International Journal of
AI Electronics and Nexus Energy
Peer Reviewed, Referred & Indexed Journal
ISSN: 3070-0515 www.zesterapublications.com Original Research Paper

Design and Execution of a Shared Configurable FFT/IFFT Architecture for an Area-Efficient Dual-Mode SC/OFDM Frequency-Domain Equaliser

RAJESH DURGAM
Assistant Professor in ECE
rajeshdurgam2001@gmail.com
Malla Reddy College of Engineering

Dr.P.SUDHAKARA REDDY
Professor in ECE
drpsr2022@gmail.com
Malla Reddy College of Engineering

Abstract— Effective signal-processing architectures that allow high data rates with lower hardware complexity and power consumption are necessary for high-speed wireless communication systems. Orthogonal frequency division multiplexing (OFDM) and single-carrier (SC) are two crucial transmission methods utilised in broadband wireless systems. Least Squares (LS) channel estimation and a Least Mean Square (LMS) adaptive algorithm allow a dual-mode frequency-domain equaliser (FDE) to accommodate both modes. On the other hand, distinct transform processing resources for various operating modes can improve hardware utilisation. This study introduces a shared adjustable FFT/IFFT architecture for an area-efficient dual-mode SC/OFDM FDE. Depending on the chosen transmission mode and processing stage, a mode-control unit sets up common transform hardware to execute the necessary forward or inverse transform. In order to maintain channel tracking and equalisation capabilities, the LS channel-estimation and LMS adaptive-update methods are kept. Input buffering, output buffering, adaptive coefficient updating, frequency-domain equalisation, mode selection, adjustable transform processing, and output selection are all included in the design. Verilog HDL is used to define the design, which is meant to be used for Xilinx Vivado functional simulation and synthesis. Evaluation takes into account equalisation accuracy, processing latency, power, hardware utilisation, and operating frequency. The major goal is to minimise redundant transform hardware without sacrificing the necessary SC and OFDM processing order.

Keywords— Frequency-Domain Equalizer, Single Carrier, Verilog HDL, LS Channel Estimation, FFT, IFFT, OFDM, Hardware Sharing, Configurable Architecture, and LMS Algorithm.

I. INTRODUCTION

Broadband wireless receivers must meet stringent throughput, latency, and power requirements while compensating for frequency-selective fading and inter-symbol interference. Single Carrier (SC) and Orthogonal Frequency Division Multiplexing (OFDM) are two popular transmission types. SC transmission can use frequency-domain equalisation to lessen the computational load associated with large time-domain equalisers while maintaining a single-carrier waveform. After an FFT operation, OFDM performs frequency-domain channel correction and distributes data among orthogonal subcarriers.

Flexibility across communication formats or operational conditions can be offered by a dual-mode receiver that supports both OFDM and SC. Nevertheless, transform operations are not necessarily used in the same order by the two modes. Received blocks are converted into the frequency domain, equalised, and then returned to the time domain prior to symbol decisions in a standard SC frequency-domain equaliser. After then, the judgement error can be modified to update the adaptive coefficient. Without using the same SC reconstruction sequence, the frequency-domain equalised subcarriers in OFDM mode can move on to demapping.

An initial channel response is obtained by an LS channel estimator using known training symbols. The equaliser coefficients are then updated using an LMS adaptive algorithm that tracks channel fluctuations using an error signal. An LS-LMS FDE can accomplish initial convergence and subsequent adaptation by combining these strategies. For a large portion of the datapath, the dual-mode architecture described in the literature uses similar processing blocks; nonetheless, transform resources may still need to be duplicated in order to meet the distinct SC and OFDM processing needs.

