

# Design and Implementing an Area and Energy Efficient High-Speed Low-Complexity Mixed-Radix Variable Length FFT Processor

Subashini C\*<sup>1</sup>, Senthil Kumar K<sup>2</sup>, Anand M<sup>2</sup>

\*<sup>1</sup> Research Scholar, Department of ECE, Dr. M.G.R Educational and Research Institute University, Chennai,

<sup>2</sup> Professor, Department of ECE, Dr. M.G.R Educational and Research Institute University, Chennai.

Email: subashini200235617@yahoo.co.in

**Abstract:** For solving destructions interrelated to orthogonal subcarriers, one of the predominant transformation method used extensively in telecommunications and digital signal processing especially in OFDM systems is Fast Fourier Transformation (FFT). Reducing the complexity and increasing the speed of a process can be obtained by using a modified FFT processor. Complexity is reduced by minimizing the number of multipliers. A Mixed-radix algorithm incorporated with Single path Delay Feedback (SDF) pipeline FFT architecture is used whereas it can be obtained by reconfiguring the FFT processor as 128-256-512-1024-2048-4096 points to reduce the computational complexity in this paper. Wherever it improves the throughput using the mixed-radix processing blocks, the area power trade-off is done. The area utilization is diminished by using higher radix butterfly structures as Radix-2<sup>5</sup> additionally. The Mixed-Radix Single-path Delay Feedback (MR-SDF-FFT) processor is synthesized by cadence using an UMC 180-nm CMOS cell library. A perfect simulation is carried out and the results are verified. The performance is evaluated by comparing the obtained results with the proposed results.

**Key words:** FFT Processor, Single-Path Delay Feedback, Power Consumption, Low Complexity, High Speed, Mixed-Radix, Variable-Length.

## 1. Background Study

Most of the earlier research works focused on estimating frequency spectrum, fast Fourier transformation and OFDM based communication in various communication based applications. All these kinds of applications have been solved mainly using signal processing methodologies. The size of the FFT varies in accordance to the OFDM systems. Some of the application standards and the required FFT size are given in Table-1. Fifty percentages of the applications are portable whereas power consumption is a major problem to be focused on them. Hence, the recent applications are motivated to design and implement a power scalable, low area utilization, high speed and variable length FFT processor.

There are three different FFT architectures are used in the communication domain. They are memory based, pipeline based and general purpose DSP based variable length FFT processors. These

three FFT processors are efficient in terms of area efficiency, throughput efficiency and flexibility in terms of neither power nor area efficient respectively. The FFT size can be varied in accordance to the application and the requirement of the application. From the above the pipeline architecture is selected to meet the real time application based requirements. Pipeline architecture in a fixed radix SDC is implemented by Hasan et al. (2003). A factor 4 is used to increase the size of the FFT. But flexibility of the FFT size is implemented can be done by utilizing mixed-radix algorithm verified based on various architectures. For example, radix 2-4-4-8-8- SDF is adopted by Guihua Liu and Quanyuan Feng (2007), radix 2-2-8-8-8 SDF is adopted by Lin et al. (2005), and radix 8-8-8-r(2/4/8) MDC is adopted by Lai and Wei (2006) and the results are verified. High flexibility with low power consumption is obtained using a butterfly-reconfigurable radix – 2/4/8 architecture by Lai and Wei (2006), and it is power consumption based architecture.

**Table-1:** FFT Size for Various Applications

Application	System	FFT Size
Wireless Network	WLAN – 802.11a	64
	WLAN -802.16e	256, 512, 1024
Wired Network (Broadband)	ADSL	512
	VDSL	256, 512, 1024, 2048, 4096, 8192
Digital Terrestrial Broadcasting	DAB	256, 512, 1024, 2048
	DVB-T	2048, 8192
	ISDBT-T	2048, 4096, 8192
	DMB-T/H	4096

In order to increase the efficiency of the FFT architecture in accordance to the emerging applications' requirements such as area, low-power and high speed efficiency this paper designed a new algorithm. The proposed algorithm used to reconfigure the structure as a radix-2, radix-2<sup>2</sup>, radix-2<sup>3</sup> and radix-2<sup>4</sup> involving variable size of FFT and it is obtained by, by-passing the stages. Also the efficacy can be increased by using two standard programmable complex multipliers and three constant complex multipliers. Comparing with the existing research works like architectures proposed in Guihua Liu and QuanyuanFeng (2007), Lin et al. (2005), Lai and Wei (2006), and other pipeline

