Reconfigurable 2, 3 and 5-point DFT processing element for SDF FFT architecture using fast cyclic convolution algorithm

Bibin Sam Paul S.✉, A.X. Glittas, M. Sellathurai and G. Lakshminarayanan

In this Letter, a reconfigurable processing element (PE) for pipelined SDF FFT architecture is presented, which can be configured to compute 2, 3 and 5-point DFTs. Foremost, the proposed PE architecture for the 5-point DFT computation is designed by factorising the 5-point DFT computation operation into 2×2 cyclic convolution units and then the 2- and 3-point DFTs structures are mapped on to it using multiplexers. Thus, all three configurations are possible. In the case of prior 5-point PE designs, the PE can start its operation only after the arrival of all the five-input data, whereas the proposed PE completes a part of computation after the arrival of the first three inputs and reuse the same hardware to process the next two inputs. As a result, the proposed PE requires less hardware, at the same time, preserving the throughput of prior PE. The proposed PE required 25% less multiplier and one adder less compared to the Winograd algorithm based 5-input PE.

Introduction: Extensive research on fast Fourier transform (FFT) algorithm has made FFT as one of the prominent computation units in application domain such as telecommunication, spectrum analysis, polynomial multiplication and wireless communication. The Cooley–Tukey FFT algorithm for composite N-length is popularly used due to the low complex uniform implementation structures. Prime factor FFT algorithm (PFA) requires relatively co-prime factors to compute an N-point DFT, whereas Winograd Fourier transform algorithm (WFTA) uses short prime length DFT processing elements (PEs) because of the minimum multiplier count used in the PE architecture [1]. Even though PFA does not require twiddle factors between stages, it failed to replace Cooley–Tukey FFT in many places because of the irregular PE structures used for implementation [2].

Recent wireless standards-based transceiver systems require FFT architectures that can compute powers-of-two FFT points (16-4096) and non-powers-of-two DFTs (12-2400) [3]. PEs capable of computing 2, 3, 5 and higher radixes are required to design such a combined multi-mode FFT architecture. Typically for low data rate applications, memory-based FFT processors with different PEs for each radix or a combined PE comprise of powers of two (2, 4, …etc.) and prime length (3, 5, …etc.) radixes are used. Winograd algorithm based unified PE [4] capable of computing 2, 3, 4, 5 and 7-point DFTs is one of the best PE for memory-based architecture.

Although non-powers-of-two PEs are hardware expensive, pipeline FFT architecture with PE at every stage is the only choice for high data rate applications. Among the variety of pipeline FFT architectures, single delay feedback (SDF) structure is preferably used to design multi-mode FFT architecture. A combined PE (radix-5, radix-3^2 and radix-2^4)-based SDF FFT architecture that supports 46 different FFT sizes defined in the 3GPP-LTE standard is designed in [5]. The trade-off between the hardware cost versus the number of different FFT points that can be realised as per 3GPP-LTE standard limits the direct implementation of higher radix PE architecture, whereas the higher radix PEs are designed from lower radix (2, 3 and 5-point) DFTs [6].

The proposed 2, 3 and 5-point PE architecture is designed based on the 5-point DFT computation PE architecture, and the lower radix DFTs (2- and 3-point) are mapped on to it using multiplexers (MUX). In an SDF FFT structure, a conventional 5-point DFT PE starts computation after four data are loaded into the scheduling buffers. The proposed PE starts computation after three data are loaded into the scheduling buffers and utilises two cycles for computation for re-using the hardware without affecting the pipeline flow of the SDF FFT structure. In the first cycle, it takes three inputs and completes a part of the 5-point DFT operation. In the next cycle, it completes the rest of the 5-point DFT operation using the same hardware. To re-use the hardware for two cycles, the PE is designed based on the fine-grained decomposition of 5-point DFT into 2×2 cyclic convolution sub-units.

4×4 *cyclic convolution architecture:* The proposed DFT computation PE architecture is designed based on the approach that a prime N-length DFT computation is formulated as an $N-1$ length cyclic convolution operation [1]. Therefore the 5-point DFT computation PE structure is designed using the 4×4 cyclic convolution operation. The cyclic convolution architecture, along with the circuit to process the DC component results in a complete PE structure for DFT computation. The architecture for the 4×4 cyclic convolution operation is designed using 2×2 cyclic convolution sub-unit [7] as per the decomposition (1) so that the proposed architecture re-uses the 2×2 cyclic convolution structure to minimise the hardware complexity. The formulation has three 2×2 cyclic convolution operation with $X_0, X_1, (X_0+X_1)$ as inputs and H_0, H_1, (H_0+H_1) as twiddle factors, respectively. Where X_i, H_i and Y_i, with $i = 0, 1$ are vectors of length two.

