International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-9 Issue-3, February 2020

Design and Implementation of FFT IP using Pipelined Hybrid Adder and Distributed Arithmetic Based Complex Multiplier

C.V.Thejashwini, A.Sumathi

Abstract: ***In current inventive technology, latency, power and area are the crucial parameters to outline any kind of the algorithm on FPGA. The fundamental tool used for DSP applications is Fast Fourier Transform. FFT plays a vital role in acquiring the signal characteristics with least use of carrying out parameters. The adder plays an utmost importance. To make the best possible adder design regarding delay and area, various works have been proposed before. In proposed system, a combination different sub adders like Carry Look ahead adder (CLA), Ripple carry adder (RCA), and Carry save adder (CSA) is proposed. This reduces the delay and area but also increases the speed. The hybrid adders is proposed to represent FFT architecture inplace of conventional adders. Hybrid adder will act as a complex adder. Speed multipliers are fundamental parts of DSP systems. Multipliers are complex process and consumes more time. In order to lower the complexity multiplication, various multiplier less method are introduced. An efficient DA based complex multiplier is proposed, inplace of regular multiplier. The pipelining technique is applied only to hybrid adder. The design of Radix-2 FFT for 8 point of FFT, 1024 point of FFT is done, programmed using Verilog language. Using Xilinx 14.5i tool with Spartan 6 kit, Simulation is achieved.***

Keywords: CSA, CLA, Distributed Arithmetic algorithm, FFT algorithm, Pipelined Hybrid adder, RCA

I. INTRODUCTION

Currently, FFT has got the great prominence in the fields of biomedical applications and communication to analyze the signal features. The conversion of original domain in to a frequency domain signal called FFT, which performs frequency analysis on Discrete Time signal. To design FFT Processor, power, area and delay are the crucial parameters. To make full use of hardware resources, algorithm of FFT is exposed. The computers and calculators are used by the engineers, scientists and accountants to solve complex arithmetic problems and also to save the effort along with time. FFT/IFFT is one of the important used blocks used in OFDM/OFDMA system. This FFT algorithm plays a crucial role in the communication field and also signal processing which is involved in Frequency Division Multiplexing signal. Adders, multiplier, subtractors are present in FFT block. The FFT is one of the most widely used algorithms for calculating the discrete Fourier transform (DFT) owing to its efficiency in reducing computation time. The following equation shows the length-*N* forward DFT x (*n*) sequence:

$$X[k]=\sum_{n=0}^{N-1}x(n)e^{(-j2\pi nk)/N} \qquad (1)$$

Where k = 0, 1, N– 1

$$x(n)=(1/n)\sum_{n=0}^{N-1}X[k]e^{(-j2\pi nk)/N} \qquad (2)$$

Where n = 0, 1, N– 1.

One such algorithm is cooley-tukey Radix-r decimation-in-frequency (DIF) FFT, which recursively divides the input sequence into N/r sequence of length r and requires logN stages of computation.

A. RADIX-2 FFT ALGORITHM

Cooley and Tukey presented the first FFT algorithm, which is Radix-2 algorithm. The DFT of a given sequence X[n] can be evaluated using the formula.

$$X(k)=\sum_{n=0}^{N-1}x(n)W_N^{nk} \quad (1)$$

Where K=0, 1….N-1

$$X_1(r)=X(2_m)$$

$$X_2(r)=X(2_m+1) \qquad (2)$$

Where m=0, 1… (N/2)-1
Wn is Twiddle factor.
Then, the N-Point DFT become,

$$X(K)=\sum_{n(even)=0}^{N-1}x(n)W_N^{nk}+\sum_{n(odd)=0}^{N-1}x(n)W_N^{nk} \qquad (3)$$

Properties of twiddle factors are used. They are symmetry and periodicity. The basic butterfly unit is shown in Figure 1.

Revised Manuscript Received on February 05, 2020.
* Correspondence Author
C.V.Thejashwini*, Department of ECE, Adhiyamaan college of Engineering, Tamilnadu, India. Email: saitheja1829@gmail.com
A.Sumathi, Department of ECE, Adhiyamaan college of Engineering, Tamilnadu, India. Email: sumathi_2005@rediffmail.com

$$X[k] = X_1(k) + W_N^k X_2(k) \quad (4)$$

$$X[k+N/2] = X_1(k).W_N^k X_2(k)$$

Where K=0, 1….N/2-1

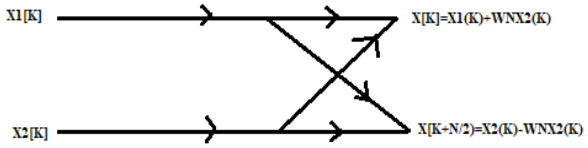

Fig 1: Basic butterfly unit.

If N is a regular power of 2, the same computational procedure can be applied recursively until the N-point DFT is evaluated as a collection of 2- point DFT's. The Radix -2 8 point DIT is shown in Figure 2.

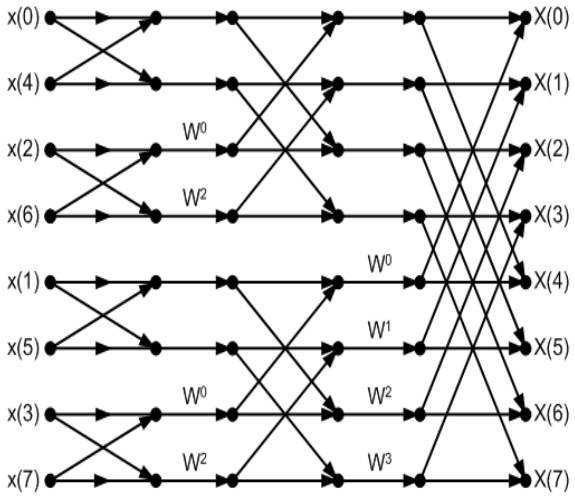

Fig 2: Radix-2 8 point DIT

B. PIPELINED FFT ARCHITECTURE

Before the invention of FFT architecture, it was very tough to process the data in signal processing applications. They developed many different architecture but each FFT architecture has a disadvantage that it needs more memory to store the twiddle factor multiplication. To increase the operation speed, there are types of techniques in signal processing architecture.

1. Parallelism
2. Pipelining

The pipelined block is categorized into 2 types.

