***Article Received:** 20 March 2023 **Revised:** 17 April 2023 **Accepted:** 20 May 2023*

Design of High Speed Memory-Based FFT Processor Using 90nm Technology

[1]T. Prasada Babu, [2]Dr.Rahul Mishra
[1]Research Scholar, Department of Electronics & Communication Engineering, Dr. A.P.J. Abdul Kalam University, Indore, M.P, India.
[2]Research Guide, Department of Electronics & Communication Engineering, Dr. A.P.J. Abdul Kalam University, Indore, M.P, India.
Corresponding Author Email : tadikondaprasadbabu@gmail.com

ABSTRACT: In order to enhance performance, the Fast Fourier Transformation is a important operation in Digital Signal Processing (DSP) systems had been extensively studied. State-of-the-art transmission technology uses Orthogonal frequency division multiplexing (OFDM), which primary operation is the Fast fourier transform (FFT). This analysis presents the design of a high-speed memory-based FFT processor using 90nm technology. The novel hybrid multiplier and hybrid adder is used in this analysis. The main objective of this method is to develop an efficient, memory-efficient FFT processor that requires less area. Using 90nm CMOS (Complementary Metal Oxide Semiconductor) technology, the proposed FFT processor was created and implemented in process. With reduced processing time, this means that the proposed FFT processor performs better than the prior memory-based FFT processors in terms of performance and the number of LUTs required which reduces area and memory utilization.

KEYWORDS: Digital Signal Processing (DSP), Fast Fourier Transform (FFT), Hybrid adder, Hybrid multiplier, Area, Speed, Memory.

I. INTRODUCTION

In wireless communication applications, one of the processes that is the most frequently used is the FFT; include the digital video broadcast terrestrial, ultra wideband, OFDM accesses, and signal processing applications [1]. They examine a number of pipelined FFT systems that make extensive use of the Singlepath delay feedback (SDF) and Multipath delay commutator (MDC) [2, 3]. For example, array signal processing, image processing, multiple-input multiple-output OFDM, and other applications need processing of numerous data streams.

In order to generate the outputs in natural order, it is necessary to perform multiple FFT operations simultaneously and to use a specific bit reversal circuit. FFT designs exist [4] that are capable of managing several separate data streams.

FFT architectures consist into four basic types:

1. Array architectures
2. Cache memory architectures
3. Memory based architectures
4. Pipelined architectures

Memory Based Architectures: FFT algorithms are typically executed in stages with memory access being used for data read and write operations at each stage [5]. Single memory and dual memory architectures are the two categories under which memory-based architectures are classified. One memory unit of at least N words is connected to the processor element over a bidirectional bus in a single memory architecture.

Cache Memory Architectures: A component of the cache memory architecture is the processor's data cache, which improves energy efficiency and memory access speed [6]. With the exception of the cache, this allows for data pre-fetching and is situated between the CPU (Central Processing Unit) and main memory, include single memory architecture. Considering of the additional hardware and complexity of the controller, this type of architecture is not commonly used.

Array Architectures: To perform the FFT computations, in array structures, several processing elements with local buffers are networked together. In addition to other factors, their large area requirements, these structures are also not often used.

Pipelined architectures: Researchers have been studying pipeline architectures since the 1970’s [7]. Based on decomposition techniques, there are various pipelined FFT structures available, following are the Radix systems: Radix-2 Multipath delay commutator (R2MDC) [8], Radix-4 single path delay commutator (R4MDC) [9], Radix-22 Single path Delay Feedback (R22SDF) and Radix-22 single path delay feedback (R22SDF).

In the present generation the advanced computing applications, communication applications, power utilization applications has been more popular [10]. They mainly depend on the size, cost and chip quality of the circuit. The primary factor in choosing an architecture considering the trade-off between speed requirements and hardware overhead. There is always a need for new low power design methodologies for Very-large-scale integration (VLSI), as the demand for more high-end, low-cost, and dependable products that run on remote power sources are continues increasing. The ability to store intermediate results in many memory banks, memory-based FFTs process the input recursively regardless of computation length through implementing a collection of PEs or a single butterfly Processing element (PE) [11, 12]. Memory-based FFTs outperform pipeline-based ones in terms of hardware efficiency [13].

