Area and Time Efficient FFT Architecture Using Hardwired Pre-Shifted Bi-Rotation Cordic Design

Manikandan .M1, Paramasivam .C2

1Department of Electronics and Communication Engineering, K.S.Rangasamy College of Technology, Tirunchengode, India.
2Department of Electronics and Communication Engineering, K.S.Rangasamy College of Technology, Tirunchengode, India.

ABSTRACT—This paper presents CORDIC based feed forward FFT architecture. It is used to implement the pipeline FFT hardware architecture. The radix 2k feed forward FFT architecture can be used for the any number of parallel sample which is power of two. It can be achieved the high throughput and low hardware requirement. The hardwired pre shifted bi-rotation cordic technique for barrel-shifter of proposed circuit. Here two proposed CORDIC cells are used to the fixed angle rotations. This cell going to implement the micro rotations and scaling interleaved, it’s implemented the two stages. The cascade proposed the bi-rotation CORDIC for higher throughput and reduced latency implementation. This method proposed optimized set of micro rotations for fixed and known angles. Shift and add circuits are used to implement the scaling factor. Fixed means square error used for analysis and reduced the error in this method. Synthesized the proposed CORDIC cells by Synopsys Design Compiler using TSMC 90-NM library, and shown that the proposed designs offer higher throughput and less area-delay product than the reference CORDIC design for fixed and known angles of rotation. We find similar results of synthesis of different Xilinx field-programmable gate-array platforms.

INDEX TERMS— FFT, Coordinate rotation digital computer (CORDIC), digital arithmetic, digital signal processing (DSP) chip, VLSI.

I. INTRODUCTION

Fast Fourier transform (FFT) is among the most widely used operations in digital signal processing. Often, a high performance FFT processor is the key component and determines most of the design metrics in many applications such as Orthogonal Frequency-Division Multiplexing (OFDM), Synthetic Aperture Radar (SAR) and software defined radio. For embedded systems, in particular portable devices; efficient hardware realization of FFT with small area, low-power dissipation and real-time computation is a significant challenge. A typical FFT processor is composed of butterfly calculation units, memory banks and control logic (address generator for data and twiddle factor accesses). In most cases, an FFT processor uses only one butterfly unit to realize all calculations iteratively, and the “in place” memory access strategy is required for the least amount of memory. With “in place” strategy, the outputs of a butterfly operation are stored back to the same memory location of the inputs, saving the memory usage by one half. However, correct memory addressing scheme is required to avoid the data conflict. This study implements an efficient addressing scheme to realize the parallel, pipelined and “in-place” memory accessing. It produces an output at every clock cycle; furthermore the memory banks and the butterfly unit are utilized with 100% efficiency within the pipeline. In FFT processors, butterfly operation is the most computationally demanding stage. Traditionally, a butterfly unit is composed of complex adders and multipliers, and the multiplier is usually the speed bottleneck in the pipeline of the FFT processor. The Coordinate Rotation Digital Computer (CORDIC) [5] algorithm is an alternative method to realize the butterfly
operation without using any dedicated multiplier hardware. CORDIC algorithm is very versatile and hardware efficient since it requires only add and shift operations, making it very suitable for the butterfly operations in FFT [6]. Instead of storing actual twiddle factors in a ROM, the CORDIC-based FFT processor needs to store only the twiddle factor angles in a ROM for the butterfly operation. Additionally, the CORDIC-based butterfly can be twice faster than traditional multiplier based butterflies in VLSI implementations. The cordic based FFT architecture is used reduced hardware source and increase high throughput.

II. RADIX-2² FFT ALGORITHM

The -point DFT of an input sequence \( X_n \) is defined as

\[
X[k] = \sum_{n=0}^{N-1} X[n] W_N^{nk} \quad k = 0,1,2 \ldots \ldots N - 1
\]

Where \( W_N^{nk} = e^{-j2\pi nk/N} \)

When \( N \) is a power of two [15], the FFT based on the Cooley-Tukey algorithm is most commonly used in order to compute the DFT efficiently. The Cooley-Tukey algorithm

Reduces the number of operations from \( o(N^2) \) for the DFT to \( o(N \log_2 N) \) for the FFT. In accordance with this, the FFT is calculated in a series of \( n = \log_2 N \) stages, where is the base of the radix, of the FFT, i.e., Figs. 1 show the flow graphs of 16-point radix-2 and radix-FFTs, respectively, decomposed using decimation in frequency (DIF). At each stage of the graphs, butterflies and rotations have to be calculated.

