International Journal for Innovative
Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

IJIEMR Transactions, online available on 20 May 2017. Link :
http://www.ijiemr.org/downloads.php?vol=Volume-6&issue=ISSUE-3

Title: Method Of Efficient Vlsi Architecture For Decimation-In-Time Fast Fourier Transform Of Real-Valued Data.

Volume o6, Issue 03, Pages: 236 – 245.

Paper Authors

***VOOTLA POORNA CHANDRIKA, ** MR. D.SRIDHAR (PH.D)**
Dept of ECE Sri Sunflower College of Engineering and Technology.

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

METHOD OF EFFICIENT VLSI ARCHITECTURE FOR DECIMATION-IN-TIME FAST FOURIER TRANSFORM OF REAL-VALUED DATA

***VOOTLA POORNA CHANDRIKA, ** MR. D.SRIDHAR (PH.D)**

*PG Scholar, Dept of ECE (VLSID), Sri Sunflower College of Engineering and Technology, Lankapalli , (A.P),India.
**Associate Professor,Head of the Department of ECE, Sri Sunflower College of Engineering and Technology, Lankapalli , (A.P),India.

Email Id: chandrika.vootla@gmail.com, sridhar.done@gmail.com.

ABSTRACT:

The decimation-in-time (DIT) fast Fourier transform (FFT) very often has advantage over the decimation-in-frequency (DIF) FFT for most real-valued applications, like speech/image/video processing, biomedical signal processing, and time-series analysis, etc., since it does not require any output reordering. Besides, the DIT FFT butterfly involves less computation time than its DIF counterpart. In this paper, we present an efficient architecture for the radix-2 DIT real-valued FFT (RFFT). We present here the necessary mathematical formulation for removing the redundancies in the radix-2 DIT RFFT, and present a formulation to regularize its flow graph to facilitate folded computation with a simple control unit. We propose here a register-based storage design which involves significantly less area at the cost of a little higher latency compared with the conventional RAM-based storage. The address generation for folded in-place DIT RFFT computation with register-based storage is challenging since both read and write operations are performed in the same clock cycle at different locations. Therefore, we present here a simple formulation of address generation for the proposed radix-2 DIT RFFT structure.

Keywords: Decimation-in-time FFT, fast Fourier transform (FFT), in-place computation, real-valued FFT.

I INTRODUCTION

In many applications, such as asymmetric digital subscriber line (ADSL) [1] and orthogonal frequency-division multiplexing (OFDM) [2], the transform length is required to be large and the previous DFT structures with computational complexity of are not practical for VLSI implementation. In recent literature, three low-cost and high-throughput systolic architectures have been presented [3]–[5] with a computational complexity of , which are regular and are suitable for VLSI implementation. Although these -point FFT algorithms are hardware efficient, they have long latency of clock cycles and their hardware utilization can be improved further. For example, the data reordering strategy used in [3] is the delay-feedback (DF) architecture [6] with a delay element utilization rate of 100%; but the utilization rate of its multipliers is just 50%. Assume we have a prefetch buffer to ensure concurrent read and write operations; the hardware can be more efficiently used [7]. By applying concurrent computation into the butterfly operations, this brief improves the throughput rate of the previously proposed FFT architectures by a factor of 2. High processing speed leads to reduced number of required delay elements since fewer intermediate results need to be stored. The latency can also be reduced by a factor of 2.The Fast Fourier Transform (FFT) is an efficient computation of the Discrete Fourier Transform (DFT) and one of the most important tools used in digital signal processing applications. Because of its well-structured form, the FFT is a benchmark in assessing digital signal processor (DSP) performance. The development of FFT algorithms has assumed an input sequence consisting of complex numbers. This is because complex phase factors, or twiddle factors, result in complex variables. Thus, FFT algorithms are designed to perform complex multiplications and additions. However, the input sequence consists of real numbers in a large number of real applications. The Fast Fourier transform (FFT) algorithm is frequently encountered in almost every application area of digital signal processing. There are several applications such as speech, audio, image, and video processing, where FFT is very often performed on real-valued signals. Efficient realization of FFT of real-valued signals has received further attention nowadays due to the emergence of biomedical signal processing, and wide applications of real-valued time-series analysis. FFT of realvalued signal exhibits conjugate symmetry which renders half the FFT outputs redundant [1]. The conjugate symmetry of realvalued FFT (RFFT) is used to compute FFTs of a pair of –point. Real-valued sequences from -point FFT of a complex-valued data.1 There has been a continued effort for several decades on reducing the hardware cost of VLSI implementation of FFT and improving its performance [2]. Some efforts have been made for efficient VLSI implementation of RFFT, as well.