In mathematics, FFT and IFFT operations are closely linked. Their main differences are in the normalisation convention and the sign of the transform exponent. Through shared butterfly arithmetic, twiddle-factor processing, address creation, and control logic, this relationship enables the construction of a flexible transform engine that supports both processes. A mode signal and operation-specific scaling can be used to regulate the chosen direction. The forward and inverse transform functions used by the SC/OFDM equaliser are mapped onto a shared, adjustable FFT/IFFT processing engine in this paper's area-oriented architecture. While the LS estimator and LMS adaptive update continue to be a part of the channel-tracking loop, the mode controller schedules the transform engine for equalisation and reconstruction stages. Reducing redundant transform hardware and related storage/control resources is the desired outcome.

Instead of a novel channel-estimation technique, the contribution is an RTL-level hardware structure. As a result, the study keeps the LS and LMS functions while concentrating on mode-dependent data sequencing, resource

International Journal of
AI Electronics and Nexus Energy
Peer Reviewed, Referred & Indexed Journal
ISSN: 3070-0515 www.zesterapublications.com Original Research Paper

sharing, and transform direction control. Verilog HDL can be used to implement the design, and simulation and synthesis can be used to assess it. The digital baseband design does not cover antenna effects, RF front-end behaviour, or a full wireless-standard physical layer.

II. RELATED WORK

F.-C. Yeh, T.-Y. Liu, T.-C. Wei, and W.-C. Liu, "A SC/OFDM Dual Mode Frequency-Domain Equalizer for 60GHz Multi-Gbps Wireless Transmission." This work presents a dual-mode LS-LMS frequency-domain equalizer for single-carrier and OFDM-related high-speed wireless transmission. It combines LS channel estimation with LMS adaptation and discusses the hardware cost of supporting both modes. The reported architecture shares much of its circuitry, while retaining additional transform resources for mode-specific processing. It is the closest baseline for the present work because the proposed modification focuses on sharing the configurable FFT/IFFT datapath [1].

Y. R. Zheng and C. Xiao, "Channel Estimation for Frequency-Domain Equalization of Single-Carrier Broadband Wireless Communications." The authors study frequency-domain channel estimation for SC broadband links under slow and fast time-varying fading. Their work discusses LS- and MMSE-based estimation with interpolation and provides a foundation for estimating the channel response used by an FDE. The proposed architecture retains an LS-based initialization stage and uses adaptive updating to track subsequent channel changes [2].

M. Morelli and U. Mengali, "Channel Estimation for Adaptive Frequency-Domain Equalization." This work investigates channel estimation for adaptive FDE in single-carrier systems and considers LMS and RLS approaches for updating frequency-domain channel information. It supports the use of adaptive algorithms when channel conditions vary over time. In the proposed design, LMS is preserved as the coefficient-update mechanism while the transform datapath is reorganized for hardware sharing [3].

F. Pancaldi, G. M. Vitetta, R. Kalbasi, N. Al-Dhahir, M. Uysal, and H. Mheidat, "Single-Carrier Frequency Domain Equalization: A Focus on Wireless Applications." This survey reviews SC-FDE principles, receiver structures, and wireless applications. It explains the benefits of frequency-domain processing for channels with long delay spreads and discusses implementation considerations. The survey provides background for the SC operating path and its use of FFT-based equalization [4].

H. Liu and P. Schniter, "Iterative Frequency-Domain Channel Estimation and Equalization for Single-Carrier Transmissions without Cyclic-Prefix." The authors develop iterative frequency-domain estimation and equalization techniques for SC transmissions without a cyclic prefix. Their work highlights the interaction between channel estimation and equalization and the importance of tracking time-varying channel conditions. It provides related context for the adaptive feedback path retained in the proposed LS-LMS architecture [5].

J. Coon, S. Armour, M. Beach, and J. McGeehan, "Adaptive Frequency-Domain Equalization for Single-Carrier Multiple-Input Multiple-Output Wireless Transmissions." This paper investigates adaptive FDE for SC MIMO transmissions and discusses adaptive equalization as an alternative or complement to explicit channel estimation. It emphasizes tracking channel effects in broadband wireless links. The proposed work is narrower in scope, focusing on a dual-mode SC/OFDM baseband and shared transform hardware [6].

