IJIRSET
International Journal of Innovative Research in
SCIENCE | ENGINEERING | TECHNOLOGY

e-ISSN: 2319-8753 | p-ISSN: 2347-6710

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

Volume 11, Issue 6, June 2022

ISSN INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 8.118

9940 572 462 | 6381 907 438 | ijirset@gmail.com | www.ijirset.com

International Journal of Innovative Research in Science, Engineering and Technology (IJIRSET)

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 8.118|

||Volume 11, Issue 6, June 2022||

| DOI:10.15680/IJIRSET.2022.1106283|

Operating Frequency Improvement of FPGA Implementation on a Pipeline Using Large-FFT Processor

Mr. B. Ilamparithi, Dr. C. Selvi, Dr. D. Sasikala

Student, Department of Electronics & Communication Engineering, Muthayammal Engineering College, Namakkal, Tamil Nadu, India

Associate Professor, Department of Electronics & Communication Engineering, Muthayammal Engineering College Namakkal, Tamil Nadu, India

Professor, Department of Electronics & Communication Engineering, Muthayammal Engineering College, Namakkal, Tamil Nadu, India

ABSTRACT: The proposed method, circuit complexity reduction in FPGA implementation of large N-point Radix-22 FFT with single-path delay feedback architecture is reported. Memory requirement of the FFT in the FPGA consists of two parts, the RAM data storage of the feedback in each stage of the data flow and the twiddle factors prepared as ROM for each complex multiplication. Through address rearrangement, the ROM sizes for the twiddle factors are significantly reduced with the removal of redundancy. As a result, the signal critical path is reduced, and the system clock frequency is increased. The proposed architecture is validated by the implementations of spartan 6 FPGA development board series.

I. INTRODUCTION

The Fast Fourier Transform (FFT) processor is one of the core components in signal processing. Small N-point FFT implementations are commonly found. An example in daily life is the Wi-Fi chips which are produced in millions annually, having either 64-point or 256-point FFT core implemented. The requirement of large N-point FFT is found in niche areas such as astrophysics and radar applications. In recent years, there have been applications of large N-point FFT in consumer automotive radar, software-defined radio and ultra-wideband spectrum scanning, etc. In the direct conversion of radio frequency (RF) signal to digital form by various high-speed analog-to-digital converters or even with down-conversion to bring the RF signal to baseband, the wide bandwidth requirement and the fast real-time applications have favored the application-specific integrated circuit (ASIC) or FPGA implementation over general purpose processors. The challenges are the processing speed and the cost involved. For example, the Sparse Fast Fourier Transform (SFFT) claims achieving million-point FFT per second with a constraint of maximum 500 non-zero frequency coefficients. Its architecture includes the Radix22 single-path delay feedback (SDF) decimation in frequency (DIF) 4096-point FFT with a maximum operating frequency of 121.2 MHz There are other studies to reduce the circuit complexity with algorithms to minimize the size and power of the FFT processor using coefficient memory reduction and switching activity analysis schemes. The architecture of Radix-22 SDF DIF N-point FFT is considered as one with the least hardware resource requirement in ultra-wideband scanning. This architecture is claimed to possess the features of simple control, N −1 data memory size requirement with log4 N −1 complex multipliers and 4 log4 N adders. The twiddle factors may either be generated within the process using CORDIC algorithm or in advance and stored as Read Only Memory (ROM) which is easier in the implementation. In large N-point FFTs, the increase of memory size is significant and reduction mechanism is necessary. With pipeline process, the individual tasks per clock cycle are simple enough to support the increase of system clock frequency. However, the clock-cycle delays introduced in the data flow upset the arrangement of simple counter for the synchronization of various controls. Thus, an alternative arrangement is needed and is proposed in this paper. In this paper, we begin with a brief explanation of the FFT architecture and describe the implementation of individual modules. Fixed-point arithmetic operation within the FFT is adopted. Scaling is introduced to convert the twiddle factors into integer values and the round-off error is reduced. The reduction of the ROM sizes of the twiddle factors for the complex multiplications through address re-arrangement is explained.

