ISSN (Print) : 0974-6846
Indian Journal of Science and Technology, *Vol 9(12), DOI: 10.17485/ijst/2016/v9i12/73128, March 2016* ISSN (Online) : 0974-5645

Efficient Implementation of Radix-4 Single path Delay Feedback (SDF) FFT Processors

K . Hanumantha Rao[1*] and C. Kumar Charlie Paul[2]

[1]St. Peter's University, Chennai - 600054, Tamil Nadu, India; hanumantharaophd.vlsi@gmail.com
[2]A. S. L. Pauls College of Engineering and Technology, Coimbatore – 641032, Tamil Nadu, India; charliepaul1970@gmail.com

Abstract

The aim of the current research work is to improve the digital architecture (Radix-4 Single path Delay Feedback) of frequency transformation process. Radix-4 structure reduces the number of steps to half than Radix-2 structure. In the proposed Radix-4 SDF FFT architecture, sequential structures are designed with the help of flip-flops. Traditional FFT structure has complex multiplier architecture to perform the twiddle factor multiplication. But it is not sufficient to large point FFT calculation. Hence, to overcome this problem, Bit Parallel Multiplication (BPM) has been introduced in this paper. In addition, data flow complexity of BPM structures have been realized and re-designed by reducing the most complexity paths. Proposed BPM based twiddle factor multiplier offers 17.5% improvements in area utilization and 7.22% improvements in latency for the value 0.707 and 15.3% improvements in area utilization and 5.97% improvements in latency for the value 0.9238. Proposed Radix-4 SDF FFT offers 47.96% improvements of total latency in Virtex E and 28.02% improvements of total latency in Virtex 6 than traditional Radix-2 SDF FFT. Power consumption of Proposed Radix-4 SDF FFT offers 69.69% improvements in Virtex E and 14.95% improvements in Virtex 6 than traditional Radix-2 SDF FFT. In future, proposed BPM based twiddle factor multiplier will be useful in Orthogonal Frequency Division Multiplexing (OFDM) and Software Defined Radio (SDR) for performing data communication process efficiently.

Keywords: Advanced Constant Multiplier, Bit Parallel Multiplication, Radix-4 Single path Delay Feedback Fast Fourier Transformation, Reconfigurable Complex Multiplier, Very Large Scale Integration

1. Introduction

The Fast Fourier Transformation (FFT) is the most crucial blocks in multi-carrier systems like Orthogonal Frequency Division Multiplexing (OFDM). In multi-carrier system, single high rate data stream is down streamed into number of multiple lower rate data streams. Frequency responses of signals are required to transmit the lower rate data streams into common channel of OFDM. FFT plays an important role in conversion of time domain signal into frequency domain signal. Butterfly structures are used in FFT for converting time domain signal into frequency domain signal. A great result has been published for Fast Fourier Transformation in 1979[1]. The most salient research wok challenges in this area include end-to-end data transfer, conversion of time to frequency, frequency to time and elaborating support for real-time multimedia streaming. Radix-2 FFT structures provide more complexity due to large computational path. The great solution will be provided in 1983[7] for reducing the large computational path. Common butterfly structures have been used in[7] instead of using independent butterfly structures for each and every input point. Despite the advantage of parallel FFT, asynchronous effect of intermediate input/output disturb the performances of parallel FFT in terms of speed. Hence, in addition with[7], pipelined structures are introduced in[8] for improving the speed of FFT processors.

Complex multiplier is used in the FFT processors to perform twiddle factor multiplication. Twiddle factor

* *Author for correspondence*

values varies from 0 to 1 for FFT calculation, therefore, fractional multiplication is to be involved in the complex multiplier. Fused floating point multiplication has been introduced in[12]. But floating point multipliers are not sufficient for improving the performance of FFT processors. To perform the complex multiplication, residue arithmetic based complex multiplier has been introduced in[12]. Residue Number System (RNS) is the best solution to perform the fractional multiplication in FFT processors. However, a number of steps have been taken out to take residues of every input point in RNS based complex multiplication. To reduce this problem, Advanced Constant Multiplier (ACM) has been proposed in 2008[7]. ACM based multiplier is better than RNS based multiplier, however switching activities of ACM based multiplier reduces the performance of FFT processors, because no control signals provided in ACM based multiplier. Hence, to reduce this problem, Reconfigurable Complex Multiplier (RCM) has been introduced in 2011[8]. A control signal is provided in RCM based complex multiplier to select the proper twiddle factor multiplier at appropriate time. In addition to reconfigurable complex multiplication, Bit Parallel Multiplier (BPM) has also introduced in 2011[8]. In Bit Parallel Multiplier, shifter and adder are involved to perform the fractional multiplication.

BPM utilize less hardware and timing consumption rather than ACM and RCM based twiddle factor multiplication[8]. Further pass logic based trivial multiplications are introduced in[1]. Almost BPM and pass logic based trivial multiplications offers same performances rather than other complex multiplications. Hence, from above consecutions, it is clear that the performance of FFT processors is based on the performance complex multipliers. In addition, Radix-4 multiplier reduces the operation sequence by half for estimating the frequency points of longer data. Radix-4 FFT also has asynchronous effect like radix-2 FFT. A great solution for this problem is pipelined based structure which is illustrated in[4]. Further the design of radix 2^2 feed-forward FFT is demonstrated in[3]. Meanwhile, BPM and shift and add based multiplier provides best fractional multiplication results rather than using fractional multiplications. Based on reducing the number of shifter and adder in BPM based multiplier structure, we can further improve the performances of FFT processors.

In this paper, we propose a structure of enhanced Radix-4 Single path Delay Feedback (R4SDF) FFT to facilitate opportunistic data forwarding path in OFDM. The R4SDF FFT has two advantages based on their architectures. One of them is reducing the complexity path to half then R2SDF FFT and another one of them is reducing the number of twiddle factor multiplication than R2SDF FFT. In our proposed work, structure of complex multiplier is minimized to reduce the hardware utilization and power consumption of complex multiplication. Structure of BPM has been realized in this work for optimizing the architecture of complex multiplier. Further, realized BPM based twiddle factor multipliers are incorporated into Radix-4 SDF FFT processors. Hence, proposed R4SDF FFT processor named as Enhanced R4SDF FFT. The design of enhanced Radix-4 SDF FFT has been implemented in Very Large Scale Integration (VLSI) System design environment. Results of enhanced R4SDF FFT have been compared with traditional R4SDF FFT. Results of traditional R2SDF FFT and enhanced R4SDF FFT have been analyzed in multiple chips like Spartan 3, Virtex E and Virtex 6 Low power from same Field Programmable Gate Array (FPGA).

