CHAPTER 5

PERFORMANCE EVALUATION OF COMBINED RADIX-2, RADIX-4 AND RADIX-8 BASED SINGLE PATH DELAY FEEDBACK (SDF) FFT

Fast Fourier Transform (FFT) algorithm plays a vital role in improving the opportunity of using the Discrete Fourier Transform (DFT) in a wide range of applications. In modern times, special kind of FFT algorithms has been designed to gather real-time processing requirements and to reduce the hardware complexity over the previous decades. FFT calculations are high and a better method than a general purpose processor is essential to fulfilling the necessities at a practical cost. The real-time processing requirements are major concerns for researchers and to reduce hardware complexity mainly on area, power and to develop processing speed of the processor. At present, FFT is the key building block for the mobile communications, particularly for OFDM transceiver system.

5.1 Radix-2 FFT and Radix-4 FFT

Fast Fourier Transform is the important method for calculating DFT. The DFT of a time domain signal is expressed by

$$X_k = \sum_{n=0}^{N-1} x_n W_N^{nk}$$

$$0 \leq k \leq N-1$$

(5.1)
In the case of Radix-2 FFT algorithm, the fundamental block of Radix-2 butterfly FFT which is shown in Figure 5.1. It shows that butterfly is simply a DFT of size-2. It takes two inputs x0, x1 and gives two outputs a0, a1.

The Radix-2 FFT algorithms (Qadeer et al.,) are the simplest form of FFT algorithms. The Radix-2 FFT with decimation in frequency (DIF) algorithm recursively partitions a DFT into two half-length DFTs of the even and odd indexed time samples. The smaller number of FFT outputs are reused to compute many outputs, thus greatly reducing the total computation cost. All the FFTs, they gain their speed by reusing the results of smaller, intermediate computation to calculate multiple DFT frequency outputs.

Radix-4 FFT algorithm which was used to enhance the speed of functioning by reducing the computational path. If base increases, the power/index will decreases. Radix-4 FFT the number of stages is reduced to 50%. It has four inputs and outputs, and it follows the in-place algorithm. The shorter FFT outputs are reused to calculate many outputs. Thus the total computational cost is greatly reduced. The Radix-4 FFTs need only 75% as many complex multiplications as the Radix-2 FFTs. The basic block of Radix-4 butterfly FFT is shown in Figure 5.2.
$$Y[K] = \sum_{n=0}^{\frac{N}{4}-1} \left\{ y(n) + W_{N/4}^{N_k} (y(n + N/4)) + W_{N^4}^{3N_k} (y(n + 3N/4)) \right\} W_{N}^{nk}$$ (5.2)

**Figure 5.2 Radix-4 Butterfly FFT**

In a comparison of Radix 2 FFT, the number of complex multiplications is reduced by 25%, but the number of complex additions is increased by 50% in Radix-4 FFT (Neuenfeld et al.,). The common method to perform a butterfly element is direct mapping the structure of the hardware. In the Radix-2 butterfly element, the hardware cost is very cheap because only two complex adders and one complex multiplier are needed. The same method can be used to perform the Radix-4 butterfly structure, where 8 complex adders and 3 complex multipliers are needed. Hardware implementation cost is three times more than Radix-2 butterfly FFT implementation. When the radix increased, the hardware cost will be automatically increased.
5.2 Radix-8 FFT

Radix-8 FFT algorithm, which was used to enhance the speed of the FFT processor. In this algorithm, the value of r is 8. In Radix-8 FFT, the numbers of computational stages are reduced to 75%. The DIF Radix-8 FFT is split into eight quarter-length DFTs of groups of every eighth sample. The shorter FFT outputs are reused to calculate many outputs the total computational cost is significantly reduced. Compared with Radix-2 and Radix-4 FFT, the computational path will be significantly reduced in Radix-8 FFT. The Radix-8 FFT is shown in Figure 5.3. The Radix-2 FFT of real addition is 1032, and real multiplication is 264. The Radix-4 FFT of real addition is 976, and the real multiplication is 208. The Radix-8 FFT of real addition is 972, and the real multiplication is 204.

![Figure 5.3 Radix-8 Butterfly FFT](image)

![Figure 5.4 Signal Flow Graph of Radix-8 Butterfly Element](image)
The Radix-8 butterfly element, which required 3 multipliers with \(-j\), 2 multipliers with \((\sqrt{2})/2\), \((1 \pm j)\) and 7 non-trivial complex multipliers. The complexity of the Radix-8 butterfly element is considerably reduced and it has only one complex multiplier and thus the area is very small.

