ISSN: 2582 - 6379
Orange Publications
International Journal for Interdisciplinary Sciences and Engineering Applications
IJISEA - An International Peer- Reviewed Journal
2021, Oct, Volume 3 Issue 1
www.ijisea.org

Development of Audio driver through ASIC implementation of FFT Engine

P Gireesha[1],G Latha[2]
PG Scholar[1] , Assistant Professor[2]
Department of Electronics and Communication Engineering
Amrita Sai Institute of Science and Technology
Paritala,Kanchikacherla,Krishna District , Andhra Pradesh , India

ABSTRACT

FFT Engine has been designed at three different level of abstraction, Algorithmic level, Microdesign or low level design and RTL code generation. FFT engine is divided into three blocks, Memory block, FFT engine and Analysis block. FFT block is commuting FFT using radix-4 algorithm and it has been implemented using Memory based architecture. FFT block will compute 216 (65536) point FFT. CORDIC multipliers are used to save the memory. Analysis block has been designed for audio application that uses audio range (20 Hz to 20 kHz) for analysis as well as for ADC application that uses entire first Nyquist zone (zero to half of the sampling frequency). FFT engine can compute the parameters SNR, SNDR ,SFDR, THD and DC component for the noise floor up to -120 dB with an error less than 0.5 dB. There is huge saving of resources utilized by FFT Engine in comparison to HDL optimized FFT block given in Simulink HDL Coder library . HDL optimized FFT block itself uses 28 multipliers, 366 adders/subtractors, 1439 registers, 34 RAMs and 924 multiplexers. Our FFT block along with Memory block and analysis block needs only 16 multipliers, 689 adders/subtractors, 87 registers, 2 RAMs and 525 multiplexers. Number of adders are more because of logarithmic block and CORDIC block that have been designed for 21 stages and 19 stages respectively and each stage needs 6 adders/subtractors. Generated RTL code (Verilog) has is simulated using ModelSim and results have been verified with the results computed by Cadence tool as well as the result computed by Fixed point FFT Engine. In CORDIC block, Adder/Subtractor block has been used where a control bit decides whether this block will act as Adder or Subtracter. HDL Coder is generating one Adder, one Subtractor and one multiplexer to select the appropriate hardware according to control signal. In digital design, one can use xor gates and adder to perform the operation. We can save huge number of adders using this technique.

Keywords: FFT Engine, CORDIC Multipliers, Microdesign, Nyquist Zone, Audio Driver

I.INTRODUCTION

Dynamic performance of an audio driver is measured in using using the parameters SNDR (Signal to Noise plus Distortion Ratio), SFDR (Spurious Free Dynamic Range), THD (Total Harmonic Distortion), SNR (Signal to Noise Ratio) and DC component in the signal. To improve the dynamic performance of audio driver system, we need to monitor the these performance parameters of the signal. We can design a system that accepts analog audio signal as input, converts it into a digital signal using an ADC, calculates its various performance parameters and feed back to a control block. Control block takes these

ISSN: 2582 - 6379
Orange Publications
International Journal for Interdisciplinary Sciences and Engineering Applications
IJISEA - An International Peer- Reviewed Journal
2021, Oct, Volume 3 Issue 1
www.ijisea.org

performance parameters and compares it with the stored reference values. If control bloc finds that any parameter is going away from given limited value then it tries to offset that parameter by sending appropriate signal to corresponding block. We can take an example of DC offset in signal. Suppose desired DC component in the audio signal is zero that is reference value for DC component. Now if measured DC component is 0.5 mV then control can send feedback signal to offset control block to subtract 0.5 mV from each sample and after subtracting this offset from each sample, if we will take the FFT of input samples then measured DC in the analog signal will be zero. Since we can not subtract each measured value from the samples so we have to decide the fixed step of increment from the least possible value to max possible value. In the same way, we can develop the technique to offset other parameters also. Block diagram shown in Figure 1 clearly represents the motivation behind the thesis work. ADC is sampling the analog signal from mixed signal system and converting it into digital format. N such samples are stored in memory to compute its FFT and half of the FFT bins are stored in memory to compute the various parameters. Control block is performs the comparison and sends the 2 appropriate feed back signal to mixed signal system.