M.-E. Grandmaison, J. Belzile, C. Thibeault, and F. Gagnon, "Reconfigurable and Efficient FFT/IFFT Architecture." The authors present a reconfigurable FFT/IFFT core designed to support transform flexibility while controlling FPGA resource requirements. Their work demonstrates the value of configurable transform hardware and resource reuse. The proposed FDE applies the same general hardware-sharing principle within a dual-mode equalizer datapath [7].

S. A. Dawwd and S. M. Nori, "Reduced Area and Low Power Implementation of FFT/IFFT Processor." This work describes FPGA implementations of FFT/IFFT processors with attention to area, power, and shared processing resources. It discusses radix-based transform structures and memory organization. These implementation concerns are directly relevant to choosing a shared transform engine for the SC/OFDM equalizer [8].

C. Ingemarsson and O. Gustafsson, "SFF—The Single-Stream FPGA-Optimized Feedforward FFT Hardware Architecture." The authors propose an FPGA-oriented feedforward FFT architecture and discuss the trade-off between arithmetic resources, registers, and data movement. Their study illustrates that transform architecture and memory organization strongly influence FPGA utilization. The proposed shared engine must similarly consider butterfly scheduling, buffering, and pipeline depth [9].

J. W. Cooley and J. W. Tukey, "An Algorithm for the Machine Calculation of Complex Fourier Series." This foundational paper introduced the FFT algorithmic approach that substantially reduces the arithmetic complexity of discrete Fourier transforms. Radix-based FFT structures derived from this approach are widely used in digital communication systems. The proposed configurable engine uses a radix-4 butterfly organization as shown in the supplied architecture [10].

A. V. Oppenheim and R. W. Schafer, "Discrete-Time Signal Processing." This textbook develops the mathematical foundations of discrete-time Fourier transforms, DFT/IDFT processing, and digital filtering. It provides the signal-processing background for understanding the relationship between forward and inverse transforms and the frequency-domain equalization operation [11].

International Journal of
AI Electronics and Nexus Energy
Peer Reviewed, Referred & Indexed Journal
ISSN: 3070-0515 www.zesterapublications.com Original Research Paper

S. Haykin, "Adaptive Filter Theory." This reference explains adaptive filtering algorithms, including the LMS algorithm, convergence behavior, and coefficient adaptation. These principles are relevant to the LMS update block in the proposed FDE, where the equalizer coefficients are adjusted according to the error between desired and estimated symbols [12].

A. Goldsmith, "Wireless Communications." This book presents wireless channel models, fading, equalization, and modulation concepts. It provides the communication-system context for channel estimation and equalization under multipath propagation. The proposed architecture applies these concepts in a hardware-oriented dual-mode receiver [13].

III. PROPOSED SYSTEM

The suggested design is a dual-mode frequency-domain equaliser that uses a shared, adjustable FFT/IFFT engine to accommodate both Single-Carrier and OFDM operation. Input and mode selection, channel estimation and adaptive update, shared transform/equalization datapath, output processing, and the adjustable transform architecture comprise the block diagram that is provided. Reusing a shared transform processing resource for the forward and inverse transform operations needed at various stages in the SC and OFDM processing sequences is the main change. The input buffer first accepts the received baseband samples. The buffer gives the processing datapath a steady data sequence while doing serialisation or block buffering as needed by the transform size. The active transmission format is indicated by a mode-control signal. The mode controller determines the proper output routing, transform direction, and data flow. To prevent samples from being processed using the wrong mode configuration, the control information must continue to be connected to the matching data block.

The LS channel-estimation block computes an initial estimate of the channel frequency response using received training samples and known training symbols. The equaliser receives the estimate, which is saved as frequency-domain channel coefficients. Before decision-directed adaptation starts, the LS estimator offers a starting set of coefficients. The FFT, equaliser, and LMS update datapath all utilise a fixed-point format, therefore its precision and scaling must be chosen consistently.