The Radix-8 butterfly element was proposed for consuming the silicon area of 21mm\(^2\) in 0.8um CMOS technology although the bit-serial arithmetic was used to reduce the large area. The pipelined techniques are used to cascade the butterfly element and thus the adders and multipliers required to calculating FFT operations are significantly reduced.

5.3 Single Path Delay Feedback (SDF) FFT

The input data sequences fed through one single path delay feedback. The butterfly-processing element performs the calculations on the data. The delay units are more efficiently utilized by sharing the same storage between input and output of butterfly unit. Butterfly units and multiplier can be utilized 50% because they are bypassed half the time. Single Path Delay Feedback (R2SDF) FFT is a pipelined based frequency transformation technique. The structure of R2SDF FFT is just like "stream-like" processing of block-based algorithm (Yang et al.,). The representation of sequential data flow in SDF FFT is shown in Figure 5.5. It has single butterfly processor for performing signed addition and signed subtraction function. Single path delay elements have been used in the feedback structure.
Figure 5.5 Representation of Sequential Dataflow in SDF FFT

The input of Figure 5.5 is given sequentially into butterfly unit. The feedback of single path delay unit performs the shifting operation of real and imaginary inputs. To reduce the number of adders and subtractors of processing element, sequential circuits based FFT processors have been used in this research work. The symbolic representation of signal flow in processing element (Butterfly Structure) of SDF FFT is illustrated in Figure 5.6.
5.4 Proposed combined Radix-2, 4 and 8 based Single Path Delay Feedback (SDF) FFT

The combined Radix-2, 4 and 8 based Single Path Delay Feedback (SDF) FFT has been designed in this proposed work. The combined Radix of FFT architecture has a lesser amount of computational path and also improves the performances of FFT processor. Single path Delay Feedback architecture, the input data sequences are pass through one single path. The butterfly processing element performs the computation on the data. The addition and subtraction operation is done in butterfly elements. The modified carry select adder circuit is used for adder operation in this architecture. This adder structure is very efficient in this architecture. The structure of combined Radix-2,
4 and 8 FFT is shown in Figure 5.7. The architecture of 16 point Single Path Delay Feedback (SDF) FFT is shown in Figure 5.8.
Figure 5.8 shows the 16 point single path delay feedback architecture. The procedural flow of this architecture is as follows, in the beginning the input data with indices 0 to 7 are stored in the shift register. The Radix-2 butterfly elements operate on these data and the remaining input data with indices 8 to 15. The resulting data from the butterfly addition operation are passed to the second stage, and the subtraction results are fed back to the shift register. After that the 8-point data addition is passed to the second stage, it has been done by using Radix-8 butterfly element and the subtraction data from the registers are passed to the second stage has been done by a Radix-2 butterfly with butterfly-processing twiddle factor coefficient. The next stages are done by using Radix-2 and Radix-4 FFT.

In the traditional R2SDF FFT, inputs are given into sequentially, and the four inputs are processed along with the help of single butterfly (Processing Element) unit. However, this architecture of hardware utilization is more and power consumption due to utilizing or storing the bulk of unnecessary intermediate processing digital signals. To overcome this problem, the design of Radix-2, Radix-4 and Radix-8 based SDF FFT architectures are combined to reduce the hardware utilization of the processor. The proposed method which significantly reduces the area, delay, and power consumption. The combined Radix-2, 4 and 8 FFT has been proposed in this architecture for reducing the computational stages. For example 64 point FFT, Radix-2 FFT has 6 stages to compute the FFT output. Radix-4 FFT has only 3 stages. Compared to Radix-2 and Radix-4 FFT, Radix-8 has only 2 stages. So combined the Radix-2, 4 and 8 for improving performance of architecture. In the proposed method, the 16 point FFT is divided into two half, the first 8 point is directly got the output by using Radix-8 FFT. The numbers of stages are reduced and also reduced the processing time. The next 8 points, used Radix-4 and Radix-2 FFT. In the normal 16 point Radix-2 FFT, 15 stages of Radix-2 FFT are used. In the proposed combined Radix-2, 4 and 8 FFT, only 5 stages of Radix-2 FFT has been used. When compared to normal Radix-2
FFT, the combined Radix-2, 4 and 8 FFT has less computational path than the existing method.

5.5 Results and Discussion

The architecture of combined Radix-2, 4 and 8 based Single path delay Feedback (SDF) FFT has been simulated by using MODELSIM 6.3C. The performance results are synthesized by Xilinx ISE 10.1i design tool. The simulation result of existing Radix-2 SDF FFT and proposed combined 16 point Radix-2, 4 and 8 FFT based SDF is shown in Figure 5.9 and Figure 5.10.