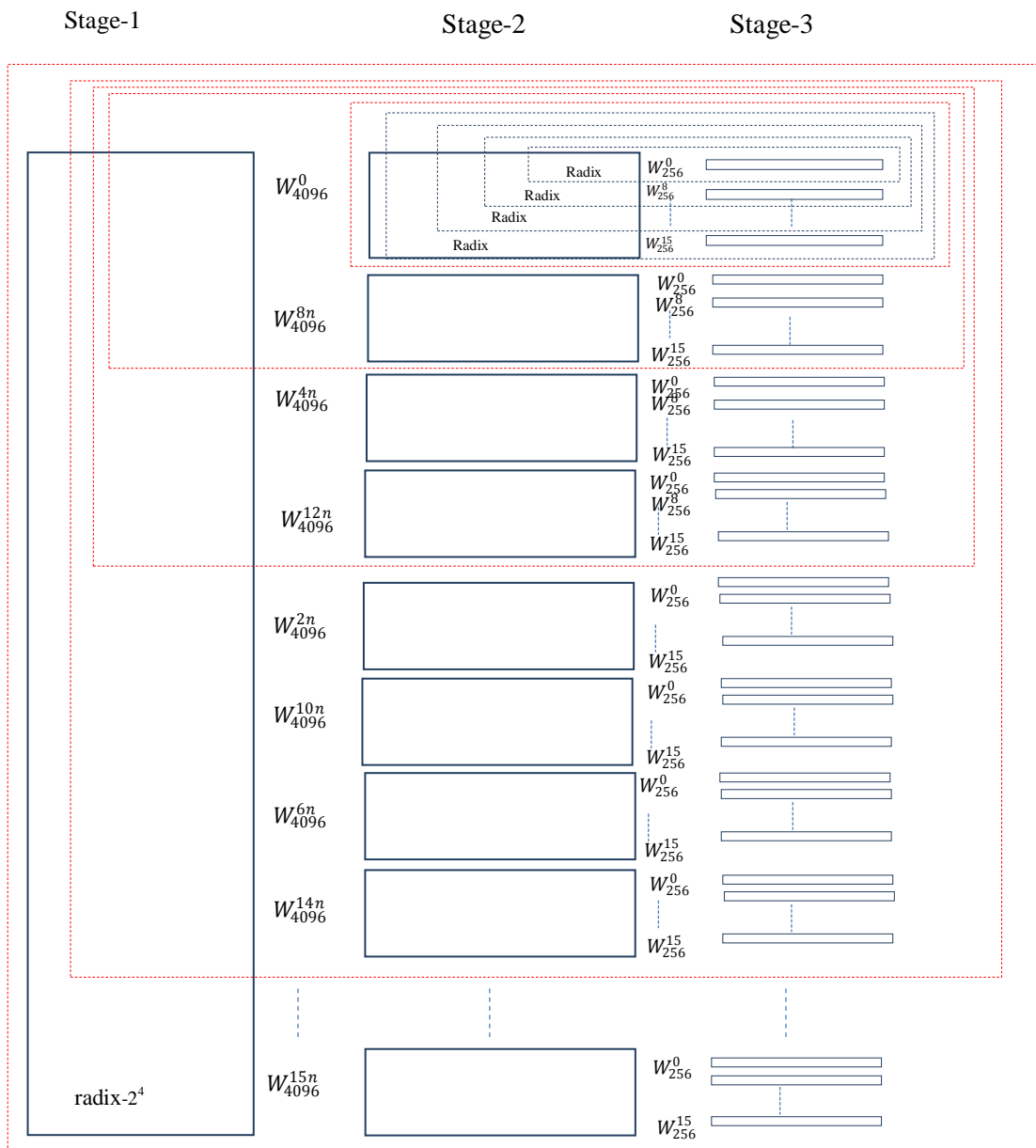
reconfigurable FFT, the hardware and power utilization is reduced using the higher level radix algorithms.

**Algorithm -1**

The core functionality of the DFT with N-point representation can be written as:

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

Where,  $W_N^{nk} = e^{-j2\pi\frac{nk}{N}}$  represents the twiddle factor, time index is denoted by  $n$ , and frequency index by  $k$ .



**Fig.1.** Variable Length FFT

Most of the existing research works were focused on reducing the complexity by utilizing various kinds of FFT architectures and algorithms. Requirements in terms of hardware is analyzed for N-length FFT based on various categories of pipeline architectures by Santhi et al. (2008). Highest throughput can be

provided by MDC-architecture comparing with the existing architectures. But it needs high memory, more number of complex adders and multipliers than the other architectures. To reduce the hardware complexity, memory size the number of multipliers and adders should be reduced whereas it can be

achieved using radix-2<sup>4</sup> SDF architecture. Thus, in the design, radix-2<sup>4</sup> algorithm is adopted. Here it is used the mixed-radix algorithm to attain the reconfigurable suppleness. The algorithm is derived in detail, below referred from Yu-Wei et al. (2005), for N-points DFT. Let,

$$N = r_1 \times r_2$$

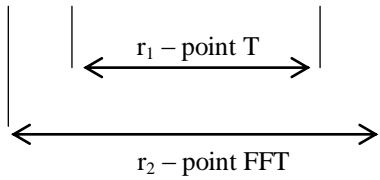
$$n = r_2 n_1 + n_2, \begin{cases} n_1 = 0, 1, \dots, r_1 \\ n_2 = 0, 1, \dots, r_2 \end{cases}$$

$$k = k_1 + r_1 k_2, \begin{cases} k_1 = 0, 1, \dots, r_1 \\ k_2 = 0, 1, \dots, r_2 \end{cases}$$