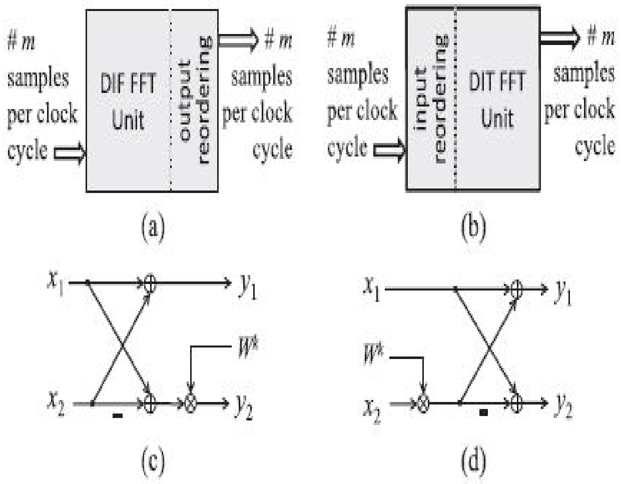


Fig. 1. The key differences in the DIT and DIF FFT computation. (a) DIF FFT processing. (b) DIT FFT processing. (c) DIF FFT butterfly. (d) DIT FFT butterfly. m is a power-of-2 integer.

II.LITERATURE SURVEY

DFT is one of the most important tools in the field of digital signal processing. Several Fast Fourier Transform (FFT) algorithms have been developed over the years due to its computational complexity. FFT plays a critical role in modern digital communications such as Digital Video Broadcasting (DVB) and Orthogonal Frequency Division Multiplexing (OFDM) systems. The design of pipelined architectures for computation of FFT of complex valued signals (CFFT) has been carried out. Different algorithms have been developed to reduce the computational complexity, of which Cooley-Tukey radix-2 FFT [1] is very popular. Algorithms such as radix-4 [2], split-radix [3] and radix-22 [4] have been developed based on the basic radix-2 FFT approach. The one of the most classical approaches for pipelined implementation of radix-2 FFT is Radix-2 multi- path delay commutator (R2MDC) [5]. A standard usage of the storage buffer in R2MDC leads to the Radix-2 Single-path delay feedback (R2SDF) [6] architecture with reduced memory. The architectures are developed for a specific-point FFT in [7] and [8], whereas hypercube theory is used to derive the architectures in [9].The method of developing these architectures from the algorithms is not well established. In additional, most of these hardware architectures are not fully utilized and require high hardware complexity. In the period of high speed digital communications, the high throughput and low power designs are essential to meet the speed and power requirements while keeping the hardware overhead to a minimum. In this paper, a new approach to design the architecture from the FFT flow graphs is presented. Folding transformation and register minimization techniques are used to derive several known FFT architectures. If the input samples are real then the spectrum is symmetric and approximately half of the operations are redundant. The applications such as speech, audio, image, radar, and biomedical signal processing, a specialized hardware implementation is best suitable to meet the real-time constraints. The implantable or portable device saves power by using this type of implementation which is a key limitation. Few pipelined architectures for real valued signals have been proposed based on the Brunn algorithm. However, these are not widely used. Different algorithms such as doubling algorithm, packing algorithm have been proposed for computation of RFFT. These approaches are based on removing the redundancies of the CFFT while the input is real. RFFT is calculated using the CFFT architecture in an efficient manner .

