Abstract

Objectives: For Orthogonal Frequency Division Multiplexing (OFDM), Fast Fourier Transform (FFT) processors are required. So, in this work, a variable point Pipelined FFT processor is designed by utilizing Verilog language. Further transmitter and receiver blocks are designed to form OFDM system. Methods: Speed enhancement is the key area which is been looked into this project work. Pipelined Architecture that is going to be implemented is RS2DF (Radix-2 Single path Delay Feedback). Standard FPGA Flow is adapted to implement this project i.e., right from specification to bit file generation. Simulations and Synthesis will be done using Questasim and Xilinx ISE. Verilog Simulations will be compared with in-built MATLAB FFT Core. MATLAB is used for reference and verification. Findings: This Project clarified the usage of FFT IP and OFDM in Xilinx ISE focusing on a specific family of FPGA. The execution of the proposed algorithm ought to perform superior to anything the base algorithm with the all-out basic way getting additionally advanced along these lines expanding the speed of activity. maximum operation frequency achieved was 236.189MHz but in base paper maximum operation frequency was 30Mhz. The systems utilized are Internal and external Pipelining of modules and Distributed LUT based idea, the target of the above methods is to lessen the basic way deferral and increment the general speed of activity of structure, the disadvantage is the increments in area and output latency. Area is not of a much issue as present day days FPGAs has immense measure of assets. Output latency should be taken consideration with additional rationale at whatever point FFT is coordinated with other system modules. Application: The project depicts technique for streamlining timing basic ways of FFT to work at higher-speeds, to be utilized as a feature of LTE Protoco1 or any wire1ess protoco1 such as WLAN and LDACS.

Keywords: Fast Fourier Transform (FFT), Hardware Description Language (HDL), Orthogonal Frequency Division Multiplexing (OFDM), Pipeline, Verilog

1. Introduction

The Fast Fourier Transform (FFT), is a standout amongst the most broadly utilized calculations for figuring the Discrete Fourier Transform (DFT) attributable to its proficiency in decreasing calculation time. Depending on Orthogonal Frequency Division Multiplexing (OFDM) rule, FFT is utilized in numerous applications for example, wide-band mobile digital communication system. Continuous FFT transformer replaces the bank of (de)modulators for each individual sub-bearer and also by utilizing it extraordinarily reduces the hardware multifaceted nature and power utilization so only the system execution is just achievable.

The DFT of any factor length N, on unit circle, ascertains the tested Fourier Transform of a discrete-time arrangement at N equally disseminated points \( \omega_k = 2\pi k/N \).
The numerical condition underneath demonstrates the length-N forward DFT of an arrangement x(n):

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

\[
k = 0, 1, ..., N - 1
\]

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

\[
n = 0, 1, ..., N - 1
\]

The un-conventionality of the DFT calculation can basically decreased by utilizing brisk algorithms that usage settled rot of summation in equations (1) & (2) – despite mishandling distinctive symmetries natural in the baffling duplications. One such algorithm is the Cooley-Tukey radix-r demolition inconsistency FFT (DIF), which iteratively separates length r input arrangements into N/r arrangements and requires log r N stages of estimations. With the information, same hardware ordinarily shares in every single stage in the progressive system of the decomposition as shown in Figures 1-2.

It is read from m/m, experienced FFT processor and into m/m rewritten. It is required that each step through the FFT processor be carried out in successful times. The popular options of the base are r = 2, 4 and 16. Expanding decomposition base prompts decrease in the quantity of goes required through the FFT processor to the detriment of gadget assets.

In communications applications, the Field Programmable Gate Array (FPGA) devices are progressively being utilized for hardware usage from a hardware viewpoint. FPGAs in trend setting innovation hubs can accomplish superior while having greater adaptability, quicker structure time and lower cost. All things considered, FPGAs are winding up progressively alluring for FFT processing applications.

The FFT pipeline processor is a particular class of processors for the figuring of DFT using quick algorithms. It is portrayed by real-time processing and without interferences as the grouping of information that goes to the processor.

Here we are implementing the architecture called RSSDF (Radix-2 Single path Delay Feedback). On a Clock frequency of the input data sampling, pipelined FFT processor is described by non-halting. A lower clock frequency is a reasonable favorable position for pipeline designs when either a high-speed processing or low power arrangement is looked for. Like-wise, the pipeline structure is exceptionally standard, which can be effectively scaled and parameterized when Hardware Description Language (HDL) is utilized in the design. It is additionally increasingly adaptable when transforms of various lengths are to be figured with a similar chip.