Now the expression (1) can be re-written as:

$$X(r_1 k_2 + k_1) = \sum_{n_2=0}^{r_2-1} \sum_{n_1=0}^{r_1-1} x(r_2 n_1 + n_2) W_N^{(r_2 n_1 + n_2)(r_1 k_2 + k_1)}$$

$$= \sum_{n_2=0}^{r_2-1} \left[ \sum_{n_1=0}^{r_1-1} x(r_2 n_1 + n_2) W_{r_1}^{n_1 k_1} W_N^{n_2 k_1} \right] W_{r_2}^{n_2 k_2}$$



The entire equation given above represents the  $r_2$  - point DFT and the red color portion of the above equation represents the  $r_1$ -point DFT. The above equation is used to separate the DFT into  $r_1$ -point and  $r_2$ -point DFT. Similarly in  $r_3$ -point and  $r_4$ -point DFT combination,  $r_3$ -point DFT is separated. Jung-Yeol et al. (2004) verified that 16-point DFT based on radix-2<sup>4</sup> FFT obtained low computational complexity. In the pipeline architecture an N-Length FFT architecture is proposed like  $N = 16 \times 16 \times 16$ . Only two complex and three constant multipliers are used in the design. Notice that the  $r_1$ , in equation (3) can be assigned as 2, 4, 8 or 16. Hence,  $N$ ,  $N/2$ ,  $N/4$  and  $N/8$  FFT can be achieved by letting  $r_1 = 16, 8, 4$  or 2. Unlike 8 FFT size is comprehended by decaying the DFT applicably as shown in Table-2.

Figure-2, shows the radix-2<sup>4</sup> FFT architecture where the size of the smallest FFT is comprehended. The lowest radix-2<sup>4</sup> stage is fixed while higher stages are reconfigurable. For large size FFT, lower stages are selected initially and allocated to fixed radix-2<sup>4</sup>. The highest stage is reconfigured as 2, 2<sup>2</sup>, 2<sup>3</sup> and 2<sup>4</sup> according to dissimilar FFT size. It is noticed that the elementary radix-2<sup>4</sup> stage assists as radix-2<sup>3</sup> FFT if we mistreat the initial butterfly from Figure-2. The same basic radix-2<sup>4</sup> stage can serve as radix-2<sup>3</sup> and radix-2 as well if we neglect the first two butterflies and first three butterflies respectively.

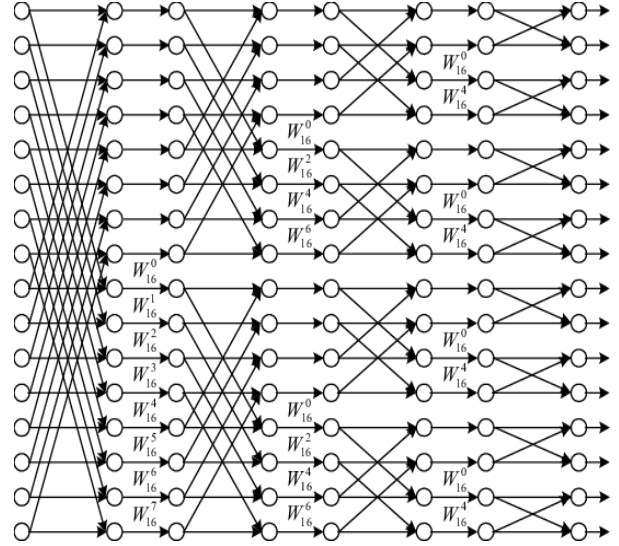


Fig.2. Radix-2<sup>4</sup> FFT

Table-2: Various Decomposition size of FFT

FFT Size	Radix-2 <sup>4</sup> stage		
	Stage-1	Stage-2	Stage-3
16	1	1	2 <sup>4</sup>
32	1	2	2 <sup>4</sup>
64	1	2 <sup>2</sup>	2 <sup>4</sup>
128	1	2 <sup>3</sup>	2 <sup>4</sup>
256	1	2 <sup>4</sup>	2 <sup>4</sup>
512	2	2 <sup>4</sup>	2 <sup>4</sup>
1024	2 <sub>2</sub>	2 <sup>4</sup>	2 <sup>4</sup>
2048	2 <sup>3</sup>	2 <sup>4</sup>	2 <sup>4</sup>
4096	2 <sup>4</sup>	2 <sup>4</sup>	2 <sup>4</sup>

For example, the input data is given to second stage of the second butterfly while computing 128-point FFT. The three stages based variable length FFT processor is illustrated in Figure-1. The table comprises of 0's and 1's, where "1" represents the signal is selected in the particular stage, "0" means not selected. The architecture has three stages and which are having four butterfly unit.

Table-3: Selected Signals of Various FFT Size

FFT Size	S1	S2	S3	S4	S5	S6	S7	S8
16	1	1	1	1	1	1	1	1
32	1	1	1	1	1	1	1	0
64	1	1	1	1	1	1	0	0
128	1	1	1	1	1	0	0	0
256	1	1	1	1	0	0	0	0
512	1	1	1	0	0	0	0	0
1024	1	1	0	0	0	0	0	0
2048	1	0	0	0	0	0	0	0
4096	0	0	0	0	0	0	0	0

Also First In First Out based memory units, complex multipliers, butterfly units and adders are placed in the FFT processor. By organizing these 8 multiplexers, it is easy to reconfigure it as 16~4096-points FFT. The process of signal selection under

various FFT is given in Table-3. For instance, S5=1, S6~S7=0 when 128-points FFT process is essential. The input data skip the first radix-2<sup>4</sup> stage and the 1<sup>st</sup> butterfly of 2<sup>nd</sup> stage and flow directly into the 2<sup>nd</sup> butterfly of 2<sup>nd</sup> stage. To incapacitate the advanced butterflies which are not cast-off whereas computing smaller size FFT, clock gating technology is executed. Thus, the power-scalable feature is achieved.

