

**FFT/IFFT Mixed Radix Processor  
For Wireless Applications****R.Radhika**

Department of Electronics and Communication Engineering Saveetha Engineering  
College, Thandalam, Chennai, India

**R.Ramesh**

Department of Electronics and Communication Engineering Saveetha Engineering  
College, Thandalam, Chennai, India

***Abstract:***

*In this project a 128-point FFT/IFFT processor for ultra wideband (UWB) systems, FFT algorithm is used to reduce the computational complexity of DFT and to eliminate the redundant complex multiplication. The proposed pipelined FFT architecture called mixed radix multipath delay feedback (MRMDF) can provide a higher throughput rate achieved by using butterfly units. In addition the number of complex multiplications reduced effectively by using a higher radix algorithm. The 128-point mixed radix FFT/IFFT algorithm is implemented to power consumption and hardware cost can also be saved. Here memory based FFT architecture will decompose a larger FFT computation into several cascaded smaller FFTs and utilize a single FFT core to reduce the hardware cost.*

***Key words:*** Fast Fourier Transforms (FFT), ultra wideband (UWB), Multiplier.

### **1.Introduction**

The Fast Fourier Transform (FFT) is a fast way to calculate the Discrete Fourier Transform [DFT]. FFT can be decomposed using DFT of even and odd points, which is called a Decimation-In-Time(DIT)FFT or it can be decomposed using a first-half/second-half approach which is called Decimation-In-Frequency .The FFT decomposes the set of data to be transformed into a series of smaller data sets to be transformed. At each stage of processing, the results of previous stage are combined finally it calculates the DFT of each small data set. DFT of the first  $N/2$  points and combine in a special way with the DFT of the second  $N/2$  points to produce a single  $N$ -point DFT. Each of these  $N/2$  point DFT can be calculated using smaller DFT in the same way. These are combined to form  $n/4$  4-point DFTs. The next stage produces  $N/8$  8-point DFTs until a single  $N$ -point DFT is produced. The DFT takes  $N^2$  operations for  $N$ -points. Since at any stage the computation required to combine smaller DFTs into larger DFTs is proportional to  $N \cdot \log_2(N)$ . Therefore the ratio between DFT computation and an FFT computation for the same  $N$  is proportional to  $N/\log_2(n)$ . Ultra Wide Band(UWB) is a wireless technology for transmitting large amount of digital data over a wide spectrum of frequency bands with very low power for short distance.

### **2.Butterflyarchitecture**

The primary computational unit in the FFT is the butterfly, which perform both complex multiplications by the twiddle factors with pieces of data in the sequence and the addition/subtraction of these products. A single radix-4 butterfly performs the computational equivalent of four radix- 4 butterfly can also be modified to function as a radix-2 butterfly.

### **3.Bit Reversal**

Bit reversal is reversing the bits in a binary word from left to right. Therefore the MSBs become LSBs and LSBs become MSBs.

### **4.Twiddle Factors**

Twiddle factors are the coefficients used to combine results from a previous stage to form inputs to the next stage. The twiddle factors needed in the computation of the FFT are represented by integer power of the following equation.

$$W_{N=e^{-i2\pi/N}}$$

Where N is the size of the FFT. This can related as,

$$W_{N=e^{i\theta_N}}$$

Where  $\theta_N$  is equal to  $2\pi/N$ .

## V. MIXED RADIX ALGORITHMS

Given a sequencex (n), an N -point discrete Fourier transform (DFT) is defined as

$$X(K) = \sum_{n=0}^{N-1} x(n)W_N^{Kn} \quad K=0, 1 \dots 127$$

Where x (n) andX (K) are complex numbers. The twiddle factor is

$$W_N^{nk} = e^{-j(2\pi nk/N)}$$

The computational complexity is  $O(N^2)$ through directly performing the required computation. By using the FFT algorithm, the computational complexity can be reduced to  $O(\log_r^N)$ , where r means the radix-r FFT. The radix-r FFT algorithm can be easily derived from the DFT by decomposing the N -point DFT into a set of recursively related N-point FFT transform, if is power of r.That is, different stages in the FFT computation have different radices. For instance, a 128-point long FFT can be computed in two stages using one stage with radix-8 PEs, followed by a stage of radix-2 PEs. This adds a bit of complexity to the algorithm compared to radix-r, but in return it gives more options in choosing the transform length. The Mixed-Radix FFT algorithm is based on sub-transform modules with highly optimized small length FFT which are combined to create large FFT. However, this algorithm does not offer the simple bit reversing for ordering the output sequences.