In the folding transformation, many butterflies in the same column can be mapped to one butterfly unit. If the FFT size is N, a folding factor of N/2 leads to 2-parallel architecture and in another design, a folding factor of N/4 leads to design 4-parallel architectures in which four samples are processed in the same clock cycle. Various folding sets lead to a family of FFT architectures. Alternatively, known FFT architectures can also be described by the proposed methodology by selecting the appropriate folding set. To reduce latency and the number of storage elements, folding sets are designed. The prior FFT architectures were derived in an informal way, and their derivations were not explained in a systematic way. This is the effort to simplify the design of FFT architectures for arbitrary level of parallelism in an efficient manner by means of the folding transformation. In this paper, the prior design architectures are explained by constructing the specific folding sets. Then new architecture is derived for radix and levels of parallelism and for either Decimation-In-Time (DIT) or Decimation-In-Frequency (DIF) flow graphs. The new architecture achieves full hardware utilization. It may be noted that all prior parallel FFT architectures did not achieve full hardware utilization. The new real FFT architecture is also presented based on higher radices. In, the parallel-pipelined architectures for the computation of RFFT based on radix-22 and radix-23 algorithms have been proposed. The real FFT architectures are not fully utilized. This drawback is removed by proposed methodology. The novel parallel-pipelined FFT architectures for the real-valued signals with full hardware utilization based on radix-24 algorithm is presented. It combines the advantages of radix-2n algorithms, which requires fewer complex multipliers when compared to radix-2 algorithm, with the reduction of operations using redundancy.

III. PROBLEM OUTLINE

EXISTING SYSTEM

Some folded pipeline architectures have been proposed for the computation of RFFT, where butterfly operations are multiplexed into a small logic unit. The structures could provide adequate throughput for some applications but the storage complexity of those structures continues to be very high. A few in-place architectures have also been proposed for RFFT using specialized packing algorithms. Memory-conflict for read/write operation is found to be the major challenge in the design of algorithms and architectures for in-place computation

1.3. PROPOSED SYSTEM

Fast Fourier transform (FFT) algorithm is frequently encountered in almost every application area of digital signal processing. There are several applications such as speech, audio, image, and video processing, where FFT is very often performed on real-valued signals. Efficient realization of FFT of real-valued signals has received further attention nowadays due to the emergence of biomedical signal processing, and wide applications of real-valued time-series analysis. FFT of real valued signal exhibits conjugate symmetry which renders half the FFT outputs redundant. The DIT-based RFFT butterfly thus involves less propagation delay than that of DIF-based RFFT butterfly although both these butterflies involve the same number of multipliers and adders. Therefore, the choice of DIT algorithm to derive RFFT structure has an advantage over DIF algorithm. In this paper, we present an efficient architecture for the DIT radix-2 RFFT algorithm. The main contributions of this paper, as discussed in the next three sections of the paper, are:

1) Mathematical formulation of the radix-2 DIT RFFT algorithm using real-valued arithmetic.

2) Derivation of a regularized flow graph for folded computation of radix-2 DIT RFFT using a simple control.

3) Formulation of address generation for folded in-place radix-2 DIT RFFT algorithm and derivation of proposed RFFT structure using register-based storage.

IV. METHODOLOGY

The proposed structure for -point in-place DIT RFFT computation is shown in Fig. 3. It consists of one arithmetic unit (AU), one data storage-unit (DSU), one twiddle-factor storage unit (TFSU), and one control-unit (CU). During every cycle, the AU receives a block of 4 samples from the DSU and a pair of twiddle factors from the TFSU. During the first 12 clock cycles of a period of 16 clock cycles, it performs a 4-point BF operation in every clock cycle to produce a 4-point intermediate- output data block which is written back into the DSU. During the last 4 clock cycles of a period of 16 clock cycles the FFT coefficients are delivered out as output of the AU. The internal structure of AU. AU uses two line-changers to steer the BF outputs along the desired path according to the MFG. Both and move the input data from ports and to port and , respectively, when the control signal is “0” otherwise they steer the input data from ports and to port and , respectively. Both and are implemented by a pair of 2-to-1 line MUXes. The MUX array receives the samples from the input port as well as from the AU. It uses signal to select input from the AU during first 12 clock cycles and from the input-port during last 4 clock cycles of each set of 16 clock cycles.

