ISSN 2278-3091
Volume 8, No.3, May - June 2019
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse87832019.pdf
https://doi.org/10.30534/ijatcse/2019/87832019

Area and Power Efficient Pipeline FFT Architecture for QPSK-OFDM

G Ganesh Kumar[1], Sibghatullah I Khan[2], M Mahaboob Basha[3]
[1,2,3]Dept. of ECE, Sreenidhi Institute of Science & Technology, Hyderabad 501301, India.
{ganeshkumarg, sibghatullahk, mahaboobbashamg}@sreenidhi.edu.in

ABSTRACT

This paper proposes four-parallel pipelined fast Fourier transform (FFT) architecture for the discrete Fourier transform (DFT) computation of quadrature-phase-shift-keying Orthogonal Frequency Division Multiplexing (QPSK-OFDM) signals. The first stage of the architecture is optimized with the multiplexer blocks that contain output values of QPSK as inputs. Furthermore, the proposed architecture employs a complex constant multipliers (CCMs) using adders and shifters that can reduce the total hardware of the design. The proposed 64-point pipeline architecture is modeled and implemented using UMC 65 nm CMOS technology with a supply voltage of 1.2 V. The results demonstrate that the proposed FFT architecture significantly reduces the hardware cost and power consumption in comparison to the recent FFT architecture.

Key words: fast Fourier transform (FFT), QPSK-OFDM, four- parallel, radix-2^2.

1. INTRODUCTION

In the recent years, the wired/wireless communication systems have to support multimedia transmission such as high-quality voice, video, etc. Therefore, there is a rapidly growing demand for high speed, efficient and reliable communication technology. Orthogonal Frequency Division Multiplexing (OFDM) is a form of multi-carrier modulation technology, which is an effective modulation scheme to meet the demand.

The key cores in the OFDM system are Fast Fourier Transform (FFT) and Inverse Fast Fourier Transform (IFFT) [1]. In conventional OFDM system, IFFT is used for trans-formation of signals at transmission section and FFT at the reception section. The implementation of an FFT processor is one of the most difficult parts in the realization of OFDM systems and its hardware complexity is very high. Therefore, many FFT algorithms and architectures are evolved in order to reduce hardware complexity with high speed.

In [2], the authors have presented the radix-2^k feedfor- ward FFT architectures. The designs in [2], include radix-2^2,

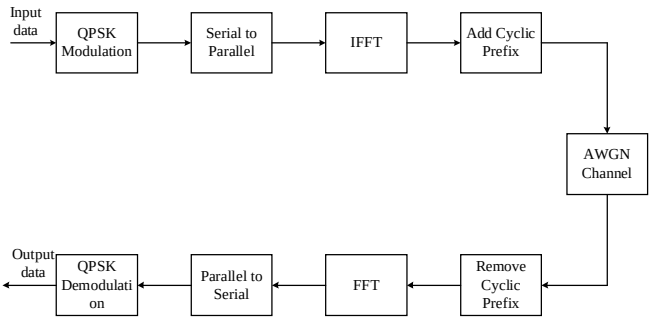

Figure 1: OFDM System block diagram

radix-2^3 and radix-2^4 architectures. It showed that radix-2^k can be used for any number of parallel samples which is a power of two. The proposed design is based on radix-2^2 feedforward FFT architecture with 4-parallel input, but the required stages to compute is reduced to half. Further, the proposed designed is optimized for OFDM communication which uses QPSK modulation.

The rest of this paper is as follows. In Section 2, the transmission and reception of quadrature-phase-shift-keying OFDM (QPSK-OFDM) signal [3-7] is demonstrated. Section 3 presents, the proposed four parallel input radix-2^2 feedfor-ward FFT architecture that is optimized for QPSK modulation. In section 4, the proposed design is compared to the previous architecture with experimental results. Finally, Section 5 concludes this paper.

2. OFDM SYSTEM

The basic block diagram of transmission and reception of an OFDM system is shown in Figure 1. The input data is mapped into the complex plane using QPSK modulation technique. The mapped data is transformed from serial format to parallel format and IFFT is applied on the parallel data to generate OFDM symbols. For the converted data, cyclic prefix is added. The transmitted data is impaired by the channel. The receiver functions to receive the data and performs the inverse operations on it.

