

PAPER • OPEN ACCESS

Design and Implementation of AGU based FFT Pipeline Architecture

To cite this article: G. Prasanna Kumar *et al* 2021 *J. Phys.: Conf. Ser.* **2089** 012070

View the [article online](#) for updates and enhancements.

You may also like

- [A real-time spectral analysis method and its FPGA implementation for long-sequence signals](#)
Pingkun Xu and Feiyun Xu
- [Design and implementation in VHDL code of the two-dimensional fast Fourier transform for frequency filtering, convolution and correlation operations](#)
Juan M Vildary, F Giacometto, C O Torres et al.
- [A high performance fast-Fourier-transform spectrum analyzer for measuring spin noise spectrums](#)
Yu Tong, , Lin Wang et al.



The Electrochemical Society
Advancing solid state & electrochemical science & technology

241st ECS Meeting

May 29 – June 2, 2022 Vancouver • BC • Canada

Extended abstract submission deadline: Dec 17, 2021

Connect. Engage. Champion. Empower. Accelerate.
Move science forward



Submit your abstract



Design and Implementation of AGU based FFT Pipeline Architecture

G. Prasanna Kumar¹, Maturi Sarath Chandra², K Shiva Prasanna³, M Mahesh⁴

¹Associate professor, ECE department, Malla Reddy Engineering College (Autonomous), Hyderabad, Telangana¹

²Assistant Professor, ECE department, Maturi Venkata Subba Rao Engineering College, Hyderabad, Telangana²

³Assistant professor, ECE department, Teegala Krishna Reddy engineering College, Telangana³

⁴Assistant professor, ECE department, Sreenidhi institute of Science and technology, Telangana⁴
Email: prasanna4600@gmail.com

Abstract. Present it is most needful task to get various applications with parallel computations by using a Fast Fourier Transform (FFT) and the derived outputs should be in regular format. This can be achieved by using an advanced technique called Multipath delay commutator (MDC) Pipelining FFT processor and this processor will be capable to perform the computation of a different data streams at a time. In this paper the design and implementation of AGU based Pipelined FFT architecture is done Calculation of a butterfly is done within 2 cycles by the instructions proposed. A Data Processing Unit (DPU) is employed in this pipeline architecture and supports the instructions & an FFT Address Generation Unit (FAGU) calculates butterfly input & output data addresses automatically. The DPU proposed system requires less area compared to commercial DSP chips. Furthermore, the proposed FAGU reduces the number of FFT computation cycles. The FFT design architecture will have real data paths. With various FFT sizes, different radix & various parallelism levels, the FFT can be mapped to the pipeline architecture. The most attractive feature of the pipelined FFT architecture is it consists of bit reversal operation so it requires little number of registers and better throughput.

Keywords: Fast Fourier transforms (FFT), parallel computation, pipelining, real data path, bit reversal, data stream and twiddle factor.

1. Introduction

In DSP (digital signal processing) applications important operations are FFT (fast fourier transform) & IFFT (inverse fast fourier transform). Basically, the FFT / IFFT operates on complex numbers & produce complex numbers. For the FFT & IFFT complex numbers computation (Complex FFT/Complex IFFT) many regular designs were presented. For real-valued samples [real FFT (RFFT) / real IFFT (RIFFT)], interest has been increasing in the computation of FFT / IFFT, real numbers were represented with several physical signals like biomedical signals. Input samples frequency spectrum (RFFT input / RFFT output) is Hermitian symmetric when they are real in the time domain so, that half of the computations are redundant approximately. For reducing the complexity of hardware this property is exploited. Here in this literature several architecture designs & implementations of RFFT has been presented. Pipeline architectures are dedicated to RFFT are suggested first. This architecture computes $2N$ - point RFFT by a N - point CFFT architecture, and more efficient compared to other architectures. These architecture is composed with real data paths only. By utilizing radix - 2 algorithm the RFFT architectures are suggested for 4 - parallel



implementations. Whether this approach will be generalized for arbitrary index and the 2 – parallel architectures will be utilized for RFFT by real data paths only have open remained questions.