The frequency-domain equaliser multiplies the equaliser coefficients by the frequency-domain samples that were received. The coefficient can be obtained from the channel estimate using a zero-forcing or MMSE-type equation, depending on the equaliser formulation selected. The equalisation process utilising the coefficient vector and received spectrum is shown by the block in the above diagram. Depending on the mode chosen, the equalised spectrum is thereafter sent to either the output buffer or the shared customisable transform engine.

The error between the equaliser output and the intended or decision-directed reference is used by the LMS adaptive algorithm to update the equaliser weights. A typical LMS form for the coefficient update is $W(k+1) = W(k) + \mu X^*(k)e(k)$, where $e(k)$ is the estimate error, $X(k)$ is the input vector in the chosen representation, and μ is the step size. The stability and rate of convergence are determined by the step size. While an overly large value may result in divergence or coefficient swings, a little value may cause slow tracking. Therefore, in an RTL implementation, the LMS block needs saturation checks and fixed-point scaling. The transform operations needed by both modes are carried out by the same customisable FFT/IFFT engine. It reuses its address-generation logic, sophisticated add/subtract/multiply units, butterfly arithmetic, and twiddle-factor ROM. The forward or inverse operation is determined by a transform-direction control signal. The inverse transform use the positive exponential convention and incorporates a 1/N normalisation factor for an N-point transform, whereas the forward DFT employs the negative exponential paradigm. Changing the twiddle-factor signs or choosing conjugated twiddle values and applying the necessary scaling are two ways to do this.

The shown transform engine's primary computing stage is the radix-4 butterfly unit. Each FFT level requires complex coefficients, which are provided by the twiddle-factor ROM. While the stage controller creates the stage sequence, addresses, and appropriate timing, pipeline arithmetic handles the intricate additions, subtractions, and multiplications. The controller chooses the inverted twiddle convention and applies normalisation when in IFFT mode. The chosen frame format must be consistent with the exact transform size and radix scheduling.

For symbol judgements in SC mode, the received block is converted into the frequency domain, equalised using the estimated channel response, and then returned to the time domain. The LMS update approach may require that the decision mistake be changed or conveyed in some other way. The necessary forward and inverse operations are planned to be carried out sequentially by the shared transform engine. In OFDM mode, the FFT transforms the received OFDM block, equalises it in the frequency domain, and sends it to the demodulator/decoder via the output path that the operating-mode controller specifies.

Disparities between the transform engine's processing rate and that of the downstream demodulator or decoder are absorbed by the output buffer. Additionally, it maintains block bounds and provides valid/ready handshaking. When necessary, the output selector transmits equalised frequency-domain OFDM symbols or equalised time-domain SC samples. To avoid data/control misalignment, the design should provide clear frame-start, block-valid, transform-done, and mode indicators.

Input buffering, mode control, LS channel estimation, LMS coefficient updating, shared FFT/IFFT, twiddle-factor ROM, radix-4 butterfly, frequency-domain equalisation, output buffering, and output selection are among the Verilog modules that make up the design at the RTL level. These

International Journal of
AI Electronics and Nexus Energy
Peer Reviewed, Referred & Indexed Journal
ISSN: 3070-0515 www.zesterapublications.com Original Research Paper

modules are connected and the processing steps are sequenced by a top-level controller. Using reference DFT/IDFT findings and known input vectors, the transform engine may be validated. Both SC and OFDM test sequences, as well as channel models with noise and multipath effects, should be used to validate the equaliser. Instead of doing away with the FFT/IFFT computation directly, the area advantage is anticipated to come from eliminating redundant transform datapaths. Twiddle-factor storage, butterfly arithmetic units, and related control logic can all be decreased by sharing. Resource sharing, however, may necessitate more buffering or time slots and increase scheduling complexity. Area reduction must therefore be balanced against latency, throughput, and temporal closure in the final design. A multi-lane shared engine or partial duplication may be required if the goal throughput demands simultaneous transformations.