$$\begin{bmatrix} Y_0 \\ Y_1 \end{bmatrix} = \begin{bmatrix} I_2 & 0 & S_2 \\ -I_2 & I_2 & -I_2 \end{bmatrix} \cdot \mathrm{diag} \begin{bmatrix} H_0 \\ H_0 + H_1 \\ H_1 \end{bmatrix} . \begin{bmatrix} X_0 \\ X_0 + X_1 \\ X_1 \end{bmatrix} \quad (1)$$

By applying the formulations stated in [8] the input, output order and it's corresponding twiddle factor for the three cyclic convolution operations in the 4×4 cyclic convolution computation are given in the following equations:

$$X_0 = [x(2), x(3)], \quad Y_0 = [y(2), y(3)], \quad H_0 = \left[\frac{W_5^2 + W_5^{-2}}{2}, \frac{W_5^2 - W_5^{-2}}{2}\right] \quad (2)$$

$$X_1 = [x(4), x(1)], \quad Y_1 = [y(4), y(1)], \quad H_1 = \left[\frac{W_5^1 + W_5^{-1}}{2}, \frac{W_5^1 - W_5^{-1}}{2}\right] \quad (3)$$

$$\begin{aligned} X_0 + X_1 &= [x(2) + x(4), x(3) + x(1)], \\ H_0 + H_1 &= [Re(W_5^2 + W_5^1), Im(W_5^2 + W_5^1)]. \end{aligned} \quad (4)$$

As like WFTA, the twiddle factors are either purely real or purely complex that requires half the hardware cost compared to a full complex multiplier known as semi-complex multiplier [4]. One more fact about the twiddle factors is that the H_0+H_1 twiddle factor given in (4) has a trivial term $(Re(W_5^2 + W_5^1) = -0.5)$.

The 4×4 cyclic convolution architecture designed using 2×2 cyclic convolution sub-units (PE-1, PE-2) and a combiner unit (CU) is shown in Fig. 1. PE-1 and PE-2 are 2×2 cyclic convolution structures designed using fast cyclic convolution algorithm [7] given in (5) and (6) with (x_0, x_1) corresponds to inputs and

$$\left(\frac{h_0 + h_1}{2}, \frac{h_0 - h_1}{2}\right)$$

corresponds to twiddle factors.

$$y_0 = (x_0 + x_1)\frac{h_0 + h_1}{2} + (x_0 - x_1)\frac{h_0 - h_1}{2}, \quad (5)$$

$$y_1 = (x_0 + x_1)\frac{h_0 + h_1}{2} - (x_0 - x_1)\frac{h_0 - h_1}{2}. \quad (6)$$

The proposed architecture computes the 4×4 cyclic convolution operation in two-time instants. PE-1 is re-used for computing the two cyclic convolution operations given in (2) and (3). PE-2 is used to compute the cyclic convolution operation given in (4) in two-time instants. To compute the sum of cyclic convolution operation PE-2 structure uses a pair of an adder and a register to accumulate the cyclic convolved output at the two-time instants. In the first cycle, PE-1 computes the cyclic convolution of X_0 and H_0 and PE-2 computes the cyclic convolution of X_0 with H_0+H_1. The computed results from PE-1 and PE-2 are stored in d_1, d_2 and d_3, d_4 registers, respectively, for using it in the next cycle. In the second cycle, PE-1 inputs are changed to X_1 and H_1 so that PE-1 computes the cyclic convolution of X_1 with H_1 and PE-2 computes the cyclic convolution of X_1 with H_0+H_1. During the second cycle, the cyclic convolved output of PE-2 is accumulated with the previously stored result in d_3 and d_4 to compute the sum of the cyclic convolution operation. The combiner unit connected to the output of PE-1, PE-2 and the delay buffers (d_1 and d_2) is used to combine the three cyclic convolved outputs as per the formulations given in [7] to get the 4×4 cyclic convolution output $(y(2), y(3), y(4)\ y(1))$. The combiner unit uses 6 adders, PE-1 and PE-2 require 4 adders each so that the overall architecture uses 14 adders, 3 semi-complex multipliers and 4 registers.