The lower edges of the butterflies are always multiplied by 1. These 1 are not depicted in order to simplify the graphs. The numbers at the input represent the index of the input sequence, whereas those at the output are the frequencies, of the output signal \( X[k] \). Finally, each number, in between the stages indicates a rotation by

\[
W_N^\phi = e^{-j2\pi \phi/N}
\]

As a consequence, samples for which \( \phi = 0 \) do not need to be rotated. Likewise, if \( \phi = [0,N/4,N/2,3N/4] \) the samples must be rotated by 0,270,180, and 90, which correspond to complex multiplications by 1,-j, j ,1 and respectively. These rotations are considered trivial, because they can be performed by interchanging the real and imaginary components and/or changing the sign of the data. Radix-2² is based on radix-2 and the flow graph DIF FFT can be obtained from the graph of a radix-2 DIF one. This can be done by breaking down each angle, at odd stage into a trivial rotation and a non-trivial one, where \( \phi = \phi \mod N/4 \) and moving the latter to the following stage. This is possible thanks to the fact that in the radix-2 DIF FFT the rotation angles at the two inputs of every butterfly, and only differ by 0 or \( \pi \). Thus, if and, the rotation is moved to the following stage in accordance with Analogously, the radix- DIT FFT can be derived from the radix-2 DIT FFT. Contrary to DIF, for DIT the non-trivial rotations are moved to the previous stage instead of the following one.

III. HARDWIRED PRE SHIFTED BI-ROTATION CORDIC ALGORITHM

This technique is used to reduce the barrel shifter complexity. It has proposed the two methods,

1) Bi-rotation CORDIC cell.
2) Case Cade Bi-rotation CORDIC.

Constants complex multiplication CORDIC circuit shown Fig 3. The control bits are stored in to the rom. SBR it is used to store the barrel –shifter and direction of micro-rotation. Micro rotations are implemented by the barrel shifter. The main contribution of the hardware is to implement the barrel-shifter and adder. Hard wired, pre shifting it’s used to the minimization of barrel shifter complexity.
A. **BI-ROTATION CORDIC CELL**

The proposed bi-rotation CORDIC circuit consists of an adder module, two multiplexer and sign bit register. The adder module consists of adder and subtracts which performs add and subtract. The output of the register sends in the two parts. One of the lines directly fed to the adder module. Another line fed into the barrel shifter. Pre-shifted by \( k(0) \) bit location shifted right by using hard wired pre shifting technique. The output of the loader is given in the register for the second CORDIC iteration. The barrel-shifters are 0 for the first rotations and 1 for second micro rotations. Flip flop is used to generate the control bit and each cycle having the value 1 and 0.

The proposed bi-rotation CORDIC cell has single rotation module Rotation module perform specific micro-rotations. Bi-rotation cell. It’s is used to reduce the adder complexity of the single rotation CORDIC. Bi-rotation CORDIC used to implement the micro rotation. Here we have to proposed the two and three stage case code bi-rotation CORDIC for higher accuracy. The structure of the function it should be shown in the fig 4. The stage one four optimized micro rotations is implemented. While the rest two is performed by stage.

IV **PROPOSED CORDIC BASED FFT**

The proposed CORDIC based FFT is nothing but feed forward FFT architecture.
butterflies. This condition is $b_{n-s}$ both for DIF and DIT decompositions and means that at any stage of the FFT, butterflies operate in pairs of data whose indices differ only in bit $s$, where the number $n = \log_2 n$ is stages of the FFT. In it can be observed that at the third stage, data with indices. On the other hand, there are two properties for rotations. At odd stages of the radix- DIF FFT only those samples whose index fulfills have to be rotated. These rotations are trivial and the symbol indicates the logic AND function. For the 16-point radix-FFT in only samples with indices 12, 13, 14, and 15 must be rotated at the first stage. For these indices is fulfilled $b_{n-s}, b_{n-s-1}$, meeting the property, since and. Conversely, at even stages rotations are non-trivial and they are calculated over indexed data for which where the symbol indicates the logic OR function. Analogously, the radix- DIT FFT can be derived from the

**B. RADIX-2² FEED FORWARD FFT ARCHITECTURE**

The radix-2² feed forward architectures. First, a 16-point 4-parallel radix- feed forward FFT architecture is explained in detail in order to clarify the approach and show how to analyse the architectures. Then, radix-feed forward architectures for different number of parallel samples are represented. Fig.2 shows a 16-point 4-parallel radix- feed forward FFT architecture. The architecture is made up of radix-2 butterflies (R2), non-trivial rotators, trivial rotators, which are diamond-shaped, and shuffling structures, which consist of buffers and multiplexers. The lengths of the buffers are indicated by a number. The architecture processes four samples in parallel in a continuous flow. The order of the data at the different stages is shown at the bottom of the figure by the indices, together with the bits that correspond to these indices. In the horizontal, indexed samples arrive at the same terminal at different time instants, whereas samples in the vertical arrive at the same time at different terminals. Finally, samples flow from left to right. Thus, indexed samples (0, 8, 4, and 12) arrive in parallel at the inputs of the circuit at the first clock cycle, whereas indexed samples (12, 13, 14, and 15) arrive at consecutive clock cycles at the lower input terminal. Taking the previous considerations into account, the architecture can be analyse. First, it can be observed that butterflies always operate in pairs of samples whose indices differ in bit, meeting the property in Table I. For instance, the pairs of data that arrive at the upper butterfly of the first stage are: (0, 8), (1, 9), (2, 10), and (3, 11). The binary representation of these pairs of numbers only differ in $s$. As, and at the first stage, so the condition is fulfilled.

