Reconfigurable FFT Processor – A Broader Perspective Survey

V.Sarada[1], T.Vigneswaran[2]
[1]ECE, SRM University,
Chennai, India.
saradasaran@gmail.com
[2]ECE, VIT University,
Chennai, India.
vigneshvlsi@gmail.com

***Abstract* -The FFT(Fast Fourier Transform) processing is one of the key procedure in the popular orthogonal frequency division multiplexing(OFDM) based communication system such as Digital Audio Broadcasting(DAB),Digital Video Broadcasting Terrestrial(DVB-T),Asymmetric Digital Subscriber Loop(ADSL) etc.These application domain require performing FFT in various size from 64 to 8192 point. Implementing each FFT on a dedicated IP presents a great overhead in silicon area of the chip. By supporting the different sizes of FFT for new wireless telecommunication standard may increase the time to market it. This consideration make FFT ideal candidate for reconfigurable implementation. Efficient implementation of the FFT processor with small area, low power and speed is very important. This survey paper aims at a study on efficient algorithm and architecture for reconfigurable FFT design and observes common traits of the good contribution.**

***Keywords* - Reconfigurable FFT, Pipeline architecture, Mixed Radix,OFDM.**

I. INTRODUCTION

The Discrete Fourier Transform (DFT) is an important technique used in Many digital signal processing (DSP) system, however due to computational intensive and multiplicative complexity for N point is $O(N^2)$. The FFT was proposed by Cooley and Tukey to reduce the multiplication complexity of operation by $O(N \log_2 N)$.The OFDM technique, due to its effectiveness in overcoming adverse channel effect [1,2] as well as spectrum utilization has become widely adopted in wireless communication standard . The FFT and IFFT (Inverse Fast Fourier Transform) are the key computation blocks in OFDM system and dominate most area and power dissipation in implementation. Therefore efficient and low power VLSI implementation of FFT processor is essential for successful deployment of OFDM based system. With reference from Table 1. FFT with various sizes are required. The design of an efficient FFT hardware ability to perform different size is the area of current research.Acheiving low power, high throughput and low area is possible by selecting proper algorithm with parallel, Pipeline or memory sharing architecture and sharing the resources.

This paper is organized as follows. Brief review of FFT algorithm in section II, discuss on various FFT architecture in section III, reconfigurable FFT in section IV, literature review in section V and finally concluding remarks in section VI.

TABLE 1
Various FFT size for OFDM applications [3]

Application	System	FFT Size
Wireless Networks	WLAN(802.11a)	64
	WLAN(802.16e)	256, 512, 1024
Wired 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

II. FFT AGORITHM

The DFT of N point input sequence is defined as given below

$$x(k) = \sum_{n=0}^{N-1} x(n) W^{nK}, k = 0,1 \ldots N-1 \quad (1)$$

Where $W_N^{nk} = e^{-i2\pi nk/N}$ which is called twiddle factor. A direct implementation of the equation (1) requires N^2 complex multiplication and N(N-1) complex additions. The cooley-tukey had proposed an algorithm using divide and conquer technique to recursively partition a DFT of size N = r1 * r2 shown in equation (2). Considering the symmetry and periodicity property computation complexity of DFT is reduced.

Symmetry property: $W_N^{k+N/2} = -W_N^K$

Periodicity property: $W_N^{k+N} = W_N^K$

$N = r_1 \times r_2$

$n = n_{1+} r_1 n_2 \qquad n_1 = 0,1 \ldots r_1 - 1 , \; n_2 = 0,1 \ldots . r_2 - 1$