In present days for RFFT, pipelined architectures has been presented, these type of architectures are composed with hybrid data paths that can contain both real & complex data paths. Twofold is the brief contribution of this. First it is observed that the implementation of 2 – parallel RFFT architectures can be done by only real data paths and opposes the hybrid path usage. Second for implementing the architecture of RFFT / RIFFT design with any size of FFT, any parallel level & any power – of – two radix, a systematic approach has been presented. Real data paths only utilized for all architectures presented here, which can lead to reducing delays, number of adders & interconnected wires. This proposed architecture needs only one kind of BF (butterfly) compared to hybrid architectures & other 4 kinds utilized. So that presented designs were regular more. However in this brief only DIF (decimation-in-frequency) architectures are suggested. Hence this approach can also be utilized for designing the architectures by DIT (decimation in time) algorithms.

2. Multipath Delay Commutator (MDC) FFT

FFT is the most basically utilized operation in the applications of wireless communications like ultra wideband, OFDM (orthogonal frequency division multiple) access, digital video broadcast terrestrial, as well as signal processing application. A FFT pipelined architecture family is discussed with very popular SDF (single – path delay feedback), MDC (multipath delay commutator). Here the applications like array signal processing, image processing, multiple I/O OFDM & etc, where more than 1 data stream is required for processing. Hence for generating the outputs in natural order simultaneous multiple FFT operations & a dedicated bit reversal circuits are needed. Multiple independent data streams are handled by FFT structures. While one FFT processor processes the all the data streams. One by one 4 independent data streams can be processed. In the same way for 2 domains 8 data streams are processed. So that in parallel multiple data stream outputs are not available. More than 1 FFT processor is required for the simultaneously processing the data streams. For wireless LAN (local area application) application 1 to 4 data streams are processed by utilizing the many data paths. For simultaneous process data of various data streams were interleaved. However these architectures does not contain any particular bit reversal circuit. For reordering the bit reversed FFT output in natural order certain circuits were presented.

For various radices, bit reversal circuits are presented in similar architecture for variable length of FFT, N is the register complexity of it. These type of circuits are suitable to bit reversing the data in the FFT pipelined structure. Bit reversal circuits only suggested. The bit reversal structure is integrated for FFT structure which can result as design total register requirement is minimized from $5N/2$ to $2N$. The parallel pipelined architectures of 2-, 4-, 8-, radix – $2k$ DIF feed forward FFT structures are suggested [10] & requires additional N registers for generating the output in normal order.

Furthermore, these 2-, 4-, & 8 – parallel FFT structure will only start their operations if $x(n + N/2)$, $x(n+3N/4)$ & $x(n+7N/8)$ samples occurs. For storing the first $N/2$, $3N/4$, $7N/8$ samples, hardware is underused & extra registers are needed. The modified MDC FFT architecture with rearranging structure & a new scheduling model is presented & the drawbacks are eliminated. The architectures operated at the incoming sample frequency rate but this structure operates at the half of clock frequency for generating the same throughput as f . The architecture throughput is twice, when all of the structures are operated at same speed. In the same manner, combined single path delay commutator – SDF FFT architecture with normal order of I/O is suggested where bit reversal is performed with $N/2$ registers only. Although number of registers required is high & its throughput is low. Less complexity FFT architectures are suggested and these only processes real – valued signals. These can generates 2 outputs per 1 clock cycle & the generated outputs are not in normal order. So, many structures need bit reversal circuits for generating output in natural order.

3. Radix-2 Pipelined FFT Architecture

The Figure 1 represents the computing thought of an N – point FFT utilizing $2 N/2$ – point FFT operations with extra 1 stage of BF operations, that is not exact structure but methodology is provided. The reordering blocks are only present to state that after the completion of $N/2$ - point DIF FFT operation reordered the $N/2$ even samples $x(2n)$ and before the starting of $N/2$ -point DIT FFT operation reordered the $N/2$ odd samples($x(2n + 1)$). For computing the N – point DIT FFT from the $2 N/2$ – point FFTs outputs, an extra BF stage operations are carried out over 2 FFTs results. So that the extra BF stage generates outputs in normal order.