2. Related Work

FFT processors are frequently used for finding the frequency response of the original discrete time signals. Advantages of FFT processors are measured in two different types of methodology. One of them is finding the redundant operation to minimize the hardware structure of FFT model and another one of them is finding logic calculation to realize the original structure of FFT models. In the past, Discrete Fourier Transformation (DFT) processors are used for converting time to frequency process. But DFT has more computational complexity than FFT processors. The computational complexity of DFT processors are denoted as O (n^2). But FFT processors have complexity of O (log N). In^2, architecture of FFT algorithm can be implemented with the help of butterfly structures. FFT algorithm save 30% to 50% multipliers on average and improve the average processing speed by a factor of 2^2. Low power consumption, high speed and less area utilization are the main factors in VLSI System design environment. Hence, aim of FFT design is to be reducing the number of utilized slices, Look up Table (LUT), delay and power consumption of FFT processors. To reduce the power consumption of FFT processors o^2 DFT processors have been designed in[11]. This architecture demonstrates reduction of redundancy mathematical

processes. However, processing time of o^2 DFT has larger and independent data access nodes are required to transmit the data. Next to o^2 DFT, parallel FFT structures have been implemented on VLSI design environment in[12]. The parallel FFT processors are based on vector rotate algorithm which improves the structure of FFT in terms of high speed and low power. Even though providing high speed, more number of clock cycles (for instance, 250ns for 8-point FFT) is required for implementing the parallel FFT processors. To minimize the number of clock cycles used, parallel pipelined FFT processors have been introduced in[6]. The parallel pipelined FFT processor has pipelining registers for reducing the asynchronous effect of input and output of intermediate blocks.

Complex multiplication of FFT processors plays an important role for improving the performances of FFT processors. For N-point FFT computation, N/2 twiddle factor has involved to convert time domain signal into frequency domain signal. As already discussed twiddle factor values of FFT varies from 0 to 1. Hence, it is essential the fractional multiplier. In[12], fused fractional multiplier has been introduced based on FFT. This fused fraction multiplier consist of all twiddle factor multiplication involved in FFT calculation. However, twiddle factor multiplication requires ROM memory to store the certain kinds of values ranging from 0 to 1. In order to reduce this problem RNS based multiplications has been designed in[13]. RNS based complex multiplier improves the performances in terms of hardware complexity and speed of the processors. Further to improve the architecture of complex multiplier, ACM based complex multiplier has been introduced in[5]. Although, every twiddle factor multiplication can be performed in every clock cycles of operation. To select the appropriate twiddle factor multiplier with proper clock cycles, Reconfigurable Complex Multiplier (RCM) based multiplier is introduced in[16]. RCM is also named as ROM-less architecture, because it removes the necessary of ROM for finding the twiddle factor multiplications. This design consumes about 33.6K gates and 9.8mW power at 20MHz. In addition to RCM, Bit Parallel Multiplier (BPM) has been designed in[16]. For every twiddle factor values, separate twiddle factor multiplier has been involved. The BPM multiplier is based on shifting and accumulation operation. Hence, based on reducing the shifter and adder, we can improve the architecture of complex multiplier. In[8], 64 point FFT design has been proposed with the help of shift and add based complex multiplier. Within 23 clock cycles, the operation of 64-point FFT has been done in[8]. The average dynamic power consumption of this FFT processor is 41mW at 20MHz. Further, pass logic trivial based multiplications have been introduced in[1]. The pass logics have been used in[1] for the design of Decimation in Frequency (DIF) FFT. It offers 9% reduction of slices, 6.5% reduction of power consumption when implemented in FPGA board. On the other hand, 28% reduction of power consumption and 27% reduction of hardware utilization, when implemented in Application Specific Integrated Circuits (ASIC). Based on Single Flux Quantum (SFQ), FFT design has been done in[5].

Radix-4 FFT structures has reduces the computational path of FFT processors by half, when compared to Radix-2 FFT structures. The architecture or real Radix-4 FFT processor by reducing its operation sequence has been implemented in[4]. It results in saving the power around 23% and 9% respectively for the 16-point and 64-point radix-4 FFT. Further to improve the architecture of Radix-4 FFT structure, Radix-2^2 Feed forward FFT has been designed in[13]. Based on structure of FFT in terms of Radix-2 and Radix-4, dual mode FFT has been designed in[14]. Finally, combined Single path Delay Commutator (SDC) – Single path Delay Feedback (SDF) FFT for reducing the computational path of FFT calculation. In this article realization of BPM based complex multiplier is introduced to every twiddle factor values. Further, realized BPM based complex multiplier is integrated into Radix-4 SDF FFT and Radix-2 FFT to improve the frequency transformation techniques.

3. Design of Radix-2 FFT

The Radix-2 algorithm is used for finding frequency response of original discrete time domain signals. Discrete Fourier Transformation (DFT) estimates the frequency spectrum of discrete time signals. The generalized N-point DFT is mathematically represented as in Equation (1):

$$Z[k]=\sum_{i=0}^{N-1} z(n) e^{-ji2\pi\frac{k}{N}}, \quad (1)$$

Where k = 0, 1.... N-1.

Where, z (n) is the discrete time input signal, Z [k] is the frequency response of z (n) or Fourier transformation of input z (n). The twiddle factor or angle rotation factor indicated as $W_n = e^{-j2\frac{\pi}{N}}$. As already discussed twiddle factor varies from 0 to 1. Hence, complex multiplier is required for all stages (0 to N-1) to multiply input signal with twiddle factor values. Hence, DFT function become

as $Z[k]=\sum_{i=0}^{N-1} z(n)W_N^{nk}$. Similarly, timing representation of frequency oriented signals is determined by using Inverse Discrete Fourier Transformation (IDFT) technique, which is the reverse process of DFT process. The generalized IDFT is mathematically represented as in Equation (2):

$$z(n)=\frac{1}{N}\sum_{i=0}^{N-1} Z[k]e^{ji2\pi i\frac{k}{N}} \quad (2)$$

Where i = 0, 1, 2, 3....N-1.

$$Z[2i]=\sum_{m=0}^{\frac{N}{2}-1}\left(z[m]+z\left[m+\frac{N}{2}\right]\right)W_{N/2}^{mm}$$

Complexity of DFT and IDFT processors is founded as O (N^2). Transformation period is long for finding the transformation process either time to frequency or frequency to time. Therefore, Cooley-Tukey suggests the Fast Fourier Transformation (FFT) algorithm to frequency transformation techniques. It reduces the complexity O (log N) from O (N^2). If follows the Radix-2 algorithm to determine the transformation points. FFT can be classified as two types as (1) Decimation in Time (DIT) FFT and (2) Decimation in Frequency (DIF) FFT. DIF FFT is used to finding the frequency spectrum of original discrete time input signals. On the other hand DIT FFT is the process of reconstructing original information signals. In this article, we take DIF FFT to exhibit the frequency transformation technique.

