# VLSI Implementation of Binary FFT Architecture by Using Retiming

L .Blessy(M.E-II VLSI Design),
*ECE Department,*
*Syed Ammal Engineering College,*
*Ramanathapuram.*

Mr.B.BanuPriya M.E.,
*Assistant Professor/ECE,*
*Syed Ammal Engineering College,*
*Ramanathapuram.*

**Abstract-***The Fast Fourier Transform (FFT) is one of the most important and fundamental algorithms in the digital signal processing area.This project presents a new type of FFT hardware architectures called binary FFT using selective rotation.The proposed architectures are based on the observation that in the radix-2 FFT algorithm only half of the samples at each stage must be rotated. Develop the binary version of FFT in order to reduce processing overhead.To design efficient transform to convert time domain to frequency domain and also implement systolic structure of FFT. Our proposed architecture has been coded in Verilog HDL and simulated using xilinx 12.1*

## IINTRODUCTION

**F**ast fourier transform(FFT) is one of the widely used algorithms in digital signal processing [1], [2]. There has been an increasing interest in the computation of FFT for real valued signals (RFFT), since virtually most of the physical signals are real. The real valued signals which are of prime importance in real-time signal processingexhibit conjugate symmetry giving rise to redundancies. This property could be exploited to reduce both arithmetic and memory complexities.The RFFT plays an important role in different fields such as communication systems, biomedical applications, sensors and radar signal processing. A dedicated RFFT processor architecture will reduce the hardware complexity in discrete multi-tone (DMT) based technologies like very high bit-rate digital subscriber line (VDSL) [3] and asymmetric digital subscriber line (ADSL) [4].In the area of sensor signal processing, the FFT is used in frequency domain beamforming, source tracking, and harmonic line association and classification [5], [6]. On the other hand RFFT is one of the key algorithms in analyzing biomedical signals such as electrocardiography (ECG), and electroencephalography (EEG) [7]. Frequency domain based features like power spectral density (PSD) can identify and/orpredict the abnormalities in these signals. A low complexity implementation of RFFT can reduce the power consumption in implantable or portable devices.Even though specific algorithms for the computation of the RFFT [8]–[13] have been proposed in the past, these algorithms lack regular geometries to design pipelined architectures. A novel approach to design efficient pipelined architectures for RFFT was presented for the first time in [15]. This architecture is obtained by modifying the radix-2 flow graph to achieve real datapaths and processes 4 samples in parallel. This approach is specific to radix-2 algorithm and was limited to a 4-parallel architecture. Further in [16], a general methodology is presented to derive FFT architectures for both real and complex inputs. Even though the redundant operations are removed from the flow graph, these architectures still compute these samples (denoted as null operations). Due to this, the datapathare complex except for the first two stages. In this paper, we have addressed the limitations of these prior approaches. A generalapproach which can be extended to other radix- $2^n$ algorithms is needed to take advantage of less number of multiplicationoperations in higher radix algorithms.The rest of this brief introduces the scheme by first summarizing FFT and the related works are considered in Section II. Then, in Section III, the proposed scheme is presented. Section IV and V presents a experimental Results and performance analysis to illustrate the effectiveness of the approach. Finally, the conclusions are summarized in Section VI.

## II RELATED WORKS AND THE SERIAL COMMUTATOR (SC) FFT

A type of FFT hardware architectures called serial commutator (SC) FFT. The SC FFT is characterized by the use of circuits for bit-dimension permutation of serial data. The proposed architectures are based on the observation that in the radix-2 FFT algorithm only half of the samples at each stage must be rotated. This fact, together with a proper data management makes it possible to allocate rotations only every other clock cycle. This allows for simplifying the rotator, halving the complexity with respect to conventional serial FFT architectures. Likewise, the proposed approach halves the number of adders in the butterflies with respect to previous architectures.