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 8.118|

||Volume 11, Issue 6, June 2022||

| DOI:10.15680/IJIRSET.2022.1106283|

Implementations of 1K- and 4K-point FFTs with memory arrangement under the same architecture are presented. Controls with different delays are introduced to match the clock-cycle delays in the data flow pipeline.

OBJECTIVE

The proposed method, circuit complexity reduction in FPGA implementation of large N-point Radix-22 FFT with single-path delay feedback architecture is reported. Memory requirement of the FFT in the FPGA consists of two parts, the RAM data storage of the feedback in each stage of the data flow and the twiddle factors prepared as ROM for each complex multiplication.

II. EXISTING SYSTEM

A power efficient hardware architecture implementing the Split-Radix Fast Fourier Transform (SRFFT) is developed through pruning unnecessary computations. SRFFT is known to offer a performance that is better than conventional FFT in terms of reduced number of required complex multiplications and hence, can lead to reduced power consumption. Leveraging this potential, a new architecture of a configurable SRFFT processor is first devised and then the architecture is developed further so that unnecessary computations, which yield zeros at the output, are pruned. This is done through stalling butterfly computations with appropriate use of a pruning matrix. The proposed processor may find applications in orthogonal frequency division multiplexing (OFDM) communication transceivers, where transmitted signals may occupy only a small portion of the whole operational spectrum. The processor is implemented on a field-programmable gate array and simulations show that maximum power saving of around 20% is achieved when computing 1024-point Fourier transform of signals with very sparse spectrum.

DISADVANTAGE:

- High in computational time
- Un compactable

III. PROPOSED SYSTEM

The proposed method, circuit complexity reduction in FPGA implementation of large N-point Radix-22 FFT with single-path delay feedback architecture is reported. Memory requirement of the FFT in the FPGA consists of two parts, the RAM data storage of the feedback in each stage of the data flow and the twiddle factors prepared as ROM for each complex multiplication. Through address rearrangement, the ROM sizes for the twiddle factors are significantly reduced with the removal of redundancy. As a result, the signal critical path is reduced, and the system clock frequency is increased. The proposed architecture is validated by the implementations of spartan 6 FPGA development board series.

ADVANTAGES

- Low computational time
- Very compact

IV. DESIGN AND ANALYSIS OF FPGA IMPLEMENTATION OF A PIPELINE LARGE-FFT PROCESSOR

There are many different FFT algorithms involving a wide range of mathematics, from simple complex-number arithmetic to group theory and number theory; this article gives an overview of the available techniques and some of their general properties, while the specific algorithms are described in subsidiary articles linked below.

The DFT is obtained by decomposing a sequence of values into components of different frequencies. This operation is useful in many fields (see discrete Fourier transform for properties and applications of the transform) but computing it directly from the definition is often too slow to be practical. An FFT is a way to compute the same result more quickly: computing the DFT of N points in the naive way, using the definition, takes O(N2) arithmetical operations, while an FFT can compute the same DFT in only ON (log N) operations. The difference in speed can be enormous, especially for long data sets where N may be in the thousands or millions. In practice, the computation time can be reduced by several orders of magnitude in such cases, and the improvement is roughly proportional to N log N. This huge improvement made the calculation of the DFT practical; FFTs are of great importance to a wide variety of

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 8.118|

||Volume 11, Issue 6, June 2022||

| DOI:10.15680/IJIRSET.2022.1106283|

applications, from digital signal processing and solving partial differential equations to algorithms for quick multiplication of large integers.

The best-known FFT algorithms depend upon the factorization of N, but there are FFTs with ON (log N) complexity for all N, even for prime N. Many FFT algorithms only depend on the fact that is an N-th primitive root of unity, and thus can be applied to analogous transforms over any finite field, such as number-theoretic transforms. Since the inverse DFT is the same as the DFT, but with the opposite sign in the exponent and a 1/N factor, any FFT algorithm can easily be adapted for it.