3.1 Radix-2 Algorithm

The Radix-2 algorithm is a special case in finding discrete response of frequency or time from discrete time or frequency signals respectively. Equation (1) can be divided into two parts as even and odd streams as in Equation (3):

$$Z[k]=\sum_{i=0}^{\frac{N}{2}-1} z(2i)W_N^{2ik}+\sum_{i=0}^{\frac{N}{2}-1} z(2i+1)W_N^{(2i+1)k} \quad (3)$$

Even stream of Equation (3) can be written as:

$$Z[2i]=\sum_{m=0}^{\frac{N}{2}-1} z(m)W_i^{2in}+\sum_{m=\frac{N}{2}}^{N-1} z(m)W_i^{2in}$$

where i = 0, 1, 2, 3....N/2-1.

$$Z[2i]=\sum_{m=0}^{\frac{N}{2}-1} z(m)W_n^{2in}+\sum_{m=0}^{\frac{N}{2}-1} x\left(m+\frac{N}{2}\right)W_N^{2i\left(m+\frac{N}{2}\right)} \quad (5)$$

Similarly, odd samples of Equation (3)
Can be derived as follows:

$$Z[2i]=\sum_{m=0}^{\frac{N}{2}-1} z(m)W_n^{2in}+\sum_{m=0}^{\frac{N}{2}-1} x\left(m+\frac{N}{2}\right)W_N^{2i\left(m+\frac{N}{2}\right)}$$

$$Z[2i+1]=\sum_{i=0}^{\frac{N}{2}-1}\left(Z[m]-z\left[m+\frac{N}{2}\right]\right)W_n^{mn}W_{N/2}^{in} \quad (5)$$

Butterfly structures can be formed from Equations (4) and (5). With the help of Equations (4) and (5), we can easily design the Radix-2 FFT structures for N-point FFT. For instance, Radix-2 8 point FFT is illustrated in Figure 1 which uses the Radix-2 algorithm to find the frequency response of time domained signals.

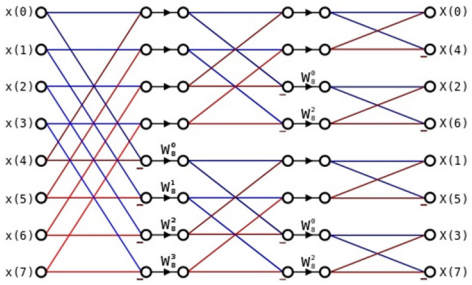

Figure 1. Radix-2 8 point DIF FFT.

Computation of N-point FFT takes longer time to fine the frequency points. As shown in Figure 1, in every stage of FFT processors, first half of the input points have processed with other half of the input points. In order to overcome this problem, Radix-2 SDF FFT has been used.

3.2 Radix-2 Single path Delay Feedback (R2SDF) FFT

SDF architecture is a parallel structure, because single butterfly structures shares the process of N-input points. For instance, 8 point Radix-2 FFT has three stages to produce the frequency analysis of input points. Therefore it requires only three butterfly structures to find the results. The block diagram of 8 point Radix-2 SDF FFT structure is illustrated in Figure 2.

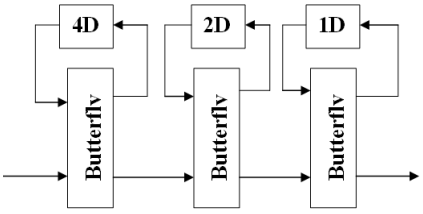

Figure 2. Structure of 8 point Radix-2 Single path Delay Feedback (R2SDF) FFT.

In each stage, N/2 number of delay elements has been used for processing first half of input with second half of input. Hence, this architecture called as Radix-2 Single path Delay Feedback (R2SDF) FFT. When

compared to normal radix-2 FFT, entire complexity has been reduced in R2SDF FFT. In[16], Reconfigurable Complex Multiplier (RCM) has been used for performing complex multiplication. In R2SDF FFT, only complex multiplier consumes more hardware complexity and power consumption while remaining part consumes only considerable number of silicon slices to design. One of the straight forward approaches of R2SDF FFT algorithm is as follows:

- The input data sequence is divided into two parallel data stream.
- Each stage of this architecture, half of the data streams is delayed via register and processed with second half of the data streams.
- For 8 point R2SDF FFT, delay for three stages is 4, 2 and 1 respectively. The total number of delay elements is 4+2+1 = 7 for performing 8 point DIF R2SDF FFT.
- In each stage, Reconfigurable Complex Multiplier is involved to perform twiddle factor multiplication.

Due to using less number of delay elements, speed of R2SDF FFT processors must be high. Similarly, less hardware utilization and power consumption has been obtained in case of R2SDF FFT processors than normal Radix-2 FFT processors. However, there are availability to reduce further hardware utilization and power consumption of FFT processors with the help of commutator structures.

3.3 Radix-2 Multipath Delay Commutator (R2MDC) FFT

MDC architecture is based on pipelining technique. In other words, it follows "stream-like" processing of block based algorithm. This architecture helps to reduce the required number of silicon slices, LUTs and registers of FFT processors. Commutator structure of MDC structure gives the solution for overcoming the problem of R2SDF FFT. Architecture of 8 point R2MDC FFT is illustrated in Figure 3.

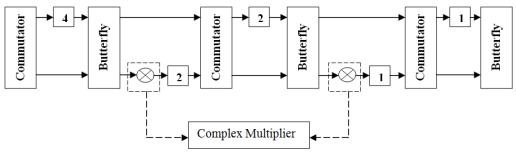

Figure 3. Structure of 8 point Radix-2 Multipath Delay Commutator (R2MDC) FFT.

In every stage of R2MDC FFT structure, single butterfly unit and commutator unit are integrated with multiple delay elements. Due to providing parallelism, hardware utilization of R2MDC FFT is dramatically reduced. Even though, speed of the R2MDC FFT is low due to utilizing multiple delay elements for computing the frequency transformation of original input discrete signals.

