Design and Implementation of VLSI Architecture for A generalized Mixed-Radix (GMR) algorithm of FFT

Badukolu.Yadagiri
Post Graduate Scholar , Indur institute of Engineering & Technology, Siddipet

A.Gopala Sharma
Professor , Indur institute of Engineering & Technology, Siddipet

Abstract - Digital signal processing is one of the core technologies, in rapidly growing application Areas, such as wireless communications, audio and video processing and industrial control. The number and variety of products that include some form of digital signal processing has grown dramatically over the last few years. DSP has become a key component, in many of the consumer, communications, medical and industrial products which implement the signal processing using microprocessors, Field Programmable Gate Arrays (FPGAs), Custom ICs etc.DSP techniques have been very successful because of the development of low-cost software and hardware support. For example, modems and speech recognition can be less expensive using DSP techniques. DSP processors are concerned primarily with real-time signal processing. Real-time processing requires the processing to keep pace with some external event, whereas non-real-time processing has no such timing constrain Fast Fourier transform (FFT) has an important role in many digital signal processing (DSP) systems. E.g., in orthogonal frequency division multiplexing (OFMD) communication systems, FFT and inverse FFT are needed. Today, various FFT processors, such as pipelined or memory-based architectures, have been proposed for different applications.In this paper we are designing Radix-2 FFTAlgorithm and Radix-4 FFTAlgorithm of

A system by using FFTcontroller . However, for long-size FFT processors, such as the 2048-point FFT, For the memory-based processor design, minimizing the necessary memory size is effective for area reduction since the memory costs a significant part of the processor. We proposed VLSI architecture for a generalized mixed-radix (GMR) algorithm of FFT.

Key Words — Radix-2 FFT Algorithm,Radix-4 FFT Algorithm,Controller

I. INTRODUCTION

In the fields of communications, signal processing, and in electrical engineering more generally, a **signal** is any time-varying or spatial-varying quantity.In the physical world, any quantity measurable through time or over space can be taken as a signal. Within a complex society, any set of human information or machine data can also be taken as a signal. Such information or machine data (for example, the dots on a screen, the ink making up text on a paper page, or the words now flowing into the reader's mind) must all be part of systems existing in the physical world – either living or non-living.Despite the complexity of such systems, their outputs and inputs can often be represented as simple quantities measurable through time or across space. In the latter half of the 20th century, electrical engineering itself separated into several disciplines, specializing in the design and analysis of physical signals and systems, on the one hand, and in the functional behavior and conceptual structure of the complex human and machine systems, on the other. These engineering disciplines have led the way in the design, study, and implementation of systems that take advantage of signals as simple measurable quantities in order to facilitate the transmission, storage, and manipulation of information.
In a communication system, a transmitter encodes a message into a signal, which is carried to a receiver by the communications channel. For example, the words "Mary had a little lamb" might be the message spoken into a telephone. The telephone transmitter converts the sounds into an electrical voltage signal. The signal is transmitted to the receiving telephone by wires; and at the receiver it is reconverted into sounds.In telephone networks, signaling, for example common channel signaling refers to phone number and other digital control information rather than the actual voice signal.
Signals can be categorized in various ways. The most common distinction is between discrete and continuous spaces that the functions are defined over, for example discrete and continuous time domains. Discrete-time signals are

often referred to as time series in other fields. Continuous-time signals are often referred to as continuous signals even when the signal functions are not continuous; an example is a square-wave signal.
A second important distinction is between discrete-valued and continuous-valued. Digital signals are sometimes defined as discrete-valued sequences of quantified values, that may or may not be derived from an underlying continuous-valued physical process. In other contexts, digital signals are defined as the continuous-time waveform signals in a digital system, representing a bit-stream. In the first case, a signal that is generated by means of a digital modulation method is considered as converted to an analog signal, while it is considered as a digital signal in the second case.
The Fourier transform (FT) has been widely used in circuit analysis and synthesis, from filter design to signal processing, image reconstruction, stochastic modeling to non-destructive measurements. The FT has also been widely used in electromagnetic from antenna theory to radio wave propagation modeling, radar cross-section prediction to multi-sensor system design. For example, the split-step parabolic equation method (which is nothing but the beam propagation method in optics) has been in use more than decades and is based on sequential FT operations between the spatial and wave number domains. Two and three dimensional propagation problems with non-flat realistic terrain profiles and inhomogeneous atmospheric variations above have been solved with this method successfully.

