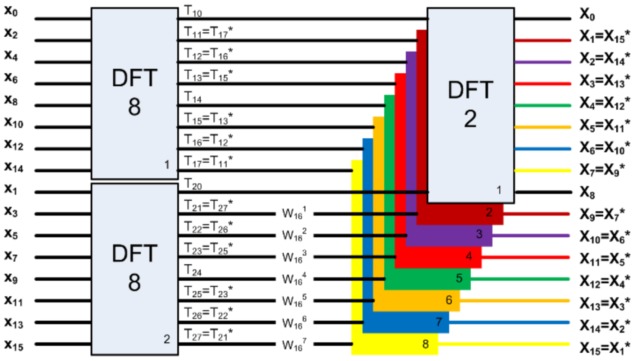

Figure 3: Complex conjugate results of 16-point DFT

III. CIRCUIT COMPLEXITY

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

From the Figure 3, it can be seen that the results of 2nd, 3rd, 4th and 8th, 7th, 6th of 2-point DFTs are complex conjugate to each other. Therefore, half 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 blocks twiddle factors W_{16}^1 or W_{16}^7, W_{16}^2 or W_{16}^6, W_{16}^3 or W_{16}^5 are not necessary anymore. Let us remove the last three blocks of 2-point DFT. As a result, twiddle factor W_{16}^5, W_{16}^6, W_{16}^7, can also be deleted. These leave connections from T_{15}, T_{16}, T_{17}, T_{25}, T_{26}, and T_{27} disconnected.

The multiplication process in the W_{16}^4 also can be removed because the magnitude of W_{16}^4 is -1. Based on previous analysis of twiddle factors multiplication indicated in Equation (6). The result 8-point DFT T_{24} may now be connected directly to the input of the fifth 2-point DFT block and assumed it as an imaginary number. Three negative circuits are required for compensation of removing three blocks of 2-point DFT. These circuits can be realised based on second complement system using an adder. The first negative circuit is used to create a negative imaginary part of X_5 and considered as imaginary part of X_{11}. The second one is used to provide a negative imaginary part of X_6 and considered as imaginary part of X_{10}. The third one is used to create a negative imaginary part of X_7 and considered as imaginary part of X_9.

Another efficiency can be applied in both blocks of 8-point DFT due to the unconnected of result T_{15}, T_{16}, T_{17}, T_{25}, T_{26}, and T_{27}. Figure 4 shows the efficient circuit design of 8-point DFT. There is no reduction can be applied to the complex conjugate of $X_4=X_{12}^*$, since they are an output of the same 2-point DFT block.

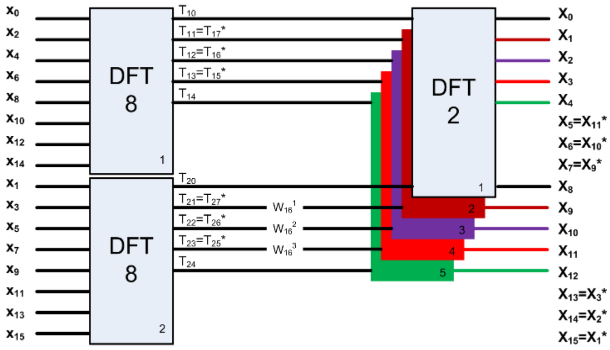

Figure 4: Propose efficient 16-point DFT

In the previous design [7], the efficient 8-point DFT has been proposed. However, the circuit will not suit the proposed 16-point DFT here, since it requires only five results of 8-point DFT (X_0, X_1, X_2, X_3, X_4). Therefore, in this design, the modified 8-point DFT for integrating together with the design 16-pint DFT will be introduced. Figure 5 shows a more efficient 8-point DFT that can be used for calculating the proposed 16-point DFT. For circuit simplicity, the result of X_3 is taken from complex conjugate of X_5, since $X_3=X_5^*$. Therefore, we can remove the last 2-point DFT. Based on this simplicity, in the output part of the modified 8-point DFT consist of only two 2-point DFT, one adder and one negative circuit for performing complex conjugate of X_5 (not shown).

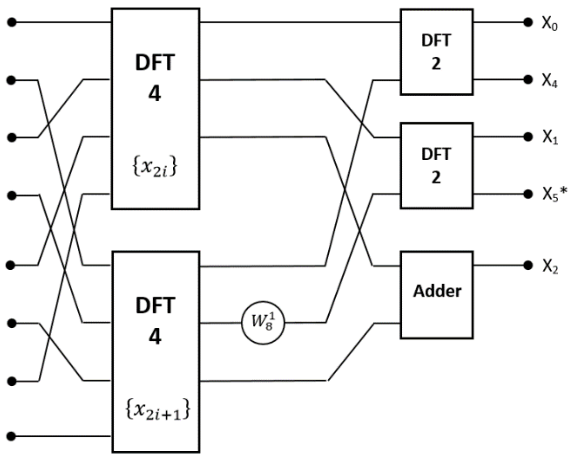

Figure 5: Propose modified 8-point DFT

In this design, the 2-point DFT can be realised using the same circuit used in [7]. The circuit consists of two adders and one inverter as can be seen in Figure 6. Real and imaginary values will be processed in the separate different circuits. Therefore, a double circuit is required when performing 2-point DFT that containing both real and imaginary data.