$k = r_2 k_1 + k_2 \qquad k_1 = 0,1 \ldots r_1 - 1 , \; k_2 = 0,1 \ldots . r_2 - 1$

$$X(k) = \sum_{n_1=0}^{r_1-1} \sum_{n_2=0}^{r_2-1} x(n_1 + r_1 n_2) W_N^{(n_1+r_1n_2)(k_2+r_2k_1)} = \sum_{n_1=0}^{r_1-1} \left\{ \sum_{n_2=0}^{r_2-1} x(n_1 + r_1 n_2) \; W_{r2}^{n_2k_2} \times W_N^{n_1k_2} \right\} W_{r_1}^{n_1k_1} \qquad (2)$$

r_2 point DFT in equation (2) is further decomposed. When N is not prime, then $N=r_1*r_2*r_3 \ldots . r_m$. N point DFT can be divided into combination of r_1, r_2, $r_3 \ldots . r_m$. when $r_1 = r_2 = r_3 = r_m = r$ and if r = 2 FFT algorithm is called fixed radix -2 algorithm. If r=4 FFT is fixed radix-4 algorithm. Decomposition of N point FFT as different combination of r_1, r_2, r_3 is called mixed radix algorithm. The basic r point of DFT is referred as butterfly unit. Dividing the output into smaller DFTs is called decimation in time algorithm (DIT), dividing the input into smaller DFTs is called decimation in frequency (DIF). The computation of DIT and DIF is similar. The DIF algorithm has the twiddle factor multiplication after DFT computation but in DIF algorithm the twiddle factor multiplication is before DFT computation. The basic butterfly diagram is shown in Fig1.

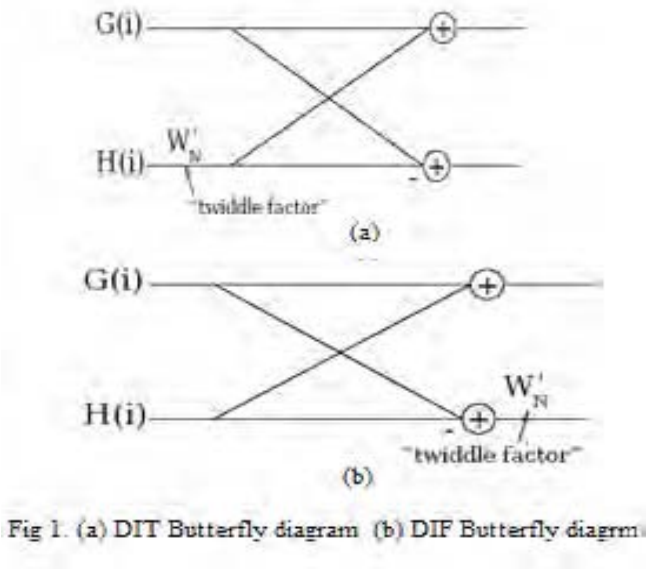

Fig 1. (a) DIT Butterfly diagram (b) DIF Butterfly diagram

Higher radix algorithm can reduce the computation complexity by reducing number of complex multiplication required. The trade off is that the hardware complexity of the butterfly grows as radix become high. Split radix FFT algorithm(SPRFFT) mixes different radix in each stage, yielding with fewer addition and multiplication than radix-r algoritm, however the split radix algorithm has an irregular signal flow graph and it can apply when N is multiple of 4. If decomposition of N is relative prime it is called Prime factor algorithm (PFA) which reduce the twiddle factor multiplication but complex re- indexing is required. The Winograd Fourier transform algorithm (WFTA) uses cyclic convolution method to compute the DFT. This algorithm minimizes the number of multiplication, however increase the arithmetic operation. Furthermore the irregularity of WFTA makes it impractical for most real application. We compare discussed algorithm from addition and multiplication point of view as shown in TABLE II.In radix–r FFT algorithm if $r=2^k$ radix-r butterfly is further decomposed by cascading k radix-2 stages known as radix-2^k [5] algorithm. Radix 2^4 is developed by He & Torkinson with rearrangement of 4 samples together in slightly different manner than radix-4 algorithm. It has same multiplicative complexity of radix-4, but has a signal flow graph similar to the radix-2 algorithm.

III. FFT ARCHITCTURS

Once the algorithm is selected, the architecture can be determined to implement the given algorithm. Various FFT algorithms have been developed after the Cooley-Tukey's publication and various implementation methods are also developed. Some of the common methods are discussed below.