1. SDF
2. MDC

C. SINGLE PATH DELAY FEEDBACK (SDF)

SDF FFT architecture has the most efficient utilization memory for pipelined FFT processor. The first half data input is saved in memory. So, the delayed input processed with the remaining input in butterfly unit. The output data fed back to input of butterfly unit through buffer for further processing.

SDF has a great advantage that it requires less memory space. Especially for applications like low power design, this SDF offers several advantage. This is the reason SDF is adopted. SDF block reduces the complex adder by 50% and also produce the output in normal order. The utilization of multiplier remains 50%. The SDF block is used to share delay elements between butterfly inputs and outputs to improve efficiency of hardware.

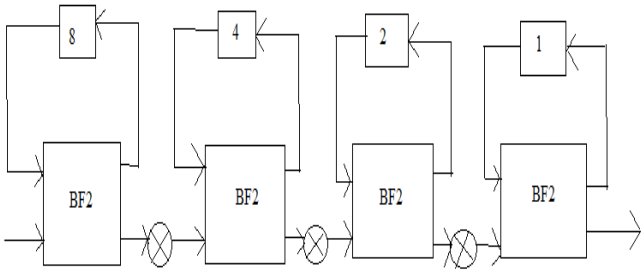

Fig 3: Radix-2 SDF architecture

II. RELATED WORK

Radix-8 booth multiplier will be implemented as a complex multiplier for inplace FFT architecture. This multiplier operates in parallel and requires less adders. Using this multiplier, power is reduced and time requirement also very less. To diminish power dissipation and area, CSLA is used. Overall delay, area and power consumption will be minimized in the proposed method with the support of CSLA adder [1].

For preparing frequency transformation technique, a new pipelined Radix-8 based FFT architecture is introduced. The objective of this paper is to improve speed and to decrease the area, delay and power. In proposed architecture, they combined SDC-SDF FFT, to elevate the processing speed of the architecture and noticed that number of stages is reduced [2].

This project explains the Radix-2 SDF FFT pipelined architecture using VHDL language. Using this divide and conquer technique, Radix-2 algorithm is obtained by integrating twiddle factor approach. Radix-2 principle has the simpler butterfly structure. In this project, they validated the design for 256 point FFT [3].

In brief, this paper purpose is to represent a Radix-2 FFT block which has high speed and also reduces the hardware resources more than 50%. This architecture also produce normal sequence output. It includes 3 SDC stage and SDF stage. By using this, we noticed the reduction in complex multipliers and complex adders. The proposed representation shows the significant reduction in latency [4].

In DSP, the most frequently used algorithm is FFT. In the proposed system, we introduced DIF for the reason being its improved accuracy. Radix-2 algorithm which is an efficient algorithm that multiples 2 signed numbers using 2's complement form. The partial products is reduced by half for this algorithm. Many fast adder exists, but adding fast with less power and low delay is still demanding. To overcome this, we proposed D-latch using CSLA to achieve less delay, Area.

Pipeline architecture can achieve a low latency and a high throughput which are suitable for real time applications. In pipeline FFT architecture, we made use of SDC.

This SDF pipeline FFT architecture is exceptional because it requires less memory space.

Especially for the application like DSP or low power design, SDF has a great advantage. SDF is selected due to this reasons.

For implementation of FFT, speed of operation is very crucial. When the CSLA is synthesized and simulated, area is less but delay is not reduced. In regular CSLA, we used several adders but in this modified method, only one adder is used to lower the delay, area and power consumption [5].

The design of Radix-2 FFT with 128 point is done and proposed adder is to increase the rapidity of FFT architecture. The number of Gates have been reduced by replacing regular CSLA with modified CSLA using D-latch, which offers great advantage of reducing area and increasing the speed. Figure 4 is shown below.

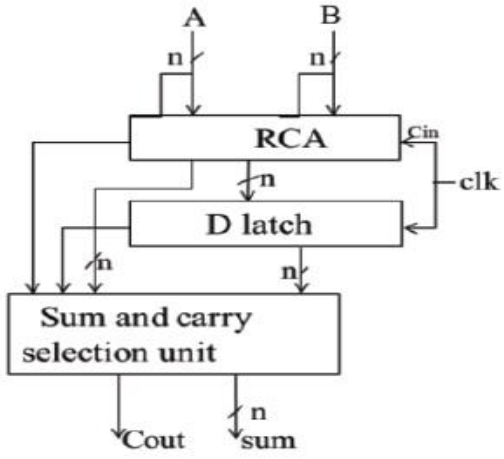

Fig 4: 32 bit CSLA using D-latch

III. PROPOSED FFT ARCHITECTURE

The architecture implemented is R2SDF, shown in figure 5. This Radix-2 SDF has same number of butterfly units and multipliers as in R2MDC approach, but with much reduced memory requirement N-1 registers. It has a minimal memory requirement.

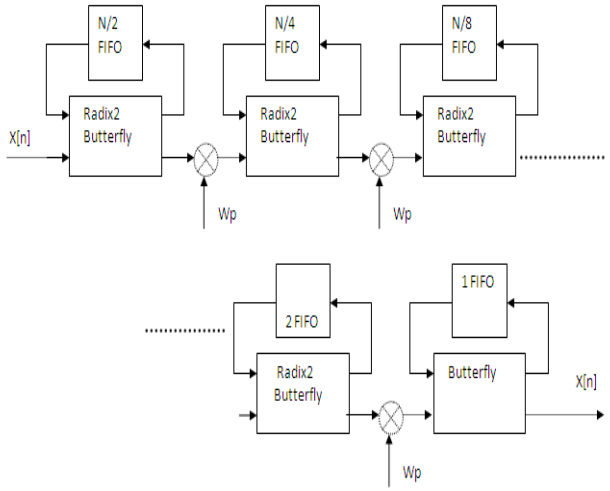

Fig 5: Top level architecture

A. INTERNAL ARCHITECTURE FOR EACH STAGE

The architecture each stage has one FIFO. Twiddle ROM and butterfly unit. In butterfly unit, the addition and subtraction operation are performed. The butterfly starts storing the input values in to FIFO when valid in is high. Butterfly has a internal control logic which consists of three states. The internal architecture is given in figure 6.

In first state, it receives real and imaginary data and stores it in to FIFO till it is full. Once it is full, it will go to next state. In this state, it reads data from FIFO as well as receives input data, processes these two inputs and generate two outputs. In the second state, the addition of inputs and subtraction of inputs will be performed. The subtraction results of inputs are kept in FIFO and addition of inputs are sent out as an output.