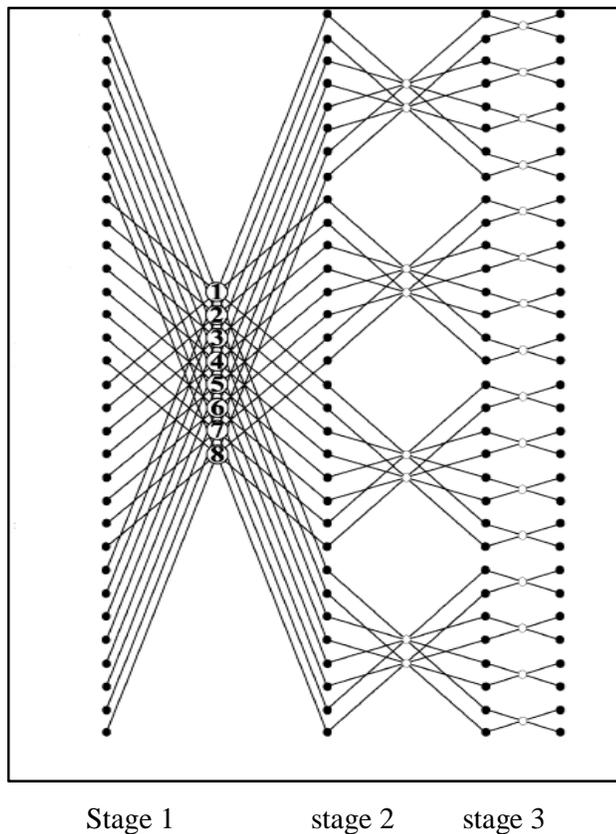


Figure 1: 32-point mixed-radix FFT algorithm

In Stage 1 and Stage 2, the radix-4 algorithm is used; in Stage 3, the radix-2 algorithm is used.

Total number of complex multiplier required for normal DFT operation is  $N^2$ . Same DFT can be done by FFT radix-2 algorithm  $N/2 \log_2 N$ . Again this multiplier can be reduced by using mixed radix algorithm.

$$N=32$$

$$\begin{aligned} \text{Complex multiplier for normal DFT} &= N^2 \\ &= 32^2 \\ &= 1024 \end{aligned}$$

$$\begin{aligned} \text{Complex multiplier for mixed radix-2} &= N/2 \log_2 N \\ &= 80 \end{aligned}$$

Radix-4 butterfly output equations

$$X = A - BWb + CWc - DWd$$

$$Y = A + iBWb - CWc - iDWd$$

When the inputs B and D in equations are tied to zero. This occurs throughout the entire radix-2 stage. The complex multipliers within the butterfly produce the products BWb, CWc, and DWd. To achieve the highest level of accuracy, the maximum numbers of bits are preserved throughout the butterfly stage. After going through the multipliers, the real and imaginary parts of the products are each 32 bits long. Adding the real components of A, BWb, CWc, and DWd together yields a sum that is 34 bits long. Rounding this data back to the 16 most significant bits involves adding 1/2 of a least significant bit of 16 bit sum and truncating the rest of the bits. The rounding performed after the addition of the real portions of the complex products are added together. Rounding is performed identically after the addition of the imaginary portions of the complex products are summed. In general, higher-radix FFT algorithm has less number of complex multiplications compared with radix-2 FFT algorithm which is the simplest form in all FFT algorithms. In an example, for the 128-point FFT, the number of nontrivial complex multiplications of the radix-8 FFT algorithm is 152, which is only 59.3% of that of the radix-2 FFT algorithm. Thus, in order to save the number of complex multiplications, we choose the radix-8 FFT algorithm. Because the 128-point FFT is not a power of 8, the mixed-radix FFT algorithm, including radix-2 and radix-8 FFT algorithms. This will be derived in below.

