Journal of Circuits, Systems, and Computers
Vol. 27, No. 3 (2018) 1830001 (21 pages)
© World Scientific Publishing Company
DOI: 10.1142/S0218126618300015

World Scientific
www.worldscientific.com

A Survey on FFT/IFFT Processors for Next Generation Telecommunication Systems*

Elango Konguvel† and Muniandi Kannan‡

Department of Electronics Engineering,
Madras Institute of Technology Campus,
Anna University, Chennai, Tamilnadu, India
†*konguart08@gmail.com*
‡*mkannan@annauniv.edu*

Received 21 September 2016
Accepted 29 May 2017
Published 7 July 2017

The Fast Fourier Transform and Inverse Fast Fourier Transform (FFT/IFFT) are the most significant digital signal processing (DSP) techniques used in Orthogonal Frequency Division Multiplexing (OFDM)-based applications which include day-to-day wired/wireless communications, broadband access, and information sharing. The advancements in telecommunication technologies require an efficient FFT/IFFT processing device to meet the necessary specifications which depend on the particular application. A real-time implementation of high-speed FFT/IFFT processor with less area that operates in minimal power consumption is essential in designing an OFDM integrated chip. A comparative study of efficient algorithms and architectures for FFT chip design is presented in this paper. It is also recommended that mixed-radix/higher-radix algorithm combined with Single-path Delay Commutator (SDC) architecture is appropriate for massive MIMO in 5G, optical OFDM, cooperative MIMO and multi-user MIMO-based applications.

Keywords: FFT; IFFT; pipelined FFT; sequential FFT; OFDM; 5G.

J CIRCUIT SYST COMP Downloaded from www.worldscientific.com
by UNIVERSITY OF QUEENSLAND on 07/08/17. For personal use only.

1. Introduction

Orthogonal Frequency Division Multiplexing (OFDM) is quite a new spectrally efficient digital modulation scheme which employs multiple carriers that are mutually orthogonal to one another over a given time interval. Multiple Input Multiple Output-Orthogonal Frequency Division Multiplexing (MIMO-OFDM) has extensive applications in the field of wireless communications such as IEEE 802.11,[1] IEEE 802.15, IEEE 802.16,[2] IEEE 802.20, WiMax and 3GPP Long Term Evolution (LTE).[3]

*This paper was recommended by Regional Editor Emre Salman.
†Corresponding author.

Table 1. Various FFT sizes and their applications.

Applications	Standard	Supporting FFT/IFFT size
Wireless networks	IEEE 802.11a/g	64
	IEEE 802.11ac	64/128/256/512
	IEEE 802.11n	64/128
	IEEE 802.16e	128/256/512/1024/2048
	IEEE 802.15	512
Wired networks	ADSL	512
	VDSL	256/512/1024/2048/4096/8192
Terrestrial broadcasting	DAB	256/512/1024/2048
	DVB-T	2048/8192
	DVB-H	2048/4096/8192
	DMB-T/H	4096
	DVB-C2	4096
	DVB-T2	1024/2048/4096/8192

J CIRCUIT SYST COMP Downloaded from www.worldscientific.com
by UNIVERSITY OF QUEENSLAND on 07/08/17. For personal use only.

The crucial blocks in a MIMO-OFDM transceiver are Inverse Fast Fourier Transform (IFFT) and Fast Fourier Transform (FFT) units, which implement efficiently the highly complex computation for Inverse Discrete Fourier Transform (IDFT) and Discrete Fourier Transform (DFT) operations, respectively.[4,5] Major digital signal processing (DSP) applications depend on the size (N) of the FFT/IFFT (64–8192 points) computation. Table 1 provides various FFT sizes with their corresponding applications.[6–8]

The recent advancements in telecommunication technologies such as massive MIMO in 5G, optical OFDM, cooperative MIMO and multi-user MIMO require an efficient FFT/IFFT processing module to meet the necessary specifications of that particular application. A real-time implementation of high-speed FFT/IFFT module with less area that operates in minimal power consumption is very essential in designing an MIMO-OFDM integrated chip. The objective of this analysis is to select an area, power and delay efficient FFT/IFFT processor for MIMO-OFDM-based applications. A comparative study of efficient algorithms and architectures for FFT chip design is presented in this paper. The efficiency of FFT/IFFT processors can be evaluated using the measures like area utilization, power consumption and delay. Therefore, the technology process of integrated chip, type of FFT/IFFT architecture, the size N of FFT/IFFT, area utilization and power consumption are the various parameters considered for the analysis and discussion.

The organization of the paper is as follows. A brief introduction to FFT algorithms is given in Sec. 2. Memory references, as well as sequential FFT algorithms, are discussed in Sec. 3. Section 4 lists the different pipelined FFT architectures. Comparative analysis and discussions of various FFT/IFFT processors are given in Sec. 5 and concluding remarks are stated in Sec. 6.

2. Preliminaries

The DFT and IDFT of N-point data sequence can be given as

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-j2\pi kn/N}, \quad 0 \le k \le N-1, \tag{1}$$

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) e^{j2\pi kn/N}, \quad 0 \le n \le N-1. \tag{2}$$

The DFT and IDFT of any given sequence of length N are $x(n)$ and $X(k)$, respectively. From Eqs. (1) and (2), it is clear that n and k are nth and kth samples of N data points, where N can be 64/128/256/512/1024/2048/4096/8192 points which depends on the specific application. The exponential term given in Eqs. (1) and (2) represents the twiddle factor needed for FFT/IFFT computation. The direct implementation of DFT requires N^2 complex multiplications and $N(N-1)$ complex additions, where N is the number of data points. Divide and conquer technique, FFT, proposed by Cooley and Tukey, considering the periodicity and symmetric properties of DFT algorithm significantly reduced the number of complex multiplications to $(N/2)\log_2 N$ and a number of complex additions to $N\log_2 N$.[9] Table 2 provides the comparison of computational complexity for direct computation algorithm versus FFT algorithm.

By using divide and conquer approach, the number of data points N is partitioned into $r_1, r_2, r_3, \ldots$ and so on, where r is called the radix of FFT. The most common and familiar FFTs are with $r = 2$, FFT algorithm is referred to as a radix-2 FFT algorithm. However, further radices of range 2–10 are also used based on its application. In single-radix FFTs, the size of data points N must be the power of radix value. For example, with radix-4, the number of data points N in the FFT should be a power of 4.

Table 2. Computational complexity of direct computation versus FFT algorithm.

	Direct computation		FFT algorithm	
Number of data points N	Complex multiplications N^2	Complex additions $N(N-1)$	Complex multiplications $(N/2)\log_2 N$	Complex additions $N\log_2 N$
16	256	240	32	64
64	4096	4032	192	384
128	16384	16256	448	896
256	65536	65280	1024	2048
512	262144	261632	2304	4608
1024	1048576	1047552	5120	10240
2048	4194304	4192256	11264	22528
4096	16777216	16773120	24576	49152
8192	67108864	67100672	53248	106496

J CIRCUIT SYST COMP Downloaded from www.worldscientific.com
by UNIVERSITY OF QUEENSLAND on 07/08/17. For personal use only.

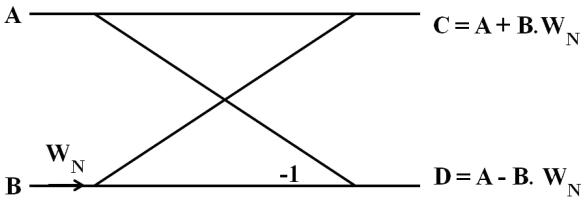

Fig. 1. Radix 2 DIT butterfly.

But, mixed-radix FFT can be done on composite sizes for decomposing a non-prime into prime factors. For example, an FFT of size 500 can be decomposed in five stages using radices of 2 and 5 since $500 = 2 \times 2 \times 5 \times 5 \times 5$ or 3 stages using radices of 5 and 10 since $500 = 5 \times 10 \times 10$. This basic r-point computation is called as butterfly unit.

