Low Power ROM less FFT Processor

Deepa Susan George & V.Sarada

SRM University ,Chennai
E-mail : deepageorge1989@gmail.com

***Abstract* – In this paper the efficient implementation of a pipeline FFT processor is presented. The design adopts a single path delay feedback as the proposed architecture. The single–path delay feedback (SDF) pipeline requires less memory space and its multiplication utilization is less. Such implementations are advantageous to low power design, especially in portable DSP devices. Read only memories are used to store twiddle factors. Inorder to achieve a ROM less FFT processor, the proposed architecture uses a reconfigurable complex multiplier and bit parallel multiplier, and using the symmetry property of twiddle factors, low power is achieved.**

I. INTRODUCTION

The Fast Fourier Transform (FFT) is a critical block and widely used in digital signal processing algorithm. With the advent of semiconductor processing technology in VLSI system, it has enabled the performance of FFT design to increase steadily. Discrete Fourier transform (DFT) is a very important technique in modern digital signal processing (DSP) and telecommunications. However, DFT is computational intensive and has a time complexity of $O(N^2)$. The Fast Fourier transform (FFT) was proposed by Cooley and Tukey to efficiently reduce the time complexity to $O(N\log 2N)$, where N denotes the FFT size. Fast Fourier Transform is a high efficient algorithm to compute the DFT. The basic idea of this approach is to decompose the N-point DFT into successively smaller DFT. This approach leads to highly efficient computation of FFT algorithm. The FFT, facilitates the efficient transformation between the time domain and the frequency domain for a sampled signal.[1]

Various FFT processors can be used for hardware implementation. These implementations can be mainly classified into memory-based and pipeline architecture styles. Memory-based architecture is widely adopted to design an FFT processor, also known as the single processing element(PE) approach. This design style is usually composed of a main PE and several memory units, thus the hardware cost and the power consumption are both lower than the other architecture style. But the disadvantage is that it has long latency, long throughput and it cannot be parallelized. In order to overcome the disadvantages of the memory based architecture style, we go for pipeline based architecture style.[2]

For a pipelined FFT processor, each stage has its own set of processing elements. All the stages are computed as soon as data are available. pipelined FFT processors have features like simplicity, modularity and high throughput. These features are important for real-time, in-place applications where the input data often arrive in a natural sequential order. We therefore select the pipeline architecture for our FFT processor implementation.

In the traditional hardware implementation of FFT processors, ROMs are used to store the wanted twiddle factors, and the wordlength complex multipliers to perform FFT computing. This increases the hardware cost so bit-parallel multipliers are used.The architecture makes also the use of symmetric property of twiddle factors. [3]

II. PROPOSED ARCHITECTURE

The proposed architecture shown in figure 1 is composed of three different types of processing elements (PEs), a complex constant multiplier and delay-line (DL) buffers . The proposed architecture uses single path delay feedback. A reconfigurable complex constant multiplier is used to eliminate the twiddle-factor ROM. Thus the new multiplication structure becomes the important component in reducing the area and hardware cost.

Based on the radix-2 FFT algorithm, the three types of processing elements are used in the design. The functions of the three PE types correspond to each of the butterfly stages .A complex multiplier is used for the computation of multiplication by a twiddle factor. PE3 stage is used to implement a simple radix-2 butterfly structure only, and serves as the submodules of the PE2 and PE1 stages. In the figure, *Iin* and *Iout* are the real parts of the input and output data, respectively. *Qin* and *Qout* denote the image parts of the input and output

data, respectively. Similarly, *DL_Iin* and *DL_Iout* stand for the real parts of input and output of the DL buffers,and *DL_Qin* and *DL_Qout* are for the image parts, respectively.The working of processing element 3 (PE3) is as follows. When So =0, DL_Iin = Iin ,Iout=DL_Iout ,When So=1 , DL_Iin = DL_Iout + (-Iin), Iout=Iin+(-DL_Iout).In PE2 stage,we need to perform the multiplication by -1.