A.Sequential architecture

The basic sequential architecture consists of a processing element (PE) that can compute a butterfly, and the memory which can be used to store the data, intermediate results and the twiddle factors. The amount of hardware involved is very small and it takes (N/2) $\log_2 N$ sequential operations to compute the FFT, but large latency and complex controller is required.

B. Fully parallel architecture

A fully parallel structure can be constructed by having a PE for each of the butterfly operations. This involves a lot of hardware and is not an attractive option for a large N point .Implementing the entire FFT structure in a parallel fashion may have the advantage of easy control (i.e. no controller required), low latency , customization of each twiddle factor as a multiplication by a constant. The disadvantages are it require huge area, and encounters more routing congestion.

C. Parallel Iterative architecture

By adding more processing elements to the architecture in each sequential pipeline stage, performance can be improved even further. The butterflies can then be computed in parallel in any stage. The total execution time for the parallel iterative Processor is ($\log_2 N$) cycles.

D. Pipeline architecture

To improve the performance of the sequential processor, parallelism can be introduced by using a separate arithmetic unit for each stage of the FFT. This increases the throughput by a factor of $\log_2 N$ when the different units are pipelined

TABLE II.
Multiplication and Addition complexity [4]

N	Radix – 2		Radix - 4		SPRFFT		PFA		WFTA	
	M	A	M	A	M	A	M	A	M	A
16	24	152	20	148	20	148				
60							200	888	136	888
64	264	1032	208	976	196	964				
256	1800	5896	1392	5488	1284	5380				
504							2524	13388	1572	14540
512	4360	13566			3076	12292				
1024	10248	30728	7856	28336	7172	27625				

M – No. of Multiplication

A – No. of Addition

This Architecture is also known as cascaded FFT architecture and will be used in most of the design. 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. The most common groups of the pipelined FFT architecture are given below and shown in Fig 2.(a-f)

- Radix-2 multipath delay commutator (R2MDC)
- Radix-2 single-path delay feedback (R2SDF)
- Radix-4 multipath delay commutator (R4MDC)
- Radix-4 single-path delay commutator (R4SDC)
- Radix-4 single-path delay feedback (R4SDF)

• Radix-2^2 single-path delay feedback (R 2^2SDF)

E. Radix-2 Multipath Delay Commutator

The Radix-2 Multipath Delay Commutator (R2MDC) architecture is the most straightforward approach to implement the radix-2 FFT. The input sequence has been broken into two parallel data stream flowing forward, with correct distance between data elements entering the butterfly scheduled by proper delays. The butterfly and the multiplier are idle half the time to wait for the new inputs. Hence the utilization of the butterfly and the multiplier is 50%.

F.Radix-2 Single-Path Delay Feedback

George A. Works introduced a feedback mechanism in order to minimize the number of delay elements. In the proposed architecture one half of outputs from each stage are fed back to the input data buffer when the input data are directly sent to the butterfly. This architecture is called Radix-2 Single-path delay Feedback (R2SDF). The number of multiplier and butterflies remains the same as R2MDC, but with much reduced memory requirement.

G. Radix-4 single-path delay feedback

A radix-4 version of R2SDF. Since we use the radix-4 algorithm we can reduce the number of multipliers to $\log_4(N) - 1$ compared to $\log_2(N) - 2$ for R2SDF. But the utilization of the butterflies are reduced to 25%. The radix-4 SDF butterflies also become more complicated than the radix-2 SDF butterflies.

H. Radix-4 Multipath Delay Commutator

This architecture is similar to R2MDC. Input data are separated by a 4-to-1 multiplexer and $3N/2$ delay elements at the first stage. A 4-path delay commutator is used between two stages. Computation is taking place only when the last 1/4 part of data is multiplexed to butterfly. The utilization of the butterflies and the multipliers are 25%.R4MDC FFT required in total multipliers $3(\log_4(N)-1)$ for an N point FFT which is more than the R2MDC or R2SDF. Moreover the memory requirement is 5N/2-4, which is the largest among the three discussed architectures. From the view of hardware and utilization, it is not a good structure.