QPSK is one of the most popular digital modulation techniques used for Satellite communication, and sending data

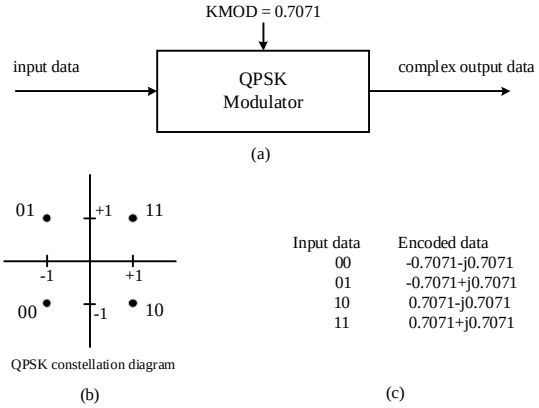

Figure 2: QPSK modulator (a) block diagram (b) Constellation diagram (c) encoded data for the input

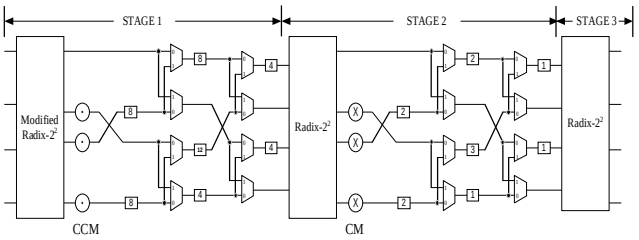

Figure 3: 64-point radix-2^2 FFT architecture optimized for QPSK modulation

over cable networks. Its popularity comes from both its easy implementation and resilience to noise. The diagram of a QPSK modulation is shown in Figure 2. The implementation of QPSK involves changing the phase of the transmitted waveform. Each finite phase change represents unique digital data. A phase modulated waveform can be generated by using the digital data to change the phase of a signal while its frequency and amplitude stay constant. A QPSK modulated carrier undergoes four distinct changes in phase that are represented as symbols which is shown in constellation diagram of Figure 2(b).

For QPSK, input bit stream need to break up to two-two bits and later these two bits are entered simultaneously to the input of the QPSK modulator. As shown in the constellation diagram of Figure 2(b), for the input of 11 the output is (1+ j*1)*KMOD, where KMOD is normalization factor. For most of the systems KMOD is 0.707. The encoded output of 0.707+j*0.707 is called as symbol which represent two binary information digits as shown in Figure 2(c). In QPSK there are four signals that are used to send the four two-bit symbols.

3. PROPOSED FFT ARCHITECTURE

For high data rate, multi-path delay commutator architectures are better. The proposed architecture has been derived using the framework presented in [2]. The design is based on analyzing the flow graph of the FFT and extracting the properties of the algorithm. The proposed 64-point four-parallel pipeline radix-2^2 feedforward FFT architecture is