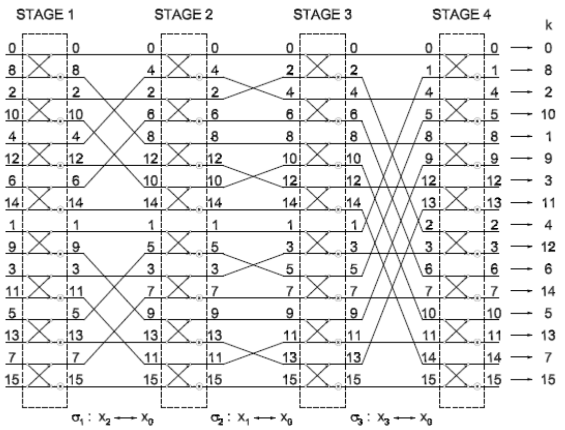

STAGE 1
STAGE 2
STAGE 3
STAGE 4
k
$\sigma_1 : x_2 \longleftrightarrow x_0$
$\sigma_2 : x_1 \longleftrightarrow x_0$
$\sigma_3 : x_3 \longleftrightarrow x_0$

Fig 1: 16-point radix-2 DIF SC FFT

The serial commutator FFT for N = 16 points and radix-2, respectively for DIF and DIT. The architectures consist of n = log2 N = 4 stages that include butterflies, rotators and circuits for data management. Rotators that carry out trivial rotations are diamond-shaped whereas general rotators are represented by a circle. Both butterflies and rotators are marked with 1=2. This means that they require half of the components in conventional butterflies and rotators: butterflies only use a real adder and a real subtracter instead of complex ones, and rotators use two real multipliers and one adder instead of four real multipliers and two adders. The half butterflyand half rotator form the processing element (PE) of the architecture. The circuits for data permutation are circuits for elementary bit exchange. These circuits have already been used for the calculation of the bit reversal. However, this is the first time that this type of circuits are used in an FFT architecture.The data management of the SC FFTs in Figure. The data management is the same for both DIF and DIF cases. Each column in Figure represents the input order to the corresponding stage. The order of arrival is from top to bottom. Therefore, x[0] and x[8] are the first and second inputs to the first stage, respectively. The figure shows that, at all the stages, consecutive samples are operated together in a butterfly. This allows for the use of the PE with half of the resources. This paper has presented the serial commutator (SC) FFT architecture. This architecture is the first FFT to use circuitsto calculate bit-dimension permutation on serial data. This creates a data management that allows for using the theoretical minimum amount of hardware resources for a serial FFT. Compared to previous designs, the SC FFT reduces either the number of rotator, or the number of adders or the memory of the design. A solution for natural I/O has also been presented, which offers comparable results to previous natural I/O FFTs.

## IIIPROPOSED METHOD

Retiming is the technique of moving the structural location of latches or registers in a digital circuit to improve its performance, area, and/or power characteristics in such a way that preserves its functional behavior at its outputs.The technique uses a directedgraph where the vertices represent asynchronous combinational blocks and the directed edges represent a series of registers or latches (the number of registers or latches can be zero). Each vertex has a value corresponding to the delay through the combinational circuit it represents. After doing this, one can attempt to optimize the circuit by pushing registers from output to input and vice versa - much like bubble pushing. Two operations can be used - deleting a register from each input of a vertex while adding a register to all outputs, and conversely adding a register to each input of vertex and deleting a register from all outputs. In all cases, if the rules are followed, the circuit will have the same functional behavior as it did before retiming.

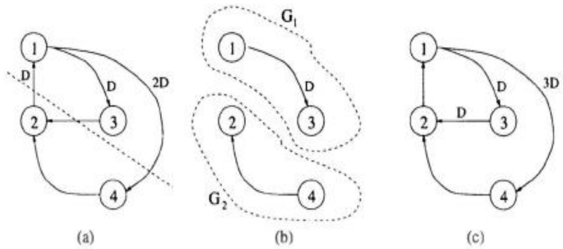


Fig 2: dfg with retiming

Cutset retiming is a useful technique that is a special case of retiming. A *cutset*is a set of edges that can be removed from the graph to create 2 disconnected subgraph. Cutset retiming only affects the weights of the edges in the cutset. If the 2 disconnected subgraphs are labeled $G_1$ and $G_2$, then cutsetretiming consists of adding *k* delays to each edge from $G_1$ to $G_2$ and removing *k* delays from each edge from $G_2$ to $G_1$. For example, a cutset is shown with a dashed line in Fig.(a). The 3 edges in the cutset are 2 → 1, 3 → 2, and 1 → 4. The 2 subgraphs $G_1$ and $G_2$ found by removing the 3 edges in the cutset are shown in Fig.(b). For $k$ = 1, the result of cutset retiming is shown in Fig. (c). The edges from $G_1$ to $G_2$ are 3 → 2 and 1→ 4, and one delay is added to each of these edges. The edge from $G_2$ to $G_1$ is 2 → 1, and one delay is subtracted from this edge.