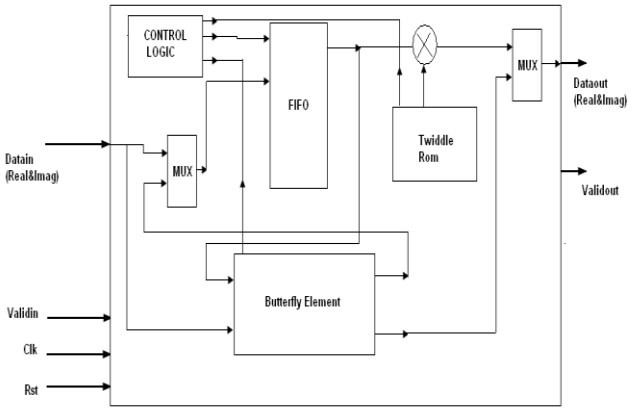

Fig 6: Internal Architecture for each stage of FFT

Depending on stage after receiving input data, state is incremented to third state. In this third state, the subtraction results are stored in the FIFO in precious state are sent to next state to perform twiddle factor multiplication. If the valid in high, the received data is kept in the FIFO again.

The Real output and imaginary output (Data out) are connected as an inputs to the next stages and the valid out output signals acts as valid in input signals to the next stage. This is how all the 10 stages are connected to form a complete 1024 point FFT. This architecture is entirely scalable to tune any size FFT.

B. ARCHITECTURE FOR 8 POINT OF FFT

The architecture for 8 point FFT is shown in figure 7.

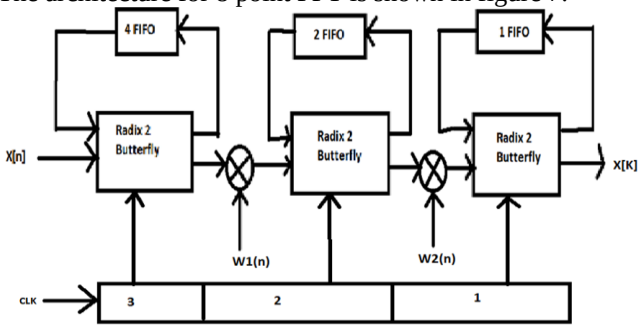

Fig 7: Block diagram for 8 point of FFT

C. ARCHITECTURE FOR 1024 POINT FFT

The architecture for 1024 point is shown in the figure 8.

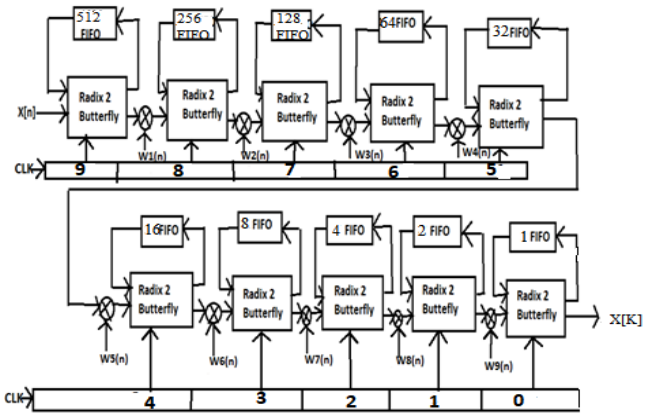

Fig 8: Block diagram for 1024 point of FFT

IV. ADDERS

The building block of ALU is Adder. Adders are generally found in microprocessor design. Increase in speed of adder will enhance the microprocessor performance.

A. RIPPLE CARRY ADDER

RCA are easily understood and more compact digital adders to generate carry and sum. Figure 9 shows the block diagram for RCA. Full adders are building blocks of RCA. RCA propagates the carry from LSB to MSB. This may not happen to all the input conditions we give, but there do exist some input combinations where a carry i.e. coming from the LSB stage may have to ripple through all the different stages and have to go to the final stage to generate sum and carry. This RCA is easy but delay is more because it should wait for the each bit of carry that is calculated by the existing adder.

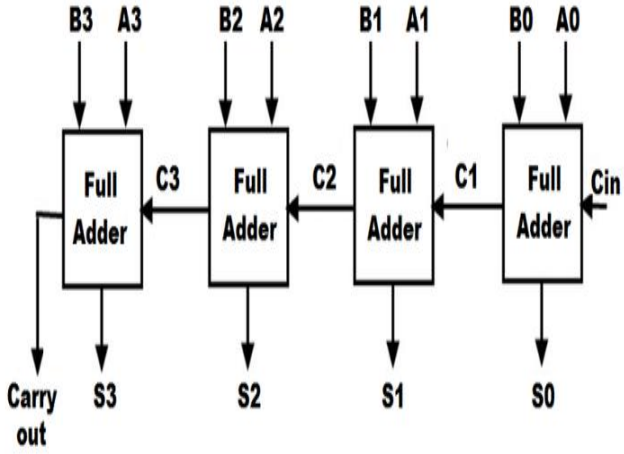

Fig 9: 4 bit RCA

B. CARRY LOOK AHEAD ADDER

CLA is a fast adder using digital logic to calculate the carry signal in advance from the input signal. Figure 10 shows the block diagram for CLA. In CLA, carry does not wait. Depending on input signal, carry generates the input. Due to this, delay is reduced. The look ahead generation fastens addition by eliminating ripple carry adder. In binary adder, each stage depends upon the carry generated by previous stage. It creates a lot of delays when we want to add two large binary number. So, in CLA, carry is generated based on input signals, due to this delay is reduced.

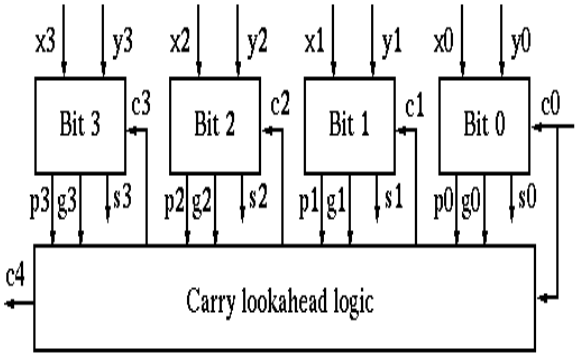

Fig 10: 4 bit CLA

C. CARRY SAVE ADDER

CSA is similar to full adder, but architecture is different. CSA has 3 inputs and process 2 outputs. The operation and principle of CSA is based on formulae.

$$A+B+C=2D+E$$

Essentially it takes 3 inputs which are three numbers, produces D and E as output. D is carry bit and S is the sum bit. CSA also called Wallace tree multiplier. It will go much faster than RCA. Both RCA and CSA have similar qualities. CSA bits will work in parallel manner. The carry does not propagate all the way, instead it is saved for future. So, CSA works faster than any adder. The figure 11 represents CSA.