I.Radix-4 Single-Path Delay Commutator

G. Bi and E. V. Jones proposed a simplified radix-4 butterfly to increase its utilization. In the simplified radix-4 butterfly, only one output is produced in comparison with 4 in the conventional butterfly. To provide the same four outputs, the butterfly works four times instead of just one. Due to this modification the butterfly has a utilization of 100%. To accommodate this change we must provide the same four data at four different times to the butterfly. A few more delay elements are required with this architecture. Furthermore, the simplified butterfly needs additional control signals, and so do the commutators.

The numbers of multipliers are less than the R4MDC FFT architecture.

J.Radix 2^2 SDF

The New efficient pipeline FFT architecture is R2^2SDF architecture. The hardware requirement of R2^2SDF has the minimum requirement for both multiplier and the storage .It uses the registers more efficiently to implement FFT function and the circuit complexity is reduced. It is constructed by two radix-2 based BFs and the coefficient multipliers. This makes it an ideal architecture for VLSI implementation of pipeline FFT processors.

The computation complexity and memory requirements for the above discussed pipeline architectures are shown in Table III.

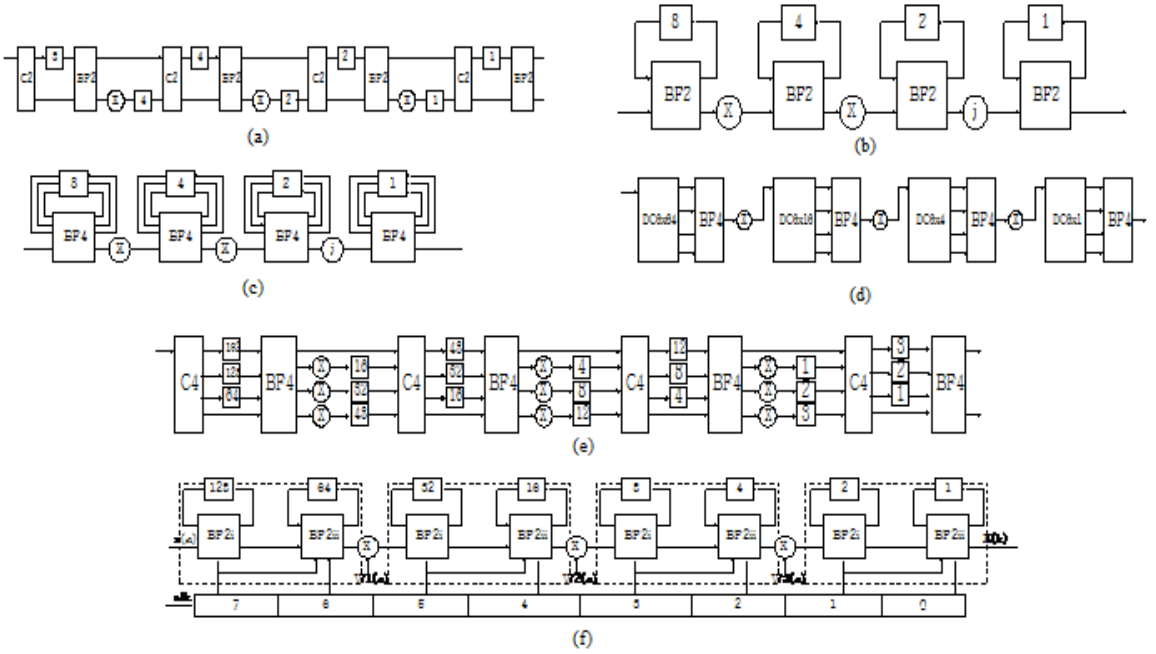

Fig 2. (a)Radix-2 multipath delay commutator (N=16) (b)Radix-2 single path delay feedback (N=16) (c)Radix-4 single path delay feedback (N=256) (d)Radix-4 single path delay commutator (N=256) (e)Radix-4 multipath delay commutator (N=256) (f)Radix-2^2 single path delay feedback (N=256)

TABLE III.
Pipeline architecture hardware comparison [5]

