Design Of A Parallel Pipelined FFT Architecture With Reduced Number Of Delays

Kiranraj A. Tank
Department of Electronics
Y.C.C.E, Nagpur, Maharashtra, India

Pradnya P. Zode
Department of Electronics
Y.C.C.E, Nagpur, Maharashtra, India

Abstract- This paper presents a novel approach to design a four and eight parallel pipelined fast Fourier transform (FFT) architecture based on canonic signed digit multiplier. This approach is based on use of decimation in time algorithm which reduces the number of delay elements up to some extent compared to decimation in frequency based design. The number of delay elements required for an N point FFT architecture is N-4 which is comparable to that of delay feedback schemes. The number of complex adders required is approximately 50% less than the other feedback designs. The proposed architecture can be extended to any radix 2n based FFT algorithm. This proposed architecture is based on feed forward designs and can be pipelined up to more stages to increase the throughput.

Keywords – Canonic signed digit (CSD), fast Fourier transform (FFT), folding, decimation in time (DIT).

I. INTRODUCTION

The most widely used algorithm in digital signal processing is fast Fourier transform (FFT). Communication biomedical and radar communication are the fields which mostly uses FFT. Multicarrier modulation system such as ultra wide band systems (UWB), digital video broadcasting (DVB), orthogonal frequency division multiplexing (OFDM) FFT and IFFT are core functions in such systems. Biomedical applications such as electrocardiograph (ECG) electroencephalogram (EEG) uses FFT in frequency domain.

Many parallel pipelined architecture have been proposed based on various algorithms. The architecture of four parallel structure was designed for OFDM system to increase the throughput of the system which uses 128 point FFT. These architecture uses radix 2^3 and radix 2^4 while some architecture use mixed radix-2 and radix-8 algorithms. These architectures have their own pros and cons. These architectures make use of delay feedback design which in turn gives less number of delay elements but the number of data path is equal to the number of butterflies used. In this project we are designing the four and eight parallel architecture which gives 100% hardware utilization along with less number of delay elements. Further we present a 128 FFT architecture which uses radix 24 algorithm.

Canonic signed digit multipliers is a very efficient method for constant fixed point multiplication by utilization of redundancy of signed digit code. Csd is the radix-2 signed digit representation for coefficient of digital set {1,0,-1}. Thus csd representation permits subtraction as well as addition of shifted data of partial products which is generated by multiplication of two numbers

II. PROPOSED ALGORITHM

A. Canonic Signed Digit Representation–

Canonic signed digit is based on a ternary number system (1, 0, -1). It is a unique representation of binary number with minimum number of 1 and -1. It allows minimum number of addition and subtraction to produce the product. In this we have to convert the multiplier coefficient from 2's compliment to its equivalent csd representation. Due to the ternary nature each non zero digit in a csd representation requires two bits one for magnitude and one for sign hence there is a need for extra space.

Figure 1 shows the flow chart of 2's compliment to csd number conversion. From the definition it can be mathematically represented as

$$x = \sum_{i=0}^{n-1} x_i 2^i \quad \text{where } x_i \in \text{-1,0,1}$$

From the mathematical equation it can be stated that the two consecutive digits are non zero. For example $(15/32)_{10}$ can be represented as $(0.100\bar{1})_{csd}$

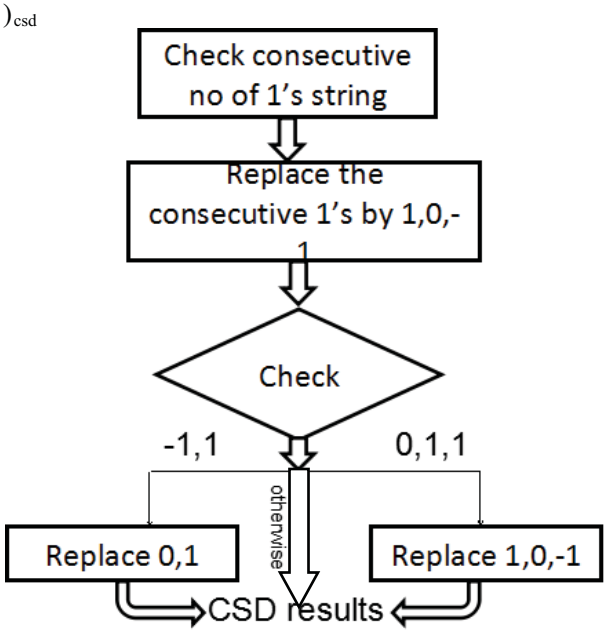

Figure 1. Flow chart diagram of 2's compliment to CSD conversion

B. CSD Multiplication –

Figure 2 represents the flow chart diagram of CSD multiplication. $N \times N$ partial products are generated first and then they are added to get the final result. At last we convert the CSD result into binary representation.