The remaining paper is structured as follows: In this section II, an explanation of the literature survey is provided, Section III presents described design of High Speed Memory-Based FFT Processor Using 90nm Technology. Section IV provides an explanation of the experimental data, and Section V concludes the paper.

II. LITERATURE SURVEY

Fahad Qureshi et al. [14], introduce the FFT which is based on the building blocks by computing the address of elements. Radix-3, radix-5 butterflies are utilized to process the elements for computation. The reconfigurable elements are processed using the FFT methods Radix-2/3/4/5. The elements are processed using the Winograd Fourier transform technique. The multiplication method uses a constant multiplier rather than a general complex valued multiplier. In both memory and pipelined FFT architectures, the processing elements are utilized. Hence, this will reduce the hardware cost for processing the elements.

Tang S. N., Tsai J. W. and Chang T. Y., et. al. [15] A new technique to simplifying the multiple-path FFT algorithm is presented, which lowers the hardware cost in the multiplication units. Also provided is a multi-data scaling approach that decreases words while maintaining the signal-to-quantization-noise ratio. An Ultrawideband (UWB) application-specific and verification-focused 2048-point FFT processor test chip with a 128-point FFT kernel had been built using UMC (Unicorn Microelectronics Corporation) 90-nm 1P9M technology. Under the same Time Response (TR) a power consumption reduction of about 30% can be attained in comparison to the four-data-path strategy. Additionally, to achieve the UWB standard with a T.R. of 409.6 MS/s at 52 MHz (Mega Hz) and a power consumption decrease of about 40%, 6.8 milliwatts is the measured power consumption of the 128-point FFT kernel test chip. Z. Qian and M. Margala et. al. [16] offers a low-power, shared-memory SRFFT (Split-Radix Fast Fourier Transform) processor design. They demonstrate that a modified radix-2 butterfly unit can be used to compute SRFFT. The butterfly unit uses more hardware resources in order to save dynamic power by taking advantage of the multiplier-gating mechanism. Furthermore, two new address generating techniques are created for the nontrivial and trivial twiddle factors. When calculating a complex-valued transform with 1024 points, the suggested design generates a power consumption that is over 20% less than the standard radix-2 shared-memory implementations, according to simulation data. Hung C. C., Chen C. M., and Huang Y. H. et. al. [17] They create a FFT processor that is partially cached to take into consideration that the OFDMA system's users receive their resources. For the changeable FFT length and modulation order, they additionally develop a constellation- and power-aware twiddle-factor multiplier. Utilizing CMOS (Complementary Metal Oxide Semiconductor) technology, they create a combined pipelined/cached FFT processor with a point count of 128 to 1024. According to the chip measurement results, between 0.09 and 1.90 nanojoule (nJ) per FFT point, the energy dissipation varies and scales with the resources available in the OFDMA system.

Chen Y., Hsiao C. F., and Lee C. Y ., et. al. [18] To allow for the simultaneous use of traditional and prime-sized 2 *n* -point FFTs, for memory-based FFT processors, a Generalized mixed-radix (GMR) method is proposed. Through index vector control, the GMR algorithm may enable multibank memory architectures to maximize throughput without memory conflict, in addition to provided procedures for data processing and I/O (Input/Output) for constant data flow to reduce memory usage. Lastly, they also suggest an index vector generator implementation for described algorithm that is low-complexity. Akanksha Dixit and Vinod Kapse, et al [19], Although it can lower power dissipation, which is a primary criterion for low power digital design, the reversible logic has received a lot of attention in recent years. It is widely used in optical information processing, low power design, and advanced computing. The logic processes in conventional digital circuits cause bits of information to be removed, which results in a large energy loss. Therefore, a significant reduction in power consumption can be achieved if logic gates are designed in a way that prevents the destruction of information bits.

