

# Unified architecture for 2, 3, 4, 5, and 7-point DFTs based on Winograd Fourier transform algorithm

Fahad Qureshi, Mario Garrido and Oscar Gustafsson

**Linköping University Post Print**



N.B.: When citing this work, cite the original article.

Original Publication:

Fahad Qureshi, Mario Garrido and Oscar Gustafsson, Unified architecture for 2, 3, 4, 5, and 7-point DFTs based on Winograd Fourier transform algorithm, 2013, Electronics Letters, (49), 5, 348-U60.

<http://dx.doi.org/10.1049/el.2012.0577>

Copyright: Institution of Engineering and Technology (IET)

<http://www.theiet.org/>

Postprint available at: Linköping University Electronic Press

<http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-74760>

# Unified Architecture Based on Winograd Fourier Transform Algorithm for 2, 3, 4, 5, and 7-point DFTs

Fahad Qureshi, Mario Garrido, and Oscar Gustafsson

In this letter, a unified hardware architecture that can be reconfigured to calculate 2, 3, 4, 5, or 7-point DFTs is presented. The architecture is based on the Winograd Fourier transform algorithm (WFTA) and the complexity is equal to a 7-point DFT in terms of adders/subtractors and multipliers plus only seven multiplexers introduced to enable reconfigurability. The processing element finds potential use in memory-based FFTs, where non-power-of-two sizes are required such as in DMB-T.

**Introduction:** The discrete Fourier transform (DFT) is an important algorithm in the field of digital signal processing. It transforms a signal from the time domain into the frequency domain, providing information about the spectrum of the signal. The direct computation of an  $N$ -point DFT requires to calculate a number of operations proportional to  $N^2$ . In order to reduce the number of arithmetic operations, many fast algorithms have been proposed, such as Cooley-Tukey [1], prime factor (PFA) [2] and Winograd Fourier transform (WFTA) [3] algorithms. Here, we refer to them collectively as fast Fourier transform (FFT) algorithms. These algorithms are based on decomposing an  $N$ -point DFT recursively into smaller DFTs, leading to a reduction of the computational complexity [4].

Most FFT algorithms and architectures have focused on power-of-two size DFTs. However, recently the interest in non-power-of-two size DFTs has increased, mainly motivated by the 3780-point DFT in Chinese digital TV (DMB-T) [5, 6] based on orthogonal frequency-division multiplexing (OFDM). In the receiving side of OFDM systems, an inverse DFT (IDFT) is usually required, which is easily computed using a DFT processor.

Most FFT architectures are not well optimised for the computation of non-power-two-point FFTs, which make use of small point DFTs with varying sizes, as well as more complex data management. Some pipelined architectures for the 3780-point DFT in DMB-T have been proposed [5, 6]. However, the streaming nature of a pipelined architecture leads to the fact that it can often process data at a much higher rate compared to the required 7.56 Mb/s. Hence, the amount of computational resources are often excessive. In [7], individual processing elements for 3 and 5-point DFTs was proposed and considered for a pipelined architecture. However, they were not based on the WFTA and have a slightly higher complexity.