PROPOSED SYSTEM:

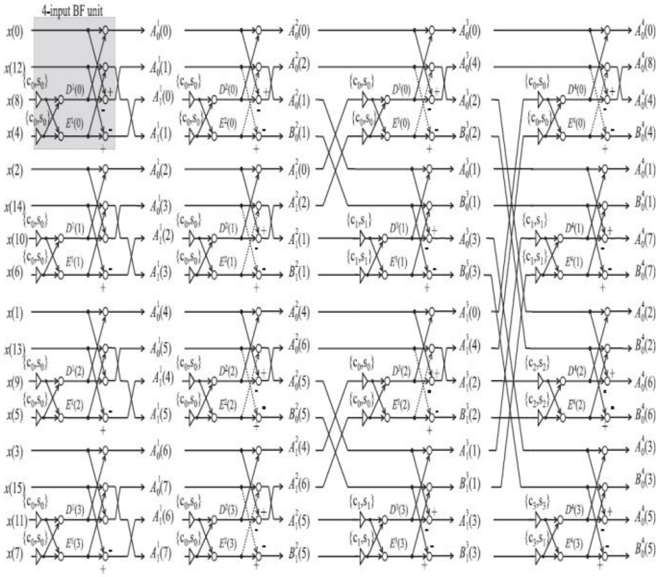


Figure 2. Modified flow graph (MFG) of 16-point DIT RFFT. The dashed line represent no signal path.

From the optimized flow graph of 16-point RFFT (Fig. 1), we find that stage-1, stage-2, stage-3, and stage-4, respectively involve (8, 4, 4, and 4) type-I BFs and (0, 0, 2, and 3) type-II BFs. Similarly, stage-1, stage-2, stage-3, stage-4, and stage-5 of the optimized flow graph of 32-point DIT RFFT, respectively, involve (16, 8, 8, 8, and 8) type-I BFs and (0, 0, 4, 6, and 7) type-II BFs. Interestingly, the numbers of type-I and type-II BFs are nearly equal after stage-2 onwards. Therefore, we use a 4-input hybrid BF comprised of a type-I BF and a type-II BF as the basic building block to derive a regular flow graph. The input pattern of the original flow graph is modified to match with the input pattern of 4-input BF. The BFs of stage-3 and stage-4 are also reordered accordingly. A few redundant arithmetic operations are introduced to transform the original flow graph to a regular form without changing intermediate signal values. The MFG of 16-point DIT RFFT is shown in Fig. 4. The dashed-lines in Fig. 4 represent absence of signal-path. As shown in Fig. 1, each stage of MFG involves four 4-point BFs.

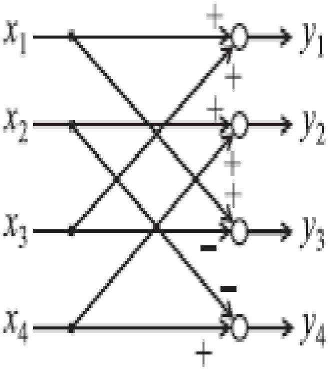


Fig 3: 4-Point butterfly operation

In the proposed structure, the in-place RFFT computation based on the MFG of Fig. 1, is to be performed by one 4-point BF processing unit and a data storage unit. The data storage unit stores the inputs as well as the intermediate results produced after the computation of different stages.

In every clock cycle, 4 input samples/intermediate results are, therefore, read from the data storage unit and accordingly four result values are written into the same storage locations. As shown in Fig. 1, the data-input pattern changes from the BF stage-2 onwards. The change in input pattern results in the increase of complexity of address generation unit which needs to be taken care of to perform appropriate read/write operations with the register banks.

The proposed structure for -point in-place DIT RFFT computation is shown in Fig. 3. It consists of one arithmetic unit (AU), one data storage-unit (DSU), one twiddle-factor storage unit (TFSU), and one control-unit (CU). During every cycle, the AU receives a block of 4 samples from the DSU and a pair of twiddle factors {cl, sl} from the TFSU. During the first 12 clock cycles of a period of 16 clock cycles, it performs a 4-point BF operation in every clock cycle to produce a 4-point intermediate- output data block which is written back into the DSU. During the last 4 clock cycles of a period of 16 clock cycles the FFT coefficients are delivered out as output of the AU.