II. DESCRIPTION OF FOURIER SERIES

According to the theory developed by Fourier, any periodic function F(t), with period T, may be represented by an infinite series of the form

$$F(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} (a_n \cos n\omega_T t + b_n \sin \omega_T t)$$

Where the coefficients a_0, a_n, and b_n for a given periodic function F(t) are calculated by the formulas

$$\omega_T = \frac{2\pi}{T}$$

$$a_0 = \frac{2}{T}\int_0^T F(t)\,dt$$

$$a_n = \frac{2}{T}\int_0^T F(t)\cos n\omega_T t\,dt \qquad n = 1, 2 \ldots$$

$$b_n = \frac{2}{T}\int_0^T F(t)\sin n\omega_T t\,dt \qquad n = 1, 2 \ldots$$

It should be noted that the first coefficient a_0 is twice the average of the function F(t) over one period. This series is called the Fourier series and the coefficients are called the Fourier coefficients.
The Fourier series expansion of a continuous and periodic waveform provides a means of expanding a function into its major sine / cosine or complex exponential terms. These individual terms represent various frequency components which make up the original waveform.
A line graph of the amplitudes of the Fourier series components can be drawn as a function of frequency. Such a graph is called a spectrum or frequency spectrum.
The periodic time T defines completely which spectral components occur in the spectrum as $f_0 = 1 / T$. The component f_0 is called the fundamental frequency or first harmonic. All the higher components are multiples of f_0 and are called the higher harmonics. The smallest spacing that can occur between frequency components in the spectrum is $f_0 = 1 / T$.
By using Euler's formula to derive complex expressions for sin(t) and cos(t), and substituting these into the Fourier series it can be shown that the complex form of the Fourier series is

$$F(t) = \sum_{n=-\infty}^{\infty} c_n e^{in2\pi f_0 t}\,dt$$

$$c_n = \frac{1}{T}\int_{-\frac{T}{2}}^{\frac{T}{2}} F(t)\, e^{-in2\pi f_0 t}\,dt \qquad n = 0, \pm 1, \pm 2 \ldots$$

Where the coefficients c_n are the complex Fourier coefficients.

Description of ***Fourier transforms:***

The transformation from the time domain to the frequency domain (and back again) is based on the Fourier transform and its inverse, which are defined as

$$S(\omega) = \int_{-\infty}^{\infty} s(t) e^{-j2\pi f t} dt \qquad (1a)$$

$$s(t) = \int_{-\infty}^{\infty} S(f) e^{j2\pi f t} df \qquad (1b)$$

The FT is valid for both periodic and non-periodic time signals that satisfy certain minimum conditions. Almost all real world signals easily satisfy these requirements (It should be noted that the Fourier series is a special case of the FT). Mathematically,

FT is defined for continuous time signals.In order to do frequency analysis, the time signal must be observed infinitely.

To compute the Fourier transform numerically on a computer, discrimination plus numerical integration are required. This is an approximation of the true (i.e., mathematical), analytically-defined FT in a synthetic (digital) environment, and is called discrete Fourier transformation (DFT). There are three difficulties with the numerical computation of the FT:

• Discrimination (introduces periodicity in both the time and the frequency domains)

Numerical integration (introduces numerical error, approximation)

- Finite time duration (introduces maximum frequency and resolution limitations

- The DFT of a continuous time signal sampled over the period of T, with a sampling rate of Δt can be given as $S(m\Delta f) = \frac{T}{N} \sum_{n=0}^{N-1} s(n\Delta t) e^{-j2\pi m\Delta f n\Delta t}$ (4)

Where Δf=1/T, and, is valid at frequencies up to $f_{max} = 1/(2\Delta t)$. Table 2 lists a simple Mat lab m-file that computes (5) for a time record s(t) of two sinusoids whose frequencies and amplitudes are user-specified. The record length and sampling time interval are also supplied by the user and DFT of this record is calculated inside a simple integration loop

Basic Discrimination DFT and FFT Requirements:

Mathematically defined Fourier transformation can be used to calculate the FT of a function at any frequency. There is no maximum frequency, or frequency resolution limit, since these are numerical FT restrictions. The maximum frequency in DFT or FFT depends on the sampling interval, and the frequency resolution is determined by the signal record length. That is N samples of a time signal recorded during a finite duration of T with a sampling period of Δt (N=T/Δt) can be transformed into N samples in the frequency domain between $-f_{max}$ and $+f_{max}$ according to $f_{max} = \frac{1}{2\Delta t}$, $\Delta f = \frac{1}{T}$.

Since sampling interval and signal record lengths are finite in numerical computations in the computers maximum frequency and the resolution are also finite. This means

- any frequency component f_c beyond $+f_{max}$ can not be observed in its actual frequency; instead it enters from left because of rotational symmetry and periodicity and appears at $-f_{max} + f_D$ where $f_D = f_c - f_{max}$.
- Similarly, any frequency component $-f_c$ beyond $-f_{max}$ can not be observed in its actual frequency; instead it enters from left because of rotational symmetry and periodicity and appears at $f_{max} - f_D$ where $f_D = |f_c - f_{max}|$.

It should be noted that, the Mat lab code in Table 2 directly integrates (4) numerically, so any number of frequency samples ($n \times \Delta f$) can be used to plot a frequency spectrum. Unfortunately, (7) still holds for the DFT. In other words, for example if one wants to discriminate two sinusoids with 50Hz and 55Hz in the frequency domain the frequency resolution Δf must be much less than their difference (5Hz).