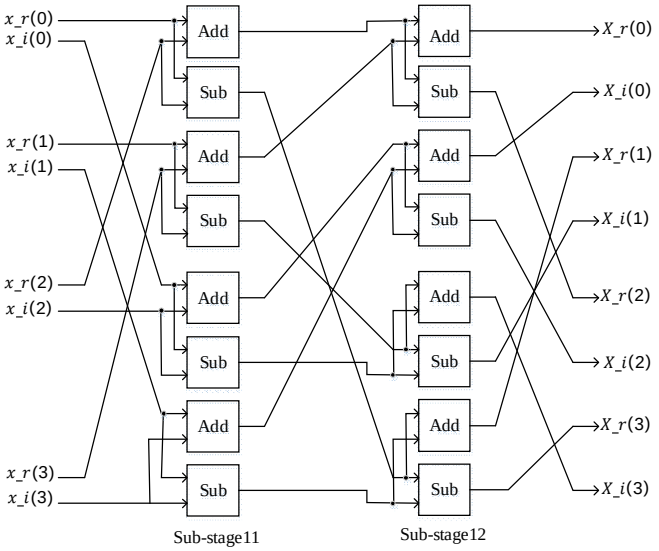

Figure 4: Radix-2^2 butterfly structure

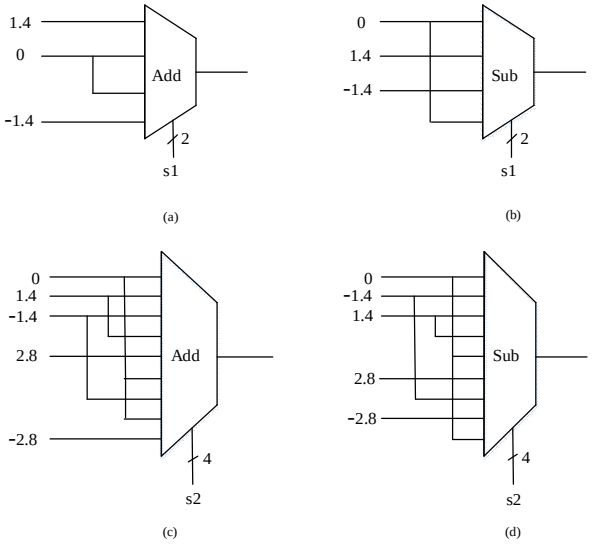

Figure 5: Multiplexers blocks at stage 1 of the architecture (a) adder for substage1, (b) subtractor for substage1 (a) adder for substage2, (b) subtractor for substage2

made up of two stages with radix-2^2 butterflies, one stage with the modified radix-2^2, complex constant multipliers (CCM), complex multipliers (CM) and shuffling structures as shown in Figure 3. The architecture processes four samples in parallel in a continuous flow. Each stage in Figure 3 requires a shuffling structure, which consist of buffers and multiplexers for storing input data and timely distributing them in a correct order for processing by the complex adder/subtractor. The buffer length is indicated by a number.

1) Radix-2^2 butterfly structure: The butterfly of a radix-2^2 structure consists of four complex inputs and four complex

outputs as shown in Figure 4. This butterfly structure performs complex addition and complex subtraction with the four complex inputs. The multiplication by *-j* is obtained by simply swapping the inputs. For a 4-point FFT, it requires only 16 real
additions/subtractions without any multipliers.

Table 1: Comparison of various N-point FFT Architectures

Radix	Complex Multipliers	Complex Adders
Radix-4 [5]	$3(\log_4 N - 1)$	$8 \log_4 N$
Radix-2^2 [2]	$3(\log_4 N - 1)$	$8 \log_4 N$
Radix-2^3 [2]	$4(\log_4 N - 1)$	$8 \log_4 N$
Radix-2^4 [2]	$3.5 \log_4 N - 4$	$8 \log_4 N$
Proposed Radix-2^2	$3(\log_4 N - 2) + 1.2$	$8(\log_4 N - 1)$

Table 2: Comparison of 64-point FFT Architectures

Design	Word Width	Area (μm^2)	Technology	Power	Normalized Power
[2]	16	3349.41	65nm,1.2 V	2.345 mW @20 MHz	1.27
Proposed	16	1948.35	65nm,1.2 V	1.089 mW @20 MHz	0.59

2) Modified Radix-2^{2s}: In the proposed FFT architecture the first stage is optimized with respect to the QPSK outputs. The QPSK modulator outputs are 0.707+j*0.707, 0.707-j*0.707,
-0.707 + j0.707 and -0.707-j*0.707. As the real or imaginary output is 0.7 or 0.7, these are coded as 0 and 1 respectively. In the first stage of Figure 3, radix-2^2 butterfly block is modified by using multiplexer blocks instead of adders/subtractors. At substage1 of Figure 4, the adder block is replaced with the multiplexer as shown in Figure 5(a). This multiplexer selects 0, 1.4 and -1.4 because these are the values of addition combination of 0.7 and -0.7. Similarly, the subtraction block also replaced with the multiplexer as shown in Figure 5(b), in substage1. Thus, the substage1 replaced the addition/subtraction blocks with the 4-input multiplexers.

The outputs of substage1 is given to the inputs of substage2, this also replaces the addition and subtraction blocks with the 8-input multiplexers as shown in Figure 5(c) and (d) respectively. The outputs of the substage2 can be 0, 1.4, -1.4, 2.4 or -2.4, which need to be multiplied by the complex multiplier. However, a complex multiplier is not required because 1.4 and 2.4 can be multiplied with a number by shifting operation so a constant complex multiplier [3] can be used. Three CCMs are employed in stage 1 and three complex multipliers are employed in stage 2 of Figure 3. The CCMs and CMs are represented as $\odot$ and $\otimes$ respectively. The hardware complexity of the design is reduced due to the modified radix-2^2 block and CCMs.