The decomposition can be categorized as Decimation In Frequency (DIF) and Decimation In Time (DIT), depending upon the partition that takes place from input and output data points respectively. Figures 1 and 2 show the basic radix-2 butterfly diagrams for DIT-FFT and DIF-FFT computations, respectively. In Figs. 1 and 2, A and B denote the complex input from previous stage whereas C and D denote the complex output of the current stage (or complex input to the next stage). The twiddle factors W_N are defined as the co-efficients which are used to compute results from the previous stage and to form inputs to the next stages of FFT algorithm. Mathematically,

$$W_N = e^{-j2\pi/N}. \tag{3}$$

The implementation of FFT/IFFT processors widely uses in-place-memory updating and pipeline architectures. An in-place memory updating architecture, a single radix-r butterfly will be sufficiently re-used for N-point FFT/IFFT computation. Radix-r butterfly will be idle in the case of writing and reading input and output values. If the re-composition has different radix points, it is called mixed-radix FFT algorithm. When the radix is higher, the number of complex multiplications and additions are lesser, which in turn reduces the chip area and cost. Continuous flow mixed radix-based FFT architecture performs the computation with two N-sample memories for the continuous data stream.

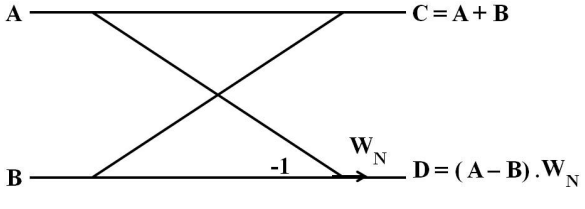

Fig. 2. Radix 2 DIF butterfly.

J CIRCUIT SYST COMP Downloaded from www.worldscientific.com
by UNIVERSITY OF QUEENSLAND on 07/08/17. For personal use only.

J CIRCUIT SYST COMP Downloaded from www.worldscientific.com
by UNIVERSITY OF QUEENSLAND on 07/08/17. For personal use only.

3. Sequential FFT Architectures

The direct implementation of the conventional Cooley–Tukey algorithm is sequential since the same radix butterfly unit is used in every stage. This radix-2 architecture requires $(N/2)\log_2 N$ complex multipliers and $N\log_2 N$ complex adders for N input data points as well as the twiddle factors for the computations are stored and accessed from a look-up table. Since FFT/IFFT operations are complex, the real and imaginary operations should be placed apart from each other. The hardware requirements for this architecture should be made up of adders, multipliers and individual memory for input/output bit reversals. Even though hardware requirements are low, latency is higher for the stream of input data points.[10,11] The sequential architecture for $N = 16$ is shown in Fig. 3. Since radix-2 butterfly computations are used, FFT for 16 data points are decomposed using four stages from stage 1 to stage 4. The inputs $x(0)$–$x(15)$ are fed into the algorithm in bit reversal (since bit-reversed indices are used to combine various stages in FFT) order whereas outputs $X(0)$–$X(15)$ are taken in natural order.

3.1. *Modified sequential architecture*

In this method, the identical twiddle factors required for the computations are grouped, so that the grouped twiddle factors are fed only once for the entire FFT computation. This reduces the latency as well as the memory occupied by the twiddle

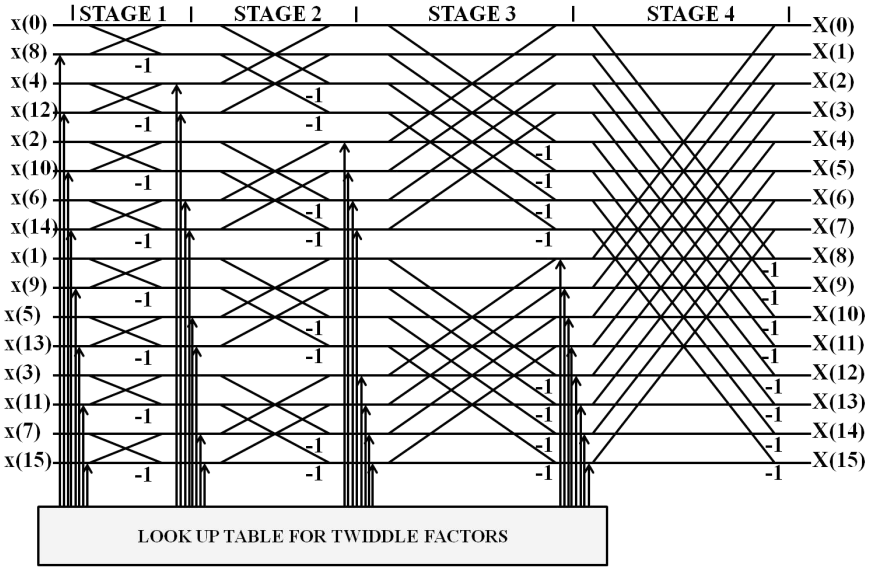

Fig. 3. Conventional DIT-FFT for $N = 16$.

factors. The identities accomplish the grouping of twiddle factors,

$$W_N^m = W_N^{N/4}.W_N^{m-N/4} = -jW_N^{m-N/4}, \quad N/4 \le m \le N/2 \tag{4}$$

$$= W_N^{N/2}.W_N^{m-N/2} = -W_N^{m-N/2}, \quad N/2 \le m \le 3N/4 \tag{5}$$

$$= W_N^{3N/4}.W_N^{m-3N/4} = jW_N^{m-3N/4}, \quad 3N/4 \le m \le N \tag{6}$$

$$= W_N^m, \quad 0 \le m \le N/4\,. \tag{7}$$

The twiddle factors are fed into the algorithm in such a way that the other computations are unaffected as well as the redundant memory modules are eliminated which is shown in Fig. 4.[12] Stage 1 is further decomposed into four substages to compute the results faster since identical twiddle factor computations are grouped. A 16-bit 64-point modified sequential algorithm along with cache memory based 1-dimensional (1D) FFT architecture[13] results in an area efficient, high-speed processor optimal for Wireless Local Area Network (WLAN)-based applications. Floating point butterfly unit using Fused-Dot-Product-Add (FDPA) based on Binary-Signed-Digit (BSD) representation is proposed to maximize the speed of the FFT computation.[14,15] Energy efficient computation units are also achieved through a stage skipping and merging to implement FFT/IFFT in Single Instruction Multiple Data (SIMD) and non-SIMD architectures.[16] The radix-2 butterfly is optimized to reduce one addition and one subtraction operation; optimized radix-2 butterfly is used in the radix-4 butterfly to reduce the number of real multipliers. Since radix-4 butterfly is also optimized, power utilization is reduced when compared with the conventional radix-4 butterflies.[17]

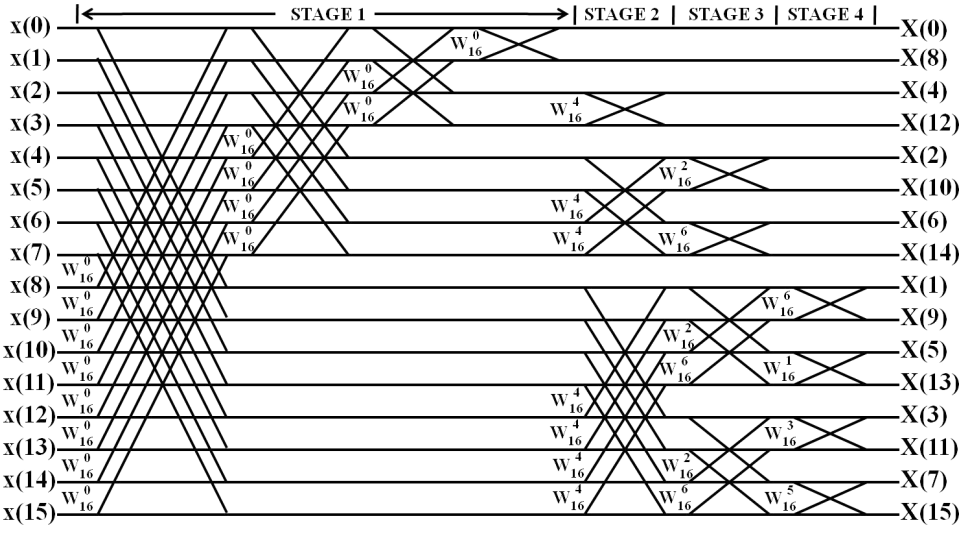

Fig. 4. Modified DIF FFT for $N = 16$.