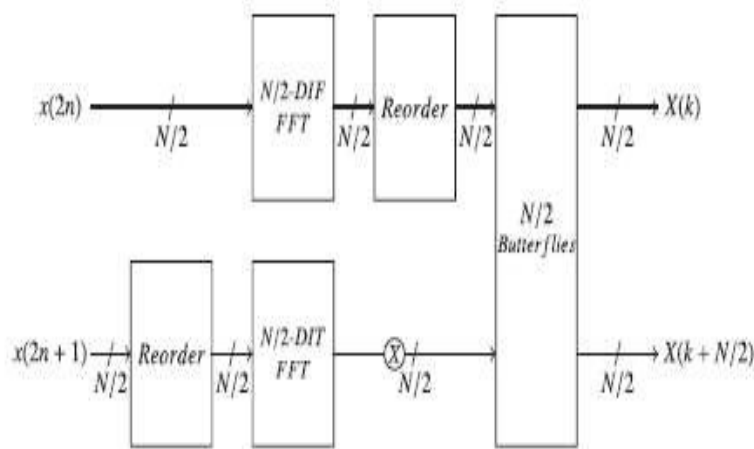


Fig. 1. Idea of the proposed method

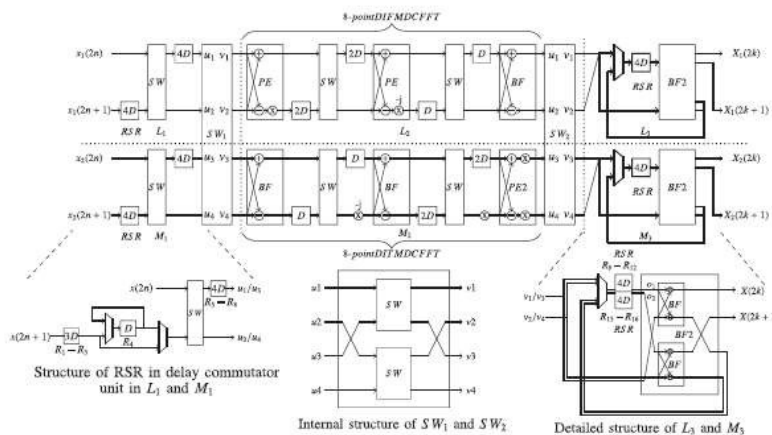


Fig. 2. Proposed 16-point radix-2 FFT architecture with outputs in natural order

The Figure 2 represents the proposed 16 – point FFT architecture. This architecture contains two 8 – point MDC FFT structures for processing the 2 data streams. At the SW1 left side delay commutator unit is presented to dissociate the even & odd samples. The delay commutators present in the shift registers that can receive inputs, and are utilized for odd input samples bit reversing. These type of shift registers are known as RSRs (reordering shift registers). The RSRs stores the outputs in

the last stage from the 8 – point DIF FFT & performs bit reverse operation. The butterfly2 performs 2 – parallel BF operations between the bit reversed data from last stage of RSR & from the outputs of 8 – point DIT FFT. So the lower & upper BF2 in last stage generates the data stream outputs of 1st & 2 nd FFTs in natural order. The 2 data paths are combined from SW2, thus the data path word length in last stage is doubled & thick lines were utilized to represent the registers and datapaths.

In Figure 1 the FFT architecture is divided into 6 levels (M1, M2, M3 ,L1, L2 & L3). The RSR registers in the M1 & L1 levels reorders the odd input data. And the even data can be partially processed by the L3 & M3 levels of RSR registers. In L2 & M2 levels the 8 – point DIT & DIF FFT operations are carried out. The data from M1 & L1 will be forwarded to M2 & L2 sequentially or vice versa by the SW1. In the same manner data from L2 & M2 will be forwarded to M3 & L3, respectively or vice versa by the SW2. The SW1 & SW2 consist 2 switches for propagating the data to various levels & swapping the data paths. In the normal mode SW1 or SW2 (switches) passes the data from u_1, u_2, u_3 & u_4 to v_1, v_2, v_3 & v_4 approximately. While in swap mode the SW1 or SW2 passes the data from u_1, u_2, u_3 & u_4 to v_3, v_4, v_1 & v_2 , respectively. In the first $N/2$ clock cycles SW1 is in swap mode & during $N/2 + 1$ to N is in normal mode. While for first $N/2$ clock cycles the SW2 is in normal mode & from $N/2 + 1$ to N in the swap mode. So at any time the SW1 & SW2 are in different modes and for each $N/2$ clock cycles they can change their modes.