The pipeline technique of R2MDC FFT utilizes 50% hardware resources. The butterfly structures of MDC structures are idle half of the time period until it can form new pair of inputs. All hardware components operate at same input frequency, termed as nyquist frequency. Like R2SDF FFT structure, R2MDC FFT structure has Reconfigurable Complex Multiplier for performing twiddle factor multiplication.

Hence, R2SDF FFT and R2MDC FFT processors have different types of advantages in terms of VLSI main concerns like utilization of hardware, delay consumption and power consumption. Hence, Radix-2 algorithm has good performance than normal Radix-2 algorithm. In each stage of Radix-2 algorithm, input points are broken into two half and then each half can be processed in a parallel manner. From this consecution, it is clear that the performances of FFT processors have been increased, if dividing the each stage of input into more than two half. For instance, in Radix-4 FFT, each stage of input is divided into four half of the inputs and then processed in a parallel manner. In this article, Radix-4 FFT is preferred for improving the performance of FFT processors.

4. Design of Radix-4 FFT

The Radix-4 FFT is an advanced technique for performing Fourier transformation of input discrete signals. As in Radix-2 algorithm, Radix-4 algorithm has both DIT FFT and DIF FFT. In Radix-4 FFT, four numbers of input points can be processed at the same time rather than processing two input points at the same time in Radix-2 FFT. Radix-4 FFT is only suitable for 2^{4n} input points. Minimum number of input points in Radix-4 FFT is four. Hence, it is possible to reduce the half of the complex multiplier in each stage. For 16 point FFT computation, Radix-4 FFT requires only 2 stages rather four stages in Radix-2 FFT. When compared to Radix-2 FFT, Radix-4 FFT takes advantages in following symmetries:

$$W_N^{nk+N/4} = -W_N^{nk+3N/4} = -jW_N^{nk} \quad (6)$$

This equation leads to a reduction of the number of multipliers (multiplication by j can be performed without multiplier). Radix-4 DIF FFT is preferred in this article. For 4 point Radix-4 DIF FFT, four point output obtained as follows:

$$X[4m] = \sum_{n=0}^{N/2-1} \left(x[n] + x[n+N/4] + x[n+N/2] + x[n+3N/4]\right) W_{N/4}^{mn} \quad (6)$$

$$X[4m+1] = \sum_{n=0}^{N/2-1} \left(x[n] - jx[n+N/4] - x[n+N/2] + jx[n+3N/4]\right) W_{N/4}^{mn} W_N^{n} \quad (7)$$

$$X[4m+2] = \sum_{n=0}^{N/2-1} \left(x[n] - x[n+N/4] + x[n+N/2] - x[n+3N/4]\right) W_{N/4}^{mn} W_N^{2n} \quad (8)$$

$$X[4m+3] = \sum_{n=0}^{N/2-1} \left(x[n] + jx[n+N/4] - x[n+N/2] - jx[n+3N/4]\right) W_{N/4}^{mn} W_N^{3n} \quad (9)$$

Where X [4m], X [4m+1], X [4m+2], X [4m+3] are the Fourier transformation of input x [n]. Hence, 4 point Radix-4 DIF FFT has a single twiddle factor to estimates the frequency points of time domain signals. The structure of 4 point Radix-4 FFT is illustrated in Figure 4.

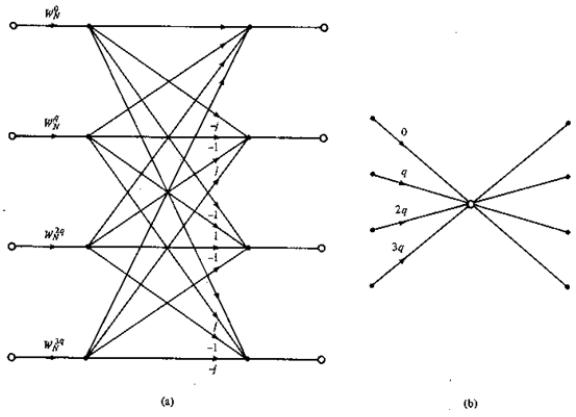

Figure 4. Structure of 4 point Radix-4 DIF FFT.

Similarly, 16 point Radix-4 DIF FFT has two stages for performing FFT computation. It has only nine twiddle factor multiplication in first stage of FFT computation. But in case of 16 point Radix-2 FFT, twelve twiddle factor multiplication involved to produce the same frequency response. Structure of 16 point Radix-4 FFT is illustrated in Figure 5.

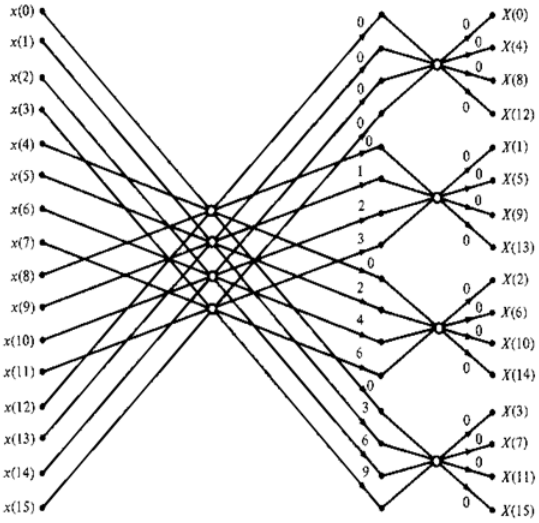

Figure 5. Structure of 16 point Radix-4 DIF FFT.

The values of twiddle factor multiplication required for Radix-2 and Radix-4 FFT is demonstrated in Table 1. It shows the effectiveness of Radix-4 FFT than Radix-2 FFT in terms reduction of number of twiddle factor multiplication. In Radix-2 FFT, there are 15 number of multipliers are required to perform FFT computation. But in case of Radix-4 FFT, only 9 numbers of multipliers are required to perform FFT computation. In this way, Radix-4 FFT reduces the hardware complexity and power consumption for designing a FFT processor. The values of twiddle factor values are demonstrated in Table 2.