J CIRCUIT SYST COMP Downloaded from www.worldscientific.com
by UNIVERSITY OF QUEENSLAND on 07/08/17. For personal use only.

3.2. *Real-valued FFT (RFFT) architecture*

The real-time bio-medical, sensor and radar-based communication processing generates redundant values in the output since the real-valued data signals exhibit conjugate symmetry. Better hardware utilization can be achieved by eliminating the redundancy using several techniques such as radix-2^5,[18] interleaved real and hybrid datapaths,[19] modified radix-2,[20] register based storage design[21] and radix-2^2 canonic based Real-valued FFT (RFFT).[22]

3.3. *Mixed radix FFT architecture*

Mixed Generalized High-Radix (GHR) FFT algorithm[23,24] uses 2D and 1D FFT factorization methods to reduce the hardware usage much less than the conventional FFT/IFFT methods. Processing speed and hardware efficiency are improved on the GHR method since it endures eight radices and 34 different FFT lengths for all LTE applications. Multiplier-less split-radix-based FFT architecture using Distributed Arithmetic (DA) for complex multiplications has been proposed.[25] Even though the number of overall arithmetic operations is reduced in this architecture, the complexity and number of operations in a butterfly are increased. Distributed Floating Point Arithmetic (DPFA) based variable length (32–2048) FFT architecture eliminates the power consuming complex twiddle factor multipliers by incorporating folded DA butterflies which reduce area usage as well as power consumption.[26] The FFT core is divided into three parts, namely dependant, in-dependant and reordering to minimize processing time in Multicore DSP implementation and is achieved with a reduction factor of 1/3.[27] A 60% resource utilization and 14% performance improvement are achieved by using the complex math processor in a low-cost radix-4/radix-2^2 butterfly-based FFT processor.[28]

A novel radix-2 split-radix FFT is proposed in which memory-sharing and clock gating are considered to avoid the inevitable switching action of multipliers which makes the circuit to operate faster. Look-up tables are used to generate the addresses for twiddle factors which eliminate the time taken for address generation.[29] An eight parallel data path for 512-point FFT computation with modified radix-25 butterfly is proposed to reduce the number of complex multipliers for twiddle factor multiplication for OFDM Wireless Personal Area Networks (WPAN) applications with a high throughput of 2.5 GS/s at 310 MHz.[30] Radix-2/4 butterfly based 64 to 128 point mixed FFT processor with Coarse Grain Reconfigurable Array (CGRA) embodied Reduced Instruction Set Computing (RISC) processor is proposed to eliminate partial execution timing constraints for IEEE 802.11 based FFT applications.[31] A parallel pipelined Real Fast Fourier Transform (RFFT) architecture in which four inputs are processed using modified processing elements that have two radix-2 butterfly units that remove redundant operations in the flow graph with conflict-free address generation scheme is proposed for real-valued signals.[32]

J CIRCUIT SYST COMP Downloaded from www.worldscientific.com
by UNIVERSITY OF QUEENSLAND on 07/08/17. For personal use only.

A scalable radix-2 based N-point FFT processor in which two radix-2 butterfly units are used for computation of input data at single clock pulse is proposed for multi-carrier systems like OFDM transceivers.[33] A Transport Triggered Architecture (TTA) and programmable Instruction Level Parallelism (ILP) based mixed radix-4/2/3 FFT processor that supports all the FFT sizes for LTE (128∼2048/1536) applications is exploited in Refs. 34 and 35. A mixed-radix comprises of radix-2, radix-2^2 and radix-2/4/8 that computes at 512–8192 points FFT operation, optimized by sub-structure sharing mechanism, results in an area efficient, low-complex processor for applications such as Digital Audio Broadcasting (DAB), Digital Video Broadcasting-Terrestrial (DVB-T) and Digital Video Broadcasting-Handheld (DVB-H).[36] By using hardware sharing schemes in radix-2^2 algorithms, hybrid architecture is developed that retains the systematic and variable characteristics of recursive DFT algorithms.[37] A 256 point mixed-radix-$2^3/2^5$ algorithm is proposed to reduce the complex multipliers by adopting 32 parallel data paths and eight sequential groups. The performance of this mixed-radix architecture is suitable for optical OFDM systems.[38] A CORDIC module is used to reduce the complexity of trigonometric computations (for twiddle factor multiplication) in the butterfly operations involved in mixed-radix FFT architecture.[39]

J CIRCUIT SYST COMP Downloaded from www.worldscientific.com
by UNIVERSITY OF QUEENSLAND on 07/08/17. For personal use only.

4. Pipelined FFT Architectures

A separate arithmetic unit can be used in each stage of the FFT unlike in the sequential architectures. This type of parallelism among each stage improves the processor performance in a significant way. A factor of $\log_2 N$ enhances the throughput of pipelined FFT architecture. Since each stage of this FFT algorithm has its basic computational units, the pipelined FFT can be called as cascaded FFT algorithm. Each stage does its operation as soon as the input data to be processed are available. These types of pipelined FFT processors are more suitable for real-time MIMO-OFDM as well as in-place signal processing applications. In pipeline architectures, Single-path Delay Feedback (SDF) and Multipath Delay Commutator (MDC) are the two popular architectures. SDF along with input scheduling enables multiple input streams processing with a single FFT/IFFT processor. MDC uses more switch boxes to resolve feedback path into feedforward paths. The MDC-based FFT architecture saves more area than SDF architecture-based FFT processor with multiple streams. Single radix-N butterflies are used at each folding stage of the MDC architecture-based FFT processor with N data streams. The pipelined FFT architectures are categorized into the following groups.

4.1. *Radix 2 multipath delay commutator* (*R2MDC*)

The most straightforward approach to implement radix-2 butterfly operation is the radix-2 Multipath delay commutator (R2MDC). The input data sequence is split

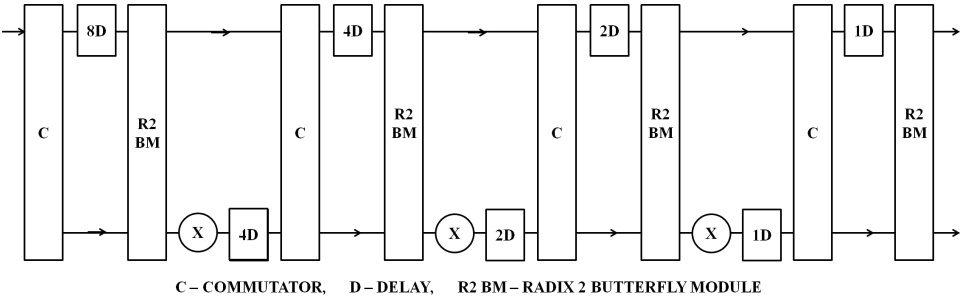

Fig. 5. Radix-2 multipath delay commutator for $N = 16$.

J CIRCUIT SYST COMP Downloaded from www.worldscientific.com
by UNIVERSITY OF QUEENSLAND on 07/08/17. For personal use only.

into two parallel data streams (for radix-2 butterfly) using two path commutator (C) and is fed into the data processing elements which are radix-2 Butterfly Modules (R2BM) at an equal distance by incorporating appropriate delay units (8D for 8 units of delay, 4D, 2D and 1D). The processing elements as well as the multiplier unit are used only half of the time and hence utilization rate of this pipelined FFT is 50%. A generalized R2MDC FFT architecture is shown in Fig. 5.

