An Area-Efficient Split-Radix FFT Processor Using Radix-4 Butterfly Units

Kasula Srujana
Departmentof Electronics and Communication Engineering
Vidya Jyothi Institute of Technology
Hyderabad, India
E-mail: srujanareddy4b4@gmail.com

G.Ravi Kishore
Departmentof Electronics and Communication Engineering
Vidya Jyothi Institute of Technology
Hyderabad, India
E-mail: ravikishore@vjit.ac.in

Abstract: Fast Fourier Transform (FFT) is one of the urgent tasks in advanced flag preparing region. The number of applications for FFTs continues to grow and is incorporated in various regions such as: communications, signal processing, instrumentation, biomedicalengineering, sonic, acoustics, numerical strategies, and applied mechanics. Split radix Fast Fourier Transform (SRFFT) approximates the base number of augmentations by hypothesis among all the FFT calculations; in this way SRFFT is a decent possibility for the execution of a low power FFT processor. In this work, we plan to actualize a novel low power Split-Radix FFT processor utilizing shared-memory design and stretch out this work to a parallel structure dependent on FPGA. We began by planning another radix-2 butterfly unit utilizing clock gating way to deal with square superfluous exchanging action in the multiplier. Contrasted with existing SRFFT processors which depend on the "L" formed butterfly, our execution improves the location process for FFT information. Besides, in light of the fact that the quantity of augmentations required by SRFFT calculation fundamentally diminishes as the FFT measure expands, it is sensible to expect the proposed engineering will spare more power with regards to bigger purposes of FFT.

Keywords: Split-Radix FFT; Low Power; Parallel Architecture, Xilinx tool

I. INTRODUCTION

Split-Radix FFT algorithm was developed by Duhamel and Hollman [1] in 1984. The idea behind their algorithm is the application of a radix-2 index mapping to the even-index terms and a radix-4 mapping to the odd-index terms, which results in an “L” shaped butterfly. Their algorithm requires least number of multiplications and additions among all the FFT algorithms with input length N equals to 2^m (m is any natural number). Since arithmetic operations significantly contribute to overall system power consumption, SRFFT is a good candidate for the implementation of a low power FFT processor. Previous research has focused on implementing SRFFT with “L” shaped butterfly. However, comparing tostandard radix-2 FFT processor, the mixed radix property of “L” shaped butterfly either increases the complexity of control logic or it requires extra hardware to buffer interim data.

We observed that flow graph of split-radix algorithm is the same as radix-2 FFT except for the location and value of twiddle factors. Therefore address generation scheme for conventional radix-2 FFT algorithm could also be applied to SRFFT. In our recent work [3], we have followed ASIC flow and developed a low power SRFFT processor using shared memory architecture. We adopted address generation scheme in [2].

II. LITERATURE REVIEW

Since the early paper by Cooley and Tukey[1] a lot of work has been done on the FFT algorithm, and this has resulted in classes of algorithms such as radix-2^m algorithms, the Winograd algorithm (WFTA), and prime factor algorithms (PFA). Among these, the radix-2 and radix-4 algorithms have been used most for practical applications. This is due to their simple structure, with a constant geometry (butterfly type) and the possibility of performing them 'in place', even if they are more costly in terms of number of multiplications than WFTA and PFA. Recently, some radix-2 algorithms have been proposed[2-4] which preserve more or less the advantages mentioned above, but require fewer multiplications than the usual radix-2^m algorithms. Unfortunately, these methods need 20 to30% more additions, and seem to be numerically ill conditioned.

We present here an algorithm which includes the advantages of these radix-2^m methods:
(a) The lowest number of multiplications, together with real factor algorithms
{b) The lowest number of additions among the 2^m algorithms
(c) The same regularity as radix-4 algorithms when implemented through an autogen method[9]
(d) As flexible as radix-2 algorithms (useful for all N = 2^n)

(e) Numerically as well conditioned as the radix-4 algorithm.

"Split-radix" algorithm: This algorithm comes from a very simple observation: a radix-2 algorithm diagram can be transformed quite straightforwardly into a radix-4 algorithm diagram only by changing the exponents of the twiddle factors. When doing so, it is quite clear that, at each stage of the algorithm, a radix-4 is best for the odd terms of the DFT, and radix-2 for the even terms of the DFT. So one might guess that restricting the transformation above locally to the lower part of the diagram only might improve the algorithm.

III. Design Method And Structure For Radix-4

A.Unit Design

The structure of our SRFFT processor is shown in figure 1. It consists of two RAM banks, two ROM banks, and one butterfly unit and address generation circuits. FFT data are stored in RAM banks and Xiao's [2] algorithm could be used to generate their corresponding addresses. However, for address generation of twiddle factors, which are stored in the ROMs, we cannot simply adopts Xiao's algorithm, since conventional radix-2 FFT only has twiddle factor at the lower leg of butterfly while SRFFT has twiddle factors at both upper leg and lower leg. In our implementation, we use a decoder (look-up table) which has the combination of pass counter P and butterfly counters B as input and generates the address of twiddle factors as well as control signal for the clock gating registers.