$$N=128,$$

$$n=64n_1+n_2 \quad n_1=0,1$$

$$n_2=0,1 \dots 63$$

$$k=k_1+2k_2$$

$$k_1=0,1$$

$$k_2=0,1 \dots 63$$

$$x(k_1+2k_2) = \sum_{n_2=0}^{63} \sum_{n_1=0}^1 x(64n_1+n_2)$$

$$W_{16}^{(2k_2+k_1)x(64n_1+n_2)}$$

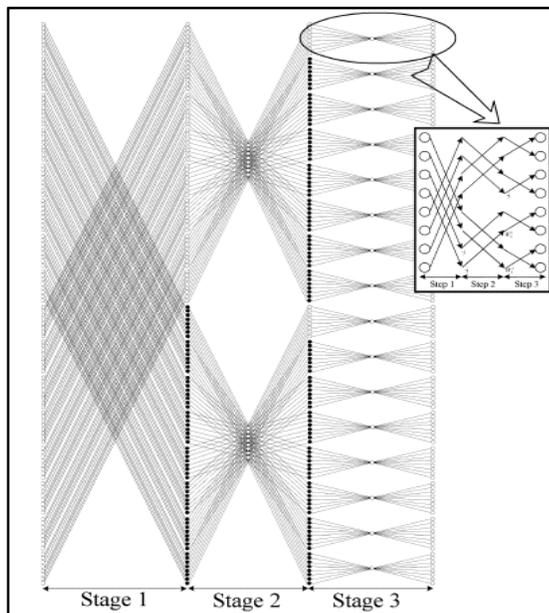


Figure 2: 128-point mixed-radix FFT algorithm

Radix-2 FFT algorithm is used in the first stage, and the radix-8 algorithm is applied in the second and third stages. Total number of complex multiplier required for normal DFT operation is  $N^2$ . Same DFT can be done by FFT radix-2 algorithm  $N/2 \log_2 N$ . Again this multiplier can be reduced by using mixed radix algorithm.

### 5. Comparison Of Normal DFT And Mixed Radix

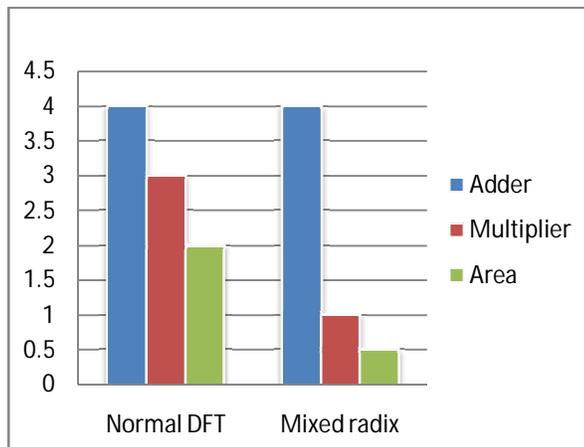


Figure 3: Comparison of Normal DFT & Mixed Radix

**6. Conclusion**

The FFT architecture, UWB specifications have been designed, simulated and simulation results are compared to normal DFT. It is found that, mixed radix architecture having better performance. The mixed radix multipath feedback has the best performance in terms of reduced the multiplier and power consumption. In future it can be implemented in FPGA for spectrum analysis.

---

**7.Reference**

1. A. Batra et al., "Multi-Band OFDM Physical Layer Proposal for IEEE 802.15 Task Group 3a," IEEE P802.15-03/268r3, Mar. 2004.
2. S. Magar, S. Shen, G. Luikuo, M. Fleming, and R. Aguilar, "An application specific DSP chip set for 100 MHz data rates," in Proc. Int.Conf. Acoustics, Speech, and Signal Processing, vol. 4, Apr. 1988, pp.1989–1992.
3. S. He and M. Torkelson, "Designing pipeline FFT processor for OFDM (de)modulation," in Proc. URSI Int. Symp. Signals, Systems, and Electronics, vol. 29, Oct. 1998, pp. 257–262.
4. J. O'Brien, J. Mather, and B. Holland, "A 200 MIPS single-chip 1 k FFT processor," in IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers, vol.36, 1989, pp.
5. B. M. Bass, "A low-power, high-performance, 1024-point FFT processor," IEEE J. Solid-State Circuits, vol. 34, no. 3, pp. 380–387, Mar.1999.
6. L. R. Rabiner and B. Gold, Theory and Application of Digital Signal Processing. Englewood Cliffs, NJ: Prentice-Hall, 1975.
7. J. Garcia, J. A. Michel, and A. M. Burón, "VLSI configurable delay Commutation for a pipeline split radix FFT architecture," IEEE Trans.SignalProcess. vol. 47, no. 11, pp. 3098–3107, Nov. 1999.
8. W.-C. Yeh and C.-W. Jen, "High-speed and low-power split-radix FFT," IEEE Trans. Acoust., Speech, Signal Process, vol. 51, no. 3, pp.864–874, Mar. 2003.
9. L. Jia, Y. Gao, J. Isoaho, and H. Tenhunen, "A new VLSI-oriented FFT Algorithm and implement," in Proc. IEEE Int. ASIC Conf., Sep. 1998, pp. 337–341.
10. K. Maharatna, E. Grass, and U. Jagdhold, "A 64-point fourier transform Chip for high-speed wireless lan application using OFDM," IEEE J. Solid-State Circuits, vol. 39, no. 3, pp. 484–493, Mar. 2004.
11. Y. Jung, H. Yoon, and J. Kim, "New efficient FFT algorithm and Pipeline implementation results for OFDM/DMT applications," IEEE Trans. Consum. Electron., vol. 49, no. 1, pp. 14–20, Feb. 2003.