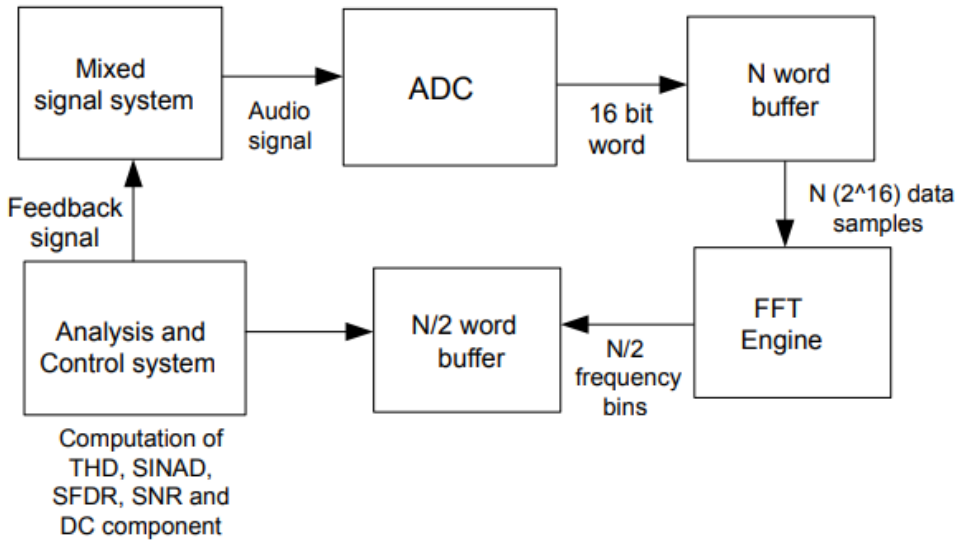

Figure 1: Shows Block diagram of FFT engine

II. BEHAVIORAL LEVEL MODELING OF FFT ENGINE

If any signal x(t) is aperiodic and continuous in nature then Fourier Transform of the signal will also be aperiodic and continuous. If it is discrete and aperiodic then also corresponding frequency domain signal will be continuous in nature. Only in case of Discrete Fourier Transform (DFT) signal is discrete in nature in both time domain as well as frequency domain. There is an inherent advantage of discrete time domain and frequency domain signal that it can be stored and processed using a digital computer and we can characterized and analyzed the signal in frequency domain. DFT of N samples of a discrete signal x(n) is given by Equation (1.1) where X(k) is DFT of x(n) and k is frequency domain index.

ISSN: 2582 - 6379
Orange Publications
International Journal for Interdisciplinary Sciences and Engineering Applications
IJISEA - An International Peer- Reviewed Journal
2021, Oct, Volume 3 Issue 1
www.ijisea.org

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

III. MODELING OF FFT ENGINE IN SIMULINK

Next step of the project is modeling of FFT engine in Simulink using available high-level resources. All the functions used in algorithm must be implemented using dedicated hardware. At this level one can choose a suitable architecture that meets our requirement of hardware. Our aim is to minimize the requirement of hardware with optimum speed of operation. Top level model integrates all the required blocks in a single test-bench. Behavioral level MATLAB algorithms are attached to corresponding Simulink blocks. Figure 3.1 shows the top level model of FFT engine.