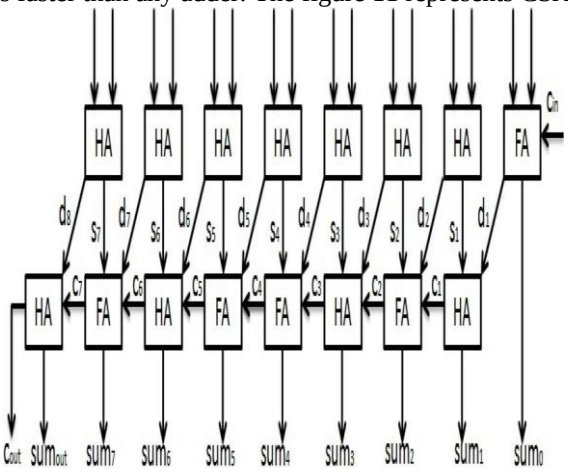

Fig 11: 8 bit CSA

V. COMPLEX ADDER

Complex adder is combination of two regular adder to perform addition. Hybrid adder will act as a complex adder.

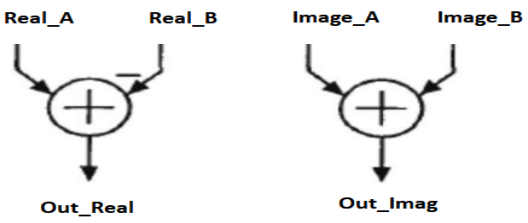

Fig 12: Complex Adder

A. PROPOSED HYBRID ADDER

In FFT, both complex addition and complex multiplication are included. The ideal way to enhance the rapidity of FFT is by using less delay adder inplace of normal adders. The different adders are joined together to form a hybrid adder. So, the delay can be reduce with the selection of adders. Here, 16 bit hybrid adder is used, which is the combination of different adder. They are 4 bit CLA, 4 bit RCA, 8 bit CSA. Once we obtain results for FFT, we can insert pipeline registers to improve speed. The block diagram for proposed hybrid adder is shown in Figure 13.

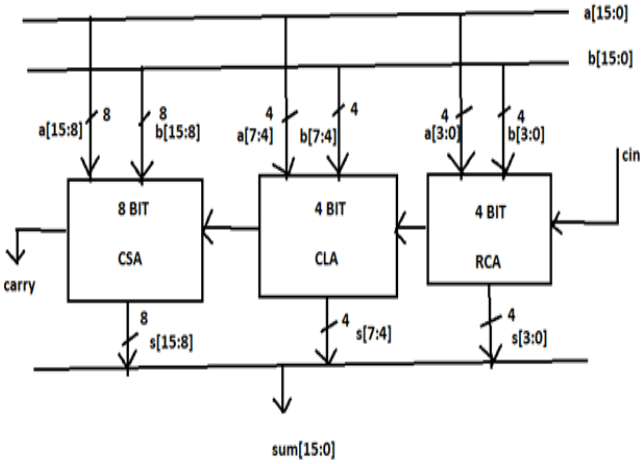

Fig 13: Proposed hybrid adder

B. PIPELINED HYBRID ADDER

To enhance high throughput of the system and for better performance, pipelining technique is used. Pipelining is done by breaking down the path in to several sub path. Between sub paths, pipelined registers are inserted to improve speed. The objective is to increase throughput and to achieve low latency and cost.

The following definitions are necessary to understand pipelining

1. Throughput = (no. of usable outputs) / (unit time)
2. Latency = Time delay from valid inputs provided until outputs valid.
3. Cost = Area + Power.

The block diagram for pipelined hybrid adder is shown in figure 14. Pipelined registers are applied to a hybrid adder circuit.

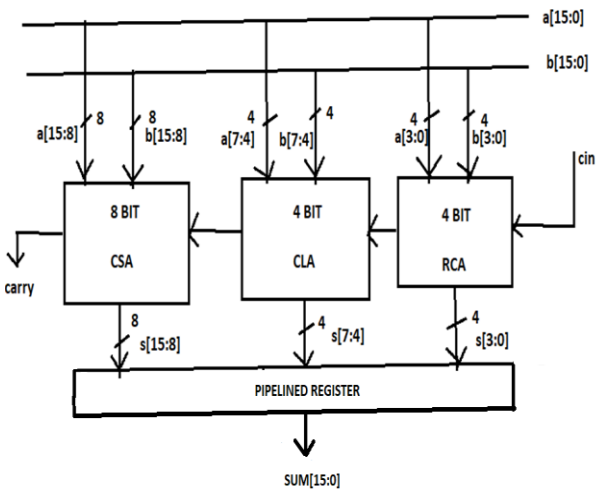

Fig 14: Pipelined hybrid adder

VI. COMPLEX MULTIPLIER

A complex multiplier consists of 2 regular adder and 4 regular multiplier, which is shown in figure 15. Instead of using the actual multiplier and adder blocks, we introduce a multiplier and adder less complex multiplier using the concept of Distributed Arithmetic (DA).

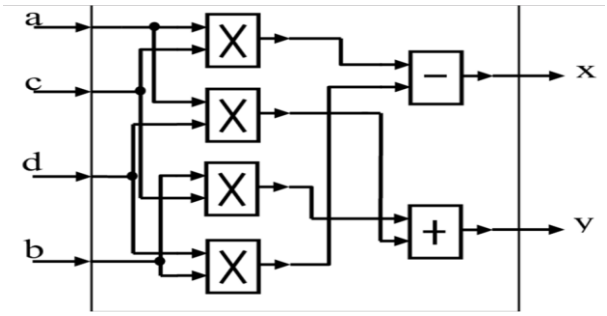

Fig 15: Complex Multiplier

The method for DA based complex multiplication can be summarized as,

$Z_R+j\ Z_I= (B_R+j\ B_I)*(T_R+j\ T_I)$ (1)

Where $Z_R=B_RT_R-B_IT_I$

$Z_I=B_RT_I+B_IT_R$

It shows that 4 real multiplication and 2 two addition are required to compute ZR and ZI. But these equations can be considered as multiply and accumulate operation.

$$y = \sum_{k=1}^{k} C_k X_k \qquad (2)$$

Let, C_k are fixed coefficients and X_k are the input words. If X_k is M-bit fractional number in 2"s complement form then it can be expressed in following form

$$X_k = -b_{k0} \sum_{m=1}^{M-1} b_{k0} 2^{-m} \qquad (3)$$

A. DISTRIBUTED ARITHMETIC BASED COMPLEX MULTIPLIER

In DSP, speed of multiplication operation has a utmost importance. Generally, multipliers are time consuming and complex process. In order to reduce the complexity, several multiplier less method were introduced. In recent years, DA is a trending architecture. The multiplier is not necessary, instead it is implemented based on LUT, DA based complex multiplier are implemented. The distributed arithmetic based complex multiplier is presented below.