Table 1. Twiddle factor multiplier required for Radix-2 and Radix-4 FFT

Radix-2 FFT				Radix-4 FFT	
Stage 1	Stage 2	Stage 3	Stage 4	Stage 1	Stage 2
W_{16}^{0}	W_{16}^{0}	W_{16}^{0}	W_{16}^{0}	W_{16}^{0}	W_{16}^{0}
W_{16}^{1}	W_{16}^{2}	W_{16}^{4}		W_{16}^{1}	
W_{16}^{2}	W_{16}^{4}			W_{16}^{2}	
W_{16}^{3}	W_{16}^{6}			W_{16}^{3}	
W_{16}^{4}				W_{16}^{4}	
W_{16}^{5}				W_{16}^{5}	
W_{16}^{6}				W_{16}^{6}	
W_{16}^{7}				W_{16}^{9}	

Table 2. Twiddle factor values for Radix-2 and Radix-4 FFT

Twiddle Factors	Values
W_{16}^{0}	1
W_{16}^{1}	0.9238-j0.3826
W_{16}^{2}	0.707-j0.707
W_{16}^{3}	0.3826-j0.9238
W_{16}^{4}	-j
W_{16}^{5}	-0.3826-j0.707
W_{16}^{6}	-0.707-j0.707
W_{16}^{7}	-0.9238-j0.3826
W_{16}^{9}	-0.9238+j0.3826

Effective complex multiplier will be required for performing twiddle factor multiplication in Radix-2 and Radix-4 structures. Reconfigurable Complex Multiplier (RCM) is used in large endeavours to perform the twiddle factor multiplication of Radix-4 structures. But in our research work, Bit Parallel Multiplier is used to improve the performances of complex multiplier and FFT processors.

5. Design of Complex Multiplier for Radix-4 FFT

Complex Multiplier performs the twiddle factor multiplication of Radix-4 FFT. In[16] design of Reconfigurable Complex Multiplier has been proposed. The structure of Complex multiplier is illustrated in Figure 6.

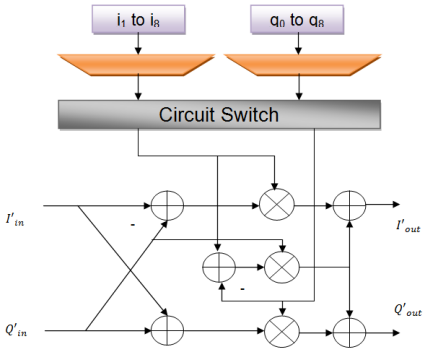

Figure 6. Complex multiplier structure for Radix-4 FFT.

Real input and imaginary input of intermediate data values are collected from Circuit Switch for performing the twiddle factor multiplication. Circuit Switch of complex multiplier will take the decision of selecting appropriate twiddle factor at proper time period. With the help of CLK control signal, design of complex multiplier circuit has been performed successfully. The values of twiddle factors are stored in form of LUTs. Hence, ROM size has been increased in the design of complex multiplier circuit. In order to overcome this problem, we prefer Bit Parallel Multiplier for performing twiddle factor multiplication. It does not require ROM to store the twiddle factor values; instead it uses the shifting and accumulation function to perform the twiddle factor multiplication. For instance, Numerical value 10 is represented in binary form as 1010. While shifting right position to one bit, it will get 0101 (5), i.e., division of two will be obtained. From Table 2, it is clear that only three values of fractional values are included in twiddle factor multiplications such as 0.707, 0.9238 and 0.3826.

Traditional Bit Parallel Multiplication (BPM) structure of 0.707 is illustrated in Figure 7. The multiplication of fractional value 0.707 with some input value is denoted as follows:

(10)

Further it can be solved as:

$$output = in \times \frac{\sqrt{2}}{2} = in \times \left[1 + \left(1 + 2^{-2}\right)\left(2^{-6} - 2^{-2}\right)\right] \quad (11)$$

As in Figure 7, multiplication of twiddle factor value 0.707 requires 4 numbers of shifters and three number of adder to perform multiplication.

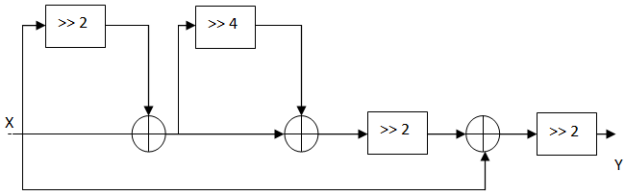

Figure 7. Bit Parallel Multiplication for twiddle factor value 0.707.

Similarly, twiddle factor multiplication of 0.9238 and 0.3826 with constant input x is illustrated in Figure 8 and Figure 9 respectively. Multiplication of twiddle factor value 0.9238 requires 5 numbers of shifters and single adder to perform multiplication. Similarly, 0.3826 require two shifters and single adder to perform multiplication.

In this research work, we can realize these BPM based complex multiplier to further reduce the hardware components. For the twiddle factor multiplication of 0.707, Equation (11) can be further simplified based on taking common factor for reducing the number. Also equation (11) can be written as:

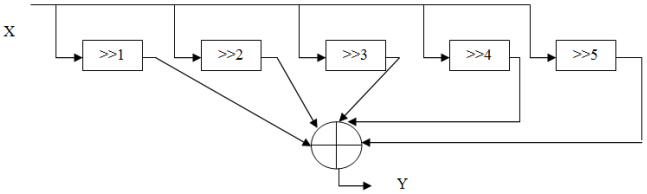

Figure 8. Bit Parallel Multiplication for twiddle factor value 0.9238.

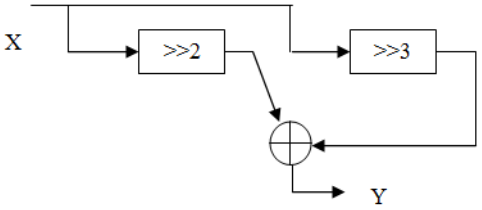

Figure 9. Bit Parallel Multiplication for twiddle factor value 0.3826.

From Equation (12), it is clear that one number of adder and shifter reduced in Equation (12) than Equation (11). The structure of twiddle factor multiplication of 0.707 after reducing the complexity is demonstrated in Figure 10. Similarly, hardware complexity of twiddle factor value 0.9238 can also be reduced by removing the two number of shifter like >>4 and >>5. Because, these two shifters didn't perform any kind of task to perform the multiplication of 0.9238. Proposed BPM for twiddle factor value 0.9238 is illustrated in Figure 11. Twiddle factor value of 0.3826 have minimum number of shifter and adder to perform the multiplication. Hence it doesn't need any simplification.