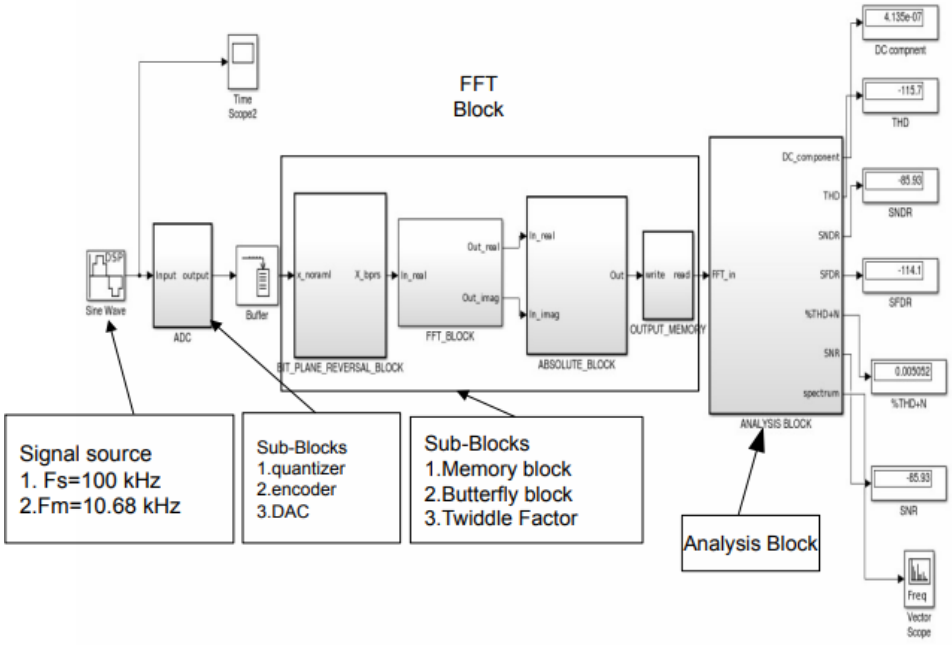

Figure 2: Shows Top level model of FFT Engine

ARCHITECTURE OF FFT BLOCK

There are three well known architecture are available for the implementation of FFT Block
1. Parallel Architecture
2. Pipelined Architecture
3. Memory based Architecture

Parallel architecture needs (N/4)*log4N radix-4 butterflies to compute 'N' point FFT using radix-4 FFT algorithm. This architecture is suitable for very high speed processing but hardware requirement is very high. Pipelined architecture is generally used for real time signal processing which computes FFT in a sequential manner. This architecture needs log4N radix-4 butterflies to compute 'N' point FFT using radix-4

ISSN: 2582 - 6379
Orange Publications
International Journal for Interdisciplinary Sciences and Engineering Applications
IJISEA - An International Peer- Reviewed Journal
2021, Oct, Volume 3 Issue 1
www.ijisea.org

FFT algorithm. Hardware requirement in this architecture is less than Parallel architecture but this architecture is slower in comparison to Parallel architecture. Memory based architecture is the slowest architecture among all the three architectures but it needs least hardware. This architecture computes the FFT using a single butterfly. A control logic block (a finite state machine) is used to manage the data to perform all the operations. So finally the Memory based architecture is used for the implementation of FFT Block because it is the most hardware efficient algorithm for the implementation of FFT Block. Figure 3.2 is showing the block diagram of the architecture of FFT block. It is a memory based architecture. There are two RAMs and nine registers in this architecture. Registers holds the data before and after processing until it gets stored back to RAM. Control logic block is responsible for the rotation of data. Butterfly block is a radix-4 butterfly without CORDIC block. In this architecture it is assumed that twiddle factors are stored in ROM in Twiddle factor block and control logic logic is providing appropriate signal to fetch the required sine and cosine value of the twiddle factor. In next step of implementation, twiddle factor block will be removed and CORDIC multipliers will be inserted in the in the Butterfly block to rotate the complex data with given angle. It will save the N word ROM used in the Twiddle factor block and real multipliers used in CORDIC block.