H. F. Luo, Liu Y. J. and Shieh M. D., et. al. [20] investigates effective memory management approaches for FFT memory-based systems. Memory-based FFT systems should use less area and power in order to minimize their requirements, a data relocation strategy that combines many banks is proposed. In terms of power consumption and area, the derived architecture performs better than traditional memory-based FFT solutions that use dual-port memory. To further minimize power dissipation, the suggested approach is expanded to a cached-memory FFT architecture. According to experimental data, the suggested memory method uses 9.6%–67.9% less power and 10.1%–29.3% less area than the multibank design.

Suzuki H., Ono T., Yamanashi Y. and Yoshikawa N., et. al. [21] utilizing Single-flux-quantum (SFQ) logic circuits, have been developing a high-speed FFT processor. A 4-bit butterfly processor and a data-shuffling circuit were developed and using the AIST 10 kA/cm ^{2} Nb advanced process 2, and a twiddle factor ROM with maximum

***Article Received:** 20 March 2023 **Revised:** 17 April 2023 **Accepted:** 20 May 2023*

frequencies of 51.6, 59.5, and 51.5 was used in earlier research for 4-bit 8-point FFT. In this work, they designed and presented residual component circuits, often known as rounding and data buffer circuits, in order to complete designing the FFT processor, with a target frequency of 50 GHz. Additionally, they created and verified through experimentation the functionality of a single-chip FFT processor based on SFQ that integrates all of the component circuits.

Ma Z. G., Xing Q. J., and Xu Y. K., et. al. [22] offers a new conflict-free access method designed for FFT processors based on memory. It is demonstrated to meet the requirements of the variable-size, parallel processing, continuous flow, and mixed radix FFT calculations. Also, an address generation unit is developed its lower hardware complexity and lower gate delay outperform those of present systems.

Lekshmi Viswanath and Ponni. M, et al [23], has introduced the FFT architecture which is reversible in nature. These are mostly used in communication, computer graphics, digital signal processing, and cryptography applications. Computations are performed effectively because of reversibility and this plays major role in entire operation. G. Kang, W. Choi and J. Park, et. al. [24] demonstrate memory customisation methods based on embedded Dynamic random access memory (eDRAM) for low-cost FFT processor architecture. The basic concept comes from that because eDRAM has regular and predictable memory access patterns, the FFT processor can be utilized to effectively modify memory. In comparison to the static RAM-based FFT architecture, the proposed eDRAM-based pipelined and cached-memory FFTs offer 26.8% and 33.2% power reductions, respectively, based on the hardware implementation results using a 0.11um CMOS technology for a 2k-point FFT.

III. HIGH SPEED MEMORY-BASED FFT PROCESSOR

The block diagram for design of high Speed memory-based FFT processor using 90nm technology is represented in below Figure 1.