## 2. Butterfly Unit

It is well known and given in Figure-2 is, all radix-2<sup>4</sup> comprises of two radix-2 units. The implementation of single flow radix-2<sup>4</sup> butterfly architecture is illustrated in Figure-2. Also the first two stages such as 1 and 2 are reconfigurable. That is radix-2, 2<sup>2</sup>, 2<sup>3</sup> and 2<sup>4</sup> butterfly units are reconfigurable to do service. A dual-port register file is utilized for implementing the FIFO operation through shift registers. The shift registers accept data from the outputs of a butterfly unit, and then shift left and feed the data back to the same butterfly unit. In case of multipliers it is implemented using CSD representation, whereas it provides a constant complex multiplier. Hence the hardware complexity and power is diminished ominously. By generating twiddle factors are required to persist in ROM in order to generate multipliers in various stages.

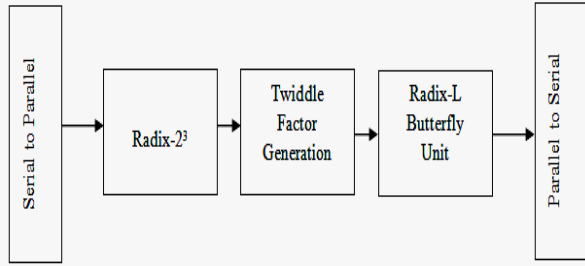


Fig.3. Variable Length SDF FFT Processor

### Twiddle Factor Generation:

In this paper twiddle factor generation is integrated with radix-2<sup>3</sup> FFT processor and it is shown in Figure-3. Here, radix-2<sup>3</sup> FFT is a single unit and it can be converted into multiple. The twiddle factors  $W_N^{il}$  will vary correspondingly when L change from 2 to 4 and length of the FFT is N, which is 2<sup>k</sup> or 2<sup>3k</sup> × 3 – point variable. For example, in this case, let L (L = 2, 3, 4) all four channels are empowered to accomplish radix-2<sup>3</sup> FFT. The twiddle factors equivalent to all channel are  $W_N^0$ ,  $W_N^1$ ,  $W_N^{2i}$  and  $W_N^{3i}$  according to

$$X(i + M \cdot j) = \sum_{l=0}^{L-1} [X_i(i) \cdot W_N^{il}] \cdot W_L^{jl}$$

In order to design and implement a twiddle factor generation unit, a ROM based LUT method is used whereas it utilizes more memory resources. In order

to investigate the efficiency of and check the suitability of twiddle factor unit into emerging applications it is applied and verify into a CORDIC (latest computer) used in real time applications. Architecture given in Figure-3 can be applied to a 4 parallel radix-2<sup>3</sup> FFT processor to a minimum of 3 CORDIC units is needed to carry out an assured parallel processing. It is used for resource consumption in high logic applications. For example a trigonometric function associated with FFT process is follows:

$$\cos 2x = \cos^2 x - \sin^2 x, \quad \cos 3x = \cos 2x \cdot \cos x - \sin 2x \cdot \sin x$$

$$\sin 2x = 2 \cdot \sin x \cdot \cos x, \quad \sin 3x = \sin 2x \cdot \cos x + \cos 2x \cdot \sin x$$

Calculating the above *cos* and *sin* functions in terms of *x* using multiple addition and multiplication functions is easier one. Calculating the twiddle factor value for  $x = 2\pi i/N$  can be expressed as follows:

$$W_N^i = \cos x - j \sin x$$

$$W_N^{2i} = \cos 2x - j \sin 2x$$

$$W_N^{3i} = \cos 3x - j \sin 3x$$

To generate  $W_N^i$  one CORDIC unit is utilized. To generate  $W_N^{2i}$  or  $W_N^{3i}$ , additions and multiplications are applied for L=3 and L=4. By verifying the results in terms of logic resources is compared between different methods in accordance to FPGA logic units. From the comparison it is noticed that the proposed method saved 31% of logic-resources. The block diagram of the twiddle factor generation unit used in this paper is illustrated in Figure-4. The address generator is fundamentally a modulo-M counter. It produces the phase  $x = 2\pi i/N$  ( $i = 0, 1, \dots, M-1$ ) for the CORDIC unit. Thus,  $W_N^0$ ,  $W_N^i$ ,  $W_N^{2i}$  and  $W_N^{3i}$  are generated for each channel.