III . MULTIPIER

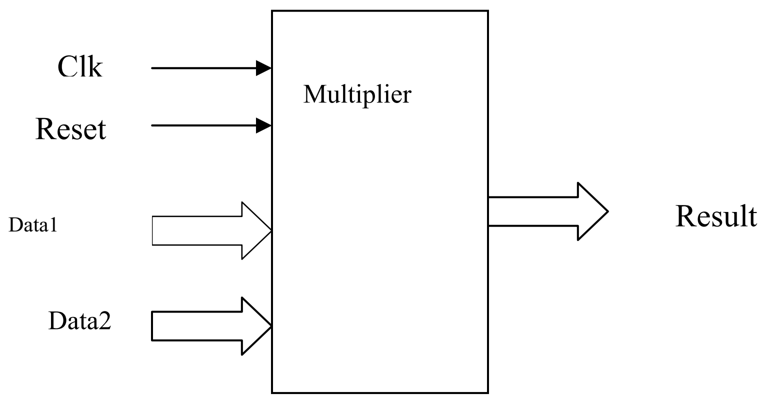

Fig 1. Block diagram of the multiplier

As shown in fig 1 the multiplier module, consists of Clk, Reset, Data1, Data2 are the inputs and Result is the output. The Data1 and Data2 input contain the 8-bits of information .The Result contains the 16-bits of information. This module is synchronous module with Positive Clk edge. If the reset is high the Result output should be zero else the multiplication operation is performed each and every clock edge of the Clk.

COMPLEX MULTIPLER:

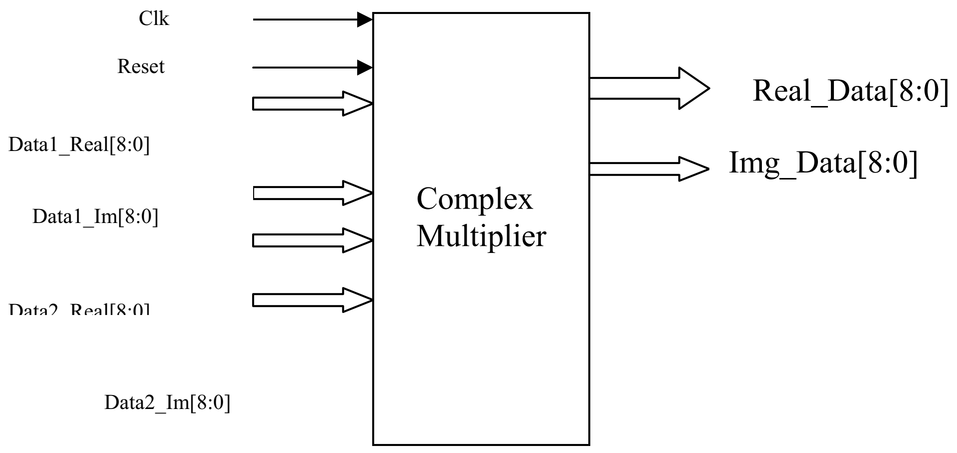

Fig 2. Block diagram of the Complex multiplier

as shown in fig.2 the Clk,Reset, Data1_real[8:0], Data1_img[8:0], Data2_real[8:0], Data2_Img[8:0] are the inputs and Real_Data[8:0] and Img_Data[8:0] are the outputs. The operation is perform on every positive Clk edge. The Reset is active low Reset. The Real Data [8:0] and Img_Data [8:0] are calculated by using the this following equations

Real_Data= Data1_real[8:0]X Data2_real[8:0]- Data1_Img[8:0] X Data2_Img[8:0];--[1]

Img_Data= Data1_real[8:0]X Data2_Img[8:0]- Data2_Real[8:0] X Data1_Img[8:0];--[2]

SIGNED ADDER:

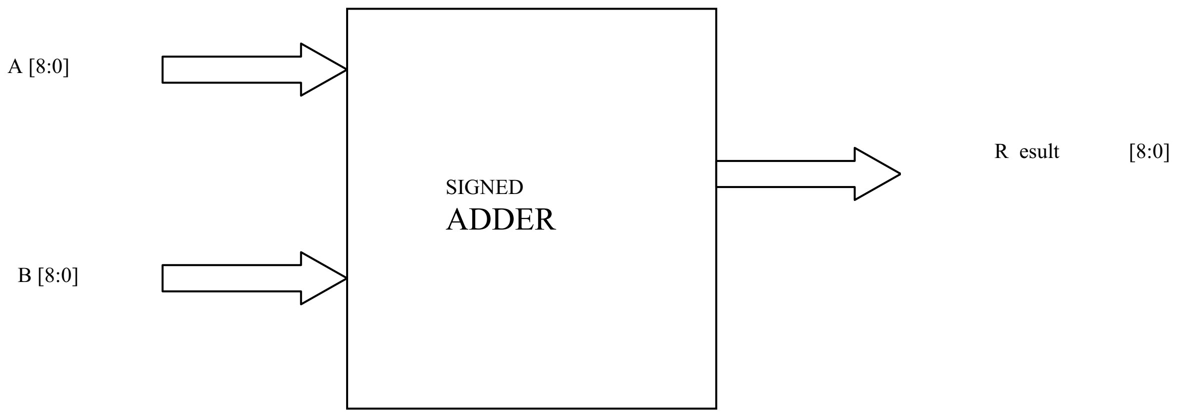

Fig 3. Block diagram of the Signed Adder

As shown in fig 3, the sign adder contains the A, B are the inputs with the 9-bits of information. The Result is the output of the signed adder with 9-bits of information. The operation is depend on the MSB bits of the A and B . If A and B MSB bits are same the operation is a simple adder operation. If MSB bit of the A is not equal to B, then compare the magnitude of the A and B, the Result magnitude is equal to the small number is subtracted from the big number and sign bit of the Result is equal to the big number sign bit.