The traditional hardware implementation of FFT processors usually employs a ROM to look up the wanted twiddle factors, and then word length complex multipliers to perform FFT computing. However, this introduces more hardware cost. So the word length multiplier is replaced by a bit parallel multiplier. But, there will be poor precision due to truncation error .So to reduce the truncation, canonic signed digit representation is used.[4][5]

The canonical signed digit (CSD) representation is one of the existing signed digit (SD) representations with unique features which make it useful in certain DSP applications focusing on lowpower,efficient-area and high-speed arithmetic . The CSD code is a ternary number system with the digit set $\{1^{-}\ 0\ 1\}$, where 1^{-} stands for 1. Given a constant, the corresponding CSD representation is unique and has two main properties: (1) the number of nonzero digits is minimal, and (2) no two consecutive digits are both non zero. The first property leads to the reduction in the number of additions in arithmetic operations. This encoding is performed from LSB to MSB.[6]

All the co-efficient values listed in figure has been realized using canonic signed digit representation .Instead of accessing those twiddle factor values from ROM,they are obtained directly.An example which shows the CSD realization of 0.707092 is shown in figure 7.

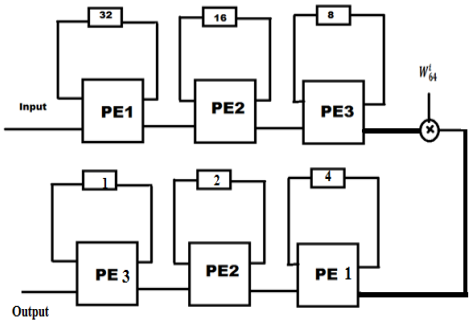


Figure 1: Proposed 64 point FFT architecture

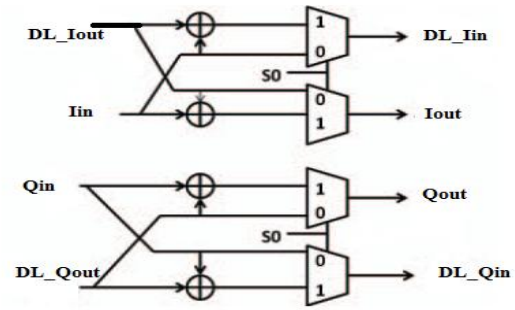


Figure 2 : PE3 Architecture

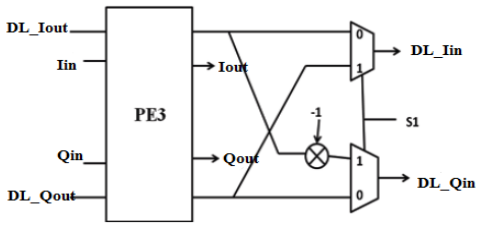


Figure 4: PE2 Architecture

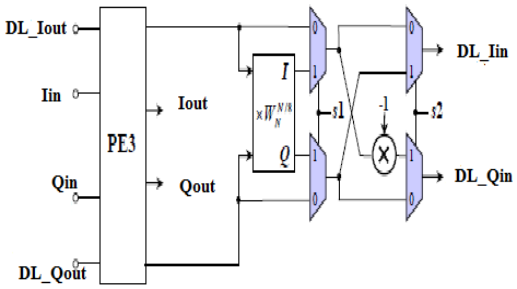


Figure 5: PE1 Architecture

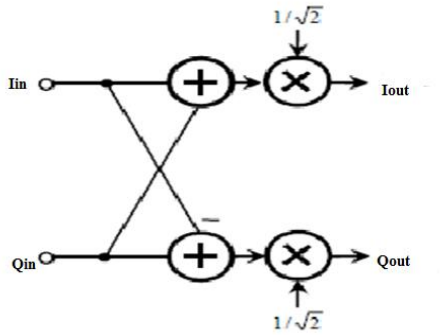


Figure 6:Multiplication by $W_N^{N/8}$

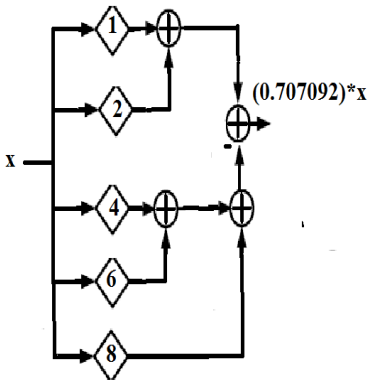


Figure7: CSD realization of multiplication of x and 0.707092

[(x>>1 + x>>2) – (x>>4+ x>>6 + x>>8)]