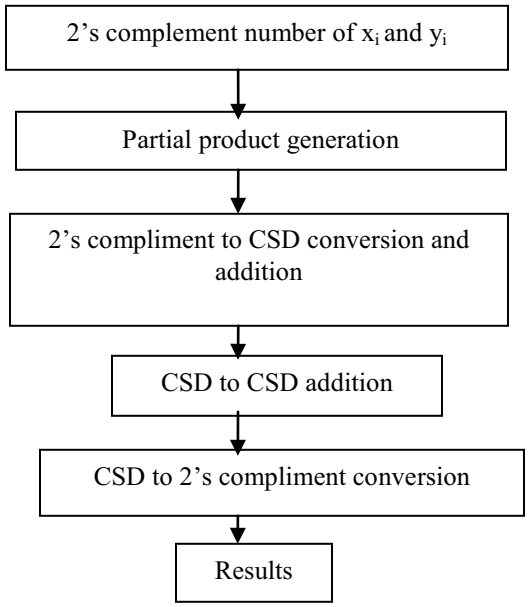

Figure 2. Flow Chart Diagram Of CSD Multiplication

III. PROPOSED FFT ARCHITECTURE

The approach for designing four parallel and eight parallel is discussed in this section. These architecture are applicable to any radix-2^n algorithm. Discussed architectures are designed using decimation in time so as to reduce the number of delays in pipelined architecture.

A. *4-Parallel Design*

The four parallel design can be explained using either ad hoc approach or folding approach. Further it can be used to design the eight parallel architecture. An example of 16 point 4 parallel architecture is shown in figure 3 which will be used to explain 4 parallel design.

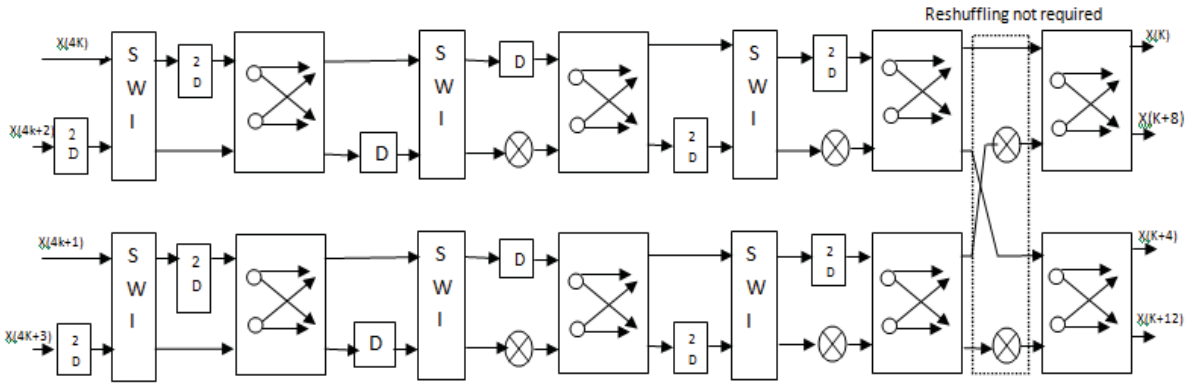

Figure 3. Proposed 4-Parallel Architecture For 16 Point Radix 2 FFT

B. *Ad Hoc Design*

Figure 4 represents the flow graph of radix-2 16 point DIT FFT algorithm where the top four butterflies A0 to A3 represents the first stage which processes the even samples and the lower four butterflies A`0 to A`3 processes the odd samples. Similarly B0 to B`3 , C0 to C`3, D0 to D`3 represents the second third and fourth stage of FFT. The output of two N/2 FFT's can be combined in the final butterfly stage. In a two parallel architecture it processes two consecutive even samples (4K , 4K+2) and two odd samples (4K+1), (4K+3). At last the reordering of the samples are not required if the input buffer is available to reorder the data before FFT processor. The output of the N/2 FFT's arrive at the same time hence final stage does not require reshuffling circuit this will reduce the required number of delay elements to implement the pipelined structure.

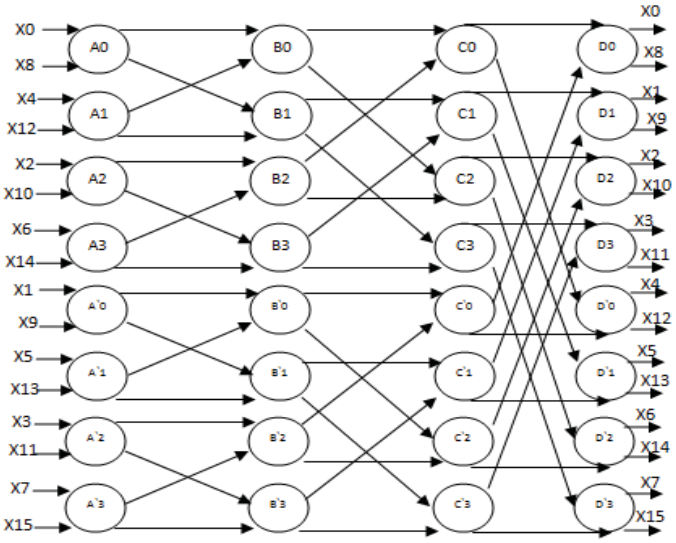

Figure 4. Flow Graph Of Radix-2 16 Point DIT FFT

C. *Folding Approach*

A={A0,A1,A2,A3} A`={A`0,A`1,A`2,A`3}
B={B3,B0,B1,B2} B`={B`3,B`0,B`1,B`2}
C={C0,C1,C2,C3} C`={C`0,C`1,C`2,C`3}
D={D0,D1,D2,D3} D={D`0,D`1,D2,D`3}
The register minimization technique and the forward and backward register allocation scheme are applied to derive the folded architecture. The final architecture will be the same as figure 3.

