VLSI Implementation of 1024-Point Pipelined FFT

1Chaithra. N, 2Rashmi. V
1,2Telecommunication Engineering, Dr. Ambedkar Institute of Technology, Bangalore, India

Abstract: The paper is the implementation of a RS2DF pipelined FFT Architecture using hardware description language (Verilog) simulated up to 50 MHz for transformation length 1024-point. A hardware oriented radix – 2^2 algorithms is derived by integrating twiddle factor decomposition technique in divide and conquer approach. Radix- 2^2 algorithm has the same multiplicative complexity, as that of radix-4 algorithm, but it retains the butterfly structure of radix-2 algorithm. The single-path delay feedback architecture is used to exploit the spatial regularity in signal flow graph of an algorithm. The Fast Fourier Transform (FFT) and its inverse (IFFT) are two important algorithms in signal processing, software defined radio, and one of the most promising modulation algorithms - Orthogonal Frequency Division Multiplexing (OFDM).

Keywords: FFT- Fast Fourier Transform, VHDL- Very High Description Language.

I. INTRODUCTION

The FFT is the most efficiently used algorithm to calculate the Discrete Fourier Transform (DFT) with respect to its efficiency in reducing computation time. The discrete Fourier transform (DFT) of length N is used to calculate the sampled Fourier transform of a discrete-time sequence where N has evenly distributed points k=2k/N on the unit circle. The following equation shows the length-N forward DFT of a sequence

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

Where k=0,1,...,N-1,

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

Where n=0,1,...,N-1.

The complexity of the DFT direct computation can be significantly reduced by using fast algorithms that uses a nested decomposition of the summation in equations one and two in addition to exploiting various symmetries inherent in the complex multiplications. Cooley- Tukey radix-r decimation in frequency (DIF)FFT is one such algorithm, which recursively divides the input sequence into N/r sequences of length r and requires log2(N) stages of computation. Each stage of the decomposition typically shares the same hardware, with the data being read from memory, passed through the FFT processor and written back into the memory. Each pass through the FFT processor is required to be performed logrN times. Popular choices of the radix are r=2, 4, and 16. Increasing the radix of the decomposition leads to a reduction in the number of passes required through the FFT processor.

II. MOTIVATION

On the perspective of the hardware specifications, Field Programmable Gate Array (FPGA) devices are increasingly being used for hardware implementations in communication applications. FPGAs at advanced technology can achieve high performance, though it can have more flexibility, faster design time, and lower cost. As such, FPGAs are becoming more attractive for FFT processing applications.

An integrated circuit, field-programmable gate array (FPGA) is designed to be configured by a customer or a designer after manufacturing hence the name “field-programmable”. The FPGA configuration is generally done using a hardware description language (HDL) which is similar to that used for an application-specific integrated circuit (ASIC).

Contemporary FPGAs have large resources of logic gates and RAM blocks to complement complex digital computations. As FPGA designs include very fast I/Os and bidirectional data buses it becomes a challenge in verifying the correct timing of the valid data within the setup and hold time. Floor planning enables allocation of the resources within FPGA to meet its time constraints. FPGAs can be used to implement and logical function that an ASIC could perform. The ability in updating the functionality after shipping, partial re-configuration of a portion of the design and the low non-recurring engineering costs which is relative to an ASIC design (notwithstanding the generally higher unit cost), offer advantages for many applications.
III. OBJECTIVE

The objective of this paper is to implement a Pipelines FFT Architecture using Verilog HDL. The Pipelined Architecture that is going to be implemented is RS2DF (Radix – 2 Single path Delay Feedback). Standard FPGA Flow is adapted to implement in this paper i.e., right from specification to bit file generation, which is going to be programmed on Spartan 3AN FPGA. Simulations and synthesis will be done using Modelsim and Xilinx ISE. The Verilog Simulations will be compared with inbuilt MATLAB FFT Core. MATLAB is used for reference and verification.

Initially 8 point FFT was done and the results were verified, then 1024 point was built and verified for its functionality. The speed achieved for this Core was 47.995MHz and Number of slices used was 2656, no of slice flip flops 1714, number of 4 input LUTs 8770, and number of Block Rams were 9. A bit file for 8 point FFT CORE was generated and burnt on Spartan3AN Evaluation board to verify its working on board and a few test cases were generated and the core was verified.

IV. METHODOLOGY

The proposed paper is designed using Verilog Hardware Description Language(HDL). The main aim of this paper is to reduce the area consumed on the FFT IC CORE. This is accomplished by reducing the number of multipliers used in the implementation of the FFT. This in turn reduces the time consumed for the processing of each stage. Functional verification is carried out using Modelsim PE Student edition. Timing analysis and calculation of the proposed pipelined FFT is completed using Xilinx 14.6 ISE. The FPGA hardware implementation is carried out on the Xilinx SPARTAN 6 FPGA development board. In this we are making a bottom-up approach, the basic building blocks are several 2-point FFT algorithm, which are integrated to obtain the 4-point FFT algorithm and similarly these 4-point are combined to get an 8-point FFT algorithm and this process of integration of small blocks to get the larger blocks continues till we get 1024-point FFT algorithm. 1024-point FFT algorithm is obtained by integrating 512-point FFT blocks.