Pipeline Architecture	Complex Multiplier. #	Complex Adders #	Complex Memory. #
R2MDC	$\log_2 N-2$	$2\log_2 N$	1.5N-2
R2SDF	$\log_2 N-2$	$2\log_2 N$	N-1
R4SDF	$\log_4 N-1$	$8\log_4 N$	N-1
R4MDC	$3(\log_4 N-1)$	$8\log_4 N$	5/2N-4
R4SDC	$\log_4 N-1$	$3\log_4 N$	N-1
$R2^2SDF$	$\log_4 N-1$	$4\log_4 N$	N-1

IV. RECONFIGURABLE FFT

Wireless communications have been evolving in a fast way. This evaluation led to exist a verity of standards like UMTS(3G),IEEE 802.11G(WLAN) and DVB.H. Extracting functional and computational similarities between multiple standard is a must to define a low cost telecommunication system. Reconfigurable Hardware can act as a general hardware solution for implementing a variety of different computation within or across application domain which require performing FFT of length from 64 to 8192. Implementing each FFT on dedicated IP is an overhead in silicon area of the chip. The different approach is used to design variable length FFT processor. Some of the approach is memory based, pipeline based and general purpose DSP.Memory based is most area efficient, but it needs many computation cycles. Pipeline based has high throughput and less power but hardware efficiency complexity is more. The third approach general purpose DSP is the flexible one for various lengths FFT but it is neither power efficient nor area efficient. In this section we reviewed the pipeline and memory based variable length FFT architecture.

A. Pipeline Based FFT

Variable length FFT uses mixed radix algorithm. Radix 2^n SDF pipeline architecture is one of the recent efficient architecture. The reason to reduce the computation complexity and selecting different size FFT ,radix 2^4-2^4-2^4 is adopted. Each radix 2^4 has three stages that consist of radix 2 butterfly. First and second radix 2^4 stages are configurable as variable point FFT bypassing higher stage. Radix 2^4 has smallest memory size, least

complex multiplier and adder. The block diagram of variable length FFT[3] is shown in Fig 3. It consists of butterfly unit, delay buffer and complex multiplier.

It uses three stages and each stage use four butterfly unit for designing 16 to 4096 point FFT. First two stages are reconfigurable as radix 2, $2^{3,}$ and 2^4 .For an example to implement 512 point FFT it requires stage 2 as radix 2^2 and stage 3 as radix 2^4

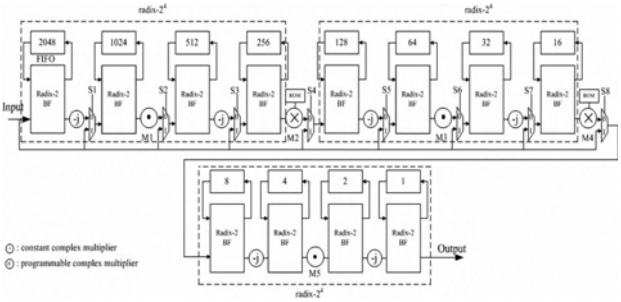

Fig 3.A Low-power Variable-length FFT Processor Base on Radix-2^4 Algorithm

B. Memory based FFT

Shared memory architecture has area efficiency and hardware simplicity. It requires only one butterfly unit as a processing element. To improve the throughput higher radix algorithm and pipelined structure butterfly units are used. The block diagram for shared memory variable length FFT is shown in Fig 4.The components are butterfly unit, memory, two commutators, address generator for data and twiddle factor, and control unit.

C. Memory

The memory consists of eight banks. It is dual port memory to improve the efficiency.

D. Butterfly unit

It Performs radix8/4/2 butterfly operation because Radix 8 algorithm can compute point 8^n only. To improve the flexibility of performing different points FFT, butterfly unit performs different radix.

Butterfly unit has pipeline structure with four stages. The first, second and third pipeline stage perform radix-2 computation and in the last pipeline stage multiplications by twiddle factor are computed. Various radix operations can be performed by the selection of control signal of MUXs.

E.Address generator

The various address generation techniques can be used to generate address for each size of FFT process .Improved address generation can make the complexity less. The portion of address is changed by a switch and barrel shifter as shown in Fig 5. The switch operates differently based on radix-r.