Table-4: Twiddle Factors Generated for Radix-2<sup>k</sup> Algorithm

Scheme	Proposed Cordic + Multiplier + Adder			Cordic
	12-bit cordic	12-bit multiplier	real 12-bit real adder	
Unit				12-cordic
LUTs per unit	1090	162	12	1090
Amount	1	7	3	3
Total LUTs	2260			3270

### FFT Decomposition for Parallel Architecture:

One of the main methodology leads to increase the throughput and decrease the power consumption in FFT processor is parallel processing. Parallel architecture increases the performance for a constant throughput, fewer clock frequency and ascended voltage. Because of in SDF architecture, time-

multiplexing is applied, the parallel processing is used for energy efficiency to control the design point in the area as well as energy and delay calculation.

Generally mixed radix FFT decomposition is used to design the mixed radix processors. Decomposition is obtained by factorizing N point into smaller integers. Leveraging the decomposition of FFT can provide area efficiency in parallel architecture. It well known and noticed that an N-point FFT is decomposed into L point and M Point FFT to increase the serial to parallel processes in term of area. Whenever  $N=L$  is obtained the single input SDF FFT is integrated with single input parallel FFT where it reduces the area. The proposed design is reconfigurable from 128-4096 point FFT. This model helps to obtain the parallel architecture of variable length FFT processor.

**Power and Area Minimization:**

The design method used for FFT comprises architecture, algorithm and the parameters used to represent the circuit operations. Several numbers of existing approaches discussed about 128-2048 point FFT processor through parallel architecture or by radix factorization. Including this delay buffers also implemented for reducing power consumption. The word length is evaluated in term of analysing the partitioning the memory cells and the memory elements. In this paper the proposed approach gives a FFT processor for parallel processing, by the basic circuit design procedure is combined with the parameter optimization. From the output it is illustrated that the proposed FFT consumes less energy comparing with the existing works.

**Radix-2/4/8 Multiplexing Butterfly Unit:**

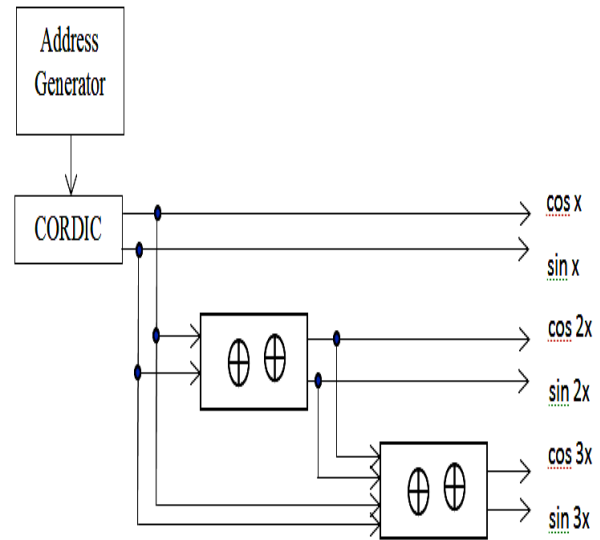
The radix-2<sup>4</sup> SDF architecture always requires small memory, less number of adders and multipliers. In this paper it is aimed to reduce the computational complexity by implementing the radix-2/4/8 algorithm. The complexity of radix2/4/8 is reduced one third complexity of the radix-2 algorithm. Also the radix-2/4/8 comprises of cascaded radix-2 where it gives a high speed and less area utilization. As discussed above, the radix-L butterfly unit should support radix-2, radix-3 and radix-4 and three modes to achieve a 2<sup>k</sup> or 2<sup>3k</sup>×3-point variable FFT processor. A number of 32 real-adders and a 3-to-1 multiplexer is required if three butterfly units are planned independently. As shown in Figure-5 we recommend a radix-2/4/8 multiplexing butterfly unit. Only 20 real-adders and 7 2-to-1 multiplexers are employed.

The proposed variable length FFT processor can perform different lengths as 512, 1024, 2048 and 4096 point FFT operations. In order to carry out variable point FFT, mixed radix FFT process is applied. This mixes radix-2/4/8, radix-2, radix-2<sup>2</sup>, radix-2<sup>3</sup> and radix 2<sup>4</sup> algorithms. The different stages

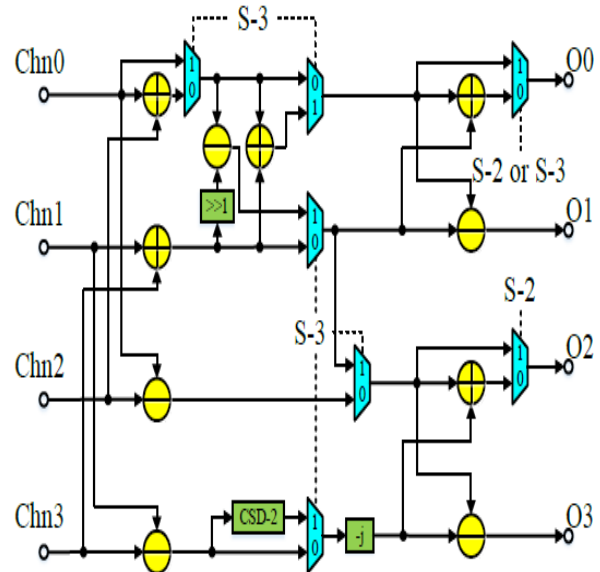
and points of the variable length FFT processor architecture are given in-detail in Table-5.