Mixed/multi-radix butterfly stages are also used with MDC based FFT/IFFT processors to reduce the memory usage or to share the memory between the butterfly operations. A modified radix-4 butterfly algorithm has been proposed that can perform a single radix-4 operation or two radix-2 operations. Eight parallel data paths are used before the modified radix-4 butterfly for higher throughput rate and low hardware complexity.[38] A simple counter is used to generate the mixed radix butterfly sequences and fixed/mixed radix memory addressing schemes are introduced in MDC-FFT architectures. In fixed radix memory addressing scheme, two memory words for real and imaginary elements, two counters for address and column counter are used. The number of counters can be reduced in fixed radix memory addressing scheme without sacrificing hardware utilization for Mixed Radix (MR) addressing operation.[40] Flexible Radix Configuration (FRC) based MDC-FFT architecture is proposed[42] to improve the throughput and power efficiency. A radix-2 DIF butterfly and pipeline data scheduling based 1024 point fixed (32-bit) FFT structure with ping-pong memory access scheme is proposed for high speed and real-time signal processing applications.[43]

4.2. *Radix 4 multipath delay commutator (R4MDC)*

The radix-4 MDC architecture is similar to radix-2 MDC in which Radix-4 Butterfly Module (R4BM) replaces radix-2 butterfly unit, and its corresponding 4-path commutator (C) is used. Since the input data sequence is separated into four parallel data streams, only 1/4 of input data is fed into the multipliers and computing elements; the utilization rate is reduced to 25%. This architecture is not

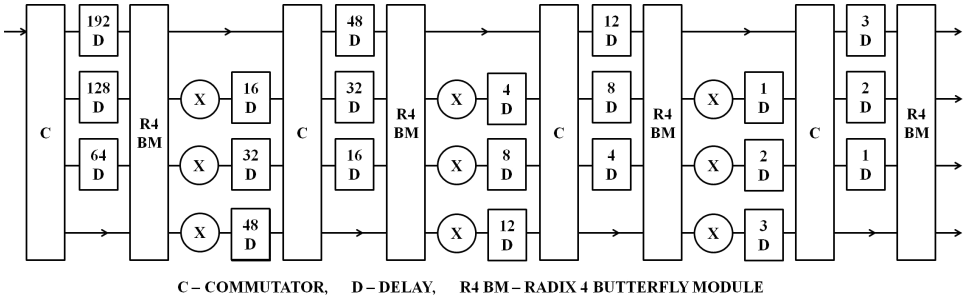

Fig. 6. Radix-4 multipath delay commutator for $N = 256$.

J CIRCUIT SYST COMP Downloaded from www.worldscientific.com
by UNIVERSITY OF QUEENSLAND on 07/08/17. For personal use only.

recommended when hardware requirement and utilization rate are taken into consideration. Radix-4 MDC architecture for 256 input data points is shown in Fig. 6.

Since larger data point is considered for computation delay, units are also larger that ranges upto 192 units of delay (192D for 192 units of delay). An input memory scheduling scheme is introduced with MDC to increase the utilization rate and obtain the variable length (128/512/1024/2048) MDC-FFT processor. Here, Radix N butterfly is used for N input data streams. One group of the memory bank is used for storing data from one input stream. $3N/4$ samples from one memory group are processed at a time in radix-4 butterfly unit. At the same instance, the elements of the other memory group are updated from the other input data streams. A configurable radix-8/radix-4 butterfly is used in the last stage with a constant multiplier for twiddle factor multiplications. A control/select input line is used for selecting the intermediate stages to the number of data points/streams.[44–46] Radix-2^k MDC architecture, for both DIF and DIT decompositions, can be used for any number of parallel input data streams. This architecture achieves high throughput with fewer hardware resources needed for recent applications.[46] The number of non-trivial multipliers is reduced by using a mixed-radix butterfly operation along with multipath delay feedback architecture for 1–4 channel 1024/2048/4096 FFT computation with less hardware complexity.[47] To lessen the complexity in twiddle factor multiplication, a radix-2^4–2^2–2^3 based MDF-FFT architecture is proposed which is optimal for IEEE 802.11ad (wireless personal area) applications[48] and higher radix butterfly units.[76] The conventional Multipath Delay Feedback (MDF) architecture inefficiently utilizes the adders and multipliers. A mixed-decimation MDF (M^2DF) architecture based on radix-2^k algorithm[49] and radix-4 algorithm[50] are proposed for better hardware utilization. A variable length radix-$2/3/2^2/2^3$ algorithm-based MDC architecture is proposed to reduce the number of operating cycles for FFT computation.[51] Many applications require simultaneous calculation of independent FFT samples; MDC architecture is configured for two data streams for achieving 100% hardware utilization.[52]

J CIRCUIT SYST COMP Downloaded from www.worldscientific.com
by UNIVERSITY OF QUEENSLAND on 07/08/17. For personal use only.

4.3. *Radix 2 single-path delay feedback (R2SDF)*

As the number of delay elements in MDC architecture is a major disadvantage, a feedback mechanism is provided in such a way that one-half of output data from each computing stage is fed back into the input of the same computing stage, as the data is directly given to the butterfly unit for computation. This technique was introduced in Ref. 53, and 100% memory utilization is achieved in Ref. 54. When the butterfly computation is radix-2, it is called Radix-2 Single-path delay feedback (R2SDF). Although the number of butterfly units and multipliers required is the same as that of Radix-2 MDC architecture, the memory (delay elements) and commutator requirement is very much reduced. A simple R2SDF architecture for $N = 16$ is shown in Fig. 7. An efficient FPGA implementation of generalized radix-2 SDF based FFT architecture by using coarse-grained hardware design is presented in Ref. 55. For OFDM baseband processing and better resource utilization, FFT/IFFT reconfiguration is achieved by FPGA-based Dynamic Partial Reconfiguration (DPR).[56] Radix-2[57] and radix-2^2[58] SDF-based pipelined FFT architectures are proposed for 16/64/256/1024 point with word-length scaling approach. A low power radix-2 64–1024 point SDF-based FFT architecture that shares hardware between 2 and 4 data streams is proposed.[59] The efficiency of the processor is doubled compared to the conventional SDF architectures.

4.4. *Radix 4 single-path delay feedback (R4SDF)*

Radix-4 single-path delay feedback (R2SDF) is similar to R2SDF in which radix-2 butterfly is replaced by radix-4 basic butterfly computation units. The number of multiplier units in radix-4 SDF architecture, as well as the utilization rate, is reduced to 25%, but hardware complexity is increased when compared with radix-2 SDF architecture. The generalized block diagram of radix-4 SDF architecture for 16 input data points is shown in Fig. 8. 1536 point FFT computation is achieved in variable length FFT processors[60,62] that suffers from very high latency and increased hardware costs. Three-stage SDF pipeline architecture and a hardware sharing mechanism are employed for FFT/IFFT processor to reduce the chip area.[61] Radix-2 SDF butterfly asymmetric with last three stages of radix-3 FFT butterfly is used for the

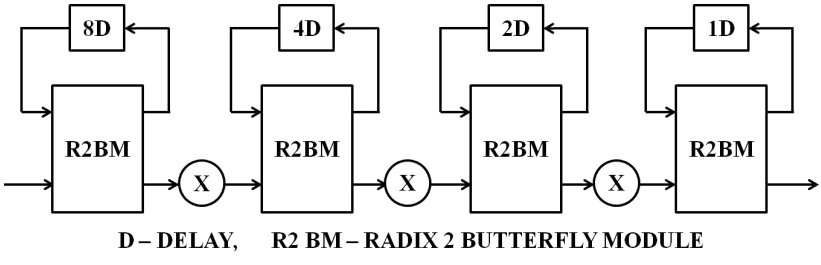

Fig. 7. Radix-2 single-path delay feedback for $N = 16$.

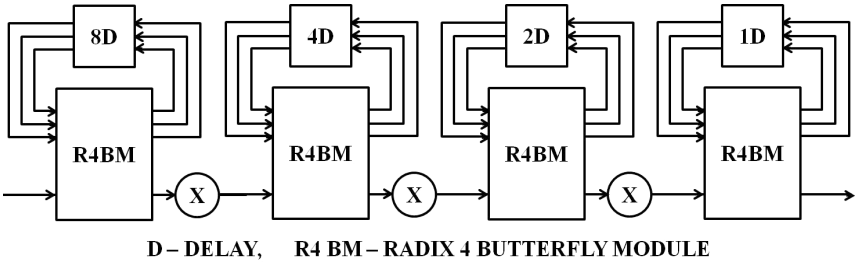

Fig. 8. Radix-4 single-path delay feedback for $N = 16$.

computation of radix-2 FFT. By using a simple multiplexer switch, the same architecture can be used to compute 128–2048/1536 FFT operations. Delay line buffers are further used to access the memory from idle processing elements to provide necessary buffering capacity for 1536 point FFT computation since it requires a larger buffer size.[62]

4.5. *Radix-4 single-path delay commutator (R4SDC)*

A simplified radix-4 butterfly is proposed in such a way that only one output is computed by using four input data from the input buffer at a time. This process is repeated in the same butterfly module (R4BM) until all the four same outputs are computed to achieve 100% utilization. Radix-4 single-path delay commutator (R4SDC) architecture needs additional computational units to provide the same input data four times. So a four path commutator (C) is used. The number of multipliers needed is much less than radix-4 MDC FFT architecture. A simplified R4SDC commutator architecture is shown in Fig. 9.