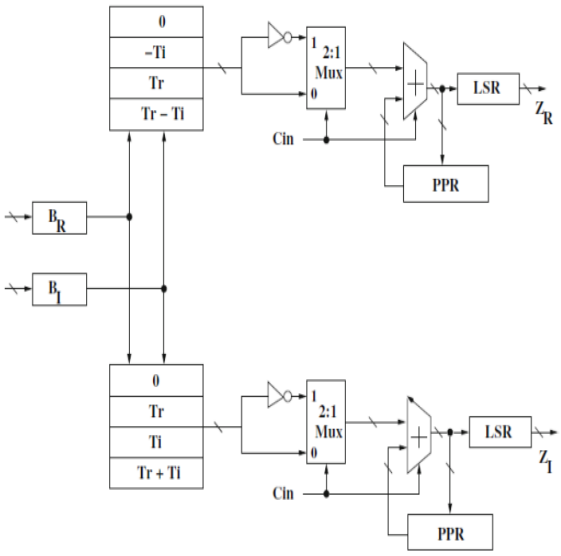

Fig 16: DA based complex multiplier

The real part and imaginary parts of incoming words BR and BI are stored in two 8 bits wide "parallel in serial out "register. Shifting is carried out starting from LSB to MSB. Each output bit of two registers is used as address lines of the ROM. The ROM stores pre calculated outcomes for both Z_R and Z_I. The size of each ROM is 4x8. The input to the 2:1 MUX is directly fed from the output of ROM and other input to MUX is inverted. Input and output bit width for MUX is also 8 bits. The selection line of MUX is, signal Cin =0, till the MSB arrives at output. It Cin=1, it selects the inverted output from ROM and it is added to the value saved in the PPR which also performs 1-bit right shift operation. Finally, the output is fetched from left shift register.

VII. EXPERIMENTAL RESULTS

Efficient FFT implementation using pipelined hybrid adder architecture is implemented in Verilog, simulated using Xilinx ISE 14.5i. The RTL defines the digital portions of the design. It actually a representation of translation of our logic to a digital circuit.

A. PIPELINED HYBRID ADDER

.RTL shows the representation of our design like AND gates, OR gates, adders, multipliers. RTL design for pipelined hybrid adder is shown in figure 17. Technology schematic for pipelined hybrid adder is shown in figure 18. Using Xilinx ISE 14.5i, simulation is achieved, shown in figure 19.

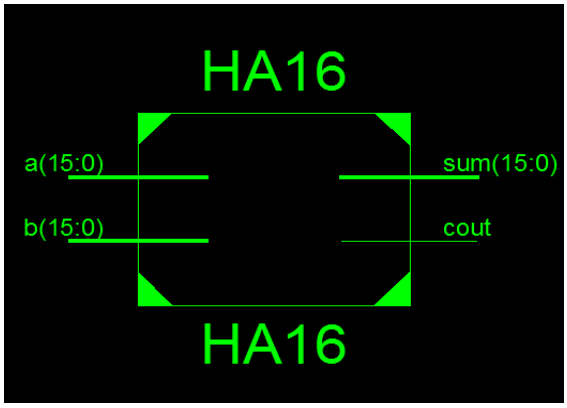

Fig 17: RTL schematic view for hybrid adder

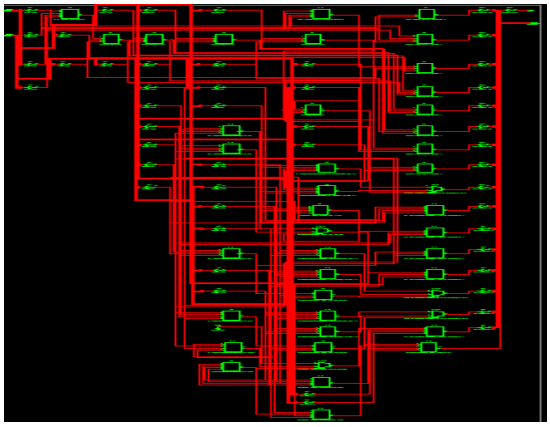

Fig 18: Technology schematic for pipelined hybrid adder

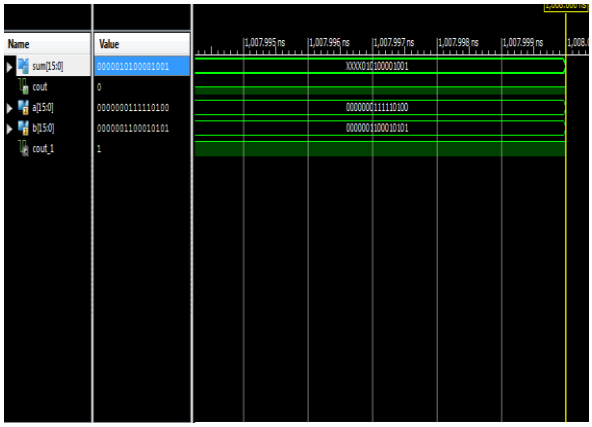

Fig 19: Simulation output for pipelined hybrid adder

B. DISTRIBUTED ARITHMETIC BASED COMPLEX MULTIPLIER

RTL design for Distributed Arithmetic based complex multiplier is shown in figure 20. Technology schematic shows the representation of our design interms of LUTs, buffers, I/Os and carry logic, which is given in figure 21. Figure 22 shows the simulation output for DA based complex multiplier.