**Table-5:** Different Point of Variable-length FFT Processor in Different Stages

Points	Stage-1	Stage-2	Stage-3	Stage4-6	Stage7-9	Stage10-12
512-points	Nil	Nil	Nil	Nil	Nil	Nil
1024-points	Nil	Nil	radix-2	Radix-2/4/8	Radix-2/4/8	Radix-2/4/8
2048-points	Nil	Radix-2 <sup>2</sup>	Radix-2 <sup>2</sup>			
4096-points	Radix-2/4/8	Radix-2/4/8	Radix-2/4/8			



**Fig.4.** Twiddle Factor Generation Diagram



**Fig.5.** Butterfly Unit of Radix 2/3/8

A two-bit control signal denoted as S-2 and S-3 is used for configuring the butterfly unit into three modes and it is given in Table-6. In this paper, four

radix-2<sup>3</sup> FFT processors are working simultaneously for configuring to radix-4 mode. By making the input and output as asymmetric during the butterfly unit configure into radix-L(L= 2, 3, 4, 8) mode. The channel input is represented as chn0, chn1 and chn3. The output is represented as O0, O1 and O2. The proposed architecture is worked on parallel or in variable length FFT. Also it takes only lesser time for execution. The proposed architecture obtained less power consumption and lesser area comparing with Yang et al. (2012). This author considering the working frequency and voltage supply is the problem and provide solution for power consumption.

**Table-6:** I/O Mode Control in Butterfly Unit

Mode	S-2, S-3	Input	Output
Radix-2	1, 0	Chn0, Chn2	O0, O2
Radix-3	0, 1	Chn0, Chn1, Chn3	O0, O2, O3
Radix-4	0, 0	Chn0, Chn1, Chn2, Chn3	O0, O1, O2, O3

Since the technology used for processing is different, the area and power given in Yang et al. (2013) are normalized as follows:

$$A_{nor} = \frac{Area}{(Technology/65)^2}$$

$$P_{nor} = \frac{Power}{(Technology/65)}$$

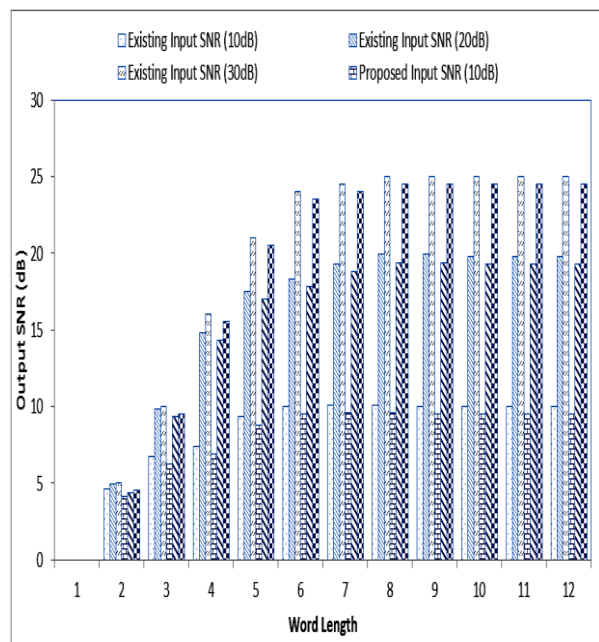
It is obtained that the normalized area and power are 1.62 mm<sup>2</sup> and 46mW, respectively. The superiority of the proposed design is obvious. Cho et al. (2013) and Wang et al. (2014) only focus on the design of 512-point FFT processor. The area is normalized using the above equations. Where  $A_{nor}$  denotes the normalized area. The normalized area in Cho et al. (2013) and Wang et al. (2014) are 1.63 mm<sup>2</sup> and 1.69 mm<sup>2</sup>, respectively. The core area of proposed design is also smaller. Similarly the power and area efficiency can be obtained by variable length FFT which is given in Table-7 comparing with the existing approaches in Yang et al. (2012), Chu Yu et al. (2014).

### 3. Simulation Results

The proposed design is investigated 4096-point FFT, it diminishes the hardware complexity and reduces the cost and power consumption. Incorporating the 4096-point FFT with radix-24 SDF parallel FFT processor simplifies the data flow and the complexity than the existing architectures. This paper focused on sharing/reusing the hardware to decrease the memory utilization and computational

complexity. The proposed design is experimented by implementing on 90nm CMOS technology. From the simulation results it is noticed that the area utilization and power consumption are reduced. It is noticed that the finite word length processing is most essential to obtain a required output-signal-to-noise-ratio and fixed hardware cost. For each point of FFT, the word length selection of an input signal sequence is feed into the FFT processor including Gaussian noise. After experiment the output signal is obtained with various input SNR and word lengths. The obtained results of the output SNR simulation is shown in Figure-5. It shows that it is necessary to control the noise level on the input signal. The word length selected by the existing approaches [9, 14] exceeds 11 bits whereas the proposed design selected the word length is of 12 bits.