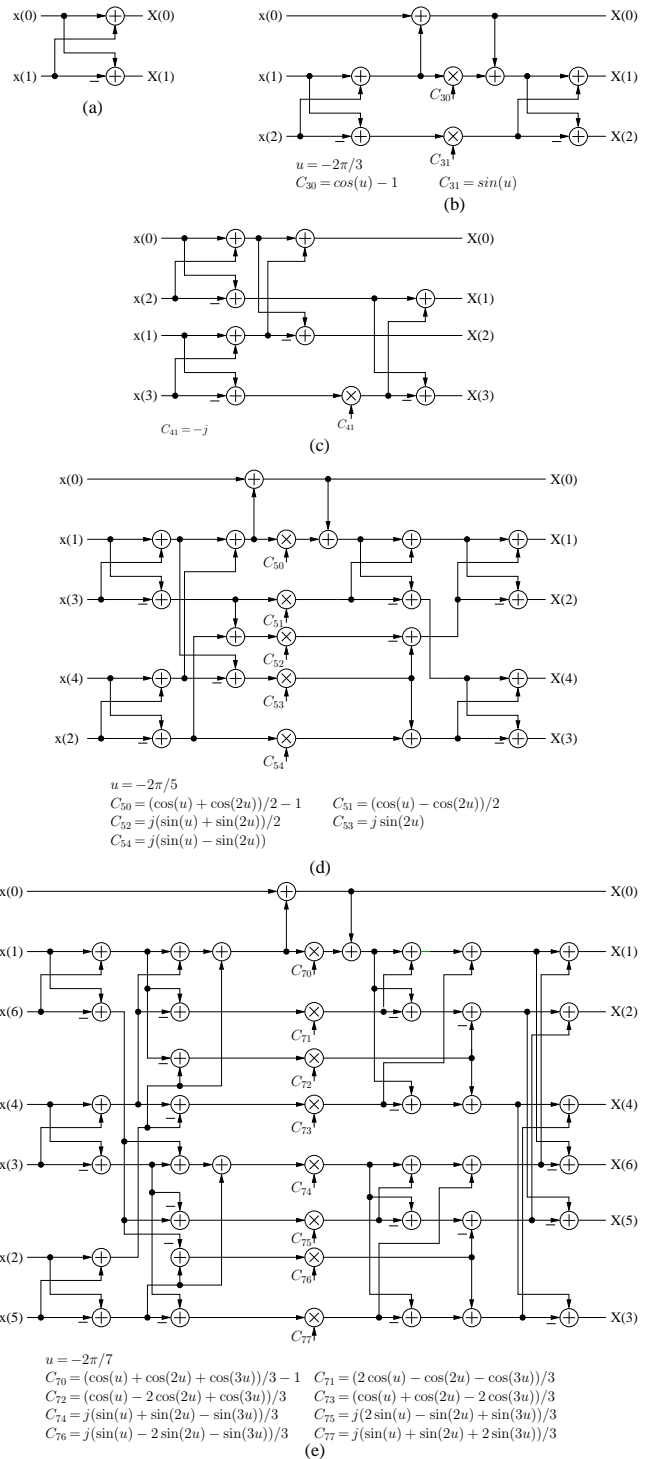
Memory-based FFTs are often more suitable for low data rate applications (where the clock frequency offered by the implementation technology is higher than the data rate), as they allow reusing the computational resources to a higher degree [8]. For a non-power-of-two memory-based FFT, a number of challenges remain. One is how to carry out the more complex data management to interconnect the small DFTs. Another one is to develop a processing element that is suitable for computing small point DFTs of different sizes. This letter presents a unified architecture to compute the 2, 3, 4, 5, and 7-point DFTs by a single processing element. This architecture can be used as the computational core of a memory-based architecture for any DFT whose size that can be decomposed into the factors 2, 3, 4, 5 and 7.

The proposed unified architecture is based on the WFTA algorithm. This algorithm has the minimum number of multiplications at the expense of introducing additions [3]. Although WFTA is very efficient for small prime size DFTs, for larger sizes the number of additions becomes too high for practical implementations.

**Architectures of 2, 3, 4, 5, and 7-Point DFTs:** Figure 1 shows the individual signal flow graphs of 2, 3, 4, 5, and 7-point DFTs. The signal flow graphs of the 3, 5, and 7-point DFTs are based on the WFTA algorithm, whereas the 4-point is based on the Cooley-Tukey FFT algorithm, and the 2-point is a direct computation. In the WFTA, the DFT computation can be written as

$$[X(0) \dots X(N-1)]^T = O \cdot M \cdot I \cdot [x(0) \dots x(N-1)]^T, \quad (1)$$

where  $I$  is a matrix corresponding to additions between inputs,  $M$  is a diagonal matrix with the multiplications, and  $O$  is a matrix corresponding to additions after the multiplications. Multiplications are performed by semi-complex multipliers at the second stage. A semi-complex



**Fig. 1** Signal flow graphs for small DFTs based on WFTA. (a) 2-point. (b) 3-point. (c) 4-point. (d) 5-point. (e) 7-point.

multiplication has a complex input but a purely real or purely imaginary coefficient and, hence, has half the implementation cost compared to a general complex multiplication. The multiplication coefficients for each size are also shown in Fig. 1. Finally, the numbers at the input represent the index of the input sequence,  $x(n)$ , whereas those at one output are the frequencies  $k$  of the output signal  $X(k)$ .