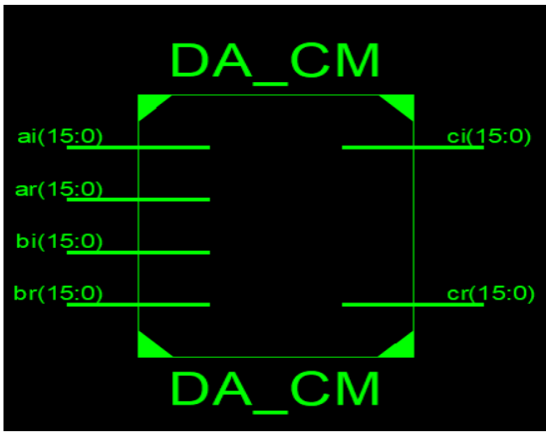

Fig 20: RTL schematic view for DA based complex multiplier

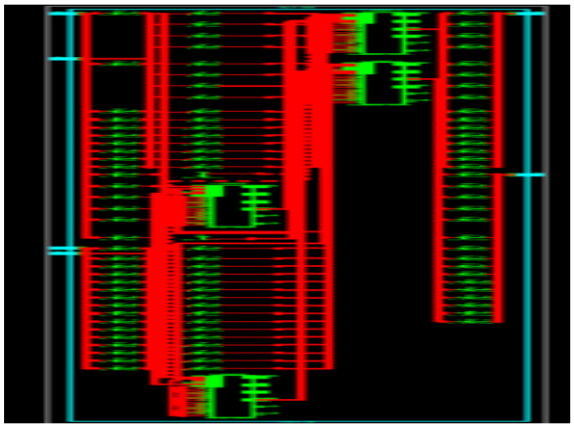

Fig 21: Technology schematic view for DA based complex multiplier

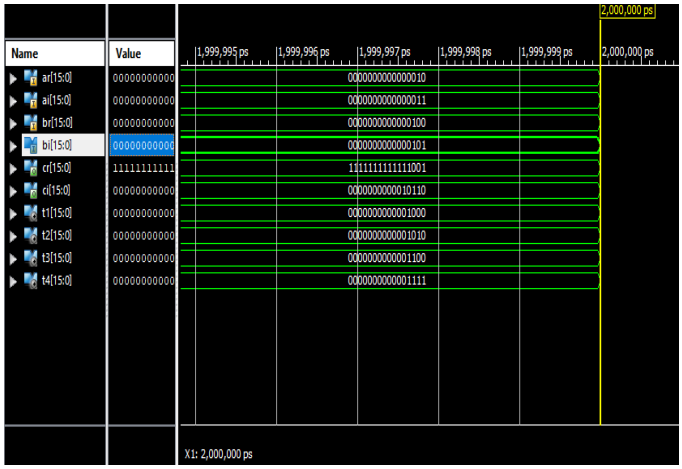

Fig 22: Simulation output for DA based complex multiplier

C. 8 POINT FFT

RTL schematic view, view technological schematic diagrams and the output waveforms of 8 point FFT is shown below.

Fig 23: RTL schematic view for 8 point FFT

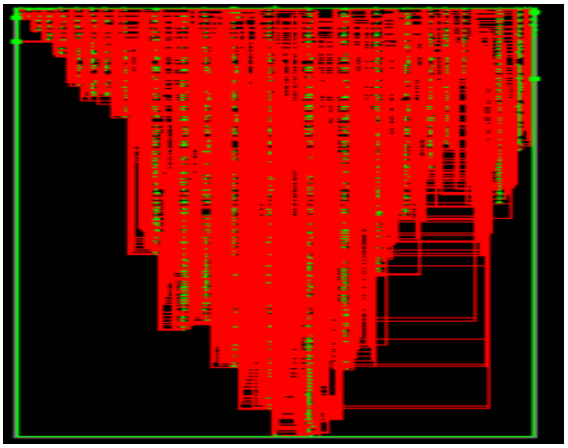

Fig 24: Technology schemativ for 8 point FFT

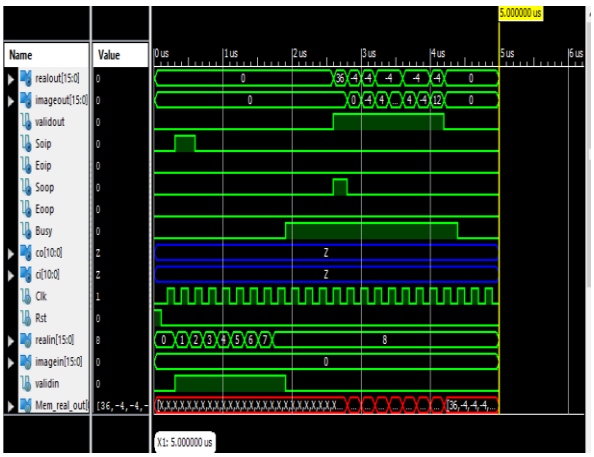

Fig 25: Simulation for 8 point FFT

D. 1024 POINT FFT

RTL schematic view, view technological schematic diagrams and the output waveforms of 8 point FFT is shown below.