Furthermore, SW1 or SW2 are in normal mode, when data transition between M_y & M_{y+1} or L_y & L_{y+1} (here y will be 1 or 2) & the SW1 or SW2 are in swap mode, when the data transition between L_y & M_{y+1} or M_y & L_{y+1} . The control signals to the switches SW1 or SW2 are provided externally like remaining control signals in the design, for every $N/2$ clock cycles these control signals swap their modes. To the FFT processor the 2 input streams are represented as X_1 & X_2 . The delay commutator disassociates the 2 input streams odd & even samples in M1 & L1 (X_1 is disassociate as $\{E_1(i, j), O_1(i, j)\}$ & X_2 is disassociate as $\{E_2(i, j), O_2(i, j)\}$ respectively). Where i denotes the data nature & j denotes the data sets number whose FFT has to be computed. The input data odd set $[x(1), x(3), x(5) \dots]$ is denoted as $O(1, j)$. $E(2, j)/O(2, j)$ & input data even set $[x(0), x(2), x(4) \dots]$ is denoted as $E(1, j)$ is the ordered or odd/even data scheduled, that are ready for applying to 8 – point DIT/ DIF FFT. The 8 – point DIT/DIF FFT outputs are denoted as $E(3, j)/O(3, j)$ that are applied to 3rd level to compute 16 – point FFT.

4. AGU Based FFT Pipelined

Each imaginary datum & real are marked as $_$ and o , respectively. The Figure 3 shows 1 real number and every complex datum(\bullet) represents 2 real numbers. Based on this each column consist N real numbers. So the structure will be transformed to Ncomplete rows by the seperation of complex data into real & imaginary parts.

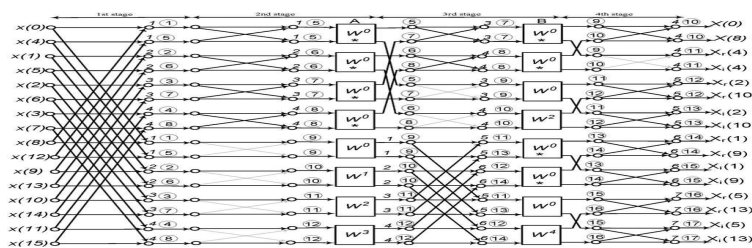


Fig. 3. Regular flow graph corresponding to Pipelined FFT

The seperation of real & imaginary parts are shown in above figure. The twiddle factors are transferred to required subsequent stages, so the transformed flow graph consist 1 column of BFs follow by 1 column of W_k - blocks & every column from the flow graph consist a total of N imaginary & real samples. First & last stages does not require W_k - blocks. After this transformation the obtained regular structure is shown in Figure 3.

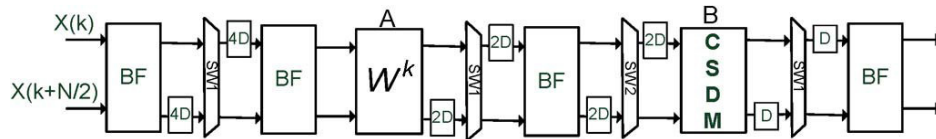


Fig. 4. Proposed two-parallel pipelined VLSI architecture for a 16-point DIF RFFT