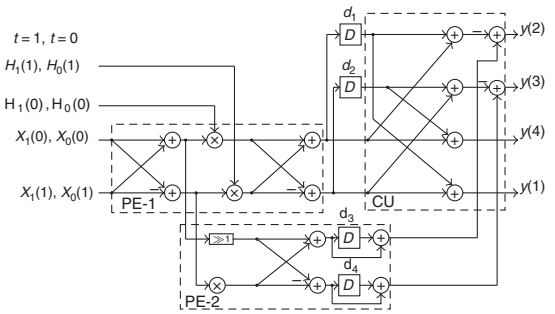

Fig. 1 4 × 4 *cyclic convolution architecture*

Proposed 2, 3 and 5-point processing element: The proposed 5-point DFT PE uses the 4 × 4 cyclic convolution architecture as a core computation unit. The 4 × 4 cyclic convolved output is added with $x(0)$ to obtain the four DFT computed results and the DC component of the spectrum is obtained by summing up all the inputs. The PE consist of three inputs of which 4 × 4 cyclic convolution architecture takes two inputs, and one input is for the DC computation circuit. The proposed PE is attached to the SDF FFT architecture due to the configurable data scheduling arrangement of the SDF structure based on the number of inputs in the PE. An SDF FFT architecture stage comprises of the proposed PE and the input scheduling buffers used for converting a single input into parallel inputs is shown in Fig. 2 to explain the operation of the PE. As the size of the delay buffers vary with the stage in which the PE is attached, for easy understanding of the PE operation assume that the PE is attached in the last stage with all the buffer size as one. The buffer size denoted as 'D' is selected according to the stage in which the PE is connected for operating in higher stages. The connection between the input of the scheduling buffers and the PE output is indicated as labels ($d1_{\text{in}}$, $d2_{\text{in}}$, $d3_{\text{in}}$, $d4_{\text{in}}$) for clarity in the diagram. The input to the SDF FFT stage is denoted as x_{in} and the output x_{out} of the stage considering only the 5-point DFT output is shown for clarity in the diagram.

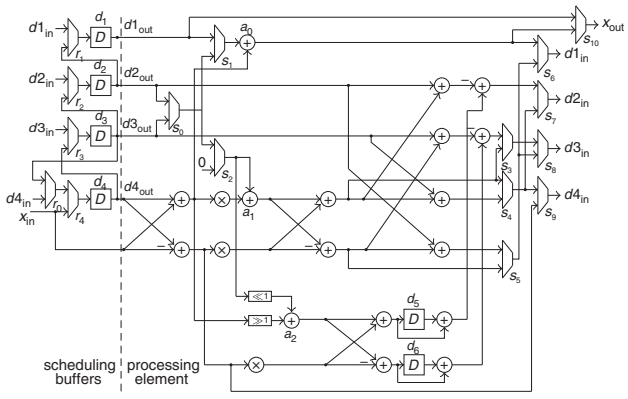

Fig. 2 *Proposed 2, 3 and 5-point PE for SDF FFT architecture*

As the cyclic convolution architecture used to design the PE requires two cycles for computation, the proposed PE takes two cycles for computation. The input data is loaded into the scheduling buffers through x_{in} for three cycles and the PE starts computation in the third cycle. In the third cycle, the PE takes three data [$x(0)$, $x(2)$, $x(3)$] in which $x(0)$, $x(2)$ are fed from scheduling buffer output ($d2_{\text{out}}$, $d4_{\text{out}}$) and $x(3)$ is direct x_{in}. In the fourth cycle, [$x(1)$, $x(4)$] are fed to the PE, where $x(1)$ from scheduling buffer output ($d4_{\text{out}}$) and $x(4)$ is direct x_{in}. As the required input order as per (3) is [$x(4)$, $x(1)$] the signs of the corresponding precomputed twiddle factors given in (3) and (4) are inverted and fed to PE structure. To operate the 4 × 4 cyclic convolution architecture for two time instants, in the third cycle the next input $x(1)$ from the scheduling buffer output ($d3_{\text{out}}$) is routed to d_4 buffer output ($d4_{\text{out}}$) by configuring the multiplexer contol signals r_0, r_4. In the same cycle, the two outputs from the PE-1 structure are routed to the unoccupied delay registers d_2 and d_3 for using it in the fourth cycle by configuring the control signals (s_4, s_7, r_2) and (s_5, s_8, r_3) of the multiplexers, respectively. The 4 × 4 cyclic convolution computation unit works as per the operations explained in the previous section and generates four cyclic convolved results.

The DC component $x(0)$ is added to the convolved output using two adders a_1 and a_2 attached to PE-1 and PE-2 structures, respectively. Since PE-2 structure involves the sum of two cyclic convolution operation $2x(0)$ is added to the PE-2 output in which the term $2x(0)$ is generated using a left shifter. The a_1 and a_2 adder inputs are connected to $x(0)$ through a single-gate multiplexer [4] so that the adder is given with $x(0)$ in the third cycle and zero in the fourth cycle. The DC component of the spectrum is computed sequentially in the third and fourth cycles utilising the input side butterfly sum output of the PE-1 structure and an adder a_0 attached at the output of the d_1 buffer ($d1_{\text{out}}$). In the third cycle, $x(0)$ is added with the butterfly output and the result is stored to d_1 buffer by configuring the control signal s_6, r_1. Subsequently, in the fourth cycle, the stored result is added with the butterfly output by configuring the control signal s_1 to compute the DC component of the spectrum. The four DFT computed outputs from CU are ordered and routed to the input scheduling buffers in the fourth cycle using multiplexers connected at the CU outputs.