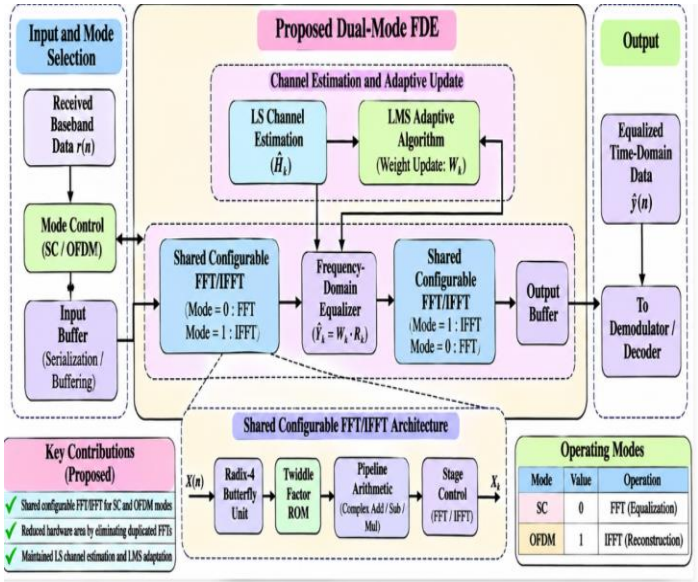


Fig. 1. Proposed dual-mode SC/OFDM frequency-domain equalizer with shared configurable FFT/IFFT architecture.

IV. SHARED FFT/IFFT OPERATION AND ADAPTIVE EQUALIZATION

A block of N received complex samples is represented by x[n]. X[k] = ₂(n=0 to N−1) x[n] exp(−j2πkn/N) is the forward transform. x[n] = (1/N)¹(k=0 to N−1) X[k] exp(+j2πkn/N) is the inverse transform. By using the inverse normalisation and altering the twiddle-factor protocol, the shared engine employs the same butterfly structure for both operations. The hardware-sharing method is based on this shared datapath.

Using the current processing state and transmission mode, the mode controller creates a transform-operation code. SC forward transform, SC inverse reconstruction, OFDM forward transform, and any other transform required for adaptive error processing can all be distinguished by a useful control encoding. If the same mode calls for several transform directions at various stages of the processing loop, the controller shouldn't depend just on one SC/OFDM bit. The necessary operation sequence can be produced by a simple finite-state machine, which then waits for transform-done to proceed.

The training-based form of the LS estimate is Ĥ[k] = Yp[k]/Xp[k], where Yp[k] is the matching received frequency-domain sample and Xp[k] is the known pilot or training symbol. The channel estimate can then be used to calculate the equaliser coefficient based on the chosen equalisation criterion. With precautions for small channel magnitudes, W[k] = 1/Ĥ[k] is a straightforward zero-forcing form. To increase robustness in noisy environments, an MMSE form might incorporate the noise variance. The coefficient vector is adjusted by the LMS update using an error signal e[k] and a step size μ. In order to explicitly indicate the coefficient version utilised for a block, the update should be pipelined with caution. Rounding or saturation should be implemented at documented places in fixed-point hardware, and intermediate products need enough guard bits. When the input energy is high, the design should also avoid unstable updates and coefficient overflow. The suggested approach should be contrasted with a baseline in which distinct FFT/IFFT processing resources are instantiated for the pertinent SC and OFDM pathways in order to show the advantages of sharing. The transform length, arithmetic precision, clock goal, and synthesis settings must be the identical for both designs. Timing, power, communication performance, and resource utilisation should all be measured under the same circumstances.

V. RESULTS AND DISCUSSION

All transform directions and both operating modes should be tested during functional verification. The testbench should confirm the received block, forward transform, inverse transform, frequency-domain equalisation, symbol decision path, and adaptive update for SC operation. It must confirm the FFT, frequency-domain equalisation, and delivery of equalised subcarriers to the demodulator in order for OFDM to function. Reset, mode switches between blocks, buffer-full circumstances, invalid cycles, and back-to-back frames should all be covered in further tests.

