ISSN (Online) 2456 -1304

International Journal of Science, Engineering and Management (IJSEM)
Vol 6, Issue 11, November 2021

Pipelined FFT Processor Power Optimization

[1] Mrs. Vani H, [2] Ravi Kumar K, [3] LakshmiNarayana Choudary, [4] Sharath Kumar, [5] Hareesh Babu

[1] Assistant Professor, Department of ECE, RYMEC, Ballari, Karnataka, India
[2][3][4][5] UG student, Department of ECE, RYMEC, Ballari, Karnataka, India
Email: [1] nani.vani36@gmail.com, [2] rav.ece.rymec@gmail.com, [3] Choudary.ece.rymec@gmail.com,
[4] sharathk.ece.rymec@gmail.com, [5] hareesh.ece.rymec@gmail.com

***Abstract---* This project presents the pipelined Fast Fourier Transform (FFT) processor power optimization. Pipelined FFT processor consists of several sub-modules like data buffer, shifter, and rotator (butterfly) which has introduced power consumption to the circuit in a hierarchical design. The objectives of this project are, first, to study the power consumption in term of power during the hierarchical condition for different type of pipelined FFT and next, the objective is to review the facility saving after the optimization process, where the planning is flattened without sub-modules. This project focuses on 64-point pipelined FFT radix-8 algorithms. The design process is in Verilog coding and simulation is in ISIM. Total power for before and after the optimization process is compared. However, all pipelined FFT show lower power consumption after the optimization process.**

***Keywords*--- pipelined, FFT, power, radix-8**

I. INTRODUCTION

An FFT algorithm uses a divide-and-conquer approach to scale back the computation complexity for DFT, i.e., one big problem is divided into a lot of different smaller problems that in the end are assembled to the solution of the original problem. In a communication system that uses an FFT algorithm there's also a requirement for an IFFT algorithm. Since the DFT and therefore the IDFT are similar both of those are often computed using basically an equivalent FFT HW, swap the important and imaginary parts of the input, compute the FFT and swap the important and imaginary data of the output. The output is now the IFFT of the input file, apart from the scaling think about the IFFT algorithm, 1/N. Usually this is not a problem, and this will therefore not be discussed henceforth.

Traditionally, direct implementation of N-point discrete Fourier transform (DFT) requires a complexity that is O (N2). Cooley-Turkey FFT algorithm published in 1965 is an efficient algorithm, exploiting the symmetry and periodicity properties of DFT to significantly lower its computational requirements to only O (N log N) computations [1]. As the semiconductor technology evolves, other technologies also flourish, especially embedded system and system-on-chip (SoC). This evolution is making the hardware implementation of FFT possible.

FFT processor is a hardware implementation for the FFT algorithm. The design of FFT processors focuses on how to proficiently map the FFT algorithms to the hardware to accommodate the serial input for computation. Some of the architecture commonly used are pipelined, memory, parallel and lots of more. Pipelined has widely adopted architecture for FFT [2]. The reason why pipelined is a very popular choice is that it has the advantage of parallelism and pipelining, making the architecture very fast [3]

This paper proposed a pipelined FFT processor with optimize power. Two differences pipelined FFT processor algorithms used are radix-4 and radix-8[4]. These two radix algorithms have better arithmetic operations and data transfer when compared to the radix-2 algorithm [5]. Since application such as wireless communication, portable computation, and many more that are suitable for FFT processor implementation, there is need to design and develop high speed and low power FFT processors [6]. Further, this paper presents an analysis of hardware utilization and performance of different points above mention FFT architecture and algorithms.

II. PROBLEM STATEMENT AND SOLUTION

The 8-point DFT, named as the base FFT operation, is implemented by the Winograd FFT Algorithm, which provides the minimum additions and multiplications (only 2 complex multiplications to the factor *W*81). As a result, the radix-8 FFT algorithm needs only 64 Complex multiplications to the twiddle factors *W*64 and 32 multiplications to the twiddle Factor *W*81 except of 4096 complex multiplications in the origin DFT. Note that the well-known radix-2 64-point FFT algorithm needs 192 complex multiplications

III. LITERATURE SURVEY

To increase the performance of the communication systems especially in OFDMA systems, they using high performance FFT algorithm for OFDMA Modulator and demodulator for their system [1]. Using radix 22 DIF (Decimation In Frequency) FFT algorithm they are implementing new architecture Radix 22 SDF. It uses two types of butterfly's architecture methods. One is same as SDF; second one is trivial twiddle factor multiplication method. They have implemented for N=1024.They achieved low power consumptions using butterfly method; speed up the design clock frequency and area improvement in OFDM Systems.