Multipath delay commutator and single path delay feedback are made parallel to achieve low latency, high throughput, and memory efficient 128–2048 point FFT/IFFT processor. An input selector or multiplexer switch is used to feed the input data to the appropriate stage.[63] A low-power 64-point pipeline FFT processor based on radix-4^3 butterflies, which computes four inputs in parallel, achieving 25% minimization in clock rate when compared with traditional MDC and SDF architectures, targeting IEEE 802.11a/g applications, is proposed.[64] R4SDC is

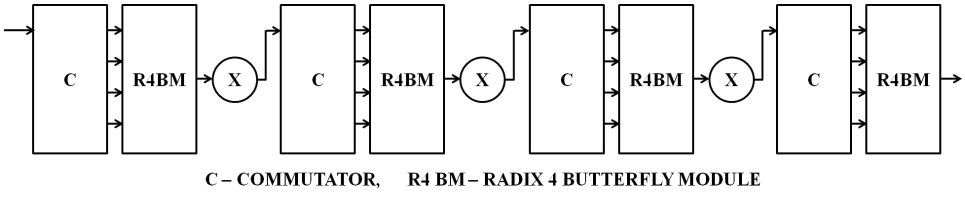

Fig. 9. Radix-4 single-path delay commutator for $N = 256$.

J CIRCUIT SYST COMP Downloaded from www.worldscientific.com
by UNIVERSITY OF QUEENSLAND on 07/08/17. For personal use only.

appropriate for MIMO-OFDM based applications for its low complexity and high hardware utilization.[65]

4.6. *Radix 2^2 single-path delay feedback ($R2^2SDF$)*

Radix-2^2 single-path delay feedback is an efficient evolving pipeline architecture for FFT computation. This architecture utilizes two radix-2 butterfly units with coefficient multipliers which are suitable for typical pipeline FFT implementation. The number of multipliers and memory requirement is very much reduced when compared with the previous designs. Radix-2^2 single-path delay feedback structure for $N = 256$ is shown in Fig. 10.

A mixed DIT/DIF FFT algorithm is used along with a modified SDF FFT architecture to obtain high utilization with same throughput and latency. In FFT computation, the final stage is computed by DIT, whereas, other stages are computed by DIF algorithm so as to maintain input and output in normal order that eliminates the usage of an additional clock. The modified SDF architecture computes radix-4 as radix-2^2, radix-8 as radix-2^3 and so on to reduce the number of complex adders and complex multipliers.[66] A mixed-radix variable length FFT processor is proposed based on Radix-$2/2^2$ butterfly structure by incorporating pipeline SDF method to minimize area with increased efficiency for floating point day-to-day FFT/IFFT applications.[67] A radix-2 algorithm based combined SDC-SDF architecture is presented with $\log_2(N-1)$ SDC stages, one SDF stage, and one-bit reverser to reduce up to 50% of complex multipliers.[68] The power consumption of

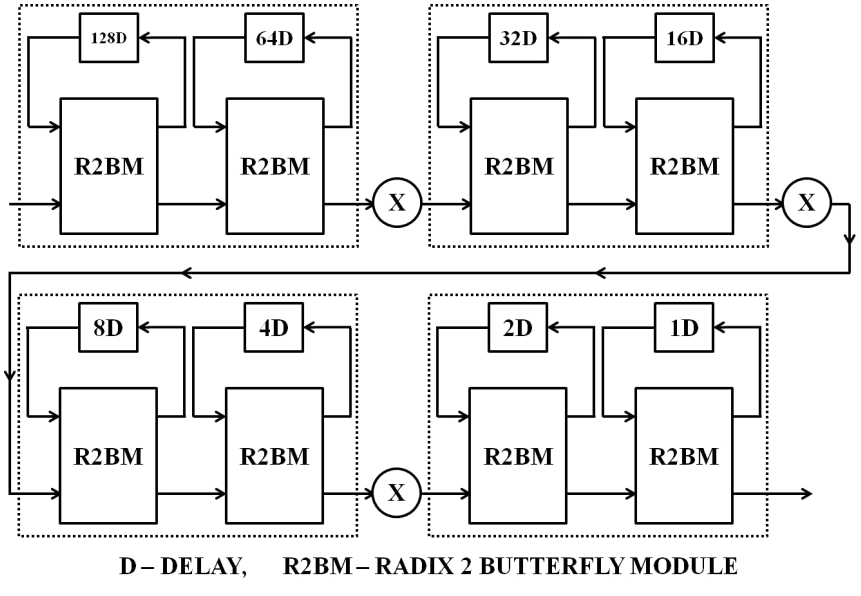

Fig. 10. Radix-2^2 single-path delay commutator for $N = 256$.

J CIRCUIT SYST COMP Downloaded from www.worldscientific.com
by UNIVERSITY OF QUEENSLAND on 07/08/17. For personal use only.

Table 3. Hardware requirements and throughput rate for various pipeline architectures.

Pipeline architecture	No. of complex adders	No. of complex multipliers	Memory requirement	Throughput rate
R2MDC	$2\log_2 N$	$\log_2(N-2)$	$(N-2)/5$	2R
R4MDC	$8\log_4 N$	$3\log_4(N-1)$	$5(N-4)/2$	4R
R2SDF	$2\log_2 N$	$\log_2(N-2)$	$N-1$	R
R4SDF	$8\log_4 N$	$\log_4(N-1)$	$N-1$	R
R4SDC	$3\log_4 N$	$\log_4(N-1)$	$N-1$	R
R2^2SDF	$4\log_4 N$	$\log_4(N-1)$	$N-1$	R

radix-4 SDF architecture is 11% lesser when compared to the radix-2 SDF architecture.[69]

The hardware comparison of pipeline architectures discussed above is shown in Table 3.[70]

5. Analysis and Discussions

The objective of this analysis is to select an area, power and delay efficient FFT/IFFT processor for MIMO-OFDM-based applications. Normalized area, energy, and throughput are the significant measures to evaluate the performances of various FFT/IFFT processors.[71] Several methods have been proposed to assess the normalized area and energy for FFT/IFFT processors,[71–74] but these evaluations were carried out for the same number of FFT/IFFT data points (N).

Since different FFT/IFFT sizes (N) were considered for our comparison, the evaluation proposed in Ref. 75 is optimal that can be given as,

$$A_{\text{Normalized}} = \frac{\text{Area}}{N \times \left(\frac{\text{Tech}}{0.18}\right)^2} \tag{8}$$

$$E_{\text{Normalized}} = \frac{\text{Power} \times \text{Exec.Time}}{N \times \left(\frac{V_{DD}}{1.8}\right)^2} \tag{9}$$

Table 4 compares the frequency, normalized area and normalized energy of the different FFT/IFFT architectures discussed in the literature. As previously stated, the processors have different fabrication CMOS library and various synthesis constraints, hence the FFT/IFFT processors with same values of N are considered for the discussion.

For FFT/IFFT processors with $N = 2048$, the MDF architecture in Ref. 75 has only 42.53% and 33.42% of the normalized area to that of SDF-based architectures in Refs. 76 and 77. The normalized area of MDF scheme in Ref. 73 is only 15.53% to that of MDC-based FFT/IFFT architecture in Ref. 44. It can be noted that the MDF architecture proposed in Ref. 75 saves more area, but execution time is almost

J CIRCUIT SYST COMP Downloaded from www.worldscientific.com
by UNIVERSITY OF QUEENSLAND on 07/08/17. For personal use only.

Table 4. Comparision among various FFT processors.