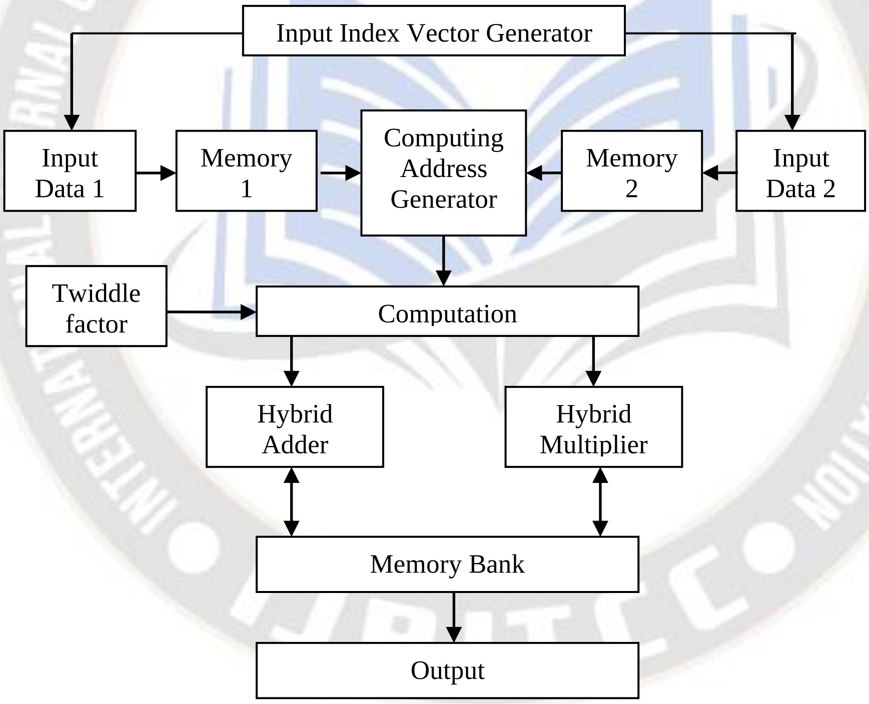

Figure 1: BLOCK DIAGRAM OF HIGH SPEED MEMORY-BASED FFT PROCESSOR

Input index vector generator will take inputs as Input 1, Input 2 and transfer the inputs to scaling unit controls the scaling operations.

Address generation unit will generate the address to obtain data and saved in the memory-1 block. Similarly same operation is performed and saved that data in memory-2 block. All these data will be computed using p×p computator. The computed data is multiplied using modular multiplication unit.

Every cycle generates concurrent data based on the compute address generator. Computation address generator block will store the intermediate results. All symbols are operated in ping pong mode which is continuous in nature. Input sampling is performed at the output side memory.

This architecture uses RAM in order to store desired Twiddle factors for computation process, which ultimately reduces the architecture to some extent. The only component needed for the twiddle factor multiplier is a ROM (Read Only Memory) and a complicated multiplier. Pre-computed twiddle factors are stored in the ROM.

For computation process the Hybrid Adders and Hybrid Multipliers are used. In Hybrid Carry Look Ahead (CLA) adder generators, propagators, generators, and carry generation cells are the main adders that they utilize. An operand's n-bits are split up into groups, each of which has a CLA. An RCA (Ripple Carry Adder) connects the groups. For ease of design and adaptability, the n-bits are separated into equal size segments. The CLA functions by analyzing input signals to determine whether or not carry will occur even before the addition is really carried out. There are three stages in the CLA circuit. The levels are

- Carry propagate/ generate Pi and Gi
- Performing actual addition operation Sum bit Si
- Carry generation Ci+1.

Although the CLA is faster, its use of composite carry expressions makes it less dependable when handling a high bit count. The CLA adders disadvantage depending on the bit count, circuit realization becomes challenging.

There are four modules that made up the overall Hybrid Multiplier system. The partial product, hybrid adder with Gate diffusion input (GDI), multiplier register, multiplicand register, and adder register are the four modules. The coprocessor in serial format is the primary application for this procedure. The coprocessor's functionality is offered by this system. In this block diagram, the multiplicand (Md) and multiplier (Mr) registers are the two available registers. The alignment of the partial products will be completed first. Subsequently, partial products use these registers to generate signals that propagate. Alignment of the dual propagator generator will be handled after this. Here, registers are used to load the operands into the multiplier first. Arithmetic circuits are used to carry out arithmetic operations such as addition and multiplication. The result that is obtained from this will be stored in the register. The root and load multiplier (Mr) from the multiplier register are used as the block's inputs.

The result register block contains the multiplier and multiplicand for the result. The multiplier and multiplicand register blocks are the values for both a(t) and b(t) are assigned. The bits will be shifted to the finite field arithmetic block by the values obtained in the multiplier register block. The arithmetic operations of addition and multiplication will be carried out by this block. The bits are moved to the result register after a specific operation. The output of the finite field arithmetic circuit will be saved in this result register. Finally, the multiplicand register will efficiently complete the parallel process.