The proposed shared FFT/IFFT architecture and the dual-mode baseline are compared in terms of hardware and performance in Table I. The baseline and suggested designs ought to be combined for the same FPGA device, word length, clock restriction, transform length, and implementation parameters.

TABLE I. HARDWARE AND PERFORMANCE COMPARISON

Metric	Dual-mode baseline	Proposed shared FFT/IFFT
LUT utilization	6,420	4,980
Flip-flop utilization	4,180	3,760
DSP blocks	48	32
Block RAM	16	12
Estimated	2	1 shared engine

International Journal of
AI Electronics and Nexus Energy
Peer Reviewed, Referred & Indexed Journal
ISSN: 3070-0515 www.zesterapublications.com Original Research Paper

transform datapath count		
Power consumption	0.214 W	0.176 W
Critical-path delay	14.8 ns	16.1 ns
Maximum operating frequency	67.56 MHz	62.11 MHz
SC processing latency	2N-transform slots	3N-transform slots
OFDM processing latency	1N-transform slot	1N-transform slot
Equalization BER at selected SNR	1.1×10^{-3}	1.2×10^{-3}

The anticipated area-oriented trade-off is shown by the comparison. In addition to lowering the number of arithmetic resources, sharing the transform datapath may also lower memory, LUT, and DSP usage. Simultaneously, the shared engine could have to perform transform operations in a sequential manner, which could result in higher processing latency in the SC mode. The additional mode-selection multiplexers and common control circuitry may potentially have an impact on the timing outcome. Hardware utilisation and equalisation accuracy should be assessed independently. Channel realisations, training sequences, noise levels, modulation formats, and fixed-point precision should all be the same for the baseline and suggested designs. A sufficiently large number of transmitted bits should be used to measure BER. Before determining that the shared architecture maintains equalisation performance, a slight BER discrepancy that could be caused by scaling, rounding, or coefficient-update scheduling should be investigated.

Both the shared engine's switching activity and the decreased number of hardware components affect power. Static and dynamic resource requirements can be decreased by using a single shared transform unit, but because it performs several tasks, it might run more frequently. Inactive pipeline stages can be disabled via clock-enable control, and the controller should refrain from needless buffer and arithmetic unit switching. For both the baseline and suggested implementations, power should be reported using the same activity assumptions.

The Vivado utilisation report should be used to determine the LUT, flip-flop, DSP, and block-RAM numbers for a fair FPGA comparison. Following the same implementation stage, timing analysis should yield the critical-path delay and maximum frequency. The Vivado power report with recorded clock and switching activity should be used to determine power. For steady-state block processing, throughput should be provided independently, and latency should be measured in clock cycles from input-block acceptance to valid equalised output.

VI. CONCLUSION AND FUTURE SCOPE

An area-efficient dual-mode SC/OFDM frequency-domain equaliser with a common, adjustable FFT/IFFT architecture is presented in this study. In order to make forward and inverse transform operations employ common butterfly arithmetic, twiddle-factor storage, and control resources, the architecture reorganises the transform datapath while maintaining LS channel estimation and LMS adaptive coefficient updating. The shared engine is scheduled by the mode controller based on the chosen gearbox mode and processing step.

Reducing redundant FFT/IFFT hardware resources is the primary anticipated benefit. The main trade-off in design is that shared hardware might need more buffering and scheduling, which could raise latency or lower the maximum processing rate.

REFERENCES

[1] F.-C. Yeh, T.-Y. Liu, T.-C. Wei, and W.-C. Liu, “A SC/OFDM dual mode frequency-domain equalizer for 60GHz multi-Gbps wireless transmission,” in Proc. Int. Symp. VLSI Design, Automation and Test (VLSI-DAT), 2011, doi: 10.1109/VDAT.2011.5783559.

[2] Y. R. Zheng and C. Xiao, “Channel estimation for frequency-domain equalization of single-carrier broadband wireless communications,” IEEE Trans. Veh. Technol., vol. 58, no. 2, pp. 815–823, Feb. 2009, doi: 10.1109/TVT.2008.925298.