Fig 26: RTL schematic view for 1024 point FFT

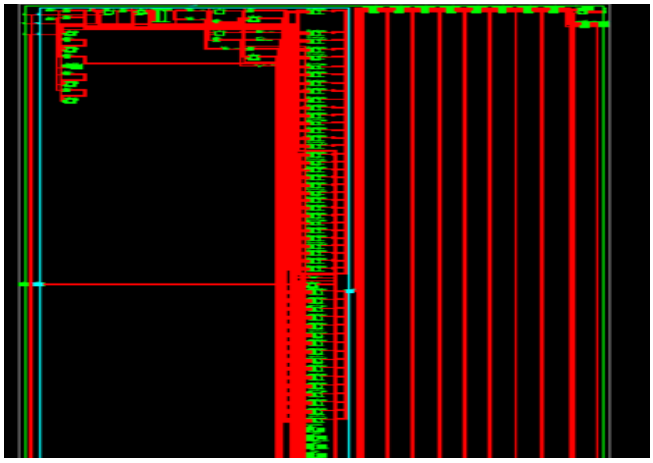

Fig 27: Technology schematic for 1024 point FFT

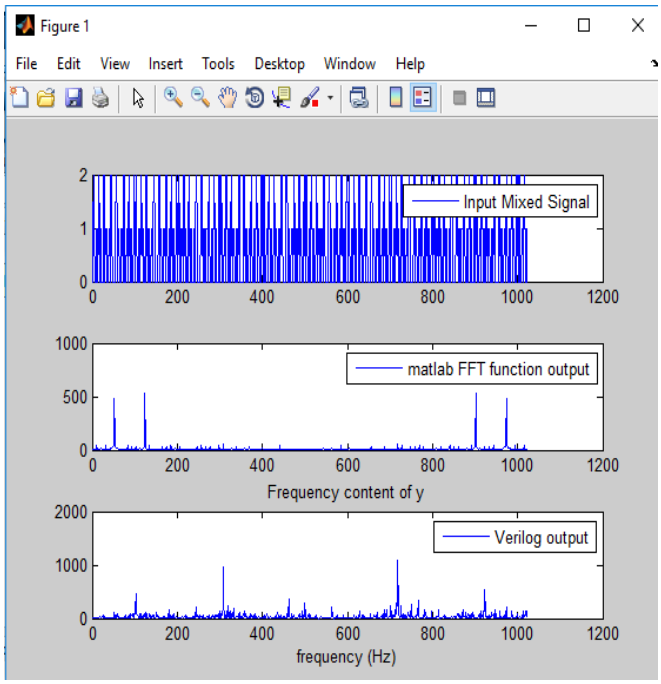

Fig 28: Matlab output for 1024 point FFT

This file also gives us input in Verilog format. But have to get the data required for our FFT to be given for Verilog Test bench. FFT_input.txt is the text file where the data is updated. Copy all the contents present in fft_input.txt and paste it in Verilog 1024 point of FFT test bench. Then RUN the file. ModelSim opens. Export real and imaginary values to text files. Copy those commands and run it in ModelSim. Then format Imag_1024 and Real_1024 text outputs shown in Fig 29 and Fig 30. Open MATLAB and run the test file for 1024 point FFT. This will show the updated plot, Fig 31.

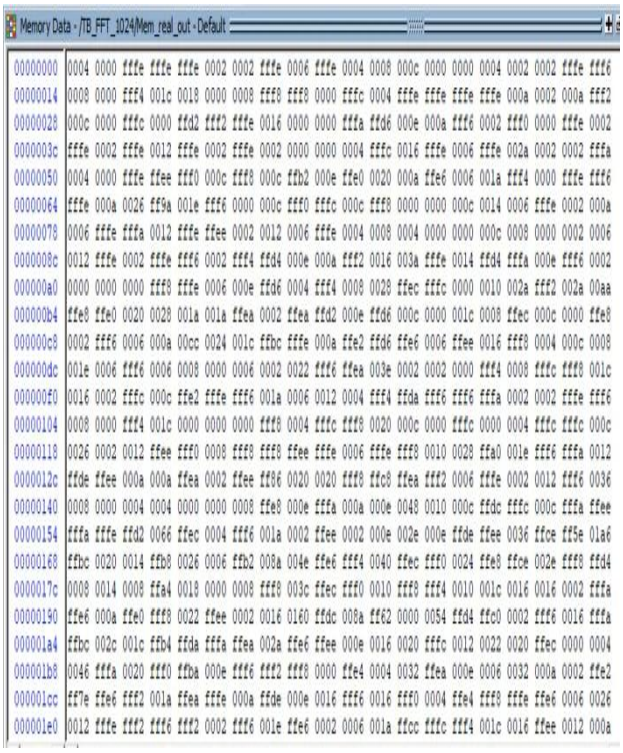

Fig 29: Memory list for Real output

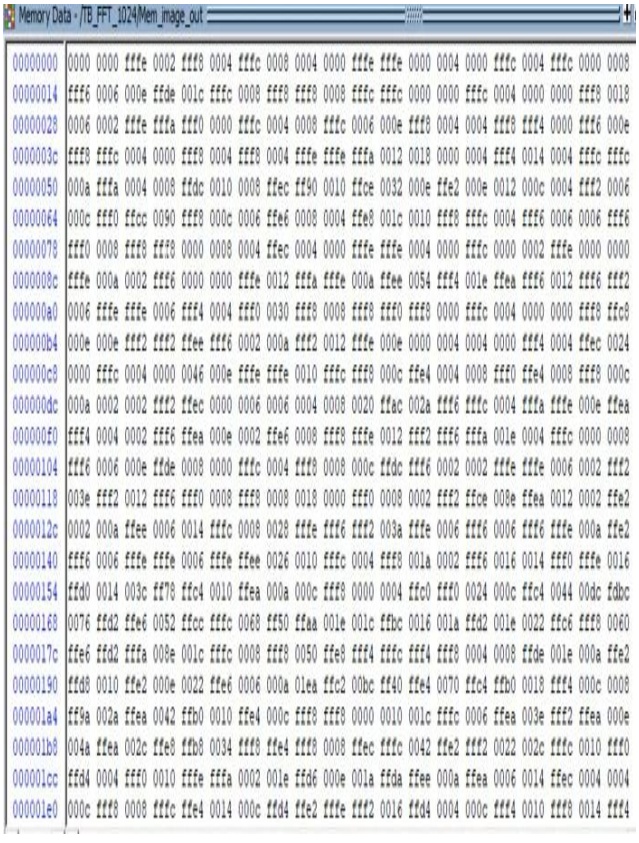

Fig 30: Memory list for Imaginary output

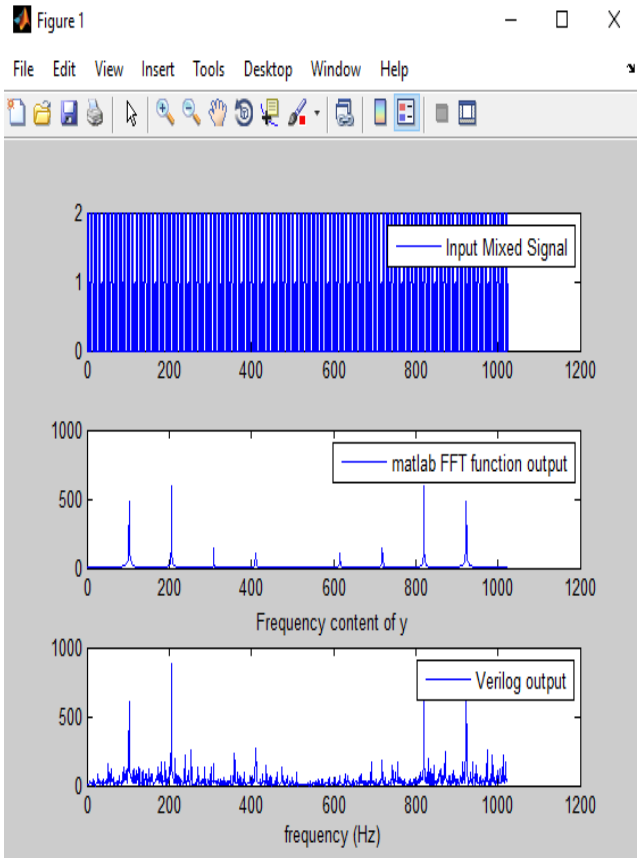

Fig 31: Final output of 1024 point FFT

Finally, the MATLAB output and Verilog output frequency is matched. For 8 point FFT, We cannot do frequency signal match, as it is too less point to represent any frequency. So, 8 point FFT is meant to compare with MATLAB output for output matching in terms of numerical. For higher point like 1024, 2048, 4096 we cannot do numerical match. Hence, we do frequency signal match.