COMPLEX BUTTERFLY MODULE **:**As shown in the complex butter module consists AR[8:0], AI[8:0], BR[8:0], BI[8:0], WR[8:0], WI[8:0] are the inputs and OUTPUTR[8:0] and OUTPUTI[8:0] are the two outputs. R stands for the real part of the sample and I stand for the imaginary part of the samples.

First compute the complex multiplication between the BR [8:0], BI [8:0], WR [8:0], WI [8:0] then the result is real and imaginary parts. These outputs are add/subtract from the AR [8:0], AI [8:0] respectively and getting the OUTPUTR [8:0] and OUTPUT I[8:0].

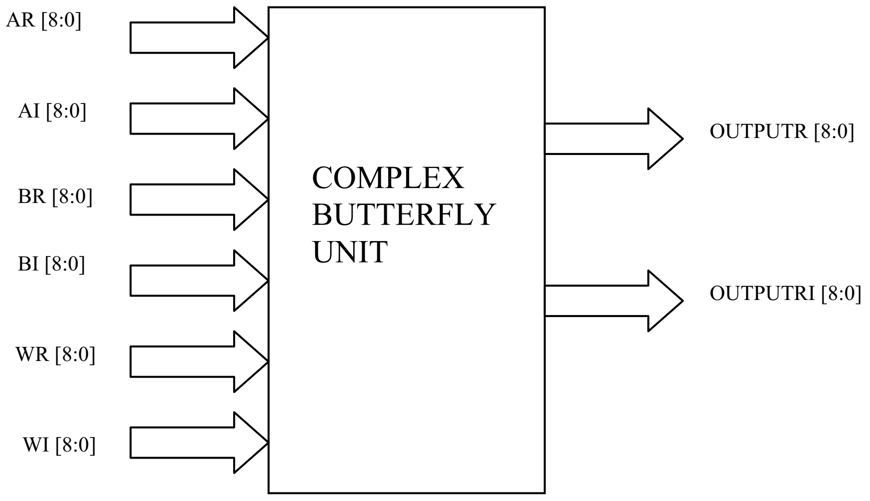

Fig 4. Block diagram of the Complex Butterfly Unit

LUT:

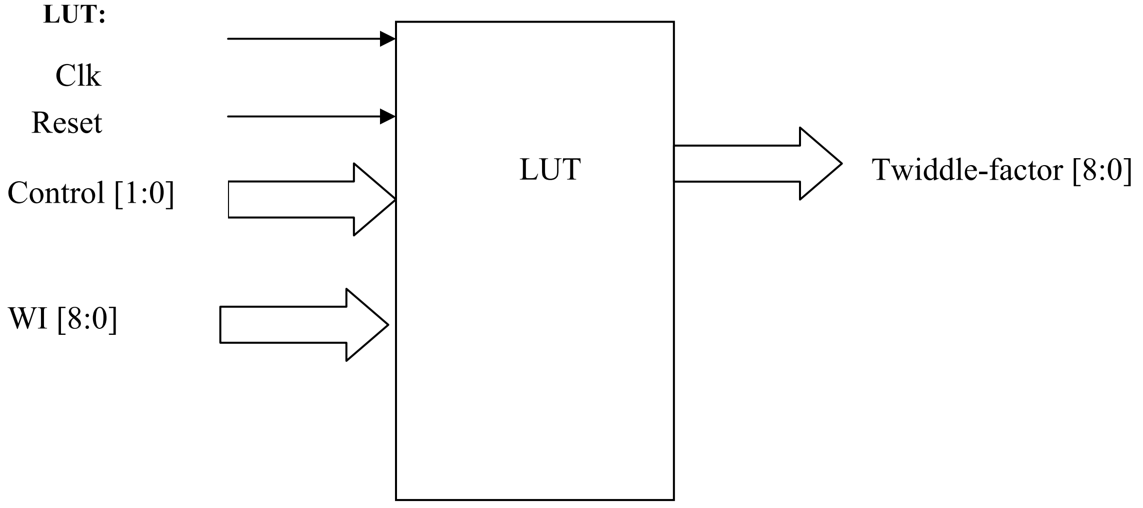

Fig 5. Block diagram of the LUT

As shown in figure5 the Clk,Reset and Control [1:0] are the inputs and twiddle-factor[8:0] is the output. This is a synchronous LUT. It performs its operation on every positive edge of the Clk. The Reset is the active low value.The control [1:0] come from the main controller .It produce all necessary control signals for different operations. The control [1:0] has the value of "00" the LUT provides the twiddle factors necessary for the first stage of the Radix_2 FFT calculation. In the same manner the LUT produce the different twiddle factors for the different stages depending the value of the controller like "01","10" and "11".