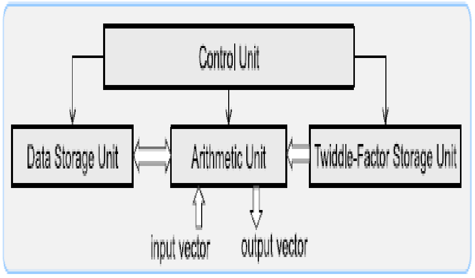


Fig 4: Proposed structure for in-place computation for 16 point DIT RFFT

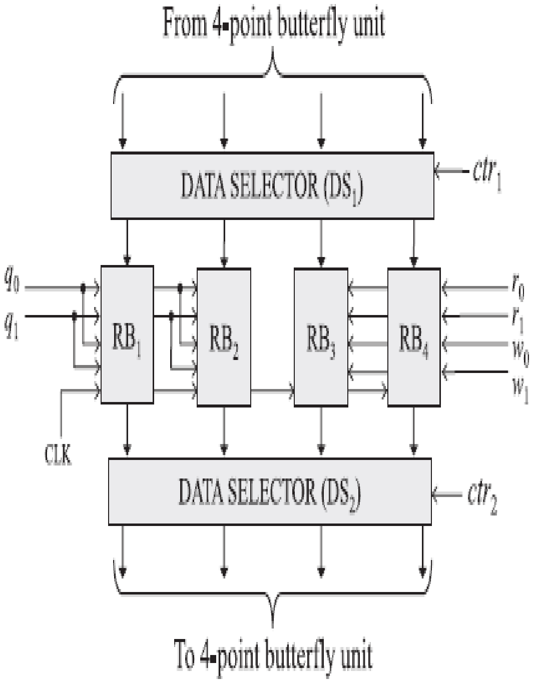


Fig 5: data storage unit of 16 point RFFT

V.RESULTS

Synthesis Result

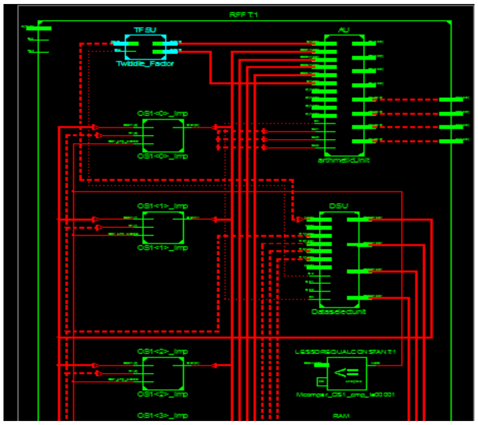
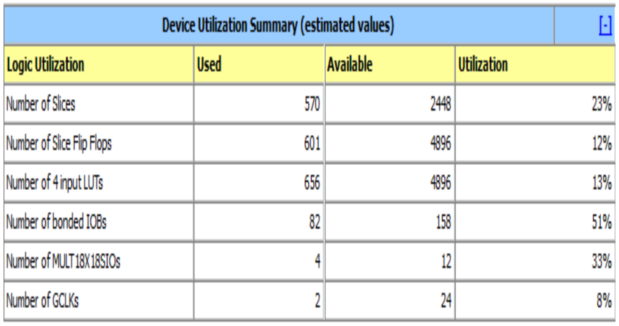


Fig 6: RTL schematic of DIT-RFFT architecture



Device Utilization Summary (estimated values)			[-]
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slices	570	2448	23%
Number of Slice Flip Flops	601	4896	12%
Number of 4 input LUTs	656	4896	13%
Number of bonded IOBs	82	158	51%
Number of MULT18X18SIOs	4	12	33%
Number of GCLKs	2	24	8%

Fig 7: design summary

Synthesis report:

Timing constraint: Default period analysis for Clock 'Clk'

Clock period: 5.532ns (frequency: 180.766MHz)

Total number of paths / destination ports: 1865 / 407

Delay: 5.532ns (Levels of Logic = 2)

Source: RAAM/C_0 (FF)

Destination: RAAM/m_0_15 (FF)

Source Clock: Clk rising

Destination Clock: Clk rising

Data Path: RAAM/C_0 to RAAM/m_0_15

Gate Net

Cell:in->out fanout Delay Delay Logical Name (Net Name)

FDC:C->Q 11 0.591 1.012 RAAM/C_0 (RAAM/C_0)