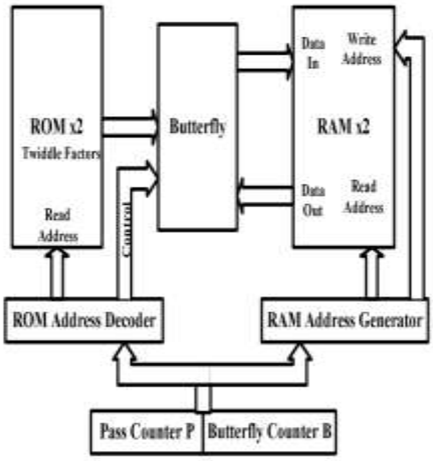


Fig1: FFT block diagram

In previous publication [3], proposed a low power radix-2 butterfly unit using tristate buffer. Since there are no internal tristate buffers in modern FPGA, we modify our butterfly unit and use clock gating approach to block unnecessary switching activity in multiplier. The structure of our butterfly unit is shown in figure 2.

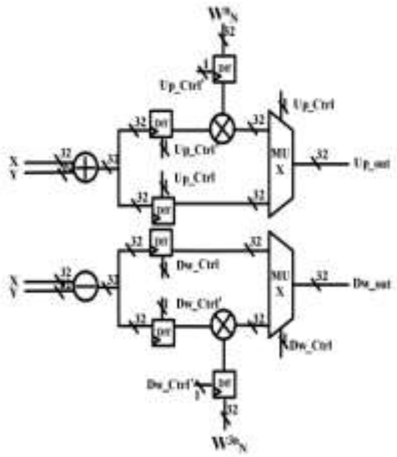


Fig.2. Low power butterfly structure

When we need to multiply twiddle factors, data go through the multiplier path. Otherwise, data just go to output.

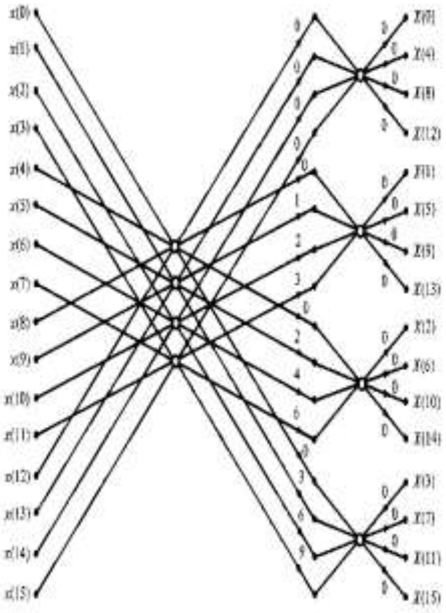
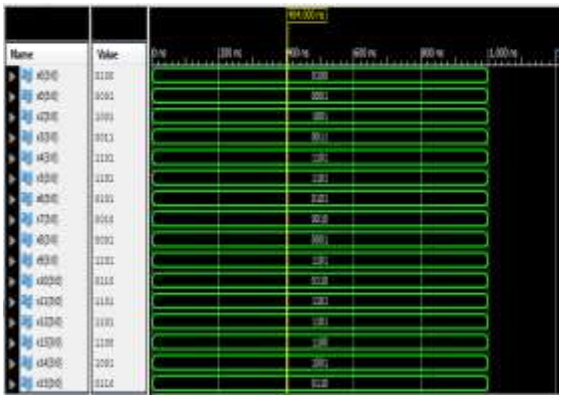