The italic & circled numbers from Figure 3 represents the computation time instances. The timing instances of circle & italic are utilized to derive the 2- parallel & 4 – prallellel architectures. This model is most suitable to normal parallel pipelined RFFT design architectures than the structures [8]. During the next step, Wk - blocks & BFs each column is mapped to a Wk - block & a BF circuit. From the data flow (Fig. 2) the 2 – parallel pipelined architecture derived is shown in Figure 3. Utilizing real data paths in the first 2 – parallel pipelined architecture for RFFT computations. The BF block internal circuit is shown in Figure 4. For 2 real inputs, the BFs subtracts or adds 2 real numbers, but for the inputs of real – imaginary, simply input data is fed to output. The switching operation between these 2 cases can be controlled by $c1$ control signal. In the third stage the multiplication by $W2$ is done only in the Wk - block which can require actual multiplication. By 2 addition operations & a scaling operation by $1/\sqrt{2}$ constant it will be implemented. By CSD (canonical-signed-digit) multiplication the scaling operation will be implemented [11]. Hence in the third stage the column of Wk - blocks are mapped to a CSDM (CSD multiplier) which is hardware efficient compared to a complex multiplier. Synchronization of data with the timing is depicted in Figure 3, delay elements & switches are denoted as shuffling circuits and are utilized. The Figure 4 represents the used switch types. The $c1$ control signal for SW1 switch is periodic with $2m$ clock cycles period. For SW2 switch, 1st $2m$ lower & upper sample inputs passes through the switch with m delay to lower & upper outputs.

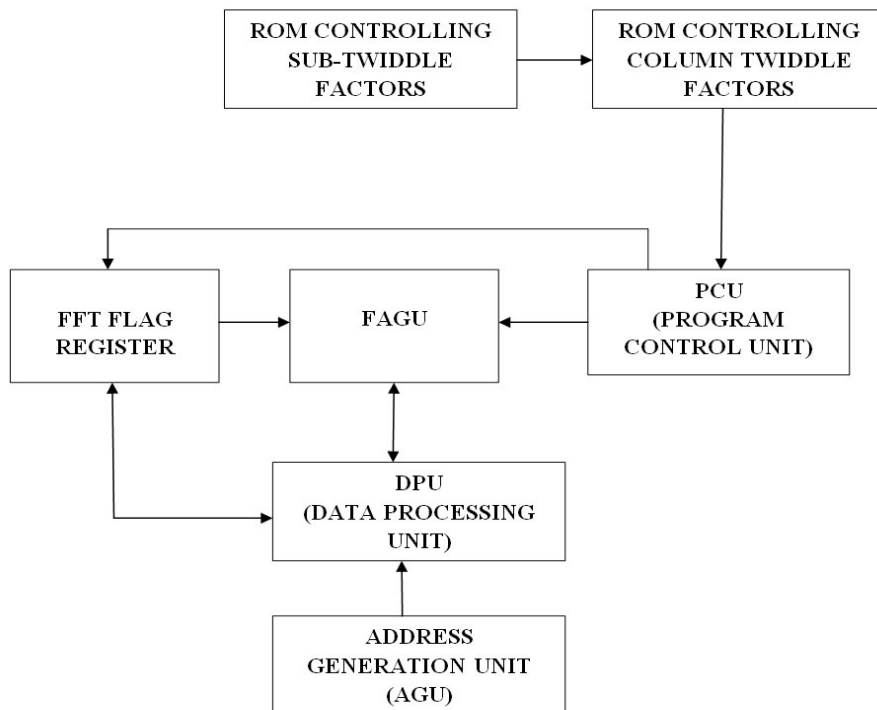


Fig. 5: AGU Based FFT Pipelined Architecture

The above figure (5) shows the block diagram of AGU based FFT pipelined architecture. This proposed FAGU contains many registers, counters, etc. The FAGU output addresses are loaded in to the address register of AGU. Then the loaded addresses are utilized to offset address of base address. For example, if R0 is the base address register & N0 is the offset register, then the calculated address is the output address of AGU. When the ADSP executing FFT instructions then the FAGU should be enabled. Either the FAGU is active or not is identified by the flag register of FFT. New instruction FFT #N is suggested for controlling the flag register of FFT. Where N denotes the number of FFT points. In FFT computations the input data addresses for a BF are marked with respect to the number of FFT points. The number of FFT points determines input data addresses order. Hence the FAGU should know the number of FFT points then only it can calculate the addresses of BFs input data.