FFT processor	Architecture	Clock frequency (MHz)	Process (nm)	FFT size	Normalized energy (nJ)	Normalized area (mm^2)
Raja[13]	Cached FFT	251	180	64	53.85	9.98
Chen[23]	GHR	122.88	180	128–2048	4.12	2.44
Cho[30]	MR-Pipelined	310	90	512	1.31	6.093
Lai[37]	Recursive Radix-2^2	25	180	512	6.875	8.50
Kim[38]	MR-MDC	430	90	128–256	—	—
Lee[41]	MR-MDC	20	180	8192	5.4	0.33
Tang[42]	FRC-MDC	300	180	512	3.26	6.25
				256	6.00	
				128	8.85	
Yang[44]	MDC	40	90	2048	5.16	6.05
				1024	5.09	
				512	4.65	
				128	4.16	
Wang[48]	Radix-2^4–2^2–2^3 MDF	220	130	512	—	3.00
Yu[62]	SDF	40	90	2048	0.313	2.813
				1536	0.328	
				1024	0.535	
				512	0.602	
				256	1.03	
				128	1.83	
Kala[64]	R4-SDC	5	130	64	2.11	24.8
Sophy[67]	MR-SDF	50/100	90	2048	3.05	39.84
				512	11.3	
				128	592	
Chen[74]	Cached FFT	51	180	128–1024	2.06	2.05
Peng[75]	MDF	35	180	128–2048/1536	1.28	0.94
Patil[76]	SDF	40	180	128–2048	1.39	2.21
Huang[77]	Memory base	324	90	512	14.41	19.21
Tang[78]	Multi-data scaling	300	90	2048	0.83	2.265
		52		128	0.568	
Huang[79]	Cascaded	324	90	512	0.820	7.265
Ahamed[80]	Feed-forward	330	65	512	1.84	21.41

J CIRCUIT SYST COMP Downloaded from www.worldscientific.com
by UNIVERSITY OF QUEENSLAND on 07/08/17. For personal use only.

4.5 times higher than other MDC[44] and SDF[76]-based FFT/IFFT schemes. Since delay elements are more in 2048 MDC/MDF-based FFT/IFFT processors, the normalized energy dissipation is less in SDF scheme which is of 4–16 times lesser when compared to the other MDC,[44] MDF[75] and SDF[76] schemes. This variation is mainly due to the usage of Radix-3 FFT in SDF pipeline stages which reduces the number of computations. On the other hand, power consumption is greatly reduced. This trend in the area, power and delay can be carried on to the FFT/IFFT processors with $N = 1024, 512, 256$ and 128.

When considering the FFT/IFFT processors with size $N = 1024$, the normalized energy consumption by the SDF[62] processor is approximately 4 times higher than that of the Cached FFT processor,[73] and is 10.5%–41.7% to that of MDC,[44]

GHR[62] and MDF[75] schemes. For the processors with FFT/IFFT size $N = 512$, the memory-based FFT architecture[77] consumes more power and occupies larger area as the number of memory elements is higher when compared with the pipelined-FFT[42,44,62,74,75] as well as mixed-radix pipelined FFT[30,67] As the clock frequency is higher, the normalized area and normalized energy also increase in feed-forward FFT architecture[81] when compared with other pipelined FFT/IFFT schemes. The normalized energy increases linearly with the normalized area as the number of logic and delay elements increase in line with the size of FFT/IFFT computation. The normalized area of mixed radix pipelined FFT[30] scheme is 15.30% of mixed radix SDF FFT scheme,[67] 28.46% of feed-forward architecture[80] and 31.71% of memory-based schemes.[77]

To reduce the delay, area and power effectively in an FFT/IFFT processor, the foremost solution is to reduce the memory usage. This can be achieved not only by selecting the proper FFT/IFFT architecture but also by adopting efficient input–output memory scheduling schemes and bit-reversal circuits. These techniques not only reduce the core size of the processor, they also reduce the power consumption and execution time of an FFT/IFFT scheme by reducing the number of accesses to the processor and main memory. Further, the reliability of core FFT processor can be improved by adopting error correction codes and Parseval checks.[82] Several strategies in implementing a reliable sequential/pipelined FFT core are discussed in Refs. 83–85.

J CIRCUIT SYST COMP Downloaded from www.worldscientific.com
by UNIVERSITY OF QUEENSLAND on 07/08/17. For personal use only.

6. Conclusion

The significance of various algorithms, architectures, and implementations of FFT/IFFT processor design for MIMO-OFDM based communication systems has been focused on in this review article. The aim of this article is to select an area efficient, low power, high-speed FFT/IFFT processor by adopting an architecture that has a desired twiddle factor multiplier circuit, an effective design of butterfly processing modules with appropriate feedback/feedforward delay units and input–output data scheduling schemes that have lesser access to memory. Higher-radix, as well as mixed-radix algorithms, based SDC pipelined FFT/IFFT architectures have less area and consume low-power at a higher throughput. It can be concluded that mixed-radix/higher-radix algorithm combined with Single-path Delay Commutator (SDC) architecture is appropriate for massive MIMO in 5G, optical OFDM, cooperative MIMO and multi-user MIMO based applications.

References

1. IEEE, IEEE Standard 802.11: The Wireless LAN Media Access Control (MAC) and Physical Layer (PHY) Specifications, IEEE Computer Society, Washington DC (1999).
2. IEEE, IEEE Standard 802.16: IEEE Standard for Local and Metropolitan Area Networks. Part16: Air interface for fixed broadband wireless access systems, IEEE Computer Society, Washington DC (2004).

J CIRCUIT SYST COMP Downloaded from www.worldscientific.com
by UNIVERSITY OF QUEENSLAND on 07/08/17. For personal use only.

3. IEEE, IEEE Standard 802.16: IEEE Standard for Local and Metropolitan Area Networks. Part16: Air interface for fixed broadband wireless access systems, IEEE Computer Society, Washington DC (2006).
4. Y. G. Li, J. H. Winters and N. R. Sollenberger, MIMO-OFDM for wireless communications: Signal detection with enhanced channel estimation, *IEEE Trans. Commun.* **50** (2002) 1471–1477.
5. A. Goldsmith, *Wireless Communications* (Cambridge University Press, 2005).
6. Asymmetric Digital Subscriber Line Transceivers 2 (ADSL2), *ITU-T Standard G.992.3*, 2005.
7. Very-High-Bit-Rate Digital Subscriber Line Transceiver 2(VDSL2), *ITUT Standard G.993.2*, Feb. 2006.
8. Q. Lu, X. Wang and J. C. Niu, A low-power variable-length FFT processor base on radix-24 algorithm, *Proc. IEEE Conf.* 2009, pp. 129–132.
9. J. G. Proakis and D. G. Manolakis, *Digital Signal Processing Principles, Algorithms and Applications*, 4th edn. (Pearson, 2007).
10. S. Josue Saenz, J. J. Raygoza, P. E. C. Becerra, A. S. O. Cisneros and J. R. Dominguez, FPGA design and implementation of radix-2 Fast Fourier Transform algorithm with 16 and 32 points, *Proc. IEEE International Autumn Meeting on Power, Electronics and Computing (ROPEC)* (Ixtapa, 2015), pp. 1–6.
11. A. Tiwari, S. Phapal, D. Kadam, S. Sivanantham and K. Sivasankaran, FPGA implementation of FFT blocks for OFDM, *Proc. Online International Conf. Green Engineering and Technologies (IC-GET)* (Coimbatore, 2015), pp. 1–4.
12. J. Raja and M. Kannan, VLSI implementation of high throughput MIMO OFDM transceiver for 4th generation systems, *Indian J. Eng. Materials Sci.* **19** (2012) 307–319.
13. J. Raja, P. Mangaiyarkarasi and K. Moorthi, Area efficient low power high performance cached FFT processor for MIMO OFDM application, *Int. J. Appl. Eng. Res.* **10** (2015) 11853–11868.
14. A. Kaivani and S. Ko, Floating-point butterfly architecture based on binary signed-digit representation, *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **24** (2016) 1208–1211.
15. A. Kaivani and S. B. Ko, Area efficient floating-point FFT butterfly architectures based on multi-operand adders, *Electron. Lett.* **51** (2015) 895–897.
16. N. Sharma, P. R. Panda and F. Catthoor, Energy efficient FFT implementation through stage skipping and merging, *Proc. Int. Conf. Hardware/Software Codesign and System Synthesis (CODES + ISSS)* (Amsterdam, 2015) pp. 153–162.
17. R. Neuenfeld, M. Fonseca and E. Costa, Design of optimized radix-2 and radix-4 butterflies from FFT with decimation in time, *IEEE 7th Latin American Symp. Circuits & Systems (LASCAS)* (Florianopolis, 2016), pp. 171–174.
18. A. A. Naoghare and A. V. Sakhare, Review on FFT architecture for real valued signals using Radix 2^5 algorithm, *Proc. Int. Conf. Pervasive Computing (ICPC)* (Pune, 2015), pp. 1–3.
19. A. Chinnapalanichamy and K. K. Parhi, Serial and interleaved architectures for computing real FFT, *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Process. (ICASSP)*, South Brisbane, QLD, 2015, pp. 1066–1070.
20. A. R. Kuralkar, Design of fast fourier transform using processing element for real valued signal, *Proc. Int. Conf. Commun. Signal Process. (ICCSP)* (Melmaruvathur, 2015), pp. 1446–1448.
21. P. K. Meher, B. K. Mohanty, S. K. Patel, S. Ganguly and T. Srikanthan, Efficient VLSI architecture for decimation-in-time fast fourier transform of real-valued data, *IEEE Trans. Circuits Syst. I Regular Papers* **62** (2015) 2836–2845.