Fig3: Signal flow graph for 16 point radix4 DIF FFT

IV. Preliminary Results



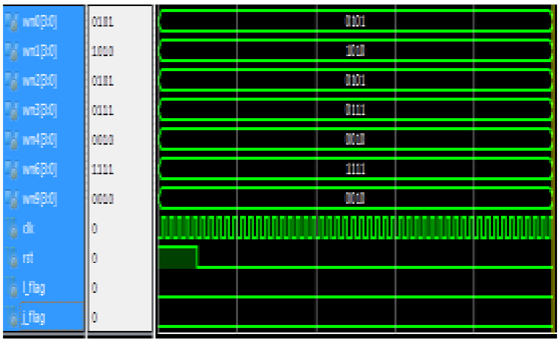


Fig4: Input for 16 Point Split Radix4 DIF FFT

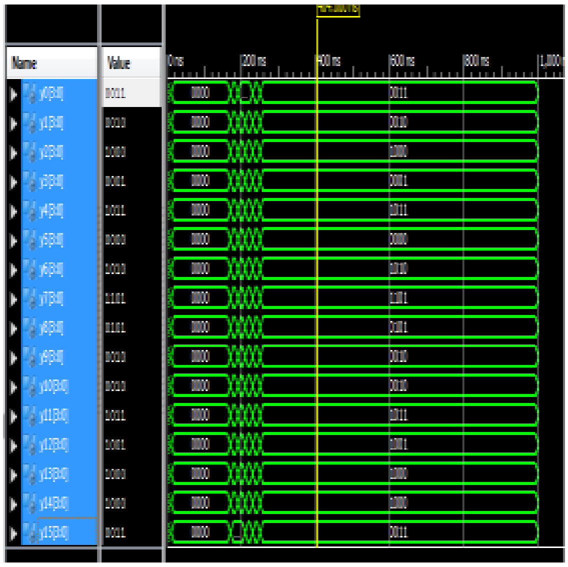


Fig5: Output for 16 Point Split Radix 4DIF FFT

Parameters		Split radix2	Split radix4
Synthesis report	Number of slices	317	254
	Number of 4 input LUTs	495	400
	Number of Bonded IOBs	146	160
	Max.delay (ns)	6.061	5.563

Table1: Analysis of Area

V. Conclusion and Future Work

In this paper we described a16 point SRFFT processor structure that is suitable for implementation on FPGA. A tradeoff has been made between hardware resources and dynamic power. Preliminary simulation results demonstrate the benefits of our approach. So far we only compare simulation results with one reference design. In future we will compare our approach with other existing works. Another aspect of ongoing work would involve developing efficient twiddle factor addressing scheme. In our implementation, we have used the look-up table approach to generate addresses of twiddle factors. If a well-defined algorithm could be found, our design could be configured to any FFT size without the need to reprogram the address decoding part. .

References

[1] P. Duharnel and H. Hollmann, "Split radix FFT algorithm," IET Electronic Letters, vol. 20, pp. 14-16, Jan. 1984.

[2] X. Xiao, E. Oruklu and J. Saniie, "An Efficient FFT Engine with Reduced Addressing Logic," IEEE Trans. on Circuits and Systems II: Express Briefs, vol. 55, pp. 1149 - 1153, Nov. 2008.

[3] Z. Qian and M. Margala. "A Novel Low-Power and In-Place Split-Radix FFT Processor," in ACM Proc. of the 24th Great Lakes Symp. on VLSI (GLVLSI'14), Houston, TX, USA, 2014, pp.81-82.

[4] M. A. Richards, "On hardware implementation of the split-radix FFT," IEEE Trans. Acoust., Speech Signal Process., vol. 36, no. 10, pp. 1575–1581, Oct. 1988.

[5] J. Chen, J. Hu, S. Lee, and G. E. Sobelman, "Hardware efficient mixed radix-25/16/9 FFT for LTE systems," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol.

23, no. 2, pp. 221–229, Feb. 2015.

[6] 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.

[7] D. Cohen, "Simplified control of FFT hardware," IEEE Trans. Acoust., Speech, Signal Process., vol. 24, no. 6, pp. 577–579, Dec. 1976.

[8] H. V. Sorensen, M. T. Heideman, and C. S. Burrus, "On computing the split-radix FFT," IEEE Trans. Acoust., Speech Signal Process., vol. 34, no. 1, pp. 152–156, Feb. 1986.

[9] X. Xiao, E. Oruklu, and J. Saniie, "An efficient FFT engine with reduced addressing logic," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 55, no. 11, pp. 1149–1153, Nov. 2008.

[10] Z. Qian, N. Nasiri, O. Segal, and M. Margala, "FPGA implementation of low-power split-radix FFT processors," in Proc. 24th Int. Conf. Field Program. Logic Appl., Munich, Germany, Sep. 2014, pp. 1–2.

[11] A. N. Skodras and A. G. Constantinides, "Efficient computation of the split-radix FFT," IEE Proc. F-Radar Signal Process., vol. 139, no. 1, pp. 56–60, Feb. 1992.

AUTHOR'S BIOGRAPHIES

KasulaSrujanareceived B.Tech. degree in E&C engineering from Malla Reddy Institute of Technology and Science, Secunderabad, India in 2015, and currently pursuing M.Tech. Degree in VLSI system design from Vidya Jyothi Institute of Technology,Hyderabad ,India .

G.RaviKishore Associate Professor in Vidya Jyothi Institute of Technology (VJIT), Hyderabad. He received B.Tech. Degree in E&C engineering from JNTUH, Hyderabad and M.Tech. degree in VLSI design from Bharat University,chennai, India.