

# Implementation of 32 bit Discrete Radix-4 FFT Unit

A. Anjana<sup>1</sup>, P. Thirumaraiselvan<sup>2</sup>

<sup>1</sup>Electronics and Communication Engineering, Adhiparasakthi Engineering College, Melmaruvathur, Tamil Nadu, India

<sup>2</sup>Electronics and Communication Engineering, Adhiparasakthi Engineering College, Melmaruvathur, Tamil Nadu, India

## Abstract

In order to perform any operation over a digital signal it needs to be converted from frequency domain to time domain. Fast Fourier Transform (FFT) operation is required at the receiver side to perform signal transformation at higher rate when compared to that of DFT. The implementation of radix 4 FFT is performed in various methods namely butterfly method, Radix- r FFT and split radix FFT. In this paper, the radix -4 FFT is implemented, which is the fastest mode of converting the signal from time domain to frequency domain. Design of radix -4 FFT is performed based on the method of discrete radix -4 FFT. Discrete radix -4 FFT is implemented based on the parallel processing method. Parallel processing method increases the speed of radix 4 FFT unit. Discrete radix-4 FFT of 32 bit input data is simulated and synthesised using “Cadence NClaunch” with 180 nm technology. The estimated delay for 32 bit discrete radix-4 FFT is 27.486 ns with the power dissipation of about 50.29 mW. The proposed discrete radix-4 FFT unit requires 6276 cells.

**Keywords:** Discrete Radix -4 FFT, addition unit, subtraction unit and multiplication unit.

## 1. Introduction

The FFT is used to implement Discrete Fourier Transform at a high speed. FFT is used in various fields such as linear filtering, spectrum analysis, digital video broadcasting and orthogonal frequency division multiplexing [1]. FFT is most widely used in communication based application such as LAN and the upcoming wireless technologies. There are various methods in generating the FFT they are radix -2, radix -4 and split radix. The number of required real addition, subtraction and multiplication units required for manipulating radix 4 FFT output. The FFT unit is manipulated based on the method of Divide and conquer, which divide the computation recursively and generate twiddle factors [2]. In this paper, radix-4 FFT is designed for 32 bit input data with reduced number of arithmetic units. Each arithmetic operation is implemented using conventional logic gates. The method of discrete radix-4 FFT unit reduces the number of arithmetic units required to perform transform of signal from one domain to other domain. The additional unit is performed by ripple carry adder, subtraction is performed using ripple borrow

subtraction and multiplication is performed using array multiplier. This method effectively reduces the number of arithmetic units used in the generation of 32 bit FFT unit when compared to that of butterfly unit [3].

## 2. Discrete radix -4 FFT

In fast Fourier transform algorithms, a butterfly is a portion of the computation that combines the smaller DFT into a larger or breaking a larger into smaller. The design of discrete radix-4 FFT unit is the efficient method to transform the signal from one domain to other domain. Discrete radix- 4 FFT unit reduces the number of arithmetic operations for generating the signal transform [1].

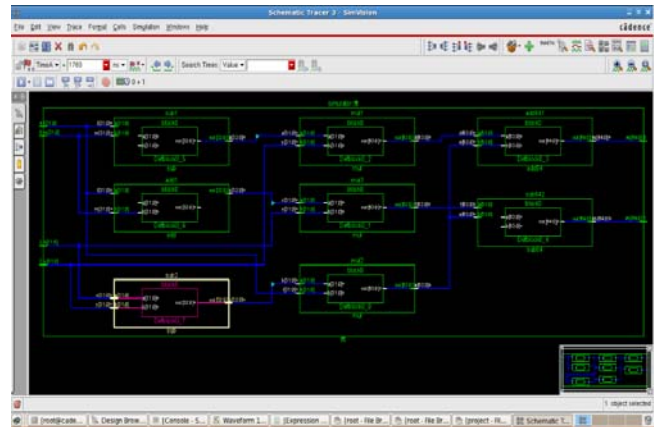


Fig. 1 RTL schematics of 32 bit Discrete Radix – 4 FFT  
Fig.1 shows the RTL schematic diagram of Discrete Radix- 4 FFT unit which has reduced number of arithmetic operations.