22. Y. Lao and K. K. Parhi, Canonic real-valued radix-2n FFT computations, *Proc. 49th Asilomar. Conf. Signals, Systems and Computers* (Pacific Grove, CA, 2015), pp. 441–446.
23. J. Chen, J. Hu, S. Lee and G. E. Sobelman, Hardware efficient mixed radix 25/16/9 FFT for LTE systems, *IEEE Trans. VLSI Syst.* **23** (2015) 221–229.
24. C.-F. Hsiao, Y. Chen and C.-Y. Lee, A generalized mixed-radix algorithm for memory-based FFT processors, *IEEE Trans. Circuits Syst. II Exp. Briefs* **57** (2010) 26–30.
25. N. Laguri and K. Anusudha, VLSI implementation of efficient split radix FFT based on distributed arithmetic, *Proc. Int. Conf. Green Comput. Electrical Eng.* (2014), pp. 1–5.
26. A. S. B. Paul, S. Raju and R. Janakiraman, Low power reconfigurable FP-FFT core with an array of folded DA butterflies, *Eurasip J. Adv. Signal Process.* **144** (2014) 1–17.
27. A. Kharin, S. Vityazev, V. Vityazev and N. Dahnoun, Parallel FFT implementation on Tms320c66x multicore DSP, *2014 6th European Embedded Design in Education and Research Conference (EDERC)*, Milano 1–5 (2014), pp. 46–49, doi: 10.1109/ICDC-Syst.2014.6926131.
28. S. Ranganathan, R. Krishnan and H. S. Sriharsha, Efficient hardware implementation of scalable FFT using configurable radix-4/2, *Proc. 2nd Int. Conf. Devices, Circuits and Systems* (2014), pp. 1–5, doi: 10.1109/ICDCSyst.2014.6926131.
29. Z. Qian, N. Nasiri, O. Segal and M. Margala, FPGA implementation of low-power split-radix FFT processors, *Proc. 24th Int. Conf. Field Programmable Logic and Applications* (2014), pp. 1–2.
30. T. Cho and H. Lee, A high-speed low-complexity modified radix-2^5 FFT processor for high rate WPAN applications, *IEEE Trans. Very Large Scale Integr.* **21** (2013) 187–191.
31. W. Hussain, X. Chen, G. Ascheid and J. Nurmi, A reconfigurable application-specific instruction-set processor for fast fourier transform processing, *Proc. IEEE 24th Int. Conf. Application–Specific Systems, Architecture and Processors* (2013), pp. 339–345.
32. M. Ayinala, Y. Lao and K. K. Parhi, An in-place FFT architecture for real-valued signals, *IEEE Trans. Circuits Syst. II* **60** (2013) 652–656.
33. D. Revanna, O. Anjum, M. Cucchi, R. Airoldi and J. Nurmi, A scalable FFT processor architecture for OFDM based communication systems, *Int. Conf. Embedded Computer Systems: Architectures, Modeling, and Simulation* (Agios Konstantinos, 2013), pp. 19–27, doi: 10.1109/SAMOS.2013.6621101
34. T. Patyk, D. Guevorkian, T. Pitkanen, P. Jaaskelainen and J. Takala, Low-power application specific FFT processor for LTE applications, *Proc. Int. Conf. Embedded Computing Systems: Architecture, Modelling and Simulation* (Agios Konstantinos, 2013), pp. 28–32, doi: 10.1109/SAMOS.2013.6621102.
35. H. Corporal, *Microprocessor Architectures: From VLIW to TTA* (John Wiley & Sons, Chichester, UK, 1997).
36. S. S. Wang and C. S. Li, An area-efficient design of variable-length fast fourier transform processor, *J. Signal Process. Syst.* **51** (2007) 245–256.
37. S. C. Lai *et al.*, Hybrid architecture design for calculating variable-length fourier transform, *IEEE Trans. Circuits Syst. II: Exp. Briefs* **63** 279–283.
38. H. S. Kang, S. H. Chang, I. K. Hwang and J. K. Lee, A design and implementation of 32-paths parallel 256-point FFT/IFFT for optical OFDM systems, *Proc. 18th Int. Conf. Adv. Commun. Technol. (ICACT)* (Pyeongchang, 2016), pp. 417–421.
39. N. Sarode, R. Atluri and P. K. Dakhole, Mixed-radix and CORDIC algorithm for implementation of FFT, *Proc. Int. Conf. Commun. Signal Process. (ICCSP)* (Melmaruvathur, 2015), pp. 1628–1634.
40. E. J. Kim and M. H. Sunwoo, High speed eight-parallel mixed-radix FFT processor for OFDM systems, *Proc. IEEE Int. Symp. Circuits and Systems* (2011), pp. 1684–1687.

J CIRCUIT SYST COMP Downloaded from www.worldscientific.com
by UNIVERSITY OF QUEENSLAND on 07/08/17. For personal use only.

J CIRCUIT SYST COMP Downloaded from www.worldscientific.com
by UNIVERSITY OF QUEENSLAND on 07/08/17. For personal use only.

41. S. Y. Lee, C. C. Chen, C. C. Lee and C. J. Cheng, A low-power VLSI architecture for a shared-memory FFT processor with a mixed-radix algorithm and a simple memory control scheme, *Proc. IEEE Int. Symp. Circuits and Systems* (Island of Kos, 2006), pp. 157–160.
42. 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* **47** (2012) 1419–1435.
43. S. Zhou, X. Wang, J. Ji and Y. Wang, Design and implementation of a 1024-point high-speed FFT processor based on the FPGA, *Proc. 6th Int. Cong. Image and Signal Processing* (2013), pp. 1112–1116.
44. K. J. Yang, S. H. Tsai and G. C. H. Chuang, MDC FFT/IFFT processor with variable length for MIMO-OFDM systems, *IEEE Trans. VLSI Syst.* **21** (2013) 720–731.
45. A. Amjadha, E. Konguvel and J. Raja, Design of multipath delay commutator architecture based FFT processor for 4th generation systems, *Int. J. Comput. Appl.* **89** (2014) 23–28.
46. M. Garrido, J. Grajal, M. A. Sanchez and O. Gustafsson, Pipelined radix-2^k feedforward FFT architectures, *IEEE Trans. VLSI Syst.* **21** (2013) 23–32.
47. Y. W. Lin and C. Y. Lee, Design of an FFT/IFFT processor for MIMO-OFDM systems, *IEEE Trans. Circuits and Syst. I* **54** (2007) 807–815.
48. C. Wang, Y. Yan and X. Fu, A high-throughput low-complexity radix-2^4-2^2-2^3 FFT/IFFT processor with parallel and normal input/output order for IEEE 802.11ad systems, *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **23** (2015) 2728–2732.
49. J. Wang, C. Xiong, K. Zhang and J. Wei, A mixed-decimation MDF architecture for radix-2^k parallel FFT, *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **24** (2016) 67–78.
50. L. P. Thakare and A. Y. Deshmukh, Area efficient FFT/IFFT processor design for MIMO OFDM system in wireless communication, *Proc. 7th Int. Conf. Emerging Trends in Engineering & Technology (ICETET)* (Kobe, 2015), pp. 10–13.
51. H. F. Luo, M. D. Shieh and K. H. Lee, A radix-$2/3/2^2/2^3$ MDC architecture for variable-length FFT processors, *Proc. Int. Conf. Consumer Electronics - Taiwan (ICCE-TW)* (Taipei, 2015), pp. 180–181.
52. A. X. Glittas, M. Sellathurai and G. Lakshminarayanan, A normal I/O order radix-2 FFT architecture to process twin data streams for MIMO, *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **24** (2016) 2402–2406.
53. S. He and M. Torkelson, A new approach to pipeline FFT processor, *Proc. IEEE Parallel Processing Symp.* (1996), pp. 766–770.
54. S. Sukhsawas and K. Benkrid, A high-level implementation of a high performance pipeline FFT on Virtex-E FPGAs, *Proc. IEEE Computer Society Annual Symp. VLSI (ISVLSI '04)* (2004), pp. 229–232.
55. I. Ali Qureshi, F. Qureshi and G. M. Shaikh, Efficient FPGA-mapping of 1024 point FFT pipeline SDF processor, *Proc. Sixth Int. Symp. Parallel Architectures, Algorithms and Programming* (2014), pp. 29–34.
56. M. L. Ferreira, A. Barahimi and J. C. Ferreira, Dynamically reconfigurable FFT processor for flexible OFDM baseband processing, *Proc. Int. Conf. Design and Technology of Integrated Systems in Nanoscale Era (DTIS)* (Istanbul, Turkey, 2016), pp. 1–6.
57. Y. Xie, Y. K. Feng, C. Yang, Y. Z. Xie and H. Chen, Design of a large point FFT processor with configurable transform length, *Proc. IET Int. Radar Conf.* (Hangzhou, 2015), pp. 1–5.