2. Proposed Methodology

Architecture we are implementing RSSDF (Radix-2 Single path Delay Feedback). This architecture stores the output
of butterfly in shift registers feedback by utilizing more
registers. Figure 3 shows the proposed block diagram of
variable point pipelined FFT processor.

An data stream experiences every multiplier at each
stage independently. R2MDC also consist the same
number of butterfly units and multipliers as like RSSDF,
but RSSDF with very less reduced memory requirements:
N-1 registers\(^{24}\). Figure 4 shows the internal architecture
of each stage of FFT, it shows when the valid in is high
butterfly starts to store the data into FIFO. Internal control
logic in butterfly consist of three states the 1st state, the
butterfly receives real and imaginary data and stores the
received data into FIFO until it is full.

- The data will go to the next state once the FIFO is
full. 1st state, butterfly receives input data as well as
reads data from the FIFO and generates 2 outputs
by processing these 2 inputs a 2nd state, in butterfly

![Proposed top-level block diagram of variable
point pipelined FFT processor.](image)

Figure 3.

- The State wills increments to the next state after
receiving entire input data by depending on the stage.
3rd state, wherein the subtraction results which is put
away in the FIFO in past state are sent to the following
state. The received data is stored in FIFO again if valid
in is high.

- Output i.e. Data out (Real and Imaginary) are con-

ected as Data in (Real and Imaginary) inputs to the
next stages and the output signal valid out connected
as valid in inputs to the next stages. Like this, all the
10 stages of 1024 point FFT are connected and work.

3. Results and Discussion

Following are the simulation results for 8-point FFT
processor. Figure 5-6 shows when valid in is kept high
realin and image in inputs are given. When valid out
goes high real out and image out can be seen. Figure 7
shows the values 1,2,3,4,5,6,7,8 are given as inputs to both
verilog test bench and MATLAB code.

Figure 8(a) shows the 1024-point FFT input (analog
view) of verilog code and Figure 8(b) shows the 1024-
point FFT output of verilog code. Figure 9-10 shows the
1042-Point FFT Verilog comparison with MATLAB.
Both MATLAB and Verilog codes output was plotted and
can conclude that the output matches. There is a loss of
precision and noise in Verilog output because 16 point
decimal format was used and compared to MATLAB

![Internal architecture of every stage of FFT.](image)

Figure 4. Internal architecture of every stage of FFT.
Figure 5. Flow diagram for the operation of each stage.

Figure 6. 8-Point FFT results.

Figure 7. Inputs for 8-Point FFT verilog test bench and MATLAB code.

Figure 8. (a) 1024-point FFT input, (b) 1024-point FFT output.

Figure 9. 1042-Point FFT verilog comparison with MATLAB.
which uses 64 bit IEEE floating point format, hence this loss in accuracy. But the thing which is important is getting the peaks at the correct frequency conforms working of the code which is achieved.

Figure 11 is the snapshot of frequency achieved. Maximum operation frequency achieved was 236.189MHz and clock period is 4.234ns on a Virtex5 of FPGA, OFDM system IFFT and FFT in its transmitter and receiver. IFFT block receives the input data and converts to time domain data. Figure 12(a) shows the time domain data. FFT block receives the time domain data and converts to spectrum data. Figure 12(b) shows the spectrum data of OFDM system.

Bit error rate is the no. of bit errors isolated by the complete no. of moved bits during a contemplated time interim. Bit error ratio is a unit less execution measure, Figure 13 shows BER of an OFDM system.

4. Conclusion

This project clarified the usage of FFT IP and OFDM in Xilinx ISE focusing on a specific family of FPGA. The execution of the proposed algorithm ought to perform
superior to anything the base algorithm with the all-out basic way getting additionally advanced along these lines expanding the speed of activity. Maximum operation frequency achieved was 236.189MHz but in base paper maximum operation frequency was 30Mhz.

The systems utilized are Internal and external Pipelining of modules and Distributed LUT based idea. The target of the above methods is to lessen the basic way deferral and increment the general speed of activity of structure. The disadvantage is the increments in area and output latency. Area isn't of a much issue as present days FPGAs have immense measure of assets. Output latency should be taken into consideration with additional rationale at whatever point FFT is coordinated with other system modules.

5. References