This paper, the radix -4 FFT is designed to reduce the delay of manipulation, power and area when compared to that of split radix- 4 FFT unit. Arithmetic units such as adder, multiplier and subtraction units are required to manipulate 32 bit discrete radix 4 FFT unit.

### 2.1 Addition unit

The addition unit in discrete radix – 4 FFT is performed based on the ripple carry adder fashion. This type of adder is called a ripple-carry adder, since each carry bit is rippled to the next of full adder. Ripple Carry Adder (RCA) is the traditional method for addition of input bits such as addend and augend. RCA provides very slow manipulation.

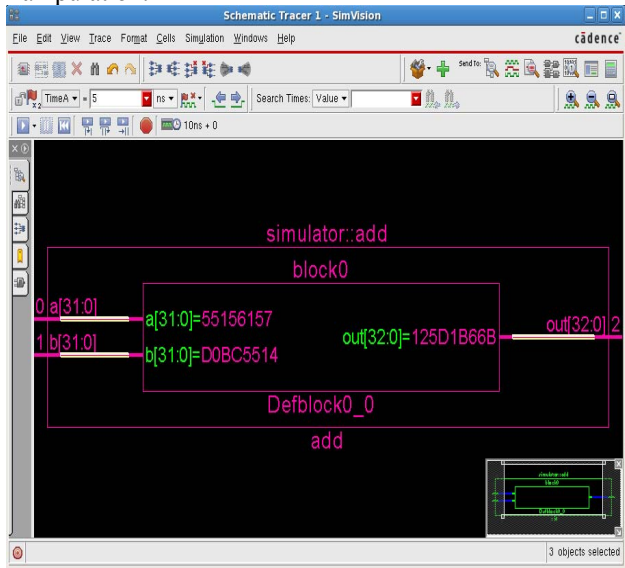


Fig. 2 RTL schematics of 32 bit RCA unit

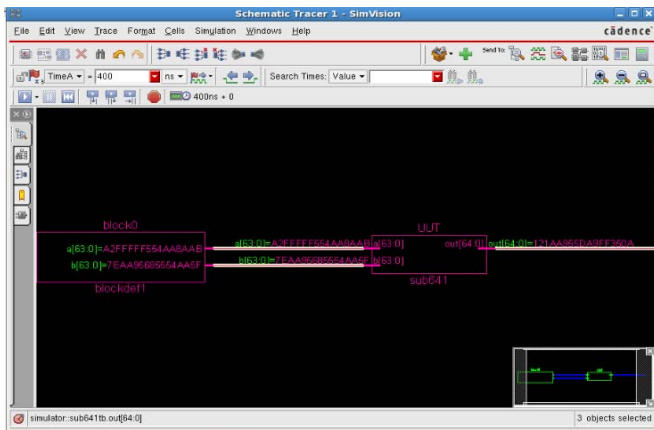


Fig. 3 RTL schematics of 64 bit RCA unit

The area required to store the RCA manipulation is less than the other addition methods. In this paper, the addition unit is designed for 32 bit and 64 bit using RCA fashion. The output of the 64 bit addition unit provides real terms. Fig. 2 Shows the RTL schematic of 32 bit addition unit and Fig. 3 shows the RTL schematic of 64 bit addition unit.

## 2.2 Subtraction Unit

Subtraction unit in 32 bit discrete radix-4 FFT unit is performed by Ripple Borrow Subtraction (RBS) unit. In RBS the borrow term is rippled from one stage of

subtraction to other subtraction stage. RPS is designed for 32 bit and 64 bit input bits for discrete radix- 4 FFT unit.