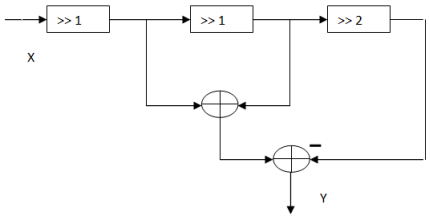

Figure 10. Proposed Bit Parallel Multiplication for twiddle factor value 0.707.

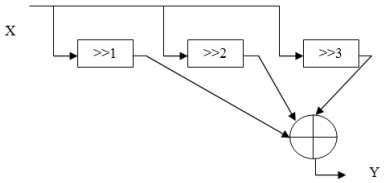

Figure 11. Proposed Bit Parallel Multiplication for twiddle factor value 0.9238.

Because of reducing such complexity, hardware utilization of complex multiplication has been completely reduced effectively. Further, realized twiddle factor values are integrated in to different types of Radix-2 and Radix-4 structures to improve the performances of frequency transformation technique.

6. Proposed Radix-4 Single path Delay Feedback FFT

Like Radix-2 FFT, Radix-4 FFT has also more complexity in computational path. Hence Single path Delay Feedback (SDF) FFT is used in this work to reduce the hardware complexity and power consumption of FFT processors. Unlike, R2SDF FFT, R4SDF FFT has only two stages for computing 16 point FFT and more than one number of delay elements has been used in R4SDF FFT per stage. The structure of 16 point Radix-4 point SDF FFT is illustrated in Figure 12.

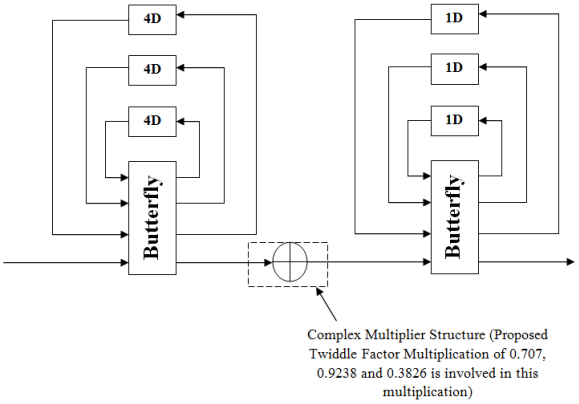

Figure 12. Structure of 16 point Radix-4 SDF FFT.

When compared to other structure of Radix-4 SDF FFT, structure of complex multiplier only consumes more hardware utilization and power consumption. Hence, in order to overcome this problem, proposed twiddle factor multiplication for 0.707 and 0.3826 are integrated in to the complex multiplier of Radix-4 FFT structure.

7. Simulation Results

The proposed realized twiddle factor multiplication, Radix-2 and Radix-4 FFT architecture has been designed through Verilog Hardware Description Language (Verilog HDL). Simulation of Synthesis results has been evaluated with the help of ModelSim 6.3C simulation software. Simulation result of traditional R2SDF FFT is illustrated in Figure 13. Similarly, simulation result of proposed R4SDF FFT is illustrated in Figure 14. In both of simulation, numbers from 0 to 15 are given to both real and imaginary terms. When compared to simulation result of R4SDF FFT with simulation result of R2SDF FFT, second half of the input only gives some numerical value changes due to reducing the structure of BPM based complex multiplier. The frequency response obtained from both R2SDF FFT and R4SDF FFT are compared and analyzed in Table 3.

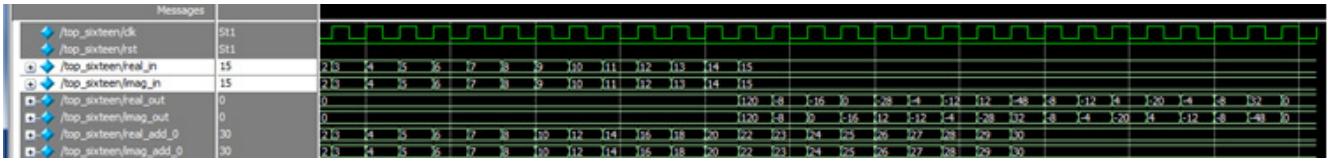

Figure 13. Simulation result of traditional 16 point R2SDF FFT.

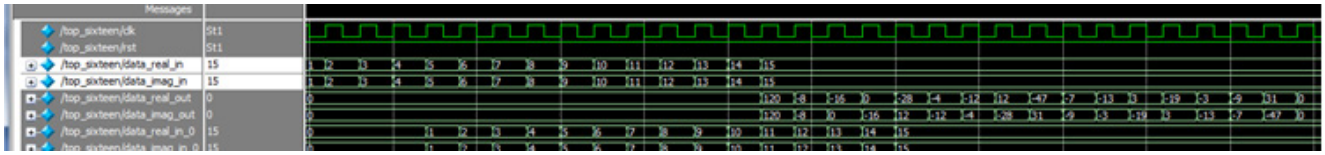

Figure 14. Simulation result of proposed 16 point R4SDF FFT.

Table 3. Frequency response obtained from both R2SDF FFT and R4SDF FFT from same input

Input	Output from R2SDF FFT	Output from proposed R4SDF FFT
0+j0	120+j120	120+j120
1+j1	-8-j8	-8-j8
2+j2	-16+j0	-16+j0
3+j3	0-j16	0-j16
4+j4	-28+j12	-28+j12
5+j5	-4-j12	-4-j12
6+j6	-12-j4	-12-j4
7+j7	12-j28	12-j28
8+j8	-48+j32	-47+j31
9+j9	-8-j8	-7-j9
10+j10	-12-j4	-13-j3
11+j11	4-j20	3-j19
12+j12	-20+j4	-19-j3
13+j13	4-j12	-3-j13
14+j14	-8-j8	-9-j7
15+j15	32-j48	31-j47

For the first half of the input, both R2SDF FFT and R4SDF FFT provide same response due to only involving signed addition. The values of second half of the proposed R4SDF FFT is slightly changed when compared to traditional R2SDF FFT due to reducing the hardware complexity of BPM based complex multiplication. However, the changes are considerable in nature. Even though, it gives the 0.01% percentage error, the synthesis performance of R4SDF FFT improves effectively than all other best existing systems.

8. Performance Evaluation

The design of proposed BPM based complex multiplier; R2SDF FFT and R4SDF FFT are synthesized by using Xilinx Integrated Software Environment (ISE) tool. Performance evaluation of traditional R2SDF FFT and proposed R4SDF FFT are evaluated in terms of utilization of silicon slices, number of LUTs, delay consumptions and power consumptions. Results of proposed R4SDF and traditional R2SDF FFT are analyzed for multiple chips like Spartan-3, Spartan-3E, Virtex E, Virtex 6 and Virtex 6E low power. Performance evaluation of traditional and proposed BPM based multiplier for 0.707 is illustrated in Table 4. Performance evaluation of traditional and proposed BPM based multiplier for 0.9238 is illustrated in Table 5.