In paper [2], they designed OFDM baseband section model for IEEE 802.11p standard. Here OFDM Uses 64-sub carriers. The OFDM contains S/P – P/S Conversion, QPSK Mapper /Demapper, 64-point FFT/IFFT using Butterfly structure, pilot insertion for boosting and CP Insertion/Removal. They are using combination circuits for designing the models, improve the system accuracy but requires the large amount of area. Finally, they suggest going with sequential circuits for designing the models for IEEE 802.11p standard. The paper [3-4] shows the design of OFDM Transmitter / Receiver module for High Speed applications in communication systems. They mainly highlighting DIF 8-point FFT/IFFT throughout the design, using multiple path modules and common path modules which is a direct method same as butterfly method. At the end they are verifying the FFT/IFFT results with MATLAB Design.

In paper [5], they have designed a single path delay feedback Radix 22 Pipelined FFT/IFFT Processor. The design is same as [2] But they include additionally the TFM Structure in butterfly structure (BF-II). It is a fully pipelined complex multiplier used to multiply the twiddle factor by the output of BF-II. This design can maintain the SNR throughout the pipeline stages of FFT/IFFT. Finally, they achieved the High Frequency and low latency design for communication systems.

IV. OBJECTIVE

To design the develop

- 64 -point radix-8 pipelined FFT.
- Pipelined mode operation, each result is outputted in one clock cycle.

V. SYSTEM DESIGN

FFT is an algorithm for the effective Discrete Fourier Transform calculation.

Discrete Fourier Transform (DFT) may be fundamental digital signal processing algorithm used in many applications, including frequency analysis and frequency domain processing. DFT is the decomposition of a sampled signal in terms of sinusoidal (complex exponential) components. The symmetry and periodicity properties of the DFT are exploited to significantly lower its computational requirements. The resulting algorithms are known as Fast Fourier Transforms (FFTs). A 64-point DFT computes a sequence *x(n)* of 64 complex valued numbers given another sequence of data *X(k)* of length 64 according to the formula.

$$X(k) = \sum_{n=0}^{63} x(n)e^{-j2\pi nk/64} \quad ; \quad k = 0 \text{ to } 63.$$

To simplify the notation, the complex-valued phase factor *e –j2pnk/*64 is usually defined as *W*64*n* where: *W*64 = cos(2p/64) – j sin(2p/64). The FFT algorithms take advantage of the symmetry and periodicity properties of *W*64*n* to greatly reduce the number of calculations that the DFT requires. In an FFT implementation the real and imaginary components of *WnN are* called twiddle factors.
The basis of the FFT is that a DFT can be divided into smaller DFTs.
In the processor FFT64 a radix-8 FFT algorithm is used. It divides DFT into two smaller DFTs of the length 8, as it is shown in the formula:

$$X(k) = X(8r+s) = \sum_{m=0}^{7} W_8^{mr} W_{64}^{ms} \sum_{l=0}^{7} x(8l+m)\, W_8^{sl}, \; r = 0 \text{ to } 7, \; s = 0 \text{ to } 7,$$

which shows that 64-point DFT is divided into two smaller 8-point DFTs. The input complex data *x*(*n*) is represented by the 2-dimensional array of data *x*(8*l*+*m*). The columns of this array are computed by 8-point DFTs. The results of them are multiplied by the twiddle factors *W*64 *ms*. And the resulting array of data *X*(8*r*+*s*) is derived by 8-point DFTs of rows of the intermediate result array. The 8-point DFT, named as the base FFT operation, is implemented by the Winograd FFT algorithm, which provides the minimum additions and multiplications (only 2 complex multiplications to the factor *W*81). As a result, the radix-8 FFT algorithm needs only 64 complex multiplications to the twiddle factors *W*64*ms* and 32 multiplications to the twiddle factor *W*81 except of 4096 complex multiplications in the origin DFT. Note that the well-known radix-2 64-point FFT algorithm needs 192 complex multiplications.