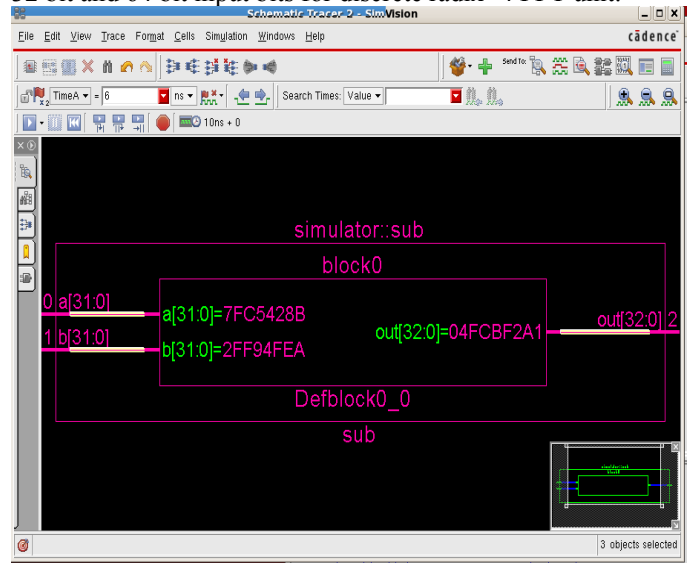
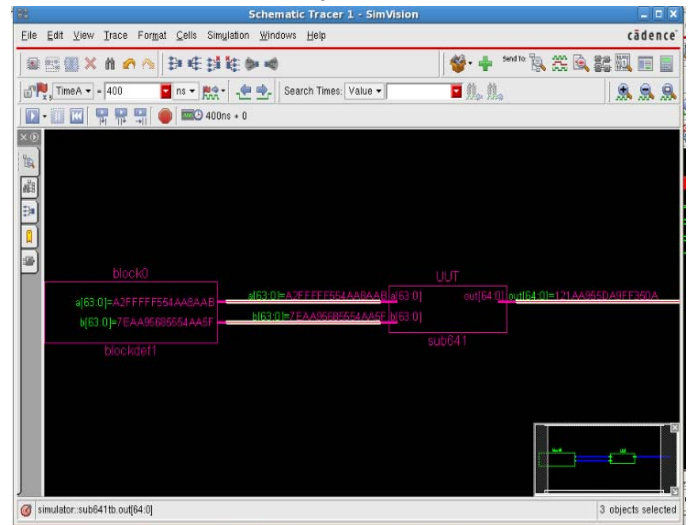


Fig. 4 RTL schematics of 32 bit RBS unit

Fig. 5 RTL schematics of



64 bit RBS unit

Fig. 4 and 5 shows the RBS unit for 32 bit and 64 bit input data. The Output of 64 bit RBS unit of discrete radix- 4 FFT provides the value of imaginary terms.

## 2.3 Multiplication Unit

Multiplication is an essential arithmetic operation for common Digital Signal Processing (DSP) applications, such as filtering and FFT. To achieve high execution speed, parallel array multipliers are widely used. But these multipliers consume more power. Power consumption has become a critical concern in today's VLSI system design.

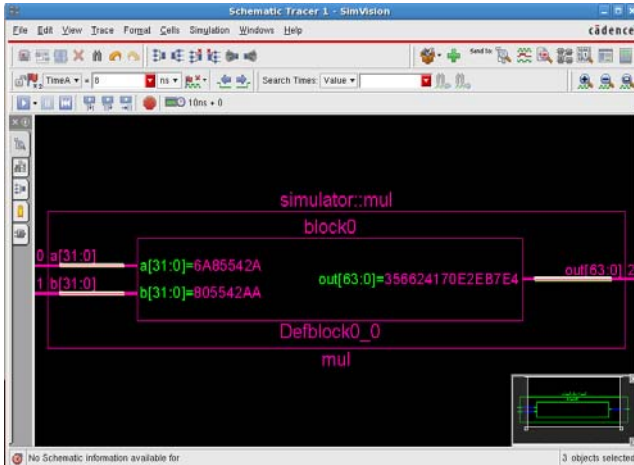


Fig. 5 RTL schematics of 32 bit Multiplication unit

Fig. 5 shows the RTL schematic of 32 bit multiplication unit. In discrete radix- 4 FFT unit the multiplication is performed in array multiplier method. The array multiplication is designed for about 32 bit input vectors with 64 bit output vectors. In discrete radix- 4 FFT unit, multiplication unit is used in the intermediate block for manipulation.