IV . RADIX_2 FFT CALCULATOR MODULE

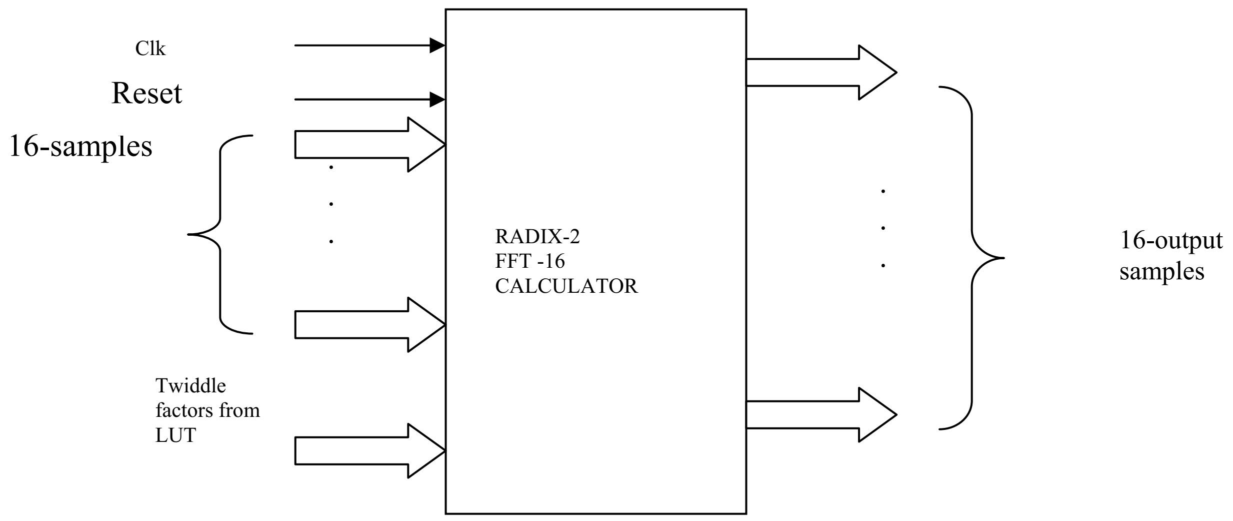

Fig 6. Block diagram of the Radix-2FET-16 CalculatorAs shown in figure the Clk, Reset, 16-samples and the twiddle factors are the inputs and 16-outputs samples are the outputs . Each and every sample is represented by using the 9-bits of information. The 8-bits are for the information and 1-bit is used for the sign bit representation.The number of samples is 16, since the four stages of complex butterfly multiplications are required. For the each and every stage requires the different twiddle factors .The required twiddle factors are delivered by using LUT table. After the fourth stage calculation we can get the 16-samples outputs with real and imaginary parts.