Memory bank address along with address generation unit will save address and generate the address for saved data. Efficient routing mechanism is provided between the memory block and commutators block. Address generator will control the computed data. In the register files are the parameters like FFT size and twiddle factors are saved. Working modes are utilized in FFT for the process of configuration.

By using continuous flow mode, the FFT is operated. This is based on the place strategy and consists of three memory banks. In stage 0, only on modular multiplier unit will be active and at other stages two radix based modular multiplier stages are activated. Based on number of clock cycles the computation is completed. The both input and output index vector generated is merged into one at output side. A forward binary form of the index is used for the input side and a reverse binary representation is used for the output side. Input data is distributed by using the index vector generator. This is obtained because of memory positions of input data is placed properly. Again the output data is reordered by using the natural sequence.

IV. RESULT ANALYSIS

This section presents the simulation results and their comparison. The entire memory based FFT processor is simulated in Xilinx technology. Processing time, Area and Memory consumption are used parameters for performance analysis. Below Table 1 shows the comparative analysis of described High Speed Memory-Based FFT Processor with Basic FFT processors in terms of Processing time, Area and Memory consumption.

Table 1: COMPARATIVE PERFORMANCE ANALYSIS

Parameters	High Speed Memory-Based FFT Processor	Basic FFT processors
Processing time (msec)	0.12	1.31
LUTs used	541	589
Memory consumption	305196 kilobytes	346127 kilobytes

Figure 2 shows the comparative processing time analysis for two models as High Speed Memory-Based FFT Processor and Basic FFT processors. It is clear that, described High Speed Memory-Based FFT Processor achieves less processing time which ultimately state that, described model attains high speed.

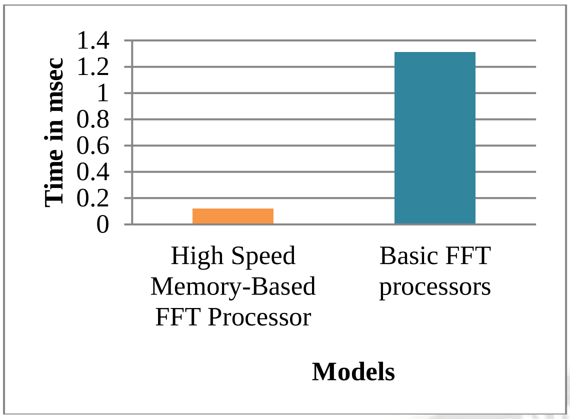

Figure 2: COMPARATIVE PROCESSING TIME ANALYSIS

Comparative analysis in terms of number of LUTS used is represented in below Figure 3. From results it is clear that, described High Speed Memory-Based FFT Processor attains less number of LUTs (Look Up Table) compared to Basic FFT processors. Therefore Area of the described model is less compared to other model.

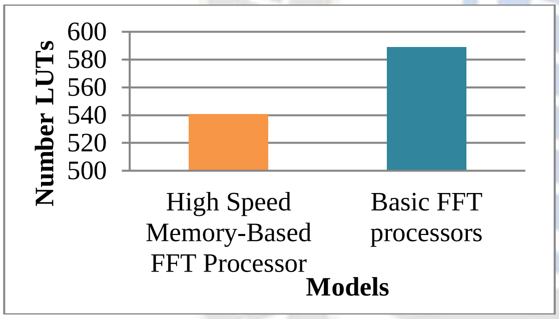

Figure 3: COMPARATIVE ANALYSIS FOR NUMBER OF LUTs

Figure 4 shows the comparative Memory Consumption analysis for two models as High Speed Memory-Based FFT Processor and Basic FFT processors. From results it is clear that, memory consumption of described model is less compared to other model.