F. Commutator

Commutator are barrel shifter which consist of eight 8x1 MUXs. First commutator assign input data to butterfly unit and the second commutator assign output data to connect memory location. The operation of two commutator changes according to each stage and FFT length.

G.Twiddle factor generator

Twiddle factor generator may be based on ROM or based on recursive feedback difference equation for the computation of sine and cosine function as shown below.

$$Sin\ n\theta = 2cos n\theta\ sin\ (n-1)\ \theta - sin\ (n-2)\ \theta$$

$$Cos\ n\theta = 2cos n\theta cos\ (n-1)\ \theta - cos\ (n-2)\ \theta \quad (3)$$

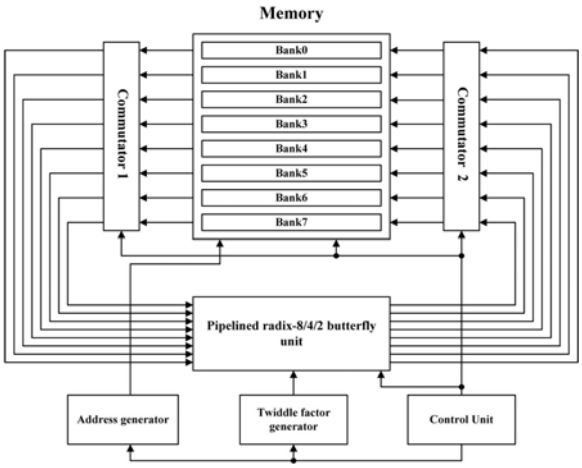

Fig4. Shared Memory variable point FFT processor

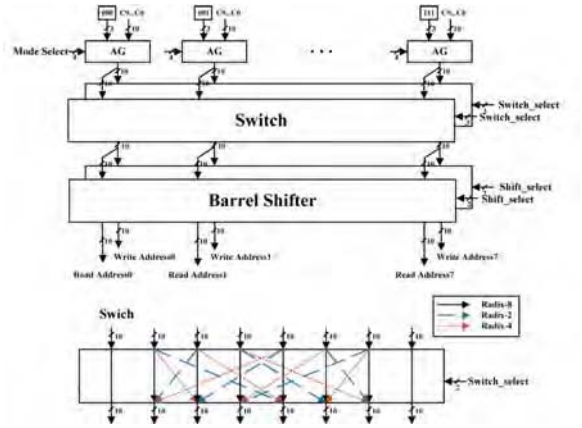

Fig 5 Address generator

V. LITERATURE SURVEY

The development of some of the Reconfigurable FFT processor is surveyed in this section. The author Y.-T. Lin et.al.[6] proposed a variable length FFT. To reduce power consumption and chip area, special current mode SRAM was adopted to replace shift register in the delay line, also complex multiplication was used which contains three real multiplication reduced sine, cosine tables. Radix 2/4/8 algorithm was adopted for variable length FFT with least hardware complexity although the adder and multiplier utilization is not as good as other architecture. The author Chi-ehen Lai et.al [7] proposed a novel Reconfigurable mixed radix FFT processor. The Variable length FFT from 16 to 4096 point is designed by proper selection of mixed radix algorithm. Pipeline based architecture achieves reconfigurable length by passing certain preceding stages to calculate FFT.Drawback of this is long data path design may occur if the processor with high flexibility is to be made. To overcome this, the author proposed reconfigurable butterfly architecture to achieve the best flexibility while keeping the datapath short. Also block float point (BFP) approach is used for better SNR(Signal to noise ratio).In the paper[8] Reconfigurability is obtained by mixed radix 2-4-4-8-8 pipeline structure for 64 to 2046 point FFT. The CORDIC algorithm is adopted to optimize power in complex multiplication. The author Hao Xiao et.al proposed a low cost reconfigurable FFT[9] using novel dual path pipelined shared memory architecture .The elaborate memory configuration scheme and mixed butterfly unit was designed to improve throughput and reduced area than the memory based architecture.