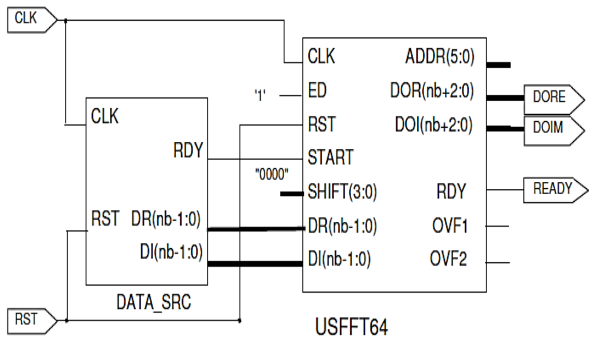


Fig. System Interconnection

Components:

BUFRAM64 – data buffer with row writing and column reading;

FFT8 – data path, which calculates the 8-point DFT;

CNORM – shifter to 0,1,2,3-bit left shift;

ROTATOR64 – complex multiplier with twiddle factor ROM;

CT64 – counter modulo 64.

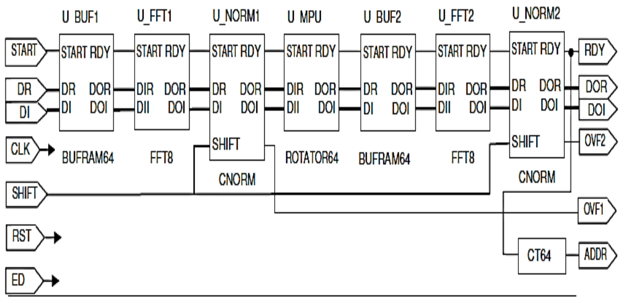


Fig. Internal Block Diagram

Highly pipelined calculations

Each base FFT operation is computed by the computational unit, called FFT8, which is the data path for FFT calculations. FFT8 calculates the 8-point DFT in the high pipelined mode. Therefore, in each clock cycle one complex number is read from the input data buffer RAM and the complex result is written in the output buffer RAM. The 8-point DFT algorithm is divided into several stages which are implemented in the stages of the FFT8 pipeline. This

supports the increasing the clock frequency up to 250 MHz and higher. The latent delay of the FFT8 unit from input the first data to output the first result is equal to 14 clock cycles.

High precision computations

In the core the inner data bit width is higher to 4 digits than the input data bit width. The main error source is the result truncation after multiplication to the factors *W*64 *ms*. Because the most of base FFT operation calculations are additions, they are calculated without errors. The FFT results have the data bit width which is higher in 3 digits than the input data bit width, which provides the high data range of results when the input data is the sinusoidal signal. The maximum result error is less than the 1 least significant bit of the input data. Besides, the normalizing shifters are attached to the outputs of FFT8 pipelines, which provide the proper bandwidth of the resulting data. The overflow detector outputs provide the opportunity to input the proper shift left bit number for these shifters.

Low hardware volume

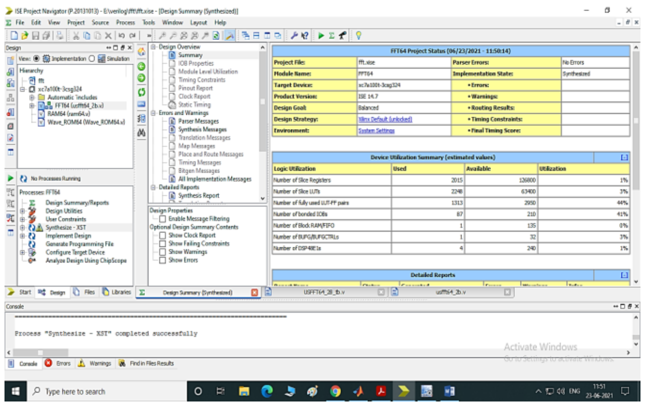
The FFT64 processor has the minimum multiplier number which is equal to 4. This fact makes this core attractive to implement in ASIC. When configuring in Xilinx FPGA, these multipliers are implemented in 4 DSP48 units respectively. The user can select the input data, output data, and coefficient widths which provide application dynamic range needs. This can minimize both logic hardware and memory volume.