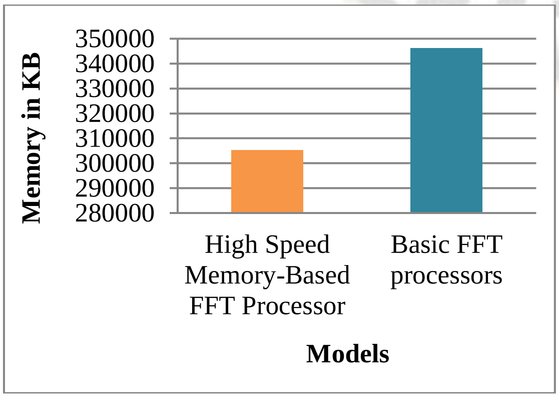

Figure 4: COMPARATIVE ANALYSIS FOR MEMORY CONSUMPTION

From overall performance analysis it is found that, described high speed memory based FFT processor is more efficient than Basic FFT processors with Processing time 0.12 msec, LUTs used are 541 and 305196 kilobytes of Memory consumption.

V. CONCLUSION

This analysis describes the design of a high-speed memory-based FFT processor using 90nm technology. In wireless communication applications, one of the most

often utilized processes is the FFT. The novel hybrid multiplier and hybrid adder is used in this paper. Processing time, Area and Memory consumption are used parameters for performance analysis. From overall performance analysis it is found that, described High Speed Memory Based FFT Processor is more efficient than Basic FFT processors with Processing time 0.12 msec which improves the speed, the LUTs used are 541 which reduce the area and 305196 kilobytes of Memory consumption.

REFERENCES

[1] M. Garrido, N. K. Unnikrishnan and K. K. Parhi, "A Serial Commutator Fast Fourier Transform Architecture for Real-Valued Signals," in IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 65, no. 11, pp. 1693-1697, Nov. 2018, doi: 10.1109/TCSII.2017.2753941.

[2] X. Y. Shih, H. R. Chou and Y. Q. Liu, "Design and Implementation of Flexible and Reconfigurable SDF-Based FFT Chip Architecture With Changeable-Radix Processing Elements," in IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 65, no. 11, pp. 3942-3955, Nov. 2018, doi: 10.1109/TCSI.2018.2860942.

[3] M. Garrido, S. J. Huang and S. G. Chen, "Feedforward FFT Hardware Architectures Based on Rotator Allocation," in IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 65, no. 2, pp. 581-592, Feb. 2018, doi: 10.1109/TCSI.2017.2722690.

[4] 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," in IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 62, no. 12, pp. 2836-2845, Dec. 2015, doi: 10.1109/TCSI.2015.2495724.

[5] M. Garrido, S. J. Huang, S. G. Chen and O. Gustafsson, "The Serial Commutator FFT," in IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 63, no. 10, pp. 974-978, Oct. 2016, doi: 10.1109/TCSII.2016.2538119.

[6] E. Bolotin, D. Nellans, O. Villa, M. O'Connor, A. Ramirez and S. W. Keckler, "Designing Efficient Heterogeneous Memory Architectures," in IEEE Micro, vol. 35, no. 4, pp. 60-68, July-Aug. 2015, doi: 10.1109/MM.2015.72

[7] Yun-Nan Chang and K. K. Parhi, "An efficient pipelined FFT architecture," in IEEE Transactions on Circuits and Systems II: Analog and Digital Signal

***Article Received:** 20 March 2023 **Revised:** 17 April 2023 **Accepted:** 20 May 2023*

Processing, vol. 50, no. 6, pp. 322-325, June 2003, doi: 10.1109/TCSII.2003.811439.

[8] A. X. Glittas, M. Sellathurai and G. Lakshminarayanan, "A Normal I/O Order Radix-2 FFT Architecture to Process Twin Data Streams for MIMO," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 24, no. 6, pp. 2402-2406, June 2016, doi: 10.1109/TVLSI.2015.2504391.

[9] S. Badar and D. R. Dandekar, "High speed FFT processor design using radix −4 pipelined architecture," 2015 International Conference on Industrial Instrumentation and Control (ICIC), Pune, India, 2015, pp. 1050-1055, doi: 10.1109/IIC.2015.7150901.

[10] M. Garrido, "The Feedforward Short-Time Fourier Transform," in IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 63, no. 9, pp. 868-872, Sept. 2016, doi: 10.1109/TCSII.2016.2534838.