### 3 RESULT AND DISCUSSION

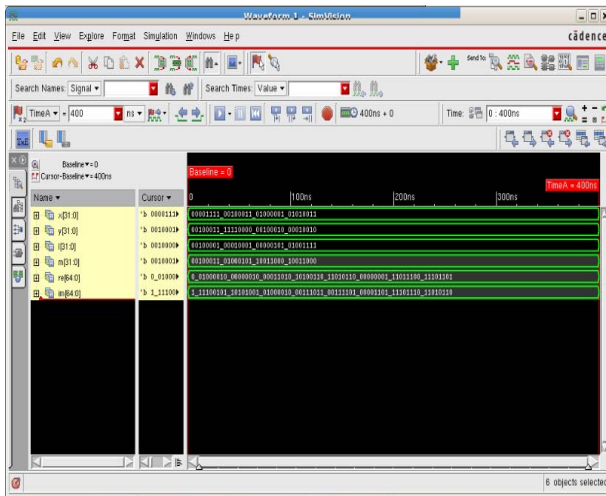


Fig. 6 RTL schematics of 32 bit Discrete Radix- 4 FFT unit

Fig. 6 shows the simulated output waveform for 32 bit Discrete Radix- 4 FFT unit with 32 bit input. This provides the output of real and imaginary values. This proposed Discrete Radix- 4 FFT unit requires very less number of arithmetic units. This requires two adder unit,

three subtraction unit and three multiplication units. The final real and imaginary value output is of 64 bit.

Table 1. Synthesis report of Discrete Radix 4 FFT unit

Table 1. shows the synthesis report comparison for the existing Split Radix-4 FFT unit and proposed Discrete Radix-4 FFT unit. The area, power and delay value are much reduced in the case of discrete radix 4 FFT designed for 32 bit. The synthesis and simulation is performed in ‘Cadence NClaunch’ at about 180nm.

### 4 CONCLUSION

The proposed DR4FFT unit design provides low arithmetic operators for generating the real and imaginary values when compared to the existing slit radix FFT algorithm which requires the multiplication unit of about 72 [3]. On the other hand, for the implemented Discrete Radix- 4 FFT unit it requires three multipliers, and three subtractions and two adders unit totally it requires about 8 arithmetic operation blocks. Based on the input of binary to the Discrete Radix- 4 FFT the output is generated at a smaller time interval of about 27.486 ns. The proposed method uses the arithmetic operations that power up Discrete Radix- 4 FFT with reduced power consumption of about 50.29 mW. Arithmetic operators such as adder, subtractor

Table 1. Synthesis report of discrete radix -4 FFT

Parameters	Existing Split radix- 4 FFT	Proposed Discrete Radix- 4 FFT
Area	-	6276 cells
Power	870 mW	50.29 mW
Timing	21.522 ms	27.486

and multiplier are used which is highly less than that of the existing method. As the result of the use of discrete method in achieving 32 bit Radix -4 FFT area is reduced to about 6276 cells. The proposed method is designed using ‘CADANCE NClaunch’ environment. From the obtained result, Discrete Radix- 4 FFT unit requires very less number of arithmetic operations of about 7 which is much reduced when compared to the existing split-radix -4 FFT unit.

### REFERENCES

[1] Siva Kumar Palaniappan and Tun Zainal Azni Zulkifli (2007) “Design of 16-point Radix-4 Fast Fourier Transform in 0.18µm CMOS Technology”, American Journal of Applied Sciences, pp. 570-575.

- [2] Sudha Kiran G and Brundavani P (2013) “ FPGA Implementation of 256-Bit, 64-Point DIT-FFT Using Radix-4 Algorithm”, International Journal of Advanced Research in Computer Science and Software Engineering, vol.3, issue 9, pp. 126-133.
- [3] Weihua Zheng, Kenli Li, and Keqin Li (2014) “Scaled Radix-2/8 Algorithm for Efficient Computation of Length-  $N=2^M$  DFTs”, IEEE Transactions On Very Large Scale Integration Systems, vol. 62, no. 10, pp. 2492-2503.