V. RADIX-4 COMPLEX BUTTERFLY MODULE :

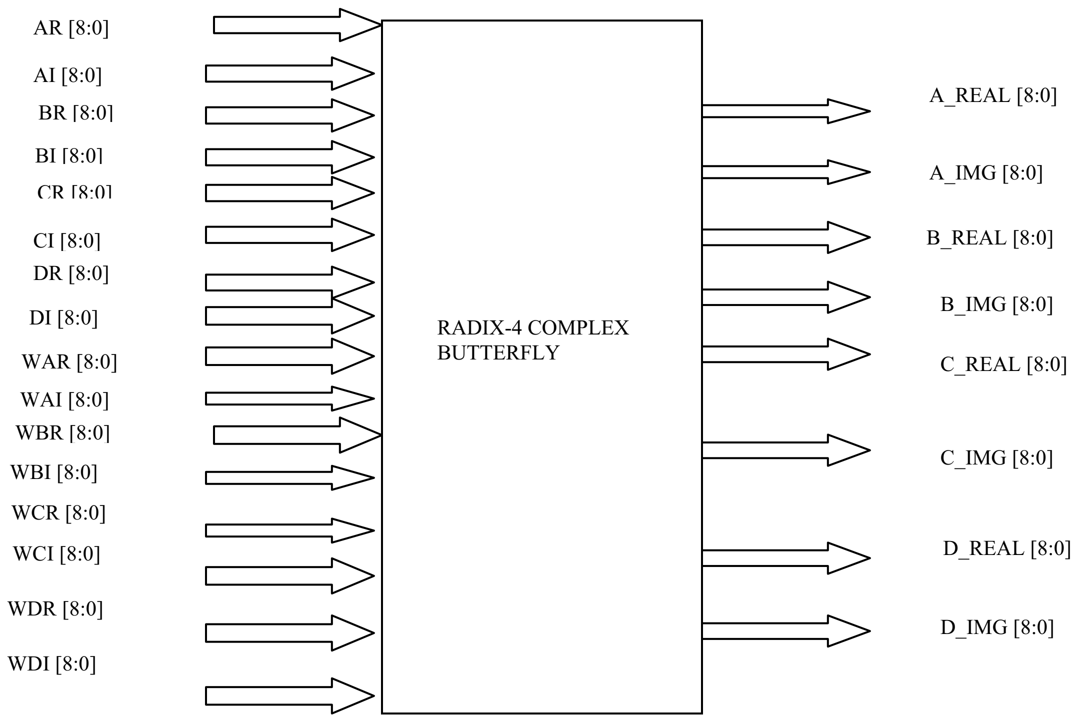

Fig 7. Block diagram of the Radix-4 Complex Butterfly\

As shown in fig .7. AR [8:0], AI [8:0], BR [8:0], BI [8:0], CR [8:0], CI [8:0], DR [8:0], DI [8:0], WAR [8:0], WAI [8:0], WBR [8:0], WBI [8:0], WCR [8:0], WCI [8:0], WDR [8:0], WDI [8:0] are the inputs of 9-bits and A_REAL [8:0], A_IMG [8:0], B_REAL [8:0], B_IMG [8:0], C_REAL [8:0], C_IMG [8:0], D_REAL [8:0], D_IMG [8:0] are the outputs of 9-bits.

The first operation in this module is calculate the complex multiplications of respective twiddle factors like AR [8:0], AI [8:0] with WAR [8:0], WAI [8:0]. In the second operation we can calculate the A_REAL [8:0], A_IMG [8:0], B_REAL [8:0], B_IMG [8:0],
C_REAL [8:0], C_IMG [8:0], D_REAL [8:0], D_IMG [8:0] by using the following equations

A_REAL =ar+(crxwr-cixwi)+(brwr-biwi)+drwr-diwi) ------------------------ 1.
A_IMG=ai+(crwi+ciwr)+(brwi+biwr)+(crwi+ciwr)+(drwi+diwr) ------------------------- 2.
B_REAL =ar-(crwr-ciwi)+(brwi+biwr)-(drwi+diwr) ----------------------------------- 3.
B_REAL =ai-(crwi+ciwr)-(brwr-biwi)+(drwr-diwi) ----------------------------------- 4.
C_REAL =ar+(crwr-ciwi)-(brwr-biwi)-(drwr-diwi) ----------------------------------- 5.
C_IMG=ai+(crwr+ciwi)-(brwi+biwr)-(drwi+diwr) ----------------------------------- 6.
D_REAL =ar-(crwr-ciwi)-(brwi+biwr)+(drwi+diwr) ----------------------------------- 7.
D_IMG=ai-(crwi+ciwr)+(brwr-biwi)-(drwr-diwi) ----------------------------------- 8.

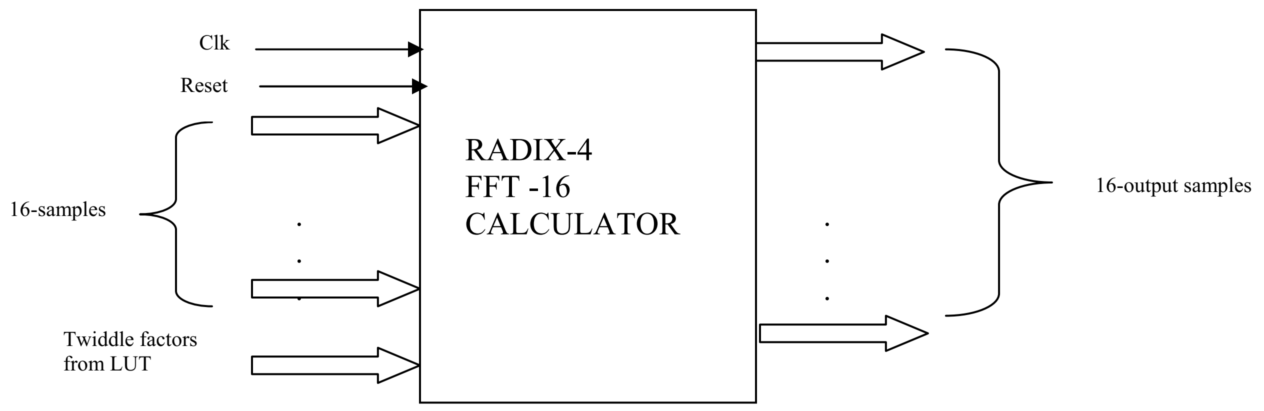

Fig 8. Block diagram of the Radix-4 FET-16 Calculator

As shown in figure 8.the Clk, Reset, 16-samples and the twiddle factors are the inputs and 16-output samples are outputs. Each and every sample is represented by using the 9-bits of information. The 8-bits are for the information and 1-bit is used for

the sign bit representation.The number of samples is 16, since the two stages of complex radix-4 butterfly multiplications are required. For the each and every stage requires the different twiddle factors .The required twiddle factors are delivered by using LUT table. After the fourth stage calculation we can get the 16-samples outputs with real and imaginary parts.