V. THE PIPELINE FFT PROCESSOR ARCHITECTURE

5.1 GENERAL:

The Radix-22 FFT equation using the common factor algorithm (CFA) to decompose the twiddle factors is shown as

$$X(k_1+2k_2+4k_3)=\sum_{n_3=0}^{N/4-1}\{H(k_1,k_2,n_3)W_N^{n_3(k_1+2k_2)}\}W_{N/4}^{n_3k_3} \quad (1)$$

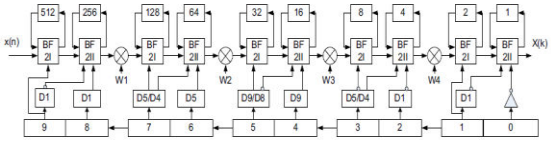

Fig. 1: Radix 2^2 SDF DIF pipeline FFT architecture for $N = 1024$.

with k1, k2 ∈ {0, 1}. H(k1, k2, n3) is named as Butterfly-II and can be expressed as

$$H(k_1,k_2,n_3)=B_{N/2}^{k_1}(n_3)+(-1)^{(k_1+2k_2)}B_{N/2}^{k_1}(n_3+N/4) \quad (2)$$

where $B_{N/2}^{k_1}(n_3)$ and $B_{N/2}^{k_1}(n_3+N/4)$ are named as Butterfly-I with

$$B_{N/2}^{k_1}(n_3)=x(n_3)+(-1)^{k_1}x(n_3+(N/2)) \quad (3)$$

$$B_{N/2}^{k_1}(n_3+N/4)=x(n_3+(N/4))+(-1)^{k_1}x(n_3+(3/4)N). \quad (4)$$

Based on the above equations, the SDF pipeline FFT architecture is divided into several modules shown in Fig. 1. Referring to the figure, data are delayed and passed to the butterfly operation module via the shift-register feedback. The two butterfly operations are equivalent to the Radix-4 FFT butterfly operation and are followed by the non-trivial complex multiplication with the twiddle factor. These modules form one stage. There are log4 N stages with different parameters and the last stage has no complex multiplication. Selection of the word length of the multiplication products are based on the balance between the minimization of round-off error and the circuit complexity. The twiddle factors are prepared in advance and stored in the ROMs. The sizes are reduced by the redundancy elimination.

A. ROM Approach of The Twiddle Factors Generation with Address Re-arrangement

The values of the twiddle factors are prepared in advance with MATLAB in HEX format and stored in the ROMs. The twiddle factors W n3(k1+2k2) N from (1) with k1, k2 ∈ {0, 1} and n3 = 0, .., N/4−1 can be categorized into four groups — W0, W2k, Wk and W3k each of size N/4. The data of each group are passed onto the next stage of FFT with N/4-point. Butterflies with the decomposed twiddle factors are shown in Fig. 2. In the next stage, the twiddle factor size of each set of FFT is 1/4 of N/4, i.e., N/42. The decomposition continues until it arrives at the 4-point FFT. Note that for the list of twiddle factors in the group W k (k = 0, .., N − 1), it is sufficient to record those for k = 0, .., N/4 − 1. As shown in Fig. 3, we define the real part and the imaginary part as Rk and −Ik, respectively, k = 0, .., N/4−1. By re-labeling these real parts