The 5-point PE architecture is designed using the inherent 3-point computation unit (2 × 2 cyclic convolution) so that 3- and 2-point DFTs are mapped on to it by configuring the control signals of the multiplexers as given Table 1. The control signals for the 2:1 multiplexers can be generated from the SDF architecture stage control signals. Overall the combined PE architecture requires 3 multipliers, 17 adders and few multiplexers. Excluding the SDF structure multiplexers, the 5-point PE requires 7 multiplexers and the 3, 2-point DFTs are mapped on to it using 3 multiplexers. The proposed 5-point DFT PE, when implemented in UMC 65 nm CMOS technology results in 20.4% reduction of gate count compared to the WFTA-based 5-input fully parallel PE.

Table 1: Control signals for DFT computation

MUX control signals																
DFT size	Clk	Scheduling buffers					Processing element									
		r_0	r_1	r_2	r_3	r_4	s_0	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8	s_9	s_{10}
5	0	—	0	0	0	0	—	—	—	—	—	—	—	—	—	—	0
	1	—	0	0	0	0	—	—	—	—	—	—	—	—	—	—	0
	2	—	0	0	0	0	—	—	—	—	—	—	—	—	—	—	0
	3	1	1	1	1	1	1	0	1	—	1	1	0	0	0	—	0
	4	0	1	1	1	1	—	1	0	1	0	0	1	1	1	1	1
3	0	—	—	—	0	0	—										
	1	—	—	—	0	0	—										
	2	0	—	—	1	1	0	0	1	0	1	1	—	—	1	1	—
2	0	—	—	—	—	0	—										
	1	0	—	—	—	1	—	—	—	—	—	—	—	—	—	0	—

Conclusion: In this Letter, a reconfigurable 2, 3 and 5-point DFT PE suitable for SDF FFT architecture is proposed. A low complex 5-point PE is designed using the granular form of the 5-point DFT computation operation using 2 × 2 cyclic convolution operation. The PE utilises the data load cycle of the SDF FFT architecture effectively for sharing the computational unit of the PE using multiplexers. The proposed PE architecture finds application in the design of the FFT system that supports multiple FFT lengths defined in the 3GPP-LTE standard.

Acknowledgment: This work was supported by UKIERI and SMDP C2SD projects.

Submitted: *13 January 2020* E-first: *7 April 2020*
doi: 10.1049/el.2019.4262
One or more of the Figures in this Letter are available in colour online.

Bibin Sam Paul S., A.X. Glittas and G. Lakshminarayanan (*Department of Electronics and Communication Engineering, National Institute of Technology Tiruchirappalli, India*)

✉ E-mail: nglbibin@gmail.com

M. Sellathurai (*School of Engineering & Physical Sciences, Heriot-Watt University, Edinburgh, UK*)

References

1 Winograd, S.: 'On computing the discrete Fourier transform', *Math. Comput.*, 1978, **2**, (141), pp. 175–199

2 Bhagat, N., Valencia, D., Alimohammad, A., *et al*.: 'High-throughput and compact FFT architectures using the good-Thomas and Winograd algorithms', *IET Commun.*, 2018, **12**, (8), pp. 1011–1018

3 Evolved Universal Terrestrial Radio Access (E-UTRA); LTE Physical Channels and Modulation, document 3GPP TS 36.211, 2012

4 Qureshi, F., Garrido, M., and Gustafsson, O.: 'Unified architecture for 2, 3, 4, 5, and 7-point DFTs based on Winograd Fourier transform algorithm', *Electron. Lett.*, 2013, **49**, (5), pp. 348–349

5 Shih, X., Chou, H., and Liu, Y.: 'VLSI design and implementation of reconfigurable 46-mode combined-radix-based FFT hardware architecture for 3GPP-LTE applications', *IEEE Trans. Circuits Syst. I*, 2018, **65**, (1), pp. 118–129

6 Chen, J., Hu, J., Lee, S., *et al*.: 'Hardware efficient mixed radix-25/16/9 FFT for LTE systems', *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, 2015, **23**, (2), pp. 221–229

7 Cheng, C., and Parhi, K.K.: 'Hardware efficient fast DCT based on novel cyclic convolution structures', *IEEE Trans. Signal Process.*, 2006, **54**, (11), pp. 4419–4434

8 Cheng, C., and Parhi, K.K.: 'Low-cost fast VLSI algorithm for discrete Fourier transform', *IEEE Trans. Circuits Syst. I*, 2007, **54**, (4), pp. 791–806