In the paper[10],the author discussed various reconfiguration levels such as function, operator and datapath level. He concluded that the reconfiguration at the datapath level presents a trade off in power consumption between the operator and the function levels. The datapath level reconfigurable unit is less costly in term of silicon area between the three approaches. The author Qingwang Lu et.al. [3]proposed variable length FFT 16 to 4896 . High radix and mixed radix are chosen to reduce computation complexity and reconfigurable flexibility. He Efficiently used mixed radix and radix 2^4 pipelined SDF. The author [11] designed a runtime reconfigurable FFT for 64,128 and 256 point architecture. The concept of mixed radix $2^2/2^3/2^4$ is used.In a modified path SDF uses 13 butterfly only and dataflow reconfiguration technique is adopted. Partial reconfiguration addresses the reduced reconfiguration overhead .The author Chia-Hsiang Yang et.al. has proposed 128 to 2048 point FFT [12] processor by using radix factorization method. The substantial power

reduction is obtained by the use of constant multiplier and efficient delay buffer. Further reduction of power may be possible with circuit level customization.

VI. CONCLUSION

In this survey paper we have focused on various algorithms and architecture of FFT processor design and the significance of reconfigurable FFT processor design in particular OFDM communication system. One of the major computations is multiplication in FFT. The aim of the research is to reduce the number of multiplication operation, increasing the efficiency in design of twiddle factor multiplier, to choose the algorithm and architecture which efficiently use the processing element and to improve the design of delay buffer in pipelining architecture. The objective is to improve performance of FFT by selecting architecture which uses pipeline and parallel method.

REFERENCES

[1] Bingham, J.A.C, "Multicarrier modulation for data transmission: an idea whose time has come", IEEE Commun. Mag., 1990, 28, (7), pp. 5–14

[2] Cimini, L.J., "Analysis and simulation of a digital mobile channel using orthogonal frequency division multiplexing", IEEE Trans. Commun., 1985, 33, (7), pp. 665–675

[3] Qingwang Lu, Xin'an Wang and JiuChong Niu , "A Low-power Variable-length FFT Processor Base on Radix-2^4 Algorithm", IEEE conference publication 2009,p 129-132.

[4] Weidong Li, "Studies on Implementation of Low Power FFT Processors", Thesis No. 1030, Department of Electrical Engineering, Linköpings university, Sweden,june 2003

[5] S. He and M. Torkelson, "A new approach to pipeline FFT processor," *10th* International Parallel Processing Symposium (IPPS '96), Apr. 1996, pp. 766-770.

[6] Y.-T. Lin, P.-Y. Tsai and T.-D. Chiueh "Low-power variable-length fast Fourier transform processor", IEE Proc.-Comput. Digit. Tech., Vol. 152,No. 4, July 2005.

[7] Chi-ehen Lai and Wei Hwang, "Low-Power Reconfigurable Mixed-Radix FFT/IFFT Processor", IEEE Asia Pacific Conference on Circuits and Systems, pp. 1931-1934, Dec 2006.

[8] Guihua Liu1,Quanyuan Feng, Institute of Microelectronic, "ASIC Design of Low-power Reconfigurable FFT Processor" ,IEEE pp 44-46 ,2007.

[9] Hao Xiao, An Pan, Yun Chen, and Xiaoyang Zeng, "Low-Cost Reconfigurable VLSI Architecture for Fast Fourier Transform",IEEE Transactions on Consumer Electronics, Vol. 54, No. 4, NOVEMBER 2008.

[10] Maroun Ojail, Raphael David, Stephane Chevobbe, DidieDemigny."Reconfiguration Levels analysis of FFT/IFFT units in wireless communication systems", 12th Euromicro Conference on Digital System Design / Architectures, Methods and Tools,2009.

[11] C.Vennila. G.Lakshminarayanan,Seok-Bum Ko."Dynamic Partial Reconfigurable FFT for OFDM based Communication Systems",Circuit System Signal process, p 1-18, October 2011.

[12] Chia-Hsiang Yang,Tsung-Han Yu,Dejan Markovic, "Power and Area Minimization of Reconfigurable FFT Processors:3GPP-LTE Example",IEEE journal of solid circuits,vol 47,no 3 march 2012.