Table 4. Performance evaluation of traditional and proposed BPM based twiddle factor multiplication values for 0.707

Devices	Methods	Used Slices	Number of 4 input LUTs	Latency (ns)	Frequency (MHz)	Percentage Reduction	
						Area	Delay
Spartan 3 (Xc3s50)	Traditional	20	38	18.751	53.33	10.52%	7.22%
	Proposed	18	34	17.397	57.48		
Spartan 3 (Xc3s200)	Traditional	20	38	18.751	53.33	10.52%	7.22%
	Proposed	18	34	17.397	57.48		
Spartan 3E (VQ100)	Traditional	20	38	14.668	68.17	10.52%	7.50%
	Proposed	18	34	13.567	73.70		
Virtex E (fg900)	Traditional	21	40	18.746	53.34	17.5%	4.02%
	Proposed	17	33	17.992	55.58		
Virtex6 (XC6VLX75T)	Traditional	7	24	3.763	265.74	-	0.5%
	Proposed	8	24	3.744	267.09		
Virtex6LP (XC6VLX-75TL)	Traditional	7	24	4.553	219.63	-	1.3%
	Proposed	7	24	4.492	222.61		

Table 5. Performance evaluation of traditional and proposed BPM based twiddle factor multiplication values for 0.9238

Devices	Methods	Used Slices	Number of 4 input LUTs	Latency (ns)	Frequency (MHz)	Percentage Reduction	
						Area	Delay
Spartan 3 (Xc3s50)	Traditional	26	48	21.336	46.80	15.38%	2.31%
	Proposed	22	42	20.842	47.98		
Spartan 3 (Xc3s200)	Traditional	26	48	21.336	46.80	15.3%	2.31%
	Proposed	22	42	20.842	47.98		
Spartan 3E (VQ100)	Traditional	26	48	16.679	59.95	15.3%	2.43%
	Proposed	22	42	16.273	61.45		
Virtex E (XCV600E)	Traditional	26	48	21.191	47.18	15.3%	5.13%
	Proposed	22	41	20.103	49.74		
Virtex6 (XC6VLX75T)	Traditional	6	21	1.974	506.58	-	5.97%
	Proposed	6	21	1.856	538.87		
Virtex6LP (XC6VLX-75TL)	Traditional	6	21	4.689	213.26	-	1.98%
	Proposed	6	21	4.596	217.58		

Further proposed twiddle factor multiplier of 0.707 and 0.9238 are integrated in to Radix-4 SDF FFT structure to improve the performance in terms of hardware utilization, delay consumption and power consumption. Proposed Radix-4 SDF FFT structure is compared with traditional Radix-2 SDF and Radix-2 MDC FFT structures. Performance evaluation of both traditional Radix-2 SDF FFT and Proposed Radix-4 SDF FFT is compared in Table 6.

Table 6. Performance evaluation of traditional R2SDF, R2MDC FFT and proposed R4SDF FFT Structure with other FFT structures

Types/Parameters	R2SDF FFT with traditional Complex Multiplier	R2SDF FFT with proposed Complex Multiplier	R2MDC FFT with traditional Complex Multiplier	R2MDC FFT with proposed Complex Multiplier	R4SDF FFT with traditional Complex Multiplier	R4SDF FFT with proposed Complex Multiplier
Family: Spartan 3, Device: Xc3s50, Package: PQ208, Speed: -5						
Number of Occupied Slices	1299	1262	1027	1021	-	-

Total Number of 4 Input LUTs	2193	2118	1584	1585	-	-
Minimum Period (ns)	22.864	21.264	19.160	18.526	-	-
Frequency (MHz)	43.737	47.028	52.193	53.978	-	-
Input arrival latency before clock (ns)	18.103	18.757	19.443	18.810	-	-
Output required latency after clock (ns)	32.660	31.060	21.079	21.079	-	-
Maximum Computational Time period (ns)	27.204	27.521	21.363	21.363	-	-
Power Consumption (mW)	1137	1078	1061	1124	-	-
Family: Virtex E, Device: Xcv600, Package: FG900, Speed: -8						
Number of Occupied Slices	1272	1233	1033	1022	996	970
Total Number of 4 Input LUTs	2087	2014	1608	1576	1592	1486
Minimum Period (ns)	29.375	28.991	23.621	21.313	27.131	25.399
Frequency (MHz)	34.043	34.493	42.335	46.920	36.858	36.382
Input arrival latency before clock (ns)	21.785	23.188	23.753	21.445	21.922	39.465
Output required latency after clock (ns)	36.393	36.009	19.742	19.582	19.642	19.482
Maximum Computational Time period (ns)	28.803	28.945	19.424	19.424	19.318	19.454
Power Consumption (W)	43.183	40.604	142.110	94.733	26.055	13.088
Family: Virtex 6, Device: Xc6vlx75T, Package: FF748, Speed: -3						
Number of Slice Registers	803	787	778	707	421	421
Number of FFs	803	787	766	707	421	421
Number of Slice LUTs	1972	1931	1479	1373	1519	1295
Number of Used Memory	90	96	48	65	82	82
Number of Shift Register	97	96	64	38	82	82
Occupied Slices	556	539	471	428	473	401
Fully used FF_LUT Pair	1983	1956	1558	1484	1569	350
Minimum Period (ns)	5.902	6.039	5.535	5.096	5.892	5.9697
Frequency (MHz)	169.445	165.594	180.654	196.242	169.733	175.545

Input arrival latency before clock (ns)	5.055	4.434	4.688	4.885	5.482	5.244
Output required latency after clock (ns)	7.240	7.377	4.478	4.688	4.801	4.801
Maximum Computational Time period (ns)	6.201	5.697	19.424	4.478	4.463	4.463
Power Consumption (W)	3.310	2.941	2.911	2.889	3.258	2.815

From Table 6, it is clear that proposed R4SDF FFT gives the better performance than traditional R2SDF FFT in terms of Number of Occupied Slices, Latency and Power consumption. When compared to the R2MDC architecture, delay of proposed R4SDF FFT increases due to architectural changes. But, R2MDC FFT has poor hardware architecture than R2SDF FFT structures. Hardware structure and power consumption of proposed R4SDF FFT are effectively reduced than other best existing FFT structures. A pin configuration of Spartan-3 (Xc3s200) is not suitable for implementing Radix-4 SDF FFT.