[11] S. N. Tang, F. C. Jan, H. W. Cheng, C. K. Lin and G. Z. Wu, "Multimode Memory-Based FFT Processor for Wireless Display FD-OCT Medical Systems," in IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 61, no. 12, pp. 3394-3406, Dec. 2014, doi: 10.1109/TCSI.2014.2327315

[12] S. Liu and D. Liu, "A High-Flexible Low-Latency Memory-Based FFT Processor for 4G, WLAN, and Future 5G," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 27, no. 3, pp. 511-523, March 2019, doi: 10.1109/TVLSI.2018.2879675.

[13] J. Greg Nash, "High-throughput programmable systolic array FFT architecture and FPGA implementations," 2014 International Conference on Computing, Networking and Communications (ICNC), Honolulu, HI, USA, 2014, pp. 878-884, doi: 10.1109/ICCNC.2014.6785453.

[14] F. Qureshi, M. Ali and J. Takala, "Multiplierless reconfigurable processing element for mixed radix-2/3/4/5 FFTs," 2017 IEEE International Workshop on Signal Processing Systems (SiPS), Lorient, France, 2017, pp. 1-6, doi: 10.1109/SiPS.2017.8110007.

[15] S. N. Tang, J. W. Tsai and T. Y. Chang, "A 2.4-GS/s FFT Processor for OFDM-Based WPAN Applications," in IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 57, no. 6, pp. 451-455, June 2010, doi: 10.1109/TCSII.2010.2048373

[16] Z. Qian and M. Margala, "Low-Power Split-Radix FFT Processors Using Radix-2 Butterfly Units," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 24, no. 9, pp. 3008-3012, Sept. 2016, doi: 10.1109/TVLSI.2016.2544838.

[17] C. M. Chen, C. C. Hung and Y. H. Huang, "An Energy-Efficient Partial FFT Processor for the OFDMA Communication System," in IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 57, no. 2, pp. 136-140, Feb. 2010, doi: 10.1109/TCSII.2010.2040318.

[18] C. F. Hsiao, Y. Chen and C. Y. Lee, "A Generalized Mixed-Radix Algorithm for Memory-Based FFT Processors," in IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 57, no. 1, pp. 26-30, Jan. 2010, doi: 10.1109/TCSII.2009.2037262.

[19] Akanksha Dixit and Vinod Kapse, “Arithmetic & Logic Unit (ALU) Design using Reversible Control Unit”, International Journal of Engineering and Innovative Technology (IJEIT) Volume 1, Issue 6, June 2014

[20] H. F. Luo, Y. J. Liu and M. D. Shieh, "Efficient Memory-Addressing Algorithms for FFT Processor Design," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 23, no. 10, pp. 2162-2172, Oct. 2015, doi: 10.1109/TVLSI.2014.2361209.

[21] T. Ono, H. Suzuki, Y. Yamanashi and N. Yoshikawa, "Design and Implementation of an SFQ-Based Single-Chip FFT Processor," in IEEE Transactions on Applied Superconductivity, vol. 27, no. 4, pp. 1-5, June 2017, Art no. 1301505, doi: 10.1109/TASC.2017.2667398

[22] Q. J. Xing, Z. G. Ma and Y. K. Xu, "A Novel Conflict-Free Parallel Memory Access Scheme for FFT Processors," in IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 64, no. 11, pp. 1347-1351, Nov. 2017, doi: 10.1109/TCSII.2017.2683643.

[23] Lekshmi Viswanath and Ponni. M, “Design and Analysis of 16 Bit Reversible ALU”, IOSR Journal of Computer Engineering (IOSRJCE), ISSN : 2278-0661 Volume 1, Issue 1 , PP 46-53 (May-June 2012),

[24] G. Kang, W. Choi and J. Park, "Embedded DRAM-Based Memory Customization for Low-Cost FFT Processor Design," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 25, no. 12, pp. 3484-3494, Dec. 2017, doi: 10.1109/TVLSI.2017.2752265.