FFTController:

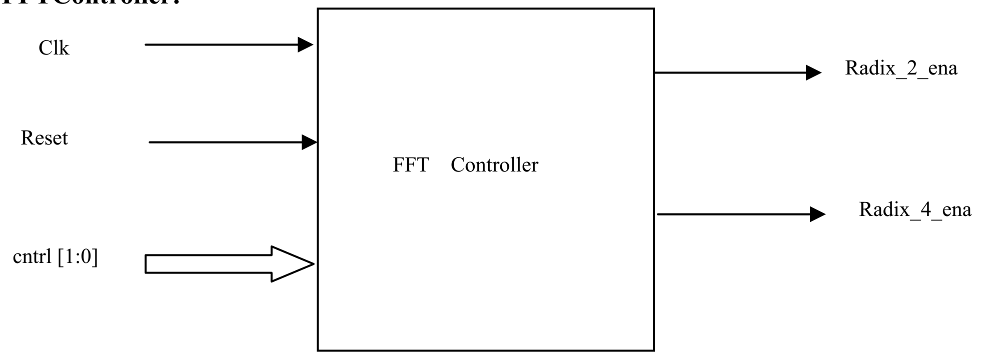

Fig 9. Block diagram of the Controller

As shown in fig9. Clk, Reset and cntrl [1:0] are the inputs and Radix_2_ena , Radix_4_ena are the outputs. Every positive Clk edge the Radix_2_ena, Radix_4_ena are enabled or disabled depending the cntrl [1:0] bits. If cntrl value is “00” Radix_2_ena, Radix_4_ena both are disabled, if “01” Radix_2_ena is enable and Radix_4_ena is disable, if “10” Radix_2_ena is disable and Radix_4_ena is enable, if “11” both are activated.