LUT3:I1->O 8 0.704 0.932 RAAM/m_0_and000011 (RAAM/N11)

LUT4:I0->O 16 0.704 1.034 RAAM/m_8_and00001 (RAAM/m_8_and0000)

FDE:CE 0.555 RAAM/m_8_0

Total 5.532ns (2.554ns logic, 2.978ns route)

(46.2% logic, 53.8% route)

RESULTS:

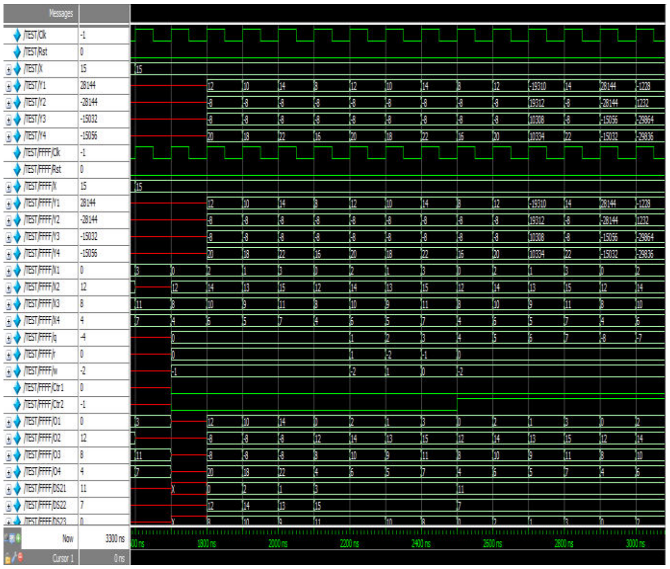
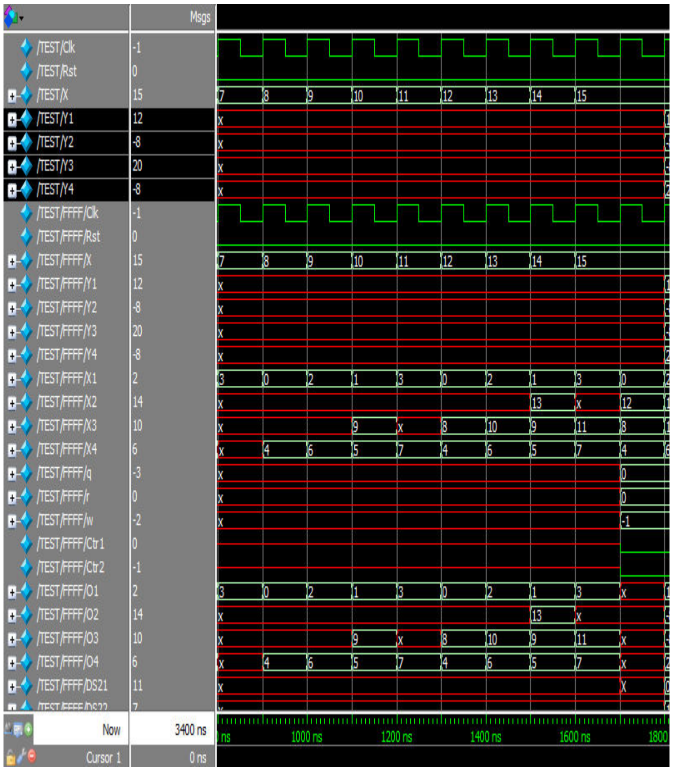


Fig: 8- point DIT RFFT RESULT



	Msgs
/TEST/Clk	-1
/TEST/Rst	0
/TEST/X	15
/TEST/Y1	12
/TEST/Y2	-8
/TEST/Y3	20
/TEST/Y4	-8
/TEST/FFFF/Clk	-1
/TEST/FFFF/Rst	0
/TEST/FFFF/X	15
/TEST/FFFF/Y1	12
/TEST/FFFF/Y2	-8
/TEST/FFFF/Y3	20
/TEST/FFFF/Y4	-8
/TEST/FFFF/X1	2
/TEST/FFFF/X2	14
/TEST/FFFF/X3	10
/TEST/FFFF/X4	6
/TEST/FFFF/q	-3
/TEST/FFFF/r	0
/TEST/FFFF/w	-2
/TEST/FFFF/Ctr1	0
/TEST/FFFF/Ctr2	-1
/TEST/FFFF/O1	2
/TEST/FFFF/O2	14
/TEST/FFFF/O3	10
/TEST/FFFF/O4	6
/TEST/FFFF/DS21	11
/TEST/FFFF/DS22	7
Now	3400 ns
Cursor 1	0 ns