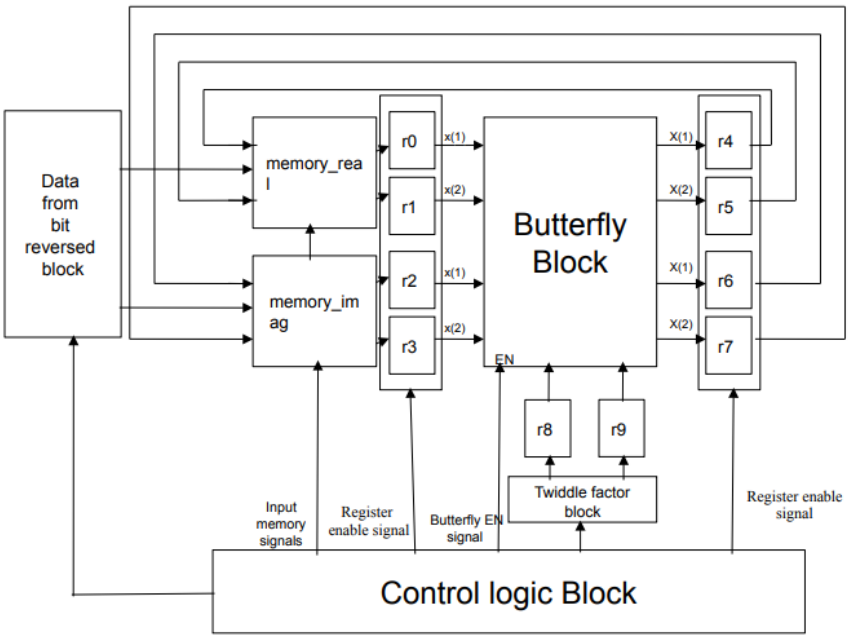

Figure 3: Shows Block diagram of the architecture of FFT block

IV. CORDIC ALGORITHM

There are three main objectives of CORDIC Algorithm in the project work

1. Our first objective is to make a complex multiplier (or to rotate a given vector having coordinate (x, y) with any arbitrary angle θ) using CORDIC algorithm.

2. Our second objective is to convert a given vector having coordinate (x, y) from Cartesian form to polar form (or to get the absolute value of a given vector and angle made with positive x axis).

ISSN: 2582 - 6379
Orange Publications
International Journal for Interdisciplinary Sciences and Engineering Applications
IJISEA - An International Peer- Reviewed Journal
2021, Oct, Volume 3 Issue 1
www.ijisea.org

3. Our third objective is to use the CORDIC algorithm for the implementation of logarithmic and square-root function.

We can make a complex multiplier with real adders and subtractors but in FFT implementation we just need to rotate a complex number with a given angle so we are just changing the phase of that complex number and magnitude of that number is always constant. If x(n) is some input sequence which length is N and X(k) is its DFT then we can use Equation 4.1 to find out X(k) as

$$X(k)=\sum_{n=0}^{n=N-1} x(n)e^{(-j\frac{2\pi}{N}(n.k))} \quad (2)$$

Here x(n) is a complex variable and we are changing its phase for every variation of n and k. Since magnitude of the product is same but the magnitude of its real and imaginary component is changing because of rotation.

EXAMPLE OF VERILOG CODE GENERATION

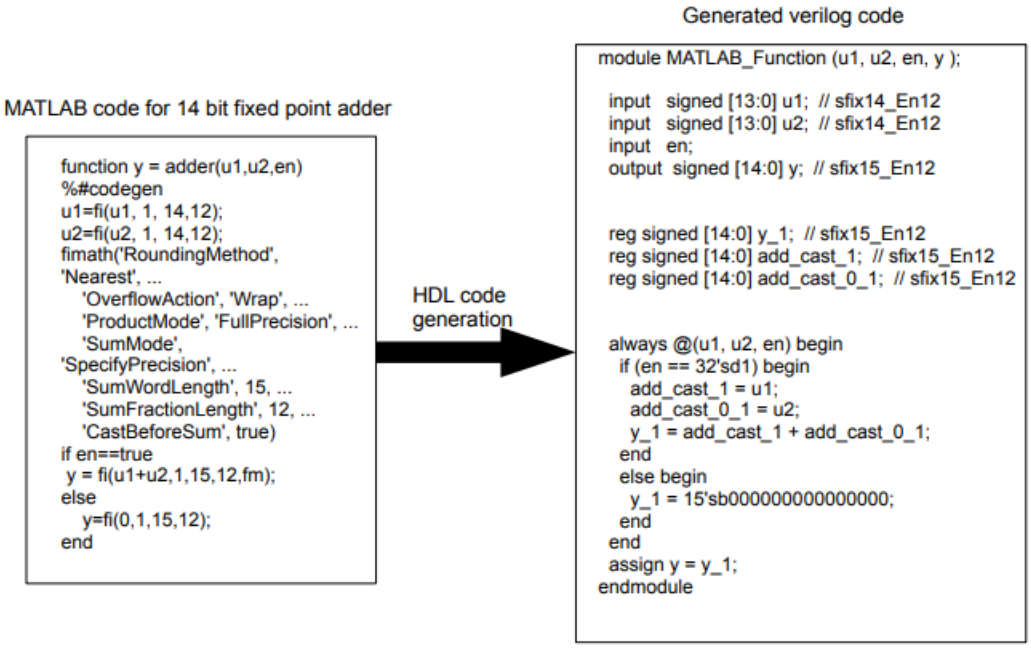

Figure 4: Shows Verilog code generation example

V. RESULTS AND DISCUSSION

FFT Engine has been designed at three different level of abstraction, Algorithmic level, Micro design or low level design and RTL code generation. FFT engine is divided into three blocks, Memory block, FFT engine and Analysis block. FFT block is commuting FFT using radix-4 algorithm and it has been