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 8.118|

||Volume 11, Issue 6, June 2022||

| DOI:10.15680/IJIRSET.2022.1106283|

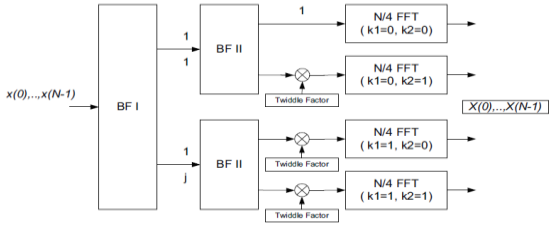

Fig. 2: Butterflies with decomposed twiddle factors

and imaginary parts as Rh and −Ih where h = 0, ., N/4− 1, the list of twiddle factors W k can be defined as

$$\begin{aligned} W^{k}{}_{(k=0,..,N/4-1)} &= R_k - jI_k = R_h - jI_h \\ {}_{(k=N/4,..,N/2-1)} &= -I_{k-N/4} - jR_{k-N/4} = -I_h - jR_h \\ {}_{(k=N/2,..,3N/4-1)} &= -R_{k-N/2} + jI_{k-N/2} = -R_h + jI_h \\ {}_{(k=3N/4,..,N-1)} &= I_{k-3N/4} + jR_{k-3N/4} = I_h + jR_h. \quad (5) \end{aligned}$$

It is thus obvious that only 1/4 of the data within the group is necessary. The other lists of twiddle factor group W 2k and W3k can also be represented by the same sets of values {Rh} and {−Ih} with some modifications. The values for the first half of the list of the twiddle factor group W 2k can be directly obtained from the two sets of values. The second half, which is on the second quadrant of the unit circle, is mapped back with the address pointers to the ROM as 2k − N/4, i.e., the pointer to N/4 is pointed back to 0, the first location of the list. The modification of the values is the swapping of assignment and 2's complement operation, i.e., −Ih − jRh. The list of twiddle factor group W 3k is divided into three parts. The first and second part follow the same arrangements. The third part with values in the third quadrant of the unit circle are mapped back with address pointers to the ROM as 3k − N/2. The modification is 2's complement operation, i.e., −Rh + jIh. The total ROM size before reduction is N × (log4 N −1)

APPLICATIONS:

- Digital signal processing
- Mobile phone
- Satellite application

SOFTWARE REQUIREMENT

- **Verification Tool**
  - Modelsim 6.4c
- **Synthesis Tool**
  - Xilinx ISE 14.5

MODELSIM

ModelSim SE - High Performance Simulation and Debug

ModelSim SE is our UNIX, Linux, and Windows-based simulation and debug environment, combining high performance with the most powerful and intuitive GUI in the industry.

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 8.118|

||Volume 11, Issue 6, June 2022||

| DOI:10.15680/IJIRSET.2022.1106283|

SYNTHESIS TOOL:

XILINX ISE

INTRODUCTION

For two-and-a-half decades, Xilinx has been at the forefront of the programmable logic revolution, with the invention and continued migration of FPGA platform technology. During that time, the role of the FPGA has evolved from a vehicle for prototyping and glue-logic to a highly flexible alternative to ASICs and ASSPs for a host of applications and markets. Today, Xilinx® FPGAs have become strategically essential to world-class system companies that are hoping to survive and compete in these times of extreme global economic instability, turning what was once the programmable revolution into the "programmable imperative" for both Xilinx and our customers.

VI. SIMULATION IMPLEMENTATION

GENERAL:

Verilog HDL is a Hardware Description Language (HDL). A Hardware Description Language is a language used to describe a digital system, for example, a computer or a component of a computer. One may describe a digital system at several levels. For example, an HDL might describe the layout of the wires, resistors and transistors on an Integrated Circuit (IC) chip, i. e., the switch level. Or it might describe the logical gates and flip flops in a digital system, i. e., the gate level. An even higher level describes the registers and the transfers of vectors of information between registers. This is called the Register Transfer Level (RTL). Verilog supports all these levels. However, this handout focuses on only the portions of Verilog which support the RTL level.