D. *EIGHT PARALLEL ARCHITECTURE*

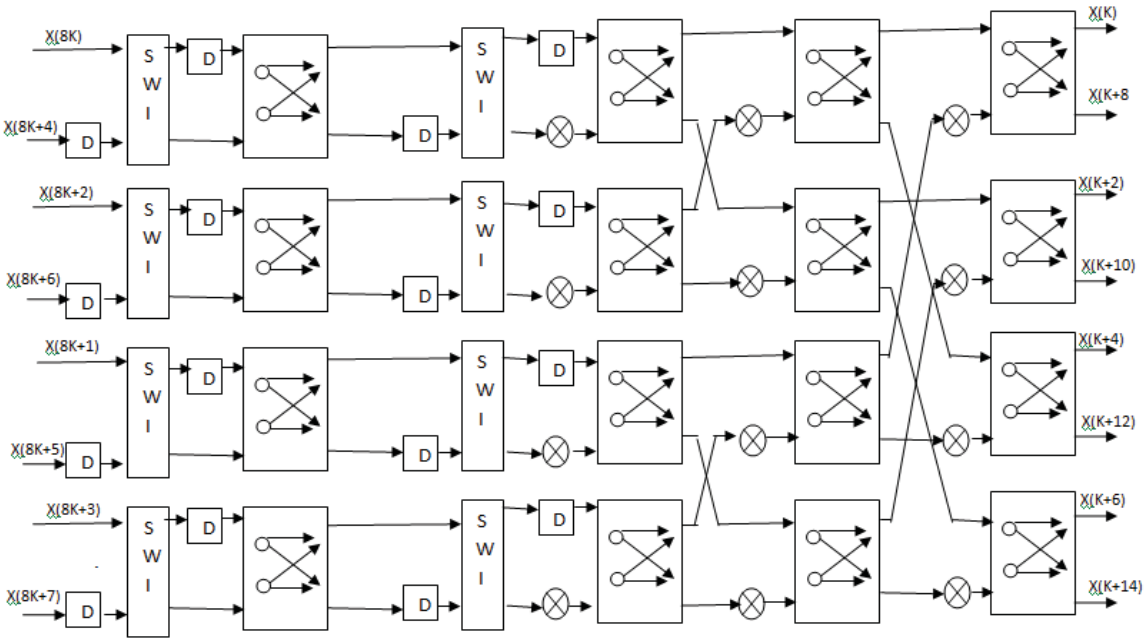

Figure 5. Eight Parallel Architecture For 16 Point Radix-2 FFT

In a same way the eight parallel architecture can be derived based on DIT flow graph. Now the even and odd input samples are 8k, 8k+4, 8k+2, 8k+6, 8k+1, 8k+5, 8k+3, 8k+7. Figure 5 shows the eight parallel architecture for a 16 point radix-2 FFT.

IV.CONCLUSION

The proposed 4 parallel and 8 parallel architecture requires 3N/2 – 4 and 3N/2 – 8 delay elements. The prior 4 parallel architecture based on DIF algorithm requires 2N – 4 delay elements. It will remain same for all radix-2^n algorithms**.** In general the delay feedback architectures requires N – 4 delay elements compared to 3N/2 – 4 in proposed architecture as there is a difference N/2 delay elements these are required to reorder the samples according to the flow graph.

REFERENCES

[1] Garrido, M.; Parhi, K.K.; Grajal, J., "A Pipelined FFT Architecture for Real-Valued Signals," Circuits and Systems I: Regular Papers, IEEE Transactions on , vol.56, no.12, pp.2634,2643, Dec. 2009

[2] Shousheng He; Torkelson, M., "Design and implementation of a 1024-point pipeline FFT processor," Custom Integrated Circuits Conference, 1998. Proceedings of the IEEE 1998 , vol., no., pp.131,134, 11-14 May 1998

[3] Sorensen, H.V.; Jones, D.L.; Heideman, M.; Burrus, C.S., "Real-valued fast Fourier transform algorithms," Acoustics, Speech and Signal Processing, IEEE Transactions on , vol.35, no.6, pp.849,863, Jun 1987

[4] Ziegler, H., "A fast Fourier transform algorithm for symmetric real-valued series," *Audio and Electroacoustics, IEEE Transactionson* , vol.20, no.5, pp.353,356, Dec 1972 doi: 101109/TAU19721162406

[5] B. R. Sekhar and K. M. M. Prabhu "Radix-2 decimation-in-frequency algorithm for the computation of the real-valued FFT", IEEE Trans. Signal Process., vol. 47, no. 4, pp.1181 -1184 1999.