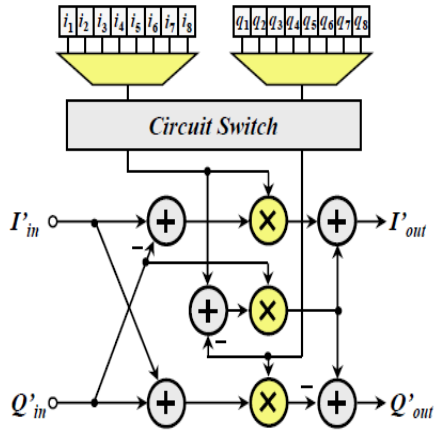


Figure 8: Complex Multiplier

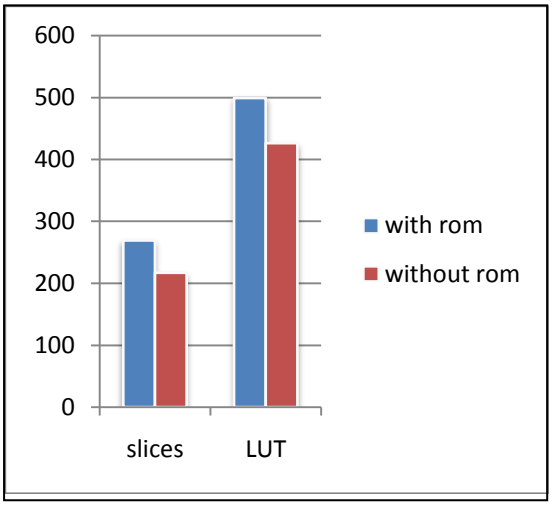
Co-efficient	Value	Co-efficient	Value
i1	0.7071	q1	0.7071
i2	0.7730	q2	0.6343
i3	0.8314	q3	0.5555
i4	0.8819	q4	0.4713
i5	0.9238	q5	0.3826
i6	0.9569	q6	0.2902
i7	0.9807	q7	0.1950
18	0.9951	q8	0.0980

Table 1: Coefficient values used in figure 8

III. CONCLUSION

In the proposed architecture, the twiddle factor values are realized using canonic signed digit representation instead of accessing the values from ROM. The simulation results shows the reduction in the number of slices and LUTs used when the ROM less architecture is used. Therefore ,there will be reduction in area and hence power and hence can be used for OFDM applications.

IV. SIMULATIONRESULTS



	Slices	LUT
With ROM	269	499
Without ROM	217	427

Table 2: No: of slices and LUTs used in ROM and ROM less architecture

V. REFERENCES

[1] J. W. Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex Fourier series," *Math. Comput.*, vol. 19, pp. 297- 301, Apr. 1965

[2] Kosher Maharatna, Eckhard Grass, and Ulrich Jagdhold, "A 64-Point fourier transform chip for high-speed wireless LAN application using OFDM," *IEEE Journal of Solid-State Circuits*, vol. 39, no. 3, pp. 484-493, Mar. 2004.

[3] Y.T. Lin, P.Y. Tsai and T.D. Chiueh, "Low-power variable-length fast Fourier transform processor," *IEEEProc. Comput. Digit. Tech.*, vol. 152,no. 4, pp. 499-506, July 2005

[4] Jia Lihong, Gao Yonghong, Isoaho Jouni, and Tenhunen Hannu, "A new VLSI-oriented FFT algorithm and implementation," in *Proc. IEEE International on ASIC Conference*, Sept. 1998, pp. 337–341. Castros s/n, Universidad de Cantabria, 3900

[5] Chin-Teng Lin, Yuan-Chu Yu, and Lan-Da Van, "A low-power 64-point FFT-IFFT design for IEEE 802.11a WLAN application," in *Proc. IEEEInt. Symp. Circuits Syst*. (ISCAS), May 2006, pp. 4523-4526.

[6] Efficient canonic signed digit recoding ,Gustavo A. Ruiz n, Mercedes Granda Departamento de Electro´nica and Computadores, Facultad de Ciencias, Avda. de Los 5 Santander, Spain

◈◈◈