VERILOG:

Verilog is one of the two major Hardware Description Languages (HDL) used by hardware designers in industry and academia. VHDL is the other one. The industry is currently split on which is better. Many feel that Verilog is easier to learn and use than VHDL. As one hardware designer puts it, "I hope the competition uses VHDL." VHDL was made an IEEE Standard in 1987, and Verilog in 1995. Verilog is very C-like and liked by electrical and computer engineers as most learn the C language in college. VHDL is very most engineers have no experience. Verilog was introduced in 1985 by Gateway Design System Corporation, now a part of Cadence Design Systems, Inc.'s Systems Division. Until May 1990, with the formation of Open Verilog International (OVI), Verilog HDL was a proprietary language of Cadence. Cadence was motivated to open the language to the Public Domain with the expectation that the market for Verilog HDL-related software products would grow more rapidly with broader acceptance of the language. Cadence realized that Verilog HDL users wanted other software and service companies to embrace the language and develop Verilog-supported design tools.

CODING IMPLEMENTATION SNAPSHOTS GENERAL:

Snapshot is nothing but every moment of the application while running. It gives the clear elaborated of application. It will be useful for the new user to understand for the future steps.

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 8.118|

||Volume 11, Issue 6, June 2022||

| DOI:10.15680/IJIRSET.2022.1106283|

Figure-1 ,2, 3 Represents control signals of the design with clock cycle delays inserted in all the stages

VARIOUS SNAPSHOTS:

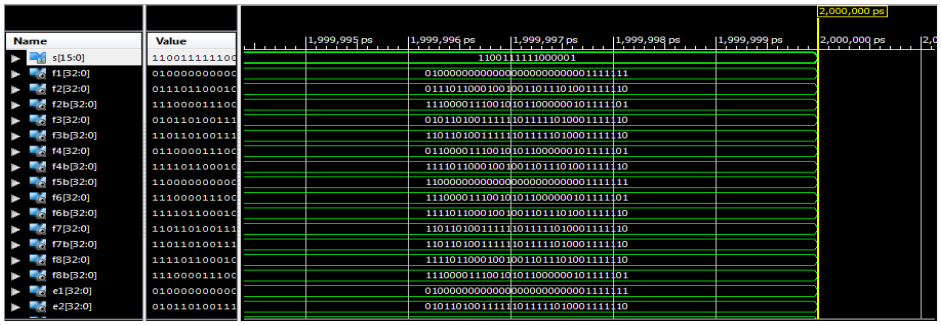

Figure-1

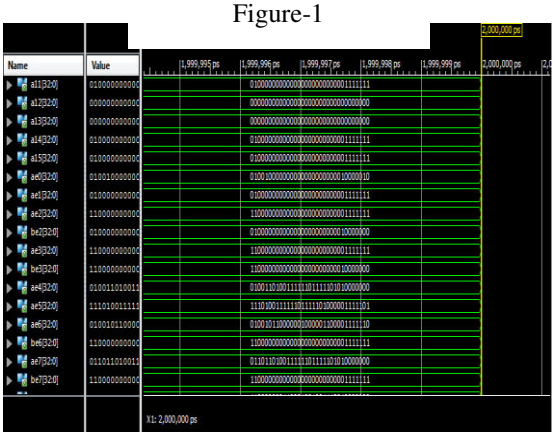

Figure-2

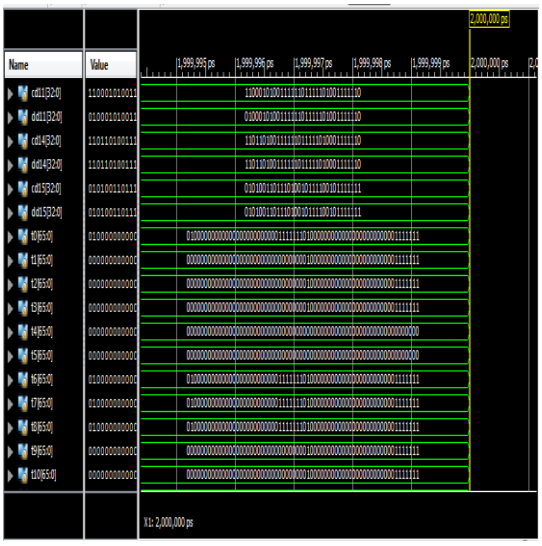

Figure-3

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 8.118|

||Volume 11, Issue 6, June 2022||

| DOI:10.15680/IJIRSET.2022.1106283|

VII. CONCLUSION

The proposed method, circuit complexity reduction in FPGA implementation of large N-point Radix-22 FFT with single-path delay feedback architecture is reported. Memory requirement of the FFT in the FPGA consists of two parts, the RAM data storage of the feedback in each stage of the data flow and the twiddle factors prepared as ROM for each complex multiplication. Through address rearrangement, the ROM sizes for the twiddle factors are significantly reduced with the removal of redundancy. As a result, the signal critical path is reduced, and the system clock frequency is increased. The proposed architecture is validated by the implementations of spartan 6 FPGA development board series.

REFERENCES

[1] Altera Cyclone IV Device Handbook. Altera Corporation, 2014.
[2] R. J. Evans, P. M. Farrell, G. Felic, H. T. Duong, H V. Le, J. Li, M. Li, W. Moran, M. Morelande, and E. Skafidas, "Consumer radar: Technology and limitations," in International Conference on Radar, 2013, pp. 21–26.
[3] J. Brakensiek, B. Oelkrug, M.Bucker, D. Uffmann, A. Droge, M. Darianian, and M. Otte, "Software radio approach for re-configurable multistandard radios," in IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, vol. 1, 2002, pp. 110–114.
[4] F. K. Jondral, "Software-defined radio: Basics and evolution to cognitive radio," EURASIP Journal on wireless communications and networking, pp. 275–283, 2005.
[5] H. Abdollahifakhr, N. Belanger, Y. Savaria, and F. Gagnon, "Power efficient hardware architecture for computing Split-Radix FFTs on highly sparsed spectrum," in IEEE International New Circuits and Systems Conference (NEWCAS), 2015, 2015, pp. 1–4.
[6] F. Qureshi, S. A. Alam, and O. Gustafsson, "4k-point fft algorithms based on optimized twiddle factor multiplication for fpgas," in 2010 Asia Pacific Conference on Postgraduate Research in Microelectronics and Electronics (PrimeAsia). IEEE, 2010, pp. 225–228. [10] F. Qureshi and O. Gustafsson, "Twiddle factor memory switching activity analysis of radix-2 2 and equivalent fft algorithms," in Proceedings of 2010 IEEE International Symposium on Circuits and Systems. IEEE, 2010, pp. 4145–4148.
[7] H. Cho, M.-S. Kim, D.-B.Kim, and J.-U. Kim, "R2aˆ 2 sdf fft implementation with coefficient memory reduction scheme," in IEEE Vehicular Technology Conference. IEEE, 2006, pp. 1–4.
[8] S. Lee, D.-b. Kim, and S.-C. Park, "Power-efficient design of memorybased fft processor with new addressing scheme," in Communications and Information Technology, 2004. ISCIT 2004. IEEE International Symposium on, vol. 2. IEEE, 2004, pp. 678–681.
[9] A. Agarwal, H. Hassanieh, O. Abari, E. Hamed, D. Katabi et al., "High-throughput implementation of a million-point sparse Fourier Transform," in Proceedings of the 24th International Conference on Field Programmable Logic and Applications (FPL). IEEE, 2014, pp. 1–6.
[10] S. He and M. Torkelson, "Design and implementation of a 1024- point pipeline FFT processor," in Proceedings of the Custom Integrated Circuits Conference. IEEE, 1998, pp. 131–134.
[11] R. W. S. Alan V. Oppenheim, Discrete-Time Signal Processing (3rd Edition). Prentice-Hall, 2009.

INNO SPACE
SJIF Scientific Journal Impact Factor
Impact Factor: 8.118

doi cross ref

ISSN INTERNATIONAL STANDARD SERIAL NUMBER INDIA

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

9940 572 462 6381 907 438 ijirset@gmail.com

www.ijirset.com

Scan to save the contact details