V. BLOCK DIAGRAM

The architecture implemented is RS2DF Radix-2 Single path delay Feedback. This architecture uses the registers more efficiently by storing the butterfly output in feedback shift registers. A stream of single data goes through the multiplier at every stage. It has the same number of butterfly units and multipliers as in R2MDC approach, but with much reduces memory requirements: N-1 registers. Its memory requirements are minimal. FFT can be decomposed using a first half/second- half approach that divides the output sequence X® into increasingly smaller subsequences; this procedure is called Decimation-In-Frequency (DIF) FFT.
VI. SOFTWARE

6.1 STATE MACHINE FLOW DIAGRAM:

The architecture’s each stage has one butterfly, floating point multiplier and Twiddle ROM. In butterfly the addition and subtraction operations are performed. Two adders and two subtractors are required in the butterfly to perform the operations. The outputs obtained are sent to the next stage of the butterfly. Thus there are 10 stages in total for a 1024-point FFT.

6.2 RADIX-2 FFT ALGORITHM CALCULATIONS:

The Discrete Fourier Transform (DFT) \( X(k) \), for \( k=0,\ldots,N-1 \) of a sequence \( x(n) \), \( n=0,1,\ldots, N-1 \) is defined as:

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

Where \( N \) is the transform size. According to the decomposition method, substituting with,

\[
n = \langle (N/2) n_1 + (N/4) n_2 + n_3 \rangle > N
\]

\[
k = \langle k_1 + 2k_2 + 4k_3 \rangle > N
\]

This yield,

\[
N/4 - 1
\]

\[
X (k_1+2k_2+4k_3) = \sum H(k_1, k_2, n_3) W_N^{-nk_3} n_3=0 +2k_2 \] \( W_N^{-nk_3} \)

Where

\[
H(k_1, k_2, n_3) = [x(n_3) + (-1)^k x(n_3 + N/2)] +(-j)^{k_2}[x(n_3 + N/4)+ (- 1)^{k_1} x (n_3 + (3N/4))]
\]

6.3 SIMULATION OUTCOMES:

After this simplification, we have a set of four DFTs of length \( N/4 \). Fig.4. shows an example of \( N=16 \) points Radix-2 decimation in frequency (DIF) .The method used for the pipelined streaming the I/O architecture of the FFT algorithm. It is to be noted that the inputs are in normal order but the outputs are in permuted or digit-reversed order. The pentagons between BFI and BFII represent the trivial multiplication by \(-j\). After these two butterflies, full twiddle factor multipliers (TFM) are expected to compute the multiplication by the twiddle factor.

Fig.4. shows the block diagram for Radix-2 Npoint FFT processor. The N-point FFT processor has \( \log_4 (N) \) stages with I as the stage number. A typical stage consists of BFI,BFII, delay-feedback, ROM, and TFM. A \( \log_2(N) \) counter is used in controlling the processor. The structure of the last stage is different according from the size of the FFT ; if \( N \) is power of 2, then the last stage is composed of BFI only. But if \( N \) is power of 4, the last stage is composed of BFI and BFII.
VII. RESULTS

7.1 FFT 1024 Simulation Results :

8-point FFT is verified, once this is verified it is sure that it will work for the other points as well i.e., a 16, 32, 64, …..,1024. The reason being is that basic building block of our architecture remains same for all stages. In order to verify for 1024 point, what was done was, 50Hz and 120Hz were generated, added up the two signals. This signal now acts as input to 1024-point FFT. The expected plot output for FFT is that it should be a frequency domain view and the peaks at 50Hz and 120Hz on the graph were obtained. Both Matlab and Verilog codes output plots as shown in the figure below.

![Matlab and Verilog code output for two input signals at frequencies 50Hz and 120Hz.](image1)

![Input to FFT 1024 in analog view.](image2)

![Inputs and outputs of FFT 1024.](image3)

VIII. CONCLUSION

The design and implementation of 1024 Radix 2 single path delay feedback pipelined FFT processor on an FPGA has been presented. The description was made by Verilog RTL in Xilinx ISE on Spartan 3AN, device name xc3s14001-4fg484, Spartan family and the functionality was verified by ModelSim Xilinx Edition. The outputs from the Verilog described architecture are validated against the standard FFT in Matlab. The multipliers, Adders/Subtractor units , control unit and their pipelining were implemented by efficient inferring the MULTI18x18SIOs Blocks in order to obtain a faster and low power design. The word length of the data and the twiddle factor were chosen to achieve an acceptable signal-to–noise ratio and also to match the feature of MULTI18x18SIOs Blocks.

Since scaling and rounding are applied in all pipeline stages, the design can maintain the SNR. The power analysis shows that the processor consumes appropriate power with respect to the transformation length and operating frequency. Number of slices which were used up are 2656 and the available sliced on board were 11264, Number of slice flip flops used were 1714, available were 22528, number of 4 input LUT used were 8770, available on board was 22528, Number of I/O used were 72 output 375. Number of 18x28 multipliers used was 3 outputs 32. The synthesis and simulation of the processor indicates the execution of 1024 data points R2SDF is 20.86ns with maximum operating frequency of 47.995 MHz The comparison with other FFT processors reveals the power of our processor. The implemented design is an easy way to increase the number of points of FFT as well as IFFT by considering simple modification.

IX. FUTURE SCOPE

The future work includes the development of the complete OFDM system and upgrade it to a multiple inputs, multiple outputs (MIMO) system by using high density FPGA device.

It also includes:

- One possible direction is Radix higher than R2SDF can be used. Higher radix design further reduces the number of logical gates and thus has the potential of power saving.
- Future work also includes the development of complete OFDM system and upgrading it to a multiple input outputs (MIMO) system by using high density FPGA device.
- In future, the proposed pipelined FFT architecture may be modified further by the use of other better number formats, such as the Q.15 number format. This results in higher precision, increased accuracy, less computation time and better efficiency.

REFERENCES

[2] Shousheng He and Mats Torkelson, A New Approach to Pipeline FFT processor, Department of Applied Electronics, Lund University S-22100 Lund, SWEDEN. Email: he@tde.1th.se; torkel@tde.1th.se