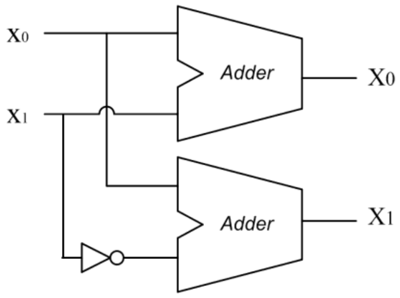

Figure 6: Circuit of 2-point DFT [7]

Figure 7 shows internal circuit that forms the calculation process of 4-point DFT [7]. This circuit performs the DFT based on the products of Rademacher functions. In the figure, primary circuit plays a crucial role in selecting whether positive or negative of x that will be passed through buffers. This selection is similar to the process of performing Walsh transforms. The circuit also determines which buffer will be used to store the selected input data (x or –x) temporarily. The last action is similar to the process of calculating Fourier transforms.

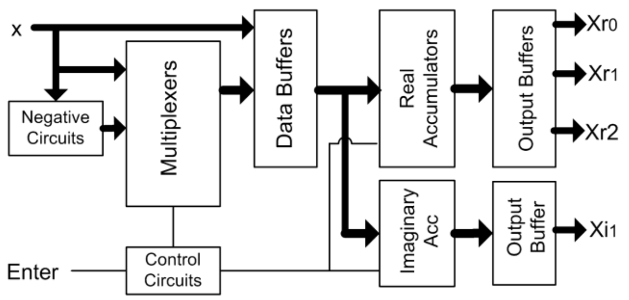

Figure 7: 4-point DFT [7]

IV. CONCLUSIONS

The designed circuit of 16-point DFT based on products of Rademacher functions has been done. Initially, the circuit consists of smaller DFT blocks which are two 8-point DFTs and eight 2-point DFTs. The analysis of type number, internal connections and the complex conjugate of the connections has been accomplished. Based on these, the efficient 16-point DFT has been gathered. The active circuit involved two modified 8-point DFTs and five 2-point DFTs. Moreover, the design of the modified 8-point DFT has been constructed. This circuit can be used in the hardware application that requires small circuit and fast computation.

REFERENCES

[1] P. Duhamel, and M. Vetterli, “Fast Fourier Transforms: a tutorial, review and a state of the art,” *Trans. Signal Processing*, vol. 19, no. 4, pp. 259-299, April 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, April 1965.

[3] S. Boussakta, and A. G. J. Holt, “Fast algorithm for calculation of both Walsh-Hadamard and Fourier transforms (FWFTs),” *Electron. Letter*, vol. 25, no. 20, pp. 1352-1354, 1989.

[4] Monir T. Hamood and, Said Boussakta, “Fast Walsh–Hadamard–Fourier transform algorithm,” *Trans. Signal Processing*, vol. 59, no. 11, pp. 5627-5631, November 2011

[5] Teng Su, and Feng. Yu, “A Family of Fast Hadamard–Fourier Transform Algorithms,” *Signal Processing Letters*, vol. 19, no. 9, pp. 583-586, September 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)*, pp:142-147, Bali, Indonesia, August 10-11, 2015

[7] Zulfikar, Z. and Walidainy, H., 2016. Design of 8-point DFT based on Rademacher Functions. *International Journal of Electrical and Computer Engineering (IJECE)*, 6(4), pp.1551-1559

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

[9] M. Y. 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 and Computer Engineering (INECCE 2011)*, pp: 440-444, 21-22 June, Pahang, Malaysia, 2011.

[10] Zulfikar, S. A. Abbasi, and A. R. M. Alamoud, “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)*, pp: 504-509, Katra, Jammu, 3-5 June 2011.

[11] Zulfikar, S. A. Abbasi, and A. R. M. Alamoud, “FPGA Based Complete Set of Walsh and Inverse Walsh Transforms for Signal Processing,” *Transaction of Electronics and Electrical Engineering*, vol. 18, no. 8, pp. 3-8, October 2012.

[12] Zulfikar, Z., Abbasi, S.A., and Alamoud, A.R.M., 2016. FPGA Hardware Realization: Addition of Two Digital Signals Based on Walsh Transforms. *International Journal of Electrical and Computer Engineering (IJECE)*, 6(6), pp.2688-2697

[13] 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

[14] John. G. Proakis, and Dimitris G. Manolakis, *Digital signal processing: principles, algorithms, and applications*, 4th ed., Pearson Prentice Hall, New Jersey, 2007.

[15] S. Salivahanan, A. Vallavaraj, and C. Gnanapriya, *Digital Signal Processing*, McGraw-Hill, New Delhi, 2000.