Software details

This paper uses the following software

- ***XILINX Software Tool***

Software Overview The ISE® software controls all aspects of the design flow. Through the Project Navigator interface, you can access all of the design entry and design implementation tools. You can also access the files and documents associated with your project. Project Navigator Interface By default, the Project Navigator interface is divided into four panel sub-windows, as seen in Figure 2-1. On the top left are the Start, Design, Files, and Libraries panels, which include display and access to the source files in the project as well as access to running processes for the currently selected source. The Start panel provides quick access to opening projects as well as frequently access reference material, documentation and tutorials. At the bottom of the Project Navigator are the Console, Errors, and Warnings panels, which display status messages, errors, and warnings. To the right is a multi-document interface (MDI) window referred to as the Workspace. The Workspace enables you to view design reports, text files, schematics, and simulation waveforms. Each window can be resized, undocked from Project Navigator, moved to a new location within the main Project Navigator window, tiled, layered, or closed. You can use the View > Panels menu commands to open or close panels. You can use the Layout > Load Default Layout to restore the default window layout.



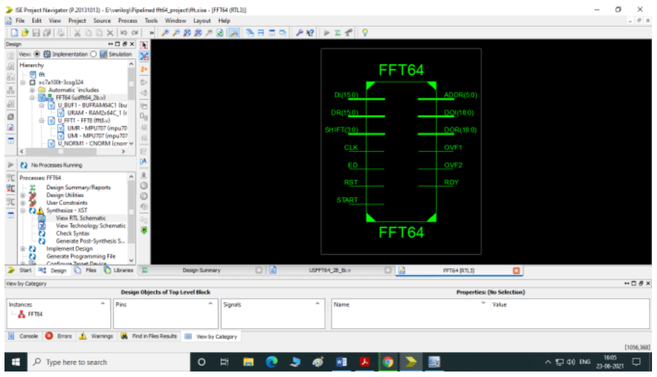
FFT64 Project Status (06/23/2021 - 11:50:14)			
Project File:	fft.xise	Parser Errors:	No Errors
Module Name:	FFT64	Implementation State:	Synthesized
Target Device:	xc7a100t-3csg324	• Errors:	
Product Version:	ISE 14.7	• Warnings:	
Design Goal:	Balanced	• Routing Results:	
Design Strategy:	Xilinx Default (unlocked)	• Timing Constraints:	
Environment:	System Settings	• Final Timing Score:	

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slice Registers	2015	126800	1%
Number of Slice LUTs	2248	63400	3%
Number of fully used LUT-FF pairs	1313	2950	44%
Number of bonded IOBs	87	210	41%
Number of Block RAM/FIFO	1	135	0%
Number of BUFG/BUFGCTRLs	1	32	3%
Number of DSP48E1s	4	240	1%

```
Process "Synthesize - XST" completed successfully
```

VI. EXPERIMENTAL RESULTS

Total power consumption in each circuit is a summation of a function of switching activity, transistor structure, capacitance and voltage. In general, power consumption in a circuit divided into two components, which are dynamic and static power. Dynamic power is made up of summation of switching power and short-circuit power. Switching power occurs when internal and net capacitance charging and discharging. While short-circuit power is the power degenerate by sudden short-circuit connection between the supply voltage and ground, during the state switching at the gate. Another power composing the overall power consumption is static power or also known as leakage power. Dynamic power is a major concern when dealing with area more than 90nm, but for a smaller area, leakage power has become the dominant power consumer. Dynamic power dissipated when switching activity occurs, but leakage power is continuous due to leakage current. Studies on power consumption are very important because one out of five chip produces fail due to excessive power consumption.