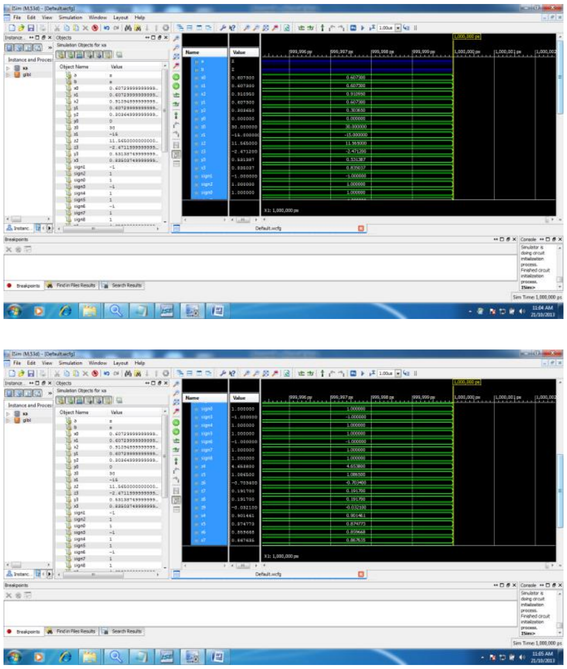
## IV EXPERIMENTAL RESULTS

The proposed circuit are simulated and synthesized by using modelsim and xilinx12.1 which occurs low area than the existing. The experimental results are given in

Table 1 and the simulation results of layout and the waveforms are shown in the fig.4 and fig.5. Then the RTL schematic of the proposed are shown in fig.5.

Table.1 comparison table

| s.no | Parameter | Existing | Proposed |
|---|---|---|---|
| 1 | Slice | 24 | 14 |
| 2 | LUT | 48 | 28 |
| 3 | IOB | 96 | 96 |


ISim (M.53d) - [Default.wcfg]
File Edit View Simulation Window Layout Help
Objects
Simulation Objects for xa
Instance and Process
Breakpoints
Find in Files Results
Search Results
Default.wcfg

```
Object Name   Value
a             z
b             z
x0            0.60729999999999...
x1            0.60729999999999...
x2            0.91094999999999...
y1            0.60729999999999...
y2            0.30364999999999...
y0            0
z0            30
z1            -15
z2            11.5650000000000...
z3            -2.47119999999999...
y3            0.53138749999999...
x3            0.83503749999999...
sign1         -1
sign2         1
sign0         1
sign3         -1
sign4         1
sign5         1
sign6         -1
sign7         1
sign8         1
```

```
Value
z
z
0.607300
0.607300
0.910950
0.607300
0.303650
0.000000
30.000000
-15.000000
11.565000
-2.471200
0.531387
0.835037
-1.000000
1.000000
1.000000
X1: 1,000,000 ps
```

```
Value
1.000000
-1.000000
1.000000
1.000000
-1.000000
1.000000
1.000000
4.653800
1.086500
-0.703400
0.191700
0.191700
-0.032100
0.901461
0.874773
0.859668
0.867635
X1: 1,000,000 ps
```

Console: Simulator is doing circuit initialization process. Finished circuit initialization process. ISim>