Hence, Figure 10 and Figure 11 are the most sufficient multiplier structures to FFT processors. In addition, Radix-4 FFT reduces the computational path effectively than Radix-2 FFT structures. Single path Delay Feedback (SDF) structures are further used to reduce the hardware complexity of both Radix-2 and Radix-4 FFT structures. From performance evaluation, it is clear that Radix-4 SDF FFT is the better solution for frequency transformation techniques of OFDM based data communication application.

9. Conclusion

This paper has been motivated by the need to reduce the complexity of complex multiplier of frequency transformation techniques like Fast Fourier Transformation (FFT) and Inverse Fast Fourier Transformation (IFFT) technique. Twiddle factor values of FFT processors are in terms of fractional values. But, fractional multiplier is not suitable for increasing the performance of FFT processors. In perspective of FFT computation, twiddle factor multiplication only has more complexity than other elements. Bit Parallel Multiplier (BPM) based twiddle factor multiplications are realized in this paper to improve the FFT/IFFT transformations. When compared to traditional twiddle factor multiplier, proposed BPM based twiddle factor multiplier offers 17.5% reduction of area utilization and 7.22% reduction of latency for the value 0.707 and 15.3% reduction of area utilization and 5.97% reduction of latency for the value 0.9238. Results of traditional and proposed BPM based twiddle factor multipliers are synthesized in multiple chips like Spartan-3 (Xc3s200), Spartan-3 (Xc3s50), Spartan-3E (VQ100), Virtex E (Xcv600E), Virtex 6 (Xc6vlx75T) and Virtex 6 low power (Xc6vlx75T). Proposed BPM based twiddle factor multiplier is integrated into 16 point Radix-4 SDF FFT for improving the performance of Radix-4 FFT transformation techniques. Results of proposed FFT processors are synthesized in multiple chips like Spartan 3 (Xc3s200), Virtex E (Xcv600E) and Virtex 6 (Xc6vlx75T). Proposed Radix-4 SDF FFT offers 23.74% reduction of Slices in Virtex E and 47.57% reduction of Slices in Virtex 6 than traditional Radix-2 SDF FFT. Similarly, proposed Radix-4 SDF FFT offers 47.96% reduction of total latency in Virtex E and 28.02% total latency in Virtex 6 than traditional Radix-2 SDF FFT. Power consumption of proposed Radix-4 SDF FFT offers 69.69% reduction in Virtex E and 14.95% reduction in Virtex 6 than traditional Radix-2 SDF FFT. Hence, overall Virtex 6 Radix-4 SDF FFT processor is suitable for low area applications and Virtex 6 Radix-4 SDF FFT processor is suitable for low latency and lower power consumption applications. In future, this proposed technique is further extended to 64, 128 bits of FFT transformations and will be integrated in OFDM applications for improving the process of data communication.

10. References

1. Arunachalam V, Raj ANJ. Efficient VLSI implementation of FFT for orthogonal frequency division multiplexing applications. IET Circuits, Devices and Systems. 2014 Nov; 8(6):526–31.
2. Cheng C, Parhi KK. Low-cost fast VLSI algorithm for discrete Fourier transform. IEEE Transactions on Circuits and Systems I: Regular Papers. 2007 April; 54(4):791–806.
3. Garrido M, Grajal J, Sanchez MA, Gustafsson O. Pipelined

Radix- 2^k Feed forward FFT Architectures. IEEE Transactions on Very Large Scale Integration (VLSI) System. 2013 Jan; 21(1):23–32.
4. Hasan M, Arslan T, Thompson JS. A novel coefficient ordering based low power pipelined radix-4 FFT processor for wireless LAN applications. IEEE Transactions on Consumer Electronics. 2003 Feb; 49(1):128–34.
5. Kim D, Choi HW. Advanced constant multiplier for multipath pipelined FFT processor. Electronics Letters. 2008 Apr; 44(8):518–9.
6. Ma YT. A VLSI-oriented parallel FFT algorithm. IEEE Transactions on Signal Processing. 1996 Feb; 44(2):445–8.
7. Mactaggart R, Jack MA. Radix-2 FFT butterfly processor using distributed arithmetic. Electronics Letters. 1983 Jan; 19(2):43–4.
8. Maharatna K, Grass E, Jagdhold U. A 64-point Fourier transform chip for high-speed wireless LAN application using OFDM. Solid-State Circuits. IEEE Transactions on Solid State Circuits. 2014 Mar; 39(3):484–93.
9. Miyaoka F, Kainuma T, Shimamura Y, Yamanashi Y, Yoshikawa N. High-speed test of a Radix-2 butterfly processing element for Fast Fourier Transforms using SFQ circuits. IEEE Transactions on Applied Superconductivity. 2011 Jun; 21(3):823–6.
10. Bhakthavatchalu R, Kannan SK, Nirmala Devi M. Verilog Design of Programmable JTAG Controller for Digital VLSI IC's. Indian Journal of Science and Technology. 2015 Aug; 8(17):1–7.
11. Roste T, Haaberg O, Ramstad T. A Radix-4 FFT processor for application in a 60-channel transmultiplexer using TTL technology. IEEE Transactions on Acoustics, Speech and Signal Processing. 1979 Dec; 27(6):746–51.
12. Swartzlande EE, Saleh HH. FFT implementation with fused floating-point operations. IEEE Transactions on Computers. 2012 Feb; 61(2):284–8.
13. Taylor FJ, Papadourakis G, Skavantzos A, Stouraitis A. A Radix-4 FFT using complex RNS arithmetic. IEEE Transactions on Computers. 1985 Jun; 34(6):573–6.
14. Wang Z, Liu X, He B, Yu F. A combined SDC-SDF architecture for normal I/O pipelined Radix-2 FFT. IEEE Transactions on Very Large Scale Integration (VLSI) System. 2015 May; 23(5):973–7.
15. Wang CC, Huang JM, Cheng HC. A 2K/8K mode small-area FFT processor for OFDM demodulation of DVB-T receivers. IEEE Transactions on Consumer Electronics. 2005 Feb; 51(1):28–32.
16. Yu C, Yen MH, Hsiung PA, Chen SJ. A low-power 64-point pipeline FFT/IFFT processor for OFDM applications. IEEE Transactions on Consumer Electronics. 2011 Feb; 57(1):40.