**Proposed Unified Architecture:** The unified architecture is based on mapping the signal flow graphs in Fig. 1 into a single processing element. As starting point, we consider a direct mapping of the 7-point DFT onto which all the other sizes are mapped. As the number of operations for 7 points is the largest and, hence, there are enough computational resources available, the main challenge is to reduce the number of multiplexers. To obtain this, it is important to find common parts in the signal flow graphs that can be mapped without multiplexers. On the one hand, multiplexer can be avoided by setting the inputs of the circuit to zero or by setting the

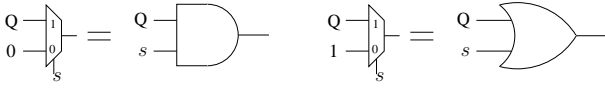


Fig. 2. Single gate multiplexers.

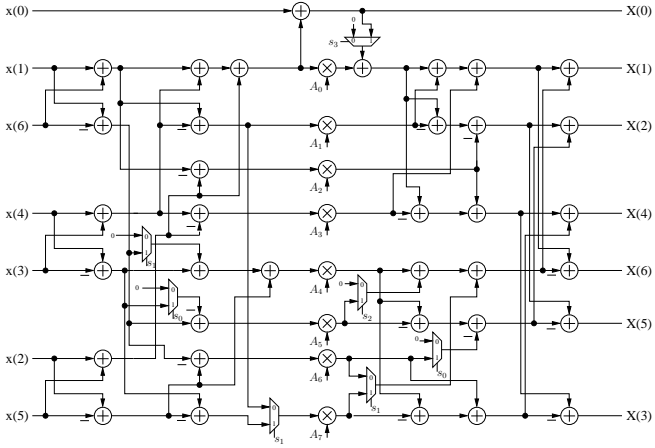


Fig. 3. Proposed unified architecture.

coefficient of a multiplier to zero. This removes unnecessary connections of the circuit. On the other hand, by setting the coefficient of a multiplier to one, this multiplier can be bypassed, which also removes the need of a multiplexer. Using these techniques, a solution with only two two-to-one multiplexers and five single gate multiplexers, controlled with four control signals, has been found. The single gate multiplexers can be implemented by a single gate as shown in Fig. 2.

The resulting architecture is shown in Fig. 3. The required multiplier coefficients are shown in Table 1, where the  $C_{XY}$  coefficient values are based on Fig. 1, and 0 and 1 are required to bypass or break the operators as discussed above. The input and output relations are shown in Table 2, where the dashes denote don't care conditions and 0 denotes that the input should be zeroed for proper operations. In both cases, zero values can be fed from the stored zero value in memory or by using a single gate multiplexer as shown in Fig. 2. Finally, the signals controlling the multiplexers are shown in Table 3, where the dashes again denote don't-care conditions.

**Conclusions:** In this letter, a reconfigurable unified processing element architecture for computing 2, 3, 4, 5, and 7-point DFTs is proposed. It is suitable as the core computational unit when computing DFTs in memory-based architectures. The processing element is suitable for any DFT size which can be decomposed into the included sizes.

**Acknowledgment:** F. Qureshi was supported by the Higher Education Commission, Pakistan. M. Garrido was supported by ELLIIT, Linköping University, Sweden. O. Gustafsson was supported by a career contract from Linköping University, Sweden.