4. COMPARISON AND ANALYSIS

Table 1 compares the proposed structure to other efficient pipelined architectures for the computation of an N-point FFT. The comparison is in terms of complex multipliers, and complex adders. Here, the multiplicative complexity ratio of one complex multiplier and one complex constant multiplier is considered as 1 and 0.4 respectively. From Table 1, it can be observed that the proposed FFT require the lowest numbers of multipliers and adders among all the designs in the literature.

The proposed 64-point FFT architecture with a word- length of 16-bit is modeled by MATLAB Simulink. After verifying the proposed FFT architecture, the model was converted into Verilog by HDL coder and functionally verified using Synopsys Verilog Compiler simulator. Then the proposed model is synthesized using Synopsys Design Compiler with UMC 65 nm, 1.2 V Standard Cell Library. The same implementation is done for the architecture in [2].

The performance [4] comparison of the proposed design with the recent 64-point FFT architecture in [2], is summarized in Table 2. To have meaningful comparison among the implementations, the total area and normalized power irrespective of frequency and FFT size were considered as comparison parameters. As the proposed design replaced the adder/subtractor blocks of first stage with multiplexers, it takes less area compared to the architecture proposed in [2]. The normalized power per FFT [1] is considered as follows:

$$\text{Normalized power per FFT} = \frac{\text{Power}}{(\text{Voltage})^2 \times (\text{FFT size}) \times \text{Frequency}} \times 1000 \quad (1)$$

With respect to normalized power under the same process technology, the power consumption of the proposed implementation is about 2.15 times lower than the design in [2]. These experiment results show that the proposed design takes less area and also consumes less power compared to the architecture in [2].

5. CONCLUSION

In this paper, we have proposed a hardware and power efficient 64-point four-parallel pipelined FFT architecture for QPSK-OFDM. The hardware cost of the architecture is reduced by employing the modified radix-2^2 using multi-plexers in the first stage. Furthermore, by employing the CCMs in the first stage, the total hardware cost of the design is further reduced. The results show that at 20 MHz frequency the proposed architecture dissipates less power compared to the architecture in [2]. As the proposed architecture is cost-effective with low power consumption, it can be useful for QPSK-OFDM.

REFERENCES

1. C. Yu, M.-H. Yen, P.-A. Hsiung, and S.-J. Chen, **A low-power 64-point pipeline FFT/IFFT processor for OFDM applications,** *IEEE Transactions on Consumer Electronics*, vol. 57, no. 1, pp. 40–40, 2011. https://doi.org/10.1109/TCE.2011.5735479
2. M. Garrido, J. Grajal, M. Sa´nchez, and O. Gustafsson, **Pipelined radix-feedforward FFT architectures**, *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 21, no. 1, pp. 23–32, 2013. https://doi.org/10.1109/TVLSI.2011.2178275
3. S.-I. Cho and K.-M. Kang, **A low-complexity 128-point mixed-radix FFT processor for MB-OFDM UWB systems**, *ETRI journal*, vol. 32, no. 1, pp. 1–10, 2010. https://doi.org/10.4218/etrij.10.0109.0232
4. M. Frigo and S. G. Johnson, **Fftw: An adaptive software architecture for the fft**, *in Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 3. 1998, pp. 1381–1384.
5. L. Jia, Y. Gao, J. Isoaho, and H. Tenhunen, **A new vlsi-oriented fft algorithm and implementation**, *in Proceedings of Eleventh Annual IEEE International ASIC Conference*, Sep 1998, pp. 337–341.
6. P Soundaryamala, D Gowri Sankar, S.Ramakrishna, STP Radika, **Initializing key for High Level Transformation for FIR filters**, *International Journal of Advanced Trends in Computer Science and Engineering*, Volume 8, No.1, January – February 2019.
7. Lavanya Maddisetti1 , Ranjan K. Senapati2 , JVR Ravindra, **Training Neural Network as Approximate 4:2 Compressor applying Machine Learning Algorithms for Accuracy Comparison**, *International Journal of Advanced Trends in Computer Science and Engineering*, Volume 8, No.2, March - April 2019. https://doi.org/10.30534/ijatcse/2019/17822019