Figure 5.9 Simulation result of existing Radix-2 SDF FFT
Figure 5.10 Simulation result of proposed combined radix-2, 4 and 8 based SDF FFT

The synthesis result of existing Radix-2 SDF FFT and proposed combined Radix-2, 4 and 8 based SDF FFT is to determine the area consumption is illustrated in Figure 5.11 and Figure 5.12. The Figure 5.11 shows that the number of LUTs and Slices required for existing Radix-2 SDF FFT. The number of Look up Tables (LUTs) is required for implementing existing Radix-2 SDF FFT is 850 and required number of
occupied slices is 609. However, in the case of proposed combined Radix-2, 4 and 8 based SDF FFT, the required number of LUTs is reduced to 319 and required slices are reduced to 251, which is shown in Figure 5.12. The numbers of gate counts are reduced, so that the slices and LUTs have been reduced in this work.

Figure 5.11 Synthesis result of existing Radix-2 SDF FFT to determine the area consumption
Figure 5.12 Synthesis result of proposed combined Radix-2, 4 and 8 based SDF FFT to determine the area consumption
Similarly, the synthesis result for both existing Radix-2 SDF FFT and proposed combined Radix-2, 4 and 8 based SDF FFT regarding delay is shown in Figure 5.13 and Figure 5.14.

Figure 5.13 Synthesis Result of existing Radix-2 SDF FFT to determine the delay consumption
Figure 5.14 Synthesis Result of proposed combined Radix-2, 4 and 8 based SDF FFT to determine the delay consumption

In the case of existing Radix-2 SDF FFT, the time required for performing operation 15.062ns, which is reduced to 9.360ns in the case of proposed combined Radix-2, 4 and 8 based SDF FFT.
Similarly, the synthesis result for both existing Radix-2 SDF FFT and proposed combined Radix-2, 4 and 8 based SDF FFT regarding power consumption is shown in Figure 5.15 and Figure 5.16.

Figure 5.15 Synthesis Result of existing Radix-2 SDF FFT to determine the power consumption
Figure 5.16 Synthesis Result of proposed combined Radix-2, 4 and 8 based SDF FFT to determine the power consumption
Table 5.1 Comparison of Existing Radix-2 SDF FFT and Proposed Combined Radix-2, 4 and 8 based SDF FFT

<table>
<thead>
<tr>
<th>Types / VLSI Concerns</th>
<th>Number of Occupied Slices</th>
<th>Number of LUTs</th>
<th>Delay(ns)</th>
<th>Power(W)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Existing Radix-2 SDF FFT</td>
<td>609</td>
<td>850</td>
<td>15.062</td>
<td>2.197</td>
</tr>
<tr>
<td>Proposed mixed Radix-2, 4 and 8 based SDF FFT</td>
<td>251</td>
<td>319</td>
<td>9.360</td>
<td>1.519</td>
</tr>
<tr>
<td>Percentage Reduction</td>
<td>58.78%</td>
<td>62.47%</td>
<td>37.85%</td>
<td>30.86%</td>
</tr>
</tbody>
</table>

In the case of existing Radix-2 SDF FFT, the power consumption is 2.197W, which is reduced to 1.519W in the case of proposed combined Radix-2, 4 and 8 based SDF FFT.
The comparison of both existing Radix-2 SDF FFT and proposed combined Radix-2, 4 and 8 based SDF FFT regarding LUTs, Slices, delay and power values is shown in Table 5.1.

Table 5.1 shows that the proposed combined Radix-2, 4 and 8 based SDF FFT offers 58.78% reduction in occupied slices, 62.47% reduction in LUTs, 37.85% reduction in delay and 30.86% reduction in power consumption than the existing Radix-2 SDF FFT. The performance evaluation of both existing Radix-2 SDF FFT and proposed combined Radix-2, 4 and 8 based SDF FFT is pictorially represented in Figure 5.17.

![Figure 5.17 Performance evaluation of existing R2SDF FFT and proposed combined Radix-2, 4 and 8 based SDF FFT](image-url)
5.6 Summary

The developed combined Radix-2, 4 and 8 based single path delay feedback FFT has been designed in this research work. The main purpose of this work is to reduce the area, latency and power consumption of the processor and also reduces the computational stages. The proposed architecture offers 58.78% reduction in slices, 62.47% reduction in LUTs, 37.85% reduction in delay and 30.86% reduction in power consumption than the existing method. Compared to the existing method, the proposed method of computational stages is reduced and gives better performances than the traditional one. Thus, this design is particularly useful for low power applications such as WLAN, OFDM, etc. For future work, plan to optimize complex multiplier for more power reduction.