58. C. Yang, Y. Z. Xie, L. Chen, H. Chen and Y. Deng, Design of a configurable fixed-point FFT processor, *Proc. IET Int. Radar Conf.* (Hangzhou, 2015), pp. 1–4.
59. M. Dali, R. M. Gibson, A. Amira, A. Guessoum and N. Ramzan, An efficient MIMO-OFDM radix-2 single-path delay feedback FFT implementation on FPGA, *Proc. Conf. Adaptive Hardware and Systems (AHS)* (Montreal, QC, 2015), pp. 1–7.
60. S. Y. Peng, K. T. Shr, C. M. Chen and Y. H. Huang, Energy-efficient 128/2048/1536-point FFT processor with resource block mapping for 3 GPP-LTE system, *Proc. Int. Conf. Green Circuits Syst.* (2010), pp. 14–17.
61. C. H. Yang, T. H. Yu, and D. Markovic, Power and area minimization of reconfigurable FFT processors: A 3 GPP-LTE example, *IEEE J. Solid-State Circuits* **47** (2012) 757–768.
62. C. Yu and M. H. Yen, Area-efficient 128- to 2048/1536-point pipeline FFT processor for LTE and mobile WiMAX systems, *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **23** (2015) 1793–1800.
63. T. Adiono and R. Mareta, Low latency parallel-pipelined configurable FFT-IFFT 128/256/512/1024/2048 for LTE, *Proc. IEEE 4th Int. Conf. Intelligent and Advanced Systems* (Kuala Lumpur, 2012), pp. 768–773.
64. S. Kala, S. Nalesh, S. K. Nandy and R. Narayan, Design of a low power 64 point FFT architecture for WLAN applications, *Proc. 25th Int. Conf. Microelectronics* (2013), pp. 1–4.
65. S. Singh and J. Kedia, Pipelined FFT architectures: A review, *Proc. Int. Conf. Electrical, Electronics, Signals, Communication and Optimization (EESCO)* (Visakhapatnam, 2015), pp. 1–5.
66. S. Lee and S. C. Park, Modified SDF architecture for mixed DIF/DIT FFT, *Proc. IEEE Int. Symp. Circuits and Syst.* (New Orleans LA, 2007), pp. 2590–2593.
67. A. Sophy, R. Srinivasan, J. Raja and S. Anand Ganesh, Variable length floating point FFT processor using radix-2^2 butterfly elements, *Int. J. Eng. Technol.* **6** (2014) 764–771.
68. 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.* **23** (2015) 973–977.
69. P. A. Sophy, R. Srinivasan, J. Raja and M. Avinash, Analysis and design of low power radix-4 FFT processor using pipelined architecture, *Proc. Int. Conf. Comput. Commun. Technol. (ICCCT)* (Chennai, 2015), pp. 227–232.
70. B. Fu and P. Ampadu, An area efficient FFT/IFFT processor for MIMO-OFDM WLAN 802.11n, *J. Signal Process. Syst.* **56** (2008) 59–68.
71. B. M. Baas, A low-power, high-performance, 1024-point FFT processor, *IEEE J. Solid-State Circuits* **34** (1999) 380–387.
72. Y. T. Lin, P. Y. Tsai and T. D. Chiueh, Low-power variable-length fast fourier transform processor, *IEEE Proc. Comput. Digit. Tech.* **152** (2005) 499–506.
73. T. D. Chiueh and P. Y. Tsai, *OFDM Baseband Receiver Design for Wireless Communications* (Wiley, New York, 2007).
74. C. M. Chen, C. C. Hung and Y. H. Huang, An energy-efficient partial FFT processor for the OFDMA communication system, *IEEE Trans. Circuits Syst. II Exp. Briefs* **57** (2010) 136–140.
75. S. Y. Peng, K. T. Shr, C. M. Chen and Y. H. Huang, Energy-efficient 128/2048/1536-point FFT processor with resource block mapping for 3GPP-LTE system, *Proc. Int. Conf. Green Circuits Syst.* (2010), pp. 14–17.
76. M. S. Patil, T. D. Chhatbar and A. D. Darji, An area efficient and low power implementation of 2048 point FFT/IFFT processor for mobile WiMAX, *Proc. Int. Conf. Signal Process. Commun.* (2010), pp. 1–4.

J CIRCUIT SYST COMP Downloaded from www.worldscientific.com
by UNIVERSITY OF QUEENSLAND on 07/08/17. For personal use only.

77. S. J. Huang and S. G. Chen, A green FFT processor with 2.5-GS/s for IEEE 802.15.3c (WPANs), *Proc. Int. Conf. Green Circuits Syst.* (2010), pp. 9–13.
78. S. N. Tang, J. W. Tsai and T. Y. Chang, A 2.4-GS/s FFT processor for OFDM-based WPAN application, *IEEE Trans. Circuits Syst.-II: Exp. Briefs* **57** (2010) 451–455.
79. S. J. Huang and S. G. Chen, A high-throughput radix-16 FFT processor with parallel and normal input/output ordering for IEEE 802.15. 3c systems, *IEEE Trans. Circuits Syst. I: Regular Papers* **59** (2012) 1752–1765.
80. T. Ahmed, M. Garrido and O. Gustafsson, A 512-point 8 parallel pipelined feedforward FFT for WPAN, *Proc. 45th Conf. Signals, Systems and Computers (ASILOMAR)* (2011), pp. 981–984.
81. Z. Gao *et al.*, Fault tolerant parallel FFTs using error correction codes and parseval checks, *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **24** (2016) 769–773.
82. P. Y. Tsai and C. Y. Lin, A generalized conflict-free memory addressing scheme for continuous-flow parallel-processing FFT processors with rescheduling, *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **19** (2011) 2290–2302.
83. M. Kannan, E. Konguvel, J. Madhumitha, M. Mohamed Kamil Ashiq and K. Abhishek, FPGA implementation of FFT architecture using modified Radix-4 algorithm, *Asian J. Res. Soc. Sci. Humanit.* **7** (2017) 47–57.
84. M. Kannan and S. K. Srivatsa, Low power hardware implementation of high speed FFT core, *J. Comput. Sci.* **3** (2007) 376–382.
85. M. Kannan and S. Srivatsa, Hardware implementation low power high speed FFT core, *Int. Arab J. Inf. Technol.* **6** (2009) 1–6.

J CIRCUIT SYST COMP Downloaded from www.worldscientific.com
by UNIVERSITY OF QUEENSLAND on 07/08/17. For personal use only.