Fig 9: 16-POINT DIT-RFFT RESULT

Here the clock and the reset are the inputs. X is the input N point sequence. Y1, Y2, Y3, Y4 are the outputs. By using the proposed architecture of DIT-RFFT architecture given in the figure 3. We are developed each block using Verilog HDL. After the completion of the each block, we are getting the FFT of the N point sequence as an output. . It consists of one arithmetic unit (AU), one data storage-unit (DSU), one twiddle-factor storage unit (TFSU), and one control-unit (CU). During every cycle, the AU receives a block of 4 samples from the DSU and a pair of twiddle factors {cl, sl} from the TFSU. All of the intermediate wires and the register values are also shown in the results.

VI.CONCLUSION

In this project, we present an area-efficient and energy-efficient architecture for radix-2 DIT real-valued FFT. Besides, we have proposed a register-based storage design which involves significantly less storage area compared with RAM-based storage unit at the cost of marginal increase in latency. The address generation for folded in-place DIT RFFT for register-based storage unit is challenging since both read and write are performed in the same clock cycle at multiple different locations. Therefore, we have regularized the flow graph of RFFT and presented a recursive formulation of the necessary address generation. The proposed structure involves significantly less area-delay product and less energy per sample than the existing folded structures for RFFT implementation.

VII REFERENCES

[1] H. Sorensen, D. Jones, M. Heideman, and C. Burrus, "Real-valued fast fourier transform algorithms," *IEEE Trans.. Acoust., Speech, Signal Process.*, vol. 35, no. 6, pp. 849–863, Jun. 1987.

[2] S.-F. Hsiao and W.-R. Shiue, "Design of low-cost and high-throughput linear arrays for DFT computations: Algorithms, architectures and implementations," *Proc. IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 47, no. 11, pp. 1188–1203, Nov. 2000.

[3] M. Garrido, K. K. Parhi, and J. Grajal, "A pipelined FFT architecture for real-valued signals," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 56, no. 12, pp. 2634–2643, Dec. 2009.

[4] M. Ayinala, M. Brown, and K. K. Parhi, "Pipelined parallel FFT architectures via folding transformation," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 20, no. 6, pp. 1068–1081, Jun. 2012.

[5] A. Wang and A. P. Chandrakasan, "Energy-aware architectures for a real valued FFT implementation," in *Proc. Int. Symp. Low Power Electron. Design*, Aug. 2003, pp. 360–365.

[6] H. Chi and Z. Lai, "A cost-effective memory-based real-valued FFT and Hermitian symmetric IFFT processor for DMT-based wire-line transmission systems," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2005, vol. 6, pp. 6006–6009.

[7] L. G. Johnson, "Conflict free memory addressing for dedicated FFT hardware," *IEEE Trans. Circuits Syst. II., Analog Digit. Signal Process.*, vol. 39, no. 5, pp. 312–316, May 1992.

VIII AUTHORS

D. SRIDHAR Working as Head of the Department of ECE, received the **M.Tech** degree in VLSI System Design from Avanthi Institute of Engineering and Technology, Narsipatnam, B.Tech degree in Electronics and Communication Engineering at Gudlavalleru Engineering College and also Pursuing his **Ph.D** in Low Power VLSI. He has total Teaching Experience (UG and PG) of 11 years. He has guided and co-guided 8 P.G students .His Research areas included VLSI system Design, Digital signal Processing, Embedded Systems

AUTHOR 2

VOOTLA POORNA CHANDRIKA, PG scholar Dept of ECE (VLSID), Sri Sunflower College of Engineering and Technology, B.Tech degree in Electronics and Communication Engineering at sri vasavi institute of engineering & technology (sviet), nandanuru.