Table-7 gives the details about normalized power followed in the proposed design whereas it is comparatively lesser than the existing approach discussed in Yang et al. (2012), and in Chu Yu et al. (2014). The reason behind of comparing with these mentioned existing approaches is they incorporated a parallel SDF variable length FFT architecture to get configurability and supports the maximum size of 2048/1536-point FFT. The existing approach in Yang et al. (2012), discussed about parallel architecture for single input stream. But our proposed method utilizes lower number of hardware in terms of constant and complex multipliers with adders. Hence our proposed design saves hardware cost with lesser memory.



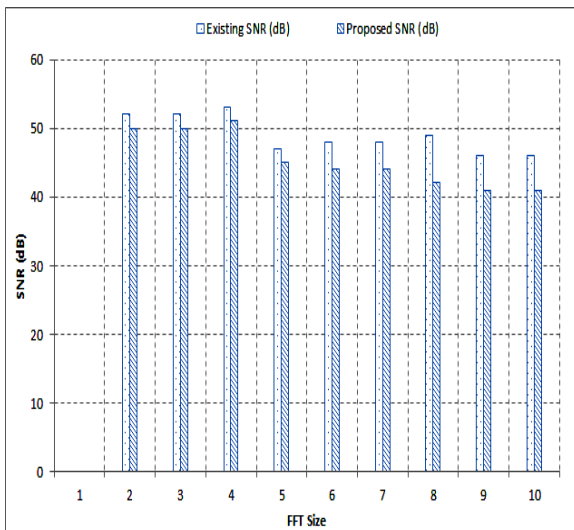
**Fig.6.** Output SNR for Various Input word length

**Table-7:** Performance Comparison

Parameters	Existing-1 [Yang et al. (2012)]	Existing-2 [Chu Yu et al. (2014)]	Proposed
FFT Size	128-2048/1536	128-2048/1536	4096
Word Length	12	12	12
Technology	65nm	90nm	90nm
Chip Core	8.55mW @ 0.45V, 20MHz	7.2mW @ 0.9V, 40MHz	7.0mW @ 0.9V, 35MHz
Normalized Power (mW)	0.54	0.33	0.31
Normalized Area(mm <sup>2</sup> )	0.171	0.408	0.4
ROM size (x 10 bits)	9216	4128	4512
Complex Multipliers	16	4	4
Constant Multipliers	760	38	38
Multiplexers	76	10	8

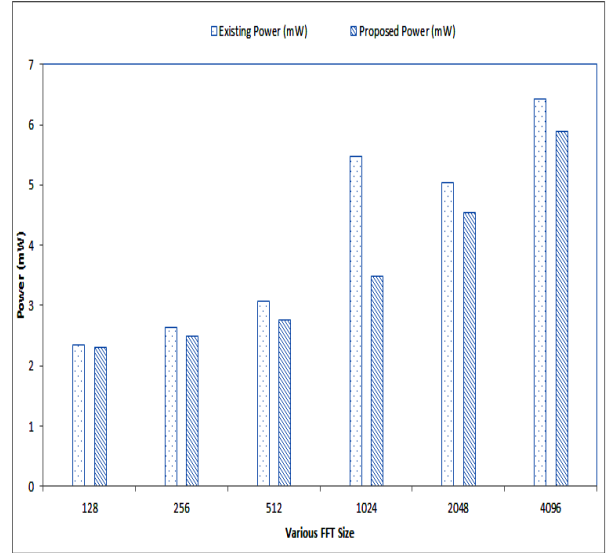
**Table-8:** Proposed Design Information

Process	90 nm CMOS
Core Size	0.87 x 0.90 mm <sup>2</sup>
Die Size	1.44 x 1.44 mm <sup>2</sup>
Gate Counts	204687
Clock Rate	40 MHz
Pin Count	88
FFT Size	128  256  512  1024 1536 2048 4096
Power (mW)	2.31 2.49 2.76 3.49  4.55  4.92  5.43
Power with I/O Pads @ 40MHz	I/O Pads   2.1 mW
	Registers   1.0 mW

**Fig.7.** Various FFT Size with SQNR

The following Table-8 shows the details of the parameters involved in the proposed design architecture. It comprises of various information like I/O pad counted, gates counted ad power consumed

for various lengths of FFT processors like 128 to 4096. It also includes the information about the core chip. From table-8, it is noticed that the memory utilized the half the amount from the total power. The power and area performance in terms of processor information it provides good balance among the area and power utilization.

**Fig.8.** Power Consumption for Various FFT Size

In order to verify the performance of the proposed approach, the variable length FFT is experimented through a simulation. The simulation is carried out in 0.9V with 35MHZ frequency. The word length is 12 bits may change due to the application. Since we are using Radix-2<sup>4</sup> variable length parallel FFT processor it is specifically used for multimedia applications. The SQNR and power consumption for various FFT size is shown in Figure-7 and in Figure-8. The power consumption of various FFT size in terms of various reconfigurable FFT can be obtained. From the result given in Figure-7 and 8, it is identified that the obtained SNR and power efficiency of the proposed approach is better than the existing approaches.