implemented using Memory based architecture. FFT block will compute 216 (65536) point FFT. CORDIC multipliers are used to save the memory. Analysis block has been designed for audio application that uses audio range (20 Hz to 20 kHz) for analysis as well as for ADC application that uses entire first Nyquist zone (zero to half of the sampling frequency). FFT engine can compute the parameters SNR, SNDR ,SFDR, THD and DC component for the noise floor up to -120 dB with an error less than 0.5 dB. There is huge saving of resources utilized by FFT Engine in comparison to HDL optimized FFT block given in Simulink HDL Coder library . HDL optimized FFT block itself uses 28 multipliers, 366 adders/subtractors, 1439 registers, 34 RAMs and 924 multiplexers. Our FFT block along with Memory block and analysis block needs only 16 multipliers, 689 adders/subtractors, 87 registers, 2 RAMs and 525 multiplexers. Number of adders are more because of logarithmic block and CORDIC block that have been designed for 21 stages and 19 stages respectively and each stage needs 6 adders/subtractors. Generated RTL code (Verilog) has is simulated using ModelSim and results have been verified with the results computed by Cadence tool as well as the result computed by Fixed point FFT Engine.

FUTURE WORK:

1. Synthesis of RTL code: Next step of design flow is the synthesis of RTL code with the tool like Design Compiler to get the gate-level netlist for the targeted technology and constraints. The designed FFT engine is an integral part of Audio Driver IC, and it will be integrated for the development of Audio Driver IC.

2. Place and Route: It is the next level of design flow. Place and Route tool takes the gate-level netlist as input and output is a GDS file, used by foundry for fabricating the ASIC.

3. Design of Control block: Computed results can be analyses through the a Control block and can be offset by sending proper feed back signal. As one can take the example of computed DC parameter, if it going beyond a specific quantized level then control block may subtract the that quantized level from each sample that will reduce the net offset in the signal by that quantized level.

REFERENCES:

[1] Xin Xiao, Erdal Oruklu, and Jafar Saniie, IEEE Transactions on circuits and systems-ii: Express Briefs, VOL. 55, NO. 11, November 2008.

[2] Prof. J.M.Rudagi , Richard Lobo, Pradeep Patil, Nikit Biraj, Naimahmed Nesaragi , International Conference on Advances in Recent Technologies in Communication and Computing , 2010.

[3] M. Hasan, T. Arslan and J.S. Thompson , A Novel Coefficient Ordering based Low Power Pipelined Radix-4 FFT Processor for Wireless LAN Applications , IEEE Transactions on Consumer Electronics, Vol. 49, No. 1, February 2003.

[4] Xiaobo Hu, Ronald G. Harber and Steven C. Bass, Expanding the Range of Convergence of the CORDIC Algorithm, IEEE Transactions on Computers, VOL. 40, NO. 1, January 1991.

[5] Texas Instruments, FFT Implementation on the TMS320VC5505, TMS320C5505, and TMS320C5515 DSPs. Available: www.ti.com/lit/an/sprabb6b/sprabb6b.pdf

[6] John G. Proakis , Dimitris K Manolakis, Digital Signal Processing: Principles, Algorithms, And Applications, Third Edition,Efficient computation of the DFT: Fast Fourier Transform algorithms, Prentice Hall International INC.

ISSN: 2582 - 6379
Orange Publications
International Journal for Interdisciplinary Sciences and Engineering Applications
IJISEA - An International Peer- Reviewed Journal
2021, Oct, Volume 3 Issue 1
www.ijisea.org

[7] Texas Instruments, Implementing the Radix-4 Decimation in Frequency (DIF) Fast Fourier Transform (FFT) Algorithm Using a TMS320C80 DSP.

www.ijisea.org