F. Qureshi, M. Garrido, and O. Gustafsson (*Linköping University, Linköping, Sweden*)

E-mail: oscar.gustafsson@liu.se

## References

- 1 J. Cooley and J. Tukey, "An algorithm for the machine calculation of complex Fourier series," *Math. Comput.*, vol. 19, pp. 297–301, 1965.

Table 1: Multiplication coefficients for different DFT sizes.

Coeff.	DFT size				
	2	3	4	5	7
$A_0$	1	$C_{30}$	0	$C_{50}$	$C_{70}$
$A_1$	0	0	0	$C_{53}$	$C_{71}$
$A_2$	0	0	0	0	$C_{72}$
$A_3$	0	0	0	0	$C_{73}$
$A_4$	0	$C_{31}$	1	$C_{52}$	$C_{74}$
$A_5$	0	0	$C_{41}$	$C_{51}$	$C_{75}$
$A_6$	0	0	0	0	$C_{76}$
$A_7$	0	0	1	$C_{54}$	$C_{77}$

Table 2: Inputs and outputs of proposed architecture.

Index IN	Input configurations					Index OUT	Output configurations				
	DFT size						DFT size				
	2	3	4	5	7		2	3	4	5	7
0	0	$x(0)$	0	$x(0)$	$x(0)$	0	$X(0)$	$X(0)$	$X(0)$	$X(0)$	$X(0)$
1	$x(0)$	$x(1)$	$x(0)$	$x(1)$	$x(1)$	1	—	$X(1)$	$X(1)$	$X(1)$	$X(1)$
2	0	0	0	0	$x(2)$	2	—	—	$X(3)$	$X(2)$	$X(2)$
3	0	0	$x(3)$	$x(2)$	$x(3)$	3	$X(1)$	—	$X(2)$	—	$X(3)$
4	0	0	$x(1)$	$x(3)$	$x(4)$	4	—	—	—	—	$X(4)$
5	0	0	0	0	$x(5)$	5	—	—	—	$X(3)$	$X(5)$
6	$x(1)$	$x(2)$	$x(2)$	$x(4)$	$x(6)$	6	—	$X(2)$	—	$X(4)$	$X(6)$

Table 3: Control signals to obtain different DFT sizes.

DFT size	$s_0$	$s_1$	$s_2$	$s_3$
2-point	—	—	—	0
3-point	0	—	0	0
4-point	0	0	1	0
5-point	0	1	1	1
7-point	1	1	1	1

- 2 C. S. Burrus and P. Eschenbacher, "An in-place, in-order prime factor FFT algorithm," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 29, no. 4, pp. 806–817, Apr. 1981.
- 3 S. Winograd, "On computing the discrete Fourier transform," *Nat. Acad. Sci. USA.*, vol. 73, no. 4, pp. 1005–1006, Apr. 1976.
- 4 P. Duhamel and M. Vetterli, "Fast Fourier transforms - a tutorial review and a state of the art," *Signal Process.*, vol. 19, no. 4, pp. 259–299, Apr. 1990.
- 5 Z.-X. Yang, Y.-P. Hu, C.-Y. Pan, and L. Yang, "Design of a 3780-point IFFT processor for TDS-OFDM," *IEEE Trans. Broadcast.*, vol. 48, no. 1, pp. 57–61, 2002.
- 6 F. Camarda, J.-C. Prevotet, and F. Nouvel, "Implementation of a reconfigurable fast Fourier transform application to digital terrestrial television broadcasting," in *Proc. Int. Workshop Field-Programmable Logic Applications*, 2009, pp. 353–358.
- 7 J. Löfgren and P. Nilsson, "On hardware implementation of radix 3 and radix 5 FFT kernels for LTE systems," in *Proc. Norchip*, 2011, pp. 1–4.
- 8 C.-F. Hsiao, Y. Chen, and C.-Y. Lee, "A generalized mixed-radix algorithm for memory-based FFT processors," *IEEE Trans. Circuits Syst. II*, vol. 57, no. 1, pp. 26–30, Jan. 2010.