This property can also be checked for the rest of the butterflies in a similar way.

**V. IMPLEMENTATION OF SCALING CIRCUIT**

Scaling circuit, it’s used to derived the shift add operations. Here CORDIC circuits realized the different level of implementation of micro-rotation [14].

$$T = \prod_{i=0}^{n-1} [1 + 2^{-2i}]^{-1/2}$$

### A. SCALING FOR BI ROTATION CORDIC

Scaling and micro-rotation could be implemented in two separated pipeline stage. The scale factor in this case should be represented by two shifts – add terms

$$KA = 1 + \delta_0 2^{-s(0)} X(1 + \delta_1 2^{-s(1)})$$

The two factor scaling can be implemented by the shift add circuit of hard wired pre shifting. It consists of adder and subtractors and a pair of single stage barrel shifter. It consists of 2:1 mux. The input of the barrel-shifter is pre shifted by $s(0)$ locations to right. The input through $s(1)-s(0)$ location to right when the control bit is 1 or control bit is 0 means no additional shifts are required. The T flip-flop is generated the control bit.
B. SCALING FOR CASCADE BI ROTATION CORDIC

It is used to implement in two or three stages for four and six micro-rotations. The scaling circuit in 43 and 45 degree rotations through. We have proposed 43 and 45 degrees based on shown in below diagram.

![Fig 7. Scaling circuit for 45° and 43°](image)

TABLE I
SYNTHESIS RESULT FOR AREA AND TIME EFFICIENT BI-ROTATION CORDIC

<table>
<thead>
<tr>
<th>Designs</th>
<th>Word size</th>
<th>Area (sq.um)</th>
<th>Clock (Ms)</th>
<th>Tpt (Ms)</th>
<th>latency</th>
<th>ACT (ns)</th>
<th>ADP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference CORDIC circuit</td>
<td>16</td>
<td>2678</td>
<td>2.32</td>
<td>35.06</td>
<td>25.76</td>
<td>26.72</td>
<td>70220</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>6565</td>
<td>4.15</td>
<td>13.88</td>
<td>71.08</td>
<td>72.08</td>
<td>473205</td>
</tr>
<tr>
<td>Hard wired pre shifted single rotation case</td>
<td>16</td>
<td>6560</td>
<td>1.41</td>
<td>542.46</td>
<td>12.36</td>
<td>1.71</td>
<td>11217</td>
</tr>
<tr>
<td>CORDIC (1stage)</td>
<td>32</td>
<td>23010</td>
<td>3.54</td>
<td>269.32</td>
<td>39.45</td>
<td>3.43</td>
<td>78924</td>
</tr>
<tr>
<td>Hard wired pre shifted Bi rotation case</td>
<td>16</td>
<td>5809</td>
<td>2.01</td>
<td>216.24</td>
<td>12.50</td>
<td>4.32</td>
<td>25094</td>
</tr>
<tr>
<td>CORDIC (2stage)</td>
<td>32</td>
<td>17667</td>
<td>2.96</td>
<td>116.26</td>
<td>42.56</td>
<td>6.92</td>
<td>122255</td>
</tr>
<tr>
<td>Cordic based feed forward fft architecture</td>
<td>16</td>
<td>4200</td>
<td>1.98</td>
<td>190.8</td>
<td>10.43</td>
<td>3.96</td>
<td>16632</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>15550</td>
<td>2.44</td>
<td>105.2</td>
<td>32.45</td>
<td>4.52</td>
<td>70286</td>
</tr>
</tbody>
</table>

VI. RESULT AND CONCLUSION

The proposed designs of the CORDIC based feed forward FFT architecture have been realized by VHDL. For basic CORDIC designs couldn’t find the any optimized specific angles for the vector-rotation and traditional hard wired, pre-shift bi-rotation CORDIC having the barrel shifter and adder complexities for fixed and known angle rotation. The proposed CORDIC based feed forward FFT architecture is used reduced the barrel and adder complexities. Improve the area and time for the proposed architecture.
The proposed CORDIC based feed forward FFT architecture require, respectively, 3.4 and 2.2 times more area over the reference design, but offer nearly 16.3 and 7.0 times more throughput, and involve nearly 4.6 and 2.5 times less ADP with nearly half and two-thirds of the latency of the other. The simulation is done in ModelSim and the code is functionally verified to be correct. Here TPT stands for throughput and calculated per second. ACT stands for average computation time measured in a nano second. ADP stands for area delay product, calculated as the product of the area and act in nano seconds.

Fig 8. Simulation result for ROM data in cascade bi-rotation CORDIC

Fig 9. Simulation result of hard wired, pre shifted bi-rotation cascade CORDIC

Fig 10. Cordic based feed forward FFT architecture

Fig 11. CORDIC based parallel radix-$2^2$ feedforward FFT

REFERENCE

Area and Time Efficient FFT Architecture Using Hardwired Pre-Shifted Bi-Rotation Cordic Design