#### 4. Conclusion

For reducing the complexity, diminishing the power consumption and area utilization, the main objective of this paper is to design and implement a parallel FFT architecture. 128 to 4096 points variable length parallel FFT processor is incorporated with mixed radix-2<sup>4</sup> algorithms in order to do that. This paper has presented a low-power and area-efficient variable-length FFT processor for 128, 256, 512, 1024, 2048 and 4096-points FFT which can be reconfigured. It's characterized of achieving low-power consumption, small area cost and high reconfigurable flexibility by adopting mixed radix-2<sup>4</sup> algorithms. Compared with other variable length FFT processor and achieves scalable power

consumption, the power comparison shows that it consumes much smaller power. In this paper, it is also discussed the design of a SDF mixed-radix variable-length FFT processor. In order to minimize the number of occupied complex multipliers, we adopt radix-2<sup>3</sup>FFT algorithm and 4 parallel processing channels. When combining with a reconfigurable radix-L butterfly unit, the processor can perform 2<sup>k</sup> or 2<sup>3k</sup>×3-point FFT. Compared with several related works with the design and optimization of CSD constant multipliers, Twiddle factor generation block and radix-2/3/4 multiplexing butterfly unit, used in the proposed design attain steiner execution latency, lesser core area and lower power consumption. FFT architecture the objective of this paper is achieved proved by simulation based experiment by integrating radix algorithm with parallel variable length.

## References

1. Cho, Taesang, and H. Lee: "A high-speed low-complexity modified radix-25, FFT processor for high rate WPAN applications," *IEEE Transactions on Very Large Scale Integration Systems* 21 (2013) 187 (DOI: 10.1109/TVLSI.2011.2182068).
2. Chu Yu, and Mao-Hsu Yen, "Area-Efficient 128- to 2048/1536-Point Pipeline FFT Processor for LTE and Mobile WiMAX Systems", *IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS*, 2014.
3. K. Chalennsuk, R.H. Spaanenburger, L. Spaanenburger, M. Seutter and H. Stoorvogel, "Flexible-length fast fourier transform for COFDM," 15<sup>th</sup> IEEE International Conference on Electronics , Circuits and Systems, pp. 534-537, Aug 2008.
4. M. Hasan, I. Arslan, and I.S. Thompson, "A delay spread based low power reconfigurable FFT processor architecture for wireless receivers," *Proceedings International Symposium on System-on-Chip*, pp. 135-138, Nov 2003.
5. Guihua Liu and QuanyuanFeng, "ASIC design of low-power reconfigurable FFT processor," 7th International Conference on ASIC, pp. 44-47 , Oct 2007
6. Y.-I. Lin, P.-Y. Tsai and T.-D. Chiueh, "Low-power variable-length fast Fourier transform Processor," *IEE Proc. Computers and Digital Techniques*, vol 152, pp 499-506, July 2005.
7. Chiehen Lai and Wei Hwang, "A Low-Power Reconfigurable Mixed Radix FFT IFFT Processor," *IEEE Asia Pacific Conference on Circuits and Systems*, pp. 1931-1934, Dec 2006.
8. M. Santhi, S.Arun Kumar, G.S.P. Kalish, K.Murali, S.Siddharth and G. Lakshminarayanan, "A Modified Radix-24 SDF Pipelined OFDM Module," *ICCCn 2008 International Conference on Computing, Communication and Networking*, pp. 1-5, Dec 2008.
9. Jung-yeol Oh and Myoung-seob Lim, "Fast Fourier transform processor based on low-power and area-efficient algorithm", *Proc. IEEE asia-pacific Conference on Advanced System Integrated Circuits*, pp. 198-201, Aug 2004.
10. Yang, C. H., T. H. Yu, and D. Markovic: "Power and Area Minimization of Reconfigurable FFT Processors: A 3GPP-LTE Example," *IEEE Journal of Solid-State Circuits* 47 (2012) 757 (DOI: 10.1109/JSSC.2011.2176163).
11. Yang, K. J., S. H. Tsai, and G. C. H. Chuang: "MDC FFT/IFFT Processor With Variable Length for MIMO-OFDM Systems," *IEEE Transactions on Very Large Scale Integration Systems* 21 (2013) 720 (DOI: 10.1109/TVLSI.2012.2194315).
12. Yu-Wei Lin, Hsuan-Yu Liu, Chen-Yi Lee, "AI-GS/s FFT/IFFT Processor for UWB Applications," *IEEE Journal of Solid-State Circuits*, vol 40, pp 1726-1735, Aug 2005.
13. C.-H. Yang, T.-H. Yu, and D. Markovi'c, "Power and area minimization of reconfigurable FFT processors: A 3 GPP-LTE example," *IEEE J. Solid-State Circuits*, vol. 47, no. 3, pp. 757-768, Mar. 2012.
14. Wang, C., Y. Yan., and X. Fu: "A High-Throughput Low-Complexity Radix-24-2<sup>2</sup>-2<sup>3</sup> FFT/IFFT Processor with Parallel and Normal Input/Output Order for IEEE 802.11ad Systems," *IEEE Transactions on Very Large Scale Integration Systems* 23 (2014) 2728 (DOI: 10.1109/TVLSI.2014.2365586).