Sim Time: 1,000,000 ps

Fig.4 simulation results

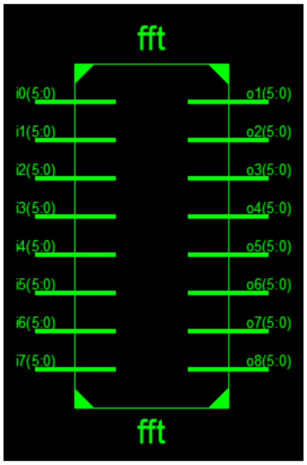


Fig.5 RTL schematic

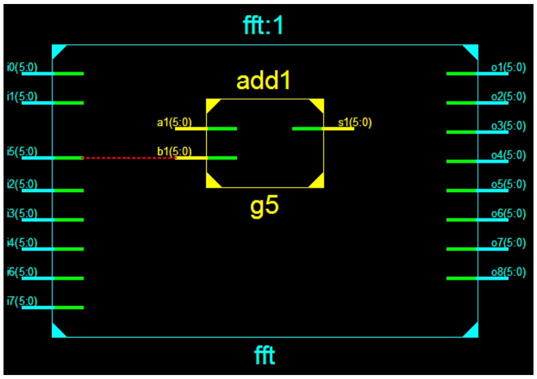


Fig 6: gatelevl=el netlist

## V PERFORMANCE ANALYSIS

The performance of the our proposed system with the existing scheme based on the area and error compensation which was given in Fig. 7

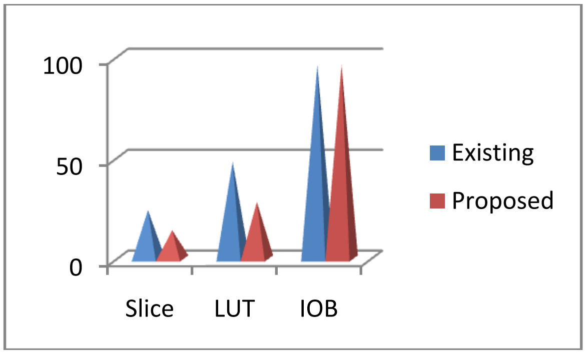


Fig.7 performance analysis of existing and proposed

## VI CONCLUSION

The project has presented the data management binary FFT architecture.This creates a data management and minimum product complexity that allows for using the theoretical minimum amount of hardware resources for a serial FFT with 100% utilization. compared to previous designs ,the proposed FFT reduces either the number of adders or the memory of the design. The solution for natural I/O has also been presented,whichoffers comparable results to previous natural I/O FFTs. Finally experimental results have been obtained to verify the architecture,leading to small area and low power consumption.

## REFERENCE

[1] A. V. Oppenheim, R.W. Schafer, and J.R.Buck, *Discrete-Time Signal Processing*, 2nd ed. Englewood Cliffs, NJ, USA: Prentice Hall, 1998.

[2] S. He and M. Torkelson, “Designing pipeline FFT processor for OFDM (de)modulation,” in *Proc. Int. Symp.Signals Syst.*, Oct. 1998, pp. 257–262.

[3] H. Chi and Z. Lai, “A cost-effective memory-based real-valued FFT and Hermitian symmetric IFFT processor for DMT-based wire-line transmission systems,” in *Proc. ISCAS*, May 2005, vol. 6, pp. 6006–6009.

[4] W. Ko, J. Kim, Y. Park, T. Koh, and D. Youn, “An efficient DMT modem for the G.LITE ADSL transceiver,” *IEEE Trans. Very LargeScale Integr. (VLSI) Syst.*, vol. 11, no. 6, pp. 997–1005, Dec. 2003.

[5] A. Wang and A. Chandrakasan, “Energy-efficient DSPs for wireless sensor networks,” *IEEE Signal Process. Mag.*, vol. 19, no. 4, pp. 68–78, Jul. 2002.

[6] S. F. Reddaway, P. Bruno, P. Rogina, and R. Pancoast, “Ultra-high performance, low-power, data parallel radar implementations,” *IEEEAerosp. Electron. Syst. Mag.*, vol. 21, no. 4, pp. 3–7, Apr. 2006.

[7] Y. Park, L. Luo, K. Parhi, and T. Netoff, “Seizure prediction with spectral power of EEG using cost-sensitive support vector machines,” *Epilepsia*, vol. 52, pp. 1761 1770, Oct. 2011.

[8] W.W. Smith and J. M. Smith, *Handbook of Real-Time Fast Fourier Transforms*. New York, NY, USA: Wiley-IEEE Press, 1995.

[9] H. Ziegler, “A fast fourier transform algorithm for symmetric realvalued series,” *IEEE Trans. Audio Electroacoust.*,vol. AU-20, pp. 353–356, Dec. 1972.

[10] J. B. Martens, “Discrete fourier transform algorithms for real valued sequence,” *IEEE Trans. Acoust., Speech, Sig. Proc.*, vol. ASSP-32, no. 4, pp. 390–396, Apr. 1984.