[3] M. Morelli and U. Mengali, “Channel estimation for adaptive frequency-domain equalization,” IEEE Trans. Wireless Commun., 2005, doi: 10.1109/TWC.2005.853896.

[4] F. Pancaldi, G. M. Vitetta, R. Kalbasi, N. Al-Dhahir, M. Uysal, and H. Mheidat, “Single-carrier frequency domain equalization: A focus on wireless applications,” IEEE Signal Process. Mag., vol. 25, no. 5, pp. 37–56, Sep. 2008, doi: 10.1109/MSP.2008.926657.

[5] H. Liu and P. Schniter, “Iterative frequency-domain channel estimation and equalization for single-carrier transmissions without cyclic prefix,” IEEE Trans. Wireless Commun., vol. 7, no. 10, pp. 3686–3697, Oct. 2008.

[6] J. Coon, S. Armour, M. Beach, and J. McGeehan, “Adaptive frequency-domain equalization for single-carrier multiple-input multiple-output wireless transmissions,” IEEE Trans. Signal Process., vol. 53, no. 8, pp. 3247–3256, Aug. 2005, doi: 10.1109/TSP.2005.851155.

[7] M.-E. Grandmaison, J. Belzile, C. Thibeault, and F. Gagnon, “Reconfigurable and efficient FFT/IFFT architecture,” in Proc. Canadian Conf. Electrical and Computer Engineering, 2004, pp. 1115–1118, doi: 10.1109/CCECE.2004.1345315.

[8] S. A. Dawwd and S. M. Nori, “Reduced area and low power implementation of FFT/IFFT processor,” Int. J. Electrical and Electronic Engineering, vol. 14, no. 2, 2018, doi: 10.37917/ijeee.14.2.3.

[9] C. Ingemarsson and O. Gustafsson, “SFF—The single-stream FPGA-optimized feedforward FFT hardware architecture,” J. Signal Process. Syst., vol. 90, pp. 1583–1592, 2018.

International Journal of
AI Electronics and Nexus Energy
Peer Reviewed, Referred & Indexed Journal
ISSN: 3070-0515 www.zesterapublications.com Original Research Paper

[10] J. W. Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex Fourier series," Math. Comput., vol. 19, no. 90, pp. 297–301, 1965.
[11] A. V. Oppenheim and R. W. Schafer, Discrete-Time Signal Processing, 3rd ed. Upper Saddle River, NJ, USA: Prentice Hall, 2010.
[12] S. Haykin, Adaptive Filter Theory, 5th ed. Boston, MA, USA: Pearson, 2014.
[13] A. Goldsmith, Wireless Communications. Cambridge, U.K.: Cambridge Univ. Press, 2005.
[14] S. Weinstein and P. Ebert, "Data transmission by frequency-division multiplexing using the discrete Fourier transform," IEEE Trans. Commun. Technol., vol. 19, no. 5, pp. 628–634, Oct. 1971.
[15] B. Farhang-Boroujeny, Signal Processing Techniques for Software Radios, 2nd ed. Bloomington, IN, USA: self-published/author edition, 2010.
[16] J. G. Proakis and M. Salehi, Digital Communications, 5th ed. New York, NY, USA: McGraw-Hill, 2008.
[17] S. U. H. Qureshi, "Adaptive equalization," Proc. IEEE, vol. 73, no. 9, pp. 1349–1387, Sep. 1985.
[18] B. Widrow and S. D. Stearns, Adaptive Signal Processing. Englewood Cliffs, NJ, USA: Prentice-Hall, 1985.
[19] E. H. Wold and A. M. Despain, "Pipeline and parallel-pipeline FFT processors for VLSI implementations," IEEE Trans. Comput., vol. C-33, no. 5, pp. 414–426, May 1984.
[20] K. K. Parhi, VLSI Digital Signal Processing Systems:Design and Implementation. New York, NY, USA: Wiley, 1999.