The switch operates similarly as SW1 after the 2m clock cycles that can illustrate the dataflow to SW2 switch type. The c2 control signal is initialized with 2m 0s follow by similar periodic pattern like c1. For a N – point RFFT pipeline architecture producing the required control signals in various stages for all the switches utilizes a single (n – 1) - bit counter, where $n = \log_2(N)$. A 2 – parallel pipelined architecture use only real data paths for deriving the 16 – point radix – 2RFFT from the presented regular structure. The architecture that requires two demultiplexers. The suggested technique to design a 2 – parallel architectures will be summarized by the following steps: i) redundant computation outputs can be eliminated & twiddle factors can be transferred to subsequent stages, as explained such that every column contains N samples; ii) separation of complex datapaths as real & imaginary datapaths; iii) transform the flow graph in to a structure with 1 BF column follow by 1 Wk – block column for every stage to obtain regularity & iv) construct an appropriate schedule & mapping regular structure in to a pipelined architecture. Each column of BF block can be mapped to 1 BF block & each Wk – block column is mapped to 1 Wk – block circuit.

5. Results

The below figure (6) shows the comparison of accuracy for FFT and AGU based FFT. The accuracy of AGU based FFT is very high.

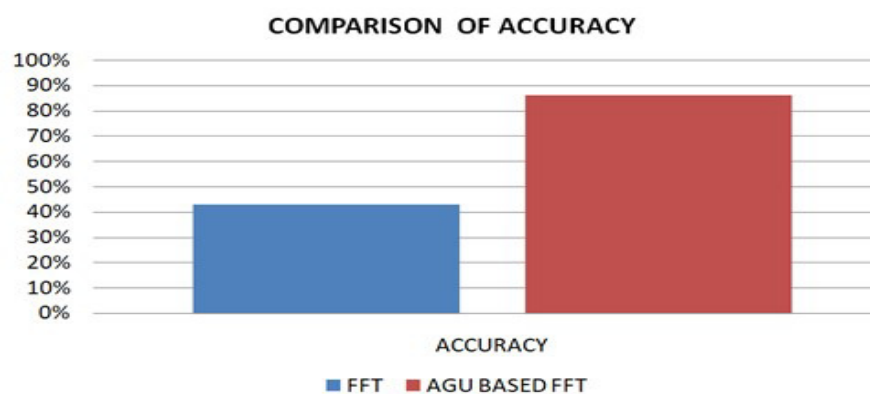


Fig. 6. Comparison of Accuracy

The below figure (7) shows the comparison of delay for FFT and AGU based FFT. The delay of AGU based FFT is reduced very effectively.

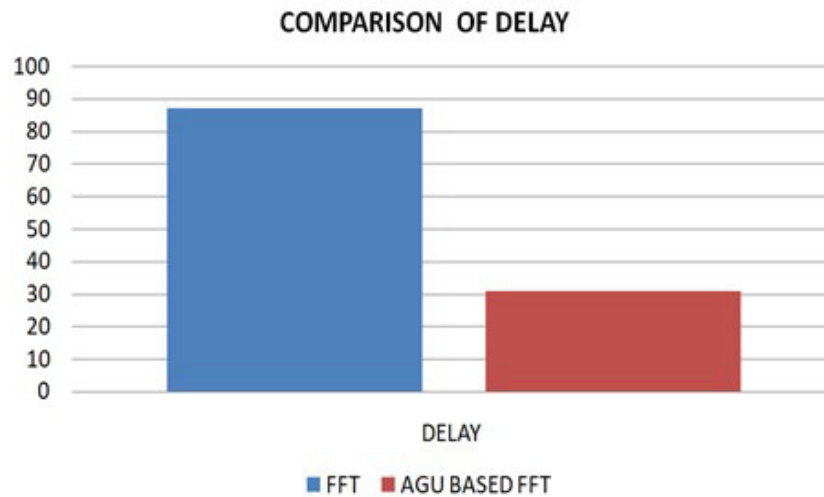


Fig. 7. Comparison of Delay

6. Conclusion

Hence in this paper the design and implementation of AGU based FFT pipelined architecture was implemented. This brief has presented a normal I/O order radix-2 multipath delay commutator FFT pipelined architecture for real-valued to computing the real signals FFT & Hermitian – symmetric signals IFFT utilizing real datapaths only, and these generated outputs are in natural order. Integrating the 2 FFT processors & registers eliminates the bit reversal circuit presented in the prior designs and these can be reused for bit reversal. By transferring the twiddle factor to subsequent stages the real structure of FFT is transformed. Every stage in the flow graph consists of 1 BF column units & 1 twiddle factor column blocks & each flow graph column consists of N samples only. These qualities made the proposed FFT processor superior in terms of performance and hardware complexity. Compared to earlier architectures this proposed FFT architecture obtains higher throughput.