VI. TOP MODULE:

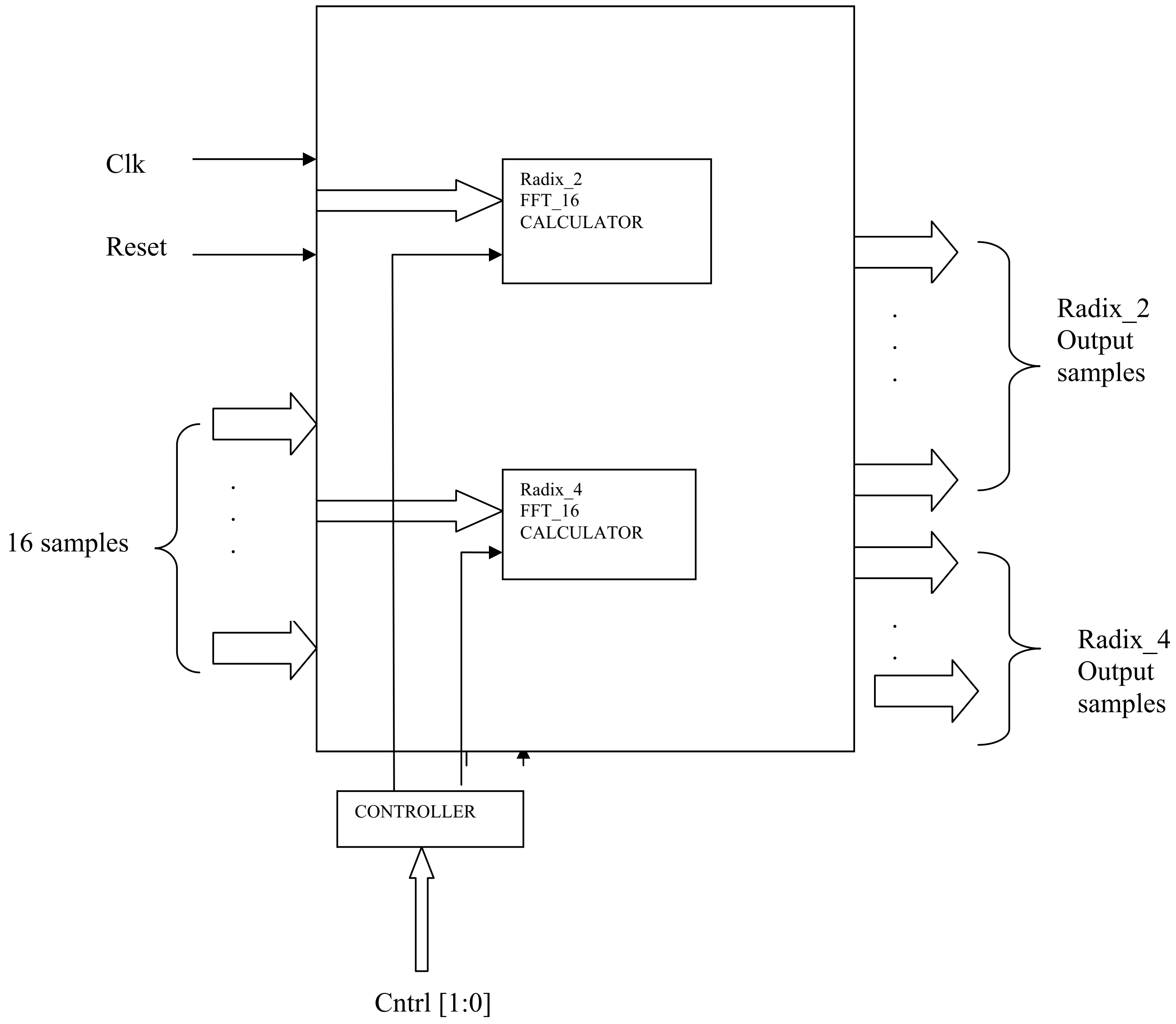

Fig 10. Block diagram of the Proposed Architecture [Top Module]

As shown in fig, Clk, Reset, cntrl [1:0] and 16_input samples are inputs. Each and every sample is represented as a real part of 9-bits and 9-bits of imaginary bits. This is synchronized design with active low Reset value. Radix-2 of 16-output samples and Radix_4 of 16-outputs samples are the outputs. Whenever the Reset is high the both outputs should be in zero value. Depending on the cntrl value we can get either Radix_2 output samples or Radix_4 output samples and both samples are active state when cntrl input value is "11".

VII. CONCULSION

This architecture is designed by using verilog language .We have used here model sim (Questa Sim 6.2b) tool for the compilation purpose. By using this project we can able to implement the VLSI architecture for the mixed radix.Mostly we are using FFT in OFDMA, DSP, Wireless communication .It has wide spread of applications in mobile communication. It increases the speed of operation.So by using this Mixed Radix FFT we can able to achieve lot of applications. We proposed VLSI architecture for a generalized mixed-radix (GMR) algorithm of FFT.

VIII. FUTURE SCOPE

This is a novel architecture for any Radix-N of FFT calculation method. It is scalable architecture for any number samples like 32,64,128,256,1024also.This is suitable for different OFDM application with little bit of hardware modification. By using this project we can drastically reduce the program execution time. We can use it in power spectrum density, speed spectrum analysis.In the future by increasing the size of the inputs of element we can able to increase size of the algorithms .It leads to achieve number of applications in the future. It will become one of the best solutions for mixed radix algorithms.

REFERENCES

[1] L. G. Johnson, "Conflict free memory addressing for dedicated FFT hardware," IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process. vol. 39, no. 5, pp. 312–316, May 1992.

[2] J. A. Hidalgo, J. Lopez, F. Arguello, and E. L. Zapata, "Area-efficient architecture for fast Fourier transform," IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process., vol. 46, no. 2, pp. 187–193, Feb. 1999.

[3] B. G. Jo and M. H. Sunwoo, "New continuous-flow mixed-radix (CFMR) FFT processor using novel in-place strategy," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 52, no. 5, pp. 911–919, May 2005.

[4] Z.-X. Yang, Y.-P. Hu, C.-Y. Pan, and L. Yang, "Design of a 3780-point IFFT processor for TDS-OFDM," IEEE Trans. Broadcast., vol. 48, no. 1, pp. 57–61, Mar. 2002.

[5] 3GPP TS 36.201 V8.3.0 LTE Physical Layer—General Description,E-UTRA, Mar. 2009.

[6] C. Burrus, "Index mappings for multidimensional formulation of the DFT and convolution," IEEE Trans. Acoust., Speech, Signal Process., vol. ASSP-25, no. 3, pp. 239–242, Jun. 1977.

[7] D. P. Kolba and T. W. Parks, "A prime factor FFT algorithm using high-speed convolution," IEEE Trans. Acoust., Speech, Signal Process., vol. ASSP-25, no. 4, pp. 281–294, Aug. 1977.

[8] A. M. Despain, "Very fast Fourier transform algorithms hardware for implementation," IEEE Trans. Comput., vol. C-28, no. 5, pp. 333–341, May 1979.

[9] C. L. Wey, S.-Y. Lin, and W. C. Tang, "Efficient memory-based FFT processors for OFDM applications," in Proc. IEEE Electro/Inf. Technol., May 17–20, 2007, pp. 345–350.

[10] Jaguar II Variable-Point (8-1024) FFT/IFFT, Drey Enterprise Inc., Crosslake, MN, 1998.

[11] R. Radhouane, P. Liu, and C. Modlin, "Minimizing the memory requirement for continuous flow FFT implementation: Continuous flow mixed mode FFT (CFMM-FFT)," in Proc. IEEE Int. Symp. Circuits Syst., May 2000, vol. 1, pp. 116–119.

[12] Y. Ma, "An effective memory addressing scheme for FFT processors," IEEE Trans. Signal Process., vol. 47, no. 3, pp. 907–911, May 1999.

[13] D. Reisis and N. Vlassopoulos, "Address generation techniques for conflict free parallel memory addressing in FFT architectures," IEEE Trans.Circuits Syst. I, Reg. Papers, vol. 55, no. 11, pp. 3438–3447, Dec. 2008.