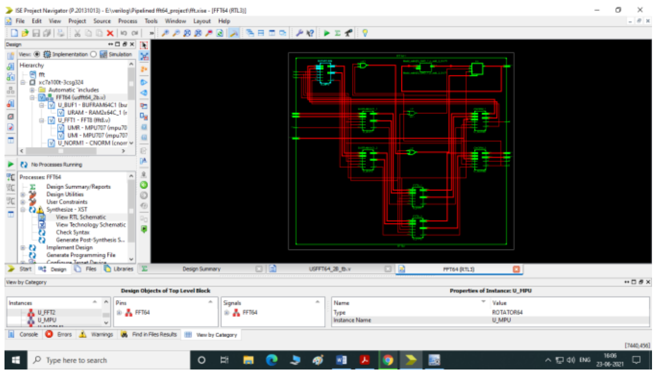
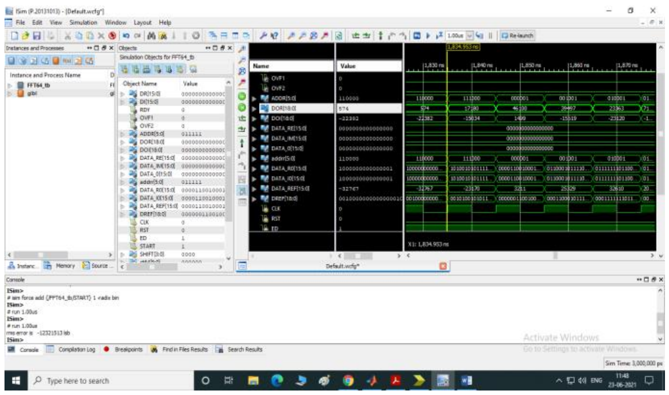


Fig. shows the RTL view of pipelined FFT before optimization process. This figure shows, the sub-modules in pipelined FFT are in hierarchical form. Using compile-ultra command in Design Compiler, the design is optimized. Design Compiler performs area-based auto ungrouping before initial mapping; in other words, the design is flattened.



Advantages, disadvantages and applications

Advantages

- ➢ Cost efficient
- ➢ Low Power
- ➢ Low Area
- ➢ Efficient
- ➢ Fast Computation

Disadvantages

- ➢ Speed is limited as compared with parallel FFT computers.

Applications

1. OFDM Communication Systems
2. FFT Computation Systems.

ISSN (Online) 2456 -1304

International Journal of Science, Engineering and Management (IJSEM)
Vol 6, Issue 11, November 2021

VII. CONCLUSION

This paper proposed pipelined Fast Fourier Transform (FFT) processor power optimization. The optimization process is circuit reduction process where some unused or similar function registers are removed. Basically, removing all hierarchy within the pipelined FFT module into one flat design. It is often concluded from results obtains that each one pipelined FFT during this paper have more power saving after the optimization process. Overall, 64-point pipelined FFT with radix-8 has the best power saving.

REFERENCES

[1] L. P. Thakare and A. Y. Deshmukh, “Area Efficient FFT/IFFT Processor Design for MIMO OFDM System in Wireless Communication,” *Int. Conf. Emerg. Trends Eng. Technol. ICETET*, vol. 2016-March, pp. 10–13, 2016.

[2] R. Neuenfeld, M. Fonseca, and E. Costa, “Design of optimized radix-2 and radix-4 butterflies from FFT with decimation in time,” *LASCAS 2016 - 7th IEEE Lat. Am. Symp. Circuits Syst. R9 IEEE CASS Flagsh. Conf.*, pp. 171–174, 2016.

[3] S. L. M. Hassan, N. Sulaiman, H. Jaafar, A. Saparon, and Y. M. Yussoff, “Implementation of pipelined FFT processor on FPGA microchip proposed for mechanical applications,” *J. Mech. Eng.*, vol. SI 2, no. 2, pp. 145–156, 2017.

[4] M. Rajasekhar, K. Manjula, M. T. Scholar, and E. Systems, “Design and Simulation of 512 Point FFT using RADIX-8 Algorithm,” vol. 05, no. 04, pp. 30–33, 2016.

[5] N. Sulaiman, “Design of a reconfigurable FFT processor using Multi-objective Genetic Algorithm,” *2010 Int. Conf. Intell. Adv. Syst.*, pp. 1–5, Jun. 2010.