References

- [1] Z. Wang, X. Liu, B. He, and F. Yu, "A combined SDC-SDF architecture for normal I/O pipelined radix-2 FFT," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 23, no. 5, pp. 973–977, May 2015.
- [2] A. X. Glittas and G. Lakshminarayanan, "Pipelined FFT architectures for real-time signal processing and wireless communication applications," in *Proc. 18th Int. Symp. VLSI Design Test*, Jul. 2014, pp. 1–2.
- [3] P. P. Boopal, M. Garrido, and O. Gustafsson, "A reconfigurable FFT architecture for variable-length and multi-streaming OFDM standards," in *Proc. IEEE ISCAS*, May 2013, pp. 2066–2070.
- [4] K.-J. Yang, S.-H. Tsai, and G. C. H. Chuang, "MDC FFT/IFFT processor with variable length for MIMO-OFDM systems," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 21, no. 4, pp. 720–731, Apr. 2013.
- [5] M. Ayinala, M. Brown, and K. K. Parhi, "Pipelined parallel FFT architectures via folding transformation," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 20, no. 6, pp. 1068–1081, Jun. 2012.
- [6] S.-N. Tang, C.-H. Liao, and T.-Y. Chang, "An area- and energy-efficient multimode FFT processor for WPAN/WLAN/WMAN systems," *IEEE J. Solid-State Circuits*, vol. 47, no. 6, pp. 1419–1435, Jul. 2012.
- [7] M. Garrido, J. Grajal, and O. Gustafsson, "Optimum circuits for bit reversal," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 58, no. 10, pp. 657–661, Oct. 2011.
- [8] Y. Voronenko and M. Püschel, "Algebraic signal processing theory: Cooley-Tukey type algorithms for real DFTs," *IEEE Trans. Signal Process.*, vol. 57, no. 1, pp. 205–222, Jan. 2009.

- [9] Y. Chen, Y.-W. Lin, Y.-C. Tsao, and C.-Y. Lee, "A 2.4-Gsample/s DVFS FFT processor for MIMO OFDM communication systems," *IEEE J. Solid-State Circuits*, vol. 43, no. 5, pp. 1260–1273, May 2008.
- [10] Y.-N. Chang, "An efficient VLSI architecture for normal I/O order pipeline FFT design," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 55, no. 12, pp. 1234–1238, Dec. 2008.
- [11] Y.-W. Lin and C.-Y. Lee, "Design of an FFT/IFFT processor for MIMO OFDM systems," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 54, no. 4, pp. 807–815, Apr. 2007.
- [12] H. F. Chi and Z. H. Lai, "A cost-effective memory-based real-valued FFT and Hermitian symmetric IFFT processor for DMT-based wire-line transmission systems," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2005, vol. 6, pp. 6006–6009.
- [13] Y. W. Lin, H. Y. Liu, and C. Y. Lee, "A 1-GS/s FFT/IFFT processor for UWB applications," *IEEE J. Solid-State Circuits*, vol. 40, no. 8, pp. 1726–1735, Aug. 2005.
- [14] A. V. Oppenheim, R. W. Schaffer, and J. R. Buck, *Discrete-Time Signal Processing.*, 2nd ed. Englewood Cliffs, NJ, USA: Prentice-Hall, 1998.
- [15] S. He and M. Torkelson, "Design and implementation of a 1024-point pipeline FFT processor," in *Proc. IEEE Custom Integr. Circuits Conf.*, Santa Clara, CA, USA, May 1998, pp. 131–134.
- [16] S. He and M. Torkelson, "A new approach to pipeline FFT processor," in *Proc. 10th Int. Parallel Process. Symp.*, 1996, pp. 766–770.