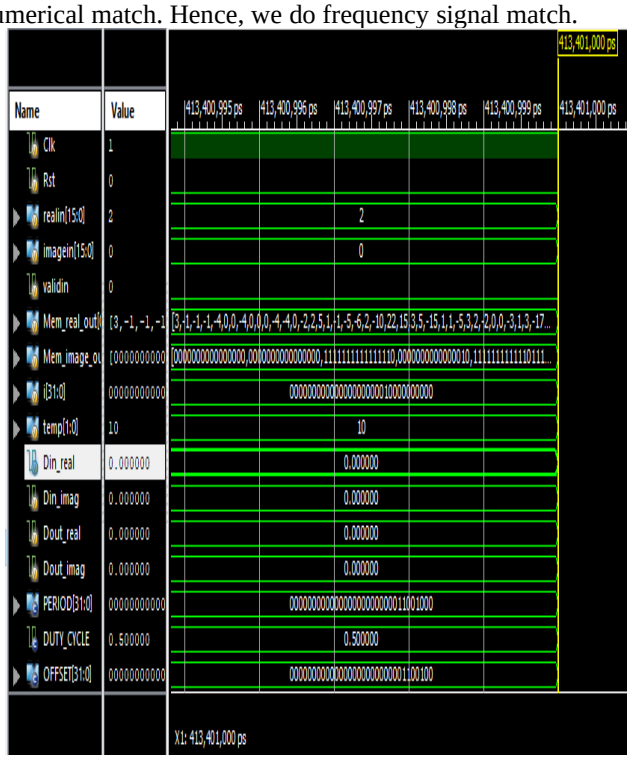

Fig 32: Simulation output for 1024 point FFT

The design summary utilization of Adders and Multipliers is shown. Table 1 gives the brief summary on Memory,Latency and Hardware resources used in different adders. The total number of LUTs and occupied slices is found to be 39 and 24 respectively. It is clear from the Table II that the LUTs is less than regular multiplier. In Table III, the total number of LUTs, Flip flops, Frequency and delay is reduced.

Table I: Performance Comparison of Adders

ADDER TYPE	NO.OF.LUTs	NO.OF. OCCUPIED SLICES	DELAY(ns)
CARRY SELECT ADDER	44	31	12.210
PROPOSED HYBRID ADDER	41	28	10.356
PIPELINED HYBRID ADDER	37	22	8.197

Table II: Performance Comparison of Multipliers

MULTIPLIER TYPE	NO.OF.LUTs	DELAY (ns)
WALLACE MULTIPLIER	45	15.825
BOOTH MULTIPLIER	37	12.583
PROPOSED DISTRIBUTED ARITHMETIC BASED COMPLEX MULTIPLIER	29	11.849

Table III: Comparison of existing system and proposed system fpr Spartan 6 FPGA

TARGET FPGA	LUT	FLIP FLOPS	FREQUECY	DELAY(ns)
Xc6slx25t-3cgs 324 Existing system(128 point)	1074	338	332 MHz	39
Xc6slx25t-3cgs 324 Proposed system(1024 point)	3182	1195	110 MHz	6.226

VII. CONCLUSION

In this paper, pipelined hybrid adder and DA based complex multiplier is proposed. 1024 point FFT architecture has also shown the significant reduction in the delay.

This type of FFT architecture can be used in major of the communication systems like Wireless modems, LTE when compared with existing system. The area and power of this proposed architecture is less.

VIII. FUTURE SCOPE

The proposed block is designed for 16 bit inputs. It can be further extended for 32 bit, 64 bit, 128 bit inputs and can also validate FFT design for 2048, 4096 point respectively. Even other design can be tried for even better analysis. In fact by combining adders including technology, used to implement a suitable less delay can be achieved.

REFERENCES

1. Kavitha M.V, S,Ranjiyha, Dr. S.G.Hiremath, Dr.Suresh H.N,"A Novel RTL Architecture for FPGA Implementation of 32-point FFT for High Speed Applications" IOSR-JCE,ISSN:2278-0661, Volume 19, Issue 2, Ver.II(Mar 2017).
2. J.Manikandan, M.Manikandan "VLSI based pipelined Architecture for Radix-8 combined SDF-SDC FFT" IJST, ISSN: 0974-5645, Vol. 9(28), July 2016.
3. Magandeep Kaur, Pragati Kapoor,"Analysis of $r2^2$ SDF pipeline FFT architecture in VLSI" IJESAT, ISSN:2250-3676, Vol. 2, Issue-3,555-560.
4. Hiremath deepika, rajeshwari.B," SDC-SDF architecture for pipelined Radix-2 FFT implementation" 2016 IEEE Region 10 conference..
5. Athkuri Mana, R.Ramavaraprasad,"Carry select adder pipelined Architecture for FFT" IJARCET, Volume 5, Issue 11, Nov 2016.

AUTHORS PROFILE

C.V.Thejashwini received her B.E degree in Eletronics and Communication Engineering from Adhiyamaan College of Engineering, India, in 2016. She is pursuing M.E VLSI design in Electronics from Adhiyamaan College of Engineering, India. Her research interests include low power techniques, Digital signal processing and VLSI

A.Sumathi received PhD degree in Information and Communication Engineering from Anna University, Chennai, India in 2009; ME degree in Applied Electronics from Anna University, Chennai, India in 2004 and BE degree in Electronics and Communication Engineering from Bharathiar University, Tamilnadu, India in 1994. She is now working as pro department of Electronics and Communication Engineering, Adhiyamaan College of Engineering, Hosur, Tamilnadu. She has published more than 25 papers in various international Journals and conference proceedings.